Box Codes

and

Convolutional Coding of Block Codes

Thesis by

Yonggang Jin

In Partial Fulfillment of the Requirements

for the Degree of

Doctor of Philosophy

California Institute of Technology

Pasadena, California

1995

(Submitted May 8, 1995)

Acknowledgements

My heartfelt thanks go to my advisors, Professor Richard M. Wilson and Dr. Gustave Solomon, for helping me choose the direction of my research, for teaching me tools and techniques required to tackle the problems, and for their great enthusiasm. This thesis, as well as my entire graduate experience, has benefited greatly from my advisors' generosity and constant encouragement.

This work would not have been possible without the numerous help from Professor Robert J. McEliece. Professor McEliece first introduced coding theory to me. He has been an invaluable source of ideas, insights and knowledge. I can hardly thank him enough for his help.

I would like to thank Dr. Laif Swanson and the Communications Systems Research Section of the Jet Propulsion Laboratory for giving me the wonderful chance to work with Dr. Solomon and other JPL researchers, and for the generous financial support.

I am grateful to Professor David B. Wales, for his patience and valuable time to familiarize himself with my research, and for his many helpful comments and suggestions.

My years at Caltech were enriched by all the wonderful people I met here. I also thank the mathematics department for its financial aid.

Abstract

PART I

A self-dual code of length 48, dimension 24, with Hamming distance essentially equal to 12 is constructed. There are only six codewords of weight 8. All the other codewords have weights that are multiples of 4 and have minimum weight equal to 12.

A (72, 36; 15) box code was constructed from a (63, 35; 8) cyclic code. The theoretical justification is presented herein.

A second (72, 36; 15) code is constructed from an inner (63, 27; 16) Bose-Chaudhuri-Hocquenghem (BCH) code and expanded to length 72 using the box code algorithm for extension. This code was simulated and verified to have a minimum distance of 15 with even weight words congruent to 0 modulo 4. The decoding for hard and soft decision is still more complex than the first code constructed above.

Finally, an (8, 4; 5) Reed-Solomon code over GF(512) in the binary representation of the (72, 36; 15) box code gives rise to a (72, 36; 16*) code, where the "16*" means that there are nine codewords of weight 8 and all the rest have weights ≥ 16.

PART II

In order to get self-dual block codes by the convolutional encoding technique developed in [18], Solomon [12] gave sufficient conditions for code length $2n + 2$ and tap polynomials $p(x)$ and $q(x)$. We present necessary and sufficient conditions for convolutional encoding of self-dual block codes of rate 1/2 with weights w, $w \equiv 0 \pmod 4$. In addition [15], we searched for the smallest possible convolutional encoding constraint lengths K for (80, 40; 16) self-dual codes (quadratic residue and non-quadratic residue) and even for (104, 52; 20) quadratic residue code.

Contents

Chapter 1 Introduction

1.1 Codes

An (n, k) *linear code* $\mathbf{C}$ of *length* n is defined as a k-dimensional subspace of $V_n(K)$, the n-dimensional vector space over a finite field K.

Let $\mathbf{x} \in V_n(K)$, $\mathbf{y} \in V_n(K)$, then the *Hamming distance* $d(\mathbf{x}, \mathbf{y})$ of $\mathbf{x}$ and $\mathbf{y}$ is defined by

$$d(\mathbf{x}, \mathbf{y}) := |\{i | 1 \leq i \leq n, x_i \neq y_i\}|.$$

The *weight* $w(\mathbf{x})$ of $\mathbf{x}$ is defined by

$$w(\mathbf{x}) := d(\mathbf{x}, \mathbf{0}).$$

The *minimum distance* of a nontrivial code $\mathbf{C}$ is

$$\min\{d(\mathbf{x}, \mathbf{y}) | \mathbf{x} \in \mathbf{C}, \mathbf{y} \in \mathbf{C}, \mathbf{x} \neq \mathbf{y}\}.$$

The *minimum weight* of $\mathbf{C}$ is

$$\min\{w(\mathbf{x}) | \mathbf{x} \in \mathbf{C}, \mathbf{x} \neq \mathbf{0}\}.$$

Note: If $\mathbf{C}$ is linear then the minimum distance is the same as the minimum weight. And we call $\mathbf{C}$ an $(n, k; d)$ code if its minimum distance is d.

In most applications, the field K is taken to be GF(2), the field with two elements 0 and 1. A code $\mathbf{C}$ is called *binary* if it is defined over GF(2).

1.2 Cyclic Codes and Mattson-Solomon Polynomials

A linear code $\mathbf{C}$ is called *cyclic* if for each *codeword* $\mathbf{x} = (x_0, x_1, \ldots, x_{n-1})$ in $\mathbf{C}$, the vector $(x_1, x_2, \ldots, x_{n-1}, x_0)$ is also in $\mathbf{C}$.

Remark. For other definitions of cyclic codes, please see [11], [8], [3] and [6].

Let $\mathbf{C}$ be an (n, k) binary cyclic code. Then there exists a binary polynomial

$$f(x) = \sum_{i=0}^{k} u_i x^i$$

which divides x^n+1 over GF(2) with the property that any codeword $\mathbf{a} = (a_0, a_1, \ldots, a_{n-1}) \in \mathbf{C}$ can be generated by the recursion [11]

$$\sum_{i=0}^{k} u_i a_{t+i} = 0.$$

Remark: The polynomial $f(x)$ is called the *parity-check polynomial* of the code and $g(x) = (x^n + 1)/f(x)$ is call the *generator polynomial* of the code.

Let β be a fixed primitive n-th root of unity. Then with every codeword $\mathbf{a}$ in $\mathbf{C}$, there is a polynomial called the *Mattson-Solomon polynomial* [7]

$$g_{\mathbf{a}}(z) = \sum_{i=0}^{n-1} c_i z^i,$$

with the following properties:

(a) The c_j are given by the Reed formula

$$c_j = \sum_{i=0}^{n-1} a_i \beta^{-ij}, \quad j = 0, 1, \ldots, n-1;$$

especially, $c_0 = \sum_{i=0}^{n-1} a_i \equiv w(\mathbf{a}) \pmod 2$ and $c_{2j} = c_j^2$ for any j.

(b) If $\mathbf{a} = (a_0, a_1, \ldots, a_{n-1})$ then $g_{\mathbf{a}}(\beta^j) = a_j$.

(c) $c_j = 0$ whenever $f(\beta^j) \neq 0$.

That is, the only c_j that can enter in the expression for $g_{\mathbf{a}}(x)$ are those c_j such that $f(\beta^j) = 0$.

1.3 Solomon-McEliece Γ_2 Formula and an Example of a Box Code

1.3.1 Solomon-McEliece Γ_2 Formula

Let $\mathbf{a} = (a_0, a_1, \ldots, a_{n-1})$ be a binary n-tuple. If w is the weight of $\mathbf{a}$, then

$$\Gamma_2(\mathbf{a}) = \sum_{i<j} a_i a_j \equiv \binom{w}{2} \pmod 2.$$

Note that $\Gamma_2 \in \mathrm{GF}(2)$.

Solomon and McEliece [17] prove that

$$\Gamma_2(\mathbf{a}) = \sum_{i=1}^{(n-1)/2} c_i c_{n-i} \quad \text{if } w(\mathbf{a}) \text{ is even,}$$

where c_j are the coefficients of the Mattson-Solomon polynomial, and

$$\Gamma_2(\mathbf{a}) = 0 \quad \text{iff} \quad w(\mathbf{a}) \equiv 0 \pmod 4$$

for any even weight cyclic codeword $\mathbf{a}$.

When $n = 7$, $\mathbf{a} \in \mathbf{C}$, we have the following straightforward properties:

(a) $\Gamma_2(\mathbf{a}) = c_1c_6 + c_2c_5 + c_3c_4 = \mathrm{Tr}(c_1c_6)$, where $\mathrm{Tr}(b) = b + b^2 + b^4 \in \mathrm{GF}(2)$;

(b) For any codeword $\mathbf{a}'$ in the extended code of $\mathbf{C}$,

$$\mathrm{Tr}(c_1 c_6) = 0 \quad \text{iff} \quad w(\mathbf{a}') \equiv 0 \pmod 4;$$

(c) $w(\mathbf{a}) = 0$ or 7 if and only if $c_1 = c_6 = 0$;

(d) $w(\mathbf{a}) = 1$ or 6 if and only if $c_1 c_6 = 1$;

(e) If $w(\mathbf{a}) = 5$, $c_1 c_6 \notin \mathrm{GF}(2)$.

1.3.2 Box Codes

Let us consider any linear code $\mathbf{C}$ of length n and any permutation $\sigma \in \mathrm{Sym}(n)$, we can arrange the codeword $\mathbf{a} = (a_0, a_1, \ldots, a_{n-1})$ as a $t \times s$ matrix M with respect to σ, where $n = ts$ as follows:

$$\mathrm{M} = \begin{pmatrix} a_{\sigma(0)} & a_{\sigma(1)} & \cdots & a_{\sigma(s-1)} \\ a_{\sigma(s)} & a_{\sigma(s+1)} & \cdots & a_{\sigma(2s-1)} \\ \vdots & \vdots & \ddots & \vdots \\ a_{\sigma((t-1)s)} & a_{\sigma((t-1)s+1)} & \cdots & a_{\sigma((t-1)s+(s-1))} \end{pmatrix}.$$

Adding t overall parity-check bits b_i; $i = 0, 1, \ldots, t-1$ for each t rows, we get a $t \times (s+1)$ matrix

$$\mathrm{M}' = \begin{pmatrix} a_{\sigma(0)} & a_{\sigma(1)} & \cdots & a_{\sigma(s-1)} & b_0 \\ a_{\sigma(s)} & a_{\sigma(s+1)} & \cdots & a_{\sigma(2s-1)} & b_1 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ a_{\sigma((t-1)s)} & a_{\sigma((t-1)s+1)} & \cdots & a_{\sigma((t-1)s+(s-1))} & b_{t-1} \end{pmatrix},$$

where

$$b_i = \sum_{j=0}^{s-1} a_{\sigma(is+j)}; \quad i = 0, 1, \ldots, t-1.$$

Then the set **C'** of all new vectors

$$\mathbf{a}' = (a_{\sigma(0)}, a_{\sigma(1)}, \ldots, a_{\sigma(s-1)}, b_0, \ldots, a_{\sigma((t-1)s)}, a_{\sigma((t-1)s+1)}, \ldots, a_{\sigma((t-1)s+(s-1))}, b_{t-1})$$

is called the $t \times (s+1)$ *box code* of **C** with respect to σ.

Note that the box code **C'** is a linear code of length $n + t$.

