PERMUTATION POLYNOMIAL BASED INTERLEAVERS FOR TURBO CODES OVER INTEGER RINGS: THEORY AND APPLICATIONS.

DISSERTATION

Presented in Partial Fulfillment of the Requirements for the Degree Doctor of Philosophy in the Graduate School of The Ohio State University

By

Jong Hoon Ryu, B.E., M.E.

* * * * *

The Ohio State University

2007

Dissertation Committee:

Hesham El Gamal, Adviser

Oscar Y. Takeshita, Co-Adviser

Lee C. Potter

Vadim Utkin

Approved by

Adviser

Co-Adviser

Graduate Program in Electrical and Computer Engineering
ABSTRACT

Turbo codes are a class of high performance error correcting codes (ECC) and an interleaver is a critical component for the channel coding performance of turbo codes. Algebraic constructions of interleavers are of particular interest because they admit analytical designs and simple, practical hardware implementation. Sun and Takeshita [33] have shown that the class of quadratic permutation polynomials over integer rings provides excellent performance for turbo codes. Recently, quadratic permutation polynomial (QPP) based interleavers have been proposed into 3rd Generation Partnership Project Long Term Evolution (3GPP LTE) draft [55] for their excellent error performance, simple implementation and algebraic properties which admit parallel processing and regularity. In some applications, such as deep space communications, a simple implementation of deinterleaver is also of importance.

In this dissertation, a necessary and sufficient condition is proven for the existence of a quadratic inverse polynomial (deinterleaver) for a quadratic permutation polynomial over an integer ring. Further, a simple construction is given for the quadratic inverse.

We also consider the inverses of QPPs which do not admit quadratic inverses. It is shown that most 3GPP LTE interleavers admit quadratic inverses. However, it is shown that even when the 3GPP LTE interleavers do not admit quadratic inverses,
the degrees of the inverse polynomials are less than or equal to 4, which allows a simple implementation of the deinterleavers. An explanation is argued for the observation.

The minimum distance and its multiplicity (or the first a few terms of the weight distribution) of error correcting codes are used to estimate the error performance at high signal-to-noise ratio (SNR). We consider efficient algorithms that find an upper bound (UB) on the minimum distance of turbo codes designed with QPP interleavers.

Permutation polynomials have been extensively studied [45, 46, 47, 48, 49, 50, 51, 52], but simple coefficient tests for permutation polynomials over integer rings are only known for limited cases. A simple necessary and sufficient coefficient test is proven for cubic permutation polynomials over integer rings. A possible application is in the design of interleavers for turbo codes.
To my family
ACKNOWLEDGMENTS

I would like to express my sincere gratitude to my adviser and co-adviser, Profs. Hesham El Gamal and Oscar Y. Takeshita, for their enthusiasm, motivation and guidance. They have patiently supported and encouraged me throughout my study. When I was in need of help, they were always willing to help me. Without their direction, knowledge, and most of all, their patience, I could not have completed the Ph.D. program.

My grateful thanks go to Profs. Lee C. Potter and Vadim Utkin for kindly agreeing to be in my dissertation committee, for spending their valuable time on reading my dissertation and for suggesting invaluable feedback and comments in spite of their busy schedules.

I would like to thank Dr. Eirik Rosnes and Yen-Lun Chen for the pleasant discussions that we have had.

I would also like to express my gratitude to IPS members including Korean student with whom I have enjoyed numerous fruitful discussions and spent memorable times.

Finally, I would like to give my thanks to my family for their affection and support throughout my life.
VITA


1998 ................................. B.E. Electronics Engineering,
                      Pusan National University,
                      Pusan, Korea.

2000 ................................. M.E. Electronics Engineering,
                      Pusan National University,
                      Pusan, Korea.

Mar. 2000 - Mar. 2001 .................. Engineer,
                      LG Electronics, Ltd., Seoul, Korea.

2001-present .......................... Ph.D. Student,
                      The Ohio State University.

PUBLICATIONS

Research Publications


FIELDS OF STUDY

Major Field: Electrical and Computer Engineering

Studies in:

- Communication and Signal Processing  Prof. Hesham El Gamal
- Communication and Signal Processing  Prof. Oscar Y. Takeshita
- Communication and Signal Processing  Prof. Lee C. Potter
- Control Theory  Prof. Vadim Utkin
# TABLE OF CONTENTS

<table>
<thead>
<tr>
<th>Chapters</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Abstract</td>
<td>ii</td>
</tr>
<tr>
<td>Dedication</td>
<td>iv</td>
</tr>
<tr>
<td>Acknowledgments</td>
<td>v</td>
</tr>
<tr>
<td>Vita</td>
<td>vi</td>
</tr>
<tr>
<td>List of Tables</td>
<td>xi</td>
</tr>
<tr>
<td>List of Figures</td>
<td>xiii</td>
</tr>
<tr>
<td>Chapters:</td>
<td></td>
</tr>
<tr>
<td>1. Introduction</td>
<td>1</td>
</tr>
<tr>
<td>1.1 Historical Remarks</td>
<td>1</td>
</tr>
<tr>
<td>1.2 Background of Turbo Codes</td>
<td>2</td>
</tr>
<tr>
<td>1.2.1 Turbo Code Encoder Structure</td>
<td>2</td>
</tr>
<tr>
<td>1.2.2 Turbo Code Decoder Structure</td>
<td>3</td>
</tr>
<tr>
<td>1.3 Interleavers for Turbo Codes</td>
<td>4</td>
</tr>
<tr>
<td>1.3.1 Classification of Interleavers</td>
<td>4</td>
</tr>
<tr>
<td>1.3.2 Interleavers for Mobile 3rd Generation (3G) Telephony Standards</td>
<td>6</td>
</tr>
<tr>
<td>1.3.3 Quadratic Permutation Polynomial Interleavers over Integer Rings</td>
<td>7</td>
</tr>
<tr>
<td>2. On Quadratic Inverses for Quadratic Permutation Polynomials over Integer Rings</td>
<td>9</td>
</tr>
<tr>
<td>2.1 Introduction</td>
<td>9</td>
</tr>
</tbody>
</table>
5.2 Permutation Polynomial over Integer Rings .................................. 76
5.3 A Simple Coefficient Test for Cubic PPs ..................................... 77
  5.3.1 $p = 2$ .............................................................................. 77
  5.3.2 $p = 3$ .............................................................................. 78
  5.3.3 $3|p - 1$ ............................................................................ 78
  5.3.4 $3 
mid p - 1$ with $p > 3$ .............................................................. 79
5.4 Examples .................................................................................. 81
5.5 Conclusion ................................................................................ 82

6. Conclusions ................................................................................ 83

Appendices:

A. Proof of Theorems and Lemmas in Chapter 2 ............................... 85
  A.1 Proof of Lemma 10 .................................................................. 85
  A.2 Proof of Lemma 11 .................................................................. 87
  A.3 Proof of Lemma 12 .................................................................. 89
  A.4 Proof of Lemma 13 .................................................................. 90
  A.5 Proof of Theorem 14 ............................................................... 91
  A.6 Proof of Theorem 15 ............................................................... 91

B. Proof of Lemmas in Chapter 3 ..................................................... 94
  B.1 Proof of Lemma 33 .................................................................. 94
  B.2 Proof of Lemma 34 .................................................................. 97

C. Proof of Lemmas in Chapter 4 ..................................................... 101
  C.1 Proof of Lemma 50 .................................................................. 101
  C.2 Proof of Lemma 51 .................................................................. 102
  C.3 Proof of Lemma 55 .................................................................. 103
  C.4 Proof of Lemma 54 .................................................................. 104
  C.5 Proof of Lemma 56 .................................................................. 105
  C.6 Proof of Lemma 65 .................................................................. 107

Bibliography .................................................................................. 109
# LIST OF TABLES

<table>
<thead>
<tr>
<th>Table</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.1</td>
<td>Self-inverting quadratic polynomials for 3GPP LTE interleavers</td>
</tr>
<tr>
<td>2.2</td>
<td>Algorithm: computation of quadratic inverses</td>
</tr>
<tr>
<td>2.3</td>
<td>Algorithm: computation of an inverse of an element over integer ring</td>
</tr>
<tr>
<td>2.4</td>
<td>Quadratic inverse polynomials for 3GPP LTE interleavers for lengths 2000 to 4000</td>
</tr>
<tr>
<td>2.5</td>
<td>Quadratic inverse polynomials for 3GPP LTE interleavers for lengths over 4000</td>
</tr>
<tr>
<td>3.1</td>
<td>Algorithm: computation of inverses of quadratic permutation polynomials</td>
</tr>
<tr>
<td>3.2</td>
<td>Least degree inverse polynomials for 3GPP LTE interleavers without quadratic inverses</td>
</tr>
<tr>
<td>4.1</td>
<td>An upper bound on the $d_{\min}$ of turbo codes designed with QPP interleavers which do not admit quadratic inverses by input weight 2 pattern analysis when the encoder has 8 states</td>
</tr>
<tr>
<td>4.2</td>
<td>An upper bound on the $d_{\min}$ of turbo codes designed with QPP interleavers which do not admit quadratic inverses by input weight 2 pattern analysis when the encoder has 16 states</td>
</tr>
<tr>
<td>4.3</td>
<td>Summary of conditions that make the pattern in Figure 4.6 a codeword of Hamming weight at most $2(2^{\nu+1} + 9)$, for $\nu \geq 3$ (1)</td>
</tr>
<tr>
<td>4.4</td>
<td>Summary of conditions that make the pattern in Figure 4.6 a codeword of Hamming weight at most $2(2^{\nu+1} + 9)$, for $\nu \geq 3$ (2)</td>
</tr>
</tbody>
</table>
4.5 Summary of conditions on $N$ (derived using the congruences in Table 4.3 and 4.4) for various upper bounds on the optimum minimum Hamming weight when $\nu = 3$. ................................. 61

4.6 Algorithm: computation of weights generated by input weight $2k$ patterns. ........................................... 72

4.7 Upper bounds on the minimum distance of turbo codes designed with QPP interleavers using input weight 4,6 pattern analysis ($N > 4000$, 8 states). ..................................................... 73

5.1 A simple coefficient test for cubic PP modulo $p^n$ ................................. 81
# LIST OF FIGURES

<table>
<thead>
<tr>
<th>Figure</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.1</td>
<td>An interleaver.</td>
</tr>
<tr>
<td>1.2</td>
<td>Turbo code encoder.</td>
</tr>
<tr>
<td>1.3</td>
<td>Turbo code decoder.</td>
</tr>
<tr>
<td>3.1</td>
<td>The Catalan numbers.</td>
</tr>
<tr>
<td>4.1</td>
<td>A codeword that can be removed by high spread interleavers</td>
</tr>
<tr>
<td>4.2</td>
<td>A codeword that cannot be removed by high spread interleavers</td>
</tr>
<tr>
<td>4.3</td>
<td>An input weight 6 critical codeword for QPP interleavers (1)</td>
</tr>
<tr>
<td>4.4</td>
<td>An input weight 6 critical codeword for QPP interleavers (2)</td>
</tr>
<tr>
<td>4.5</td>
<td>An input weight 6 critical codeword for QPP interleavers (3)</td>
</tr>
<tr>
<td>4.6</td>
<td>An input weight 4 critical codeword for QPP interleavers</td>
</tr>
<tr>
<td>4.7</td>
<td>A generalized linear permutation polynomial.</td>
</tr>
<tr>
<td>4.8</td>
<td>The relationship among QPP, ARP, DRP and GLPP.</td>
</tr>
<tr>
<td>4.9</td>
<td>An input weight 6 pattern.</td>
</tr>
<tr>
<td>4.10</td>
<td>An input weight 4 pattern.</td>
</tr>
<tr>
<td>4.11</td>
<td>An input weight 8 pattern.</td>
</tr>
</tbody>
</table>
CHAPTER 1

INTRODUCTION

1.1 Historical Remarks

In modern digital communication applications, high reliability of transmission is required. In his landmark paper [1], Shannon proved in 1948 that if the information transmission rate is less than a quantity called channel capacity or Shannon capacity, asymptotically error free communication is possible over an additive white Gaussian noise (AWGN) channel.

Unfortunately, although the existence of a code that achieves the capacity was shown in Shannon’s paper [1], a practical channel coding scheme with reasonable encoding and decoding complexity was not shown in the paper [1].

After Shannon’s proof [1], there was a great deal of effort to find a channel code that approaches the capacity with reasonable complexity. The task, however, seemed to be formidable until the invention of turbo codes.

In 1993, Berrou et al. [3] introduced a new class of error correcting codes called turbo codes. A turbo code is able to achieve the capacity within a fraction of a decibel (dB) when used with a well designed interleaver and iterative decoding. It is remarkable that although Low Density Parity Check (LDPC) [22, 23, 24, 25, 26,
27, 28, 29, 30] codes, another important channel codes which approach the capacity, were invented by Gallager [22] in 1963, it was largely forgotten for nearly 30 years until rediscovered by MacKay et al. [24, 25].

1.2 Background of Turbo Codes

In his 1948 paper, Shannon showed that a random code approaches the capacity as the codelength increases. However, for a random code, encoding and decoding is impractical. Following Shannon’s proof, a channel code with reasonable encoding and decoding complexity is expected to satisfy two conditions:

1. randomness, as in Shannon’s proof, that enables the code to approach the capacity as the codelength increases.

2. a simple structure that allows reasonable encoding and decoding complexity.

Berrou et al. [3] showed that both the condition can be fulfilled by adopting an interleaver and parallel concatenated convolutional codes. An interleaver between two concatenated convolutional codes induces randomness, and the reasonable encoding and decoding complexity is accomplished by using iterative decoding for each concatenated convolutional code.

1.2.1 Turbo Code Encoder Structure

A turbo code encoder consists of constituent encoders and an interleaver [Figure 1.1]. The constituent encoders are concatenated in parallel via the interleaver.

In Figure 1.2, an information sequence \( u \) is fed directly to the constituent encoder 1. The same information sequence is interleaved and fed to the constituent code 2 and the output sequences \( v_1, v_2 \) together with the information sequence \( v_0 \) are scrambled
and transmitted over the channel.

\begin{figure}[h]
\centering
\includegraphics[width=0.8\textwidth]{interleaver.png}
\caption{An interleaver.}
\end{figure}

\begin{figure}[h]
\centering
\includegraphics[width=0.8\textwidth]{turbo_encoder.png}
\caption{Turbo code encoder.}
\end{figure}

\subsection{Turbo Code Decoder Structure}

Figure 1.3 shows turbo decoder. Turbo decoder consists of two Bahl, Cocke, Jelinek and Raviv (BCJR) [32] algorithms. For each BCJR algorithm, the inputs to the algorithm are the received sequence

\begin{equation}
\mathbf{r} = (r_0(0)r_0(1)r_0(2), r_1(0)r_1(1)r_1(2), ..., r_{N+\nu-1}(0)r_{N+\nu-1}(1)r_{N+\nu-1}(2)) \quad (1.1)
\end{equation}
and a priori log-likelihood ratios or L-values of the information bits $L_a(r_k(0))$, $k = 0, 1, ..., N - 1$. In equation (1.1), $r_k(0)$ is the $k$-th received information bit, $r_k(1)$ and $r_k(2)$ are $k$-th received parity bits, and $\nu$ is the constraint length of the encoder.

In Figure 1.3, $L_c = 4\frac{E_s}{N_0}$ is the channel reliability factor. At the beginning of the decoding, $L_a^{(1)}$, a priori L-value to the first decoder is set to 0’s and at the end of the decoder 1, the extrinsic a posteriori L-value $L^{(1)}$ is computed. The a priori L-value $L_a^{(2)}$ that is passed to the decoder 2 is interleaved value $L_e^{(1)} = L^{(1)} - (L_c r^{(0)} + L_a^{(1)})$, where

\[ r^{(0)} = (r_0(0), r_1(0), r_2(0), ..., r_{N+\nu-1}(0)). \]

Decoder 2 computes $L^{(2)}$, $L_e^{(2)} = L^{(2)} - (L_c r^{(0)} + L_a^{(2)})$ and the interleaved $L_e^{(2)}$ is passed to the decoder 1 as a priori L-value $L_a^{(1)}$. After iterations, $L^{(2)}$ is deinterleaved and hard-decisioned to compute the estimated information bits.

1.3 Interleavers for Turbo Codes

An interleavers is a critical components of turbo codes. In Subsection 1.3.1, interleavers are classified according to their construction methods. Turbo codes are extensively used in many applications including mobile telephony standards. In Subsection 1.3.2, interleavers used in 3GPP are introduced. In Subsection 1.3.3, QPP based interleavers are presented.

1.3.1 Classification of Interleavers

There are a number of interleavers [3, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 33, 34] for turbo codes and each interleaver was constructed on different criteria. In the following, interleavers are classified in terms of construction methods.
1. Purely random interleavers(Random interleavers with a structure)

Purely random interleavers were suggested in the original turbo codes [3] and the interleaver index is selected in a purely random way. Although the performance of the turbo codes using purely random interleavers is remarkable, it can be further improved by using more sophisticated interleavers.

The S-random interleaver is constructed at random, but it requires the constraint \( |\pi(i) - \pi(j)| > S \) for all \( i, j \) such that \( |i - j| < S \), where \( \pi \) is a permutation by an interleaver and \( S \) is a quantity called maximum spread. The
spread $S$ is closely related to the weight generated by input 2 sequences, and increasing $S$ removes low weight sequences by input 2 sequences.

2. Structured interleavers with a random nature

Dithered Relative Prime (DRP) interleavers [10, 11, 12, 13, 14, 15, 16] and Almost Regular Permutation (ARP) interleavers [20] belong to the this class. DRP and ARP interleavers show excellent error performance.

3. Algebraic/Number theoretic interleavers

Linear interleavers, Takeshita-Costello interleavers [34], permutation polynomial over finite field based interleavers [19] and permutation polynomial over integer ring based interleavers [33, 35, 36, 37, 38, 39, 40] belong to this class. One of the advantages of these interleavers over others is simple representation of the interleaver. Consequently, they are suitable for applications that require high-speed, low-power consumption and little memory.

1.3.2 Interleavers for Mobile 3rd Generation (3G) Telephony Standards

Turbo codes are/will be used in many applications including

1. 3rd Generation mobile telephony.

2. deep space satellite communications such as MediaFLO, terrestrial mobile television system.

3. New NASA missions such as Mars Reconnaissance Orbiter (MRO).

4. IEEE 802.16, a wireless metropolitan network.
One of the most important applications of turbo codes is 3G mobile telephony. There are 188 interleavers for turbo codes in the 3GPP LTE [55] as follows,

1. $40 \leq N \leq 512$, interleaver set contains all multiples of 8.
2. $512 < N \leq 1024$, interleaver set contains all multiples of 16.
3. $1024 < N \leq 2048$, interleaver set contains all multiples of 32.
4. $2048 < N \leq 6144$, interleaver set contains all multiples of 64.

### 1.3.3 Quadratic Permutation Polynomial Interleavers over Integer Rings

Recently, quadratic permutation polynomial based interleavers have been proposed into the 3GPP LTE draft. Existing release-6 interleavers [55] use a table and simple arithmetic for generating the interleaver indexes efficiently. It has been demonstrated that QPP interleavers have better performance compared to the existing release-6 interleavers [55] with less or equal complexity. QPP interleavers have major advantages over other earlier interleaver constructions because it simultaneously provides

1. Excellent error performance with practical codelength [33, 36, 55].
2. Structures suitable for parallel processing [35].
3. Efficient implementation of interleaver [41]/deinterleaver [37, 38] with low-power consumption and little memory requirements.
4. Regularity [33].
5. Extensionability to higher order permutation polynomial based interleaver [36, 39].

This dissertation is organized as follows. In Chapter 2, we investigate quadratic inverses of QPP. In the beginning of researches on QPP, it was conjectured that most of good QPPs admit quadratic inverses. However, later, good interleavers which do not admit quadratic inverses have found [36, 55]. In Chapter 3, we investigate the inverses of QPP. In Chapter 4, using the results obtained in Chapters 2 and 3, we investigate efficient algorithms that find an upper bound on the minimum distance of turbo codes designed with QPP interleavers. Finally, in Chapter 5, we find the necessary and sufficient condition for a cubic polynomial to be a permutation polynomial.
CHAPTER 2

ON QUADRATIC INVERSES FOR QUADRATIC PERMUTATION POLYNOMIALS OVER INTEGER RINGS

2.1 Introduction

Interleavers for turbo codes [3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 33, 34, 35, 36, 37, 38, 41, 42, 43, 55] have been extensively investigated. Recently, Sun and Takeshita [33] suggested the use of permutation polynomial-based interleavers over integer rings. In particular, quadratic polynomials were emphasized; this quadratic construction is markedly different from and superior to the one proposed earlier by Takeshita and Costello [34]. The algebraic approach was shown to admit analytical design of an interleaver matched to the constituent convolutional codes. The resulting performance was shown to be better than that of S-random interleavers [6, 7] for short to medium block lengths and parallel concatenated turbo codes [33, 35]. Other interleavers [9, 10] better than S-random interleaver for parallel concatenated turbo codes have also been investigated but they are not algebraic.

This work was motivated by work at the Jet Propulsion Laboratory (JPL) [41, 42] for the Mars Laser Communication Demonstration (MLCD). The interleaver in [42] is used in a serially concatenated turbo code. The work in [42] shows that the quadratic
interleavers proposed in [33] can be efficiently implemented in Field-Programmable Gate Array (FPGA) using only additions and comparisons. A turbo decoder needs also a deinterleaver. In [41], the inverse polynomial of a quadratic polynomial is computed by brute force using the fact that permutation polynomials form a group under function composition. It is also shown that the inverse polynomial of a quadratic permutation polynomial may not be quadratic by a particular counterexample. Therefore three natural questions arise: When does a quadratic permutation polynomial over an integer ring have a quadratic inverse polynomial? How do we compute it efficiently? Are there self-inverting quadratic permutation polynomials?

In this chapter, we derive a necessary and sufficient condition for a quadratic permutation polynomial over integer rings to admit a quadratic inverse. The condition consists of simple arithmetic comparisons. In addition, we provide a simple algorithm to compute the inverse polynomial. Further, we argue that this restriction does not seem to effectively narrow the pool of good quadratic interleavers for turbo codes. Finally, we show that self-inverting quadratic permutation polynomials exist. This chapter is organized as follows. In Section 2.2, we briefly review permutation polynomials [45, 46, 47, 48, 49, 50] over the integer ring $\mathbb{Z}_N$ and relevant results. The main result is derived in Section 2.3, and examples are given in Section 2.4. Finally, conclusions are discussed in Section 2.5.

2.2 Permutation Polynomial over Integer Rings

In this section, we revisit the relevant facts about permutation polynomials and other additional results in number theory.

Given an integer $N \geq 2$, a polynomial $\overline{q}(x) = q_0 + q_1 x + q_2 x^2 + \cdots + q_k x^k \pmod{N}$,
where $q_0, q_1, \ldots, q_k$ and $k$ are non-negative integers, is said to be a permutation polynomial over $\mathbb{Z}_N$ when $\overline{q}(x)$ permutes \{0, 1, 2, \ldots, N - 1\} [46, 47, 48, 49, 50]. It is immediate that the constant $q_0$ in $\overline{q}(x)$ only causes a “cyclic shift” to the permuted values. Therefore we define the polynomial $q(x) = \overline{q}(x) - q_0$ without losing generality in our quest for a quadratic inverse polynomial by the following lemma.

**Lemma 1** Suppose that the inverse of a permutation polynomial $q(x)$ is $r(x)$. Then the inverse permutation polynomial of $\overline{q}(x)$ is $r(x - q_0)$. Conversely, suppose that the inverse of a permutation polynomial $\overline{q}(x)$ is $s(x)$. Then the inverse permutation polynomial of $q(x)$ is $s(x + q_0)$.

**Proof:** Suppose the inverse of $q(x)$ is $r(x)$. Then $r(q(x)) = x$. Consequently, $\overline{q}(r(x - q_0)) = q(r(x - q_0)) + q_0 = x - q_0 + q_0 = x$. The other direction can be proved similarly.

Further, it is well known that an inverse permutation polynomial always exists because permutation polynomials forms a group under function composition [41, 48, 49, 50]. The condition for a quadratic polynomial to be a permutation polynomial over $\mathbb{Z}_p$, where $p$ is any prime, is shown in the following two lemmas.

**Lemma 2** ([46]) Let $p = 2$. A polynomial $q(x) = q_1x + q_2x^2 \pmod{p}$ is a permutation polynomial over $\mathbb{Z}_p$ if and only if $q_1 + q_2$ is odd.

**Lemma 3** ([48]) Let $p \neq 2$. A polynomial $q(x) = q_1x + q_2x^2 \pmod{p}$ is a permutation polynomial over $\mathbb{Z}_p$ if and only if $q_1 \not\equiv 0 \pmod{p}$ and $q_2 \equiv 0 \pmod{p}$, i.e., there are no quadratic permutation polynomials modulo a prime $p \neq 2$. 

11
The following theorem and corollary give the necessary and sufficient conditions for a polynomial to be a permutation polynomial over integer ring \( \mathbb{Z}_{p^n} \), where \( p \) is any prime number and \( n \geq 2 \).

**Theorem 4 ([45, 33])** Let \( p \) be a prime number and \( n \geq 2 \). A polynomial \( q(x) = q_1x + q_2x^2 \pmod{p^n} \) is a permutation polynomial over \( \mathbb{Z}_{p^n} \) if and only if \( q_1 \not\equiv 0 \pmod{p} \) and \( q_2 \equiv 0 \pmod{p} \).

**Corollary 5 ([46])** Let \( p = 2 \) and \( n \geq 2 \). A polynomial \( q(x) = q_1x + q_2x^2 \pmod{p^n} \) is a permutation polynomial if and only if \( q_1 \) is odd and \( q_2 \) is even.

Corollary 5 can be easily verified from Theorem 4. However, since our proofs in the Appendix A can be simplified using Corollary 5, we keep it for its simplicity. In this thesis, let the set of primes be \( \mathcal{P} = \{2, 3, 5, \ldots\} \). Then an integer \( N \) can be factored as \( N = \prod_{p \in \mathcal{P}} p^{n_{N,p}} \), where \( p \)'s are distinct primes, \( n_{N,p} \geq 1 \) for a finite number of \( p \) and \( n_{N,p} = 0 \) otherwise. For a quadratic polynomial \( q(x) = q_1x + q_2x^2 \pmod{N} \), we will abuse the previous notation by writing \( q_2 = \prod_{p \in \mathcal{P}} p^{n_{q,p}} \), i.e., the exponents of the prime factors of \( q_2 \) will be written as \( n_{q,p} \) instead of the more cumbersome \( n_{q_2,p} \) because we will only be interested in the factorization of the second degree coefficient.

For a general \( N \), the necessary and sufficient condition for a polynomial to be a permutation polynomial is given in the following theorem.

**Theorem 6 ([33])** For any \( N = \prod_{p \in \mathcal{P}} p^{n_{N,p}} \), \( q(x) \) is a permutation polynomial modulo \( N \) if and only if \( q(x) \) is also a permutation polynomial modulo \( p^{n_{N,p}} \), \( \forall p \) such that \( n_{N,p} \geq 1 \).

Using this theorem, verifying whether a polynomial is a permutation polynomial modulo \( N \) reduces to verifying the polynomial modulo each \( p^{n_{N,p}} \) factor of \( N \).
Corollary 7  Let \( N = \prod_{p \in P} p^{n_{p_N}} \) and denote \( y \) divides \( z \) by \( y \mid z \). The necessary and sufficient condition for a quadratic polynomial \( q(x) = q_1 x + q_2 x^2 \) (mod \( N \)) to be a permutation polynomial can be divided into two cases.

1. \( 2 \mid N \) and \( 4 \nmid N \) (i.e., \( n_{N,2} = 1 \))
   \[ q_1 + q_2 \text{ is odd, } \gcd(q_1, \frac{N}{2}) = 1 \text{ and } q_2 = \prod_{p \in P} p^{n_{H,p}}, n_{q,p} \geq 1, \forall p \text{ such that } p \neq 2 \text{ and } n_{N,p} \geq 1. \]

2. Either \( 2 \nmid N \) or \( 4 \mid N \) (i.e., \( n_{N,2} \neq 1 \))
   \[ \gcd(q_1, N) = 1 \text{ and } q_2 = \prod_{p \in P} p^{n_{H,p}}, n_{q,p} \geq 1, \forall p \text{ such that } n_{N,p} \geq 1. \]

Proof: This is a direct consequence of Lemmas 2, 3, Theorems 4, 6 and Corollary 7.

The following theorem and lemma are also necessary for deriving the main theorem (Theorem 15) in this chapter.

Theorem 8 ([45])  Let \( a, b \) and \( N \) be integers. The linear congruence \( au \equiv b \) (mod \( N \)) has at least one solution if and only if \( d \mid b \), where \( d = \gcd(a, N) \). If \( d \mid b \), then it has \( d \) mutually incongruent solutions. Let \( u_0 \) be one solution, then the set of the solutions is

\[ u_0, u_0 + \frac{N}{d}, u_0 + \frac{2N}{d}, \ldots, u_0 + \frac{(d - 1)N}{d}, \]

where \( u_0 \) is the unique solution of \( \frac{d}{2} u \equiv \frac{b}{d} \) (mod \( \frac{N}{d} \)).

Lemma 9 ([45])  Let \( M \) be an integer. Suppose that \( M \mid N \) and that \( v \equiv w \) (mod \( N \)). Then \( v \equiv w \) (mod \( M \)).

