Computing Methods for the Weight Distributions of Linear Block Codes and the Weight Distributions of the Extended Binary Primitive BCH Codes of Lengths 64 and 128

by

Yoshihisa Desaki

April 1997

Dissertation submitted to Graduate School of Engineering Science of
Osaka University in partial fulfillment of the requirements for
the degree of Doctor of Philosophy in Engineering

Abstract

In high speed and high reliable digital data communication systems, error control coding is one of the most important technology. To design the systems, it is needed to estimate the error performance of codes precisely. Since it is easy to design the encoder and decoder of linear block codes, they are frequently used for the current communication systems. The weight distribution of a linear block code is a fundamental and important factor to determine its error performance. The formulas of the weight distribution are known only for limited classes of codes.

In this dissertation, we propose methods for computing the weight distributions of linear codes, and estimate the weight distributions of the extended binary primitive BCH codes of lengths 64 and 128. For the binary codes derived from shortened Reed-Solomon codes, the number of codewords with small weight is also investigated.

In Chapter 1, related topics on the weight distributions are briefly summarized to provide a background for this research.

In Chapter 2, we present a method for computing the weight distribution of a linear block code using the trellis structure of the code. This method is called trellis-based computing method. The computational complexity of the method is given in terms of the dimensions of subcodes and the related codes. Since the complexity can be evaluated easily, we can choose the trellis diagram by which the computational complexity becomes relatively small.

In Chapter 3, we also present a computing method using the invariant property of a code for some bit position permutations. In this method, the set of cosets of a subcode, which forms the original code, is partitioned into blocks. Two cosets are in the same block if and only if a permutation of coordinate places changes a coset into the other one. Since two cosets in the same block have the same weight distribution, it is enough to compute the number of cosets and the weight distributions of the representative coset in each block, which can be computed by the trellis-based computing method.

The computational complexity depends on the number of blocks. We investigate it for the case when we partition the set of cosets of the subcode of the code which is contained in a Reed-Muller code hierarchy and is invariant under the affine group.

In Chapter 4, by using the methods presented in Chapters 2 and 3, we compute the weight distributions of all the extended binary primitive BCH codes of lengths 64 and 128 for which the weight distributions have not been known so far. From the results, we also computed the probabilities of an undetectable error when the codes are used only for error detection in a binary symmetric channel, and determined whether or not the probability of each code monotonically increases as the bit error rate does.

It is still infeasible to compute the whole weight distribution for larger codes. Also, when shortened codes of various lengths are used, we need to know the weight distribution for large number of the codes. In Chapter 5, we investigate the binary code derived from a shortened Reed-Solomon code. For the binary code with a generator polynomial $(X-\alpha)$, a formula is shown for the exact number of codewords with weight 2, and an upper bound on the number of codewords with weight greater than 2 are derived. We also investigate the upper bound of the number of codewords with weight 4 and 6 for the binary code with the generator polynomial $(X-1)(X-\alpha)$. With these results, we computed the upper and lower bounds on the probability of an undetectable error for the case when $n = 2^{16}$, $m = 32$, the generator polynomial is $(X-\alpha)$, α is the root of the polynomial $g_0(X) = X^{32} + X^{26} + X^{23} + X^{22} + X^{16} + X^{12} + X^{11} + X^{10} + X^8 + X^7 + X^5 + X^4 + X^2 + X + 1$ and the polynomial basis is used. We also computed the upper and lower bounds on the probability of an undetectable error for the case when $n = 2^{12}$, the generator polynomial is $(X-1)(X-\alpha)$ and other conditions are the same as those in the above case. For the former case, those bounds are tight when the bit error rate of the channel is less than 10^{-7}, and for the latter case, when the bit error rate of the channel is less than 10^{-6}.

Acknowledgment

I am deeply indebted to many people for the advice, feedback and support they gave to me in the course of this work. I would especially like to thank Professor emeritus Tadao Kasami, currently Professor of Nara Institute of Science and Technology for his invaluable support, discussions and encouragement throughout the work.

I am grateful to Professor Nobuki Tokura for his invaluable suggestions and discussions on the work. I am also obliged to Professor Hideo Miyahara and Professor Toshinobu Kashiwabara for their helpful comments and suggestions.

I am extremely thankful to Associate Professor Toru Fujiwara for his invaluable discussions and great support throughout the work. I would like to thank Professor Shu Lin of University of Hawaii, Associate Professor Toyoo Takata of Nara Institute of Science and Technology, Dr. Robert H. Morelos-Zaragoza of Tokyo University and Dr. Hiroshi Yamamoto for their helpful comments and suggestions.

I would like to thank Professor Kazuhiko Iwasaki of Tokyo Metropolitan University, Professor Hiroyuki Seki of Nara Institute of Science and Technology, Dr. Masahiro Higuchi, Dr. Yukiya Miura of Tokyo Metropolitan University and Dr. Ryuichi Nakanishi of Nara Institute of Science and Technology for their kind and helpful support. I am thankful to Dr. Yasunori Ishihara of Nara Institute of Science and Technology, Dr. Yuich Kaji of Nara Institute of Science and Technology and Dr. Hajime Watanabe of Nara Institute of Science and Technology for their valuable support. I am also grateful to Ms. Yukiko Tanobe and Ms. Machiko Uehara for their kind support.

Lastly, I would like to thank all the students of information theory and logics laboratory of Osaka University and information and communication engineering laboratory of Tokyo Metropolitan University.

List of Publications

Journal Papers

(1) Yoshihisa Desaki, Toru Fujiwara and Tadao Kasami: "A Method for Computing the Weight Distribution of a Block Code by Using Its Trellis Diagram," *IEICE Trans. Fundamentals*, vol. E77-A, no. 8, pp. 1230–1237, Aug. 1994.

(2) Yoshihisa Desaki, Toru Fujiwara and Tadao Kasami: "The Weight Distributions of Extended Binary Primitive BCH Codes of Length 128," to appear in *IEEE Trans. Inf. Theory*, 1997.

International Conferences

(3) Yoshihisa Desaki, Toru Fujiwara, Tadao Kasami and Shu Lin: "Upper and Lower Bounds on the Undetected Error Probability of Binary Codes Derived from Shortened Reed-Solomon Codes," *Proc. of the International Symposium on Information Theory and Its Applications*, pp. 598–602, Singapore, Nov. 1992.

(4) Yoshihisa Desaki, Toru Fujiwara and Tadao Kasami: "The weight distributions of the (128, 64, 22) and (128, 71, 20) extended binary primitive BCH codes," *Proc. of the International Symposium on Information Theory and Its Applications*, pp. 594–597, Victoria, Sep. 1996.

Workshops

(5) Yoshihisa Desaki, Toru Fujiwara and Tadao Kasami: "A Method for Computing the Weight Distributions of Block Codes by Using Trellis Diagrams," *Proc. of the 16-th Symposium on Information Theory and Its Applications*, pp. 25–28, Kanazawa, Dec. 1993.

(6) Yoshihisa Desaki, Toru Fujiwara and Tadao Kasami: "A Method for Computing the Weight Distribution of a Linear Block Code by Using Weight Distributions

of Cosets with Respect to Its Subcode," *Proc. of the 17-th Symposium on Information Theory and Its Applications*, pp. 213–216, Hiroshima, Dec. 1994.

(7) Yoshihisa Desaki, Toru Fujiwara and Tadao Kasami: "The weight distributions of extended codes of binary primitive BCH codes of length 127 with designed distances 11, 13, 15, 23, 25, 29 and 31," *IEICE Technical Report*, IT95-25, Jul. 1995.

(8) Yoshihisa Desaki, Toru Fujiwara and Tadao Kasami: "The Weight Distributions of (128, 71) Extended Cyclic Codes which Contain the Extended Cyclic Third Order RM Code and Their Dual Codes," *Proc. of the 18-th Symposium on Information Theory and Its Applications*, pp. 651–654, Hanamaki, Dec. 1995.

(9) Yoshihisa Desaki, Toru Fujiwara and Tadao Kasami: "The weight distributions of the (128,64,22) and (128,71,20) extended binary primitive BCH codes," *IEICE Technical Report*, IT95-71, Mar. 1996.

Contents

Chapter 1

Introduction

In recent years, digital communication systems have made remarkable advances, and the demands for high speed and high reliability have been increased. Error correcting/detecting codes have been to meet the demands. Especially, linear block codes are frequently used for the current communication systems, since it is easy to design the encoder and decoder of them. The Hamming weight distribution (or simply the weight distribution) is important to estimate the performance of the used code. When a linear block code is used only for error detection in a q-ary symmetric channel, the probability of an undetectable error can be computed by the Hamming weight distribution of the code.

Let C be an (n, k) linear block code. Theoretically, the weight distribution of the code can be computed by generating all the codewords or generating all the codewords of its dual code and applying the MacWilliams' identity to it. If this is done by generating all linear combinations of row vectors from a generator matrix (or a parity-check matrix), the time complexity is $O(n2^k)$ (or $O(n2^{n-k})$). We call this computation method a brute-force method. The formulas of the weight distribution are known only for limited classes of codes, say Hamming codes, single, double and triple error-correcting binary primitive BCH codes. Then, expected is the research of the formula or the efficient computing method for the weight distributions which may use the knowledge of the structural property of a particular class of codes. Also, we may choose the other strategy, that is, to compute the some part of weight distribution and to estimate the upper and lower bounds from the results instead of computing the exact probability from the whole weight distribution. For example, it is known that the probability derived from only the numbers of codewords with small weight is a simple lower bound. Several computing methods for the number of codewords with minimum

weight for some classes of codes are known. D. Coppersmith and G. Seroussi proposed the efficient method to compute the minimum weight and the number of codewords with minimum weight for cyclic codes[1], and A.M. Barg and I.I. Dumer proposed the improved method[2]. Moreover, M. Mohri and M. Morii investigated the improved method in detail, and reduced the computational complexity furthermore[3].

In this dissertation, we propose efficient computing methods for the whole weight distribution of a general linear block code. By using the methods, we also compute the weight distributions of the extended binary primitive BCH codes of lengths 64 and 128.

In Chapter 2, we present a method for computing the number of codewords of weight less than or equal to the given integer in a linear block code by using its trellis diagram, which is called a trellis-based computing method. When the code length is a given integer, the weight distribution of the code can be computed. The time and space complexities are also analyzed. In this method, the computation is done by tracing the paths in a trellis diagram and counting the weight of the label on each branch. Then, the complexities of the trellis-based computing method depends on the structure of the used trellis diagram. For example, the sectionalization is one of such factors that have an influence on the complexities. Since the computational complexity can be represented in terms of the dimensions of subcodes and the related codes, we can choose the trellis diagram with a lower computational complexity by examining the dimensions. Some measures are proposed for the complexity and examine them for several section trellis diagrams of (128,36) extended binary primitive BCH code. In general, the computing time for this algorithm is not greater than that for a brute-force method. As another computing method for the weight distributions of linear block codes, T. Tanigawa et al. proposed a method which uses a code tree in [4]. A code tree is also a trellis diagram, and it is not minimal. Then, the capability of the trellis-based computing method is higher than that of the method by a code tree.

In general, the same label set of parallel branches between state pairs in a section may appears many times. Then, if the number of distinct weight distributions of the label sets is small enough to make a table for them, we can reduce the computational complexity for the trellis-based computing method by constructing the table.

When the error performance is precisely analyzed, we may need to know detailed weight distribution, say split weight distribution or joint weight distribution[5]. For example, joint weight distribution was used in the analysis of multi-stage decoding[6]. It is easy to modify the proposed algorithm to compute the detailed weight distribution.

In Chapter 3, we also propose a computation method which uses the invariant property of a code for permutation groups. The method is called a coset-based computing method. By the method, the set of cosets of a subcode, which forms the original code, is partitioned into blocks. Two cosets are in the same block if and only if a permutation of coordinate places changes a coset into the other one. Consequently, two cosets in the same block have the same weight distribution. Therefore, it is enough to compute the number of cosets and the weight distributions of the representative coset in each block. Then, the computation complexity is reduced.

In recent years, X. Hou has been examining the equivalent property of the cosets of a Reed-Muller code[7, 8]. He classified the set of cosets and derived a representative coset for each block. For several parameters, the weight distribution is also computed for a representative coset. Then, once it is known which representative cosets compose the given code, the weight distribution of the code can be easily computed. However, it is generally difficult to efficiently decide which representative coset is a component. In this chapter, a concrete study is given about it for some cases. It is known that a cyclic code is given by the direct sum of some irreducible cyclic codes. First, we consider the case when we partition the set of cosets of the cyclic subcode which is the largest cyclic subcode except for the original code by a cyclic group. Second, we investigate the equivalent property among the cosets of the subcode of the code which is contained in a Reed-Muller code hierarchy and is invariant under an affine group. A lemma is derived for the latter case, which is the generalization of the Lemma 5.1 in [8]. Note that the coset-based computing method can be simultaneously used with the trellis-based computing method presented in Chapter 2. That is, it is feasible that first, the coset-based computing method is used to reduce the computation of the weight distribution of a code into that of each representative coset of the subcode, and second, the trellis-based computing method is applied to the estimation of the weight distribution of each coset.

In Chapter 4, the extended binary primitive BCH codes of lengths 64 and 128 are taken as target codes. The weight distributions for the codes of length 128 have been known for the nontrivial codes with designed distances 4, 6 and 8[5, 9, 10], and 48, 56 and 64[10, 11]. In this chapter, by using the above methods, we compute the weight distributions of all the remaining extended binary primitive BCH codes of lengths 64 and 128 except for the codes of which the weight distributions are already known. If $K > N/2$, first the weight distribution of the dual code is computed and the MacWilliams' identity is applied to it, except for the case when $N = 128, K = 57$. The weight distributions of their dual codes are also computed. From the results, it turned out that (128, 64) extended binary primitive BCH code is formally self-dual, that is, the weight distributions of the code and its dual code are the same. From each weight distribution, we compute the probability of an undetectable error, denoted $P_{ue}(C, \varepsilon)$, when the code C is used only for error detection in a binary symmetric channel with bit-error rate $1/2^7 \leq \varepsilon \leq 1/2$. A code C is called *proper* if $P_{ue}(C, \varepsilon)$ is monotonously increasing for $0 \leq \varepsilon \leq 1/2$. For the choice of a code for some real applications, it is important whether a code is proper or not. Hence, it is examined whether each code is proper within the accuracy of our computation. Note that the above range of ε is sufficient to estimate it, since it is known that for any C of length n and minimum distance d, $P_{ue}(C, \varepsilon)$ is monotonously increasing for $0 \leq \varepsilon \leq d/n$.

It is still infeasible to compute the weight distributions for codes with large parameters. Also, when shortened codes of various lengths are used, we need to know the weight distributions for large number of the codes. This may be also infeasible. For such cases, another strategy is chosen, that is, to compute the some part of weight distribution and to estimate upper and lower bounds on the probability instead of computing the whole weight distribution and examining the exact probability. For example, it is difficult to compute the weight distribution of the binary codes derived from a shortened Reed-Solomon code. The binary codes often appear in a built-in self-test (BIST) in VLSI as it is explained below.

Signature analysis is a widely used data compaction method for built-in self-test in VLSI. Usually, a linear feedback shift register (LFSR) is used for a signature register which is a circuit to compress the output from a circuit under test (CUT). One problem

caused by data compaction is error aliasing. An aliasing error occurs when the output of CUT contains error and we fail to detect it. It is important to evaluate the aliasing probability of a signature register [12–17]. When we assume that any bit of the output of CUT is in error with a constant probability ε [12–17], the aliasing probability of the widely used multiple input signature registers is equal to the probability of an undetectable error $P_e(C, \varepsilon)$ of the binary code C derived from a shortened Reed-Solomon code when the code is used only for error detection in the binary symmetric channel with bit error rate ε.

In Chapter 5, we investigate the number of codewords with small weights in the binary codes derived from some shortened Reed-Solomon codes. By using the results, upper and lower bounds on the probability of an undetectable error are derived. In the following, let $\mathrm{RS}_n^{(d,a)}[g(X)]$ (or $\mathrm{RS}_n^{(d,a)}$) denote the shortened Reed-Solomon code over $GF(2^m)$ of length n and the minimum distance d whose generator polynomial is $(X - \alpha^a)(X - \alpha^{a+1}) \cdots (X - \alpha^{a+d-2})$ where α is a root of the primitive polynomial $g(X)$ of degree m. The binary code derived from $\mathrm{RS}_n^{(d,a)}$ by using a basis $\overline{\beta} = \{\alpha^{b_1}, \alpha^{b_2}, \cdots, \alpha^{b_m}\}$ is denoted by $\mathrm{RS}_n^{(d,a)}(\bar{\beta})$. We mainly consider the case where $\overline{\beta}$ is the polynomial basis $\overline{\beta}^{(P)} \triangleq \{1, \alpha, \alpha^2, \cdots, \alpha^{m-1}\}$, since LFSRs whose aliasing probability is equal to $P_e(\mathrm{RS}_n^{(d,a)}(\bar{\beta}^{(P)}), \varepsilon)$ are often used.

The probability of an undetectable error is computed from the binary weight distribution. When the Reed-Solomon code is not shortened, the formulas for the binary weight distributions are known for $\mathrm{RS}_{2^m-1}^{(d,1)}$ with $2 \leq d \leq 5$[18, 19], but when shortened, they are still unknown. It becomes infeasible to compute the whole weight distributions for many code lengths even for $m = 32$ and $d = 1$ which is an ordinary parameter for the LFSR used in a practical BIST. Then, the upper and lower bounds on the probability instead of examining the exact value are proposed. They are derived from the number of codewords with small weights, and the numbers are examined for some cases. For example, the exact number of codewords with weight 2 and an upper bound on the number of codewords with weight greater than 2 are derived for $\mathrm{RS}_n^{(2,1)}(\bar{\beta})$. These results are independent of the chosen primitive element, although in general, the binary weight distribution depends on the primitive element used to derive the binary image. We also derived upper bounds on the numbers of codewords

with weight 4 and 6 in $\mathrm{RS}_n^{(3,0)}(\bar{\beta})$. Using these results, upper and lower bounds on the probability of an undetectable error are computed for the case when $n = 2^{16}$, $m = 32$, the generator polynomial is $(X - \alpha)$, α is the root of the polynomial $g_0(X) = X^{32} + X^{26} + X^{23} + X^{22} + X^{16} + X^{12} + X^{11} + X^{10} + X^8 + X^7 + X^5 + X^4 + X^2 + X + 1$ and the polynomial basis is used. The upper and lower bounds on the probability of an undetectable error are also computed for the case when $n = 2^{12}$, the generator polynomial is $(X - 1)(X - \alpha)$ and other conditions are the same as those in the above case. For the former case, those bounds are tight when the bit error rate of the channel is less than 10^{-7}, and for the latter case, when the bit error rate of the channel is less than 10^{-6}.

Chapter 2

A Computing Method Using the Trellis Diagram of a Linear Block Code

In this chapter, we show a computing method for the number of codewords of weight less than or equal to a given integer in a linear block code by using its trellis diagram. When we choose the code length for the integer, we can compute the weight distribution of the given code. The time and space complexities are also analyzed. In general, the label set of parallel branches between state pairs in a section may appears many times for a trellis diagram of a linear block code. For the case, a further reduction of time complexity can be made.

2.1 Minimal Trellis Diagram

For simplicity, we only consider the trellis structure of binary linear block codes in this dissertation. The extension to nonbinary linear codes or to nonlinear codes is straightforward.

Consider a binary linear (n, k) block code C. By an n-section trellis diagram for C[20, 21], we mean a directed graph, denoted T, such that (1) T has an initial state s_0 and a final state s_f (a state is simply a node in the graph), (2) each branch (an edge in the graph) has a label and two branches diverging from the same state have different labels, and (3) there is a directed path from s_0 to s_f with label sequence $u_1 u_2 \cdots u_n$ if and only if $(u_1, u_2, \ldots, u_n)$ is a codeword in C. In the followings, a binary sequence of length ℓ is regarded as a binary ℓ-tuple, and vice versa. It is clear that the paths

connecting the initial state s_0 and the final state s_f of T represent all the codewords in C, and a branch represents a code bit. A trellis diagram for C with minimum number of states is said to be minimal, and a minimal trellis diagram is unique within graph isomorphism[20, 21].