The concept of box codes was first introduced by G. Solomon in the late 1980's when he was visiting the electrical engineering department of the California Institute of Technology. A box code can be regarded as a generalization of an extended code. Many good codes are box codes generated from other codes. A good example [17] is that the extended (24, 12; 8) Golay code is a 3×8 box code generated from the (8, 4; 5) Reed-Solomon code over the field GF(8) of eight elements with respect to certain basis for GF(8) over GF(2). We here present the theoretical proof of the example.

Example: (see [17])

Consider the (7, 4; 4) Reed-Solomon code over GF(8) given by the parity-check polynomial

$$f(x) = \prod_{i=0}^{3}(x + \beta^i),$$

where β be a root of $g(x) = x^3 + x^2 + 1$.

It is easy to see that β is a primitive 7th root of unity and that $\{\beta, \beta^2, \beta^4\}$ is a self-complementary normal basis for GF(8) over GF(2).

Then the MS polynomial for any codeword **a** is

$$g_{\mathbf{a}}(x) = c_0 + c_1 x + c_2 x^2 + c_3 x^3,$$

where

$$\mathbf{a} \;=\; (a_0, a_1, a_2, a_3, a_4, a_5, a_6) \in GF(8)^7$$

$$x \in GF(8) = \{0, 1, \beta, \beta^2, \beta^3, \beta^4, \beta^5, \beta^6\};$$
$$c_i \in GF(8), \quad 0 \leq i \leq 3.$$

In binary representation by using the normal basis above, we obtain three 7-tuple vectors:

$\mathbf{a} =$	a_0	a_1	a_2	a_3	a_4	a_5	a_6	□
$\mathrm{Tr}(g_{\mathbf{a}}(x)\beta)$	$a_0^{(0)}$	$a_1^{(0)}$	$a_2^{(0)}$	$a_3^{(0)}$	$a_4^{(0)}$	$a_5^{(0)}$	$a_6^{(0)}$	□
$\mathrm{Tr}(g_{\mathbf{a}}(x)\beta^2)$	$a_0^{(1)}$	$a_1^{(1)}$	$a_2^{(1)}$	$a_3^{(1)}$	$a_4^{(1)}$	$a_5^{(1)}$	$a_6^{(1)}$	□
$\mathrm{Tr}(g_{\mathbf{a}}(x)\beta^4)$	$a_0^{(2)}$	$a_1^{(2)}$	$a_2^{(2)}$	$a_3^{(2)}$	$a_4^{(2)}$	$a_5^{(2)}$	$a_6^{(2)}$	□

Consider the extended (8, 4; 5) Reed-Solomon code; we obtain a 3×8 box code (see the table above), where □ represents the overall parity symbol.

$$\mathrm{Tr}(g_{\mathbf{a}}(x)\beta^j) = \mathrm{Tr}[(c_1\beta^j + (c_2\beta^j)^4)x + (c_3\beta^j)^2 x^6 + c_0\beta^j],$$
$$\Gamma_2[\mathrm{Tr}(g_{\mathbf{a}}(x)\beta^j)] = \mathrm{Tr}(c_1 c_3^2 \beta^{3j} + c_2^4 c_3^2 \beta^{6j}).$$

So

$$\sum_{j=1,2,4} \Gamma_2[\mathrm{Tr}(g_{\mathbf{a}}(x)\beta^j)] = 0 \quad \text{by} \quad \mathrm{Tr}(\beta^3) = \mathrm{Tr}(\beta^6) = 0.$$

By the Solomon-McEliece Γ_2 formula, the weight of each new codeword is multiple of 4. And the (8, 4; 5) Reed-Solomon code over GF(8) has minimun distance 5. Then the box code has minimum distance 8, and therefore, it is the (24, 12; 8) Golay code.

Moreover we studied the properties of the box codes generated for the (8, 4; 5) extended Reed-Solomon codes over GF(64) and GF(512).

In 1993, we [14] constructed a (72, 35; 16) box code from the (63, 35; 8) cyclic code by using the Mattson-Solomon polynomials and the Solomon-McEliece Γ_2 Formula

[17]. A similar method yields a (72, 36; 15) box code with the (72, 35; 16) code as a subcode. This work is related to the Research Problem (19.3) of MacWilliams and Sloane [6], which asks whether there is a (72, 36; 16) even self-dual code.

1.4 Convolutional Coding of Block Codes

Convolutional codes have also been used successfully in practice despite the lack of a deep mathematics theory. In 1979, Solomon and van Tilborg [18] developed a connection between quadratic residue codes, quasi-cyclic codes and convolutional codes. For example [18], the (24, 12; 8) Golay code can be encoded convolutionally by two polynomials $p(x) = x^3 + x^2 + 1$ and $q(x) = x^3 + x + 1$.

Let us consider the $(24, 12; 8)$ Golay code and two polynomials $p(x) = x^3 + x^2 + 1$ and $q(x) = x^3 + x + 1$. Let $i_0, i_1, \ldots, i_{10}, i_{11}$ be the 12 information bits.

In vector and matrix notation, the codeword of the information bits $i_0, i_1, \ldots, i_{10}, i_{11}$ is the vector

$$(i_0, i_1, \ldots, i_{10}, i_{11})G(P|Q),$$

where

$$G(P|Q) = \begin{pmatrix} & & & 1 & & & & 1 \\ & P & & 1 & & Q & & 1 \\ & & & \vdots & & & & \vdots \\ & & & 1 & & & & 1 \\ 1 & 1 \cdots & 1 & 1 & 0 & 0 \cdots & 0 & 0 \end{pmatrix}.$$

P is the associated circulant of order 11 of $p(x)$, and Q is the associated circulant of order 11 of $q(x)$.

Note that for any polynomial $f(x) = \sum_{i=0}^{m-1}$ of degree $m-1$, the associated circulant F of order n $(n \geq m)$ is a square $n \times n$ matrix with the top row $(f_0, f_1, \ldots, f_{m-1}, 0, \ldots, 0)$

and every other row the right cyclic permutation of its preceding row.

Solomon and van Tilborg [18] give the convolutional encoding of the Golay code by the two polynomials $p(x)$ and $q(x)$.

Remark: Such polynomials $p(x)$ and $q(x)$ are call *taps* or *tap polynomials* of the convolutional coding. And we call this code can be convolutionally encoded by the polynomial matrix $[p(x) \;\; q(x)]$.

Arrange the 12 information bits $i_0, i_1, \ldots, i_{10}, i_{11}$ in special way (See Figure 1.1).

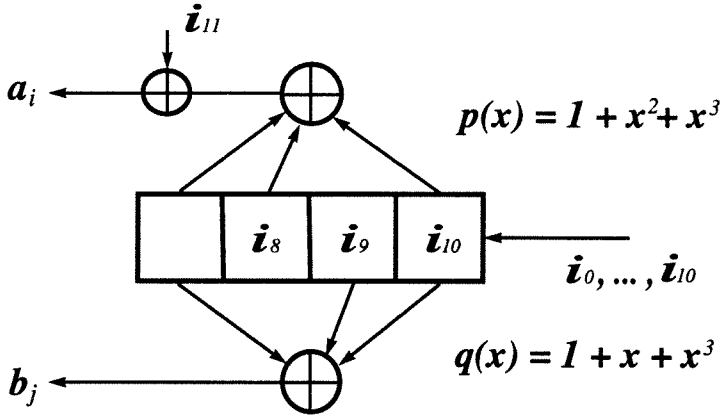

Figure 1.1: Binary Convolutional Encoder for the (24, 12; 8) Golay Code.

Output sequences

$$a_0, a_1, \ldots, a_{10}, a_\infty \quad \text{with} \quad a_\infty = \sum_{i=0}^{10} a_i;$$

$$b_0, b_1, \ldots, b_{10}, b_\infty \quad \text{with} \quad b_\infty = \sum_{i=0}^{10} b_i.$$

Codeword is

$$(a_0, a_1, \ldots, a_{10}, a_\infty, b_0, b_1, \ldots, b_{10}, b_\infty).$$

Recently Solomon [12] gave sufficient conditions for code length $2n + 2$ and two polynomials $p(x)$ and $q(x)$ in order to convolutionally encode even self-dual block

codes. The conditions are $n \equiv 3 \pmod 4$, $\gcd(p(x), x^n + 1) = 1$, and $q(x) = \tilde{p}(x)$, where $\tilde{p}(x)$ is the reciprocal polynomial of $p(x)$. We eventually found the necessary and sufficient conditions [16]. In addition, the previously found constraint length $K = 9$ [18] for the (48, 24; 12) quadratic residue code was lowered to $K = 8$ [12] by Solomon. We have found in our search for the smallest possible constraint lengths for (80, 40; 16) even self-dual codes (quadratic residue and non-quadratic residue, respectively), the constraint lengths $K = 14$, and $K = 13$; and we found $K = 21$ for the (104, 52; 20) quadratic residue code [15]. The smaller the K, the less complex the sequential or Viterbi decoder.

Moreover, we generalized this idea to convolutional encoding of some cyclic codes of rate $\frac{t}{t+1}$. For example, the (33, 22; 6) cyclic code with generator polynomial $f(x) = x^{11} + x^9 + x^6 + x^5 + x^2 + 1$ and the (65, 52; 6) cyclic code with generator polynomial $f(x) = x^{13} + x^{12} + x^9 + x^4 + x + 1$ can be encoded convolutionally.

Chapter 2 Box Codes

2.1 Codes of Lengths 48 and 72

The self-dual (48, 24; 12) Quadratic Residue Code has had a history of difficulty and complexity in decoding for 5 errors algebraically as well as decoding for soft decision. (For the concepts of algebraic decoding (hard decision) and decoding for soft decision, please see [8].) This led us to apply the techniques of box codes as successfully developed for Golay Codes to rate 1/2 codes of length 48. See [13], [10]. Subcodes of dimension 23 and Hamming distance 12 were easily found. In addition the box structure gave parity information to detect odd errors in rows which simplify decoding procedures.

Remark: For the concepts of algebraic decoding (hard decision) and decoding for soft decision, please see [8].

The attempt to avoid the six codewords of weight 8 in the natural box code construction yielded two self-dual (48, 24; 12) codes [13]. Upon closer examination of computer simulation, these codes contained 42 words of weight 8.

In [13], two codes constructed were designed to be self-dual. The (48, 23; 12) systematic subcodes of each were easily found. The 24th coordinate in each was more elaborately constructed with the proviso that odd parities of the rows were induced to be used as tools in an erasure-error correcting decoding procedure. A search of the codeword weights' structure indicated the presence of 42 words of weight 8 and 40 in both these codes. The remaining non-zero words were of minimum weight 12. There exists a straight systematic construction of the Reed-Solomon (8, 4; 5) code over GF(64) for the 24th dimension given below, still using the particular binary representation in [13], which yields only 6 codewords of weight 8 and 40. This gives a box code with even parity on the rows. So for a low signal-to-noise ratio, this

code and the previously constructed codes of dimension 48, rate 1/2 are effectively of minimum distance 12. The decoding procedure for soft decision mentioned in [13] is still applicable and preferred over any current soft decoding of the (48, 24; 12) Quadratic Residue code.

