Decoding Algorithms of Error-Correcting Codes over Symbol-Pair Read Channels

Takita, Makoto

(Degree)
博士（工学）
(Date of Degree)
2018-03-25
(Date of Publication)
2019-03-01
(Resource Type)
doctoral thesis
(Report Number)
甲第7183号
(URL)
https://hdl.handle.net/20.500.14094/D1007183

博士論文

Decoding Algorithms of Error-Correcting Codes over Symbol-Pair Read Channels
(シンボルペア通信路における誤り訂正符号の復号法)

2018年1月

神戸大学大学院工学研究科
TAKITA Makoto
瀧田　慎

Acknowledgments

I would like to thank everyone who contributed to this thesis. First of all, I would like to express my sincere gratitude to my superior Prof. Masakatu Morii for introducing me to coding theory as my research topic. I also would like to express my sincere gratitude to Associate Prof. Masanori Hirotomo of Saga University for kind research guidance from the beginning of the research to the present. Their guidance throughout my research helps me to establish the basis of my career in this field. I also would like to thank the judging committee members, Prof. Masatoshi Kitamura, Prof. Seiichi Ozawa, and Associate Prof. Yoshiaki Shiraishi for reviewing this manuscript and giving me valuable feedbacks. Moreover, I would like to thank Associate Prof. Yoshiaki Shiraishi and Assistant Prof. Yasuhiro Takano for giving me helpful advice and supporting my research in the laboratory. I have learned from them about various techniques for guidance for my junior through the seminar and the teaching assistant of experiments.

I would like to thank Associate Prof. Minoru Kuribayashi for giving me helpful advice and supporting my research when I was a bachelor and a master student. He taught me various things about research life and recommended me to go on to a doctoral course. I would not have gone to the doctoral course without his advice. Moreover, I would like to thank my classmates of the laboratory: Naoki Aoyama, Takahiro Iriyama, Yoshihiro Kusanagi, Sho Sakikoyama, and Chikara Sasakura. They encouraged me through research discussion and daily talks. I also would like to thank Dr. Yuhei Watanabe and Keisuke Furumoto for many advice as a senior of doctoral course. I am grateful to all members of the laboratory in Kobe University.

I would like to thank Associate Prof. Manabu Kobayashi, Associate Prof. Brian Kurkoski, Assistant Prof. Kenji Yasunaga, Associate Prof. Manabu Hagiwara, and anonymous reviewers for reviewing published papers and giving me valuable feedback. I wish to thank all people whom have done constructive discussions and talks in various symposiums and technical committees.

I greatly appreciate my parents, grandparents, sisters for their longtime support. I would never be possible to write this thesis without them.

Abstract

In the information society, it is important to make sure the reliability of information transmitted over the networks, also it is important to keep accurate information stored in recording media. The transmitted data and recorded data may be corrupted due to various factors. The error-correcting codes are used for correctness of the data, so coding theory is one of academical research topics.

There are many classes of error-correcting codes. Depending on the ability of the code, the code appropriate for the application is selected. The main ability of the code is equal to the number of correctable errors. Even if the ability of the code is high, depending on the decoding algorithm, the ability can not be used to the utmost. Thus, the performance of the algorithm used for encoding and decoding is also important.

In this thesis, I study the new error-correcting codes called *symbol-pair codes*, which are proposed by Cassuto and Blaum. They presented a coding framework for channels whose outputs are overlapping pairs of symbols in storage applications. Such channels are called symbol-pair read channels. The pair distance and pair error are used in symbol-pair read channels. The study on the decoding algorithm to correct pair errors because the symbol-pair read channel a new framework proposed in 2010.

I focus on the decoding method of symbol-pair codes. Cassuto et al. and Yaakobi et al. discuss the decoding method of the codes. However, their decoding methods can not be user to the utmost of the capability of the codes.

In this thesis, I propose multiple new decoding methods. By newly defining the syndromes of the symbol-pair code, I propose the first decoding method that can correct all pair errors within the capability of the code. Based on the research results, I discuss the reduction of the calculation complexity and the design of the decoding circuit in hardware.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Motivation

In the information society, it is important to ensure the reliability of information transmitted over the networks such as the Internet, Local Area Network (LAN). They use optical communications, satellite communications, Wireless Fidelity (Wi-Fi) and so on. Moreover, it is also important to keep accurate information stored in recording media such as hard disk drives (HDD), solid state drives (SSD), and compact discs (CD).

In data communications over the networks and the data readout from recording media, data is corrupted due to the radio interference, the equipment failure and so on. The error-correcting codes are used to restore the corrupted data to the appropriate data. By using the error-correcting codes, redundant information is added to transmit data. Redundant information can detect or correct errors occurred in the received data. Thus, the reliability of information is improved by using the error-correcting codes.

The error-correcting codes have a lot of classes such as the Hamming codes, the BCH codes, and the Reed-Solomon codes [1, 2]. Recently, for treating the large data, the low-density parity-check (LDPC) codes and Spatially Coupled LDCP codes are studied. These codes are selected according to the communication channel to be used. To select codes, the length of the code, the number of errors which the code can correct, and algorithms for encoding and decoding are important. In general, the number of correctable errors of the code is determined by the minimum distance among sequences, which are encoded by the codes. The code has the capability to correct errors up to the half of the minimum distance. Whether errors within the correcting capability actually can be corrected depends on the algorithm for decoding. Thus, the decoding algorithm which can correct all errors within the capability is important.

In this thesis, I study the new error-correcting codes called *symbol-pair codes*. The symbol-pair codes are used over *symbol-pair read channels* which are new channels presented by Yuval Cassuto and Mario Blaum [3, 4].

Recently, the capacity of magnetic recording devices is increasing due to high density data recording. Thus, the distance between elements for recording data is small Moreover, with the development of reading technology, two adjacent data will be read at the same time. Cassuto et al. presented a coding framework for channels whose outputs are overlapping pairs of symbols in storage applications [3, 4]. They defined errors and a metric over the symbol pair read channel as *pair errors* and pair distance, respectively. They also defined the parameter, which is determined by the capability for correcting pair error, as minimum pair distance. Moreover, Cassuto and Lisyn constructed cyclic symbol-pair codes using algebraic methods [5]. They showed that there are some symbol-pair codes whose rates are higher than the codes for the Hamming metric with the same relative distance. Yaakobi, Bruck, and Siegel proved a lower bound on the minimum pair distance of cyclic codes and presented a decoding algorithm for cyclic codes over symbol-pair read channels [6, 7]. Yaakobi's decoding algorithm uses decoders for cyclic codes in the Hamming metric.

In this thesis, I focus on the decoding method of symbol-pair codes. Cassuto et al. proposed a decoding algorithm by reducing the pair-decoding problem to error-and-erasure decoding in the Hamming metric[3, 4]. They considered erasure which indicates the pairs of received pair vectors in conflict. In Ref.[6, 7], Yaakobi et al. proved that, for cyclic codes with the minimum Hamming distance d_H, the minimum pair distance holds:

$$d_p \geq d_H + \left\lceil \frac{d_H}{2} \right\rceil .$$

Furthermore, they presented a decoding algorithm for correcting pair errors using a decoder for cyclic codes with the error-correcting capability t_H. Their algorithm can correct up to t_0-pair errors, where

$$t_0 = \left\lfloor \frac{3t_H + 1}{2} \right\rfloor .$$

Both of decoding algorithms presented by Cassuto et al. and Yaakobi et al. cannot correct all pair errors within the pair error correcting capability.

1.2 This Study

In this thesis, I propose a new syndrome decoding algorithm which can correct all pair errors within the pair error-correcting capability. Moreover, I also propose decoding algorithms based on the research result obtained by studying syndrome decoding algorithm.

1.2.1 Syndrome Decoding of Linear Codes over Symbol-Pair Read Channels

In Chapter 3, I propose the new decoding algorithm of linear codes over symbol-pair read channels. I newly define a parity-check matrix and two types of syndromes of symbol-pair codes, which are called *symbol-pair syndrome* and *neighbor-symbol syndrome*. Further, I prove that the pair of two syndromes is unique for each error vector whose number of pair errors is not more than the half of the minimum pair distance.

1.2.2 A Decoding Algorithm of Cyclic Codes over Symbol-Pair Read Channels

The proposed syndrome decoding algorithm can correct all pair error vectors whose pair errors are within half of the minimum pair distance. However, this algorithm requires a decoding table for matching the syndromes with the pair error vectors, and it has a drawback in that the space complexity is high for long codes.

In the Chapter 4, I propose an efficient decoding algorithm for cyclic codes over symbol-pair read channels using a decoder for cyclic codes. The proposed algorithm corrects pair errors based on the relationship between the pair errors and the syndromes.

Whereas Yaakobi's algorithm corrects pair errors in the Hamming metric, the proposed algorithm corrects pair errors in the pair metric. Thus, the proposed algorithm can correct pair errors that cannot be corrected by Yaakobi's algorithm.

1.2.3 Algebraic Decoding of BCH Codes over Symbol-Pair Read Channels

In chapter 5, I focus on the decoding problem for BCH codes over symbol-pair read channels.

In the decoding problem in the Hamming metric, the error-locator polynomial is obtained by algebraic methods, such as the Euclidean algorithm [10] or Berlekamp-Massey algorithm [11]. Moreover, there are some methods that directly derived the error-locator polynomial or error positions when several errors are corrected [2, 12].

I consider correcting pair errors with algebraic methods in the pair metric. I define *error-locator polynomial* and *conflict-locator polynomial* of the BCH codes over symbol-pair read channels. Moreover, I derive a relation between the conflict-locator polynomial and the error-locator polynomial. In addition, I propose new decoding algorithms that calculate the error-locator polynomial with an algebraic method by using syndrome. Furthermore, I compare the proposed decoding algorithms with existing decoding algorithms and discuss the validity of calculating the error-locator polynomial of pair errors.

1.2.4 Error-Trapping Decoding over Symbol-Pair Read Channels

Above decoding algorithms assume the decoding by software. Since the symbol-pair read channels are the model for the storage applications, it is natural to embed the decoder in hardware. A small decoder is required for embedding in the hardware.

In chapter 6, I discuss error-trapping decoding for cyclic codes over symbol-pair read channels. I propose a new error-trapping decoding algorithm under some restrictions on the pair error patterns that we intend to correct. It corrects all pair error patterns whose pair errors within the pair error-correcting capability under the restrictions. I firstly discuss problems in the existing error-trapping decoding algorithms when it is used for cyclic codes over symbol-pair read channels. I solve these problems by using the neighbor-symbol syndrome and propose a new error-trapping decoding algorithm. Next, I show a circuitry that implements the proposed algorithm. Finally, I discuss modifying the restrictions on the correctable error patterns. I show necessity that I need to find covering polynomials suitable for the symbol-pair read channels, and show how to modify the restrictions by using the covering polynomials.

1.3 Outline

The remainder of the thesis is organized as follows. Chapter 2 explains describe error-correcting codes and describe definitions and terms related to symbol-pair read channels. Chapter 3 discusses a decoding method of linear codes over symbol-pair read channels. Chapter 4 discusses a decoding method of cyclic codes over symbol-pair read channels. Chapter 5 discusses a decoding method of BCH codes over symbol-pair read channels. Chapter 6 discusses a decoding method of cyclic codes by hardware over symbol-pair read channels. Finally, Section 7 concludes this thesis.

Chapter 2

Error-Correcting Codes and Symbol-Pair Read Channels

2.1 Error-Correcting Codes

The transmission and storage of digital information process transfer data from an information source to a destination. A simple model of a typical transmission or storage system may be represented in Fig. 2.1. The *digital source* is a sequence of binary digits (bits) called into the *information sequence* $\boldsymbol{i}$. The *channel encoder* transform the information sequence $\boldsymbol{i}$ into a discrete *encoded sequence* $\boldsymbol{w}$ called a *codeword.* The codeword enters *channel* or *recoding media* and is corrupted by noise. The channel outputs corresponding to the encoded sequence $\boldsymbol{w}$ is called the *received sequence* $\boldsymbol{u}$. The *channel decoder* transforms the received sequence $\boldsymbol{u}$ into a binary sequence $\hat{\boldsymbol{i}}$ called the *estimated information sequence.*

The encoder for a block code divides the information sequence into message block of k information bits each. A message block is represented by the binary k-tuple $\boldsymbol{i} = (i_0, i_1, \ldots, i_{k-1})$, called a message. The encoder transforms each message $\boldsymbol{i}$ independently into an n-tuple $\boldsymbol{w} = (w_0, w_1, \ldots, w_{n-1})$ of discrete symbol, called *codeword.* Therefore, corresponding to the 2^k different possible messages. There are 2^k different possible codewords at the encoder output. This set of 2^k codewords of length n is called an (n, k) block code.

The *minimum distance* is an important parameter of block codes. The minimum

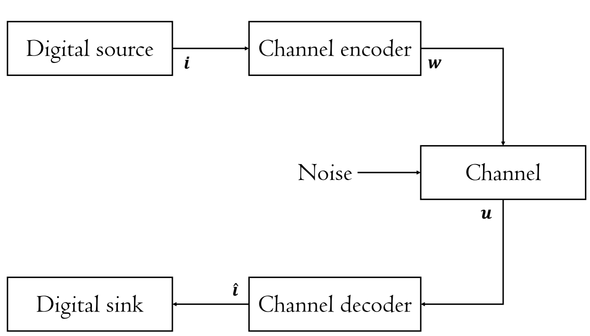

Figure 2.1: Channel coding system

distance determines the error detecting and error-correcting capabilities. Let $\boldsymbol{x} = (x_0, x_1, \ldots, x_{n-1})$ be a binary n-tuple. The *Hamming weight* of $\boldsymbol{x}$, denoted by $W_H(\boldsymbol{x})$, is defined as the number of nonzero components of $\boldsymbol{x}$. Let $\boldsymbol{x}$ and $\boldsymbol{y}$ be two n-tuple. The *Hamming distance* between $\boldsymbol{x}$ and $\boldsymbol{y}$, denoted by $D_H(\boldsymbol{x}, \boldsymbol{y})$, is defined as the number of places where they differ, that is,

$$D_H(\boldsymbol{x}, \boldsymbol{y}) = \sum_{i=0}^{n-1} D_H(x_i, y_i), \tag{2.1}$$

$$D_H(x_i, y_i) = \begin{cases} 0 & \text{if } x_i = y_i \\ 1 & \text{otherwise} \end{cases}. \tag{2.2}$$

The Hamming weight is the Hamming distance between $\boldsymbol{x}$ and all-zero tuple $\mathbf{0}$, that is,

$$W_H(\boldsymbol{x}) = D_H(\boldsymbol{x}, \mathbf{0}). \tag{2.3}$$

The Hamming distance is a metric function that satisfies the triangle inequality. Let $\boldsymbol{x}$, $\boldsymbol{y}$, and $\boldsymbol{z}$ be three n-tuples. Then,

$$D_H(\boldsymbol{x}, \boldsymbol{y}) + D_H(\boldsymbol{y}, \boldsymbol{z}) \geq D_H(\boldsymbol{x}, \boldsymbol{z}). \tag{2.4}$$

In follows from the definition of the Hamming distance and the definition of modulo-2 addition that the Hamming distance between two n-tuple $\boldsymbol{x}$ and $\boldsymbol{y}$ is equal to the Hamming weight of the sum of $\boldsymbol{x}$ and $\boldsymbol{y}$, that is,

$$D_H(\boldsymbol{x}, \boldsymbol{y}) = W_H(\boldsymbol{x} + \boldsymbol{y}). \tag{2.5}$$

Given a block code C, the *minimum distance* of C, denoted by d_H, is can compute the Hamming distance between any two distinct codewords, that is,

$$d_H \triangleq \min_{\boldsymbol{x}, \boldsymbol{y} \in C, \boldsymbol{x} \neq \boldsymbol{y}} D_H(\boldsymbol{x}, \boldsymbol{y}). \tag{2.6}$$

A block code with minimum Hamming distance d_H guarantees correction of all the error pattern of

$$t_H = \left\lfloor \frac{d_H - 1}{2} \right\rfloor \tag{2.7}$$

or fewer errors, where $\lfloor (d_H - 1)/2 \rfloor$ denotes the largest integer no greater than $(d_H - 1)/2$. The parameter t_H is called the error-correcting capability.

For a block code with 2^k codewords and length n, the code is called a *linear* (n, k) block code if and only if its 2^k codewords form a k-dimensional subspace of the vector space of all the n-tuples over the field GF(2)[2]. Thus, for any two codeword $\boldsymbol{w}_1$ and $\boldsymbol{w}_2$ in a code C, and any two element c_1 and c_2 in GF(2), C is a linear code if the following equation is satisfied.

$$c_1 \boldsymbol{w}_1 + c_2 \boldsymbol{w}_2 \in C. \tag{2.8}$$

If C is a linear block code, the sum of two codewords is also a codeword. From (2.5), the Hamming distance between two codewords in C is equal to the Hamming weight of the third codeword in C. Then, the minimum Hamming distance is obtained by the *minimum Hamming weight*, defined by

$$w_H \triangleq \min_{\boldsymbol{x} \in C, \boldsymbol{x} \neq \mathbf{0}} W_H(\boldsymbol{x}). \tag{2.9}$$

A linear code with 2^k codewords, length n, and the minimum distance d is called a (n, k, d) linear code.

Any set of basis vectors for a linear block code C can be considered as rows of a matrix $\mathbf{G}$, called a *generator matrix* of C. The row space of $\mathbf{G}$ is the linear code C, and the vector is a codeword if and only if it is a linear combination of rows of G. For a code C with generator matrix $\mathbf{G}$, the codeword $\boldsymbol{w}$ of a digital source i is given by

$$\boldsymbol{w} = \boldsymbol{i}\mathbf{G}. \tag{2.10}$$

There is another useful matrix associated with every linear block code. For any $k \times n$ generator matrix $\mathbf{G}$ with linearly independent rows, there exists an $(n-k) \times n$ matrix $\mathbf{H}$ with linearly independent rows such that any vector in the row space of $\mathbf{G}$ is orthogonal to the rows of $\mathbf{H}$, and any vector that is orthogonal to the rows of $\mathbf{H}$ is in the row space of $\mathbf{G}$. The matrix $\mathbf{H}$ is called a *parity-check matrix* of the code. Then, the generator matrix $\mathbf{G}$ of a code C and its parity-check matrix $\mathbf{H}$ satisfy the following equation.

$$\mathbf{G}\mathbf{H}^{\mathrm{T}} = \mathbf{0}. \tag{2.11}$$

Thus, the codeword $\boldsymbol{w}$ of the code C satisfy $\boldsymbol{w}\mathbf{H}^{\mathrm{T}} = \mathbf{0}$.

For the any codeword $\boldsymbol{w} = (w_0, w_1, \ldots, w_{n-1})$ of a linear code, the code is *cyclic code* if and only if $(w_{n-1}, w_0, w_1 \ldots, w_{n-2})$ is also codeword.

Let $\boldsymbol{w}$ be a codeword of (n, k) linear code. If $\boldsymbol{w}$ is transmitted over a noisy channel, the received vector $\boldsymbol{u}$ may be suffered errors. Let $\boldsymbol{e}$ be an *error vector*, the received vector is $\boldsymbol{u} = \boldsymbol{w} + \boldsymbol{e}$. For the received vector $\boldsymbol{u}$, a *syndrome* is defined by

$$\boldsymbol{s} = (s_1, s_2, \ldots, s_{n-k}) = \boldsymbol{u}\mathbf{H}^{\mathrm{T}}. \tag{2.12}$$

Since $\boldsymbol{u} = \boldsymbol{w} + \boldsymbol{e}$ and $\boldsymbol{w}$ satisfies $\boldsymbol{w}\mathbf{H}^{\mathrm{T}} = \mathbf{0}$, the syndrome is

$$\boldsymbol{s} = \boldsymbol{u}\mathbf{H}^{\mathrm{T}} = (\boldsymbol{w} + \boldsymbol{e})\mathbf{H}^{\mathrm{T}} = \boldsymbol{w}\mathbf{H}^{\mathrm{T}} + \boldsymbol{e}\mathbf{H}^{\mathrm{T}} = \boldsymbol{e}\mathbf{H}^{\mathrm{T}} \tag{2.13}$$

Thus, the syndrome is not dependent on codewords and depends on error vectors.

For the code C with the parity-check matrix $\mathbf{H}$, if the syndromes of the error vector $\boldsymbol{e}$ in the set of errors E are all different, the errors are corrected by using a correspondence table of the syndrome and the error vector. In general, for the linear code C with the error-correcting capability t_H, the set of errors E which the Hamming weight is t_H or less satisfies the above condition.

The decoding method using syndrome of linear codes is executed in the following steps.

[**Syndrome decoding algorithm of linear codes**]

Step 1. Calculate the syndrome $\boldsymbol{s}$ from received vector $\boldsymbol{u}$.

Step 2. Refer the correspondence table, and estimate corresponding error $\boldsymbol{e}$ by the syndrome $\boldsymbol{s}$.

Step 3. Decode the received vector $\boldsymbol{u}$ into the codeword as $\hat{\boldsymbol{w}} = \boldsymbol{u} - \boldsymbol{e}$

The syndrome decoding uses for any linear codes. However, the size of the correspondence table depends on the code length and the correcting capability, thus a large memory is needed to hold the table for the large codes, so it is not practical. There are codes which have an algebraic structure for efficient encoding and decoding. The typical algebraic code is BCH code, and the Euclidean algorithm and Berlekamp-Massey algorithm are decoding algorithm for the BCH codes.

2.2 Symbol-Pair Read Channels and Its Decoding

2.2.1 Symbol-Pair Read Channels

Cassuto and Blaum proposed new error-correcting codes which are called *symbol-pair codes* [3][4]. High-density data storage technologies require high-capacity storage at a relatively low cost. They presented a coding framework for channels whose outputs are overlapping pairs of symbols in storage applications. Such channels are called *symbol-pair read channels*.

In this section, I briefly review definitions and terms related to symbol-pair codes presented in Ref. [3].

Definition 1. Let $\boldsymbol{x} = [x_0, x_1, \ldots, x_{n-1}]$ be a vector in n-dimensional vector space Σ. The *symbol-pair read vector* of $\boldsymbol{x}$ is defined as

$$\pi(\boldsymbol{x}) = [(x_0, x_1), (x_1, x_2), \ldots, (x_{n-2}, x_{n-1}), (x_{n-1}, x_0)].$$

This work focus in this work on binary vectors, so $\Sigma = \{0, 1\}$.

To distinguish received pair vectors $\overleftrightarrow{\boldsymbol{u}}$ from standard symbol vectors, I will denote them with the symbol $\leftrightarrow$ as follows:

$$\overleftrightarrow{\boldsymbol{u}} = [(u_{l,0}, u_{r,1}), (u_{l,1}, u_{r,2}), \ldots, (u_{l,n-1}, u_{r,0})]. \tag{2.14}$$

In each pair, l and r denote the left and right symbols, respectively. Furthermore, the i-th pair of the pair vector is defined as $\overleftrightarrow{u}_i = (u_{l,i}, u_{r,i+1})$.

Every vector $\boldsymbol{x}$ has a pair representation in n-dimensional pair vector space $(\Sigma, \Sigma)^n$. However, not all pair vectors in $(\Sigma, \Sigma)^n$ have a corresponding vector in Σ^n, because $u_{r,i}$ and $u_{l,i+1}$ in two adjacent pairs may have different readings.

For example, the pair vector $[(0, 0), (1, 0), (0, 1), (1, 1), (1, 0)]$ does not have a corresponding vector in Σ^n since $x_1 = 0$ at the right of the first pair, but $x_1 = 1$ at the left of the second pair. If pair vectors have corresponding vectors in Σ^n, they are said to be *consistent*.

The main error model that considered for codes over symbol-pair read channels is defined as follows.

Definition 2. Let $\boldsymbol{x} = [x_0, \ldots, x_{n-1}]$ be a vector in Σ^n. A pair vector $\overleftrightarrow{\boldsymbol{u}} = [(u_{l,0}, u_{r,1}), (u_{l,1}, u_{r,2}), \ldots, (u_{l,n-1}, u_{r,0})]$ is the result of a *t-pair error* from $\boldsymbol{x}$, if t is the number of elements in $\{i : (u_{l,i}, u_{r,i+1}) \neq (x_i, x_{i+1})\}$.

Next, I explain the necessary and sufficient condition on codes for correcting pair errors. A central element in the characterization of correctability is the pair distance, defined as follows.

Definition 3. Let $\overleftrightarrow{\boldsymbol{u}}$ and $\overleftrightarrow{\boldsymbol{v}}$ be two pair vectors in $(\Sigma, \Sigma)^n$. The *pair distance* between $\overleftrightarrow{\boldsymbol{u}}$ and $\overleftrightarrow{\boldsymbol{v}}$ is defined as

$$D_p(\overleftrightarrow{\boldsymbol{u}}, \overleftrightarrow{\boldsymbol{v}}) = |\{i : (u_{l,i}, u_{r,i+1}) \neq (v_{l,i}, v_{r,i+1})\}|.$$

For notational aesthetics, when a consistent pair vector is used as an argument to the pair distance, its notation may appear

$$D_p(\boldsymbol{x}, \boldsymbol{y}) \triangleq D_p(\pi(\boldsymbol{x}), \pi(\boldsymbol{y})).$$

Similar to the pair distance, the pair weight can be defined as below.

Definition 4. Let $\overleftrightarrow{\boldsymbol{u}}$ be a pair vector in $(\Sigma, \Sigma)^n$. The *pair weight* of $\overleftrightarrow{\boldsymbol{u}}$ is defined as

$$W_p(\overleftrightarrow{\boldsymbol{u}}) = |\{i : (u_{l,i}, u_{r,i+1}) \neq (0,0)\}|.$$

For $\boldsymbol{x}, \boldsymbol{y}$ in Σ^n, let $0 < D_H(\boldsymbol{x}, \boldsymbol{y}) < n$ be the Hamming distance between $\boldsymbol{x}$ and $\boldsymbol{y}$. Then

$$D_H(\boldsymbol{x}, \boldsymbol{y}) + 1 \leq D_p(\boldsymbol{x}, \boldsymbol{y}) \leq 2D_H(\boldsymbol{x}, \boldsymbol{y}). \tag{2.15}$$

In the extreme case in which $D_H(\boldsymbol{x}, \boldsymbol{y}) = 0$ or $D_H(\boldsymbol{x}, \boldsymbol{y}) = n$, clearly $D_p(\boldsymbol{x}, \boldsymbol{y}) = D_H(\boldsymbol{x}, \boldsymbol{y})$. If the codes have all vectors in Σ^n, the minimum pair distance between distinct codewords is 2, hence it can detect a single pair error. Further, for a set $S_H = \{j : x_j \neq y_j\}$, If $S_H = \bigcup_{l=1}^{L} B_l$ is a minimal partition of the set S_H to subsets of consecutive indices, then

$$D_p(\boldsymbol{x}, \boldsymbol{y}) = D_H(\boldsymbol{x}, \boldsymbol{y}) + L. \tag{2.16}$$

In addition, the minimum pair distance of a code $C \subset \Sigma^n$ is defined as

$$d_p = \min_{\boldsymbol{x}, \boldsymbol{y} \in C, \boldsymbol{x} \neq \boldsymbol{y}} D_p(\pi(\boldsymbol{x}), \pi(\boldsymbol{y})). \tag{2.17}$$

A code over a symbol-pair read channel is linear if the code recorded in storage applications is linear; thus, the minimum pair distance equals the minimum pair weight.

The pair distance has triangle inequality just like the Hamming distance; hence, the set $(\Sigma, \Sigma)^n$ with the pair distance is a metric space. Based on the properties of the pair distance, the necessary and sufficient condition for correctability of t-pair errors is given by the following theorem.

Theorem 1. [3] A code C can correct t-pair errors if and only if $d_p \geq 2t + 1$.

From Theorem 1, the pair error-correcting capability t_p of the code with the minimum pair distance d_p is given by

$$t_p = \left\lfloor \frac{d_p - 1}{2} \right\rfloor. \tag{2.18}$$

2.2.2 Decoding of Symbol-Pair Codes

Cassuto-Blaum Decoding Algorithm [3]

In Ref. [3], Cassuto and Blaum proposed a decoding algorithm by reducing the pair-decoding problem to error and erasure decoding in the Hamming metric.

