Capability of Error-Trapping Technique In Decoding Cyclic Codes

Anader Benyamin-Seeyar

A Thesis

in

The Department

of

Electrical Engineering

Presented in Partial Fulfillment of the Requirements
for the Degree of Doctor of Philosophy at
Concordia University
Montréal, Québec, Canada

March 1985

ABSTRACT

Capability of Error-Trapping Technique In Decoding Cyclic Codes

Anader Benyamin-Seeyar, Ph.D.
Concordia University, 1985

This thesis is concerned with an "error-trapping" technique based on the permutation decoding concept for decoding binary cyclic codes. Permutation decoding, whenever applicable, is easily implementable. Here, we investigate the capability of this decoding technique by establishing exact lower bounds on the code length, n, of an (n,k,t) cyclic code in order for it to be permutation decodable. These bounds are obtained by using group (T,U) permutations, where (T) is a group of cyclic shifts and (U) is a sequence of squaring (or square-rooting) operations on the received code word. A code word is said to be one-step (T,U) permutation decodable if all errors can be trapped by using cyclic shifts. It is said to be i-step (T,U) permutation decodable if (i-1) is the maximum number of squaring permutations required to trap all errors. In this work, we present exact lower bounds on n for: 1) Two-step (T,U) permutation decodable binary cyclic codes with t being odd or even valued, 2) Three-step (T,U) permutation decodable binary cyclic codes with t odd-valued and t=2, and 3) some results for the (n,k,t) cyclic codes which, in general, are either not permutation decodable or can not be decoded with a certain number of steps of permutation are also obtained. The derivation of these results involves only the symbol positions of the errors and, consequently, the results are directly applicable to cyclic codes over GF(2^m), for m>1.

ACKNOWLEDGEMENTS

The author wishes to express his deep gratitude to his supervisor Professor V.K. Bhargava for his invaluable advice and continuous encouragement received throughout this research.

The author wishes to express his indebtedness to:

Professor S.G.S. Shiva, who made himself easily available for discussions and provided valuable suggestions and constructive criticism during the preparation of this thesis.

Professor S.V. Kanetkar for his valuable assistance and fruitful discussions during the early stages of this work.

Thanks are in order to my friends and colleagues in the Electrical Engineering Department at Concordia University.

Thanks are due to Ms. Marie Berryman for her excellent typing.

Finally, I must thank my wife Concepcion. Without her encouragement and support, none of this would have been possible.

This research was supported in part by the Natural Sciences Research Council of Canada.

TABLE OF CONTENTS

Page

LIST OF FIGURES

LIST OF TABLES

CHAPTER I

INTRODUCTION TO CODING

1.1 Introduction

In recent years the subject of "error-control coding" has become important in communications and computer technology. The overall objective is that of protecting digital data transmission and storage of digital information against the errors that occur during transmission through a communication channel. The possibility of achieving reliable transmission over a noisy channel was originally introduced by Shannon in 1948 [1]. The most remarkable results proved by Shannon [2] was to demonstrate the existence of codes that achieve reliable communication if and only if the information transmission rate is less than a quantity called the "channel capacity". However, his results were only "existence" type.

Since the publication of these results, a great deal of effort has been devoted to finding efficient and practical encoding and decoding schemes that permit reliable communication at high data rates over various types of noisy channels as promised by Shannon's theory. This in fact has recently been emphasized by J. Wolf [3] where he states: "Coding is playing a more and more important role in helping us to achieve the promises of Shannon and Wiener".

This chapter introduces the coding problems, describes the role of coding in communication, and defines the types of codes along with some important fundamental concepts of coding theory. In Sections 1.4 and 1.5 we give a description of cyclic codes and some important classes of cyclic codes. Findally, in Sections 1.6 and 1.7, we describe the topic of our investigation and the plan of the thesis..

1.2 The Coding Problem

In a communication system as shown in Figure 1-1., the symbol that comes out of the decoder should match

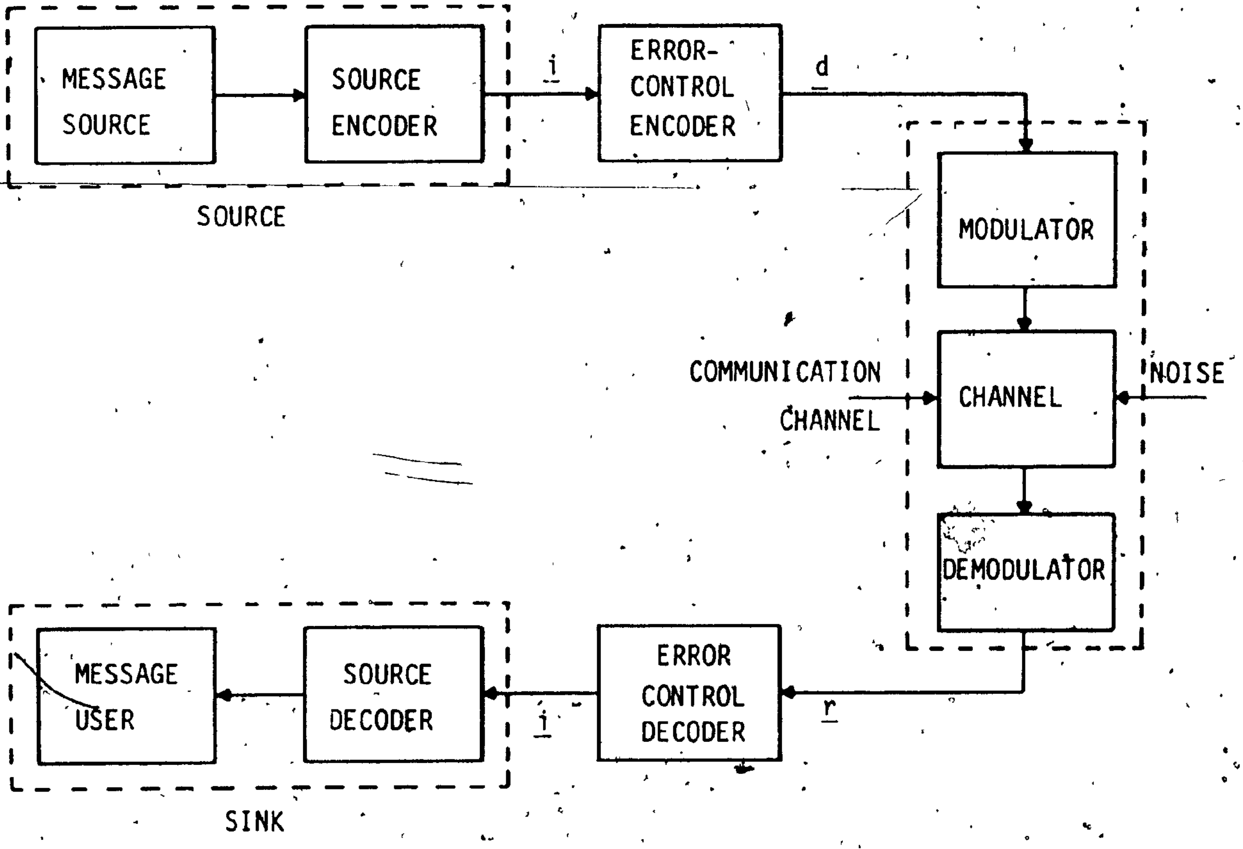


Figure 1-1. A communication system

the symbol that entered the encoder [4]. But in a practical system messages are first encoded into signals before transmission. These signals enter the channel and are likely to be disturbed by noise.

On memoryless channels, the noise affects each transmitted signal independently, while on channels with memory, the noise is not independent from transmission to transmission. As a consequence, transmission errors tend to occur in clusters or bursts. Hence, transmission errors occuring on memoryless channels are called "random-errors"; and on channels with memory are called "burst-errors". Some channels contain a combination of both random and burst errors and are called compound channels. Here, throughout the thesis we will be considering the memoryless channel with random errors only [5,6].

The channel output then enters the decoder which makes a decision concerning which message was sent and then delivers this message to destination. The coding problem has two aspects: one, the performance of the code, i.e. the associated probability of error and two, the complexity of the decoder. Ideally we would like to find good codes which possess a decoding algorithm of moderate complexity [4-12].

There are two basic techniques for controlling transmission errors in data communication systems; forward-error-control (FEC) schemes, and automatic-repeat-request (ARQ) schemes. Hybrid FEC/ARQ schemes have also been proposed [13,14].

In an FEC communication system, an error-correcting code is used for combating transmission errors. The receiver detects the presence of errors then automatically <u>corrects</u> them. On the other hand, in an ARQ communication system, when the receiver detects the presence of errors, it automatically requests for retransmission of the message [13,15]. In this thesis we are only concerned with FEC coding techniques.

There are two fundamentally different classes of FEC codes; namely "block codes" and "tree codes". The encoder for a block code is a memoryless device which breaks the continuous sequence of information symbols into k-symbol blocks. It then maps each k-symbol block into an n-symbol block, called a "code word", where n is greater than k. If the code is linear, then k is referred to as the dimension of the code, and n as its length. The rate of the code, R is defined by

$$R = k/n, \tag{1.1}$$

In the case of a convolutional encoder, the output n_o-tuple is a function of the present k_o-tuple input and the previous ν k_o-tuples. Hence, the encoder has a memory order of ν. The rate, R, of the tree code is defined as,

$$R = k_o/n_o \quad , \tag{1.2}$$

The most important tree codes are those known as convolutional codes. Convolutional codes are tree codes that satisfy certain additional linearity and time-invariance properties (linear tree codes).

For the purpose of this thesis, we restrict our attention to block codes. The study of block codes has led to codes which have considerably mathematical structure. In many cases this mathematical structure can be exploited to arrive at practical decoding algorithms. The readers interested in tree codes may find References [6,7,10,15] useful.

A block code of length n and size M is a collection of M distinct vectors, the code words, each vector having n elements belonging to some finite field $GF(q)$, where q is a prime power [16-19]. The simplest and most widely used finite field is the binary field, $q=2$, consisting of the digits 0 and 1. Such codes are called binary codes*. The code rate R, is defined as

$$R = \log_q M / n, \tag{1.3}$$

since the code words are distinct, then $1 \leq M \leq q^n$, and so $0 \leq R \leq 1$.

The block codes of practical importance, introduced by Slepian [19,20] in 1956, are called "group codes". These codes are a generalization of the error-correcting codes of Hamming [21]. The code words of group codes were shown to correspond to the elements of a suitably defined group [17,18]. Further, Elias [22] has shown that on the binary symmetric channel, there exist group codes having a rate arbitrarily close to channel capacity with an arbitrarily small probability of error. However, like Shannon, Elais only gives an existance proof.

* It is relatively easy to extend most of the results of binary codes to the nonbinary $(q > 2)$ case.

A subclass of group codes are called linear coes. A code is linear if the code words are all the solutions to a set of r homogeneous linear equations, called "generalized parity-check" equations. The coefficients of these equations are elements from $F = GF(q)$. Let $k = n-r$ then if the equations are linearly independent [15], $M = q^k$, $R = k/n$, and the code is termed an (n,k) "linear" code.

Another class of block codes discovered by Nordstrom and Robinson [23], which are not linear and are said to be "nonlinear" block codes. It is interesting to note that in several cases it is possible to construct a non-linear block code with more code words than the best linear code with the same length and minimum distance [12,24]. A generalization by Preparata yielded a class of high-rate nonlinear binary codes [25].

Whenever the information symbols and the parity symbols can be separated as in Figure 1.2, we say that the encoding is systematic. Hereafter we will be considering only (block)

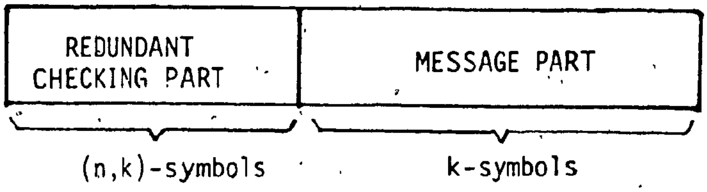


Figure 1-2. Systematic format of a block code word

linear codes.

1.3 Notations and Definitions

Consider a q-ary linear block code of length n, i.e., a code whose symbols come from the Galois field $F=GF(q)$ with q distinct symbols, q being a prime power ($q=p^{\nu}$, p prime). Each code word represents an n-tuple and hence a vector with n elements over GF(q). Using the symbols of F one forms all n-tuples, i.e., F^n, and call these n-tuples words and n the word length. We shall denote the set of all words by F^n and interpret this as a n-dimensional vector space over GF(q). A linear block code C is defined as a subspace of F^n. Such a code is called an (n,k) linear block code.

An important concept in this study of block codes is the distance function called "Hamming distance" [21] between two code words, which by definition is the number of coordinate places in which they differ. Then, the minimum Hamming distance of the code C, denoted by d, is the minimum pair-wise Hamming distance between any two code words. A related concept is the "Hamming weight" of a code word which is defined as the number of non-zero components in the code word. For any group code, it is easily shown that the minimum distance is equal to the minimum weight of the non-zero code words (any two code words differ in at least d places) [25,26]. This property does not necessarily hold for non-group codes.

The error-correcting and detecting capability of a given code is determined by the minimum distance (or simply distance). A block code with distance d is capable of correcting all patterns of t or fewer errors if and only if the distance d is at least $2t+1$ (or $t = \lfloor (d-1)/2 \rfloor$*). Similarly, it is possible to detect all patterns of $t+\lambda$ or fewer errors ($\lambda \geq t$) if $2t+\lambda<d$ [11,15].

We shall now examine two ways of describing a linear block code:

The first is given by a $k \times n$ generator matrix $\underline{G}$ which has as its rows a set of basis vectors of the linear subspace C. The row space of $\underline{G}$ is the linear code C; any code word is a linear combination of the rows of $\underline{G}$. The rows of $\underline{G}$ are linearily independent. Any one-to-one pairing of k-tuples and code words can be used as an encoding procedure, but the most natural approach is to use the following:

$$\underline{c} = \underline{i} \cdot \underline{G} \tag{1.4}$$

where $\underline{i}$, the information word, is a k-tuple of information symbols to be encoded and $\underline{c}$ is the corresponding code word. We now come to the second description of a linear code C.

Let C be an (n,k) linear code with generator matrix $\underline{G}$. C being a subspace of F^n, it has an orthogonal complement (with respect

* $\lfloor x \rfloor$ denotes the greatest-integer not greater than x.

to the usual inner product) $C^{\perp}$ which is a subspace of F^n of dimension $n-k$. Naturally $C^{\perp}$ is also a linear code called the dual of C. Let $\underline{H}$ be a generator matrix for $C^{\perp}$, the $\underline{H}$ is a full rank $(n-k)xn$ matrix and C consists of all n-tuples $\underline{C}$ such that

$$\underline{C} \cdot \underline{H}^T = \underline{0} , \tag{1.5}$$

where $\underline{H}^T$ is the transpose of $\underline{H}$. We say that C is the null-space of $\underline{H}$ and a vector for $C^{\perp}$ is called a parity check equation for the code C. $\underline{H}$ is called a parity check matrix for C. A $(n-k)xn$ matrix $\underline{H}$ is a parity check matrix for C if and only if $\underline{H}$ has full rank $n-k$ and

$$\underline{G} \cdot \underline{H}^T = \underline{0} \tag{1.6}$$

Two linear codes in F^n are called equivalent if there is a permutation of coordinates followed by a scalar multiplication on coordinates, which transforms one code into the other. That is, if $(c_1, \ldots, c_n)$ is the sequence of coordinate functions of one code and $(c_1', \ldots, c_n')$ that for the other, then the codes are equivalent provided there is a permutation π of $\{1, \ldots, n\}$ and non-zero scalars $\alpha_1, \ldots, \alpha_n$ from the ground field $GF(q)$, such that

$$\alpha_i \cdot c_{\pi(1)} = c_1' , \ldots , \alpha_n \cdot c_{\pi(n)} = c_n' , \tag{1.7}$$

Implicit in Equation (1.7) is that the codes have the same dimension and minimum distances [28].

The permutations of coordinate places which send C into itself (code words go into (possible different) code words) form the "automorphism group" of C, denoted by Aut (C). Auto (C) is a subgroup of the symmetric group S_n consisting of all $n!$ permutations of n symbols.

The most important class of linear block codes is the class of cyclic codes introduced by Prange [29] in 1957. A cyclic code is a linear code with the additional property that any cyclic permutation of the symbols of a code word is also a code word. Cyclic codes can be efficiently encoded by means of simple feedback shift registers [30]. Additional properties of cyclic codes are given in the following section.

1.4 Cyclic Codes

It is perhaps a remarkable fact that most of the important block codes found to date can be reformulated to be cyclic codes or closely related to cyclic codes (for example certain nonlinear codes [25]). Cyclic codes are adequately described in the literature [4-12] and their description here will be brief. An (n,k) linear code C over $GF(q)$ is called cyclic if every cyclic permutation of the symbols of a code word is also a code word. To illustrate, if the n-tuple $(c_0,c_1, \ldots, c_{n-1})$ represents a code word, then the n-tuple

$\underline{c}' = (c_{n-1}, c_0, c_1, \ldots, c_{n-2})$ obtained by shifting the coordinates of $\underline{c}$ cyclically one place to the right is also a code word. If the c_i's belong to a finite field $F=GF(q)$ of q elements, an (n,k) cyclic code has q^k code words*.

In treating cyclic codes, it is convenient to identify a code word with a code word polynomial. The code word polynomial associated with the code word $\underline{c}$ is

$$\underline{c} = (c_0, c_1, \ldots, c_{n-1}) \longleftrightarrow$$

$$C(x) = c_0 + c_1 \cdot x + c_2 \cdot x^2 + \ldots + c_{n-1} \cdot x^{n-1}, \qquad (1.8)$$

where the $c_i \in F$. Thus, each code word corresponds to a polynomial of degree $n-1$ or less. A cyclic shift of this code word $\underline{c}$ is equivalent to multiplication by x and reducing exponents modulo n(mod n). However, reducing exponents mod n is equivalent to reducing the polynomial mod (x^n-1). From now on we do not distinguish between <u>words</u> of length n and <u>polynomials</u> of degree $<n$ [9,11,15,31]. We note that if

$$\underline{c} \longleftrightarrow C(x) \quad \text{and} \quad \underline{y} \longleftrightarrow Y(x) \qquad (1.9)$$

then

$$\alpha_1 \cdot \underline{c} + \alpha_2 \cdot \underline{y} \longleftrightarrow \alpha_1 \cdot C(x) + \alpha_2 \cdot Y(x),$$

* This statement is also true of any (n,k) linear code whose elements come from a finite field of q elements.

where α_1 and $\alpha_2 \epsilon GF(q)$, and that a vector space of n-tuples is also a vector space of polynomials, and conversely.

We next consider the important structural properties of (n,k) cyclic codes.

It can be shown that a cyclic code C is an ideal in the ring, $F[x]/(x^n-1)$, of polynomials mod (x^n-1) over GF(q) [18,32]. Every ideal is generated by a polynomial, g(x), which divides (x^n-1). We shall call g(x) the generator polynomial of the cyclic code. For an (n,k) cyclic code C over GF(q), g(x) is the unique monic* ploynomial in C which has degree (n-k). Any cyclic code word $C(x) \epsilon C$ can be written uniquely as [15,18].

$$C(x) = I(x) \cdot g(x)$$

$$= (i_0 + i_1 \cdot x + \ldots + i_{k-1} \cdot x^{k-1}) \cdot g(x) \qquad (1.10)$$

where the polynomial I(x) corresponds to the k information symbols $(i, i_1, \ldots, i_{k-1}) \epsilon GF(q)$. In addition, for a given $g(x) \neq x-1$, we will always take the block length n of the corresponding cyclic code to be the least integer such that g(x) divides (x^n-1) [18]**. That is $g(x) | (x^n-1)$ and $g(x) \not| (x^{\nu}-1)$, $0<\nu<n$. The quotient $h(x)=(x^n-1)/g(x)$

* A polynomial is called monic if the coefficient of the highest power of x is 1.

** A necessary and sufficient condition that (x^n-1) not to have any factors of multiplicity greater than 1 over GF(q), is that (n,q)=1.

is called the check polynomial (recursion polynomial) of the cyclic code C. We note that the dimension of the cyclic code is equal to the degree of the check polynomial. The cyclic code generated by $h(x)$ is the dual of C, and $C^{\perp}$ is obtained by reversing the order of the coordinates of the words in the cyclic code generated by $h(x)$ [15]. Many properties of the code C can be given from its dual. For example, the weight distribution (i.e., the number of code words of weight i, A_i, for $i=0,1,\ldots,n$) of the code C can be calculated from that of its dual [8].

1.5 Some Important Classes of Cyclic Codes

Here, we briefly introduce the characteristics of some important classes of cyclic codes or codes related to cyclic codes. Additional properties of these codes as well as the relationships with other classes of codes may be found in the following papers on classical algebraic coding theory [4,8,9,10,15,18].

The most important class of linear block codes are the BCH (Bose-Chaudhuri-Hocquenghem) codes [33,34]. These codes were introduced by Bose and Chaudhuri and independently by Hocquenghem [35]. They became prominient when Peterson [18] showed that they are cyclic and introduced a decoding algorithm for them. These codes are cyclic with coefficients from any finite field $F=GF(q)$ and have the following parameters:

$$n = (q^s-1)/c, \quad \text{for} \quad s > 2$$

$$n-k \leq 2\cdot s\cdot t, \qquad (1.11)$$

$$d \geq 2\cdot t+1,$$

where t is any integer and c is any divisor of (q^s-1).

An important and popular subclass of the class of BCH codes is the class of Reed-Solomon codes [8,18,31]. These are BCH codes in which the block length divides the multiplicative order of the finite field GF(q), that is, $s=c=1$. These nonbinary codes defined over GF(q) have the parameters:

$$\begin{aligned} &n=q-1 && \text{symbols} \\ &n-k=2\cdot t && \text{symbols,} \\ &d=2\cdot t+1 && \text{symbols.} \end{aligned} \tag{1.12}$$

Since $d=n-k+1$ for Reed-Solomon codes, these codes are called "maximum distance separable". Typically, $q=2^m$ is selected. This means that the code provides correction of 2^m-ary symbols and hence multiple-burst errors. Moreover, these codes are very useful in two-level "concatenated" coding schemes [36].

The class of "quadratic residue" (QR) codes is a special subclass of the class of cyclic codes whose minimum distances typically compare to those of BCH codes of comparable lengths. The QR codes have the following parameter:

$$\begin{aligned} &n=p, \text{ a prime, of the form } 8\mu\pm1, \\ &k = (p+1)/2. \\ &d > \sqrt{n}. \end{aligned} \tag{1.13}$$

This inequality can be strengthened to : $d^2-d+1 \geq n$, for $n\equiv-1 \bmod 4$ [8,18,26].

Some other important codes that have received special attention are the famous Hamming and Golay codes. These codes as well as Reed-Muller codes are equivalent to cyclic codes [8,19].

The binary Hamming codes is the first class of linear codes devised for error-correction [21]. For any positive integer $m>3$, there exists a Hamming code with the following parameters:

$$\begin{aligned} &n=2^m-1, \\ &k=2^m-m-1, \\ &n-k=m, \\ &d=3 \quad (t=1). \end{aligned} \qquad (1.14)$$

Hamming codes correct all single error patterns and no others. This is a very interesting structure that makes the code a "perfect" code* [8]. Thus, Hamming codes form a class of single-error correcting perfect codes.

The general Hamming codes $(n=(q^m-1)/(q-1), k=n-m)$ with minimum distance $d=3$ codes over $GF(q)$ are perfect single-error-correcting codes. They are also cyclic if m and $(q-1)$ are prime [11].

* A t-error-correcting code over $GF(q)$ is called perfect if every code vector is at a distance of at most t from a code word.

Besides the Hamming codes, the only other non trivial binary perfect code is the (23,12) Golay code with minimum distance d=7. This code is the only known multiple-error-correcting binary perfect code which is capable of correcting any combination of 3 or fewer random errors in a block of 23 bits. Based on the following number theoretic fact:

$$[\binom{23}{0} + \binom{23}{1} + \binom{23}{2} + \binom{23}{3}] = 2^{11} , \qquad (1.15)$$

the code is a perfect code (perfect codes are rare [8]). The extended (24,12) Golay code with minimum distance d = 8 has been widely used.

Hamming and Golay codes were discovered prior to BCH codes, but later were shown to be equivalent to BCH codes [18].

The above list is by no means exhaustive. For example, we have not mentioned codes based on the combinatorial configurations of finite geometries, Goppa codes [37-39], Quasi-Cyclic codes [40-43] to name a few.

For the sake of brevity, from now on we shall refer to a t-error-correcting (n,k) cyclic code, as an (n,k,t) cyclic code.

