NAIST-IS-DT9861014

Doctor's Thesis

Soft-Decision Decoding Algorithms for Binary Linear Block Codes

Hitoshi Tokushige

December 2000

Department of Information Processing
Graduate School of Information Science
Nara Institute of Science and Technology

Doctor's Thesis
submitted to Graduate School of Information Science,
Nara Institute of Science and Technology
in partial fulfillment of the requirements for the degree of
DOCTOR of ENGINEERING

Hitoshi Tokushige

Thesis committee:	Hiroyuki Seki, Professor
	Masaki Koyama, Professor
	Toru Fujiwara, Professor
	Yuichi Kaji, Associate Professor

Soft-Decision Decoding Algorithms for Binary Linear Block Codes[*]

Hitoshi Tokushige

Abstract

This thesis consists of two parts. In the first part, effective solutions are presented for two basic problems related to the implementation of recursive maximum likelihood decoding (RMLD) of Reed-Muller codes. A $(64, 40)$ subcode of the third order Reed-Muller code of length 64 ($\mathrm{RM}_{3,6}$) is considered as an inner code in a concatenated coding system for NASA's high-speed satellite communications. In this system, because the error performance of the inner code is amplified by the outer code, a subcode with lower error probability is more desirable. Furthermore, the overall decoder for the $(64, 40)$ subcode of $\mathrm{RM}_{3,6}$ code consists of 32 identical RMLD decoders and each such decoder processes a $(64, 35)$ subcode or its coset parallel. The RMLD algorithm is computationally more efficient than the Viterbi decoding algorithm. However, the computational complexity of the RMLD algorithm depends on the sectionalization of a code trellis. In general, minimization of the computational complexity results in non-uniform sectionalization of a code trellis. From an implementation point of view, uniform sectionalization of a code trellis and regularity among the trellis sections are desirable.

First, we consider linear subcodes of $\mathrm{RM}_{r,m}$ whose bases are formed from the monomial basis of $\mathrm{RM}_{r,m}$ by deleting ΔK monomials of degree r where $\Delta K < \binom{m}{r}$. For such subcodes, a procedure for computing the number of minimum weight codewords is presented and we show how to delete ΔK monomials in order to obtain a subcode with the smallest number of codewords of the minimum

[*] Doctor's Thesis, Department of Information Processing, Graduate School of Information Science, Nara Institute of Science and Technology, NAIST-IS-DT9861014, December 2000.

weight. For $\Delta K \leq 3$, a formula for the number of codewords of the minimum weight is presented. For $(64, 40)$ subcodes, there are three equivalent classes. For each class, the number of minimum weight codewords, that of the second smallest weight codewords and simulation results on error probabilities of soft-decision maximum likelihood decoding are presented.

Second, we consider how to choose the $(64, 35)$ subcode of $\mathrm{RM}_{3,6}$ whose bases are formed from the monomial basis of $\mathrm{RM}_{3,6}$ by deleting seven monomials to minimize the total number of additions and comparisons in ACS(add-compare-select) procedure which roles a mainly part in RMLD.

In the second part, two new soft-decision iterative decoding algorithms are presented. Several iterative soft-decision decoding algorithms have been proposed to achieve a good error performance and a small computational complexity. In these decoding algorithms, an algebraic decoder with respect to chosen input words is iteratively used. Their performances are degraded mainly by the decoding failure of algebraic decoding and the duplication in generating candidate codewords.

We introduce "multiple GMD decoding" for binary linear block codes. In this decoding algorithm, GMD-like decoding is iterated around a few appropriately selected search centers. The original GMD decoding by Forney is a GMD-like decoding around the binary hard-decision sequence. Compared with the original GMD decoding, this decoding algorithm provides better error performance with increasing the number of iterations of erasure and error correction moderately. To reduce the number of iterations, we derive new sufficient conditions on the optimality of decoded codewords. For extended BCH codes, EBCH(64, 24), EBCH(128, 85) and EBCH(128, 99), simulation results show that the new approach provides better error performance than that of the original GMD decoding by adding two GMD-like decoding around two appropriately chosen centers to the original GMD decoding with relative small increment of iteration number.

Finally, we present a new method of choosing a sequence of search centers around which successive bounded distance-$t_0 \triangleq \lfloor (\text{minimum distance} - 1)/2 \rfloor$ decodings are carried out. To reduce the number of iterations of bounded distance decoding algorithm without any loss of error performance, we show new effective sufficient conditions on the optimality of decoded codewords as early termination

conditions.

Keywords:

minimum weight codeword, Reed-Muller Code, extended BCH Codes, subcode, recursive maximum likelihood decoding, transitive invariant code, soft-decision iterative decoding, multiple GMD decoding, early termination conditions

Acknowledgements

I am deeply indebted to many people for their assistance. I would especially like to Professor Hiroyuki Seki and Emeritus Professor Tadao Kasami for their invaluable support, advice and encouragement throughout the work.

I am also grateful to Professor Masaki Koyama of Nara Institute of Science and Technology and Professor Toru Fujiwara of Osaka University for their valuable suggestions on this work. I would like to express my gratitude to Professor Toyoo Takata of Iwate Prefecture University and Associate Professor Yuichi Kaji for their invaluable support. I would also like to thank Professors Shu Lin of the University of California at Davis and Marc P. C. Fossorier of the University of Hawaii for their helpful advice and proofreading of various drafts.

I would like to thank Associate Professor Kenichi Tomita of Shiga University and Dr. Takuya Koumoto of NTT DoCoMo for their technical support and advice.

Finally, I would like to thank all the members of foundations of information science laboratory of Nara Institute of Science and Technology, who have always given me support and encouragement.

List of Publications

Journal Papers

[1] T. Kasami, H. Tokushige, T. Fujiwara, H. Yamamoto and S. Lin, “A Recursive Maximum Likelihood Decoding Algorithm for Some Transitive Invariant Binary Codes,” *IEICE Trans. Fundamentals*, vol. E81-A, no. 9, pp. 1916–1924, Sept. 1998.

[2] H. Tokushige, T. Takata and T. Kasami, “On the Number of Minimum Weight Codewords of Subcodes of Reed-Muller Codes,” *IEICE Trans. Fundamentals*, vol. E81-A, no. 10, pp. 1990–1997, Nov. 1998.

[3] H. Tokushige, Y. Tang, T. Koumoto and T. Kasami, “An Improvement to GMD-like Decoding Algorithms,” *IEICE Trans. Fundamentals*, vol. E83-A, no. 10, pp. 1963–1965, Oct. 2000.

International Conference Papers

[1] H. Tokushige, T. Koumoto and T. Kasami, “An Improvement to GMD–like Decoding Algorithms,” *Proc. of 2000 IEEE International Symposium on Information Theory*, Sorrento, Italy, pp. 396, Jun. 2000.

[2] H. Tokushige, Y. Tang, T. Koumoto and T. Kasami, “An Improvement to Chase-like Decoding Algorithm,” *Proc. of the International Symposium on Information Theory and Its Applications*, Honolulu, USA, pp. 331–334, Nov. 2000.

Other Papers

[1] H. Tokushige, T. Takata and T. Kasami, "On the Number of Minimum Weight Codewords of a Subcode of a Reed-Muller Code," *Technical Report of IEICE*, IT97-45, pp. 25–28, Sept. 1997.

[2] H. Tokushige, T. Takata, T. Kasami and T. Fujiwara, "On Subcodes of Reed-Muller Codes," *Proc. of the Symposium on Information Theory and Its Applications*, Ehime, Japan, pp. 317–320, Dec. 1997.

[3] S. Kinuta, T. Tanoue, H. Tokushige and T. Kasami, "On the Binary Images of Shortened (8, 5) RS Codes over GF(2^8)," *Technical Report of IEICE*, IT99-36, pp. 19–24, Jul. 1999.

[4] H. Tokushige, T. Tanoue, S. Kinuta and T. Kasami, "On the Binary Images of Shortened (8, 5) RS Codes over GF(2^8)," *Proc. of the Symposium on Information Theory and Its Applications*, Yuzawa, Japan, pp. 331–334, Dec. 1999.

[5] H. Tokushige, T. Koumoto and T. Kasami, "An Impovement to GMD-like Decoding Algorithms," *Proc. of the Symposium on Information Theory and Its Applications*, Yuzawa, Japan, pp. 645–648, Dec. 1999.

[6] H. Tokushige, K. Nakamaye, T. Koumoto, Y. Tang and T. Kasami, "Selection of Search Centers in Iterative Soft-decision Decoding Algorithms," *Proc. of the Symposium on Information Theory and Its Applications*, Aso, Kumamoto, Japan, pp. 73–76, Oct. 2000.

[7] T. Kasami, H. Tokushige and Y. Kaji, "Search Procedures in Top-Down Recursive Maximum Likelihood Decoding Algorithm," *Proc. of the Symposium on Information Theory and Its Applications*, Aso, Kumamoto, Japan, pp. 535–538, Oct. 2000.

Technical Report

[1] T. Kasami, H. Tokushige, T. Fujiwara, H. Yamamoto and S. Lin, "A Recursive Maximum Likelihood Decoding Algorithm for Reed-Muller Codes and

Related Codes," *Technical Report of NAIST*, no. NAIST-IS-97003, Jan. 1997.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

The notation $\mathrm{RM}_{r,m}$ denotes the Reed-Muller code (RM code) of length 2^m and degree r. A $(64, 40)$ subcode of $\mathrm{RM}_{3,6}$ is considered as an inner code in a concatenated coding system for NASA's high-speed satellite communications [1, 3]. The inner code is decoded by a maximum likelihood decoder. Since the Reed-Solomon$(255, 223)$ code over $\mathrm{GF}(2^8)$ is used as the outer code, the number of information bits must be a multiple of 8. This is the reason why a $(64, 40)$ subcode of $\mathrm{RM}_{3,6}$ is used as the inner code. Therefore, the problem is how to choose such a subcode. In this system, because the error performance of the inner code is amplified by the outer code, a subcode with lower error probability is more desirable. The number of minimum weight codewords of the subcode is a reasonable design index.

A trellis-based recursive maximum likelihood decoding (RMLD) algorithm has been proposed [5]. This RMLD algorithm is more efficient than the conventional Viterbi decoding algorithm in both computational complexity for software implementation and hardware requirement for IC (integrated circuit) implementation. Furthermore, it allows parallel/pipeline processing to speed up the decoding process. The RMLD algorithm is devised based on a divide-and-conquer approach. A code trellis is first divided into appropriate sections. A metric table for each trellis section is formed and each table contains only the metrics of the distinct composite branches in the section and their labels. Metric tables for long trellis sections are formed recursively from tables for shorter trellis sections. At the end of the recursion process, there is only one table which contains only the most likely codeword for a given received sequence and its corresponding metric. Computational complexity of this decoding algorithm depends on the sectionalization of a code trellis. Minimization of the computational complexity in general results in non-uniform sectionalization of a code trellis in which the trellis sections are not equal in length and do not have a regular structure. These facts cause implementation problems and require more circuits in IC implementation. Therefore, in some applications, it is desirable to trade-off computational complexity for simplicity and regularity in the trellis to reduce circuit requirements and gain decoding speed.

In chapter 2, we consider how to evaluate the number of codewords of the minimum weight for linear subcodes of $\mathrm{RM}_{r,m}$ whose bases are formed from the monomial basis of $\mathrm{RM}_{r,m}$ by deleting ΔK monomials of degree r where $\Delta K < \binom{m}{r}$. For $\Delta K \leq 3$, a formula or an effective method which gives the number of codewords of the minimum weight is presented. We also show how to delete ΔK monomials in order to obtain the subcode with the smallest number of codewords of the minimum weight. For $(64, 40)$ subcodes, there are three equivalent classes, each of which consists of all equivalent codes. For each class, the number of minimum weight codewords, the number of second smallest weight codewords and simulation results on error probabilities of soft-decision maximum likelihood decoding are shown. The class with the smallest number of minimum weight codewords provides the best error performance among three equivalent classes.

In chapter 3, we study the RMLD for a class of codes that are transitive invariant. This class of codes includes RM codes, extended and permuted primitive BCH codes (EBCH codes), and their subcodes. For this class of codes, the binary uniform sectionalization of a code trellis results in a simple regular structure among the sections so that the metric table construction procedure can be applied uniformly at each decoding recursion level. The metric tables at the same recursion level have the same size and structure. This simplifies the implementation of an RMLD algorithm. Furthermore, for transitive invariant codes, the binary uniform sectionalization of a code trellis results in almost the same computational complexity as an optimum sectionalization does. This provides an excellent trade-off between computational complexity and code trellis regularity for simple implementation.

The overall decoder for the $(64, 40)$ subcode of $\mathrm{RM}_{3,6}$ code consists of 32 identical RMLD decoders based on binary sectionalization, each such decoder processes a $(64, 35)$ subcode or its coset parallel. Then, the problem is how to choose a subcode to minimize the computational complexity. Because ACS(add-compare-select) procedure roles a main part in RMLD, the total number of additions and comparisons in ACS procedure is used as an evaluation index. We consider how to choose the $(64, 35)$ subcode of $\mathrm{RM}_{3,6}$ whose bases are formed from the monomial basis of $\mathrm{RM}_{3,6}$ by deleting seven monomials to minimize the total number of additions and comparisons in ACS.

The iterative decoding algorithms, such as GMD and Chase II decoding algorithms, use an algebraic decoder iteratively with respect to successively chosen input words, called the search centers. Their performances are degraded by the decoding failure of algebraic decoding and the duplication in generating candidate codewords as simulation results show. By these facts, they do not achieve both a good error performance and a low computer complexity. We present new two iterative decoding algorithms and sufficient conditions on the optimality on the decoded codeword to avoid these drawbacks.

In chapter 4, we present "multiple GMD decoding" [11, 13] for binary linear block codes. Some improved versions of GMD decoding have been proposed [12, 17, 18] and performance analysis of GMD decoding has been presented [16]. Simulation results in [12] for several examples codes show that better error performance than for that of the improved version in [17] is provided by adding two bounded distance-t_0($\triangleq \lfloor$(the minimum distance $-$ 1)/2$\rfloor$) decodings around two appropriately chosen centers to the original GMD decoding.

In this decoding algorithm, GMD-like decoding is iterated around a few appropriately selected search centers. The original GMD decoding by Forney [15] is a GMD-like decoding around the binary hard-decision sequence. For extended BCH codes, EBCH(64, 24), EBCH(128, 85) and EBCH(128, 99), compared with the original GMD decoding, this decoding algorithm provides better error performance by moderately increasing the number of iterations of erasure and error correction. To reduce the number of iterations, several sufficient conditions on the optimality of decoded codewords have been introduced [12, 20, 21, 22]. We derive new effective sufficient conditions and show the effectiveness by simulation for EBCH(64, 24), EBCH(64, 45), EBCH(128, 78), EBCH(128, 85) and EBCH(128, 99).

In chapter 5, we present a new method of choosing the search centers of successive bounded distance-t_0 decodings for binary linear block codes. For BCH codes, BCH(63, 30), BCH(63, 45) and BCH(127, 92) codes, with the minimum distance, 13, 7 and 11, respectively, simulation results show the effectiveness of the choice of search centers. To reduce the number of iterations of bounded distance decoding algorithm without any loss of error performance, we derive new effective sufficient conditions on the optimality of decoded codewords as early termination conditions and show the effectiveness by simulation for BCH(63, 30), BCH(63, 45), BCH(127, 85) and BCH(127, 92).

Chapter 2

On the Number of Minimum Weight Codewords of Subcodes of Reed-Muller Codes

2.1 Definitions

We consider the number of minimum weight codewords in $(2^m, \sum_{i=0}^{r} \binom{m}{i} - \Delta K)$ linear subcodes of $\mathrm{RM}_{r,m}$ for $0 < \Delta K < \binom{m}{r}$. The notation $N_{\min}(C)$ denotes the number of minimum weight codewords in a subcode C. Let $x_1, x_2, \ldots, x_m$ be m variables. Let $P_{r,m}$ denote the set of binary polynomials with m variables of degree r or less and let $M_{r,m}$ denote the set of monomials in $P_{r,m}$. For a codeword of $\mathrm{RM}_{r,m}$, there is a unique polynomial in $P_{r,m}$ which represents the codeword [2, Ch.13, §3]. Hereafter, the polynomial is used in place of a codeword.

For $0 < h \leq r$, define $J_h \triangleq \{\{j_1, j_2, \ldots, j_h\} : 1 \leq j_1 < j_2 < \cdots < j_h \leq m\}$, $J_0 \triangleq \{\emptyset(\text{empty set})\}$ and $J \triangleq \bigcup_{h=0}^{r} J_h$. For α_1 and α_2 in J, let $\alpha_1 \cap \alpha_2$ denote the ordered set of those integers which are contained in both of α_1 and α_2. For $\alpha = \{j_1, j_2, \ldots, j_h\} \in J_h$ with $1 \leq h \leq r$, define m_α as the product of variables $x_{j_1}, x_{j_2}, \ldots, x_{j_h}$ and let $m_\emptyset \triangleq 1$. Polynomial $f \in P_{r,m}$ can be expressed uniquely as a linear sum of monomials in $M_{r,m}$, and for $\alpha \in J$, let $c_\alpha(f)$ denote the coefficient of m_α in the sum of f. For a set X, $|X|$ denotes the cardinality of X. For a binary $r \times n$ matrix B and $\alpha = \{j_1, j_2, \ldots, j_h\} \in J_h$ with $j_h \leq n \leq m$, let B_α denote the submatrix of B consisting of the j_1-th column, the j_2-th column, ..., the j_h-th column of B.

2.2 Minimum Weight Codewords

Let $P_{r,m,\min}$ denote the set of those polynomials in $P_{r,m}$ which represent codewords of minimum weight 2^{m-r}. Then, as shown in [2, Theorems 5, 7 and 8 in §4, Ch.13], $f \in P_{r,m,\min}$ if and only if f is of the following form :

$$f = \prod_{i=1}^{r}(a_i + \sum_{j=1}^{m} a_{ij}x_j),$$

where

$$\text{rank}\begin{pmatrix} a_{11} & \cdots & a_{1m} \\ \vdots & \ddots & \vdots \\ a_{r1} & \cdots & a_{rm} \end{pmatrix} = r. \tag{2.1}$$

Let $\Lambda_{r,m}$ denote the set of those $r \times (m+1)$ binary matrices whose submatrices consisting of the first m columns have rank r. For $A \in \Lambda_{r,m}$, define

$$p(A) \quad \triangleq \quad \prod_{i=1}^{r}(a_{i\,m+1} + \sum_{j=1}^{m} a_{ij}x_j), \tag{2.2}$$

where a_{ij} denotes the (i,j) element of A for $1 \leq i \leq r$ and $1 \leq j \leq m+1$. For two polynomials f and g in $P_{r,m}$ and $0 \leq i < r$, we write $f \equiv g$, mod $P_{i,m}$ if and only if $f + g \in P_{i,m}$. Then, $p(A)$ is uniquely expanded as follows:

$$\begin{aligned} p(A) \quad &= \quad \prod_{i=1}^{r}(a_{i\,m+1} + \sum_{j=1}^{m} a_{ij}x_j) \\ &\equiv \quad \sum_{\alpha \in J_r} \det(A_\alpha)m_\alpha, \ \text{ mod } P_{r-1,m}. \end{aligned} \tag{2.3}$$

2.3 The Number of Minimum Weight Codewords of a Subcode Spanned by Monomials

For simplicity, we consider the case where the subcode C of a RM code is spanned by monomials in $M_{r,m} \setminus \Delta M$ where $\Delta M \subseteq M_{r,m} \setminus M_{r-1,m}$ and $|\Delta M| = \Delta K$. Let

ΔJ denote the subset of J_r such that $\Delta M = \{m_\alpha : \alpha \in \Delta J\}$. The subcode C is denoted also by $C(\Delta M)$ or $C(\Delta J)$. Then, $f = p(A)$ with $A \in \Lambda_{r,m}$ is a codeword in $C(\Delta J)$, if and only if

$$\det(A_\alpha) = 0, \text{ for } \alpha \in \Delta J. \tag{2.4}$$

For different $\alpha_1, \alpha_2, \ldots, \alpha_h \in J$, define $F(\alpha_1, \alpha_2, \ldots, \alpha_h) \overset{\triangle}{=} \{f \in P_{r,m,\min} : c_{\alpha_i}(f) = 1 \text{ for } 1 \leq i \leq h\}$ and $\nu(\alpha_1, \alpha_2, \ldots, \alpha_h) \overset{\triangle}{=} |F(\alpha_1, \alpha_2, \ldots, \alpha_h)|$. Then, for ΔJ consisting of different $\alpha_1, \alpha_2, \ldots, \alpha_h$, $N_{\min}(C(\Delta J))$ can be expressed by the principle of inclusion and exclusion as follows:

$$\begin{aligned} N_{\min}(C(\Delta J)) &= 2^r \prod_{i=0}^{m-r-1} \frac{2^{m-i}-1}{2^{m-r-i}-1} - \sum_{1 \leq i \leq h} \nu(\alpha_i) + \sum_{1 \leq i_1 < i_2 \leq h} \nu(\alpha_{i_1}, \alpha_{i_2}) \\ &\quad \cdots + (-1)^s \sum_{1 \leq i_1 < i_2 < \cdots < i_s \leq h} \nu(\alpha_{i_1}, \alpha_{i_2}, \ldots, \alpha_{i_s}) \\ &\quad \cdots + (-1)^h \nu(\alpha_1, \alpha_2, \ldots, \alpha_h), \end{aligned} \tag{2.5}$$

where the first term of the right-hand side is the number of minimum weight codewords of $\mathrm{RM}_{r,m}$ [2, Ch.13, §5].