Let T be an n-section minimal trellis diagram for a binary linear (n, k) block code C. For a nonnegative integer h not greater than n, let S_h denote the set of states of T just after the h-th bit position, where S_0 consists of the initial state s_0 only and S_n consists of the final state s_f only. For two states s and s', let $L(s, s')$ denote the set of all label sequences of paths from s to s', and $A_i(s, s')$ denote the number of label sequences of paths of weight i in $L(s, s')$. Also, let $W(s, s')$ denote the weight distribution of $L(s, s')$, $\{A_i(s, s') : 0 \leq i \leq n\}$. For convenience, we define that

$$A_i(s, s) = \begin{cases} 1, & \text{for } i = 0, \\ 0, & \text{otherwise.} \end{cases} \tag{2.1}$$

Then $L(s_0, s_f) = C$ and $W(s_0, s_f)$ is the weight distribution of C.

For a set of integers $U \triangleq \{h_0, h_1, h_2, \ldots, h_l\}$ with $0 = h_0 < h_1 < \cdots < h_l = n$, a section minimal trellis diagram for C, denoted $T(U)$, can be obtained from the n-section minimal trellis diagram T by deleting every state in S_h for $h \in \{0, 1, \ldots, n\} - U$ and every branch to or from a deleted state and by writing a branch with label λ from a state $s \in S_{h_j}$ to a state $s' \in S_{h_{j+1}}$ for $0 \leq j < l$, if and only if there is a path with label λ from s to s' in T[22]. $T(\{0, 1, \ldots, n\})$ is T itself. This section minimal trellis diagram $T(U)$ may have parallel branches between two adjacent states with different labels. Every branch from a state in S_{h_j} to a state in $S_{h_{j+1}}$ for $0 \leq j < l$ represents $(h_{j+1} - h_j)$ code bits.

2.2 Two Algorithms for Computing the Weight Distribution

For a section trellis diagram $T(U)$ of C, let $T'(U)$ be a graph which is obtained from $T(U)$ by replacing the label of each branch by the Hamming weight of the label. In this section, we show a method for computing the weight distribution of C when $T'(U)$ is given for $U \triangleq \{h_0, h_1, h_2, \ldots, h_l\}$ with $0 = h_0 < h_1 < \cdots < h_l = n$. $T'(U)$ can be constructed from a generator matrix of C[20, 21].

For a branch b in $T'(U)$, let $s_s(b)$ be the starting state of b. The label of b in $T'(U)$ is denoted by $w_H(b)$. For a state s, let $IB(s)$ denote the set of incident branches to s in $T'(U)$. For simplicity, $A_i(s_0, s)$ is denoted by $A_i(s)$. Then, for a state $s \in S_{h_j}$ with $0 < j \leq l$,

$$A_i(s) = \sum_{b \in IB(s), w_H(b) \leq i} A_{i-w_H(b)}(s_s(b)), \quad \text{for} \quad 0 \leq i \leq n. \tag{2.2}$$

By using (2.1) and (2.2), the weight distribution of C can be computed. For a given integer w, an algorithm, denoted Algorithm I, to compute $A_i(s_f)$ with $0 \leq i \leq w$ is as follows:

[**Algorithm I** to compute $A_i(s_f)$ with $0 \leq i \leq w$ for a given integer w]

(L1) $A_0(s_0) := 1$;
(L2) **for** $j := 1$ **to** l
(L3) **for** each state s in S_{h_j} **do begin**
(L4) **for** $i = 0$ **to** $\min\{h_j, w\}$
(L5) $A_i(s) := 0$;
(L6) **for** each branch $b \in IB(s)$
(L7) **for** $i = w_H(b)$ **to** $\min\{h_{j-1} + w_H(b), w\}$
(L8) $A_i(s) := A_i(s) + A_{i-w_H(b)}(s_s(b))$;
(L9) **end;**
(L10) **end.**

This algorithm can be used to compute the weight distributions of the shortened codes of various code lengths simultaneously. For two integers h and h' such that $0 \leq h < h' \leq n$, let $C_{h,h'}$ be the linear subcode of C consisting of all codewords whose components are all zero except for the $h' - h$ components from the $(h+1)$-th bit position to the h'-th bit position. Let $C^{tr}_{h,h'}$ be defined as

$$C^{tr}_{h,h'} \triangleq \{(v_{h+1}, v_{h+2}, \ldots, v_{h'}) : (\underbrace{0, \ldots, 0}_{h}, v_{h+1}, v_{h+2}, \ldots, v_{h'}, \underbrace{0, \ldots, 0}_{n-h'}) \in C_{h,h'}\}. \tag{2.3}$$

The code $C^{tr}_{0,h}$ is a shortened code of length h, denoted C_h. For each S_h with $0 \leq h \leq n$, let s^0_h be the state in S_h for which

$$00 \cdots 0 \in L(s^0_h, s_f). \tag{2.4}$$

The state s_h^0 is unique for each h. Then, the weight distribution of C_h is given by $\{A_i(s_0, s_h^0) : 0 \leq i \leq h\}$. Therefore, we can compute the weight distributions of the codes, C_h with $h \in U$ simultaneously by the above algorithm.

Next, we show a modified algorithm for computing the weight distribution. For a state s in $T'(U)$, define $A'_i(s) \triangleq A_i(s, s_f)$. Then, the numbers of label sequences in $L(s_0, s_f)$ with weight i which pass through the state s is given by

$$\sum_{j=0}^{i} A_j(s) A'_{i-j}(s). \tag{2.5}$$

Therefore, we can compute the weight distribution by computing $A_i(s)$ with $0 \leq i \leq h_p$ and $A'_i(s)$ with $0 \leq i \leq n - h_p$ for all the state s in S_{h_p} for an integer p with $0 < p < l$.

For a branch b in $T'(U)$, let $s_e(b)$ be the end state of b. For a state s, let $IB'(s)$ denote the set of branches diverging from s in $T'(U)$. Then, for a state $s \in S_{h_j}$ with $0 \leq j < l$,

$$A'_i(s) = \sum_{b \in IB'(s), w_H(b) \leq i} A'_{i - w_H(b)}(s_e(b)), \quad \text{for} \quad 0 \leq i \leq n. \tag{2.6}$$

By using the above equation, we can also compute $A'_i(s)$ for any i and s in a similar way as for $A_i(s)$.

The modified algorithm by using (2.1), (2.5) and (2.6), denoted Algorithm II, consists of the following three parts:

[**Algorithm II** (Outline)]

(1) Compute $A_i(s)$ with $0 \leq i \leq h_p$ for each state s in S_{h_p} for a given integer p by the same way as in Algorithm I.

(2) Compute $A'_i(s)$ with $0 \leq i \leq n - h_p$ for each state s in S_{h_p} in a similar way.

(3) Compute $A_i(s_f)$ with $0 \leq i \leq n$ by using

$$A_i(s_f) = \sum_{s \in S_{h_p}} \sum_{j=0}^{i} A_j(s) A'_{i-j}(s). \tag{2.7}$$

Algorithm II is more efficient than Algorithm I in several cases, say $A_i(s_f) = 0$ for many i, as shown in the following section.

When we only compute the number of codewords with minimum weight, we can reduce the computational complexity of the above two algorithms furthermore. That is, we only need to compute $A_0(s)$ and $A_{i_0}(s)$ (or $A'_0(s)$ and $A'_{i'_0}(s)$) for the minimum

positive integer i_0 (or i'_0) such that $A_{i_0}(s) \neq 0$ (or $A'_{i'_0} \neq 0$). In step (L6), it is enough to consider only the minimum weight branch among the parallel branches.

This method is called a trellis-based computing method. The simple straightforward method examining all the codewords can be viewed as the trellis-based computing method using the one-section trellis diagram $T(\{0, n\})$.

2.3 Complexity analysis of the Two Algorithms

In this section, we analyze the complexities of the above two algorithms. Before that we briefly review the results in [22] on the structural complexity of a section minimal trellis diagram for a binary linear code. The numbers of branches and states in $T(U)$ are expressed in terms of the dimensions of specific subcodes of C.

For integers h and h' with $0 \leq h < h' \leq n$, let $K_{h,h'}$ be the dimension of $C_{h,h'}$, i.e.,

$$K_{h,h'} = \log_2 |C_{h,h'}|, \tag{2.8}$$

where for a set X, $|X|$ denotes the number of elements in X. For convenience, $K_{h,h}$ is defined as zero. For integers h, h' and h'' such that $0 \leq h < h' < h'' \leq n$, let $K_{h,h',h''}$ be defined as

$$K_{h,h',h''} \triangleq K_{h,h''} - K_{h,h'} - K_{h',h''}. \tag{2.9}$$

For simplicity, we write K_h for $K_{0,h,n}$. Then, it is shown in [21, Appendix A] that $|S_h| = 2^{K_h}$. Let h and h' be two integers such that $0 \leq h < h' \leq n$ again. For a binary n-tuple $\mathbf{v} = (v_1, v_2, \ldots, v_n)$, let $p_{h,h'}\mathbf{v}$ denote the binary $(h'-h)$-tuple $(v_{h+1}, v_{h+2}, \ldots, v_{h'})$ and let $p_{h,h'}[C]$ be defined as $p_{h,h'}[C] \triangleq \{p_{h,h'}\mathbf{v} : \mathbf{v} \in C\}$. For $\mathbf{u} \in p_{0,n/2}[C]$, let $\sigma\mathbf{u}$ denote the last state of the path labeled $\mathbf{u}$ from the initial state s_0 in T. The following theorem[22] holds on the structure of a minimal trellis diagram.

Theorem 1: For $1 \leq h \leq n$, let S_h be the set of states of the n-section minimal trellis diagram just after the h-th bit position for a binary linear (n, k) code C. Let $q_{h,h'}$ be defined as

$$q_{h,h'} \triangleq K_h - K_{0,h,h'} = K_{h'} - K_{h,h',n}. \tag{2.10}$$

Then, S_h and $S_{h'}$ can be partitioned into $2^{q_{h,h'}}$ blocks of the same size S_{h1}, $S_{h2}, \ldots, S_{h2^{q_{h,h'}}}$ and $S_{h'1}$, $S_{h'2}$, ..., $S_{h'2^{q_{h,h'}}}$, in such a way that (1) $S_{h'i} = \{\sigma\mathbf{v} : \mathbf{v} \in V_i\}$,

where $V_1, V_2, \ldots, V_{2^{q_{h,h'}}}$ are the cosets of $p_{0,h'}[C]$ with respect to $p_{0,h'}[C_{0,h'} + C_{h,n}]$, (2) there is a path from $s \in S_h$ to $s' \in S_{h'}$, if and only if $s \in S_{hi}$ and $s' \in S_{h'i}$ for the same i, and (3) for $s \in S_{hi}$ and $s' \in S_{h'i}$ with $1 \leq i \leq 2^{q_{h,h'}}$, the number of paths from s to s' is $2^{K_{h,h'}}$. $\Delta\Delta$

By using this theorem, the number of branches in the j-th section, which is the section from $(h_{j-1}+1)$-th to h_j-th bit positions, is given by

$$\left(|S_{h_{j-1}}|/2^{q_{h_{j-1},h_j}}\right) \cdot |S_{h_j}| \cdot 2^{K_{h_{j-1},h_j}} = 2^{K - K_{0,h_{j-1}} - K_{h_j,n}}. \tag{2.11}$$

Define $B_{h_{j-1},h_j} \triangleq K - K_{0,h_{j-1}} - K_{h_j,n}$. Then the total number BR of branches in the entire trellis diagram $T(U)$ is given by

$$BR \triangleq \sum_{j=1}^{l} 2^{B_{h_{j-1},h_j}}. \tag{2.12}$$

Now we consider the time complexity of Algorithm I. This algorithm consists of two parts: part (1): the construction of $T'(U)$ and part (2): the computation of $A_i(s_f)$. We can use the total number BR of branches in the trellis diagram given by (2.12) as the complexity measure for the part (1). In the part (2), for every state s in the trellis diagram, we compute $A(s) \triangleq \{A_i(s) : 0 \leq i \leq \min\{w, h_j\}\}$ where $s \in S_{h_j}$ from $A(s_s(b))$ for $b \in IB(s)$ by using (2.1) and (2.2). The number of additions to obtain $A(s)$ for a state $s \in S_{h_j}$ is upper bounded by

$$(\min\{w, h_{j-1}\} + 1)\,|IB(s)| = (\min\{w, h_{j-1}\} + 1)\,2^{K_{0,h_j} - K_{0,h_{j-1}}}. \tag{2.13}$$

Consider the case where $s \in S_{h_j}$ and $|L(s_0, s_s(b))| = 2^{K_{0,h_{j-1}}}$ is less than $\min\{w, h_{j-1}\} + 1$ for $b \in IB(s)$. This often occurs when j is small. The number of additions to obtain $A(s)$ can be reduced in the following way: Let $w_1^{(b)}, w_2^{(b)}, \cdots, w_{|L(s_0,s_s(b))|}^{(b)}$ be the weights of label sequences in $L(s_0, s_s(b))$. Then, $A(s)$ can be obtained by computing $w_H(b) + w_i^{(b)}$ for $b \in IB(s)$ and $1 \leq i \leq |L(s_0, s_s(b))|$. The number of additions to obtain $A(s)$ is

$$|L(s_0, s_s(b))| \times |IB(s)| = 2^{K_{0,h_j}}. \tag{2.14}$$

Therefore, we can use

$$\begin{aligned} ADD_1 &\triangleq \sum_{1 \leq j \leq l} \sum_{s \in S_{h_j}} \sum_{b \in IB(s)} \min\{w+1, h_{j-1}+1, |L(s_0, s_s(b))|\} \\ &= \sum_{1 \leq j \leq l} 2^{B_{h_{j-1},h_j}} \min\left\{w+1, h_{j-1}+1, 2^{K_{0,h_{j-1}}}\right\}. \end{aligned} \tag{2.15}$$

as the complexity measure for the part (2).

The time complexity of Algorithm I is given by $c_1 BR + c_2 ADD_1$, where c_1 and c_2 are the cost of an operation to compute the weight of a branch and that of an addition, respectively. The order of the complexity is wBR.

Next, we consider the time complexity of Algorithm II. This algorithm consists of four parts: part (0): the construction of $T'(U)$ and three parts mentioned before. As the complexity measure for the part (0), we can use the same measure BR as for Algorithm I. For the parts (1) and (2), the measure is given by

$$\begin{aligned} ADD_2 &\triangleq \sum_{1\le j\le p}\sum_{s\in S_{h_j}}\sum_{b\in IB(s)} \min\{w+1, h_{j-1}+1, |L(s_0, s_s(b))|\} \\ &\quad + \sum_{p<j\le l}\sum_{s\in S_{h_{j-1}}}\sum_{b\in IB'(s)} \min\{w+1, n-h_j+1, |L(s_e(b), s_f)|\} \\ &= \sum_{1\le j\le p} 2^{B_{h_{j-1},h_j}} \min\left\{w+1, h_{j-1}+1, 2^{K_{0,h_{j-1}}}\right\} \\ &\quad + \sum_{p<j\le l} 2^{B_{h_{j-1},h_j}} \min\left\{w+1, n-h_j+1, 2^{K_{h_j,n}}\right\}. \end{aligned} \tag{2.16}$$

The measure for the part (3) is the number of multiplications to compute the weight distribution by using $A(s)$ and $A'(s)$ for all the states $s \in S_{h_p}$, which is upper bounded by

$$MLT_2 \triangleq 2^{K_{h_p}} \times \min\{w+1, h_p+1, 2^{K_{0,h_p}}\} \times \min\{w+1, n-h_p+1, 2^{K_{h_p,n}}\}. \tag{2.17}$$

Note that we need the same number of additions as that of multiplications to obtain $A(s_f)$ from the values of $A(s)$ and $A'(s)$ with $s \in S_{h_p}$. If $A_i(s_f) = 0$ for many i, we can reduce the number of multiplications. It is shown by MacEliece[5, 23] that the weight of every codeword of a binary cyclic code is divisible by 2^{h-1}, where h is the smallest number such that h nonzeros of the code have product 1. For example, all weights in $\mathrm{RM}_{m,r}$, that is, r-th order Reed-Muller code of length 2^m are multiple of $2^{\lceil m/r\rceil - 1}$[5]. In this case, $A_{i+i'}(s) = 0$ (or $A'_{i+i'}(s) = 0$) for $1 \le i' < 2^{h-1}$, if $A_i(s) \neq 0$ (or $A'_i(s) \neq 0$). Therefore, $A_i(s) = 0$ (or $A'_i(s) = 0$) for many i at every state s. By finding the sets of integers i such that $A_i(s) \neq 0$ and $A'_i(s) \neq 0$ before multiplications, we can reduce the number of multiplications.

The time complexity of Algorithm II is given by $c_1 BR + c_2 ADD_2 + (c_2 + c_3) MLT_2$, where c_3 is the cost of a multiplication. The order of the complexity is $wBR + w^2|S_{h_p}|$.

We compare the time complexities of these two algorithms. It is easy to see that $ADD_1 > ADD_2$. For simplicity, we consider Algorithm II with $p = n/2$, and the case where $U = \{0, n/4, n/2, 3n/4, n\}$ and $w = n$. Then, $MLT_2 \approx 2^{K_{n/2}}(n/2+1)^2$. Since the term with $j = 3$ is dominant in $ADD_1 - ADD_2$ in most cases, $ADD_1 - ADD_2 \approx 2^{B_{n/2,3n/4}}\{(n/2+1) - (n/4+1)\}$. Therefore, $ADD_1 \approx ADD_2 + 2MLT_2$ if $n = 2^{K_{n/2,n} - K_{3n/4,n} - 1}$. When $B_{n/2,3n/4}$ is relatively large, Algorithm II is more efficient than Algorithm I. The number of multiplications MLT_2 becomes much smaller than ADD_2 when $A_i(s_f) = 0$ for many i. Therefore, in such a case, Algorithm II is more efficient than Algorithm I, even if the difference between ADD_1 and ADD_2 is not large.

The time complexities of these algorithms depend on the choice of U. In general, it is hard to determine U that minimizes the time complexity. Since it is easy to evaluate the time complexity for a given U, we may find a good U by computer search. For an example code, the complexities are compared with respect to various U in the following section. A good U for one algorithm is also good for the other algorithm in most cases. But it is not known whether optimum U for one algorithm is optimum for the other or not.

Next, we consider the space complexity of Algorithm I. It is easy to see that to compute $A(s)$ for a state $s \in S_{h_j}$, we only need $A(s')$ for states s' in $S_{h_{j-1}}$, and the information on the j-th section of the trellis diagram. Therefore, we can compute the weight distribution by constructing the trellis diagram section by section, and the amount of space needed for computation is dominated by that to store the values of $A(s)$, which is $O(\max_{0<j\le l}\{h_{j-1}|S_{h_{j-1}}| + h_j|S_{h_j}|\})$. It is easy to see that $O(\max_{0<j\le l}\{h_{j-1}|S_{h_{j-1}}| + h_j|S_{h_j}|\} + n|S_{h_p}|)$, where $0 < p < l$, is the space complexity of Algorithm II.

The trellis diagrams for some linear block codes are loosely connected and have parallel structure in the sense that the trellis diagram consists of parallel sub-trellis diagrams without cross connection between them. From Theorem 1, the entire section minimal trellis diagram for C consists of the first tree type section, $2^{q_{h_1,h_{l-1}}}$ structurally identical (except branch labels) sub-trellis diagrams without cross connection between them, and the last tree type section. For example, the 4-section trellis diagram $T(\{0, n/4, n/2, 3n/4, n\})$ for $\mathrm{RM}_{6,2}$ consists of the first tree type section, 64 structurally

identical 16-state sub-trellis diagrams without cross connection between them, and the last tree type section. From Theorem 1, each parallel sub-trellis diagram combined with the first and the last tree type section is the minimal trellis diagram for a coset of C with respect to $C_{0,h_{l-1}} + C_{h_1,n}$. We can reduce the space complexity by computing the weight distribution of each coset independently. Then the space complexity of Algorithm I (or Algorithm II) is reduced to $O(\max_{0<j\leq l}\{h_{j-1}|S_{h_{j-1}}| + h_j|S_{h_j}|\}/2^{q_{h_1,h_{l-1}}})$ (or $O(\{\max_{0<j\leq l}\{h_{j-1}|S_{h_{j-1}}| + h_j|S_{h_j}|\} + n|S_{h_p}|\}/2^{q_{h_1,h_{l-1}}})$, where $0 < p < l$).