[Cassuto-Blaum Decoding Algorithm]
Let $\overleftrightarrow{\boldsymbol{u}} = [(u_{l,0}, u_{r,1}), (u_{l,1}, u_{r,2}), \ldots, (u_{l,n-1}, u_{r,0})]$ be the received pair-vector. Define the n symbols of the vector $\boldsymbol{z}$ as

$$z_i = \begin{cases} u_i & \text{if } u_{l,i} = u_{r,i-1} \\ * & \text{otherwise} \end{cases}.$$

The symbol $*$ represents symbol erasure and is used when symbol hypotheses from the two pairs are in conflict. The vector $\boldsymbol{z}$ is now input to an error/erasure decoder in the Hamming metric. ■

While any code can be decoded using this algorithm, a critical question is whether this algorithm provides a guarantee to find the unique codeword within a pair-ball of radius $\lfloor (d_p - 1)/2 \rfloor$ around the received pair vector. Cassuto and Blaum mentioned that the answer turns out to be no in general, and yes for interleaved codes. The following example proves that this algorithm is inferior to a bounded-distance pair decoder.

Example 1. Suppose a single pair error-correcting code with the two codewords $\{[00000], [01100]\}$ of minimum pair distance 3 is used, and the pair vector $\overleftrightarrow{\boldsymbol{u}} = [(0,0),(1,1),(0,0),(0,0),(0,0)]$ is received. Cassuto-Blaum decoding algorithm transforms $\overleftrightarrow{\boldsymbol{u}}$ into $\boldsymbol{z} = [0,*,*,0,0]$. Then a Hamming-metric decoder fails to decode, because both codewords are equally likely given the decoder input $\boldsymbol{z}$. On the other hand, a pair-decoder can discern that $\overleftrightarrow{\boldsymbol{u}}$ is at pair distance 1 to [00000] and at pair distance 2 to [01100], hence successfully choosing the vector [00000] from within the radius-1 pair-ball.

From the above description, the Cassuto-Blaum decoding algorithm cannot correct all pair errors within the pair error-correcting capability.

Yaakobi-Bruck-Siegel Decoding Algorithm

In Ref [6], Yaakobi, Bruck and Siegel proved that, for *linear cyclic* codes with the minimum Hamming distance d_H, the minimum pair distance holds

$$d_p \geq d_H + \left\lceil \frac{d_H}{2} \right\rceil,$$

and they showed how to use decoders of cyclic codes in order to construct a decoder for symbol-pair codes which corrects up to

$$t_0 = \lfloor (3t_H + 1)/2 \rfloor \tag{2.19}$$

pair errors.

Given a linear cyclic code C with the minimum Hamming distance $d_H = 2t_H + 1$, I assume it has a decoder D_C that can correct up to t_H errors. The decoder D_C is defined as a map $D_C : \Sigma^n \to C \cup \{F\}$ and the notation $D_C(\boldsymbol{u}) = \hat{\boldsymbol{c}}$ indicates that the decoder's input is a received word $\boldsymbol{u}$ and its output is a decoded codeword $\hat{\boldsymbol{c}}$ or the decoder failure symbol F. The double-repetition code of C is the code

$$C_2 = \{(\boldsymbol{c}, \boldsymbol{c}) : \boldsymbol{c} \in C\}.$$

This code is of length $2n$ and the minimum Hamming distance satisfies $2d_H$. The code C_2 can correct up to $2t_H$ errors and I assume that it has a decoder $D_{C_2} : \Sigma^n \times \Sigma^n \to \Sigma^n \cup \{F\}$ having the same properties as the decoder D_C. Every codeword in C_2 consists of two identical codewords from C. Thus, I assume that the decoder D_{C_2} returns only one copy of the decoded codeword from C.

Let $\boldsymbol{c} \in C$ and let $\pi(\boldsymbol{c}) \in \pi(C)$ be its pair vector. Let $\overleftrightarrow{\boldsymbol{u}} = \pi(\boldsymbol{c}) + \overleftrightarrow{\boldsymbol{e}}$ be a received word, where $\boldsymbol{e} \in (\Sigma, \Sigma)^n$ is the error vector and $w_H(\boldsymbol{e}) \leq \lfloor (3t_H + 1)/2 \rfloor$. The decoder of symbol-pair codes is defined as a map $D_\pi : (\Sigma, \Sigma)^n \to \{0,1\}^n$ which receives the word $\overleftrightarrow{\boldsymbol{u}}$ and returns $\hat{\boldsymbol{c}} \in C$. Let

$$\overleftrightarrow{\boldsymbol{u}} = [(u_{l,0}, u_{r,1}), (u_{l,1}, u_{r,2}), \ldots, (u_{l,n-1}, u_{r,0})]$$

be a received vector, and three vectors are defined as follows:

$$\boldsymbol{u}_L = [u_{l,0}, u_{l,1} \ldots, u_{l,n-1}], \tag{2.20}$$
$$\boldsymbol{u}_R = [u_{r,1}, u_{r,2}, \ldots, u_{r,0}], \tag{2.21}$$
$$\boldsymbol{u}_S = [u_{l,0} \oplus u_{r,1}, u_{l,1} \oplus u_{r,2}, \ldots, u_{l,n-1} \oplus u_{r,0}]. \tag{2.22}$$

Since the vector $\boldsymbol{u}$ suffers from at most t_0 pair errors, the vectors $\boldsymbol{u}_L$ and $\boldsymbol{u}_R$ have at most t_0 errors as well, respectively. The vector $\boldsymbol{u}_S$ has at most t_0 errors with respect to the codeword $\boldsymbol{c}' = [c_0 + c_1, c_1 + c_2, \ldots, c_{n-1} + c_0]$. Then, the following lemma holds.

Lemma 1. [6] If the codeword $\boldsymbol{c}' \in C$ is successfully decoded then I can recover the codeword $\boldsymbol{c}$.

The codeword $\boldsymbol{c}$ satisfies $c_i = c_0 + \sum_{j=0}^{i-1} c'_j$. Hence, if $\tilde{\boldsymbol{c}} = [\tilde{c}_0, \tilde{c}_1, \ldots, \tilde{c}_{n-1}]$ is defined by $\tilde{c}_i = \sum_{j=0}^{i-1} c'_j$ then the codeword $\boldsymbol{c}$ is either $\tilde{\boldsymbol{c}}$ or $\tilde{\boldsymbol{c}} + \boldsymbol{1}$, depending on the value of c_0. The distance between $\boldsymbol{u}_L$ and $\boldsymbol{c}$ is at most t_0 and $d_H(\tilde{\boldsymbol{c}}, \tilde{\boldsymbol{c}} + \boldsymbol{1}) = n$. Hence, if $d_H(\boldsymbol{u}_L, \tilde{\boldsymbol{c}}) < d_H(\boldsymbol{u}_L, \tilde{\boldsymbol{c}} + \boldsymbol{1})$ then $\boldsymbol{c} = \tilde{\boldsymbol{c}}$ and otherwise $\boldsymbol{c} = \tilde{\boldsymbol{c}} + \boldsymbol{1}$. Therefore, it is possible to recover the codeword $\boldsymbol{c}$ from the codeword $\boldsymbol{c}'$. By abuse of notation, I denote by $\boldsymbol{c}'^{*}$ an operator that calculates, as explained in Lemma 1, the codeword $\boldsymbol{c}$ from $\boldsymbol{c}'$, and so $\boldsymbol{c}'^{*} = \boldsymbol{c}$.

The number of pair errors in the vector $\boldsymbol{u}$ is at most t_0. Each pair error corresponds to one or two bit error in the pair. Let E_1 be the number of single-bit pair errors and E_2 be the number of double-bit pair errors, where $E_1 + E_2 \leq t_0$. Then, the following lemma holds.

Lemma 2. [6] If $\boldsymbol{c} \in C$, $\overleftrightarrow{\boldsymbol{u}} = \pi(\boldsymbol{c}) + \overleftrightarrow{\boldsymbol{e}}$ and $w_H(\overleftrightarrow{\boldsymbol{e}}) \leq t_0$, then either $D_C(\boldsymbol{u}_S) = \boldsymbol{c}'$ or $D_{C_2}((\boldsymbol{u}_L, \boldsymbol{u}_R)) = \boldsymbol{c}$.

Lemma 2 implies that at least one of the two decoders succeeds. Yaakobi et al. showed that the decoder's output $D_\pi(\overleftrightarrow{\boldsymbol{u}}) = \hat{\boldsymbol{c}}$ is calculated as follows.
Yaakobi-Bruck-Siegel Decoding Algorithm [6]

Step1. $\boldsymbol{c}_1 = D_C(\boldsymbol{u}_S), e_1 = d_H(\boldsymbol{c}_1, \boldsymbol{u}_S)$.

Step2. $\boldsymbol{c}_2 = D_{C_2}((\boldsymbol{u}_L, \boldsymbol{u}_R)), e_2 = d_H((\boldsymbol{c}_2, \boldsymbol{c}_2), (\boldsymbol{u}_L, \boldsymbol{u}_R))$.

Step3. If $\boldsymbol{c}_1 = F$ or $w_H(\boldsymbol{c}_1)$ is odd then $\hat{\boldsymbol{c}} = \boldsymbol{c}_2$.

Step4. If $e_1 \leq \lfloor (t_H + 2)/2 \rfloor$, then $\hat{\boldsymbol{c}} = \boldsymbol{c}_1^*$.

Step5. If $e_1 > \lfloor (t_H + 2)/2 \rfloor$, let $e_1 = \lfloor (t_H + 2)/2 \rfloor + a, (1 \leq a \leq \lceil t_H/2 \rceil - 1)$

a) If $e_2 \leq t_0 + a$ then $\hat{\boldsymbol{c}} = \boldsymbol{c}_2$.

b) Otherwise, $\hat{\boldsymbol{c}} = \boldsymbol{c}_1^*$. ■

Note that this algorithm can correct $t_0 = \lfloor (3t_H + 1)/2 \rfloor$ pair errors for a given cyclic code with the minimum Hamming distance $d_H = 2t_H + 1$. On the other hand, from (2.15), the minimum pair distance of the cyclic code satisfies

$$d_H + 1 = 2t_H + 2 \leq d_p \leq 4t_H + 2 = 2d_H.$$

Hence, it is possible to be symbol-pair code which corrects up to $\lfloor (d_p - 1)/2 \rfloor = 2t_H$ pair errors. For such a code, Yaakobi-Bruck-Siegel decoding algorithm cannot correct all error within the pair error-correcting capability.

Chapter 3

Syndrome Decoding of Linear Codes over Symbol-Pair Read Channels

3.1 Abstract

Cassuto et al. and Yaakobi et al. presented decoding algorithms for symbol-pair codes. However, their decoding algorithms cannot always correct errors whose number is not more than half the minimum pair distance. In this chapter, I newly define a parity-check matrix and two types of syndromes of symbol-pair codes. Further, I prove that the pair of two syndromes is unique for each error vector whose number of pair errors is not more than the half of the minimum pair distance.

3.2 Parity-Check Matrix and Syndrome of Symbol-Pair Codes

I firstly define the parity-check matrix of symbol-pair codes. Let

$$\mathbf{H} = \begin{bmatrix} \boldsymbol{h}_0 \\ \boldsymbol{h}_1 \\ \vdots \\ \boldsymbol{h}_{n-k-1} \end{bmatrix} = \begin{bmatrix} h_{0,0} & h_{0,1} & \cdots & h_{0,n-1} \\ h_{1,0} & h_{1,1} & \cdots & h_{1,n-1} \\ \vdots & \vdots & & \vdots \\ h_{n-k-1,0} & h_{n-k-1,1} & \cdots & h_{n-k-1,n-1} \end{bmatrix}$$

be a parity-check matrix of linear block codes. By representing each row of the parity-check matrix by symbol-pair vector,

$$\begin{aligned} \pi(\mathbf{H}) &= \begin{bmatrix} \pi(\boldsymbol{h}_0) \\ \pi(\boldsymbol{h}_1) \\ \vdots \\ \pi(\boldsymbol{h}_{n-k-1}) \end{bmatrix} \\ &= \begin{bmatrix} (h_{0,0}, h_{0,1}) & (h_{0,1}, h_{0,2}) & \cdots & (h_{0,n-1}, h_{0,0}) \\ (h_{1,0}, h_{1,1}) & (h_{1,1}, h_{1,2}) & \cdots & (h_{1,n-1}, h_{1,0}) \\ \vdots & \vdots & & \vdots \\ (h_{n-k-1,0}, h_{n-k-1,1}) & (h_{n-k-1,1}, h_{n-k-1,2}) & \cdots & (h_{n-k-1,n-1}, h_{n-k-1,0}) \end{bmatrix}. \end{aligned} \tag{3.1}$$

I call it a *symbol-pair parity-check matrix.*

Let $\overleftrightarrow{\boldsymbol{u}}$ be a received pair vector. Then, I calculate the syndrome by multiplying $\overleftrightarrow{\boldsymbol{u}}$ by the transpose of the symbol-pair parity-check matrix.

$$\begin{aligned}\overleftrightarrow{\boldsymbol{s}} &\triangleq [(s_{l,0}, s_{r,1}), (s_{l,1}, s_{r,2}), \ldots, (s_{l,n-k-1}, s_{r,0})] \\ &= \overleftrightarrow{\boldsymbol{u}} \cdot \pi(\mathbf{H})^{\mathrm{T}}.\end{aligned} \tag{3.2}$$

I call it a *symbol-pair syndrome*. This calculation requires the inner product of the pair vector and each row of the symbol-pair parity-check matrix.

The inner product of two pair vectors $\overleftrightarrow{\boldsymbol{u}} = [(u_{l,0}, u_{r,1}), (u_{l,1}, u_{r,2}), \ldots, (u_{l,n-1}, u_{r,0})]$ and $\overleftrightarrow{\boldsymbol{v}} = [(v_{l,0}, v_{r,1}), (v_{l,1}, v_{r,2}), \ldots, (v_{l,n-1}, v_{r,0})]$ is calculated as follows:

$$\begin{aligned}\overleftrightarrow{\boldsymbol{u}} \cdot \overleftrightarrow{\boldsymbol{v}} = [((u_{l,0} \cdot v_{l,0} + u_{l,1} \cdot v_{l,1} + \cdots + u_{l,n-1} \cdot v_{l,n-1}, \\ u_{r,1} \cdot v_{r,1} + u_{r,2} \cdot v_{r,2} + \cdots + u_{r,0} \cdot v_{r,0})].\end{aligned} \tag{3.3}$$

For example, the inner product of $[(0,1), (1,0), (1,0)]$ and $[(0,0), (0,1), (1,0)]$ is calculated by

$$\begin{aligned}&[(0,1), (1,0), (1,0)] \cdot [(0,0), (0,1), (1,0)] \\ &= [(0 \cdot 0{+}1 \cdot 0{+}1 \cdot 1, 1 \cdot 0{+}0 \cdot 1{+}0 \cdot 0)] \\ &= [(1,0)].\end{aligned}$$

When a symbol-pair code is constructed from a linear block code with the minimum Hamming distance d_H, the minimum pair distance is more than d_H because of (2.15). Thus, the symbol-pair code constructed from the t_H-error-correcting code can correct more than t_H-pair errors.

In the symbol-pair syndrome calculation of (3.2), the right and left symbols in pairs of the symbol-pair syndrome are independently calculated by the same way as the syndrome of t_H-error-correcting code. Therefore, for distinct pair error patterns whose number is beyond t_H in the right or left symbol, the pair error patterns may have the same symbol-pair syndromes.

Example 2. Consider the $(3,2)$ single-parity-check code C_1. Although this code is a single error detecting code, its symbol-pair code has the minimum pair distance 3 and is the 1-pair error-correcting code. The symbol-pair parity-check matrix of this code is given by

$$\pi(\mathbf{H}) = \begin{bmatrix} (1,1) & (1,1) & (1,1) \end{bmatrix}.$$

Then, the symbol-pair syndrome of three error vectors

$$\begin{aligned}\overleftrightarrow{\boldsymbol{e}}_1 &= [(0,1), (0,0), (0,0)], \\ \overleftrightarrow{\boldsymbol{e}}_2 &= [(0,0), (0,1), (0,0)], \\ \overleftrightarrow{\boldsymbol{e}}_3 &= [(0,0), (0,0), (0,1)].\end{aligned}$$

is $[(0,1)]$. Since the symbol-pair syndrome is not unique for each error vector, I cannot find error vectors by the symbol-pair syndromes. Therefore, I cannot correct pair errors by using only symbol-pair syndrome while the pair errors are within pair error capability.

3.3 Proposed Syndrome Decoding Algorithm

I introduce a new syndrome which indicates the pairs of received pair vector in conflict. The syndrome is defined as follows:

$$\boldsymbol{S} \triangleq [S_0, S_1, \ldots, S_{n-1}], \tag{3.4}$$

$$S_i = \begin{cases} 0 & \text{if } u_{l,i} = u_{r,i} \\ 1 & \text{otherwise} \end{cases}. \tag{3.5}$$

It is called a *neighbor-symbol syndrome*. If the left symbol of i-th pair and the right symbol of $(i-1)$-th pair are consistent then the i-th symbol of the neighbor-symbol syndrome is 0, otherwise, the i-th symbol of the neighbor-symbol syndrome is 1.

The i-th symbol of neighbor-symbol syndrome is easily obtained by XORing

$$S_i = u_{l,i} \oplus u_{r,i},$$

i.e., the neighbor-symbol syndrome is calculated as follows:

$$\boldsymbol{S} = [u_{l,0} \oplus u_{r,0}, u_{l,1} \oplus u_{r,1}, \ldots, u_{l,n-1} \oplus u_{r,n-1}].$$

When a pair vector $[(0,0),(1,1),(1,0),(0,0),(0,0)]$ is received, the neighbor-symbol syndrome is $\boldsymbol{S} = [01000]$. It implies that the left symbol of the second pair and the right symbol of the first pair of received pair vector are in conflict.

Using the neighbor-symbol syndrome and the symbol-pair syndrome, I derive the following theorem.

Theorem 2. If a code C can correct t_p-pair errors, the pair of the symbol-pair syndrome and the neighbor-symbol syndrome $(\overleftrightarrow{\boldsymbol{s}}, \boldsymbol{S})$ is unique for each error vector $\overleftrightarrow{\boldsymbol{e}}$ where $W_p(\overleftrightarrow{\boldsymbol{e}}) \leq t_p$.

Proof: Consider the case that a code C can correct t_p-pair errors. Let $\overleftrightarrow{\boldsymbol{e}}$ and $\overleftrightarrow{\boldsymbol{e}}'$ be distinct error vectors whose pair weight is not more than t_p. The symbol-pair syndrome and neighbor-symbol syndrome of $\overleftrightarrow{\boldsymbol{e}}$ and $\overleftrightarrow{\boldsymbol{e}}'$ are written as

$$\begin{aligned}
\overleftrightarrow{\boldsymbol{s}} &= \overleftrightarrow{\boldsymbol{e}} \cdot \pi(\mathbf{H})^{\mathrm{T}}, \\
\boldsymbol{S} &= [e_{l,0} \oplus e_{r,0}, e_{l,1} \oplus e_{r,1}, \ldots, e_{l,n-1} \oplus e_{r,n-1}], \\
\overleftrightarrow{\boldsymbol{s}}' &= \overleftrightarrow{\boldsymbol{e}}' \cdot \pi(\mathbf{H})^{\mathrm{T}}, \\
\boldsymbol{S}' &= [e'_{l,0} \oplus e'_{r,0}, e'_{l,1} \oplus e'_{r,1}, \ldots, e'_{l,n-1} \oplus e'_{r,n-1}].
\end{aligned}$$

Suppose that the symbol-pair syndrome and the neighbor-symbol syndrome of $\overleftrightarrow{\boldsymbol{e}}$ are same as those of $\overleftrightarrow{\boldsymbol{e}}'$.

$$\begin{aligned}
\overleftrightarrow{\boldsymbol{s}} &= \overleftrightarrow{\boldsymbol{s}}', \\
\boldsymbol{S} &= \boldsymbol{S}'.
\end{aligned}$$

If the symbol-pair syndrome of $\overleftrightarrow{\boldsymbol{e}}$ is equal to that of $\overleftrightarrow{\boldsymbol{e}}'$, error vectors satisfy

$$\overleftrightarrow{\boldsymbol{e}}' = \overleftrightarrow{\boldsymbol{e}} \oplus [(c_{1,0}, c_{2,1}), (c_{1,1}, c_{2,2}), \ldots, (c_{1,n-1}, c_{2,0})],$$

where $\boldsymbol{c}_1 = (c_{1,0}, \ldots, c_{1,n-1})$ and $\boldsymbol{c}_2 = (c_{2,0}, \ldots, c_{2,n-1})$ are codewords of C. Furthermore, if the neighbor-symbol syndrome of $\overleftrightarrow{\boldsymbol{e}}$ is equal to that of $\overleftrightarrow{\boldsymbol{e}}'$, the conflict pairs of $\overleftrightarrow{\boldsymbol{e}}$ are corresponding to that of $\overleftrightarrow{\boldsymbol{e}}'$. Then, $[(c_{1,0}, c_{2,1}), (c_{1,1}, c_{2,2}), \ldots, (c_{1,n-1}, c_{2,0})]$ is a consistent pair vector. Hence $\boldsymbol{c}_1$ and $\boldsymbol{c}_2$ are the same nonzero codewords. Therefore, the error vectors satisfy

$$\overleftrightarrow{\boldsymbol{e}}' = \overleftrightarrow{\boldsymbol{e}} \oplus \pi(\boldsymbol{c}_1).$$

Since the minimum pair distance of C is $d_p = 2t_p + 1$, the pair weight of the codeword $W_p(\boldsymbol{c}_1)$ is more than d_p. Therefore, I obtain the following relation:

$$\begin{aligned} W_p(\overleftrightarrow{\boldsymbol{e}}') &\geq W_p(\boldsymbol{c}_1) - W_p(\overleftrightarrow{\boldsymbol{e}}) \\ &\geq 2t_p + 1 - t_p \\ &= t_p + 1. \end{aligned}$$

It is conflicted that the pair weight of error vectors $\overleftrightarrow{\boldsymbol{e}}$ and $\overleftrightarrow{\boldsymbol{e}}'$ is not more than t_p. Consequently, the pair of the symbol-pair syndrome and the neighbor-symbol syndrome $(\overleftrightarrow{\boldsymbol{s}}, \boldsymbol{S})$ is unique for each error vector $\overleftrightarrow{\boldsymbol{e}}$ where $W_p(\overleftrightarrow{\boldsymbol{e}}) \leq t_p$. □

I propose a decoding algorithm of symbol-pair codes using the symbol-pair syndrome and the neighbor-symbol syndrome.

Syndrome Decoding Algorithm of Symbol-Pair Codes

Step1. Calculate the symbol-pair syndrome and the neighbor-symbol syndrome of the received pair vector $\overleftrightarrow{\boldsymbol{u}}$.

$$\begin{aligned} \overleftrightarrow{\boldsymbol{s}} &= \overleftrightarrow{\boldsymbol{u}} \cdot \pi(\mathbf{H})^{\mathrm{T}}. \\ \boldsymbol{S} &= [u_{l,0} \oplus u_{r,0}, u_{l,1} \oplus u_{r,1}, \ldots, u_{l,n-1} \oplus u_{r,n-1}]. \end{aligned}$$

Step2. Locate the coset leader $\overleftrightarrow{\boldsymbol{e}}$ whose symbol-pair syndrome and neighbor-symbol syndrome are equal to $\overleftrightarrow{\boldsymbol{s}}$ and $\boldsymbol{S}$.

Step3. Assume that $\overleftrightarrow{\boldsymbol{e}}$ is the error vector caused by the channel, decode the received pair vector $\overleftrightarrow{\boldsymbol{w}}$ into the pair vector.

$$\overleftrightarrow{\boldsymbol{w}} = \overleftrightarrow{\boldsymbol{u}} \oplus \overleftrightarrow{\boldsymbol{e}}$$

Step4. Transform the pair vector $\overleftrightarrow{\boldsymbol{w}}$ into the codeword $\boldsymbol{w} = (w_0, w_1, \ldots, w_{n-1})$ by

$$w_i = \triangleleft w_i,$$

and output the codeword $\boldsymbol{w}$. ■

For symbol-pair codes with the minimum pair distance $d_p = 2t_p + 1$, the proposed algorithm can correct all pair error patterns whose number of pair errors is not more than t_p. I show an example of pair error correction by the proposed algorithm using the $(3, 2)$ single-parity-check code.

Example 3. Consider again about the $(3, 2)$ single-parity-check code C_1 in Example 2. Its symbol-pair code is 1-pair error-correcting code. Table 3.1 shows the standard array for the symbol-pair code of the $(3, 2)$ single-parity-check code. It can be constructed by the similar way of Hamming-metric codes [1][2]. When the standard array is formed, each coset leader is chosen to be a pair vector of least pair weight from the remaining pair vectors. Table 3.1 lists 10 cosets including $1+\binom{3}{1}(2^2-1)$ coset leaders corresponding to correctable error patterns within 1-pair error. The decoder has the correctable pair error patterns corresponding to the symbol-pair syndromes and the neighbor-symbol syndromes. When a codeword $\boldsymbol{c} = [011]$ is sent and 1-pair error is occur in the channel, $\overleftrightarrow{\boldsymbol{u}} = [(0,1),(1,0),(1,0)]$ is a received pair vector. The proposed algorithm decodes $\overleftrightarrow{\boldsymbol{u}}$ into the codeword as follows:

Table 3.1: Standard array for the symbol-pair code of the $(3, 2)$ single-parity-check code

Symbol-pair syndrome $\overleftrightarrow{\boldsymbol{s}}$	Neighbor-symbol syndrome $\boldsymbol{S}$	Coset			
		Coset leader			
[(0,0)]	[000]	[(0,0),(0,0),(0,0)]	[(0,1),(1,1),(1,0)]	[(1,0),(0,1),(1,1)]	[(1,1),(1,0),(0,1)]
[(1,0)]	[100]	[(1,0),(0,0),(0,0)]	[(1,1),(1,1),(1,0)]	[(0,0),(0,1),(1,1)]	[(0,1),(1,0),(0,1)]
[(0,1)]	[010]	[(0,1),(0,0),(0,0)]	[(0,0),(1,1),(1,0)]	[(1,1),(0,1),(1,1)]	[(1,0),(1,0),(0,1)]
[(1,1)]	[110]	[(1,1),(0,0),(0,0)]	[(1,0),(1,1),(1,0)]	[(0,1),(0,1),(1,1)]	[(0,0),(1,0),(0,1)]
[(1,0)]	[010]	[(0,0),(1,0),(0,0)]	[(0,1),(0,1),(1,0)]	[(1,0),(1,1),(1,1)]	[(1,1),(0,0),(0,1)]
[(0,1)]	[001]	[(0,0),(0,1),(0,0)]	[(0,1),(1,0),(1,0)]	[(1,0),(0,0),(1,1)]	[(1,1),(1,1),(0,1)]
[(1,1)]	[011]	[(0,0),(1,1),(0,0)]	[(0,1),(0,0),(1,0)]	[(1,0),(1,0),(1,1)]	[(1,1),(0,1),(0,1)]
[(1,0)]	[001]	[(0,0),(0,0),(1,0)]	[(0,1),(1,1),(0,0)]	[(1,0),(0,1),(0,1)]	[(1,1),(1,0),(1,1)]
[(0,1)]	[100]	[(0,0),(0,0),(0,1)]	[(0,1),(1,1),(1,1)]	[(1,0),(0,1),(1,0)]	[(1,1),(1,0),(0,0)]
[(1,1)]	[101]	[(0,0),(0,0),(1,1)]	[(0,1),(1,1),(0,1)]	[(1,0),(0,1),(0,0)]	[(1,1),(1,0),(1,0)]

Step1. Calculate the symbol-pair syndrome and the neighbor-symbol syndrome of $\overleftrightarrow{\boldsymbol{u}} = [(0, 1), (1, 0), (1, 0)]$.

$$\begin{aligned}\overleftrightarrow{\boldsymbol{s}} &= [(0, 1), (1, 0), (1, 0)] \cdot \begin{bmatrix} (1, 1) & (1, 1) & (1, 1) \end{bmatrix}^{\mathrm{T}} \\ &= [(0, 1)], \\ \boldsymbol{S} &= [001].\end{aligned}$$

Step2. Locate the coset leader $\overleftrightarrow{\boldsymbol{e}}$ whose symbol-pair syndrome and neighbor-symbol syndrome are equal to $[(0, 1)]$ and $[001]$. Then

$$\overleftrightarrow{\boldsymbol{e}} = [(0, 0), (0, 1), (0, 0)]$$

is assumed to be the error pattern.