We consider how to evaluate $\nu(\alpha_1, \alpha_2, \ldots, \alpha_h)$ for $1 \leq h \leq \Delta K$. The following lemma holds.

Lemma 1 $\nu(\alpha_1, \alpha_2, \ldots, \alpha_h)$ is equal to the number of $r \times (m+1)$ binary matrices A's such that

(1) A_{α_1} is the identity matrix,

and

(2) $\det(A_{\alpha_i}) = 1$, for $2 \leq i \leq h$.

(Proof) (i) Let A be a matrix satisfying the above conditions. Then, $p(A) \in F(\alpha_1, \alpha_2, \ldots, \alpha_h)$. Let A and A' be two different matrices satisfying the above conditions. Let $a_{i,j}$ and $a'_{i,j}$ denote the (i, j) elements of A and A', respectively. Without loss of generality, let $\alpha_1 = \{1, 2, \ldots, r\}$. If the $(m+1)$-th columns of A and A' are different, then substitute 0 for variables x_j for $r < j \leq m$ in $p(A)$ and $p(A')$. Then the resulting polynomials are different polynomials $\prod_{i=1}^{r}(x_i + a_{i\,m+1})$ and $\prod_{i=1}^{r}(x_i + a'_{i\,m+1})$. If A and A' have different (i, j) elements, say $a_{ij} = 1$ and

$a'_{ij} = 0$ where $1 \leq i \leq r$ and $r < j \leq m$, then substitute 0 for variables x_t for $t \in \{i\} \cup \{r+1, r+2, \ldots, m\} \setminus \{j\}$. Then, the resulting polynomial of $p(A)$ contains monomials $x_1 \ldots x_{i-1} x_j x_{i+1} \ldots x_r$ of degree r and that of $p(A')$ contains no monomial of degree r. Hence, $p(A) \neq p(A')$.

(ii) Let B be an $r \times (m+1)$ binary matrix such that $p(B) \in F(\alpha_1, \alpha_2, \ldots, \alpha_h)$. For $1 \leq i \leq r$, let the i-th row of B be $(b_{i_1}, b_{i_2}, \ldots, b_{i_m}, b_{i_{m+1}})$. For $1 \leq i < i' \leq r$, let B' denote the matrix obtained from B by replacing the i'-th row of B with $(b_{i'_1} + b_{i_1}, b_{i'_2} + b_{i_2}, \ldots, b_{i'_m} + b_{i_m}, b_{i'_{m+1}} + b_{i_{m+1}} + 1)$. Then, $p(B') = p(B)$. By using this kind of row operation and permuting the rows, we can derive a matrix A such that $p(B) = p(A)$ and A meets the above conditions. $\triangle\triangle$

Renumbering the suffices of variables induces a permutation of the bit positions of codewords. Hence, an equivalent code is derived by the renumbering. Since equivalent codes have the same weight distribution, there is no loss of generality in assuming that $\alpha_1 = \{1, 2, \ldots, r\}$. Hereafter in this section, we assume this. For subsets H_1 and H_2 of $\{1, 2, \ldots, m+1\}$, A_{H_1,H_2} denotes the submatrix of A consisting of (i, j) elements of A where $i \in H_1$ and $j \in H_2$. For $\alpha \in J$, define $\alpha^0 \triangleq \{1, 2, \ldots, m+1\} \setminus \alpha$, $\alpha^1 = \alpha$ and $\alpha^* \triangleq \{1, 2, \ldots, m+1\}$. For a sequence $\beta = b_1 b_2 \cdots b_h$ over $\{0, 1, *\}$ with $1 \leq h \leq \Delta K$, define

$$n_\beta \triangleq \alpha_1^{b_1} \cap \alpha_2^{b_2} \cap \cdots \cap \alpha_h^{b_h}. \tag{2.6}$$

(i) When $\Delta K = 1$, from Lemma 1, $\nu(\alpha_1)$ is given as follows:

$$\nu(\alpha_1) = 2^{(m-r+1)r}. \tag{2.7}$$

From (2.5), we have that

$$N_{\min}(C) = 2^r \prod_{i=0}^{m-r-1} \frac{2^{m-i} - 1}{2^{m-r-i} - 1} - 2^{(m-r+1)r}. \tag{2.8}$$

(ii) Consider the case of $\Delta K = 2$ and $\Delta J = \{\alpha_1, \alpha_2\}$. Define $l \triangleq |\alpha_1 \cap \alpha_2|$. By renumbering variables, we can assume that $\alpha_1 = \{1, 2, \ldots, r\}$ and $\alpha_2 = \{r+1-l, r+2-l, \ldots, 2r-l\}$ where $0 \leq l < r$. Then, $\det(A_{\alpha_2}) = 1$, if and only if the $(r-l) \times (r-l)$ submatrix $A_{n_{10}, n_{01}}$ is regular (refer to Fig. 2.1). The number of such regular submatrices is given by

$$\prod_{j=0}^{r-l-1} (2^{r-l} - 2^j).$$

There is no restriction of submatrix $A_{n_{11},n_{01}}$. Then, the number of matrices A such that A_{α_1} is the identity matrix and $\det(A_{\alpha_2}) = 1$ is

$$2^{r\{m+1-(2r-l)\}} \cdot 2^{(r-l)l} \cdot \prod_{j=0}^{r-l-1} (2^{r-l} - 2^j).$$

From Lemma 1,

$$\nu(\alpha_1, \alpha_2) = 2^{r\{m+1-(2r-l)\}} \cdot 2^{(r-l)l} \cdot \prod_{j=0}^{r-l-1} (2^{r-l} - 2^j). \tag{2.9}$$

It follows from (2.5), (2.7) and (2.9) that

$$\begin{aligned} N_{\min}(C) &= 2^r \prod_{i=0}^{m-r-1} \frac{2^{m-i}-1}{2^{m-r-i}-1} - \{\nu(\alpha_1) + \nu(\alpha_2) - \nu(\alpha_1, \alpha_2)\} \\ &= 2^r \prod_{i=0}^{m-r-1} \frac{2^{m-i}-1}{2^{m-r-i}-1} - 2 \cdot 2^{(m-r+1)r} \\ &\quad + 2^{r\{m+1-(2r-l)\}} \cdot 2^{(r-l)l} \cdot \prod_{j=0}^{r-l-1} (2^{r-l} - 2^j), \end{aligned} \tag{2.10}$$

where $\Delta M = \{m_{\alpha_1}, m_{\alpha_2}\}$ and l denotes the number of integers appearing in both of α_1 and α_2. The value of $N_{\min}(C)$ in (2.10) takes the minimum, if and only if $l = 0$, that is, α_1 and α_2 are disjoint. This result is generalized in Theorem 1.

(iii) Consider the case of $\Delta K = 3$ and $\Delta J = \{\alpha_1, \alpha_2, \alpha_3\}$. By renumbering of variables, there is no loss of generality in assuming that $\alpha_1 = \{1, 2, \ldots, r\}$ and the integers in nonempty sets $n_{100}, n_{101}, n_{110}$ and n_{111} are arranged as shown in Fig 2.2, that is, for any $i_1 \in n_{100}, i_2 \in n_{101}, i_3 \in n_{110}$ and $i_4 \in n_{111}$, $1 \leq i_1 < i_2 < i_3 < i_4 \leq r$ (for an empty set, skip it). Then, since $A_{\alpha_1} (= A_{n_{1**},n_{1**}})$ is an identity matrix and therefore $A_{n_{10*},n_{11*}}$ is a zero matrix and $A_{n_{11*},n_{11*}}$ is an identity matrix, we have that $\det(A_{\alpha_2}) = 1$, if and only if

(L1) submatrix $A_{n_{10*},n_{01*}}$ is regular.

Note that $|n_{10*}| = r - |n_{11*}| = |n_{01*}|$. Similarly, $\det(A_{\alpha_3}) = 1$ if and only if

(L2) submatrix $A_{n_{1*0},n_{0*1}}$ is regular.

Note that $|n_{1*0}| = r - |n_{1*1}| = |n_{0*1}|$ (refer to Fig. 2.2). From (L1) and (L2), we see that

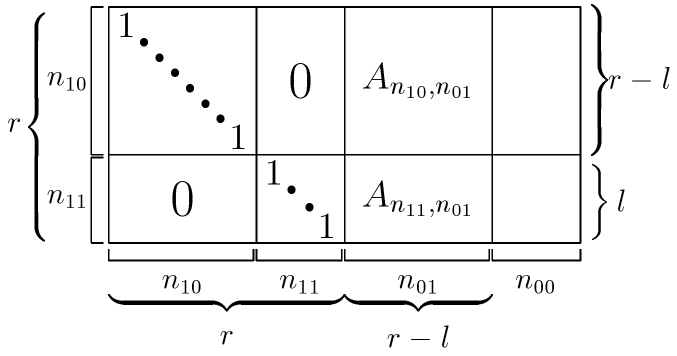

Figure 2.1. The matrix A by renumbering variables.

(L3) the columns of $A_{n_{10*},n_{011}}$ are linearly independent, and

(L4) the columns of $A_{n_{1*0},n_{011}}$ are linearly independent.

The submatrix which consists of common rows and columns of $A_{n_{10*},n_{01*}}$ and $A_{n_{1*0},n_{0*1}}$ is $A_{n_{100},n_{011}}$. Let N_{23} denote the number of pairs of $A_{n_{10*},n_{011}}$ and $A_{n_{1*0},n_{011}}$ satisfying (L3) and (L4). If one of $|n_{100}|$, $|n_{101}|$, $|n_{110}|$ and $|n_{011}|$ is zero, N_{23} can be easily counted. For the case where they are all nonzeros, how to evaluate N_{23} is shown in Appendix A.

Since the first to the $|n_{010}|$-th columns of $A_{n_{10*},n_{01*}}$ are linearly independent of the $|n_{011}|$ rest linearly independent columns of $A_{n_{10*},n_{01*}}$, the number of $A_{n_{10*},n_{010}}$ consisting of such columns is given by

$$\prod_{j=|n_{011}|}^{|n_{01*}|-1} (2^{|n_{10*}|} - 2^j) = \prod_{j=|n_{011}|}^{|n_{10*}|-1} (2^{|n_{10*}|} - 2^j). \tag{2.11}$$

Similarly, the number of $A_{n_{1*0},n_{001}}$ is given by

$$\prod_{j=|n_{011}|}^{|n_{0*1}|-1} (2^{|n_{1*0}|} - 2^j) = \prod_{j=|n_{011}|}^{|n_{1*0}|-1} (2^{|n_{1*0}|} - 2^j). \tag{2.12}$$

The elements of the $(m+1)$-th column and submatrices $A_{n_{11*},n_{010}}$, $A_{n_{111},n_{0*1}}$, $A_{n_{101},n_{001}}$ and $A_{n_{1**},n_{000}}$ are arbitrary. The total number of these elements is given by

$$2^{|n_{11*}|\cdot|n_{010}|+|n_{111}|\cdot|n_{0*1}|+|n_{101}|\cdot|n_{001}|} \times 2^{r(|n_{000}|+1)}. \tag{2.13}$$

Then, it follows from the definition of N_{23}, (2.11), (2.12) and (2.13) that the number of matrices A such that A_{α_1} is the identity matrix, $\det(A_{\alpha_2}) = 1$ and $\det(A_{\alpha_3}) = 1$ is

$$\begin{aligned} N_{23} \quad &\times \quad \prod_{j=|n_{011}|}^{|n_{10*}|-1} (2^{|n_{10*}|} - 2^j) \times \prod_{j=|n_{011}|}^{|n_{10*}|-1} (2^{|n_{1*0}|} - 2^j) \\ &\times \quad 2^{|n_{11*}|\cdot|n_{010}|+|n_{111}|\cdot|n_{0*1}|+|n_{101}|\cdot|n_{001}|} \times 2^{r(|n_{000}|+1)}, \end{aligned} \tag{2.14}$$

(refer to Fig. 2.2). From Lemma 1, $\nu(\alpha_1, \alpha_2, \alpha_3)$ is equal to the value of the formula (2.14).

For $\Delta K = 4$, the evaluation of $\nu(\alpha_1, \alpha_2, \alpha_3, \alpha_4)$ where $\alpha_1 \cap \alpha_2 \cap \alpha_3 \cap \alpha_4 \neq \emptyset$ becomes more complicated. However, the method presented in Appendix A can be applied to more general cases.

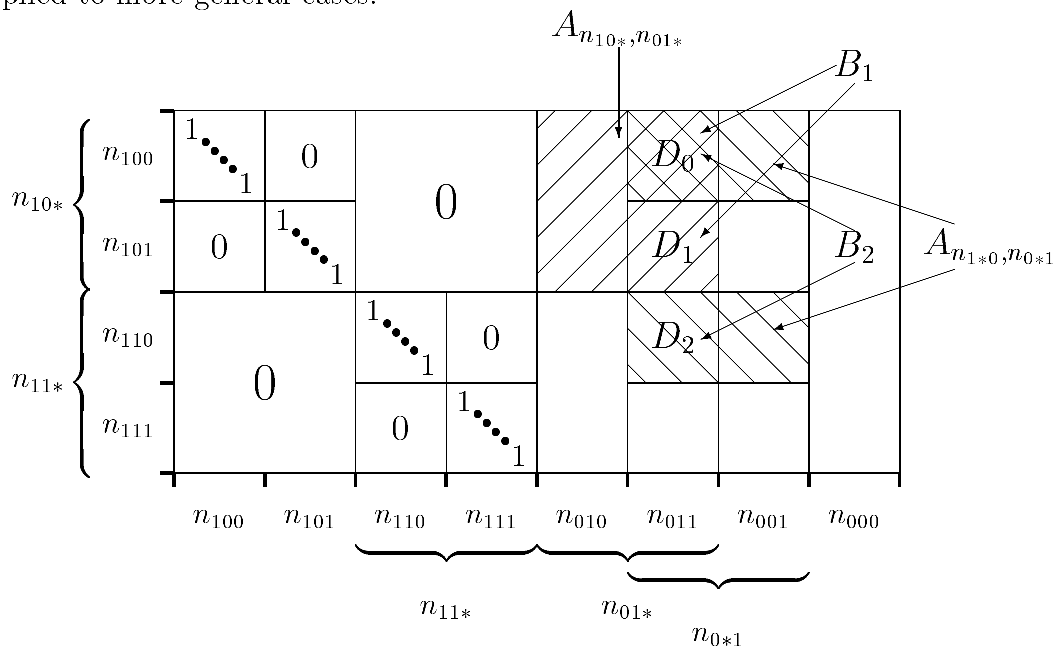

Figure 2.2. The matrix A by renumbering variables.

2.4 Subcodes with the Smallest Number of Minimum Weight Codewords

We consider how to delete ΔK monomials in order to obtain the subcode with the smallest number of minimum weight codewords. We first prove the following lemma.

Lemma 2 We assume that (a) there is an integer, say m, which is not contained in any $\alpha \in \Delta J$ and (b) there are two or more sets $\beta_1, \beta_2, \dots, \beta_p$ in ΔJ which have a common integer, say $m-1$, and no other sets in ΔJ contain $m-1$. Let $\beta'_p \in J_r$ denote the set obtained from β_p by replacing $m-1$ with m and $\Delta J' \stackrel{\triangle}{=} (\Delta J \setminus \{\beta_p\}) \cup \left\{\beta'_p\right\}$. Then, we have the following inequality:

$$N_{\min}(C(\Delta J)) > N_{\min}(C(\Delta J')). \tag{2.15}$$

(Proof) For a subset J' of J, define $\Lambda(J') \stackrel{\triangle}{=} \{A \in \Lambda_{r,m} : \det(A_\alpha) = 0, \alpha \in J'\}$. Then [2, Theorems 3 and 5, Appendix B],

$$\begin{aligned} N_{\min}(C(J')) &= |\{p(A) : A \in \Lambda(J')\}| \\ &= \frac{|\Lambda(J')|}{\prod_{i=0}^{r-1}(2^r - 2^i)}. \end{aligned}$$

Hence it is sufficient to show that

$$|\Lambda(\Delta J)| > |\Lambda(\Delta J')|. \tag{2.16}$$

For a binary $r \times (m+1)$ matrix A, let A_0 denote the $r \times (m-1)$ submatrix consisting of the first to the $(m-2)$-th columns and the $(m+1)$-th column of A. That is, $A_0 = A_{\{1,2,\dots,m+1\}\setminus\{m-1,m\}}$. For a binary $r \times (m-1)$ matrix D, define

$$\begin{aligned} \Lambda(\Delta J, D) &\stackrel{\triangle}{=} \{A \in \Lambda(\Delta J) : A_0 = D\}, \\ \Lambda(\Delta J', D) &\stackrel{\triangle}{=} \{A' \in \Lambda(\Delta J') : A'_0 = D\}. \end{aligned}$$

We will show that for any binary $r \times (m-1)$ matrix D,

$$|\Lambda(\Delta J, D)| \geq |\Lambda(\Delta J', D)|, \tag{2.17}$$

and that there are binary $r \times (m-1)$ matrices D's such that

$$|\Lambda(\Delta J, D)| > |\Lambda(\Delta J', D)|. \tag{2.18}$$

Since $|\Lambda(\Delta J)| = \sum_D |\Lambda(\Delta J, D)|$ and $|\Lambda(\Delta J')| = \sum_D |\Lambda(\Delta J', D)|$, where $\sum_D$ denotes the sum over all binary $r \times (m-1)$ matrices, (2.17) and (2.18) imply (2.16) and therefore (2.15).

There are the following three cases of D to be considered. Let D_o denote the submatrix of D consisting of the first $m-2$ columns of D.

(i) The rank of $D_{\beta_p \setminus \{m-1\}} \leq r-2$: For any binary $r \times (m+1)$ matrix A such that $A_0 = D$, $\det(A_{\beta_p}) = \det(A_{\beta'_p}) = 0$. Hence, if $A' \in \Lambda(\Delta J', D)$, then $A' \in \Lambda(\Delta J, D)$. That is, (2.17) holds.

(ii) The rank of $D_{\beta_p \setminus \{m-1\}} = r-1$:

(ii.1) First, we assume that the rank of D_o is r. If there is a set $\alpha \in \Delta J \setminus \{\beta_1, \beta_2, \dots, \beta_p\}$ such that $\det(D_\alpha) = 1$, then $\Lambda(\Delta J, D) = \Lambda(\Delta J', D) = \emptyset$. We consider the case where for any set α in $\Delta J \setminus \{\beta_1, \beta_2, \dots, \beta_p\}$, $\det(D_\alpha) = 0$. Then, for any binary $r \times (m+1)$ matrix A (or A') such that $A_0 = D$ and $\det(A_{\beta_i}) = 0$ with $1 \leq i \leq p$ (or $A'_0 = D$, $\det(A'_{\beta_i}) = 0$ with $1 \leq i < p$ and $\det(A'_{\beta'_p}) = 0$), $A \in \Lambda(\Delta J, D)$ (or $A' \in \Lambda(\Delta J', D)$).