1.6 Advances in This Work

In this thesis we are concerned with the decoding of cyclic codes. The most effective and easily implementable decoding technique is the error-trapping technique. Here, we investigate the capability of error-trapping techniques, specially, based on the "permutation decoding" concept. To appreciate the simplicity of this form of decoder see for example Figure 2-1 on page 34 of this thesis.

Permutation decoding can be implemented by a simple logic circuit. Since it has been used only with cyclic shift permutations, this technique was in the past only applicable to low rate codes. In this thesis we show that by applying permutations other than cyclic shifting this decoder can be used for higher rate codes. More specifically we shall obtain exact lower bounds on the block length n in order for an (n,k,t) cyclic code to be permutation decodable using cyclic shifting and squaring (all cyclic codes are invariant under these type of permutations).

Finally, we present some general results for cyclic codes which are not permutation decodable using cyclic shifting and squaring.

Since the derivation of the results involves only the positions of the errors, the results are applicable not only to binary cyclic codes over $GF(2)$, but also to cyclic codes over $GF(2^m)$, for $m>1$.

1.7 Plan of the Thesis

This thesis is divided into six chapters and a very brief description of these chapters now follows:

In the previous sections we introduced the coding problem as well as the important role of codes in a communication system. We then gave a brief review of techniques for controlling transmission errors. Some notations and definitions useful for this work were given in Section 1.3. Finally, we defined cyclic codes and some important classes of cyclic codes.

Since our main thrust is on decoding, we dedicate Chapter 2 to a review of decoding procedures available. We begin by defining the decoding procedure for cyclic codes. Then, we introduce the error-trapping technique as a variation of the Meggitt decoder for cyclic codes. In view of this technique, we describe some important and related decoding schemes based on the concept of permutation decoding for cyclic codes. Finally, from the implementation point of view we give a practical permutation decoder in Section 2.5.

In Chapter 3, we investigate the capability of permutation decoding by applying 2-step permutations to cyclic codes. In this direction, we establish exact lower bounds on the code length n; for the (n,k,t) cyclic codes with t being odd or even valued in order for the code to be two step permutation decodable. In Section 3.5 we present a summary of the results obtained in Chapter 3 and numerical results derived there from.

In Chapter 4, we present the exact lower bounds on n, for the (n,k,t) cyclic codes with odd-valued t and to the case of even $t=2$, which are 3-step permutation decodable. The improvements on the exact lower bounds with respect to the results given in Chapter 3, are summarized in Section 4.4 and some numerical results based on the results of this Chapter are presented.

Chapter 5 contains some general results based on certain cyclic codes which are not permutation decodable using cyclic shifting and squaring. The results that we obtain in this chapter are summarized in Section 5.3.

Finally, in Chapter 6, we summarize the results obtained in this thesis and we make some remarks on the results and conclude the thesis with some suggestions for further work.

The main contributions of the author are in Chapter 3 to 5. In these chapters we investigate the capability of permutation decoding for various steps of permutations that are introduced in Chapter 2.

In passing it may be noted that some of the results given in this thesis have been presented or will be published elsewhere [44-46].

CHAPTER 2

DECODING TECHNIQUES FOR CYCLIC CODES

2.1 Introduction

Encoding of cyclic codes was described in Chapter 1. Since our main concern is with decoding, therefore in this Chapter we review decoding procedures available for error-trapping technique and introduce the concept of permutation decodable cyclic codes.

Section 2.2 is on the general decoding procedures, while Section 2.3 describes some of the important and related decoding techniques based on the concept of error-trapping. In Section 2.4 we present a comprehensive description of permutation decodable cyclic codes. Finally, in Section 2.5 we present a permutation decoder which can be realized as a simple logic circuit.

2.2 Decoding Procedure

Let $C(x)$ be the transmitted codeword, $E(x)$ be the channel noise error pattern (EP), and $R(x)$ be the received code word where:

$$C(x) = \sum_{i=0}^{n-1} c_i x^i ,$$

$$E(x) = \sum_{i=0}^{n-1} e_i x^i , \qquad (2.1)$$

$$R(x) = \sum_{i=0}^{n-1} r_i x^i,$$

and where c_i, e_i and r_i are elements of $GF(q^s)$, $i=0,1,\ldots,n-1$. The received word $R(x)$, at the decoder is:

$$R(x) = C(x) + E(x) . \tag{2.2}$$

A decoder must process the received word so as to remove the error word $E(x)$, the information is then recovered from $C(x)$.

The basic step of the decoding procedure consists in computing the "syndrome", $S(x)$, according to the following operation. From the received Work defined in Equation (2.2) we get

$$\underline{R} \cdot \underline{H}^T = \underline{E} \cdot \underline{H}^T . \tag{2.3}$$

where the product between $\underline{H}^T$ and any code word $C(x)$ will give an "all-zero" vector of dimension $n-k$. From Equation (2.3) the product $\underline{S} \underline{\Delta} \underline{R} \cdot \underline{H}^T$ is called the error syndrome and is independent of the transmitted code word $C(x)$. The decoder must then produce an estimate $\hat{E}(x)$ of the $E(x)$, in order to obtain an estimate $\hat{I}(x)$ of the information polynomial.

In the sequel we shall describe the decoding schemes which are pertinent to our research problem.

2.3 Some Decoding Schemes for Cyclic Codes

In the literature, based on the algebraic structure of cyclic codes, various decoding schemes have been proposed; see for example References [8,9,10,15,18]. In general, there are two major categories of decoding techniques for the correction of errors of weight t or less. The decoding techniques in the first category are all based on certain mathematical structures which are designed into the codes and basically involve solving sets of equations to determine the location and values of the errors. The decoding schemes of the second category, while accomplishing the same goal, are based on simple structural aspects of cyclic codes. They determine the correctable error patterns of the received word in a more direct fashion.

The most prominent among the first category of decoding techniques is the iteration decoding algorithm for BCH codes [9,31,33,40]. Decoding a received BCH code word requires execution of three successive computational processes with all computations performed over the field GF(q^s). These processes are syndrome calculation, solution of the "key equation", and Chien search [9,11,18]. The complexity, execution time, and cost of this decoder and most of the first category decoders, are in general much higher than those of the second category decoders [15, 18,32].

In this thesis, we are more concerned with the second category of decoding schemes. As will be seen, they are conceptually simple and quite easy to implement.

A typical decoder from the second category for any (n,k,t) cyclic code of block length n, number of information places k, and error-correcting capability t, is the Meggitt decoder [47], for the correction of burst and random errors. From the implementation aspect, the Meggitt decoder is limited to codes for which n-k and t are small.

A practical variation of Meggitt decoding is called "Error-Trapping" decoding [15,48]. A decoder based on this technique for an (n,k,t) cyclic code can correct all error vectors $\underline{e}$ of weight t, or less, which contain a string of at least k consecutive zeros. In such a decoder, if the weight of the sybdrome for cyclic code is at most t, then the information symbols (positions) are correct (i.e., the errors are "trapped" in the parity check region). Moreover, if the weight of the syndrome is greater than t, then at least one information symbol is incorrect [49]. This decoding technique is most effective and it can be very simple, economical and quite easy to implement, whenever applicable to the correction of random and burst errors.

Since 1962, several refinements and generalizations of this simple decoding technique have been devised in an effort to extend the capability and effectiveness of the decoder to multiple-error-correcting cyclic codes [29,49,50].

In this section we briefly present some of the most important schemes which are of interest.

In 1962, Prange [29], introduced a class of algorithms for decoding cyclic codes based on the use of information sets as a variation of the error-trapping technique. In an (n,k,t) code, an information set is defined to be any set of k positions in the code wode in which the values can be specified independently. The remaining n-k positions are referred to as the parity set whose elements are linear combinations of the contents of the information set.

Information set decoding algorithms are based on the fact, that if there are no errors in the information set positions then the remaining symbols in the transmitted code word can be reconstructed. In accordance with this property, the information set decoding procedure may be described in three steps: first, select a collection of information sets; second, re-encode each of these sets assuming that the symbols in the information sets are free of errors; and third, compare the resulting code words to the received sequence for final corrections.

For the first step, there already exist several methods for selecting an appropriate collection of information sets. However, all these methods are ad hoc in nature. It is supposed that in the collection at least one information set exists, where information set positions are free of errors. As for the second step, by using this set, produce a code word. As for the third step, the hypothesized error pattern is obtained by comparing the resulting code word with the received sequence. This latter pattern should be zero in the information set positions and thus, should contain nonzero terms only in the remaining, or parity set, positions. This implies that the portion of the error pattern contained in the parity set is, in fact, identical to the syndrome [10]. Such a parity set is said to "cover" the error pattern, and a collection of all the parity sets which cover all the error patterns of a particular code is said to form a "covering".

In constructing a candidate code word, one can either re-encode the selected information set using the appropriate generator matrix $\underline{G}$, or one can simply compute a syndrome using the appropriate parity check matrix $\underline{H}$, to determine the error pattern [10,29,49].

A notable decoder similar to the information set error-trapping technique has been given by Kasami [50]. He has shown that all the error patterns can be corrected by choosing a set of polynomials $[Q_j(x)]_{j=1}^{N}$ of degree k-1 or less, which are called "covering polynomials". For any correctable error vector $\underline{e}$ of weight t or less, there is one polynomial $Q_j(x)$ in this set such that some cyclic permutation of $\underline{e}$ agrees with $Q_j(x)$ in k information positions. Recently, Kasami's decoder has been further modified with the view of applying it to non-binary cyclic codes [51].

Although these decoding techniques are rather simple in principle, they may require the use of a large number of distinct information sets or covering patterns in order to correct the dominant error patterns. This number, for codes with large code length n, and t, can be prohibitively large. At the same time finding the covering patterns for a specific code is not an easy problem [10].

Another effective decoding scheme belonging to the second category for certain classes of cyclic codes is the "majority-logic" decoding [4, 11,13,15]. Majority-logic decoding is the simplest form of "threshold decoding" [5,13,40]. For cyclic codes, the majority-logic decoders can always be implemented as Meggitt decoders, a simple logic circuitry that operates on a syndrome to produce a likely estimate of some selected error digits. This decoding technique is applicable to both block and convolutional codes. Unfortunately, only a small class of codes can be decoded in this way, which have the necessary structural characteristics [10,11].

Here, we are going to consider in detail a decoding technique known as "permutation decoding" which is essentially an error-trapping technique introduced by Prange [29]. A serial decoder based on his treatment was given by MacWilliams [49], who made use of code-preserving permutation sets to obtain k error-free positions from which the rest of the codeword could be reconstructed. Permutation decoding is best suited to codes which are invariant under a large group of permutations [9,18,47,49].

For every cyclic code C in the vector space F^n of dimension n, with symbols from the finite field $F=GF(q)$, where q is the size of the field, there are various code preserving permutations. In this thesis we shall apply the following group (T,U) permutations:

1) The group (T) permutations:

$$T^{\beta} : \omega \rightarrow \omega + \beta \quad (\text{addition mod } n) \tag{2.4}$$

where $\beta = 0,1,2,\ldots,n-1$, ω stands for a symbol position* of the code, and

$$T^n = I \text{ (identity)} : \omega \rightarrow \omega + n = \omega.$$

Thus, a cyclic code in F^n is a code which is invariant under group (T) permutations. These permutations correspond to a mapping of the form:

$$x^{\omega} \rightarrow x^{\omega+\beta} \ (\text{mod } x^n-1). \tag{2.5}$$

* We identify the symbol positions by the numbers $0,1,\ldots,n-1$. This notation is convenient for the description of the permutations.

2) The group (U) permutations:

$$U^i : \omega \rightarrow q^i \cdot \omega \text{ (multiplication mod } n). \qquad (2.6)$$

The group (U) permutations correspond to the mapping:

$$x^j \rightarrow (x^j)^{q^i} \quad (\text{mod } x^n - 1)$$
$$\text{for } j = 0,1,\ldots,n-1 \qquad (2.7)$$

If q is relatively prime to n, the length of the code C, then the mapping in Equation (2.7) is an automorphism of $F(x)$ (i.e., the ring of polynomials in x over $GF(q)$) and every ideal in $F(x)$ is preserved by the mapping in Equation (2.7). Thus, if we raise any codeword to the power q, then we get another code word of the same weight, and the code is invariant under group (U) permutations.

From now on throughout the thesis, we shall assume that the cyclic codes are binary, i.e. that q=2 (This analysis can be extended to non binary cyclic codes with $q>2$ [29]). Therefore, n is an odd number (in order to be prime to $q=2$). In Equation (2.6), for n odd, there exists a least integer ν such that $2^\nu=1$ mod n, and so we can restrict to be between 0 and $\nu-1$ inclusively.

In fact, the group (T) is generated by the cyclic shift permutation T whose order is n. The group (U) is generated by the squaring (the group (U) permutations contain the square-rooting permutations which corresponds to the permutations with $i \rightarrow -i$ in Equation (2.6) or (2.7)) permutation U whose order is ν.

It is easy to show that:

$$T^2 \cdot U = U \cdot T \quad , \tag{2.8}$$

hence, we may represent every permutation in the group (T,U) in the form $U^i \cdot T^\beta$, with $0 \leq \beta \leq n-1$, $0 \leq i \leq \nu-1$. Every power of U leaves the zero position unchanged, and no power of T (except the identity) leaves "0" unchanged. Thus, $U^i T^\beta = U^h \cdot T^\ell$, if and only if $i \equiv h \bmod \nu$ and $\beta \equiv \ell \bmod n$. It follows that the binary cyclic codes for n-odd are invariant under the group (T,U) permutations, and the group is of order $n \cdot \nu$ [49].

In addition to the above permutations, there exist other sets of permutations for which certain codes are also invariant. For example, the extended binary QR codes are invariant under the doubly-transitive-projective unimodular group. Also, Golay codes and extended Golay codes are preserved under the Mathieu group [9,52,53]. In this connection, we may point out that the Golay code is permutation decodable in the sense of the definition we have used in this work.

2.4 Permutation Decoding of Cyclic Codes

Let C be an (n,k,t) cyclic code over $GF(2)$. The code is generated by $g(x) = \frac{x^n+1}{h(x)}$, where n the code length, is odd and $h(x)$, the parity check polynomial, has degree k. Given any k-bit information polynomial $I(x)$, the corresponding code polynomial is of the form $C(x) = I(x) \cdot g(x)$.

Clearly, the group (T,U) permutations given in Equations (2.4) and (2.6) correspond to polynomial representations, and can be applied to binary cyclic code polynomials, $C(x) \in C$ as follows:

$$T^{\beta}[C(x)] = x^{\beta}C(x) \bmod(x^{n}+1), \quad \text{for} \quad 0 \leq \beta \leq n-1,$$
$$U^{i}[C(x)] = [C(x)]^{2^{i}} \bmod(x^{n}+1), \quad \text{for} \quad 0 \leq i \leq \nu-1, \tag{2.9}$$

Since every binary cyclic code is preserved by this group of permutations, each of $T^{\beta}[C(x)]$ and $U^{i}[C(x)]$ are code words if $C(x)$ is a code word. That is, the new polynomials $(x^{\beta}\cdot C(x) \bmod(x^{n}+1))$ and $([C(x)]^{2^{i}} \bmod(x^{n}+1))$ are simply $C(x)$ with the location index permuted by $\omega \rightarrow \omega+\beta$, and $\omega \rightarrow 2^{i}\cdot\omega$, respectively. Consequently if $C(x) \in C$, then $U^{i}\cdot T^{\beta}[C(x)] = ([C(x)]^{2^{i}} \cdot x^{\beta} \bmod(x^{n}+1)) \in C$ is also a codeword.

Suppose $R(x) = C(x) + E(x)$ is the received code word polynomial where $E(x)$, the error pattern polynomial, is of weight t or less. Dividing $R(x)$ by $g(x)$, we have $R(x) = a(x)\cdot g(x) + S(x)$, where $S(x)$, a polynomial of degree $\leq n-k-1$ is called the syndrome of the received word $R(x)$. Since $R(x)-E(x) = C(x) = I(x)\cdot g(x)$, the syndrome $S(x)$ is also the remainder after dividing $E(x)$ by $g(x)$. Therefore, it can be shown that the syndromes of $(x^{\beta}\cdot R(x) \bmod(x^{n}+1))$ and $([R(x)]^{2^{i}} \bmod(x^{n}+1))$ are equal to the remainders of the divisions of $(x^{\beta}\cdot E(x) \bmod(x^{n}+1))$ and $([E(x)]^{2^{i}} \bmod(x^{n}+1))$ by $g(x)$, respectively. Consequently, the syndrome of the permuted received word

$$S_{i\beta}(x) = (x^{\beta}\cdot[R(x)]^{2^{i}} \bmod(x^{n}+1)) \bmod g(x)$$

is

$$S_{i\beta}(x) = (x^{\beta}\cdot[E(x)]^{2^{i}} \bmod(x^{n}+1)) \bmod g(x)\,{}^{*} \qquad (2.10)$$

$$= x^{\beta}\cdot[E(x)]^{2^{i}} \bmod g(x) .$$

We shall use the $S_{i\beta}(x)$ in Equation (2.10) to develop the permutation decoding procedure.

We define $E(x)$ to be a permutation decodable (PD) pattern by cyclic shift and squaring if there are values of i and β such that $E_{i\beta}(x) = (x^{\beta}\cdot[E(x)]^{2^{i}} \bmod(x^{n}+1))$ has degree $n-k-1$ or less. That is, all the errors in the permuted $E(x)$ are confined to the first $n-k$ parity-check positions and $S_{i\beta}(x) = E_{i\beta}(x)$. In this case, the error pattern is $(x^{-\beta}\cdot S_{i\beta}(x))^{2^{-i}} \bmod(x^{n}+1)$. If these conditions on i and β hold for every $E_{i\beta}(x)$, such that the error $E(x)$ is PD with $i=0,1,\ldots,s<\nu$, then we say that the code C is $(s+1)$-step PD. In this connection, we note that $S_{i\beta}(x)$ has a weight t or less if and only if $E_{i\beta}(x)$ has a degree $n-k-1$ or less. This provides the decoder with a way of finding whether or not an error $E(x)$ of a weight t or less is PD.

Here, in the analysis of an (n,k,t) cyclic code, we represent an n-symbol error pattern $E(x)$ with weight t, by

$$E(x) = 1+x^{\beta_1}+x^{\beta_2}+\ldots+x^{t-1} \qquad (2.11)$$

[*] Observe that since $g(x)$ divides $X^{n}+1$ then the reduction modulo $X^{n}+1$ can be dropped.

where $0<\beta_1<\beta_2< \ldots < \beta_{t-1} \leq n-1$ are the error positions. For simplicity of analysis it is assumed that one of the errors is fixed at the position 0 $(\beta_0=0)$. If the code is 1-step PD, then $E(x)$ must satisfy one of the following conditions:

$$\begin{cases} \beta_1 \geq k+1 , \\ \text{or} \\ \beta_{i+1} - \beta_i \geq k+1, \quad \text{for some } 1 \leq i \leq t-2, \\ \text{or} \\ \beta_{t-1} \leq n-k-1 . \end{cases} \qquad (2.12)$$

Each of these conditions implies that some cyclic shift of the pattern will bring all the errors to the first ~~(or~~ last) n-k positions, and hence the last (first) k positions become error-free.

$$\begin{array}{l} 0 \qquad\qquad\qquad\qquad\qquad n-k-1 \qquad\qquad\qquad\quad n-1 \\ \underbrace{x \;\; x \;\; x \;\; \ldots \;\; x \;\; x \;\; x \;\; x}_{n-k \text{ positions}} \;\; \underbrace{x \;\; \ldots \;\; x \;\; x \;\; x}_{k \text{ positions}} \end{array}$$

Hence, in general, a code is s-step PD if for all error patterns $E(x) = 1+x^{\beta_1}+\ldots+x^{\beta_{t-1}}$, there is an integer j , $0 \leq i \leq s$, such that $\beta'_\ell = 2^j \cdot \beta_\ell \mod n$, $\ell = 1,2,\ldots,t-1$ meet conditions (2.12).

It is clear that in the anslysis, conditions (2.12) will be used over and over again.

From the above, it can be seen that permutation decoding involves only the positions of the errors. Therefore, the results that we obtain in this work are applicable not only to binary cyclic codes, but also to cyclic codes over $GF(2^m)$, for $m>1$. For example, the (7,3,2) Reed-Solomon code with $t=2$ over $GF(2^3)$ is 1-step PD [46].

Next, we give a decoder based on the group (T,U) permutation that are used in this work.

2.5 Permutation Decoder

A decoder based on the permutation decoding concept can be practically implemented. Consider an (n,k,t) binary cyclic code with generator polynomial $g(x)$. Suppose that we shift the received polynomial $R(x)$, from higher-order-term-first, into an $(n-k)$-stage shift register division circuit to produce $\rho_0(x) = R(x) \bmod g(x)$. As soon as the entire $R(x)$ has been shifted into the circuit, the weight of $\rho_0(x)$ in the register is tested by an $(n-k)$-input threshold gate whose output is "1" when t or fewer of its inputs are nonzero; otherwise, it is "0". This will imply that errors of $E(x)$ are confined to the first $n-k$ check positions, or that some errors in $E(x)$ lie in the last k positions, respectively. Then the decoding procedures can proceed as follows:

If the code is 1-step PD, then the errors of $E(x)$ will be brought to the first $n-k$ check positions with at most $(n-k-1)$ cyclic shifts of $\rho_0(x)$.

If the code is 2-step PD, and $(x^i \cdot \rho_0(x) \bmod (x^n+1)) \bmod g(x)$ has weight > t, for i=0,1, ..., n-k-1, the threshold gate output is 0. Then $\rho_0(x)$ through a wired squaring circuit is squared and the produced $\rho_0^2(x) \bmod (x^n+1)$ polynomial, from higher-order-term-first, is shifted into the division circuit. The weight of $(x^i \cdot \rho_0^2(x) \bmod (x^n+1)) \bmod g(x)$ polynomials are tested. We find the errors with at most 2(n-k-1) cyclic shifts of $x^i \cdot \rho_0(x) \bmod g(x)$, and $(x^i \cdot \rho_0^2(x) \bmod (x^n+1)) \bmod g(x)$.

In general, if the code is s-step PD, then at most $s \cdot (n-k-1)$ cyclic shifts of $\rho_0(x)$, $\rho_0^2(x)$, ..., $\rho_0^{2^{s-1}}(x)$ will complete the decoding procedures.

As an example, the implementation of a decoder for the binary (31,15,3) BCH code generated by

$$\begin{aligned} g(x) &= p_0(x) \cdot p_1(x) \cdot p_3(x) \cdot p_5(x) \\ &= (1+x)(1+x^2+x^5)(1+x^2+x^3+x^4+x^5)(1+x+x^2+x^4+x^5) \\ &= 1+x+x^4+x^8+x^{11}+x^{12}+x^{15}+x^{16} \end{aligned}$$

where $p_i(x)$ is the binary minimal polynomial of α^i, i=0,1,3,...,d-2 and α is a primitive element of $GF(2^5)$ [46], is shown in Figure 2-1.