2.3 Quadratic Inverse Polynomial

In this section, we derive the necessary and sufficient condition for a quadratic polynomial to admit at least one quadratic inverse in Theorem 15. We also explicitly
find the quadratic inverse in Algorithm 2.2. If $N = 2$, the inverse polynomial of a quadratic permutation polynomial can be easily constructed. If $N \neq 2$ is a prime number, by Lemma 3, there are no quadratic permutation polynomials. If $N$ is a composite number, it has been shown that the inverse polynomial may not be quadratic by a particular counterexample [41]. However, in the following lemma, it is shown that for any quadratic permutation polynomial, there exists at least one quadratic polynomial that inverts it at three points $x = 0, 1, 2$. The reason why we look at this partially inverting polynomial is because it becomes the basis for the true quadratic inverse polynomial if it exists.

**Lemma 10** Let $N$ be a composite number and let $f(x) = f_1 x + f_2 x^2 \pmod{N}$ be a quadratic permutation polynomial. Then there exists at least one quadratic polynomial $g(x) = g_1 x + g_2 x^2 \pmod{N}$ that inverts $f(x)$ at these three points: $x = 0, 1, 2$. If $N$ is odd, there is exactly one quadratic polynomial $g(x) = g_1 x + g_2 x^2 \pmod{N}$ and the coefficients of the polynomial can be obtained by solving the linear congruences.

\begin{align*}
g_2(f_1 + f_2)(f_1 + 2f_2)(f_1 + 3f_2) & \equiv -f_2 \pmod{N}. \quad (2.1) \\
g_1(f_1 + f_2) + g_2(f_1 + f_2)^2 & \equiv 1 \pmod{N}. \quad (2.2)
\end{align*}

If $N$ is even, there are exactly two quadratic polynomials $g_1(x) = g_{1,1} x + g_{1,2} x^2 \pmod{N}$, $g_2(x) = g_{2,1} x + g_{2,2} x^2 \pmod{N}$ and the coefficients of the polynomial $g_1(x) = g_{1,1} x + g_{1,2} x^2 \pmod{N}$ can be obtained by solving the linear congruences.

\begin{align*}
g_{1,2}(f_1 + f_2)(f_1 + 2f_2)(f_1 + 3f_2) & \equiv -f_2 \pmod{\frac{N}{2}}. \quad (2.3) \\
g_{1,1}(f_1 + f_2) + g_{1,2}(f_1 + f_2)^2 & \equiv 1 \pmod{N}. \quad (2.4)
\end{align*}

After computing $(g_{1,1}, g_{1,2})$, $(g_{2,1}, g_{2,2})$ can be obtained by $g_{2,1} \equiv g_{1,1} + \frac{N}{2} \pmod{N}$ and $g_{2,2} \equiv g_{1,2} + \frac{N}{2} \pmod{N}$.
Each of the above four linear congruences (2.1), (2.2), (2.3) and (2.4) are guaranteed to have exactly one solution by Lemma 10 and Theorem 8. It is well known that linear congruences can be efficiently solved by using the extended Euclidean algorithm [44]. For example, in congruence (2.2), the unknown value to be solved is \( g_1; f_1 \) and \( f_2 \) are given and \( g_2 \) can be calculated from (2.1). The congruence (2.2) can be rewritten as

\[
g_1 \equiv (f_1 + f_2)^* \cdot (1 - g_2(f_1 + f_2)^2) \pmod{N},
\]

where \((f_1 + f_2)^*\) means the arithmetic inverse of \((f_1 + f_2) \pmod{N}\). By an arithmetic inverse of \(s\) modulo \(N\), we mean a number \(s^*\) such that \(ss^* \equiv 1 \pmod{N}\). The Algorithm 2.3 provided in Table 2.2 can be used to calculate such an inverse.

In the following lemma, we show that the polynomials \(g(x), g_1(x)\) and \(g_2(x)\) obtained by solving the congruences (2.1), (2.2), (2.3) and (2.4) are permutation polynomials.

**Lemma 11** The polynomials \(g(x), g_1(x)\) and \(g_2(x)\) obtained in Lemma 10 are permutation polynomials.

**Proof:** See Appendix A.2.

From Lemmas 10 and 11, there exists at least one quadratic permutation polynomial \(g(x)\) that inverts any quadratic permutation polynomial \(f(x)\) at three points \(x = 0, 1, 2\). However, it does not necessarily mean that \(g(x)\) is an inverse polynomial of \(f(x)\).

In the following lemma, we show that some exponents \(n_{g,p}\)'s of the \(g_2\) which was obtained in Lemma 10 are determined by the exponents \(n_{f,p}\)'s.

**Lemma 12** Let \(N = \prod_{p \in \mathbb{P}} p^{n_{p,N}}\), \(f(x) = f_1x + f_2x^2 \pmod{N}\) be a quadratic permutation polynomial and \(g(x) = g_1x + g_2x^2 \pmod{N}\) be a quadratic permutation polynomial.
polynomial in Lemmas 10 and 11. Then, \( f_2 = \prod_{p \in P} p^{n_{f,p}} \) and \( g_2 = \prod_{p \in P} p^{n_{g,p}} \) satisfy Corollary 7. Furthermore, the following holds.

case a: \( 2 \nmid N \) (i.e., \( n_{N,2} = 0 \))

If \( N \) contains \( p \) as a factor (i.e., \( n_{N,p} \geq 1 \)) then

\[
\begin{cases}
  n_{g,p} = n_{f,p} & \text{if } 1 \leq n_{f,p} < n_{N,p} \\
  n_{g,p} \geq n_{N,p} & \text{if } n_{f,p} \geq n_{N,p}
\end{cases}
\]

case b: \( 2 \mid N \) and \( 4 \nmid N \) (i.e., \( n_{N,2} = 1 \))

\( N \) contains \( p = 2 \) as a factor but we do not need to consider how \( n_{g,2} \) is determined by \( n_{f,2} \). The reason for this is explained in the proof of Theorem 15.

If \( p \neq 2 \) and \( N \) contains \( p \) as a factor (i.e., \( n_{N,p} \geq 1 \)) then

\[
\begin{cases}
  n_{g,p} = n_{f,p} & \text{if } 1 \leq n_{f,p} < n_{N,p} \\
  n_{g,p} \geq n_{N,p} & \text{if } n_{f,p} \geq n_{N,p}
\end{cases}
\]

case c: \( 4 \mid N \) (i.e., \( n_{N,2} \geq 2 \))

\( N \) contains \( 2^2 \) as a factor (i.e., \( n_{N,2} \geq 2 \)).

If \( p = 2 \),

\[
\begin{cases}
  n_{g,2} = n_{f,2} & \text{if } 1 \leq n_{f,2} < n_{N,2} - 1 \\
  n_{g,2} \geq n_{N,2} - 1 & \text{if } n_{f,2} \geq n_{N,2} - 1
\end{cases}
\]

If \( p \neq 2 \) and \( N \) contains \( p \) as a factor (i.e., \( n_{N,p} \geq 1 \)) then

\[
\begin{cases}
  n_{g,p} = n_{f,p} & \text{if } 1 \leq n_{f,p} < n_{N,p} \\
  n_{g,p} \geq n_{N,p} & \text{if } n_{f,p} \geq n_{N,p}
\end{cases}
\]

Proof: See Appendix A.3.

Before proceeding further, we need the following lemma.

**Lemma 13** Let \( t(x) = t_1 x + t_2 x^2 + t_3 x^3 + t_4 x^4 \) (mod \( N \)) and \( t(0) \equiv t(1) \equiv t(2) \equiv 0 \) (mod \( N \)). Then \( t(x) \equiv 0 \) (mod \( N \), \( \forall x \in [0, N - 1] \)) if and only if \( 24 t_4 \equiv 0 \) (mod \( N \)) and \( 6 t_3 + 36 t_4 \equiv 0 \) (mod \( N \)).
Proof: See Appendix A.4.

Combining Lemmas 10 and 13 gives the following theorem.

**Theorem 14** Let $f(x)$ be a quadratic permutation polynomial and let $g(x)$ be a quadratic polynomial in Lemma 10. Then $g(x)$ is a quadratic inverse polynomial of $f(x)$ if and only if $12f_2g_2 \equiv 0 \pmod{N}$.

Proof: See Appendix A.5.

We now state our main theorem. It states that the necessary and sufficient condition for the existence of a quadratic inverse for a quadratic permutation polynomial $f(x)$ can be simply checked by inequalities involving the exponents for the prime factors of $N$ and the second degree coefficient of $f(x)$.

**Theorem 15 (main Theorem)** Let $N = \prod_{p \in \mathbb{P}} p^{n_{N,p}}$, $f(x)$ be a quadratic permutation polynomial and $f_2 = \prod_{p \in \mathbb{P}} p^{n_{f,p}}$ be the second degree coefficient of $f(x)$. Then $f(x)$ has at least one quadratic inverse polynomial if and only if

$$n_{f,2} \geq \max \left( \left\lceil \frac{n_{N,2}-2}{2} \right\rceil, 1 \right) \quad \text{if} \quad n_{N,2} > 1$$

$$0 \quad \text{if} \quad n_{N,2} = 0, 1$$

$$n_{f,3} \geq \max \left( \left\lceil \frac{n_{N,3}-1}{2} \right\rceil, 1 \right) \quad \text{if} \quad n_{N,3} > 0$$

$$0 \quad \text{if} \quad n_{N,3} = 0$$

$$n_{f,p} \geq \left\lceil \frac{n_{N,p}}{2} \right\rceil \quad \text{if} \quad p \neq 2, 3.$$

Proof: See Appendix A.6.

An interesting question of practical significance is if an interleaver can be its own inverse [34] because the same hardware can be used for both interleaving and deinterleaving. It is shown in [34] that this type of restriction do not affect turbo decoding performance using interleavers therein proposed. We identified six self-inverting quadratic polynomials in [55] and summarized them in Table 2.1.
<table>
<thead>
<tr>
<th>length</th>
<th>QPP</th>
</tr>
</thead>
<tbody>
<tr>
<td>48</td>
<td>$7x + 12x^2$</td>
</tr>
<tr>
<td>176</td>
<td>$21x + 44x^2$</td>
</tr>
<tr>
<td>240</td>
<td>$29x + 60x^2$</td>
</tr>
<tr>
<td>304</td>
<td>$37x + 76x^2$</td>
</tr>
<tr>
<td>864</td>
<td>$17x + 48x^2$</td>
</tr>
<tr>
<td>2688</td>
<td>$127x + 504x^2$</td>
</tr>
</tbody>
</table>

Table 2.1: Self-inverting quadratic polynomials for 3GPP LTE interleavers.

### 2.3.1 Algorithms for Finding the Quadratic Inverse Polynomials

Algorithm 2.2 is provided in Table 2.2. It finds the quadratic inverse polynomial for a given quadratic permutation polynomial $f(x) = f_1x + f_2x^2 \pmod{N}$. In Table 2.2, Algorithm 2.3 is provided to calculate the arithmetic inverse of $s \pmod{M}$, which is required in Algorithm 2.2.

### 2.4 Examples

We present three examples to illustrate the necessary and sufficient conditions of Theorem 15. The first example considers interleavers investigated in [42]. The second example is a generalization of an example in [41]. The third example shows that the verification procedure simplifies when $N$ is a power of 2, as it was chosen in [33], for a fair comparison with [34]. Quadratic inverses of 3GPP LTE interleavers are shown in Tables 2.4 and 2.5. Among 188 interleavers in [55], 153 interleavers admit quadratic inverses.

**Example 1:** Let $N = 15120 = 2^4 \cdot 3^3 \cdot 5 \cdot 7$, $f_1 \equiv 11 \pmod{15120}$ and $f_2 \equiv 2 \cdot 3 \cdot 5 \cdot 7 \cdot m \equiv 210m \pmod{15120}$, where $m$ is any non-negative integer. Let $m = 1$. 

18
An algorithm for finding the quadratic inverse permutation polynomial(s) for a quadratic permutation polynomial \( f(x) = f_1 x + f_2 x^2 \pmod{N} \)

1. Factor \( N \) and \( f_2 \) as products of prime powers and find the respective exponents of each prime factor, i.e., find \( n_{N, p} \)'s and \( n_{f_2, p} \)'s for \( N = \prod_{p \in P} p^{n_{N, p}} \), \( f_2 = \prod_{p \in P} p^{n_{f_2, p}} \), respectively. We can also get \( \left\lceil \frac{N}{p} \right\rceil \) and \( \left\lceil \frac{f_2}{p} \right\rceil \) has two quadratic inverse polynomials. By Algorithms 2.2 and 2.3, we can get \( g_{1, 1} \equiv 14891 \pmod{15120} \) and \( g_{1, 2} \equiv 210 \pmod{15120} \), respectively. We can also get \( g_{2, 1} \equiv g_{1, 1} + \frac{15120}{2} \equiv 7331 \pmod{15120} \) and \( g_{2, 2} \equiv g_{1, 2} + \frac{15120}{2} \equiv 7770 \pmod{15120} \) by Algorithm 2.2.

If \( m > 1 \), there are also two inverses since \( m \) only increases \( n_{f_2, p} \), for some

### Table 2.2: Algorithm : computation of quadratic inverses.

| \( n_{f_2} \) | \( n_{f_2} = 1 \geq \max(\lceil \frac{4 - 2}{2} \rceil, 1) \) | \( n_{f, 3} = 1 \geq \max(\lceil \frac{3 - 1}{2} \rceil, 1) \) | \( n_{f, 5} = 1 \geq \lceil \frac{1}{2} \rceil \) | \( n_{f, 7} = 1 \geq \lceil \frac{1}{2} \rceil \) | \( 15120 \) is even, by Lemma 10 and Theorem 15, \( f(x) \) has two quadratic inverse polynomials. By Algorithms 2.2 and 2.3, we can get \( g_{1, 1} \equiv 14891 \pmod{15120} \) and \( g_{1, 2} \equiv 210 \pmod{15120} \), respectively. We can also get \( g_{2, 1} \equiv g_{1, 1} + \frac{15120}{2} \equiv 7331 \pmod{15120} \) and \( g_{2, 2} \equiv g_{1, 2} + \frac{15120}{2} \equiv 7770 \pmod{15120} \) by Algorithm 2.2.

}\( m > 1 \), there are also two inverses since \( m \) only increases \( n_{f_2, p} \), for some
An algorithm for finding the arithmetic inverse $s^*$ for $s \pmod{M}$

\[
s^* = 1; \\
r = 0; \\
while (M \neq 0) \\
    c \equiv s \pmod{M}; \\
    quot = \lfloor \frac{s}{M} \rfloor; \\
    s = M; \\
    M = c; \\
    r' = s^* - quot \cdot r; \\
    s^* = r; \\
    r = r'; \\
end \\
Return s^*
\]

Table 2.3: Algorithm: computation of an inverse of an element over integer ring.

$p$'s. Thus, regardless of the values $m$ and $f_1$, there exist two quadratic inverse polynomials for $f(x)$.

Example 2: Let $N = 5^3$ and $f_2 \equiv 5m \pmod{5^3}$, where $m$ is an integer such that $5 \nmid m$. In this case, regardless of the values $m$ and $f_1$, there are no quadratic inverse polynomial, since $n_{f,5} = 1 \nleq \lceil \frac{3}{2} \rceil$. However, if $f_2 \equiv 5^2 m \pmod{5^3}$, where $5 \nmid m$, regardless of $m$ and $f_1$, there exists one quadratic inverse polynomial since $5^3$ is odd and $n_{f,5} = 2 \geq \lceil \frac{3}{2} \rceil$.

Example 3: Let $N = 2^{10}$ and $f_2 \equiv 2^4 \pmod{2^{10}}$. In this case, regardless of the value $f_1$, there exist two inverses since $2^{10}$ is even and $n_{f,2} = 4 \geq \max(\lceil \frac{10-2}{2} \rceil, 1)$. Specifically, if $f_1$ is 1, the two inverses are $g_1(x) = x + 496x^2 \pmod{2^{10}}$ and $g_2(x) = 513x + 1008x^2 \pmod{2^{10}}$, and if $f_1$ is 15, the two inverses are $g_1(x) = 751x + 272x^2 \pmod{2^{10}}$ and $g_2(x) = 239x + 784x^2 \pmod{2^{10}}$, respectively.
<table>
<thead>
<tr>
<th>length</th>
<th>QPP</th>
<th>Inverse QPP</th>
</tr>
</thead>
<tbody>
<tr>
<td>2016</td>
<td>$113x + 420x^2$</td>
<td>$6x + 785x^2$</td>
</tr>
<tr>
<td>2048</td>
<td>$31x + 64x^2$</td>
<td>$991x + 64x^2$</td>
</tr>
<tr>
<td>2176</td>
<td>$171x + 136x^2$</td>
<td>$1667x + 680x^2$</td>
</tr>
<tr>
<td>2240</td>
<td>$209x + 420x^2$</td>
<td>$1329x + 700x^2$</td>
</tr>
<tr>
<td>2304</td>
<td>$253x + 216x^2$</td>
<td>$1621x + 648x^2$</td>
</tr>
<tr>
<td>2368</td>
<td>$367x + 444x^2$</td>
<td>$1455x + 444x^2$</td>
</tr>
<tr>
<td>2432</td>
<td>$265x + 456x^2$</td>
<td>$569x + 760x^2$</td>
</tr>
<tr>
<td>2496</td>
<td>$181x + 468x^2$</td>
<td>$2365x + 156x^2$</td>
</tr>
<tr>
<td>2560</td>
<td>$39x + 80x^2$</td>
<td>$2199x + 720x^2$</td>
</tr>
<tr>
<td>2624</td>
<td>$27x + 164x^2$</td>
<td>$243x + 820x^2$</td>
</tr>
<tr>
<td>2688</td>
<td>$127x + 504x^2$</td>
<td>$127x + 504x^2$</td>
</tr>
<tr>
<td>2752</td>
<td>$143x + 172x^2$</td>
<td>$2319x + 172 + x^2$</td>
</tr>
<tr>
<td>2816</td>
<td>$43x + 88x^2$</td>
<td>$1539x + 440x^2$</td>
</tr>
<tr>
<td>2880</td>
<td>$29x + 300x^2$</td>
<td>$1589x + 1380x^2$</td>
</tr>
<tr>
<td>3008</td>
<td>$157x + 188x^2$</td>
<td>$661x + 564x^2$</td>
</tr>
<tr>
<td>3072</td>
<td>$47x + 96x^2$</td>
<td>$2255x + 96x^2$</td>
</tr>
<tr>
<td>3136</td>
<td>$13x + 28x^2$</td>
<td>$965x + 1204x^2$</td>
</tr>
<tr>
<td>3200</td>
<td>$111x + 240x^2$</td>
<td>$591x + 560x^2$</td>
</tr>
<tr>
<td>3264</td>
<td>$443x + 204x^2$</td>
<td>$851x + 1020x^2$</td>
</tr>
<tr>
<td>3328</td>
<td>$51x + 104x^2$</td>
<td>$3067x + 1352x^2$</td>
</tr>
<tr>
<td>3392</td>
<td>$51x + 212x^2$</td>
<td>$3259x + 1060x^2$</td>
</tr>
<tr>
<td>3456</td>
<td>$451x + 192x^2$</td>
<td>$811x + 1536x^2$</td>
</tr>
<tr>
<td>3520</td>
<td>$257x + 220x^2$</td>
<td>$2753x + 1540x^2$</td>
</tr>
<tr>
<td>3584</td>
<td>$57x + 336x^2$</td>
<td>$1289x + 560x^2$</td>
</tr>
<tr>
<td>3648</td>
<td>$313x + 228x^2$</td>
<td>$169x + 1596x^2$</td>
</tr>
<tr>
<td>3712</td>
<td>$271x + 232x^2$</td>
<td>$2671x + 232x^2$</td>
</tr>
<tr>
<td>3776</td>
<td>$179x + 236x^2$</td>
<td>$443x + 1180x^2$</td>
</tr>
<tr>
<td>3840</td>
<td>$331x + 120x^2$</td>
<td>$3811x + 600x^2$</td>
</tr>
<tr>
<td>3904</td>
<td>$363x + 244x^2$</td>
<td>$3619x + 1220x^2$</td>
</tr>
<tr>
<td>3968</td>
<td>$375x + 248x^2$</td>
<td>$2439x + 248x^2$</td>
</tr>
</tbody>
</table>

Table 2.4: Quadratic inverse polynomials for 3GPP LTE interleavers for lengths 2000 to 4000

### 2.5 Conclusion

We derived in Theorem 15 a necessary and sufficient condition for the existence of a quadratic inverse for a quadratic permutation polynomial over integer rings.
Further, we described a simple algorithm (Algorithm 2.2) to find the coefficients of the quadratic inverse polynomial. We also found that almost all good interleavers searched in [33] admit a quadratic inverse despite the fact that they were not designed with this remarkable property in mind. Also we have found that 153 interleavers among 188 3GPP LTE interleavers [55] admit a quadratic inverse.

Table 2.5: Quadratic inverse polynomials for 3GPP LTE interleavers for lengths over 4000

<table>
<thead>
<tr>
<th>length</th>
<th>QPP</th>
<th>Inverse QPP</th>
</tr>
</thead>
<tbody>
<tr>
<td>4032</td>
<td>$127x + 168x^2$</td>
<td>$2143x + 840x^2$</td>
</tr>
<tr>
<td>4096</td>
<td>$31x + 64x^2$</td>
<td>$991x + 64x^2$</td>
</tr>
<tr>
<td>4224</td>
<td>$43x + 264x^2$</td>
<td>$2947x + 1320x^2$</td>
</tr>
<tr>
<td>4288</td>
<td>$33x + 134x^2$</td>
<td>$1137x + 1474x^2$</td>
</tr>
<tr>
<td>4352</td>
<td>$477x + 408x^2$</td>
<td>$885x + 1224x^2$</td>
</tr>
<tr>
<td>4416</td>
<td>$35x + 138x^2$</td>
<td>$2555x + 138x^2$</td>
</tr>
<tr>
<td>4480</td>
<td>$233x + 280x^2$</td>
<td>$4057x + 1960x^2$</td>
</tr>
<tr>
<td>4544</td>
<td>$357x + 142x^2$</td>
<td>$4509x + 3266x^2$</td>
</tr>
<tr>
<td>4608</td>
<td>$337x + 480x^2$</td>
<td>$4273x + 1824x^2$</td>
</tr>
<tr>
<td>4800</td>
<td>$71x + 120x^2$</td>
<td>$2231x + 1080x^2$</td>
</tr>
<tr>
<td>4864</td>
<td>$37x + 152x^2$</td>
<td>$4141x + 1672x^2$</td>
</tr>
<tr>
<td>4928</td>
<td>$39x + 158x^2$</td>
<td>$551x + 4582x^2$</td>
</tr>
<tr>
<td>5120</td>
<td>$39x + 80x^2$</td>
<td>$919x + 2000x^2$</td>
</tr>
<tr>
<td>5184</td>
<td>$31x + 96x^2$</td>
<td>$4543x + 1632x^2$</td>
</tr>
<tr>
<td>5312</td>
<td>$41x + 166x^2$</td>
<td>$745x + 498x^2$</td>
</tr>
<tr>
<td>5376</td>
<td>$251x + 336x^2$</td>
<td>$2099x + 1680x^2$</td>
</tr>
<tr>
<td>5440</td>
<td>$43x + 170x^2$</td>
<td>$3827x + 4250x^2$</td>
</tr>
<tr>
<td>5508</td>
<td>$43x + 174x^2$</td>
<td>$3507x + 4350x^2$</td>
</tr>
<tr>
<td>5632</td>
<td>$45x + 176x^2$</td>
<td>$4005x + 528x^2$</td>
</tr>
<tr>
<td>5760</td>
<td>$161x + 120x^2$</td>
<td>$161x + 840x^2$</td>
</tr>
<tr>
<td>5824</td>
<td>$89x + 182x^2$</td>
<td>$409x + 546x^2$</td>
</tr>
<tr>
<td>5888</td>
<td>$323x + 184x^2$</td>
<td>$3819x + 2392x^2$</td>
</tr>
<tr>
<td>5952</td>
<td>$47x + 186x^2$</td>
<td>$3071x + 930x^2$</td>
</tr>
<tr>
<td>6144</td>
<td>$263x + 480x^2$</td>
<td>$2231x + 2784x^2$</td>
</tr>
</tbody>
</table>
An interesting question of practical significance is if an interleaver can be its own inverse [34]. We have found six self-inverting interleavers in 3GPP LTE interleavers [55].
3.1 Introduction

Interleavers for turbo codes have been extensively investigated \([3, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 33, 34, 35, 36]\). Today the focus on interleaver constructions is not only for good error correction performance of the corresponding turbo codes but also for their hardware efficiency with respect to power consumption and speed. Recently, permutation polynomial-based interleavers over integer rings have been suggested in \([33]\). In particular, QPPs were emphasized because of their simple construction and analysis. Their performance was shown to be excellent \([33, 35, 36]\). The practical suitability of QPP interleavers has been considered in a deep space application \([42]\) and in 3GPP long term evolution (LTE) \([55]\).

The inverse function for a QPP is also a permutation polynomial but is not necessarily a QPP \([42]\). However, there exists a simple criterion for a QPP to admit a QPP inverse \([37]\). A simple rule for finding good QPPs has been suggested in \([36]\). Some examples in \([36]\) do not have QPP inverses. Most of QPP interleavers proposed in 3GPP LTE \([55]\) admit a quadratic inverse with the exception of 35 of them. In this chapter, we extend the result in \([37]\) and provide a necessary and sufficient condition.
that determines the least degree inverse of a QPP. The condition consists of simple arithmetic comparisons. We further provide an algorithm to explicitly find the inverse polynomials.

This chapter is organized as follows. In Section 3.2, we briefly review permutation polynomials [45, 46, 51, 52] over the integer ring $\mathbb{Z}_N$ and relevant results. The main result is derived in Section 3.3, and examples are given in Section 3.4. Finally, conclusions are discussed in Section 3.5.

3.2 Permutation Polynomial over Integer Rings

The following definitions will be used in the rest of this chapter.

Definition 16 ([51, 52]) Two polynomials $p(x) = \sum_{k=1}^{K} p_k x^k$ and $q(x) = \sum_{k=1}^{K} q_k x^k$ of degree $K$ are congruent polynomials modulo $N$ if $p_k \equiv q_k \pmod{N}$, $\forall k \in \{1, ..., K\}$.

Definition 17 ([51, 52]) Two polynomials $q(x)$ and $\tilde{q}(x)$ are equivalent polynomials modulo $N$ if $q(x) \equiv \tilde{q}(x) \pmod{N}$, $\forall x \in \{0, 1, ..., N-1\}$.

Definition 18 ([51, 52]) A polynomial $\hat{q}(x)$ of degree $K$ is a null polynomial of degree $K$ modulo $N$ if $\hat{q}(x) \equiv 0 \pmod{N}$, $\forall x \in \{0, 1, ..., N-1\}$. Specially, $\hat{q}(x) = 0$ is a trivial null polynomial of degree 0 modulo $N$.

Note that congruent polynomials modulo $N$ are equivalent polynomials modulo $N$, although the converse is not always true.

Corollary 19 ([51, 52]) If two equivalent polynomials $q(x)$ and $\tilde{q}(x)$ are not congruent polynomials modulo $N$, there exists a non-trivial null polynomial $\hat{q}(x)$ with degree $K \geq 1$ such that $q(x) - \tilde{q}(x) \equiv \hat{q}(x) \pmod{N}$.
Let $2|N$. A typical example of non-trivial null polynomial is $q(x) = \frac{N}{2}x + \frac{N}{2}x^2 \pmod{N}$.

**Definition 20** Let $q(x) = \sum_{k=1}^{K} q_k x^k \pmod{N}$ and let us denote the least degree of equivalent polynomials of $q(x)$ by $\deg_L\{q(x)\}$. If a polynomial $q(x)$ does not have an equivalent polynomial with degree less than $K$, $\deg_L\{q(x)\} = \deg\{q(x)\} = K$ and $\deg_L\{q(x)\} < \deg\{q(x)\} = K$, otherwise. Let us denote $q^{(L)}(x) = \sum_{k=1}^{L} q_k x^k \pmod{N}$ if $\deg_L\{q(x)\} = \deg\{q(x)\}$.

The following lemma was proposed in [51, 52]. However, the proof is shown for its simplicity.

**Lemma 21 ([51, 52])** There is an equivalent polynomial with $\deg_L\{q(x)\} \leq N - 1$ for a polynomial $q(x) = \sum_{k=1}^{K} q_k x^k \pmod{N}$ if $K \geq N$.

**Proof:** Let $\hat{q}(x) = q_K \cdot \left[ \prod_{k=0}^{N-1} (x - k) \right] \cdot x^{K-N}$. Then $\hat{q}(x) \equiv 0 \pmod{N}, \forall x \in \{0, 1, ..., N - 1\}$. Let $\tilde{q}(x) = q(x) - \hat{q}(x)$, then $\tilde{q}(x) \equiv q(x)$ but $\deg_L\{\tilde{q}(x)\} < \deg_L\{q(x)\}$. By applying Lemma 21 repeatedly, we can get a polynomial with least degree less or equal to $N - 1$.

In the following lemma, the number of equivalent polynomials of a quadratic permutation polynomial is shown.

**Lemma 22** If $N$ is odd, then there exists no other equivalent polynomial for a quadratic permutation polynomial and if $N$ is even, then there exists one equivalent polynomial which is not congruent with the quadratic permutation polynomial.