In [24], the state complexities of minimal trellises have been analyzed for Reed-Muller codes and the extended and permuted $(64, 24)$, $(64, 45)$, and double-error correcting $(2^m, 2^m - 2m - 1)$ BCH codes. The state complexities of some other permuted BCH codes have also been investigated[25, 26]. Since the following upper bound on the number of branches holds for the n-section minimal trellis diagram T,

$$BR \leq \sum_{h=1}^{n} 2|S_h|, \tag{2.18}$$

these results show that for these codes, BR is much smaller than $2n \times \min\{2^k, 2^{n-k}\}$. That is, to compute the weight distributions of these codes, the algorithms proposed here are efficient.

2.4 Example

Consider the extended (128, 36) code of the binary primitive (127, 36) BCH code with minimum weight 32 as C. The numbers of branches and states of a minimal trellis diagram depend on the choice of the order of the bit positions. To get a relatively simple minimal trellis diagram, we consider an equivalent code of C which contains the second order Reed-Muller code with the standard binary order. The specific bit permutation is given in [24].

A trellis diagram T is said to be reversible if the graph T^R obtained from T by reversing the direction of each branch without changing the label and exchanging the initial state and the final state is identical to T. For an integer l by which n is divisible, define

$$U_l \triangleq \{nj/l : 0 \leq j \leq l\}. \tag{2.19}$$

The trellis diagram $T(U_l)$ is the l-section minimal trellis diagram with the section length $h_j - h_{j-1} + 1 = n/l$ for $1 \leq j \leq l$, and is reversible (see Section 2.5). Since C is an extended code, (i) all weights are even and (ii) $A_i(s_f) = A_{n-i}(s_f)$. It follows from (i), (ii) and (iii) the designed distance is 32 that the number of multiplications in Algorithm II is upper bounded by $MLT_2' \triangleq MLT_2/13$. In Table 2.1, the measures of time complexities for the trellis diagrams $T(U_l)$ are listed for $l = 2^i$ with $1 \leq i \leq 7$, where ADD_2 and MLT_2' are evaluated for $p = n/2$. Note that for this example code, the actual number of multiplications is much smaller than MLT_2', since all the weights are multiple of 4.

Although the ratios of c_1 to c_3 and c_2 to c_3 depend on the computer used, the ratio c_2 to c_3 is almost equal to 1 for most cases. The ratio of c_1 to c_3 also depends on the implementation. We can store the label of a branch of length $h_j - h_{j-1} + 1$ in the j-th section in $\lceil (h_j - h_{j-1} + 1)/L \rceil$ words in the computer, when L bits binary sequence is stored in a word. By finding the Hamming weight of the label in a word with the table look up, we can obtain the weight of a branch by $\lceil (h_j - h_{j-1} + 1)/L \rceil - 1$ additions. For example, L may be 16.

From Table 2.1, we see that Algorithm II with $l = 4$ is appropriate to compute the weight distribution. The structural complexity of the reversible trellis diagram $T(U_4)$ is as follows: (1) The number of states at the end of the j-th section with $1 \leq j \leq 2$ is 2^{22}. (2) The number of parallel components in the second section is 2^{17} and therefore one parallel component is a 2^5-state sub-trellis diagram. (3) The number of parallel branches between any connected states is 2. (4) The total number of branches in the first section is 2^{23} and that in the second section is 2^{28}.

Since each parallel component has only 32 states, the space complexity is enough small to compute on a workstation. Given a generator matrix of the code, it takes about 79 minutes (CPU time) to obtain the weight distribution by using a UNIX workstation, DEC Station 3100.

We have also applied the proposed algorithm to compute a detailed weight distribution of the inner code for the error performance analysis of a concatenated code. Each 8-bit byte in a codeword of the inner code is also a symbol of the outer code. Therefore, for each integer i and each binary 8-tuple $\mathbf{v}$, we need to know the number

Table 2.1 The measures of the time complexity of the two algorithms for the l-section minimal trellis diagrams, $T(U_l)$ for the extended and permuted (128, 36) code of the binary primitive (127, 36) BCH code.

l	BR	ADD_1	ADD_2
2	1.073×10^9	3.543×10^{10}	1.073×10^9
4	5.536×10^8	1.880×10^{10}	1.090×10^9
8	5.537×10^8	2.122×10^{10}	1.627×10^9
16	8.305×10^8	3.375×10^{10}	2.978×10^9
32	1.519×10^9	6.338×10^{10}	5.814×10^9
64	2.966×10^9	1.253×10^{11}	1.155×10^{10}
128	2.966×10^9	1.253×10^{11}	1.155×10^{10}

$MLT_2' = 1.363 \times 10^9$ for each 2^i-section minimal trellis diagram where $1 \leq i \leq 7$.

of codewords whose weight is i and a specific 8-bit byte is $\mathbf{v}$. We have computed the detailed weight distribution of a $(64, 40)$ subcode of the third order Reed-Muller code of length 64 on the workstation.

2.5 Symmetric Property of Minimal Trellis Diagram

For an i-tuple $\mathbf{u} = (u_1, u_2, \cdots, u_i)$, let $\mathbf{u}^R$ denote $(u_i, u_{i-1}, \cdots, u_1)$, and for a code C, let C^R denote $\{\mathbf{u}^R : \mathbf{u} \in C\}$. It is known that $C = C^R$ for Reed-Muller codes, extended and permuted primitive BCH codes and their dual codes. The minimal trellis diagram T is reversible for a code C such that $C = C^R$.

In this section, we show that computational complexity to obtain the weight distribution of a linear code C with an even code length n can be reduced by modifying Algorithm II when $C = C^R$. The result (Theorem 2 below) may also be used to simplify the maximum likelihood decoder of the code.

From the definition, $C^{tr}_{0,n/2} = p_{0,n/2}[C_{0,n/2}]$, and $C^{tr}_{0,n/2}$ is a linear subcode of $p_{0,n/2}[C]$. The concatenation of two $n/2$-tuples $\mathbf{u} = (u_1, u_2, \ldots, u_{n/2})$ and $\mathbf{v} =$

$(v_1, v_2, \ldots, v_{n/2})$ is defined as the following n-tuple:

$$\mathbf{u} \circ \mathbf{v} \triangleq (u_1, u_2, \ldots, u_{n/2}, v_1, v_2, \ldots, v_{n/2}). \tag{2.20}$$

For $\mathbf{u}_1 \in p_{0,n/2}[C]$, let $R_H(\mathbf{u}_1)$ be defined as

$$R_H(\mathbf{u}_1) \triangleq \{\mathbf{u}_2^R : \mathbf{u}_1 \circ \mathbf{u}_2 \in C\}. \tag{2.21}$$

Theorem 2: Suppose that

$$C = C^R. \tag{2.22}$$

Let C^{SR} denote the following subcode of C:

$$C^{\mathrm{SR}} \triangleq \{\mathbf{u} \in C : \mathbf{u}^R = \mathbf{u}\}. \tag{2.23}$$

For any $\mathbf{u} \in p_{0,n/2}[C]$ and $\mathbf{v} \in R_H(\mathbf{u})$, (1) $\sigma\mathbf{v}$ is unique, and (2) if $\mathbf{u} \in p_{0,n/2}[C^{\mathrm{SR}}]$, $\sigma\mathbf{v} = \sigma\mathbf{u}$ and otherwise, $\sigma\mathbf{u}$ and $\sigma\mathbf{v}$ are in $\{\sigma\mathbf{w} : \mathbf{w} \in V\}$ for a coset V of $p_{0,n/2}[C]$ with respect to $p_{0,n/2}[C^{\mathrm{SR}}]$.

(Proof)

See Appendix for its proof. $\Delta\Delta$

For a state $s \in S_{n/2}$, let $\mathbf{u}_s$ be an $n/2$-tuple in $p_{0,n/2}[C]$ such that $\sigma\mathbf{u}_s = s$, and let $R(s)$ denote the state $\sigma\mathbf{v}$, where $\mathbf{v} \in R_H(\mathbf{u}_s)$. From (1) of Theorem 2, we see that we can reduce the computing time by the following way:

(1) Compute $A_i(s)$ with $0 \leq i \leq n/2$ for each $s \in S_{n/2}$.

(2) Compute $A_i(s_f)$ with $0 \leq i \leq n$ by

$$A_i(s_f) = \sum_{s \in S_{n/2}} \sum_{j=0}^{i} A_j(s) A_{i-j}(R(s)). \tag{2.24}$$

The technique to reduce the space complexity described in Section 2.2 can be also applied. The left half of the minimal trellis diagram $T(\{0, h, \ldots, n/2, \ldots\})$, consists of the first tree type section and $2^{q_{h,n/2}}$ structurally identical (except branch labels) sub-trellis diagrams without cross connection between them[22]. From Theorem 1, the set of all label sequences of paths in each parallel sub-trellis diagram combined with the first tree type section is a coset of $p_{0,n/2}[C]$ with respect to $p_{0,n/2}[C_{0,n/2} + C_{h,n}]$. It

follows from this fact and (2) of Theorem 2, the computation of (2.24) can be performed independently for each subset of $S_{n/2}$,

$$\{\sigma\mathbf{v}: \ \mathbf{v} \text{ is in a coset of } p_{0,n/2}[C] \text{with respect to } p_{0,n/2}[C_{0,n/2} + C_{h,n} + C^{\mathrm{SR}}]\}. \quad (2.25)$$

The number of states in the subset is given by

$$2^{K_{h,n}-2K_{0,n/2}} \left|C^{SR}\right| / \left|C^{\mathrm{SR}}_{h,n-h}\right|. \quad (2.26)$$

The derivation of (2.26) is shown in Appendix. For these purposes, we need to obtain a basis of C^{SR}. The following Lemma 1 gives us an efficient algorithm for finding it.

Lemma 1: Let $\{\mathbf{u}_1, \mathbf{u}_2, \cdots, \mathbf{u}_k\}$ be a basis of (n,k) linear code C, where for a non-negative integer k_0^{SR}, $\mathbf{u}_i$ with $1 \leq i \leq k_0^{\mathrm{SR}}$ are in C^{SR} and $\mathbf{u}_i$ with $k_0^{\mathrm{SR}} < i \leq k$ are not in C^{SR}. Let C_0^{SR} be the linear (n, k_0^{SR}) subcode of C^{SR} spanned by $\{\mathbf{u}_1, \mathbf{u}_2, \cdots, \mathbf{u}_{k_0^{\mathrm{SR}}}\}$. Note that $C_0^{\mathrm{SR}} = \{\mathbf{0}\}$ if $k_0^{\mathrm{SR}} = 0$. Then the followings hold:
(1) If there is $\mathbf{u}_i$ with $k_0^{\mathrm{SR}} < i \leq k$ such that $\mathbf{u}_i$ and $\mathbf{u}_i^R$ belong to different cosets of C with respect to C_0^{SR}, then

$$\mathbf{u}_i + \mathbf{u}_i^R \in C^{\mathrm{SR}} - C_0^{\mathrm{SR}}. \quad (2.27)$$

(2) Suppose that $\mathbf{u}_i + \mathbf{u}_i^R \in C_0^{\mathrm{SR}}$ for every i with $k_0^{\mathrm{SR}} < i \leq k$. If $\mathbf{u}_i + \mathbf{u}_i^R$ with $k_0^{\mathrm{SR}} < i \leq k$ are linearly independent, then

$$C_0^{\mathrm{SR}} = C^{\mathrm{SR}}, \quad (2.28)$$

and otherwise, for any binary nonzero tuple $(a_{k_0^{\mathrm{SR}}+1}, a_{k_0^{\mathrm{SR}}+2}, \cdots, a_k)$ such that

$$\sum_{i=k_0^{\mathrm{SR}}+1}^{k} a_i(\mathbf{u}_i + \mathbf{u}_i^R) = \mathbf{0}, \quad (2.29)$$

we have that

$$\sum_{i=k_0^{\mathrm{SR}}+1}^{k} a_i\mathbf{u}_i \in C^{\mathrm{SR}} - C_0^{\mathrm{SR}}. \quad (2.30)$$

(Proof)

See Appendix for its proof. $\Delta\Delta$

2.6 An Improved Computing Method by Using a Trellis Diagram

In this section, we show an idea to reduce the computational complexity for the trellis-based computing method.

Define $h \triangleq h_{j-1}$ and $h' \triangleq h_j$ for $1 \leq j \leq l$. It is known that a nonempty set $L(s, s')$ of parallel branch labels between two states, $s \in S_h$ and $s' \in S_{h'}$, is a coset in $p_{h,h'}[C]/C^{\mathrm{tr}}_{h,h'}$[22, 27]. Any coset in $p_{h,h'}[C]/C^{\mathrm{tr}}_{h,h'}$ may appears many times as the label set of parallel branches between state pairs in the section. For example, consider the second section of the 4-section minimal trellis diagram $T(\{0, 32, 64, 96, 128\})$ for $\mathrm{RM}_{7,3}$. The second section is the subgraph of the trellis diagram obtained by truncating $T(\{0, 32, 64, 96, 128\})$ except from time 32 to time 64. Now, the following equation holds.

$$C^{\mathrm{tr}}_{0,n/2} = C^{\mathrm{tr}}_{n/2,n} = \mathrm{RM}_{m-1,r-1}. \tag{2.31}$$

Then, we see that $C^{\mathrm{tr}}_{32,64}$ is the (32, 6) $\mathrm{RM}_{5,1}$ code. It is also easy to show that $p_{32,64}[C]$ is the $(32, 26)$ $\mathrm{RM}_{5,3}$ code. Hence, $L(s, s')$ with $s \in S_{32}, s' \in S_{64}$ is a coset of the $(32, 6)$ code and has 64 labels. There are 2^{20} cosets in $p_{32,64}[C]/C^{\mathrm{tr}}_{32,64} = \mathrm{RM}_{5,3}/\mathrm{RM}_{5,1}$. Using the analysis in [27], we can show that every coset in $\mathrm{RM}_{5,3}/\mathrm{RM}_{5,1}$ appears 2^6 times as the label set of parallel branches.

When each coset in $p_{h,h'}[C]/C^{\mathrm{tr}}_{h,h'}$ appears many times as the label set, we can reduce the computing time by computing $W(s, s')$ with $s \in S_h$ and $s' \in S_{h'}$ in the following way: (1) Construct a table to find the weight distribution of any coset in $p_{h,h'}[C]/C^{\mathrm{tr}}_{h,h'}$ from the syndrome of the coset leader. To compute the weight distribution of a coset of $C^{\mathrm{tr}}_{h,h'}$, we can use trellis-based computing method recursively. (2) For each $L(s, s')$, find its weight distribution, $W(s, s')$, from the table.

Chapter 3

A Computing Method Using an Invariant Property of a Linear Block Code for Permutation Groups

In this chapter, we present a computation method for the weight distribution using an invariant property of a given code for permutation groups. The set of cosets of a subcode, which forms the original code, is partitioned into blocks. Two cosets are in the same block if and only if a permutation of coordinate places changes a coset into the other one. Consequently, two cosets in the same block have the same weight distribution. Therefore, it is enough to compute the number of blocks and the weight distributions of the representative coset in each block. Here, we examine the number of blocks and which cosets are in a same block for the two special cases.

3.1 Outline of the Method

In this section, we present another method with reduced computational complexity which uses the invariant property of a code for groups of bit position permutations. For a vector $\mathbf{u} = (u_1, u_2, \cdots, u_n)$ and a permutation π on $\{1, 2, \cdots, n\}$, let $\pi(\mathbf{u})$ be the permuted vector $(u_{\pi(1)}, u_{\pi(2)}, \cdots, u_{\pi(n)})$. For a set D of vectors of length n, let $\pi(D)$ denote $\{\pi(\mathbf{u}) : \mathbf{u} \in D\}$. We say that the set D is invariant under a permutation π if $\pi(D) = D$, and that D is invariant under a set Π of permutations if $\pi(D) = D$ for any $\pi \in \Pi$.

Let C be a binary linear block code and let C_0 be a linear subcode of C. For a group Π of permutations under which C_0 is invariant, we define an equivalence relation on C/C_0. Two cosets, $\mathbf{u}+C_0$ and $\mathbf{v}+C_0$, are equivalent if and only if there is a permutation $\pi \in \Pi$ such that $\pi(\mathbf{u}+C_0) = \mathbf{v}+C_0$. We call an equivalence class a block. A block containing $\mathbf{v}+C_0$ is the set of cosets, $\{\pi(\mathbf{v})+C_0 : \pi \in \Pi\}$. All the cosets in a block have the same weight distribution. Let the blocks be denoted by $B_1, B_2, \cdots, B_b$. For $1 \leq i \leq b$, let $W^{(i)}(x)$ denote the weight enumerator of a coset in B_i. To obtain the weight enumerator of C, it is enough to obtain the size of each block, $|B_i|$, and the weight enumerator $W^{(i)}(x)$ of representative coset for each block. Then, the weight enumerator of C is given by

$$\sum_{i=1}^{b} |B_i| W^{(i)}(x). \tag{3.1}$$

We must devise an efficient method for finding the number of cosets and the representative coset in each block. In the following, we consider the case where Π is the cyclic group or the affine group[5, 28]. Note that BCH codes are invariant under the cyclic group and the extended codes of binary primitive BCH codes are invariant under the affine group.

3.2 Partition of the Cosets with respect to the Cyclic Group

Consider the case where C is a cyclic code of length $n = 2^m - 1$. Let $C^{(\mathrm{irr})}(u)$ be the irreducible cyclic code of length $2^m - 1$ whose nonzeros are α^u and its conjugates, where α is a primitive element of GF(2^m). It is known that any cyclic code of length $2^m - 1$ can be represented as the direct sum of irreducible codes[5, Theorem 7, p. 220]. Suppose that $C \neq C^{(\mathrm{irr})}(0)$. Then, C can be expressed as

$$C = C^{(\mathrm{irr})}(u) + C_0, \quad \text{for} \quad u \neq 0, \tag{3.2}$$

where C_0 is a cyclic subcode of C. If u is relatively prime to $2^m - 1$, then for any nonzero codeword $\mathbf{v}$ of $C^{(\mathrm{irr})}(u)$, the following equation holds.

$$C^{(\mathrm{irr})}(u) = \{\sigma_0^j(\mathbf{v}) | 0 \leq j < n\} \cup \{\mathbf{0}\}, \tag{3.3}$$

where $\sigma_0^j(\mathbf{v})$ is the vector given by cyclically shifting $\mathbf{v}$ j times to the left, that is, for $\mathbf{v} = (v_1, v_2, \cdots, v_n)$,

$$\sigma_0^j(\mathbf{v}) = (v_{j+1}, \cdots, v_n, v_1, \cdots, v_j). \tag{3.4}$$

Now, consider the partition of C/C_0 with respect to the cyclic group of permutations. There are $(n+1)$ cosets of C_0,

$$C_0 \quad \text{and} \quad \sigma_0^j(\mathbf{v}) + C_0, \quad \text{for} \quad 0 \leq j < n, \tag{3.5}$$

in C/C_0, and C/C_0 is partitioned into two blocks. The block containing the coset C_0 consists of C_0 only, since C_0 is a cyclic subcode of C. The remaining n cosets form the other block, since for any j and j' with $0 \leq j < j' < n$, $\sigma_0^j(\mathbf{v}) + C_0$ can be translated into $\sigma_0^{j'}(\mathbf{v}) + C_0$ by the cyclic permutation $\sigma_0^{j'-j}$.

Therefore, we have the following lemma.

Lemma 2: Consider a cyclic code C of length $2^m - 1$. Suppose that $C^{(\mathrm{irr})}(u) \subseteq C$ for an integer u which is relatively prime to $2^m - 1$. Let C_0 be the cyclic subcode of C such that $C = C^{(\mathrm{irr})}(u) + C_0$, and let $W_D(x)$ denote the weight enumerator of a set D of vectors. Then, for any nonzero codeword $\mathbf{v} \in C^{(\mathrm{irr})}(u)$, the weight enumerator of C is given by

$$(2^m - 1)W_{\mathbf{v}+C_0}(x) + W_{C_0}(x). \tag{3.6}$$

$\Delta\Delta$

It is straightforward to modify Lemma 2 for a extended cyclic code. Let $\mathbf{v}_{\mathrm{ex}}$ denote the extended vector of $\mathbf{v}$, and let $D_{\mathrm{ex}} = \{\mathbf{v}_{\mathrm{ex}} : \mathbf{v} \in D\}$ for a set D of vectors. Hence, the following corollary holds.