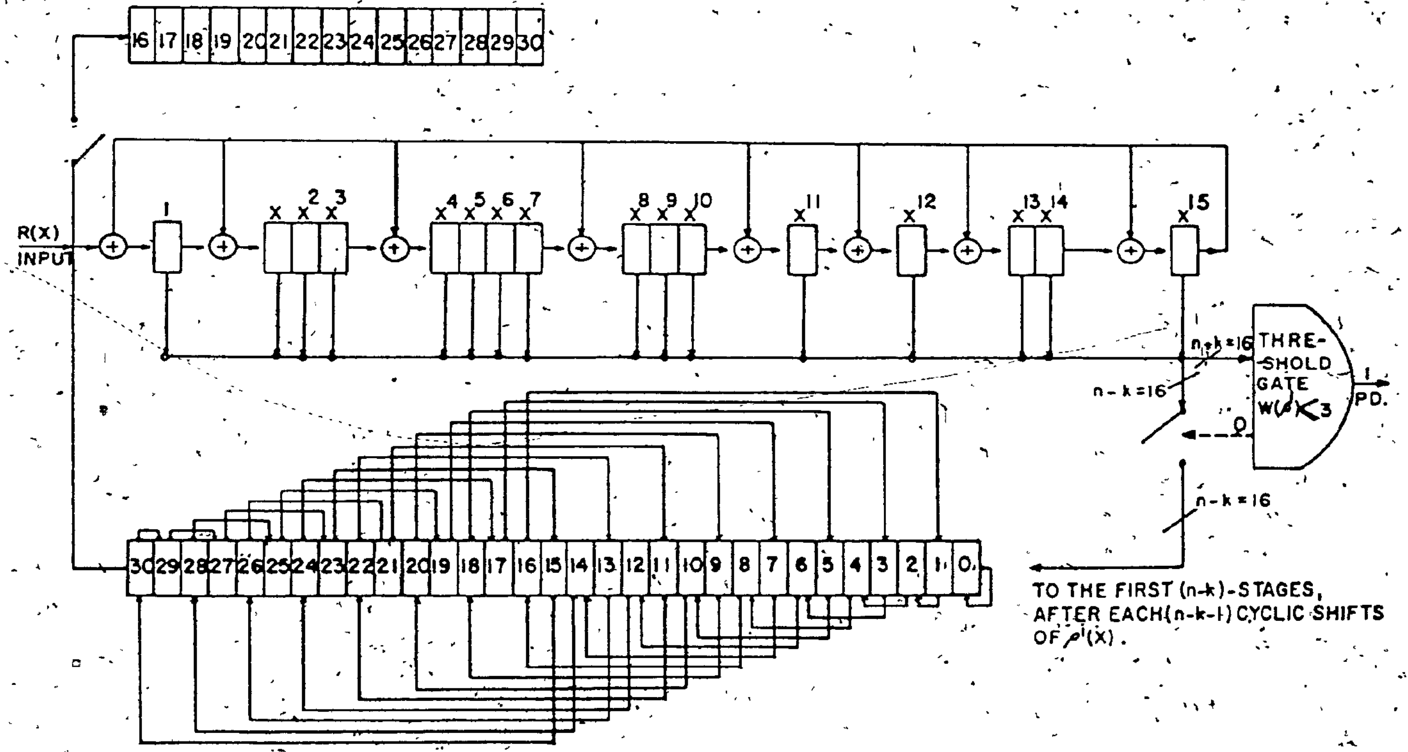


Figure 2-1. The implementation of the permutation decoder for binary (31,15,3) BCH code generated by: $g(x) = 1+x+x^4+x^8+x^{11}+x^{12}+x^{15}+x^{16}$.

This permutation decoder, in addition to the binary cyclic codes can be generalized to nonbinary codes over GF(2^m) with some modifications. As an example consider the (31,15,8) Reed-Solomon code with minimum distance, d=17, over GF(2^m), where m=5. The field GF(2^5) is formed by a primitive polynomial $p(x) = 1+x^2+x^5$ with α as the primitive element of the field. The generator polynomial $g(x)$ for this code is

$$g(x) = \prod_{i=1}^{16} (x+\alpha^i) = g_{16} \cdot x^{16}+g_{15}x^{15} +\ldots+ g_1x+g_0,$$

The feedback multipliers $g_0, g_1, \ldots, g_{16}$ are powers of α. Each nonbinary element of GF(2^5) can be expressed as a 5-tuple over GF(2). Therefore, the hardware implementation of this nonbinary code will be 5-times larger. This implies that to accommodate each symbol we require a 5-stage shift register. For the division circuit part we require addition and multiplication operations in GF(2^5), and for the squaring circuit part, squaring operations are needed. These operations must be done on the field elements of GF(2^5) for the selected minimal polynomial $p(x)$, and they can be implemented similarly to those circuits given in References [9,15]. The decoding procedures in this case will follow the same arguments as for the binary case.

In addition, each cyclic shift of patterns GF(2^m), will require m-bit shifts at a time.

In Chapters to follow, we will present exact lower bounds on the code length n of an (n,k,t) binary cyclic code C. This is achieved by applying 2-step and 3-step group (T,U) permutations.

Furthermore, some general results for cyclic codes which are not permutation decodable* are also obtained.

* When we say the code is not PD, for some (s+1)-step permutations, we mean that the consideration of $S_{0\beta}(x)$, $S_{1\beta}(x)$, ..., $S_{s\beta}(x)$ will not yield (decode) E(x).

CHAPTER 3

2-STEP PERMUTATION DECODABLE CYCLIC CODES

3.1 Introduction

In Chapter 2 we gave a brief review of several decoding techniques of the error-trapping type available for cyclic codes. In that direction, the concept of permutation decoding technique and the procedure of the decoder were introduced. In this chapter our main goal is to derive exact bounds on n in order for an (n,k,t) cyclic code to be permutation decodable (PD). Therefore, we investigate progressively the exact lower bound on the code length n, for the (n,k,t) PD cyclic code to be PD using only the permutation $U^i \cdot T^\beta$ for $i=0,1,\ldots,s<\nu$ and $\beta=0,1,\ldots,n-1$, where ν is the order of group (U) permutations.

In this chapter, we examine the capability of 2-step permutation decoding for cyclic codes. In Section 3.2, for the purpose of comparision, we provide a brief summary of the lower bounds previously presented in the literature. In Section 3.3, we introduce some preliminary notations and definitions in order to make the chapter self-contained. Section 3.4 explores the exact lower bounds on code length for (n,k,t) 2-step PD cyclic codes. In Section 3.5 we given the results for the codes which are exceptions to the bounds obtained in the previous section i.e. k=2 and t=2. Finally, Section 3.6 is a summary of the results derived in this chapter and also presents numerical results.

3.2 Summary of Known Results

Concerning the available results about the permutation decodability of cyclic codes using the group (T,U) permutations it is known that:

1) A binary (n,k,t) cyclic code C is 1-step PD if and only if

$$\frac{k}{n} < \frac{1}{t} , \tag{3.1}$$

that is, with just cyclic shift permutations [49]. This implies that all single-error-correcting cyclic codes are 1-step PD.

In References [46,54,55], the authors have mainly elaborated on the permutation decodability of certain t=2 and 3 error-correcting cyclic codes. In Reference [46] we have presented the following new result,

2) The triple-error-correcting (n,k,3) cyclic codes with n odd, and $n \le 2(n-k)+1$ or equivalently

$$\frac{k}{n} \le \frac{1}{2} + \frac{1}{n} , \tag{3.2}$$

are PD*, if $n \bmod 3 \neq 0$. If $n \bmod 3 = 0$, the only error patterns not PD are of the form

[*] When we say that an (n,k,t) cyclic code is PD, we mean that such a code will be permutation decodable for some (s+1)-step permutations, with $0 \le s < \upsilon$.

$$x^{i} \cdot (1+x^{\frac{n}{3}} + x^{\frac{2n}{3}}) \text{ . for } 0 \le i \le n-1. \qquad (3.3)$$

3) In Reference [56] it is shown that double-error correcting cyclic codes with $\frac{k}{n} < \frac{2}{3}$ are 2-step PD. Later on we shall obtain this result; but in a manner different from that found in [56].

4) It can also be shown that the double-error correcting (n,k,2) cyclic codes with

$$\frac{k}{n} \le \frac{4}{5} - \frac{1}{n} . \qquad (3.4)$$

are PD.

In References [56,57] further investigation has been done; the main result is the following:

5) The (n,k,t) binary cyclic codes with t being even and rate

$$\frac{1}{t} < \frac{k}{n} < \frac{2}{2t-1} = \frac{1}{t-\frac{1}{2}} \qquad (3.5)$$

are 2-step PD. This result with respect to Equation (3.1) for even t, the lower bound on n will be decreased only by $\frac{k}{2}$. Later on in this chapter, we present a tighter lower bound which is exact and is as tight as it can be. Also in References [56,57] the following conjecture was made:

6) The (n,k,t) binary cyclic codes with t being even, $t \ge 4$ and rate

$$\frac{1}{t} < \frac{k}{n} \leq \frac{1}{t-1} \qquad (3.6)$$

are 2-step PD.

3.3 Notation and Definitions

Let an n-symbol error polynomial $E(x) = 1+x^{e_1}+x^{e_2}+ ,,,+ x^{e_{t-1}}$ with weight t, over GF(2), be represented by a vector $\underline{e} = \{e_i, i=0,1 ,\ldots, t-1\}$ called the error pattern (EP) where $0,e_1,e_2,\ldots, e_{t-1}$ are the error positions $0 < e_1 < e_2 <\ldots< e_{t-1}$. For simplicity in analysis it is assumed that one of the errors is fixed at the zero position $(e_0=0)$. If the code is not correctly decodable by successive cyclic shifts (the group (T) permutations only), then $E(x)$ must satisfy the following conditions:

$$\begin{cases} e_1 \leq k, \\ e_{i+1}-e_i \leq k, \\ e_{t-1} \geq n-k, \end{cases} \qquad (3.7)$$

for all $i=1,2,\ldots, t-2$.

Hence, there is no gap of length $\geq k$ in the error pattern, and we say that the EP is not 1-step PD. By the "gap-length" G_ℓ, in an EP we mean the number of consecutive error-free coordinate positions between any two consecutive error locations.

The pattern associated with the result of permutation by U^i of

$E(x)$, i.e., the pattern associated with $E_{i\beta}(x) = (x^{\beta}\cdot[E(x)]^{2i} \bmod(x^n+1))$, will be referred to as "the EP in R_i-domain", for any $i=0,1,\ldots,s < \nu$, and for all values of β. It is desirable, if possible, to have an EP in an R_i-domain, having a minimum i, such that there exists at least a gap of length $\geq k$ in the pattern. Then, by definition, the pattern is $(i+1)$-step PD. Note that, since the codes C are cyclic, we can cyclically shift the EP whenever desired.

In this work we will be concerned with the parity of the error patterns for all R_i-domains with $i\geq 1$. Suppose that the EP in R_i-domain is not $(i+1)$-step PD. Then, the corresponding error location in R_{i+1}-domain can be classified into two classes. Namely, class 1 containing the fixed-position (F) errors, class 2 containing the nonfixed-position (NF) errors. The F and NF errors in R_{i+1}-domain correspond to the error positions in R_i-domain which are located before and after the $\lfloor\frac{n}{2}\rfloor$-th location in the pattern, respectively. Obviously, for n odd, the F and NF error positions are even and odd-valued, respectively. Here, in figures and tables we present pictorially F as "0" and NF as "x" positions. For example, in Figure 3-1 we are considering the error polynomial $E(x) = 1+x^7+x^{17}$ in R_0-domain for the (23,12,3) Golay code, which is not 1-step PD, and the corresponding EP in R_1-domain is not 2-step PD either.

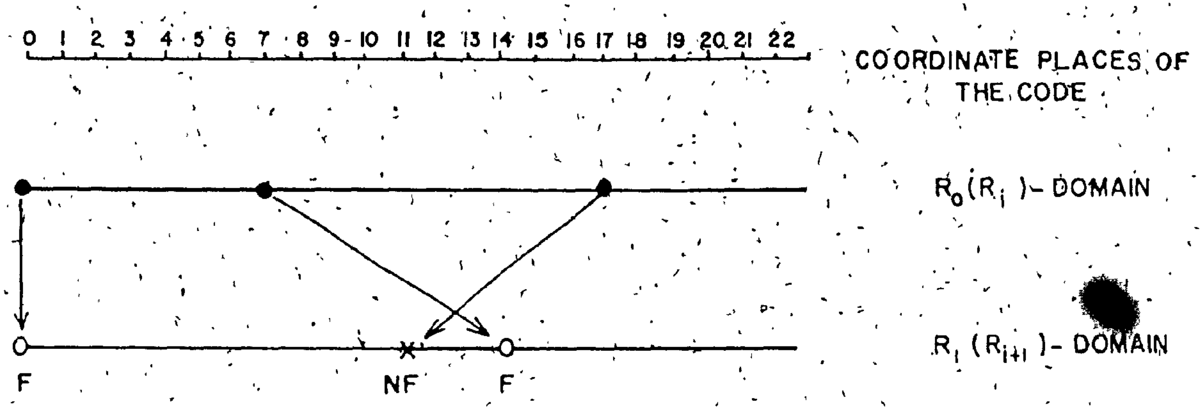


Figure 3-1. The error polynomial $E(x) = 1+x^7+x^{17}$, in R_0- and R_1-domains for the (23,12,3) Golay code.

We then distinguish four types of patterns; namely, $\alpha,\beta,\gamma,$ and σ, in any EP of an R_i-domain, for $i \geq 1$. The two principal patterns, namely the α-type pattern and the β-type pattern, correspond to any pair of consecutive error locations in an EP. These patterns are defined as:

1) α-type pattern of the forms

$$(0 \quad 0), \text{ or } (x \quad x),$$

2) β-type patterns of the forms

$$(0 \quad x), \text{ or } (x \quad 0).$$

The other two are combinations of these two patterns; namely, the γ-type pattern and the σ-type pattern, correspond to any three and four consecutive error locations in an EP, respectively. These patterns are defined as:

3) γ-type pattern; the combination of two β-type patterns is of the form:

$$(0 \quad x \quad 0), \text{ or } (x \quad 0 \quad x) \quad .$$

4) σ-type pattern, the combination of two α-type patterns, is of the form:

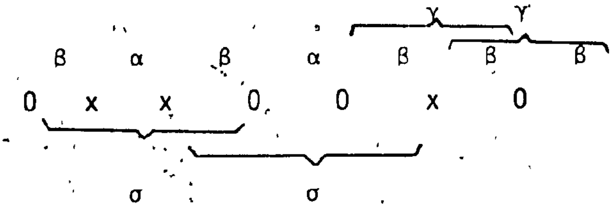
$$(0 \quad x \quad x \quad 0), \text{ or } (x \quad 0 \quad 0 \quad x).$$

The number of α-type, β-type, γ-type, and σ-type patterns in an EP are denoted by N_α, N_β, N_γ, and N_σ, respectively. In fact, in the worst case analysis, $N_\alpha = N_\sigma$.

EXAMPLE 3-1.

As an example, consider the error polynomial $E(x)$ in R_i-domain, $i \geq 1$, for the (49,15,7) cyclic code as

$$E(x) = 1 + x^3 + x^{15} + x^{20} + x^{30} + x^{37} + x^{44}$$



As is depicted in this EP, with reference to NOTE 3.3, there are $N_\alpha=2$ of α-type and $N_\beta=5$ of β-type patterns. In the same EP, there are $N_\gamma=2$ of γ-type and $N_\sigma=2$ of σ-type patterns.

We note that there may be many other patterns in an EP. However, in our analysis in which we consider the worst cases, we will be only concerned with these four patterns.

Finally, we proceed to give the main results and their proofs after the following important notes and Lemma 3.1.

NOTE 3.1.

All the variables with subscripts "e" or "o" are considered as even, or odd-valued variables, respectively.

NOTE 3.2

It should be clear that whether or not a cyclic code exists for a given rate is not pertinent in the derivation of the bounds on n. That is, when we say that an (n,k,t) code is (s+1)-step PD, we mean that such a code is decodable with $S_{0\beta}(x), S_{1\beta}(x),\ldots,S_{s\beta}(x)$; we do not imply that such a code exists.

NOTE 3.3

The zero coordinate place, in the analysis of the EPs in all R_i-domains, $i\geq 1$, has a "double parity" feature. This can easily be seen if we consider the errors as being located around a ring where the 0 and n coordinate places coincide with each other. For n odd, the zero location is considered as an even-valued number (F) for the first gap, and it is considered as an odd-valued number (NF) for the last gap in the pattern.

NOTE 3.4

According to NOTE 3.3, the zero location has a double parity feature and is considered an odd-valued number (NF) equal to n for the last NF in R_{i+1}-domain, $i\geq 0$. Therefore, the zero location and the last NF error positions correspond to the "end-gap" in R_i-domain. We denote the end-gap in a domain as EG in the text.

NOTE 3.5

In this thesis, whenever we refer to a code we mean a binary cyclic code.

LEMMA 3.1

For t odd (even) the number of α-type patterns N_α in an EP is even (odd), where the zero is being counted as an even-valued number.

Proof of Lemma 3.1

The lemma will be proven by mathematical induction on t.

For $t=1$, obviously $N_\alpha=0$.

Furthermore, for $t=2$, $N_\alpha=1$, because it has been assumed that one error is located at the zero coordinate place. This error will have a double parity feature. Therefore, an additional error of either parity (odd or even) will produce one and only one α-type pattern in comparison with the zero located error. Thus, the lemma is true in this case.

Finally, let us assume that for $t=\tau$, an odd number, N_α is even, and for $t=\tau+1$, an even number, N_α is odd. Then, since each additional error in the pattern causes N_α to increase or decrease by unity, the parity of N_α for $t=\tau+1$ is different from that of N_α for $t=\tau$. Thus, for odd τ, $\tau+1$ is even, and N_α is odd; and for even τ, $\tau+1$ is odd, and N_α is even. This completes the proof of the lemma.

Now, by applying 2-step permutation, we obtain the following exact lower bounds on the code length n.

3.4 2-Step Permutation Decoding

We categorize the 2-step PD codes into two groups:

I) the codes with an odd-valued t.

II) the codes with an even-valued t_o,

as follows.

3.4.1. The Case of t odd

The following theorem and its related corollaries present tight lower bounds on n for the (n,k,t_o) 2-step PD codes, with k even or odd.

Before going to the next theorem, the following corollary is given.

COROLLARY 3.1

The (n,k_e,t_o) codes with $n=t_o\cdot(k_e-1)+2\ell$, $\ell > \frac{t_o}{2}$, and the (n,k_o,t_o) codes with $n=t_ok_o+2\ell$, $\ell \geq 1$, are 1-step PD.

The proof of Corollary 3.1 is a direct consequence of Equation (3.1).

The corresponding case where $1 \leq \ell < \frac{t_o}{2}$ will be considered in the following Theorem.

THEOREM 3.1

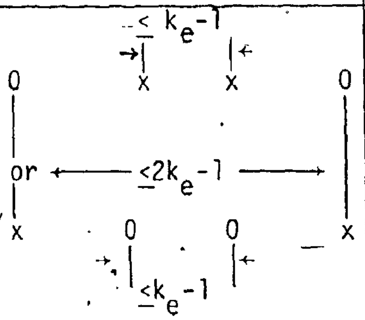
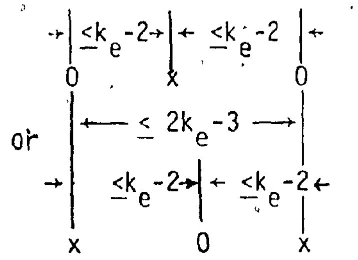
The (n,k_e,t_o) codes C, with $n=t_o(k_e-1)+2\ell$, for $1 \leq \ell < \frac{t_o}{2}$, $t_o \geq 3$ and $k_e \geq 4$, are 2-step PD.

Proof of Theorem 3.1

This theorem will be established by using the principle of contradiction. Suppose C is not 1-step or 2-step PD. Then, the different types of patterns γ and σ in R_1-domain, for any error pattern which is not 1-step or 2-step PD, are, in the worst case, as shown in Table 3-1. Note that the other types of patterns will result in lower gap-lengths (see Appendix).

TABLE 3-1

Gap-lengths associated with σ- and γ-types of patterns (with gap-length $\leq k_e$) in R_0- and R_1-domains

Type of Patterns	No. of Patterns	The type of patterns and gap-lengths in R_1-domain	The corresponding gap-lengths in R_0-domain
σ	N_σ	 $\leq k_e-1$ 0 x x 0 or $\leq 2k_e-1$ x 0 0 x $\leq k_e-1$	$\rightarrow \leq \frac{k_e-2}{2} + k_e-1$
γ	N_γ	 $\leq k_e-2$ $\leq k_e-2$ 0 x 0 or $\leq 2k_e-3$ $\leq k_e-2$ $\leq k_e-2$ x 0 x	$\rightarrow \leq k_e-2$
	1	x 0 1st NF last F	$\rightarrow \leq k_e-2$ (central-gap)

NOTE: The central-gap is referred to as the gap in R_0-domain which corresponds to first NF and last F error positions in R_1-domain.

In Tables 3-1 the third column shows the type of patterns and the associated qap-lengths in R_1-domain. Each σ-type pattern in R_1-domain corresponds to two separate gaps in R_0-domains.. One of these corresponds to the "internal" α-type pattern (the pattern of the form (x x) in the first row or of the form (0 0) in the second row), and it has a gap-length $g_1 \leq \frac{k_e-2}{2}$. The other gap corresponds to the "external" α-type pattern (the pattern of the form (0 0) in the first row or of the form (x x) in the second row), and it has a gap-length $g_2 \leq k_e-1$. Thus, the total gap-lengths in R_0-domain is g_1+g_2 as shown in the last column of the table. Each γ-type pattern in R_1-domain corresponds to a gap of length $g_\gamma \leq k_e-2$ in R_0-domain. Hence, the total number of gaps is t_o which is

$$t_o = 2N_\sigma + N_\gamma + 1$$

where $2N_\sigma$ is the total number of gaps associated with σ-type patterns, N_γ is the total number of gaps associated with γ-type patterns, and the unity term signifies with central-gap which corresponds to the first NF and the last F error positions of the EP in R_1-domain. The central-gap in column 3 can be considered to be $g_c \leq k_e-1$. Here, we have assumed that by cyclic shits (whenever desired) of the EP, the central-gap g_c in R_0-domain is made $\leq k_e-2$. This is possible if $n < k \cdot t$. This is indeed the case presently under consideration, otherwise the codes will be 1-step PD.

Thus, in R_0-domain, by using Table 3-1, the following relationship can be obtained:

$$t_o + N_\sigma(g_1+g_2) + N_\gamma g_\gamma + g_c \geq t_o(k_e-1) + 2\ell,$$

or

$$N_\sigma(k_e-4) \leq -4\ell , \qquad \text{for } \ell \geq 1. \qquad (3.7)$$

If the LHS of Equation (3.7) is required to be non-negative ($k_e \geq 4$ and $N_\sigma \geq 0$), then ℓ is required to be non-positive. However, this latter requirement is in contradiction with $\ell \geq 1$.

Q.E.D.

The following two corollaries are concerned with the (n,k,t) codes which are not 1-step or 2-step PD.

COROLLARY 3.2

The (n,k_o,t_o) codes with $n=t_o \cdot k_o$ are not 2-step PD.

COROLLARY 3.3

If the (n,k,t) code is not 2-step PD, then the (n',k,t) code with $n' = n-2\ell$, $\ell \geq 1$ is not 2-step PD.

The proofs of Corollaries 3.2 and 3.3 are not given here since these corollaries are special cases of Theorems 5.2 and 5.3, respectively, which will be given in Chapter 5.

According to Theorem 3.1 and Corollaries 3.1, 3.2, and 3.3, the bounds on n, for 2-step PD codes with t odd, are established.

3.4.2 The Case of t Even

The following Theorems 3.2 to 3.4 specify the exact lower bounds on n, for 2-step PD (n,k,t_e) codes, with k odd or even. First, consider the codes with odd k.

THEOREM 3.2

The (n,k_0,t_e) codes with $n = k_0(t_e-1)+2\ell$, for $1 \leq \ell < \frac{k_0}{2}$, $k_0 \geq 3$, $t_e \geq 4$, are 2-step PD.

Proof of Theorem 3.2

According to Corollary 3.1, the $(n,k_0,t_0 = t_e-1)$ codes with $n=t_0 \cdot k_0+2\ell$, $\ell \geq 1$, are 1-step PD. Here, we first prove that the $(n,k_0,t_0 = t_e-1)$ codes have at least two gaps of length $\geq k_0$ in R_0-or R_1-domain. Then, by adding an extra error $t_e = t_0+1$ to the EPs, these patterns remain 2-step PD.

We proceed to prove this theorem by using the principle of contradiction.

Consider the EPs of weight $t_0 = t_e-1$ with exactly one gap of length $\geq k_0$, and referred to them as "Λ_0-gap", in R_0-domain, and as "Λ_1-gap", in R_1-domain. Then assume that:

$$k_0 \leq \Lambda_i < 2k_0 \quad \text{for } i=0,1 \tag{3.8}$$

Otherwise, if either of the gaps Λ_0 or Λ_1 were $\geq 2k_0$, then by introducing an additional error to the patterns in each of these gaps, as pictorially shown in Figure 3-2,

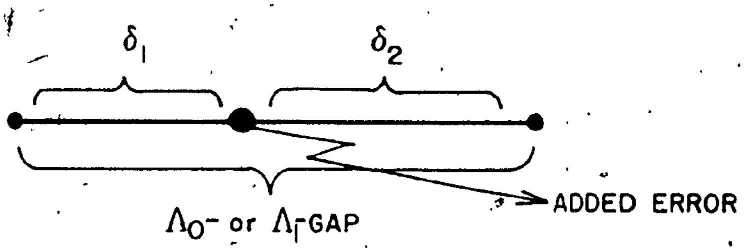


Figure 3-2. The additional error in either Λ_0-, or Λ_1- gap

the resulting two gaps δ_1 and δ_2 should satisfy

$$\delta_1 + \delta_2 + 1 \geq 2k_0,$$

or

$$\delta_1 + \delta_2 \geq 2k_0-1.$$

This will imply that δ_1 or $\delta_2 \geq k_0$, in which case the EPs are decodable. Therefore, the assumed Λ_0- and Λ_1-gaps should be such that

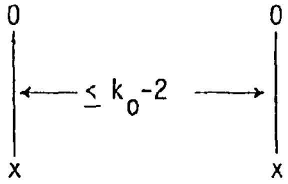
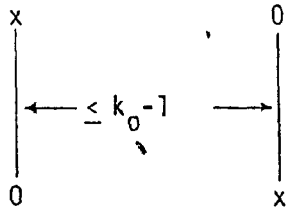
$$k_0 \leq \Lambda_i \leq 2k_0-1, \quad \text{for } i=0,1 \tag{3.9}$$

The different types of pattern in R_1-domain and their corresponding gap-lengths in R_0-domain for any EP which does not have a gap-length

$\geq k_0$, except for Λ_0 and Λ_1, are, in the worst case, as shown in Table 3-2.