In [10], a code of length 72 and distance 15 was constructed specifically designed to have simplified soft decoding. Again the (72, 35; 16) subcode was constructed with even parity on the nine rows in a non-systematic manner as a subcode of the Reed-Solomon (8, 4; 5) code over GF(512). The 36th dimension was constructed to give odd parity on the rows and yield a code of minimum distance 15. The full code was designed to have a systematic encoding. This code, however, upon investigation, was found to have a very small number of codewords of length 11.

To meet this emergency, a new (72, 36; 15) box code is constructed here with rows of even or odd parity, and so it possesses, perhaps, a simple hard decision 7-8 error correcting procedure. This code has been simulated and verified to have minimum distance of 15 and even weight words congruent to 0 modulo 4.

2.2 (8, 4; 5) Reed-Solomon Code over GF(64)

Representing the (8, 4; 5) Reed-Solomon code over GF(64) in binary using the particular normal basis in [13], one can generate a rate 1/2 self-dual code of length 48 and dimension 24 with weights that are multiples of 4.

This binary representation of the (8, 4; 5) Reed-Solomon code over GF(64) yields six (8, 7; 2) codewords whose decomposition via Mattson-Solomon into two cyclic code components and a constant component looks like (6, 4; 3), (6, 2; 5) codes over GF(8) and a (6, 6; 1) binary code respectively.

In particular, let γ be a root of the polynomial $f(x) = x^6 + x^5 + x^4 + x + 1$, where γ is a primitive generator of the 63 roots of unity. Represent the elements of

GF(64) in the normal representation using the roots of $f(x)$. The roots are γ^j; $j \in J$; $J = \{1, 2, 4, 8, 16, 32\}$.

NOTE: For this particular choice of $f(x)$, we have

$$\begin{aligned} \mathrm{Tr}(\gamma^j) &= 1; \quad j \in J; \quad J = \{1, 2, 4, 8, 16, 32\}, \\ \mathrm{Tr}(\gamma^i \gamma^k) &= 0; \quad i \neq k; \quad i, k \in J. \end{aligned}$$

And such a basis $\{\gamma^1, \gamma^2, \gamma^4, \gamma^8, \gamma^{16}, \gamma^{32}\}$ is called the *self-complementary normal basis* for GF(64) over GF(2). A useful theorem can be found in A. Lempel and M.J. weinberger [2]:

Theorem 2.2.1. GF(q^n) has a self-complementary normal basis over GF(q) if and only if n is odd or $n \equiv 2 \pmod 4$ and q is even. ∎

A. Encoding

Let β be a root of the polynomial $g(x) = x^3 + x^2 + 1$. β is an element of GF(8), a subfield of GF(64), and $\beta = \gamma^9$.

Now use the parity-check polynomial $h(x) = \prod_{i=0}^{3}(x + \beta^i)$ to generate an extended (8, 4; 5) Reed-Solomon code over GF(64). This means that the initial shift register contains four elements in GF(64) expressed as coefficients in the normal representation above. The cyclic portion of the code is of length 7, and the overall parity symbol is the eighth dimension. Representing the binary code as components $\mathrm{Tr}(P_{\mathbf{a}}(x)\gamma^j)$; $j = 1, 2, 4, 8, 16, 32$.

The general Mattson-Solomon polynomial of a codeword **a**, similar to the Golay codeword over GF(8) is $P_{\mathbf{a}}(x) = C_0 + C_1 x + C_2 x^2 + C_3 x^3$ where $C_i \in$ GF(64) for $0 \leq i \leq 3$ and $x \in$ GF(8).

Encode the codeword in its cyclic portion. The extended codeword **a** expressed in terms of the Mattson-Solomon polynomial is

$$\mathbf{a} = (P_{\mathbf{a}}(\beta^i); 0 \leq i \leq 6, P_{\mathbf{a}}(0)).$$

Writing the codewords in binary, using the normal basis γ^j, $j \in J$ above, there are six binary codewords of weight 8:

$$\mathrm{Tr}(P_{\mathbf{a}}(x)\gamma^j); \quad j = 1, 2, 4, 8, 16, 32,$$

where Tra denotes the value in GF(2) given by the Trace of an element $a \in$ GF(64):

$$\mathrm{Tr}(a) = a + a^2 + a^4 + a^8 + a^{16} + a^{32}.$$

Consider one of the six binary words in its Mattson-Solomon setting,

$$\begin{aligned} \mathrm{Tr}(P_{\mathbf{a}}(x)\gamma^j) &= \mathrm{Tr}((C_0 + C_1x + C_2x^2 + C_3x^3)\gamma^j) \\ &= \mathrm{Tr}(C_0\gamma^j) + \mathrm{Tr}'[(C_1x + C_2x^2 + C_3x^3)\gamma^j \\ &\quad + ((C_1x + C_2x^2 + C_3x^3)\gamma^j)^8], \\ \text{where} \quad \mathrm{Tr}'(a) &= a + a^2 + a^4; \quad a \in \mathrm{GF}(8). \end{aligned}$$

Set $C_0 = 0$ temporarily as this does not effect the arguments to follow.

$$\begin{aligned} \mathrm{Tr}(P_{\mathbf{a}}(x)\gamma^j) &= \mathrm{Tr}'[(C_1\gamma^j + (C_1\gamma^j)^8 + (C_2\gamma^j)^4 + (C_2\gamma^j)^{32})x \\ &\quad + ((C_3\gamma^j)^2 + (C_3\gamma^j)^{16})x^6]. \end{aligned}$$

For each codeword **a**, we have 6 coefficients of x (x^6) with respect to the 6 values of j. Consider all codewords of the Reed-Solomon code, we have a code of length 6 over GF(8). We call this code the *coefficient code* of x (x^6).

Lemma 2.2.2. The coefficient code of x is a (6, 4; 3) code over GF(8). The coefficient code of x^6 is a (6, 2; 5) code over GF(8). The code is indexed by the values of $\gamma^j; j \in J = \{1, 2, 4, 8, 16, 32\}$.

Proof. The set $\gamma^j; j \in J = \{1, 2, 4, 8, 16, 32\}$ is a linearly independent set and thus one can only take zero values one less than the number of terms in the coefficients.

For the coefficients of x, if there are at least 4 of them are zeros, then $C_1 = C_2 = 0$ which implies that all the coefficients are zeros. Hence the coefficient code of x is a (6, 4; 3) code.

Same argument applies to coefficient code of x^6. ∎

Theorem 2.2.3. The Reed-Solomon code determined by codewords with Mattson-Solomon polynomials $P_{\mathbf{a}}(x); \mathrm{Tr}(C_0) = 0$ form a (48, 23; 12) binary code with weight multiples of 4.

Proof. The multiple 4 property of the weights follows using the Solomon-McEliece Γ_2 Formula.

$$\begin{aligned}\mathrm{Tr}(P_{\mathbf{a}}(x)\gamma^j) &= \mathrm{Tr}'[(C_1\gamma^j + (C_1\gamma^j)^8 + (C_2\gamma^j)^4 + (C_2\gamma^j)^{32})x \\ &\quad + ((C_3\gamma^j)^2 + (C_3\gamma^j)^{16})x^6],\end{aligned}$$

where Tr is defined in GF(64) and Tr′ is defined in GF(8).

Now

$$\begin{aligned}\Gamma_2(\mathrm{Tr}P_{\mathbf{a}}(x)\gamma^j) &= \mathrm{Tr}'(C_1C_3^2\gamma^{3j} + C_1^8C_3^2\gamma^{10j} + C_1C_3^{16}\gamma^{17j} + C_1^8C_3^{16}\gamma^{24j} \\ &\quad + C_2^{32}C_3^2\gamma^{34j} + C_2^4C_3^2\gamma^{6j} + C_2^{32}C_3^{16}\gamma^{48j} + C_2^4C_3^{16}\gamma^{20j}) \\ &= \mathrm{Tr}(C_1C_3^2\gamma^{3j} + C_1^8C_3^2\gamma^{10j} + C_2^4C_3^2\gamma^{6j} + C_2^4C_3^{16}\gamma^{20j}),\end{aligned}$$

and therefore $\sum_{j\in J}\Gamma_2(\mathrm{Tr}P_{\mathbf{a}}(x)\gamma^j) = 0$. ∎

Recall that the normal basis was chosen so that $\mathrm{Tr}(\gamma^j) = 1; j \in J$, and $\mathrm{Tr}(\gamma^i\gamma^k) = 0;\ i \neq k;\ i, k \in J$.

It has been demonstrated that the binary weight of any codeword in the Reed-Solomon code above is a multiple of 4. Since the symbol distance of the code is greater than 5, we have narrowed the weights down to $8, 12, 16, 20, \ldots, 40$.

2.3 Structure of the Code

Using the same arguments in [13], one shows the minimum weight of the code for $\mathrm{Tr}(C_0) = 0$ is equal to 12. We proved that these six are the only codewords of weight 8. A counting argument on the weights would do the same. Since all words have weight multiples of 4, the code is self-dual.

Theorem 2.3.1. The 6×8 box code generated by extended (8, 4; 5) Reed-Solomon code over GF(64) with respect to the chosen normal basis γ^j; $j \in J = \{1, 2, 4, 8, 16, 32\}$ over GF(2) has exactly six codewords of weight 8.

Proof. A codeword of weight 8 occurs only if $\mathrm{Tr}(C_0) \neq 0$ and $C_1 = C_6 = 0$. By property (c) of Section 1.3.1, there must be one and only one row with all 1's and the rest five rows are all 0's. This happens when the original Reed-Solomon codeword is $(\gamma^j, \gamma^j, \gamma^j, \gamma^j, \gamma^j, \gamma^j, \gamma^j)$; for $j = 1, 2, 4, 8, 16, 32$. ∎

2.4 (72, 36; 15) Code

In [10], an alternate (72, 36; 15) box code was constructed from the (63, 35; 8) cyclic code, generated by the check polynomial $f(x) = \prod f_i(x); i = 1, 3, 5, 7, 9, 13, 21$ where $f_i(x)$ is a polynomial irreducible over GF(2) with β^i as a root where β is a primitive 63rd root of unity. We now present the theoretical justification.