Let $a_{i,j}$ denote the (i,j) element of A and $a'_{i,j}$ denote the (i,j) element of A'. We will compare $|\Lambda(\Delta J, D)|$ with $|\Lambda(\Delta J', D)|$. We can expand $\det(A_{\beta_i})$ on the $(m-1)$-th column of A_{β_i} for $1 \leq i \leq p$ as follows:

$$\det(A_{\beta_i}) = \sum_{j=1}^{r} B_{ij} a_{j,m-1} = 0, \text{ for } 1 \leq i \leq p, \tag{2.19}$$

where B_{ij} is a cofactor and is dependent on D only. There is no restriction on $a_{1,m}, a_{2,m}, \dots, a_{r,m}$. Let ρ denote the rank of the coefficient matrix of (2.19) whose (i,j) element is B_{ij} for $1 \leq i \leq p$ and $1 \leq j \leq r$. Since $|\Lambda(\Delta J, D)|$ is the number of $a_{1,m-1}, a_{2,m-1}, \dots, a_{r,m-1}, a_{1,m}, a_{2,m}, \dots, a_{r,m}$ satisfying (2.19),

$$|\Lambda(\Delta J, D)| = 2^{2r-\rho}. \tag{2.20}$$

We can expand $\det(A'_{\beta_i})$ on the $(m-1)$-th column of A'_{β_i} for $1 \leq i < p$ and $\det(A'_{\beta'_p})$ on the m-th column of $A'_{\beta'_p}$ as follows:

$$\det(A'_{\beta_i}) = \sum_{j=1}^{r} B_{ij} a'_{j,m-1} = 0, \text{ for } 1 \leq i < p, \tag{2.21}$$

$$\det(A'_{\beta'_p}) = \sum_{j=1}^{r} B_{pj} a'_{j,m} = 0. \tag{2.22}$$

Let ρ' denote the rank of the coefficient matrix of (2.21) whose (i,j) elements is B_{ij} for $1 \leq i < p$ and $1 \leq j \leq r$. Then,

$$\rho' = \rho \text{ or } \rho - 1. \tag{2.23}$$

The number of $a'_{1,m-1}, a'_{2,m-1}, \ldots, a'_{r,m-1}$ satisfying (2.21) is $2^{r-\rho'}$. Since the rank of $A'_{\beta'_p \setminus \{m\}} (= D_{\beta_p \setminus \{m-1\}})$ is $r-1$, one of B_{p_j} with $1 \leq j \leq r$ is not zero. Hence, the number of $a'_{1,m}, a'_{2,m}, \ldots, a'_{r,m}$ satisfying (2.22) is 2^{r-1}. Consequently, we have that

$$|\Lambda(\Delta J', D)| = 2^{2r-\rho'-1}. \tag{2.24}$$

From (2.20), (2.23) and (2.24), we see that

$$|\Lambda(\Delta J, D)| \geq |\Lambda(\Delta J', D)|. \tag{2.25}$$

(ii.2) Next, we assume that the rank of D_o is $r-1$. It is sufficient to consider the case that the first $r-1$ rows of D_o are linearly independent and the last row of D_o is a zero row. Since the rank of $D_{\beta_p \setminus \{m-1\}}$ is $r-1$, if $A \in \Lambda(\Delta J, D)$, then $a_{r,m-1} = 0$ from $\det(A_{\beta_p}) = 0$ and $a_{r,m} = 1$ from $A \in \Lambda_{r,m}$. Conversely, if $a_{r,m-1} = 0$ and $a_{r,m} = 1$, then $\det(A_\alpha) = 0$ for $\alpha \in \Delta J$ and $\det(A_{\beta'_p}) = 1$, that is, $A \in \Lambda(\Delta J, D)$. Hence, $|\Lambda(\Delta J, D)| = 2^{2r-2}$. Similarly, if $A' \in \Lambda(\Delta J', D)$, then $a'_{r,m} = 0$ from $\det(A'_{\beta'_p}) = 0$ and $a'_{r,m-1} = 1$ from $A' \in \Lambda_{r,m}$. Hence, $|\Lambda(\Delta J', D)| \leq 2^{2r-2}$. Consequently

$$|\Lambda(\Delta J, D)| \geq |\Lambda(\Delta J', D)|. \tag{2.26}$$

Note that if the rank of $D_{\beta_{p-1} \setminus \{m-1\}}$ is $r-1$, then $a'_{r,m-1} = 1$ implies $\det(A'_{\beta_{p-1}}) = 1$. That is, $\Lambda(\Delta J', D)$ is empty. Let D be a binary $r \times (m-1)$ matrix as follows:

(a) the last row of D is a zero row, (b) the rank of $D_{\beta_p \backslash \{m-1\}}$ is $r-1$ and (c) there is a one-to-one correspondence between the set of columns of $D_{\beta_p \backslash \{m-1\}}$ and that of $D_{\beta_{p-1} \backslash \{m-1\}}$ such that the corresponding columns are the same column vector. Then, the rank of $D_{\beta_{p-1} \backslash \{m-1\}}$ is $r-1$, and therefore,

$$|\Lambda(\Delta J, D)| > |\Lambda(\Delta J', D)|. \tag{2.27}$$

$\triangle\triangle$

Suppose that $r\Delta K \leq m$. From Lemma 2, we can remove the overlap among ΔJ step by step to decrease the number of minimum weight codewords. Thus we have the following theorem.

Theorem 1 For $r\Delta K \leq m$, the value of $N_{\min}(C(\Delta J))$ takes the minimum, if and only if the sets in ΔJ are mutually disjoint. $\triangle\triangle$

In case that ΔJ consists of mutually disjoint sets, a formula for $N_{\min}(C(\Delta J))$ is readily derived from (2.5) and Lemma 1.

$\mathrm{RM}_{r,m}$ code is spanned by the set of codewords with the minimum weight [2, Ch.13, §5]. We can consider subcodes of RM codes which are spanned by a set of codewords with the minimum weight. The basis of this subcode is formed from that of a RM code by deleting ΔK codewords of the minimum weight which are linearly independent each other. Then, for $r\Delta K \leq m$, we can replace the ΔK codewords to ΔK monomials of degree r by an affine transformation. Hence, this case is reduced to the case that we have considered in Sections 2.3 and 2.4.

Example 1 Consider the number of minimum weight codewords of $(64, 40)$ subcodes of the $(64, 42)$ RM code, that is, $\mathrm{RM}_{3,6}$. Table 2.1 shows the number of minimum weight codewords of a subcode whose basis is formed from that of the $(64, 42)$ RM code by deleting two monomials. By renumbering the suffices of variables, there are exactly three cases, that is, $\Delta M_0 = \{x_1x_2x_3, x_4x_5x_6\}$, $\Delta M_1 = \{x_1x_2x_3, x_3x_4x_5\}$ and $\Delta M_2 = \{x_1x_2x_3, x_2x_3x_4\}$. The code $C(\Delta M_0)$ with the smallest number of minimum weight codewords has the smallest block error probabilities at $E_b/N_0 = 2.0, 3.0, 4.0$ and 5.0dB as is shown in Table 2.1.

The result in Sections 2.3 and 2.4 can be generalized to the case where monomials of degree $r-1$ may be deleted, that is, $\Delta M \subseteq M_{r,m} \backslash M_{r-2,m}$, by using the following

Table 2.1. Minimum Weight Codewords of $(64, 40)$ subcode of $\mathrm{RM}_{3,6}$.

ΔM	The number of minimum weight codewords	The block error probabilities at E_b/N_0 for soft-decision maximum likelihood decoding			
		2.0dB	3.0dB	4.0dB	5.0dB
ΔM_0	4312	8.54×10^{-2}	1.33×10^{-2}	9.68×10^{-4}	2.93×10^{-5}
ΔM_1	4504	8.67×10^{-2}	1.36×10^{-2}	9.97×10^{-4}	3.38×10^{-5}
ΔM_2	5016	8.99×10^{-2}	1.46×10^{-2}	1.09×10^{-3}	3.66×10^{-5}

fact: For $\alpha = (j_1, j_2, \ldots, j_{r-1}) \in J_{r-1}$, let A_α denote the $r \times r$ submatrix of A consisting of the j_1-th, the j_2-th, $\ldots$, the j_{r-1}-th and the $(m+1)$-th columns of A. Then, we have that

$$p(A) \equiv \sum_{\alpha \in J_{r-1} \cup J_r} \det(A_\alpha) m_\alpha, \ \bmod\ P_{r-2,m}. \tag{2.28}$$

We consider the case where the subcode C of a RM code is spanned by monomials in $M_{r,m} \setminus \Delta M$ where $\Delta M \subseteq M_{r-2,m}$, $|\Delta M| = 2$ and $\Delta M = \{\alpha_1, \alpha_2\}$ for $\alpha_1 \in J_{r-1}$ and $\alpha_2 \in J_r$.

Consider the case of $|\Delta M| = 1$ and $\alpha_1 \in J_{r-1}$. There is no loss of generality in assuming that $\alpha_1 = (1, 2, \ldots, r-1)$. The first to the $r-1$-th columns of A are linearly independent and there exists unique j satisfying the following conditions:

(1) $r \leq j \leq m$,

(2) the first to the $r-1$-th and the i-th columns of A for $r \leq i < j$ are linearly depend,
and

(3) the first to the $r-1$-th and the j-th columns of A are linearly independent.

For any j, we evaluate the number N_j of matrices such that

(1′) the submatrix consisting of the first to the $r-1$-th and the j-th columns of A is the identity matrix,
and

(2′) the first to the $r-1$-th and $m+1$-th columns of A are linearly independent.

Then, the total sum of N_j for $r \leq j \leq r$ is $\nu(\alpha_1)$, that is

$$\nu(\alpha_1) = \sum_{j=r}^{m} N_j. \tag{2.29}$$

Consider the case of $|\Delta M| = 2$. Let l denotes the number of integer appearing in both of α_1 and α_2. By renumbering variables, we can assume that $\alpha_1 = (r+1-l, r+2-l, \ldots, 2r-1-l)$ and $\alpha_2 = (1, 2, \ldots, r)$ where $0 \leq l < r$. Then, $\det(A_{\alpha_1}) = 1$, if and only if the $(r-l) \times (r-l)$ submatrix consisting of the first $r-l$ rows and the j_1-th, the j_2-th, ..., the j_{r-1}-th and the $(m+1)$-th columns of A is regular. There is no restriction of the submatrix consisting of columns of A, excluding the first $(2r-1-l)$ columns an the $(m+1)$-th column. Then, the number of matrices A such that A_{α_2} is the identity matrix and $\det(A_{\alpha_1}) = 1$ is

$$2^{r\{m+1-(2r-l)\}} \cdot 2^{(r-l)l} \cdot \prod_{j=0}^{r-l-1} (2^{r-l} - 2^j). \tag{2.30}$$

From Lemma 1,

$$\nu(\alpha_1, \alpha_2) = 2^{r\{m+1-(2r-l)\}} \cdot 2^{(r-l)l} \cdot \prod_{j=0}^{r-l-1} (2^{r-l} - 2^j). \tag{2.31}$$

It follows from (2.5), (2.7), (2.29) and (2.31) that

$$\begin{aligned} N_{\min}(C) &= 2^r \prod_{i=0}^{m-r-1} \frac{2^{m-i}-1}{2^{m-r-i}-1} - \sum_{j=r}^{m} N_j - 2^{(m-r+1)r} \\ &\quad + 2^{r\{m+1-(2r-l)\}} \cdot 2^{(r-l)l} \cdot \prod_{j=0}^{r-l-1} (2^{r-l} - 2^j). \end{aligned} \tag{2.32}$$

Example 2 Consider the number of minimum weight codewords of $(64, 40)$ subcodes of the $(64, 42)$ RM code. Table 2.2 shows the number of minimum weight codewords of a subcode of a RM code is spanned by monomials in $M_{r,m} \setminus \Delta M$ where $\Delta M \subseteq M_{r-2,m}$, $|\Delta M| = 2$ and $\Delta M = \{\alpha_1, \alpha_2\}$ for $\alpha_1 \in J_{r-1}$ and $\alpha_2 \in J_r$. By renumbering the suffices of variables, there are exactly three cases, that is, $\Delta M = \{x_1x_2x_3, x_4x_5\}$, $\{x_1x_2x_3, x_3x_4\}$ and $\{x_1x_2x_3, x_2x_3\}$.

Table 2.2. $N_{\min}(C(\Delta M))$ where $\Delta M \subseteq M_{r,m} \setminus M_{r-2,m}$, $|\Delta M| = 2$ and $\Delta M = \{\alpha_1, \alpha_2\}$ for $\alpha_1 \in J_{r-1}$ and $\alpha_2 \in J_r$.

ΔM	The number of minimum weight codewords
$x_1x_2x_3, x_4x_5$	4568
$x_1x_2x_3, x_3x_4$	4760
$x_1x_2x_3, x_2x_3$	5272

2.5 Conclusion

We have presented a formula or an effective method which gives the number of minimum weight codewords in a $(2^m, \sum_{i=0}^{r} \binom{m}{i} - \Delta K)$ linear subcode C of $\mathrm{RM}_{r,m}$ code which is spanned by monomials with m variables of degree r or less over GF(2) for $\Delta K \leq 3$. Next, it has been shown in Theorem 1 how to delete ΔK monomials in order to obtain the subcode with the smallest number of codewords of the minimum weight for $r\Delta K \leq m$. In Example 1, we have shown the numbers of minimum weight codewords of all such $(64, 40)$ subcodes of $\mathrm{RM}_{3,6}$ and those error probabilities of soft-decision maximum likelihood decoding by simulation.

Chapter 3

A Recursive MLD Algorithm for Some Transitive Invariant Binary Block Codes

3.1 A Review of RMLD Algorithm

Consider a binary (n, k) linear block code C for error control over an AWGN channel using BPSK signaling.

For a positive integer n, let V^n denote the vector space of all the n-tuples over GF(2). For an n-tuple $\boldsymbol{u} = (u_1, u_2, \ldots, u_n) \in V^n$, let $\mathrm{wt}(\boldsymbol{u})$ be the Hamming weight (or simply weight) of $\boldsymbol{u}$. Let x and y be two nonnegative integers such that $0 \leq x < y \leq n$. For an n-tuple $\boldsymbol{u} = (u_1, u_2, \ldots, u_n) \in V^n$, define $p_{x,y}\boldsymbol{u} \triangleq (u_{x+1}, u_{x+2}, \ldots, u_y)$, and $p_{x,y}(C) \triangleq \{p_{x,y}\boldsymbol{u} : \boldsymbol{u} \in C\}$. Then $p_{x,y}(C)$ is a linear code of length $y - x$. Let $C_{x,y}$ denote the subcode of C which consists of those codewords whose components are all zero except for the $y - x$ components from the $(x+1)$-th bit position to the y-th bit position. For simplicity, we use the notation $C^{\mathrm{tr}}_{x,y}$, for the truncated code $p_{x,y}(C_{x,y})$.

For a linear block code A and its linear subcode $B, A/B$ denotes the set of cosets of B in A, called partition of A with respect to B. For two binary tuples $\boldsymbol{u} = (u_1, u_2, \ldots, u_i)$ and $\boldsymbol{v} = (v_1, v_2, \ldots, v_j)$, let $\boldsymbol{u} \circ \boldsymbol{v}$ denote the concatenation of $\boldsymbol{u}$ and $\boldsymbol{v}$, $(u_1, u_2, \ldots, u_i, v_1, v_2, \ldots, v_j)$, and for block codes A and B, $A \circ B$ denotes $\{\boldsymbol{u} \circ \boldsymbol{v} : \boldsymbol{u} \in A, \boldsymbol{v} \in B\}$. For a set X, $|X|$ denotes the cardinality of X.

For integers x and y such that $0 \leq x < y \leq n$, let $L_{x,y}$ denote the cosets of $C^{\text{tr}}_{x,y}$ in the partition $p_{x,y}(C)/C^{\text{tr}}_{x,y}$, i.e.,

$$L_{x,y} \overset{\triangle}{=} p_{x,y}(C)/C^{\text{tr}}_{x,y}. \tag{3.1}$$

$L_{x,y}$ plays a key role in RMLD algorithm.

Let D be a coset in $L_{x,y}$. For a vector $\boldsymbol{u} \in D$, let $m(\boldsymbol{u})$ denote the correlation metric of $\boldsymbol{u}$ with the received sequence (from the $(x+1)$-th bit position to the y-th bit position). Define

$$m(D) \overset{\triangle}{=} \max_{\boldsymbol{u} \in D} m(\boldsymbol{u}) \tag{3.2}$$

which is called the metric of coset D. Let $l(D)$ denote the vector in D that has the largest (correlation) metric in D. This $(y-x)$-tuple $l(D)$ is called the label of coset D.

Form a metric table for $L_{x,y}$ which consists of the pairs,

$$(m(D), l(D)),$$

for all the cosets in $L_{x,y}$. This metric table is called $L_{x,y}$-table. Note that for $x = 0$ and $y = n$, $L_{0,n} = \{C\}$. When $L_{0,n}$-table is constructed, it stores the most likely codeword and its correlation metric. The RMLD algorithm for decoding C is simply a recursive procedure for constructing $L_{x,y}$-tables for long trellis sections from short trellis sections.

Assume that $y - x \geq 2$. Let z be an integer such that $x < z < y$. For each $D \in L_{x,y}$, define $A_z(D) \overset{\triangle}{=} \{D_z \in L_{x,z} : D_z \subseteq p_{x,z}(D)\}$. Since D is in $p_{x,y}(C)/C^{\text{tr}}_{x,y}$ and D_z is in $p_{x,z}(C)/C^{\text{tr}}_{x,z}$, D_z is in $p_{x,z}(p_{x,y}(C)/(C^{\text{tr}}_{x,z} \circ C^{\text{tr}}_{z,y}))$, that is, for each D_z, there is a binary $(y-x)$-tuple $\boldsymbol{u}$ such that $D_z = \{p_{x,z}\boldsymbol{u} + \boldsymbol{v} : \boldsymbol{v} \in C^{\text{tr}}_{x,z}\}$ and there is exactly one coset in $L_{z,y}$, $\{p_{z,y}\boldsymbol{u} + \boldsymbol{v} : \boldsymbol{v} \in C^{\text{tr}}_{z,y}\}$, denoted $\text{adj}(D_z, D)$, such that

$$D_z \circ \text{adj}(D_z, D) \subseteq D. \tag{3.3}$$

Hence $|A_z(D)|$, denoted $A_{x,y;z}$, is given by

$$A_{x,y;z} = |C^{\text{tr}}_{x,y}|/(|C^{\text{tr}}_{x,z}| \cdot |C^{\text{tr}}_{z,y}|). \tag{3.4}$$

Then, it follows from (3.3) that

$$D = \bigcup_{D_z \in A_z(D)} D_z \circ \mathrm{adj}(D_z, D), \tag{3.5}$$

and

$$\begin{aligned} m(D) &= \max_{D_z \in A_z(D)} \{m(D_z) + m(\mathrm{adj}(D_z, D))\}, & (3.6) \\ l(D) &= l(D'_z) \circ l(\mathrm{adj}(D'_z, D)), & (3.7) \end{aligned}$$

where the maximum of the right-hand side of (3.6) is taken by $D'_z \in A_z(D)$. Note that $m(D'_z)$ and $l(D'_z)$ are stored in the $L_{x,z}$-table and $m(\mathrm{adj}(D'_z, D))$ and $l(\mathrm{adj}(D'_z, D))$ are stored in the $L_{z,y}$-table. Therefore, from (3.6) and (3.7), $L_{x,y}$-table can be constructed from $L_{x,z}$- and $L_{z,y}$-tables.

The $L_{0,n}$-table for C (or decoding C) can be obtained by executing the following recursive procedure RMLD-$C(x, y)$:

(1) Construct $L_{x,x+1}$-table directly, or

(2) if $y - x \geq 2$, then choose an integer z such that $x < z < y$. Execute RMLD-$C(x, z)$ and RMLD-$C(z, y)$ to form $L_{x,z}$- and $L_{z,y}$-tables. Construct the $L_{x,y}$-table from $L_{x,z}$-table, $L_{z,y}$-table and $A_z(D)$ by using (3.6) and (3.7).

△△

A straightforward method for solving (3.6) is **add-compare-select(ACS)** which is used in conventional Viterbi algorithm. Let $\gamma(x, y; z)$ denote the number of additions and comparisons in **ACS** for solving (3.6). Then

$$\begin{aligned} \gamma(x, y; z) &= \sum_{D \in L_{x,y}} (2|A_z(D)| - 1) \\ &= (2A_{x,y;z} - 1)\, |L_{x,y}|\, . & (3.8) \end{aligned}$$

3.2 Transitive Invariant Block Codes

For a positive integer m, let P_m denote the set of Boolean polynomials of m variables, $x_1, x_2, \ldots, x_m$, which take values 0 or 1. Consider a Boolean polynomial $f \overset{\triangle}{=} f(x_1, x_2, \ldots, x_m)$ in P_m. For each combination of values of $x_1, x_2, \ldots, x_m$, the polynomial f takes a truth value either 0 or 1. For the 2^m combinations of values of $x_1, x_2, \ldots, x_m$, the truth values of f forms a 2^m-tuple over GF(2).