**Proof:** The statement is equivalent to the following statement.

\[
\begin{align*}
\begin{cases}
N \text{ is odd: } \hat{q}(x) = \hat{q}_1 x + \hat{q}_2 x^2 \equiv 0 & \text{if and only if } \hat{q}_1, \hat{q}_2 \equiv 0 \\
N \text{ is even: } \hat{q}(x) = \hat{q}_1 x + \hat{q}_2 x^2 \equiv 0 & \text{if and only if } \hat{q}_1, \hat{q}_2 \equiv 0 \text{ or } \hat{q}_1, \hat{q}_2 \equiv \frac{N}{2}
\end{cases}
\end{align*}
\]
i.e., there only exists a trivial null polynomial when \( N \) is odd and there exists a non-trivial null polynomial \( \frac{N}{2}x + \frac{N}{2}x^2 \) when \( N \) is even.

The proof is as follows. \( \hat{q}(x) = \hat{q}_1x + \hat{q}_2x^2 \equiv 0 \pmod{N} \) if and only if \( \hat{q}(0) \equiv 0 \pmod{N} \) and \( \hat{q}(x+1) - \hat{q}(x) = 2\hat{q}_2x + \hat{q}_1 + \hat{q}_2 \equiv 0 \pmod{N} \), \( \forall x \in \{0, 1, ..., N-1\} \).

From \( \hat{q}(1) - \hat{q}(0) = \hat{q}_1 + \hat{q}_2 \equiv 0 \pmod{N} \) and \( \hat{q}(2) - \hat{q}(1) = \hat{q}_1 + 3\hat{q}_2 \equiv 0 \pmod{N} \), it is easily shown that \( 2\hat{q}_2 \equiv 0 \pmod{N} \). If \( N \) is odd, \( \hat{q}_1 \equiv 0 \pmod{N} \), \( \hat{q}_2 \equiv 0 \pmod{N} \) and if \( N \) is even, \( \hat{q}_1 \equiv 0, \frac{N}{2} \pmod{N} \), \( \hat{q}_2 \equiv 0, \frac{N}{2} \pmod{N} \), respectively.

In Lemma 34, the necessary and sufficient condition for a polynomial with degree \( K \) to be a null polynomial is shown. The following lemma will be used in the rest of this chapter.

**Lemma 23** Let \( q_1, q_2, N \) be integers in Corollary 7 and let \( l \) be an integer. Let us take odd \( q_1 \) when \( n_{N,2} = 1 \). Then \( \gcd(q_1 + lq_2, N) = 1 \), \( \forall l \).

**Proof:**

1. Either \( 2 \nmid N \) or \( 4 \mid N \) (i.e., \( n_{N,2} \neq 1 \))

   Suppose that \( \gcd(q_1 + lq_2, N) \neq 1 \). Then there exists a prime \( p \) such that \( p \mid (q_1 + lq_2) \) and \( p \mid N \). By Corollary 7, if \( p \mid N \), then \( p \mid lq_2 \) but \( p \nmid q_1 \). A contradiction.

2. \( 2 \mid N \) and \( 4 \nmid N \) (i.e., \( n_{N,2} = 1 \))

   There are two equivalent quadratic polynomials. Take a polynomial with odd \( q_1 \). Then by Corollary 7 and similar argument in (1), \( \gcd(q_1 + lq_2, N) = 1 \).

By the Lemma 23, it is shown that the inverse of \( (q_1 + lq_2) \) exists when \( q_1 \) is odd and \( n_{N,2} = 1 \). Since we only need to find the inverse of one of the equivalent polynomials, it does not impose any restriction even if we assume \( q_1 \) is odd when \( n_{N,2} = 1 \). We only consider the polynomial with odd \( q_1 \) when \( n_{N,2} = 1 \), since the existence of the inverse
of $q_1 + lq_2$ will be used in the rest of the correspondence. The following corollary will be used in Lemma 33.

**Corollary 24** Let $q_1$, $q_2$, $N$ be the integers in Corollary 7 and $l$, $k$ be integers. Let us take odd $q_1$ when $n_{N,2} = 1$. Then \( \gcd \left\{ \prod_{i=1}^{k} \prod_{l=1}^{2i-1} (q_1 + lq_2), N \right\} = 1, \forall k \geq 1. \)

**Proof:** This is a direct consequence of Lemma 23.

### 3.3 Inverses of Quadratic Permutation Polynomials

In this section, we derive the necessary and sufficient condition for a quadratic polynomial to admit a least degree inverse in Theorem 43. We also explicitly find the inverse in Algorithm 3.1.

In the following lemma, we identify the problem of finding an inverse with solving a system of linear congruences.

**Lemma 25** Let \( f(x) = f_1x + f_2x^2 \mod N \) be a permutation polynomial. Then there exists a polynomial \( g^{(L)}(x) \) such that \( L < N \) and \( (g^{(L)} \circ f)(x) \equiv x \mod N \), \( \forall x \in \{0, 1, ..., N - 1\} \). Further, finding \( g^{(L)}(x) \) is equivalent to solving a system of linear congruences,

\[
Ag^{(L)} \equiv b \mod N,
\]

where

\[
a_{i,j} = (if_1 + if_2)^j, 1 \leq i, j \leq N - 1,
\]

\[
g^{(L)} = [g_1, g_2, ..., g_L, 0, ..., 0]^T \quad \text{and} \quad b = [b_1, b_2, ..., b_{N-1}]^T = [1, 2, ..., N - 1]^T.
\]

**Proof:** Since the set of permutation polynomials forms a group under function composition, the existence of an inverse permutation polynomial for a quadratic permutation polynomial is guaranteed. Let \( g(x) \) be an inverse polynomial of \( f(x) \) and
suppose that \( \text{deg}\{g(x)\} \geq N \). Then by Lemma 21, it can be reduced to an equivalent polynomial with degree less than \( N \).

Since the number of permutation polynomials with degree up to \( N - 1 \) is finite, there exists an inverse polynomial \( g^{(L)}(x) \) of \( f(x) \), where \( L < N \).

The equivalence of \( (g^{(L)} \circ f)(x) \equiv x \) and \( A g^{(L)} \equiv b \) is shown by evaluating \( (g^{(L)} \circ f)(x) = \sum_{k=1}^{N-1} g_k(f_1 x + f_2 x^2)^k \equiv x \pmod N \) at each point \( x \in \{1, 2, ..., N - 1\} \). Note that \( (g^{(L)} \circ f)(0) \equiv 0 \) trivially holds.

From now on, we identify \( g^{(L)} = [g_1, g_2, ..., g_L, 0, ..., 0]^T \) with \( g^{(L)}(x) = \sum_{k=1}^{L} g_k x^k \pmod N \) by Lemma 25.

Following definitions and theorem will be used in the proof of Theorem 30.

**Definition 26 ([53])** A principal ideal ring \( \mathfrak{B} \) is a commutative unit ring in which every ideal is principal.

An integer ring \( \mathbb{Z}_N \) is a principal ideal ring. Consequently, definitions and theorems that are applied to a principal ideal ring \( \mathfrak{B} \) can be also applied to an integer ring \( \mathbb{Z}_N \).

**Definition 27 ([53])** A matrix \( M \) with elements in \( \mathfrak{B} \) is a unit if there exists a matrix \( M' \) with elements in \( \mathfrak{B} \) such that \( MM' = I \).

**Theorem 28 ([53])** \( M \) is a unit if and only if \( \text{det}(M) \) is a unit.

**Definition 29 ([53])** An unimodular matrix is a unit with elements in \( \mathfrak{B} \).

Before proceeding further, we show the following theorem which will be used for proving Lemma 33.

**Theorem 30 ([53, 54])** Let \( A \) be an \( N - 1 \) by \( N - 1 \) matrix with elements in \( \mathfrak{B} \).

Then there exist unimodular matrices \( U \) and \( V \) with elements in \( \mathfrak{B} \), \( U \) being \( N - 1 \) by
$N - 1$ and $V$ being $N - 1$ by $N - 1$, such that $UEV = A$, where $E$ is in Smith normal form, with zero elements everywhere except in the main diagonal where there may appear non-zero elements $e_1, e_2, ..., e_r$ having the property that $e_k | e_{k+1}$ and $r \leq N - 1$.

In the following corollary, it is shown that solving $Ag \equiv b \pmod{N}$ is equivalent to solving $Eh \equiv d \pmod{N}$, where $h = Vg$ and $d = U^{-1}b$ by using Theorem 30.

**Corollary 31 ([58])** $Ag \equiv b \pmod{N}$ is solvable if and only if $e_k \cdot h_k \equiv d_k \pmod{N}$, $\forall k$ is solvable, where $h = Vg$ and $d = U^{-1}b$.

**Proof:** Let $\alpha$ and $\beta$ be $N - 1$ by 1 matrices with elements in $\mathbb{Z}_N$. Then by Theorem 30,

$$
\begin{align*}
Ag \equiv b \pmod{N} \iff& \quad Ag = b + N \cdot \alpha \\
& \iff \quad UEVg = b + N \cdot \alpha \\
& \iff \quad EVg = U^{-1}b + N \cdot U^{-1} \alpha \\
& \iff \quad Eh = d + N \cdot \beta \\
& \iff \quad Eh \equiv d \pmod{N} \\
& \iff \quad e_k \cdot h_k \equiv d_k \pmod{N}, \forall k.
\end{align*}
$$

In the following corollary, it is shown that the degrees of $h$ and $g$ in Corollary 31 are equal.

**Corollary 32 ([54])** The degree of $h^{(L)}$ in $Eh^{(L)} \equiv d \pmod{N}$ and the degree of $g^{(L)}$ in $Ag^{(L)} \equiv b \pmod{N}$ are equal. Further, the number of the solutions of $Eh^{(L)} \equiv d \pmod{N}$ are equal to those of $Ag^{(L)} \equiv b \pmod{N}$.
Since $A$ in Lemma 25 has elements in $\mathbb{Z}_N$, there exist matrices $U$, $E$ and $V$ with elements in $\mathbb{Z}_N$ as in Theorem 30. In the following lemma, we find $U$, $E$ and $V$ in Theorem 30.

**Lemma 33** Suppose $A$ is a matrix in Lemma 25. Then the following $U$, $E$ and $V$ satisfy the equation $UEV = A$ in Theorem 30.

$U$ is an $N−1$ by $N−1$ lower triangular matrix such that

$$u_{i,j} = \begin{cases} \binom{i}{j} \cdot \prod_{l=i+1}^{i+j-1} (f_1 + l f_2) & \text{if } i \geq j \\ 0 & \text{otherwise} \end{cases}$$

$E$ is an $N−1$ by $N−1$ diagonal matrix such that $e_i = i!$, where $1 \leq i \leq N−1$.

$V$ is an $N−1$ by $N−1$ upper triangular matrix such that

$$v_{i,j} = \begin{cases} 1 & \text{if } i = j \\ y^{i,j} W^{i,j} z_{i,j} & \text{if } i < j \\ 0 & \text{otherwise} \end{cases},$$

where $y^{i,j}$ is an $1$ by $j$ row vector such that $y^{i,j}_1 = (i f_1 + i^2 f_2)^{l-1}$ and $W^{i,j}$ is a $j$ by $j$ upper triangular matrix such that

$$W^{i,j} = \begin{cases} \mathbf{I} & \text{if } i = 1 \\ \prod_{l=i-1}^{l-1} W^{(l)} & \text{otherwise} \end{cases}$$

and $z_{i,j} = [0, 0, ..., 0, 1]^T$ is a $j$ by $1$ matrix.

$W^{(l)}$ is a $j$ by $j$ upper triangular matrix such that

$$w^{(l)}_{s,t} = \begin{cases} 1 & \text{if } s = t \\ (l f_1 + l^2 f_2)^{l-s-1} & \text{otherwise} \end{cases}.$$

**Proof:** See Appendix B.1.

We now prove the necessary and sufficient condition for a polynomial to be a null polynomial using Lemma 33.
Lemma 34 Let \( f(x) = \sum_{i=1}^{K} f_i x^i \pmod{N} \). A polynomial \( f(x) \) of degree \( K \) is a null polynomial of degree \( K \) modulo \( N \) if and only if \( \gcd(K!, N) \neq 1 \). Further, The number of null polynomials of degree \( K \) is \( \prod_{i=1}^{K} \gcd(i!, N) \) and null polynomials degree up to \( K \) are \( \sum_{i=1}^{K} \left\{ \frac{N}{\gcd(i!, N)} \cdot n \cdot \prod_{k=0}^{i-1} (x + k) \right\} \), where \( n = 0, 1, 2, \ldots, \gcd(i!, N) - 1 \).

Proof: See Appendix B.2.

Before proceeding further, we briefly review Catalan numbers \( C(k) \), where \( k = 0, 1, 2, \ldots \) [56], which will be used in Conjecture 35. Many problems in combinatorics have Catalan numbers as the solutions. Catalan numbers have various interpretations [57] and one of them is shown in Figure 3.3. In Figure 3.3, Catalan numbers \( C(k) \) are interpreted as the number of different ways a convex polygon with \( k + 2 \) sides can be cut into triangles. In Figure 3.3, the cases when \( k = 2, 3, 4 \) are shown, i.e., \( C(2) = 2, C(3) = 5 \) and \( C(4) = 14 \).

![Figure 3.1: The Catalan numbers.](image-url)
Catalan numbers have both binomial coefficient form and recursive form. Let $C(k)$, where $k \geq 0$, be a sequence of integers known as Catalan numbers [56]. The $k$th Catalan numbers are given by

$$C(k) = \frac{1}{k + 1} \binom{2k}{k} = \frac{(2k)!}{(k + 1)! \cdot k!}.$$  

The Catalan numbers also satisfy the recurrence relation,

$$C(0) = 1 \text{ and } C(k + 1) = \frac{2(2k + 1)}{k + 2} C(k).$$  

We now state the conjecture. In the following conjecture, we show $\mathbf{d} = \mathbf{U}^{-1}\mathbf{b}$ is described by a simplified form.

**Conjecture 35** Let $d_k = \frac{(-f_2)^{k-1} \cdot k! \cdot C(k-1)}{\prod_{l=1}^{k-1} (f_1 + lf_2)}$. Then

$$\sum_{j=1}^{k} u_{k,j} \cdot d_j = k, \text{ where } k = 1, 2, ..., N - 1,$$

i.e.,

$$k = \sum_{j=1}^{k} \left\{ \frac{\binom{k}{j}}{j} \cdot \prod_{l=k}^{k+j-1} (f_1 + lf_2) \right\} \cdot \left\{ \frac{(-f_2)^{j-1} \cdot j! \cdot C(j-1)}{\prod_{l=1}^{2j-1} (f_1 + lf_2)} \right\},$$

where $k = 1, 2, ..., N - 1$.

Conjecture 35 has been verified to be correct up to $k_{\text{conj}} = 50$ by using a computer.

We show the following lemma which will be used for proving Lemma 41.

**Lemma 36** ([42]) Let $f(x)$ be a quadratic permutation polynomial modulo $N$, where $N = \prod_{p \in \mathcal{P}} p^{n_{N,p}}$ and let $g^{(L)}(x)$ be a least degree inverse of $f(x)$. Then $L \leq \max_{p \in \mathcal{P}} n_{N,p}$.

**Lemma 37** Let $N = \prod_{p \in \mathcal{P}} p^{n_{N,p}}$. If $N \leq 2^{k_{\text{conj}}}$, then $\max_{p \in \mathcal{P}} n_{N,p} \leq k_{\text{conj}}$.  

33
Proof: Trivial.

**Corollary 38** Consider $d$ and $N$ in Lemma 33, Conjecture 35 and Lemma 37. $d_k \equiv 0 \pmod{N}$, where $k > k_{\text{conj}}$.

**Proof:** By Lemmas 36 and 37, $L \leq k_{\text{conj}}$. Suppose that the degree of $h^{(L)}$ in $Eh^{(L)} \equiv d$ is larger than $k_{\text{conj}}$. Then $h_k^{(L)} \neq 0$ for some $k > k_{\text{conj}}$ and $g_k^{(L)} \neq 0$ by Corollary 32. Thus the least degree of the inverse is larger than $k_{\text{conj}}$, which contradicts Lemma 36.

From now on, we consider the case when $N \leq 2^{k_{\text{conj}}}$. Since $2^{k_{\text{conj}}}$ is a large number, the investigation on the inverse of the quadratic permutation polynomial is not restricted in practice under this assumption. Under Conjecture 35 and Corollary 38, we prove the following two lemmas which will be used for proving Lemma 41.

**Lemma 39** $\gcd(e_k!, N)|d_k, \forall k \in \{1, \ldots, N-1\}$, consequently $e_k \cdot h_k \equiv d_k$ always has solutions and the number of solutions is $\gcd(e_k, N) = \gcd(k!, N)$.

**Proof:** It is trivial that $\gcd(e_k!, N)|d_k, \forall k \in \{1, \ldots, N-1\}$ by Lemma 33 and Conjecture 35. The latter statement can be shown by Theorem 8.

We prove the following lemmas which are necessary for proving Lemma 41.

**Lemma 40** $d_l|d_{l'}$, where $l$ and $l'$ are integers such that $1 \leq l \leq l' \leq N-1$.

**Proof:** This can be easily proved by using the recurrence relation of Catalan numbers and Corollary 38.

**Lemma 41** Let $N \leq 2^{k_{\text{conj}}}$. Then there exists a polynomial $g^{(L)}$ such that $Ag^{(L)} \equiv b$ if and only if $d_L \not\equiv 0 \pmod{N}$ and $d_{L+1} \equiv 0 \pmod{N}$.
Proof:

($\implies$)

Let the least degree of $h$ such that $Eh \equiv d$ be $K$. By Conjecture 35, Corollary 38 and Lemma 40, there exists a $L$ such that $d_L \not\equiv 0 \pmod{N}$ and $d_{L+1} \equiv 0 \pmod{N}$. Since $d_L \not\equiv 0 \pmod{N}$, $K \geq L$ and since $d_{L+1} \equiv 0 \pmod{N}$, $K \leq L$. Consequently, $K = L$. By Corollary 32, $\deg_L \{g\} = L$.

($\impliedby$)

By Lemma 39, there exists a solution $h$ for the equation $Eh \equiv d$. The least degree of $h$ cannot be less than $L$ since $d_L \not\equiv 0 \pmod{N}$. The least degree of $h$ cannot be larger than $L$ since $d_k \equiv 0 \pmod{N}$, $\forall k > L$ by Lemma 40.

**Lemma 42** Let $N \leq 2^{k_{conj}}$ and let $f(x) = f_1 x + f_2 x^2 \pmod{N}$ be a permutation polynomial. Then there exists a polynomial $g^L(x)$ such that $L \leq k_{conj}$ and $(g^L \circ f)(x) \equiv x \pmod{N}$, $\forall x \in \{0, 1, ..., N-1\}$ if and only if $d_L \not\equiv 0$ and $d_{L+1} \equiv 0 \pmod{N}$.

**Proof:**

This is a direct consequence of Lemmas 25 and 41.

Note that since $d_1 \not\equiv 0$ by Corollary 36, finding $d_L \not\equiv 0$ and $d_{L+1} \equiv 0 \pmod{N}$ in Lemmas 41 and 42 is equivalent to finding the smallest $L$ such that $d_{L+1} \equiv 0 \pmod{N}$.

**Theorem 43 (main Theorem)** Let $N = \prod_{p \in \mathcal{P}} p^{a_{N,p}} \leq 2^{k_{conj}}$, $\phi(k) = k! \cdot C(k-1) = \prod_{l=k}^{2k-2} l$ and $f(x)$ be a quadratic permutation polynomial.

Decompose $\phi(k) = k! \cdot C(k-1) = \prod_{l=k}^{2k-2} l$ into prime factors and denote each exponent of prime factors as $n_{\phi(k),p}$. Then $f(x)$ has $\prod_{k=1}^{L} \gcd(k!, N)$ inverse polynomials.
with the least degree $L$ if and only if there is a smallest integer $L$ such that

$$
n_{f,2} \geq \left\{ \begin{array}{ll}
\max \left( \left\lceil \frac{n_{N,2} - n_{\phi(L+1),2}}{L} \right\rceil, 1 \right) & \text{if } n_{N,2} > 1 \\
0 & \text{if } n_{N,2} = 0, 1
\end{array} \right.,
$$

$$
n_{f,p} \geq \left\{ \begin{array}{ll}
\max \left( \left\lceil \frac{n_{N,p} - n_{\phi(L+1),p}}{L} \right\rceil, 1 \right) & \text{if } n_{N,p} > 0 \\
0 & \text{if } n_{N,p} = 0
\end{array} \right..$$

If $L = 2$, it is reduced to the main Theorem 15 in Chapter 2.

**Proof:** We first consider the number of equivalent inverse polynomials with the least degree. By Corollary 39, the number of equivalent polynomial of $h^{(L)}$ is $\prod_{k=1}^{L} \gcd (k!, N)$, and by Corollary 32, the number of equivalent polynomials of $g^{(L)}$ and $h^{(L)}$ are equal.

We show that the conditions $d_{L} \not\equiv 0$ and $d_{L+1} \equiv 0 \pmod{N}$ are equivalent to the conditions on $n_{f,2}$ and $n_{f,p}$. Then by Lemma 41, the main Theorem holds.

We only prove for the condition on $n_{f,2}$. The other condition can be similarly proved.

($\implies$)

$d_{L+1} \equiv 0 \pmod{N}$ implies $2^{n_{N,2}}|2^{n_{\phi(L+1),2}+L-n_{f,2}}$.

If $n_{N,2} = 0, 1$, $2^{n_{N,2}}|2^{n_{\phi(L+1),2}+L-n_{f,2}}$ trivially holds, since $1 \leq n_{\phi(L+1),2}$.

If $n_{N,2} > 1$, $2^{n_{N,2}}|2^{n_{\phi(L+1),2}+L-n_{f,2}}$ implies $n_{N,2} \leq n_{\phi(L+1),2} + L \cdot n_{f,2}$, which gives the condition on $n_{f,2}$. Note that by Corollary 7, $n_{f,2} \geq 1$ when $n_{N,2} > 1$.

($\iff$)

Suppose the condition on $n_{f,2}$ is satisfied. Since $n_{f,2} \geq \left\lfloor \frac{n_{N,2} - n_{\phi(L+1),2}}{L} \right\rfloor \geq \frac{n_{N,2} - n_{\phi(L+1),2}}{L}$,

$L \cdot n_{f,2} + n_{\phi(L+1),2} \geq n_{N,2}$. Thus, $2^{n_{N,2}}|2^{n_{\phi(L+1),2}+L-n_{f,2}}$ and consequently, $d_{L+1} \equiv 0$.

**Corollary 44** Let $N = \prod_{p \in P} p^{n_{N,p}} \leq 2^{k_{\text{com}}}$. Let $\mathcal{F}$ be the set of quadratic permutation polynomials modulo $N$ and let $L'$ be the largest degree of inverse polynomials of $\mathcal{F}$. Then

$$L' = \min_{k} \{ k | k + n_{\phi(k+1),p} \geq n_{N,p}, \forall p \in P \},$$

36
which is less or equal to the bound $\max_{p \in P} \{n_{N,p}\}$ derived in [42].

Proof: Consider $L \cdot n_{f,p} + n_{\phi(k+1),p} \geq n_{N,p}$ in the proof of main Theorem 43. $L$ is largest when $n_{f,p} = 1$, $\forall p$ such that $n_{N,p} \geq 1$. It is immediate that $\min_k \{k | k + n_{\phi(k+1),p} \geq n_{N,p}, \forall p \in P\} \leq \max_{p \in P} \{n_{N,p}\}$. Thus the bound derived in [42] is larger or equal to the exact value derived in Corollary 44.

3.4 Examples

We present four examples to illustrate the necessary and sufficient conditions of Theorem 43. The first two examples consider interleavers that were investigated in [36]. The third example was from [33]. The fourth example shows the exact maximum degree for inverse polynomials can be smaller than an upper bound derived in [42].

1. Let $f(x) = f_1x + f_2x^2 \mod N$, where $N = 1504 = 2^5 \cdot 47$, $f_1 = 23$ and $f_2 = 2 \cdot 47$. The smallest $L$ such that $d_{L+1} = \frac{(-f_2)^{L \cdot \phi(L+1)}}{\prod_{t=1}^{L+1}(f_1 + tf_2)} \equiv 0$ is 3.

By Lemma 33

$$E = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 6 \end{bmatrix}, \quad V = \begin{bmatrix} 1 & 117 & 153 \\ 0 & 1 & 539 \\ 0 & 0 & 1 \end{bmatrix},$$

and $d = [797, 188, 752]^T$. We can compute the inverse of $f_1 + lf_2$ by referring the Table 2.3 in Chapter 2. Let us now solve $Eh^{(3)} \equiv d$ (mod 1504). From $e_1 \cdot h_1 \equiv d_1$, $e_2 \cdot h_2 \equiv d_2$, $e_3 \cdot h_3 \equiv d_3$, we get $h_1 = 797$, $h_2 = 94,846$, $h_3 = 376,1128$, respectively. Note that for each case, $\gcd(1!, N) = 1$, $\gcd(2!, N) = 2$ and $\gcd(3!, N) = 2$, so there are one $h_1$, two $h_2$’s and two $h_3$’s that satisfy the congruences $e_1 \cdot h_1 \equiv d_1$, $e_2 \cdot h_2 \equiv d_2$ and $e_3 \cdot h_3 \equiv d_3$, respectively. However, in
practice, we do not need to find all the $h_k$’s. We need only one $h_k$ for each $k$. Let us choose $h_1 = 797$, $h_2 = 94$ and $h_3 = 376$. For given $h^{(3)}$, we can get $g^{(3)}$ by solving $Vg^{(3)} \equiv h \pmod{1504}$. Note that $g_3 = h_3 = 376$, $g_2 = h_2 - v_{23} \cdot g_3 \pmod{1504} = 470$, $g_1 = h_1 - v_{12} \cdot g_2 - v_{13} \cdot g_3 \pmod{1504} = 1079$.

2. Let $f(x) = f_1 x + f_2 x^2 \mod N$, where $N = 5472 = 2^5 \cdot 3^2 \cdot 19$, $f_1 = 7 \cdot 11 \pmod{5472}$ and $f_2 = 2 \cdot 3 \cdot 19 \pmod{5472}$. Similarly, we can get $g^{(3)} = [1445, 3534, 456]^T$.

3. Let $f(x) = f_1 x + f_2 x^2 \mod N$, where $N = 16384 = 2^{14}$, $f_1 = 15 \pmod{2^{14}}$ and $f_2 = 2^5 \pmod{2^{14}}$. Similarly, we can get $g^{(3)} = [12015, 9760, 6144]^T$.

4. Let $N = 2^6$. Then by Corollary 44, least degree of the inverse polynomial is 3, which shows the upper bound 6 obtained by the technique in [42] is not tight.

Recently 188 QPP based interleavers have been proposed in 3GPP LTE [55]. Most of the interleavers proposed in [55] admit a quadratic inverse with the exception of 35 of them. In Table 3.2, all of the interleavers that do not admit quadratic inverses are listed with their respective least degree inverse polynomials computed using Algorithm 3.1.

### 3.5 Conclusion

We derived in Theorem 43 a necessary and sufficient condition to determine the degree of the least degree inverse polynomial for a QPP. We also provided an algorithm to explicitly compute the inverse polynomials.

Recently QPP interleavers were proposed in 3GPP LTE [55]. Most of the QPP interleavers in [55] admit a QPP inverse. We applied the theory in this chapter to
An algorithm for finding the inverse permutation polynomial(s) with least degree for a quadratic permutation polynomial $f(x) = f_1x + f_2x^2 \pmod{N}$

1. Factor $N$ and $f_2$ as products of prime powers and find the respective exponents of each prime factor, i.e., find $n_{N,p}$'s and $n_{f_2,p}$'s for $N = \prod_{p \in P} p^{n_{N,p}}$, $f_2 = \prod_{p \in P} p^{n_{f_2,p}}$.

If $n_{N,2} = 1$ and $n_{f_1,1}$ is even, set $f_1 = f_1 + \frac{N}{2} \pmod{N}$ and $f_2 = f_2 + \frac{N}{2} \pmod{N}$.

Recompute $n_{N,p}$'s and $n_{f,p}$'s.

2. Find the smallest $L \geq 2$ such that $d_{L+1} = \frac{(-f_2)^{L-1} \cdot C(L)}{\prod_{l=1}^{L+1} (f_1 + lf_2)} \equiv 0 \pmod{N}$,

where $C(0) = 1$ and $C(k) = \frac{1}{k+1} \binom{2k}{k}$.

When computing the inverse of $f_1 + lf_2$, refer to Table 2.2 in Chapter 2.


4. (1) When finding only one inverse (For most applications, this is sufficient).

Compute $h_k = \frac{(-f_2)^{k-1} \cdot C(k)}{\prod_{l=1}^{k+1} (f_1 + lf_2)}$, where $k = 1, 2, ..., L$.

(2) When finding all the inverses.

Find $h$ such that $Eh \equiv d \pmod{N}$ by solving $k! \cdot h_k \equiv d_k$ for $k = 1, 2, ..., L$.

$k! \cdot h_k \equiv d_k$ can be solved by using Theorem 8 in Chapter 2.

5. Find $g$ such that $Vg \equiv h \pmod{N}$ by computing $g_k$ in the descending order, i.e., $g_L, g_{L-1}, ..., g_2, g_1$.

Note that $g_L = h_L$ and $g_k = h_k - \sum_{l=k+1}^{L} v_{k,l} \cdot g_l \pmod{N}$ for $k = 1, 2, ..., L - 1$.