TABLE 3.2

Gap-lengths associated with α- and β-types of patterns (with gap lengths < k_0) in R_0- and R_1-domains.

Type of Patterns	No. of Patterns	The type of patterns and gap-length in R_1-domain	The corresponding gap-lengths in R_0-domain
α	N_α		$\leq \frac{k_0-3}{2}$
β	N_β		$\leq k_0-1$
α or β	1	$k_0 \leq \Lambda_1 \leq 2k_0-1$	$k_0 \leq \Lambda_0 \leq 2k_0-1$

Note that N_α and N_β are respectively the numbers of α-type and β-type patterns in R_1-domain, without considering the type of the pattern associated with the Λ_1-gap in the EP. Note further that the β-type patterns do not correspond to a snecific gap in R_0-domain. Thus, for the worst case analysis, each α-type pattern in R_0-domain corresponds to a gap-length $g_\alpha \leq \frac{k_0-3}{2}$ and the β-type patterns in R_1-domain corresponds to a total of $N'_\beta = t_0-N_\alpha-1$ gaps in R_0-domain each of which is of length $g_\beta \le k_0-1$ where the minus unity term in N'_β signifies the Λ_0-gap. Therefore, from Table 3-2, the estimation of Λ_0-gap in R_0-domain is given by

$$t_0 + N_\alpha g_\alpha + N'_\beta g_\beta + \Lambda_0 \ge t_0 k_0 + 2\ell,$$

or

$$\Lambda_0 \ge \frac{N_\alpha}{2}(k_0+1) + (k_0-1) + 2\ell. \tag{3.10}$$

From Equation (3.10), if $N_\alpha \geq 2$, then

$$\Lambda_0 \ge 2k_0 + 2\ell > 2k_0,$$

which contradicts Equation (3.9). Thus, N_α has to be less than 2.

Depending on the pattern associated with the Λ_1-gap, if it is of type α, then for t odd N_α is even (Lemma 3.1), and there should be another gap of type α in R_1-domain; and if it is of type β, then N_α should be zero. We investigate these two possibilities through the following two lemmas:

LEMMA 3.2

The pattern associated with the gap Λ_1 is not of type α.

LEMMA 3.3

The pattern associated with the gap Λ_1 is not of type β.

Proof of Lemma 3.2

From Equations (3.9) and (3.10) we have,

$$2k_0-1 \geq \Lambda_0 \geq \frac{3k_0-1}{2} + 2\ell \ , \quad \text{for} \quad \ell \geq 1 \tag{3.11}$$

We can assume that by cyclic shifts, if necessary, on the EP, the Λ_0-gap is the first gap in R_0-domain. Thus, the corresponding gap in R_1-domain is referred to as the "F-F region", which is shown pictorially in Figure 3-3 (The F-F region is $\geq 3k_0+4\ell$).

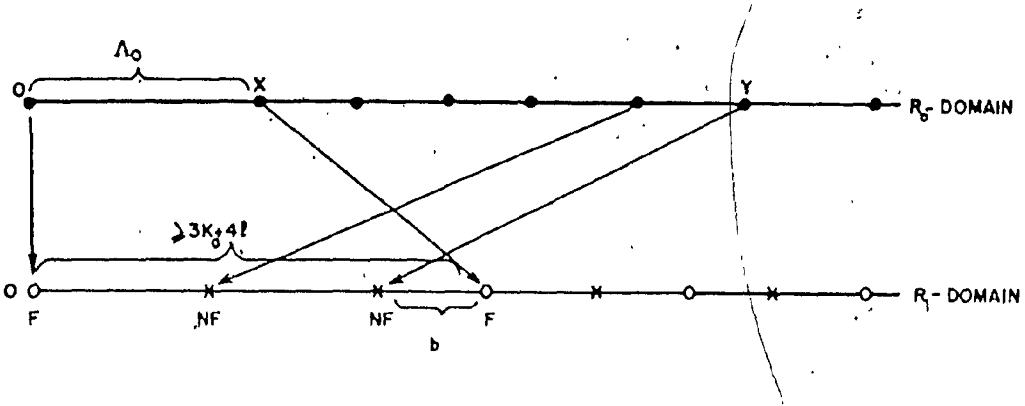


Figure 3-3. The graphical correspondence between both R_0-and R_1-domains.

The Λ_1-gap should occur in the F-F region. Otherwise, the additional error $t_e=t_0+1$ if positioned in either Λ_0- or Λ_1-gap, will be a decodable EP by one or the other of these gaps. There must not be more than three NF types of error positioned in the F-F region. Otherwise, $N_\alpha \geq 2$ which is not admissible. Moreover, since the Λ_1-gap is supposed to be in the F-F region (it has α-type pattern by assumption), there must be more than one NF error positioned in the region.

Suppose, there are only two NF errors in the F-F region. This will imply that there two NF errors are the errors associated with the Λ_1-gap, and that there should be another α-type pattern in the region outside the F-F in the EP.

Then, in the worst case, only two cases need to be investigated namely:

Case i

Let the α-type pattern outside the F-F region be of the form (x x) so that the last error position is an NF and that the last F in R_1-domain corresponds to the $(\frac{t_o-1}{2})$-th error in R_0-domain. Thus, the second F and the second NF error positions in the F-F region correspond to the second and the $(\frac{t_o+3}{2})$-th error positions in R_0-domain (denoted by X and Y, respectively in Figure 3-3). In this case, call the NF-F gap in the F-F region as the "b-gap". Then, we can estimate the length of this gap in connection with the corresponding error locations in R_0-domain, as follows;

$$b = 2X - (2Y-n) - 1,$$

and an upperbound for Y is

$$Y \leq \Lambda_0 + \frac{t_o-1}{2} \cdot (k_o-1) + \frac{t_o+1}{2} .$$

Therefore,

$$b \geq 2(\Lambda_0+1) - [2(\Lambda_o + \frac{t_o-1}{2} (k_o-1) + \frac{t_o+1}{2}) - (t_o \cdot k_o+2\ell)] - 1$$

$$= k_o-1+2\ell > k_o \ , \quad \text{for} \quad \ell \geq 1 \qquad (3.12)$$

This implies that, in addition to the Λ_1-gap, there is another gap $b \geq k_o$, which is a contradiction.

Case ii

Now, let the α-type pattern outside the F-F region be of the form (0 0). Then, the last error position in R_1-domain will be an error of type F. Thus, this error will correspond to the $(\frac{t_o+1}{2})$-th error in the R_0-domain. Therefore, the first NF in the F-F region corresponds to the $(\frac{t_o+3}{2})$-th error in the R_0-domain. Furthermore, it is assumed that the first gap is not $\geq k_o$ (otherwise the EP is decodable in R_1-domain). Therefore, the $(\frac{t_o+3}{2})$-th (i.e. the $e_{\frac{t_o+1}{2}}$) error in the R_0-domain should be:

$$2\, e_{\frac{t_o+1}{2}} - n \leq k_o \ ,$$

or

$$e_{\frac{t_o+1}{2}} \leq \frac{n+k_o}{2} \ .$$

Then, the last error position e_{t_o-1} in the R_0-domain satisfies:

$$e_{t_o-1} \leq e_{\frac{t_o-1}{2}} + \frac{t_o-3}{2} \cdot k_o \ .$$

Substituting for $e_{\frac{t_o+1}{2}}$ we get:

$$e_{t_o-1} \leq \frac{n+k_o}{2} + \frac{t_o \cdot k_o}{2} - \frac{3k_o}{2} = n-\ell-k_o$$

or

$$e_{t_0-1} < n-k_0 \,, \quad \text{for } \ell \geq 1.$$

Hence, the end-gap is $EG(n,e_{t_0-1}) = n-e_{t_0-1} \geq k_0$, and hence the EP is decodable in the R_0-domain, which is again a contradiction.

Suppose that there are three NF types of errors in the F-F region. In this case, one gap of NF-NF should be Λ_1-gap (otherwise $N_\alpha \geq 2$). Thus, in the region outside the F-F region, the error positions should be an alternation of NF and F, such that the last error in the R_1-domain is an error of type F corresponding to the $(\frac{t_0-1}{2})$-th error position in the R_0-domain. Therefore, if we calculate the end-gap in R_1-domain, we get

$$EG(n,2\, e_{\frac{t_0-1}{2}}) \geq t_0 \cdot k_0 + 2\ell - 2[2k_0 - 1 + \frac{t_0+5}{2}(k_0-1) + \frac{t_0-3}{2}] - 1$$

$$= k_0 - 1 + 2\ell \,, \quad \text{for } \ell \geq 1,$$

which is a contradiction. This concludes Lemma 3.2 which entails that Λ_1-gap cannot be of α-type pattern.

Proof of Lemma 3.3

From Equations (3.9) and (3.10) we have:

$$\Lambda_0 \geq k_0 - 1 + 2\ell > k_0 \qquad (3.13)$$

According to the arguments given in Lemma 3.2, there should be only one error of the type NF in the F-F region. It was assumed that the Λ_1-gap, being of type β, is located in the F-F region, consequently, $N_\alpha = 0$, i.e. the last error in the R_1-domain is an error of type F which corresponds to the $(\frac{t_0+1}{2})$-th error position in R_0-doamin. Thus, in the F-F region, the second F-type and NF-type errors correspond to the second and the $(\frac{t_0+3}{2})$-th error positions in the R_0-domain, which are denoted by X and Y, respectively, and which are shown in Figure 3-3. Similarly, it was shown that the b-gap in Equation (3.12) is greater than k_0 (that is the NF-F gap $> k_0$).

Now, in calculating the first gap-length in the F-F region, the F-NF gap should, in the worst case, be:

$$G_\ell(\text{F-NF}) = n-2(n-Y) - 1 \tag{3.14}$$

But, substituting for Y and Λ_0 from Lemma 3.2 and Equation (3.13), respectively, we obtain

$$2(n-Y) = t_0 \cdot k_0 + k_0 - 2 - 2\Lambda_0 + 4\ell \leq (t_0-1) \cdot k_0. \tag{3.15}$$

Then, from Equations (3.14) and (3.15) we get

$$G_\ell(\text{F-NF}) \geq n - (t_0-1) \cdot k_0-1 = k_0-1 + 2\ell > k_0. \tag{3.16}$$

This shows that both the F-NF and NF-F gaps are greater than k_0, which clearly is a contradiction. Thus, the Λ_1-gap cannot be of type β, and this completes the proof of Lemma 3.3.

These two lemmas contradict the above possibilities and hence lead to Theorem 3.2

Q.E.D.

Now we proceed to consider the codes with even k.

THEOREM 3.3

The (n,k_e,t_e) codes with $n = (k_e-1)(t_e-1) + 2$, $k_e \geq 4$, $t_e \geq 4$, are not 2-step PD.

Proof of Theorem 3.3:

It is sufficient to prove the theorem to find an error vector of weight t_e which does not have any gap of length $\geq k_e$ in R_0- or R_1-domain.

Consider, in R_0-domain, the error vector $\{e_i,\ i=0,1,\ldots,\ t_e-1\}$:

$$\{e_i\} = \begin{cases} e_0 = 0, & \\ e_i = i(k_e-1) - 1, & \text{for } 1 \leq i \leq \frac{t_e}{2} - 1, \\ e_i = i(k_e-1), & \text{for } \frac{t_e}{2} \leq i \leq t_e-1. \end{cases} \tag{3.17}$$

The gaps between any consecutive error position pairs in Equation (3.17) are obtained as

$$\begin{cases} G_\ell(e_1,e_0) = k_e-3, \\ G_\ell(e_{i+1},e_i) = k_e-2, \quad \text{for } 1 \leq i \leq t_e-1, \text{ except for } i = \frac{t_e}{2} - 1, \\ G_\ell(e_{\frac{t_e}{2}},\ e_{\frac{t_e}{2}-1}) = k_e-1. \end{cases} \tag{3.18}$$

The end-gap is obtained as $EG(n,e_{t_e-1}) = n-e_{t_e-1}-1 = 1$. Thus, from Equation (3.18) we can conclude that the pattern is not 1-step PD. The corresponding R_1-domain error vector $\{e_i'\}$ becomes,

$$\{e_i'\} = \begin{cases} e_0' = e_0 = 0, \\ e_{2i+2}' = 2e_{i+1} = 2\,(i+1)(k_e-1) - 2, & 0 \le i \le \frac{t_e}{2} - 1, \\ e_{2i+1}' = 2e_{(i+\frac{t_e}{2})} - n = (2i+1)(k_e-1) - 2. \end{cases} \qquad (3.19)$$

From equation (3.19), the gaps between any consecutive error position pairs in R_1-domain are:

$$\begin{cases} G_\ell(e_1',e_0') = k_e-4, \\ G_\ell(e_{i+1}',e_i') = k_e-2, & \text{for } 1 \le i \le t_e-1, \\ EG(n,e_{t_e-1}') = 3. \end{cases}$$

Thus, the pattern is not 2-step PD either.

Q.E.D.

Before introducing the next theorem, the following corollary is given.

<u>COROLLARY 3.4</u>

The (n,k_e,t_e) codes with $n = (t_e-1)(k_e-1) + 2(\ell+1)$, $\ell > \frac{t_e+k_e-4}{2}$, are 1-step PD.

The proof of Corollary 3.4 is a direct consequence of Equation (3.1). The case where $1 \le \ell \le \frac{t_e+k_e-4}{2}$ will be considered in Theorem 3.4.

<u>THEOREM 3.4</u>

The (n,k_e,t_e) codes with $n = (t_e-1)(k_e-1) + 2(\ell+1)$, for $k_e \ge 4$, $t_e \ge 4$, $1 \le \ell \le \frac{t_e+k_e-4}{2}$, are 2-step PD.

Proof of Theorem 3.4:

This Theorem will be established by using the principle of contradiction. Let us consider the γ- and σ-types of patterns in R_1-domain and their corresponding gap-lengths in R_0-domain, as given in Table 3.1.

It is assumed that these patterns are not 1-step or 2-step PD, implying that there is no gap-length $\geq k_e$. Then, in the worst case, relationships similar to those given in Theorem 3.1 can be obtained as

$$t_e = 2N_\sigma + N_\gamma + 1,$$

and

$$N_\sigma(g_1+g_2) + N_\gamma g_\gamma + g_c + t_e \geq (t_e-1)(k_e-1) + 2(\ell+1).$$

From these two we have

$$N_\sigma \leq \frac{2(k_e-2\ell-3)}{k_e-4} = 2 - \frac{4\ell-2}{k_e-4}. \tag{3.20}$$

From Equation (3.20), we conclude that N_σ, for $\ell \geq 1$, can not be greater than unity and with reference to Lemma 3.1 for even t, $N_\sigma = N_\alpha$ can only be an odd number. Consequently, $N_\sigma = 1$ and the following relationship in R_1-domain should hold

$$(N_\sigma = 1)(2k_e-1) + (t_e-2) + (k_e-2)(t_e-3) \geq (t_e-1)(k_e-1) + 2(\ell+1),$$

or

$$2 \geq 2(\ell+1),$$

This contradicts of the assumption that $\ell \geq 1$. Hence the proof of Theorem 3.4 is completed.

Q.E.D.

The following corollaries are applied in establishing the bounds on n.

COROLLARY 3.5

The (n,k_o,t_e) codes with $n=k_o(t_e-1)$, and the (n,k_e,t_e) codes with $n = (t_e-1)(k_e-1)$, are not 2-step PD.

COROLLARY 3.6.1

The (n,k_e,t_e) codes with $n = 3/4\ k_e \cdot t_e$, $k_e \cdot t_e \not\equiv 0 \bmod 8$, are not 2-step PD.

COROLLARY 3.6.2

The (n,k_e,t_o) codes with $n = 3/4\ k_e \cdot (t_o-1)$, $k_e \cdot (t_o-1) \not\equiv 0 \bmod 8$, are not 2-step PD.

COROLLARY 3.6.3

The (n,k_o,t_e) codes with $n = 3/4\ t_e \cdot (k_o-1)$, $t_e \cdot (k_o-1) \not\equiv 0 \bmod 8$, are not 2-step PD.

The proofs of Corollaries 3.5, 3.6.1, 3.6.2, and 3.6.3 are not given here since these corollaries are special cases of Corollary 5.2, Theorem 5.3, and Corollary 5.3 which will be given in Chapter 5.

So far, we have obtained lower bounds on n, both for odd and for even values of t for 2-step PD codes. However, those which have not been considered, are given in the following section as "special cases".

3.5 Special Cases

Exceptions to the bounds given above are for the codes with k=2, or with t=2. The following theorems are concerned with these cases.

THEOREM 3.5.1

The (n,k,t_o) codes with $k=2$ and $n > \frac{3t_o-1}{2}$ for odd value t, are 2-step PD.

Proof of Theorem 3.5.1

Let us assume that the codes with $n > \frac{3t_0-1}{2}$ are neither 1-step or 2-step PD. In this case, the different types of R_1-domain patterns and their corresponding R_0-domain gap-lengths are given in Table 3-3.

TABLE 3-3

Gap-lengths associated with α- and β-types of patterns (with gap-lengths <2) in R_0- and R_1-domains

Type of Patterns	No. of Patterns	The type of patterns and gap-lengths in R_1-domain	The corresponding gap-lengths in R_0-domain
α	N_α	$\overset{0}{\underset{x}{\mid}} \longleftarrow 1 \longrightarrow \overset{0}{\underset{x}{\mid}}$	$\longrightarrow \quad = 0$
β	N_β	$\overset{0}{\underset{x}{\mid}} \longleftarrow 1 \longrightarrow \overset{x}{\underset{0}{\mid}}$	$\longrightarrow \quad \le 1$

From Table 3-3, it is obvious that, for t-odd, $N_\alpha \le \frac{t_0-1}{2}$. Hence, if $n > \frac{t_0-1}{2} + t_0 = \frac{3t_0-1}{2}$, the codes will be 2-step PD. If $n \le \frac{3t_0-1}{2}$, by the following two lemmas, we prove that the codes are not 2-step PD.

LEMMA 3.4

The $(n,2,t_0)$ codes with $n = \frac{3t_0-1}{2}$, $(\frac{t_0-1}{2}$ even) are not 2-step PD.

LEMMA 3.5

The $(n,2,t_0)$ codes with $n = \frac{3t_0-3}{2}$, $(\frac{t_0-1}{2}$ odd) are not 2-step PD.

Proof of Lemma 3.4:

We take the following general EP:

$$\{e_i\} = \begin{cases} e_0 = 0, \\ e_1 = 1, \\ e_i = e_{i-2}+3 \ , \quad \text{for} \quad 2 \le i \le \frac{t_o-1}{2} \ , \\ e_{(\frac{t_o+1}{2})} = e_{(\frac{t_o-1}{2})}+2, \\ e_i = e_{i-2}+3 \ , \quad \text{for} \ \frac{t_o+3}{2} \le i \le t_o-1, \end{cases} \qquad (3.21)$$

in the R_0-domain. From Equation (3.21), the gaps between any pair of consecutive error positions are of the length not greater than 1. The end-gap is

$$EG(n,e_{t_o-1}) = \frac{3t_o-1}{2} - 3 \cdot \frac{t_o-1}{2} - 1 = 0.$$

Therefore, the EP is not 1-step. Now, for the corresponding EP in the R_1-domain the $\{e_i', \ i = 0,1,\ldots, t_o-1\}$ is

$$\{e_i'\} = \begin{cases} \left.\begin{matrix} e_{2i}' = e_{2i} \\ \\ e_{2i+1}' = e_{2i+1}+1 \end{matrix}\right\} , & \text{for} \quad 0 \le i \le \frac{t_o-5}{4} \ , \\ e_i' = e_i \ , & \text{for} \quad \frac{t_o-1}{2} \le i \le t_o-1. \end{cases} \qquad (3.22)$$

From Equations (3.21) and (3.22), the difference between the pattern

$\{e_i'\}$ in R_1-domain and $\{e_i\}$ in R_0-domain lies in the odd-valued error positions only which are located before $e'_{\frac{t_o-1}{2}}$. Such errors are shifted one position to the right, that is, the gaps of lengths 0 and 1 have interchanged to 1 and 0, respectively. Thus, the pattern $\{e_i\}$ is not 2-step PD. This completes the proof of Lemma 3.4.

The proof of Lemmas 3.5 for Theorem 3.5.1 is basically the same as that of Corollary 3.6, since $n = \frac{3t_o-3}{2} = \frac{3\cdot 2\cdot(t_e=t_o-1)}{4}$.

From Lemmas 3.4 and 3.5, Theorem 3.5.1 is true for $n = \frac{3t_o-1}{2}$ and $n = \frac{3t_o-3}{2}$, respectively. Then, according to Corollary 3.3, Theorem 3.5.1 is also true for $n < \frac{3t_o-3}{2}$.

Q.E.D.

THEOREM 3.5.2

The (n,k,t_e) codes with $k = 2$ and $n > \frac{3t_e}{2}$ for even value t, are 2-step PD.

Proof of Theorem 3.5.2:

By using Table 3-3 for the $(n,2,t_e)$ codes with even t, it is obvious that the number of the α-type patterns in EPs which are not 2-step PD, should satisfy

$$N_\alpha \leq \frac{t_e}{2} .$$

This implies that the codes with $n > \frac{3t_e}{2}$ are 2-step PD. For the codes with $n \leq \frac{3t_e}{2}$, in the following two lemmas we prove that they are not 2-step PD.

LEMMA 3.6

The $(n,2,t_e)$ codes, with $n = \frac{3t_e}{2}$, $\frac{t_e}{2}$ odd, are not PD.

The proof of Lemma 3.6 is contained in Corollary 3.6.1 for the special case $k_e=2$.

LEMMA 3.7

The $(n,2,t_e)$ codes, with $n = \frac{3t_e-2}{2}$, $\frac{t_e}{2}$ even, are not 2-step PD.

Proof of Lemma 3.7:

Consider the EP

$$\{e_i\} = \begin{cases} \left.\begin{aligned} e_{2i} &= 3i \\ e_{2i+1} &= 3i+1 \end{aligned}\right\}, & \text{for } 0 \le i \le \frac{t_e}{4}, \\ \\ \left.\begin{aligned} e_{2i} &= e_{2i-1}+1 \\ e_{2i+1} &= e_{2i}+2 \end{aligned}\right\}, & \text{for } \frac{t_e}{4}+1 \le i \le \frac{t_e}{2}-1. \end{cases} \qquad (3.23)$$

in R_0-domain. From Equation (3.23), it can be found that all the consecutive errors have gap-lengths not greater than 1. The end-gap is

$$EG(n,e_{t_e-1}) = \frac{3t_e-2}{2} - [(3 \cdot \frac{t_e}{4} + 1) + 3 \cdot (\frac{t_e}{2} - 1 - \frac{t_e}{4})] - 1 = 0.$$

Therefore, the pattern $\{e_i\}$ is not 1-step PD. The corresponding EP in R_1-domain is

$$\{e'_i\} = \begin{cases} e'_i = i & , \text{ for } 0 \le i \le 3 \\ \left.\begin{array}{l} e'_{2(i+1)} = e'_{2i}+3 \\ \\ e'_{2i+3} = e'_{2i+1}+3 \end{array}\right\} & , \text{ for } 1 \le i \le \frac{t_e}{2} - 2. \end{cases}$$

In the pattern $\{e'_i\}$ none of the gap-lengths are greater than 1 $(EG(n,e_{t_o-1}) = 1)$, and thus the EP is not 2-step PD. This completes the proof of Lemma 3.7.

According to Corollary 3.3, the results of Lemma 3.6 and 3.7 are also true for $n < \frac{3t_e}{2}$.

Q.E.D.

The next result is the special case for codes with $t = 2$.