Step3. Decode the received vector $\overleftrightarrow{\boldsymbol{u}}$ into the pair vector as follows:

$$\begin{aligned}\overleftrightarrow{\boldsymbol{w}} &= \overleftrightarrow{\boldsymbol{u}} \oplus \overleftrightarrow{\boldsymbol{e}} \\ &= [(0, 1), (1, 0), (1, 0)] \oplus [(0, 0), (0, 1), (0, 0)] \\ &= [(0, 1), (1, 1), (1, 0)].\end{aligned}$$

Step4. Transform the pair vector $\overleftrightarrow{\boldsymbol{w}}$ into the codeword as follows:

$$\overleftrightarrow{\boldsymbol{w}} = [(0, 1), (1, 1), (1, 0)] \to \boldsymbol{w} = [011].$$

The proposed algorithm corrects 1-pair error and outputs the codeword [011].

3.4 Conclusion

In this chapter, I have proposed a new syndrome decoding algorithm of symbol-pair codes. This algorithm uses two types of syndromes which are the symbol-pair syndrome and the neighbor-symbol syndrome.

I have shown the significant property that the pair of the symbol-pair syndrome and the neighbor-symbol syndrome is unique for each error vector whose number of pair errors is not more than the half of the minimum pair distance. It leads simple and elegant decoding algorithm that can correct pair errors up to the decoding radius.

Chapter 4

A Decoding Algorithm of Cyclic Codes over Symbol-Pair Read Channels

4.1 Abstract

Yaakobi et al. proved a lower bound on the minimum pair distance of cyclic codes. Furthermore, they provided a decoding algorithm for correcting pair errors using a decoder for cyclic codes, and showed the number of pair errors that can be corrected by their algorithm. However, their algorithm cannot correct all pair error vectors within half of the minimum pair distance.

In this chapter, I propose an efficient decoding algorithm for cyclic codes over symbol-pair read channels. It is based on the relationship between pair errors and syndromes. In addition, I show that the proposed algorithm can correct more pair errors than Yaakobi's algorithm.

4.2 Relationship Between Syndromes and Error Vectors

Let $\pi(\boldsymbol{c}) = [(c_0, c_1), (c_1, c_2), \ldots, (c_{n-1}, c_0)]$ be a pair vector of a codeword $\boldsymbol{c} = [c_0, c_1, \ldots, c_{n-1}]$ of a cyclic code C. Let $\boldsymbol{c}_L = [c_0, c_1, \ldots, c_{n-1}]$ be the left vector of $\pi(\boldsymbol{c})$, and let $\boldsymbol{c}_R = [c_1, c_2, \ldots, c_0]$ be the right vector of $\pi(\boldsymbol{c})$. Here, $\boldsymbol{c}_L$ equals $\boldsymbol{c}$ and $\boldsymbol{c}_R$ equals the vector cyclic-shifting $\boldsymbol{c}$ to the left by one bit, so $\boldsymbol{c}_L$ and $\boldsymbol{c}_R$ are the codewords of the cyclic code C. Furthermore, let $\boldsymbol{c}_S = [c_0 \oplus c_1, c_1 \oplus c_2, \ldots, c_{n-1} \oplus c_0]$ be a vector XORing $\boldsymbol{c}_L$ and $\boldsymbol{c}_R$. Note that $\boldsymbol{c}_S$ also is a codeword of the cyclic code C since $\boldsymbol{c}_S$ is the vector XORing the codewords of the cyclic code C. Therefore, when $\boldsymbol{c}_L$, $\boldsymbol{c}_R$, and $\boldsymbol{c}_S$ suffer from errors, they can be corrected by the decoder D_C for the cyclic code C. In addition, the codeword $\boldsymbol{c}$ can be transformed from $\boldsymbol{c}_S$ as follows. First, $\tilde{\boldsymbol{c}}$ is calculated by as

$$\tilde{c}_i = \begin{cases} 0 & \text{if } i = 0 \\ \sum_{j=0}^{i-1} c_{S,j} & \text{otherwise} \end{cases} . \tag{4.1}$$

Next, $\boldsymbol{c}$ is calculated as

$$\boldsymbol{c} = \begin{cases} \tilde{\boldsymbol{c}} & \text{if } D_H(\boldsymbol{u}_L, \tilde{\boldsymbol{c}}) < D_H(\boldsymbol{u}_L, \tilde{\boldsymbol{c}} + \mathbf{1}) \\ \tilde{\boldsymbol{c}} + \mathbf{1} & \text{otherwise} \end{cases} , \tag{4.2}$$

where $D_H(\boldsymbol{x}, \boldsymbol{y})$ denotes the Hamming distance between the two vectors $\boldsymbol{x}$ and $\boldsymbol{y}$.

When $\boldsymbol{c}_L$, $\boldsymbol{c}_R$, and $\boldsymbol{c}_S$ suffer from errors, I consider that they are corrected independently. Let $\overleftrightarrow{\boldsymbol{u}} = [(u_{l,0}, u_{r,1}), (u_{l,1}, u_{r,2}), \ldots, (u_{l,n-1}, u_{r,0})]$ be a received pair vector. As in the case of (2.20)–(2.22), let $\boldsymbol{u}_L$, $\boldsymbol{u}_R$, and $\boldsymbol{u}_S$ be the left vector of $\overleftrightarrow{\boldsymbol{u}}$, the right vector of $\overleftrightarrow{\boldsymbol{u}}$, and the vector XORing the left and right vectors of $\overleftrightarrow{\boldsymbol{u}}$, respectively. Additionally, the neighbor-symbol syndrome is calculated as follows:

$$\boldsymbol{S} = [u_{l,0} \oplus u_{r,0}, u_{l,1} \oplus u_{r,1}, \ldots, u_{l,n-1} \oplus u_{r,n-1}]. \tag{4.3}$$

I consider the t_H-error correcting cyclic code C over a symbol-pair read channel. I define the notation $D_C(\boldsymbol{u}) = \hat{\boldsymbol{c}}$ indicates that the decoder's input is a received word $\boldsymbol{u}$ and its output is a decoded codeword $\hat{\boldsymbol{c}}$ or the decoder failure symbol F. If $\boldsymbol{c} \in C$ is the transmitted word and $D_H(\boldsymbol{c}, \boldsymbol{u}) \leq t_H$, then it is guaranteed that $\hat{\boldsymbol{c}} = \boldsymbol{c}$. However, if $D_H(\boldsymbol{c}, \boldsymbol{u}) > t_H$, then either $\hat{\boldsymbol{c}} = F$, indicating that more than t_H errors have occurred, or $\hat{\boldsymbol{c}} = \boldsymbol{c}'$, where $\boldsymbol{c}'$ is a codeword different from $\boldsymbol{c}$, whose Hamming distance from the received word $\boldsymbol{u}$ is at most t_H, i.e., $D_H(\boldsymbol{c}', \boldsymbol{u}) \leq t_H$. Then, $\boldsymbol{u}_L$ can be corrected by the decoder D_C if $\boldsymbol{u}_L$ has at most t_H errors. However, $\boldsymbol{u}_L$ cannot be corrected by the decoder D_C if $\boldsymbol{u}_L$ has more than t_H errors. Similarly, $\boldsymbol{u}_R$ and $\boldsymbol{u}_S$ can be corrected by the decoder D_C if they have at most t_H errors, but they cannot be corrected by the decoder D_C if they have more than t_H errors. I consider that these uncorrectable errors are corrected by using the neighbor-symbol syndrome. The pair error vector is written as follows:

$$\begin{aligned}
\boldsymbol{e}_L &= [e_{l,0}, e_{l,1} \ldots, e_{l,n-1}], \\
\boldsymbol{e}_R &= [e_{r,1}, e_{r,2}, \ldots, e_{r,0}], \\
\boldsymbol{e}_S &= [e_{l,0} \oplus e_{r,1}, e_{l,1} \oplus e_{r,2}, \ldots, e_{l,n-1} \oplus e_{r,0}].
\end{aligned}$$

Then, the neighbor-symbol syndrome $\boldsymbol{S}$ has the following relation:

$$\begin{aligned}
\boldsymbol{S} &= [u_{l,0} \oplus u_{r,0}, u_{l,1} \oplus u_{r,1}, \ldots, u_{l,n-1} \oplus u_{r,n-1}] \\
&= [(c_0 \oplus e_{l,0}) \oplus (c_0 \oplus e_{r,0}), \ldots, (c_{n-1} \oplus e_{l,n-1}) \oplus (c_{n-1} \oplus e_{r,n-1})] \\
&= [e_{l,0} \oplus e_{r,0}, \ldots, e_{l,n-1} \oplus e_{r,n-1}] \\
&= \boldsymbol{e}_L \oplus (\boldsymbol{e}_R \gg 1),
\end{aligned} \tag{4.4}$$

where $(\boldsymbol{e}_R \gg 1)$ is the vector cyclic-shifting $\boldsymbol{e}_R$ to the right by one bit.

Therefore, when $\boldsymbol{e}_L$ is found, $\boldsymbol{e}_R$ is obtained by

$$\boldsymbol{e}_R = (\boldsymbol{S} \oplus \boldsymbol{e}_L) \ll 1. \tag{4.5}$$

When $\boldsymbol{e}_R$ is found, $\boldsymbol{e}_L$ is also obtained by

$$\boldsymbol{e}_L = \boldsymbol{S} \oplus (\boldsymbol{e}_R \gg 1). \tag{4.6}$$

4.3 Classification of Pair Error Vectors

I classify the pair error vectors into four cases according to the number of errors in the left and right vectors of $\overleftrightarrow{\boldsymbol{e}}$. When the Hamming weight of a vector $\boldsymbol{x}$ is denoted by $W_H(\boldsymbol{x})$, the classification is as follows.

Case 1: $W_H(\boldsymbol{e}_L) \leq t_H$ and $W_H(\boldsymbol{e}_R) \leq t_H$.

Case 2: $W_H(\boldsymbol{e}_L) \leq t_H$ and $t_H < W_H(\boldsymbol{e}_R) \leq t_p$.

Case 3: $t_H < W_H(\boldsymbol{e}_L) \le t_p$ and $W_H(\boldsymbol{e}_R) \le t_H$.

Case 4: $t_H < W_H(\boldsymbol{e}_L) \le t_p$ and $t_H < W_H(\boldsymbol{e}_R) \le t_p$.

For each case, I discuss how to correct errors in the left and right vectors by using the t_H-error correcting decoder D_C and the neighbor-symbol syndrome.

Case 1: When both left and right vectors have at most t_H errors, $\boldsymbol{e}_L$ and $\boldsymbol{e}_R$ can be calculated from $\boldsymbol{u}_L$ and $\boldsymbol{u}_R$ by the decoder D_C, respectively.

Case 2: When the left vector has at most t_H errors, the decoding operation $\boldsymbol{c}_L = D_c(\boldsymbol{u}_L)$ succeeds and the left error vector is obtained by $\boldsymbol{e}_L = \boldsymbol{c}_L \oplus \boldsymbol{u}_L$. Then, $\boldsymbol{e}_R$ can be calculated by (4.5). Therefore, $\overleftrightarrow{\boldsymbol{e}} = (\boldsymbol{e}_L, \boldsymbol{e}_R)$ is the correctable pair error vector within the pair error correcting capability t_p. On the other hand, when the right vector has more than t_H errors, the decoding operation $D_C(\boldsymbol{u}_R)$ fails and outputs the failure symbol F or a miscorrected codeword $\boldsymbol{c}'_R$. Then, a different error vector is obtained by $\boldsymbol{e}'_R = \boldsymbol{c}'_R \oplus \boldsymbol{u}_R$, and a different error vector $\boldsymbol{e}'_L$ is calculated by (4.6). Therefore, the pair error vector $\overleftrightarrow{\boldsymbol{e}}' = (\boldsymbol{e}'_L, \boldsymbol{e}'_R)$ is different from $\overleftrightarrow{\boldsymbol{e}}$. Moreover, the symbol-pair syndrome of $\overleftrightarrow{\boldsymbol{e}}'$ is equal to that of $\overleftrightarrow{\boldsymbol{e}}$. From Theorem 2, $\overleftrightarrow{\boldsymbol{e}}'$ is not the pair error vector within the pair error correcting capability, and the pair weight of $\boldsymbol{e}'$ is more than t_p. Consequently, even if $\overleftrightarrow{\boldsymbol{e}}'$ is obtained by the decoding failure of operation $D_C(\boldsymbol{u}_R)$, it can be distinguished from $\overleftrightarrow{\boldsymbol{e}}$ by examining the pair weight of $\overleftrightarrow{\boldsymbol{e}}'$.

Case 3: In this case, the number of errors occurring the left and right vectors in Case 2 is swapped. Therefore, $\overleftrightarrow{\boldsymbol{e}} = (\boldsymbol{e}_L, \boldsymbol{e}_R)$ can be calculated by replacing $\boldsymbol{u}_L$ and $\boldsymbol{u}_R$.

Case 4: When both left and right vectors have more than t_H errors, $\boldsymbol{e}_L$ and $\boldsymbol{e}_R$ cannot be obtained by the decoding operations $D_C(\boldsymbol{u}_L)$ and $D_C(\boldsymbol{u}_R)$. As in Case 2, even if a different pair vector $\overleftrightarrow{\boldsymbol{e}}'$ is obtained by the decoding failure, it can be distinguished from $\overleftrightarrow{\boldsymbol{e}}$ by examining the pair weight of $\overleftrightarrow{\boldsymbol{e}}'$. In this case, I consider that the vector $\boldsymbol{u}_S$ is corrected by the decoder D_C. I calculate the codeword $\boldsymbol{c}$ from $\boldsymbol{c}_S$ by (4.1) and (4.2) if $\boldsymbol{c}_S$ is obtained from $\boldsymbol{u}_S$ by using the decoder D_C.

As a result, I propose a new decoding algorithm for cyclic codes over symbol-pair read channels as follows.

Proposed Decoding Algorithm for Cyclic Codes

Input: Received pair vector $\overleftrightarrow{\boldsymbol{u}} = (\boldsymbol{u}_L, \boldsymbol{u}_R)$, pair error correcting capability $t_p = \lfloor (d_p - 1)/2 \rfloor$.

Output: Corrected codeword $\hat{\boldsymbol{c}}$.

Step 1. Calculate the neighbor-symbol syndrome by (3.4).

Step 2. Decode $\boldsymbol{u}_L$ to $\boldsymbol{c}_L$ by the decoder D_C.

$$\boldsymbol{c}_L := D_C(\boldsymbol{u}_L).$$

If $\boldsymbol{c}_L = F$, go to Step 3; otherwise, calculate $\boldsymbol{e}_L$ and $\boldsymbol{e}_R$ as follows:

$$\begin{aligned} \boldsymbol{e}_L &:= \boldsymbol{c}_L \oplus \boldsymbol{u}_L, \\ \boldsymbol{e}_R &:= (\boldsymbol{e}_L \oplus \boldsymbol{S}) \lll 1. \end{aligned}$$

Examine the pair weight of $\overleftrightarrow{\boldsymbol{e}} = (\boldsymbol{e}_L, \boldsymbol{e}_R)$. If $W_p(\overleftrightarrow{\boldsymbol{e}}) \le t_p$, let $\hat{\boldsymbol{c}} := \boldsymbol{c}_L$. Then, output $\hat{\boldsymbol{c}}$ and terminate this algorithm; otherwise, go to Step 3.

Step 3. Decode $\boldsymbol{u}_R$ to $\boldsymbol{c}_R$ by the decoder D_C.

$$\boldsymbol{c}_R := D_C(\boldsymbol{u}_R).$$

If $\boldsymbol{c}_R = F$, go to Step 4; otherwise, calculate $\boldsymbol{e}_R$ and $\boldsymbol{e}_L$ as follows:

$$\begin{aligned} \boldsymbol{e}_R &:= \boldsymbol{c}_R \oplus \boldsymbol{u}_R, \\ \boldsymbol{e}_L &:= (\boldsymbol{e}_R \gg 1) \oplus \boldsymbol{S}. \end{aligned}$$

Examine the pair weight of $\overleftrightarrow{\boldsymbol{e}} = (\boldsymbol{e}_L, \boldsymbol{e}_R)$. If $W_p(\overleftrightarrow{\boldsymbol{e}}) \le t_p$, let $\hat{\boldsymbol{c}} := (\boldsymbol{c}_R \gg 1)$. Then, output $\hat{\boldsymbol{c}}$ and terminate this algorithm; otherwise, go to Step 4.

Step 4. Decode $\boldsymbol{u}_S$ to $\boldsymbol{c}'$ by the decoder D_C.

$$\boldsymbol{c}' := D_C(\boldsymbol{u}_S).$$

If $\boldsymbol{c}' = F$, then output the decoding failure; otherwise, calculate $\hat{\boldsymbol{c}}$ from $\boldsymbol{c}'$ by (4.1) and (4.2), and output $\hat{\boldsymbol{c}}$.

In Step 2 of the proposed algorithm, the decoder D_C attempts to correct the error from $\boldsymbol{u}_L$. If the error vector is in Case 1 or Case 2, $\boldsymbol{c}_L$ can be obtained by the result of the decoding operation $D_C(\boldsymbol{u}_L)$; then, the error vector can be obtained by $\boldsymbol{e}_L = \boldsymbol{c}_L \oplus \boldsymbol{u}_L$, and $\boldsymbol{e}_R$ can be calculated by (4.5). Then, the pair weight of $\overleftrightarrow{\boldsymbol{e}} = (\boldsymbol{e}_L, \boldsymbol{e}_R)$ is t_p or less. Therefore, these error vectors in Case 1 and Case 2 can be corrected in Step 2. If the error vector is in Case 3 or Case 4, the decoding operation $D_C(\boldsymbol{u}_L)$ fails and the decoder D_C outputs the failure symbol F or a miscorrected codeword $\boldsymbol{c}'_L$. Then, a different error vector is obtained by $\boldsymbol{e}'_L = \boldsymbol{c}'_L \oplus \boldsymbol{u}_L$, and a different error vector $\boldsymbol{e}'_R$ is calculated by (4.5). However, the pair weight of $\overleftrightarrow{\boldsymbol{e}}' = (\boldsymbol{e}'_L, \boldsymbol{e}'_R)$ is more than t_p; hence, $\overleftrightarrow{\boldsymbol{e}}'$ can be distinguished from $\overleftrightarrow{\boldsymbol{e}}$ and the process goes to Step 3. In Step 3, the decoder D_C attempts to correct the error from $\boldsymbol{u}_R$. If the error vector is in Case 3, $\boldsymbol{c}_R$ can be obtained by the result of the decoding operation $D_C(\boldsymbol{u}_R)$; then, the error vector can be obtained by $\boldsymbol{e}_R = \boldsymbol{c}_R \oplus \boldsymbol{u}_R$, and $\boldsymbol{e}_L$ can be calculated by (4.6). Then, the pair weight of $\overleftrightarrow{\boldsymbol{e}} = (\boldsymbol{e}_L, \boldsymbol{e}_R)$ is t_p or less. Therefore, these error vectors in Case 3 can be corrected in Step 3. If the error vector is in Case 4, the decoding operation $D_C(\boldsymbol{u}_R)$ fails and the decoder D_C outputs the failure symbol F or a miscorrected codeword $\boldsymbol{c}'_R$. Then, a different error vector is obtained by $\boldsymbol{e}'_R = \boldsymbol{c}'_R \oplus \boldsymbol{u}_R$, and a different error vector $\boldsymbol{e}'_L$ is calculated by (4.6). However, the pair weight of $\overleftrightarrow{\boldsymbol{e}}' = (\boldsymbol{e}'_L, \boldsymbol{e}'_R)$ is more than t_p; hence, $\overleftrightarrow{\boldsymbol{e}}'$ can be distinguished from $\overleftrightarrow{\boldsymbol{e}}$ and the process goes to Step 4. In Step 4, the decoder D_C attempt to correct the error from $\boldsymbol{u}_S$. The error is uncorrectable in Step 2 and Step 3. If $\boldsymbol{c}'$ is obtained by the decoding operation $D_C(\boldsymbol{u}_S)$, $\boldsymbol{c}$ can be calculated from $\boldsymbol{c}'$ using (4.1) and (4.2). If the decoding operation $D_C(\boldsymbol{u}_S)$ fails and the decoder D_C outputs the failure symbol F, the proposed decoding algorithm fails.

4.4 Pair Error Correcting Capability of this algorithm

I consider the pair error correcting capability of the proposed decoding algorithm. I assume to correct a pair error vector $\overleftrightarrow{\boldsymbol{e}} = (\boldsymbol{e}_L, \boldsymbol{e}_R)$. If $W_H(\boldsymbol{e}_L) \le t_H$ or $W_H(\boldsymbol{e}_R) \le t_H$, i.e., the vector is included in Case 1, Case 2, or Case 3, then this algorithm can correct up to t_p-pair errors in the vector in Step 2 or Step 3. If $W_H(\boldsymbol{e}_L) > t_H$ and $W_H(\boldsymbol{e}_R) > t_H$, i.e., the vector is included in Case 4, then this algorithm corrects pair errors in the vector in Step 4. Therefore, to analyze the error vectors corrected in Step 4, I have to consider only the error vectors included in Case 4.

Theorem 3. If both left and right vectors of the received word of a cyclic code over a symbol-pair read channel have more than t_H errors, the number of correctable pair errors by the t_H-error correcting decoder is $t'_0 = \lfloor (3t_H + 2)/2 \rfloor$.

Proof : Suppose that a received pair vector $\overleftrightarrow{\boldsymbol{u}}$ has t-pair errors. Let E_S be the number of pair errors $(1, 1)$ in $\overleftrightarrow{\boldsymbol{u}}$, E_L be the number of pair errors $(1, 0)$ in $\overleftrightarrow{\boldsymbol{u}}$, and E_R be the number of pair errors $(0, 1)$ in $\overleftrightarrow{\boldsymbol{u}}$. Thus,

$$t = E_S + E_L + E_R. \tag{4.7}$$

First, I consider the number of errors in $\boldsymbol{u}_S$ which is the vector XORing the left and right vectors of $\overleftrightarrow{\boldsymbol{u}}$ since $\boldsymbol{u}_S$ is corrected in Step 4. The XOR result of the left and right symbols of the pair error $(1, 1)$ in $\overleftrightarrow{\boldsymbol{u}}$ is equal to 0 in $\boldsymbol{u}_S$, and the XOR result of the left and right symbols of the pair error $(1, 0)$ and $(0, 1)$ in $\overleftrightarrow{\boldsymbol{u}}$ is equal to 1 in $\boldsymbol{u}_S$. Then, the total number of errors in $\boldsymbol{u}_S$ is $E_L + E_R$.

Next, I consider the upper bound on $E_L + E_R$ under the condition that $\overleftrightarrow{\boldsymbol{u}}$ is included in Case 4. Since $\overleftrightarrow{\boldsymbol{u}}$ is included in Case 4, the total number of errors in the left vector satisfies

$$E_S + E_L \geq t_H + 1. \tag{4.8}$$

Similarly, the total number of errors in the right vector satisfies

$$E_S + E_R \geq t_H + 1. \tag{4.9}$$

From (4.8) and (4.9),

$$2E_S + E_L + E_R \geq 2t_H + 2. \tag{4.10}$$

From (4.7), the left-hand side of (4.10) is written as

$$\begin{aligned} 2E_S + E_L + E_R &= 2(t - E_L - E_R) + E_L + E_R \\ &= 2t - E_L - E_R. \end{aligned} \tag{4.11}$$

Thus, (4.10) is written as

$$2t - E_L - E_R \geq 2t_H + 2. \tag{4.12}$$

Therefore, the upper bound on $E_L + E_R$ is given by

$$E_L + E_R \leq 2t - 2t_H - 2. \tag{4.13}$$

Finally, I consider the number of correctable pair errors. The maximum value of $E_L + E_R$ is $2t - E_L - E_R$ from (4.13); hence, $\boldsymbol{u}_S$, which has $2t - E_L - E_R$ errors, can be corrected by the t_H-errors-correcting decoder under the following condition:

$$2t - 2t_H - 2 \leq t_H. \tag{4.14}$$

From (4.14), $t \leq \lfloor (3t_H + 2)/2 \rfloor$ is obtained. Therefore, the number of correctable pair errors is $t'_0 = \lfloor (3t_H + 2)/2 \rfloor$. □

From Theorem 1 and Theorem 3, the pair error correcting capability of the proposed decoding algorithm is

$$t'_p = \min\left(\left\lfloor \frac{d_p - 1}{2} \right\rfloor, \left\lfloor \frac{3t_H + 2}{2} \right\rfloor \right). \tag{4.15}$$

Table 4.1: Comparison of the pair error correcting capability of binary (n, k) BCH codes over symbol-pair read channels

Code					Proposed algorithm	Yaakobi's algorithm
(n, k)	d_H	d_p	t_H	t_p	Pair error correcting capability t'_p	Pair error correcting capability t_0
(31,21)	5	9	2	4	4	3
(127,64)	21	34	10	16	16	15
(127,85)	13	22	6	10	10	9
(127,99)	9	15	4	7	7	6
(255,223)	9	15	4	7	7	6

4.5 Discussion

First, I compare the error correcting capability of the proposed algorithm with that of existing algorithms. Yaakobi et al. presented a decoding algorithm that can correct up to t_0-pair errors, where t_0 is given by (2.19). Their algorithm cannot correct all pair errors in a code that has the error correcting capability t_p, if t_p is greater than t_0. Table 4.1 lists examples of such codes as binary primitive BCH codes. The minimum pair distance d_p is calculated by the algorithm presented in Ref. [9]. For these codes, the proposed algorithm can correct the pair error vectors that cannot be corrected by Yaakobi's algorithm. I present an example of such a pair error vector.