Table 3.1: Algorithm: computation of inverses of quadratic permutation polynomials.
<table>
<thead>
<tr>
<th>length</th>
<th>QPP</th>
<th>Least Degree Inverse Polynomial</th>
</tr>
</thead>
<tbody>
<tr>
<td>928</td>
<td>$15x + 58x^2$</td>
<td>$31x + 290x^2 + 232x^3$</td>
</tr>
<tr>
<td>1056</td>
<td>$17x + 66x^2$</td>
<td>$673x + 726x^2 + 88x^3$</td>
</tr>
<tr>
<td>1184</td>
<td>$19x + 74x^2$</td>
<td>$187x + 666x^2 + 296x^3$</td>
</tr>
<tr>
<td>1248</td>
<td>$19x + 78x^2$</td>
<td>$11x + 78x^2 + 104x^3$</td>
</tr>
<tr>
<td>1312</td>
<td>$367x + 82x^2$</td>
<td>$799x + 410x^2 + 328x^3$</td>
</tr>
<tr>
<td>1376</td>
<td>$21x + 86x^2$</td>
<td>$1245x + 1290x^2 + 344x^3$</td>
</tr>
<tr>
<td>1504</td>
<td>$49x + 846x^2$</td>
<td>$353x + 282x^2 + 376x^3$</td>
</tr>
<tr>
<td>1632</td>
<td>$25x + 102x^2$</td>
<td>$729x + 1122x^2 + 136x^3$</td>
</tr>
<tr>
<td>1696</td>
<td>$55x + 954x^2$</td>
<td>$663x + 530x^2 + 424x^3$</td>
</tr>
<tr>
<td>1760</td>
<td>$27x + 110x^2$</td>
<td>$1043x + 990x^2 + 440x^3$</td>
</tr>
<tr>
<td>1824</td>
<td>$29x + 114x^2$</td>
<td>$21x + 798x^2 + 152x^3$</td>
</tr>
<tr>
<td>1888</td>
<td>$45x + 354x^2$</td>
<td>$21x + 1534x^2 + 472x^3$</td>
</tr>
<tr>
<td>1952</td>
<td>$59x + 610x^2$</td>
<td>$1555x + 610x^2 + 488x^3$</td>
</tr>
<tr>
<td>2112</td>
<td>$17x + 66x^2$</td>
<td>$673x + 726x^2 + 88x^3$</td>
</tr>
<tr>
<td>2944</td>
<td>$45x + 92x^2$</td>
<td>$229x + 1748x^2 + 736x^3$</td>
</tr>
<tr>
<td>4160</td>
<td>$33x + 130x^2$</td>
<td>$3057x + 1430x^2 + 1560x^3$</td>
</tr>
<tr>
<td>4288</td>
<td>$33x + 134x^2$</td>
<td>$1137x + 1474x^2 + 536x^3$</td>
</tr>
<tr>
<td>4416</td>
<td>$35x + 138x^2$</td>
<td>$2555x + 138x^2 + 552x^3$</td>
</tr>
<tr>
<td>4544</td>
<td>$357x + 142x^2$</td>
<td>$4509x + 3266x^2 + 568x^3$</td>
</tr>
<tr>
<td>4672</td>
<td>$37x + 146x^2$</td>
<td>$2557x + 3358x^2 + 1752x^3$</td>
</tr>
<tr>
<td>4736</td>
<td>$71x + 444x^2$</td>
<td>$567x + 3996x^2 + 1184x^3$</td>
</tr>
<tr>
<td>4928</td>
<td>$39x + 462x^2$</td>
<td>$4391x + 3542x^2 + 616x^3$</td>
</tr>
<tr>
<td>4992</td>
<td>$127x + 234x^2$</td>
<td>$2175x + 2314x^2 + 4888x^3 + 104x^4$</td>
</tr>
<tr>
<td>5056</td>
<td>$39x + 158x^2$</td>
<td>$551x + 4582x^2 + 1896x^3$</td>
</tr>
<tr>
<td>5184</td>
<td>$31x + 96x^2$</td>
<td>$4543x + 1632x^2 + 288x^3$</td>
</tr>
<tr>
<td>5248</td>
<td>$113x + 902x^2$</td>
<td>$209x + 3034x^2 + 4264x^3 + 328x^4$</td>
</tr>
<tr>
<td>5312</td>
<td>$41x + 166x^2$</td>
<td>$745x + 498x^2 + 664x^3$</td>
</tr>
<tr>
<td>5440</td>
<td>$43x + 170x^2$</td>
<td>$3827x + 4250x^2 + 680x^3$</td>
</tr>
<tr>
<td>5504</td>
<td>$21x + 86x^2$</td>
<td>$3997x + 4386x^2 + 5160x^3 + 344x^4$</td>
</tr>
<tr>
<td>5568</td>
<td>$43x + 174x^2$</td>
<td>$3507x + 4350x^2 + 232x^3$</td>
</tr>
<tr>
<td>5696</td>
<td>$45x + 178x^2$</td>
<td>$981x + 5518x^2 + 2136x^3$</td>
</tr>
<tr>
<td>5824</td>
<td>$89x + 182x^2$</td>
<td>$409x + 546x^2 + 728x^3$</td>
</tr>
<tr>
<td>5952</td>
<td>$47x + 186x^2$</td>
<td>$3071x + 930x^2 + 744x^3$</td>
</tr>
<tr>
<td>6016</td>
<td>$23x + 94x^2$</td>
<td>$4839x + 5358x^2 + 1880x^3 + 376x^4$</td>
</tr>
<tr>
<td>6080</td>
<td>$47x + 190x^2$</td>
<td>$2943x + 950x^3 + 2280x^3$</td>
</tr>
</tbody>
</table>

Table 3.2: Least degree inverse polynomials for 3GPP LTE interleavers without quadratic inverses.
CHAPTER 4

AN UPPER BOUND ON THE MINIMUM DISTANCE OF TURBO CODES DESIGNED WITH QUADRATIC PERMUTATION POLYNOMIALS OVER INTEGER RINGS

4.1 Introduction

In this chapter, we consider an upper bound on the minimum distance of turbo codes designed with QPP interleavers.

The minimum distance $d_{\text{min}}$ and its multiplicity of error correcting can be used to estimate the error rates at high SNR using union bounds. At low to moderate SNR, however, the first a few terms of the weight distribution are also important for the error performance of error correcting codes.

For some error correcting codes, for example, some of BCH codes and Reed-Solomon codes which are defined over finite fields, weight distributions are known. However, for turbo codes, no algebraic method for finding the weight distribution has been suggested yet.

This chapter is organized as follows. In Section 4.2, we briefly review various distance measurement methods. In Sections 4.3-4.5, we investigate an upper bound on the minimum distance of turbo codes designed with QPP interleavers.
4.2 Distance Measurement Methods

In this section, we briefly review distance measurement methods for turbo codes [59, 60, 61, 62, 63, 64, 65, 66, 67]. In Subsection 4.2.1, true distance measurement methods, Garello algorithm [59] and Rosnes algorithm [60], improvement of Garello algorithm, are explained. In Subsection 4.2.2, fast distance measurement methods, Berrou’s impulse algorithm [66] and its variants [14, 16] are explained. Finally, in Subsection 4.2.3, a distance measurement method that is applicable to high spread interleaver is discussed.

4.2.1 Garello/Rosnes Algorithm

Error performance estimation of a turbo code at low error rates using simulations typically take a long time. It may take a few days or even a few weeks. Further, at very low error rate, the simulation may not be practical. An alternative is to compute an upper bound of error curves using the minimum distance of error correcting codes and its multiplicity. However, when a brute force algorithm is used to find the accurate $d_{\text{min}}$ and its multiplicity, all the $2^N$ codewords need to be computed, where $N$ is the interleaver size. Even for $N = 100$, the brute force search becomes impractical.

Garello [59] proposed a Viterbi-like algorithm that finds the minimum distance and its multiplicity (or the first a few distance profile and their multiplicities). Garello algorithm finds a list of all codewords with estimated weight less than an upper bound. Rosnes algorithm improves Garello algorithm by reducing the computation for the weight estimation of each codewords. Both Garello and Rosnes algorithm finds the true distance spectrum of turbo codes and Rosnes algorithm reduces the computation.
However, for turbo codes with interleaver length 2000 or higher, especially for high minimum distance interleaver, both algorithm is impractical.

4.2.2 Berrou’s Impulse Method and its Variants

Berrou et al. introduced in [66] a fast and probabilistic method for estimating minimum distance, which is based on the assumption that turbo decoder is maximum likelihood (ML) decoder. A low amplitude impulse is artificially inserted into the all-zero codeword at a specific position and this amplitude is increased until the decoder fails. The lowest amplitude at which the decoder fails is the estimate of $d_{\text{min}}$.

Garello et al. suggested a method called all-zero method [67], which is similar to Berrou algorithm [66]. Crozier et al. [14, 16] suggested a faster algorithm that improves the all-zero method [66]. However, these methods are not guaranteed to find the true minimum distance of turbo codes.

4.2.3 Crozier’s Method

Low weights of turbo code codewords are generated by combinations of low input weight (IW) patterns. For input weight 2 patterns, a quantity called spread [7, 10, 12], which is closely related to the weight generated by input weight 2 patterns, is used to estimate the minimum distance of turbo codes. Spread which is associated with two indexes $i$ and $j$ is defined as [10, 12]

$$S''(i, j) = |i - j| + |\pi(i) - \pi(j)|.$$ 

The minimum spread associated with index $i$ is defined as

$$S'(i) = \min_{j, j \neq i} [S''(i, j)].$$
The overall minimum spread is defined as

\[ S = \min_i [S'(i)]. \]

Typically, good interleavers have large spreads \([7, 36, 10, 12]\). Crozier \([10, 12]\) considered low weight input patterns, which cannot be removed even for high spread interleavers, that construct low weight codewords. In Figure 4.1, a codeword that can be removed for high spread interleavers is shown, and in Figure 4.2, a codeword that cannot be removed even for high spread interleavers is shown.

In \([10, 12]\), low input weights that generate low weight codewords are classified.

Typical examples are \([IW : 22 : 22]\), \([IW : 222 : 222]\), \([IW : 33 : 222]\), etc. For \([IW : 22 : 22]\), this label contains four numbers. The first two numbers indicate the pattern combinations before and after interleaving. Thus for \([IW : 22 : 22]\), there are two input weight 2 patterns in the first constituent codes and there are two input weight 2 patterns in the second constituent codes. By considering these bad error patterns, it has been demonstrated that an upper bound on the minimum distance of turbo codes was obtained.
4.3 An Upper Bound on the Minimum Distance of Turbo Codes Designed with QPP Interleavers by Algebraic Methods

It is observed that the most frequent patterns are of input weight $2k$, where $m = 1, 2, 3...$ for turbo codes designed with QPP interleavers [33]. In this section, we consider an upper bound on the minimum distance of turbo codes using input weight $2k$ pattern analysis. It is shown that for input weight 2, 4, 6, we can obtain an upper bound using algebraic methods.

4.3.1 Preliminary Work

In this subsection, we show some preliminary works which will be used in the rest of this section.

**Lemma 45** Let $f(x) = f_1x + f_2x^2 \pmod{N}$ be a QPP, where $N = \prod_{p \in \mathcal{P}} p^{n_p}$. Let us partition the set of $f_2 > 0$ such that any two $f_2$ and $\overline{f}_2$ are in the same set if and only if $\gcd (2f_2, N) = \gcd (2\overline{f}_2, N)$. Then the set of $f_2$ forms equivalence classes.

**Proof:** Trivial.
Proposition 46 Let \( f_2 \equiv \overline{f_2} \pmod{N} \), where \( 0 < \overline{f_2} < N \). Then, \( f_2 \) and \( \overline{f_2} \) are in the same equivalence class.

Proof: This can be easily shown, since this proposition was used in Chapter 2 and 3 without proof, we show the proof of this proposition.

1. \( p^{n_{N,p}} | p^{n_{F,p}} \), i.e., \( n_{N,p} \leq n_{F,p} \).

Consider \( f_2 \equiv \overline{f_2} \pmod{N} \), i.e., \( \overline{f_2} = f_2 - k \cdot N \). Since \( p^{n_{N,p}} | p^{n_{F,p}} \) and \( p^{n_{N,p}} | N \),

\( p^{n_{N,p}} | p^{n_{F,p}} \).

2. \( p^{n_{N,p}} \nmid p^{n_{F,p}} \), i.e, \( n_{F,p} < n_{N,p} \).

Since \( p^{n_{F,p}} | p^{n_{N,p}} \) and \( p^{n_{F,p}} | p^{n_{F,p}} \). However, \( p^{n_{F,p}+1} \nmid p^{n_{F,p}} \) since it is, \( p^{n_{F,p}+1} \mid p^{n_{F,p}} \) must holds, which is a contradiction to the assumption.

In summary,

\[
n_{F,p} = n_{F,p} \text{ if } n_{F,p} < n_{N,p} \\
\geq n_{N,p} \text{ if } n_{F,p} \geq n_{N,p}
\]

Consequently, \( \gcd(2f_2, N) = \gcd(2\overline{f_2}, N) \).

Corollary 47 Let \( f'_2 \equiv f_2 + \frac{N}{2} \pmod{N} \), where \( N \) is even. Then \( f_2 \) and \( \overline{f_2} \) are in the same equivalence class.

Proof: \( f'_2 \equiv f_2 + \frac{N}{2} \pmod{N} \) implies \( 2f'_2 \equiv 2f_2 + 2 \cdot \frac{N}{2} \pmod{N} \). Thus \( \gcd(2f_2, N) = \gcd(2\overline{f_2}, N) \) by the above proposition.

The above lemma and corollary show that we do not need to consider whether \( f_2 \geq N \), \( \frac{N}{2} \leq f_2 < N \) or not when we compute \( \gcd(2f_2, N) = \gcd(2\overline{f_2}, N) \). For a quadratic inverse, more generally, for an inverse of QPP, the exponents of primes are critical.
The above lemma and corollary also show that when we compute the inverses of QPP, we do not consider whether \( N \leq \frac{N}{f_2} \leq f_2 < N \) or not. From now on, we consider only QPP which is not equivalent to a linear PP, since it is shown that linear interleavers show bad performance [34], i.e., we consider only the case when \( \frac{N}{f_2} \mid f_2 \).

### 4.3.2 Input Weight 2 Pattern Analysis

In this subsection, we solve an equation generated by input weight 2 patterns. In this subsection, interleaver length \( N \) can be any number such that \( N > 0 \).

We first show an equation that describes input weight 2 patterns.

\[
|f(x + mt) - f(x)| \equiv nt \pmod{N}
\]

\[
\iff f_1(x + mt) + f_2(x + mt)^2 - f_1x - f_2x^2 \equiv \pm nt \pmod{N}
\]

\[
\iff mf_1t + 2mf_2tx + m^2t^2f_2 \equiv \pm nt \pmod{N}
\]

\[
\iff t \cdot 2mf_2x \equiv t \cdot (\pm n - mf_1 - m^2tf_2) \pmod{N}
\]

\[
\iff t \cdot 2mf_2x \equiv -t \cdot (m^2tf_2 + mf_1 + n) \pmod{N},
\]

where \( m, n \) are integers and \( t = 2^\nu - 1 \), where \( \nu \) is the constraint length of a convolutional code. Note that if \( m \) is odd, \( n \) must be odd and if \( m \) is even, \( n \) must be even.

We slightly abuse the notation and let \( n \) be any integer except 0. Then,

\[
t \cdot 2mf_2x \equiv -t \cdot (m^2tf_2 + mf_1 + n) \pmod{N}, \tag{4.1}
\]

Equation (4.1) has at least one solution if and only if

\[
\gcd(t \cdot 2mf_2, N) \mid \{-t \cdot (m^2tf_2 + mf_1 + n)\}, \tag{4.2}
\]

which is equivalent to

\[
-t \cdot (m^2tf_2 + mf_1 + n) \equiv 0 \pmod{\gcd(t \cdot 2mf_2, N)}. \tag{4.3}
\]
Equation (4.3) looks like a Diophantine-equation. However, it is not, since the modulus is not independent of $m$. Further, there is a second-order term in the equation. Equation (4.3) completely describes the multiplicities and the distances generated by input weight 2 patterns. Specifically, for given $f_1$, $f_2$ and $N$, if $m$, $n$ satisfies equation (4.3), there is an input weight 2 pattern that generates weight $4(|m| + |n|) + 6$, $8(|m| + |n|) + 6$ for $t = 7, 15$ with multiplicity $\gcd(t \cdot 2mf_2, N)$ by the following corollary. Thus by finding the set of $(m, n)$ satisfying equation (4.3), we can find all the weight distributions generated by 2 patterns.

**Corollary 48 ([21])** For $m$, $n$ used in the equation (4.3), weights of turbo code with $1101/1011$, $11101/10011$ encoder/decoder are $4(|m| + |n|) + 6$ and $8(|m| + |n|) + 6$ respectively.

We now show that equation (4.3) has at least one solution set $(m, n)$, which implies that for every $f_1$, $f_2$ and $N$, there exists input weight 2 pattern.

**Lemma 49** Equation (4.3) has at least one solution set $(m, n)$.

**Proof:** Let $m = 1$ and $n = tf_2 - f_1$. Then,

$$-t \cdot (m^2tf_2 + mf_1 + n) \equiv 0 \pmod{\gcd(t \cdot 2mf_2, N)}$$

$$\iff -t \cdot (2tf_2) \equiv 0 \pmod{\gcd(t \cdot 2f_2, N)},$$

since $\gcd(t \cdot 2f_2, N) \mid \{-t \cdot 2tf_2\}$.

For each $f_1$ and $f_2$, finding $(m, n)$ for equation (4.3) takes a long time, since the number of QPPs is huge. However, there are two lemmas that allow finding $(m, n)$ simpler.
Lemma 50 Let us partition the set of $f_2$ as in Lemma 45. Then we only need to solve equation (4.3) for one representative $f_2$ in each equivalence class.

Proof: See Appendix C.1.

Although we only need to solve equation (4.3) for one representative $f_2$ in each equivalence class, we need to solve equation (4.3) for every $f_1$. This also takes a long time. However, we do not need to do it using the following lemma.

Lemma 51 Let us partition the set of $f_2$ as in Lemma 45. For each representative in each equivalence classes, we only need to solve equation (4.3) for only the following range,

$$\begin{align*}
t &= 7 \begin{cases} 
1 \leq f_1 \leq 2f_{2,\text{min}} & \text{if } n_{N,7} = 0 \text{ or } n_{N,7} > 0, n_{f,7} < n_{N,7} \\
1 \leq f_1 \leq \frac{2f_{2,\text{min}}}{7} & \text{otherwise}
\end{cases}
\end{align*}$$

$$\begin{align*}
t &= 15 \begin{cases} 
1 \leq f_1 \leq 2f_{2,\text{min}} & \text{if } (n_{N,3} = 0 \text{ or } n_{N,3} > 0, n_{f,3} < n_{N,3}) \text{ and } (n_{N,5} = 0 \text{ or } n_{N,5} > 0, n_{f,5} < n_{N,5}) \\
1 \leq f_1 \leq \frac{2f_{2,\text{min}}}{5} & \text{if } (n_{N,3} = 0 \text{ or } n_{N,3} > 0, n_{f,3} < n_{N,3}) \text{ and } (n_{N,5} > 0, n_{N,5} \leq n_{f,5}) \\
1 \leq f_1 \leq \frac{2f_{2,\text{min}}}{3} & \text{if } (n_{N,5} = 0 \text{ or } n_{N,5} > 0, n_{f,5} < n_{N,5}) \text{ and } (n_{N,3} > 0, n_{N,3} \leq n_{f,3}) \\
1 \leq f_1 \leq \frac{2f_{2,\text{min}}}{15} & \text{otherwise}
\end{cases}
\end{align*}$$

, where $f_{2,\text{min}}$ is the smallest element in each equivalence classes.

Proof: See Appendix C.2.

4.3.3 An Upper Bound on the Minimum Distance of Turbo Codes Designed with QPP interleavers without QPP Inverse Using Input Weight 2 Pattern Analysis

In this subsection, we find an upper bound on $d_{\text{min}}$ of turbo codes designed with QPP interleavers which do not admit quadratic inverses using the results in Subsection 4.3.2. We first restrict $N$ as in the following lemma.
Lemma 52 Let $f(x) = f_1x + f_2x^2 \pmod{N}$ be a QPP, where $N = \prod_{p \in P} p^{n_{N,p}}$. Suppose that $N$ is such that
\[
\begin{cases}
  n_{N,2} \geq 1 \\
  n_{N,3} \leq 3 \\
  n_{N,p} \leq 2 & \text{if } p \neq 2, p \neq 3
\end{cases}
\]
In this case, a QPP do not admit quadratic inverse because of the deficiency of the exponent of 2.

Proof: This can be easily shown by checking the necessary and sufficient condition for a QPP to admit quadratic inverse (Theorem 15).

Corollary 53 Let $N \geq 1024$ and let $N$ be the length of 3GPP LTE interleavers. Then the only length that does not satisfy the Lemma 52 is $5184 = 2^6 \cdot 3^4$.

Proof: This can be verified using brute-force search. Corollary 53 holds since all the 3GPP LTE interleaver lengths are divided by either 32 or 64. The reason that we only consider these length will be clearer at the end of this subsection.

In the following lemma, we show that equation (4.3) can be further simplified.

Lemma 54 Let $N = \prod_{p \in P} p^{n_{N,p}}$ in Lemma 52. Let $f_{2,\min} = 2^{n_{f,2} \cdot 2^{n_{f,7}}} \prod_{p \in P \setminus \{2,7\}} p^{n_{N,p}}$, where $n_{f,2} = \max(\lceil \frac{n_{N,2} - 2}{2} \rceil - 1, 1)$. If $m$ is odd,
\[
\gcd(t \cdot 2mf_{2,\min}, N) = \begin{cases}
  2f_{2,\min} & \text{if } n_{N,7} = 0 \\
  t \cdot \frac{2f_{2,\min}}{7} & \text{if } n_{N,7} > 0 \text{ and } n_{f,7} = n_{N,7}
\end{cases}
\]
Consequently, when $m$ is odd,
\[
\text{Eq.(4.3) } \iff \begin{cases}
  m_1 + n \equiv f_{2,\min} \pmod{2f_{2,\min}} & \text{if } n_{N,7} = 0 \\
  m_1 + n \equiv f_{2,\min} \pmod{\frac{2f_{2,\min}}{7}} & \text{if } n_{N,7} > 0 \text{ and } n_{f,7} = n_{N,7} \\
  m_1 + n \equiv f_{2,\min} \pmod{2f_{2,\min}} & \text{if } n_{N,7} > 0 \text{ and } n_{f,7} < n_{N,7}
\end{cases}
\]
and when \( m \) is even,

\[
\{ \begin{array}{ll}
    m f_1 + n \equiv 0 & (\text{mod } \gcd(2m f_{2\text{min}}, N)) \\
    m f_1 + n \equiv 0 & (\text{mod } \gcd\left(\frac{2m f_{2\text{min}}}{7}, N\right)) \\
    m f_1 + n \equiv 0 & (\text{mod } \gcd(2m f_{2\text{min}}, N))
\end{array} \right.
\]

if \( n_{N,7} = 0 \)

\[
\{ \begin{array}{ll}
    n_{f,7} = n_{N,7} & \text{if } n_{N,7} > 0 \text{ and } \\
    n_{f,7} < n_{N,7} & \end{array} \right.
\]

\[\leq \Rightarrow\]

In summary, when \( m \) is odd

\[
\{ \begin{array}{ll}
    m f_1 + n \equiv f_{2\text{min}} & (\text{mod } 2f_{2\text{min}}) \\
    m f_1 + n \equiv f_{2\text{min}} & (\text{mod } \frac{2f_{2\text{min}}}{7})
\end{array} \right.
\]

if \( n_{N,7} = 0 \) or \( n_{N,7} > 0, n_{f,7} < n_{N,7} \),

otherwise

and when \( m \) is even,

\[
\{ \begin{array}{ll}
    m f_1 + n \equiv 0 & (\text{mod } \gcd(2m f_{2\text{min}}, N)) \\
    m f_1 + n \equiv 0 & (\text{mod } \gcd\left(\frac{2m f_{2\text{min}}}{7}, N\right))
\end{array} \right.
\]

if \( n_{N,7} = 0 \) or \( n_{N,7} > 0, n_{f,7} < n_{N,7} \),

otherwise

\[\leq \Rightarrow\]

Proof: See Appendix C.4.

Note that if \( m \) is 1, we do not need to restrict \( N \) in Lemma 52. Further, for even \( m \),

without restriction, the results in Corollary 53 hold.

In the following, we find an upper bound on \( d_{\text{min}} \) of turbo codes designed with QPP

interleavers for \( N \) in Lemma 52 by the following lemma. We first prove the following

lemma.

\[\text{Lemma 55}\]

In Lemma 54, we only need to compute \((m, n)\) for \( 1 \leq f_1 \leq f_{2\text{min}} \) or \( 1 \leq f_1 \leq \frac{f_{2\text{min}}}{7} \) for \( t = 7 \). For \( t = 15 \), similar result can be shown.

Proof: See Appendix C.3.

In the following, we compute an upper bound by input weight 2 pattern analysis.

\[\text{Lemma 56}\]

Let also \( f_{2\text{min}} = 2^{n_{f,2}} \cdot \prod_{p \in \mathbb{F} \backslash \{2\}} p^{n_{p,v}} \),

where \( n_{f,2} = \max\left(\left\lceil \frac{n_{N,2} - 2}{2} \right\rceil - 1, 1\right) \). Then,
1. $t = 7$

(a) $n_{N,2} \geq 2$

\[
\begin{align*}
UB & \leq 22 & \text{if } f_{2,\min} = 4 \\
UB & \leq 2f_{2,\min} + 6 & \text{if } n_{N,7} = 0 \text{ or } n_{N,7} > 0, n_{f,7} < n_{N,7} \\
UB & \leq \frac{2f_{2,\min}}{7} + 6 & \text{otherwise}
\end{align*}
\]

(b) $n_{N,2} = 1$

\[
\begin{align*}
UB & \leq 2f_{2,\min} + 10 & \text{if } n_{N,7} = 0 \text{ or } n_{N,7} > 0, n_{f,7} < n_{N,7} \\
UB & \leq \frac{2f_{2,\min}}{7} + 10 & \text{otherwise}
\end{align*}
\]

2. $t = 15$

(a) $n_{N,2} \geq 2$

\[
\begin{align*}
UB & \leq 38 & \text{if } f_{2,\min} = 4 \\
UB & \leq 4f_{2,\min} + 6 & \text{if } (n_{N,3} = 0 \text{ or } n_{N,3} > 0, n_{f,3} < n_{N,3}) \text{ and } (n_{N,5} = 0 \text{ or } n_{N,5} > 0, n_{f,5} < n_{N,5}) \\
UB & \leq \frac{4f_{2,\min}}{5} + 6 & \text{if } (n_{N,3} = 0 \text{ or } n_{N,3} > 0, n_{f,3} < n_{N,3}) \text{ and } (n_{N,5} > 0, n_{N,5} \leq n_{f,5}) \\
UB & \leq \frac{4f_{2,\min}}{3} + 6 & \text{if } (n_{N,5} = 0 \text{ or } n_{N,5} > 0, n_{f,5} < n_{N,5}) \text{ and } (n_{N,3} > 0, n_{N,3} \leq n_{f,3}) \\
UB & \leq \frac{4f_{2,\min}}{15} + 14 & \text{otherwise}
\end{align*}
\]

(b) $n_{N,2} = 1$

\[
\begin{align*}
UB & \leq 4f_{2,\min} + 14 & \text{if } (n_{N,3} = 0 \text{ or } n_{N,3} > 0, n_{f,3} < n_{N,3}) \text{ and } (n_{N,5} = 0 \text{ or } n_{N,5} > 0, n_{f,5} < n_{N,5}) \\
UB & \leq \frac{4f_{2,\min}}{5} + 14 & \text{if } (n_{N,3} = 0 \text{ or } n_{N,3} > 0, n_{f,3} < n_{N,3}) \text{ and } (n_{N,5} > 0, n_{N,5} \leq n_{f,5}) \\
UB & \leq \frac{4f_{2,\min}}{3} + 14 & \text{if } (n_{N,5} = 0 \text{ or } n_{N,5} > 0, n_{f,5} < n_{N,5}) \text{ and } (n_{N,3} > 0, n_{N,3} \leq n_{f,3}) \\
UB & \leq \frac{4f_{2,\min}}{15} + 14 & \text{otherwise}
\end{align*}
\]

Proof: See Appendix C.5.

We now show the reason we restricted $N$ as follows. Suppose that we find an upper bound on $d_{\min}$ of turbo codes designed with QPP interleavers with length $N = 2^6 \cdot 3^4$. Let $f_2 = 2^8 \cdot 3$. Then this QPP does not admit quadratic inverse and when Lemma 56
was used, the upper bound becomes very large. However, when we restrict $N$ as in Lemma 52, such a problem does not occur. Since a QPP admits non-QPP inverse when the exponent of 2 is small, the upper bound can be tightened. In Table 4.1 and 4.2, we used equation (4.3) to compute an upper bound on $d_{\text{min}}$ and compared it to the theoretical upper bound.