Corollary 1: For any nonzero codeword $\mathbf{v} \in C^{(\mathrm{irr})}(u)$, the weight enumerator of C_{ex} is given by

$$(2^m - 1)W_{\mathbf{v}_{\mathrm{ex}}+C_{0,\mathrm{ex}}}(x) + W_{C_{0,\mathrm{ex}}}(x), \tag{3.7}$$

where $C_{0,\mathrm{ex}}$ denotes the extended code of C_0. $\Delta\Delta$

3.3 Partition of the Cosets with respect to the Affine Group

For an integer i with $0 \leq i < 2^m$, let $(i_1, i_2, \cdots, i_m)$ be the binary representation of i, i.e.,

$$i = \sum_{j=1}^{m} i_j 2^{j-1}, \tag{3.8}$$

and for an m-variable Boolean polynomial $F(\mathbf{X})$ (or simply F), where $\mathbf{X} = (X_1, X_2, \cdots, X_m)$ with Boolean variable X_i for $1 \leq i \leq m$, its vector representation $\mathbf{v}_F$ is defined as $(v_0, v_1, \cdots, v_{2^m-1})$, where $v_i = F(i_1, i_2, \cdots, i_m)$. We say that this 2^m-tuple is in standard order of bit positions[24]. For a set D of vectors, we write $F \in D$ if and only if $\mathbf{v}_F \in D$.

We consider a linear block code C of length 2^m such that

$$\forall F(\mathbf{X}) \in C \, \forall \mathbf{a} \in \mathrm{GF}(2)^m [F(\mathbf{X} + \mathbf{a}) \in C], \tag{3.9}$$

$$\mathrm{RM}_{m,r-2} \subseteq C \subseteq \mathrm{RM}_{m,r}, \quad \text{and} \quad C \not\subseteq \mathrm{RM}_{m,r-1}. \tag{3.10}$$

For a code C which satisfies (3.9) and (3.10), we have the following lemma.

Lemma 3: Let C be a linear block code of length 2^m which satisfies (3.9) and (3.10). For a Boolean polynomial $F \in C$ of degree r and $\mathbf{a} \in \mathrm{GF}(2)^m$, let $F_{\mathbf{a}}$ be the Boolean formula which is obtained from $F(\mathbf{X} + \mathbf{a}) - F(\mathbf{X})$ by deleting all the terms with the degree less than or equal to $r - 2$.

(1) Let ΔF be a set of Boolean polynomials $\{F_{\mathbf{a}} : \mathbf{a} \in \mathrm{GF}(2)^m\}$. Then, ΔF is a linear subcode of $C \cap \mathrm{RM}_{m,r-1}$.

(2) Let $C(F)$ be a linear subcode of C such that $C \cap \mathrm{RM}_{m,r-1} = C(F) + \Delta F$ and $C(F) \cap \Delta F = \{\mathbf{0}\}$. Then, $C(F)$ is invariant under the bit permutation $\mathbf{X} \to \mathbf{X} + \mathbf{a}$ for $\mathbf{a} \in \mathrm{GF}(2)^m$.

(3) The weight enumerator of the coset $F + C \cap \mathrm{RM}_{m,r-1} = F + C(F) + \Delta F$ is given by

$$|\Delta F| W_{F+C(F)}(x). \tag{3.11}$$

(Proof)

(1) Since C satisfies (3.9), $F_{\mathbf{a}} \in C$. The degree of $F_{\mathbf{a}}$ is equal to $r - 1$ or 0, and $F_{\mathbf{a}} \in C \cap \mathrm{RM}_{m,r-1}$. We will prove that

$$F_{\mathbf{a}} + F_{\mathbf{b}} = F_{\mathbf{a}+\mathbf{b}}, \quad \text{for} \quad \mathbf{a}, \mathbf{b} \in \mathrm{GF}(2)^m. \tag{3.12}$$

Let $\mathbf{a} = (a_1, a_2, \cdots, a_m)$ and $\mathbf{b} = (b_1, b_2, \cdots, b_m)$. For a monomial $G \triangleq X_{i_1} X_{i_2} \cdots X_{i_r}$ with $1 \leq i_1 < i_2 < \cdots < i_r \leq m$, we have that

$$G_{\mathbf{a}} + G_{\mathbf{b}} = \sum_{j=1}^{r} (a_{i_j} + b_{i_j}) X_{i_1} X_{i_2} \cdots X_{i_{j-1}} X_{i_{j+1}} \cdots X_{i_r} = G_{\mathbf{a}+\mathbf{b}}. \tag{3.13}$$

(3.12) follows from the fact that $F_{\mathbf{a}}$ (or $F_{\mathbf{b}}$) is zero or can be expressed as a sum of some monomials and (3.13).

(2) Since both C and $\mathrm{RM}_{m,r-1}$ contain $\mathrm{RM}_{m,r-2}$, $C \cap \mathrm{RM}_{m,r-1} = C(F) + \Delta F$ also contains $\mathrm{RM}_{m,r-2}$. From the definition of ΔF, $C(F)$ contains $\mathrm{RM}_{m,r-2}$. The degree of the coset leader of a coset in $C(F)/\mathrm{RM}_{m,r-2}$ is $r-1$ but that of $\mathrm{RM}_{m,r-2}$ itself. For an Boolean polynomial $H(\mathbf{X})$ with degree $r-1$ or less,

$$H(\mathbf{X}) + \mathrm{RM}_{m,r-2} = H(\mathbf{X} + \mathbf{a}) + \mathrm{RM}_{m,r-2}, \tag{3.14}$$

since $H(\mathbf{X}+\mathbf{a}) - H(\mathbf{X}) \in \mathrm{RM}_{m,r-2}$. Hence, $C(F)$ is invariant under the bit permutation $\mathbf{X} \rightarrow \mathbf{X} + \mathbf{a}$ for $\mathbf{a} \in \mathrm{GF}(2)^m$.

(3) Consider $F_{\mathbf{a}}, F_{\mathbf{b}} \in \Delta F$ with $\mathbf{a}, \mathbf{b} \in \mathrm{GF}(2)^m$. Then,

$$F(\mathbf{X}) + F_{\mathbf{a}}(\mathbf{X}) + C(F) = F(\mathbf{X} + \mathbf{a}) + C(F), \tag{3.15}$$

$$F(\mathbf{X}) + F_{\mathbf{b}}(\mathbf{X}) + C(F) = F(\mathbf{X} + \mathbf{b}) + C(F). \tag{3.16}$$

Since $F(\mathbf{X}+\mathbf{a})$ can be obtained from $F(\mathbf{X}+\mathbf{b})$ by the bit permutation $\mathbf{X} \rightarrow \mathbf{X}+\mathbf{a}+\mathbf{b}$ and $C(F)$ is invariant under the bit permutations $\mathbf{X} \rightarrow \mathbf{X} + \mathbf{a}$ for $\mathbf{a} \in \mathrm{GF}(2)^m$. Therefore, $F + F_{\mathbf{a}} + C(F)$ and $F + F_{\mathbf{b}} + C(F)$ have the same weight distribution, and we have (3.11). $\Delta\Delta$

Lemma 3 is a generalization of Lemma 5.1 in [8] which is used to reduce the computational complexity to obtain the weight distribution of cosets of $\mathrm{RM}_{8,2}$ in $\mathrm{RM}_{8,3}$[7, 8]. The special case of Lemma 3 where $C = \mathrm{RM}_{m,r}$ is Lemma 5.1 in [8]. As shown in Lemma 5.2 in [8], the size of ΔF is given by

$$\begin{aligned} \log_2 |\Delta F| = \min\{ & s : \exists A \in \mathrm{GL}(m,2), g \in P_s \\ & \text{such that } F(\mathbf{X}A) \in g + \mathrm{RM}_{m,r-1} \} \leq m, \end{aligned} \tag{3.17}$$

where P_s is the set of all Boolean polynomials of s variables, and $\mathrm{GL}(m,2)$ is the general linear group which is the set of all $m \times m$ invertible matrices over GF(2)[5].

From Lemmas 2 and 3, we have the following theorem.

Theorem 3: Let π_0 be a permutation on $\{1, 2, \cdots, 2^m\}$ from the cyclic order to the standard order of bit positions[24]. Let $C_{\rm ex}$ be an extended cyclic code of length 2^m such that

(a) $C_{\rm ex}$ is invariant under the affine group of permutations,

(b) $\mathrm{RM}_{m,r-2} \subseteq \pi_0(C_{\rm ex}) \subseteq \mathrm{RM}_{m,r}$, $\pi_0(C_{\rm ex}) \not\subseteq \mathrm{RM}_{m,r-1}$ and

(c) $\pi_0(C_{\rm ex}) = \pi_0(C_{\rm ex}^{(\rm irr)}(u)) + \pi_0(C_{\rm ex}) \cap \mathrm{RM}_{m,r-1}$ for an integer u relatively prime to $2^m - 1$, where $C_{\rm ex}^{(\rm irr)}(u)$ is the extended code of the irreducible cyclic code $C^{(\rm irr)}(u)$.

Then, the weight enumerator $W_{C_{\rm ex}}(x)$ of $C_{\rm ex}$ is given by

$$W_{C_{\rm ex}}(x) = (2^m - 1)|\Delta F| W_{F+C(F)}(x) + W_{\pi_0(C_{\rm ex}) \cap \mathrm{RM}_{m,r-1}}(x), \tag{3.18}$$

where $F \in \pi_0(C_{\rm ex})$ is any Boolean polynomial of degree r and $C(F)$ is such a subcode of $\pi_0(C_{\rm ex}) \cap \mathrm{RM}_{m,r-1}$ that satisfies $\pi_0(C_{\rm ex}) \cap \mathrm{RM}_{m,r-1} = C(F) + \Delta F$ and $C(F) \cap \Delta F = \{\mathbf{0}\}$.

$\Delta\Delta$

Chapter 4

The Weight Distributions of Extended Codes of Binary Primitive BCH Codes of Lengths 64 and 128

For the extended binary primitive BCH codes of length 128, the formulas of the weight distributions are known for some high-rate codes for which the number of information bits is $127 - 7t$ with $1 \leq t \leq 3$[5, 10] and for some low-rate codes for which that is $7t' + 1$ with $1 \leq t' \leq 3$[11]. By using the techniques in Chapters 2 and 3, we compute the weight distributions of all the remaining extended binary primitive BCH code of length 128. We also compute the weight distributions of all the extended binary primitive BCH code of length 64.

4.1 Computing Method

We discuss an efficient method for computing the weight distribution of $(128, k)$ extended binary primitive BCH code with $29 \leq k \leq 99$ and $(64, k)$ extended binary primitive BCH code with any k.

Let α be a primitive element of GF(2^m). The cyclic r-th order RM code of length $2^m - 1$, denoted c-RM$_{m,r}$, is defined as a cyclic code generated by the polynomial which has α^u as a root if and only if the following holds[9, Theorem 8.1, p. 229],

$$0 < \sum_{j=1}^{m} u_j \leq m - r - 1, \quad \text{where} \quad u = \sum_{j=1}^{m} u_j 2^{j-1}. \tag{4.1}$$

The extended code of c-$\mathrm{RM}_{m,r}$ is equivalent to $\mathrm{RM}_{m,r}$[28, p. 323]. Let π_0 be a permutation from the cyclic order to the standard order of bit positions[24]. By this permutation, the extended code of c-$\mathrm{RM}_{m,r}$ is transformed into $\mathrm{RM}_{m,r}$. If $C = \pi_0(D)$ for the extended code D of a cyclic code, the property (3.9) holds for C if D is invariant under the affine group. Note that the extended code of a BCH code and its dual code are invariant under the affine group[5, 28].

For a linear block code C, $C^{\perp}$ denotes the dual code of C. Let $\mathrm{BCH}(2^m-1, k, 2t+1)$ (or $\mathrm{BCH}(2^m-1, k)$ for simplicity) denote a binary primitive $(2^m-1, k)$ BCH code with designed distance $2t+1$. Let $\mathrm{EBCH}(2^m, k, 2t+2)$ (or $\mathrm{EBCH}(2^m, k)$ for simplicity) also denote the extended code of $\mathrm{BCH}(2^m-1, k, 2t+1)$. The zeros of a cyclic code are the roots of its generator polynomial, and the nonzeros of a cyclic code are the roots of its check polynomial. For a binary BCH code with designed distance $2t+1$, its zeros consist of α^h with $0 < h < 2t+1$ and their conjugates. Table 4.1 shows the nonzeros of several cyclic Reed-Muller codes, BCH codes and the dual codes of BCH codes considered in this dissertation.

(A) $\mathrm{EBCH}(128, k)$ with $k = 29, 36, 43$ and 50:

$\mathrm{EBCH}(128, 29)$ and $\mathrm{EBCH}(128, 36)$ are small enough to compute their weight distributions by using the trellis-based computing method only.

Consider the case where $k = 43$ or 50. From Table 4.1, we have that

$$\pi_0(\mathrm{EBCH}(128, 43)) \subseteq \pi_0(\mathrm{EBCH}(128, 50)) \subseteq \mathrm{RM}_{7,3}. \tag{4.2}$$

We can apply Lemma 3 with $r = 3$ to reduce the computational complexity to obtain the weight distributions of certain cosets of $\mathrm{RM}_{7,1}$. But we do not use this approach, since $\mathrm{RM}_{7,1}$ is a small code and hence the improved trellis-based computing method for $\mathrm{RM}_{7,1}$ does not reduce the computational complexity very much.

We only use Corollary 1. We choose $\mathrm{EBCH}(128, k-7)$ as $C_{0,\mathrm{ex}}$ for $\mathrm{EBCH}(128, k)$. Once we have computed the weight distribution of $\mathrm{EBCH}(128, k-7)$, we only need to compute the weight distribution of one coset of $\mathrm{EBCH}(128, k-7)$ to obtain the weight distribution of $\mathrm{EBCH}(128, k)$.

Since $\pi_0(\mathrm{EBCH}(128, k-7))$ contains $\mathrm{RM}_{7,2}$ and has a relatively simple trellis structure, the weight distribution of the coset is computed by the improved trellis-

based computing method efficiently.

(B) EBCH$(128, k)$ with $k = 78, 85, 92$ and 99:

For these codes, we compute the weight distribution of the dual codes first. Then, we compute the weight distributions of the original codes with MacWilliams' identity. From Table 4.1, we have that

$$\pi_0(\text{EBCH}(128,99)^\perp) \subseteq \pi_0(\text{EBCH}(128,92)^\perp) \subseteq \pi_0(\text{EBCH}(128,85)^\perp)$$
$$\subseteq \pi_0(\text{EBCH}(128,78)^\perp) \subseteq \text{RM}_{7,3}. \tag{4.3}$$

We only use Corollary 1 with $C_{0,\text{ex}} = \text{EBCH}(128, k+7)^\perp$.

Table 4.1 Nonzeros of several codes of length 128.

Code	NZ
c-RM$_{7,4}$	7, 11, 13, 15, 19, 21, 23, 27, 29, 31, 43, 47, 55, 63
c-RM$_{7,3}$	15, 23, 27, 29, 31, 43, 47, 55, 63
BCH$(127,57)^\perp$	7, 15, 23, 27, 29, 31, 43, 47, 55, 63
BCH$(127,64)^\perp$	7, 15, 23, 27, 29, 31, 47, 55, 63
BCH$(127,71)^\perp$	7, 15, 23, 29, 31, 47, 55, 63
BCH$(127,78)^\perp$	15, 23, 29, 31, 47, 55, 63
BCH$(127,85)^\perp$	15, 29, 31, 47, 55, 63
BCH$(127,92)^\perp$	15, 31, 47, 55, 63
BCH$(127,99)^\perp$	15, 31, 47, 63
BCH$(127,64)$	21, 23, 27, 29, 31, 43, 47, 55, 63
BCH$(127,57)$	23, 27, 29, 31, 43, 47, 55, 63
BCH$(127,50)$	27, 29, 31, 43, 47, 55, 63
BCH$(127,43)$	29, 31, 43, 47, 55, 63
BCH$(127,36)$	31, 43, 47, 55, 63
BCH$(127,29)$	43, 47, 55, 63
c-RM$_{7,2}$	31, 47, 55, 63

Nonzeros are $\{\alpha^h : h \in NZ\}$ and their conjugates.

(C) EBCH(128, 57):

EBCH(128, 57) is also a subcode of $\mathrm{RM}_{7,3}$. We may compute the weight distribution by using Corollary 1 with $C_{0,\mathrm{ex}} = \mathrm{EBCH}(128, 50)$. But there is the following more efficient way.

From Table 4.1, we have that

$$\mathrm{EBCH}(128, 57)^{\perp} = C^{(\mathrm{irr})}(7) + \mathrm{RM}_{7,3}. \tag{4.4}$$

By using Corollary 1 with $C_{0,\mathrm{ex}} = \mathrm{RM}_{7,3}$, we can compute the weight distribution of $\mathrm{EBCH}(128, 57)^{\perp}$. All we need to compute is the weight distribution of a coset of $\mathrm{RM}_{7,3}$, since that of $\mathrm{RM}_{7,3}$ was already computed[29]. Although $\mathrm{RM}_{7,3}$ is larger than EBCH(128, 50), we found that the computational complexity to obtain the weight distribution of a coset of $\mathrm{RM}_{7,3}$ is less than that of EBCH(128, 50) by comparing the trellis complexities of them.

(D) EBCH(128, 64):

For this code, we use Theorem 3. As mentioned above, the condition (a) is satisfied. Condition (b) is satisfied for $m = 7$ and $r = 4$. From Table 4.1, we have that

$$\begin{aligned} \pi_0(\mathrm{EBCH}(128, 64)) &= \pi_0(C_{\mathrm{ex}}^{(\mathrm{irr})}(21)) + \pi_0(\mathrm{EBCH}(128, 64)) \cap \mathrm{RM}_{7,3} \\ &= \pi_0(C_{\mathrm{ex}}^{(\mathrm{irr})}(21)) + \pi_0(\mathrm{EBCH}(128, 57)). \end{aligned} \tag{4.5}$$

Then, the condition (c) is also satisfied.

To reduce the computational complexity, we should choose the boolean polynomial F such that the number of codewords in ΔF is as large as possible. In this case, we compute $|\Delta F|$ for every F in (128,7) $\pi_0(C_{\mathrm{ex}}^{(\mathrm{irr})}(21))$ code, and it turned out that $|\Delta F| = 2^7$ for any $F \in \pi_0(C_{\mathrm{ex}}^{(\mathrm{irr})}(21))$.

After computing the weight distribution of EBCH(128, 57), all we have to do is to compute the weight distribution of the coset $F + C(F)$, where $C(F)$ is a (128, 50) subcode of EBCH(128, 57). This computation is done by using the improved trellis-based computing method.

(E) EBCH(128, 71):

For this code, we first compute the weight distribution of its dual code. Theorem 3

is also applied to the dual code. From Table 4.1, the following equation holds,

$$\pi_0(\mathrm{EBCH}(128,71)^{\perp}) = \pi_0(C_{\mathrm{ex}}^{(\mathrm{irr})}(7)) + \pi_0(\mathrm{EBCH}(128,71)^{\perp}) \cap \mathrm{RM}_{7,3}. \tag{4.6}$$

Three conditions for the theorem are also satisfied. In this case, it also holds that $|\Delta F| = 2^7$ for any $F \in \pi_0(C_{\mathrm{ex}}^{(\mathrm{irr})}(7))$.

The weight distribution of the (128,50) code, $\pi_0(\mathrm{EBCH}(128,71)^{\perp}) \cap \mathrm{RM}_{7,3}$, and that of one coset of the (128,43) code, $C(F)$, are computed by using the improved trellis-based computing method.

(F) EBCH$(64,k)$ with any k:

For these codes, we may use a trellis-based computing method only. It is feasible to compute the weight distributions in reasonable time.