THEOREM 3.6

The $(n,k,2)$ codes, with $t_e=2$ and $n > \frac{3k}{2}$, are 2-step PD.

Proof of Theorem 3.6:

Clearly, for $n > 2k$ the codes are 1-step PD. Now, we prove that for $\frac{3k}{2} < n < 2k$, the codes are 2-step PD.

Suppose an error pattern $\{e_0 = 0, e_1 = x\}$ which is not 1-step PD, that is,

$$x \le k \quad \text{and} \quad n - x \le k$$

or equivalently

$$n-k \le x \le k. \tag{3.24}$$

If such an EP is assuméd not to be 2-step PD, then the corresponding pattern in R_1-domain should satisfy:

$$\begin{cases} 2x-n \le k, \\ n-(2x-n) \le k. \end{cases} \tag{3.25}$$

From Equation (3.24) and (3.25), we obtain

$$n \le \frac{3k}{2} , \tag{3.26}$$

which contradicts the assumption.

Now, we prove that, if $n \le \frac{3k}{n}$, then the codes are not 2-step PD. For example, according to Equation (3.25), the EP $\{e_0 = 0, e_1 = \lceil \frac{n}{3} \rceil\}$, where $\lceil x \rceil$ denotes the smallest integer not less than x, is not 1-step PD for $k > \frac{2n}{3}$. Furthermore, the corresponding EP in R_1-domain has the following forms

$$\begin{cases} \text{first gap} = 2 \cdot \lceil \frac{n}{3} \rceil - 1 \le k-1, \\ \text{last gap} \;\, = n-2 \cdot \lceil \frac{n}{3} \rceil - 1 \le \frac{k}{2} - 1. \end{cases}$$

Thus, the pattern itself is not 2-step PD.

Q.E.D.

NOTE 3.6 Theorem 3.6 was given in Reference [56], and had been treated differently there.

Next, we present the summary and numerical results of this chapter.

3.6 Summary and Numerical Results

In this chapter we examined the capability of 2-step permutation decoding in correcting errors of weight t (or less), for the cases to t being odd or even ofthe (n,k,t) binary cyclic codes.

The main results of analyzing the capability of the decoder are presented in Theorem 3.1 to 3.6 and their related lemmas and corollaries. Exact lower bounds on the code length n are established by applying 2-step permutations to the cyclic codes C. A summary of the results obtained is given below.

3.5.1. Main Results

Theorem 3.1 to 3.4 and their related corollaries, establish the exact lower bounds on n for both values of t being odd or even with two exceptions; for the case $t=2$ and for the case $k=2$. The bounds for 2-step PD codes can be specified as follows.

3.5.1a For t odd-valued (t_o)

i) k odd-valued (n,k_o,t_o) codes:

$$n > k_o \cdot t_o . \tag{3.27}$$

ii) k even-valued (n,k_e,t_o) codes:

$$n > t_o \cdot (k_e-1) . \tag{3.28}$$

3.5.1b For t even-valued (t_e)

i) k odd-valued (n,k_o,t_e) codes:

$$n > (t_e-1) \cdot k_o . \tag{3.29}$$

ii) k even-valued (n,k_e,t_e) codes:

$$n > (t_e-1) \cdot (k_e-1) + 2 . \tag{3.30}$$

Exceptions to the bounds given above are for the codes with $t=2$, or with $k=2$. These cases are examined in Theorems 3.5.1, 3.5.2, and 3.6.

3.6.2 Special Cases

1) For $t=2$, $(n,k,2)$ codes:

$$n > 3 \cdot k/2 . \tag{3.31}$$

2) For $k=2$, $(n,2,t)$ codes:

i) t odd-valued

$$n > (3 \cdot t_o - 1)/2 \tag{3.32}$$

ii) t even-valued

$$n > 3 \cdot t_e/2 . \tag{3.33}$$

We can restate the cases (i) and (ii) for the case $k=2$, as if, $n > 3 \cdot t/2$, then the $(n,2,t)$ codes is 2-step PD.

Next, numerical results, based upon the complete lower bounds on n, for some values of t for 2-step PD, codes are given.

3.6.3 Numerical Results

Based on the results obtained in this chapter, we give four tables 3-4 to 3-7, for some specific numbers of correctable errors $t_o=5,9$, and $t_e=6,10$, as examples. The tables show the exact lower bounds on code length n, for a given information length k, for 2-step PD codes.

In the next chapter, we derive similar results for codes which are 3-step PD.

TABLE 3-4

2-Step PD codes of length n, and t_o=5.

k =	2	4	6	8	10	12	14	16	• • •
n The Code Length	x	17	27	37	47	57	67	77	
	9	19	29	39	49	59	69	79	
	11	21	31	41	51	61	71	81	
	13	23	33	43	53	63	73	83	
	15	25	35	45	55	65	75	85	

TABLE 3-5

2-Step PD codes of length n, and t_e=6.

k =	2	3	4	5	6	7	8	9	10	11	12	13	14	15	• • •
n The Code Length	x	17	19	27	29	37	39	47	49	57	59	67	69	77	
	11	--	21	--	31	--	41	--	51	--	61	--	71	--	
	13	--	23	--	33	--	43	--	53	--	63	--	73	--	
	15	--	25	--	35	--	45	--	55	--	65	--	75	--	

Note: In these tables, the occurence of an "x" or a "--" in a place means that no codes exist in that place.

TABLE 3-6

2-Step PD codes of length n, and $t_o=9$.

k =	2	4	6	8	10	12	14	16	. . .
n The Code Length	X	29	47	65	83	101	119	137	
	X	31	49	67	85	103	121	139	
	15	33	51	69	87	105	123	141	
	17	35	53	71	89	107	125	143	
	19	37	55	73	91	109	127	145	
	21	39	57	75	93	111	129	147	
	23	41	59	77	95	113	131	149	
	25	43	61	79	97	115	133	151	
	27	45	63	81	99	117	135	153	

TABLE 3-7

2-Step PD codes of length n, and $t_e=10$.

k =	2	3	4	5	6	7	8	9	10	11	12	13	14	. . .
n The Code Length	X	29	31	47	49	65	67	83	85	101	103	119	121	
	X	--	33	--	51	--	69	--	87	--	105	--	123	
	17	--	35	--	53	--	71	--	89	--	107	--	125	
	19	--	37	--	55	--	73	--	91	--	109	--	127	
	21	--	39	--	57	--	75	--	93	--	111	--	129	
	23	--	41	--	59	--	77	--	95	--	113	--	131	
	25	--	43	--	61	--	79	--	97	--	115	--	133	
	27	--	45	--	63	--	81	--	99	--	117	--	135	

CHAPTER 4

3-STEP PERMUTATIONS IN DECODING CYCLIC CODES

4.1 Introduction.

In addition to the results given in the previous chapter for 2-step PD codes, the exact lower bounds can be improved by applying higher order permutation from the group (T,U). In this chapter, we examine the capability of the permutation technique in decoding cyclic codes by applying 3-step permutations.

In Section 4.2, with respect to the definitions and notations given in Section 3.3, we develop the results for 3-step permutations with $U^i \cdot T^\beta$, $i=0,1,2$, $\beta=0,1,\ldots,n-1$ to the codes with t odd and to the case of $t=2$. Section 4.3 examines the results for codes which are exception to the bounds given in Section 4.2. Finally, Section 4.4 summarizes the results obtained in this chapter and presents some numerical results.

4.2 3-Step Permutation Decoding

Here, we extend the results to the codes with t odd, and to the case of double-error-correction.

4.2.1. The Case of t odd:

The following theorems and corollaries specify the improvement of the lower bounds on n for the (n,k_e,t_o) 3-step PD codes with k even.

Before presenting Theorem 4.1, the following corollaries are given.

COROLLARY 4.1

The (n,k_e,t_0) codes with $n=(k_e-1)\cdot t_0$. and the (n,k_0,t_0) codes with $n=k_0\cdot t_0$, are not 3-step PD.

COROLLARY 4.2

If the (n,k,t) code is not 3-step PD, then the (n',k,t) codes, with $n'=n-4\ell$ for $\ell\geq 1$, are not 3-step PD.

The proof of Corollaries 4.1 and 4.2 are not given here since these Corollaires are special cases of Theorems 5.1 and 5.2, respectively for $s=3$ which will be given in Chapter 5.

COROLLARY 4.3

The (n,k_e,t_0) codes with $n=(k_e-1)t_0-2(2\ell+1)$, $\ell\geq\frac{t_0-1}{2}$, $k_e\geq 2(\ell+2)$, are not 3-step PD.

The proof of Corollary 4.3, is contained in Corollaries 4.1 and 4.2, which imply that the codes with $n=(k_e-1)\cdot t_0-2(2\ell+1)=(k_e-3)\cdot t_0-4\ell'$, for $\ell'\geq 0$, are not 3-step PD.

The corresponding case where $0\leq\ell\leq\frac{t_0-3}{2}$ will be considered in Theorems 4.1 and 4.2.

THEOREM 4.1

The (n,k_e,t_0) codes with $n=(k_e-1)\cdot t_0-2(2\ell+1)$, $0\leq\ell\leq\frac{t_0-3}{2}$, $k_e > 2(\ell+2)$, are 3-step PD.

Proof of Theorem 4.1:

To prove this Theorem, we use the principle of contradiction. Let us assume that the (n,k_e,t_0) codes with $n=(k_e-1)\cdot t_0-2(2\ell+1)$ are not

3-step PD. Moreover, let us assume that $\{e_i, i=0,1,\ldots, t_0-1\}$ is a pattern in the R_0-domain which is not 1-step, 2-step, or 3-step PD. According to the different types of patterns γ, σ in the EPs given in Table 3-1, in the worst case, the following relationships should hold:

$$\begin{cases} 2N_\sigma + N_\gamma = t_0-1, \\ t_0 + N_\sigma\,(g_1+g_2) + N_\gamma \cdot g_\gamma+g_c \geq t_0(k_e-1) - 2(2\ell+1). \end{cases}$$

From these two we have

$$N_\sigma \leq \frac{4(2\ell+1)}{k_e-4}\ , \tag{4.1}$$

in the R_1-domain. Thus, from Equation (4.1) and Lemma 3.1, for odd t, $N_\sigma=N_\alpha$ can be obtained as

$$\begin{cases} N_\sigma = 0\ , & \text{for } k_e \geq 4 \\ N_\sigma = 2\ , & \text{for } k_e \leq 4\ell+6, \\ N_\sigma \geq 4\ , & \text{for } k_e \leq 2(\ell+2). \end{cases} \tag{4.2}$$

Note that codes with $N_\sigma \geq 4$ (for $k_e \leq 2(\ell+2)$) do not satisfy the assumptions of the theorem and are thus excluded from further consideration.

We proceed to the cases $N_\sigma=0$ and $N_\sigma=2$, by considering the EPs in the R_2-domain with respect to (w.r.t) the EPS in R_1-and R_0-domains. Let us denote the number of the α-type and the σ-type patterns in the EP in R_2-domain as N'_α and N'_σ respectively. Then, according to

Equation (4.2) we can conclude that $N'_\sigma \leq 2$. Thus, we have the following two lemmas:

LEMMA 4.1

Suppose that $N_\sigma=0$; then the corresponding pattern in R_2-domain for $N'_\sigma = 0$ or 2, will be 3-step PD.

LEMMA 4.2

Suppose that $N_\sigma=2$. Then, the corresponding patterns in R_2-domain, with $N'_\sigma = 0$ or 2, will be 3-step PD.

Proof of Lemma 4.1

We prove this lemma by using the principle of contradiction. The proof of the lemma will be divided into the following two cases.

Case 4.1.1: $N'_\sigma = 0$.

In this case, if for $n=t_o\cdot(k_e-1) -2(2\ell+1)$ the pattern is not decodable, then there should be a collection of gaps of lengths (k_e-2-2j) in R_1-domain, where $1\leq j\leq 2\ell+1$ (for $N_\sigma=N_\alpha=0$). Now, in the worst case (i.e., when $j=1$), the estimate for the total code length n, should satisfy

$$2\left\lceil\frac{2\ell+1}{2}\right\rceil\cdot(k_e-4)+\left(t_o-2\left\lceil\frac{2\ell+1}{2}\right\rceil\right)(k_e-2)+t_o \geq n = t_o\cdot(k_e-1)-2(2\ell+1)$$

or

$$t_o\cdot(k_e-1) - 4(\ell+1) \geq t_o\cdot(k_e-1) - 2(2\ell+1) \qquad (4.3)$$

in the R_2-domain, which is contradicts $\ell \geq 1$.

Case 4.1.2: $N_o'=2$

In this case, there should be two even-valued gaps of lengths $\leq \frac{k_e-2}{2}$, or

$$k_e = 4i+2 \quad , \text{ for some } i \geq 0 \tag{4.4}$$

In addition, in R_1-domain, there should be a number x, of gaps of lengths (k_e-2-2j) for $j \geq 1$. In the worst case (when $j = 1$), the following relationship,

$$t_o + x\cdot(k_e-4) + 2\cdot(\frac{k_e-2}{2}) + (t_o-2-2x)(k_e-2) \geq n \quad , \tag{4.5}$$

should be satisfied in R_1-domain. From Equations (4.4) and (4.5), we get the value for x as

$$x = \frac{4\ell-k_e-4}{2} > 0 \tag{4.6}$$

which is an odd number. For the corresponding EP in R_2-domain, the estimate for the total code length n, is

$$(t_o-4)+2\cdot\lceil\frac{x}{2}\rceil\cdot(k_e-4)+(4k_e-6)+(t_o-2\cdot\lceil\frac{x}{2}\rceil-5)\cdot(k_e-2) \geq n. \tag{4.7}$$

By substituting for $2\cdot\lceil\frac{x}{2}\rceil = \frac{4\ell-k_e+4}{2}$ in Equation (4.6), we get:

$$t_o \cdot (k_e-1) - 4(\ell+1) \geq t_o \cdot (k_e-1) - 2(2\ell+1).$$

which is again a contradiction. This completes the proof of Lemma 4.1.

Proof of Lemma 4.2

This Lemma will be established through contradiction. Similar to the proof of Lemma 4.1 we consider the following two cases.

Case 4.2.1: $N'_\sigma = 0$:

In this case, let us assume that there exist an EP such that $N_\sigma=2$ and $N'_\sigma=0$ which is not 3-step PD. Therefore, for such an EP, in the worst case, the estimate for the code length n, as shown in Figure 4.1, should satisfy

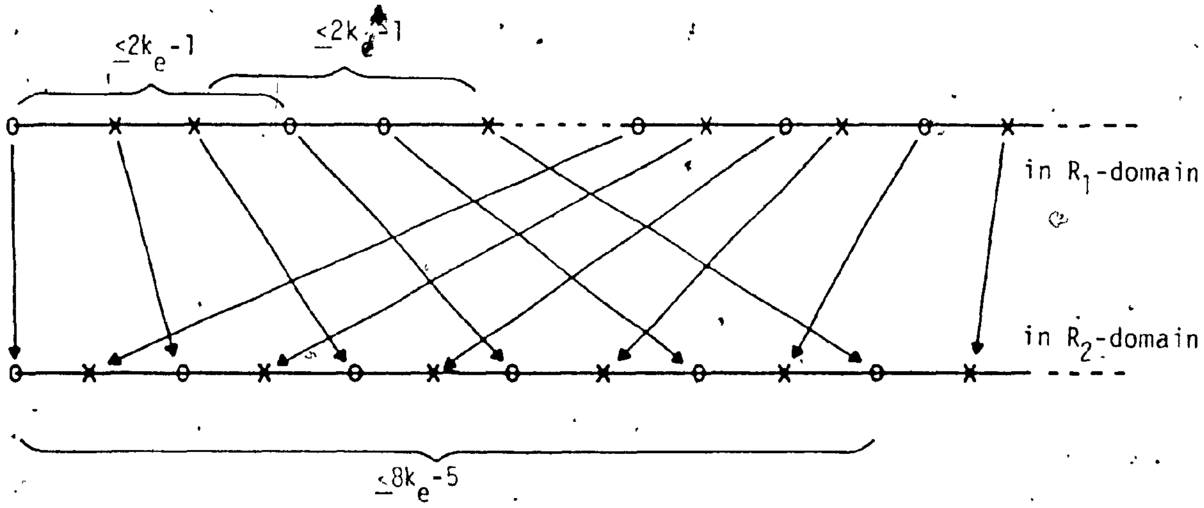


Figure 4-1. A sketch representing the case of $N_\sigma=2$ and $N'_\sigma=0$ in the worst case.

$$(8k_e-5)+(t_o-10)\cdot(k_e-2)+(t_o-9) \geq t_o\cdot(k_e-1) -2(2\ell+1),$$

or

$$k_e \leq 2\ell+4 \qquad (4.8)$$

in R_2-domain for $t_o \geq 11$ (The case $t_o < 11$ is not possible). The condition in Equation (4.8) does not satisfy the assumptions of Theorem 4.1.

Case 4.2.2: $N'_\sigma=2$.

With reference to Figure 4-1, it can easily be seen that this case is a subcase of Case 4.2.1 and since Case 4.2.1 is 3-step PD, Case 4.2.2 is obviously 3-step PD.

This leads to Lemma 4.2.

The proofs of Lemma 4.1 and 4.2 completes the proof of Theorem 4.2.

Q.E.D.

Now we consider the case of $k_e=2(\ell+2), 0\leq\ell\leq \frac{t_o-3}{2}$, in the following theorem.

THEOREM 4.2

The (n,k_e,t_o) codes with $n=(k_e-1)\cdot t_o-2(2\ell+1)$, $0\leq\ell\leq \frac{t_o-3}{2}$ and $k_e = 2(\ell+2)$, are not 3-step PD.

Proof of Theorem 4.2

To prove this theorem, it is sufficient to find an EP which is not 1-step, 2-step, or 3-step PD. Thus, let us consider the EP

$$\{e_i\} = \begin{cases} e_0 = 0 \ , \\ e_1 = 2, \\ e_i = e_{i-1}+(k_e-1) & , \text{ for } 2 \le i \le \frac{t_o-1}{2} \\ e_{\frac{t_o+1}{2}} = e_{\frac{t_o-1}{2}} + \frac{k_e}{2} \\ e_{\frac{t_o+3}{2}} = e_{\frac{t_o+1}{2}} + (\frac{k_e+2}{2}), \\ e_i = e_{i-1}+(k_e-1) & , \text{ for } \frac{t_o+5}{2} \le i \le t_o-1, \ t_o > 5 \end{cases} \tag{4.9}$$

in R_1-domain. From Equation (4.9) it can be seen that none of the gap-lengths is greater than (k_e-2). The end-gap is

$$\begin{aligned} EG(n,e_{t_o-1}) &= n - [2+(\frac{t_o-1}{2} - 1)(k_e-1)+ \frac{k_e}{2} + \frac{k_e+2}{2} + (t_o - \frac{t_o+5}{2})(k_e-1)-1 \\ &= k_e-2. \end{aligned}$$

It is sufficient to prove that this pattern does not have any gap of length $\ge k_e$, in R_0 and R_2-domains.

First, we consider the corresponding EP in R_0-domain. The correspondence between the EPs in R_1- and R_0-domains is given in Table 4-1 for $(\frac{k_e}{2})$ odd, and in Table 4-2 for $(\frac{k_e}{2})$ even.

TABLE 4-1

For odd-valued $\frac{k_e}{2}$.

Gap-lengths in R_o-domain associated with any two consecutive errors with the same parity in R_1-domain.

†Distances D in R_1-domain	Corresponding gap-lengths in R_o-domain
$D(e_1,e_o) = 2$	$= 0$
$D(e_{i+2},e_i) = 2k_e-2, \quad 1 \le i \le \frac{t_o-5}{2}, \quad t_o>5$	$= k_e-2$
$D(e_{\frac{t_o+1}{2}},e_{\frac{t_o-3}{2}}) = \frac{3k_3-2}{2}$	$= \frac{3(k_e-2)}{4}$
$D(e_{\frac{t_o+3}{2}},e_{\frac{t_o+1}{2}}) = \frac{k_e}{2}+1$	$= \frac{k_e-2}{4}$
‡$D(e_{\frac{t_o+5}{2}},e_{\frac{t_o-1}{2}}) = 2k_e$	$= k_e-1$
$D(e_{i+2},e_i) = 2k_e-2, \quad \frac{t_o+3}{2} \le i \le t_o-3,$ if $t_o > 7$	$= k_e-2$
$D(e_{t_o-1},e_2) = D$ (first NF, last F) $= \frac{n-(e_{t_o-1}-e_2)}{2} - 1$	$= k_e-1$ (central-gap)
$D(n,e_{t_o-2}) = 2k_e-2$	$= k_e-2$ (end-gap)

†The distances "D" are taken to be between any two consecutive errors with the same parity

‡When $t_o=5$, then the end-gap in R_o-domain will correspond to $G_\ell(n,e_2) = G_\ell(n,e_{\frac{t_o-1}{2}}) = 2k_e-1$, and is equal to k_e-1.

TABLE 4-2

For even-valued $\frac{k_e}{2}$.

Gap-length in R_0-domain associated with any two consecutive errors with the same parity in R_1-domain.

Distances in R_1-domain	Corresponding gap-lengths in R_0-domain
$D(e_1,e_0) = 2$	$= 0$
$D(e_{i+2},e_i) = 2k_e-2 \quad 1\le i\le \frac{t_0-5}{2}, t_0>5$	$= k_e-2$
$D(e_{\frac{t_0+1}{2}},e_{\frac{t_0-1}{2}}) = \frac{k_e}{2}$	$= \frac{k_e-4}{4}$
$D(e_{\frac{t_0+3}{2}},e_{\frac{t_0-3}{2}}) = 2k_e$	$= k_e-1$
†$D(e_{\frac{t_0+5}{2}},e_{\frac{t_0+1}{2}}) = \frac{3k_e}{2}, \quad t_0>5$	$= \frac{3k_e-4}{4}$
$D(e_{i+2},e_i) = 2k_e-2, \quad \frac{t_0+3}{2} \le i \le t_0-3, \ t_0>7$	$= k_e-2$
$D(e_{t_0-1},e_2) = D$ (first NF, last F) = $\frac{n-(e_{t_0-1}-e_2)}{2} - 1$	$= k_e-1$ (central-gap)
$D(n,e_{t_0-2}) = 2k_e-2$	$= k_e-2$ (end-gap)

† When $t_0=5$, then the end-gap will correspond to $G_e(n,e_3)$, where $e_{\frac{t_0+5}{2}}$ is considered to be n.

From Tables 4-1 and 4-2, we can conclude that the pattern $\{e_i\}$ is not 1-step or 2-step PD.

Second, we consider the EP corresponding to $\{e_i\}$ in R_2-domain and denote it by $\{e_i', i=0,1,\ldots, t_0-1\}$. From Equation (4.9) we obtain

$$\{e_i'\} = \begin{cases} e_0' = e_0 = 0, \\ e_1' = 2e_{\frac{t_0+1}{2}}-n = 1, \\ e_2' = 2e_1 = 4, \\ e_i' = e_{i-1}+(k_e-1)\ , \qquad 3 \le i \le t_0-1\ . \end{cases} \tag{4.10}$$

Again, it can be seen that none of the gap-lengths in R_2-domain are greater than (k_e-2). The end-gap is

$$EG(n,e_{t_0-1}') = n - [(k_e-1)(t_0-3) + 4] - 1 = k_e-2.$$

Thus, the pattern $\{e_i\}$ is not 3-step PD. Q.E.D.

The following corollary in establishing the bounds on n is applied for codes with k even.

<u>COROLLARY 4.4</u>

The (n,k_e,t_0) codes with $n=3/4\ k_e\cdot(t_0-1)$, $k_e(t_0-1) \equiv 0 \bmod 8$, are not 3-step PD.

The proof of this corollary is not given here since it is an special case of Corollary 5.4 which will be given in Chapter 5.

Next, for the case of k being odd, we have the following results.

COROLLARY 4.5

The (n,k_0,t_0) codes with $n = t_0\cdot k_0-2(2\ell+1)$, $k_0 = 2\ell+3$ and $\ell \geq \frac{t_0-1}{2}$, are not 3-step PD.

The proof of Corollary 4.5 is contained in those of Corollaries 4.1 and 4.2 which imply that the codes with $n = k_0\cdot t_0-2(2\ell+1) = (k_0-2)\cdot t_0-4\ell'$, for $\ell' \geq 0$, are not 3-step PD.

The corresponding case where $0 \leq \ell \leq \frac{t_0-3}{2}$ will be considered in Theorem 4.3.

THEOREM 4.3

The (n,k_0,t_0) codes with $n = t_0\cdot k_0-2(2\ell+1)$, $k_0 = 2\ell+3$, $0 \leq \ell \leq \frac{t_0-3}{2}$, are 3-step PD.