This approach saves computation. For example, let $N = 4096$. $f_2$ ranges from 2 to $2^{11}$. Then the number of $f_2$’s is $2^{10}$. For each $f_2$, there are $2^{11} f_1$’s. Consequently,

\begin{table}
\begin{tabular}{|c|c|c|c|}
\hline
length & factor & UB by computation in Lemma 56 & UB in Lemma 56 \\ 
\hline
1024 & $2^{10}$ & 22 & 22 \\ 
1088 & $2^6 \cdot 17$ & 46 & 74 \\ 
1120 & $2^5 \cdot 5 \cdot 7$ & 22 & 26 \\ 
1216 & $2^6 \cdot 19$ & 54 & 82 \\ 
1280 & $2^8 \cdot 5$ & 46 & 46 \\ 
1344 & $2^6 \cdot 3 \cdot 7$ & 22 & 22 \\ 
1536 & $2^9 \cdot 3$ & 54 & 54 \\ 
1568 & $2^5 \cdot 7^2$ & 30 & 34 \\ 
1792 & $2^8 \cdot 7$ & 22 & 22 \\ 
2016 & $2^5 \cdot 3^2 \cdot 7$ & 30 & 42 \\ 
2048 & $2^{11}$ & 38 & 38 \\ 
2240 & $2^6 \cdot 5 \cdot 7$ & 22 & 26 \\ 
2688 & $2^7 \cdot 3 \cdot 7$ & 30 & 30 \\ 
3072 & $2^{10} \cdot 3$ & 54 & 54 \\ 
3136 & $2^6 \cdot 7^2$ & 30 & 34 \\ 
3584 & $2^9 \cdot 7$ & 22 & 22 \\ 
4032 & $2^6 \cdot 3^2 \cdot 7$ & 30 & 42 \\ 
4096 & $2^{12}$ & 38 & 38 \\ 
4480 & $2^7 \cdot 5 \cdot 7$ & 46 & 46 \\ 
4928 & $2^6 \cdot 7 \cdot 11$ & 38 & 50 \\ 
5376 & $2^8 \cdot 3 \cdot 7$ & 30 & 30 \\ 
5824 & $2^6 \cdot 7 \cdot 13$ & 38 & 58 \\ 
\hline
\end{tabular}
\end{table}

Table 4.1: An upper bound on the $d_{\text{min}}$ of turbo codes designed with QPP interleavers which do not admit quadratic inverses by input weight 2 pattern analysis when the encoder has 8 states.

53
<table>
<thead>
<tr>
<th>length</th>
<th>factor</th>
<th>UB by computation in 56</th>
<th>UB in Lemma 56</th>
</tr>
</thead>
<tbody>
<tr>
<td>1024</td>
<td>$2^{10}$</td>
<td>42</td>
<td>42</td>
</tr>
<tr>
<td>1056</td>
<td>$2^5 \cdot 3 \cdot 11$</td>
<td>74</td>
<td>106</td>
</tr>
<tr>
<td>1120</td>
<td>$2^5 \cdot 5 \cdot 7$</td>
<td>58</td>
<td>74</td>
</tr>
<tr>
<td>1152</td>
<td>$2^7 \cdot 3^2$</td>
<td>58</td>
<td>58</td>
</tr>
<tr>
<td>1248</td>
<td>$2^5 \cdot 3 \cdot 13$</td>
<td>74</td>
<td>122</td>
</tr>
<tr>
<td>1280</td>
<td>$2^8 \cdot 5$</td>
<td>42</td>
<td>42</td>
</tr>
<tr>
<td>1344</td>
<td>$2^6 \cdot 3 \cdot 7$</td>
<td>58</td>
<td>74</td>
</tr>
<tr>
<td>1440</td>
<td>$2^5 \cdot 3^2 \cdot 5$</td>
<td>42</td>
<td>42</td>
</tr>
<tr>
<td>1536</td>
<td>$2^9 \cdot 3$</td>
<td>42</td>
<td>42</td>
</tr>
<tr>
<td>1600</td>
<td>$2^6 \cdot 5^2$</td>
<td>42</td>
<td>58</td>
</tr>
<tr>
<td>1728</td>
<td>$2^6 \cdot 3^3$</td>
<td>58</td>
<td>90</td>
</tr>
<tr>
<td>1760</td>
<td>$2^5 \cdot 5 \cdot 11$</td>
<td>74</td>
<td>106</td>
</tr>
<tr>
<td>1920</td>
<td>$2^7 \cdot 3 \cdot 5$</td>
<td>42</td>
<td>42</td>
</tr>
<tr>
<td>2048</td>
<td>$2^{12}$</td>
<td>74</td>
<td>74</td>
</tr>
<tr>
<td>2112</td>
<td>$2^6 \cdot 3 \cdot 11$</td>
<td>74</td>
<td>106</td>
</tr>
<tr>
<td>2240</td>
<td>$2^6 \cdot 5 \cdot 7$</td>
<td>58</td>
<td>74</td>
</tr>
<tr>
<td>2304</td>
<td>$2^8 \cdot 3^2$</td>
<td>58</td>
<td>58</td>
</tr>
<tr>
<td>2496</td>
<td>$2^6 \cdot 3 \cdot 13$</td>
<td>74</td>
<td>122</td>
</tr>
<tr>
<td>2560</td>
<td>$2^9 \cdot 5$</td>
<td>42</td>
<td>42</td>
</tr>
<tr>
<td>2880</td>
<td>$2^6 \cdot 3^2 \cdot 5$</td>
<td>42</td>
<td>42</td>
</tr>
<tr>
<td>3072</td>
<td>$2^{10} \cdot 3$</td>
<td>42</td>
<td>42</td>
</tr>
<tr>
<td>3520</td>
<td>$2^6 \cdot 5 \cdot 11$</td>
<td>74</td>
<td>106</td>
</tr>
<tr>
<td>3840</td>
<td>$2^8 \cdot 3 \cdot 5$</td>
<td>42</td>
<td>42</td>
</tr>
<tr>
<td>4096</td>
<td>$2^{12}$</td>
<td>74</td>
<td>74</td>
</tr>
<tr>
<td>4160</td>
<td>$2^6 \cdot 5 \cdot 13$</td>
<td>74</td>
<td>122</td>
</tr>
<tr>
<td>5760</td>
<td>$2^7 \cdot 3^2 \cdot 5$</td>
<td>58</td>
<td>58</td>
</tr>
<tr>
<td>6144</td>
<td>$2^{11} \cdot 3$</td>
<td>74</td>
<td>74</td>
</tr>
</tbody>
</table>

Table 4.2: An upper bound on the $d_{\min}$ of turbo codes designed with QPP interleavers which do not admit quadratic inverses by input weight 2 pattern analysis when the encoder has 16 states.

The number of QPPs for this length is $2^{21}$. If an interleaver has a quadratic inverse, $f_2$ ranges from $2^5$ to $2^{11}$. Thus the number of QPP with quadratic inverse is $2^{17}$. Consequently, the number of QPPs without quadratic inverse is $2^{21} - 2^{17} \approx 2$ million.
4.3.4 An Upper Bound on the Minimum Distance of Turbo Codes Designed with QPP interleavers with QPP Inverse Using Input Weight 4, 6 Pattern Analysis

In this subsection, we find an upper bound on $d_{\text{min}}$ of turbo codes designed with QPP interleavers which admit quadratic inverses using input weight 4, 6 pattern analysis. Let $f(x) = f_1 x + f_2 x^2 \pmod{N}$ denote a QPP with a quadratic inverse denoted by $g(x) = g_1 x + g_2 x^2 \pmod{N}$. Furthermore, we will consider component encoders with a primitive feedback polynomial of degree $\nu$.

A general upper bound of $2(2^{\nu+1} + 9)$ on $d_{\text{min}}$

![Diagram](image)

Figure 4.3: An input weight 6 critical codeword for QPP interleavers (1).

Let us look at the pattern in Figure 4.3. To make the pattern in Figure 4.3 a codeword, we need to make sure that $f(g(f(x) + a) + 2a) \equiv f(g(f(x) + a) + 2a) + a - a \pmod{N}$ with $a$ equal to the length of a input weight 2 minimum parity weight
pattern. With a primitive feedback polynomial \( a = 2^\nu - 1 \) and the corresponding parity weight with a monic feedforward polynomial is \( 2 + 2^{\nu-1} \). This congruence is equivalent, with \( x = 0 \), to

\[
4 \cdot a^3 \cdot f_2g_2(1 + 2f_1 + 2af_2) \equiv 0 \pmod{N}.
\] (4.4)

The Hamming weight of the codeword in Figure 4.3 is at most \( 2(2^{\nu+1} + 9) \). Equality holds if the patterns do not interfere with each other as in Figure 4.3. With \( \nu = 3 \), the weight is at most 50.

From Theorem 14 in Chapter 2, we know that \( 12f_2g_2 \equiv 0 \pmod{N} \), and it follows that \( 4f_2g_2 \equiv 0 \pmod{N} \) if 27 is not a divisor of \( N \). Thus, the congruence in equation (4.4) holds for all QPPs with a quadratic inverse, for a given value of \( N \), if 27 is not a divisor of \( N \).

![Figure 4.4: An input weight 6 critical codeword for QPP interleavers (2).](image-url)
In Figure 4.4 another input weight 6 critical pattern is depicted. To make this pattern a codeword, \( f(g(f(x) + 2a) - a) \equiv f(g(f(x + a) + a) - 2a) + a \pmod{N} \), which is equivalent, with \( x = 0 \), to

\[
4 \cdot a^3 \cdot f_2 g_2 (1 - 2f_1 - 2af_2) \equiv 0 \pmod{N}.
\] (4.5)

As for the codeword in Figure 4.3, the Hamming weight of the codeword in Figure 4.4 is at most \( 2(2^{\nu} + 1 + 9) \). Assume \( 27 | N \). Then, \( f_2 = 3 \cdot c \), for some integer \( c \), since \( 3 | f_2 \). Furthermore, \( f_1 = 1 + 3 \cdot k \) or \( 2 + 3 \cdot k \), for some integer \( k \), since \( \gcd(f_1 + f_2, N) = 1 \). If \( f_1 = 1 + 3 \cdot k \), then the congruence in equation (4.4) reduces to

\[
4 \cdot a^3 \cdot f_2 g_2 (1 + 2(1 + 3 \cdot k) + 2a \cdot 3 \cdot c) = 12 \cdot a^3 \cdot f_2 g_2 (1 + 2 \cdot k + 2a \cdot c) \equiv 0 \pmod{N},
\]

which is always true, since \( 12f_2g_2 \equiv 0 \pmod{N} \) by Theorem 14. Furthermore, if \( f_1 = 2 + 3 \cdot k \), then the congruence in equation (4.5) reduces to

\[
4 \cdot a^3 \cdot f_2 g_2 (1 - 2(2 + 3 \cdot k) - 2a \cdot 3 \cdot c) = 12 \cdot a^3 \cdot f_2 g_2 (-1 - 2 \cdot k - 2a \cdot c) \equiv 0 \pmod{N},
\]

which is always true, since \( 12f_2g_2 \equiv 0 \pmod{N} \) by Theorem 14. Thus, there is an upper bound of \( 2(2^{\nu} + 1 + 9) \) on \( d_{\min} \) for QPPs with a quadratic inverse for all values of \( N \).

### 4.3.5 Various Partial Upper Bounds on the Minimum Distance

Consider the pattern in Figure 4.5. This is a codeword if \( g(f(x + b) + a) \equiv g(f(x) + a) + b \pmod{N} \) and \( g(f(x + c) + a) \equiv g(f(x) + a) + c \pmod{N} \), which is equivalent, with \( x = 0 \), to \( 2bg_2(f_1 + bf_2) \equiv 0 \pmod{N} \) and \( 2cg_2(f_1 + cf_2) \equiv 0 \pmod{N} \). Choose \( a = 2^{\nu} - 1 \) and \( b \) and \( c \) such that they correspond to a minimum parity weight pattern.

With the Universal Mobile Telecommunications System (UMTS) component encoders we choose \( b = 8 \) and \( c = 12 \). Then, the two congruences above reduces
to $112g_2(f_1 + 8f_2) \equiv 0 \pmod{N}$ and $168g_2(f_1 + 12f_2) \equiv 168g_2f_1 \equiv 0 \pmod{N}$ by Theorem 14. Furthermore, if $168g_2f_1 \equiv 0 \pmod{N}$, then $2 \cdot 168g_2f_1 + 224 \cdot 12g_2f_2 = 3 \cdot 112g_2(f_1 + 8f_2) \equiv 0 \pmod{N}$ by Theorem 14, from which it follows that $112g_2(f_1 + 8f_2) \equiv 0 \pmod{N}$ if 9 is not a divisor of $N$. The congruence $168g_2f_1 \equiv 0 \pmod{N}$ is satisfied (for all valid values of $f_1$ and $f_2$) if $n_{G,2} + 3 \geq n_{N,2}$, $n_{G,3} + 1 \geq n_{N,3}$, $n_{G,7} + 1 \geq n_{N,7}$, and $n_{G,p} \geq n_{N,p}$, $p \neq 2, 3, 7$. Using Theorem 14, it follows that the congruences are satisfied for all valid values of $g_2$, for a given value of $N$, if the following conditions are satisfied.

$$n_{N,2} \leq \begin{cases} 3 + \left\lceil \frac{n_{N,2} - 2}{2} \right\rceil & \text{if } n_{N,2} > 1 \\ 3 & \text{if } n_{N,2} = 0, 1 \end{cases}$$

$$n_{N,3} \leq \begin{cases} 1 + \left\lceil \frac{n_{N,3} - 1}{2} \right\rceil & \text{if } n_{N,3} > 0 \\ 1 & \text{if } n_{N,3} = 0 \end{cases}$$

$$n_{N,7} \leq 1 + \left\lceil \frac{n_{N,7}}{2} \right\rceil$$

$$n_{N,p} \leq \left\lceil \frac{n_{N,p}}{2} \right\rceil \quad \text{if } p \neq 2, 3, 7,$$
which reduces (when we use the fact that 9 should not be a divisor of $N$) to

$$n_{N,p} \leq \begin{cases} 5 & \text{if } p = 2 \\ 3 & \text{if } p = 7 \\ 1 & \text{otherwise} \end{cases} \quad \text{(4.6)}$$

The Hamming weight of the codeword in Figure 4.5 is at most $12 + 3 \cdot 2^{\nu-1} + 2$ (minimum parity weight of an input weight 3 pattern). With the UMTS component encoders, this weight is 38. Thus, there is an upper bound of 38 on $d_{\min}$ with the UMTS component encoders for QPPs with a quadratic inverse when the interleaver length $N$ satisfies equation (4.6).

Let us now consider the pattern in Figure 4.6. To make this pattern a codeword, we need to make sure that $g(f(x + a) + b) \equiv g(f(x) + c) + d \pmod{N}$ for some nonzero integers $a, b, c,$ and $d,$ such that we get 4 patterns. The Hamming weight of the codeword in Figure 4.6 is $12 + 2^{\nu-1}(|a| + |b| + |c| + |d|)/(2^{\nu} - 1)$. The congruence $g(f(x + a) + b) \equiv g(f(x) + c) + d \pmod{N}$ is equivalent, with $x = 0,$ to

$$\left(b^2 - c^2\right)g_2 + (b - c)g_1 + a - d + 2abg_2(f_1 + af_2) \equiv 0 \pmod{N} \quad \text{(4.7)}$$

If we choose $c = b$ and $d = a,$ we get $2abg_2(f_1 + af_2) \equiv 0 \pmod{N}.$ Furthermore, let $a = (2^{\nu} - 1)a'$ and $b = (2^{\nu} - 1)b'$

We will now consider the case of $\nu = 3$ in more detail. The congruence in the second row and third column of Table 4.3 and 4.4 (with $\nu = 3$) is satisfied if $n_{G,2} + 1 \geq n_{N,2}$ and $n_{G,7} + 2 \geq n_{N,7}.$ Using Theorem 15, it follows that the two inequalities above
\[
\begin{array}{|c|c|}
\hline
([a'], [b']) & \text{Equation (4.7) with } c = b \text{ and } d = a \\
(1, 1) & (2^\nu - 1)^2 \cdot 2g_2(f_1 \pm (2^\nu - 1)f_2) \equiv 0 \\
(1, 2) & (2^\nu - 1)^2 \cdot 2^2g_2(f_1 \pm (2^\nu - 1)f_2) \equiv 0 \\
(2, 1) & (2^\nu - 1)^2 \cdot 2^2g_2(f_1 \pm 2(2^\nu - 1)f_2) \equiv 0 \\
(2, 2) & (2^\nu - 1)^2 \cdot 2^2g_2(f_1 \pm 2(2^\nu - 1)f_2) \equiv 0 \\
(1, 3) & (2^\nu - 1)^2 \cdot 2 \cdot 3g_2(f_1 \pm (2^\nu - 1)f_2) \equiv 0 \\
(3, 1) & (2^\nu - 1)^2 \cdot 2 \cdot 3g_2(f_1 \pm 3(2^\nu - 1)f_2) \equiv 0 \\
\hline
\end{array}
\]

**Table 4.3:** Summary of conditions that make the pattern in Figure 4.6 a codeword of Hamming weight at most \(2(2^\nu + 9)\), for \(\nu \geq 3\) (1).

\[
\begin{array}{|c|c|c|}
\hline
([a'], [b']) & \text{Simplified congruence by Theorem 14} & \text{Weight} \\
(1, 1) & (2^\nu - 1)^2 \cdot 2g_2f_1 \equiv 0 \text{ when } (2^\nu \nmid N \text{ or } 3 \mid (2^\nu - 1)) \text{ and } 8 \nmid N & 12 + 2^\nu+1 \\
(1, 2) & (2^\nu - 1)^2 \cdot 2^2g_2f_1 \equiv 0 \text{ when } 27 \nmid N \text{ or } 3 \mid (2^\nu - 1) & 12 + 3 \cdot 2^\nu \\
(2, 1) & (2^\nu - 1)^2 \cdot 2^2g_2f_1 \equiv 0 \text{ when } 27 \nmid N \text{ or } 3 \mid (2^\nu - 1) & 12 + 3 \cdot 2^\nu \\
(2, 2) & (2^\nu - 1)^2 \cdot 2^2g_2f_1 \equiv 0 \text{ when } 27 \nmid N \text{ or } 3 \mid (2^\nu - 1) & 12 + 2^\nu+2 \\
(1, 3) & (2^\nu - 1)^2 \cdot 2 \cdot 3g_2f_1 \equiv 0 \text{ when } 8 \nmid N & 12 + 2^\nu+2 \\
(3, 1) & (2^\nu - 1)^2 \cdot 2 \cdot 3g_2f_1 \equiv 0 \text{ when } 8 \nmid N & 12 + 2^\nu+2 \\
\hline
\end{array}
\]

**Table 4.4:** Summary of conditions that make the pattern in Figure 4.6 a codeword of Hamming weight at most \(2(2^\nu + 9)\), for \(\nu \geq 3\) (2).

are satisfied for all values of \(g_2\) if

\[
\begin{align*}
n_{N, 2} & \leq \begin{cases} 
1 + \max \left( \left\lceil \frac{n_{N, 2} - 2}{2} \right\rceil, 1 \right) & \text{if } n_{N, 2} > 1 \\
1 & \text{if } n_{N, 2} = 0, 1 
\end{cases} \\
n_{N, 3} & \leq \begin{cases} 
\max \left( \left\lceil \frac{n_{N, 3} - 1}{2} \right\rceil, 1 \right) & \text{if } n_{N, 3} > 0 \\
0 & \text{if } n_{N, 3} = 0 
\end{cases} \\
n_{N, 7} & \leq 2 + \left\lceil \frac{n_{N, 7}}{2} \right\rceil \\
n_{N, p} & \leq \left\lceil \frac{n_{N, p}}{2} \right\rceil \quad \text{if } p \neq 2, 3, 7,
\end{align*}
\]

60
which reduces (when we use the facts that $27 \mid N$ and $8 \mid N$) to

$$n_{N,p} \leq \begin{cases} 
2 & \text{if } p = 2 \\
5 & \text{if } p = 7 \\
1 & \text{otherwise}
\end{cases} \tag{4.8}$$

In a similar fashion, we can use the congruences in the other rows together with
Theorem 15 to derive other conditions on $N$ for various upper bounds on the optimum minimum Hamming weight. The conditions and the corresponding upper bounds are summarized in Table 4.5 when $\nu = 3$.

<table>
<thead>
<tr>
<th>$n_{N,2}$</th>
<th>$n_{N,3}$</th>
<th>$n_{N,7}$</th>
<th>$n_{N,p} \neq 2, 3, 7$</th>
<th>Upper bound</th>
<th>Row from Table 4.3 and 4.4</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\leq 2$</td>
<td>$\leq 1$</td>
<td>$\leq 5$</td>
<td>$\leq 1$</td>
<td>28</td>
<td>2</td>
</tr>
<tr>
<td>$\leq 3$</td>
<td>$\leq 1$</td>
<td>$\leq 5$</td>
<td>$\leq 1$</td>
<td>36</td>
<td>3 and 4</td>
</tr>
<tr>
<td>$\leq 5$</td>
<td>$\leq 1$</td>
<td>$\leq 5$</td>
<td>$\leq 1$</td>
<td>44</td>
<td>5</td>
</tr>
<tr>
<td>$\leq 2$</td>
<td>$\leq 2$</td>
<td>$\leq 5$</td>
<td>$\leq 1$</td>
<td>44</td>
<td>6 and 7</td>
</tr>
</tbody>
</table>

Table 4.5: Summary of conditions on $N$ (derived using the congruences in Table 4.3 and 4.4) for various upper bounds on the optimum minimum Hamming weight when $\nu = 3$.

4.4 Generalized Linear Permutation Polynomial

In this section, we define a generalized linear permutation polynomial (GLPP) and show the relationship among QPP interleavers, ARP interleavers, DRP interleavers and GLPP interleavers. We first define a generalized linear permutation polynomial.

**Definition 57** Let $N$ be an interleaver length. Let $L < N$ such that $L|N$. Let $\mathcal{F} = \{f_0, f_1, \ldots, f_{L-1}\}$, where $f_i(x) = f_{i,0} + f_{i,1}x \mod N$ and $\text{Dom}(f_i) = \{l, L + l, 2L + l, \ldots, (\frac{N}{L} - 1)L + l\}$. $\mathcal{F}$ is a generalized linear permutation polynomial if and
only if $f_i$ is bijective $\forall i = 0, 1, \ldots, L - 1$ and $\text{Ran}(f_i) \cap \text{Ran}(f_j) = \phi, \forall i, j$ such that $i \neq j$.

We now show that a QPP is a GLPP.

**Lemma 58** A QPP is a GLPP. However a GLPP is not necessarily a QPP.

**Proof**: Let $f(x) = f_1 x + f_2 x^2 (\mod N)$ a QPP and $N = \prod_{p \in \mathcal{P}} p^{n_{N,p}}$. Let $L = \prod_{p \in \mathcal{P}} p^{n_{L,p}}$ such that

$$n_{L,2} = \max \left( \left\lceil \frac{n_{N,2} - n_{f,2} - 1}{2} \right\rceil, 0 \right)$$

$$n_{L,p} = \max \left( \left\lceil \frac{n_{N,p} - n_{f,p}}{2} \right\rceil, 0 \right), \text{ where } p \neq 2$$

Then either $\frac{N}{2} | (f_2 \cdot L^2)$ or $N | (f_2 \cdot L^2)$ holds. By definition, $L | N$ and $L < N$.

Let $x = Ly + l$, where $y = 0, 1, \ldots, \frac{N}{L} - 1$. Then,
1. $N|(f_2 \cdot L^2)$

\[
f(x) = f(Ly + l) = f_1(y) = f_1x + f_2x^2 = f_1(Ly + l) + f_2(Ly + l)^2 = f(l) + Ly(f_1 + 2 \cdot f_2 \cdot l) + f_2L^2y^2 = f(l) + Ly(f_1 + 2 \cdot f_2 \cdot l)
\]

Since $\gcd(f_1 + 2 \cdot f_2 \cdot l, N) = 1$, $f_1(y)$ is bijective. Consequently, $f_{l,1} = f_1 + 2f_2l$, $f_{l,0} = f(l) - lf_{l,1}$.

2. $\frac{N}{2}|(f_2 \cdot L^2)$

\[
f(x) = f(Ly + l) = f_1(y) = f_1x + f_2x^2 = f_1(Ly + l) + f_2(Ly + l)^2 = f(l) + Ly(f_1 + 2 \cdot f_2 \cdot l) + f_2L^2y^2 = f(l) + Ly(f_1 + 2 \cdot f_2 \cdot l) - f_2L^2y + f_2L^2y + f_2L^2y^2 = f(l) + Ly(f_1 + 2 \cdot f_2 \cdot l - L \cdot f_2)
\]

Consequently, $f_{l,1} = f_1 + 2f_2l - Lf_2$, $f_{l,0} = f(l) - lf_{l,1}$.

Thus, a QPP is a GLPP. For a QPP, $f_{l,1}^{QP} = f_1 + 2f_2l$ or $f_{l,1}^{QP} = f_1 + 2f_2l - Lf_2$.

Let us choose $f_{l,1}$ of a GLPP such that $f_{l,1} \neq f_1 + 2f_2l$ and $f_{l,1} \neq f_1 + 2f_2l - Lf_2$.

Thus, in general, a GLPP is not a QPP.
We now show that a DRP is a GLPP.

**Lemma 59** A DRP is a GLPP. However a GLPP is not necessarily a DRP.

**Proof**: Let \( I_1(x) \) be a bijective function with \( \text{Dom}(I_1) = \text{Ran}(I_1) = \{0, 1, 2, ..., L_1 - 1\} \) and let \( v_2(x) = I_2(x) = ax + b \) be a bijective function with \( \text{Dom}(I_2) = \text{Ran}(I_2) = \{0, 1, 2, ..., N - 1\} \). Let also \( I_3(x) \) be a bijective function with \( \text{Dom}(I_3) = \text{Ran}(I_3) = \{0, 1, 2, ..., L_2 - 1\} \). Let \( L \) be \( \text{lcm}(L_1, L_2) \) \([12]\). Then,

\[
\begin{align*}
  f(Ly + l) &= v_3(v_2(v_1(Ly + l))) \\
  &= v_3 \left( v_2 \left( aL_1 \left[ \frac{Ly + l}{L_1} \right] + I_1(l \pmod{L_1}) \right) \right) \\
  &= v_3 \left( v_2 \left( Ly + L_1 \left[ \frac{l}{L_1} \right] + I_1(l \pmod{L_1}) \right) \right) \\
  &= v_3 \left( aL_1 \left[ \frac{l}{L_1} \right] + aI_1(l \pmod{L_1}) + b \right) \\
  &= L_2 \left\lfloor aL_2 \left[ \frac{Ly + aL_1 \left[ \frac{l}{L_1} \right] + aI_1(l \pmod{L_1}) + b}{L_2} \right] \right\rfloor \\
  &\quad + I_3 \left( aL_1 \left[ \frac{l}{L_1} \right] + aI_1(l \pmod{L_1}) + b \pmod{L_2} \right) \\
  &= aL_1 \left[ \frac{l}{L_1} \right] + aI_1(l \pmod{L_1}) + b \pmod{L_2} \\
  &\quad + I_3 \left(aL_1 \left[ \frac{l}{L_1} \right] + aI_1(l \pmod{L_1}) + b \pmod{L_2} \right) \\
\end{align*}
\]

Consequently, \( f_{l,1} = a \),

\[
\begin{align*}
  f_{l,0} &= L_2 \left\lfloor aL_2 \left[ \frac{l}{L_1} \right] + aI_1(l \pmod{L_1}) + b \right\rfloor \\
  &\quad + I_3 \left(aL_1 \left[ \frac{l}{L_1} \right] + aI_1(l \pmod{L_1}) + b \pmod{L_2} \right) - al \\
\end{align*}
\]
Thus, a DRP is a GLPP. However, a GLPP is not necessarily a DRP, since for a DRP, \( f_{l,1} = a, \forall l \), but for a GLPP, it is not.

We now show that an ARP is a GLPP.

**Lemma 60** An ARP is a GLPP. However a GLPP is not necessarily an ARP.

*Proof:* Let \( \alpha \) be an 1 by \( L_3 \) vector and \( \beta \) be an 1 by \( L_4 \) vector. Let also \( c \) be a constant. Then,

\[
 f(Ly + l) = f_1(Ly + l) + c + f_1 \alpha_l \cdot (Ly + l \pmod{L_3}) + \beta_l \cdot (Ly + l \pmod{L_4})
\]

Consequently, \( f_{l,1} = f_1 \), \( f_{l,0} = c + f_1 \alpha_l \cdot (Ly + l \pmod{L_3}) + \beta_l \cdot (Ly + l \pmod{L_4}) \).

Thus, an ARP is a GLPP. A GLPP is not necessarily an ARP, since for an ARP, \( f_{l,1} = f_1, \forall l \), but for a GLPP, it is not.

We now show that a permutation polynomial (PP) is not a GLPP in general.

**Lemma 61** An \( n \)th order PP is not necessarily a GLPP and a GLPP is not necessarily an \( n \)th order PP.

*Proof:*

\((\Rightarrow)\)

Let \( N \) be a prime number. Since \( L | N \) and \( L < N \), \( L = 1 \). Thus only trivial linear PPs exist for this length. However, it is shown that there exists an \( n \)th order PP [48] and it is not a GLPP.

\((\Leftarrow)\)

Let \( K | N \). Then for \( n \)th order PP, \( K | f(K) \), however, for a GLPP, this is not necessarily true.

In Figure 4.8, Lemmas 58, 59, 60, and 61 are shown.
4.5 Algorithms for Finding an Upper Bound on the Minimum Distance of Turbo Codes Designed with QPP interleavers

In the following lemma, we show that a QPP can be decomposed into two linear interleavers. As a consequence of it, some QPP interleavers have input weight 4 patterns.