Place the codewords in the usual 9×7 box code matrices corresponding to their values $7i + 9j \pmod{63}$ for $0 \leq i \leq 8$, $0 \leq j \leq 6$. Let $z = xy$ where $x^7 = 1, y^9 = 1, x = \beta^{9j}$, and $y = \beta^{7i}$. Indexing the rows by y, the Mattson-Solomon polynomial for each row y is

$$\begin{aligned} P_y(x) &= \mathrm{Tr}(C_1 z + C_3 z^3 + C_5 z^5 + C_7 z^7 + C_{13} z^{13}) \\ &\quad + C_9 z^9 + C_{18} z^{18} + C_{36} z^{36} + C_{21} z^{21} + C_{42} z^{42} \\ &= C_{21} y^3 + C_{21}^2 y^6 + \mathrm{Tr}(C_7 y^7) + \mathrm{Tr}'[(C_9^4 + C_1 y + C_1^8 y^8) x \\ &\quad + (C_5^4 y^2 + C_3^{16} y^3 + C_{13} y^4 + C_{13}^8 y^5 + C_3^2 y^6 + C_5^{32} y^7) x^6]. \end{aligned}$$

Where Tr is defined in GF(64) and Tr′ is defined in GF(8).

Thus the coefficient code of x is a (9, 3; 7) code over GF(8) and the coefficient code of x^6 is a (9, 6; 4) code over GF(8).

Construct an 8th column on the nine rows by the usual parity rule. The 8th column will have the same Γ_2 value as the original (63, 35; 8) code. Then we have the following lemma immediately:

Lemma 2.4.1. The extended box code is a (72, 35; 16) code.

Proof. Consider the Solomon-McEliece Formula. The sum Γ_2 over the nine rows gives 0, showing that the weight of every codeword is multiple of 4. And the properties of the coefficient codes of x and x^6 imply the minimum weight to be 16. ∎

Now adjoin a vector of all ones in the original 9×7 matrix setting. This will make the rows have odd parity and will complement the column sums. It is easy to show that all odd-weight codewords have weights of form $4m - 1$. We will prove that the minimum code distance is 15.

Examining the degree of the Mattson-Solomon polynomial and the properties of the coefficients of x and x^6, one can easily see that the weight of the inner cyclic codeword is less than or equal to 54 because the degree of the Mattson-Solomon polynomial is at most 56 and $x + 1$ is not a factor of the parity-check polynomial $f(x)$. If the inner weight is 54, the nine rows weight patterns 6 6 6 6 6 6 6 6 6, 7 6 6 6 6 6 6 6 5, and 7 7 6 6 6 6 6 6 4 could generate codewords of weights less than 15. If the inner weight is 52, the weight pattern 6 6 6 6 6 6 6 6 4 could generate codeword of weight less than 15.

Lemma 2.4.2. For the original cyclic (63, 35; 8) code, none of the weight patterns above are possible.

Proof. Weight pattern 6 6 6 6 6 6 6 6 6 gives the sum Γ_2 to be 1. This is impossible.

Let

$$\begin{aligned} P(y) &= C_9^4 + C_1 y + C_1^8 y^8, \\ Q(y) &= C_5^4 y^2 + C_3^{16} y^3 + C_{13} y^4 + C_{13}^8 y^5 + C_3^2 y^6 + C_5^{32} y^7. \end{aligned}$$

Then

$$\Sigma_y P(y)Q(y) = 0,$$
$$\deg\,(P^6(y) - Q(y)) \leq 7.$$

If the weight pattern is 7 6 6 6 6 6 6 6 5, then $\Sigma_y P(y)Q(y) = 1 + \alpha \neq 0$ for some $\alpha \neq 1$; $\alpha \in$ GF(8). See property (e) in Section 1.3.1.

If the weight patterns are 6 6 6 6 6 6 6 6 4 or 7 7 6 6 6 6 6 6 4, $P(y) = Q(y) = 0$ or $P(y)Q(y) = 1$ for eight values of y by properties (c) and (d) in Section 1.3.1. This means that the polynomial $P^6(y) - Q(y)$ has 8 zeros since $P(y), Q(y) \in$ GF(8). Then $P^6(y) = Q(y)$. But $P(y)Q(y) = 0$ and $P^6(y) = Q(y)$ do not give weight 4 for any row indexed by y. ∎

Theorem 2.4.3. The box code is a (72, 36; 15) code, where the even-weight subcode is a (72, 35; 16) with all codewords having weights of form $4m$, and the odd-weight subcode is a (72, 35; 15) code with all codewords having weights of form $4m - 1$. ∎

2.5 (72, 36; 15) Alternate Code

One can construct the cyclic (63, 27; 16) code generated by the check polynomial $f(x) = \prod f_i(x)$; $i = 1, 3, 5, 9, 11$. The cyclic decomposition in the box code setting yields a (9, 5; 5) code over GF(8) for the coefficient code of x and a (9, 4; 6) code over GF(8) for the coefficient code of x^6. This does extend to a (72, 36; 15) code, too. In fact, this code has been simulated and verified to have a minimum distance

of 15 with even weight codewords congruent to 0 modulo 4. If we try all possibilities for the check polynomial $g(x) = \prod f_i(x);\ i = 7, 21$, which totals to 256 codewords, we are left with an inner cyclic code that can algebraically correct 7 errors. This leaves soft decoding still very complex and unworkable.

Note that because of the Mattson-Solomon decomposition here, we may correct five errors easily by hard decision, but six and higher takes more trials. Similarly, a soft decision would require $8^4 = 2^{12}$ trials. This requires more trials for both hard and soft decision than the alternate code mentioned above [10]. Note the (9, 3; 7) code over GF(8) is the coefficient code of x^6 as opposed to the (9,4;6) code over GF(8) for the coefficient code of x.

2.6 (8, 4; 5) Reed-Solomon Code over GF(512)

The extended (8, 4; 5) Reed-Solomon code over GF(512) in the binary representation of [10] gives rise to a systematic (72, 36; 16*) code with nine words of weight 8 and all the rest have weights ≥ 16. The normal basis consists of $\gamma^i;\ i = 2^j;\ 0 \leq j \leq 8$ with γ a root of $f(x) = x^9 + x^8 + x^6 + x^5 + x^4 + x + 1$. The proof that there are no words of weight 12 is a simple counting argument. We prove there are no words of weight 60 in the code of dimension 35 given by $C_0 = 0$.

Represent the elements of GF(512) in the normal representation using the roots of $f(x)$. The roots are $\gamma^j;\ j \in J;\ J = \{1, 2, 4, 8, 16, 32, 64, 128, 256\}$.

For this particular choice of $f(x)$, we have

$$\begin{aligned} \mathrm{Tr}(\gamma^j) &= 1; \quad j \in J;\ \ J = \{1, 2, 4, 8, 16, 32, 64, 128, 256\}, \\ \mathrm{Tr}(\gamma^i \gamma^k) &= 0; \quad i \neq k;\ \ i, k \in J. \end{aligned}$$

Represent the binary code as components $\mathrm{Tr}(P(x)\gamma^i);\ i = 1, 2, 4, 8, 16, 32, 64, 128, 256$, giving nine words of length 8.

A. Encoding

Let β be a root of the polynomial $g(x) = x^3 + x^2 + 1$. β is an element of GF(8), a subfield of GF(512) and $\beta = \gamma^{73}$.

Now use parity-check polynomial $h(x) = \prod_{i=0}^{3}(x + \beta^i)$ to generate the extended (8, 4; 5) Reed-Solomon code over GF(512). This means that the initial shift register contains four elements in GF(512) expressed as coefficients in the normal representation above. The cyclic portion of the code is of length 7, the overall parity symbol; the eighth dimension is the usual sum over the seven symbols.

In the (72, 36) binary representation of the Reed-Solomon (8, 4; 5) code over GF(512), any codeword with the coeffiecients of x and x^6 non-zero, has minimum weight 16. When these are zero, then clearly there are only nine words of weight 8, which come from the encoding of the symbol γ^j; $j \in J$; $J = \{1, 2, 4, 8, 16, 32, 64, 128, 256\}$. A similar proof would argue that there are only six words of weight 8 in the binary representation of the (8, 4; 5) Reed-Solomon code over GF(64).

To prove there are no words of weight 12, a counting argument notes that there are no words of weight 60 in the even weight codes, where $\mathrm{Tr}C_0 = 0$. Words of weight 60 must possess a weight distribution over the nine words in any permutation of 8 8 8 6 6 6 6 6 6. This implies that three rows are zero and six rows are non-zero with weights 6 or $\Gamma_2 = 1$, to say the least. However, in [10], we note that this cannot be. The coefficient codes are (9, 3; 7) and (9, 6; 4) codes over GF(8), so there are at least 7 rows with $\Gamma_2 = 1$. ∎

Chapter 3 Convolutional Coding of Block Codes

3.1 Convolutional Techniques for Block Codes

Let $f(x) = \sum_{i=0}^{m-1} f_i x^i$ be a binary polynomial of degree $m-1$. We define $w(f)$ to be the number of non-zero coefficients of $f(x)$ and call $f(x)$ an even polynomial or odd polynomial if $w(f)$ is even or odd respectively. The associated circulant F of order n ($n \geq m$) is a square $n \times n$ matrix with the top row $(f_0, f_1, \ldots, f_{m-1}, 0, \ldots, 0)$ and every other row the right cyclic permutation of its preceding row. If A and B are two square matrices of order n, we define $G(A|B)$ to be the following $(n+1) \times (2n+2)$ matrix:

$$\begin{pmatrix} & & & 1 & & & & 1 \\ & A & & 1 & & B & & 1 \\ & & & \vdots & & & & \vdots \\ & & & 1 & & & & 1 \\ 1 & 1 & \cdots & 1 & 1 & 0 & 0 \cdots 0 & 0 \end{pmatrix}.$$

Let us consider a binary convolutional code of rate $\frac{1}{2}$ with constraint length K. Let $n \geq K$ be an odd integer and $i_0, i_1, \ldots, i_{n-1}$ be n information bits to be encoded. The taps are described by two odd polynomials

$$p(x) = \sum_{i=0}^{K-1} p_i x^i \quad \text{and} \quad q(x) = \sum_{i=0}^{K-1} q_i x^i,$$

where $(p_0, p_{K-1}) = (1,1)$, and $\gcd(p(x), q(x), x^n + 1) = 1$.

Extend the information bits to $i_{-K+1}, \ldots, i_{-1}, i_0, i_1, \ldots, i_{n-1}$, where $i_{-j} = i_{n-j}$ for $1 \leq j \leq K-1$, i.e., sequences of length $n+K-1$, in which the first $K-1$ symbols are repeated at the end.