Proof of Theorem 4.3

Theorem 4.3 will be established through contradiction. Let us assume that the (n,k_0,t_0) codes, with $n = t_0\cdot k_0-2(2\ell+1)$ and $k_0 = 2\ell+3$, are not 3-step PD. Moreover, let us assume $\{e_i\}$ is an EP in R_0-domain which is not 1-step, 2-step or 3-step PD. With reference to Equation (3.7), for the different types of patterns γ,σ in the EPs given in Table 3-1, in the worst case the following relationship should hold:

$$\begin{cases} 2\,N_\sigma + N_\gamma + 1 = t_0, \\ N_\sigma\cdot(g_1+g_2) + N_\gamma g_\gamma + g_c + t_0 \geq t_0 k_0 - 2(2\ell+1). \end{cases}$$

From these two we have

$$N_\sigma \le \frac{2(4\ell+1)}{k_0+1} \qquad (4.11)$$

in R_1-domain. Here, $g_1 = \frac{k_0-3}{2}$, $g_2 = k_0-1$, and $g_c = k_0-2$. Consequently, from Equation (4.11) and Lemma 3.1, we have $N_\sigma \le 2$. The remaining part of the proof is very similar to those of Lemma 4.1 and 4.2, and will be omitted.

Q.E.D.

So far we have obtained the lower bounds on n for 3-step PD codes for odd t. Furthermore, in connection with even values of t, we only consider the case of $t_e=2$, as follows.

4.2.2. The Case of $t_e=2$

The following Theorems specify the improvement of the bound given in Theorem 3.6 for double-error-correction.

THEOREM 4.4

The (n,k_0,t_e) codes with $n = k_0 + \frac{k_0-1}{2}$ for $t_e = 2$ and $k_0 > 3$, are 3-step PD.

Proof of Theorem 4.4

According to the proof of Theorem 3.6, in Equation (3.24) we have stated that an error pattern of weight $t=2$, such as $\{e_i\} = \{e_0=0, e_1 = x\}$, is not 1-step PD if

$$n-k \le x \le k. \qquad (4.12)$$

Accordingly, in Equation (3.26) the $\{e_i\}$ pattern, is not 2-step PD if

$$\begin{cases} 2x-n \leq k \\ n-(2x-n) \leq k \end{cases} \tag{4.13}$$

From Equations (4.12), (4.13), and for the case of $n = \frac{3k_0-1}{2}$ for $k_0 > 3$, we get

$$\frac{2n-k_0}{2} \leq x \leq k_0 \ ,$$

or

$$k_0 - \frac{1}{2} \leq x \leq k_0 \ ,$$

this implies that

$$x = k_0 \tag{4.14}$$

That is, the only error pattern which is not 2-step PD is $\{e_i\} = \{0,k_0\}$ in R_0-domain. Hence, the corresponding error pattern in R_1-domain is

$$\{e_i'\} = \begin{cases} e_0' = e_0 = 0, \\ e_1' = 2x-n = \frac{k_0+1}{2} \end{cases} \tag{4.15}$$

Similarly, the corresponding pattern in R_2-domain is

$$\{e_i''\} = \begin{cases} e_0'' = e_0 = 0 \\ e_1'' = 2e_1' = k_0+1. \end{cases} \qquad (4.16)$$

From Equation (4.16) we conclude that the pattern is 3-step PD.

Q.E.D.

THEOREM 4.5

The $(n,k,2)$ codes with $t_e = 2$ and $n = \frac{3k}{2} - \ell$, $1 \leq \ell < \frac{k}{2}$, are not 3-step PD.

Proof of Theorem 4.5

From Equations (4.12) and (4.13) in Theorem 4.4, we can restate that, for an $(n,k,2)$ code eith $n = \frac{3k}{2} - \ell$, $1 \leq \ell < \frac{k}{2}$, there can be an EP of weight $t=2$ such as $\{e_i\} = \{e_0=0,\ e_1=x\}$ for the case of

$$k-\ell = \frac{2n-k}{2} \leq x \leq \text{[illegible]}, \qquad (4.17)$$

which is not 2-step PD. Let us assume

$$x = k-\ell > 0\ , \qquad (4.18)$$

in R_0-domain. Then, according to Equation (4.17), the $\{e_i\}$ pattern is not 2-step PD. Now, considerino the corresponding error pattern in R_2-domain, $\{e_i''\}$, we have:

$$\{e_i''\} = \begin{cases} e_0'' = 4\cdot e_0 = 0, \\ \\ e_1'' = 2\cdot(2x-n) = k-2\ell > 0, \end{cases} \qquad (4.19)$$

where $e_1' = 2x-n = \frac{k}{2} - \ell$ is the corresponding error location of $e_1=x$ in R_1-domain. From Equation (4.19) we have:

$$\begin{cases} \text{first gap} = k-2\ell-1 < k, \\ \text{last gap} = n-(k-2\ell)-1 = \frac{k}{2} + \ell-1 < k, \end{cases} \qquad (4.20)$$

in R_2-domain. Thus, the $\{e_i\}$ pattern is not 3-step PD.

Q.E.D.

In establishing the bounds on n, for the codes with even t, the following corollaries may be applied.

COROLLARY 4.6

The (n,k_o,t_e) codes with $n = k_o\cdot(t_e-1)$, and the (n,k_e,t_e) codes with $n = (t_e-1)(k_e-1)$, are not 3-step PD.

COROLLARY 4.7

The (n,k_o,t_e) codes with $n = \frac{3}{4}\cdot t_e\cdot(k_o-1)$, $t_e\cdot(k_o-1) \not\equiv 0 \bmod 8$ and the (n,k_e,t_e) codes with $n = \frac{3}{4}\cdot t_e\cdot k_e$, $t_e\cdot k_e \not\equiv 0 \bmod 8$, are not 3-step PD.

The proofs of Corollaries 4.6 and 4.7 are not given here since these corollaries are special cases of Corollary 5.2, Theorem 5.3, and

Corollary 5.3 which will be given in Chapter 5.

The codes which have not been discussed so far, are considered in the following section as "special cases".

4.3 Special Cases

The next results are for the codes with $k = 2$.

THEOREM 4.6

The (n,k,t_0) codes, with $n = \frac{3t_0-1}{2}$, $\frac{t_0-1}{2}$ even, and for $k = 2$ are 3-step PD.

Proof of Theorem 4.6

According to the different types of patterns α,β in the EP given in Table 3-3, the number of α-type patterns, for $n = \frac{3t_0-1}{2}$ and $k = 2$, can, in the worst case, be obtained as

$$N_\alpha + t_0 > n = \frac{3t_0-1}{2},$$

or

$$N_\alpha \geq \frac{t_0-1}{2}, \tag{4.21}$$

in R_1-domain. Here, each α-type gap is of length 1. On the other hand, if an EP is not decodable in a given domain, then there should at most be a number of $(\frac{t_0-1}{2})$-gaps each of which is of length 1. Therefore, in R_1-domain,

$$N_\alpha \leq \frac{t_0-1}{2}. \tag{4.22}$$

From Equations (4.21) and (4.22), it can be concluded that

$$N_\alpha = \frac{t_o-1}{2} . \tag{4.23}$$

This implies that for N_α as given in Equation (4.23), all the EPs which are not 1-step or 2-step PD are of the form

$$\{e_i\} = \begin{cases} e_0 = 0, \\ e_{2i+1} = e_{2i}+1 \quad , & \text{for } 0 \le i \le \frac{t_o-3}{2} \\ e_{2i+2} = e_{2i+1}+2, \end{cases}$$

in R_1-domain, or are the cyclic shifts of this pattern.

Now, considering the EP corresponding to $\{e_i\}$ in R_2-domain, we have

$$\{e_i'\} = \begin{cases} e_0' = e_0 = 0, \\ e_{2i+2}' = 2\cdot e_{i+1}, & \text{for } 0 \le i \le \frac{t_o-3}{2} . \\ e_{2i+1}' = (2\cdot e_{i+\frac{t_o+1}{2}}) \bmod n, \end{cases} \tag{4.24}$$

From Equation (4.24), it can be verified that

$$e_{4j-1}' - e_{4j-2}' = 3, \quad \text{for } 1 \le j \le \frac{t_o-1}{2} .$$

This implies that the pattern $\{e_i\}$ and the cyclic shifts of $\{e_i\}$ are decodable in R_2-domain.

Q.E.D.

THEOREM 4.7

The (n,k,t_o) codes with $n \le \frac{3t_o-3}{2}$ and $k = 2$, are not 3-step PD.

Proof of Theorem 4.7

To prove this theorem, we consider the following two lemmas and one corollary.

LEMMA 4.3

The $(n,2,t_o)$ codes, with $n = \frac{3t_o-5}{2}$, $\frac{t_o-1}{2}$ even, $t_o > 5$, are not 3-step PD.

LEMMA 4.4

The $(n,2,t_o)$ codes, with $n = \frac{3t_o-7}{2}$, $\frac{t_o-1}{2}$ odd, and $t_o > 7$, are not 3-step PD.

Proof of Lemma 4.3

We prove this lemma by introducing an EP in R_2-domain which is not 3-step PD.

In particular, consider the following EP

$$\{e_i\} = \begin{cases} e_0 = 0, \\ e_1 = 2, \\ e_i = e_{i-1}+1, & \text{for } 2 \le i \le 5 , \\ \left.\begin{array}{l} e_{2i+4} = e_{2i+3}+2, \\ e_{2i+5} = e_{2i+4}+1, \end{array}\right\} & \text{for } 1 \le i \le \frac{t_o-9}{2} \text{ and } t_o \ge 13, \\ e_{t_o-4+i} = e_{t_o-5+i}, & \text{for } 1 \le i \le 2 \\ e_{t_o-1} = e_{t_o-2}+2 , \end{cases} \tag{4.25}$$

in R_2-domain. The pattern $\{e_i\}$ in Equation (4.25) has no gap-lengths greater than unity. The end-gap is

$$EG(n,e_{t_o-1}) = \frac{3t_o-5}{2} - [2+4+3\cdot(\frac{t_o-9}{2})+2+2] - 1 = 0$$

Therefore, the EP is not decodable. Thus, the corresponding EP in R_1-domain becomes

$$\{e_i'\} = \begin{cases} e_0' = 0, & \\ e_i' = e_{i-1}'+1, & \text{for } 1 \le i \le 4 \\ \left.\begin{array}{l} e_{2i+3}' = e_{2i+2}'+2, \\ e_{2i+4}' = e_{2i+3}'+1, \end{array}\right\} & \text{for } 1 \le i \le \frac{t_o-9}{4} \text{ and } t_o \ge 13 \\ e_{\frac{t_o+1}{2}}' = e_{\frac{t_o-1}{2}}'+1, & \\ e_{\frac{t_o+3}{2}}' = e_{\frac{t_o+1}{2}}'+2, & \\ e_{\frac{t_o+5}{2}}' = e_{\frac{t_o+3}{2}}'+1, & \\ \left.\begin{array}{l} e'_{2i+\frac{t_o+3}{2}} = e'_{2i+\frac{t_o+1}{2}}+2, \\ e'_{2i+\frac{t_o+5}{2}} = e'_{2i+\frac{t_o+3}{2}}+1; \end{array}\right\} & \text{for } 1 \le i \le \frac{t_o-9}{2} \text{ and } t_o \ge 13 \\ e_{t_o-1}' = e_{t_o-2}'+1. & \end{cases} \quad (4.26)$$

Therefore, all the gap-lengths between any pair of consecutive errors in $\{e'_1\}$ are also ≤ 1. The end-gap is

$$EG(n,e'_{t_o-1}) = \frac{3t_o-5}{2} - [4+6 \cdot (\frac{t_o-9}{4}) + 5] - 1 = 1.$$

Thus, the pattern $\{e'_i\}$ is not decodable. Finally, the corresponding EP in R_0-domain can be obtained from $\{e'_i\}$ as given in Table 4-3.

TABLE 4.3

Gap-lengths in R_0-domain associated with any two consecutive errors with the same parity in the R_1-domain

Distances in R_1-domain	Corresponding gap-lengths in R_0-domain
$D(e'_{2i+4},e'_{2i+1}) = 4$, for $1 \leq i \leq \frac{t_o-9}{4}$ and $t_o \geq 13$	$= 1$
$D(e'_{2i+\frac{t_o-5}{2}},e'_{2i+\frac{t_o-1}{2}}) = 4$	$= 1$
$D(e'_{\frac{t_o+5}{2}},e'_{\frac{t_o-1}{2}}) = 4$	$= 1$
$D(e_{t_o-2},e_1) = D(\text{first NF, last F}) = \frac{n-(e_{t_o-2},e_1)}{2} - 1 = 2$	$= 0$ (central-gap)
$D(n,e_{t_o-1}) = 2$	$= 0$ (end-gap)

From Table 4-3, a total of $\frac{t_0-5}{2}$ gaps are of length 1, and the rest of the gaps are of the length 0, resulting in $n = t_0 + \frac{t_0-5}{2} = \frac{3t_0-5}{2}$. Thus, the pattern $\{e_i\}$ is not decodable in R_0-domain, and it is not 3-step PD. This completes the proof of Lemma 4.3.

Proof of Lemma 4.4

To prove this lemma, we follow the same steps as in the proof of Lemma 4.3. Here, we consider the EP

$$\{e_i\} = \begin{cases} e_0 = 0, & \\ e_1 = 2, & \\ e_i = e_{i-1}+1, & \text{, for } 2 \le i \le 5 \text{ ,} \\ \left.\begin{array}{l} e_{2i+4} = e_{2i+3}+2, \\ \\ e_{2i+5} = e_{2i+4}+1, \end{array}\right\} & \text{, for } 1 \le i \le \frac{t_0-9}{2} \text{ and } t_0 \ge 11, \\ e_{t_0-4+i} = e_{t_0-5+i}+1 & \text{, for } 1 \le i \le 3, \end{cases} \tag{4.27}$$

in R_2-domain. The pattern $\{e_i\}$ in Equation (4.27) has no gap-lengths greater than unity. The end-gap is

$$EG(n,e_{t_0-1}) = \frac{3t_0-7}{2} - (2+4+3\cdot(\frac{t_0-9}{2})+3) - 1 = 0.$$

Therefore, the EP is not decodable. Hereafter, the derivation of the corresponding EPs in R_1- and R_0-domains is the same as that in Lemma 4.3. Thus, it can easily be verified that the resulting EPs in these domains are not decodable. This completes the proof of Lemma 4.4.

COROLLARY 4.8

The $(n,2,t_o)$ codes, with $n = \frac{3t_o-3}{2}$, $\frac{t_o-1}{2}$ being odd; with $n = \frac{3t_o-9}{2}$, $\frac{t_o-1}{2}$ being even and $t_o > 9$, are not 3-step PD.

Proof of Corollary 4.8

According to Corollaries 4.1 and 4.4, the codes with $n = \frac{3t_o-3}{2} = \frac{3\cdot 2\cdot(t_e=t_o-1)}{4}$, and those with $n = \frac{3t_o-9}{2} = \frac{3\cdot 2\cdot(t_e=t_o-3)}{4}$, are not 3-step PD.

Note that for the $(n,2,t_o)$ codes; with $n = \frac{3t_o-5}{2}$ and $t_o=5$; with $n = \frac{3t_o-7}{2}$ and $t_o=7$; and with $n = \frac{3t_o-9}{2}$ and $t_o=9$, one has $n=t_o$. This is a special case of Theorem 5.1 for $k_o=1$ which will be given in Chapter 5.

Thus, the proof of Theroem 4.7 follows from Lemma 4.3 and Corollaries 4.2 and 4.8 for the codes with $n \leq \frac{3t_o-3}{2}$ and $\frac{t_o-1}{2}$ being even, and from Lemma 4.4 and Corollaries 4.2 and 4.8 for the codes with $n \leq \frac{3t_o-3}{2}$ and $\frac{t_o-1}{2}$ being odd.

Q.E.D,

4.4 Summary and Numerical Results

In this chapter we studied 3-step permutation decodable cyclic codes. In this direction, the exact lower bounds on the code length n, of the (n,k,t) codes with t odd and the case of $t=2$ were obtained. A combined summary and numerical results of this chapter are given below.

4.1.1 Main Results

In addition to the improved bounds given in the previous chapter, for 2-step PD codes, Theorem 4.1,4.3,4.6,4.7 assert that the following codes with t odd and k being odd or even to be 3-step PD. Theorems 4.2 and 4.7 and Corollaries 4.1 to 4.8 assert that the rest of the codes remain under the same bounds given for 2-step PD.

4.4.1a. For t odd-valued (t_0)

i) k odd-valued (n,k_0,t_0) codes:

$$n = t_0 \cdot k_0 - 2(2.\ell+1), \quad \text{if } k_0=2\ell+3,$$
$$\text{and } 0 \leq \ell < (t_0-1)/2 . \tag{4.28}$$

ii) k even-valued (n,k_e,t_0) codes:

$$n = t_0 \cdot (k_e-1) - 2(2.\ell+1) \quad \text{if } k_e > 2.\ell+4,$$
$$\text{and } 0 \leq \ell < (t_0-1)/2. \tag{4.29}$$

According to Theorem 4.4, the following double-error-correcting codes are 3-step PD.

4.4.1b For t=2

k odd-valued $(n,k_0,2)$ codes:

$$n = k_0 + \frac{k_0-1}{2} \quad , \quad \text{for} \quad k_0 > 3. \tag{4.30}$$

Finally, the exceptional case which is not considered in Section 4.4.1a is the case of t odd and $k=2$. Theorems 4.6 and 4.7 show the following codes to be 3-step PD.

4.4.1c <u>Special Case</u>:

For $k=2$, $(n,2,t_o)$ codes:

$$n \geq (3 \cdot t_o - 1)/2. \tag{4.31}$$

Note that the above lower bounds on n are an improvement over the corresponding bounds for 2-step decodable codes.

4.4.3. <u>Numerical Results</u>

Based on the results obtained in this chapter, we give three Tables 4-4, 4-5 and 4-6, for some specific numbers of correctable errors $t_o=5,9,$ and $t_e=2$ as examples. These tables show the exact lower bounds on code length n, for a given information length k, for 3-step PD codes.

TABLE 4-4

3-step PD codes of length n, and $t_0=5$

k =	2	3	4	5	6	8	10	12	14	16	18	20	22	. . .
n The Code Length	X	13	X	19	X	29	39	49	59	69	79	89	99	
	7	--	X	--	23	33	43	53	63	73	83	98	103	
	9	--	17	--	27	37	47	57	67	77	87	97	107	
	11	--	21	--	31	41	51	61	71	81	91	101	111	
	15	--	25	--	35	45	55	65	75	85	95	105	115	

TABLE 4-5

3-step PD codes of length n, and $t_0=9$

k =	2	3	4	5	6	7	8	9	10	12	14	16	18	. . .
n The Code Length	X	25	X	39	X	53	X	67	X	85	103	121	139	
	X	--	X	--	X	--	X	--	71	89	107	125	143	
	13	--	29	--	43	--	57	--	75	98	111	129	147	
	15	--	31	--	47	--	61	--	79	97	115	133	151	
	17	--	33	--	49	--	65	--	83	101	119	137	155	
	19	--	35	--	51	--	69	--	87	105	123	141	159	
	21	--	37	--	55	--	73	--	91	109	127	145	163	
	23	--	41	--	59	--	77	--	95	113	131	149	167	
	27	--	45	--	63	--	81	--	99	117	135	153	171	

TABLE 4-6

3-step RD codes of length n, and $t_e=2$

k =	3	5	7	9	11	13	15	17	19	21	23	25	...
n	5	7	11	13	17	19	23	25	29	31	35	37	
The Code Length	--	9	--	15	--	21	--	27	--	33	--	39	

CHAPTER 5

CERTAIN ~~NON~~ PERMUTATION DECODABLE CYCLIC CODES

5.1 Introduction

This chapter deals with some general results based on certain cyclic codes which are not PD. When we say an (n,k,t) cyclic code is not PD, it implies that there will be at least one EP which can not be permutation decoded.

The results of this chapter are general and can be applied to all group (T,U) permutations and they are used in establishing the exact lower bounds on the code length n, of an (n,k,t) cyclic code. Section 5.2 presents these results as "certain non PD cyclic codes". In Section 5.3, we summarize the results that are obtained in this Chapter.

5.2 Certain Non-PD Codes

Here, we give the following three general theorems and related corollaries for both values of t being odd or even, and also for both values of k being odd or even.

THEOREM 5.1

The (n,k_0,t_0) codes with $n=t_0 \cdot k_0$, are not PD.

Proof of Theorem 5.1:

This theorem will be proven by considering two different EPs, of weight t_0 or less, which are not PD in any R_i-domain, $i\geq 0$

Specially, let us consider the EPs:

$$\{e_i\} = \{e_i = i\cdot k_0, \quad \text{for } i=0,1,2,\ldots, t_0-1\},$$

and (5.1)

$$\{\varepsilon_i\} = \{\varepsilon_i = i\cdot t_0, \quad \text{for } i=0,1,2,\ldots, k_0-1 \text{ and } k_0\geq t_0\},$$

in R_0-domain, for $n=t_0\cdot k_0$. With reference to their definitions, these two EPs, are not 1-step PD.

Now, the EPs $\{e_i'\}$ and $\{\varepsilon_i'\}$ in R_1-domain, which correspond, respectively, in R_0-domain, to the EPs = $\{e_i\}$ and $\{\varepsilon_i\}$ are:

$$\{e_i'\} = \begin{cases} e_i' = 2e_i = 2\cdot i\cdot k_0\,, & \text{for } 0\leq i\leq \frac{t_0-1}{2}, \\ e_i' = 2e_i \bmod n = 2e_i-t_0\cdot k_0 & \\ \qquad = (2i-t_0)\cdot k_0, & \text{for } \frac{t_0+1}{2}\leq i\leq t_0-1, \end{cases}$$

and

$$\{\varepsilon_i'\} = \begin{cases} \varepsilon_i' = 2\varepsilon_i = 2\cdot i\cdot t_0\,, & \text{for } 0\leq i\leq \frac{k_0-1}{2}\,, \\ \varepsilon_i' = 2\varepsilon_i \bmod n = 2\varepsilon_i-t_0\cdot k_0 & \\ \qquad = (2i-k_0)\cdot t_0, & \text{for } \frac{k_0+1}{2}\leq i\leq k_0-1 \text{ and } k_0\leq t_0. \end{cases}$$

That is:

$$\{e'_i = i \cdot k_o, \quad \text{for } i=0,1, \ ,\ldots, \ t_o-1\} = \{e_i\} ,$$

and

$$\{\varepsilon'_i = i \cdot t_o \ , \quad \text{for } i=0,1, \ \ldots, \ k_o-1 \ \text{ and } \ k_o \geq t_o\} = \{\varepsilon_i\}.$$

Thus, by squaring (higher power of group (U) permutations) the $\{e'_i\}$ and $\{\varepsilon'_i\}$ successively, the resulting EPs do not differ from those in Equation (5.1)*. Hence these patterns are not PD.

Q.E.D.

Theorem 5.1 leads to the following two corollaries:

<u>COROLLARY 5.1</u>

The (n,k_o,t_o) codes with $n=\tau_o \cdot k_o$ for $\tau_o \leq t_o$, and the codes with $n=\delta_o \cdot t_o$ for $\delta_o \leq k_o$, are not PD.

<u>COROLLARY 5.2</u>

The (n,k_e,t_o) codes with $n=t_o \cdot (k_e-1)$, the (n,k_o,t_e) codes with $n=k_o \cdot (t_e-1)$, and the (n,k_e,t_e) codes with $n=(k_e-1)(t_e-1)$, are not PD.

<u>Proof of Corollaries 5.1 and 5.2</u>:

These Corollaries are a logical consequence of Theorem 5.1. This is so because if a (n,k,t) code is not PD, then by replacing k with $k+1$, or t with $t+1$ (or both), the resulting code is obviously not PD.

* This operation can be identified as idempotent in literature.

THEOREM 5.2

If the (n,k,t) code is not s-step PD for some $s<\nu$, then the (n',k,t) codes with $n'=n-2^{s-1}\cdot\ell$ for $\ell\geq 1$ are not s-step PD.