**Lemma 62** Suppose that $N = \prod_{p \in P} p^{n_{N,p}}$, where $n_{N,2} \geq 1$ and $n_{N,p} \leq 1$ for all $p$. Let also be $n_{f,2} \geq n_{N,2} - 3$. Then, a QPP interleaver can be decomposed into two linear interleavers.

**Proof:** By Lemma 58, $n_{f,p} \geq 1$ so $n_{L,p} = 0$, where $p \neq 2$. Consequently, $L = 2$ for all QPPs that satisfy the assumption.
Lemma 63 Let a QPP satisfy Lemma 62. Let also be the encoder of each constituent codes be 1101/1011. Then, $d_{\text{min}} \leq 44$.

Proof: There are two linear interleavers. The sub-interleavers behave exactly the same as a linear interleaver. Consequently, input weight 4 patterns exist [34]. Thus in the first and second constituent code, the error pattern has weight 10. There are four error pattern like this and the input sequence has weight 4. Consequently, the weight is 44.
For example, let \( N = 2^{12} \). Then for all the QPPs \( f(x) = f_1 x + c \cdot 2^9 x^2 \), where \( c \) is any number, \( d_{\text{min}} \leq 44 \).

In the following, we first state lemmas for input weight 6 patterns. Lemmas and algorithms for input weight 4 and 8 can be obtained using similar methods. Let us first define formal derivative for \( f(x) \).

**Definition 64** Let \( f(x) = \sum_{k=1}^{K} f_k x^k \pmod{N} \). \( f'(x) \) is a formal derivative for \( f(x) \) if and only if \( f(x) = \sum_{k=1}^{K} k \cdot f_k x^{k-1} \pmod{N} \).

By Definition 64, for a QPP \( f(x) = f_1 x + f_2 x^2 \), a formal derivative \( f'(x) \) is \( f(x) = f_1 + 2f_2 x \).

**Lemma 65** Consider an input weight 6 pattern in Figure 4.9. Let \( f(x) \) has \( L \) sub-linear interleavers. Then finding the input weight 6 pattern is equivalent to solving a system of linear congruences

\[
\begin{align*}
Ez &\equiv d \pmod{N} \\
y &\equiv V^{-1}z \pmod{N} \\
x &\equiv Ly + i \pmod{N},
\end{align*}
\]
where

\[
E = \begin{bmatrix}
-L \cdot f'(i_1) & 0 & 0 \\
0 & -L \cdot f'(i_2 + m_2 t) & 0 \\
0 & 0 & L \{ -f'(i_1 + m_1 t) \cdot f'(i_2) \cdot f'(i_1 + m_1 t) + f'(i_1) \cdot f'(i_2 + m_2 t) \cdot f'(i_3 + m_3 t) \}
\end{bmatrix}
\]

\[
V = \begin{bmatrix}
1 & -\frac{f'(i_2)}{f'(i_1)} & 0 \\
0 & 1 & -\frac{f'(i_3 + m_3 t)}{f'(i_2 + m_2 t)} \\
0 & 0 & 1
\end{bmatrix},
\]

\[
U = \begin{bmatrix}
1 & 0 & 0 \\
0 & \frac{f'(i_1 + m_1 t)}{f'(i_1)} & -\frac{f'(i_1 + m_1 t) \cdot f'(i_2)}{f'(i_1) \cdot f'(i_2 + m_2 t)} \\
0 & 1 & 0
\end{bmatrix},
\]

\[
b = [f(i_1) - f(i_2) + n_1 t, f(i_2 + m_2 t) - f(i_3 + m_3 t) + n_3 t, f(i_1 + m_1 t) - f(i_3) + n_2 t]^T,
\]

\[
d = U^{-1}b, \ i = [i_1, i_2, i_3]^T.
\]

**Proof:** See Appendix C.6.

For input weight 4 and 8 patterns, we have the following results.

![Figure 4.10: An input weight 4 pattern.](image-url)
Lemma 66 Consider an input weight $4$ pattern in Figure 4.10. Let $f(x)$ has $L$ sub-linear interleavers. Then finding the input weight $4$ pattern is equivalent to solving a system of linear congruences

\[ Ez \equiv d \pmod{N} \]
\[ y \equiv V^{-1}z \pmod{N} \]
\[ x \equiv Ly + i \pmod{N} \]

where

\[ E = \begin{bmatrix} -L \cdot f'(i_1) & 0 \\ 0 & -L \cdot \left\{ \frac{f'(i_1 + m_1 t)}{f'(i_1)} \cdot f'(i_2) - f'(i_2 + m_2 t) \right\} \end{bmatrix} \] \hspace{1cm} (4.9)

\[ V = \begin{bmatrix} 1 & -f'(i_2) \\ 0 & 1 \end{bmatrix} \] \hspace{1cm} (4.10)

\[ U = \begin{bmatrix} 1 & f'(i_1 + m_1 t) \\ -\frac{f'(i_1)}{f'(i_1)} & 1 \end{bmatrix} \]

\[ \mathbf{b} = [f(i_1) - f(i_2) + n_1 t, f(i_1 + m_1 t) - f(i_2 + m_2 t) + n_2 t]^T, \]

\[ \mathbf{d} = U^{-1}\mathbf{b}, \; \mathbf{i} = [i_1, i_2]^T. \]

Proof: See Appendix C.6. Lemma 66 can be proved using a similar method.

Lemma 67 Consider an input weight $8$ pattern in Figure 4.11. Let $f(x)$ has $L$ sub-linear interleavers. Then finding the input weight $8$ pattern is equivalent to solving a system of linear congruences

\[ Ez \equiv d \pmod{N} \]
\[ y \equiv V^{-1}z \pmod{N} \]
\[ x \equiv Ly + i \pmod{N} \]
Figure 4.11: An input weight 8 pattern.

where $E$ is a 4 by 4 diagonal matrix such that

\[
e_1 = -L \cdot f'(i_1)
\]
\[
e_2 = -L \cdot f'(i_2 + m_2 t)
\]
\[
e_3 = -L \cdot f'(i_3 + m_3 t)
\]
\[
e_4 = -L \cdot \left\{ \frac{f'(i_1 + m_1 t)}{f'(i_1)} \cdot \frac{f'(i_2)}{f'(i_2 + m_2 t)} \cdot \frac{f'(i_3)}{f'(i_3 + m_3 t)} - \frac{f'(i_3 + m_3 t)}{f'(i_3 + m_3 t)} \cdot f'(i_4 + m_4 t) \right\}
\]

\[
V = \begin{bmatrix}
1 & -\frac{f'(i_2)}{f'(i_1)} & 0 & 0 \\
0 & 1 & 0 & -\frac{f'(i_4)}{f'(i_3 + m_3 t)} \\
0 & 0 & 1 & -\frac{f'(i_4 + m_4 t)}{f'(i_4 + m_4 t)} \\
0 & 0 & 0 & 1
\end{bmatrix}
\]

\[
U = \begin{bmatrix}
1 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 \\
0 & 0 & 1 & 0 \\
\frac{f'(i_1 + m_1 t)}{f'(i_1)} & -\frac{f'(i_1 + m_1 t) \cdot f'(i_2)}{f'(i_1) \cdot f'(i_2 + m_2 t)} & \frac{f'(i_3)}{f'(i_3 + m_3 t)} & 1
\end{bmatrix}
\]

\[
b = [f(i_1) - f(i_2) + n_1 t, f(i_2 + m_2 t) - f(i_4) + n_3 t, f(i_3 + m_3 t) - f(i_4 + m_4 t) + n_4 t, f(i_1 + m_1 t) - f(i_3) + n_2 t]^T,
\]

\[
d = U^{-1}b, i = [i_1, i_2, i_3, i_4]^T
\]
Proof: See Appendix C.6. Lemma 67 can be using a similar method.

In Algorithm 4.6, algorithms for finding the weight and multiplicities generated by input weight $2k$ patterns are shown. Let an interleaver length be $N$, and the num-

<table>
<thead>
<tr>
<th>Algorithms for finding the weight distribution generated by input weight $2k$ patterns</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Input</strong>: An estimated upper bound</td>
</tr>
<tr>
<td>1. Find all non-isomorphic graphs for given input weight $2k$ patterns.</td>
</tr>
<tr>
<td>2. Find sets of $(m_1, m_2, ..., m_k)$ and $(n_1, n_2, ..., n_k)$ that may generate codewords with weights less than or equal to the given estimated upper bound.</td>
</tr>
<tr>
<td>3. For $i_1 = 0 : L - 1$</td>
</tr>
<tr>
<td>For $i_2 = 0 : L - 1$</td>
</tr>
<tr>
<td>...</td>
</tr>
<tr>
<td>For $i_K = 0 : L - 1$</td>
</tr>
<tr>
<td>(1) For each candidate $(m_1, m_2, ..., m_k)$ and $(n_1, n_2, ..., n_k)$, check if there are solutions for $\mathbf{Ez} \equiv \mathbf{d} \pmod{N}$.</td>
</tr>
<tr>
<td>If yes, find $\mathbf{y}$ and $\mathbf{x}$. Save $\mathbf{x}$ and increase multiplicity by 1.</td>
</tr>
<tr>
<td>If not, go to (1).</td>
</tr>
<tr>
<td>end</td>
</tr>
<tr>
<td>end</td>
</tr>
<tr>
<td>end</td>
</tr>
</tbody>
</table>

Table 4.6: Algorithm: computation of weights generated by input weight $2k$ patterns.

number of possible sets of $(m_1, m_2, ..., m_k)$ and $(n_1, n_2, ..., n_k)$ be $H$. If we use a brute-force search, the number of candidate codewords are $H \cdot N$. Using methods in Lemmas 66, 65 and 67, the number of candidate codewords are $H \cdot L^K$. Since $L \ll N$, $L^K < N$. Specifically, for most 3GPP LTE interleaver, $L = 4$. Consequently, for input weight 6 patterns and interleaver length $N = 4096$, $L^K = 64 < N = 4096$. In Table 4.7, upper bounds on the minimum distance of turbo codes designed with QPP interleavers using input weight 4, 6 pattern analysis are shown.
<table>
<thead>
<tr>
<th>$N$</th>
<th>$f_1$</th>
<th>$f_2$</th>
<th>UB(4)</th>
<th>multiplicity(4)</th>
<th>UB(6)</th>
<th>multiplicity(6)</th>
</tr>
</thead>
<tbody>
<tr>
<td>4032</td>
<td>127</td>
<td>168</td>
<td>52</td>
<td>4032</td>
<td>50</td>
<td>8064</td>
</tr>
<tr>
<td>4096</td>
<td>31</td>
<td>64</td>
<td>44</td>
<td>512</td>
<td>50</td>
<td>8704</td>
</tr>
<tr>
<td>4160</td>
<td>33</td>
<td>130</td>
<td>60</td>
<td>2080</td>
<td>58</td>
<td>16640</td>
</tr>
<tr>
<td>4224</td>
<td>43</td>
<td>264</td>
<td>60</td>
<td>8448</td>
<td>50</td>
<td>8448</td>
</tr>
<tr>
<td>4288</td>
<td>33</td>
<td>134</td>
<td>60</td>
<td>2144</td>
<td>58</td>
<td>17152</td>
</tr>
<tr>
<td>4352</td>
<td>477</td>
<td>408</td>
<td>44</td>
<td>1088</td>
<td>50</td>
<td>9792</td>
</tr>
<tr>
<td>4416</td>
<td>35</td>
<td>138</td>
<td>60</td>
<td>2208</td>
<td>58</td>
<td>17664</td>
</tr>
<tr>
<td>4480</td>
<td>233</td>
<td>280</td>
<td>52</td>
<td>4480</td>
<td>50</td>
<td>8960</td>
</tr>
<tr>
<td>4544</td>
<td>357</td>
<td>142</td>
<td>68</td>
<td>4544</td>
<td>58</td>
<td>18176</td>
</tr>
<tr>
<td>4608</td>
<td>337</td>
<td>480</td>
<td>44</td>
<td>768</td>
<td>50</td>
<td>9984</td>
</tr>
<tr>
<td>4672</td>
<td>37</td>
<td>146</td>
<td>60</td>
<td>2336</td>
<td>58</td>
<td>18688</td>
</tr>
<tr>
<td>4736</td>
<td>71</td>
<td>444</td>
<td>76</td>
<td>8288</td>
<td>66</td>
<td>37888</td>
</tr>
<tr>
<td>4800</td>
<td>71</td>
<td>120</td>
<td>52</td>
<td>2400</td>
<td>50</td>
<td>9600</td>
</tr>
<tr>
<td>4864</td>
<td>37</td>
<td>152</td>
<td>76</td>
<td>9728</td>
<td>50</td>
<td>9728</td>
</tr>
<tr>
<td>4928</td>
<td>39</td>
<td>462</td>
<td>44</td>
<td>1232</td>
<td>50</td>
<td>616</td>
</tr>
<tr>
<td>4992</td>
<td>127</td>
<td>234</td>
<td>60</td>
<td>1248</td>
<td>58</td>
<td>9984</td>
</tr>
<tr>
<td>5056</td>
<td>39</td>
<td>158</td>
<td>60</td>
<td>2528</td>
<td>58</td>
<td>20224</td>
</tr>
<tr>
<td>5120</td>
<td>39</td>
<td>80</td>
<td>44</td>
<td>640</td>
<td>50</td>
<td>10880</td>
</tr>
<tr>
<td>5184</td>
<td>31</td>
<td>96</td>
<td>44</td>
<td>6912</td>
<td>50</td>
<td>1728</td>
</tr>
<tr>
<td>5248</td>
<td>113</td>
<td>902</td>
<td>68</td>
<td>2624</td>
<td>58</td>
<td>10496</td>
</tr>
<tr>
<td>5312</td>
<td>41</td>
<td>166</td>
<td>60</td>
<td>2656</td>
<td>58</td>
<td>21248</td>
</tr>
<tr>
<td>5376</td>
<td>251</td>
<td>336</td>
<td>60</td>
<td>16128</td>
<td>50</td>
<td>10752</td>
</tr>
<tr>
<td>5440</td>
<td>43</td>
<td>170</td>
<td>60</td>
<td>2720</td>
<td>58</td>
<td>21760</td>
</tr>
<tr>
<td>5504</td>
<td>21</td>
<td>86</td>
<td>60</td>
<td>1376</td>
<td>58</td>
<td>11008</td>
</tr>
<tr>
<td>5568</td>
<td>43</td>
<td>174</td>
<td>60</td>
<td>2784</td>
<td>58</td>
<td>22272</td>
</tr>
<tr>
<td>5632</td>
<td>45</td>
<td>176</td>
<td>76</td>
<td>11264</td>
<td>50</td>
<td>11264</td>
</tr>
<tr>
<td>5696</td>
<td>45</td>
<td>178</td>
<td>60</td>
<td>2848</td>
<td>58</td>
<td>22784</td>
</tr>
<tr>
<td>5760</td>
<td>161</td>
<td>120</td>
<td>60</td>
<td>4320</td>
<td>50</td>
<td>11520</td>
</tr>
<tr>
<td>5824</td>
<td>89</td>
<td>182</td>
<td>44</td>
<td>1456</td>
<td>50</td>
<td>728</td>
</tr>
<tr>
<td>5888</td>
<td>323</td>
<td>184</td>
<td>76</td>
<td>11776</td>
<td>50</td>
<td>11776</td>
</tr>
<tr>
<td>5952</td>
<td>47</td>
<td>186</td>
<td>60</td>
<td>2976</td>
<td>58</td>
<td>23808</td>
</tr>
<tr>
<td>6016</td>
<td>23</td>
<td>94</td>
<td>60</td>
<td>1504</td>
<td>58</td>
<td>12032</td>
</tr>
<tr>
<td>6080</td>
<td>47</td>
<td>190</td>
<td>60</td>
<td>3040</td>
<td>58</td>
<td>24320</td>
</tr>
<tr>
<td>6144</td>
<td>263</td>
<td>480</td>
<td>76</td>
<td>3072</td>
<td>50</td>
<td>12288</td>
</tr>
</tbody>
</table>

Table 4.7: Upper bounds on the minimum distance of turbo codes designed with QPP interleavers using input weight 4, 6 pattern analysis ($N > 4000$, 8 states).
4.6 Conclusion

In this chapter, we considered an upper bound on the minimum distance of turbo codes designed with QPP interleavers. It was observed that the most frequent patterns are of input weight $2^k$, where $m = 1, 2, 3, \ldots$ for QPP [33]. It was shown that for input weight 2, 4, 6 patterns, an upper bound on the minimum distance of turbo codes designed with QPP can be found using algebraic methods. Using these methods, we have shown an upper bound on the minimum distance of turbo codes designed with some 3GPP LTE interleavers. For input weight 4, 6 and 8 patterns, we have proposed efficient algorithms for finding an upper bound on the minimum distance of turbo codes designed with QPP interleavers.
CHAPTER 5

A SIMPLE COEFFICIENT TEST FOR CUBIC PERMUTATION POLYNOMIALS OVER INTEGER RINGS

5.1 Introduction

In this chapter, we show the necessary and sufficient condition for a cubic polynomial to be a permutation polynomial.

As shown by Sun and Takashita [33] and Takashita [36], a class of structured interleavers for turbo codes based on permutation polynomials over integer rings exhibit excellent performance. Moreover, PP-based turbo codes have completely algebraic structure and efficient implementation [41]. Therefore a deeper understanding of PP-based interleavers is desirable. PPs over finite fields have been studied in [48]. In [46], PPs over $\mathbb{Z}_{2^w}$ are investigated and the necessary and sufficient condition for the coefficients of a polynomial to be a PP is given when $N = 2^w$. For arbitrary $N$, a simple necessary and sufficient coefficient test is given in [33, 35], but only up to QPPs. However, as argued in [36], simple tests for PPs of degrees larger than two deserve attention in turbo coding. In this chapter, we find a simple necessary and sufficient coefficient test for cubic PPs. This chapter is organized as follows. In Section 5.2, we briefly review the necessary and sufficient condition for a general PP. The proposed
coefficient test for a cubic PP is derived in Section 5.3, and examples are given in Section 5.4. Finally, conclusions are discussed in Section 5.5.

5.2 Permutation Polynomial over Integer Rings

We use the same notation $P$ and $n_{N,p}$ defined in Chapter 1 and 2. For a general $N$, the necessary and sufficient condition for a polynomial to be a permutation polynomial is given in the following theorem.

**Theorem 68 ([33])** For any $N = \prod_{p \in P} p^{n_{N,p}}$, $f(x)$ is a PP modulo $N$ if and only if $f(x)$ is also a PP modulo $p^{n_{N,p}}$, $\forall p$ such that $n_{N,p} \geq 1$.

With this theorem, verifying whether a polynomial is a PP modulo $N$ reduces to verifying the polynomial modulo each $p^{n_{N,p}}$ factor of $N$. For each $p^n$ factor of $N$, the necessary and sufficient condition is shown in the following theorem.

**Theorem 69 ([45])** $f(x)$ is a PP modulo $p^n$, with $n > 1$, if and only if $f(x)$ is a PP modulo $p$ and $f'(x) \neq 0 \pmod{p}$, for every integer $x$.

Let a cubic polynomial of the form $f(x) = f_1x + f_2x^2 + f_3x^3 \pmod{N}$. We are setting a possible constant coefficient $f_0$ to zero without loss of generality. Our aim is to provide a direct test on the coefficients of $f(x)$ that ensures the necessary and sufficient condition in Theorem 68.
5.3 A Simple Coefficient Test for Cubic PPs

The necessary and sufficient condition on the coefficients of a cubic polynomial \( f(x) \pmod{N} \) to be a permutation polynomial can be tested by the following three-step algorithm:

1. Factor \( N \) as \( \prod_{p \in \mathbb{P}} p^{n_{N,p}} \).

2. For each \( p \) and the corresponding exponent \( n_{N,p} \) in the previous step, test if the conditions in Table 5.1 are satisfied.

3. \( f(x) \) is a cubic permutation polynomial if and only if all tests in step 2 are satisfied.

The efficiency of the algorithm is apparent. While a brute force test requires the evaluation of a polynomial at \( N \) points, the above test only requires a small number of computations for each distinct factor of \( N \) (for practical values of \( N \) in turbo coding, the complexity of the factorization of \( N \) is negligible). Table 5.1 can be interpreted as a simpler equivalent test for Theorem 69. The proof that Table 5.1 is equivalent to Theorem 69 follows from the remaining of this section. An interesting observation is that Table I (or step 2) splits the set of primes into four disjoint subsets for the proposed test. Each disjoint subset is treated in Sections 5.3.1, 5.3.2, 5.3.3, and 5.3.4.

5.3.1 \( p = 2 \)

For \( p = 2 \), a simple coefficient test for an arbitrary-degree PP is well known and derived in [46]. The conditions for cubic PPs are listed in Table 5.1.
5.3.2  $p = 3$

$p = 3$ and $n = 1$

**Theorem 70** $f(x) = f_1 x + f_2 x^2 + f_3 x^3 \pmod{3}$ is a PP if and only if $f_1 + f_3 \neq 0$ and $f_2 = 0 \pmod{3}$.

**Proof:** Clearly $f(0) = 0$, then $f(x)$ is a PP if and only if $f(1) \neq 0$, $f(2) \neq 0$, and $f(1) \neq f(2) \pmod{3}$. By plugging the values into the function, we have $f_1 + f_2 + f_3 \neq 0$, $2f_1 + f_2 + 2f_3 \neq 0$, and $f_1 + f_3 \neq 0 \pmod{3}$, respectively. The first two conditions lead to $f_2 = 0 \pmod{3}$ by inspection of the coefficients.

$p = 3$ and $n > 1$

**Theorem 71** $f(x) = f_1 x + f_2 x^2 + f_3 x^3 \pmod{3^n}$, with $n > 1$, is a PP if and only if $f_1 + f_3 \neq 0$, $f_2 = 0$, and $f_1 \neq 0 \pmod{3}$.

**Proof:** Let $f(x)$ be a PP $(\pmod{3^n})$. By Theorem 69, $f(x)$ is a PP modulo 3 and $f'(x) = f_1 + 2f_2 x + 3f_3 x^2 = f_1 + 2f_2 x \neq 0 \pmod{3}$. Since $f(x)$ is a PP modulo 3, $f_1 + f_3 \neq 0$ and $f_2 = 0 \pmod{3}$, by Theorem 70. Let $x = 0$, $f'(0) = f_1 \neq 0 \pmod{3}$. Thus we have $f_1 \neq 0$, $f_2 = 0$, and $f_1 + f_3 \neq 0 \pmod{3}$. For the converse, let $f_1 \neq 0$, $f_2 = 0$, and $f_1 + f_3 \neq 0 \pmod{3}$. Since $f_1 + f_3 \neq 0$ and $f_2 = 0 \pmod{3}$, $f(x)$ is a PP modulo 3 by Theorem 70. The formal derivative is reduced to $f'(x) = f_1 + 2f_2 x = f_1 \neq 0 \pmod{3}$. Thus $f(x)$ is a PP $(\pmod{3^n})$, by Theorem 69.

5.3.3  $3|p - 1$

$3|p - 1$ and $n = 1$

**Theorem 72** Let $p$ be a prime, with $3|p - 1$. $f(x) = f_1 x + f_2 x^2 + f_3 x^3 \pmod{p}$ is a PP if and only if $f_1 \neq 0$, $f_2 = 0$, $f_3 = 0 \pmod{p}$.
Proof: By Corollary 7.5 of [48], for a valid PP, if \( d \mid p - 1 \), where \( d \) is the degree of the polynomial, then \( f_d = 0 \mod p \). Since \( 3 \mid p - 1 \), we have \( f_3 = 0 \mod p \). Similarly, because \( 2 \mid p - 1 \), for any \( p \), we have \( f_2 = 0 \mod p \). Then \( f(x) = f_1x \) is a PP if and only if \( f_1 \neq 0 \mod p \).

3\( |p - 1 \) and \( n > 1 \)

**Theorem 73** Let \( p \) be a prime, with \( 3 \mid p - 1 \). \( f(x) = f_1x + f_2x^2 + f_3x^3 \mod p^n \), with \( n > 1 \), is a PP if and only if \( f_1 \neq 0, f_2 = 0, f_3 = 0 \mod p \).

Proof: Let \( f(x) \) be a PP \( \mod p^n \), with \( 3 \mid p - 1 \). By Theorem 69, \( f(x) \) is a PP modulo \( p \), with \( 3 \mid p - 1 \). Then we have \( f_1 \neq 0, f_2 = 0, f_3 = 0 \mod p \), by Theorem 72. The other direction is similar to the argument in the proof of Theorem 71.

5.3.4 3\( \nmid p - 1 \) with \( p > 3 \)

3\( \nmid p - 1 \), with \( p > 3 \), and \( n = 1 \)

The following Lemma and Propositions are necessary for deriving the conditions in Theorem 78.

**Lemma 74** Let \( f(x) = f_1x + f_2x^2 + f_3x^3 \mod p \), where \( f_3 \neq 0 \mod p \). Then \( f(x) \) can be factored as \( f(x) = c(x + l)^3 + m \mod p \) if and only if \( f_2^2 = 3f_1f_3 \mod p \).

Proof: Suppose \( f(x) = c(x + l)^3 + m \). We want to prove that \( f_2^2 = 3f_1f_3 \mod p \).

To match the second and third-degree coefficients, let \( c = f_3, l = \frac{f_2}{3f_3}, \) and \( m = -\frac{f_3^3}{27f_3^3} \).

Then we have \( f(x) = f_3(x + \frac{f_2}{3f_3})^3 - \frac{f_3^3}{27f_3^2} = f_3x^3 + f_2x^2 + \frac{f_3^2}{3f_3}x \). Consequently, \( f_1 = \frac{f_2^2}{3f_3} \mod p \), which is equivalent to \( f_2^2 = 3f_1f_3 \mod p \). The other direction can be proved by the reverse way.
Proposition 75 ([48]) A polynomial \( f(x) \) is a PP (mod \( p \)) if and only if \( cf(x+l) + m \), with \( c \neq 0 \) (mod \( p \)), is a PP (mod \( p \)).

Definition 76 ([48]) Let \( \overline{f}(x) = \sum_{i=1}^{d} f_i x^i \) (mod \( p^n \)). We say that a polynomial \( \overline{f}(x) \) is in normalized form if \( f_d = 1 \), \( f(0) = 0 \), and when \( p \nmid d \), \( f_{d-1} = 0 \).

Proposition 77 ([48]) The only cubic PP in normalized form, with \( 3 \nmid p - 1 \), is \( \overline{f}(x) = x^3 \) (mod \( p \)).

Theorem 78 Let \( p \) be a prime with \( 3 \nmid p - 1 \). \( f(x) = f_1 x + f_2 x^2 + f_3 x^3 \) (mod \( p \)) is a PP if and only if \( f_2^2 = 3f_1 f_3 \) and \( f_3 \neq 0 \) (mod \( p \)).

Proof: By Propositions 75 and 77, all cubic PP(mod \( p \)), with \( 3 \nmid p - 1 \), can be obtained by \( \overline{f}(x) = x^3 \) and the formula \( cf(x+l) + m \). Thus, \( f(x) = f_1 x + f_2 x^2 + f_3 x^3 \) (mod \( p \)) is a PP if and only if \( f_2^2 = 3f_1 f_3 \) and \( f_3 \neq 0 \) (mod \( p \)), by Lemma 74.

\( 3 \nmid p - 1 \), with \( p > 3 \), and \( n > 1 \)

To derive the conditions of Theorem 81, the following Proposition and Lemma are necessary.

Proposition 79 The only solution to \( y^2 = 0 \) (mod \( p \)) is \( y = 0 \) (mod \( p \)).

Lemma 80 Let \( p > 3 \) be a prime, with \( 3 \nmid p - 1 \). If \( f_2^2 = 3f_1 f_3 \) (mod \( p \)), with \( f_3 \neq 0 \) (mod \( p \)), then the equation \( f'(x) = f_1 + 2f_2 x + 3f_3 x^2 = 0 \) (mod \( p \)) has a solution of \( x = -\frac{f_2}{3f_3} \) (mod \( p \)).

Proof: By ”completing the square,” the equation \( f'(x) = f_1 + 2f_2 x + 3f_3 x^2 = 0 \) (mod \( p \)) is transformed to \( y^2 = u \), with \( y = x + \frac{f_2}{3f_3} \) and \( u = \frac{1}{(3f_3)^2} (f_2^2 - 3f_1 f_3) \). Since \( u = \frac{1}{(3f_3)^2} (f_2^2 - 3f_1 f_3) = 0 \), the solution to \( y^2 = u \) is \( y = 0 \), by Proposition 79. Hence \( x = y - \frac{f_2}{3f_3} = -\frac{f_2}{3f_3} \) (mod \( p \)).
\[ p = 2 \quad n = 1 \quad (f_1 + f_2 + f_3) \text{ is odd.} \]
\[ n > 1 \quad f_1 \text{ is odd, } f_2 \text{ is even, and } f_3 \text{ is even.} \]

\[ p = 3 \quad n = 1 \quad f_1 + f_3 \neq 0, f_2 = 0 \pmod{3}. \]
\[ n > 1 \quad f_1 \neq 0, f_1 + f_3 \neq 0, f_2 = 0 \pmod{3}. \]

\[ 3 \nmid p - 1 \quad n = 1 \quad f_1 \neq 0, f_2 = 0, f_3 = 0 \pmod{p}. \]
\[ n > 1 \quad f_1 \neq 0, f_2 = 0, f_3 = 0 \pmod{p}. \]

Table 5.1: A simple coefficient test for cubic PP modulo \( p^n \)

**Theorem 81** Let \( p > 3 \) be a prime, with \( 3 \nmid p - 1 \). \( f(x) = f_1 x + f_2 x^2 + f_3 x^3 \pmod{p^n} \), with \( n > 1 \), is a PP if and only if \( f_1 \neq 0, f_2 = 0, f_3 = 0 \pmod{p} \).