For a nonnegative integer l less than 2^m, let $(b_{l_1}, b_{l_2}, \ldots, b_{l_m})$ be the standard binary representation of l such that

$$l = b_{l_1} + b_{l_2}2 + \cdots + b_{l_m}2^{m-1}. \tag{3.9}$$

For a given Boolean polynomial $f \in P_m$, we form the following 2^m-tuple:

$$\boldsymbol{v} = (v_1, v_2, \ldots, v_{2^m}) \tag{3.10}$$

where

$$v_{l+1} = f(b_{l_1}, b_{l_2}, \ldots, b_{l_m}) \tag{3.11}$$

with $0 \leq l < 2^m$. We say that the Boolean polynomial f represents the vector $\boldsymbol{v}$. We use the notation $\boldsymbol{v}_m(f)$ (or simply $\boldsymbol{v}(f)$ when there is no confusion) for the vector represented by f.

There are 2^{2^m} Boolean polynomials in P_m. It follows from (3.10) and (3.11) that these polynomials uniquely define all the 2^m-tuples over GF(2). For a subset $X \subseteq P_m$, define

$$\boldsymbol{v}_m(X) \triangleq \{\boldsymbol{v}_m(f) : f \in X\}$$

which is a subset of the vector space of all the 2^m-tuples over GF(2). Therefore, for a binary block code C of length 2^m, there exists a subset P_C of P_m such that $C = \boldsymbol{v}_m(P_C)$. This Boolean polynomial representation of block codes of length 2^m is quite useful in analyzing their structural properties [2]. The most well known example is the Boolean polynomial representation of RM codes. For $0 \leq r \leq m$, let $P_{r,m}$ denote the set of polynomials of degree r or less in P_m. Let $\mathrm{RM}_{r,m}$ denote the r-th order RM code of length $n \triangleq 2^m$. Then, the $\mathrm{RM}_{r,m}$ is given by $\boldsymbol{v}_m(P_{r,m})$, i.e.,

$$\mathrm{RM}_{r,m} = \boldsymbol{v}_m(P_{r,m}) = \{\boldsymbol{v}_m(f) : f \in P_{r,m}\}. \tag{3.12}$$

Let C be a binary block code of length 2^m which is specified by a subset P_C of polynomials in P_m. We introduce the following definition:

Definition 1 C is said to be binary transitive invariant (or b-transitive invariant), if and only if for any $f(x_1, x_2, \ldots, x_m) \in P_C$ and $(a_1, a_2, \ldots, a_m) \in V_m$,

$$f(x_1 + a_1, x_2 + a_2, \ldots, x_m + a_m) \in P_C. \tag{3.13}$$

That is, if $f \in P_C$ represents a vector $\boldsymbol{v} = (v_1, v_2, \ldots, v_{2^m}) \in \boldsymbol{v}_m(P_C)$, then for $(a_1, a_2, \ldots, a_m) \in V_m$, $f(x_1 + a_1, x_2 + a_2, \ldots, x_m + a_m)$ represents another vector $\boldsymbol{u} = (u_1, u_2, \ldots, u_{2^m}) \in \boldsymbol{v}_m(P_C)$.

△△

This transitive operation simply permutes the components of $\boldsymbol{v}_m(f)$. For $0 \leq l < 2^m$, the component v_{l+1} of $\boldsymbol{v}_m(f)$ at the position $l + 1 = \sum_{i=1}^{m} b_{l_i} 2^{i-1} + 1$ is permuted to the position

$$l' + 1 = \sum_{i=1}^{m} (b_{l_i} + a_i) 2^{i-1} + 1.$$

RM codes [4] and extended and permuted binary primitive BCH codes [9] are b-transitive invariant. Let C_w denote the set of codewords in C with weight w, i.e.,

$$C_w \triangleq \{\boldsymbol{u} \in C : \mathrm{wt}(\boldsymbol{u}) = w\}.$$

It is clear that, for a b-transitive invariant code C, the subcode C_w is also b-transitive invariant. For a polynomial $f(x_1, x_2, \ldots, x_m) \in P_m$, define ∂f as follows:

$$\partial f \triangleq \{f(x_1 + a_1, x_2 + a_2, \ldots, x_m + a_m) - f(x_1, x_2, \ldots, x_m) : (a_1, a_2, \ldots, a_m) \in V_m\}. \tag{3.14}$$

Then we have the following lemma:

Lemma 3 If C is a binary linear code of length 2^m, then there is a subset Q_C of P_m such that C is spanned by $\boldsymbol{v}_m(Q_C)$. C is b-transitive invariant if and only if for $f \in Q_C$, each polynomial in ∂f can be expressed as a linear sum of polynomials in Q_C.

△△

For two monomials $t_1 = x_{i_1}x_{i_2}\cdots x_{i_p}$ and $t_2 = x_{j_1}x_{j_2}\cdots x_{j_q}$, t_1 is said to be a subterm of t_2 or t_2 is said to be a superterm of t_1, if and only if $\{i_1, i_2, \ldots, i_p\} \subseteq \{j_1, j_2, \ldots, j_q\}$. Then, for a monomial t in P_m, each polynomial in ∂t (defined by (3.14)) can be expressed as a linear sum of subterms of t other than t itself. Suppose that a linear code C is spanned by $\boldsymbol{v}_m(M_B)$(or simply M_B), where M_B is a set of monomials in P_m. Then, it follows from Lemma 3 that C is b-transitive invariant, if and only if the following closure condition (CS) on subterms holds:
(CS) For $t \in M_B$, all subterms of t are in M_B.

Let $f(x_1, x_2, \ldots, x_m)$ be a polynomial in P_m and $\{b_1, b_2, \ldots, b_h\}$ be a set of binary constants (i.e. $b_i = 0$ or 1 for $1 \le i \le h$). Let $f_{b_1 b_2 \cdots b_h}$ be the polynomial obtained from $f(x_1, x_2, \ldots, x_m)$ by setting $x_{m-h+1} = b_1, x_{m-h+2} = b_2, \ldots, x_m = b_h$. Then $f_{b_1 b_2 \cdots b_h}$ is a Boolean polynomial of $m-h$ variables, $x_1, x_2, \ldots, x_{m-h}$ in P_{m-h}. Let β represent the binary sequence $b_1 b_2 \cdots b_h$, i.e., $\beta \triangleq b_1 b_2 \cdots b_h$. For simplicity, we use the notation f_β for $f_{b_1 b_2 \cdots b_h}$. Let j be the integer defined by $\beta = b_1 b_2 \cdots b_h$ as follows:

$$j \triangleq \sum_{i=1}^{h} b_i 2^{i-1}. \tag{3.15}$$

Then we can readily see that

$$p_{j2^{m-h},(j+1)2^{m-h}}(\boldsymbol{v}_m(f)) = \boldsymbol{v}_{m-h}(f_\beta). \tag{3.16}$$

For $f(x_1, x_2, \ldots, x_m) \in P_m$, binary sequences $\beta = b_1 b_2 \cdots b_h$ and $\beta' = b'_1 b'_2 \cdots b'_h$ with $1 \le h \le m$, define

$$\begin{aligned} & f'(x_1, x_2, \ldots, x_m) \\ \triangleq\ & f(x_1, x_2, \ldots, x_{m-h}, x_{m-h+1} + b_1 + b'_1, x_{m-h+2} + b_2 + b'_2, \ldots, x_m + b_h + b'_h), \end{aligned} \tag{3.17}$$

$j \triangleq \sum_{i=1}^{h} b_i 2^{i-1}$ and $j' \triangleq \sum_{i=1}^{h} b'_i 2^{i-1}$. From (3.16), we have

$$\begin{aligned} p_{j2^{m-h},(j+1)2^{m-h}}(\boldsymbol{v}_m(f)) &= \boldsymbol{v}_{m-h}(f_\beta) = \boldsymbol{v}_{m-h}(f'_{\beta'}) \\ &= p_{j'2^{m-h},(j'+1)2^{m-h}}(\boldsymbol{v}_m(f')). \end{aligned} \tag{3.18}$$

If C is b-transitive invariant, then $f' \in P_C$ if and only if $f \in P_C$. Hence, from (3.18),

$$p_{j2^{m-h},(j+1)2^{m-h}}(C) = p_{j'2^{m-h},(j'+1)2^{m-h}}(C). \tag{3.19}$$

Similarly, we have that

$$C^{\mathrm{tr}}_{j2^{m-h},(j+1)2^{m-h}} = C^{\mathrm{tr}}_{j'2^{m-h},(j'+1)2^{m-h}}. \tag{3.20}$$

From (3.1), (3.19) and (3.20), we have the following theorem.

Theorem 2 Suppose a linear block code C of length 2^m is b-transitive invariant. Then for $1 \leq h \leq m$ and $0 \leq j < 2^h$, $L_{j2^{m-h},(j+1)2^{m-h}}$ for C is the same for all j. △△

Theorem 2 says that for a b-transitive invariant code C of length 2^m, if we divide the code (or code trellis) into sections of length 2^{m-h}, all the sections have the same structural properties with respect to the operations of the RMLD algorithm proposed in [5].

3.3 Decoding of Binary Transitive Invariant Codes with the RMLD Algorithm

Let C be a binary linear block code of length 2^m which is b-transitive invariant. Suppose C is decoded with the RMLD algorithm. From Theorem 2, we see that if we sectionalize the code uniformly at each recursion level, all the sections will have the same $L_{x,y}$. Consequently, the RMLD-$C(x,y)$ procedure is applied uniformly among all the sections to construct $L_{x,y}$-tables from tables at a lower level based on (3.6) and (3.7). This uniformity (or regularity) is very advantageous in implementation. In hardware implementation, this results in identical circuits for all the sections [1].

Based on Theorem 2, we propose the following binary (uniform) sectionalization. At each level, we always choose x and y such that $0 \leq x < y < 2^m$, $y - x \geq 2$, $x + y$ even and $z = (x+y)/2$. At the 0-th level (or top level), we set

$x = 0$, $y = 2^m$ and $z = 2^{m-1}$. At the first level, the code is partitioned into two sections with

$$(x, y) \in \{(0, 2^{m-1}), (2^{m-1}, 2^m)\}.$$

Each section is of length 2^{m-1}. At the second level, each section at the first level is partitioned into two equal sections, each section is of length 2^{m-2}. This results in four sections with

$$\begin{aligned}(x, y) \in \{&(0, 2^{m-2}), (2^{m-2}, 2^{m-1}),\\ &(2^{m-1}, 2^{m-2} + 2^{m-1}), (2^{m-2} + 2^{m-1}, 2^m)\}.\end{aligned}$$

For $0 \leq h < m$, at the h-th level, the code is partitioned into 2^h sections with

$$(x, y) \in \{(j2^{m-h}, (j + 1)2^{m-h}) : 0 \leq j < 2^h\}. \tag{3.21}$$

Each section is of length 2^{m-h}. The above sectionalization process continues until $y - x = 2$. At the bottom of the sectionalization, each section consists of two code bits. This results in a sectionalization tree shown in Figure 3.1.

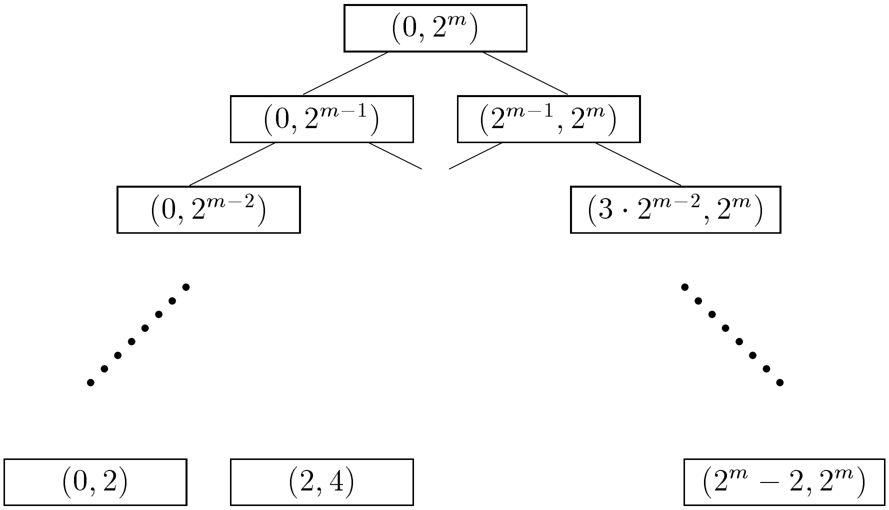

Figure 3.1. Binary Sectionalization.

In decoding C using the RMLD algorithm, all the $L_{x,y}$-tables at the bottom of the sectionalization tree are constructed directly using the methods proposed in [5]. As the recursion moves up to higher levels of the tree, each $L_{x,y}$-table is constructed from two tables, $L_{x,z}$-table and $L_{z,y}$-table, with $z = (x+y)/2$ at a lower level. Suppose the h-th level of recursion has been completed. We have all the $L_{x,y}$-tables for each (x, y) in the boundary of (3.21). The decoder moves up to the $(h-1)$-th level. At this level, the $L_{x,y}$-table with $x = j2^{m-h+1}$ and $y = (j+1)2^{m-h+1}$ is constructed from $L_{x,z}$-table and $L_{z,y}$-table with $z = (2j+1)2^{m-h}$. The recursion process continues until it reaches to the top of the tree. At this point, $L_{0,n}$-table is constructed, which contains only the most likely codeword and its metric. Decoding is then completed.

The above binary sectionalization is most suitable for parallel/pipeline processing, while the decoder is processing all the sections of a received word in parallel at one recursion level, it is also processing other received words at other recursion levels. This parallel/pipeline architecture is desired in high-speed data communication where a decoder must operate at a high speed.

For a transitive invariant linear block code of length 2^m using RMLD algorithm based on binary sectionalization, it follows from (3.8) and Theorem 2 that the total number of additions and comparison required to construct all the $L_{x,y}$-tables based on (3.6) using ACS procedure is given by

$$\gamma = \sum_{h=0}^{m-1} (2A_{0,2^{m-h}:2^{m-h-1}} - 1) \left| L_{0,2^{m-h}} \right| \cdot 2^h. \tag{3.22}$$

For a given code C, the computational complexity can be evaluated once $A_{0,2^{m-h};2^{m-h-1}}$ and $L_{0,2^{m-h}}$ for $0 \le h < m-1$ are known. For a nonlinear subcode C' of a binary linear block code C, $L_{x,y}$ is defined as $\{D \cap p_{x,y}(C') \neq \emptyset : D \in p_{x,y}(C)/C_{x,y}^{\rm tr}\}$. Then RMLD can be generalized to decoding of C' [9]. If C and C' are transitive invariant, then Theorem 2 can be extended to C'. For example, Theorem 2 holds for C_w with $0 < w \le 2^m$.

3.4 Structural and Computational Complexity Analysis

In the following, we first analyze the structure of

$$L_{x,y} = p_{x,y}(C)/C_{x,y}^{\rm tr}$$

and $A_{x,y;z}$. Once their structures are known, the computational complexity of the RMLD algorithm can be evaluated. The analysis is based on Boolean polynomial representation of C.

Suppose that a binary transitive invariant linear code C is spanned by $\boldsymbol{v}_m(M_B)$ where M_B is a set of monomials in P_m. Using binary sectionalization, we set

$$\begin{cases} x \triangleq j2^{m-h}, \\ y \triangleq (j+1)2^{m-h}, \end{cases} \tag{3.23}$$

for $0 \leq h < m$ and $0 \leq j < 2^h$. Let $\mathbf{0}_l$ denote the all-zero l-tuple over GF(2). It is easy to see that

(1) $p_{x,y}(C)$ is spanned by

$$M_{B,\mathbf{0}_h} \triangleq \{x_{i_1}x_{i_2}\cdots x_{i_p} \in M_B : 1 \leq i_1 < i_2 < \cdots < i_p \leq m-h\} \tag{3.24}$$

and

(2) $C^{\mathrm{tr}}_{x,y}$ is spanned by

$$\begin{aligned} M_B/x_{m-h+1}x_{m-h+2}\cdots x_m \quad \triangleq \quad & \{x_{i_1}x_{i_2}\cdots x_{i_p} : 1 \leq i_1 < i_2 < \cdots < i_p \leq m-h \\ & \text{and } x_{i_1}x_{i_2}\cdots x_{i_p}x_{m-h+1}x_{m-h+2}\cdots x_m \in M_B\}. \end{aligned} \tag{3.25}$$

Then, it follows from (3.1) and (3.4) that for code C,

$$\log_2 |L_{x,y}| = |M_{B,\mathbf{0}_h}| - |M_B/x_{m-h+1}x_{m-h+2}\cdots x_m|, \tag{3.26}$$

$$\begin{aligned} \log_2 A_{x,y;z} = \; & |M_B/x_{m-h+1}x_{m-h+2}\cdots x_m| \\ & -2|M_B/x_{m-h}x_{m-h+1}\cdots x_m|. \end{aligned} \tag{3.27}$$

Now we consider RM codes. Recall that the $\mathrm{RM}_{r,m}$ is specified by the Boolean polynomials in $P_{r,m}$. Let $M_{r,m}$ be the set of all monomials in $P_{r,m}$. Since $\mathrm{RM}_{r,m}$ is spanned by the vectors in $\boldsymbol{v}_m(M_{r,m})$, we obtain

$$M_{r,m,\mathbf{0}_h} = M_{\min\{r,m-h\},m-h} \tag{3.28}$$

$$\begin{aligned} M_{r,m}/x_{m-h+1}x_{m-h+2}\cdots x_m = \; & M_{r-h,m-h} \\ & \text{(empty for } r < h\text{).} \end{aligned} \tag{3.29}$$

It follows from (3.24) to (3.29) that we have Theorem 3 which gives the structure of $L_{x,y}$.

Theorem 3 For integers x and y defined by (3.23),

(1)

$$p_{x,y}(\mathrm{RM}_{r,m}) = \mathrm{RM}_{\min\{r,m-h\},m-h}, \tag{3.30}$$

(2)

$$[\mathrm{RM}_{r,m}]^{\mathrm{tr}}_{x,y} = \begin{cases} \mathrm{RM}_{r-h,m-h}, & \text{if } r \geq h, \\ \{\mathbf{0}_{2^{m-h}}\}, & \text{otherwise.} \end{cases} \tag{3.31}$$

(3) $L_{x,y}$ for $\mathrm{RM}_{m,r}$ is the same for all j such that $0 \leq j < 2^h$, and

$$L_{x,y} = \begin{cases} \mathrm{RM}_{\min\{r,m-h\},m-h}/\mathrm{RM}_{r-h,m-h}, & \text{for } r \geq h, \\ \mathrm{RM}_{\min\{r,m-h\},m-h}/\{\mathbf{0}_{2^{m-h}}\}, & \text{otherwise.} \end{cases} \tag{3.32}$$

(4)

$$\log_2 A_{x,y;z} = \begin{pmatrix} m-h-1 \\ r-h \end{pmatrix}. \tag{3.33}$$

△△

Based on (3.22), (3.32), and (3.33), we can evaluate exactly the computational complexity of the RMLD algorithm for a RM code with binary sectionalization.