Proof of Theorem 5.2:

Let $\{e_i^{s-2};\ i=0,1,\ldots,t-1\}$ be an EP in the R_{s-2}-domain of the (n,k,t) code, which is not 1-, 2-,..., or s-step PD. Now, let us assume that $\{\varepsilon_i^{s-2};\ i=0,1,\ldots,t-1\}$ is an EP in R_{s-2}-domain for the (n',k,t) code, with $n'=n-2^{s-1}$ (i.e. $\ell=1$) w.r.t. the pattern $\{e_i^{s-2}\}$ defined by:

$$\{\varepsilon_i^{s-2}] = \begin{cases} \varepsilon_0^{s-2} = e_0^{s-2} = 0, & \\ \varepsilon_i^{s-2} = e_i^{s-2} & \text{for } e_i^{s-2} \leq \frac{n'-1}{2} \qquad \text{(a)}, \\ \varepsilon_i^{s-2} = e_i^{s-2} & \text{for } \frac{n'-1}{2} < e_i^{s-2} \leq \frac{n-1}{2} \qquad \text{(b)}, \\ \varepsilon_i^{s-2} = e_i^{s-2} - 2^{s-2} & \text{for } \frac{n-1}{2} < e_i^{s-2} < n-2^{s-2} \qquad \text{(c)}, \\ \varepsilon_i^{s-2} = e_i^{s-2} - 2^{s-1} & \text{for } n-2^{s-2} < e_i^{s-2} < n \qquad \text{(d)}. \end{cases} \tag{5.2}$$

In Equation (5.2), (a),(b),(c), and (d) are used to specify the various categories of error positions associated with e_i^{s-2}-values in R_{s-2}-domain. From Equation (5.2), we can define a difference vector $\{\delta\varepsilon_i^{s-2}\}$ in R_{s-2}-domain as:

$$\{\delta\varepsilon_i^{s-2}\} = \begin{cases} \delta\varepsilon_i^{s-2} = 0 & \text{if } e_i^{s-2} \leq \frac{n-1}{2}, \\ \delta\varepsilon_i^{s-2} = 2^{s-2} & \text{if } \frac{n-1}{2} < e_i^{s-2} < n-2^{s-2}, \\ \delta\varepsilon_i^{s-2} = 2^{s-1} & \text{if } n-2^{s-2} < e_i^{s-2} < n, \end{cases} \tag{5.3}$$

which represents the difference between the errors $\{e_i^{s-2}\}$ and $\{\varepsilon_i^{s-2}\}$, that is $\delta\varepsilon_i^{s-2} = e_i^{s-2} - \varepsilon_i^{s-2}$ for $i=0,1, \ldots, t-1$.

Thus, the relationship between the difference vectors $\{\delta\varepsilon_i^{s-j}\}$, (associated with R_{s-j}-domain) and the difference vectors $\{\delta\varepsilon_i^{s-j+1}\}$, (associated with R_{s-j+1}-domain), for $3 \leq j \leq s$ and $s \geq 3$, can be obtained as:

$$\{\delta\varepsilon_i^{s-j}\} = \begin{cases} \delta\varepsilon_0^{s-j} = 0 \\ \delta\varepsilon_i^{s-j} = \dfrac{\delta\varepsilon_i^{s-j+1}}{2} + 2^{s-2} & \text{for odd-valued } \varepsilon_i^{s-j+1}(e_i^{s-j+1}), \\ \delta\varepsilon_i^{s-j} = \dfrac{\delta\varepsilon_i^{s-j+1}}{2.} & \text{for even-valued } \varepsilon_i^{s-j+1}(e_i^{s-j+1}). \end{cases} \tag{5.4}$$

Note that for odd-valued ε_i^{s-j} or e_i^{s-j} we have:

$$\varepsilon_i^{s-j} = \frac{\varepsilon_i^{s-j+1}+n'}{2} \quad \text{and} \quad e_i^{s-j} = \frac{e_i^{s-j+1}+n}{2}, \quad \text{and thus}$$

$$\delta\varepsilon_i^{s-j} = e_i^{s-j} - \varepsilon_i^{s-j} = \frac{\delta\varepsilon_i^{s-j+1}}{2} + 2^{s-2}.$$

To prove the theorem, it should be shown that there will be no gap increase in any $\{\varepsilon_i^{s-j}\}$ w.r.t. $\{e_i^{s-j}\}$ in R_{s-j}-domain for $1 \leq j \leq s$. This can be established using the following two lemmas:

LEMMA 5.1

The pattern $\{\varepsilon_i^{s-2}\}$ given in Equation (5.2) is not s-step PD if there are no errors located in $\{e_i^{s-2}\}$ in the intervals $(n'+1)/2 \leq e_i^{s-2} \leq (n-1)/2$ and $n > e_{t-1}^{s-2} > n-2^{s-2}$ for $i \leq t-1$.

We shall assume that the conditions of Lemma 5.1 are satisfied, that is, by the use of proper cyclic shifts. On the other hand, if these conditions are not satisfied, we have the following lemma.

LEMMA 5.2

The $\{\varepsilon_i^{s-2}\}$ defined in Equation (5.2) is not PD even if there are errors located in $\{e_i^{s-2}\}$ in either or in both intervals $(n'+1)/2 \leq e_i^{s-2} \leq (n-1)/2$ or $n > e_i^{s-2} > n-2^{s-2}$, for $i \leq t-1$.

Proofs of the following lemmas are rather long. This is partly because there are many cases to be considered.

Proof of Lemma 5.1:

If the conditions of the lemma are satisfied, then $\{\varepsilon_i^{s-2}\}$ given by Equation (5.2) reduces to

$$\{\varepsilon_i^{s-2}\} = \begin{cases} \varepsilon_0^{s-2} = e_0^{s-2} = 0, & \\ \varepsilon_i^{s-2} = e_i^{s-2} & \text{for } e_i^{s-2} \leq \frac{n'-1}{2} \quad \text{(a)} \\ \varepsilon_i^{s-2} = e_i^{s-2} - 2^{s-2} & \text{for } \frac{n-1}{2} < e_i^{s-2} < n \quad \text{(c)} \end{cases} \qquad (5.5)$$

Now, we prove that $\{\varepsilon_i^{s-2}\}$ will not be decodable in R_{s-2}-domain. This can be shown as follows:

(1) by assumption $\{e_i^{s-2}\}$ is not decodable;

(2) in accordance with the definition of $\{\varepsilon_i^{s-2}\}$ in Equation (5.5), the distances between any two consecutive pairs of errors with the same parity (i.e., both F or NF) from

either category (a) or category (c) w.r.t. $\{e_i^{s-2}\}$ are exactly the same; and

(3) the distance between the last odd-valued (NF) and even-valued (F) error positions of category (a), and the first NF and F error positions from categroy (c) have reduced by 2^{s-2} in comparison with the corresponding distances of the error positions in $\{e_i^{s-2}\}$.

To be more specific, we derive the following gap relationships between $\{\varepsilon_i^{s-2}\}$ and $\{e_i^{s-2}\}$ in R_{s-2}-domain:

$$\left\{\begin{array}{l}
\underset{\{\varepsilon_i^{s-2}\}}{D[LNF^a(LF^a),FNF^c(FF^c)]} = \underset{\{e_i^{s-2}\}}{D[LNF^a(LF^a),FNF^c(FF^c)]} - 2^{s-2}, \\
\underset{\{\varepsilon_i^{s-2}\}}{D[FNF^a,LF^c]} = \underset{\{e_i^{s-2}\}}{D[FNF^a,LF^c]} - 2^{s-2}, \\
EG(n',\varepsilon_{t-1}^{s-2}) = EG(n,e_{t-1}^{s-2}) - 2^{s-2};
\end{array}\right. \qquad (5.6)$$

where LNF (LF) and FNF (FF) denote the last NF (F) and the first NF (F) for a given category*, respectively. It should be noted that the gaps which have not been included in these expressions have the same lengths as those of their corresponding gaps in $\{e_i^{s-2}\}$. It should further be noted that there will be no interchange of error

[*] Superscripts a, and c show the error positions for categories (a), and (c) respectively.

positions in $\{\varepsilon_i^{s-2}\}$ w.r.t. $\{e_i^{s-2}\}$, since all the error positions in $\{\varepsilon_i^{s-2}\}$ from category (c) are greater than $[(n-1)/2] - 2^{s-2} = (n'-1)/2$. It should finally be noted that from Equation (5.6), $\{\varepsilon_i^{s-2}\}$ is not decodable in R_{s-2}-domain.

Now, let us consider the corresponding patterns in R_{s-1}-domain which are $\{e_i^{s-1} = 2\cdot e_i^{s-2} \bmod n\}$, and $\{\varepsilon_i^{s-1} = 2\,\varepsilon_i^{s-2} \bmod n'\}$, for $i=0,1,2,\ldots, t-1$. Then, from Equation (5.5) we get:

$$\{\varepsilon_i^{s-1}\} = \begin{cases} 2\,\varepsilon_i^{s-2} = 2\,e_i^{s-2} & \text{for} \quad e_i^{s-2} \le \dfrac{n'-1}{2}\,, \\ 2\,\varepsilon_i^{s-2} \bmod n' = 2\,\varepsilon_i^{s-2} - (n-2^{s-1}) = 2(\varepsilon_i^{s-2} + 2^{s-2}) - n & \\ \qquad = 2\,e_i^{s-2} - n = 2\,e_i^{s-2} \bmod n & \\ & \text{for} \quad e_i^{s-2} > \dfrac{n-1}{2}\,. \end{cases} \tag{5.7}$$

From Equation (5.7) we can conclude that in R_{s-1}-domain, $\{\varepsilon_i^{s-1}\}$ is exactly the same as $\{e_i^{s-1}\}$ which by assumption is not PD.

Now, consider $\{e_i^{s-j}\}$ and $\{\varepsilon_i^{s-j}\}$ the corresponding patterns from R_{s-j}- to R_0-domains for $3 \le j \le s$. These patterns, and consequently the difference vectors in R_{s-j}-domains, can be obtained from Equation (5.4). For s=2, R_{s-2}-domain is R_0-domain, and thus the lemma is proven for this case.

For the case $s > j = 3$, in accordance with Equation (5.6), some of the gaps in $\{\varepsilon_i^{s-3}\}$ have the same lengths as those in $\{e_i^{s-3}\}$, with the latter gaps corresponding to consecutive error positions having the same parity and with the same category in $\{\varepsilon_i^{s-2}\}$ w.r.t. $\{e_i^{s-2}\}$. In

R_{s-3}-domain, the gaps corresponding to the first expression in Equation (5.6), will produce two gap-length decrease, i.e., by an amount $2^{s-2}/2 = 2^{s-3}$, as do the gaps corresponding to the second expression. Moreover, the end-gap in R_{s-3}-domain, which corresponds to the last NF error position in R_{s-2}-domain, can be obtained as:

$$EG(n',\varepsilon_{t-1}^{s-3}) = n' - \varepsilon_{t-1}^{s-3} - 1 = EG(n,e_{t-1}^{s-3}) - (2^{s-2} - \delta\varepsilon_{t-1}^{s-3}) \quad (5.8)$$

Where $\delta\varepsilon_{t-1}^{s-3} = 2^{s-3}+2^{s-2}$. Thus, this end-gap is also reduced by an amount 2^{s-3}. This implies that all the elements of $\{\delta\varepsilon_i^{s-3}\}$, for $s>3$, are of even values. Therefore, the parity of $\{\varepsilon_i^{s-3}\}$ remains the same, and no gap-length increase occurs w.r.t. $\{e_i^{s-2}\}$. Hence, if $\{e_i^{s-3}\}$ is not PD, then $\{\varepsilon_i^{s-3}\}$ cannot be decodable in R_{s-3}-domain.

So far, we have shown that some gap-length reductions in R_{s-2}-domain are of the order 2^{s-2}, and that the gap-length reductions in R_{s-3}-domain are of the order $2^{s-2}/2 = 2^{s-3}$. Proceeding in this way, we can conclude that for the case $s>j>3$, the gap-length reductions in each R_{s-j}-domain will be of the order $2^{s-j+1}/2 = 2^{s-j}$. Consequently, the elements of the difference vector in each domain are of even values, and there will be no exchange of error positions and no change of parities in $\{\varepsilon_i^{s-j}\}$ w.r.t. $\{e_i^{s-j}\}$ for $s>j>3$.

It should be pointed out that for each reduced gap-length in the "present" R_{s-j+1}-domain, there will be two gap-length reductions of the order 2^{s-j} (two consecutive F or NF error positions surrounding the reduced gap by 2^{s-j+1} in the present domain) in the "previous" R_{s-j}-domain; and that it is possible that some of the gap-lengths are reduced more than once. However, this does not imply that there will be some exchange of error positions in the pattern.

In general, the end-gap in the previous R_{s-j}-domain corresponds to the last NF error position in the present R_{s-j+1}-domain. Thus, from Equation (5.8) we get:

$$EG(n'-\varepsilon_{t-1}^{s-j}) = n' - \varepsilon_{t-1}^{s-j} - 1 = EG(n,e_{t-1}^{s-j}) - (2^{s-1} - \delta\varepsilon_{t-1}^{s-j})$$

where $\delta\varepsilon_i^{s-j} < 2^{s-1}$. Then, for $s>j\geq 3$, none of $\{\varepsilon_i^{s-j}\}$ will be PD in any R_{s-j}-domain.

Finally, if we consider $s = j \geq 3$, then the pattern in R_{s-j} ($= R_0$)-domain can be derived from the corresponding pattern in R_1-domain. Note that $\{\varepsilon_i^1\}$ has the same arrangement as $\{e_i^1\}$, and both these patterns have the same parity of error positions, but their differences lie in some gap-length reductions of order 2 or multiples of 2 only. Hence, the corresponding pattern $\{\varepsilon_i^0\}$ in R_0-domain has the same arrangement as $\{\varepsilon_i^0\}$. Note that in R_0-domain the elements of $\{\delta\varepsilon_i^0\}$ are not necessarily of even values, and, based on $\{\varepsilon_i^1\}$, there will be no increase of gap lengths.

Thus, in the patterns $\{\varepsilon_i^{s-j}\}$, for $3\leq j\leq s$, there will be no gap-length increase w.r.t. $\{e_i^{s-j}\}$. Hence, if $\{e_i^{s-j}\}$ is not decodable in any R_{s-j}, then $\{e_i^{s-j}\}$ cannot be decodable in any R_{s-j}-domain, for $j = 1,2,\ldots, s$. This completes the proof of Lemma 5.1.

<u>Proof of Lemma 5.2</u>

Based on the structure of the error patterns given in Equation (5.2) and the inclusion of errors in categories (b) and (d) in $\{\varepsilon_i^{s-2}\}$ and $\{e_i^{s-2}\}$, it can easily be shown from the proof of Lemma 5.1 that $\{\varepsilon_i^{s-2}\}$

is not decodable. As we have shown in Equation (5.7) the corresponding pattern $\{\varepsilon_i^{s-1}\}$ in R_{s-1}-domain is such that the error positions from categories (a) and (c) of $\{\varepsilon_i^{s-2}\}$ correspond exactly to those of $\{e_i^{s-1}\}$ which are located before the code length n'. This is so because in R_{s-1}-domain all the errors located in $\{e_i^{s-2}\}$ in the intervals $(n'-1)/2 < e_i^{s-2} \leq (n-1)/2$ and $n-2^{s-2} < e_i^{s-2} < n'$ are; $2e_i^{s-2} \bmod n > n'$. Thus, in R_{s-1}-domain, the errors in $\{\varepsilon_i^{s-2}\}$ associated with categories (b) and (d) give rise to some additional errors between the gaps produced by the errors associated with categories (a) and (c). Therefore, the patterns $\{\varepsilon_i^{s-1}\}$ will not be decodable in R_{s-1}-domain.

Now, we proceed to consider the corresponding EPs in R_{s-j}-to R_0-domains for $3\leq j\leq s$. Note that for $s=2$, R_{s-2}-domain is R_0-domain, and thus the lemma is proven for this case.

For the general case, considering $\{\varepsilon_i^{s-2}\}$, in Equation (5.2) two cases can occur:

i) If there are no interchange of error positions either from category (c) to (b), or from category (d) to (c),

ii) If there is an interchange of error positions either from category (c) to (b), or category (d) to (c) (or both). We investigate these two cases in R_{s-j}-domains for $3\leq j\leq s$, as follows:

Case i:

In this case, all the errors in $\{\varepsilon_i^{s-2}\}$ associated with categories (c) and (d) are greater than those of categories (b) and (c), respectively. Therefore, expressions similar to Equation (5.6) can be derived (assuming that there are errors from both categories (b) and (d)) in

R_{s-2}-domain as;

$$\begin{cases} \underset{\{\varepsilon_i^{s-2}\}}{D[LNF^b(LF^b),FNF^c(FF^c)]} = \underset{\{e_i^{s-2}\}}{D[LNF^b(LF^b),FNF^c(FF^c)]} - 2^{s-2}. \\ \underset{\{\varepsilon_i^{s-2}\}}{D[LNF^c(LF^c),FNF^d(FF^d)]} = \underset{\{e_i^{s-2}\}}{D[LNF^c(LF^c),FNF^d(FF^d)]} - 2^{s-2}, \\ \underset{\{\varepsilon_i^{s-2}\}}{D[FNF^a,LF^d]} = \underset{\{e_i^{s-2}\}}{D[FNF^a,LF^d]} - 2^{s-1}, \\ EG(n';\varepsilon_{t-1}^{s-2}) = EG(n,e_{t-1}^{s-2}). \end{cases} \qquad (5.9)$$

Note that the gaps which have not been included in these expressions have the same lengths as those of their corresponding gaps in $\{e_i^{s-2}\}$.

In accordance with the proof of Lemma 5.1, we first consider the case $s>j\geq 3$. Then, from Equation (5.4) and the first two expressions in Equation (5.9), it can be seen that the corresponding gap-lengths in R_{s-3}-domain will be reduced by 2^{s-3}. The gap-length in the third expressions in Equation (5.9), which corresponds to the central-gap in R_{s-3}-domain, will either remain the same or will be reduced by 2^{s-3} if LF^d does not exist, as in $\{e_i^{s-3}\}$. The end-gap in this domain depends on whether the last NF error position is from category (c) or (d). Thus, by using Equations (5.4) and (5.8) we have:

$$\begin{cases} \delta\varepsilon_{t-1}^{s-3} = 2^{s-2} + 2^{s-3} & \text{if last NF is } LNF^{c}, \\ \delta\varepsilon_{t-1}^{s-2} = 2^{s-2} + 2^{s-2} = 2^{s-1} & \text{if last NF if } LNF^{d}, \end{cases} \qquad (5.10)$$

which shows that the end-gap in $\{\varepsilon_i^{s-3}\}$ will be reduced by 2^{s-3} w.r.t. $\{e_i^{s-3}\}$ for category (c), and it will remain the same for category (d). Hence, $\{\varepsilon_i^{s-3}\}$ has no gap-length increase w.r.t. $\{e_i^{s-3}\}$, it has the same parity as $\{e_i^{s-3}\}$, and its gap-length reductions w.r.t. $\{e_i^{s-3}\}$ ~~are of the order 2^{s-3}. Therefore, $\{\varepsilon_i^{s-3}\}$ is not decodable in R_{s-3}~~ domain. Now, proceeding in the same way as in Lemma 5.4, it can easily be shown that the elements of the difference vectors associated with the patterns $\{\varepsilon_i^{s-j}\}$ in R_{s-4}-, R_{s-5}-, ..., and R_1-domains are all of even values; that the parities of the error positions remain the same w.r.t. the patterns $\{e_i^{s-j}\}$; and that all the gap-length reductions w.r.t. $\{e_i^{s-j}\}$ are of order 2^{s-j} or multiplies of 2^{s-j}. Clearly, there can be no gap-length increase in any of these domains.

Finally, in R_0-domain, for $s=j\geq 3$, $\{\varepsilon_i^0\}$ will have the same pattern as $\{\varepsilon_i^0\}$ with reduced gap-lengths only.

Case ii:

In this case, an interchange of error positions in $\{\varepsilon_i^{s-2}\}$ w.r.t. $\{e_i^{s-2}\}$ in R_{s-2}-domain simply means that there is an overlapping of two or more adjacent gaps from two adjacent categories. However, this interchange of error positions will not produce any increase in the gap-lengths. Two cases can be distinguished when an error has exchanged its position with that of another error from a previous category in R_{s-2}-domain. We will consider these two cases in the following.

As the first case, consider that these two errors are not of the same parity, and denote them by e_F^{s-2} and e_{NF}^{s-2}. Moreover, denote by e_{F-}^{s-2} (e_{NF-}^{s-2}) and e_{F+}^{s-2} (e_{NF+}^{s-2}) the two errors which are immediately before and after the error e_F^{s-2} (e_{NF}^{s-2}), respectively, and which are of the same parity as e_F^{s-2} (e_{NF}^{s-2}). The errors e_{F-}^{s-2}, e_F^{s-2}, and e_{F+}^{s-2} produce two consecutive gaps in R_{s-3}-domain, as do the errors e_{NF-}^{s-2}, e_{NF}^{s-2}, and e_{NF+}^{s-2}. The former two gaps and the latter two will not have any overlapping in R_{s-3}-domain. Thus, in this domain the errors corresponding to e_F^{s-2} and e_{NF}^{s-2} are arranged in $\{\varepsilon_i^{s-3}\}$ in the same order as they are in $\{e_i^{s-2}\}$. Therefore, $\{\varepsilon_i^{s-3}\}$ will have the same properties in R_{s-3}-domain as in case i, and the lemma can accordingly be proved from R_{s-3} to $-R_0$-domains.

As the second case, consider that the above two errors are of the same parity, and denote them by e_{F1}^{s-2} (e_{NF1}^{s-2}) and e_{F2}^{s-2} (e_{NF2}^{s-2}). Then, the corresponding errors e_{F1}^{s-3} (e_{NF1}^{s-3}) and e_{F2}^{s-3} (e_{NF2}^{s-3}) will also interchange their positions in $\{\varepsilon_i^{s-3}\}$ w.r.t. $\{e_i^{s-3}\}$ in R_{s-3}-domain. Therefore, as in R_{s-2}-domain, there will be an overlapping of two or more adjacent gaps in R_{s-3}-domain. As mentioned in the first case, such an interchange of error positions will not produce any increase in the gap-lengths.

Now, considering the above two errors in R_{s-3}-domain, one can distinguish between two subclasses: 1) when these two errors are not of the same parity, and 2) when they are of the same parity. As for 1), the problem under consideration in this domain reduces to that considered in the first case. As for 2), considering the corresponding errors e_{F1}^{s-4} (e_{NF1}^{s-4}) and e_{F2}^{s-4} (e_{NF2}^{s-4}) in R_{s-4}-domain, the position of these errors in $\{\varepsilon_i^{s-4}\}$ w.r.t. $\{e_i^{s-4}\}$ will be interchanged, thus producing an overlapping of two or more adjacent gaps, but not producing any gap-length increase. Therefore, the problem is to find an R_{s-j}-domain, for $5\leq j<s$, such that the parities of e_{F1}^{s-j} (e_{NF1}^{s-j}) and e_{F2}^{s-j} (e_{NF2}^{s-j}) are different.

In this way, by starting from the two errors e_{F1}^{s-2} (e_{NF1}^{s-2}) and e_{F2}^{s-2} (e_{NF2}^{s-2}), we can find an R_{s-j}-domain such that the positions of e_{F1}^{s-j} (e_{NF1}^{s-j}) and e_{F2}^{s-j} (e_{NF2}^{s-j}) are of different parities. Thus, considering e_{F1}^{s-2} (e_{NF1}^{s-2}) and e_{F2}^{s-2} (e_{NF2}^{s-2}) we have the following three parts:

1) if

$$e_{F1}^{s-2}(e_{NF1}^{s-2}) - e_{F2}^{s-2}(e_{NF2}^{s-2}) = 2(2\lambda+1),$$

for $0\leq\lambda\leq 2^{s-5}-1$ and $s\geq 5$, then the position of the two errors e_{F1}^{s-3} (e_{NF1}^{s-3}) and e_{F2}^{s-3} (e_{NF2}^{s-3}) in R_{s-3}-domain will be interchanged and their parities will become different w.r.t. $\{e_i^{s-3}\}$. Therefore, as for the first case, there two errors together with their "immediately before" and "immediately after" errors which have the same parities will produce two non-overlapping gaps in R_{s-4}-domain. Thus, e_{F1}^{s-4} (e_{NF1}^{s-4}) and e_{F2}^{s-4} (e_{NF2}^{s-4}) will be arranged in $\{\varepsilon_i^{s-4}\}$ in the same order as they are in $\{e_i^{s-4}\}$.