**Proof:** Let \( f(x) \) be a PP \( \pmod{p^n} \), with \( 3 \nmid p - 1 \). By Theorem 69, \( f(x) \) is a PP modulo \( p \) and \( f'(x) = f_1 + 2 f_2 x + 3 f_3 x^2 \neq 0 \pmod{p} \). Since \( f(x) \) is a PP modulo \( p \), with \( 3 \nmid p - 1 \), there are only two possible cases: \( f_2^2 = 3 f_1 f_3 \), where \( f_3 \neq 0 \pmod{p} \), by Theorem 78 and \( f_1 \neq 0, f_2 = 0, f_3 = 0 \pmod{p} \), by the condition for a linear PP [33], [45]. Assume \( f_2^2 = 3 f_1 f_3 \), where \( f_3 \neq 0 \pmod{p} \), then by Lemma 80, there always exists a solution \( x = -\frac{f_2}{3f_3} \pmod{p} \) such that \( f'(x) = 0 \pmod{p} \), which is a contradiction to \( f'(x) \neq 0 \), for all \( x \). Thus the assumption fails. Now assume \( f_1 \neq 0, f_2 = 0, f_3 = 0 \pmod{p} \), then \( f'(x) = f_1 \neq 0 \pmod{p} \) holds trivially. The other direction is evident.

### 5.4 Examples

We present two examples to illustrate the proposed coefficient test in Section 5.3. Both examples are important because there are no QPPs for \( N = 11 \) and all QPPs for \( N = 420 \) degenerate to first degree PPs as discussed in [36].
Example 1: $N = 11$

In this case, $3 
mid (11 - 1)$ and $n = 1$. From Table 5.1, the condition for a cubic polynomial to be a PP is $f_2^2 = 3f_1f_3$, where $f_3 \neq 0 \pmod{11}$. For instance, $f(x) = 2x + x^2 + 2x^3 \pmod{11}$ is a valid PP, because $1^2 = 3 \cdot 2 \cdot 2 \pmod{11}$. However, $f(x) = x + x^2 + 2x^3 \pmod{11}$ is not a valid PP, because $1^2 \neq 3 \cdot 1 \cdot 2 \pmod{11}$.

Example 2: $N = 2^2 \times 3 \times 5 \times 7 = 420$

In this case, we must check the conditions for $p = 2, 3, 5, 7$ separately.

For instance, $f(x) = 5x + 14x^3 \pmod{420}$ is a valid PP, because all conditions in Table 5.1 are satisfied. However, $f(x) = 5x + 30x^2 + 14x^3 \pmod{420}$ is not a valid PP, because $f_2 \neq 0 \pmod{7}$.

5.5 Conclusion

Although PPs have been extensively studied, to the best of our knowledge, little is known about simple tests on the coefficients of a polynomial to verify if it is a PP over integer rings. In this chapter, we derived a simple necessary and sufficient condition on the coefficients of a cubic polynomial to make it a PP. The method is much more efficient than a brute force test. It is expected that the result of this paper will be useful in the understanding and design of PP-based turbo codes [36].
CHAPTER 6

CONCLUSIONS

In this dissertation, we investigated several topics in the analysis of turbo codes designed with permutation polynomial-based interleavers. In order to save both power consumption and storage, not only simple implementation of interleavers but also simple and efficient implementation of deinterleavers is of importance.

We derived a necessary and sufficient condition for the existence of a quadratic inverse for a quadratic permutation polynomial over integer rings. Further, we described a simple algorithm to find the coefficients of the quadratic inverse polynomial.

We also investigated a necessary and sufficient condition for the existence of an inverse for a quadratic permutation polynomial over integer rings along with a simple algorithm. Specifically, we have shown that most 3GPP LTE interleavers admit quadratic inverses, and when the interleavers do not admit quadratic inverses, the degrees of the inverses are low.

The minimum distance and its multiplicity of turbo codes are used to estimate the error performance in high SNR region. We derived an upper bound on the minimum distance of turbo codes designed with QPP interleavers using algebraic methods. We also investigated efficient algorithms for finding an upper bound on the minimum distance of turbo codes designed with QPP interleavers.
We derived a simple necessary and sufficient condition on the coefficients of a cubic polynomial to make it a PP. The method is much more efficient than a brute force test. It is expected that this result will be useful in the understanding and design of PP-based turbo codes.
A.1 Proof of Lemma 10

Let $N = \prod_{p \in \mathbb{P}} p^{\alpha_p}$. $g(x) = g_1x + g_2x^2 \pmod{N}$ inverts $f(x)$ at two points: $x = 1$ and $x = 2$ (in addition, $g(x)$ trivially inverts $f(x)$ at a third point $x = 0$) if and only if the following two congruences have at least one solution set $(g_1, g_2)$.

\begin{align*}
(g \circ f)(1) &= g(f_1 + f_2) \equiv g_1(f_1 + f_2) + g_2(f_1 + f_2)^2 \equiv 1. \quad (A.1) \\
(g \circ f)(2) &= g(2f_2 + 4f_2) = g_1(2f_1 + 4f_2) + g_2(2f_1 + 4f_2)^2 \equiv 2. \quad (A.2)
\end{align*}

By multiplying $(2f_1 + 4f_2)$ to equation (A.1) and $(f_1 + f_2)$ to equation (A.2), we get

\begin{align*}
g_1(2f_1 + 4f_2)(f_1 + f_2) + g_2(2f_1 + 4f_2)(f_1 + f_2)^2 &\equiv 2f_1 + 4f_2. \quad (A.3) \\
g_1(2f_1 + 4f_2)(f_1 + f_2) + g_2(2f_1 + 4f_2)^2(f_1 + f_2) &\equiv 2(f_1 + f_2). \quad (A.4)
\end{align*}

By subtracting equation (A.3) from equation (A.4),

\[
2g_2(f_1 + f_2)(f_1 + 2f_2)(f_1 + 3f_2) \equiv -2f_2 \pmod{N}. \quad (A.5)
\]

It can be shown that there exists at least one $g_2$ that satisfies equation (A.5) as follows.
If either $2 \nmid N$ or $4 \mid N$, suppose $\gcd(f_1 + f_2, N) \neq 1$. Then there is a prime number $p$ such that $p|(f_1 + f_2)$ and $p|N$. However, $p \nmid f_1$ and $p|f_2$ by Corollary 7. Thus, $p \nmid (f_1 + f_2)$ for all $p$'s such that $p|N$. A contradiction. Therefore $\gcd(f_1 + f_2, N) = 1$.

Similarly, $\gcd(f_1 + 2f_2, N) = 1$ and $\gcd(f_1 + 3f_2, N) = 1$. Thus, $\gcd((f_1 + f_2)(f_1 + 2f_2)(f_1 + 3f_2), N) = 1$. Consequently, if $2 \nmid N$, $\gcd(2(f_1 + f_2)(f_1 + 2f_2)(f_1 + 3f_2), N) = 1$ and if $4|N$, $\gcd(2(f_1 + f_2)(f_1 + 2f_2)(f_1 + 3f_2), N) = 2$.

If $2|N$ and $4 \nmid N$, $\gcd(2, N) = 2$ and $\gcd((f_1 + f_2)(f_1 + 2f_2)(f_1 + 3f_2), p) = 1$, where $p \neq 2$ by Corollary 7. Thus, $\gcd(2(f_1 + f_2)(f_1 + 2f_2)(f_1 + 3f_2), N) = 2$. In summary, if $N$ is an even number, we have exactly two solution sets, and if $N$ is an odd number, we have exactly 1 solution set by Theorem 8.

When $N$ is an even number, let $(g_{1,1}, g_{1,2})$ and $(g_{2,1}, g_{2,2})$ be the solution sets. Then,

$$g_{1,2}(f_1 + f_2)(f_1 + 2f_2)(f_1 + 3f_2) \equiv -f_2 \pmod{N/2}.$$ \hspace{1cm} (A.6)

and $g_{2,2} \equiv g_{1,2} + \frac{N}{2} \pmod{N}$ by Theorem 8.

When $N$ is an odd number, let $(g_1, g_2)$ be the solution set. Then, the congruence equation (A.5) can be rewritten as [45]

$$g_2(f_1 + f_2)(f_1 + 2f_2)(f_1 + 3f_2) \equiv -f_2 \pmod{N},$$ \hspace{1cm} (A.7)

since $\gcd(2, N) = 1$.

After computing $g_2$ using equation (A.7), or $g_{1,2}$, $g_{2,2}$ using equation (A.6) and Theorem 8, we can compute the corresponding $g_1$ or $g_{1,1}$, $g_{2,1}$ using equation (A.1) respectively. Specifically, it can be verified that $g_{2,1} \equiv g_{1,1} + \frac{N}{2} \pmod{N}$. Thus, for a given quadratic permutation polynomial $f(x)$, we can find at least one quadratic polynomial $g(x)$ that inverts the polynomial $f(x)$ at three points $x = 0, 1, 2$. 

86
A.2 Proof of Lemma 11

case a: $2 \nmid N$

In Lemma 10, $f(x)$ is a permutation polynomial. We can thus apply Corollary 7 to equation (A.5) and reducing it to (mod $p$) by Lemma 9, where $p$ is a prime number such that $p|N$.

$$2g_2 \cdot f_1 \cdot f_1 \equiv 0 \pmod{p}.$$  

Thus $p|g_2$, since $\gcd(2 \cdot f_1 \cdot f_1, p) = 1$, $\forall p$ such that $p|N$.

By multiplying $(2f_1+4f_2)^2$ to equation (A.1) and $(f_1+f_2)^2$ to equation (A.2), we get

$$g_1(2f_1 + 4f_2)^2(f_1 + f_2) + g_2(2f_1 + 4f_2)^2(f_1 + f_2)^2 \equiv (2f_1 + 4f_2)^2(A.8)$$

$$g_1(2f_1 + 4f_2)(f_1 + f_2)^2 + g_2(2f_1 + 4f_2)^2(f_1 + f_2)^2 \equiv 2(f_1 + f_2)^2. \ (A.9)$$

By subtracting equation (A.9) from equation (A.8),

$$g_1(f_1 + f_2)(2f_1 + 4f_2)(f_1 + 3f_2) \equiv 2f_1^2 + 12f_1f_2 + 14f_2^2 \pmod{N}. \ (A.10)$$

By Lemma 9, Corollary 7 and equation (A.10),

$$2g_1 \cdot f_1 \cdot f_1 \equiv 2f_1^2 \pmod{p}.$$  

Thus, if $p|g_1$, then $p|2f_1^2$, which is a contradiction from Corollary 7.

By Corollary 7, $g(x)$ is a permutation polynomial.

case b: $2|N$ and $4 \nmid N$

We apply Corollary 7. First we prove that $g_1(x)$ and $g_2(x)$ obtained in Lemma 10
are permutation polynomials modulo 2. Since \( f(x) \) is a quadratic permutation polynomial, \( f_1 + f_2 \) is an odd number from Lemma 2. Thus, \( (f_1 + f_2)^2 \) is odd. Let one solution set of Lemma 10 be \((g_{1,1}, g_{1,2})\). Suppose \( g_{1,1} + g_{1,2} \) is even, i.e., suppose both of \( g_{1,1} \) and \( g_{1,2} \) are even or odd numbers. Then the LHS of equation (A.1) becomes an even number. A contradiction, since an even number modulo an even number must be an even number but RHS is an odd number and \( N \) is even. Therefore, \( g_{1,1} + g_{1,2} \) must be an odd number. By Lemma 2, \( g_1(x) \) is a permutation polynomial modulo 2. Since \( g_{1,1} + g_{1,2} \) is odd and the second solution set is given as \( g_{2,1} \equiv g_{1,1} + \frac{N}{2} \pmod{N} \) and \( g_{2,2} \equiv g_{1,2} + \frac{N}{2} \pmod{N} \), \( g_{2,1} + g_{2,2} \) must be odd. Consequently, \( g_2(x) \) is a permutation polynomial modulo 2. For \( p \)'s such that \( p \neq 2 \), using a similar argument as in case (a), it can be easily verified that \( g_1(x) \) and \( g_2(x) \) are permutation polynomials.

case c: \( 4 | N \)

We apply Corollary 7. First we prove that \( g_1(x) \) and \( g_2(x) \) obtained in Lemma 10 are permutation polynomials modulo \( 2^n \) where \( n \geq 2 \).

\( f_2 \) is even by Corollary 5 and \( \frac{N}{2} \) is even since \( 4 | N \). By Corollary 5, \( f_1 + f_2, f_1 + 2f_2, f_1 + 3f_2 \) are all odd numbers. Thus \( (f_1 + f_2)(f_1 + 2f_2)(f_1 + 3f_2) \) is an odd number. Consequently, \( g_{1,2} \) must be an even number in equation (A.6) by reducing it \( \pmod{2} \) and using Lemma 9. From equation (A.1), since \( f_1 + f_2 \) is odd and \( g_{1,2}(f_1 + f_2)^2 \) is even, \( g_{1,1} \) must be an odd number. Finally, \( g_{2,1} \equiv g_{1,1} + \frac{N}{2} \pmod{N} \) is an odd number since \( g_{1,1} \) is an odd number, and \( g_{2,2} \equiv g_{1,2} + \frac{N}{2} \pmod{N} \) is an even number since \( g_{1,2} \) is an even number. Consequently, by Corollary 5, \( g_1(x) \) and \( g_2(x) \) are permutation polynomials.
modulo $2^n$, where $n \geq 2$. For $p$’s such that $p \neq 2$, using a similar argument as in case (a), it can be easily verified that $g_1(x)$ and $g_2(x)$ are permutation polynomials.

A.3 Proof of Lemma 12

case a: $2 \nmid N$

Suppose that $n_{f,p} < n_{N,p}$ and $n_{f,p} < n_{g,p}$, where $p$ is a prime number such that $p|N$. From Lemma 9 and equation (A.7)

\[ 0 \equiv -f_2 \pmod{p^\min(n_{g,p},n_{N,p})}. \]

A contradiction.

Now suppose that $n_{f,p} < n_{N,p}$ and $n_{g,p} < n_{f,p}$, again, from Lemma 9 and equation (A.7)

\[ g_2 \cdot f_1 \cdot f_1 \cdot f_1 \equiv 0 \pmod{p^{n_{f,p}}}. \]

The LHS cannot be 0, since $\gcd(f_1 \cdot f_1 \cdot f_1, p^{n_{f,p}}) = 1$ by Corollary 7. A contradiction. Thus $n_{g,p} = n_{f,p}$.

If $n_{f,p} \geq n_{N,p}$, from Lemma 9 and (A.7),

\[ g_2 \cdot f_1 \cdot f_1 \cdot f_1 \equiv 0 \pmod{p^{n_{N,p}}}, \]

which forces $n_{g,p} \geq n_{N,p}$.

case b: $2 | N$ and $4 \nmid N$

Using a similar argument as above, it is easily verified by using Lemma 9 and equation (A.6).
Using a similar argument as above, it is easily verified by using Lemma 9 and equation (A.6).

A.4 Proof of Lemma 13

(\implies)

Define \(T_0(x) = T(x) = t_1x + t_2x^2 + t_3x^3 + t_4x^4 \pmod{N}\) and \(T_n(x) = T_{n-1}(x + 1) - T_{n-1}(x), \forall n \geq 1\). If \(T(x) \equiv 0 \pmod{N}, \forall x \in [0, N-1]\) then \(T_n(x) \equiv 0 \pmod{N}, \forall x \in [0, N-1], \forall n \geq 0\). After some computation, it can be easily shown that

\[
T_1(x) = (t_1 + t_2 + t_3 + t_4) + (2t_2 + 3t_3 + 4t_4)x + (3t_3 + 6t_4)x^2 + 4t_4x^3 \equiv 0 \pmod{N}.
\]
\[
T_2(x) = (2t_2 + 6t_3 + 14t_4) + (6t_3 + 24t_4)x + 12t_4x^2 \equiv 0 \pmod{N}.
\]
\[
T_3(x) = (6t_3 + 36t_4) + 24t_4x \equiv 0 \pmod{N}.
\]

Consequently, in order to ensure \(T_3(x) \equiv 0 \pmod{N}\) for \(x \in [0, N-1]\),

\[
24t_4 \equiv 0 \pmod{N}.
\]
\[
6t_3 + 36t_4 \equiv 0 \pmod{N}.
\]

(\iff)

Define \(T_0(x), T_1(x), T_2(x)\) and \(T_3(x)\) as above. Then by assumption, \(T_3(x) \equiv 0 \pmod{N}, \forall x \in [0, N-1]\), and \(T_2(0) = T(2) - 2T(1) + T(0) \equiv 0 \pmod{N}\). By induction, \(T_2(x) \equiv 0 \pmod{N}, \forall x \in [0, N-1]\) since \(T_2(x + 1) = T_2(x) + T_3(x)\). By the same procedure, \(T_1(x) \equiv 0 \pmod{N}\) and \(T(x) = T_0(x) \equiv 0 \pmod{N}\).
A.5 Proof of Theorem 14

\[(g \circ f)(x) \equiv x \pmod{N}\] if and only if \(g(x)\) is the quadratic inverse polynomial of \(f(x)\).

\[(g \circ f)(x) \equiv g_1(f_1x + f_2x^2) + g_2(f_1x + f_2x^2)^2 \pmod{N} \equiv f_1g_1x + (f_2g_1 + f_1^2g_2)x^2 + 2f_1f_2g_2x^3 + f_2^2g_2x^4 \equiv x \pmod{N}.

Thus, \(g(x)\) is the quadratic inverse polynomial of \(f(x)\) if and only if the following condition is satisfied.

\[(f_1g_1 - 1)x + (f_2g_1 + f_1^2g_2)x^2 + 2f_1f_2g_2x^3 + f_2^2g_2x^4 \equiv 0 \pmod{N} \quad (A.11)\]

Let \(T(x) = (g \circ f)(x) - x = (f_1g_1 - 1)x + (f_2g_1 + f_1^2g_2)x^2 + 2f_1f_2g_2x^3 + f_2^2g_2x^4\). By Lemma 10, \(T(0) = g(f(0)) - 0 \equiv 0 \pmod{N}\), \(T(1) = g(f(1)) - 1 \equiv 0 \pmod{N}\), \(T(2) = g(f(2)) - 2 \equiv 0 \pmod{N}\). Applying Lemma 13, we get

\[24f_2^2g_2 \equiv 0 \pmod{N}.
\]

\[36f_2^2g_2 + 12f_1f_2g_2 \equiv 0 \pmod{N}.
\]

These can be further reduced to

\[12f_2g_2 \equiv 0 \pmod{N},\]

since \(\gcd(f_1 + f_2, N) = 1\), by Corollary 7.

A.6 Proof of Theorem 15

\[\Leftarrow\]

By Lemmas 10 and 11, a quadratic permutation polynomial \(f(x)\) has at least one
quadratic permutation polynomial \( g(x) = g_1x + g_2x^2 \pmod{N} \) that inverts \( f(x) \) at three points \( x = 0, 1, 2 \). Since it is required for a quadratic inverse polynomial to invert \( f(x) \) at these points, we only need to check whether \( g(x) \) is a quadratic inverse polynomial or not.

We show that if \( g(x) \) is a quadratic inverse polynomial, then the condition on \( n_{f,p} \), where \( p = 2 \), holds. The conditions for \( n_{f,p} \), where \( p \neq 2 \), can be done similarly.

If \( n_{N,2} = 0, 1 \), whether \( g(x) \) is a quadratic inverse or not, \( n_{f,2} \geq 0 \) trivially holds and this is why we do not need to determine \( n_{g,2} \) in Lemma 12, case (b) when \( n_{N,2} = 1 \).

If \( n_{N,2} = 2, 3, 4 \), by Corollary 7, \( n_{f,2} \geq 1 \), since \( f(x) \) is a permutation polynomial.

Suppose that \( g(x) \) is a quadratic inverse polynomial but \( n_{f,2} < \left\lceil \frac{n_{N,2} - 2}{2} \right\rceil \), for \( n_{N,2} \geq 5 \).

Since \( g(x) \) is a quadratic inverse polynomial, \( 12f_2g_2 \equiv 0 \pmod{N} \) holds by Theorem 14, i.e., \( \prod_{p \in P} p^{n_{f,p}p} | (2^2 \cdot 3 \cdot \prod_{p \in P} p^{n_{f,p}} \cdot \prod_{p \in P} p^{n_{g,p}}) \).

We divide it into two cases.

1. \( n_{N,2} \) is odd

   \[ \left\lceil \frac{n_{N,2} - 2}{2} \right\rceil = \frac{n_{N,2} - 1}{2}, \text{ thus } n_{f,2} \leq \frac{n_{N,2} - 1}{2} - 1. \]

   By Lemma 12, \( n_{g,2} = n_{f,2} \), thus

   \[ 2 + n_{f,2} + n_{g,2} \leq n_{N,2} - 1 < n_{N,2}, \text{ which is a contradiction since } N|12f_2g_2 \text{ implies } n_{N,2} \leq 2 + n_{f,2} + n_{g,2}. \]

2. \( n_{N,2} \) is even

   \[ \left\lceil \frac{n_{N,2} - 2}{2} \right\rceil = \frac{n_{N,2} - 2}{2}, \text{ thus } n_{f,2} \leq \frac{n_{N,2} - 2}{2} - 1. \]

   By Lemma 12, \( n_{g,2} = n_{f,2} \), thus

   \[ 2 + n_{f,2} + n_{g,2} \leq n_{N,2} - 2 < n_{N,2}, \text{ which is a contradiction since } N|12f_2g_2 \text{ implies } n_{N,2} \leq 2 + n_{f,2} + n_{g,2}. \]

Similarly, it can be shown that if \( g(x) \) is a quadratic inverse polynomial, then the conditions on \( n_{f,p} \), where \( p \neq 2 \) is satisfied.
We show that if the conditions on $n_{f,p}$ holds, then $12f_2g_2 \equiv 0 \pmod{N}$, i.e.,

\[
\prod_{p \in \mathcal{P}} p^{n_{N,p}} | (2^2 \cdot 3 \cdot \prod_{p \in \mathcal{P}} p^{n_{f,p}} \cdot \prod_{p \in \mathcal{P}} p^{n_{g,p}})
\]

holds. We only show that if the condition on $n_{f,p}$, where $p = 2$, is satisfied, $2^{n_{N,2}} | (2^2 \cdot 2^{n_{f,2}} \cdot 2^{n_{g,2}})$ holds and the case where $p \neq 2$ can be done similarly.

We divide it into three cases.

1. $n_{N,2} = 0, 1$

   If $n_{f,2} \geq 0$, then $n_{N,2} \leq 2 + n_{f,2}$ holds. Thus $2^{n_{N,2}} | (2^2 \cdot 2^{n_{f,2}} \cdot 2^{n_{g,2}})$

2. $n_{N,2} = 2, 3, 4$

   If $n_{f,2} \geq 1$, then as required by Lemma 12, $n_{g,2} \geq 1$. Thus, $n_{N,2} \leq 2 + n_{f,2} + n_{g,2}$ holds and consequently $2^{n_{N,2}} | (2^2 \cdot 2^{n_{f,2}} \cdot 2^{n_{g,2}})$

3. $n_{N,2} \geq 5$

   If $n_{N,2} \geq 5$, then $\left\lceil \frac{n_{N,2}-2}{2} \right\rceil > 1$. By Lemma 12, if $n_{N,2} - 1 > n_{f,2} > \left\lceil \frac{n_{N,2}-2}{2} \right\rceil$, then $n_{g,2} = n_{f,2}$. Consequently, if $n_{N,2}$ is even, $2 + n_{f,2} + n_{g,2} = 2 + 2 \cdot n_{f,2} \geq 2 + n_{N,2}-2 = n_{N,2}$, and if $n_{N,2}$ is odd, $2 + n_{f,2} + n_{g,2} = 2 + 2 \cdot n_{f,2} \geq 2 + n_{N,2}-1 > n_{N,2}$. Thus $2^{n_{N,2}} | (2^2 \cdot 2^{n_{f,2}} \cdot 2^{n_{g,2}})$. If $n_{f,2} \geq n_{N,2} - 1$, by Lemma 12, $n_{g,2} \geq n_{N,2} - 1$. Thus $2 + n_{f,2} + n_{g,2} \geq 2 \cdot n_{N,2} > n_{N,2}$ and consequently $2^{n_{N,2}} | (2^2 \cdot 2^{n_{f,2}} \cdot 2^{n_{g,2}})$.
APPENDIX B

PROOF OF LEMMAS IN CHAPTER 3

B.1 Proof of Lemma 33

We define $N - 1$ by $N - 1$ upper triangular matrices $V^{i,j}$ such that

$$v^{i,j}_{m,n} = \begin{cases} 
1 & \text{if } m = n \\
-v^{i,j}_{i,j} & \text{if } m = i, n = j \\
0 & \text{otherwise}
\end{cases}$$

We also define $N - 1$ by $N - 1$ upper triangular matrices $U^{i,j}$ such that

$$U^{i,j} = \begin{cases} 
A \cdot \prod_{k=2}^{i-1} \left\{ \prod_{l=1}^{k-1} V^{l,k} \right\} \cdot \prod_{l=1}^{j} V^{l,j} & \text{if } j = 2 \\
A \cdot V^{1,2} & \text{otherwise}
\end{cases},$$

where $j \geq 2$ and $j > i$.

We prove $U^{N-2,N-1} = A \cdot \prod_{k=2}^{N-1} \left\{ \prod_{l=1}^{k-1} V^{l,k} \right\} = UE$.

Since

$$V \cdot \prod_{k=2}^{N-1} \left\{ \prod_{l=1}^{k-1} V^{l,k} \right\} = V \cdot V^{1,2} \cdot V^{1,3} \cdot V^{2,3} \cdots V^{1,N-1} \cdot V^{2,N-1} \cdots V^{N-2,N-1} = I,$$

$$V^{-1} = \prod_{k=2}^{N-1} \left\{ \prod_{l=1}^{k-1} V^{l,k} \right\}.$$

Thus the desired result follows.

We use two-fold induction.
In the first induction, we show that $U_{j, j}^{j-1}, j \geq 2$ is as follows,

$$u_{m,n}^{j-1,j} = \begin{cases} n! \cdot u_{m,n} & \text{if } n \leq j \\ a_{m,n} & \text{if } j < n < N \end{cases}$$

If this argument holds, it is immediately shown that $U_{N-2,N-1}^{N-2} = UE$.

We first show that the above claim holds for $j = 2$.

By definition, $U_{1,2}^{1,2} = AV_{1,2}^{1,2}$. Since $v_{1,2}^{1,2} = -v_{1,2}$ and $v_{1,2} = y_{1,2}W_{1,2}^{1,2}z^2 = f_1 + f_2$,

$$u_{m,2}^{1,2} = -v_{1,2} \cdot a_{m,1} + a_{m,2}$$

$$= -(f_1 + f_2)(mf_1 + m^2f_2) + (mf_1 + m^2f_2)^2$$

$$= (mf_1 + m^2f_2) \cdot \{(m-1)f_1 + (m-1)(m+1)f_2\}$$

$$= 2! \cdot \binom{m}{2} \cdot (f_1 + mf_2)\{f_1 + (m+1)f_2\}$$

$$= 2! \cdot \binom{m}{2} \cdot \prod_{l=m}^{m+2-1} (f_1 + lf_2)$$

$$= 2! \cdot u_{m,2}^{1,2}$$

Since multiplying $V_{1,2}^{1,2}$ to $A$ does not change the other columns except the second, the above claim holds for $j = 2$.

Suppose that the above statement holds for $j$. We show that it also holds for $j + 1$.

In order for that, we use another induction and prove that following claim for $j \geq 3$.

$$u_{m,j}^{i,j} = \left[ \prod_{l=0}^{i} (m - l)\{f_1 + (m + l)f_2\} \right] \cdot y_{m,j}^{i+1,j}W_{i+1,j}^{i+1,j}z^j$$

When $i = 1$, the above claim holds, since

$$u_{m,j}^{1,j} = -v_{1,j} \cdot u_{m,1}^{j-2,j-1} + u_{m,j}^{j-2,j-1}$$

$$= -v_{1,j} \cdot u_{m,1}^{j-2,j-1} + a_{m,j}$$

$$= -y_{1,j}^{1,j}W_{1,j}^{1,j}z^j \cdot \binom{m}{1} (f_1 + mf_2) + (mf_1 + m^2f_2)^j$$

95
\[-(f_1 + f_2)^{j-1} \cdot (mf_1 + m^2f_2) + (mf_1 + m^2f_2)^j \]
\[= (mf_1 + m^2f_2) \cdot \{(mf_1 + m^2f_2)^{j-1} - (f_1 + f_2)^{j-1}\} \]
\[= (mf_1 + m^2f_2) \cdot \{(m-1)f_1 + (m-1)(m+1)f_2\} \cdot \sum_{l=0}^{j-2} \{(mf_1 + m^2f_2)^{j-2-l} \cdot (f_1 + f_2)^l\} \]
\[= m(m-1)(f_1 + mf_2)\{f_1 + (m+1)f_2\}y^{m,j}W^{2,j}z^j \]
\[= \left[ \prod_{l=0}^{1} (m-l)\{f_1 + (m+l)f_2\} \right] \cdot y^{m,j}W^{2,j}z^j \]

Note that \(U^{1,j} = U^{j-2,j-1}V^{1,j}\). Suppose that the above claim holds for \(i\).