Example 4. Suppose the binary $(31, 21)$ BCH code whose generator polynomial is $g(x) = x^{10} + x^9 + x^8 + x^6 + x^5 + x^3 + 1$. Yaakobi's decoding algorithm can correct up to $t_0 = 3$ pair errors, and the proposed decoding algorithm can correct up to $t'_p = 4$ pair errors. Let

$$\overleftrightarrow{\boldsymbol{u}} = [(1,1),(0,0),(0,0),(0,0),(1,0),(0,0),(1,0),(1,0),\\ (0,0),(0,0),\ldots,(0,0)]$$

be a received pair vector that has 4 pair errors, when the all-zero codeword [00000000 00000000 00000000 0000000] is read. Then, the left vector, right vector, XORing vector, and neighbor-symbol syndrome are as follows:

$$\begin{aligned}
\boldsymbol{u}_L &= [10001011\ 00000000\ 00000000\ 0000000],\\
\boldsymbol{u}_R &= [10000000\ 00000000\ 00000000\ 0000000],\\
\boldsymbol{u}_S &= [00001011\ 00000000\ 00000000\ 0000000],\\
\boldsymbol{S} &= [11001011\ 00000000\ 00000000\ 0000000].
\end{aligned}$$

I consider correcting $\overleftrightarrow{\boldsymbol{u}}$ by Yaakobi's algorithm and the proposed algorithm using the t_H-error correcting decoder for cyclic codes D_C, and the error/erasure decoder D_{C_2}, which indicates the pairs of received pair vector in conflict.
Yaakobi's Decoding Algorithm: In Step 1, $\boldsymbol{u}_S$ is input to the decoder D_C. However, the decoder D_C outputs the failure symbol F or a different vector from $\boldsymbol{c}'$, since the number of errors in $\boldsymbol{u}_S$ is 3. In Step 2, $\overleftrightarrow{\boldsymbol{u}}$ is input the error/erasure decoder D_{C_2}. However, the decoder D_{C_2} outputs the failure symbol F or a different vector from $\boldsymbol{c}$, since the number of erasures in $\overleftrightarrow{\boldsymbol{u}}$ is 5. In Step 3 and later, the error vector is calculated

from the obtained vectors in Step 1 and Step 2. However, the all-zero codeword cannot be obtained since the corrections are not successful in Step 1 and Step 2.
Proposed Decoding Algorithm for Cyclic Codes: In Step 2, $\boldsymbol{u}_L$ is input to the decoder D_C. The decoder D_C outputs $\boldsymbol{c}_L = [10001011\ 00000000\ 00000001\ 0000000]$. Then, $\boldsymbol{e}_L$ and $\boldsymbol{e}_R$ are calculated as follows:

$$\begin{aligned}\boldsymbol{e}_L &= \boldsymbol{u}_L \oplus \boldsymbol{c}_L \\ &= [00000000\ 00000000\ 00000001\ 0000000], \\ \boldsymbol{e}_R &= (\boldsymbol{S} \oplus \boldsymbol{e}_L) \ll 1 \\ &= [10010110\ 00000000\ 00000010\ 0000001].\end{aligned}$$

Examining the pair weight of $\overleftrightarrow{\boldsymbol{e}} = (\boldsymbol{e}_L, \boldsymbol{e}_R)$, this algorithm confirms that $W_p(\overleftrightarrow{\boldsymbol{e}}) = 7 > t_p$. Thus, the proposed decoding algorithm performs Step 3. In Step 3, $\boldsymbol{u}_R$ is input to the decoder D_C. The decoder D_C outputs $\boldsymbol{c}_R = [00000000\ 00000000\ 00000000\ 0000000]$. Then, $\boldsymbol{e}_R$ and $\boldsymbol{e}_L$ are calculated as follows:

$$\begin{aligned}\boldsymbol{e}_R &= \boldsymbol{u}_R \oplus \boldsymbol{c}_R \\ &= [10000000\ 00000000\ 00000000\ 0000000], \\ \boldsymbol{e}_L &= \boldsymbol{S} \oplus (\boldsymbol{e}_R \gg 1) \\ &= [10001011\ 00000000\ 00000000\ 0000000].\end{aligned}$$

Examining the pair weight of $\overleftrightarrow{\boldsymbol{e}} = (\boldsymbol{e}_L, \boldsymbol{e}_R)$, this algorithm confirm that $W_p(\overleftrightarrow{\boldsymbol{e}}) = 4 \leq t_p$. Thus, the proposed algorithm outputs $\hat{\boldsymbol{c}} = \boldsymbol{e}_R \gg 1 = [000 \cdots 0]$. Therefore, this error correction is successful. ■

Next, I compare the number of correctable pair error vectors of the proposed algorithm with that of Yaakobi's algorithm. I consider binary (n, k) cyclic codes over symbol-pair read channels. Then, the total number of error vectors that have at most t pair errors is given by

$$N = \sum_{i=1}^{t} \binom{n}{i} 3^i. \tag{4.16}$$

In the case of the binary $(31, 21)$ BCH code,

125,643 pair error vectors can be corrected by Yaakobi's algorithm since it can correct 3-pair errors, and 2,684,308 pair error vectors can be corrected by the proposed algorithm since it can correct 4-pair errors. Furthermore, I consider how many correctable pair error vectors are included in each case when these vectors are decoded by the proposed algorithm. For an (n, k) code, the number of pair error vectors included in each case is given as follows.
Case 1: In this case, both left and right vectors have at most t_H errors. Then, all pair error vectors that have at most t_H-pair errors are included in Case 1. In addition, I consider $(t_H + i)$ pair error vectors. If the number of both pair errors $(1, 0)$ and $(0, 1)$ is at least i, the pair error vectors are included in Case 1. Thus, the number of correctable pair error vectors included in Case 1 is given by

$$\begin{aligned}N_1 = &\sum_{i=1}^{t_H} \binom{n}{i} 3^i \\ &+ \sum_{i=1}^{t'_p - t_H} \binom{n}{t_H + i} \sum_{j=0}^{t_H - i} \sum_{k=0}^{t_H - i - j} \frac{(t_H + i)!}{(i+j)!(i+k)!(t_H - i - j - k)!}.\end{aligned} \tag{4.17}$$

Table 4.2: Number of correctable pair error vectors

	$(31, 21)$ BCH code	$(127, 99)$ BCH code
Case 1	247,008 (188,790)	20,155,493,860,890 (18,764,847,312,750)
Case 2	881,020 (849,555)	66,249,786,102,525 (65,140,827,099,975)
Case 3	881,020 (849,555)	66,249,786,102,525 (65,140,827,099,975)
Case 4	665,260 (660,765)	46,598,517,334,275 (46,375,979,787,225)
Total	2,674,308 (2,548,665)	199,253,583,400,215 (195,422,481,299,925)

The first term is the number of pair error vectors that have at most t_H-pair errors. In the second term, $\binom{n}{t_H+i}$ is the number of patterns of pair error positions, and the fraction represents the permutations of pair errors $(1,0)$, $(0,1)$, and $(1,1)$ in pair error positions with repetitions permitted. For the numerator of the fraction, $i+j$ is the number of $(0,1)$, $i+k$ is the number of $(0,1)$, and $t_H - i - j - k$ is the number of $(1,1)$.
Case 4 : In this case, both left and right vectors have more than t_H errors. I consider $t_H + i$ pair error vectors. If the number of pair errors $(1,1)$ is $j \geq t_H - i + 2$ and the number of both pair errors $(1,0)$ and $(0,1)$ is at least l, where

$$l = \begin{cases} t_H + 1 - j & \text{if } j < t_H + 1 \\ 0 & \text{otherwise} \end{cases}, \tag{4.18}$$

then pair error vectors are included in Case 4. Thus, the number of pair error vectors included in Case 4 is given by

$$N_4 = \sum_{i=1}^{t'_p - t_H} \binom{n}{t_H + i} \sum_{j=t_H-i+2}^{t_H+i} \sum_{k=0}^{t_H+i-j-2l} \frac{(t_H+i)!}{(l+k)!(t_H+i-j-l-k)!j!}. \tag{4.19}$$

As in (4.17), $\binom{n}{t_H+i}$ is the number of patterns of pair error positions, and the fraction represents the permutations of pair errors $(1,0)$, $(0,1)$, and $(1,1)$ in pair error positions with repetitions permitted. For the numerator of the fraction, $l+k$ is the number of $(0,1)$, $t_H + i - j - l - k$ is the number of $(0,1)$, and j is the number of $(1,1)$.
Case 2 and Case 3: In these cases, either the left or the right vector has at most t_H errors and the other vector has more than t_H errors. The numbers of pair error vectors included in Case 2 and Case 3 are equal since the two cases are symmetrical about the number of errors in the left and right vectors. Thus, the number of pair error vectors included in Case 2 and Case 3 is given by

$$N_2 = N_3 = \frac{N - N_1 - N_4}{2}. \tag{4.20}$$

Table 4.2 lists the number of pair error vectors that can be corrected by the proposed algorithm with the $(31, 21)$ BCH code and the $(127, 99)$ BCH code. In Table 4.2, the numbers in parentheses are the numbers of pair error vectors that cannot be corrected by Yaakobi's algorithm. Such pair error vectors have t'_p pair errors; hence, the number of pair error vectors are given by

$$N' = \binom{n}{t'_p} 3^{t'_p}, \tag{4.21}$$

$$N'_1 = \binom{n}{t'_p} \sum_{j=0}^{t'_p} \sum_{k=0}^{t'_p-j} \frac{t'_p!}{(t'_p - t_H + j)!(t'_p - t_H + k)!(t'_p - j - k)!}, \tag{4.22}$$

$$N'_4 = \binom{n}{t'_p} \sum_{j=2t_H-t'_p+2}^{t'_p} \sum_{k=0}^{t'_p-j-2l} \frac{t'_p!}{(l+k)!(t'_p - j - l - k)!j!}, \tag{4.23}$$

$$N'_2 = N'_3 = \frac{N' - N'_1 - N'_4}{2}. \tag{4.24}$$

The pair error vectors included in Case 1 and Case 2 are corrected in Step 2 of the proposed algorithm. Next, the pair error vectors included in Case 3 are corrected in Step 3, and then, the pair error vectors included in Case 4 are corrected in Step 4.

Finally, I compare the complexity of the proposed algorithm with that of existing algorithms. The proposed decoding algorithm uses decoders for cyclic codes, and it does not need a decoding table such as that in the case of syndrome decoding proposed in Chapter 3. Therefore, the pair errors can be corrected efficiently by decoding algorithms such as the Euclidean algorithm and Berlekamp-Massey algorithm. In Yaakobi's algorithm, the decoding algorithm for cyclic codes is executed two times. On the other hand, in the proposed algorithm, the decoding algorithm for cyclic codes is executed three times at most. However, if the proposed algorithm is terminated in Step 2, it is faster than Yaakobi's algorithm since the decoding algorithm for cyclic codes is executed only one time.

4.6 Conclusion

In this chapter, I have proposed a new decoding algorithm for cyclic codes over symbol-pair read channels using a decoder for cyclic codes. The proposed algorithm corrects pair errors on the basis of the relationship between the pair errors and the syndromes. In addition, the proposed algorithm can correct pair errors that cannot be corrected by Yaakobi's algorithm since the proposed algorithm corrects pair errors in the pair metric. Furthermore, I compared the error correcting capability and complexity of the proposed algorithm with those of existing algorithms. The results showed that the proposed algorithm can correct more pair errors than Yaakobi's algorithm.

Chapter 5

Algebraic Decoding of BCH Codes over Symbol-Pair Read Channels

5.1 Abstract

In this chapter, I discuss an algebraic decoding of BCH codes over symbol-pair read channels. I define a polynomial that represents the positions of the pair errors as an error-locator polynomial and also define a polynomial that represents the positions of the pairs of a received pair vector in conflict as a conflict-locator polynomial. I propose algebraic methods for correcting two-pair and three-pair errors for BCH codes. First, I show the relation between the error-locator polynomial and the conflict-locator polynomial. Second, I show the relation among these polynomials and the syndromes. Finally, I provide how to correct the pair errors by solving equations including the relational expression by algebraic methods.

5.2 Polynomial Representation of Syndromes for Codes over Symbol-Pair Read Channels

In this work, I consider binary BCH codes. Let α be a primitive element of GF(2^m). The t_H-error-correcting BCH codes whose length $n = 2^m - 1$ have $\alpha, \alpha^2, \ldots, \alpha^{2t_H}$ as their roots. Let $\mathbf{H}$ be a parity-check matrix of the BCH codes. A symbol-pair parity-check matrix $\pi(\mathbf{H})$ is given by representing each row of $\mathbf{H}$ as the symbol-pair vector, so $\pi(\mathbf{H})$ of a BCH code is given by

$$\begin{aligned}&\pi(\mathbf{H})\\ &= \begin{bmatrix} (1,\alpha) & (\alpha,\alpha^2) & \cdots & (\alpha^{n-2},\alpha^{n-1}) & (\alpha^{n-1},1) \\ (1,\alpha^2) & (\alpha^2,\alpha^4) & \cdots & (\alpha^{2(n-2)},\alpha^{2(n-1)}) & (\alpha^{2(n-1)},1) \\ \vdots & \vdots & & \vdots & \vdots \\ (1,\alpha^{2t_H}) & (\alpha^{2t_H},\alpha^{4t_H}) & \cdots & (\alpha^{2t_H(n-2)},\alpha^{2t_H(n-1)}) & (\alpha^{2t_H(n-1)},1) \end{bmatrix}.\end{aligned}$$

Then, the symbol-pair syndrome is calculated by the received pair vector as follows:

$$\begin{aligned}\overleftrightarrow{\boldsymbol{s}} &= [(s_{l,1}, s_{r,1}), (s_{l,2}, s_{r,2}), \ldots, (s_{l,2t_H}, s_{r,2t_H})]\\ &\triangleq \overleftrightarrow{u} \cdot \pi(\mathbf{H})^{\mathrm{T}}.\end{aligned}$$

The symbol-pair syndrome also can be calculated by the received polynomials. When the received pair vector is given by $\overleftrightarrow{\boldsymbol{u}} = [(u_{l,0}, u_{r,1}), (u_{l,1}, u_{r,2}), \ldots, (u_{l,n-1}, u_{r,0})]$, the symbol-pair syndrome is calculated as follows:

$$\begin{aligned}
s_{l,1} &= 1 \cdot u_{l,0} + \alpha \cdot u_{l,1} + \cdots + \alpha^{n-2} \cdot u_{l,n-2} + \alpha^{n-1} \cdot u_{l,n-1}, \\
s_{r,1} &= \alpha \cdot u_{r,1} + \alpha^{2} \cdot u_{r,2} + \cdots + \alpha^{n-1} \cdot u_{r,n-1} + 1 \cdot u_{r,0}, \\
s_{l,2} &= 1 \cdot u_{l,0} + \alpha^{2} \cdot u_{l,1} + \cdots + \alpha^{2(n-2)} \cdot u_{l,n-2} + \alpha^{2(n-1)} \cdot u_{l,n-1}, \\
s_{r,2} &= \alpha^{2} \cdot u_{r,1} + \alpha^{4} \cdot u_{r,2} + \cdots + \alpha^{2(n-1)} \cdot u_{r,n-1} + 1 \cdot u_{r,0}, \\
&\vdots \\
s_{l,2t_H} &= 1 \cdot u_{l,0} + \alpha^{2t_H} \cdot u_{l,1} + \cdots + \alpha^{2t_H(n-2)} \cdot u_{l,n-2} + \alpha^{2t_H(n-1)} \cdot u_{l,n-1}, \\
s_{r,2t_H} &= \alpha^{2t_H} \cdot u_{r,1} + \alpha^{4t_H} \cdot u_{r,2} + \cdots + \alpha^{2t_H(n-1)} \cdot u_{r,n-1} + 1 \cdot u_{r,0}.
\end{aligned}$$

Here, I define the left and right received polynomials as follows:

$$u_L(x) = u_{l,0} + u_{l,1}x + \cdots + u_{l,n-1}x^{n-1}, \tag{5.1}$$

$$u_R(x) = u_{r,1} + u_{r,2}x + \cdots + u_{r,0}x^{n-1}. \tag{5.2}$$

Then, the symbol-pair syndrome is calculated as follows:

$$s_{l,i} = u_L(\alpha^i), \quad i = 1, 2, \ldots, 2t_H, \tag{5.3}$$

$$s_{r,i} = \alpha^i u_R(\alpha^i), \quad i = 1, 2, \ldots, 2t_H. \tag{5.4}$$

The neighbor-symbol syndrome, which indicates the pairs of the received pair vectors in conflict, is also represented by using the received polynomial. In Chapter 3, this syndrome is defined as follows:

$$\boldsymbol{S} \triangleq [S_0, S_1, \ldots, S_{n-1}],$$

$$S_i = \begin{cases} 0 & \text{if } u_{l,i} = u_{r,i} \\ 1 & \text{otherwise} \end{cases}.$$

The neighbor-symbol syndrome is calculated by using the left and right received polynomials as

$$S(x) = u_L(x) \oplus (xu_R(x) \bmod x^n - 1). \tag{5.5}$$

In Chapter 3, I have shown the relation between these two types of syndromes and the correctable pair errors.

Theorem 2 (Reproduce)**.** If a code C can correct t_p-pair errors, the pair consisting of the symbol-pair and neighbor-symbol syndromes $(\overleftrightarrow{\boldsymbol{s}}, \boldsymbol{S})$ is unique for each error vector $\overleftrightarrow{\boldsymbol{e}}$, where $W_p(\overleftrightarrow{\boldsymbol{e}}) \leq t_p$.

5.3 Error-Locator and Conflict-Locator Polynomials

First, I define the *error-locator polynomial* that represent the positions of the pair error. Suppose that the pair error polynomial $\overleftrightarrow{e}(x)$ has t errors at the positions $0 \leq p_1 < p_2 < \cdots < p_t \leq n$. The error value $\overleftrightarrow{e}_{p_i} = (e_{l,p_i}, e_{r,p_i+1})$ of the position p_i is either $(1,1)$, $(1,0)$, or $(0,1)$, where $i = 1, 2, \ldots, t$. Then, the pair error polynomial is defined as

$$\overleftrightarrow{e}(x) = \overleftrightarrow{e}_{p_1}x^{p_1} + \overleftrightarrow{e}_{p_2}x^{p_2} + \cdots + \overleftrightarrow{e}_{p_t}x^{p_t}. \tag{5.6}$$

Moreover, the left and right error polynomials are defined as follows:

$$e_L(x) = e_{l,p_1}x^{p_1} + e_{l,p_2}x^{p_2} + \cdots + e_{l,p_t}x^{p_t}, \tag{5.7}$$

$$e_R(x) = e_{r,p_1+1}x^{p_1} + e_{r,p_2+1}x^{p_2} + \cdots + e_{r,p_t+1}x^{p_t}. \tag{5.8}$$

Note that the factors of $e_R(x)$ are one bit ahead of the factors of $e_L(x)$. Then, the relations between the error polynomials and the two types of syndromes are as follows:

$$\begin{aligned} s_{l,i} &= e_L(\alpha^i), \quad i = 1, 2, \ldots, 2t_H, \\ s_{r,i} &= \alpha^i e_R(\alpha^i), \quad i = 1, 2, \ldots, 2t_H, \\ S(x) &= e_L(x) \oplus (x e_R(x) \bmod x^n - 1). \end{aligned}$$

Because it is difficult to determine the error positions and error values directly from the syndromes, I consider finding a polynomial that represents the positions of a pair error. Suppose that pair errors occur at the positions $0 \le p_1 < p_2 < \cdots < p_t \le n$; I define the left and right error-locator polynomials as follows:

$$\sigma_L(x) = \sum_{i=0}^{E_L} \sigma_{l,i}x^i \triangleq \prod_{\substack{i=0 \\ e_{l,i} \neq 0}}^{n-1} (1 - \alpha^i x), \tag{5.9}$$

$$\sigma_R(x) = \sum_{i=0}^{E_R} \sigma_{r,i}x^i \triangleq \prod_{\substack{i=0 \\ e_{r,i+1} \neq 0}}^{n-1} (1 - \alpha^{i+1} x), \tag{5.10}$$

where E_L and E_R denote the number of errors in the left and right symbols of the pair error vector, respectively. Since the factors of $e_R(x)$ are one bit ahead of the factors of $e_L(x)$, the exponent of α in (5.10) is one-bit ahead of the exponent of α in (5.9).

In the decoding algorithm, if the left and right error-locator polynomials are obtained, the pair error polynomial is found as follows. From (5.9), $e_{l,i} \neq 0$ if a root of $\sigma_L(x)$ is α^{n-i}; therefore, $e_{l,i} = 1$ in (5.7). Moreover, from (5.10), $e_{r,i+1} \neq 0$ if a root of $\sigma_R(x)$ is $\alpha^{n-(i+1)}$; therefore, $e_{r,i+1} = 1$ in (5.8). Thus, the left and right error polynomials $e_L(x)$ and $e_R(x)$ are obtained by finding the roots of $\sigma_L(x)$ and $\sigma_R(x)$, and the pair error polynomial $\overleftrightarrow{e}(x)$ is found from (5.6).

Next, I define *conflict-locator polynomial* that represents the conflict positions. The neighbor-symbol syndrome shows the conflict positions in that either $u_{r,i}$ or $u_{l,i}$ in two adjacent pairs suffers from errors. Thus, I define the conflict-locator polynomial by using the neighbor-symbol syndrome $\boldsymbol{S}$ as

$$\tau(x) = \sum_{i=0}^{W_H(\boldsymbol{S})} \tau_i x^i \triangleq \prod_{\substack{i=0 \\ S_i \neq 0}}^{n-1} (1 - \alpha^i x), \tag{5.11}$$

where $W_H(\boldsymbol{S})$ is the Hamming weight of $\boldsymbol{S}$.

Consider that both $u_{r,i}$ and $u_{l,i}$ suffer from errors. The positions of such errors do not appear in the neighbor-symbol syndrome. To correct pair errors, it is necessary to consider that such errors occur. If the received pair vector suffers from t-pair errors, I consider the number of positions at which both $u_{r,i}$ and $u_{l,i}$ suffer from errors.

Theorem 4. Let t' be the number of positions at which both $u_{r,i}$ and $u_{l,i}$ suffer from errors. Then, the relation among t', E_L, E_R, and $W_H(\boldsymbol{S})$ is

$$W_H(\boldsymbol{S}) = E_L + E_R - 2t'. \tag{5.12}$$

Proof: If all error positions in the pair error vector are represented by the neighbor-symbol syndrome, then $W_H(\boldsymbol{S}) = E_L + E_R$. If both $u_{r,i}$ and $u_{l,i}$ suffer from errors, such positions are not represented by the neighbor-symbol syndrome. Then, $W_H(\boldsymbol{S})$ decreases by 2. Thus, if the number of such positions is t', $W_H(\boldsymbol{S}) = E_L + E_R - 2t'$. $\square$

From Theorem 4, t' is obtained by

$$t' = \frac{E_L + E_R - W_H(\boldsymbol{S})}{2}, \tag{5.13}$$

where $W_H(\boldsymbol{S})$ is obtained by the received pair vector. Then, t' satisfies the following inequalities:

$$t' \leq E_L, \quad t' \leq E_R, \quad t' \leq \left\lfloor \frac{E_L + E_R}{2} \right\rfloor, \quad t' < t. \tag{5.14}$$

I show the relation between the left and right error-locator polynomials and the conflict-locator polynomial.

Theorem 5. Let $k_1, k_2, \ldots, k_{t'}$ be the positions at which both $u_{r,i}$ and $u_{l,i}$ suffer from errors. Then, the left and right error-locator polynomials $\sigma_L(x)$ and $\sigma_R(x)$ and the conflict-locator polynomial $\tau(x)$ satisfy

$$\sigma_L(x)\sigma_R(x) = \tau(x) \prod_{i=0}^{t'} (1 - \alpha^{k_i} x)^2. \tag{5.15}$$

Proof: All error positions in the pair error vector are represented by the multiplication of $\sigma_L(X)$ and $\sigma_R(x)$ as

$$\sigma_L(x)\sigma_R(x) = \prod_{\substack{i=0 \\ e_{l,i} \neq 0}}^{n-1} (1 - \alpha^i x) \prod_{\substack{i=0 \\ e_{r,i+1} \neq 0}}^{n-1} (1 - \alpha^{i+1} x).$$

If both $u_{r,i}$ and $u_{l,i}$ suffer from errors, $\sigma_L(x)$ and $\sigma_R(x)$ have common factors. These factors are represented by $\gcd(\sigma_L(x), \sigma_R(x))$, which denotes the greatest common divisor of polynomials of $\sigma_L(x)$ and $\sigma_R(x)$. Then, $u_{r,i}$ and $u_{l,i}$ do not appear as the conflict positions. Let k_i be such positions; then, $\gcd(\sigma_L(x), \sigma_R(x))$ is written as follows:

$$\gcd(\sigma_L(x), \sigma_R(x)) = \prod_{\substack{i=0 \\ e_{r,i} \neq 0 \,\wedge\, e_{l,i} \neq 0}}^{n-1} (1 - \alpha^i x) \triangleq \prod_{i=0}^{t'} (1 - \alpha^{k_i} x). \tag{5.16}$$

If either $u_{r,i}$ or $u_{l,i}$ suffers from errors, such positions are represented as $\tau(x)$. Thus, all error positions in the pair error vectors are represented by using (5.16) and $\tau(x)$; therefore,

$$\prod_{\substack{i=0 \\ e_{l,i} \neq 0}}^{n-1} (1 - \alpha^i x) \prod_{\substack{i=0 \\ e_{r,i+1} \neq 0}}^{n-1} (1 - \alpha^{i+1} x) = \tau(x) \prod_{i=0}^{t'} (1 - \alpha^{k_i} x)^2.$$

Thus, $\sigma_L(x)\sigma_R(x) = \tau(x) \prod_{i=0}^{t'} (1 - \alpha^{k_i} x)^2$. $\square$

Table 5.1: Classification of the pair error vectors

		$E_L + E_R$	(E_L, E_R)
Two-pair error	Case I	4	(2,2)
	Case Ⅱ	3	(2,1), (1,2)
	Case Ⅲ	2	(2,0), (0,2), (1,1)
Single-pair error	Case Ⅳ	2	(1,1)
	Case V	1	(1,0), (0,1)

5.4 Decoding Problem for Two-Pair Error Correction

In this section, I consider a decoding problem for two-pair error correction using single-error-correcting cyclic Hamming codes.

5.4.1 Classification of the Pair Error Vector

I classify pair error vectors according to $E_L + E_R$, as indicated in Table 5.1. In each case, I consider (5.15) when $W_H(\boldsymbol{S})$ is obtained.
Case I: In the case of $(E_L, E_R) = (2,2)$, (5.14) holds, $t' = 0$, and $t' = 1$. From (5.12), $W_H(\boldsymbol{S}) = 4$ if $t' = 0$, and $W_H(\boldsymbol{S}) = 2$ if $t' = 1$. Then, (5.15) is written as follows:

$$\sigma_L(x)\sigma_R(x) = \begin{cases} \tau(x) & \text{if } W_H(\boldsymbol{S}) = 4 \\ \tau(x)(1-\alpha^{k_1}x)^2 & \text{if } W_H(\boldsymbol{S}) = 2 \end{cases}. \tag{5.17}$$

Case Ⅱ: In the case of $(E_L, E_R) = (2,1)$, (5.14) holds, $t' = 0$, and $t' = 1$. From (5.12), $W_H(\boldsymbol{S}) = 3$ if $t' = 0$, and $W_H(\boldsymbol{S}) = 1$ if $t' = 1$. Then, (5.15) is written as follows:

$$\sigma_L(x)\sigma_R(x) = \begin{cases} \tau(x) & \text{if } W_H(\boldsymbol{S}) = 3 \\ \tau(x)(1-\alpha^{k_1}x)^2 & \text{if } W_H(\boldsymbol{S}) = 1 \end{cases}. \tag{5.18}$$

Similarly, in the case of $(E_L, E_R) = (1,2)$, (5.15) is written as (5.18).
Case Ⅲ: In the case of $(E_L, E_R) = (2,0)$, (5.14) holds, and $t' = 0$. If $t' = 0$, $W_H(\boldsymbol{S}) = 2$ from (5.12); then, (5.15) is written as follows:

$$\sigma_L(x)\sigma_R(x) = \tau(x). \tag{5.19}$$

Similarly, in the case of $(E_L, E_R) = (0,2)$, (5.15) is written as (5.19).

In the case of $(E_L, E_R) = (1,1)$, (5.14) holds, $t' = 0$, and $t' = 1$. From (5.12), $W_H(\boldsymbol{S}) = 2$ if $t' = 0$, and $W_H(\boldsymbol{S}) = 0$ if $t' = 1$. Then, (5.15) is written as follows:

$$\sigma_L(x)\sigma_R(x) = \begin{cases} \tau(x) & \text{if } W_H(\boldsymbol{S}) = 2 \\ (1-\alpha^{k_1}x)^2 & \text{if } W_H(\boldsymbol{S}) = 0 \end{cases}. \tag{5.20}$$

Case Ⅳ: In the case of $(E_L, E_R) = (1,1)$, (5.14) holds, and $t' = 0$. If $t' = 0$, $W_H(\boldsymbol{S}) = 2$ from (5.12); then, (5.15) is written as follows:

$$\sigma_L(x)\sigma_R(x) = \tau(x). \tag{5.21}$$

Case V: In the case of $(E_L, E_R) = (1,0)$, (5.14) holds, and $t' = 0$. If $t' = 0$, $W_H(\boldsymbol{S}) = 1$ from (5.12); then, (5.15) is written as follows:

$$\sigma_L(x)\sigma_R(x) = \tau(x). \tag{5.22}$$

Similarly, in the case of $(E_L, E_R) = (0,1)$, (5.15) is written as (5.22).

5.4.2 Relation between the Error-Locator Polynomials and the Symbol-Pair Syndrome

For any t-pair error, I derive the following theorem.

Theorem 6. The coefficients of the first-order terms of the left and right error-locator polynomials $\sigma_{l,1}$ and $\sigma_{r,1}$ and the symbol-pair syndromes $s_{l,1}$ and $s_{r,1}$ satisfy the following relations:

$$\sigma_{l,1} = s_{l,1}, \tag{5.23}$$
$$\sigma_{r,1} = s_{r,1}. \tag{5.24}$$

Proof: When the left and right error polynomials are given by (5.7) and (5.8), the left and right error-locator polynomials are given by (5.9) and (5.10). Then, $\sigma_{l,1}$ and $\sigma_{r,1}$ are

$$\begin{aligned}\sigma_{l,1} &= e_{l,p_1}\alpha^{p_1} + e_{l,p_2}\alpha^{p_2} + \cdots + e_{l,p_t}\alpha^{p_t},\\ \sigma_{r,1} &= e_{r,p_1+1}\alpha^{p_1+1} + e_{r,p_2+1}\alpha^{p_2+1} + \cdots + e_{r,p_t+1}\alpha^{p_t+1}.\end{aligned}$$

In addition, $s_{l,1}$ and $s_{r,1}$ are

$$\begin{aligned}s_{l,1} &= e_{l,p_1}\alpha^{p_1} + e_{l,p_2}\alpha^{p_2} + \cdots + e_{l,p_t}\alpha^{p_t},\\ s_{r,1} &= e_{r,p_1+1}\alpha^{p_1+1} + e_{r,p_2+1}\alpha^{p_2+1} + \cdots + e_{r,p_t+1}\alpha^{p_t+1}.\end{aligned}$$

Thus, $\sigma_{l,1} = s_{l,1}$ and $\sigma_{r,1} = s_{r,1}$. □

If $E_L = E_R$, for any t-pair error, I derive the following theorem.