The two output sequences depend on the input sequence in the following way:

$$\begin{aligned} a_j &= \sum_{l=0}^{K-1} p_l i_{j-l}, \quad 0 \le j \le n-1, \\ b_j &= \sum_{l=0}^{K-1} q_l i_{j-l}, \quad 0 \le j \le n-1. \end{aligned}$$

In terms of polynomials, when we write

$$I(x) = \sum_{l=0}^{n-1} i_l x^l, \; a(x) = \sum_{l=0}^{n-1} a_l x^l, \text{ and } b(x) = \sum_{l=0}^{n-1} b_l x^l,$$

the relations are

$$\begin{aligned} a(x) &\equiv I(x)p(x) \pmod{x^n+1}, \\ b(x) &\equiv I(x)q(x) \pmod{x^n+1}. \end{aligned}$$

In vector notation,

$$(i_0, i_1, \ldots, i_{n-1})(P|Q) = (a_0, \ldots, a_{n-1}, b_0, \ldots, b_{n-1})$$

where p and Q are $n \times n$ circulants with top row $(p_0, p_1, \ldots, p_{K-1}, 0, \ldots, 0)$, respectively $(q_0, q_1, \ldots, q_{K-1}, 0, \ldots, 0)$. So, the binary block code $\mathbf{C}$ via convolutional construction above has generator matrix

$$G = (P|Q).$$

The following three lemmas can be found in [18].

Lemma 3.1.1. $\dim(\mathbf{C}) = \text{rank}(P|Q) = n$.

Proof. If $I(x) = \sum_{l=0}^{n-1} i_l x^l$, then

$$(i_0, \ldots, i_{n-1})(P|Q) = (0, \ldots, 0|0, \ldots, 0)$$

iff $I(x)p(x) \equiv I(x)q(x) \equiv 0 \pmod{x^n+1}$. $I(x)p(x) = h(x)(x^n+1) = g(x)(x^n+1)$ for some $h(x)$ and $g(x)$. $\gcd(p(x), q(x), x^n+1) = 1$ implies that $I(x) = 0$. ∎

If $g(x)$ is a binary polynomial of degree $\leq n-1$ and $\gcd(g(x), x^n+1) = 1$, then $\frac{1}{g(x)} \pmod{x^n+1}$ is well-defined.

Lemma 3.1.2. The code **C** generated by $(P|Q)$ can also be generated by $(\hat{P}|\hat{Q})$, where $\hat{P}$ and $\hat{Q}$ are circulants of $\hat{p}(x) = \frac{p(x)}{g(x)}$, respectively, $\hat{q}(x) = \frac{q(x)}{g(x)}$. Especially, **C** can be generated by $(I|F)$ where I is identity matrix and F is circulant of $f(x) = \frac{q(x)}{p(x)}$.

Proof. Consider input polynomials $g(x)$ and $\frac{1}{g(x)}$. ∎

Especially, if $\gcd(p(x),\ x^n+1) = 1$, we have

Lemma 3.1.3. The code **C** generated by $(P|Q)$ is equivalent to the code **D** generated by $(I|F^d)$, where I is identity matrix and F^d is circulant of $\frac{x^d q(x)}{p(x)} \pmod{x^n+1}$ for integer d. ∎

Now let us generalize the convolutional construction by adding one more information bit i_n.

And extend codeword $(a_0, \ldots, a_{n-1}, b_0, \ldots, b_{n-1})$ to

$$(a_0, \ldots, a_{n-1}, a_\infty, b_0, \ldots, b_{n-1}, b_\infty)$$

where

$$a_\infty = \sum_{i=0}^{n-1} a_i, \quad b_\infty = \sum_{i=0}^{n-1} b_i.$$

Then the extended new code $\mathbf{C}'$ is generated by $G(P|Q)$, i.e.,

$$\begin{pmatrix} & & & & 1 & & & & & 1 \\ & & P & & 1 & & & Q & & 1 \\ & & & & \vdots & & & & & \vdots \\ & & & & 1 & & & & & 1 \\ 1 & 1 & \cdots & 1 & 1 & 0 & 0 & \cdots & 0 & 0 \end{pmatrix},$$

or

$$\left(\begin{array}{cccc|c|cccc|c} & & & & 1 & & & & & 1 \\ & & & & 1 & & & & & 1 \\ & I & & & \vdots & & & F & & \vdots \\ & & & & 1 & & & & & 1 \\ \hline 1 & 1 & \cdots & 1 & 1 & 0 & 0 & \cdots & 0 & 0 \end{array}\right),$$

which remains rate $\frac{1}{2}$.

Solomon and van Tilborg [18] proved

Theorem 3.1.4. Let **U** be the extension by an overall parity bit of the $(2n+1, n+1)$ binary quadratic residue code generated by $f_{QR}(x) = \prod_{i \in QR}(x + \alpha^i)$, where α is a primitive $(2n+1)^{st}$ root of unity, $QR = \{j^2 (\mathrm{mod} 2n+1); 1 \leq j \leq 2n\}$ and $2n+1$ is a prime of the form $8l \pm 1$.

If $(c_0, c_1 \ldots, c_{2n})$ is a codeword of the $(2n+1, n+1)$ QR code and for some $r, t \in QR$, $s \in NQR$ such that

$$\begin{aligned} \{r \cdot t^i; i = 0, 1, \ldots, n-1\} &= QR, \\ \{s \cdot t^i; i = 0, 1, \ldots, n-1\} &= NQR. \end{aligned}$$

Let

$$p_i = c_{r \cdot t^i}, \quad \text{and} \quad q_i = c_{s \cdot t^i}, \quad i = 0, 1, \ldots, n-1.$$

Note that $p(x)$ and $q(x)$ may not be reciprocal each other.

Then

$$\left(\begin{array}{cccc|c|cccc|c} & & & & 1 & & & & & 1 \\ & & & & 1 & & & & & 1 \\ & P & & & \vdots & & & Q & & \vdots \\ & & & & 1 & & & & & 1 \\ \hline 1 & 1 & \cdots & 1 & 1 & 0 & 0 & \cdots & 0 & 0 \end{array}\right)$$

is the generator matrix for the $(2n+2, n+1)$ extended QR code. ∎

3.2 Sufficient Conditions

For any two polynomials $p(x)$ and $q(x)$ of degree at most $n-1$, let P and Q be the associated circulants and consider the block code **C'** with generator matrix $G(P|Q)$. To get self-dual codes of rate 1/2 with weights of all codewords multiples of 4 by the convolutional technique, we apply a theorem that can be found in G. Solomon [12].

Theorem 3.2.1. Let $n \equiv 3 \pmod 4$, $p(x)$ a polynomial of degree $K-1$, $K \leq n$, with $\gcd(p(x), x^n+1) = 1$. Let $q(x) = \tilde{p}(x) = x^{K-1}p(x^{-1})$ be the reciprocal polynomial of $p(x)$. Then the code with generator matrix $G(P|Q)$ is a self-dual code of length $2(n+1)$, and all of the weights are multiples of 4. ∎

In [12] and [15] we used Theorem 3.2.1 and looked for the smallest possible constraint lengths K that could generate certain self-dual codes using tap polynomials $p(x)$ and $q(x)$. However, one finds that the reciprocal relation in Theorem 3.1.1 is not necessary even when the degrees of $p(x)$ and $q(x)$ are the same.

Example 3.2.2. Let $n = 11$, $p(x) = x^{10}+x^8+x^5+x+1$, $q(x) = x^{10}+x^9+x^7+x^4+1$. $\gcd(p(x), x^{11}+1) = \gcd(q(x), x^{11}+1) = 1$ and $G(P|Q)$ generates a $(24, 12)$ self-dual code with all the weights multiples of 4. One can check that $p(x)\tilde{p}(x) \equiv q(x)\tilde{q}(x) \equiv x^{10} \pmod{x^{11}+1}$.

3.3 Necessary Conditions

We examine some properties of the two tap polynomials $p(x)$ and $q(x)$ of self-dual codes with all weights multiples of 4.

Theorem 3.3.1. Let $p(x)$ and $q(x)$ be two polynomials of degree at most $n-1$ and P and Q the associated circulants of order n. If $G(P|Q)$ is a generator matrix

for a self-dual $(2n+2, n+1)$ code with the weight of every codeword multiple of 4, then

(1) $n \equiv 3 \pmod 4$,

(2) $w(p) \equiv w(q) \pmod 4$,

(3) both $w(p)$ and $w(q)$ are odd,

(4) $\gcd(p(x), q(x), x^n + 1) = 1$.

Proof. (1) The last row of $G(P|Q)$ gives us a codeword of weight $n+1 \equiv 0 \pmod 4$, which implies $n \equiv 3 \pmod 4$;

(2) Consider the sum of the first row and the last row. There is a codeword of weight $n - w(p) + w(q) + 1 \equiv 0 \pmod 4$. Then $w(p) \equiv w(q) \pmod 4$ by (1);

(3) Observe that the first row of $G(P|Q)$ gives us a codeword of weight $w(p) + w(q) + 2 \equiv 0 \pmod 4$. If $w(p) \equiv w(q) \equiv 0$ or $2 \pmod 4$, then $w(p) + w(q) + 2 \equiv 2 \pmod 4$. This is in contradiction to the observation. Hence both $w(p)$ and $w(q)$ are odd;

(4) If $\gcd(p(x), q(x), x^n + 1) \neq 1$, then the n rows of $(P|Q)$ are linearly dependent, which implies that there is a polynomial $I(x)$ such that $I(x)p(x) \equiv I(x)q(x) \equiv 0 \pmod{x^n + 1}$. $w(I)$ is even since both $w(p)$ and $w(q)$ are odd. Then the first n rows of $G(P|Q)$ are linearly dependent. Hence the rank of $G(P|Q)$ is less than $n+1$. This is in contradiction to the dimension of the code. ∎

3.4 The Λ_2 Mapping and Its Property

From now on let n, $p(x)$ and $q(x)$ satisfy the four conditions stated in Theorem 3.3.1 unless we specify otherwise.

For a fixed odd n, let β be a primitive n^{th} root of unity and $\mathcal{F}$ the extension field of $GF(2)$ containing β. Given any polynomial $I(x) = \sum_{i=0}^{n-1} a_i x^i$ there is an associated polynomial

$$g_I(z) = \sum_{j=0}^{n-1} c_j z^j$$

with

$$c_j = \sum_{i=0}^{n-1} a_i \beta^{-ij}, \quad j = 0, 1, \ldots, n-1,$$

where $g_I(z)$ and c_j's are the MS (Mattson-Solomon) polynomial and the coefficients of the MS polynomial of $I(x)$, respectively.

Let

$$\Gamma_2(I(x)) = \sum_{i=1}^{(n-1)/2} c_i c_{n-i}.$$

The Solomon-McEliece Γ_2 formula [17] implies

Lemma 3.4.1. For $n \equiv 3 \pmod 4$, then

$$w(I) \equiv 0, 3 \pmod 4 \text{ iff } \Gamma_2(I) \equiv 0 \pmod 2. \ \blacksquare$$

Now we define S_n for any odd n to be the set of all binary polynomials of degree at most $n-1$ and a mapping

$$\Lambda_2 : \ S_n \longrightarrow \mathcal{F}^{(n+1)/2}$$

with

$$\Lambda_2(I(x)) = (c_0 c_0, c_1 c_{n-1}, \ldots, c_i c_{n-i}, \ldots, c_{(n-1)/2} c_{(n+1)/2}).$$

Note that $\mathcal{F}^{(n+1)/2}$ is an $(n+1)/2$-dimensional vector space over $\mathcal{F}$ but Λ_2 is not a linear mapping in general.