4.2 Weight Distributions

The weight distributions are listed in Tables 4.2 to 4.9. Only the numbers $A_{\mathrm{ex},w}$ of codewords of weight w with $0 < w \leq n/2$ which are not zero are listed in these tables. The number of codewords of weight w with $n/2 < w \leq n$ equals that of weight $n - w$. We also compute the weight distributions of their dual codes by MacWilliams' identity. Then, it turned out that both EBCH(n,k) and EBCH$(n,n-k)^{\perp}$ have the same weight distribution when $k = 29, 36, 43, 64, 85, 92$ and 99 for $n = 128$ and $k = 7$ and 57 for $n = 64$. Especially, EBCH$(128, 64, 22)$ is formally self-dual, that is, the dual code has the same weight distribution with the original code, although they are not the same code[5, p. 596].

The weight distribution of a binary primitive BCH code can be easily obtained from that of its extended code[5, p. 232], [28, p. 246]. Let A_w and $A_{\mathrm{ex},w}$ denote the number of codewords of weight w of a binary primitive BCH code of length $2^m - 1$ and that of its extended code, respectively. Then,

$$A_{2i-1} = \frac{2i}{2^m} A_{\mathrm{ex},2i}, \quad \text{for} \quad 0 < i \leq 2^{m-1}, \tag{4.7}$$

$$A_{2i} = \frac{2^m - 2i}{2^m} A_{\mathrm{ex},2i}, \quad \text{for} \quad 0 \leq i < 2^{m-1}. \tag{4.8}$$

Table 4.2 The weight distributions of EBCH$(128, 29, 44)$ (or EBCH$(128, 99, 10)^{\perp}$) and EBCH$(128, 36, 32)$ (or EBCH$(128, 92, 12)^{\perp}$).

w	$A_{\mathrm{ex},w}$ for EBCH$(128, 29, 44)$ (or EBCH$(128, 99, 10)^{\perp}$)	$A_{\mathrm{ex},w}$ for EBCH$(128, 36, 32)$ (or EBCH$(128, 92, 12)^{\perp}$)
32	0	10668
36	0	16256
40	0	2048256
44	373888	35551872
48	2546096	353494848
52	16044672	2028114816
56	56408320	7216135936
60	116750592	14981968512
64	152623774	19484794406

Table 4.3 The weight distributions of EBCH$(128, 43, 30)$ (or EBCH$(128, 85, 14)^{\perp}$), EBCH$(128, 50, 28)$ and EBCH$(128, 57, 24)$.

w	$A_{\mathrm{ex},w}$ for EBCH$(128, 43, 30)$ (or EBCH$(128, 85, 14)^{\perp}$)	$A_{\mathrm{ex},w}$ for EBCH$(128, 50, 28)$	$A_{\mathrm{ex},w}$ for EBCH$(128, 57, 24)$
24	0	0	597408
28	0	186944	24579072
32	124460	19412204	2437776684
36	8810752	1103839296	141621881856
40	263542272	33723852288	4315318568736
44	4521151232	579267441920	74150180302848
48	44899876672	5744521082944	735289205007168
52	262118734080	33558415333632	4295496356229120
56	915924097536	117224663972352	15004724612905792
60	1931974003456	247312085243776	31655991621445632
64	2476672341286	316992306111910	40574965317267238

Table 4.4 The weight distributions of EBCH$(128, 64, 22)$ (or EBCH$(128, 64, 22)^{\perp}$), EBCH$(128, 71, 20)$ and EBCH$(128, 78, 16)$.

w	$A_{\mathrm{ex},w}$ for EBCH$(128, 64, 22)$ (or EBCH$(128, 64, 22)^{\perp}$)	$A_{\mathrm{ex},w}$ for EBCH$(128, 71, 20)$	$A_{\mathrm{ex},w}$ for EBCH$(128, 78, 16)$
16	0	0	387096
18	0	0	5462016
20	0	2674112	213018624
22	243840	37486336	5309859840
24	6855968	839699616	107350803840
26	107988608	13825045248	1766071867392
28	1479751168	188001347136	24074650400768
30	16581217536	2140095182336	273932927993856
32	161471882796	20510697927468	2625267567169884
34	1292241296640	166689980438016	21336485108951040
36	9106516329984	1156658661471040	148052866301892608
38	53383279307904	6886497209935616	881470039149213696
40	278420690161824	35363776220195360	4526561735332554624
42	1218666847725184	157207798773129984	20122606565844068352
44	4782630191822848	607468163067994304	77755925658495682560
46	15858705600596992	2045773679068686336	261859003134276581376
48	47425684161326912	6023796954778012480	771046023044966543784
50	120442185147493376	15537040516548126720	198874124912401137254 4
52	277061634654099456	35191124114633006464	450446382891185969971 2
54	543244862505775360	70078589269156969984	897005932881366583296 0
56	967799721857135168	122925566952088660288	1573447271016983141248 0
58	1473287478189735168	190054082758956107264	2432692269013718774169 6
60	2041819511308530688	259342737902840355456	3319587022194492448358 4
62	2421550630907043328	312380032198035579904	3998464407989233708646 4
64	2617075886216910118	332409207867786543910	4254837887630251351495 0

Table 4.5 The weight distributions of EBCH$(128, 85, 14)$ (or EBCH$(128, 43, 30)^{\perp}$), EBCH$(128, 92, 12)$ (or EBCH$(128, 36, 32)^{\perp}$) and EBCH$(128, 99, 10)$ (or EBCH $(128, 29, 44)^{\perp}$).

w	$A_{\mathrm{ex},w}$ for EBCH$(128, 85, 14)$ (or EBCH$(128, 43, 30)^{\perp}$)	$A_{\mathrm{ex},w}$ for EBCH$(128, 92, 12)$ (or EBCH$(128, 36, 32)^{\perp}$)	$A_{\mathrm{ex},w}$ for EBCH$(128, 99, 10)$ (or EBCH$(128, 29, 44)^{\perp}$)
10	0	0	796544
12	0	1194816	90180160
14	341376	45646848	6463889536
16	22121368	2751682584	347764539928
18	856967552	110071456768	14127559573120
20	27230880768	3484410778688	445754705469248
22	680417833472	87099309355008	11149685265467776
24	13721772977024	1756359917165952	224811690627712384
26	226128254847488	28944450656120832	3700489537780219110 4
28	3081454360189952	394426389988237184	50486556173121673600
30	35064826913355520	4488297727663171584	574502176730571255552
32	336014520825141340	43009842715896693084	5505259786944679990620
34	2731238665152128768	34959871757858753 1264	44748635720273383143168
36	18949612280501341184	242554919872597678976	3104702962799943092975 36
38	112834993226032103936	1444288602806763978 3424	1848689417301349247899904
40	579364846705294996864	7415866532060410558 0416	9492309123731911851566976
42	2575849616631486204416	329708906635048784769024	4220274021289462404510 3744
44	995215572807115388 2112	1273875330862725405590976	1630560417423899918822325 12
46	33519982404512223401600	4290559778009132197764096	5491916536029199089614841 60
48	98687914666573428364840	1263204709961981875163 9976	1616902022803263350264149928
50	254574296248800159922816	3258552533730703659129 1392	4170947258582865019960480640
52	576536456040619165149184	7379663146092432776151 1104	9445968792041391795950926784
54	1148237129819878789497856	1469744221488665142430 84288	1881272610465098466814531 2896
56	2013890548891825020657408	2577778683006800236932 47232	3299556702053516278220243 4304
58	3114034684742715393815552	3985966282327258318095 23712	5102036860227828704470159 9232
60	4248814088020530790422528	5438479459612333934726 54592	6961253682581094321172612 1216
62	5118344400874949289841152	6551483932680756588723 8080	8385899464831778035255231 5392
64	5445862703373444517825478	6970700962464131491457 13094	8922497198963119451267798 6758

Table 4.6 The weight distributions of EBCH$(64, 7, 31)$ (or EBCH$(64, 57, 3)^{\perp}$), EBCH$(64, 10, 27)$ and EBCH$(64, 16, 23)$.

w	$A_{ex,w}$ for EBCH$(64, 7, 31)$ (or EBCH$(64, 57, 3)^{\perp}$)	$A_{ex,w}$ for EBCH$(64, 10, 27)$	$A_{ex,w}$ for EBCH$(64, 16, 23)$
24	0	0	5040
28	0	448	12544
32	126	126	30366

Table 4.7 The weight distributions of EBCH$(64, 18, 21)$, EBCH$(64, 24, 15)$ and EBCH$(64, 30, 13)$.

w	$A_{ex,w}$ for EBCH$(64, 18, 21)$	$A_{ex,w}$ for EBCH$(64, 24, 15)$	$A_{ex,w}$ for EBCH$(64, 30, 13)$
14	0	0	8064
16	0	2604	30828
18	0	10752	631680
20	0	0	1128960
22	4224	216576	14022144
24	5040	291648	14629440
26	24192	1645056	105057792
28	12544	888832	65046016
30	69888	4419072	282933504
32	30366	1828134	106764966

Table 4.8 The weight distributions of EBCH$(64, 36, 11)$, EBCH$(64, 39, 9)$ and EBCH$(64, 45, 7)$.

w	$A_{\mathrm{ex},w}$ for EBCH$(64, 36, 11)$	$A_{\mathrm{ex},w}$ for EBCH$(64, 39, 9)$	$A_{\mathrm{ex},w}$ for EBCH$(64, 45, 7)$
8	0	0	27288
10	0	13888	501760
12	30240	172704	12738432
14	354816	2874816	182458368
16	3583020	29210412	1862977116
18	27105792	214597824	13739292672
20	145061280	1168181280	74852604288
22	603113472	4794749760	306460084224
24	1853011776	14924626752	956270217000
26	4517259264	35889146496	2294484111360
28	8269968448	66620912960	4268285380352
30	12166253568	96671788416	6180152832000
32	13547993382	109123263270	6991765639110

Table 4.9 The weight distributions of EBCH$(64, 51, 5)$ and EBCH$(64, 57, 3)$ (or EBCH$(64, 7, 31)^{\perp}$).

w	$A_{\mathrm{ex},w}$ for EBCH$(64, 51, 5)$	$A_{\mathrm{ex},w}$ for EBCH$(64, 57, 3)$ (or EBCH$(64, 7, 31)^{\perp}$)
4	0	10416
6	20160	1166592
8	1067544	69194232
10	37051840	2366570752
12	801494400	51316746768
14	11684617344	747741998592
16	119266575708	7633243745820
18	879321948288	56276359749120
20	4789977429888	306558278858160
22	19616032446528	1255428754917120
24	61193769988008	3916392495228360
26	146864398476096	9399341113166592
28	273137809339136	17480786291963792
30	395577405119232	25316999607653376
32	447418802536902	28634752793916486

4.3 Probability of an Undetectable Error

For an (n, k) binary linear code C, let $P_{\text{ue}}(C, \varepsilon)$ denote the probability of an undetectable error when C is used for error detection in a binary symmetric channel with bit-error rate ε. Then, using the weight enumerator $W_C(x)$ of C, $P_{\text{ue}}(C, \varepsilon)$ can be expressed as follows:

$$P_{\text{ue}}(C, \varepsilon) = (1-\varepsilon)^n \left\{ W_C\left(\frac{\varepsilon}{1-\varepsilon}\right) - 1 \right\}. \tag{4.9}$$

The probabilities of an undetectable error for EBCH$(128, k)$ with $29 \leq k \leq 99$ are computed from their weight distributions and are shown in Figure 1.

A code C is called *proper* if $P_{\text{ue}}(C, \varepsilon)$ is monotonously increasing as ε increases. From Figure 1, we see that EBCH$(128, k)$ with $k = 36, 57$ and 78 are not proper. From [30, Theorem 3.4.2, p. 74], if the probability of an undetectable error for the dual code $C^{\perp}$ of C, $P_{\text{ue}}(C^{\perp}, \varepsilon)$, is greater than 2^{-k} for some ε, then $P_{\text{ue}}(C, \frac{1-2\varepsilon}{2-2\varepsilon}) > 2^{-(n-k)} > P_{\text{ue}}(C, 1/2)$, and hence C is not proper. By examining the probability of an undetectable error for the dual code, we see that EBCH$(128, k)$ with $k = 71$ and 92 are also not proper. For the remaining cases, EBCH$(128, k)$ with $k = 29, 43, 50, 64, 85$ and 99, the probabilities of an undetectable error are monotonously increasing as ε increases within the accuracy of our computation.

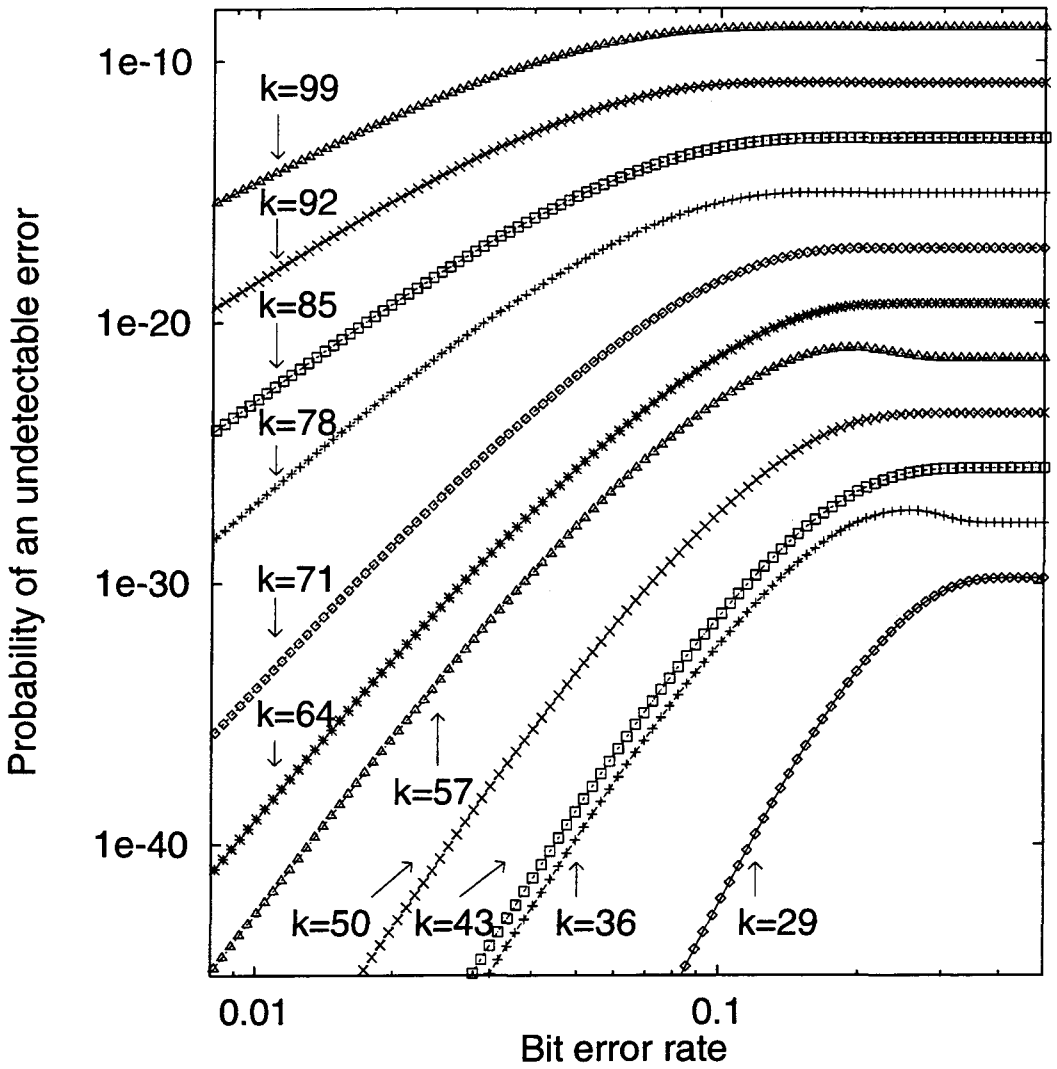

Figure 1: The probability of an undetectable error for EBCH$(128, k)$.

Chapter 5

Upper and Lower Bounds on the Undetected Error Probability of Binary Codes Derived from Shortened Reed-Solomon Codes

Although we already have two efficient computing method for a weight distribution of a linear block code given in Chapter 2 and 3, it may be still infeasible to compute the weight distributions for codes with large parameters. For such codes, we may choose the other strategy, that is, to compute the some part of weight distribution and to estimate the upper and lower bounds on the probability of an undetectable error from the results instead of computing the whole weight distribution and examining the exact probability. In this chapter, we take the binary codes derived from some shortened Reed-Solomon codes as examples. From the results, we also derive upper and lower bounds on the probability of an undetectable error when the binary codes are used only for error detection in a binary symmetric channel.

5.1 Binary Weight Distribution of a Code over $GF(2^m)$

Let $\bar{\beta} = \{\beta_1, \beta_2, \ldots, \beta_m\}$ be a basis of the Galois field $GF(2^m)$. Then each element γ in $GF(2^m)$ can be expressed as a linear sum of $\beta_1, \beta_2, \ldots, \beta_m$ as follows:

$$\gamma = c_1\beta_1 + c_2\beta_2 + \cdots + c_m\beta_m, \tag{5.1}$$

where $c_i \in GF(2)$ for $1 \leq i \leq m$. Thus γ can be represented by the m-tuple $(c_1, c_2, \ldots, c_m)$ over $GF(2)$. Let C be an $(n,\ k)$ linear block code with symbols from the Galois field $GF(2^m)$. If each code symbol of C is represented by the corresponding m-tuple over the binary field $GF(2)$ using the basis $\bar{\beta}$ of $GF(2^m)$, we obtain a binary $(mn,\ mk)$ linear block code, called a binary image of C. This binary code is denoted by $C(\bar{\beta})$. For a binary linear code C_B, let $A_i[C_B]$ be the number of codewords of weight i in C_B. The weight distribution of a binary image of C, $\{A_i[C(\bar{\beta})] : 0 \leq i \leq mn\}$, is called a binary weight distribution of C. In general, a binary weight distribution depends on the choice of basis. But there are bases for which the binary weight distributions are the same.

Lemma 4: Let C be a linear code of length n over $GF(2^m)$, and let $\bar{\beta}_1 = \{\beta_1, \beta_2, \ldots, \beta_m\}$ be a basis of $GF(2^m)$.

(1) For any nonzero element γ in $GF(2^m)$, define $\bar{\beta}_\gamma \triangleq \{\gamma\beta_1, \gamma\beta_2, \ldots, \gamma\beta_m\}$. Then,

$$A_i[C(\bar{\beta}_\gamma)] = A_i[C(\bar{\beta}_1)], \qquad \text{for } 0 \leq i \leq mn. \tag{5.2}$$

(2) Let $N = 2^m - 1$, and let α be a primitive element of $GF(2^m)$. There is a basis

$$\bar{\beta} = (\alpha^{b_1}, \alpha^{b_2}, \ldots, \alpha^{b_m}), \qquad 0 = b_1 < b_2 < \cdots < b_m \leq \frac{(m-1)N}{m}, \tag{5.3}$$

such that

$$A_i[C(\bar{\beta})] = A_i[C(\bar{\beta}_1)], \qquad \text{for } 0 \leq i \leq mn. \tag{5.4}$$

(Proof)

(1) For a codeword $(v_1, v_2, \ldots, v_n) \in C$ and a basis $\bar{\beta}_0$, let $(v_1, v_2, \ldots, v_n)_{\bar{\beta}_0}$ denote the corresponding codeword in $C(\bar{\beta}_0)$. For any codeword $(v_1, v_2, \ldots, v_n)_{\bar{\beta}_1} \in C(\bar{\beta}_1)$, $(\gamma v_1, \gamma v_2, \ldots, \gamma v_n)_{\bar{\beta}_\gamma}$ is a codeword of the same weight in $C(\bar{\beta}_\gamma)$, and for any codeword $(v_1, v_2, \ldots, v_n)_{\bar{\beta}_\gamma} \in C(\bar{\beta}_\gamma)$, $(\gamma^{-1}v_1, \gamma^{-1}v_2, \ldots, \gamma^{-1}v_n)_{\bar{\beta}_1}$ is a codeword of the same weight in $C(\bar{\beta}_0)$. Therefore, equation (5.2) holds.