Example 3 A $(64,40)$ subcode of $\mathrm{RM}_{3,6}$ is being considered as an inner code in a concatenated coding system for NASA's high-speed satellite communications [3]. The overall decoder for the RM subcode consists of 32 identical maximum likelihood decoding (MLD) decoders, each such decoder processes a $(64,35)$ subcode C or its coset in parallel. In this example, we assume that an RMLD decoder based on binary sectionalization is used, and consider how to choose the $(64,35)$ subcode C of $\mathrm{RM}_{3,6}$ to minimize the total number γ of additions and comparisons in ACS. We assume that a subset M of $M_{r,m}$ satisfies the closure condition (CS). Let $C(M)$ denote the subcode of $\mathrm{RM}_{r,m}$ spanned by M, and let t be a monomial in M whose superterms are not in M. For $0 \leq h < m$, it follows from (3.26) and (3.28) that the difference between the values of $\log_2 |L_{0,2^{m-h}}|$ (or

$\log_2 A_{0,2^{m-h};2^{m-h-1}}$) for $C(M)$ and $C(M\backslash\{t\})$ is independent of M. Let $\triangle L^{(h)}(t)$ (or $\triangle A^{(h)}(t)$) denote the difference. Table 3.1 shows $\triangle L^{(h)}(t)$ with $1 \leq h \leq 5$ and $\triangle A^{(h)}(t)$ with $0 \leq h \leq 5$ for $t \in M_{3,6}\backslash M_{1,6}$. For a subset $\triangle M$ of M consisting of different monomials $t_1, t_2, \ldots, t_k$ such that there is no superterm of t_i in $M\backslash\{t_1, t_2, \ldots, t_{i-1}\}$, the difference between the values of $\log_2 |L_{0,2^{m-h}}|$ (or $\log_2 A_{0,2^{m-h};2^{m-h-1}}$) for $C(M)$ and $C(M\backslash\triangle M)$ is given by $\sum_{i=1}^{k} \triangle L^{(h)}(t_i)$, denoted $\triangle L^{(h)}(\triangle M)$ (or $\sum_{i=1}^{k} \triangle A^{(h)}(t_i)$, denoted $\triangle A^{(h)}(\triangle M)$). The total number γ for $C(M_{6,3}\backslash\triangle M)$ with $|\triangle M| = 7$ is given by

$$\begin{aligned}
\gamma &= (2 \cdot 2^{10-\triangle A^{(0)}(\triangle M)} - 1) \cdot 2^0 \cdot 2^0 \\
&+ (2 \cdot 2^{6-\triangle A^{(1)}(\triangle M)} - 1) \cdot 2^{10-\triangle L^{(1)}(\triangle M)} \cdot 2^1 \\
&+ (2 \cdot 2^{3-\triangle A^{(2)}(\triangle M)} - 1) \cdot 2^{10-\triangle L^{(2)}(\triangle M)} \cdot 2^2 \\
&+ (2 \cdot 2^{1-\triangle A^{(3)}(\triangle M)} - 1) \cdot 2^{7-\triangle L^{(3)}(\triangle M)} \cdot 2^3 \\
&+ (2 \cdot 2^{0-0} - 1) \cdot 2^{4-\triangle L^{(4)}(\triangle M)} \cdot 2^4 \\
&+ (2 \cdot 2^{0-0} - 1) \cdot 2^{2-0} \cdot 2^5. \qquad (3.34)
\end{aligned}$$

There are two cases to be considered.

(i) From the closure condition (CS) and $|\triangle M| = 7$, $\triangle M$ can contain at most one monomial in $M_{2,6}$. Consider the case where $\triangle M$ contains one monomial t in $M_{2,6}$. By considering the effect to γ by t and its all superterms based on Table 3.1, we see that we should choose $x_1x_2, x_1x_2x_i$ with $3 \leq i \leq 6$, and then $x_1x_3x_4$ and $x_2x_3x_4$. This choice of $\triangle M$, denoted ΔM_1, yields the smallest $\gamma = 7039$. If we choose one monomial of form $x_{i_1}x_{i_2}x_5$ other than $x_1x_2x_5$ instead of one of $x_1x_3x_4$ and $x_2x_3x_4$, then we have $\gamma = 8959$, which is the third smallest value of γ.

(ii) $\triangle M \subseteq M_{3,6}\backslash M_{2,6}$: Since each monomial of degree 3 can be chosen independently without violating the closure condition (CS), from Table 3.1 and (3.34) we see that we should choose $x_1x_2x_3$ first, $\{x_{i_1}x_{i_2}x_4 : 1 \leq i_1 < i_2 < 4\}$ next and any three monomials in $\{x_{i_1}x_{i_2}x_5 : 1 \leq i_1 < i_2 < 5\}$. For this choice of $\triangle M$, denoted ΔM_2, $\gamma = 7807$, which is the second smallest value of γ.

Table 3.2 summarizes these choices of $\triangle M$. An RMLD decoder for $C(M_{3,6}\backslash\triangle M_2)$ is designed with VLSI [1], and a decoder for $C(M_{3,6}\backslash \triangle M_1)$ is being designed.

3.5 Conclusion

We have studied a class of linear block codes which are transitive invariant. A condition for a code to be transitive invariant has been proved. We have shown that transitive invariant block codes have uniform structure. This structure is advantageous for implementation of the RMLD algorithm based on a binary uniform sectionalization.

It follows from Examples 1 and 3 that $C(M_{3,6} \setminus \Delta M'_2)$ with the smallest value of γ and $C(M_{3,6} \setminus \Delta M'_3)$ with the second smallest value of γ are subcodes of $C(M_{3,6} \setminus \Delta M_1)$ which has the second smallest number of minimum weight codewords and that $C(M_{3,6} \setminus \Delta M'_1)$ with the third smallest value of γ is a subcode of $C(M_{3,6} \setminus \Delta M_0)$ with the smallest number of minimum weight codewords.

Table 3.1. $\Delta L^{(h)}(t)$ and $\Delta A^{(h-1)}(t)$ with $1 \leq h \leq 5$ for $t \in M_{3,6} \setminus M_{1,6}$.

	$\Delta L^{(h)}(t)^{\dagger}$				$\Delta A^{(h)}(t)^{\dagger}$			
$t \quad \setminus \quad h$	1	2	3	4	0	1	2	3
$x_4x_5x_6$	-1	-1	-1	0	-1	-1	-1	1
$x_ix_5x_6$ $(1 \leq i < 4)$	-1	-1	0	0	-1	-1	1	0
$x_{i_1}x_{i_2}x_6$ $(1 \leq i_1 < i_2 < 5)$	-1	0	0	0	-1	1	0	0
$x_{i_1}x_{i_2}x_5$ $(1 \leq i_1 < i_2 < 5)$	1	0	0	0	1	0	0	0
$x_{i_1}x_{i_2}x_4$ $(1 \leq i_1 < i_2 < 4)$	1	1	0	0	1	0	0	0
$x_1x_2x_3$	1	1	1	0	1	0	0	0
x_5x_6	-1	-1	0	0	-1	-1	1	0
x_ix_6 $(1 \leq i < 5)$	-1	0	0	0	-1	1	0	0
x_ix_5 $(1 \leq i < 5)$	1	0	0	0	1	0	0	0
x_ix_4 $(1 \leq i < 4)$	1	1	0	0	1	0	0	0
x_ix_3 $(1 \leq i < 3)$	1	1	1	0	1	0	0	0
x_1x_2	1	1	1	1	1	0	0	0

$^{\dagger}\Delta L^{(5)}(t) = \Delta A^{(4)}(t) = \Delta A^{(5)}(t) = 0$ for $t \in M_{3,6} \setminus M_{1,6}$.

Table 3.2. $\Delta L^{(h)}(\Delta M')$ and $\Delta A^{(h-1)}(\Delta M')$ with $1 \leq h \leq 5$ and γ for $C(M_{3,6} \setminus \Delta M')$.

	$\Delta L^{(h)}(t)^{\ddagger}$				$\Delta A^{(h)}(t)^{\ddagger}$		γ
$\Delta M' \ \setminus \ h$	1	2	3	4	1	2	
$\Delta M'_1$	5	4	2	1	5	1	8959
$\Delta M'_2$	5	5	2	1	5	1	7039
$\Delta M'_3$	7	4	1	0	7	0	7807

[‡] $\Delta L^{(5)}(t) = \Delta A^{(3)}(t) = \Delta A^{(4)}(t) = \Delta A^{(5)}(t) = 0$ for $C(M_{3,6} \setminus \Delta M')$.

Chapter 4

An Improvement to GMD-like Decoding Algorithms

4.1 Definitions

Suppose a binary (N, K) linear block code C with minimum weight $d_{\min}$ is used for error control over the AWGN channel using BPSK signaling. Each codeword is transmitted with the same probability. Let $\boldsymbol{r} = (r_1, r_2, \ldots, r_N)$ be a received sequence and let $\boldsymbol{z} = (z_1, z_2, \ldots, z_N)$ be the binary hard-decision sequence obtained from $\boldsymbol{r}$ using the hard-decision function: $z_i = 1$ for $r_i > 0$ and $z_i = 0$ for $r_i \leq 0$.

For a positive integer n, let V^n denote the vector space of all binary n-tuples. For $\boldsymbol{u} = (u_1, u_2, \ldots, u_N) \in V^N$, the correlation between $\boldsymbol{u}$ and the received sequence $\boldsymbol{r}$ is given by $M(\boldsymbol{u}) = \sum_{i=1}^{N} r_i(2u_i - 1)$. Then $M(\boldsymbol{z}) = \sum_{i=1}^{N} |r_i| \geq M(\boldsymbol{u})$ for any $\boldsymbol{u} \in V^N$. Define $D_{-1}(\boldsymbol{u}) \triangleq \{i : u_i \neq z_i, \text{ and } 1 \leq i \leq N\}$ and

$$L(\boldsymbol{u}) \triangleq \sum_{i \in D_{-1}(\boldsymbol{u})} |r_i| = (M(\boldsymbol{z}) - M(\boldsymbol{u}))/2. \tag{4.1}$$

$L(\boldsymbol{u})$ is called the **correlation discrepancy** of $\boldsymbol{u}$ with respect to $\boldsymbol{z}$. For a subset U of V^N, let $\underline{L}[U]$ be defined as

$$\underline{L}[U] \triangleq \begin{cases} \min_{\boldsymbol{u} \in U} L(\boldsymbol{u}), & \text{for } U \neq \phi, \\ \infty, & \text{for } U = \phi. \end{cases} \tag{4.2}$$

If $U \neq \phi$, let $\boldsymbol{v}[U]$ denote an N-tuple in U such that $L(\boldsymbol{v}[U]) = \underline{L}[U]$.

The maximum likelihood decoder decodes the received sequence $\boldsymbol{r}$ into the **optimal codeword** $\boldsymbol{c}_{\text{opt}}$ for which

$$L(\boldsymbol{c}_{\text{opt}}) = \underline{L}[C]. \tag{4.3}$$

If $\boldsymbol{z}$ is a codeword, then $\boldsymbol{z}$ is the optimal codeword. A candidate codeword $\boldsymbol{c}$ is said to be better (or more likely) than another candidate codeword $\boldsymbol{c}'$ if $L(\boldsymbol{c}) \leq L(\boldsymbol{c}')$. A candidate codeword $\boldsymbol{c}$ is said to be the best if $L(\boldsymbol{c})$ is the minimum among a specified set of candidate codewords.

For integers i and i' with $1 \leq i \leq i' \leq N$, define $[i, i'] \triangleq \{i, i+1, \ldots, i'\}$, and for $\boldsymbol{u} = (u_1, u_2, \ldots, u_N) \in V^N$, define $p_{i,i'}(\boldsymbol{u}) \triangleq (u_i, u_{i+1}, \ldots, u_{i'})$ and for $\boldsymbol{u}' \in V^N$, let $d_{\mathrm{H},i,i'}(\boldsymbol{u}, \boldsymbol{u}')$ denote the Hamming distance between $p_{i,i'}(\boldsymbol{u})$ and $p_{i,i'}(\boldsymbol{u}')$. For simplicity, we assume that the bit positions $1, 2, \ldots, N$ are ordered according to the reliability order given as follows:

$$|r_i| \leq |r_j|, \text{ for } 1 \leq i < j \leq N. \tag{4.4}$$

4.2 GMD-Like decoding

For nonnegative integers s and t such that $s + 2t < d_{\min}$ and $\boldsymbol{v} \in V^N$, the decoding which corrects s erasures in the first s bit positions and t or less errors in the remaining bit positions of input $\boldsymbol{v}$ is called **(s,t)-decoding with respect to $\boldsymbol{v}$**. Then, the (s,t)-decoding with respect to $\boldsymbol{v}$ outputs a unique codeword, if exists, in

$$R \triangleq \{\boldsymbol{x} \in V^N : d_{\mathrm{H},s+1,N}(\boldsymbol{x}, \boldsymbol{v}) \leq t\}. \tag{4.5}$$

R is called the search region of (s,t)-decoding with respect to $\boldsymbol{v}$. Define

$$\rho \triangleq (d_{\min} + p)/2, \tag{4.6}$$

where

$$p = \begin{cases} 0, & \text{for even } d_{\min}, \\ 1, & \text{for odd } d_{\min}. \end{cases} \tag{4.7}$$

For $\boldsymbol{v} \in V^N$, a GMD-like decoding with search center $\boldsymbol{v}$, denoted GMD($\boldsymbol{v}$), is defined as the iterative decoding consisting of ρ stages whose j-th stage is the $(2j-p-1, \rho-j)$-decoding with respect to $\boldsymbol{v}$ for $1 \le j \le \rho$. The original GMD proposed by Forney [15] is GMD($\boldsymbol{z}$). From (4.5), the search region of the j-th stage, denoted $R(\boldsymbol{v}, j)$, is

$$R(\boldsymbol{v}, j) = \{\boldsymbol{x} \in V^N : d_{\mathrm{H},2j-p,N}(\boldsymbol{x}, \boldsymbol{v}) \le \rho - j\}. \tag{4.8}$$

Since the union of search regions from the first stage to the j-th stage, denoted $R_{\mathrm{p}}(\boldsymbol{v}, j)$, is

$$R_{\mathrm{p}}(\boldsymbol{v}, j) = \bigcup_{j'=1}^{j} \{\boldsymbol{x} \in V^N : d_{\mathrm{H},2j'-p,N}(\boldsymbol{x}, \boldsymbol{v}) \le \rho - j'\}, \tag{4.9}$$

the region which has not been searched (for candidate codewords) yet up to the j-th stage of GMD($\boldsymbol{v}$), denoted $\bar{R}_{\mathrm{p}}(\boldsymbol{v}, j)$, is given by

$$\bar{R}_{\mathrm{p}}(\boldsymbol{v}, j) = \{\boldsymbol{x} \in V^N : d_{\mathrm{H},2j'-p,N}(\boldsymbol{x}, \boldsymbol{v}) > \rho - j' \text{ for } 1 \le j' \le j\}. \tag{4.10}$$

Define $\bar{R}_{\mathrm{GMD}(\boldsymbol{v})} \triangleq \bar{R}_{\mathrm{p}}(\boldsymbol{v}, \rho)$, which denotes the region which is not searched for candidate codewords by GMD($\boldsymbol{v}$).

For a positive integer h, h-GMD decoding is defined as an iterative decoding algorithm which consists of successive GMD($\boldsymbol{v}^{(1)}$), GMD($\boldsymbol{v}^{(2)}$), ..., GMD($\boldsymbol{v}^{(h)}$). For $i \ge 1$, the N-tuple $\boldsymbol{v}^{(i)} \in V^N$ is called the i-th search center of the h-GMD decoding. The first search center $\boldsymbol{v}^{(1)}$ is chosen as the hard-decision sequence $\boldsymbol{z}$, i.e. $\boldsymbol{v}^{(1)} \triangleq \boldsymbol{z}$. For $i > 1$, the i-th search center $\boldsymbol{v}^{(i)}$ is chosen as the best word in the region which has not been searched by $(i-1)$-GMD decoding, that is,

$$\begin{aligned} \boldsymbol{v}^{(i)} &= \boldsymbol{v}\left[\bigcap_{i'=1}^{i-1} \bar{R}_{\mathrm{GMD}(\boldsymbol{v}^{(i')})}\right] \\ &= \boldsymbol{v}[\boldsymbol{x} \in V^N : d_{\mathrm{H},2j'-p,N}(\boldsymbol{x}, \boldsymbol{v}^{(i')}) > \rho - j' \\ &\qquad\qquad \text{for } 1 \le j' \le j \text{ and } 1 \le i' < i\}]. \end{aligned} \tag{4.11}$$

It is shown in [12] that $\boldsymbol{v}^{(2)} = (v_1^{(2)}, v_2^{(2)}, \ldots, v_N^{(2)})$ is given by

$$v_l^{(2)} = \begin{cases} z_l + 1, & \text{if } l+p \text{ is even and } 1 \le (l+p)/2 \le \rho, \\ z_l, & \text{otherwise.} \end{cases} \tag{4.12}$$

A simple and efficient algorithm for finding $\boldsymbol{v}^{(3)}$ has been derived [23]. For $1 \leq i \leq h$ and $1 \leq j \leq \rho$, the j-th stage of the i-th GMD($\boldsymbol{v}^{(i)}$) of h-GMD decoding is also called the (i,j)-th stage. Let $C_j^{(i)}$ denote the set of candidate codewords obtained from the $(1,1)$-th stage to the (i,j)-th stage. If $C_j^{(i)} \neq \phi$, let $\boldsymbol{c}_{\mathrm{best}}^{(i)}(j)$ be the best candidate codeword in $C_j^{(i)}$. If $C_j^{(i)} = \phi$, for convenience, define $\boldsymbol{c}_{\mathrm{best}}^{(i)}(j) \triangleq *$ and $L(*) = \infty$. Then

$$L(\boldsymbol{c}_{\mathrm{best}}^{(i)}(j)) = \underline{L}[C_j^{(i)}]. \tag{4.13}$$

After the (h,ρ)-th stage, $\boldsymbol{c}_{\mathrm{best}}^{(h)}(\rho)$ is output as the decoded codeword. If $\boldsymbol{c}_{\mathrm{best}}^{(h)}(\rho) = *$, then the decoding fails.

4.3 Early Termination Conditions

To reduce the number of (s,t)-decodings without any loss of error performance, we introduce new effective sufficient conditions on the optimality of decoded codewords as early termination conditions. Just after the (i,j)-th stage, $(\bigcap_{i'=1}^{i-1} \bar{R}_{\mathrm{GMD}(\boldsymbol{v}^{(i')})}) \cap \bar{R}_{\mathrm{p}}(\boldsymbol{v}^{(i)}, j)$, denoted $\bar{R}(i,j)$, is the region which has not yet been searched for candidate codewords. Suppose that $\boldsymbol{c}_{\mathrm{best}}^{(i)}(j) \neq *$. If there is a better candidate than $\boldsymbol{c}_{\mathrm{best}}^{(i)}(j)$, then it is in the following region:

$$\bar{R}(i,j) \cap \bigcap_{\boldsymbol{c} \in C_j^{(i)}} \bar{O}_{d_{\min}}(\boldsymbol{c}), \tag{4.14}$$

where $\bar{O}_{d_{\min}}(\boldsymbol{c}) \triangleq \{\boldsymbol{x} \in V^N : d_{\mathrm{H},1,N}(\boldsymbol{x}, \boldsymbol{c}) \geq d_{\min}\}$. Hence, the following condition $\mathrm{Cond}_{\mathrm{S}}^{(i)}(j)$ is a sufficient condition on the optimality of $\boldsymbol{c}_{\mathrm{best}}^{(i)}(j)$:

$$\mathrm{Cond}_{\mathrm{S}}^{(i)}(j) : L(\boldsymbol{c}_{\mathrm{best}}^{(i)}(j)) \leq \underline{L}\left[\bar{R}(i,j) \cap \bigcap_{\boldsymbol{c} \in C_j^{(i)}} \bar{O}_{d_{\min}}(\boldsymbol{c})\right]. \tag{4.15}$$

Since simulation results show that two or more candidate codewords are generated very rarely by GMD-like decoding, the following simpler condition has almost the same effectiveness:

$$\mathrm{Cond}_{\mathrm{S},1}^{(i)}(j) : L(\boldsymbol{c}_{\mathrm{best}}^{(i)}(j)) \leq \underline{L}\left[\bar{R}(i,j) \cap \bar{O}_{d_{\min}}(\boldsymbol{c}_{\mathrm{best}}^{(i)}(j))\right]. \tag{4.16}$$

For $1 \leq l \leq 4$, the following sufficient condition $\mathrm{Cond}_{\mathrm{S},l}$ on the optimality of decoded codewords which are independent of search regions have been derived [14, 21, 22].

$$\mathrm{Cond}_{\mathrm{S},l} : L(\boldsymbol{c}_{\mathrm{best}}) \leq \underline{L}\left[\bigcap_{\boldsymbol{c}\in C_l} \bar{O}_{d_{\min}}(\boldsymbol{c})\right], \tag{4.17}$$

where C_l denotes the $\min\{l, |C_j^{(i)}|\}$ best codewords among the candidate codewords generated up to the current (i,j)-th stage and $\boldsymbol{c}_{\mathrm{best}}$ denotes the best in C_l. $\mathrm{Cond}_{\mathrm{S},1}$ is called Taipale-Pursley condition [14]. $\mathrm{Cond}_{\mathrm{S},1}^{(i)}(j)$ is stronger than $\mathrm{Cond}_{\mathrm{S},1}$, because $\bar{R}(i,j)$ is taken into account. As shown in Section 4.4, the condition $\mathrm{Cond}_{\mathrm{S},1}^{(1)}(j)$ is more effective than the condition $\mathrm{Cond}_{\mathrm{S},1}$.