We present a useful property of Λ_2.

Lemma 3.4.2. For any odd n, $\Lambda_2(S_n)$ contains a basis for $\mathcal{F}^{(n+1)/2}$.

Proof. We will construct a set of $(n+1)/2$ vectors in $\Lambda_2(S_n)$ which forms a basis for $\mathcal{F}^{(n+1)/2}$.

Let

$$\begin{aligned} e_0(x) &= 1, \\ e_1(x) &= 1+x, \\ e_2(x) &= 1+x^2, \\ &\vdots \\ e_i(x) &= 1+x^i, \quad \text{for } i \neq 0, \\ &\vdots \\ e_{(n-1)/2}(x) &= 1+x^{(n-1)/2}, \end{aligned}$$

and

$$\gamma_i = \beta^i + \beta^{-i}, \quad \text{for } i = 0, 1, \ldots, (n-1)/2.$$

It is easy to see that all γ_i are distinct, and

$\Lambda_2(e_0(x)) = (1, 1, \ldots, 1)$, $\Lambda_2(e_m(x)) = (0, \beta^m+\beta^{-m}, \ldots, \beta^{mi}+\beta^{-mi}, \ldots, \beta^{m(n-1)/2}+\beta^{-m(n-1)/2})$ for $1 \leq m \leq (n-1)/2$. So

$$\Lambda_2(e_m(x)) = (\gamma_0^m, \gamma_1^m, \ldots, \gamma_{(n-1)/2}^m), \quad \text{for } m = 1, 2.$$

Moreover, for $3 \leq m \leq (n-1)/2$, if m is odd

$$\begin{aligned} \beta^{mi} + \beta^{-mi} &= (\beta^i + \beta^{-i})^m + \binom{m}{1}(\beta^{(m-2)i} + \beta^{-(m-2)i}) + \cdots \\ &+ \binom{m}{(m-1)/2}(\beta^i + \beta^{-i}), \end{aligned}$$

then

$$\begin{aligned}\Lambda_2(e_m(x)) &= (\gamma_0^m, \gamma_1^m, \ldots, \gamma_{(n-1)/2}^m) + \binom{m}{1}\Lambda_2(e_{m-2}(x)) + \cdots \\ &+ \binom{m}{(m-1)/2}\Lambda_2(e_1(x));\end{aligned}$$

if m is even

$$\begin{aligned}\beta^{mi} + \beta^{-mi} &= (\beta^i + \beta^{-i})^m + \binom{m}{1}(\beta^{(m-2)i} + \beta^{-(m-2)i}) + \cdots \\ &+ \binom{m}{m/2-1}(\beta^{2i} + \beta^{-2i}) + \binom{m}{m/2},\end{aligned}$$

then

$$\begin{aligned}\Lambda_2(e_m(x)) &= (\gamma_0^m, \gamma_1^m, \ldots, \gamma_{(n-1)/2}^m) + \binom{m}{1}\Lambda_2(e_{m-2}(x)) + \cdots \\ &+ \binom{m}{m/2-1}\Lambda_2(e_2(x)) + \binom{m}{m/2}\Lambda_2(e_0(x)).\end{aligned}$$

Consider the square matrix of order $(n+1)/2$

$$\begin{pmatrix} \Lambda_2(e_0(x)) \\ \Lambda_2(e_1(x)) \\ \vdots \\ \Lambda_2(e_{(n-1)/2}(x)) \end{pmatrix};$$

any m^{th} row is a linear combination of $(\gamma_0^m, \gamma_1^m, \ldots, \gamma_{(n-1)/2}^m)$ and its previous rows.

So

$$\det\begin{pmatrix} \Lambda_2(e_0(x)) \\ \Lambda_2(e_1(x)) \\ \vdots \\ \Lambda_2(e_{(n-1)/2}(x)) \end{pmatrix} = \det\begin{pmatrix} 1 & 1 & \cdots & 1 \\ \gamma_0 & \gamma_1 & \cdots & \gamma_{(n-1)/2} \\ \gamma_0^2 & \gamma_1^2 & \cdots & \gamma_{(n-1)/2}^2 \\ \vdots & \vdots & \ddots & \vdots \\ \gamma_0^{(n-1)/2} & \gamma_1^{(n-1)/2} & \cdots & \gamma_{(n-1)/2}^{(n-1)/2} \end{pmatrix}$$
$$= \prod_{i>j}(\gamma_i - \gamma_j) \neq 0.$$

Therefore, $\{\Lambda_2(e_0(x)), \Lambda_2(e_1(x)), \ldots, \Lambda_2(e_{(n-1)/2}(x))\}$ forms a basis for $\mathcal{F}^{(n+1)/2}$. ∎

3.5 Necessary and Sufficient Conditions

Theorem 3.5.1. Let $p(x) = \sum_{i=0}^{K_1-1} p_i x^i$ and $q(x) = \sum_{i=0}^{K_2-1} q_i x^i$ be two polynomials with $\max(K_1 - 1, K_2 - 1) \leq n - 1$ and $p_{K_1-1} = q_{K_2-1} = p_0 = q_0 = 1$. Then $G(P|Q)$ is a generator matrix for a self-dual $(2n + 2, n + 1)$ code with the weight of every codeword multiple of 4 if and only if

(1) $n \equiv 3 \pmod 4$,

(2) $w(p) \equiv w(q) \pmod 4$,

(3) both $w(p)$ and $w(q)$ are odd,

(4) $\gcd(p(x), q(x), x^n + 1) = 1$,

(5) $x^{-K_1+1}p(x)\tilde{p}(x) + x^{-K_2+1}q(x)\tilde{q}(x) \equiv 0 \pmod{x^n + 1}$.

Note that for any $g(x) \in S_n$ with $\gcd(g(x), x^n + 1) = 1$, $g^{-1}(x) \pmod{x^n + 1}$ is well-defined in S_n, so x^{-K} is well-defined. (5) does not guarantee $p(x)$ and $q(x)$ are reciprocal to each other even if $K_1 = K_2$; see Example 3.2.2.

Proof. Suppose $G(P|Q)$ generates a self-dual $(2n+2, n+1)$ code with the weight of every codeword a multiple of 4. We only need to prove (5) because of Theorem

3.3.1 Consider any input/information polynomial $I(x) = \sum_{j=0}^{n-1} a_j x^j$, then the parity sequences of length $n = 4m - 1$ generated by $p(x)$ and $q(x)$ are of the form $I(x)p(x)$ and $I(x)q(x) \pmod{x^n + 1}$. Let c_i be the MS coefficients of the information sequence $I(x)$ of length n, then for parity sequences $I(x)p(x)$ and $I(x)q(x) \pmod{x^n + 1}$, their MS coefficients are $c_i p(\beta^{-i})$ and $c_i q(\beta^{-i})$, respectively.

Now consider sequences $I(x)p(x)$ and $I(x)q(x)$. Their Γ_2 formulae [17] are

$$\begin{aligned} \Gamma_2(I(x)p(x)) &= \sum_{i=1}^{(n-1)/2} c_i c_{n-i} p(\beta^i) p(\beta^{-i}) \\ \text{and} \quad \Gamma_2(I(x)q(x)) &= \sum_{i=1}^{(n-1)/2} c_i c_{n-i} q(\beta^i) q(\beta^{-i}) \end{aligned}$$

respectively.

Since $n \equiv 3 \pmod 4$, the weight of each codeword of length $2n + 2$ is multiple of 4 if and only if $\Gamma_2(I(x)p(x)) + \Gamma_2(I(x)q(x)) = 0$; see Lemma 3.4.4. And

$$\begin{aligned} \Gamma_2(I(x)p(x)) + \Gamma_2(I(x)q(x)) &= \sum_{i=1}^{(n-1)/2} c_i c_{n-i} [p(\beta^i)p(\beta^{-i}) + q(\beta^i)q(\beta^{-i})] \\ &= \sum_{i=0}^{(n-1)/2} c_i c_{n-i} [p(\beta^i)p(\beta^{-i}) + q(\beta^i)q(\beta^{-i})] \\ &= 0 \end{aligned}$$

consequently. (Note that $p(1)p(1) + q(1)q(1) = 0$ and we define $c_n = c_0$.)

In vector notation, it becomes

$$\Lambda_2(I(x)) \cdot (\vec{A} + \vec{B}) = 0$$

if we denote

$$\begin{aligned} \vec{A} &= (p(1)p(1), \ldots, p(\beta^i)p(\beta^{-i}), \ldots, p(\beta^{(n-1)/2})p(\beta^{-(n-1)/2})), \\ \text{and} \quad \vec{B} &= (q(1)q(1), \ldots, q(\beta^i)q(\beta^{-i}), \ldots, q(\beta^{(n-1)/2})q(\beta^{-(n-1)/2})). \end{aligned}$$

By Lemma 3.4.2 we can see that $\vec{A}+\vec{B}=\vec{0}$. That is $p(x)p(x^{-1})+q(x)q(x^{-1})=0$ for $x=\beta^i, i=0,1,\ldots,(n-1)/2$, and then also for $x=\beta^i, i=0,1,\ldots,n-1$. Therefore x^n+1 is a factor of $p(x)p(x^{-1})+q(x)q(x^{-1})$, i.e., $x^{-K_1+1}p(x)\tilde{p}(x)+x^{-K_2+1}q(x)\tilde{q}(x)\equiv 0 \pmod{x^n+1}$.

The other way is trivial if we can see that the rank of $G(P|Q)$ is $n+1$. One notices that (4) guarantees $\text{rank}(P|Q)=n$, and the last row of $G(P|Q)$ is not a linear combination of the first n rows. ∎

Corollary 3.5.2. Let $f(x)=\sum_{i=0}^{m-1} f_i x^i$ be a polynomial of degree $m-1\leq n-1$ with $f_0=f_{m-1}=1$ and I_n the identity matrix of order n. Then $G(I_n|F)$ is a generator matrix for a self-dual $(2n+2,n+1)$ code with the weight of every codeword being multiple of 4 if and only if

(1) $n\equiv 3 \pmod 4$,

(2) $w(f)\equiv 1 \pmod 4$,

(3) $f(x)\tilde{f}(x)\equiv x^{m-1} \pmod{x^n+1}$.

Note that such an $f(x)$ satisfies $\gcd(f(x),x^n+1)=1$.