(2) From (1) of this theorem, we see that there is a basis

$$\bar{\beta}' = \left\{\alpha^{b'_1}, \alpha^{b'_2}, \ldots, \alpha^{b'_m}\right\}, \qquad 0 = b'_1 < b'_2 < \cdots < b'_m < N, \tag{5.5}$$

such that

$$A_i[C(\bar{\beta}')] = A_i[C(\bar{\beta})], \qquad \text{for } 0 \le i \le mn. \tag{5.6}$$

Suppose that $b'_m > \frac{(m-1)N}{m}$. Since

$$\sum_{j=2}^{m}(b'_j - b'_{j-1}) = b'_m > \frac{(m-1)N}{m}, \tag{5.7}$$

there is j with $2 \le j \le m$ such that $(b_j - b_{j-1}) > \frac{N}{m}$. Consider a basis $\bar{\beta}$ which consists of $\alpha^{b'_i}\alpha^{-b'_j}$ with $1 \le i \le m$, that, is,

$$\bar{\beta} = \left\{\alpha^{b_1}, \alpha^{b_2}, \ldots, \alpha^{b_m}\right\}, \tag{5.8}$$

where

$$b_i \triangleq \begin{cases} b'_{i+j-1} - b'_j, & \text{for } 1 \le i \le m-j+1 \\ b'_{i+j-1-m} - b'_j + N, & \text{for } m-j+2 \le i \le m. \end{cases} \tag{5.9}$$

Then, we have that

$$0 = b_1 < b_2 < \cdots < b_m \le \frac{(m-1)N}{m}. \tag{5.10}$$

$\Delta\Delta$

5.2 The Number of Codewords with Small Weights of Some Shortened Reed-Solomon Codes

5.2.1 Shortened Reed-Solomon Codes Generated by $(X - \alpha)$

For a basis $\bar{\beta}$ in (5.3), let B be the set of integers defined by

$$B \triangleq \{b_1(=0), b_2, \ldots, b_m\}. \tag{5.11}$$

Let $\Gamma^{(n)}$ be a set of monomials (single term polynomials) of degree $n-1$ or less whose coefficient is in $\bar{\beta}$, that is,

$$\Gamma^{(n)} \triangleq \{\alpha^{b_i}X^j : 1 \le i \le m, 0 \le j < n\}. \tag{5.12}$$

We partition $\Gamma^{(n)}$ into subsets. For an integer l with $0 \le l < N$, define

$$\Gamma_l^{(n)} \triangleq \left\{\alpha^{b_i}X^j \in \Gamma^{(n)} : b_i + j \equiv l \pmod N\right\}. \tag{5.13}$$

For b_i and l, the number of integers j which satisfy $b_i + j \equiv l \pmod{N}$ is at most one. Therefore, we have that $\left|\Gamma_l^{(n)}\right| \leq m$. For an integer u with $1 \leq u \leq m$, let q_u be the number of sets $\Gamma_l^{(n)}$ such that $\left|\Gamma_l^{(n)}\right| = u$. For a given basis, it is easy to derive formulas of $q_1, q_2, \ldots, q_m$. Such formulas for the polynomial basis are given in the following example.

Example 1: We consider the case where $\bar{\beta}$ is the polynomial basis, that is,

$$\bar{\beta} = \bar{\beta}^{(P)} \triangleq \left\{1, \alpha, \alpha^2, \ldots, \alpha^{m-1}\right\}. \tag{5.14}$$

(1) If $1 < n < m$, we have that

$$q_i = \begin{cases} 2, & \text{for } 1 \leq i < n \\ m - n + 1, & \text{for } i = n \\ 0, & \text{for } n < i \leq m. \end{cases} \tag{5.15}$$

(2) If $m \leq n \leq 2^m - m$, we have that

$$q_i = \begin{cases} 2, & \text{for } 1 \leq i < m \\ n - m + 1, & \text{for } i = m. \end{cases} \tag{5.16}$$

(3) If $2^m - m < n < N$, we have that

$$q_i = \begin{cases} 0, & \text{for } 1 \leq i \leq n + m - N + 1 \\ n + m - N + 1, & \text{for } i = n + m - N \\ 2, & \text{for } n + m - N < i < m \\ n - m + 1, & \text{for } i = m. \end{cases} \tag{5.17}$$

$\Delta\Delta$

Next, we consider the number of codewords of weight 2 in $\mathrm{RS}_n^{(2,1)}(\bar{\beta})$, denoted by $A_2[\mathrm{RS}_n^{(2,1)}(\bar{\beta})]$.

We associate a polynomial with $\mathbf{v}$. For a vector $\mathbf{v} = (v_0, v_1, \ldots, v_{n-1})$, the corresponding polynomial is $\sum_{j=0}^{n-1} v_j X^j$.

For a binary linear code C_B, let $U_i[C_B]$ be the set of polynomials which correspond to codewords of weight i in C_B. From the definition, we have that

$$\begin{aligned} U_2[\mathrm{RS}_n^{(2,1)}(\bar{\beta})] &= \{f_1(X)+f_2(X) : f_1(X), f_2(X) \in \Gamma_l^{(n)} \text{ for an integer } l \\ &\qquad \text{and} \quad f_1(X) \neq f_2(X)\}. \end{aligned} \tag{5.18}$$

It follows from the definition of q_i and (5.18) that we have the following theorem.

Theorem 4: The number of codewords of weight 2 in $\mathrm{RS}_n^{(2,1)}(\bar{\beta})$ is given by

$$A_2[\mathrm{RS}_n^{(2,1)}(\bar{\beta})] = \sum_{i=2}^{m} q_i \binom{i}{2}. \tag{5.19}$$

$\Delta\Delta$

We also show that $A_2[\mathrm{RS}_n^{(2,1)}(\bar{\beta}^{(P)})]$ is the largest among all the bases. For two integers $b_{i_1}, b_{i_2} \in B$ with $i_1 < i_2$, define

$$Q_{b_{i_1},b_{i_2}} \triangleq \{\alpha^{b_{i_1}} X^{j_1} + \alpha^{b_{i_2}} X^{j_2} : 0 \leq j_1, j_2 < n \text{ and } b_{i_1}+j_1 \equiv b_{i_2}+j_2 \pmod N\}. \tag{5.20}$$

Then,

$$A_2[\mathrm{RS}_n^{(2,1)}(\bar{\beta})] = \sum_{i_1=1}^{m} \sum_{i_2=i_1+1}^{m} \left|Q_{b_{i_1},b_{i_2}}\right|. \tag{5.21}$$

From the definition of $Q_{b_{i_1},b_{i_2}}$,

$$\begin{aligned} Q_{b_{i_1},b_{i_2}} &= \{(\alpha^{b_{i_1}} X^{b_{i_2}-b_{i_1}} + \alpha^{b_{i_2}})X^j : 0 \leq j < n-(b_{i_2}-b_{i_1})\} \\ &\quad \cup \{(\alpha^{b_{i_2}} X^{N-b_{i_2}+b_{i_1}} + \alpha^{b_{i_1}})X^j : 0 \leq j < n+b_{i_2}-b_{i_1}-N\}. \end{aligned} \tag{5.22}$$

For a positive integer b, define

$$\tau(b) \triangleq \max\{n-b, 0\} + \max\{n+b-N, 0\}. \tag{5.23}$$

Then, it follows from (5.22) and (5.23) that

$$\left|Q_{b_{i_1},b_{i_2}}\right| = \tau(b_{i_2}-b_{i_1}), \quad \text{for } b_{i_1} < b_{i_2}. \tag{5.24}$$

Therefore, we have that

$$A_2[\mathrm{RS}_n^{(2,1)}(\bar{\beta})] = \sum_{i_1=1}^{m} \sum_{i_2=i_1+1}^{m} \tau(b_{i_2}-b_{i_1}). \tag{5.25}$$

Lemma 5: For two integers i_1, i_2 with $1 \leq i_1 < i_2 \leq m$, define $i \triangleq i_2 - i_1, b \triangleq b_{i_2} - b_{i_1}$. Then, we have that

$$\tau(b) \leq \tau(i). \tag{5.26}$$

(Proof)

From the definition of i, b and B, we have that

$$0 < i \leq m-1 \qquad \text{and} \qquad i \leq b \leq \frac{(m-1)N}{m}. \tag{5.27}$$

For any positive integer m, the following inequality holds.

$$m - 1 \leq N/m. \tag{5.28}$$

There are two cases to be considered.

(1) First, we consider the case where $n < N/2$. It follows from the definition of $\tau(b)$ that

$$\tau(b) = \begin{cases} n-b, & \text{for } 0 < b < n \\ 0, & \text{for } n \leq b \leq N-n \\ n+b-N, & \text{for } N-n < b \leq \frac{(m-1)N}{m}. \end{cases} \tag{5.29}$$

Since $\tau(b)$ decreases monotonically as b increases for $0 < b \leq N-n$, it follows from (5.27) that (5.26) holds if $b \leq N-n$. If $N-n < b \leq \frac{(m-1)N}{m}$, it follows from (5.28) that

$$\tau(b) = n+b-N \leq n + \frac{m-1}{m}N - N = n - \frac{N}{m} \leq n-(m-1). \tag{5.30}$$

Since $N-n < \frac{(m-1)N}{m}$, we have that $n > N/m \geq m-1 \geq i$. Therefore, we have that

$$\tau(i) = n - i. \tag{5.31}$$

The inequality (5.26) follows from (5.27), (5.30) and (5.31).

(2) Next, we consider the case where $n > N/2$. In this case, we have that

$$\tau(b) = \begin{cases} n-b, & \text{for } 0 < b < N-n \\ 2n-N, & \text{for } N-n \leq b \leq n \\ n+b-N, & \text{for } n < b \leq \frac{(m-1)N}{m}. \end{cases} \tag{5.32}$$

Since $\tau(b)$ decreases monotonically as b increases for $0 < b \leq n$, it follows from (5.27) that (5.26) holds for $0 < i \leq b \leq n$. If $n < b \leq \frac{(m-1)N}{m}$, we can show that $\tau(b) \leq \tau(i)$ by the similar way used for the case where $n < N/2$ and $N-n \leq b \leq \frac{(m-1)N}{m}$. $\Delta\Delta$

It follows from (5.25) and Lemma 5 that the following theorem holds.

Theorem 5: For any basis $\bar{\beta}$, we have that

$$A_2[\mathrm{RS}_n^{(2,1)}(\bar{\beta})] \leq A_2[\mathrm{RS}_n^{(2,1)}(\bar{\beta}^{(P)})] = \sum_{i=1}^{m-1}(m-i)\tau(i). \tag{5.33}$$

The equality in (5.33) holds if $\bar{\beta} = \bar{\beta}^{(P)}$. ΔΔ

Next theorem shows that there is a basis $\bar{\beta}$ for which $A_2[\mathrm{RS}_n^{(2,1)}(\bar{\beta})] = 0$.

Theorem 6: Define an integer ρ_0 as

$$\rho_0 \triangleq \left\lfloor \log_2 \frac{2^m - 1}{m} \right\rfloor. \tag{5.34}$$

There is a basis $\bar{\beta}$ such that

$$A_2[\mathrm{RS}_n^{(2,1)}(\bar{\beta})] = 0, \qquad \text{for } n \leq \max\{2^{\rho_0}, 2^m - 1 - (m-1)2^{\rho_0+1}\}. \tag{5.35}$$

(Proof)

For any integer ρ with $0 < \rho < 2^m - 1$ and $a_0, a_1, \ldots, a_{m-1} \in GF(2)$,

$$\sum_{i=0}^{m-1} a_i \alpha^{i2^\rho} = \left(\sum_{i=0}^{m-1} a_i \alpha^i \right)^{2^\rho} = 0, \tag{5.36}$$

if and only if $a_i = 0$ for $0 \leq i < m$. Therefore, $(1, \alpha^{2^\rho}, \alpha^{2 \cdot 2^\rho}, \ldots, \alpha^{(m-1)\cdot 2^\rho})$ is a basis. Define

$$\rho \triangleq \begin{cases} \rho_0, & \text{if } 2^{\rho_0} \geq 2^m - 1 - (m-1)2^{\rho_0+1} \\ \rho_0 + 1, & \text{otherwise.} \end{cases} \tag{5.37}$$

Then, it is easy to see that (5.35) holds for the basis $\bar{\beta} = (1, \alpha^{2^\rho}, \alpha^{22^\rho}, \ldots, \alpha^{(m-1)2^\rho})$.

ΔΔ

Next, we consider $A_w[\mathrm{RS}_n^{(2,1)}(\bar{\beta})]$ for $w \geq 3$. The following simple bound holds.

Theorem 7:

$$A_w[\mathrm{RS}_n^{(2,1)}(\bar{\beta})] \leq \frac{\min\{m,n\}}{w} \binom{mn}{w-1}. \tag{5.38}$$

ΔΔ

For a primitive polynomial of degree m and an integer n with $m+1 \leq n \leq N$, let HM_n (or $\mathrm{HM}_n[g(X)]$) denote the shortened Hamming code of length n with generator polynomial $g(X)$. By using a relation between the codewords in the shortened Hamming code $\mathrm{HM}_{\min\{n+b_m,N\}}$ and those of the binary image of the shortened Reed-Solomon code $\mathrm{RS}_n^{(2,1)}(\bar{\beta})$, we improve the bound on $A_w[\mathrm{RS}_n^{(2,1)}(\bar{\beta})]$ for small w. For a polynomial $\sum_{l=1}^{w} \alpha^{b_{i_l}} X^{j_l}$, define

$$\sigma^*(\sum_{l=1}^{w} \alpha^{b_{i_l}} X^{j_l}) \triangleq \sum_{l=1}^{w} X^{((j_l+b_{i_l}) \bmod N)}. \tag{5.39}$$

Then, $\alpha^{b_i} X^j \equiv \sigma^*(\alpha^{b_i} X^j) \pmod{g(X)}$. For monomials $f_i(X)$ with $1 \leq i \leq w$ in $\Gamma^{(n)}$, let

$$f_1(X) + f_2(X) + f_3(X) + \cdots + f_w(X) \tag{5.40}$$

be a codeword of weight w in $\mathrm{RS}_n^{(2,1)}(\bar{\beta})$. If $\sigma^*(f_{l_1}(X)) \neq \sigma^*(f_{l_2}(X))$ for any $f_{l_1}(X), f_{l_2}(X) (1 \leq l_1 < l_2 \leq w)$,

$$\sigma^*(f_1(X)) + \sigma^*(f_2(X)) + \sigma^*(f_3(X)) + \cdots + \sigma^*(f_w(X)) \tag{5.41}$$

is a codewords of weight w in $\mathrm{HM}_{\min\{n+b_m,N\}}$. We partition the set $U_w[\mathrm{RS}_n^{(2,1)}(\bar{\beta})]$ into two subsets. Let

$$U_w^{(1)}[\mathrm{RS}_n^{(2,1)}(\bar{\beta})] \triangleq \{f(X) \in U_w[\mathrm{RS}_n^{(2,1)}(\bar{\beta})] : \sigma^*(f(X)) \in U_w[\mathrm{HM}_{\min\{n+b_m,N\}}]\}, \tag{5.42}$$

$$U_w^{(2)}[\mathrm{RS}_n^{(2,1)}(\bar{\beta})] \triangleq U_w[\mathrm{RS}_n^{(2,1)}(\bar{\beta})] - U_w^{(1)}[\mathrm{RS}_n^{(2,1)}(\bar{\beta})]. \tag{5.43}$$

First, we consider codewords in $U_w^{(1)}[\mathrm{RS}_n^{(2,1)}(\bar{\beta})]$. For $X^j \in \Gamma^{(N)}$ and an integer i with $1 \leq i \leq m$, define

$$\sigma^i(X^j) \triangleq \alpha^{b_i} X^{((j-b_i) \bmod N)}. \tag{5.44}$$

Note that $X^j \equiv \sigma^i(X^j) \pmod{g(X)}$ for $1 \leq i \leq m$. Then, $U_w^{(1)}[\mathrm{RS}_n^{(2,1)}(\bar{\beta})]$ is the set of polynomials of degree $(n-1)$ or less in the set,

$$\{\sigma^{i_1}(f_1(X)) + \sigma^{i_2}(f_2(X)) + \sigma^{i_3}(f_3(X)) + \cdots + \sigma^{i_w}(f_w(X)) : 1 \leq i_1, i_2, \ldots i_w \leq m\}, \tag{5.45}$$

where

$$f_1(X) + f_2(X) + f_3(X) + \cdots + f_w(X) \tag{5.46}$$

is a codeword in the Hamming code, $\mathrm{HM}_{\min\{n+b_m,N\}}$, of weight w.

Therefore, we can compute $\left|U_w^{(1)}[\mathrm{RS}_n^{(2,1)}(\bar{\beta})]\right|$ by generating all the codewords of weight w in $\mathrm{HM}_{\min\{n+m-1,N\}}$. By using a similar method used in an algorithm to determine the minimum distance of a given shortened Hamming code in [31], we can compute $\left|U_w^{(1)}[\mathrm{RS}_n^{(2,1)}(\bar{\beta})]\right|$. The order of computing time is $O(\min\{n^{w-h-1}, n^h\})$, and the space complexity is $O(n^h)$, where h is a predesigned integer. We can also show that the following simple bound on $\left|U_w^{(1)}[\mathrm{RS}_n^{(2,1)}(\bar{\beta})]\right|$ holds.

Theorem 8: Let $d^{(H)}$ be the minimum weight of the shortened Hamming code, $\mathrm{HM}_{\min\{n+b_m,N\}}$.

(1) For an integer w with $2 \leq w < d^{(H)}$,

$$|U_w^{(1)}[\mathrm{RS}_n^{(2,1)}(\bar{\beta})]| = 0. \tag{5.47}$$

(2) For an integer w with $w \geq d^{(H)}$,

$$|U_w^{(1)}[\mathrm{RS}_n^{(2,1)}(\bar{\beta})]| \leq m^w A_w[\mathrm{HM}_{\min\{n+b_m,N\}}]. \tag{5.48}$$

$\Delta\Delta$

Next, we derive a formula for $|U_w^{(2)}[\mathrm{RS}_n^{(2,1)}(\bar{\beta})]|$ with a relatively small w in $4 \leq w \leq d^{(H)}+1$. The following lemma holds.

Lemma 6: (1) For an odd integer w with $3 \leq w \leq d^{(H)}+1$,

$$U_w^{(2)}[\mathrm{RS}_n^{(2,1)}(\bar{\beta})] = \emptyset. \tag{5.49}$$

(2) For an even integer w with $4 \leq w \leq d^{(H)}+1$,

$$\begin{aligned} U_w^{(2)}[\mathrm{RS}_n^{(2,1)}(\bar{\beta})] &= \{\mathbf{v}_1 + \mathbf{v}_2 + \cdots + \mathbf{v}_{w/2} : \mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_{w/2} \in U_2[\mathrm{RS}_n^{(2,1)}(\bar{\beta})] \\ &\qquad \text{and the weight of } \mathbf{v}_1 + \mathbf{v}_2 + \cdots + \mathbf{v}_{w/2} \text{ is w}\}. \end{aligned} \tag{5.50}$$

$\Delta\Delta$

By using (5.50), we can derive a formula for $|U_w^{(2)}[\mathrm{RS}_n^{(2,1)}(\bar{\beta})]|$ with a relatively small w. For example,

$$\left|U_4^{(2)}[\mathrm{RS}_n^{(2,1)}(\bar{\beta})]\right| = \binom{A_2[\mathrm{RS}_n^{(2,1)}(\bar{\beta})]}{2} - \sum_{i=4}^{m} 2q_i \binom{i}{4} - \sum_{i=3}^{m} 3q_i \binom{i}{3}. \tag{5.51}$$

5.2.2 Shortened Reed-Solomon Codes Generated by $(X-1)(X-\alpha)$

It is known that the minimum Hamming weight of $\mathrm{RS}_n^{(3,0)}$ is not less than 3. Since, $\mathrm{RS}_n^{(3,0)}(\bar{\beta})$ only contains codewords of even weight, the minimum Hamming weight of $\mathrm{RS}_n^{(3,0)}(\bar{\beta})$ is not less than 4. Note that $\mathrm{RS}_n^{(3,0)}(\bar{\beta})$ is a subcode of $\mathrm{RS}_n^{(2,1)}(\bar{\beta})$. For polynomials $f_1(X)$ and $f_2(X)$ in $U_2[\mathrm{RS}_n^{(2,1)}(\bar{\beta})]$, $f_1(1) = f_2(1)$ if and only if there are integers i_1, i_2 for which $f_1(X), f_2(X) \in Q_{b_{i_1}, b_{i_2}}$. Therefore, we have the following theorem.