For simplicity, the following sufficient conditions on the optimality of decoded codewords are used as early termination conditions at the (i,j)-th stage of 3-GMD decoding in the simulation reported in Section 4.4.

(i) $i = 1$:
$\mathrm{Cond}_{\mathrm{S},1}^{(1)}(j)$, and $\mathrm{Cond}_{\mathrm{S},2}$,

(ii) $i = 2$:
$\mathrm{Cond}_{\mathrm{S},0}^{(2)}$:$L(\boldsymbol{c}_{\mathrm{best}}^{(2)}(j)) \geq \underline{L}[\bar{R}(2,j)]$, and $\mathrm{Cond}_{\mathrm{S},2}$,

(iii) $i = 3$:
$\mathrm{Cond}_{\mathrm{S},2}$.

The above conditions will be called $\mathrm{Cond}_{\mathrm{S,NEW}}$. Simulation results show that $\mathrm{Cond}_{\mathrm{S},1}^{(1)}(j)$ and $\mathrm{Cond}_{\mathrm{S},0}^{(2)}$ are more effective than $\mathrm{Cond}_{\mathrm{S},2}$.

4.4 Simulation Results of h-GMD decoding with $2 \leq h \leq 3$

Figures 4.1 to 4.3 show simulation results of block error probabilities for EBCH(64, 24), EBCH(128, 85) and EBCH(128, 99) with the minimum distance, 16, 14 and 10, respectively. For comparison, the block error probabilities for bounded distance-$t_0(\triangleq \lfloor (d_{\min} - 1)/2 \rfloor)$ decoding and Chase decoding algorithm II [19] are shown.

Tables 4.1 and 4.2 show the following statistics with respect to signal to noise ratios 2.0 E_b/N_0 and 4.0 E_b/N_0 in dB for EBCH(64, 24), respectively. The number of trials is 50000. In the first part, the number of trials in which $\boldsymbol{z} \in C$ is shown. The remaining parts concern those trials in which $\boldsymbol{z} \notin C$. The second part concerns bounded distance-t_0 decoding with input $\boldsymbol{z}$, and the numbers of decoding failures, correct decoding and incorrect decoding are shown, respectively. The third part shows the numbers of decoding failure, correct decoding and incorrect decoding, respectively, of 3-GMD decoding. The first subpart shows the occurrence number of the event that the first candidate codeword is generated at the (i, j)-th stage for $1 \leq j \leq d_{\min}/2$ and $1 \leq i \leq 3$. The second to fourth subparts show the occurrence number of the event that the best candidate codeword (= the output of 3-GMD decoding) is generated at the (i, j)-th stage and the numbers of $\boldsymbol{c}_{\mathrm{best}^{(i)}(j)}$'s which satisfy sufficient conditions on the optimality of decoded codewords $\mathrm{Cond}_{\mathrm{S},1}$ and $\mathrm{Cond}_{\mathrm{S,NEW}}$, respectively. $\mathrm{Cond}_{\mathrm{S},1}$ has effect only for early stages.

The number of iteration of h-GMD decoding is at most $h(d_{\min} + p)/2$. This number of iteration can be reduced considerably by using $\mathrm{Cond}_{\mathrm{S,NEW}}$. Define the rate of reduction as the ratio of the number of iterations of (s, t)-decoding to 3ρ. Since sufficient conditions can be used only when at least one candidate codeword has been generated, rates μ_{TP} and μ_{NEW} are the averages of reduction rates by using $\mathrm{Cond}_{\mathrm{S},1}$ and $\mathrm{Cond}_{\mathrm{S,NEW}}$, respectively, as early termination conditions over all the trials where at least one candidate codeword is generated. Tables 4.3 lists μ_{TP} and μ_{NEW} in percentage for EBCH(64, 24), EBCH(64, 45), EBCH(128, 78), EBCH(128, 85) and EBCH(128, 99) with respect to signal to noise ratios $2.0E_b/N_0$ and $4.0E_b/N_0$.

4.5 Conclusion

We have introduced "multiple GMD decoding" for binary linear block codes. For extended BCH codes, simulation results show that the new approach provides better error performance than that of the original GMD decoding by adding two GMD-like decoding around two appropriately chosen centers to the original GMD decoding with relative small increment of iteration number.

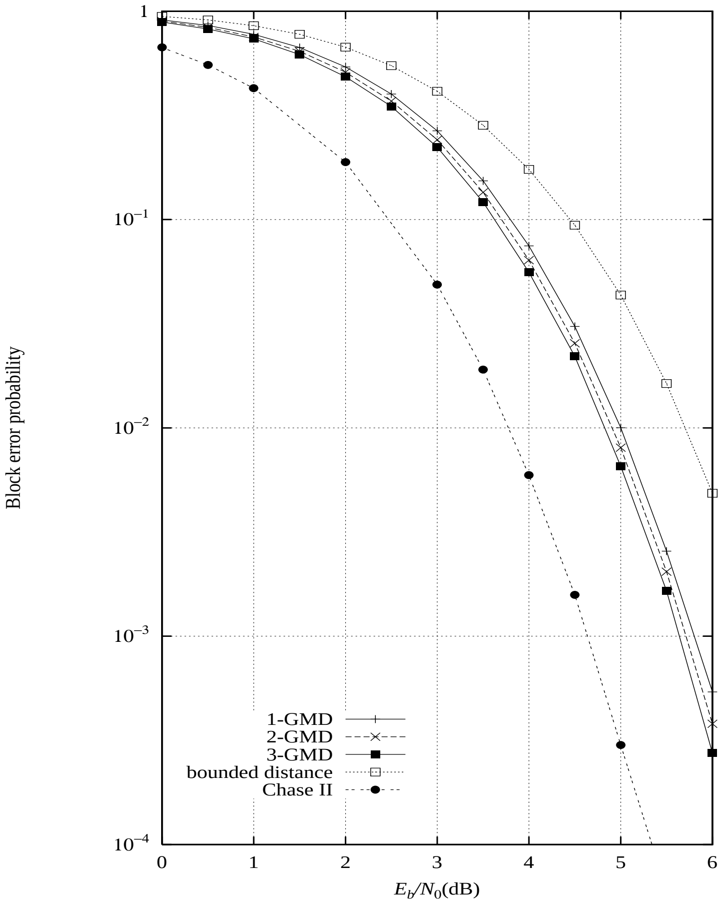

Figure 4.1. Block error probability of EBCH(64, 24)

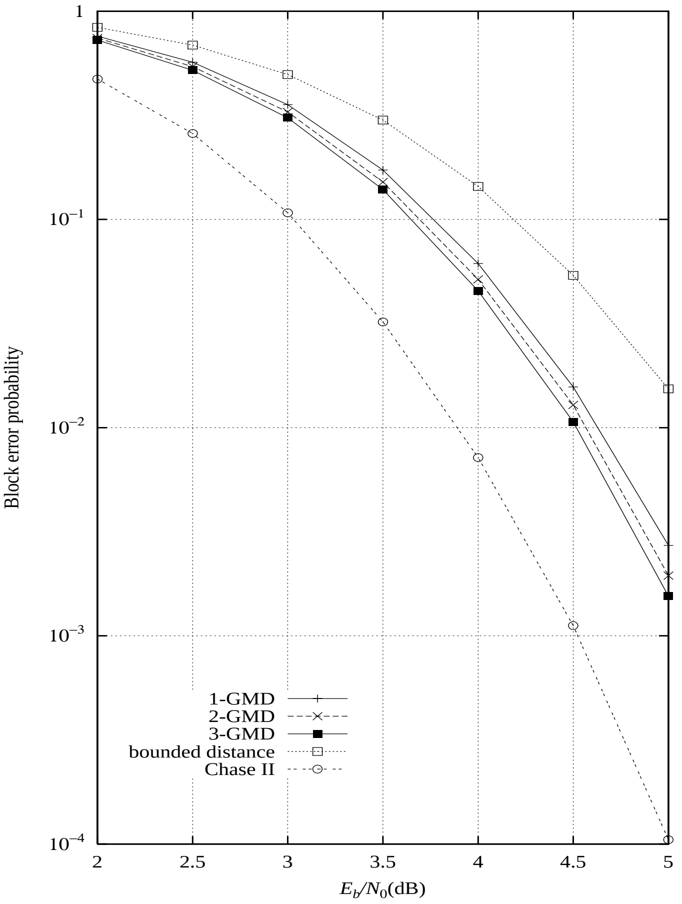

Figure 4.2. Block error probability of EBCH(128, 85)

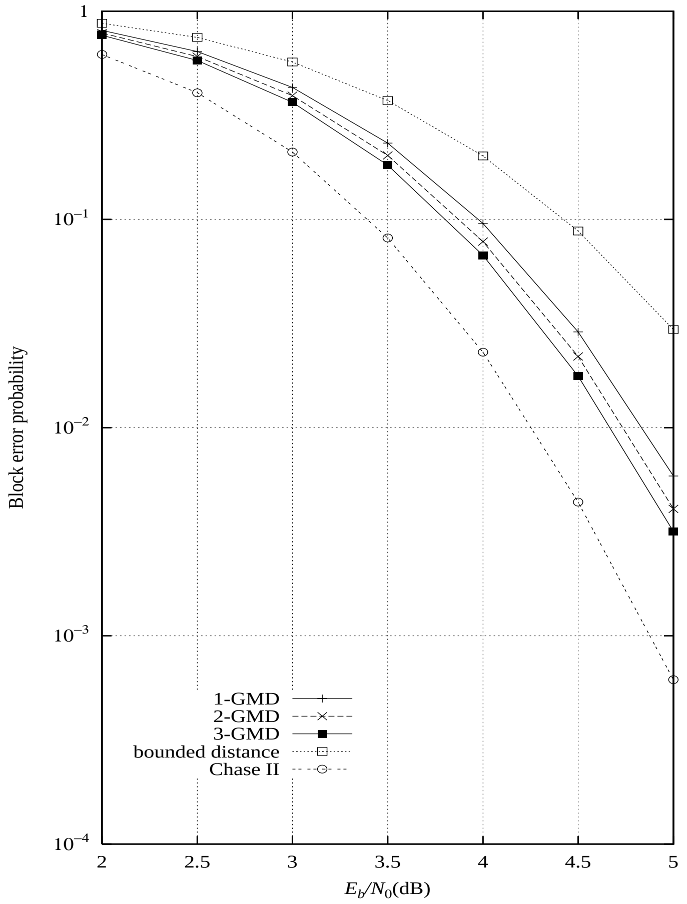

Figure 4.3. Block error probability of EBCH(128, 99)

Table 4.1. EBCH(64, 24) at $E_b/N_0 = 2.0$ dB

$\boldsymbol{z} \in C$									4
bounded distance-$\lfloor (d_{\min} - 1)/2 \rfloor$ decoding						decoding failure			33610
						correct decoding			16373
						incorrect decoding			17
3-GMD decoding						decoding failure			24104
						correct decoding			25774
						incorrect decoding			122
	$i\backslash j$	1	2	3	4	5	6	7	8
the occurance of the first candidate codeword	1	19698	1481	882	485	292	140	38	9
	2	1167	189	85	61	44	21	8	2
	3	939	138	82	57	34	29	10	1
the occurance of the best candidate codeword	1	19690	1482	882	486	292	141	38	9
	2	1166	189	85	61	44	21	8	3
	3	944	138	82	57	34	29	10	1
the number of codewords $\boldsymbol{c}_{\text{best}}^{(i)}(j)$ which satisfy $\text{Cond}_{\text{S},1}$	1	8800	135	57	22	7	2	0	0
	2	0	0	0	0	0	0	0	0
the number of codewords $\boldsymbol{c}_{\text{best}}^{(i)}(j)$ which satisfy $\text{Cond}_{\text{S,new}}$	1	8800	361	253	231	267	302	269	162
	2	47	9	3	2	0	0	1	1

Table 4.2. EBCH(64, 24) at $E_b/N_0 = 4.0$ dB

$\boldsymbol{z} \in C$									192
bounded distance-$\lfloor (d_{\min} - 1)/2 \rfloor$ decoding						decoding failure			8704
						correct decoding			41293
						incorrect decoding			3
3-GMD decoding						decoding failure			2780
						correct decoding			47208
						incorrect decoding			12
	$i\backslash j$	1	2	3	4	5	6	7	8
the occurance of the first candidate codeword	1	43599	1068	622	391	227	110	45	13
	2	366	73	32	40	17	11	4	2
	3	276	51	28	25	7	10	9	2
the occurance of the best candidate codeword	1	43595	1067	623	392	227	111	45	13
	2	368	73	32	40	17	11	4	2
	3	275	51	28	25	7	11	9	2
the number of codewords $\boldsymbol{c}_{\text{best}}^{(i)}(j)$ which satisfy $\text{Cond}_{\text{S},1}$	1	37162	318	123	35	13	1	0	0
	2	0	0	0	0	0	0	0	0
the number of codewords $\boldsymbol{c}_{\text{best}}^{(i)}(j)$ which satisfy $\text{Cond}_{\text{S,new}}$	1	37162	653	400	342	389	426	314	186
	2	88	9	4	4	2	4	1	0

Table 4.3. The average number of iteration (%) of 3-GMD

Code	$E_b/N_0 = 2.0$dB		$E_b/N_0 = 4.0$dB	
	μ_{TP}	μ_{NEW}	μ_{TP}	μ_{NEW}
EBCH(64, 24)	66.7%	61.6%	23.6%	19.8%
EBCH(64, 45)	65.0%	59.1%	17.8%	15.0%
EBCH(128, 78)	78.8%	73.2%	21.6%	18.6%
EBCH(128, 85)	78.4%	74.3%	21.1%	18.1%
EBCH(128, 99)	82.2%	78.3%	24.8%	20.9%

Chapter 5

An Improvement to Chase-like Decoding Algorithm

5.1 Definitions

Suppose a binary linear $(N, K, d_{\min})$ block code C is used for error control over the AWGN channel using BPSK signaling. Each codeword is transmitted with the same probability. For a positive integer n, let V^n denote the vector space of all binary n-tuples. Let $\boldsymbol{r} = (r_1, r_2, \ldots, r_N)$ be a received sequence and let $\boldsymbol{z} = (z_1, z_2, \ldots, z_N)$ be the hard-decision sequence from $\boldsymbol{r}$ by using hard-decision function, $z_i = 1$ for $r_i > 0, z_i = 0$ for $r_i \leq 0$. For simplicity, we assume that the bit positions $1, 2, \ldots, N$ are ordered according to the reliability order given as $|r_i| \leq |r_j|$, for $1 \leq i < j \leq N$. For $\boldsymbol{u} = (u_1, u_2, \ldots, u_N) \in V^N$, the correlation between $\boldsymbol{u}$ and the received sequence $\boldsymbol{r}$ is given by $M(\boldsymbol{u}) = \sum_{i=1}^{N} r_i(2u_i - 1)$. Then, $M(\boldsymbol{z}) = \sum_{i=1}^{N} |r_i| \geq M(\boldsymbol{u})$ for any $\boldsymbol{u} \in V^N$. For $\boldsymbol{u} \in V^N$, define $D_{-1}(\boldsymbol{u}) \triangleq \{i : u_i \neq z_i, \text{ and } 1 \leq i \leq N\}$ and

$$L(\boldsymbol{u}) \triangleq \sum_{i \in D_{-1}(\boldsymbol{u})} |r_i| = (M(\boldsymbol{z}) - M(\boldsymbol{u}))/2.$$

$L(\boldsymbol{u})$ is called the correlation discrepancy of $\boldsymbol{u}$ with respect to $\boldsymbol{z}$. For $U \subseteq V^N$, let $\underline{L}[U]$ be defined as

$$\underline{L}[U] \triangleq \begin{cases} \min_{\boldsymbol{u} \in U} L(\boldsymbol{u}), & \text{for } U \neq \emptyset, \\ \infty, & \text{for } U = \emptyset. \end{cases}$$

The maximum likelihood decoder outputs the optimal codeword $\boldsymbol{c}_{\text{opt}}$ which

$$L(\boldsymbol{c}_{\text{opt}}) = \underline{L}[C]$$

from the received sequence $\boldsymbol{r}$. A codeword $\boldsymbol{c}$ is said to be better than another codeword $\boldsymbol{c}'$ if $L(\boldsymbol{c}) \leq L(\boldsymbol{c}')$. For a nonempty subset $U \subseteq V^N$, a codeword $\boldsymbol{c}$ is said to be the best in U if $L(\boldsymbol{c}) = \underline{L}[U]$. For $U \neq \phi$, let $\boldsymbol{v}[U]$ denote a binary N-tuple in U such that $L(\boldsymbol{v}[U]) = \underline{L}[U]$. For integers i and j such that $1 \leq i < j \leq N$, define $[i,j] \triangleq \{i, i+1, \ldots, j\}$. For $\boldsymbol{u} = (u_1, u_2, \ldots, u_N)$ and $\boldsymbol{v} = (v_1, v_2, \ldots, v_N) \in V^N$, define $d_{H,[i,j]}(\boldsymbol{u}, \boldsymbol{v}) \triangleq |\{k \in [i,j] : u_k \neq v_k\}|$ and for a nonnegative integer t, $O(\boldsymbol{v}, t) \triangleq \{\boldsymbol{u} \in V^N : d_{H,[1,N]}(\boldsymbol{u}, \boldsymbol{v}) \leq t\}$ and $\bar{O}(\boldsymbol{v}, t) \triangleq V^N \setminus O(\boldsymbol{v}, t)$.

For a nonnegative integer $l = \sum_{i=1}^{N} b_i 2^{i-1}$ with $b_i \in \{0, 1\}$, define $b(l) \triangleq (b_1, b_2, \ldots, b_N)$. For $0 \leq l < 2^N$, define $E_l \triangleq \{b(i) : 0 \leq i < l\}$. For a positive integer l less than $N - t_0 + 1$, a Chase-like decoding algorithm, denoted Chase(l), is defined as an iterative decoding algorithm which performs bounded distance-t_0 decoding around $\boldsymbol{z} + \boldsymbol{b}(i)$ for $0 \leq i < l$. The original Chase decoding algorithm II introduced by Chase [24] is Chase($2^{\lfloor d_{\min}/2 \rfloor}$).

5.2 Decoding algorithm

If $\boldsymbol{z} \in C$, then $L(\boldsymbol{z}) = \underline{L}[C]$, that is, $\boldsymbol{z}$ is the optimal decoded codeword. Assume that $\boldsymbol{z} \notin C$.

For $\boldsymbol{v} \in V^N$, the bounded distance-t_0 decoding around the vector $\boldsymbol{v}$, denoted BDD($\boldsymbol{v}$), outputs a unique codeword in $O(\boldsymbol{v}, t_0)$, if it exists.

In this paper, we consider the following iterative decoding algorithm. For a positive integer h, a nonnegative integer t and $\boldsymbol{v}^{(i)} \in V^N$ with $1 \leq i \leq h$, (h,t)-IBDD decoding is defined as an iterative decoding algorithm which consists of successive BDD($\boldsymbol{v}^{(1)}$), BDD($\boldsymbol{v}^{(2)}$), ... ,BDD($\boldsymbol{v}^{(h)}$), where $\boldsymbol{v}^{(i)}$ is called the i-th search center of the (h,t)-IBDD decoding. Define $R(i) \triangleq \cup_{1 \leq i' \leq i} O(\boldsymbol{v}^{(i')}, t_0)$ as the region which has been searched up to the i-th stage of (h,t)-IBDD decoding and $\bar{R}(i) \triangleq V^N \setminus R(i)$.

We adopt the following choice of search centers, called Selection(t). The first search center $\boldsymbol{v}^{(1)}$ is $\boldsymbol{z}$. For $i > 1$, search center $\boldsymbol{v}^{(i)}$ is chosen as

$$\boldsymbol{v}^{(i)} = v[\cap_{1 \leq i' \leq i-1} \bar{O}(\boldsymbol{v}^{(i')}, t)].$$

If $t = t_0$, then $\boldsymbol{v}^{(i)} = v[\bar{R}(i-1)]$.