In order to compute \(u_{m,j}^{i+1,j}\), we multiply \(V_{i+1,j}^{i,j}\) to \(U_{i,j}^{i,j}\), which is equivalent to multiply \(-v_{i+1,j}^{i,j}\) to \(u_{m,i+1}^{i,j}\) and add it to \(u_{m,j}^{i,j}\).

Thus,

\[u_{m,j}^{i+1,j} \]
\[= -v_{i+1,j}^{i,j} \cdot u_{m,i+1}^{i,j} + u_{m,j}^{i,j} \]
\[= -v_{i+1,j}^{i,j} \cdot (i+1)! \cdot u_{m,i+1}^{i,j} + u_{m,j}^{i,j} \]
\[= -y^{i+1,j}W^{i+1,j}z^j \cdot (i+1)! \cdot \left( \begin{array}{c} m+i \\ i+1 \end{array} \right) \cdot \prod_{l=m}^{m+i} (f_1 + lf_2) + \]
\[\left[ \prod_{l=0}^{i} (m-l)\{f_1 + (m+l)f_2\} \right] \cdot y^{m,j}W^{i+1,j}z^j \]
\[= -y^{i+1,j}W^{i+1,j}z^j \cdot \left[ \prod_{l=0}^{i} (m-l)\{f_1 + (m+l)f_2\} \right] + \]
\[\left[ \prod_{l=0}^{i} (m-l)\{f_1 + (m+l)f_2\} \right] \cdot y^{m,j}W^{i+1,j}z^j \]
\[= \left[ \prod_{l=0}^{i} (m-l)\{f_1 + (m+l)f_2\} \right] \cdot (y^{m,j} - y^{i+1,j})W^{i+1,j}z^j \]

96
\[ y^{m,j} - y^{i+1,j} = \left[ 1, mf_1 + m^2 f_2, \ldots, (mf_1 + m^2 f_2)^{j-1} \right] - \left[ 1, (i + 1)f_1 + (i + 1)^2 f_2, \ldots, ((i + 1)f_1 + (i + 1)^2 f_2)^{j-1} \right] = \{m - (i + 1)\} \{f_1 + (m + i + 1)f_2\} \cdot \left[ 0, 1, \ldots, \sum_{l=0}^{j-2} \{m(f_1 + m f_2)\}^{j-2-l} \{(i + 1)(f_1 + (i + 1)f_2)\}^l \right] = \{m - (i + 1)\} \{f_1 + (m + i + 1)f_2\} \cdot y^{m,j} W^{(i+1)} \]

Thus,
\[ u^{i+1,j}_{m,j} = \prod_{l=0}^{i} (m - l) \{f_1 + (m + l)f_2\} \cdot \left[ m - (i + 1) \right] \{f_1 + (m + i + 1)f_2\} \cdot y^{m,j} W^{(i+1)} W^{i+1,j} z^j \]

Consequently, the second induction holds.

When \( i = j - 1 \), \( y^{m,j} W^{j,j} z^j = 1 \). Thus \( u^{j-1,j}_{m,j} = \prod_{l=0}^{j-1} (m - l) \{f_1 + (m + l)f_2\} = j! \cdot \binom{m}{j} \cdot \prod_{l=m}^{m+j-1} (f_1 + l f_2) = j! \cdot u_{m,j} \). Consequently, the first induction holds.

By Corollary 24, \( \det(U) = \prod_{i=1}^{N-1} \prod_{l=i}^{2i-1} (f_1 + l f_2) \), \( \det(V) = 1 \) are units. Consequently, by Theorem 28, \( U \) and \( V \) are unimodular.

**B.2 Proof of Lemma 34**

Let \( f(x) = \sum_{l=1}^{K} f_l x^l \pmod{N} \). Let us recall the notation used in Theorem A.4.

**Definition 82 (Theorem A.4)** Let \( f_0(x) = f(x) = \sum_{l=1}^{K} f_l x^l \pmod{N} \). Let us define \( f_n(x) = f_{n-1}(x + 1) - f_{n-1}(x) \).

We first prove the following lemma.
Lemma 83  \( f(x) = \sum_{l=1}^{K} f_l x^l \pmod{N} \equiv 0 \pmod{N} \) if and only if \( f_n(0) \equiv 0 \pmod{N} \), \( \forall n = 0, 1, ..., K \).

Proof:

( \implies )

If \( f(x) \equiv 0 \pmod{N} \), then \( f_n(x) \equiv 0 \pmod{N} \), \( \forall n = 0, 1, ..., K \). Substituting \( x = 0 \) in each \( f_n(x) \), we have the desired result.

( \impliedby )

Suppose that \( f_n(0) \equiv 0 \pmod{N} \), \( \forall n = 0, 1, ..., K \) holds. Since \( f_K(x) \) is a constant, \( f_K(x) = f_K(0) = 0 \pmod{N} \), \( \forall x = 0, 1, ..., N - 1 \). Consider \( f_K(x) = f_{K-1}(x + 1) - f_{K-1}(x) \). Since \( f_{K-1}(0) = 0 \), \( f_{K-1}(1) = f_K(0) + f_{K-1}(0) = 0 \). By induction, \( f_{K-1}(x) = 0 \pmod{N} \), \( \forall x = 0, 1, ..., N - 1 \). Again by induction on \( n \), \( f(x) = 0 \pmod{N} \).

In the following lemma, we get \( f_n(x) \) explicitly.

Lemma 84  \( f_n(x) = \sum_{l=0}^{n} \binom{n}{l} \cdot (-1)^l f(x + n - l) \)

Proof:  We use induction. Let \( n = 0, 1 \). Then the above claim trivially holds.

Let \( n > 1 \) and suppose the claim holds. Then, \( f_n(x) = \sum_{l=0}^{n} \binom{n}{l} \cdot (-1)^l f(x + n - l) \) and \( f_n(x + 1) = \sum_{l=0}^{n} \binom{n}{l} \cdot (-1)^l f(x + 1 + n - l) \).

\[
\begin{align*}
f_{n+1}(x) &= f_n(x + 1) - f_n(x) \\
&= \binom{n}{0} f(x + 1 + n) + \sum_{l=1}^{n} \left\{ \binom{n}{l} (-1)^l - \binom{n}{l-1} (-1)^{l-1} \right\} f(x + 1 + n - l) \\
&\quad - \binom{n}{n} (-1)^n f(x) \\
&= \binom{n+1}{0} f(x + 1 + n) + \sum_{l=1}^{n} \left\{ \binom{n+1}{l} (-1)^l \right\} f(x + n + 1 - l)
\end{align*}
\]
\[ f(x) = \sum_{l=0}^{n+1} \binom{n+1}{l}(-1)^{n+1} f(x) \]

\[ \sum_{l=0}^{n+1} \binom{n+1}{l} \cdot (-1)^l f(x + n + 1 - l). \]

Note that by Pascal’s triangle,\[ \binom{n+1}{l} = \binom{n}{l} + \binom{n}{l-1}. \]

By induction, the claim holds.

**Corollary 85** \[ f(x) = \sum_{l=1}^{K} f_l x^l \pmod{N} \] if and only if \[ f(l) \equiv 0 \pmod{N}, \forall n = 0, 1, ..., K. \]

**Proof:** Let us define a \( K + 1 \) by \( K + 1 \) matrix \( D \) such that \[ d_{i,j} = \begin{cases} \binom{i-1}{i-j}(-1)^{i-j} & \text{if } i \geq j \\ 0 & \text{otherwise} \end{cases}. \]

Then by Lemma 84, it can be easily shown that \[ D \cdot [f_0(0), f_1(0), ..., f_K(0)]^T = [f(0), f(1), ..., f(K)]^T. \]

Consequently, \[ D \cdot [f_0(0), f_1(0), ..., f_K(0)]^T \equiv 0 \iff [f(0), f(1), ..., f(K)]^T \equiv 0. \]

Since \( D \) is unimodular, by Corollaries 31 and 32, the lemma holds.

Now we prove Lemma 34. Let \( f = [f_1, ..., f_K]^T \) and let \( A' \) such that \[ a'_{i,j} = i^j \pmod{N}. \]

Then \[ [f(0), f(1), ..., f(K)]^T \equiv 0 \iff A'x \equiv 0 \pmod{N}. \]
Note that \( f(0) \equiv 0 \pmod{N} \) trivially holds. In Lemma 33, let \( f_1 = 1 \) and \( f_2 = 0 \). Then \( A' \) is \( A \) when \( f_1 = 1 \) and \( f_2 = 0 \). By Lemma 33, it can be easily shown that \( U' \), where \( U' \) is \( U \) when \( f_1 = 1 \) and \( f_2 = 0 \), is unimodular. Consequently, \( f(x) = \sum_{l=1}^{K} f_l x^l \pmod{N} \equiv 0 \pmod{N} \) is a non-trivial null polynomial with degree \( K \) if and only if \( \gcd(K!, N) \neq 1 \). If \( f(x) \) is a non-trivial null polynomials with degree \( K \), the number of non-trivial null polynomial with degree \( K \) is \( \gcd(K!, N) - 1 \), since there is one trivial null polynomial. Since there exist \( \gcd(K!, N) - 1 \) non-trivial null polynomials with degree \( K \), the number of null polynomials with degree up to \( K \) is \( \prod_{l=1}^{K} \gcd(l!, N) \).

We now show the closed form of null polynomials with degree up to \( K \). Consider \( \frac{N}{\gcd(l!, N)} \cdot n \cdot \prod_{k=0}^{l-1} (x + k) \). Since \( l! \mid \prod_{k=0}^{l-1} (x + k) \ \forall x \), \( \frac{N}{\gcd(l!, N)} \cdot n \cdot \prod_{k=0}^{l-1} (x + k) \equiv 0 \pmod{N} \), where \( n = 0, 1, 2, ..., \gcd(l!, N) - 1 \). Note that \( \frac{N}{\gcd(l!, N)} \cdot n \cdot \prod_{k=0}^{l-1} (x + k) \) are equivalent but not congruent. Since the number of \( \frac{N}{\gcd(l!, N)} \cdot n \cdot \prod_{k=0}^{l-1} (x + k) \) is \( \gcd(l!, N) \), null polynomials degree up to \( K \) are \( \sum_{l=1}^{K} \left\{ \frac{N}{\gcd(l!, N)} \cdot n \cdot \prod_{k=0}^{l-1} (x + k) \right\} \), where \( n = 0, 1, 2, ..., \gcd(l!, N) - 1 \).
APPENDIX C

PROOF OF LEMMASES IN CHAPTER 4

C.1 Proof of Lemma 50

1. $n_{f,2} < n_{N,2} - 1$

Let $f_{2,\text{min}}$ be the smallest one in each equivalence class. Let also $\overline{f}_2$ be an element in that equivalence class. Then $\overline{f}_2 = (2k + 1)f_{2,\text{min}}$, where $k$ is an integer. Consequently,

$$-t \cdot (m^2 tf_2 + mf_1 + n) \equiv 0 \pmod{\gcd(t \cdot 2mf_2, N)}$$

$$\iff -t \cdot \{m^2 t \cdot (2k + 1)f_{2,\text{min}} + mf_1 + n\} \equiv 0 \pmod{\gcd(t \cdot 2mf_{2,\text{min}}, N)}$$

$$\iff -t \cdot \{2k \cdot m^2 tf_{2,\text{min}} + m^2 t \cdot f_{2,\text{min}} + mf_1 + n\} \equiv 0 \pmod{\gcd(t \cdot 2mf_{2,\text{min}}, N)}$$

$$\iff -t \cdot (m^2 t \cdot f_{2,\text{min}} + mf_1 + n) \equiv 0 \pmod{\gcd(t \cdot 2mf_{2,\text{min}}, N)},$$

since $\gcd(t \cdot 2mf_{2,\text{min}}, N)\{-t \cdot 2k \cdot m^2 tf_{2,\text{min}}\}$.  

101
Let \( f_{2,\text{min}} \) be the smallest one in each equivalence class. Then, \( n_{f,2,\text{min}} = n_{N,2} - 1 \), where \( n_{f,2,\text{min}} \) is the exponent of 2 in \( f_{2,\text{min}} \). Let also \( f_2 \) be an element in that equivalence class. Then, either \( f_2 = (2k + 1)f_{2,\text{min}} \) or \( f_2 = 2k \cdot f_{2,\text{min}} \) holds. However, when \( f_2 = 2k \cdot f_{2,\text{min}} \),

\[
\frac{f_2 + \frac{N}{2}}{2} = f_{2,\text{min}} \cdot \text{(an even number + an odd number)}
\]

\[= \text{an odd number} \cdot f_{2,\text{min}}
\]

Since the analysis using \( f_2 \) or \( f_2 + \frac{N}{2} \) should be the same, we use \( f_2 + \frac{N}{2} \) when \( f_2 = 2k \cdot f_{2,\text{min}} \). By using similar method as above, it is shown that one representative in each equivalence class is sufficient to analyze equation 4.3.

C.2 Proof of Lemma 51

We prove only for the case \( t = 7 \). For the case \( t = 15 \), the proof is similar.

1. \( n_{N,7} = 0 \) or \( n_{N,7} > 0, n_{f,7} < n_{N,7} \)

Let \( f_1 = f_1 + k \cdot 2f_{2,\text{min}} \), where \( 1 \leq f_1 < 2f_{2,\text{min}} \) and \( f_{2,\text{min}} \) is the smallest element in each equivalence classes as in Lemma 50. Then, from equation (4.3)

\[-t \cdot (m^2tf_2 + mf_1 + n) \equiv 0 \pmod{\gcd(t \cdot 2mf_2, N)}
\]

\[\iff -t \cdot \{m^2tf_2 + m(f_1 + k \cdot 2f_{2,\text{min}}) + n\} \equiv 0 \pmod{\gcd(t \cdot 2mf_{2,\text{min}}, N)}
\]

\[\iff -t \cdot (m^2tf_2 + mf_1 + n) \equiv 0 \pmod{\gcd(t \cdot 2mf_{2,\text{min}}, N)}
\]

, since \( \gcd(t \cdot 2mf_{2,\text{min}}, N) \mid \{t \cdot m \cdot k \cdot 2f_{2,\text{min}}\} \). Note that since \( \gcd(2f_2, N) = \gcd(2f_{2,\text{min}}, N) \), \( \gcd(t \cdot 2mf_2, N) = \gcd(t \cdot 2mf_{2,\text{min}}, N) \)
2. \( n_{N,7} \leq n_{f,7} \)

Let \( f_1 = \overline{f}_1 + k \cdot \frac{2f_{2,\text{min}}}{7} \), where \( 1 \leq \overline{f}_1 < \frac{2f_{2,\text{min}}}{7} \). Note that \( 7^{n_{N,7}} \mid \gcd (t \cdot 2mf_2, N) \) and \( 7^{n_{N,7}+1} \nmid \gcd (t \cdot 2mf_2, N) \). Consequently,

\[
-t \cdot (m^2tf_2 + mf_1 + n) \equiv 0 \pmod{\gcd (t \cdot 2mf_2, N)}
\]

\[
\iff -t \cdot \left\{ m^2tf_2 + m \left( \overline{f}_1 + k \cdot \frac{2f_{2,\text{min}}}{7} \right) + n \right\} \equiv 0 \pmod{\gcd (2mf_{2,\text{min}}, N)}
\]

\[
\iff -t \cdot (m^2tf_2 + m\overline{f}_1 + n) - k \cdot m \cdot 2f_{2,\text{min}} \equiv 0 \pmod{\gcd (2mf_{2,\text{min}}, N)}
\]

\[
\iff -t \cdot (m^2tf_2 + m\overline{f}_1 + n) \equiv 0 \pmod{\gcd (2mf_{2,\text{min}}, N)}
\]

C.3 Proof of Lemma 55

We only prove for \( n_{N,7} = 0 \) or \( n_{N,7} > 0 \), \( n_{f,7} < n_{N,7} \) and \( t = 7 \) case. Let \( m \) be an odd number and suppose we have \((m, n)\) for \( mf_1 + n \equiv f_{2,\text{min}} \pmod{2f_{2,\text{min}}} \). Let \( \overline{f}_1 = 2f_{2,\text{min}} - f_1 \). Then

\[
m\overline{f}_1 + n \equiv f_{2,\text{min}} \pmod{2f_{2,\text{min}}}
\]

\[
\iff m(2f_{2,\text{min}} - f_1) + n \equiv f_{2,\text{min}} \pmod{2f_{2,\text{min}}}
\]

\[
\iff m(-f_1) + n \equiv f_{2,\text{min}} \pmod{2f_{2,\text{min}}}
\]

\[
\iff mf_1 - n \equiv f_{2,\text{min}} \pmod{2f_{2,\text{min}}}
\]

Consequently, \((m, -n)\) is a solution for \( \overline{f}_1 \). When \( m \) is even,

\[
mf_1 + n \equiv 0 \pmod{\gcd (2mf_{2,\text{min}}, N)}
\]

\[
\iff m(2f_{2,\text{min}} - f_1) + n \equiv 0 \pmod{\gcd (2mf_{2,\text{min}}, N)}
\]

\[
\iff mf_1 - n \equiv 0 \pmod{\gcd (2mf_{2,\text{min}}, N)}
\]

Thus, \((m, -n)\) is a solution for \( \overline{f}_1 \).
C.4 Proof of Lemma 54

We prove only one case. The other cases can be done similarly. Let \( n_{N,7} > 0 \) and \( n_{f,7} = n_{N,7} \). Let \( m \) be odd. Then \( \gcd(t \cdot 2mf_{2,\min}, N) = 2f_{2,\min} = t \cdot \frac{2f_{2,\min}}{7} \)

\[
-t \cdot (m^2t f_{2,\min} + mf_1 + n) \equiv 0 \pmod{\gcd(t \cdot 2mf_{2,\min}, N)}
\]

\[
\iff -t \cdot (m^2t f_{2,\min} + mf_1 + n) \equiv 0 \pmod{\frac{2f_{2,\min}}{7}}
\]

\[
\iff -(m^2t f_{2,\min} + mf_1 + n) \equiv 0 \pmod{\frac{2f_{2,\min}}{7}}.
\]

Since

\[
m^2t f_{2,\min} \equiv m^2t \frac{2f_{2,\min}}{7} \pmod{\frac{2f_{2,\min}}{7}}
\]

\[
\equiv \text{an odd number} \cdot \frac{f_{2,\min}}{7} \pmod{\frac{2f_{2,\min}}{7}}
\]

\[
\equiv \frac{f_{2,\min}}{7} \pmod{\frac{2f_{2,\min}}{7}}.
\]

Consequently,

\[
-(m^2t f_{2,\min} + mf_1 + n) \equiv 0 \pmod{\frac{2f_{2,\min}}{7}}
\]

\[
\iff m^2t f_{2,\min} + mf_1 + n \equiv 0 \pmod{\frac{2f_{2,\min}}{7}}
\]

\[
\iff mf_1 + n \equiv -\frac{f_{2,\min}}{7} \pmod{\frac{2f_{2,\min}}{7}}
\]

\[
\iff mf_1 + n \equiv \frac{f_{2,\min}}{7} \pmod{\frac{2f_{2,\min}}{7}}.
\]

When \( m \) is even,

\[
-t \cdot (m^2t f_{2,\min} + mf_1 + n) \equiv 0 \pmod{\gcd(t \cdot 2mf_{2,\min}, N)}
\]

\[
\iff -t \cdot (m^2t f_{2,\min} + mf_1 + n) \equiv 0 \pmod{t \cdot \gcd(2mf_{2,\min}, N)}.
\]

104
\( \iff - (m^2 t_{f_{2,\text{min}}} + m f_1 + n) \equiv 0 \pmod{\gcd(2m f_{2,\text{min}}, N)} \)

\( \iff - (\text{an even number} \cdot m t_{f_{2,\text{min}}} + m f_1 + n) \equiv 0 \pmod{\gcd(2m f_{2,\text{min}}, N)} \)

\( \iff - (m f_1 + n) \equiv 0 \pmod{\gcd(2m f_{2,\text{min}}, N)} \)

\( \iff m f_1 + n \equiv 0 \pmod{\gcd(2m f_{2,\text{min}}, N)} \)

C.5 Proof of Lemma 56

The proof is given for \( t = 7 \) and \( n_{N,7} = 0 \) or \( n_{N,7} > 0, n_{f,7} < n_{N,7} \) case. The others can be similarly shown.

1. \( n_{N,2} = 3 \)

We divide into three cases.

\[
\begin{align*}
\begin{cases}
\frac{f_{2,\text{min}}}{2} & \leq f_1 \leq f_{2,\text{min}} - 1 \\
\frac{f_{2,\text{min}}}{4} + 1 & \leq f_1 \leq \frac{f_{2,\text{min}}}{2} - 1 \\
1 & \leq f_1 \leq \frac{f_{2,\text{min}}}{4} - 1
\end{cases}
\end{align*}
\]

(a) \( \frac{f_{2,\text{min}}}{2} \leq f_1 \leq f_{2,\text{min}} - 1 \) Let \( (m, n) = (1, f_{2,\text{min}} - f_1) \). Then \( (m, n) \) satisfies 

\[ m f_1 + n \equiv f_{2,\text{min}} \pmod{2 f_{2,\text{min}}} \]

and \( 1 \leq n = f_{2,\text{min}} - f_1 \leq \frac{f_{2,\text{min}}}{2} - 1 \). Thus 

\[ |m| + |n| \leq \frac{f_{2,\text{min}}}{2}. \]

(b) \( \frac{f_{2,\text{min}}}{4} + 1 \leq f_1 \leq \frac{f_{2,\text{min}}}{2} - 1 \) Let \( (m, n) = (3, f_{2,\text{min}} - 3f_1) \). Then \( (m, n) \) satisfies 

\[ m f_1 + n \equiv f_{2,\text{min}} \pmod{2 f_{2,\text{min}}} \]

and \( -\frac{f_{2,\text{min}}}{2} + 3 \leq n = f_{2,\text{min}} - 3f_1 \leq \frac{f_{2,\text{min}}}{4} - 3 \). Thus 

\[ |m| + |n| \leq \max \left( \left| \frac{f_{2,\text{min}}}{2} - 3 \right| + 3, \left| \frac{f_{2,\text{min}}}{4} - 3 \right| + 3 \right) \leq \frac{f_{2,\text{min}}}{2}. \]

(c) \( 1 \leq f_1 \leq \frac{f_{2,\text{min}}}{4} - 1 \) Let \( (m, n) = (2, -2f_1) \). Then \( (m, n) \) satisfies 

\[ m f_1 + n \equiv 0 \pmod{\gcd(2m f_{2,\text{min}}, N)} \]

and \( -\frac{f_{2,\text{min}}}{2} + 2 \leq n = -2f_1 \leq -2 \). Thus 

\[ |m| + |n| \leq \max \left( \left| \frac{f_{2,\text{min}}}{2} - 2 \right| + 2, 2 + 2 \right) \leq \frac{f_{2,\text{min}}}{2}. \]

From above, for all cases, \( |m| + |n| \leq \frac{f_{2,\text{min}}}{2} \). Consequently, the upper bound of the distance is \( 4 \cdot \frac{f_{2,\text{min}}}{2} + 6 = 2 f_{2,\text{min}} + 6 \).
2. \( n_{N,2} = 2 \)

We divide into three cases.

\[
\begin{cases}
\frac{f_{2,\text{min}}}{2} + 1 \leq f_1 \leq f_{2,\text{min}} - 1 \\
\frac{f_{2,\text{min}}}{4} \leq f_1 \leq f_{2,\text{min}} - 1 \\
1 \leq f_1 \leq \frac{f_{2,\text{min}}}{4} - 2
\end{cases}
\]

For \( \frac{f_{2,\text{min}}}{4} \leq f_1 \leq \frac{f_{2,\text{min}}}{2} - 1 \) let \((m, n) = (3, f_{2,\text{min}} - 3f_1)\).

Then \((m, n)\) satisfies \( mf_1 + n \equiv 0 \pmod {\gcd (2mf_{2,\text{min}}, N)} \) and \(-\frac{f_{2,\text{min}}}{2} + 3 \leq n = f_{2,\text{min}} - 3f_1 \leq f_{2,\text{min}} - \frac{f_{2,\text{min}}}{4} \). Thus,

\[|m| + |n| \leq \max \left( \left| \frac{f_{2,\text{min}}}{2} - 3 \right| + 3, \frac{f_{2,\text{min}}}{4} + 3 \right). \quad (C.1)\]

\( \max \left( \left| \frac{f_{2,\text{min}}}{2} - 3 \right| + 3, \frac{f_{2,\text{min}}}{4} + 3 \right) \leq \frac{f_{2,\text{min}}}{2} \) holds when \( 12 \leq f_{2,\text{min}} \). Consequently for \( f_{2,\text{min}} = 4 \), we need to compute it directly using \( |m| + |n| \leq 4 \).

3. \( n_{N,2} = 1 \)

For \( f_{2,\text{min}} = 8k + 2 \), we divide into three cases.

\[
\begin{cases}
\frac{f_{2,\text{min}}}{2} \leq f_1 \leq f_{2,\text{min}} - 1 \\
\frac{f_{2,\text{min}}}{4} + 1 \leq f_1 \leq \frac{f_{2,\text{min}}}{2} - 2 \\
1 \leq f_1 \leq \frac{f_{2,\text{min}}}{4} - 2
\end{cases}
\]

and use similar method as above.

For \( f_{2,\text{min}} = 8k + 6 \), we divide into three cases.

\[
\begin{cases}
\frac{f_{2,\text{min}}}{2} \leq f_1 \leq f_{2,\text{min}} - 1 \\
\frac{f_{2,\text{min}}}{4} + 1 \leq f_1 \leq \frac{f_{2,\text{min}}}{2} - 2 \\
1 \leq f_1 \leq \frac{f_{2,\text{min}}}{4} - 1
\end{cases}
\]

and use similar method as above.

Suppose that \( \overline{f}_2 < f_{2,\text{min}} \), where \( \overline{f}_2 \) and \( f_{2,\text{min}} \) are not in the same equivalence class.

In this case, we divide it into two cases. For example, let \( n_{N,2} = 3 \).

\[
\begin{cases}
\frac{f_{2,\text{min}}}{4} + 1 \leq f_1 \leq f_{2,\text{min}} - 1 \\
1 \leq f_1 \leq \frac{f_{2,\text{min}}}{4} - 1
\end{cases}
\]
For \( \frac{f_{2,\min}}{4} + 1 \leq f_1 \leq f_{2,\min} - 1 \), we use \((m, n) = (1, \overline{f}_2 - f_1)\) and for \( 1 \leq f_1 \leq \frac{f_{2,\min}}{4} - 1 \), we use \((m, n) = (2, -2f_1)\). Then it can be shown that \(|m| + |n| \leq \frac{3f_2}{4}\). Since \( \overline{f}_2 \leq \frac{f_{2,\min}}{2} \), the weight \( \overline{f}_2 \) generates is equal to or less than the weight \( f_{2,\min} \) generates.

### C.6 Proof of Lemma 65

Let \( m_1, m_2, m_3 \) and \( n_1, n_2, n_3 \) be non-zero integers. In Figure 4.9, input weight 6 error events occur if and only if

\[
|f(x_2) - f(x_1)| \equiv n_1 t \pmod{N} \\
|f(x_3) - f(x_1 + m_1 t)| \equiv n_2 t \pmod{N} \\
|f(x_3 + m_3 t) - f(x_2 + m_2 t)| \equiv n_3 t \pmod{N}
\]

which is equivalent to

\[
f(i_2) + Lf'(i_2)y_2 - f(i_1) - Lf'(i_1)y_1 \equiv n_1 t \quad (C.2) \\
f(i_3) + Lf'(i_3)y_3 - f(i_1 + m_1 t) - Lf'(i_1 + m_1 t)y_1 \equiv n_2 t \quad (C.3) \\
f(i_3 + m_3 t) + Lf'(i_3 + m_3 t)y_3 - f(i_2 + m_2 t) - Lf'(i_2 + m_2 t)y_2 \equiv n_3 t \quad (C.4)
\]

Let \( x_i = Ly_i + i_i \), where \( 1 \leq y_i \leq \frac{N}{L} \), \( L \) is the number of sub-interleavers and \( N \) is interleaver length.

\[
f(x_1) = f_1 x_1 + f_2 x_1^2 = f_1 (Ly_i + i_i) + f_2 (Ly_i + i_i)^2 = f(i_i) + L \cdot y_i \cdot f'(i_i)
\]

Similarly, \( f(x_1 + m_1 t) = f(x_1 + m_1 t) + L \cdot y_i \cdot f'(x_1 + m_1 t) \). Using this notation, equations C.2, C.3 and C.4 can be represented by matrix form,

\[
Ay \equiv b \pmod{N},
\]

107
where
\[
A = \begin{bmatrix}
-L \cdot f(i_1) & L \cdot f(i_2) & 0 \\
0 & -L \cdot f'(i_2 + m_2t) & L \cdot f'(i_3 + m_3t) \\
-L \cdot f'(i_1 + m_1t) & 0 & L \cdot f'(i_3)
\end{bmatrix},
\]

\[
b = [f(i_1) - f(i_2) + n_1t, f(i_2 + m_2t) - f(i_3 + m_3t) + n_3t, f(i_1 + m_1t) - f(i_3) + n_2t]^T.
\]

By Theorem 30, \(A\) can be represented as \(A = UEV\). It can be easily verified that \(U, E\) and \(V\) satisfies \(A = UEV\).
BIBLIOGRAPHY