Theorem 9: (1) The following equation holds.

$$\left|U_4^{(2)}[\mathrm{RS}_n^{(2,1)}(\bar{\beta})] \cap U_4[\mathrm{RS}_n^{(3,0)}(\bar{\beta})]\right| = \sum_{i_1=1}^{m} \sum_{i_2=i_1+1}^{m} \binom{\tau(b_{i_2} - b_{i_1})}{2}. \tag{5.52}$$

(2) For any basis $\bar{\beta}$,

$$\left|U_4^{(2)}[\mathrm{RS}_n^{(2,1)}(\bar{\beta})] \cap U_4[\mathrm{RS}_n^{(3,0)}(\bar{\beta})]\right| \leq \left|U_4^{(2)}[\mathrm{RS}_n^{(2,1)}(\bar{\beta}^{(P)})] \cap U_4[\mathrm{RS}_n^{(3,0)}(\bar{\beta}^{(P)})]\right|. \tag{5.53}$$

The equality holds if $\bar{\beta} = \bar{\beta}^{(P)}$.

$\Delta\Delta$

Theorem 10:

$$A_4[\mathrm{RS}_n^{(3,0)}(\bar{\beta})] \leq \left|U_4^{(2)}[\mathrm{RS}_n^{(2,1)}(\bar{\beta})] \cap U_4[\mathrm{RS}_n^{(3,0)}(\bar{\beta})]\right| + m^4 A_4[\mathrm{HM}_{n+b_m}]. \tag{5.54}$$

The equality holds if $A_4[\mathrm{HM}_{n+b_m}] = 0$. $\Delta\Delta$

Next we derive an upper bound on $A_w[\mathrm{RS}_n^{(3,0)}(\bar{\beta})]$. Suppose that $\sum_{i=1}^{m} \alpha^{b_i} \sum_{l=1}^{j_i} X^{P_l^{(i)}}$ is a code polynomial of weight $2w$ in $\mathrm{RS}_n^{(3,0)}(\bar{\beta})$. Then,

$$\sum_{i=1}^{m} j_i = w, \quad 0 \leq j_i \leq w \quad \text{and } 0 \leq P_l^{(i)} < n \quad \text{for } 1 \leq i \leq m \quad \text{and } 1 \leq l \leq j_i. \tag{5.55}$$

By counting the number of polynomials $\sum_{i=1}^{m} \alpha^{b_i} \sum_{l=1}^{j_i} X^{P_l^{(i)}}$, we can obtain an upper bound on $A_{2w}[\mathrm{RS}_n^{(3,0)}(\bar{\beta})]$, denoted $\bar{A}_{2w}[\mathrm{RS}_n^{(3,0)}(\bar{\beta})]$. For example, when $2w = 6$, we have that

$$\bar{A}_6[\mathrm{RS}_n^{(3,0)}(\bar{\beta})] = m\binom{n}{5} + 2\binom{m}{2}\binom{n}{4}\binom{n}{1} + 6\binom{m}{3}\binom{n}{2}^2\binom{n}{1}. \tag{5.56}$$

5.2.3 Other Shortened Reed-Solomon Codes

It is known that the minimum Hamming weight of $\mathrm{RS}_n^{(3,1)}$ is not less than 3, and that $\mathrm{RS}_n^{(3,1)}(\bar{\beta})$ is a subcode of $\mathrm{RS}_n^{(2,1)}(\bar{\beta})$.

We derive a lower bound on $A_4[\mathrm{RS}_n^{(3,1)}(\bar{\beta})]$ by counting the number of codewords in a subset of $U_4^{(2)}[\mathrm{RS}_n^{(2,1)}(\bar{\beta})] \cap U_4[\mathrm{RS}_n^{(3,1)}(\bar{\beta})]$.

Consider a codeword of weight 4 in $\mathrm{RS}_n^{(3,1)}(\bar{\beta})$,

$$\alpha^{b_{i_1}} X^{j_1} + \alpha^{b_{i_2}} X^{j_2} + \alpha^{b_{i_3}} X^{j_3} + \alpha^{b_{i_4}} X^{j_4}, \tag{5.57}$$

such that $\alpha^{b_{i_1}} X^{j_1} + \alpha^{b_{i_2}} X^{j_2}$ and $\alpha^{b_{i_3}} X^{j_3} + \alpha^{b_{i_4}} X^{j_4}$ are divisible by $(X-\alpha)$, and $\alpha^{b_{i_1}} X^{j_1} + \alpha^{b_{i_3}} X^{j_3}$ and $\alpha^{b_{i_2}} X^{j_2} + \alpha^{b_{i_4}} X^{j_4}$ are divisible by $(X-\alpha^2)$. For simplicity, we assume that $b_m + 2n \leq N$. Then,

$$\begin{cases} b_{i_1} + j_1 = b_{i_2} + j_2 \\ b_{i_3} + j_3 = b_{i_4} + j_4 \\ b_{i_1} + 2j_1 = b_{i_3} + 2j_3 \\ b_{i_2} + 2j_2 = b_{i_4} + 2j_4. \end{cases} \tag{5.58}$$

Without loss of generality, we assume that

$$i_1 = \min\{i_1, i_2, i_3, i_4\}. \tag{5.59}$$

Then, it follows from (5.58) that $i_2 > i_1$, $i_3 > i_1$ and $\min\{j_2, j_3, j_4\} = j_4$. Therefore, for each tuple of integers, (i_1, i_2, i_3, j_1) such that

$$\begin{cases} 1 \leq i_1 < i_2 \leq m \\ i_3 \in I_3(i_1, i_2) \triangleq \{i_3 | i_1 < i_3, b_{i_2} + b_{i_3} - b_{i_1} \in B \\ \quad \text{and } b_{i_3} - b_{i_1} \text{ is an even integer}\} \\ (b_{i_3} - 3b_{i_1})/2 + b_{i_2} \leq j_1 < n, \end{cases} \tag{5.60}$$

there is a polynomial in $U_4^{(2)}[\mathrm{RS}_n^{(2,1)}(\bar{\beta})] \cap U_4[\mathrm{RS}_n^{(3,1)}(\bar{\beta})]$. It is easy to see that for different tuples i_1, i_2, i_3, j_1, the polynomials are different. Therefore, we have the following theorem:

Theorem 11: If $b_m + n \leq N$, we have that

$$A_4[\mathrm{RS}_n^{(2,1)}(\bar{\beta})] \geq \sum_{i_1=1}^{m} \sum_{i_2=i_1+1}^{m} \sum_{i_3 \in I_3(i_1,i_2)} \max\left\{n - (b_{i_3} - 3b_{i_1})/2 - b_{i_2}, 0\right\}, \tag{5.61}$$

where $I_3(i_1, i_2)$ is defined in (5.60). The equality holds if and only if the minimum weight of the shortened Hamming code, HM_{2n+b_m}, is not less than 5. $\Delta\Delta$

We can also show that $A_4[\mathrm{RS}_n^{(4,0)}(\bar{\beta})]$ (or $A_4[\mathrm{RS}_n^{(4,1)}(\bar{\beta})]$) is equal to zero if the minimum weight of HM_{2n+b_m} (or HM_{3n+b_m}) is not less than 5.

5.3 Upper and Lower Bounds on the Probability of an Undetectable Error

In this section, we discuss the probability of an undetectable error when the code C is used only for error detection in the binary symmetric channel with bit-error rate ε. Let $P_{\mathrm{ue}}(C, \varepsilon)$ denote the probability of an undetectable error. By using the result in Section 5.2, we can derive the following upper and lower bounds on the probability of an undetectable error of $\mathrm{RS}_n^{(2,1)}(\bar{\beta})$.

$$\begin{aligned} P_{\mathrm{ue}}(\mathrm{RS}_n^{(2,1)}(\bar{\beta}), \varepsilon) &\leq A_2[\mathrm{RS}_n^{(2,1)}(\bar{\beta})]\varepsilon^2(1-\varepsilon)^{mn-2} \\ &\quad + \sum_{i=3}^{mn} \binom{mn}{i-1} \frac{\min\{m,n\}}{i} \varepsilon^i (1-\varepsilon)^{mn-i}, \qquad (5.62) \\ P_{\mathrm{ue}}(\mathrm{RS}_n^{(2,1)}(\bar{\beta}), \varepsilon) &\geq A_2[\mathrm{RS}_n^{(2,1)}(\bar{\beta})]\varepsilon^2(1-\varepsilon)^{mn-2} \\ &\quad + \left|U_4^{(2)}[\mathrm{RS}_n^{(2,1)}(\bar{\beta})]\right| \varepsilon^4 (1-\varepsilon)^{mn-4}. \qquad (5.63) \end{aligned}$$

If $n + b_m$ is not large, we can compute $A_3[\mathrm{HM}_{n+b_m}]$ and $A_4[\mathrm{HM}_{n+b_m}]$ by generating all the codewords of weight 3 and 4. By using $A_3[\mathrm{HM}_{n+b_m}]$ and $A_4[\mathrm{HM}_{n+b_m}]$, the following improved upper bound holds.

$$\begin{aligned} P_{\mathrm{ue}}(\mathrm{RS}_n^{(2,1)}(\bar{\beta}), \varepsilon) &\leq A_2[\mathrm{RS}_n^{(2,1)}(\bar{\beta})]\varepsilon^2(1-\varepsilon)^{mn-2} + m^3 A_3[\mathrm{HM}_{n+b_m}]\varepsilon^3(1-\varepsilon)^{mn-3} \\ &\quad + \{m^4 A_4[\mathrm{HM}_{n+b_m}] + \left|U_4^{(2)}[\mathrm{RS}_n^{(2,1)}(\bar{\beta})]\right|\}\varepsilon^4(1-\varepsilon)^{mn-4} \\ &\quad + \sum_{i=5}^{mn} \binom{mn}{i-1} \frac{\min\{m,n\}}{i} \varepsilon^i (1-\varepsilon)^{mn-i}, \qquad (5.64) \\ P_{\mathrm{ue}}(\mathrm{RS}_n^{(2,1)}(\bar{\beta}), \varepsilon) &\geq A_2[\mathrm{RS}_n^{(2,1)}(\bar{\beta})]\varepsilon^2(1-\varepsilon)^{mn-2} + A_3[\mathrm{HM}_{n+b_m}]\varepsilon^3(1-\varepsilon)^{mn-3} \\ &\quad + \left|U_4^{(2)}[\mathrm{RS}_n^{(2,1)}(\bar{\beta})]\right| \varepsilon^4 (1-\varepsilon)^{mn-4}. \qquad (5.65) \end{aligned}$$

Next we show bounds on $P_{\mathrm{ue}}(\mathrm{RS}_n^{(3,0)}(\bar{\beta}), \varepsilon)$.

$$P_{\mathrm{ue}}(\mathrm{RS}_n^{(3,0)}(\bar{\beta}), \varepsilon) \leq \{\left|U_4^{(2)}[\mathrm{RS}_n^{(2,1)}(\bar{\beta})] \cap U_4[\mathrm{RS}_n^{(3,0)}(\bar{\beta})]\right|$$

$$+m^2 A_4[\mathrm{HM}_{n+b_m}]\}\varepsilon^4(1-\varepsilon)^{mn-4}$$
$$+\sum_{i=3}^{8}\bar{A}_{2i}[\mathrm{RS}_n^{(2,1)}(\bar{\beta})]\varepsilon^{2i}(1-\varepsilon)^{mn-2i}$$
$$+\sum_{i=9}^{\lfloor\frac{mn}{2}\rfloor}\binom{mn}{2i-1}\frac{\min\{m,n\}}{2i}\varepsilon^{2i}(1-\varepsilon)^{mn-2i}, \qquad (5.66)$$

$$P_{\mathrm{ue}}(\mathrm{RS}_n^{(3,0)}(\bar{\beta}),\varepsilon) \geq \left|U_4^{(2)}[\mathrm{RS}_n^{(2,1)}(\bar{\beta})]\cap U_4[\mathrm{RS}_n^{(3,0)}(\bar{\beta})]\right|\varepsilon^4(1-\varepsilon)^{mn-4}. \qquad (5.67)$$

Note that the bounds proposed in this dissertation are tight for relatively small ε. Bounds which are tight for large ε are presented in [17].

In Figure 2, the bounds given by (5.62) to (5.65) on the probability of an undetectable error, $P_{\mathrm{ue}}(\mathrm{RS}_n^{(2,1)}[g_0(X)](\bar{\beta}^{(P)}))$ are shown where $n = 2^{16}$ and $g_0(X) = X^{32}+X^{26}+X^{23}+X^{22}+X^{16}+X^{12}+X^{11}+X^{10}+X^8+X^7+X^5+X^4+X^2+X+1$. In this figure, UPE1, LPE1, UPE2 and LPE2 correspond to (5.62), (5.63), (5.64) and (5.65), respectively. The code generated by this polynomial is adopted for error detection in IEEE standard 802.3. Since $A_3[\mathrm{HM}_{n+b_m}] = 0$ for $n+b_m = 2^{16}+31$, the values of these two lower bounds are the same. Since the upper bound in [17] is tighter than those given by (5.62) and (5.64) for $\varepsilon \geq 6.7\times 10^{-6}$, the bound in [17] is shown for this range of ε.

In Figure 3, the bounds given by (5.66) and (5.67) on the probability of an undetectable error, $P_{\mathrm{ue}}(\mathrm{RS}_n^{(3,0)}[g_0(X)](\bar{\beta}^{(P)}))$ are shown where $n = 2^{12}$. We have computed $A_4[\mathrm{HM}_{n+b_m}]$ with $n+b_m = 2^{12}+31$ for 250 polynomials which are the minimal polynomials of α^j with $1 \leq j \leq 997$ where α is a root of $g_\alpha(X) = X^{32}+X^{22}+X^2+X+1$. For the codes generated by these polynomials, we have that

$$4137432360 \leq \sum_{i_1=1}^{m}\sum_{i_2=i_1+1}^{m}\binom{\tau(b_{i_2}-b_{i_1})}{2} + m^2 A_4[\mathrm{HM}_{n+b_m}] \leq 4157805864. \qquad (5.68)$$

The values of the upper bound given by (5.66) are very close for these polynomials.

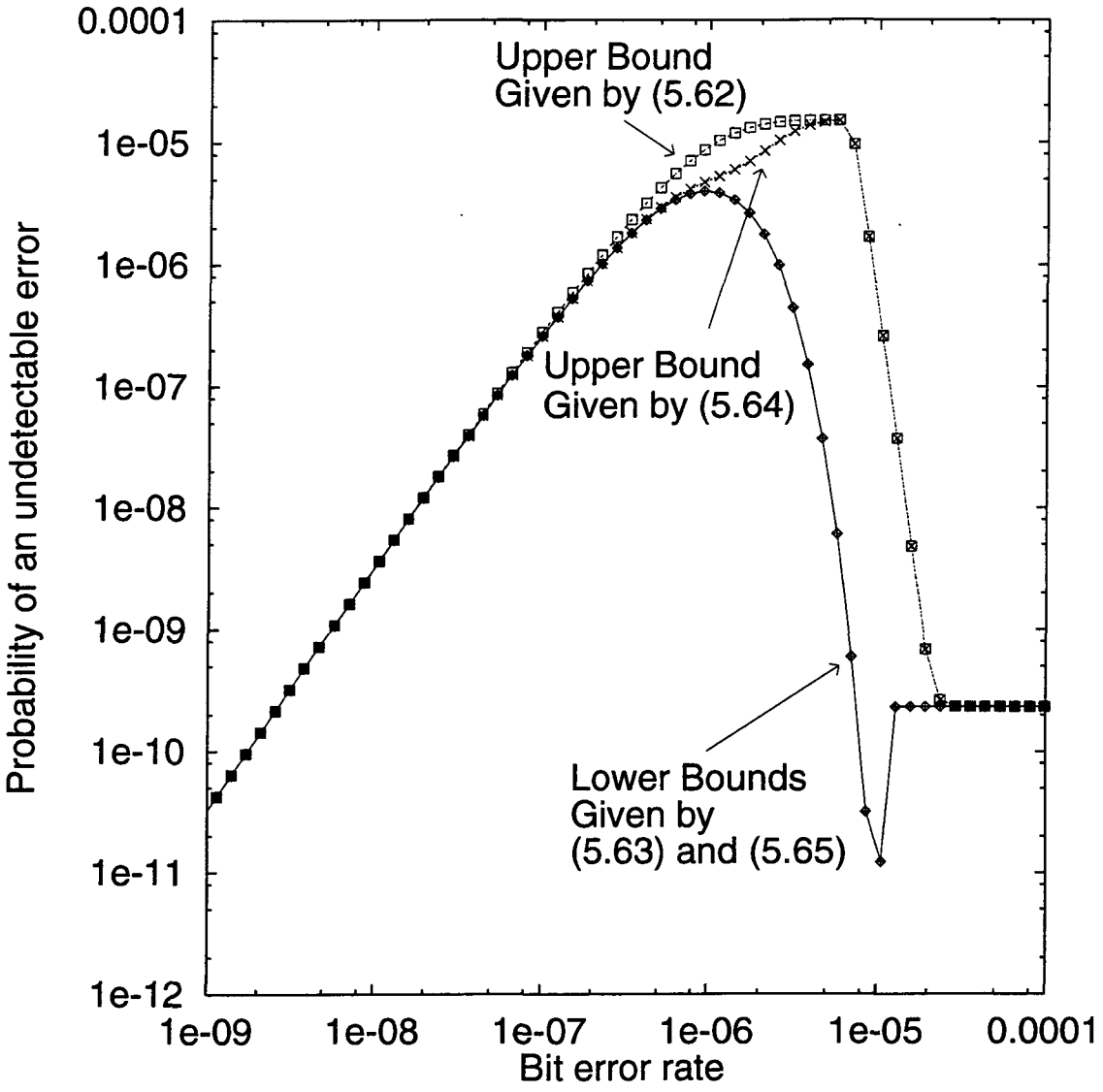

Figure 2: The upper and lower bounds on the probability of an undetectable error for $\mathrm{RS}_{2^{16}}^{(2,1)}[g_0(X)]$ generated by $(X-\alpha)$, where $g_0(X) = X^{32} + X^{26} + X^{23} + X^{22} + X^{16} + X^{12} + X^{11} + X^{10} + X^8 + X^7 + X^5 + X^4 + X^2 + X + 1$.