Proof. Let $p(x)=1$, $q(x)=f(x)$ and use Theorem 3.5.1. ∎

Corollary 3.5.3. Let $p(x)=\sum_{i=0}^{K-1} p_i x^i$ and $q(x)=\sum_{i=0}^{K-1} q_i x^i$ be two polynomials of the same degree $K-1\leq (n+1)/2$ with $p_{K-1}=q_{K-1}=p_0=q_0=1$. If $\gcd(p(x),q(x))=1$ and $G(P|Q)$ is a generator matrix for a self-dual $(2n+2,n+1)$ code with the weight of every codeword multiple of 4, then $p(x)=\tilde{q}(x)$.

Proof. By Theorem 3.5.1, $p(x)\tilde{p}(x)+q(x)\tilde{q}(x)\equiv 0 \pmod{x^n+1}$. Since degree $(p(x)\tilde{p}(x)+q(x)\tilde{q}(x))\leq 2(K-1)-1\leq n$ and 0 is a root of $p(x)\tilde{p}(x)+q(x)\tilde{q}(x)$, then $p(x)\tilde{p}(x)=q(x)\tilde{q}(x)$. Since $\gcd(p(x),q(x))=1$, this leads to $p(x)=\tilde{q}(x)$. ∎

We now prove that such extended quadratic residue codes can be generated by matrices of the form $G(I|F)$. That is, we can find a codeword $\mathbf{a}=(a_0,a_1,\ldots,a_{2n})$ of

the $(2n+1, n+1)$ quadratic residue code with $a_0 = a_1 = 1$ and $a_i = 0$ for $i \neq 0, 1$ a quadratic residue.

Theorem 3.5.4. Let $\mathbf{U}$ be the extended $(2n+2, n+1)$ quadratic residue code generated by parity-check polynomial $g(x) = (x+1)\prod_{i \in QR}(x + \alpha^i)$, where α is a primitive $(2n+1)^{st}$ root of unity, $QR = \{j^2 (\text{mod } 2n+1); 1 \leq j \leq 2n\}$ and $2n+1$ is a prime of the form $8l - 1$. Then $\mathbf{U}$ can be generated by a matrix of the form $G(I|F)$ where F is the associated circulant of order n of some polynomial $f(x)$ with degree $n-1$.

Remark. We can find a polynomial $f(x)$ such that 1 and $f(x)$ can be used as taps in convolutional encoder for $(2n+2, n+1)$ QR code.

Proof. Let the parity-check polynomial

$$g(x) = \sum_{i=0}^{n+1} u_i x^i.$$

Let $a_0 = a_1 = 1$ and $a_i = 0$ if $i \neq 0, 1$ is a quadratic residue. Let us regard a_i as variables for all nonresidues i, then there are n variables. Since we want the codeword $\mathbf{a} = (a_0, a_1, \ldots, a_{2n})$ to satisfy

$$\sum_{i=0}^{n+1} u_i a_{t+i} = 0, \text{ for } t = 0, 1, \ldots, n-1,$$

we have n equations for n unknowns. So there is a solution for the n unknowns, and therefore there is a codeword $\mathbf{a} = (a_0, a_1, \ldots, a_{2n})$ such that $a_0 = a_1 = 1$ and $a_i = 0$ if $i \neq 0, 1$ is a quadratic residue. And the solution is nontrivial because $(1, 1, 0, \ldots, 0)$ is obviously not a codeword.

Let j be a multiplicative generator of the quadratic residues of $2n+1$, i.e., $j^n \equiv 1 \pmod{2n+1}$ and $\{j^i \pmod{2n+1},\ i = 0, 1, \ldots, n-1\} = QR$. Pick s to be any nonresidue; then $\{s \cdot j^i \pmod{2n+1},\ i = 0, 1, \ldots, n-1\} = NQR$. The following two

polynomials can be chosen as the convolutional encoder taps

$$f^{(1)}(x) = \sum_{i=0}^{n-1} f_i^{(1)} x^i \quad \text{with} \quad f_i^{(1)} = a_{j^i},$$
$$f^{(2)}(x) = \sum_{i=0}^{n-1} f_i^{(2)} x^i \quad \text{with} \quad f_i^{(2)} = a_{s \cdot j^i}.$$

It is easy to see that $f^{(1)}(x) = 1$. Let $f(x) = f^{(2)}(x)$ and F the associated circulant of $f(x)$ of order n, then $G(I|F)$ is a generator matrix for $\mathbf{U}$. ∎

Generally, from a convolutional coding point of view, if we regard equivalent codes to be the same, we want to find polynomials $p(x)$ and $q(x)$ with $q(x) \equiv f(x)p(x) \pmod{x^n + 1}$ and $\gcd(p(x), x^n + 1) = 1$ such that the maximum degree of $p(x)$ and $q(x)$ is as small as possible.

From Theorem 1.5 in Solomon and van Tilborg [18], one can find $p'(x)$ and $q'(x)$ with $q'(x) \equiv x^{d'} f(x)p'(x) \pmod{x^n + 1}$ for some $0 \leq d' \leq n-1$ and $q'(0) = p'(0) = 1$ such that max degree $(p'(x), q'(x)) \leq (n-1)/2$. If $\gcd(p'(x), x^n + 1) = 1$ we can choose $p(x) = \sum_{i=0}^{K_1-1} p_i x^i$, $q(x) = \sum_{i=0}^{K_2-1} q_i x^i$ with $q(x) \equiv x^d f(x)p(x) \pmod{x^n + 1}$ for some $0 \leq d \leq n-1$, $(p_0, p_{K_1-1}) = (q_0, q_{K_2-1}) = (1, 1)$ and $\gcd(p(x), q(x)) = 1$ by dividing $p'(x)$ and $q'(x)$ by their common factors. Note that max degree $(p(x), q(x))$ is at most max degree $(p'(x), q'(x))$. Moreover we have

Theorem 3.5.5. Under the conditions stated above, $p(x)$ and $q(x)$ are reciprocal to each other.

Proof. Without loss of generality, suppose $K_1 \geq K_2$. By Theorem 3.5.1

$$x^{-K_1+1} p(x)\tilde{p}(x) + x^{-K_2+1} q(x)\tilde{q}(x) \equiv 0 \pmod{x^n + 1},$$

and then

$$p(x)\tilde{p}(x) + x^{K_1-K_2} q(x)\tilde{q}(x) \equiv 0 \pmod{x^n + 1}.$$

Since

$$\text{degree}(p(x)\tilde{p}(x) + x^{K_1-K_2}q(x)\tilde{q}(x)) \leq 2(K_1 - 1) \leq n - 1,$$

$$p(x)\tilde{p}(x) + x^{K_1-K_2}q(x)\tilde{q}(x) = 0$$

$K_1 = K_2$ because $p(0) = \tilde{p}(0) = q(0) = \tilde{q}(0) = 1$. Therefore,

$p(x)\tilde{p}(x) + q(x)\tilde{q}(x) = 0$, and $\gcd(p(x), q(x)) = 1$ implies $p(x) = \tilde{q}(x)$. ∎

The theorems above explain why we can always find two reciprocal polynomials $p(x)$ and $q(x)$ to encode QR codes convolutionally in Solomon and van Tilborg [18].

3.6 Convolutional Encoding of Quadratic Residue Codes

3.6.1 (80, 40; 16) QR Code

The vector (c_i) is a codeword of the (79, 40; 15) QR code generated by check polynomial $g(x) = x^{40} + x^{39} + x^{37} + x^{35} + x^{32} + x^{29} + x^{28} + x^{24} + x^{22} + x^{18} + x^{17} + x^{16} + x^{15} + x^{13} + x^{12} + x^{11} + x^6 + x^4 + x^3 + 1$, where

$$\begin{aligned} c_i &= 1 \text{ for } i = 0, 1, 14, 15, 24, 30, 34, 35, 37, 39, 41, \\ &\quad 43, 47, 53, 57, 58, 61, 66, 68, 69, 70, 71, 74, \\ c_i &= 0 \text{ otherwise.} \end{aligned}$$

Let

$$\begin{aligned} f_i^{(1)} &= c_{1\cdot 44^i}, \\ f_i^{(2)} &= c_{3\cdot 44^i}, \quad \text{for } i = 0, 1, \ldots, 38. \end{aligned}$$

Then

$$\begin{aligned} f^{(1)}(x) &= 1, \\ f^{(2)}(x) &= x + x^2 + x^3 + x^6 + x^8 + x^{10} + x^{11} + x^{14} \\ &+ x^{15} + x^{17} + x^{19} + x^{21} + x^{22} + x^{23} + x^{25} + x^{26} \\ &+ x^{27} + x^{28} + x^{30} + x^{32} + x^{33}. \end{aligned}$$

Since

$$f^{(2)}(x) = \frac{x^d q(x)}{p(x)} \pmod{x^{39} + 1},$$

where

$$q(x) = 1 + x^2 + x^4 + x^5 + x^{11} + x^{12} + x^{13},$$

$$p(x) = 1 + x + x^2 + x^8 + x^9 + x^{11} + x^{13};$$

$$d = 27,$$

$$\gcd(p(x), x^{39} + 1) = 1,$$

and $K = 14$.

By previous results in Solomon and van Tilborg [18], we can use $p(x)$ and $q(x)$ as taps in the convolutional encoder with the tail bitting sequence of information of length 39. Appending the overall parity checks to the length 39 parity sequences and then adding the 40th information bit to the $p(x)$ sequence gives us the appropriate QR code. We have thus found an encoding with constraint length $K = 14$.

3.6.2 (80, 40; 16) Non-QR Code

On the other hand, let us use as the encoding taps the following polynomials in the convolutional encoder:

$$p(x) = 1 + x + x^2 + x^4 + x^5 + x^{10} + x^{12},$$

$$q(x) = 1 + x^2 + x^7 + x^8 + x^{10} + x^{11} + x^{12},$$

$$K = 13.$$

Adjoining the parity checks to the parity sequences and then adding the 40th information bit to the $p(x)$ sequence, we construct an even self-dual (80, 40; 16) block code. The minimum distance is verified by computer simulation. The code may not be QR code.

Problem 3.6.2 Is the (80, 40; 16) code a non-QR code?