Theorem 7. The coefficients of the i-th order terms of $\sigma_L(x)$ and $\sigma_R(x)$ are represented as $\sigma_{l,i}$ and $\sigma_{r,i}$. If $E_L = E_R$, $\sigma_{r,i} = \alpha^i\sigma_{l,i}$.

Proof: Since $E_L = E_R$, the left and right error polynomials are $e_L(x) = x^{p_1} + x^{p_2} + \cdots + x^{p_t}$ and $e_R(x) = x^{p_1} + x^{p_2} + \cdots + x^{p_t}$. From (5.9) and (5.10), the left and right error-locator polynomials are written as follows:

$$\begin{aligned}\sigma_L(x) =&(1-\alpha^{p_1}x)(1-\alpha^{p_2}x)\ldots(1-\alpha^{p_t}x),\\ \sigma_R(x) =&(1-\alpha^{p_1+1}x)(1-\alpha^{p_2+1}x)\ldots(1-\alpha^{p_t+1}x).\end{aligned}$$

Then, the coefficients of $\sigma_L(x)$ are written as follows:

$$\begin{aligned}\sigma_{l,1} &= \alpha^{p_1} + \alpha^{p_2} + \cdots + \alpha^{p_t},\\ \sigma_{l,2} &= \alpha^{p_1+p_2} + \alpha^{p_1+p_3} + \alpha^{p_2+p_3} + \cdots + \alpha^{p_{t-2}+p_t} + \alpha^{p_{t-1}+p_t},\\ &\vdots\\ \sigma_{l,i} &= \alpha^{p_1+\cdots+p_i} + \alpha^{p_1+\cdots+p_{i-1}+p_{i+1}} + \alpha^{p_1+\cdots+p_{i-2}+p_i+p_{i+1}} + \cdots\\ &\quad + \alpha^{p_{t-i}+p_{t-i+1}+p_{t-i+3}+\cdots+p_t} + \alpha^{p_{t-i}+p_{t-i+2}+\cdots+p_t} + \alpha^{p_{t-i+1}+\cdots+p_t},\\ &\vdots\\ \sigma_{l,t} &= \alpha^{p_1+p_2+\cdots+p_t}.\end{aligned}$$

In addition, the coefficients of $\sigma_R(x)$ are written as follows:

$$\begin{aligned}
\sigma_{r,1} &= \alpha^{p_1+1} + \alpha^{p_2+1} + \cdots + \alpha^{p_t+1}, \\
\sigma_{r,2} &= \alpha^{(p_1+1)+(p_2+1)} + \alpha^{(p_1+1)+(p_3+1)} + \alpha^{(p_2+1)+(p_3+1)} \\
&\quad + \cdots + \alpha^{(p_{t-2}+1)+(p_t+1)} + \alpha^{(p_{t-1}+1)+(p_t+1)}, \\
&\vdots \\
\sigma_{r,i} &= \alpha^{(p_1+1)+\cdots+(p_i+1)} + \alpha^{(p_1+1)+\cdots+(p_{i-1}+1)+(p_{i+1}+1)} \\
&\quad + \alpha^{(p_1+1)+\cdots+(p_{i-2}+1)+(p_i+1)+(p_{i+1}+1)} + \cdots \\
&\quad + \alpha^{(p_{t-i}+1)+(p_{t-i+1}+1)+(p_{t-i+3}+1)+\cdots+(p_t+1)} \\
&\quad + \alpha^{(p_{t-i}+1)+(p_{t-i+2}+1)+\cdots+(p_t+1)} + \alpha^{(p_{t-i+1}+1)+\cdots+(p_t+1)}, \\
&\vdots \\
\sigma_{r,t} &= \alpha^{(p_1+1)+(p_2+1)+\cdots+(p_t+1)}.
\end{aligned}$$

Thus, $\sigma_{r,i}$ is expressed in terms of $\sigma_{l,i}$ as

$$\begin{aligned}
\sigma_{r,i} &= \alpha^i\alpha^{p_1+\cdots+p_i} + \alpha^i\alpha^{p_1+\cdots+p_{i-1}+p_{i+1}} + \alpha^i\alpha^{p_1+\cdots+p_{i-2}+p_i+p_{i+1}} \\
&\quad + \cdots + \alpha^i\alpha^{p_{t-i}+p_{t-i+1}+p_{t-i+3}+\cdots+p_t} + \alpha^i\alpha^{p_{t-i}+p_{t-i+2}+\cdots+p_t} \\
&\quad + \alpha^i\alpha^{p_{t-i+1}+\cdots+p_t} \\
&= \alpha^i\sigma_{l,i}.
\end{aligned}$$

□

From Theorem 7, if $(E_L, E_R) = (2, 2)$, I obtain

$$\sigma_{r,1} = \alpha\sigma_{l,1}, \tag{5.25}$$

$$\sigma_{r,2} = \alpha^2\sigma_{l,2}. \tag{5.26}$$

5.4.3 Method for Finding the Error-Locator Polynomial

In this section, I show how to find the left and right error-locator polynomials by using the equations derived in Sec. 5.4.1 and Sec. 5.4.2. In each case, the conflict-locator polynomial $\tau(x)$ and the symbol-pair syndromes $s_{l,1}$ and $s_{r,1}$ are known; therefore, $\sigma_{l,1} = s_{l,1}$ and $\sigma_{r,1} = s_{r,1}$ from (5.23) and (5.24). **Case I:** In the case of $(E_L, E_R) = (2, 2)$ and $W_H(\boldsymbol{S}) = 4$, the coefficients of the error-locator polynomial and conflict-locator polynomial in (5.17) are related by the following equations:

$$\begin{cases} \sigma_{l,1} + \sigma_{r,1} = \tau_1 \\ \sigma_{l,2} + \sigma_{l,1}\sigma_{r,1} + \sigma_{r,2} = \tau_2 \\ \sigma_{r,1}\sigma_{l,2} + \sigma_{l,1}\sigma_{r,2} = \tau_3 \\ \sigma_{l,2}\sigma_{r,2} = \tau_4 \end{cases}.$$

Then, $\sigma_{l,2}$ and $\sigma_{r,2}$ are obtained by

$$\sigma_{l,2} = \frac{\tau_2 + \alpha(s_{l,1})^2}{1+\alpha^2}, \quad \sigma_{r,2} = \alpha^2\sigma_{l,2}. \tag{5.27}$$

Table 5.2: Methods for calculating $\sigma_{l,2}$ and $\sigma_{r,2}$ in Step 3-1 of proposed decoding algorithm I

$W_H(\boldsymbol{S})$	Condition	$\sigma_{l,2}$	$\sigma_{r,2}$
4	Calculate a candidate set of $\sigma_L(x)$ and $\sigma_R(x)$.	(5.27)	
3	Calculate two candidate sets of $\sigma_L(x)$ and $\sigma_R(x)$.	(5.29)	0
		0	(5.30)
2	If the coefficient matrix of (5.28) is nonsingular, calculate a candidate set of $\sigma_L(x)$ and $\sigma_R(x)$.	(5.28)	(5.26)
	If the coefficient matrix of (5.28) is singular, calculate three candidate sets of $\sigma_L(x)$ and $\sigma_R(x)$.	τ_2	0
		0	τ_2
		0	0
1	If the coefficient matrix of (5.31) is nonsingular, calculate a candidate set of $\sigma_L(x)$ and $\sigma_R(x)$.	(5.31)	0
	If the coefficient matrix of (5.32) is nonsingular, calculate a candidate set of $\sigma_L(x)$ and $\sigma_R(x)$.	0	(5.32)
	If the coefficient matrices of both (5.31) and (5.32) are singular, go to Step 4.	—	—
0	Calculate a candidate set of $\sigma_L(x)$ and $\sigma_R(x)$.	0	0

In the case of $(E_L, E_R) = (2, 2)$ and $W_H(\boldsymbol{S}) = 2$, the coefficients of the error-locator polynomial and conflict-locator polynomial in (5.17) are related by the following equations:

$$\begin{cases} \sigma_{l,1} + \sigma_{r,1} = \tau_1 \\ \sigma_{l,2} + \sigma_{l,1}\sigma_{r,1} + \sigma_{r,2} = \tau_2 + \alpha^{2k_1} \\ \sigma_{r,1}\sigma_{l,2} + \sigma_{l,1}\sigma_{r,2} = \tau_1\alpha^{2k_1} \\ \sigma_{l,2}\sigma_{r,2} = \tau_2\alpha^{2k_1} \end{cases}.$$

They are transformed as follows:

$$\begin{cases} (1+\alpha^2)\sigma_{l,2} + \alpha^{2k_1} = \tau_2 + \alpha(s_{l,1})^2 \\ (\alpha+\alpha^2)s_{l,1}\sigma_{l,2} + \tau_2\alpha^{2k_1} = 0 \end{cases}, \tag{5.28}$$

where $\sigma_{l,2}$ and α^{2k_1} are unknown. $\sigma_{l,2}$ is obtained by solving (5.28); then, $\sigma_{r,2}$ is obtained by (5.26).

Case II: In the case of $(E_L, E_R) = (2, 1)$ and $W_H(\boldsymbol{S}) = 3$, the coefficients of the error-locator polynomial and conflict-locator polynomial in (5.18) are related by the following equations:

$$\begin{cases} \sigma_{l,1} + \sigma_{r,1} = \tau_1 \\ \sigma_{l,2} + \sigma_{l,1}\sigma_{r,1} = \tau_2 \\ \sigma_{r,1}\sigma_{l,2} = \tau_3 \end{cases}.$$

Then, $\sigma_{l,2}$ is obtained by

$$\sigma_{l,2} = \tau_2 + s_{l,1}s_{r,1}. \tag{5.29}$$

Similarly, in the case of $(E_L, E_R) = (1, 2)$ and $W_H(\boldsymbol{S}) = 3$, $\sigma_{r,2}$ is obtained by

$$\sigma_{r,2} = \tau_2 + s_{l,1}s_{r,1}. \tag{5.30}$$

In the case of $(E_L, E_R) = (2,1)$ and $W_H(\boldsymbol{S}) = 1$, the coefficients of the error-locator polynomial and conflict-locator polynomial in (5.18) are related by the following equations:

$$\begin{cases} \sigma_{l,1} + \sigma_{r,1} = \tau_1 \\ \sigma_{l,2} + \sigma_{l,1}\sigma_{r,1} = \alpha^{2k_1} \\ \sigma_{r,1}\sigma_{l,2} = \tau_1\alpha^{2k_1} \end{cases}.$$

They are transformed as follows:

$$\begin{cases} \sigma_{l,2} + \alpha^{2k_1} = s_{l,1}s_{r,1} \\ s_{r,1}\sigma_{l,2} + \tau_1\alpha^{2k_1} = 0 \end{cases}, \tag{5.31}$$

where $\sigma_{l,2}$ and α^{2k_1} are unknown. $\sigma_{l,2}$ is obtained by solving (5.31). Similarly, in the case of $(E_L, E_R) = (1,2)$ and $W_H(\boldsymbol{S}) = 1$, I derive the following equations:

$$\begin{cases} \sigma_{r,2} + \alpha^{2k_1} = s_{l,1}s_{r,1} \\ s_{l,1}\sigma_{r,2} + \tau_1\alpha^{2k_1} = 0 \end{cases}, \tag{5.32}$$

where $\sigma_{r,2}$ and α^{2k_1} are unknown. $\sigma_{r,2}$ is obtained by solving (5.32).

Case III: In the case of $(E_L, E_R)=(2,0)$ and $W_H(\boldsymbol{S})=2$, $\sigma_L(x) = \tau(x)$ is obtained by (5.19). Similarly, in the case of $(E_L, E_R) = (0,2)$ and $W_H(\boldsymbol{S}) = 2$, $\sigma_R(x) = \tau(x)$ is obtained.

In the case of $(E_L, E_R) = (1,1)$ and $W_H(\boldsymbol{S}) = 2$ or $W_H(\boldsymbol{S}) = 0$, the only unknown quantities are $\sigma_{l,1}$ and $\sigma_{r,1}$; then, $\sigma_{l,1} = s_{l,1}$ and $\sigma_{r,1} = s_{r,1}$ are obtained by (5.23) and (5.24).

Case IV: In the case of $(E_L, E_R) = (1,1)$ and $W_H(\boldsymbol{S}) = 2$, the only unknown quantities are $\sigma_{l,1}$ and $\sigma_{r,1}$; then, $\sigma_{l,1} = s_{l,1}$ and $\sigma_{r,1} = s_{r,1}$ are obtained by (5.23) and (5.24).

Case V: In the case of $(E_L, E_R) = (1,0)$ and $W_H(\boldsymbol{S}) = 1$, the only unknown quantities are $\sigma_{l,1}$; $\sigma_{l,1} = s_{l,1}$ is obtained by (5.23). Similarly, in the case of $(E_L, E_R) = (0,1)$ and $W_H(\boldsymbol{S}) = 1$, $\sigma_{r,1} = s_{r,1}$ is obtained by (5.24).

5.4.4 Decoding Algorithm for Two-Pair Error Correction

I propose the following decoding algorithm for two-pair error correction.

Proposed Decoding Algorithm I (Two-Pair Error Correction)

Input: Received pair vector $\overleftrightarrow{\boldsymbol{u}}$ (left and right received polynomials $u_L(x)$ and $u_R(x)$).

Output: Corrected codeword $\hat{\boldsymbol{c}}$ or failure symbol F.

Step 1. Calculate the symbol-pair syndromes $\overleftrightarrow{s}$ and neighbor-symbol syndrome $S(x)$:

$$\begin{aligned} s_{l,1} &:= u_L(\alpha), \\ s_{r,1} &:= \alpha^i u_R(\alpha), \\ S(x) &:= u_L(x) \oplus (x u_R(x) \bmod x^n - 1). \end{aligned}$$

Step 2. Calculate $W_H(\boldsymbol{S})$ and the conflict-locator polynomial $\tau(x)$:

$$\tau(x) := \prod_{\substack{i=0 \\ S_i \neq 0}}^{n-1} (1 - \alpha^i x).$$

Table 5.3: Classification of the pair error vector

		$E_L + E_R$	(E_L, E_R)
Three-pair error	Case 1	6	(3,3)
	Case 2	5	(3,2), (2,3)
	Case 3	4	(3,1), (1,3), (2,2)
	Case 4	3	(3,0), (0,3), (2,1), (1,2)
Two-pair error	Case 5	4	(2,2)
	Case 6	3	(2,1), (1,2)
	Case 7	2	(2,0), (0,2), (1,1)
Single-pair error	Case 8	2	(1,1)
	Case 9	1	(1,0), (0,1)

Step 3. Correct two-pair errors.

Step 3-1. Calculate $\sigma_L(x) = 1 + \sigma_{l,1}x + \sigma_{l,2}x^2$ and $\sigma_R(x) = 1 + \sigma_{r,1}x + \sigma_{r,2}x^2$ by classification. Set $\sigma_{l,1} := s_{l,1}$ and $\sigma_{r,1} := s_{r,1}$ and also calculate $\sigma_{l,2}$ and $\sigma_{r,2}$ as summarized in Table 5.2.

Step 3-2. Estimate the pair error vector $\overleftrightarrow{\boldsymbol{e}}$ by using the left and right error-locator polynomials derived in Step 3-1 and examine the pair weight $W_p(\overleftrightarrow{\boldsymbol{e}})$. If the pair error vector satisfies $W_p(\overleftrightarrow{\boldsymbol{e}}) = 2$, $\pi(\hat{\boldsymbol{c}}) := \overleftrightarrow{\boldsymbol{u}} + \overleftrightarrow{\boldsymbol{e}}$, and output $\hat{\boldsymbol{c}}$. Then, terminate this algorithm; otherwise, go to Step 4.

Step 4. Correct a single-pair error.

Step 4-1. Calculate $\sigma_L(x) = 1 + \sigma_{l,1}x$ and $\sigma_R(x) = 1 + \sigma_{r,1}x$ by using (5.23) and (5.24); $\sigma_{l,1} := s_{l,1}$ and $\sigma_{r,1} := s_{r,1}$.

Step 4-2. Estimate $\overleftrightarrow{\boldsymbol{e}}$ by using the left and right error-locator polynomials derived in Step 4-1 and examine $W_p(\overleftrightarrow{\boldsymbol{e}})$. If the pair error vector satisfies $W_p(\overleftrightarrow{\boldsymbol{e}}) = 1$, $\pi(\hat{\boldsymbol{c}}) := \overleftrightarrow{\boldsymbol{u}} + \overleftrightarrow{\boldsymbol{e}}$, and output $\hat{\boldsymbol{c}}$. Then, terminate this algorithm; otherwise, output the failure symbol F and terminate this algorithm. ■

This algorithm corrects two-pair errors in Step 3 and corrects a single-pair error in Step 4. The reason for this process is to avoid that miscorrection of the received pair vectors that suffer from two-pair errors when a single-pair error is corrected. Moreover, the pair error vector is uniquely-determined by examining the pair weight of $\overleftrightarrow{\boldsymbol{e}}$ in Step 3-2 and Step 4-2 from Theorem 2.

5.5 Decoding Problem for Three-Pair Error Correction

In this section, I consider a decoding problem for three-pair error correction using the two-error correcting BCH codes.

5.5.1 Classification of Pair Error Vector

I classify the pair error vectors according to $E_L + E_R$, as indicated as Table 5.3. In each case, I consider (5.15) when $W_H(\boldsymbol{S})$ is obtained.

Case 1: In the case of $(E_L, E_R) = (3,3)$, (5.14) holds, $t' = 0$, $t' = 1$, and $t' = 2$. From (5.12), $W_H(\boldsymbol{S}) = 6$ if $t' = 0$, $W_H(\boldsymbol{S}) = 4$ if $t' = 1$, and $W_H(\boldsymbol{S}) = 2$ if $t' = 2$. Then, (5.15) is written as follows:

$$\sigma_L(x)\sigma_R(x) = \begin{cases} \tau(x) & \text{if } W_H(\boldsymbol{S}) = 6 \\ \tau(x)(1-\alpha^{k_1}x)^2 & \text{if } W_H(\boldsymbol{S}) = 4 \\ \tau(x)\prod_{i=1}^{2}(1-\alpha^{k_i}x)^2 & \text{if } W_H(\boldsymbol{S}) = 2 \end{cases}. \tag{5.33}$$

Case 2: In the case of $(E_L, E_R) = (3,2)$, (5.14) holds, $t' = 0$, $t' = 1$, and $t' = 2$. From (5.12), $W_H(\boldsymbol{S}) = 5$ if $t' = 0$, $W_H(\boldsymbol{S}) = 3$ if $t' = 1$, and $W_H(\boldsymbol{S}) = 1$ if $t' = 2$. Then, (5.15) is written as follows:

$$\sigma_L(x)\sigma_R(x) = \begin{cases} \tau(x) & \text{if } W_H(\boldsymbol{S}) = 5 \\ \tau(x)(1-\alpha^{k_1}x)^2 & \text{if } W_H(\boldsymbol{S}) = 3 \\ \tau(x)\prod_{i=1}^{2}(1-\alpha^{k_i}x)^2 & \text{if } W_H(\boldsymbol{S}) = 1 \end{cases}. \tag{5.34}$$

Similarly, in the case of $(E_L, E_R) = (2,3)$, (5.15) is written as (5.34).
Case 3: In the case of $(E_L, E_R) = (3,1)$, (5.14) holds, $t' = 0$, and $t' = 1$. From (5.12), $W_H(\boldsymbol{S}) = 4$ if $t' = 0$, and $W_H(\boldsymbol{S}) = 2$ if $t' = 1$. Then, (5.15) is written as follows:

$$\sigma_L(x)\sigma_R(x) = \begin{cases} \tau(x) & \text{if } W_H(\boldsymbol{S}) = 4 \\ \tau(x)(1-\alpha^{k_1}x)^2 & \text{if } W_H(\boldsymbol{S}) = 2 \end{cases}. \tag{5.35}$$

Similarly, in the case of $(E_L, E_R) = (1,3)$, (5.15) is written as (5.35).

In the case of $(E_L, E_R) = (2,2)$, (5.14) holds, $t' = 0$, $t' = 1$, and $t' = 2$. From (5.12), $W_H(\boldsymbol{S}) = 4$ if $t' = 0$, $W_H(\boldsymbol{S}) = 2$ if $t' = 1$, and $W_H(\boldsymbol{S}) = 0$ if $t' = 2$. Then, (5.15) is written as follows:

$$\sigma_L(x)\sigma_R(x) = \begin{cases} \tau(x) & \text{if } W_H(\boldsymbol{S}) = 4 \\ \tau(x)(1-\alpha^{k_1}x)^2 & \text{if } W_H(\boldsymbol{S}) = 2 \\ \prod_{i=1}^{2}(1-\alpha^{k_i}x)^2 & \text{if } W_H(\boldsymbol{S}) = 0 \end{cases}. \tag{5.36}$$

Case 4: In the case of $(E_L, E_R) = (3,0)$, (5.14) holds, and $t' = 0$. If $t' = 0$, $W_H(\boldsymbol{S}) = 3$ from (5.12); then, (5.15) is written as follows:

$$\sigma_L(x)\sigma_R(x) = \tau(x). \tag{5.37}$$

Similarly, in the case of $(E_L, E_R) = (0,3)$, (5.15) is written as (5.37).

In the case of $(E_L, E_R) = (2,1)$, (5.14) holds, $t' = 0$, and $t' = 1$. From (5.12), $W_H(\boldsymbol{S}) = 3$ if $t' = 0$, and $W_H(\boldsymbol{S}) = 1$ if $t' = 1$. Then, (5.15) is written as follows:

$$\sigma_L(x)\sigma_R(x) = \begin{cases} \tau(x) & \text{if } W_H(\boldsymbol{S}) = 3 \\ \tau(x)(1-\alpha^{k_1}x)^2 & \text{if } W_H(\boldsymbol{S}) = 1 \end{cases}. \tag{5.38}$$

Similarly, in the case of $(E_L, E_R) = (1,2)$, (5.15) is written as (5.38).
Cases 5–9: These cases are the same as Cases I–V in Sec. 5.4, respectively.

5.5.2 Relation between the Error-Locator Polynomial and the Symbol-Pair Syndrome

From Theorem 7, if $(E_L, E_R) = (3, 3)$, I obtain (5.25) and (5.26), and

$$\sigma_{r,3} = \alpha^3 \sigma_{l,3}. \tag{5.39}$$

If $E_L = 3$ or $E_R = 3$, I have the following theorem.

Theorem 8. If $E_L = 3$, the left error-locator polynomial is $\sigma_L(x) = 1 + \sigma_{l,1}x + \sigma_{l,2}x^2 + \sigma_{l,3}x^3$; then, $\sigma_{l,3}$ is given by

$$\sigma_{l,3} = (s_{l,1})^3 + s_{l,3} + s_{l,1}\sigma_{l,2}. \tag{5.40}$$

Similarly, if $E_R = 3$, the right error-locator polynomial is $\sigma_R(x) = 1 + \sigma_{r,1}x + \sigma_{r,2}x^2 + \sigma_{r,3}x^3$; then, $\sigma_{r,3}$ is given by

$$\sigma_{r,3} = (s_{r,1})^3 + s_{r,3} + s_{r,1}\sigma_{r,2}. \tag{5.41}$$

Proof: If $E_L = 3$, the left error polynomial is $e_L(x) = x^{p_1} + x^{p_2} + x^{p_3}$, and the coefficients of each term of the left error-locator polynomial are

$$\begin{aligned}
\sigma_{l,1} &= \alpha^{p_1} + \alpha^{p_2} + \alpha^{p_3}, \\
\sigma_{l,2} &= \alpha^{p_1+p_2} + \alpha^{p_2+p_3} + \alpha^{p_3+p_1}, \\
\sigma_{l,3} &= \alpha^{p_1+p_2+p_3}.
\end{aligned}$$

(5.40) is derived by the following transformation:

$$\begin{aligned}
\sigma_{l,3} &= \alpha^{p_1+p_2+p_3} \\
&= (\alpha^{p_1} + \alpha^{p_2} + \alpha^{p_3})^3 + \alpha^{3p_1} + \alpha^{3p_2} + \alpha^{3p_3} \\
&+ (\alpha^{p_1} + \alpha^{p_2} + \alpha^{p_3})(\alpha^{p_1+p_2} + \alpha^{p_2+p_3} + \alpha^{p_3+p_1}) \\
&= (s_{l,1})^3 + s_{l,3} + s_{l,1}\sigma_{l,2}.
\end{aligned}$$

(5.41) is derived by a similar transformation. □

If $E_L = 2$ or $E_R = 2$, the following theorem holds.

Theorem 9. If $E_L = 2$, the left error-locator polynomial is $\sigma_L(x) = 1 + \sigma_{l,1}x + \sigma_{l,2}x^2$; then, $\sigma_{l,2}$ is given by

$$\sigma_{l,2} = \frac{(s_{l,1})^3 + s_{l,3}}{s_{l,1}}. \tag{5.42}$$

Similarly, if $E_R = 2$, the right error-locator polynomial is $\sigma_R(x) = 1 + \sigma_{r,1}x + \sigma_{r,2}x^2$; then, $\sigma_{r,2}$ is given by

$$\sigma_{r,2} = \frac{(s_{r,1})^3 + s_{r,3}}{s_{r,1}}. \tag{5.43}$$

Proof: If $E_L = 2$, (5.42) is derived by assigning $\sigma_{l,3} = 0$ in (5.40). Similarly, if $E_R = 2$, (5.43) is derived by assigning $\sigma_{r,3} = 0$ in (5.41). □

5.5.3 Method for Finding the Error-Locator Polynomial

In this section, I show how to find the left and right error-locator polynomials by using the equations derived in Sec. 5.5.1 and Sec. 5.5.2. In each case, the conflict-locator polynomial $\tau(x)$ and the symbol-pair syndromes $s_{l,1}$ and $s_{r,1}$ are known; therefore, $\sigma_{l,1} = s_{l,1}$ and $\sigma_{r,1} = s_{r,1}$ from (5.23) and (5.24).

Case 1: In the case of $(E_L, E_R) = (3,3)$ and $W_H(\boldsymbol{S}) = 6$, the coefficients of the error-locator polynomial and conflict-locator polynomial in (5.33) are related by the following equations:

$$\begin{cases} \sigma_{l,1} + \sigma_{r,1} = \tau_1 \\ \sigma_{l,2} + \sigma_{l,1}\sigma_{r,1} + \sigma_{r,2} = \tau_2 \\ \sigma_{l,3} + \sigma_{r,1}\sigma_{l,2} + \sigma_{l,1}\sigma_{r,2} + \sigma_{r,3} = \tau_3 \\ \sigma_{r,1}\sigma_{l,3} + \sigma_{l,2}\sigma_{r,2} + \sigma_{l,1}\sigma_{r,3} = \tau_4 \\ \sigma_{r,2}\sigma_{l,3} + \sigma_{l,2}\sigma_{r,3} = \tau_5 \\ \sigma_{l,3}\sigma_{r,3} = \tau_6 \end{cases}.$$

Then, $\sigma_{l,2}$ and $\sigma_{r,2}$ are obtained by

$$\sigma_{l,2} = \frac{\tau_2 + s_{l,1}s_{r,1}}{1+\alpha^2}, \quad \sigma_{r,2} = \alpha^2\sigma_{l,2} \tag{5.44}$$

Then, $\sigma_{l,3}$ and $\sigma_{r,3}$ are obtained by (5.40) and (5.41).