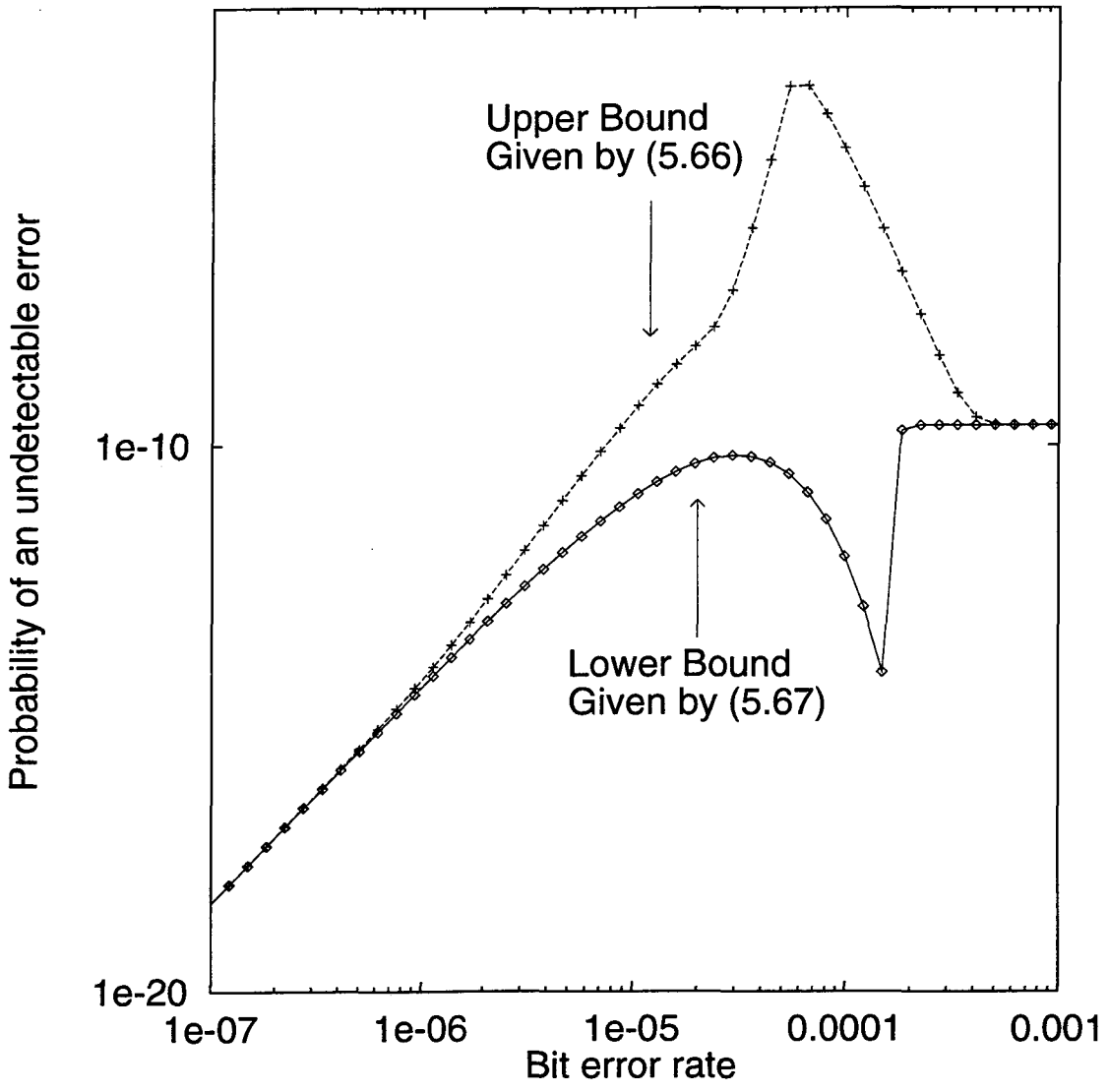

Figure 3: The upper and lower bounds on the probability of an undetectable error for $\mathrm{RS}_{2^{12}}^{(3,0)}[g_0(X)]$ generated by $(X-1)(X-\alpha)$, where $g_0(X) = X^{32}+X^{26}+X^{23}+X^{22}+X^{16}+X^{12}+X^{11}+X^{10}+X^8+X^7+X^5+X^4+X^2+X+1$.

Chapter 6

Conclusions

In this dissertation, computing methods for the weight distributions of linear block codes are proposed.

First, we consider the method for computing the number of codewords of weight less than or equal to a given integer in a linear block code by using its trellis diagram. When we give the code length as the parameter integer, we can compute the weight distribution of the code. We also show that the time and space complexities are given in terms of the dimensions of subcodes and the related codes. Then, we can choose the trellis diagram by which the computational complexity becomes small by using the dimensions which can be computed easily. This method is very efficient for the codes which have relatively simple trellis diagram, say Reed-Muller codes and some BCH codes.

We also present another computing method which uses the invariant property of a code for permutation groups. Two lemmas are derived to reduce the computation of the weight distribution of a given code into that of its subcode. One is for the case when the code is invariant under cyclic permutations, and another is for the case when the code is invariant under affine permutations. For the extended binary primitive BCH codes, we can apply the two lemmas for the computation of the weight distributions. For the extended BCH codes of length 64 and 128 whose weight distributions have not been computed, we computed their weight distributions by using the above two methods simultaneously. From the results, we also computed the probabilities of an undetectable error when the codes are used only for error detection in a binary symmetric channel, and determined whether each code is proper or not.

It may be still infeasible to compute the whole weight distribution for the codes with large parameters with the methods. Also, when shortened codes of various lengths

are used, we need to know the weight distribution for large number of the codes. For such cases, we may compute the some part of weight distribution and to estimate the upper and lower bounds from the results instead of computing the exact probability from the whole weight distribution. In this dissertation, we take the binary code derived from a shortened Reed-Solomon code as a target code. For example, a formula is shown for the exact number of codewords with weight 2 in the binary image of a shortened Reed-Solomon code generated by $(X-\alpha)$. By using the results, we estimate the upper and lower bound on the probability of an undetectable error for the case when $n = 2^{16}$, $m = 32$, the generator polynomial is $(X-\alpha)$, α is the root of the polynomial $g_0(X) = X^{32}+X^{26}+X^{23}+X^{22}+X^{16}+X^{12}+X^{11}+X^{10}+X^8+X^7+X^5+X^4+X^2+X+1$ and the polynomial basis is used. For this case, those bounds are tight when the bit error rate of the channel is less than 10^{-7}.

For the trellis-based computing method proposed in Chapter 2, the computational complexity depends on the structure of the given trellis diagram. In general, the structural complexity varies by permuting the bit positions, while the weight distribution of the code represented by the trellis diagram is invariant. Then, it is important to examine the property of a trellis diagram for a group of bit permutations. This is our future work.

Appendix

(Proof of Theorem 2)

It is shown in [22] that for $\mathbf{u}_1$ and $\mathbf{u}_2$ in $p_{0,n/2}[C]$, $\sigma\mathbf{u}_1 = \sigma\mathbf{u}_2$, if and only if

$$\mathbf{u}_1 + \mathbf{u}_2 \in C^{tr}_{0,n/2}. \tag{6.1}$$

For $\mathbf{v}, \mathbf{v}' \in R_H(\mathbf{u})$, we have that $(\mathbf{v} + \mathbf{v}') \circ \mathbf{0} \in C$, since $\mathbf{v} \circ \mathbf{u}^R$ and $\mathbf{v}' \circ \mathbf{u}^R$ are in C. This implies that

$$\mathbf{v} + \mathbf{v}' \in C^{tr}_{0,n/2}. \tag{6.2}$$

Part (1) of the theorem follows from (6.1) and (6.2).

If $\mathbf{u} \in p_{0,n/2}[C^{\mathrm{SR}}]$, $\mathbf{u} \in R_H(\mathbf{u})$, and therefore $\sigma\mathbf{v} = \sigma\mathbf{u}$ from part (1) of this theorem. It follows from (2.21) and (2.22) that $\mathbf{u} \circ \mathbf{v}^R$ and $\mathbf{v} \circ \mathbf{u}^R$ are in C. Therefore, $(\mathbf{u} \circ \mathbf{v}^R) + (\mathbf{v} \circ \mathbf{u}^R) = (\mathbf{u} + \mathbf{v}) \circ (\mathbf{u} + \mathbf{v})^R \in C^{\mathrm{SR}}$, and hence

$$\mathbf{u} + \mathbf{v} \in p_{0,n/2}[C^{\mathrm{SR}}]. \tag{6.3}$$

The remaining part of the theorem follows from (6.3). $\Delta\Delta$

(Derivation of (2.26))

It follows from the definition of the subset and (6.1) that the number of states in the subset is given by

$$S_B = \frac{\left|p_{0,n/2}[C_{0,n/2} + C_{h,n}]\right| \cdot \left|p_{0,n/2}[C^{SR}]\right|}{\left|p_{0,n/2}[C_{0,n/2} + C_{h,n}] \cap p_{0,n/2}[C^{\mathrm{SR}}]\right| \cdot \left|C^{tr}_{0,n/2}\right|}. \tag{6.4}$$

First, we show that

$$p_{0,n/2}[C_{0,n/2} + C_{h,n}] \cap p_{0,n/2}[C^{\mathrm{SR}}] = C^{tr}_{0,n/2} + p_{0,n/2}[C^{\mathrm{SR}}_{h,n-h}], \tag{6.5}$$

where, for two linear codes A and B, $A + B$ denote the linear code $\{\mathbf{u} + \mathbf{v} : \mathbf{u} \in A, \mathbf{v} \in B\}$.

For any $\mathbf{u} \in p_{0,n/2}[C_{0,n/2} + C_{h,n}] \cap p_{0,n/2}[C^{\mathrm{SR}}]$, there are $\mathbf{u}_1 \in C_{0,n/2}$ and $\mathbf{u}_2 \in C_{h,n}$ such that $\mathbf{u} = p_{0,n/2}(\mathbf{u}_1 + \mathbf{u}_2)$. Define $\mathbf{u}'_1 \triangleq p_{0,n/2}\mathbf{u}_1$ and $\mathbf{u}'_2 \triangleq p_{0,n/2}\mathbf{u}_2$. Since $\mathbf{u} \in p_{0,n/2}[C^{\mathrm{SR}}]$, $(\mathbf{u}'_1 + \mathbf{u}'_2) \circ (\mathbf{u}'_1 + \mathbf{u}'_2)^R \in C$. This and the fact that $\mathbf{u}'_1 \circ \mathbf{0}$ and $(\mathbf{u}'_1 \circ \mathbf{0})^R$ are in C imply $\mathbf{u}'_2 \circ \mathbf{u}'^R_2 \in C$. Therefore, $\mathbf{u}'_2 \in p_{0,n/2}[C^{\mathrm{SR}}_{h,n-h}]$. Since $\mathbf{u}'_1 \in C^{tr}_{0,n/2}$, we have that

$$p_{0,n/2}[C_{0,n/2} + C_{h,n}] \cap p_{0,n/2}[C^{\mathrm{SR}}] \subseteq C^{tr}_{0,n/2} + p_{0,n/2}[C^{\mathrm{SR}}_{h,n-h}]. \tag{6.6}$$

Conversely, consider $\mathbf{v}_1 \in C^{tr}_{0,n/2}$ and $\mathbf{v}_2 \in p_{0,n/2}[C^{\mathrm{SR}}_{h,n-h}]$. Let $\mathbf{0}$ denote the all-zero $n/2$-tuple. Then, $\mathbf{v}_1 + \mathbf{v}_2 \in p_{0,n/2}[C_{0,n/2} + C_{h,n}]$, since $\mathbf{v}_1 \circ \mathbf{0} \in C_{0,n/2}$ and $\mathbf{v}_2 \circ \mathbf{v}_2^R \in C^{\mathrm{SR}}_{h,n-h} \subseteq C_{h,n}$. It follows from (2.22) that $\mathbf{v}_1 \circ \mathbf{0} + (\mathbf{v}_1 \circ \mathbf{0})^R = \mathbf{v}_1 \circ \mathbf{v}_1^R \in C^{\mathrm{SR}}$. Hence $\mathbf{v}_1 = p_{0,n/2}(\mathbf{u} \circ \mathbf{u}^R) \in p_{0,n/2}[C^{\mathrm{SR}}]$. Since $\mathbf{v}_2 \circ \mathbf{v}_2^R \in C^{\mathrm{SR}}$, $\mathbf{v}_2 \in p_{0,n/2}[C^{\mathrm{SR}}]$. Hence, $\mathbf{v}_1 + \mathbf{v}_2 \in p_{0,n/2}[C^{\mathrm{SR}}]$. This implies that

$$p_{0,n/2}[C_{0,n/2} + C_{h,n}] \cap p_{0,n/2}[C^{\mathrm{SR}}] \supseteq C^{tr}_{0,n/2} + p_{0,n/2}[C^{\mathrm{SR}}_{h,n-h}]. \tag{6.7}$$

(6.5) follows from (6.6) and (6.7).

From Theorem 1 and (6.1), we have that

$$\frac{\left|p_{0,n/2}[C_{0,n/2} + C_{h,n}]\right|}{\left|C^{tr}_{0,n/2}\right|} = 2^{K_{n/2} - q_{h,n/2}}. \tag{6.8}$$

It follows from (6.4), (6.5) and (6.8) that

$$\begin{aligned} S_B &= \frac{2^{K_{n/2}-q_{h,n/2}} \left|p_{0,n/2}[C^{SR}]\right|}{\left|C^{tr}_{0,n/2} + p_{0,n/2}[C^{\mathrm{SR}}_{h,n-h}]\right|} \\ &= \frac{2^{K_{n/2}-q_{h,n/2}} \left|p_{0,n/2}[C^{SR}]\right| \cdot \left|p_{0,n/2}[C_{h,n/2}]\right|}{\left|C^{tr}_{0,n/2}\right| \cdot \left|p_{0,n/2}[C^{\mathrm{SR}}_{h,n-h}]\right|} \\ &= \frac{2^{K_{h,n}-2K_{0,n/2}} \left|C^{SR}\right|}{\left|C^{\mathrm{SR}}_{h,n-h}\right|}. \end{aligned} \tag{6.9}$$

$\Delta\Delta$

(Proof of Lemma 1)

For any codeword $\mathbf{u}$ in C, $\mathbf{u} + \mathbf{u}^R \in C^{\mathrm{SR}}$. (2.27) follows from this equation and the assumption of $\mathbf{u}_i$ and $\mathbf{u}_i^R$. Next, we prove (2.28). Suppose that $C_0^{\mathrm{SR}} \subsetneq C^{\mathrm{SR}}$. Since C^{SR} is linear, there is a codeword $\mathbf{u}$ in $C^{\mathrm{SR}} - C_0^{\mathrm{SR}}$ of the form $\sum_{i=k_0^{\mathrm{SR}}+1}^{k} a_i \mathbf{u}_i$ for a binary nonzero tuple $(a_{k_0^{\mathrm{SR}}+1}, a_{k_0^{\mathrm{SR}}+2}, \cdots, a_k)$. Since $\mathbf{u} = \mathbf{u}^R$, (2.29) holds. A contradiction. Finally, we prove (2.30). Define $\mathbf{u} \triangleq \sum_{i=k_0^{\mathrm{SR}}+1}^{k} a_i \mathbf{u}_i$. (2.29) implies that $\mathbf{u} = \mathbf{u}^R$. Since $\{\mathbf{u}_1, \mathbf{u}_2, \cdots, \mathbf{u}_k\}$ are linearly independent, $\mathbf{u} \notin C_0^{\mathrm{SR}}$. $\Delta\Delta$

References

[1] D. Coppersmith and G. Seroussi, "On the minimum distance of some quadratic residue codes," *IEEE Trans. Inform. Theory*, vol. IT-30, no. 2, pp. 407–411, Mar. 1984.

[2] A.M. Barg and I.I. Dumer, "On computing the weight spectrum of cyclic codes," *IEEE Trans. Inform. Theory*, vol. 38, no. 4, pp. 1382–1386, Jul. 1992.

[3] M. Mohri and M. Morii, "On Computing the Number of Codewords with Minimum Weight for Cyclic Codes," *Trans.of IEICE*, vol. J79-A, no. 4, pp. 963–972, Apr. 1996 (in Japanese).

[4] T. Tanigawa, M. Morii and H. Sasano, "On computing the weight spectrum of linear block codes — Improved Cedervall-Johannesson algorithm for searching a code-tree —," *IEICE Technical Report*, IT92-95, Nov. 1992.

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

[6] M. Kataoka, T. Takata, T. Kasami and S. Ujita, "Error Performance of Multi-Stage Hard-decision Bounded Distance Decoding for Multi-Level Block Modulation Codes," *IEICE Trans.*, vol. E74, no. 9, pp. 2555–2562, Sep. 1991.

[7] X. Hou, "$GL(m,2)$ Acting on $R(r,m)/R(r-1,m)$," *Discrete Math.*, vol. 149, pp. 99–122, 1996.

[8] X. Hou, "Classification of $R(3,8)/R(2,8)$," *unpublished.*

[9] S. Lin and D.J. Costello, Jr., *Error Control Coding: Fundamentals and Applications*. Englewood Cliffs, NJ: Prentice Hall, 1983.

[10] T. Kasami, "Weight Distributions of Bose-Chaudhuri-Hocquenghem Codes," *Combinational Math. and its Applications*, Univ. of North Carolina Press, Chapel Hill, NC, 1969.

[11] E.R. Berlekamp, *Algebraic Coding Theory*. New York: McGraw-Hill, 1968.

[12] T.W. Williams, W.Daehn, M. Gruetzner and C.W. Starke, "Bounds and analysis of aliasing errors in linear feedback shift registers," *IEEE Trans. on Computer-Aided Design*, vol. CAD-7, no. 1, pp. 75-83, Jan. 1988.

[13] A. Ivanov and V.K. Agarwal, "An iterative technique for calculating aliasing probability of linear feedback signature registers," *Proc. of the 18th international Symposium on Fault Tolerant Computing*, pp. 70-75, June 1988.

[14] S.K. Gupta and D.K. Pradhan, "A new framework for designing and analyzing BIST techniques: computation of exact aliasing probability," *Proc. of 1988 International Test Conference*, pp. 329-342, 1988.

[15] K. Iwasaki, "Analysis and proposal of signature circuits for LSI testing," *IEEE Trans. on Computer-Aided Design*, vol. CAD-7, no. 1, pp. 84-90, Jan. 1988.

[16] K. Iwasaki, "Design of signature circuits based on weight distributions of error-correcting codes," *Proc. of 1990 International Test Conference*, pp. 779-785, Sept. 1990.

[17] T. Fujiwara, S. Feng and T. Kasami, "An Approximation to Aliasing Probability of Some Signature Analysis Registers," *IEICE*, vol. J73-A, no. 10, pp. 1669-1677, Oct. 1990.

[18] T. Kasami and S. Lin, "On the binary weight distribution of some Reed-Solomon codes," *Proc. of the 7th Symposium on Information Theory and Its Applications*, pp. 49-54, Nov. 1984.

[19] T. Kasami and S. Lin, "The binary weight distribution of the extended $(2^m, 2^m-4)$ code of the Reed-Solomon code over $GF(2^m)$ with generator polynomial $(X-\alpha)(X-\alpha^2)(X-\alpha^3)$," *Linear Algebra and its Applications*, 98, pp. 291-307, 1988.

[20] J. Wolf, "Efficient Maximum Likelihood Decoding of Linear Block Codes Using a Trellis," *IEEE Trans. Inform. Theory*, vol. IT-24, no. 1, pp. 76–80, Jan. 1978.

[21] G.D. Forney, Jr., "Coset Codes—Part II: Binary Lattices and Related Codes," *IEEE Trans. Inform. Theory*, vol. IT-34, no. 4, pp. 1152–1187, Sep. 1988.

[22] T. Kasami, T. Takata, T. Fujiwara and S. Lin, "On Structural Complexity of the L-section Minimal Trellis Diagrams for Binary Linear Block Codes," *IEICE Trans. Fundamentals*, vol. E76-A, no. 9, pp. 1411–1421, Sep. 1993.

[23] R.J. MacEliece, "On periodic sequences from $GF(q)$," *J. Comb. Theory*, vol. 10A, no. 1, pp. 80–91, Jan. 1971.

[24] 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, May 1993.

[25] A. Vardy and Y. Be'ery, "Maximum Likelihood Soft Decision Decoding of BCH Codes," *Proc. of 1993 IEEE International Symposium on Inf. Theory*, p. 29, San Antonio, TX, Jan. 1993.

[26] 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 16th Symposium on Information Theory and Its Applications*, pp. 21–24, Kanazawa, Oct. 1993.

[27] T. Kasami, T. Fujiwara, Y. Desaki and S. Lin, "On Branch Labels of Parallel Components of the L-section Minimal Trellis Diagrams for Binary Linear Block Codes," *IEICE Trans. Fundamentals*, vol. E77-A, no. 6, pp. 1058–1068, Jun. 1994.

[28] W.W. Peterson and E.J. Weldon, Jr., *Error-Correcting Codes*. Cambridge, MA: MIT Press, 1972.

[29] M. Sugino, Y. Ienaga, N. Tokura and T. Kasami, "Weight Distribution of (128, 64) Reed-Muller code," *IEEE Trans. Inform. Theory*, vol. IT-17, no. 4, pp. 627–628, Sep. 1971.

[30] T. Kløve and V.I. Korzhik, *Error-Detecting Codes*. Norwell, MA: Kluwer academic publishers, 1995.

[31] T. Fujiwara, T. Kasami and S. Lin, "Error Detecting Capabilities of the Shortened Hamming Codes Adopted for Error Detection in IEEE Standard 802.3," *IEEE Trans. on Communications*, vol. COM-37, no. 9, pp. 986–989, 1989.

[32] 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.