Let $C(i)$ denote the set of candidate codewords obtained from the first stage to the i-th stage. If $C(i) \neq \emptyset$, let $\boldsymbol{c}_{\text{best}}(i)$ be the best codeword in $C(i)$. If $C(i) = \emptyset$, define $\boldsymbol{c}_{\text{best}}(i) \triangleq *$ and $L(*) = \infty$. Then $L(\boldsymbol{c}_{\text{best}}(i)) = \underline{L}[C(i)]$. After the h-th stage, $\boldsymbol{c}_{\text{best}}(h)$ is output as the decoded codeword. If $\boldsymbol{c}_{\text{best}}(h) = *$, then the decoding fails.

5.3 Early termination conditions

To reduce the number of bounded distance-t_0 decodings in (h,t)-IBDD without any degradation of error performance, we present new effective sufficient conditions on the optimality of decoded codewords as early termination conditions. Suppose that $\boldsymbol{c}_{\text{best}}(i) \neq *$. If there is a better codeword than $\boldsymbol{c}_{\text{best}}(i)$, it is in the following region: $\bar{R}(i) \cap \bigcap_{\boldsymbol{c} \in C(i)} \bar{O}(\boldsymbol{c}, d_{\min}-1)$. The following condition $\text{Cond}_S(i)$ is a sufficient condition on the optimality of $\boldsymbol{c}_{\text{best}}(i)$,

$$\text{Cond}_S(i) : L(\boldsymbol{c}_{\text{best}}(i)) \leq \underline{L}[\bar{R}(i) \cap \bigcap_{\boldsymbol{c} \in C(i)} \bar{O}(\boldsymbol{c}, d_{\min}-1)].$$

For $1 \leq l \leq 4$, the following sufficient condition $\text{Cond}_{S,l}$ on the optimality of decoded codewords which are independent of search regions have been derived.

$$\text{Cond}_{S,l} : L(\boldsymbol{c}_{\text{best}}(i)) \leq \underline{L}\left[\cap_{\boldsymbol{c} \in C_l} \bar{O}(\boldsymbol{c}, d_{\min}-1)\right],$$

where C_l denotes the $\min\{l, |C(i)|\}$ best codewords among the candidate codewords generated up to the current i-th stage. $\text{Cond}_{S,1}$ is Taipale-Pursley condition [12], denoted Cond_{TP}. $\text{Cond}_S(i)$ is stronger than $\text{Cond}_{S,1}$, because $\bar{R}(i)$ is taken into account. $\text{Cond}_S(i)$ is denoted Cond_{NEW}.

5.4 Simulation results

By simulation, we have evaluated the effectiveness of the choice of parameter t in Selection(t) with $1 \leq t \leq t_0$ for several BCH codes [31]. For $t = t_0$, $\boldsymbol{v}^{(i)}$ is the best in the region which was not searched before the i-th stage of (h,t_0)-IBDD decoding. From this fact, t_0 was thought to be a reasonable choice.

The simulation results show, however, that it is better to choose t somewhat smaller than t_0. Figures 5.1, 5.2 and 5.3 show simulation results of block error probabilities of (h, t)-IDD with properly chosen parameter t for BCH(63, 30, 13) code, BCH(63, 45, 7) code and BCH(127, 92, 11), respectively. For comparison, the block error probabilities for bounded distance-t_0 decoding and Chase(h) are shown.

The reduction rate of (h, t)-IBDD is defined as $1 - (\nu/h)$, where ν is the number of iterations of bounded distance-t_0 decoding. Since sufficient conditions can be used only when at least one candidate codeword has been generated, rates μ_{TP} and μ_{NEW} denote the averages of reduction rates by using $\mathrm{Cond}_{\mathrm{TP}}$ and $\mathrm{Cond}_{\mathrm{NEW}}$, respectively, as early termination conditions over all the trials where at least one candidate codeword is generated. Figures 5.4, 5.5, 5.6 and 5.7 show simulation results of the average of reduction rates of $(32, 4)$-IBDD decoding for BCH(63, 30, 13), BCH(63, 45, 7), BCH(127, 85, 13) and BCH(127, 92, 11). To compute search centers $\boldsymbol{v}^{(i)}$ and the right-hand side of $\mathrm{Cond}_S(i)$, an integer programming approach [18] is adopted and *lp_solve 3.0* is tentatively used.

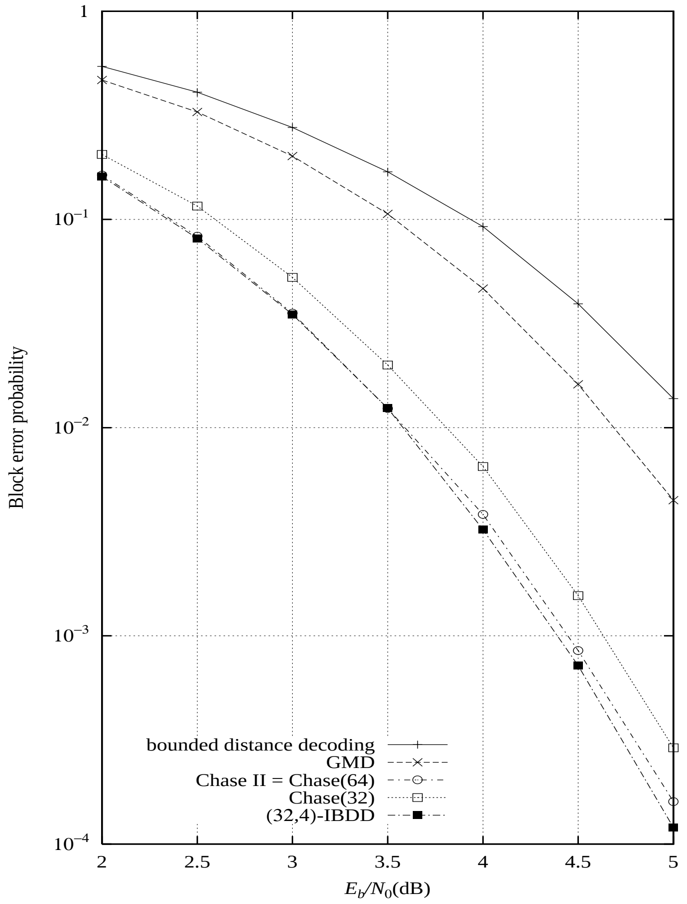

Figure 5.1. Block error probability for BCH(63, 30, 13)

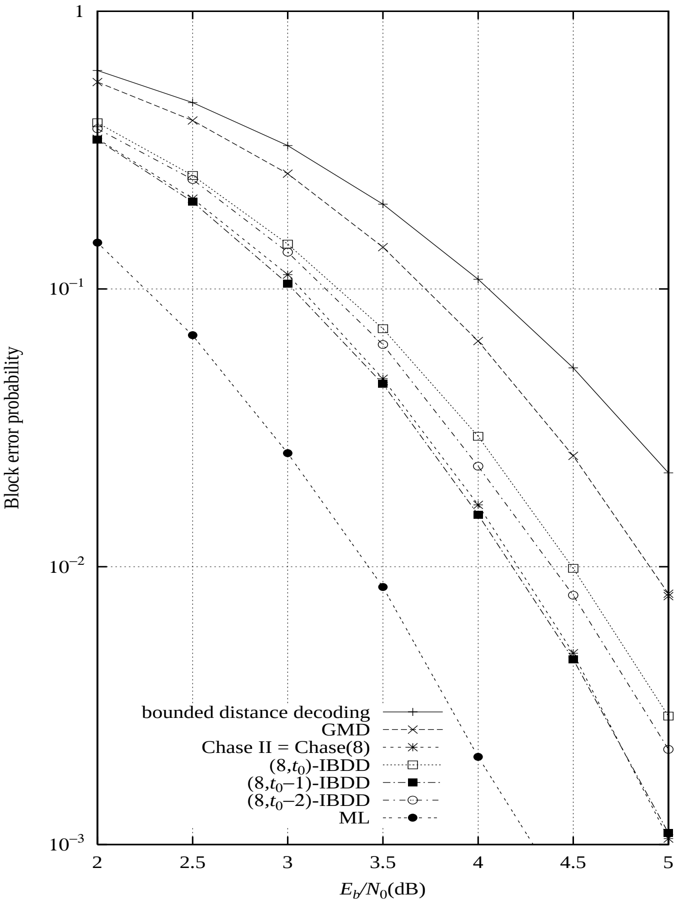

Figure 5.2. Block error probability for BCH(63, 45, 7)

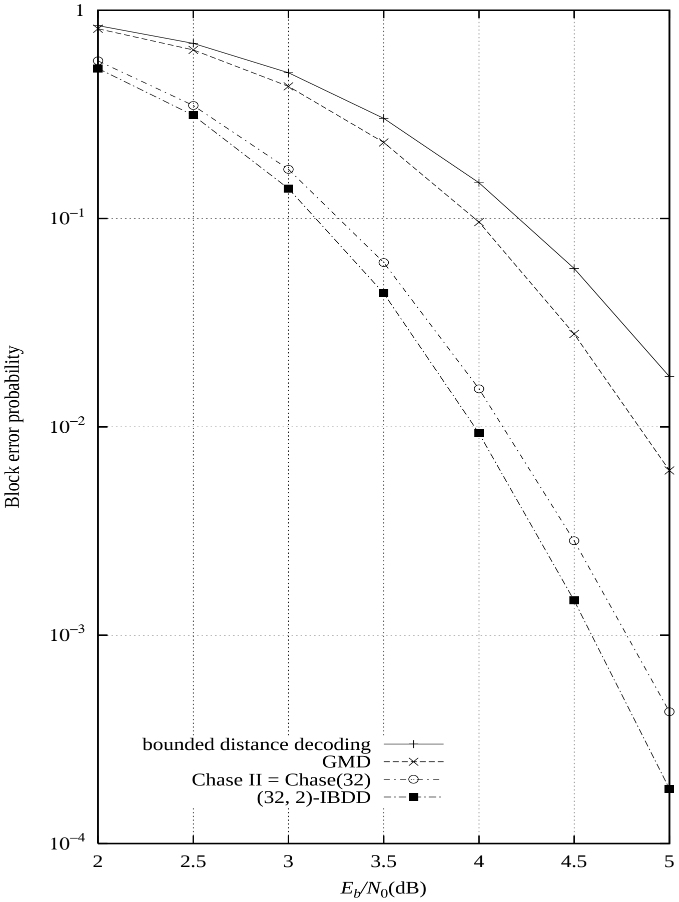

Figure 5.3. Block error probability for BCH(127, 92, 11)

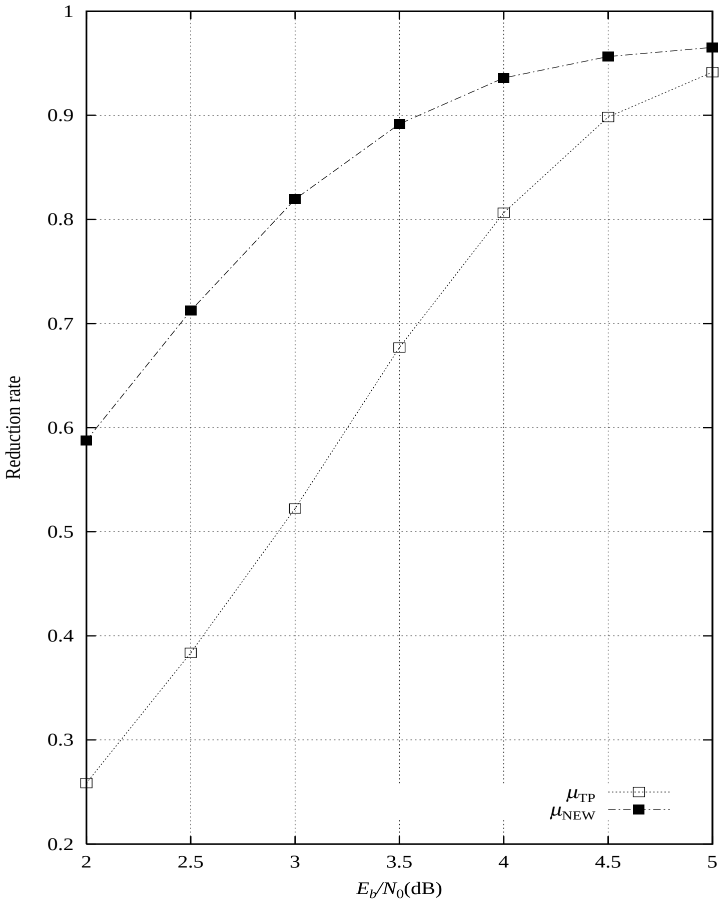

Figure 5.4. Reduction rate of $(32, 4)$-IBDD decoding for BCH(63, 30, 13)

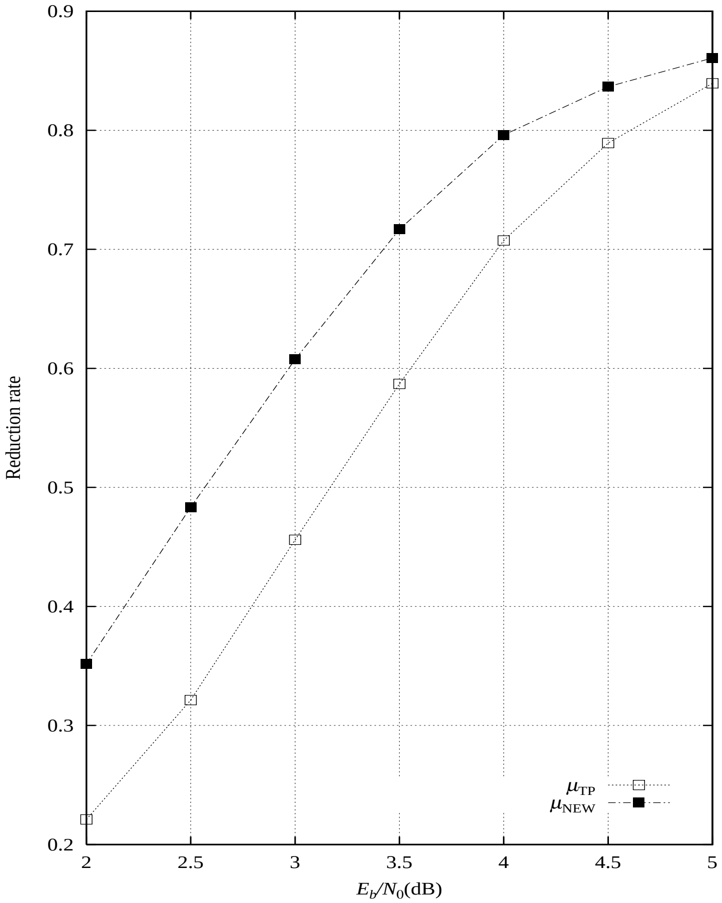

Figure 5.5. Reduction rate of $(32, 2)$-IBDD decoding for BCH(63, 45, 7)

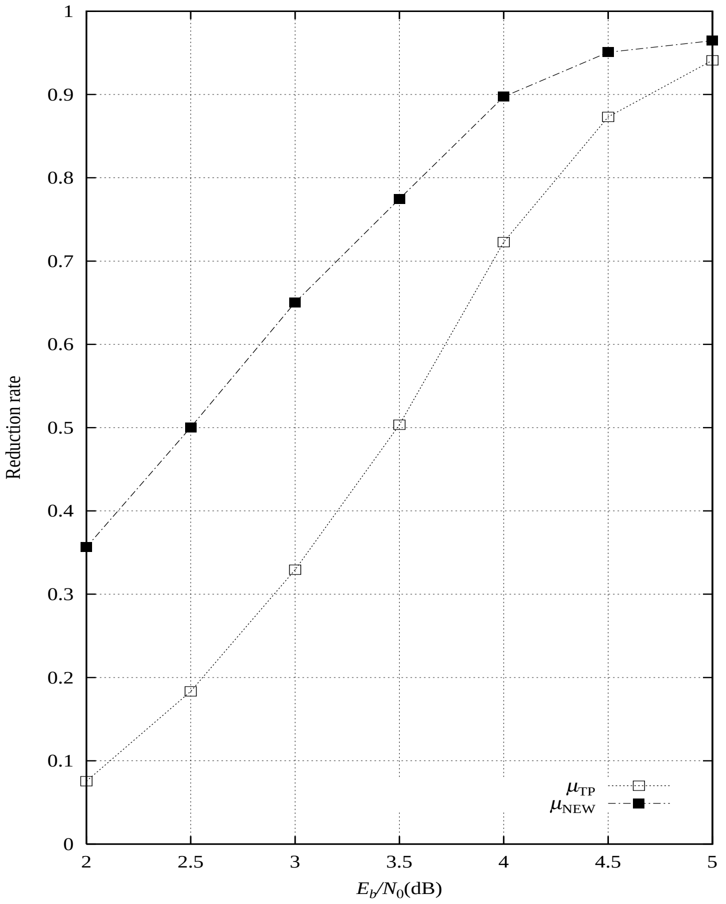

Figure 5.6. Reduction rate of (32, 4)-IBDD decoding for BCH(127, 85, 13)

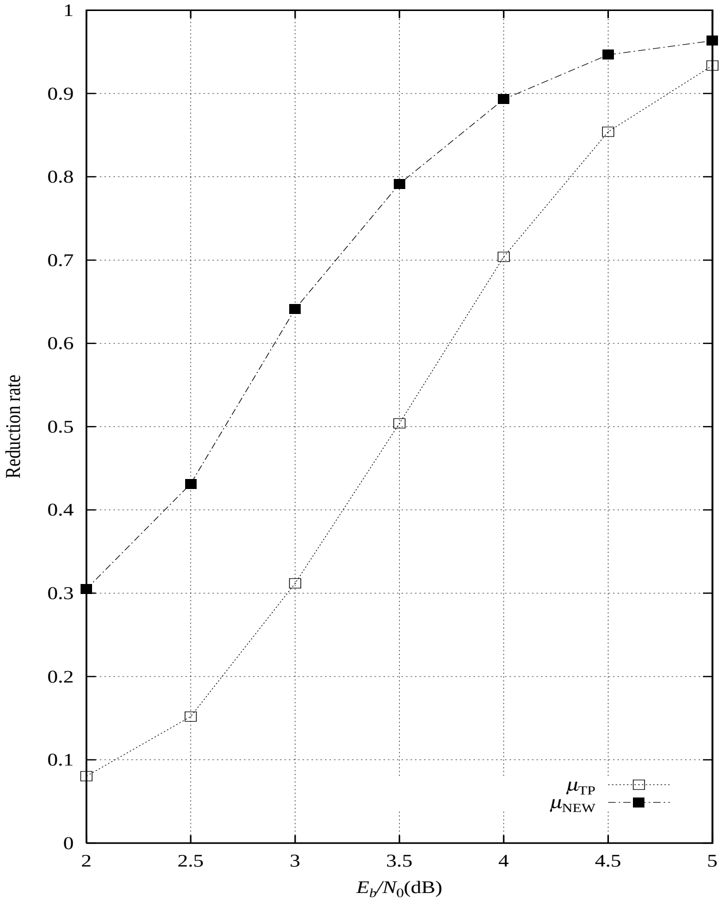

Figure 5.7. Reduction rate of $(32, 2)$-IBDD decoding for BCH(127, 92, 11)

Chapter 6

Conclusion

In this thesis, first, we have presented a formula for an effective method which gives the number of minimum weight codewords in a $(2^m, \sum_{i=0}^{r} \binom{m}{i} - \Delta K)$ linear subcode C of $\mathrm{RM}_{r,m}$ code which is spanned by monomials with m variables of degree r or less over GF(2) for $\Delta K \leq 3$. Next, we have shown in Theorem 1 how to delete ΔK monomials in order to obtain the subcode with the smallest number of codewords of the minimum weight for $r\Delta K \leq m$. In Example 1, we have shown the numbers of minimum weight codewords of all such $(64, 40)$ subcodes of $\mathrm{RM}_{3,6}$ and those error probabilities of soft-decision maximum likelihood decoding by simulation.

Second, we have studied a class of linear block codes which are transitive invariant. A condition for a code to be transitive invariant has been proved. We have shown that transitive invariant block codes have a uniform structure. This structure is advantageous for implementation of the RMLD algorithm based on a binary uniform sectionalization. We have shown that the binary sectionalization results in almost the same computational complexity as an optimum sectionalization for many transitive invariant example codes.

Third, we have introduced “multiple GMD decoding” for binary linear block codes. For extended BCH codes, simulation results show that the new approach provides better error performance than that of the original GMD decoding by adding two GMD-like decodings around two appropriately chosen centers to the original GMD decoding with a relatively small increment of iteration number.