In the case of $(E_L, E_R) = (3,3)$ and $W_H(\boldsymbol{S}) = 4$, the coefficients of error-locator polynomial and conflict-locator polynomial in (5.33) are related by the following equations:

$$\begin{cases} \sigma_{l,1} + \sigma_{r,1} = \tau_1 \\ \sigma_{l,2} + \sigma_{l,1}\sigma_{r,1} + \sigma_{r,2} = \tau_2 + \alpha^{2k_1} \\ \sigma_{l,3} + \sigma_{r,1}\sigma_{l,2} + \sigma_{l,1}\sigma_{r,2} + \sigma_{r,3} = \tau_3 + \tau_1\alpha^{2k_1} \\ \sigma_{r,1}\sigma_{l,3} + \sigma_{l,2}\sigma_{r,2} + \sigma_{l,1}\sigma_{r,3} = \tau_4 + \tau_2\alpha^{2k_1} \\ \sigma_{r,2}\sigma_{l,3} + \sigma_{l,2}\sigma_{r,3} = \tau_3\alpha^{2k_1} \\ \sigma_{l,3}\sigma_{r,3} = \tau_4\alpha^{2k_1} \end{cases}.$$

They are transformed as follows:

$$\begin{cases} (1+\alpha^2)\sigma_{l,2} + \alpha^{2k_1} = \tau_2 + \alpha(s_{l,1})^2 \\ (1+\alpha+\alpha^2+\alpha^3)s_{l,1}\sigma_{l,2} + \alpha^{2k_1} = \tau_3 + (1+\alpha^3)((s_{l,1})^3 + s_{l,3}) \end{cases}, \tag{5.45}$$

where $\sigma_{l,2}$ and α^{2k_1} are unknown. $\sigma_{l,2}$ is obtained by solving (5.45), and $\sigma_{r,2}$ is obtained by (5.26). Then, $\sigma_{l,3}$ and $\sigma_{r,3}$ are obtained by (5.40) and (5.41).

In the case of $(E_L, E_R) = (3,3)$ and $W_H(\boldsymbol{S}) = 2$, the coefficients of the error-locator polynomial and conflict-locator polynomial in (5.33) are related by the following equations:

$$\begin{cases} \sigma_{l,1} + \sigma_{r,1} = \tau_1 \\ \sigma_{l,2} + \sigma_{l,1}\sigma_{r,1} + \sigma_{r,2} = \tau_2 + \alpha^{2k_1} + \alpha^{2k_2} \\ \sigma_{l,3} + \sigma_{r,1}\sigma_{l,2} + \sigma_{l,1}\sigma_{r,2} + \sigma_{r,3} = \tau_1(\alpha^{2k_1} + \alpha^{2k_2}) \\ \sigma_{r,1}\sigma_{l,3} + \sigma_{l,2}\sigma_{r,2} + \sigma_{l,1}\sigma_{r,3} = \tau_2(\alpha^{2k_1} + \alpha^{2k_2}) \\ \sigma_{r,2}\sigma_{l,3} + \sigma_{l,2}\sigma_{r,3} = \tau_1\alpha^{2k_1+2k_2} \\ \sigma_{l,3}\sigma_{r,3} = \tau_2\alpha^{2k_1+2k_2} \end{cases}.$$

They are transformed as follows:

$$\left\{\begin{array}{c} (1+\alpha^2)\sigma_{l,2}+(\alpha^{2k_1}+\alpha^{2k_2})=\tau_2+\alpha(s_{l,1})^2 \\ (1+\alpha+\alpha^2+\alpha^3)s_{l,1}\sigma_{l,2}+\tau_1(\alpha^{2k_1}+\alpha^{2k_2}) \\ =(1+\alpha^3)((s_{l,1})^3+s_{l,3}) \end{array}\right. , \tag{5.46}$$

where $\sigma_{l,2}$ and $\alpha^{2k_1}+\alpha^{2k_2}$ are unknown. $\sigma_{l,2}$ is obtained by solving (5.46), and $\sigma_{r,2}$ is obtained by (5.26). Then, $\sigma_{l,3}$ and $\sigma_{r,3}$ are obtained by (5.40) and (5.41).

Case 2: In the case of $(E_L, E_R) = (3,2)$ and $W_H(\boldsymbol{S}) = 5$, the coefficients of the error-locator polynomial and conflict-locator polynomial in (5.34) are related by the following equations:

$$\left\{\begin{array}{r} \sigma_{l,1}+\sigma_{r,1}=\tau_1 \\ \sigma_{l,2}+\sigma_{l,1}\sigma_{r,1}+\sigma_{r,2}=\tau_2 \\ \sigma_{l,3}+\sigma_{r,1}\sigma_{l,2}+\sigma_{l,1}\sigma_{r,2}=\tau_3 \\ \sigma_{r,1}\sigma_{l,3}+\sigma_{l,2}\sigma_{r,2}=\tau_4 \\ \sigma_{r,2}\sigma_{l,3}=\tau_5 \end{array}\right. .$$

Then, $\sigma_{r,2}$ and $\sigma_{l,2}$ are obtained by

$$\sigma_{r,2}=\frac{(s_{r,1})^3+s_{r,3}}{s_{r,1}}, \quad \sigma_{l,2}=\tau_2+s_{l,1}s_{r,1}+\sigma_{r,2}. \tag{5.47}$$

Then, $\sigma_{l,3}$ is obtained by (5.40). Similarly, in the case of $(E_L, E_R) = (2,3)$ and $W_H(\boldsymbol{S}) = 5$, $\sigma_{l,2}$ and $\sigma_{r,2}$ are obtained by

$$\sigma_{l,2}=\frac{(s_{l,1})^3+s_{l,3}}{s_{l,1}}, \quad \sigma_{r,2}=\tau_2+s_{l,1}s_{r,1}+\sigma_{l,2}. \tag{5.48}$$

Then, $\sigma_{r,3}$ is obtained by (5.41).

In the case of $(E_L, E_R) = (3,2)$ and $W_H(\boldsymbol{S}) = 3$, the coefficients of the error-locator polynomial and conflict-locator polynomial in (5.34) are related by the following equations:

$$\left\{\begin{array}{r} \sigma_{l,1}+\sigma_{r,1}=\tau_1 \\ \sigma_{l,2}+\sigma_{l,1}\sigma_{r,1}+\sigma_{r,2}=\tau_2+\alpha^{2k_1} \\ \sigma_{l,3}+\sigma_{r,1}\sigma_{l,2}+\sigma_{l,1}\sigma_{r,2}=\tau_3+\tau_1\alpha^{2k_1} \\ \sigma_{r,1}\sigma_{l,3}+\sigma_{l,2}\sigma_{r,2}=\tau_2\alpha^{2k_1} \\ \sigma_{r,2}\sigma_{l,3}=\tau_3\alpha^{2k_1} \end{array}\right. .$$

They are transformed as follows:

$$\left\{\begin{array}{l} \sigma_{l,2}+\alpha^{2k_1}=\tau_2+s_{l,1}s_{r,1}+\sigma_{r,2} \\ \sigma_{l,3}+s_{r,1}\sigma_{l,2}+\tau_1\alpha^{2k_1}=\tau_3+s_{l,1}\sigma_{r,2} \\ s_{r,1}\sigma_{l,3}+\sigma_{r,2}\sigma_{l,2}+\tau_2\alpha^{2k_1}=0 \end{array}\right. , \tag{5.49}$$

where $\sigma_{l,3}$, $\sigma_{l,2}$, and α^{2k_1} are unknown, and $\sigma_{r,2}$ is known because it is obtained by (5.43). $\sigma_{l,3}$ and $\sigma_{l,2}$ are obtained by solving (5.49). Similarly, in the case of $(E_L, E_R) = (2,3)$ and $W_H(\boldsymbol{S}) = 3$, I derive the following equations:

$$\left\{\begin{array}{l} \sigma_{r,2}+\alpha^{2k_1}=\tau_2+s_{l,1}s_{r,1}+\sigma_{l,2} \\ \sigma_{r,3}+s_{l,1}\sigma_{r,2}+\tau_1\alpha^{2k_1}=\tau_3+s_{r,1}\sigma_{l,2} \\ s_{l,1}\sigma_{r,3}+\sigma_{l,2}\sigma_{r,2}+\tau_2\alpha^{2k_1}=0 \end{array}\right. , \tag{5.50}$$

where $\sigma_{r,3}$, $\sigma_{r,2}$, and α^{2k_1} are unknown, and $\sigma_{l,2}$ is obtained by (5.42). $\sigma_{r,3}$ and $\sigma_{r,2}$ are obtained by solving (5.50).

In the case of $(E_L, E_R) = (3, 2)$ and $W_H(\boldsymbol{S}) = 1$, the coefficients of the error-locator polynomial and conflict-locator polynomial in (5.34) are related by the following equations:

$$\left\{\begin{array}{r@{\,}l} \sigma_{l,1} + \sigma_{r,1} &= \tau_1 \\ \sigma_{l,2} + \sigma_{l,1}\sigma_{r,1} + \sigma_{r,2} &= \alpha^{2k_1} + \alpha^{2k_2} \\ \sigma_{l,3} + \sigma_{r,1}\sigma_{l,2} + \sigma_{l,1}\sigma_{r,2} &= \tau_1(\alpha^{2k_1} + \alpha^{2k_2}) \\ \sigma_{r,1}\sigma_{l,3} + \sigma_{l,2}\sigma_{r,2} &= \alpha^{2k_1+2k_2} \\ \sigma_{r,2}\sigma_{l,3} &= \tau_1\alpha^{2k_1+2k_2} \end{array}\right. .$$

They are transformed as follows:

$$\left\{\begin{array}{l} (s_{l,1}s_{r,1}+\sigma_{r,2})\,\sigma_{l,2}+\alpha^{2k_1+2k_2} = s_{r,1}((s_{l,1})^3+s_{l,3}) \\ s_{l,1}\sigma_{r,2}\sigma_{l,2} + \tau_1\alpha^{2k_1+2k_2} = ((s_{l,1})^3 + s_{l,3})\sigma_{r,2} \end{array}\right. , \tag{5.51}$$

where $\sigma_{l,2}$ and $\alpha^{2k_1+2k_2}$ are unknown, and $\sigma_{r,2}$ is obtained by (5.43). $\sigma_{l,2}$ is obtained by solving (5.51); then, $\sigma_{l,3}$ is obtained by (5.40). Similarly, in the case of $(E_L, E_R) = (2, 3)$ and $W_H(\boldsymbol{S}) = 1$, I derive the following equations:

$$\left\{\begin{array}{l} (s_{l,1}s_{r,1}+\sigma_{l,2})\,\sigma_{r,2}+\alpha^{2k_1+2k_2} = s_{l,1}((s_{r,1})^3+s_{r,3}) \\ s_{r,1}\sigma_{l,2}\sigma_{r,2} + \tau_1\alpha^{2k_1+2k_2} = ((s_{r,1})^3 + s_{r,3})\sigma_{l,2} \end{array}\right. , \tag{5.52}$$

where $\sigma_{r,2}$ and $\alpha^{2k_1+2k_2}$ are unknown, and $\sigma_{l,2}$ is obtained by (5.42). $\sigma_{r,2}$ is obtained by solving (5.52); then, $\sigma_{r,3}$ is obtained by (5.41).

Case 3: In the case of $(E_L, E_R) = (3, 1)$ and $W_H(\boldsymbol{S}) = 4$, the coefficients of the error-locator polynomial and conflict-locator polynomial in (5.35) are related by the following equations:

$$\left\{\begin{array}{r@{\,}l} \sigma_{l,1} + \sigma_{r,1} &= \tau_1 \\ \sigma_{l,2} + \sigma_{l,1}\sigma_{r,1} &= \tau_2 \\ \sigma_{l,3} + \sigma_{r,1}\sigma_{l,2} &= \tau_3 \\ \sigma_{r,1}\sigma_{l,3} &= \tau_4 \end{array}\right. .$$

Then, $\sigma_{l,2}$ and $\sigma_{l,3}$ are obtained by

$$\sigma_{l,2} = \tau_2 + s_{l,1}s_{r,1}, \quad \sigma_{l,3} = \frac{\tau_4}{s_{r,1}}. \tag{5.53}$$

Similarly, in the case of $(E_L, E_R) = (1, 3)$ and $W_H(\boldsymbol{S}) = 4$, $\sigma_{r,2}$ and $\sigma_{r,3}$ are obtained by

$$\sigma_{r,2} = \tau_2 + s_{l,1}s_{r,1}, \quad \sigma_{r,3} = \frac{\tau_4}{s_{l,1}}. \tag{5.54}$$

In the case of $(E_L, E_R) = (3, 1)$ and $W_H(\boldsymbol{S}) = 2$, the coefficients of the error-locator polynomial and conflict-locator polynomial in (5.35) are related by the following equations:

$$\left\{\begin{array}{r@{\,}l} \sigma_{l,1} + \sigma_{r,1} &= \tau_1 \\ \sigma_{l,2} + \sigma_{l,1}\sigma_{r,1} &= \tau_2 + \alpha^{2k_1} \\ \sigma_{l,3} + \sigma_{r,1}\sigma_{l,2} &= \tau_1\alpha^{2k_1} \\ \sigma_{r,1}\sigma_{l,3} &= \tau_2\alpha^{2k_1} \end{array}\right. .$$

They are transformed as follows:

$$\begin{cases} \sigma_{l,2} + \alpha^{2k_1} = \tau_2 + s_{l,1}s_{r,1} \\ \sigma_{l,3} + s_{r,1}\sigma_{l,2} + \tau_1\alpha^{2k_1} = 0 \\ s_{r,1}\sigma_{l,3} + \tau_2\alpha^{2k_1} = 0 \end{cases}, \tag{5.55}$$

where $\sigma_{l,3}$, $\sigma_{l,2}$, and α^{2k_1} are unknown. $\sigma_{l,3}$ and $\sigma_{l,2}$ are obtained by solving (5.55). Similarly, in the case of $(E_L, E_R) = (1, 3)$ and $W_H(\boldsymbol{S}) = 2$, I derive the following equations:

$$\begin{cases} \sigma_{r,2} + \alpha^{2k_1} = \tau_2 + s_{l,1}s_{r,1} \\ \sigma_{r,3} + s_{l,1}\sigma_{r,2} + \tau_1\alpha^{2k_1} = 0 \\ s_{l,1}\sigma_{r,3} + \tau_2\alpha^{2k_1} = 0 \end{cases}, \tag{5.56}$$

where $\sigma_{r,3}, \sigma_{r,2}$, and α^{2k_1} are unknown. $\sigma_{r,3}$ and $\sigma_{r,2}$ are obtained by solving (5.56).

In the case of $(E_L, E_R) = (2, 2)$ and $W_H(\boldsymbol{S}) = 4$, $W_H(\boldsymbol{S}) = 2$ or $W_H(\boldsymbol{S}) = 0$, and $\sigma_{l,2}$ and $\sigma_{r,2}$ are obtained by (5.42) and (5.43).

Case 4: In the case of $(E_L, E_R) = (3, 0)$ and $W_H(\boldsymbol{S}) = 3$, $\sigma_L(x) = \tau(x)$ is obtained by (5.37). Similarly, in the case of $(E_L, E_R) = (0, 3)$ and $W_H(\boldsymbol{S}) = 3$, $\sigma_R(x) = \tau(x)$ is obtained.

In the case of $(E_L, E_R) = (2, 1)$ and $W_H(\boldsymbol{S}) = 3$ or $W_H(\boldsymbol{S}) = 1$, $\sigma_{l,2}$ is obtained by (5.42). Similarly, in the case of $(E_L, E_R) = (1, 2)$ and $W_H(\boldsymbol{S}) = 3$ or $W_H(\boldsymbol{S}) = 1$, $\sigma_{r,2}$ is obtained by (5.43).

Cases 5–9: For any (E_L, E_R) and $W_H(\boldsymbol{S})$, either $\sigma_{l,1}$, $\sigma_{r,1}$, $\sigma_{l,2}$, or $\sigma_{r,2}$ is unknown. Then, $\sigma_{l,1}$, $\sigma_{r,1}$, $\sigma_{l,2}$, and $\sigma_{r,2}$ are obtained by (5.23), (5.24), (5.42), and (5.43), respectively.

5.5.4 Decoding Algorithm for Three-Pair Error Correction

I propose the following decoding algorithm for three-pair error correction.

Proposed Decoding Algorithm II (Three-Pair Error Correction)

Input: Received pair vector $\overleftrightarrow{\boldsymbol{u}}$ (left and right received polynomials $u_L(x)$ and $u_R(x)$).

Output: Corrected codeword $\hat{\boldsymbol{c}}$ or failure symbol F.

Step 1. Calculate the symbol-pair syndromes $\overleftrightarrow{s}$ and neighbor-symbol syndrome $S(x)$:

$$\begin{aligned} s_{l,i} &:= u_L(\alpha^i), \quad i = 1, 3, \\ s_{r,i} &:= \alpha^i u_R(\alpha^i), \quad i = 1, 3, \\ S(x) &:= u_L(x) \oplus (x u_R(x) \bmod x^n - 1). \end{aligned}$$

Step 2. Calculate $W_H(\boldsymbol{S})$ and the conflict-locator polynomial $\tau(x)$:

$$\tau(x) := \prod_{\substack{i=0 \\ S_i \neq 0}}^{n-1} (1 - \alpha^i x).$$

Step 3. Correct three-pair errors.

Step 3-1. Calculate $\sigma_L(x) = 1 + \sigma_{l,1}x + \sigma_{l,2}x^2 + \sigma_{l,3}x^3$ and $\sigma_R(x) = 1 + \sigma_{r,1}x + \sigma_{r,2}x^2 + \sigma_{r,3}x^3$ by classification. Set $\sigma_{l,1} := s_{l,1}$ and $\sigma_{r,1} := s_{r,1}$ and also calculate $\sigma_{l,2}$, $\sigma_{r,2}$, $\sigma_{l,3}$, and $\sigma_{r,3}$ as summarized in Table 5.4.

Step 3-2. Estimate the pair error vector $\overleftrightarrow{\boldsymbol{e}}$ by using the left and right error-locator polynomials derived in Step 3-1 and examine the pair weight $W_p(\overleftrightarrow{\boldsymbol{e}})$. If the pair error vector satisfies $W_p(\overleftrightarrow{\boldsymbol{e}}) = 3$, $\pi(\hat{\boldsymbol{c}}) := \overleftrightarrow{\boldsymbol{u}} + \overleftrightarrow{\boldsymbol{e}}$, and output $\hat{\boldsymbol{c}}$. Then, terminate this algorithm; otherwise, go to Step 4.

Step 4. Correct two-pair errors.

Step 4-1. Calculate $\sigma_L(x) = 1 + \sigma_{l,1}x + \sigma_{l,2}x^2$ and $\sigma_R(x) = 1 + \sigma_{r,1}x + \sigma_{r,2}x^2$ by using (5.23), (5.24), (5.42), and (5.43).

Step 4-2. Estimate $\overleftrightarrow{\boldsymbol{e}}$ from the left and right error-locator polynomials derived in Step 4-1 and examine $W_p(\overleftrightarrow{\boldsymbol{e}})$. If the pair error vector satisfies $W_p(\overleftrightarrow{\boldsymbol{e}}) = 2$, $\pi(\hat{\boldsymbol{c}}) := \overleftrightarrow{\boldsymbol{u}} + \overleftrightarrow{\boldsymbol{e}}$, and output $\hat{\boldsymbol{c}}$. Then, terminate this algorithm; otherwise, go to step 5.

Step 5. Correct a single-pair error.

Step 5-1. Calculate $\sigma_L(x) = 1 + \sigma_{l,1}x$ and $\sigma_R(x) = 1 + \sigma_{r,1}x$ by using (5.23) and (5.24).

Step 5-2. Estimate $\overleftrightarrow{\boldsymbol{e}}$ from the left and right error-locator polynomials derived in Step 5-1 and examine $W_p(\overleftrightarrow{\boldsymbol{e}})$. If the pair error vector satisfies $W_p(\overleftrightarrow{\boldsymbol{e}}) = 1$, $\pi(\hat{\boldsymbol{c}}) := \overleftrightarrow{\boldsymbol{u}} + \overleftrightarrow{\boldsymbol{e}}$, and output $\hat{\boldsymbol{c}}$. Then terminate this algorithm; otherwise, output the failure symbol F and terminate this algorithm. ■

This algorithm corrects three-pair errors in Step 3, two-pair errors in Step 4, and a single-pair error in Step 5. The reason for this process is to avoid the miscorrection of received pair vectors that suffer from three-pair errors when single-pair or two-pair errors are corrected. Moreover, the pair error vector is uniquely determined by examining the pair weight of $\overleftrightarrow{\boldsymbol{e}}$ from Theorem 2. I present an example of three-pair error correction by using the proposed decoding algorithm II.

Example 5. Suppose the binary (15,7) BCH code whose primitive polynomial is $p(x) = 1+x+x^4$ and generator polynomial is $g(x) = 1+x+x^2+x^4+x^8$. The code can correct up to two errors in the Hamming metric and up to three-pair errors in the pair metric. Let

$$\begin{aligned}\overleftrightarrow{\boldsymbol{u}} = [&(0,0),(1,1),(0,0),(0,1),(0,0),(0,0),(1,1),(0,0),\\ &(0,0),(0,0),(0,0),(0,0),(0,0),(0,0),(0,0)]\end{aligned}$$

be a received pair vector that has three-pair errors when the all-zero codeword [00000 00000 00000] is read. Then, the left and right received polynomials are $u_L(x) = x + x^6$ and $u_R(x) = x + x^3 + x^6$. In Step 1, the symbol-pair syndromes and the neighbor-symbol syndrome are calculated as follows:

$$\begin{aligned}
s_{l,1} &= u_L(\alpha) = \alpha + \alpha^6 = \alpha^{11},\\
s_{l,3} &= u_L(\alpha^3) = \alpha^3 + (\alpha^3)^6 = 0,\\
s_{r,1} &= \alpha u_R(\alpha) = \alpha(\alpha + \alpha^3 + \alpha^6) = \alpha^6,\\
s_{r,3} &= \alpha^3 u_R(\alpha) = \alpha^3(\alpha^3 + (\alpha^3)^3 + (\alpha^3)^6) = \alpha^{12},\\
S(x) &= x + x^2 + x^4 + x^6 + x^7.
\end{aligned}$$

In Step 2, the Hamming weight of $S(x)$ is examined, and the conflict-locator polynomial is derived as follows:

$$\begin{aligned} W_H(\boldsymbol{S}) &= 5, \\ \tau(x) &= (1-\alpha x)(1-\alpha^2 x)(1-\alpha^4 x)(1-\alpha^6 x)(1-\alpha^7 x) \\ &= 1+\alpha x+\alpha^{10}x^2+\alpha^{14}x^3+\alpha^{13}x^4+\alpha^5 x^5. \end{aligned}$$

In Step 3-1, the left and right error-locator polynomials are calculated. First, $\sigma_{l,1}=\alpha^{11}$ and $\sigma_{r,1}=\alpha^6$ from (5.23) and (5.24). Next, two candidate sets of $\sigma_L(x)$ and $\sigma_R(x)$ are calculated. The first is obtained by (5.47) as follows:

$$\begin{aligned} \sigma_{r,2} &= \frac{(\alpha^6)^3+\alpha^{12}}{\alpha^6} = \alpha^4, \\ \sigma_{l,2} &= \alpha^{10}+\alpha^{11}\alpha^6+\alpha^4 = 0. \end{aligned}$$

Then, $\sigma_{l,3}$ is obtained by (5.40) as

$$\sigma_{l,3} = (\alpha^{11})^3 = \alpha^3.$$

The other is obtained by (5.48) as follows:

$$\begin{aligned} \sigma_{l,2} &= \frac{(\alpha^{11})^3}{\alpha^{11}} = \alpha^7, \\ \sigma_{r,2} &= \alpha^{10}+\alpha^{11}\alpha^6+\alpha^7 = \alpha^3. \end{aligned}$$

Then, $\sigma_{r,3}$ is obtained by (5.41) as

$$\sigma_{r,3} = (\alpha^6)^3+\alpha^{12}+\alpha^6\alpha^3 = \alpha^{13}.$$

In Step 3-2, two-pair error vectors are estimated by the two candidates sets of $\sigma_L(x)$ and $\sigma_R(x)$ derived in Step 3-1. The first is $\sigma_L(x) = 1+\alpha^{11}x+\alpha^3x^3$ and $\sigma_R(x) = 1+\alpha^6x+\alpha^4x^2$. Then, all values of x do not satisfy $\sigma_L(x)=0$ or $\sigma_R(x)=0$; therefore, the error positions are not estimated. The other is $\sigma_L(x) = 1+\alpha^{11}x+\alpha^7x^2$ and $\sigma_R(x) = 1+\alpha^6x+\alpha^3x^2+\alpha^{13}x^3$. Then, the left error positions are 1 and 6 since $x=\alpha^{14}$ and $x=\alpha^9$ satisfy $\sigma_L(x)=0$, and the right error positions are 1, 3, and 6 since $x=\alpha^{13}$, $x=\alpha^{11}$, and $x=\alpha^8$ satisfy $\sigma_R(x)=0$. The left and right error polynomials are $e_L(x)=x+x^6$ and $e_R(x)=x+x^3+x^6$. Examining the pair weight of $\overleftrightarrow{\boldsymbol{e}}=(e_L,e_R)$, this algorithm confirms that $W_p(\overleftrightarrow{\boldsymbol{e}})=3$; then, $\pi(\hat{\boldsymbol{c}}) = \overleftrightarrow{\boldsymbol{u}}+\overleftrightarrow{\boldsymbol{e}} = \pi(\boldsymbol{0})$, and $\hat{\boldsymbol{c}}=\boldsymbol{0}$ is output. Thus, this error correction is successful.

■

5.6 Discussion

In this section, I compare the complexities of the proposed algorithms with those of existing algorithms and discuss the limitation of the proposed algorithms. The proposed algorithms and the existing algorithms in Refs. [6, 20] find a plurality of candidates of pair error pattern until the algorithms terminate. The complexity of these algorithms is dominated by the complexity of finding the candidate sets. In addition, the complexity of finding one candidate set is almost equal. Thus, as an indicator of the comparison, I consider the number N of determined candidates sets of the error-locator polynomials until the algorithm terminates. Note that N implies the number of times that the decoder for cyclic codes is executed in the existing algorithms.

In the Yaakobi's decoding algorithm [6], $N = 2$ because their algorithm uses the decoder for cyclic codes twice in Step 1 and Step 2 regardless of the pair error vectors. In the decoding algorithm proposed in Chapter 4, the condition under which the algorithm terminates depends on the numbers of the left and right error symbols. Then, $N = 1$ if the algorithm terminates in Step 2, $N = 2$ if the algorithm terminates in Step 3, and $N = 3$ if the algorithm terminates in Step 4. In the proposed algorithms, the condition under which the algorithm terminates depends on the numbers of left and right error symbols (E_L, E_R) and the Hamming weight of the neighbor-symbol syndrome $W_H(\boldsymbol{S})$. In Example 1, $N = 2$ since two candidates sets of error-locator polynomials are determined until the algorithm terminates. Table 5.5 and Table 5.6 summarize the values of N for all patterns of (E_L, E_R) and $W_H(\boldsymbol{S})$ if the proposed decoding algorithms I and Ⅱ are used.

A limitation of the proposed algorithms is that the number of correctable pair errors is limited to two-pair and three-pair errors. Thus, the disadvantage is that the proposed algorithms correct fewer pair errors than the existing algorithm. On the other hand, the advantage is that they define the error-locator polynomial and conflict-locator polynomial for pair error correction, and new decoding algorithms are proposed by using those polynomials. In future works, I will be to clarify the method for finding the error-locator polynomial regardless of the number of pair errors that are intended to be corrected.

The proposed decoding algorithms are different from the existing algorithm since the proposed decoding algorithms do not reduce the pair-decoding problem to the error-decoding problem in the Hamming metric. Then, I showed that there are some pair error vectors that are corrected more efficiently than the existing decoding algorithms from the results of a comparison and demonstrated the validity of the approach when pair errors are corrected with algebraic methods in the pair metric.

5.7 Conclusion

In this chapter, I have discussed the algebraic decoding of BCH codes over symbol-pair read channels. I have defined the error-locator polynomial and conflict-locator polynomial for symbol-pair read channels and have derived the relation between the two types of polynomials. In addition, I have discussed the decoding problem for two-pair and three-pair error correction and proposed two new algorithms that correct two-pair and three-pair errors with algebraic methods. The proposed algorithms are able to correct pair errors in the pair metric and independent of existing cyclic decoders. In future work, I will generalize the proposed decoding algorithm to correct pair errors within half of the minimum pair distance.