3.6.3 (104, 52; 20) QR Code

The vector (c_i) is a codeword of the $(103, 52; 19)$ QR code generated by check polynomial $g(x) = x^{52} + x^{51} + x^{50} + x^{48} + x^{45} + x^{42} + x^{38} + x^{37} + x^{36} + x^{35} + x^{33} + x^{28} + x^{27} + x^{26} + x^{21} + x^{17} + x^{16} + x^{12} + x^{10} + x^8 + x^4 + x^3 + x^2 + 1$, where

$$\begin{aligned} c_i &= 1 \text{ for } i = 0, 1, 6, 10, 11, 12, 31, 37, 39, 43, 45, \\ &\qquad 47, 48, 53, 54, 73, 75, 85, 87, 88, 89, 99, 101, \\ c_i &= 0 \text{ otherwise.} \end{aligned}$$

Let

$$\begin{aligned} f_i^{(1)} &= c_{1\cdot 2^i}, \\ f_i^{(2)} &= c_{3\cdot 2^i}, \text{ for } i = 0, 1, \ldots, 50. \end{aligned}$$

Then

$$\begin{aligned} f^{(1)}(x) &= 1, \\ f^{(2)}(x) &= x + x^2 + x^4 + x^6 + x^7 + x^8 + x^{10} + x^{12} \\ &+ x^{19} + x^{21} + x^{26} + x^{29} + x^{30} + x^{31} + x^{36} \\ &+ x^{38} + x^{43} + x^{44} + x^{46} + x^{48} + x^{50}. \end{aligned}$$

Since

$$f^{(2)}(x) = \frac{x^d q(x)}{p(x)} \pmod{x^{51}+1},$$

where

$$\begin{aligned} q(x) &= 1 + x^4 + x^5 + x^9 + x^{10} + x^{12} + x^{15} \\ &+ x^{16} + x^{17} + x^{19} + x^{20}, \\ p(x) &= 1 + x + x^3 + x^4 + x^5 + x^8 + x^{10} + x^{11} \\ &+ x^{15} + x^{16} + x^{20}; \\ d &= 30, \quad \gcd(p(x),\ x^{51}+1) = 1, \\ \text{and} \quad K &= 21. \end{aligned}$$

We can use $p(x)$ and $q(x)$ above in the convolutional encoder. We have thus found an encoding with constraint length $K = 21$.

3.7 Convolutional Coding of Some Cyclic Codes and Further Problem

3.7.1 The (24, 12; 8) Golay Code and a (33, 22; 6) Cyclic Code

We know that the (24, 12; 8) Golay code can be convolutionally encoded by $[x^3 + x^2 + 1 \quad x^3 + x + 1]$. Without the overall parity checks and the 12th information bit, this convolutional algorithm gives us a (22, 11; 6) code.

Now let us consider the rate 2/3 (33, 22; 6) code convolutionally encoded by

$$\begin{bmatrix} x^3 + x^2 + 1 & 0 & x^3 + x + 1 \\ 0 & x^3 + x + 1 & x^3 + x^2 + 1 \end{bmatrix}.$$

Let $a_{-3}, a_{-2}, a_{-1}, a_0, a_1, \ldots, a_{10}$ and $b_{-3}, b_{-2}, b_{-1}, b_0, b_1, \ldots, b_{10}$ be two sets of in-

formation sequences, where $a_{-j} = a_{11-j}, b_{-j} = b_{11-j}, j = 1, 2, 3$. Parity sequences are c_j, d_j and e_j, where $c_j = a_{j-3} + a_{j-2} + a_j$, $d_j = b_{j-3} + b_{j-1} + b_j$ and $d_j = a_{j-3} + a_{j-1} + a_j + b_{j-3} + b_{j-2} + b_j, j = 0, 1, \ldots, 10$. Then

Theorem 3.7.1. The (33, 22; 6) code above is isomorphic to the cyclic code with generator polynomial $f(x) = (x+1)(x^{10}+x^9+x^5+x+1) = x^{11}+x^9+x^6+x^5+x^2+1$.

Proof. Consider the cyclic code. Make three rows for the 33-bit codeword

$$\begin{array}{ccccccccccc} 0 & 3 & 6 & 9 & 12 & 15 & 18 & 21 & 24 & 27 & 30 \\ 2 & 5 & 8 & 11 & 14 & 17 & 20 & 23 & 26 & 29 & 32 \\ 4 & 7 & 10 & 13 & 16 & 18 & 22 & 25 & 28 & 31 & 1 \end{array}.$$

Note that the codeword of the generator polynomial gives

$$\begin{array}{ccccccccccc} 1 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{array},$$

and the cyclic shift by two gives

$$\begin{array}{ccccccccccc} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{array}.$$

Therefore, the permutation of the three rows give an isomorphism between the two codes. ∎

3.7.2 Convolutional Encoding of a (65, 52; 6) Cyclic Code

Theorem 3.7.2. The (65, 52; 6) cyclic code with generator polynomial $f(x) = x^{13} + x^{12} + x^9 + x^4 + x + 1$ can be encoded convolutionally by

$$\begin{bmatrix} x^4+x+1 & x^4+x^3+1 & 0 & 0 & 0 \\ x^3+x+1 & 0 & x^3+x^2+1 & 0 & 0 \\ x^3+x^2+1 & 0 & 0 & x^3+x+1 & 0 \\ x^4+x^3+1 & 0 & 0 & 0 & x^4+x+1 \end{bmatrix}.$$

Proof. Consider the cyclic code. Make five rows for the 65-bit codeword

$$\begin{array}{cccccccccccccc} 0 & 5 & 10 & 15 & 20 & 25 & 30 & 35 & 40 & 45 & 50 & 55 & 60 \\ 1 & 6 & 11 & 16 & 21 & 26 & 31 & 36 & 41 & 46 & 51 & 56 & 61 \\ 17 & 22 & 27 & 32 & 37 & 42 & 47 & 52 & 57 & 62 & 2 & 7 & 12 \\ 48 & 53 & 58 & 63 & 3 & 8 & 13 & 18 & 23 & 28 & 33 & 38 & 43 \\ 64 & 4 & 9 & 14 & 19 & 24 & 29 & 34 & 39 & 44 & 49 & 54 & 59 \end{array}.$$

A codeword $f(x)(x^8 + x^4 + 1) = x^{21} + x^{20} + x^{16} + x^5 + x + 1$ gives

$$\begin{array}{cccccccccccccc} 1 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{array}.$$

Similarly, $x^{32}+x^{27}+x^{17}+x^{15}+x^5+1$ gives

$$\begin{matrix} 1 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{matrix},$$

$x^{63}+x^{53}+x^{48}+x^{15}+x^{10}+1$ gives

$$\begin{matrix} 1 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{matrix},$$

and $x^{64}+x^{20}+x^{19}+x^{15}+x^4+1$ gives

$$\begin{matrix} 1 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{matrix}.$$

So the rate 4/5 cyclic code can be convolutionally encoded by

$$\begin{bmatrix} x^4+x+1 & x^4+x^3+1 & 0 & 0 & 0 \\ x^3+x+1 & 0 & x^3+x^2+1 & 0 & 0 \\ x^3+x^2+1 & 0 & 0 & x^3+x+1 & 0 \\ x^4+x^3+1 & 0 & 0 & 0 & x^4+x+1 \end{bmatrix}. \blacksquare$$

3.7.3 Further Problem

Problem 3.7.3. Let $\mathbf{C}$ be a rate $(n-1)/n$ cyclic code with generator polynomial $g(x)$. If $g(x)$ is self-reciprocal, i.e., $\tilde{g}(x) = g(x)$. Then the code can be convolutionally encoded by

$$\begin{bmatrix} p_1(x) & q_1(x) & 0 & \cdots & 0 \\ p_2(x) & 0 & q_2(x) & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ p_{n-1}(x) & 0 & 0 & \cdots & q_{n-1}(x) \end{bmatrix},$$

where $\tilde{p}_i(x) = q_i(x)$ for $1 \leq i \leq n-1$.

Bibliography

[1] R.W.D. Booth, M.A. Herro and G. Solomon, *Convolutional coding techniques for certain quadratic residue codes*, International Telemetering Conference, (XI) Proceedings (Silver Springs, Maryland) 1975.

[2] A. Lempel and M.J. Weinberger, *Self-Complementary Normal Bases in Finite Fields*, SIAM J., Disc. Math., Vol. 1, No. 2, May 1988, pp. 193–198.

[3] J.H. van Lint, *Introduction to Coding Theory*, GTM 86, New York–Heidelberg–Berlin, Springer–Verlag, 1982.

[4] J.H. van Lint and R.M. Wilson, *A Course in Combinatorics*, Cambridge University Press, 1992.

[5] J.H. van Lint and R.M. Wilson, *On the Minimium Distance of Cyclic Codes*, IEEE Transactions on Information Theory, Vol. IT-32, No. 1, January 1986, pp. 23–40.

[6] F.J. MacWilliams and N.J.A. Sloane, *The Theory of Error-Correcting Codes*, North-Holland, Amsterdam, 1986.

[7] H.F. Mattson and G. Solomon, *A New Treatment of BCH Codes*, J. Soc. Industr. Appl. Math., 9 (1961), pp. 654–669.

[8] R.J. McEliece, *The Theory of Information and Coding*, Encylopedia of Math. and its Applications. vol. 3. Reading, Mass.: Addison-Wesley, 1977.

[9] W.W. Peterson and E.J. Weldon, *Error-correcting Codes* (2nd ed.), Cambridge, Mass.: MIT Press, 1972.

[10] G. Solomon, *A (72, 36; 15) Box Code* The Telecommunications and Data Acquisition Progress Report 42–112, Vol. October-December 1992, Jet Propulsion Laboratory, Pasadena, California, February 15, 1992, pp. 19–21.

[11] G. Solomon, *Algebraic Coding Theory*, in A.V. Balakrishnan, ed., *Communication Theory*, Chapter 6, Inter–University Electronics Series, Vol. 6, McGraw–Hill, 1968.

[12] G. Solomon, *Convolutional Encoding of Self-Dual Codes*, The Telecommunications and Data Acquisition Progress Report 42–116, Vol. October-December 1993, Jet Propulsion Laboratory, Pasadena, California, February 15, 1994, pp. 110–113.

[13] G. Solomon, *Self-Dual (48, 24; 12) Codes*, The Telecommunications and Data Acquisition Progress Report 42–111, Vol. July-September 1992, Jet Propulsion Laboratory, Pasadena, California, November 15, 1992, pp. 75–79.

[14] G. Solomon and Y. Jin, *Box Codes of Lengths 48 and 72*, The Telecommunications and Data Acquisition Progress Report 42–115, Vol. July-September 1993, Jet Propulsion Laboratory, Pasadena, California, November 15, 1993, pp. 105–109.

[15] G. Solomon and Y. Jin, *Convolutional Encoding of Self-Dual Codes (II)*, The Telecommunications and Data Acquisition Progress Report 42–118, Vol. April-June 1994, Jet Propulsion Laboratory, Pasadena, California, August 15, 1994, pp. 22–25.

[16] G. Solomon and Y. Jin, *Convolutional Encoding of Self-Dual Codes (III)*, The Telecommunications and Data Acquisition Progress Report, Jet Propulsion Laboratory, Pasadena, California, in press.

[17] G. Solomon and R.J. McEliece, *Weights of Cyclic Codes*, Journal of Combinatorial Theory, Vol. 1, No. 4, December 1966, pp. 459–475.

[18] G. Solomon and H.C.A. van Tilborg, *A Connection Between Block and Convolutional Codes*, SIAM J. APPL. MATH. Vol. 37, No. 2, Oct. 1979, pp. 358–369.