Finally, we have presented a new method of choosing a sequence of search centers around which successive bounded distance-t_0 decodings are carried out. For BCH codes, simulation results show that the new approach provides better error performance than that of Chase-like decoding which consists of the same number of bounded distance-t_0 decodings. To reduce the number of iterations of bounded distance decoding algorithm without any loss of error performance, we show new effective sufficient conditions on the optimality of decoded codewords as early termination conditions.

Appendix

A Formula for Evaluation N_{23}

In this Appendix, we consider the evaluation of N_{23} in a somewhat general framework. In the evaluation of N_{23}, the following B_1, B_2, D_0, D_1 and D_2 represent $A_{n_{10*},n_{011}}A_{n_{1*0},n_{011}}, A_{n_{100},n_{011}}, A_{n_{101},n_{011}}$ and $A_{n_{110},n_{011}}$, respectively. For $h \in \{0,1,2\}$, let D_h be a binary $w_h \times l$ matrix where $w_h > 0$ and $l > 0$, and for $i \in \{1,2\}$, let B_i denote the binary $(w_0 + w_i) \times l$ matrix whose submatrix consisting of the first w_0 rows is D_0 and whose submatrix consisting of the last w_i rows is D_i, that is,

$$B_i \triangleq \begin{bmatrix} D_0 \\ D_i \end{bmatrix}, \text{ for } i \in \{1,2\}. \tag{A$\cdot$1}$$

We consider the following condition:

$$\text{rank}(B_i) = l, \text{ for } i \in \{1,2\}, \tag{A$\cdot$2}$$

that is, the l columns of B_i are linearly independent.
The following conditions is necessary for the condition (A$\cdot$2). Hereafter, we assume it.

$$w_0 + \min\{w_1, w_2\} \geq l. \tag{A$\cdot$3}$$

For the evaluation of N_{23}, w_0, w_1, w_2 and l represent $|n_{100}|$, $|n_{101}|$, $|n_{110}|$ and $|n_{011}|$, respectively, and since $|n_{100}|+|n_{101}| = |n_{10*}| = |n_{01*}|$ and $|n_{100}|+|n_{110}| = |n_{1*0}| = |n_{0*1}|$ as shown in Section 2.3, $|n_{100}| + \min\{|n_{101}|, |n_{110}|\} \geq |n_{011}|$, that is (A$\cdot$3) holds.

For $1 \leq j \leq l$, let $\varphi_{D_0}(j)$ denote the number of linearly independent columns in the first j columns of D_0. Define $\Delta\varphi_{D_0}(j) \triangleq \varphi_{D_0}(j) - \varphi_{D_0}(j-1)$ for $1 \leq j \leq l$, and $\varphi_{D_0}(0) \triangleq 0$. Since $0 \le \Delta\varphi_{D_0}(j) \le 1$ for $1 \le j \le l$ and $\varphi_{D_0}(l) \le \min\{w_0, l\}$, $\Delta\varphi_{D_0} \triangleq (\Delta\varphi_{D_0}(1), \Delta\varphi_{D_0}(2), \dots, \Delta\varphi_{D_0}(l))$ is a binary l-tuple of weight $\min\{w_0, l\}$ or less.

Conversely, for a binary l-tuple $\boldsymbol{v} = (v_1, v_2, \dots, v_l)$ of weight $\min\{w_0, l\}$ or less, define $\phi_{\boldsymbol{v}}(j)$ for $0 \le j \le l$ recursively as follows:

$$\phi_{\boldsymbol{v}}(0) = 0, \tag{A$\cdot$4}$$

$$\phi_{\boldsymbol{v}}(j) = \phi_{\boldsymbol{v}}(j-1) + v_j. \tag{A$\cdot$5}$$

Let $V^l_{w_0}$ denote $\left\{\boldsymbol{v} \in V^l : \text{ the weight of } \boldsymbol{v} \le w_0\right\}$ where V^l denotes the set of binary l-tuples. Then, the following lemma is a direct consequence of the definition of φ_{D_0}.

Lemma 4 For $\boldsymbol{v} \in V^l_{w_0}$, the number of binary $w_0 \times l$ matrices D_0's such that $\varphi_{D_0} = \phi_{\boldsymbol{v}}$ is given by

$$\prod_{j=1}^{l} N_{\phi_{\boldsymbol{v}}}(j), \tag{A$\cdot$6}$$

where

$$N_\phi(j) \triangleq \begin{cases} 2^{w_0} - 2^{\phi_{\boldsymbol{v}}(j-1)}, & \text{if } \phi_{\boldsymbol{v}}(j) > \phi_{\boldsymbol{v}}(j-1), \quad \text{(A$\cdot$7)} \\ 2^{\phi_{\boldsymbol{v}}(j-1)}, & \text{if } \phi_{\boldsymbol{v}}(j) = \phi_{\boldsymbol{v}}(j-1). \quad \text{(A$\cdot$8)} \end{cases}$$

△△

Next, we consider the condition on D_i with $i \in \{1, 2\}$ that $\text{rank}(B_i) = l$ for a given D_0. For $1 \le j \le l$ and $h \in \{0, 1, 2\}$, let $\boldsymbol{d}_{h,j}$ denote the j-th column of D_h, and for $i \in \{1, 2\}$, let $\boldsymbol{b}_{i,j}$ denote the j-th column of B_i. For $1 \le j \le l$, there are two cases:

1. $\varphi_{D_0}(j) > \varphi_{D_0}(j-1)$: Then, $\boldsymbol{d}_{0,j}$ is linearly independent of $\boldsymbol{d}_{0,1}, \dots, \boldsymbol{d}_{0,j-1}$, and therefore, for any $\boldsymbol{d}_{i,j}$, $\boldsymbol{b}_{i,j}$ is linearly independent of $\boldsymbol{b}_{i,1}, \dots, \boldsymbol{b}_{i,j-1}$.

2. $\varphi_{D_0}(j) = \varphi_{D_0}(j-1)$: Then, there is a binary $(j-1)$-tuple $(c_1, c_2, \dots, c_{j-1})$ such that

$$\boldsymbol{d}_{0,j} = \sum_{s=1}^{j-1} c_s \boldsymbol{d}_{0,s}. \tag{A$\cdot$9}$$

Let Γ_j denote the set of all binary $(j-1)$-tuples $(c_1, c_2, \ldots, c_{j-1})$'s for which (A· 9) holds. Then,

$$|\Gamma_j| = 2^{j-1-\varphi_{D_0}(j-1)}. \tag{A· 10}$$

For B_i to satisfy (A· 2), the following inequality must hold for any j such that $\varphi_{D_0}(j) = \varphi_{D_0}(j-1)$:

$$\boldsymbol{d}_{i,j} \neq \sum_{s=1}^{j-1} c_s \boldsymbol{d}_{i,s}, \text{for } (c_1, c_2, \ldots, c_{j-1}) \in \Gamma_j. \tag{A· 11}$$

Conversely suppose that for any j' such that $1 \leq j' < j$ and $\varphi_{D_0}(j') = \varphi_{D_0}(j'-1)$,

$$\boldsymbol{d}_{i,j'} \neq \sum_{s=1}^{j'-1} c_s \boldsymbol{d}_{i,s}, \text{for } (c_1, c_2, \ldots, c_{j'-1}) \in \Gamma_{j'}. \tag{A· 12}$$

Then, $\boldsymbol{b}_{i,1}, \boldsymbol{b}_{i,2}, \ldots, \boldsymbol{b}_{i,j-1}$ are linearly independent. Since (A· 9) holds for any $(c_1, c_2, \ldots, c_{j-1}) \in \Gamma_j$, we see that for different $(c_1, c_2, \ldots, c_{j-1})$ and $(c'_1, c'_2, \ldots, c'_{j-1})$ in Γ_j,

$$\sum_{s=1}^{j-1} c_s \boldsymbol{d}_{i,s} \neq \sum_{s=1}^{j-1} c'_s \boldsymbol{d}_{i,s}, \tag{A· 13}$$

that is,

$$\left| \left\{ \sum_{s=1}^{j-1} c_s \boldsymbol{d}_{i,s} : (c_1, c_2, \ldots, c_{j-1}) \in \Gamma_j \right\} \right| = |\Gamma_j| = 2^{j-1-\varphi_{D_0}(j-1)}. \tag{A· 14}$$

In order that (A· 11) holds, the following inequality must hold from (A· 10):

$$w_i > j - 1 - \varphi_{D_0}(j-1). \tag{A· 15}$$

Suppose that (A· 15) holds for any j such that $1 \leq j \leq l$ and $\varphi_{D_0}(j) = \varphi_{D_0}(j-1)$. Then, for such j, $\boldsymbol{d}_{i,j}$ satisfying (A· 11) can be chosen. Summarizing the above argument, we have the following lemma:

Lemma 5 For a given D_0, there are binary $w_1 \times l$ matrix D_1 and $w_1 \times l$ matrix D_2 for which (A· 2) holds, if and only if for any j such that $1 \leq j \leq l$ and

$\varphi_{D_0}(j) = \varphi_{D_0}(j-1)$, (A$\cdot$15) holds for $i \in \{0,1\}$. If this condition holds, then the number of pairs of matrices D_1 and D_2 satisfying (A$\cdot$2) is given as follows:

$$\prod_{j=1}^{l} N_1'(j) N_2'(j) \tag{A$\cdot$16}$$

where for $i \in \{0,1\}$,

$$N_i'(j) \triangleq \begin{cases} 2^{w_i}, & \text{if } \varphi_{D_0}(j) > \varphi_{D_0}(j-1), \quad \text{(A}\cdot\text{17)} \\ 2^{w_i} - 2^{j-1-\varphi_{D_0}(j-1)}, & \text{if } \varphi_{D_0}(j) = \varphi_{D_0}(j-1). \quad \text{(A}\cdot\text{18)} \end{cases}$$

$\triangle\triangle$

Since $\varphi_{D_0}(j) - \varphi_{D_0}(j-1)$ is 0 or 1 for $1 \le j \le l$, $j - 1 - \varphi_{D_0}(j-1)$ is monotonously nondecreasing as j increases. If $\varphi_{D_0}(j) > \varphi_{D_0}(j-1)$ for all j less than l, then $\varphi_{D_0}(l-1) = l-1$. Hence, (A$\cdot$15) can be replaced by

$$\varphi_{D_0}(l-1) > l - 1 - \min\{w_1, w_2\}. \tag{A$\cdot$19}$$

Define

$$\begin{aligned} V^l_{w_0,w_1,w_2} \quad &\triangleq \quad \{\boldsymbol{v} \in V^l; \text{the weight of } \boldsymbol{v} \le w_0, \\ &\qquad \text{the weight of the first to the } (l-1)\text{-th component} \\ &\qquad\qquad \ge l - \min\{w_1, w_2\}\}. \end{aligned} \tag{A$\cdot$20}$$

From (A$\cdot$3), $V^l_{w_0,w_1,w_2}$ is not empty. Then, for a given D_0, there are D_1 and D_2 satisfying (A$\cdot$2), if and only if

$$\varphi_{D_0} \in V^l_{w_0,w_1,w_2}. \tag{A$\cdot$21}$$

From the definition (A$\cdot$20), we have that

$$\left|V^l_{w_0,w_1,w_2}\right| = 2 \sum_{i=l-\min\{w_1,w_2\}}^{w_0-1} \binom{l-1}{i} + \varepsilon \binom{l-1}{w_0}, \tag{A$\cdot$22}$$

where if $w_0 + \min\{w_1, w_2\} = l$, $\varepsilon = 1$ and otherwise, $\varepsilon = 0$.

The number of matrices D_0, D_1 and D_2 satisfying (A$\cdot$2) is obtained by multiplying (A$\cdot$6), (A$\cdot$16) and (A$\cdot$22). By substituting $|n_{100}|$, $|n_{101}|$, $|n_{110}|$ and $|n_{011}|$ for w_0, w_1, w_2 and l, respectively, N_{23} is derived.

References

[1] T. Fujiwara, T. Komura, T. Onoye, Y. Kaji, T. Kasami and S. Lin, "IC Implementation of a Recursive Maximum Likelihood Decoding Algorithm for Reed-Muller and Related Codes," *Proc. of the 4th International Symposium on Communication Theory and Applications*, pp. 2–7, Lake District, U.K., Jul. 1997.

[2] F.J. MacWilliams and N.J.A. Sloane, *The Theory of Error-Correcting Codes*, Amsterdam, The Netherlands: North-Holland, 1977.

[3] H.T. Moorthy, S. Lin and G.T. Uehara, "Good Trellises for IC Implementation of Viterbi Decoders for Linear Block Codes," *IEEE Trans. Communication*, vol. 45, no. 1, pp. 52–63, Jan. 1997.

[4] W.W. Peterson and E.J. Weldon Jr., *Error Correcting Codes*, 2nd ed., MIT Press, Cambridge, 1972.

[5] T. Fujiwara, H. Yamamoto, T. Kasami and S. Lin, "A Trellis-Based Recursive Maximum Likelihood Decoding Algorithm for Linear Block Codes," *IEEE Trans. Inform. Theory*, vol. 44, no. 2, pp. 714–729, Mar. 1998.

[6] H. Yamamoto, H. Nagano, T. Fujiwara, T. Kasami and S. Lin, "Recursive MLD Algorithm Using the Detail Trellis Structure for a Linear Block Codes and Its Average Complexity Analysis," *Proc. of the International Symposium on Information Theory and Its Applications*, pp. 704–708, Victoria, Canada, Sep. 1996.

[7] T. Kasami, T. Koumoto, T. Fujiwara, H. Yamamoto, Y. Desaki and S. Lin, "Low Weight Subtrellises for Binary Linear Block Codes and Their Appli-

cations," *IEICE Trans. Fundamentals*, vol. E80-A, no. 11, pp. 2095–2103, Nov. 1997.

[8] G.D. Forney Jr., "Dimension/Length Profiles and Trellis Complexity of Linear Block Codes," *IEEE Trans. Inform. Theory*, vol. 40, no. 6, pp. 1741–1752, Nov. 1994.

[9] T. Kasami, T. Takata, T. Fujiwara and S.Lin, "On Complexity of Trellis Structure of Linear Block Codes," *IEEE Trans. Inform. Theory*, vol. 39, no. 3, pp. 1057–1064, 1993.

[10] T. Fujiwara, T. Kasami, R. Morelos-Zaragoza and S. Lin, "The State Complexity of Trellis Diagram for a Class of Generalized Concatenated Codes," *Proc. of the Symposium on Information Theory and Its Applications*, pp. 21–24, Kanazawa, Japan, Oct. 1993.

[11] G. D. Forney, Jr., "Generalized Minimum Distance Decoding," *IEEE Trans. Inform. Theory*, vol. IT-2, pp. 125–181, April 1966.

[12] D. Taipale and M. B. Pursley, "An Improvement to Generalized Minimum-Distance Decoding," *IEEE Trans. Inform. Theory*, vol. 37, pp. 167–172, Jan. 1991.

[13] B. Shen, K. k. Tzeng and C. Wang, "A Bounded-Distance Decoding Algorithm for Binary Linear Block Codes Achieving the Minimum Effective Error Coefficient," *IEEE Trans. Inform. Theory*, vol. 42, pp. 1987–1991, Nov. 1996.

[14] M. P. C. Fossorier and S. Lin, "Generalized Chase and GMD Decodings," *Proc. of the International Symposium on Information Theory and Its Applications*, pp. 415–418, Mexico City, Mexico, Oct. 1998.

[15] T. Koumoto and T. Kasami, "Analysis and Improvement on GMD-like Decoding Algorithms," *Proc. of the International Symposium on Information Theory and Its Applications*, pp. 419–422, Mexico City, Mexico, Oct. 1998.

[16] E. Fishler, O. Amrani and Y. Be'ery, "Geometrical and Performance Analysis of GMD and Chase-Decoding Algorithms," *IEEE Trans. Inform. Theory*, vol. 45, pp. 1402–1422, July 1999.

[17] T. Kasami, Y. Tang, T. Koumoto and T. Fujiwara, "Sufficient Conditions for Ruling-Out Useless Iterative Steps in a Class of Iterative Decoding Algorithms," *IEICE Trans. Fundamentals*, vol. E82-A, pp. 2061–2073, Oct. 1999.

[18] T. Kasami, "On Integer Programming Problems Related to Soft-Decision Iterative Decoding Algorithms," *Proc. of the 13th International Symposium AAECC-13*, Lecture Notes in Computer Science, vol. 1719, pp. 43–54, Springer Verlag, Honolulu, Hawaii, U.S.A., Nov. 1999.

[19] H. Tokushige, T. Koumoto and T. Kasami, "An Improvement to GMD-like Decoding Algorithms," *Proc. of the Symposium on Information Theory and Its Applications*, pp. 645–648, Yuzawa, Japan, Dec. 1999.

[20] T. Kaneko, T. Nishijima, H. Inazumi and S. Hirasawa, "An Efficient Maximum-Likelihood-Decoding Algorithm for Linear Block Codes with Algebraic Decoder," *IEEE Trans. Inform. Theory*, vol. 40, pp. 320–327, Mar. 1994.

[21] T. Kasami, T. Koumoto, T. Takata and S. Lin, "The Effectiveness of the Least Stringent Sufficient Condition on the Optimality of Decoded Codewords," *Proc. of the 3rd International. Symposium on Communication Theory and Applications*, pp. 324–333, Ambleside, UK, July 1995.

[22] Y. Tang, T. Kasami and T. Fujiwara, "An Optimality Testing Algorithm for a Decoded Codeword of Binary Block Codes and Its Computational Complexity," *Proc. of the 13th International Symposium AAECC-13*, Lecture Notes in Computer Science, vol. 1719, pp.201–210, Springer Verlag, Honolulu, Hawaii, U.S.A., Nov. 1999.

[23] Y. Tang, "Optimality and Ruling-out Conditions and their Evaluation Methods for Soft-decision Iterative Decoding Algorithms for Binary Lin-

ear Block Codes", Doctor thesis, *Nara Institute of Science and Technology*, NAIST-IS-DT9761027, Feb. 2000.

[24] D. Chase, "A Class of Algorithms for Decoding Block Codes with Channel Measurement Information," *IEEE Trans. Inform. Theory*, vol. IT-18, no. 1, pp. 170-182, Jan. 1972.

[25] N. Tendolkar and C. P. Hartmann, "Generalization of Chase Algorithms for Soft Decision Decoding of Binary Linear Codes," *IEEE Trans. Inform. Theory*, vol. IT-30, pp. 714–721, Sept. 1984.

[26] T. Kaneko, T. Nishijima and S. Hirasawa, "An Improvement of Soft-Decision Maximum-Likelihood Decoding Algorithm Using Hard-Decision Bounded-Distance Decoding," *IEEE Trans. Inform. Theory*, vol. 43, pp. 1314–1319, July 1997.

[27] T. Koumoto, T. Kasami and S. Lin, "A Sufficient Condition for Ruling Out Some Useless Test Error Patterns in Iterative Decoding Algorithms," *IEICE Trans. on Fundamentals*, vol. E81-A, No. 2, pp. 321–326, Feb. 1998.

[28] M. P. C. Fossorier and S. Lin, "Generalized Chase and GMD Decodings," *Proc. of the International Symposium on Information Theory and Its Applications*, pp. 415–418, Mexico City, Mexico, Oct. 1998.

[29] M. Kobayashi, T. Matsushima and S. Hirasawa, "On Reducing Complexity of Chase Decoding Method using Decoding Algorithms Beyond the BCH Bound," *Proc. of the Symposium on Information Theory and Its Applications*, pp. 857–860, Ehime, Japan, Dec. 1997.

[30] T. Kasami, Y. Tang, T. Koumoto and T. Fujiwara, "Sufficient Conditions for Ruling-Out Useless Iterative Steps in a Class of Iterative Decoding Algorithms," *IEICE Trans. Fundamentals*, vol. E82-A, no. 10, pp. 2061–2073, Oct. 1999.

[31] H. Tokushige, K. Nakamaye, T. Koumoto, Y. Tang and T. Kasami, "Selection of Search Centers in Iterative Soft-decision Decoding Algorithms," *Proc. of the Symposium on Information Theory and Its Applications*, Aso, Kumamoto, Japan, pp. 73–76, Oct. 2000.

[32] Y. Tang, T. Kasami and T. Fujiwara, "On the Computation of the Search Centers and the Evaluation of the Testing Conditions for the h-Chase Decoding," *Proc. of the Symposium on Information Theory and Its Applications*, Aso, Kumamoto, Japan, pp 77–80, Oct. 2000.