Table 5.4: Methods for calculating $\sigma_{l,2}$, $\sigma_{r,2}$, $\sigma_{l,3}$, and $\sigma_{r,3}$ in Step 3-1 of proposed decoding algorithm Ⅱ

<table>
<tr><th>$W_H(\boldsymbol{S})$</th><th>Condition</th><th>$\sigma_{l,2}$</th><th>$\sigma_{r,2}$</th><th>$\sigma_{l,3}$</th><th>$\sigma_{r,3}$</th></tr>
<tr><td>6</td><td>Calculate a candidate set of $\sigma_L(x)$ and $\sigma_R(x)$.</td><td colspan="2">(5.44)</td><td>(5.40)</td><td>(5.41)</td></tr>
<tr><td rowspan="2">5</td><td rowspan="2">Calculate two candidate sets of $\sigma_L(x)$ and $\sigma_R(x)$.</td><td colspan="2">(5.47)</td><td>(5.40)</td><td>0</td></tr>
<tr><td colspan="2">(5.48)</td><td>0</td><td>(5.41)</td></tr>
<tr><td rowspan="4">4</td><td>If the coefficient matrix of (5.45) is nonsingular, calculate a candidate set of $\sigma_L(x)$ and $\sigma_R(x)$.</td><td>(5.45)</td><td>(5.26)</td><td>(5.40)</td><td>(5.41)</td></tr>
<tr><td rowspan="3">If the coefficient matrix of (5.45) is singular, calculate three candidate sets of $\sigma_L(x)$ and $\sigma_R(x)$.</td><td>(5.53)</td><td>0</td><td>(5.53)</td><td>0</td></tr>
<tr><td>0</td><td>(5.54)</td><td>0</td><td>(5.54)</td></tr>
<tr><td>(5.42)</td><td>(5.43)</td><td>0</td><td>0</td></tr>
<tr><td rowspan="6">3</td><td>If the coefficient matrix of (5.49) is nonsingular, calculate a candidate set of $\sigma_L(x)$ and $\sigma_R(x)$.</td><td>(5.49)</td><td>(5.43)</td><td>(5.49)</td><td>0</td></tr>
<tr><td>If the coefficient matrix of (5.50) is nonsingular, calculate a candidate set of $\sigma_L(x)$ and $\sigma_R(x)$.</td><td>(5.42)</td><td>(5.50)</td><td>0</td><td>(5.50)</td></tr>
<tr><td rowspan="4">If the coefficient matrices of both (5.49) and (5.50) are singular, calculate four candidate sets of $\sigma_L(x)$ and $\sigma_R(x)$.</td><td>τ_2</td><td>0</td><td>τ_3</td><td>0</td></tr>
<tr><td>0</td><td>τ_2</td><td>0</td><td>τ_3</td></tr>
<tr><td>(5.42)</td><td>0</td><td>0</td><td>0</td></tr>
<tr><td>0</td><td>(5.43)</td><td>0</td><td>0</td></tr>
<tr><td rowspan="4">2</td><td>If the coefficient matrix of (5.46) is nonsingular, calculate a candidate set of $\sigma_L(x)$ and $\sigma_R(x)$.</td><td>(5.46)</td><td>(5.26)</td><td>(5.40)</td><td>(5.41)</td></tr>
<tr><td>If coefficient matrix of (5.46) is singular and the coefficient matrix of (5.55) is nonsingular, calculate a candidate set of $\sigma_L(x)$ and $\sigma_R(x)$.</td><td>(5.55)</td><td>0</td><td>(5.55)</td><td>0</td></tr>
<tr><td>If the coefficient matrix of (5.46) is singular and the coefficient matrix of (5.56) is nonsingular, calculate a candidate set of $\sigma_L(x)$ and $\sigma_R(x)$.</td><td>0</td><td>(5.56)</td><td>0</td><td>(5.56)</td></tr>
<tr><td>If the coefficient matrix of (5.46) is singular, calculate a candidate set of $\sigma_L(x)$ and $\sigma_R(x)$.</td><td>(5.42)</td><td>(5.43)</td><td>0</td><td>0</td></tr>
<tr><td rowspan="4">1</td><td>If the coefficient matrix of (5.51) is nonsingular, calculate a candidate set of $\sigma_L(x)$ and $\sigma_R(x)$.</td><td>(5.51)</td><td>(5.43)</td><td>(5.40)</td><td>0</td></tr>
<tr><td>If the coefficient matrix of (5.52) is nonsingular, calculate a candidate set of $\sigma_L(x)$ and $\sigma_R(x)$.</td><td>(5.42)</td><td>(5.52)</td><td>0</td><td>(5.41)</td></tr>
<tr><td rowspan="2">If the coefficient matrices of both (5.51) and (5.52) are singular, calculate two candidate sets of $\sigma_L(x)$ and $\sigma_R(x)$.</td><td>(5.42)</td><td>0</td><td>0</td><td>0</td></tr>
<tr><td>0</td><td>(5.43)</td><td>0</td><td>0</td></tr>
<tr><td>0</td><td>Calculate a candidate set of $\sigma_L(x)$ and $\sigma_R(x)$.</td><td>(5.42)</td><td>(5.43)</td><td>0</td><td>0</td></tr>
</table>

Table 5.5: N in proposed decoding algorithm I

		(E_L, E_R)					
		Two-pair error				Single-pair error	
		$(2,2)$	$(2,1)$	$(2,0)$	$(1,1)$	$(1,1)$	$(1,0)$
$W_H(\boldsymbol{S})$	4	1	—	—	—	—	—
	3	—	2	—	—	—	—
	2	1	—	4	4	5	—
	1	—	2	—	—	—	4
	0	—	—	—	1	—	—

Table 5.6: N in proposed decoding algorithm II

		(E_L, E_R)											
		Three-pair error						Two-pair error				Single-pair error	
		$(3,3)$	$(3,2)$	$(3,1)$	$(2,2)$	$(3,0)$	$(2,1)$	$(2,2)$	$(2,1)$	$(2,0)$	$(1,1)$	$(1,1)$	$(1,0)$
$W_H(\boldsymbol{S})$	6	1	—	—	—	—	—	—	—	—	—	—	—
	5	—	2	—	—	—	—	—	—	—	—	—	—
	4	1	—	4	4	—	—	5	—	—	—	—	—
	3	—	2	—	—	6	6	—	8	—	—	—	—
	2	1	—	4	4	—	—	5	—	8	8	9	—
	1	—	2	—	—	—	4	—	6	—	—	—	8
	0	—	—	—	1	—	—	—	—	—	2	—	—

Chapter 6

Error-Trapping Decoding over Symbol-Pair Read Channels

6.1 Abstract

The Kasami's error-trapping decoder [13] is known as the efficient decoder for cyclic codes by hardwares. In this chapter, I discuss error-trapping decoding for cyclic codes over symbol-pair read channels. I propose a new error-trapping decoding algorithm under some restrictions on the pair error patterns that I intend to correct. It corrects all pair error patterns whose pair errors within the pair error correcting capability under the restrictions. I firstly discuss problems in the existing error-trapping decoding algorithms when it is used for cyclic codes over symbol-pair read channels. I solve this problems by using the neighbor-symbol syndrome defined in Chapter 3, and propose a new error-trapping decoding algorithm. Next, I show a circuitry that implements the proposed algorithm. Finally, I discuss modifying the restrictions on the correctable error patterns. I show necessity that I need to find covering polynomials suitable for the symbol-pair read channels, and show how to modify the restrictions by using the covering polynomials.

6.2 Error-Trapping Decoding for Cyclic Codes over Symbol-Pair Read Channels

In this section, I propose an error-trapping decoding algorithm and a decoder for cyclic codes over symbol-pair read channels.

Consider an (n, k) cyclic code with a generator polynomial $g(x)$. Let $c(x) = c_0 + c_1x + \cdots + c_{n-1}x^{n-1}$ be a code polynomial, left and right polynomials of $\pi(c(x))$ are defined as

$$c_L(x) = c_0 + c_1x + \cdots + c_{n-2}x^{n-2} + c_{n-1}x^{n-1}, \tag{6.1}$$
$$c_R(x) = c_1 + c_2x + \cdots + c_{n-1}x^{n-2} + c_0x^{n-1}. \tag{6.2}$$

I define a pair error pattern, left and right error patterns as

$$\overleftrightarrow{e}(x) = \overleftrightarrow{e}_0 + \overleftrightarrow{e}_1x + \overleftrightarrow{e}_2x^2 + \cdots + \overleftrightarrow{e}_{n-1}x^{n-1}, \tag{6.3}$$
$$e_L(x) = e_{l,0} + e_{l,1}x + \cdots + e_{l,n-2}x^{n-1} + e_{l,n-1}x^{n-1}, \tag{6.4}$$
$$e_R(x) = e_{r,1} + e_{r,2}x + \cdots + e_{r,n-1}x^{n-1} + e_{r,0}x^{n-1}. \tag{6.5}$$

Moreover, I define the received pair polynomial and the left and right received polynomials as follows:

$$\overleftrightarrow{u}(x) = \overleftrightarrow{u}_0 + \overleftrightarrow{u}_1 x + \overleftrightarrow{u}_2 x^2 + \cdots + \overleftrightarrow{u}_{n-1} x^{n-1}, \tag{6.6}$$

$$u_L(x) = u_{l,0} + u_{l,1} x + \cdots + u_{l,n-2} x^{n-1} + u_{l,n-1} x^{n-1}, \tag{6.7}$$

$$u_R(x) = u_{r,1} + u_{r,2} x + \cdots + u_{r,n-1} x^{n-1} + u_{r,0} x^{n-1}, \tag{6.8}$$

where $u_L(x) = c_L(x) + e_L(x)$ and $u_R(x) = c_R(x) + e_R(x)$. If $\overleftrightarrow{u}(x)$ is obtained, I define symbol-pair syndrome polynomial $\overleftrightarrow{s}(x) = (s_L(x), s_R(x))$ as follows:

$$s_L(x) \triangleq u_L(x) \bmod g(x), \tag{6.9}$$

$$s_R(x) \triangleq u_R(x) \bmod g(x). \tag{6.10}$$

Suppose that pair errors are confined to the $n-k$ low-order positions, namely the pair error pattern and the left and right error patterns are represented as

$$\overleftrightarrow{e}(x) = \overleftrightarrow{e}_0 + \overleftrightarrow{e}_1 x + \cdots + \overleftrightarrow{e}_{n-k-1} x^{n-k-1}, \tag{6.11}$$

$$e_L(x) = e_{l,0} + e_{l,1} x + \cdots + e_{l,n-k-1} x^{n-k-1}, \tag{6.12}$$

$$e_R(x) = e_{r,1} + e_{r,2} x + \cdots + e_{r,n-k} x^{n-k-1}. \tag{6.13}$$

From the result of dividing $u_L(x)$ and $u_R(x)$ by $g(x)$,

$$s_L(x) = u_L(x) \bmod g(x) = e_L(x), \tag{6.14}$$

$$s_R(x) = u_R(x) \bmod g(x) = e_R(x). \tag{6.15}$$

Thus, $\overleftrightarrow{e}(x)$ equals $\overleftrightarrow{s}(x)$.

Suppose that pair errors are confined to $n-k$ consecutive pairs, the pair error pattern is represented as

$$\overleftrightarrow{e}(x) = \overleftrightarrow{e}_i x^i + \overleftrightarrow{e}_{i+1} x^{i+1} + \cdots + \overleftrightarrow{e}_{(n-k)+i-1} x^{(n-k)+i-1}. \tag{6.16}$$

If $\overleftrightarrow{e}(x)$ is cyclically shifted $n-i$ times to the right, pair errors will be confined to the $n-k$ low-order positions.

$$\overleftrightarrow{e}^{(n-i)}(x) = \overleftrightarrow{e}_i + \overleftrightarrow{e}_{i+1} x + \cdots + \overleftrightarrow{e}_{(n-k)+i-1} x^{n-k-1}, \tag{6.17}$$

$$e_L^{(n-i)}(x) = e_{l,i} + e_{l,i+1} x + \cdots + e_{l,(n-k)+i-1} x^{n-k-1}, \tag{6.18}$$

$$e_R^{(n-i)}(x) = e_{r,i+1} + e_{r,i+2} x + \cdots + e_{r,(n-k)+i} x^{n-k-1}, \tag{6.19}$$

where $\overleftrightarrow{e}^{(n-i)}(x)$ is the $(n-i)$-th cyclic shift of $\overleftrightarrow{e}(x)$. From the result of dividing $u_L^{(n-i)}(x)$ and $u_R^{(n-i)}(x)$ by $g(x)$,

$$s_L{}^{(n-i)}(x) = u_L^{(n-i)}(x) \bmod g(x) = e_L^{(n-i)}(x), \tag{6.20}$$

$$s_R{}^{(n-i)}(x) = u_R^{(n-i)}(x) \bmod g(x) = e_R^{(n-i)}(x), \tag{6.21}$$

where $s_L{}^{(n-i)}(x)$ and $s_R{}^{(n-i)}(x)$ are the symbol-pair syndrome of $\overleftrightarrow{u}^{(n-i)}(x)$. Thus, $\overleftrightarrow{e}^{(n-i)}(x) = (e_L^{(n-i)}(x), e_R^{(n-i)}(x))$ equals $(s_L{}^{(n-i)}(x), s_R{}^{(n-i)}(x))$, so $\overleftrightarrow{e}(x)$ equals $x^i \overleftrightarrow{s}^{(n-i)}(x)$.

In the symbol-pair read channels, if assuming that a pair error pattern equals the symbol-pair syndrome, I have the following problem. The pair error patterns cannot be distinguished by using the symbol-pair syndromes when t_p-pair error-correcting code is used. The reason is that there is a plurality of error patterns that they have t_p or fewer pair errors and their symbol-pair syndromes are the same.

Example 6. Suppose that the $(7,4)$ cyclic Hamming code which can correct up to two-pair errors is used and a symbol-pair syndrome is $\overleftrightarrow{s}(x) = (1,1) + (1,1)x$. There are some error patterns that they have two or fewer pair errors confined to $n-k$ consecutive pairs with the symbol pair syndromes $\overleftrightarrow{s}(x) = (1,1) + (1,1)x$. These error patterns are $\overleftrightarrow{e}(x) = (1,1)x^3$, $\overleftrightarrow{e}'(x) = (1,1) + (1,1)x$ and $\overleftrightarrow{e}''(x) = (1,1)x^4 + (1,1)x^6$. ■

Thus, these pair error patterns cannot be distinguished by using the symbol-pair syndromes.

To distinguish those pair error patterns, I use neighbor-symbol syndrome which represents the conflict positions defined by (3.4) and (3.5). If the received pair vector is given by (6.6), the neighbor-symbol syndrome is defined as follows:

$$S(x) = u_L(x) + (xu_R(x) \bmod x^n - 1). \tag{6.22}$$

By using the symbol-pair syndrome and the neighbor-symbol syndrome to determined the pair error pattern, I have the following theorem.

Theorem 10. Suppose that the number of pair errors in $\overleftrightarrow{u}(x)$ is t_p or less and they are confined to $n-k$ consecutive pairs. The pair error pattern is uniquely determined by the symbol-pair syndrome $\overleftrightarrow{s}(x)$ if $\overleftrightarrow{s}(x)$ satisfies the following conditions.

Condition 1. The pair weight of $\overleftrightarrow{s}(x)$ is t_p or less.

Condition 2. The conflict positions of $\overleftrightarrow{s}(x)$ equal the $n-k$ low-order positions of $S(x)$ calculated from $\overleftrightarrow{u}(x)$.

Proof: An pair error pattern $\overleftrightarrow{e}(x) = (e_L(x), e_R(x))$ with t_p or fewer pair errors that are confined to $n-k$ consecutive pairs is represented as $\overleftrightarrow{e}(x) = x^j \overleftrightarrow{b}(x) = (x^j b_L(x), x^j b_R(x))$, where $\overleftrightarrow{b}(x)$ has t_p or fewer terms and has degree $n-k-1$ or less. From the result of dividing the left and right error patterns by the generator polynomial $g(x)$,

$$x^j b_L(x) = a_L(x)g(x) + s_L(x), \tag{6.23}$$
$$x^j b_R(x) = a_R(x)g(x) + s_R(x). \tag{6.24}$$

Because $x^j b_L(x) + s_L(x)$ and $x^j b_R(x) + s_R(x)$ are multiples of $g(x)$, they are code polynomials in the Hamming metric. In addition, if the symbol-pair syndrome $\overleftrightarrow{s}(x) = (s_L(x), s_R(x))$ satisfies Condition 2, namely, the conflict positions of $\overleftrightarrow{s}(x)$ equal the $n-k$ low-order positions of $S(x)$ calculated from $\overleftrightarrow{u}(x)$, the neighbor-symbol syndrome of $x^j \overleftrightarrow{b}(x) + \overleftrightarrow{s}(x)$ is zero. Thus, $x^j \overleftrightarrow{b}(x) + \overleftrightarrow{s}(x)$ is a code polynomial over symbol-pair read channels. The symbol-pair syndrome $\overleftrightarrow{s}(x)$ cannot satisfy Condition 1 unless $x^j \overleftrightarrow{b}(x) = \overleftrightarrow{s}(x)$. Suppose that the pair weight of $\overleftrightarrow{s}(x)$ is t_p or less, and $x^j \overleftrightarrow{b}(x) \neq \overleftrightarrow{s}(x)$. Thus, $x^j \overleftrightarrow{b}(x) + \overleftrightarrow{s}(x)$ is a nonzero code polynomial with the pair weight less than $2t_p + 1$ since the pair weight of $\overleftrightarrow{b}(x)$ is t_p or less. This is impossible, since a t_p-pair error-correcting code must have the minimum pair weight of at least $2t_p + 1$. Thus, I conclude that the pair error pattern is determined uniquely if $\overleftrightarrow{s}(x)$ satisfies Conditions 1 and 2. □

From theorem 10, I propose an error-trapping decoding algorithm for cyclic codes over symbol-pair read channels.

Proposed error-trapping decoding algorithm

Input: Received polynomial $\overleftrightarrow{u}(x) = (u_L(x), u_R(x))$.

Output: Corrected code polynomial $\hat{c}(x)$ or failure symbol F.

Step 1: Calculate $\overleftrightarrow{s}(x) = (s_L(x), s_R(x))$ and $S(x)$.

$$\begin{aligned} s_L(x) &:= u_L(x) \bmod g(x), \\ s_R(x) &:= u_R(x) \bmod g(x), \\ S(x) &:= u_L(x) + (x u_R(x) \bmod x^n - 1). \end{aligned}$$

Step 2: Examine $\overleftrightarrow{s}(x)$ whether it satisfies Conditions 1 and 2. If $\overleftrightarrow{s}(x)$ satisfies both of Conditions 1 and 2, go to Step 5. Otherwise, set $j = 0$ and go to Step 3.

Step 3: If $j < n$, cyclically shift $\overleftrightarrow{u}(x)$ to the right, and calculate $\overleftrightarrow{s}^{(j+1)}(x) = ({s_L}^{(j+1)}(x), {s_R}^{(j+1)}(x))$ and $S^{(j+1)}(x)$.

$$\begin{aligned} \overleftrightarrow{u}^{(j+1)}(x) &:= x\overleftrightarrow{u}^{(j)}(x), \\ {s_L}^{(j+1)}(x) &:= u_L^{(j+1)}(x) \bmod g(x), \\ {s_R}^{(j+1)}(x) &:= u_R^{(j+1)}(x) \bmod g(x), \\ S^{(j+1)}(x) &:= u_L^{(j+1)}(x) + (x u_R^{(j+1)}(x) \bmod x^n - 1). \end{aligned}$$

Note that $\overleftrightarrow{u}^{(0)}(x) = \overleftrightarrow{u}(x)$ when $j = 0$. If $j = n$, output failure symbol F and terminate this algorithm.

Step 4: Examine $\overleftrightarrow{s}^{(j+1)}(x)$ whether it satisfies Conditions 1 and 2. If $\overleftrightarrow{s}^{(j+1)}(x)$ satisfies both of Conditions 1 and 2, go to Step 5. Otherwise, set $j = j + 1$ and return to Step 3.

Step 5: Calculate $\pi(\hat{c}(x)) := \overleftrightarrow{u}(x) + x^{n-j}\overleftrightarrow{s}^{(j)}(x)$. Output $\hat{c}(x)$ and terminate this algorithm. ■

Based on the proposed decoding algorithm, I construct an error-trapping decoder over symbol-pair read channels. The schematic circuitry is shown in Fig. 6.1. In the each circuitry, $\oplus$ represents XOR operation. I show elements for constructing the decoder.

The neighbor-symbol syndrome calculation circuit is implemented as shown in Fig. 6.2. The neighbor-symbol syndrome is calculated as follows.

Step 1. Input each $n - 1$ bits from the first bit of the left and right received polynomials $u_L(x)$ and $u_R(x)$ with gate 1 turned on and gate 2 turned off.

Step 2. Input each n-th bit of $u_L(x)$ and $u_R(x)$ with gate 1 turned off and gate 2 turned on.

The symbol-pair syndrome calculation circuit and the threshold gate 1 are implemented as shown in Fig. 6.3. The symbol-pair syndrome calculation circuit is constructed by two syndrome registers in Hamming metric. The threshold gate 1 examines Condition 1. The pair weight of the symbol-pair syndrome is tested by the threshold gate 1 whose output is 1 when t_p or fewer of its inputs are 1; otherwise, it is zero. Note that inputs of threshold gate 1 are calculated OR operation of corresponding bits of two syndrome registers.

The threshold gate 2 is implemented as shown in Fig. 6.4. Condition 2 is tested by

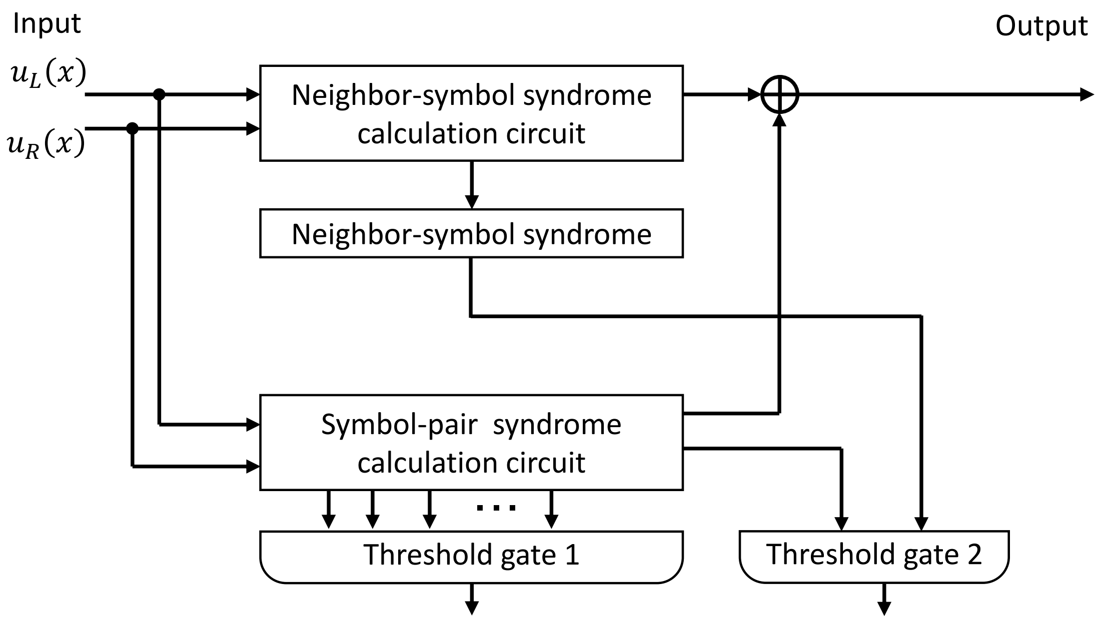

Figure 6.1: Schema of error-trapping decoder for symbol-pair read channels

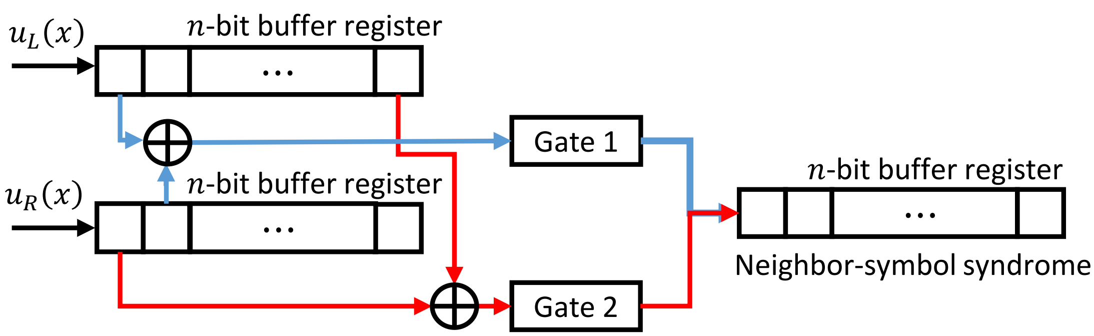

Figure 6.2: Neighbor-symbol syndrome calculation circuit

the threshold gate 2 whose output is 1 when all of its inputs are 0; otherwise, it is zero. Note that inputs of threshold gate 2 are calculated XOR operation of corresponding bits of two syndrome registers and the neighbor-symbol syndrome.

An error-trapping decoder for the (7,4) cyclic code over the symbol-pair syndrome is implemented as shown in Fig. 6.5. The decoding algorithm is described in the following steps:

Step 1. Input $n-1$ bits from the first bit of the left and right received polynomials $u_L(x)$ and $u_R(x)$ with gates 1, 2 and 3 turned on and all the other gates turned off. Next, input n-th bits of $u_L(x)$ and $u_R(x)$ with gates 1, 2 and 4 turned on and all the other gates turned off.

Step 2. Examine Conditions 1 and 2 by the threshold gates 1 and 2, respectively. If both of outputs of threshold gates 1 and 2 are 1, go to Step 5. Otherwise, go to Step 3.

Step 3. Cyclically shift each circuit with gates 5, 6 and 7 turned on and all the

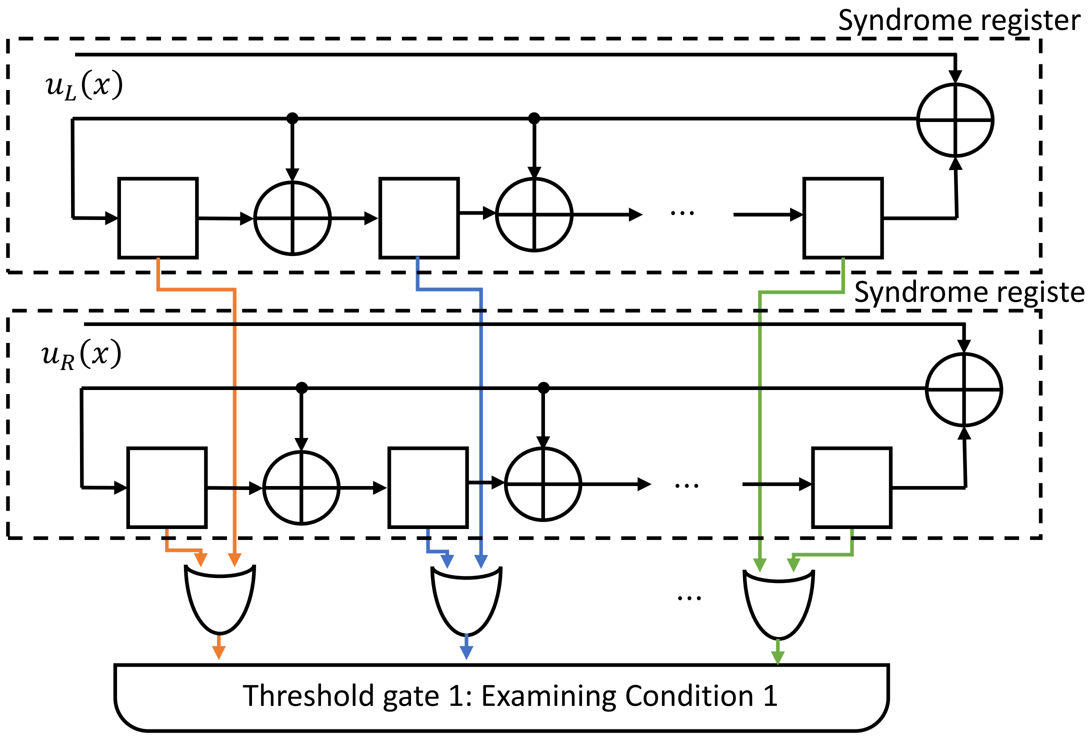

Figure 6.3: Symbol-pair syndrome calculation circuit and Threshold gate 1

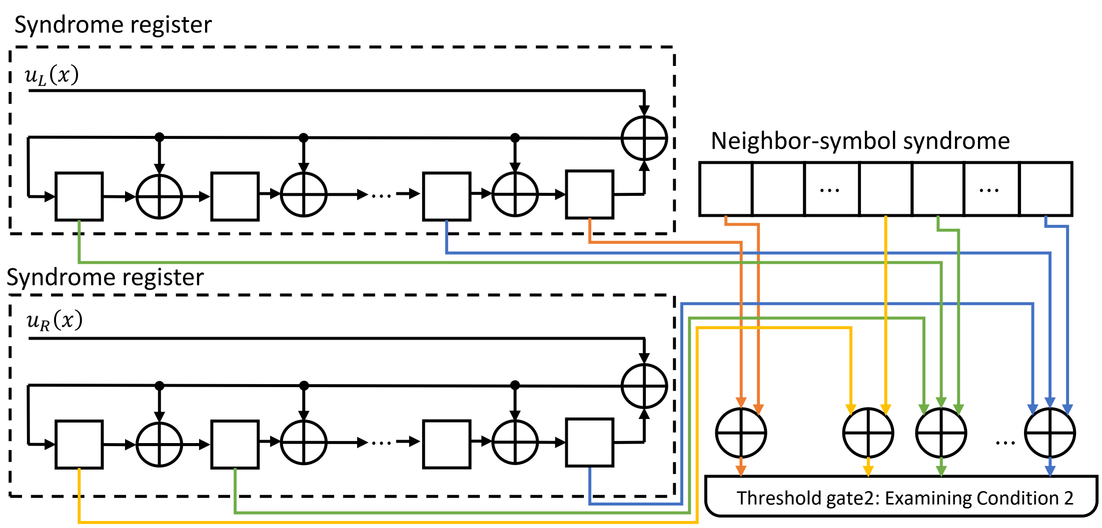

Figure 6.4: Threshold gate 2

other gates turned off.

Step 4. Examine Conditions 1 and 2 by the threshold gates 1 and 2, respectively. If both of outputs of threshold gates 1 and 2 are 1, go to Step 5. Otherwise, return to Step 3. If Step 3 is repeated $n-1$ times, the decoding is a failure and terminate this algorithm.

Step 5. For the number of times that repeated Step 3, output the left received

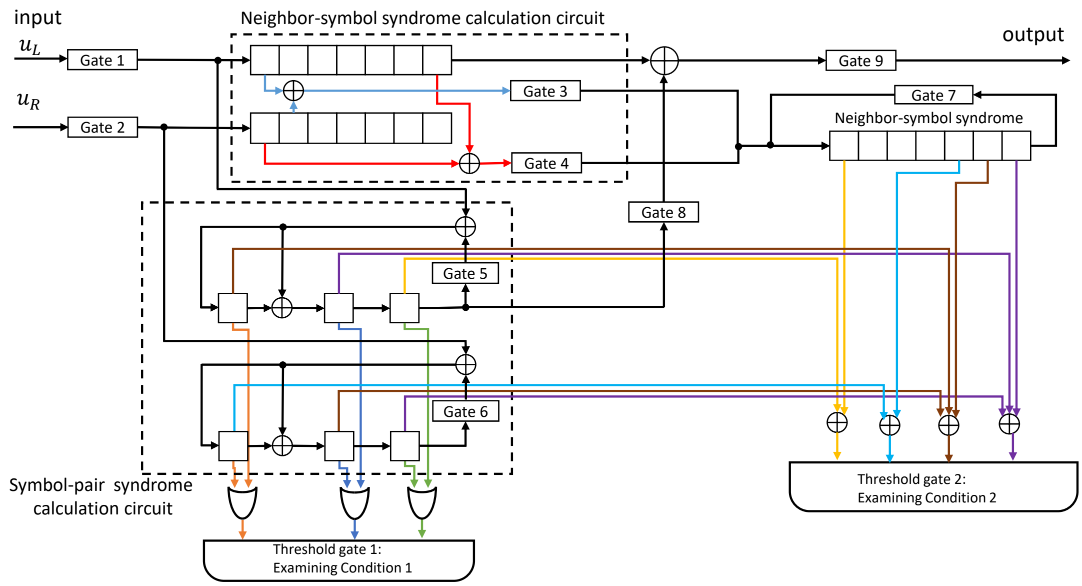

Figure 6.5: Error-trapping decoder for (7,4) cyclic Hamming code over symbol-pair read channels

polynomial with gate 9 turned on and all the other gates turned off. Next, gate 8 turned on, output the result of XOR of the left received polynomial and the left of the symbol-pair syndrome. After output n times, terminate this algorithm. ■

I compare the error correction performance of the proposed error-trapping decoder with Yaakobi's decoder [6, 7]. Yaakobi's decoder can correct all pair errors within $t_0 = \lfloor (3t_H + 1)/2 \rfloor$, where $t_H = \lfloor (d_H - 1)/2 \rfloor$ and d_H is the minimum Hamming distance of codes. The proposed decoder can correct all t_p pair errors under the condition that the errors are confined to $n - k$ consecutive pairs. Since the number of error patterns with t_p pair errors under such a condition is less than that with t_0 pair errors, the proposed decoder correct fewer pair errors than Yaakobi's decoder. However, it has shown that there are codes whose pair error correcting capability t_p is greater than t_0 [9], so Yaakobi's decoder cannot correct all pair errors within t_p. On the other hand, I can reduce the restriction of the proposed decoder by a few additional circuits. As the result, the decoder can correct pair errors outside an $(n - k)$-pair span. I show the method in the next section.

6.3 Improved Error-Trapping Decoding over Symbol-Pair Read Channels

The proposed error-trapping decoding over symbol-pair read channels discussed in Section 6.2 corrects the error patterns with t_p or fewer pair errors and all of them are confined to $n - k$ consecutive pairs. In this section, I reduce the restriction that all of the pair errors are confined to $n - k$ consecutive pairs. I propose an improved error-trapping decoding algorithm that corrects the error patterns whose most pair errors are confined to $n-k$ consecutive pairs and fewer pair errors are outside an $(n-k)$-pair span. By calculating the symbol-pair syndrome of pair errors outside the $(n - k)$-pair span in advance, I can only consider the symbol-pair syndrome of pair errors confined to $n - k$

Table 6.1: The result of Step 2 in the example

$\overleftrightarrow{\phi}_j(x)$	$W_p(\overleftrightarrow{s}(x)+\overleftrightarrow{\rho}_j(x))$	$t_p-W_p(\overleftrightarrow{\phi}_j(x))$	Condition 1'	$S(x)$ of $(\overleftrightarrow{s}(x)+\overleftrightarrow{\rho}_j(x)+x^3\overleftrightarrow{\phi}_j(x))$	Condition 2'
$\overleftrightarrow{\phi}_1(x)$	2	2	✓	$1+x+x^3$	×
$\overleftrightarrow{\phi}_2(x)$	2	1	×	-	-
$\overleftrightarrow{\phi}_3(x)$	1	1	✓	0	×
$\overleftrightarrow{\phi}_4(x)$	1	1	✓	$1+x^2+x^6$	✓

consecutive pairs. The pair error pattern $\overleftrightarrow{e}(x) = \overleftrightarrow{e}_0 + \overleftrightarrow{e}_1 x + \cdots + \overleftrightarrow{e}_{n-1}x^{n-1}$ can be divided into two parts:

$$\overleftrightarrow{e}_P(x) = \overleftrightarrow{e}_0 + \overleftrightarrow{e}_1 x + \cdots + \overleftrightarrow{e}_{n-k-1}x^{n-k-1}, \tag{6.25}$$

$$\overleftrightarrow{e}_I(x) = \overleftrightarrow{e}_{n-k}x^{n-k} + \cdots + \overleftrightarrow{e}_{n-1}x^{n-1}, \tag{6.26}$$

I consider that $\overleftrightarrow{e}_P(x)$ is the pair error pattern whose pair errors are confined to $n-k$ consecutive pairs and $\overleftrightarrow{e}_I(x)$ is the pair error pattern whose pair errors outside the $(n-k)$-pair span. From the result of dividing $\overleftrightarrow{e}_I(x)$ by the generator polynomial $g(x)$,

$$\overleftrightarrow{e}_I(x) = \overleftrightarrow{q}(x)g(x) + \overleftrightarrow{\rho}(x), \tag{6.27}$$

where $\overleftrightarrow{\rho}(x)$ is the remainder with degree $n-k-1$ or less. By adding $\overleftrightarrow{e}_P(x)$,

$$\overleftrightarrow{e} = \overleftrightarrow{e}_P(x) + \overleftrightarrow{e}_I(x) = \overleftrightarrow{q}(x)g(x) + \overleftrightarrow{\rho}(x) + \overleftrightarrow{e}_P(x) \tag{6.28}$$

Because $\overleftrightarrow{e}_P(x)$ has degree $n-k-1$ or less, $\overleftrightarrow{\rho}(x) + \overleftrightarrow{e}_P(x)$ must be the remainder resulting from dividing the pair error pattern $\overleftrightarrow{e}(x)$ by $g(x)$. Thus, symbol-pair syndrome is

$$\overleftrightarrow{s}(x) = \overleftrightarrow{\rho}(x) + \overleftrightarrow{e}_P(x). \tag{6.29}$$

By transforming (6.29)

$$\overleftrightarrow{e}_P(x) = \overleftrightarrow{\rho}(x) + \overleftrightarrow{s}(x). \tag{6.30}$$

Thus, if the error pattern $\overleftrightarrow{e}_I(x)$ is known, the error pattern $\overleftrightarrow{e}_P(x)$ can be found. However, I cannot know the error pattern $\overleftrightarrow{e}_I(x)$ and I also cannot calculate the symbol-pair syndrome of $\overleftrightarrow{e}_I(x)$. I find a set of polynomials have degree $k-1$ or less such that, for any correctable error pattern $\overleftrightarrow{e}(x)$, there is one polynomial $\overleftrightarrow{\phi}_j(x)$ such that $x^{n-k}\overleftrightarrow{\phi}_j(x)$ matches $\overleftrightarrow{e}_I(x)$ or a cyclic shift of $\overleftrightarrow{e}_I(x)$. The polynomials are called covering polynomials. Over the symbol-pair read channels, since the coefficient of each term of $\overleftrightarrow{e}(x)$ is (1,0), (0,1) or (1,1), I have to find the covering polynomial for any combination coefficient of each term of $\overleftrightarrow{e}(x)$. Let $\overleftrightarrow{\rho}_j(x)$ be the remainder resulting from dividing $x^{n-k}\overleftrightarrow{\phi}_j(x)$ by $g(x)$. For this improvement, I modify Conditions 1 and 2 as follows.

> Condition 1'. For either of $j = 0, 1, 2, \ldots,$ the pair weight of $\overleftrightarrow{s}(x) + \overleftrightarrow{\rho}_j(x)$ is $t_p - W_p(\overleftrightarrow{\phi}(x))$ or less.

Condition 2'. For j satisfying Conditions 1', The conflict positions of $\overleftrightarrow{s}(x) + \overleftrightarrow{\rho}_j(x) + x^{n-k}\overleftrightarrow{\phi}_j(x)$ equals the $n-k$ low-order positions of $S(x)$ of $\overleftrightarrow{u}(x)$

If $\overleftrightarrow{s}(x)$ satisfies Conditions 1' and 2', $\overleftrightarrow{e}(x)$ equals $\overleftrightarrow{s}(x)+\overleftrightarrow{\rho}_j(x)+x^{n-k}\overleftrightarrow{\phi}_j(x)$. This improvement needs additional circuitry in the threshold gates 1 and 2. The complexity of the additional circuitry depends on how many pair errors outside an $(n-k)$ pair span are to be corrected.

I show the difference from the decoding algorithm discussed in Section 6.2 by the next example.

Example 7. Suppose the binary (7,4) cyclic Hamming code whose the primitive polynomial and the generator polynomial $p(x) = g(x) = 1 + x + x^3$. The code corrects up to two pair errors. The decoding algorithm discussed in Section 6.2 corrects only pair error patterns whose pair errors are confined to $n-k$ consecutive pairs. The code corrects any error patterns that have two or fewer pair errors if I consider correcting single pair error outside an $(n-k)$-pair span by the improvement discussed in this section. In this example, the set of covering polynomials is chosen as follows: $\{\overleftrightarrow{\phi}_1(x) = (0,0)$, $\overleftrightarrow{\phi}_2(x) = (1,0)x^3$, $\overleftrightarrow{\phi}_3(x) = (0,1)x^3$, $\overleftrightarrow{\phi}_4(x) = (1,1)x^3\}$. Thus, I have $\overleftrightarrow{\rho}_1(x) = (0,0)$, $\overleftrightarrow{\rho}_2(x) = (1,0)+(1,0)x^2$, $\overleftrightarrow{\rho}_3(x) = (0,1)+(0,1)x^2$ and $\overleftrightarrow{\rho}_4(x) = (1,1)+(1,1)x^2$. Let $\overleftrightarrow{u}(x) = (1,0)x^2 + (1,1)x^6$ be a received pair polynomial that has two pair errors, when the all-zero code polynomial $c(x) = 0$ is read. In Step 1, $\overleftrightarrow{s}(x)$ and $S(x)$ are calculated.

$$\begin{aligned}
s_L(x) &= x^2 + x^6 \bmod g(x) = 1,\\
s_R(x) &= x^6 \bmod g(x) = 1 + x^2,\\
S(x) &= 1 + x^2 + x^6.
\end{aligned}$$

Thus, $\overleftrightarrow{s}(x) = (1,1) + (0,1)x^2$. In Step 2, $\overleftrightarrow{s}(x)$ is examined whether it satisfies Conditions 1' and 2'. The results are presented in Table. 6.1. From the results, $\overleftrightarrow{s}(x)$ is satisfies Conditions 1' and 2' in the case of $\overleftrightarrow{\phi}_4(x)$, so the algorithm go to Step 5. In Step 5, $\pi(\hat{c}(x)) = \overleftrightarrow{u}(x) + \overleftrightarrow{s}(x) + \overleftrightarrow{\rho}_4(x) + x^3\overleftrightarrow{\phi}_4(x) = (0,0)$, so $\hat{c}(x) = 0$ is output. Thus, this error correction is successful. ■

In this study, I tentatively choose the set of covering polynomials by a brute force method. To find the set of covering polynomials for a specific code is not an easy problem. In the Hamming metric, several methods for finding the set can be found [2]. I need to give the method to find the set of covering polynomial for the symbol-pair read channels. That will probably be a future problem.

6.4 Conclusion

I have discussed the error-trapping decoding for cyclic codes over symbol-pair read channels. By using the two kinds of syndromes for symbol-pair read channels, I have proposed a new error-trapping decoding algorithm that puts restrictions on the correctable pair error patterns. I have shown a circuitry that implements proposed error-trapping decoding for the $(7,4)$ cyclic Hamming code. In addition, I have shown how to modify the restrictions on the $(7,4)$ cyclic Hamming code. By the improvement, the code can correct all pair error patterns whose pair errors are within the correction capability.

Chapter 7

Conclusion of This Study

In this thesis, I have studied error-correcting codes over symbol-pair read channels, which are channels for reading data accurately from a magnetic recoding media. I have discussed how to correct pair errors by using the conventional error-correcting codes. The existing decoding algorithms [3, 6] cannot correct all error patterns within the pair error correcting capability.

In Chapter 3, I newly defined the syndrome of symbol-pair codes and proved a one-to-one relationship between the syndrome and the error pattern within the correcting capability. This research result is useful for the study of symbol-pair codes by other researchers. In particular, based on the results obtained by the syndrome decoding method, Horii et al. discuss the linear programming decoding of binary linear codes [15, 16]. Moreover, Kasai et al. propose a symbol-tuple error channels extending the symbol-pair read channels and discuss the LDPC codes over the channels. They propose the iterative decoding method for the LDPC coded over the channels.

In Chapter 4, I proposed decoding method to solve the problem of space computational complexity of the syndrome decoding method. I discussed the relation between the syndromes and the pair error pattern. As a result, I proposed a new decoding method by using the output of the decoder of cyclic codes. However, although this decoding method improves the error correcting ability compared to the Yaakobi's algorithm, it does not always correct all pair errors within the pair error correctability. This is due to the fact that the pair error decoding problem is dropped into the conventional error decoding algorithm in order to utilize the conventional decoder.

In Chapter 5, I discussed the decoding pair errors only by algebraic calculation from syndromes. I newly defined error-locator polynomial and conflict-locator polynomial and derived the relation between the these polynomials. By solving the relational equations algebraically, the pair errors can be corrected. However, due to the complexity of the relational equations, the proposed decoding method is limited to three pair errors correction. It is considered that the complexity of the relation is caused by assuming that the conventional BCH codes are read over symbol-pair read channels. I consider that it is important to analyze algebraic structure for decoding and design algebraic codes suitable for symbol-pair read channels.

In Chapter 6, I assumed to incorporate a decoding circuit to hardware, and also discuss how to correct pair errors with registers and a few logic circuits. Based on the idea of the error-trapping decoding method known as a decoding method of cyclic codes by using shift registers, I proposed a decoding method by trapping pair errors and designed a circuit to realize it.

References

[1] W. W. Peterson and E. J. Weldon, Jr., Error-Correcting Codes, Second Ed., MIT Press, Cambridge, 1972.

[2] S. Lin and D. J. Costello, Jr. Error Control Coding, Second Ed.,Prentice-Hall, Englewood Cliffs, NJ, 2004.

[3] Y. Cassuto and M. Blaum, “Codes for symbol-pair read channels,” IEEE Trans. Inform. Theory, vol.57, no.12, pp.8011–8020, Dec. 2011.

[4] Y. Cassuto and M. Blaum, “Codes for symbol-pair read channels,” Proc. 2010 IEEE Intl. Symp. on Inform. Theory, Austin, Texas, pp. 988–992, June 2010.

[5] Y. Cassuto and S. Litsyn, “Symbol-pair codes: Algebraic constructions and asymptotic bounds,” Proc. IEEE Int. Symp. on Inf. Theory, St. Petersburg, Russia, pp. 2348–2352, Jul. 2011.

[6] E. Yaakobi, J Bruck, and P.H. Siegel, “Decoding of Cyclic codes over symbol-pair read channels,” Proc. IEEE Intl. Symp. Inform. Theory, Cambridge, MA, pp.2891–2895, Jul. 2012.

[7] E. Yaakobi, J. Bruck and P. H. Siegel, “Constructions and decoding of cyclic codes over b-symbol read channels,” IEEE Trans. Inf. Theory, vol.62, no.4, pp.1541–1551, Apr. 2016.

[8] Y. M. Chee, L. Ji, H. Kiah, C. Wang, J. Yin, “Maximum distance separable codes for symbol-pair read channels,” IEEE Trans. Inf. Theory, vol.59, no.11, pp.7259–7267, Nov. 2013.

[9] T. Sugimoto, M. Hirotomo, and M. Morii, “On computing the minimum pair distance of symbol-Pair codes,” Proc. the 36th Symp. Inf. Theory and its Applications, pp1–6, Nov. 2013. (in Japanese)

[10] Y. Sugiyama, M. Kasahara, M. Hirasawa, and T. Namekawa “A method for solving key equation for decoding Goppa codes,” Inf. Control. ,vol.27, pp.87–99, Jan. 1975.

[11] J.L. Massey, “Shift register synthesis and BCH decoding,” IEEE Trans. Inf. Theory, vol.15, pp.122–127, Jan. 1969.

[12] F. Polkinghorn, “Decoding of double and triple error correcting bose-chaudhuri coes,” IEEE Trans. Inf. Theory, vol.12, pp.480–481, Oct. 1966

[13] T. Kasami, “A decoding method for multiple-error-correcting cyclic codes by using threshold logics,” Conv. Rec. Inf. Process. Soc. Jap., Tokyo, November, 1961. (in Japanese)

[14] T. Kasami, “A decoding procedure for multiple-error-correcting cyclic codes,” IEEE Trans. Inf. Theory, vol.10, no.2, pp.134–139, April. 1964.

[15] S. Horii, T. Matsushima and S. Hirasawa, “Linear programming decoding of binary linear codes for symbol-pair read channels,” Proc. 2016 IEEE Int. Symp. Information Theory (ISIT 2016), pp.1944–1948, Jul. 2016.

[16] S. Horii, T. Matsushima and S. Hirasawa, “Linear Programming Decoding of Binary Linear Codes for Symbol-Pair Read Channel,” IEICE Trans. Fundamentals, vol.EA99-A, no.12, pp.2170-2178,Dec. 2016.

[17] K. Kasai, M. Hirotomo and M. Morii, “Coding symbol-tuple error channels by LDPC codes,” Proc. Symp. on Inf. Theory and its Applications (SITA2015), pp.253–257, Nov. 2015. (in Japanese)

[18] K. Kasai, M. Hirotomo and M. Morii, “LDPC codes and iterative decoding over symbol-tuple error channels,” Proc. 9th Int. Symp. on Turbo codes, pp.276–279, Sept. 2016.

[19] M. Takita, M. Hirotomo and M. Morii, “Syndrome decoding of symbol-pair codes,” IEICE Trans. Fundamentals, vol.EA98-A, no.12, pp.2423–2428, Dec. 2015.

[20] M. Takita, M. Hirotomo, and M. Morii,“A Decoding Algorithm for Cyclic Codes over Symbol-Pair Read Channels,” IEICE Trans. Fundamentals, vol.EA98-A, no.12, pp.2415–2422, 2015.

[21] M. Takita, M. Hirotomo and M. Morii, “Algebraic Decoding of BCH Codes over Symbol-Pair Read Channels: Cases of Two-Pair and Three-Pair Error Correction,” IEICE Trans. Fundamentals, vol.EA99-A, no.12, pp.2179–2191,Dec. 2016.

[22] M. Takita, M. Hirotomo and M. Morii,“Error-Trapping Decoding for Cyclic Codes over Symbol-Pair Read Channels,” Proc. 2016 Int. Symp. Information Theory and its Applications (ISITA2016), pp.717–721, Nov. 2016.

List of Publications

Journal and Transactions

1. M. Takita, M. Hirotomo and M. Morii, “Syndrome decoding of symbol-pair codes,” IEICE Trans. Fundamentals, vol.EA98-A, no.12, pp.2423–2428, Dec. 2015.

2. M. Takita, M. Hirotomo, and M. Morii,“A Decoding Algorithm for Cyclic Codes over Symbol-Pair Read Channels,” IEICE Trans. Fundamentals, vol.EA98-A, no.12, pp.2415–2422, 2015.

3. M. Takita, M. Hirotomo and M. Morii, “Algebraic Decoding of BCH Codes over Symbol-Pair Read Channels: Cases of Two-Pair and Three-Pair Error Correction,” IEICE Trans. Fundamentals, vol.EA99-A, no.12, pp.2179–2191,Dec. 2016.

4. M. Takita, M. Hirotomo, and M. Morii,“Error-Trapping Decoding for Cyclic Codes over Symbol-Pair Read Channels,” IEICE Trans. Fundamentals, vol.EA100-A, no.12, pp.2578–2584, 2017.

International Conference

1. M. Hirotomo, M. Takita, and M. Morii, “Syndrome Decoding of Symbol-Pair Codes,”Proc. IEEE Information Theory Workshop (ITW2014), pp.162–166, 2014.

2. M. Takita, M. Hirotomo, and M. Morii, “Error-Trapping Decoding for Cyclic Codes over Symbol-Pair Read Channels,” Proc. The International Symposium on Information Theory and Its Applications (ISITA2016), pp.717–721, 2016.

Symposiums and Technical Committees

1. M. Takita, M. Hirotomo, and M. Morii, “Syndrome Decoding of Symbol-Pair Codes,” IEICE Tech. Rep., vol.113, no.483, IT2013–67, pp.81–86, March 2014 (in Japanese).

2. M. Takita, M. Hirotomo, and M. Morii, “On Decoding Method of Cyclic Codes over Symbol-Pair Read Channels,” IEICE Tech. Rep., vol.114, no.224, IT2014–42, pp.7–12, Sept. 2014. (in Japanese).

3. M. Takita, M. Hirotomo, and M. Morii, “On Decoding Method of Symbol-Pair Codes Constructed from Cyclic Codes,” Proc. Symp. on Inf. Theory and its Applications (SITA2015), pp.559–564, Dec. 2014. (in Japanese)

4. M. Takita, M. Hirotomo, and M. Morii, “On Decoding Method of Symbol-Pair Codes Constructed from Cyclic Codes,” Proc. Symp. on Inf. Theory and its Applications (SITA2015), pp.559–564, Dec. 2014. (in Japanese)

5. T. Hatanaka, Y. Saji, M. Takita and M. Morii, “Camera Based Data Recognition for Enbedding Invisible Information in Images – New Type for Two-Dimensional Barcode and Its Application –,” IEICE Tech. Rep., vol.115, no.80, IA2015-14, pp.75–80, June 2015. (in Japanese).

6. M. Takita, M. Hirotomo, and M. Morii, “On the Decoding Problem of Algebraic Codes over Symbol-Pair Read Channels,” IEICE Tech. Rep., vol.115, no.214, IT2015–35, pp.7–12, Sept. 2015. (in Japanese).

7. M. Takita, M. Hirotomo, and M. Morii, “On the Decoding Problem of Algebraic Codes over Symbol-Pair Read Channels (II),” Proc. Symp. on Inf. Theory and its Applications (SITA2015), pp.258–263, Nov. 2015. (in Japanese)

8. M. Takita, M. Hirotomo, and M. Morii, “Coded Caching for Hierarchical Networks with Different Number of Layers,” Proc. Symp. on Inf. Theory and its Applications (SITA2015), pp.92–97, Dec. 2016. (in Japanese)

9. M. Takita, M. Hirotomo, and M. Morii, “Coded Caching for Hierarchical Networks with Different Number of Layers (II),” IEICE Tech. Rep., vol.117, no.208, IT2017-49, pp. 61-66, Sept. 2017. (in Japanese).

10. 土田敏生, 伊東春香, 瀧田愼, 廣友雅徳, 野村健太, 白石善明, 毛利公美, 福田洋治, 森井昌克, “LDPC符号を用いたシンドローム復号問題に基づく署名方式,” 信学技報, vol.117, no.316, ICSS2017-42, pp.23-27, 2017年11月.

11. 西澤昭人, 古本啓祐, 瀧田愼, 白石善明, 高野泰洋, 毛利公美, 森井昌克, “Named Data Networking上のRaptor 符号化によるコンテンツ配信の評価環境,” 平成29年電気関係学会関西連合大会（予稿集）, pp.284-285, 2017年11月.

12. 土田敏生, 瀧田愼, 古本啓祐, 白石善明, 高野泰洋, 毛利公美, 森井昌克, “グループ暗号通信のためのマルチホップ無線ネットワーク上での分散秘密の配布,” 平成29年電気関係学会関西連合大会（予稿集）, pp.299-300, 2017年11月

13. 近藤秀紀, 永井達也, 古本啓祐, 瀧田愼, 白石善明, 高野泰洋, 毛利公美, 森井昌克, “Windows API のログからのマルウェアの疑似復元の検討,” 平成29年電気関係学会関西連合大会（予稿集）, pp.332-333, 2017年11月

14. M. Takita, M. Hirotomo, and M. Morii, “Coded Caching Scheme in Multi-Rate Wireless Network,” Proc. Symp. on Inf. Theory and its Applications (SITA2017), pp.565–570, Nov. 2017. (in Japanese)

15. C. Yu, M. Takita, and M. Morii, “Improvement of Error Correcting Capability in QR Code by Erasure Error Correction Using Pad Codewords Space,” Proc. Symp. on Inf. Theory and its Applications (SITA2017), pp.367–372, Nov. 2017. (in Japanese)

16. H. Hatanaka, M. Takita, H. Okuma, and M. Morii, “Implementation and Evaluation of Coded Caching by Network Simulator,” IEICE Tech. Rep., vol.117, no.353, IN2017-52, pp.37–42, Dec. 2017. (in Japanese)

Invited Talk

1. 瀧田慎,“シンボルペア通信路に対する誤り訂正符号,” 誤り訂正符号のワークショップ, 2016年9月.

Doctor Thesis, Kobe University
“Decoding Algorithms of Error-Correcting Codes over Symbol-Pair Read Channels,” 67 pages
Submitted on January 19, 2018
The date of publication is printed in cover of repository version published in Kobe University Repository Kernel.