2) if

$$e_{F1}^{s-2}(e_{NF1}^{s-2}) - e_{F2}^{s-2}(e_{NF2}^{s-2}) = 2^{\mu+2},$$

for $0\leq\mu\leq$ s-5 and $s\geq 5$, the position of the two errors e_{F1}^{s-3} (e_{NF1}^{s-3})

and e_{F2}^{s-3} (e_{NF2}^{s-3}) will be interchanged but their parities will remain the same w.r.t. $\{e_i^{s-3}\}$. This process can go ahead at most up to $R_{s-3-\mu}$-domain. In this domain, the distance between the two errors $e_{F1}^{s-3-\mu}$ $(e_{NF1}^{s-3-\mu})$ and $e_{F2}^{s-3-\mu}$ $(e_{NF2}^{s-3-\mu})$ reduces to that in part 1. Thus, the corresponding errors $e_{F1}^{s-5-\mu}$ $(e_{NF1}^{s-5-\mu})$ and $e_{F2}^{s-5-\mu}$ $(e_{NF2}^{s-5-\mu})$ in $R_{s-5-\mu}$-domain will be arranged in $\{\varepsilon_i^{s-5-\mu}\}$ in the same order as they are in $\{e_i^{s-5-\mu}\}$, but they will produce no gap-length increase.

3) if

$$e_{F1}^{s-2}\ (e_{NF1}^{s-2}) - e_{F2}^{s-2}\ (e_{NF2}^{s-2}) = 2^{\mu+2}(2\lambda+1) < 2^{s-1} ,$$

for $0\leq\mu\leq s-6$, $1\leq\lambda\leq 2^{s-5}-1$, and $s\geq 6$, then it can be shown (in accordance with parts 1 and 2) that the position of the two errors $e_{F1}^{s-5-\mu}$ $(e_{NF1}^{s-5-\mu})$ and $e_{F2}^{s-5-\mu}$ $(e_{NF2}^{s-5-\mu})$ in $R_{s-5-\mu}$-domain will be rearranged in $\{\varepsilon_i^{s-5-\mu}\}$ in the same order as they are in $\{e_i^{s-5-\mu}\}$.

Thus, we have shown that for both cases i and ii, from R_{s-j}-domain to R_0-domain for $3\leq j\leq s$, there will be no gap-length increase in $\{\varepsilon_i^{s-j}\}$ w.r.t. $\{e_i^{s-j}\}$. This completes the proof of the lemma.

Lemmas 5.1 and 5.2 establish Theorem 5.2 for the case $\ell=1$.

For the case $\ell=2$, the $(n^{(2)},k,t)$ codes with $n^{(2)}=n'-2^{s-2}$ are not s-step PD, because the (n',k,t) codes are not s-step PD as for the case $\ell=1$.

Similarly, for the case $\ell=3$, the $(n^{(3)},k,t)$ codes with $n^{(3)}=n^{(2)}-2^{s-2}$ are not s-step PD as for the case $\ell=2$.

In this way, it can be shown that for $\ell>1$, the $(n^{(\ell)},k,t)$ codes with $n^{(\ell)} = n^{(\ell-1)}-2^{s-2}$ are not s-step PD, because the $(n^{(\ell-1)},k,t)$

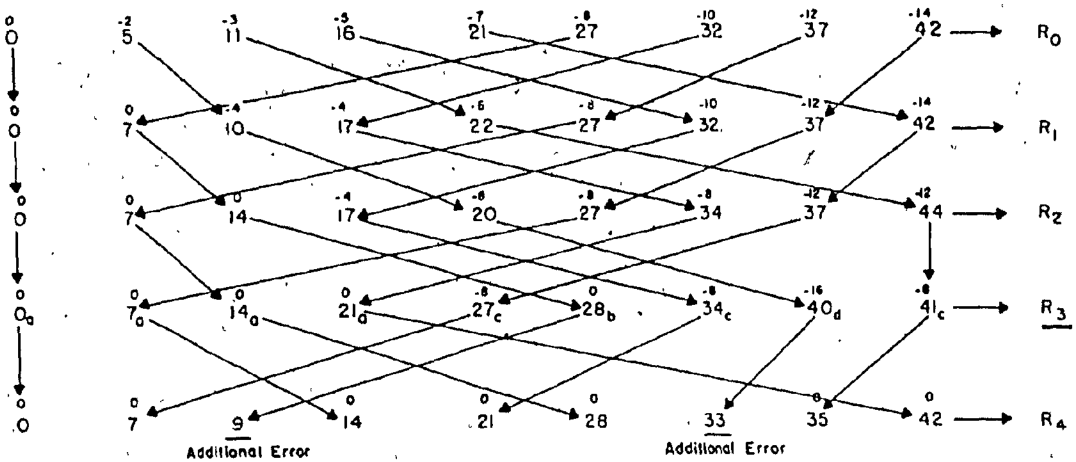
codes are not s-step PD. Q.E.D.

EXAMPLE 5.1:

As an example to illustrate Theorem 5.2, consider the case $s=5$. Then, since the (63,7,9) code is not 5-step PD (see Theorem 5.1), the (47,7,9) code is not 5-step PD either as shown in Figure 5-1. The pattern for the (63,7,9) code is $\{e_i^{s-j} = 7\cdot i$, for $i=0,1, \ldots, t-1 = 8$ and $1\leq j\leq 5\}$. For the (47,7,9) code, we construct all the patterns in five different domains, starting with R_3-domain, as

$$\{\varepsilon_i^3\} = \begin{cases} \varepsilon_0^3 = e_0^3 = 0, & \\ \varepsilon_i^3 = e_i^3, & \text{for } e_i^3 \leq 24 \quad \text{(a)}, \\ \varepsilon_i^3 = e_i^3, & \text{for } 24 < e_i^3 \leq 31 \quad \text{(b)}, \\ \varepsilon_i^3 = e_i^3-8, & \text{for } 31 < e_i^3 \leq 55 \quad \text{(c)}, \\ \varepsilon_i^3 = e_i^3-16, & \text{for } 55 < e_i^3 < 63 \quad \text{(d)}, \end{cases}$$

by using Equation (5.2). Thus, this example corresponds to Lemma 5.2. For the details of the patterns with their difference values w.r.t. $\{e_i\}$, see Figure 5-1.



Legend

Upper numbers: Negative of the difference vector in each domain $-\{\delta\varepsilon_i^{s-j}\}$.

Lower numbers: Error positions placed orderly $\{\varepsilon_i^{s-j}\}$.

Figure 5-1. An illustrative example of the pattern $\{e_i\}$ for the (47,7,9) code which is not 5-step PD, derived from the pattern $\{e_i\}$ for the (63,7,9) code.

The next general result is concerned with some other codes which are not PD.

THEOREM 5.3

The (n,k_e,t_e) codes with $n = \frac{3}{4} t_e \cdot k_e$ and $k_e \cdot t_e \not\equiv 0 \bmod 8$, are not PD.

Proof of Theorem 5.3

This theorem is established by considering two pairs of EPs, which are not PD in any R_i-domain, $i \geq 0$. Let us consider the EPs $\{e_i\}$ and $\{\varepsilon_i\}$, for $i=0,1, \ldots, t_e-1$, in R_0-domain, as:

$$\begin{cases} \varepsilon_{2i} = e_{2i} = \frac{3}{2} k_e \cdot i, \\ e_{2i+1} = e_{2i} + \frac{k_e}{2}, & \text{for } 0 \leq i \leq \frac{t_e}{2} - 1, \\ \varepsilon_{2i+1} = \varepsilon_{2i} + k_e. \end{cases} \tag{5.11}$$

From Equation (5.11) it is obvious that all the gaps are of lengths not greater than (k_e-1). The end-gap for both EPs are:

$$\begin{cases} EG(n, e_{t_e-1}) = \frac{3}{4} k_e \cdot t_e - [\frac{3t_e \cdot k_e}{4} - k_e] - 1 = k_e - 1, \\ EG(n, \varepsilon_{t_e-1}) = \frac{3}{4} k_e \cdot t_e - [\frac{3t_e \cdot k_e}{4} - \frac{k_e}{2}] - 1 = \frac{k_e}{2} - 1. \end{cases}$$

Thus, $\{e_i\}$ and $\{\varepsilon_i\}$ given in Equation (5.11) are not 1-step PD. The corresponding EPs in R_1-domain, $\{e_i'\}$ and $\{\varepsilon_i'\}$ become:

$$\{e_i'\} = \begin{cases} e_{2i}' = e_{2i}, & \text{for } 0 \leq i \leq \frac{t_e}{2} - 1 \\ e_{2i+1}' = e_{2i} + k_e, \end{cases}$$

which is the same as $\{\varepsilon_i\}$. That is $\{e_i'\} = \{\varepsilon_i\}$. Similarly,

$$\{\varepsilon_i^{\prime}\} = \begin{cases} \varepsilon_{2i}^{\prime} = \varepsilon_{2i}, \\ \\ \varepsilon_{2i+1}^{\prime} = \varepsilon_{2i} + \dfrac{k_e}{2} . \end{cases} \qquad \text{for } 0 \leq i \leq \frac{t_e}{2} - 1,$$

That is $\{\varepsilon_i^{\prime}\} = \{e_i\}$. Thus, the patterns $\{e_i\}$ and $\{\varepsilon_i\}$ will interchange with the corresponding patterns $\{\varepsilon_i^{\prime}\}$ and $\{e_i^{\prime}\}$ in R_1-domain, respectively. This interchange of patterns will also take place from R_1- to R_2-doamins, and so on. Therefore, by using any higher group (U) permutations the patterns corresponding to $\{e_i\}$ and $\{\varepsilon_i\}$ will still remain not PD.

Simiarly, let us consider the EPs, $\{e_i\}$ and $\{\varepsilon_i\}$ for $i=0,1,\ldots, k_e-1$ and $k_e \leq t_e$, in R_0-domain, as:

$$\{e_i\} = \begin{cases} e_{2i} = \dfrac{3}{2}\, t_e \cdot i, \\ \\ e_{2i+1} = e_{2i} + \dfrac{t_e}{2} , \end{cases} \qquad \text{for } 0 \leq i \leq \frac{k_e}{2} - 1$$

and

$$\{\varepsilon_i\} = \begin{cases} \varepsilon_{2i} = \dfrac{3}{2}\, t_e \cdot i \\ \\ \varepsilon_{2i+1} = \varepsilon_{2i} + t_e . \end{cases} \qquad \text{for } 0 \leq i \leq \frac{k_e}{2} - 1 ,$$

In this case too, the patterns $\{e_i\}$ and $\{\varepsilon_i\}$ are not 1-step PD and they will interchange with the corresponding patterns from one domain to the next higher domain. Thus, these patterns will remain not PD.

Q.E.D.

From the proof of Theorem 5.3, the following corollary is an immeidate consequence.

COROLLARY 5.4

The (n,k_e,t_o) codes with $n = \frac{3k_e}{4}(t_o-1)$, and the (n,k_o,t_e) codes with $n = \frac{3\cdot t_e}{4}\cdot(k_o-1)$ for $k_e(t_o-1)$ or $t_e\cdot(k_o-1) \not\equiv 0$ mod 8, respectively, are not PD.

Proof of Corollary 5.5:

This corollary is a logical consequence of Theorem 5.3. This is so because if an (n,k,t) code is not PD, then by replacing k with $k+1$, and t with $t+1$ (or both), the resulting code is obviously not PD.

5.3 Summary of the Results

In this chapter we obtained some general results based on certain cyclic codes which are, in general, not PD. These results are used in establishing the exact bounds given in previous two chapters and also are applicable to all group (T,U) permutations.

Theorems 5.1 to 5.3 and their related corollaries assert the following results for both even and odd values of t, and both even and odd values of k.

In general, the following codes are not permutation decodable:

I. For any t (odd or even), the codes (n,k,t) with

$$n = \delta_0 \cdot \tau_0 , \tag{5.12}$$

where $\delta_0 \leq k$, and $\tau_0 \leq t$, are not PD.

II. If the (n,k,t) code is not s-step PD, then, the (n',k,t) codes with

$$n' = n-2^{s-1} \cdot \ell, \quad \text{for} \quad \ell \geq 1 \tag{5.13}$$

are not s-step PD.

III. For any t (odd or even), the codes (n,k,t) with

$$n = \frac{3}{4} \delta_e \cdot \tau_e , \tag{5.14}$$

where $\delta_e \leq k$, $\tau_e \leq t$, and $\delta_e \cdot \tau_e \not\equiv 0 \bmod 8$, are not PD.

CHAPTER 6

CONCLUSIONS AND SUGGESTIONS FOR FUTURE WORK

In this chapter, we summarize the results presented in the thesis. We then briefly discuss the results obtained and conclude the chapter with suggestions for further research.

6.1 Summary

The main emphasis in this thesis has been on analyzing the capability of error-trapping decoding, specially, those which are based on the "permutation decoding" concept. In this respect, we have derived exact lower bounds on the code length n, for given information length k, of the "multiple-error-correcting" binary (n,k,t) cyclic codes by applying cyclic (T) and squaring (U) (or square rooting) group (T,U) permutations.

Chapter 1 was a brief introduction to coding theory with special emphasis on cyclic codes.

In Chapter 2 we discussed the various decoding procedures available for decoding cyclic codes. We introduced the error-trapping technique and related decoding schemes based on the concept of permutation decoding for cyclic codes. From the implementation point of view, we described a practical permutation decoder at the end of the chapter.

In Chapter 3, we developed the relationship that must exist between n,k and t in order for a binary (n,k,t) cyclic code to be 2-step permutation decodable. In this respect, we established the exact lower bounds on code length n, for the (n,k,t) cyclic codes with t being odd- or even valued.

In Chapter 4, we derived the exact lower bounds on code length n, by applying 3-step permutations to the (n,k,t) cyclic codes with odd-valued t, and to the case of even $t=2$.

Chapter 5 described certain classes of cyclic codes which are not permutation decodable.

The following main results were obtained:

Result I

The lower bounds on n, by applying 2-step permutations (for $U^i \cdot T^\beta$ for $i=0,1$, and $\beta = 0,1, \ldots, n-1$) are:

a) For t odd-valued (t_o):

i) k odd-valued (n,k_o,t_o) codes:

$$n > k_o \cdot t_o \; . \tag{6.1}$$

ii) k even-valued (n,k_e,t_o) codes:

$$n > t_o \cdot (k_e-1) \; . \tag{6.2}$$

b) For t even-valued (t_e):

i) k odd-valued (n,k_o,t_e) codes

$$n > (t_e-1) \cdot k_o \; . \tag{6.3}$$

ii) k even-valued (n,k_e,t_e) codes:

$$n > (t_e-1) \cdot (k_e-1)+2 \tag{6.4}$$

c) Special cases:

1) For $t=2$, $(n,k,2)$ codes:

$$n > 3 \cdot k/2 \tag{6.5}$$

2) For $k=2$, $(n,2,t)$ codes:

i) t odd-valued

$$n > (3 \cdot t_o - 1)/2 \tag{6.6}$$

ii) t even-valued

$$n > 3 \cdot t_e/2 \tag{6.7}$$

We can restate the cases (i) and (ii) for the case $k=2$, as if, $n > 3 \cdot t/2$, then the $(n,2,t)$ code is 2-step PD.

Result II

The extent of decodability of 3-step PD codes ($u^i \cdot T^\beta$ for $i=0,1,2$; and $\beta = 0,1, \ldots, n-1$) only for t odd-valued codes.

In addition to the improved bounds given for 2-step PD codes, the following codes are 3-step PD.

a) The (n,k_o,t_o) codes:

$$n = t_o \cdot k_o - 2(2\ell+1), \quad \text{if } k_o = 2\ell+3 ,$$
$$\text{and } 0 \leq \ell < (t_o - 1)/2 . \tag{6.8}$$

b) The (n,k_e,t_o) codes:

$$n = t_o \cdot (k_e - 1) - 2(2\ell+1) \quad \text{if } k_e > 2\ell+4 ,$$

$$\text{and} \quad 0 \le \ell < (t_0-1)/2 \qquad (6.9)$$

c) Special cases:

1) for $k=2$, $(n,2,t_0)$ codes:

$$n \ge (3 \cdot t_0-1)/2 \qquad (6.10)$$

2) for $t=2$, $(n,k,2)$ codes:

$$n = k_0 + \frac{k_0-1}{2}, \quad \text{for} \quad k_0 > 3$$

The rest of the codes are under the same bounds given for 2-step PD.

Result III

In general, the following codes are not permutation decodable:

1) If the (n,k,t) codes is not s-step PD, then the (n',k,t) codes with

$$n' = n-2^{s-1} \cdot \ell, \quad \text{for} \quad \ell \ge 1 \qquad (6.11)$$

are not s-step PD.

2) for any t (odd or even), the codes (n,k,t) with

$$n = \delta_0 \cdot \tau_0, \qquad (6.12)$$

where $\delta_0 \le k$, and $\tau_0 \le t$, are not PD.

3) For any t (odd or even), the codes (n,k,t) with

$$n = \frac{3}{4}\delta_e \cdot \tau_e, \qquad (6.13)$$

where $\delta_e \leq k$, $\tau_e \leq t$, and $\delta_e \cdot \tau_e \not\equiv 0 \bmod 8$, are not PD.

Since the derivation of these results involves only positions of the errors, the results are applicable not only to binary cyclic codes over GF(2), but also to cyclic codes over $GF(2^m)$, for $m>1$.

6.2 Concluding Remarks

With regard to the (n,k,t) cyclic codes we have presented exact lower bounds on the code length n, or equivalently the upper bounds on the code rate k/n in order for the code to be PD. In this work, by comparing the results of 2-step and 3-step PD codes with those of 1-step PD codes, we have proved the following facts:

1) For 2-step PD cyclic codes, the lower bounds on n for the (n,k_e,t_o) and the (n,k_o,t_e) cyclic codes have been improved over those of 1-step PD codes by as much as (t_e) and (k_e), respectively. Also, the lower bound for the (n,k_e,t_e) cyclic codes has been improved by as must as (t_e+k_e-4). However, the lower bound for the (n,k_o,t_o) cyclic codes has not been improved.

2) In the case of 2-step PD codes, if t odd is increased to t+1, the lower bound on n increases only by 2.

As far as 2-step PD codes are concerned it is better to have codes with even t rather than odd t.

3) For 3-step PD codes, the lower bounds on code length n for the (n,k_o,t_o) and the (n,k_e,t_o) cyclic codes have been improved by as much as $(2\cdot t_o-4)$ and $(3\cdot t_o-4)$, respectively.

Based on the results obtained in this thesis and the above mentioned facts, we give Tables 6-1 to 6-4 for some specific numbers of correctable errors $t_o = 5,9$ and $t_e = 6,10$, as examples. These tables show the improvements on the code length n, by increasing the number of steps of permutations from 1-step to 3-step permutations.

From these tables it can be seen that when codes with high rate are required, it would be sufficient to use an (n,k,t) cyclic code to correct at most, say, t_1 errors, where $t_1 < t$. It would then be a great advantage to decode such a code with a simple error-correcting procedure. One such application is in ARQ Systems [14]. In this connection, let us consider Tables 3-4, 3-5, 4-4, 4-5, 6-1, and 6-2 for the (n,k,t) cyclic codes with $t_o=5$ and $t_o=9$, which can be decoded simply, using 2-step or 3-step permutations. From these tables one can observe that if an (n,k,t) cyclic code with $t=9$ cannot be decoded by certain steps of permutations (Table 6-2), it may be possible that this code fall within the bounds given in Table 6-1 for $t_1=5$. If this is sufficient as error correcting capability for the code, then the error-trapping technique based on the permutation concept can be applied.

TABLE 6-1

The exact lower bounds on the code length n, for $t_0=5$,

	EXACT LOWER BOUNDS ON n		
K	1-Step PD Codes	2-Step PD Codes	3-Step PD Codes
2	11	9	7
3	17	17	13
4	21	17	17
5	27	27	19
6	31	27	23
8	41	37	29
10	51	47	39
12	61	57	49
14	71	67	59
16	81	77	69
⋮	⋮	⋮	⋮

TABLE 6-2

The exact lower bounds on the code length n, for $t_o=9$

	EXACT LOWER BOUNDS ON n		
K	1-Step PD Codes	2-Step PD Codes	3-Step PD Codes
2	19	15	13
3	29	29	25
4	37	29	29
5	47	47	39
6	55	47	43
7	65	65	53
8	73	66	57
9	83	83	67
10	91	83	71
12	109	101	85
14	127	119	103
16	145	137	121
⋮	⋮	⋮	⋮

TABLE 6-3

The exact lower bounds on the code length n, for $t_e=6$.

	THE EXACT LOWER BOUNDS ON n	
K	1-Step PD Codes	2-Step PD Codes
2	13	11
3	19	17
4	25	19
5	31	27
6	37	29
7	43	37
8	49	39
9	55	47
10	61	49
11	67	57
12	73	59
13	79	67
14	85	69
15	91	77
16	97	79
.	.	.
.	.	.
.	.	.

TABLE 6-4

The exact lower bounds on the code length n, for $t_e=10$.

	THE EXACT LOWER BOUNDS ON n	
K	1-Step PD Codes	2-Step PD Codes
2	21	17
3	31	29
4	41	31
5	51	47
6	61	49
7	71	65
8	81	67
9	91	83
10	101	85
11	111	101
12	121	103
13	131	119
14	141	121
15	151	137
16	161	139
⋮	⋮	⋮

6.3 Suggestions for Future Research

The permutation decodina concent in error-trapping technique for decoding cyclic codes can be studied further by considering the following:

1) By increasing the number of steps of the group (T,U) permutations the lower bounds on n improves, so that the capability of error-trapping technique can be extended to higher rate codes.

2) In addition to the group (T,U) permutations, there are other sets of permutations for which certain cyclic codes are invariant. For example, the extended binary QR codes are invariant under the Doubly-Transitive-Projective-Unimodular group, and thus the error-trapping technique is applicable. At the present time, little theoretical progress has been made on this problem.

REFERENCES

[1] Shannon, C.E., "A Mathematical Theory of Communications", Bell System Tech. J., Vol. 27, pp. 379-423 (Part I) and 623-656 (Part II), July 1948.

[2] Shannon, C.E., "Communication in the Presence of Noise", Proc. IRE, Vol. 37, pp. 10-21, January 1949.

[3] Wolf, J.K., "Statistical Communication Theory", IEEE Communications Magazine, pp. 121-122, May 1984.

[4] Lucky, R.W., Salz, J., and Weldon, E.J., "Principles of Data Communication", McGraw-Hill, New York, 1968.

[5] Massey, J.L., "Threshold Decoding", MIT Press, Cambridge, Mass., 1963.

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

[7] Gallager, R.G., "Information Theory and Reliable Communication", Wiley, New York, 1968.

[8] MacWillians, F.J., and Sloane, N.J.A., "The Theory of Error-Correcting Codes", North-Holland, Amsterdam, 1977.

[9] Berlekamp, E.R., "Algebraic Coding Theory", McGraw-Hill, New York, 1968.

[10] Clark, G.C., and Cain, J.B., "Error-Correcting Coding for Digital Communications", Plenum Press, New York, 1981.

[11] Blahut, R.E., "Theory and Practice of Error Control Codes", Addison Wesley, 1983.

[12] McEliece, R.J., "The Theory of Information and Coding", Addison-Wesley, 1977.

[13] Bhargava, V.K., Haccoun, D., Matyas, R., and Nuspl, P., "Digital Communication by Satellite", Wiley, New York, 1981.

[14] Lin,S., and Yu, P.S., "A Hybid ARQ Scheme with Parity Retransmission for Error Control of Satellite Channels", IEEE Communications, Vol. COM-30, pp. 1701-1719, July 1982.

[15] Lin, S., and Costelloa, D.J., "Error-Control Coding; Fundamentals and Applications", Prentice-Hall, Englewood Cliffs, New Jersey, 1983.

[16] Albert, A.A., "Fundamental Concepts of Higher Algebra", University of Chicago Press, Chicago, 1956.

[17] Birkhoff, G., and Maclane, S., "A Survey of Modern Algebra", Third Edition, The MacMillan Co., 1965.

[18] Peterson, W.W., "Error-Correcting Codes", MIT Press, Cambridge Mass., 1961.

[19] Slepian, D., "A Class of Binary Signalling Alphabets", Bell System Tech. J., Vol. 35, pp. 203-234, January 1956.

[20] Slepian, D., "Some Further Theory of Group Codes", Bell System Tech. J., Vol. 39, pp. 1219-1252, September 1960.

[21] Hamming, R.W., "Error Detecting and Error Correcting Codes", Bell System Tech J., Vol. 29, pp. 147-160, April 1950.

[22] Elias, P., "Coding for Noisy Channels", IRE Convention Record, Part 4, Vol. 3, pp. 37-46, March 1955.

[23] Nordstrom, A.W., and Robinson, J., "An Optimum Nonlinear Code", IEEE Infor. Control, Vol. 11, pp. 613-616, December 1967.

[24] Sloane, N.J.A., and Whitehead, D.S., "A New Family of Single-Error-Correcting Codes", IEEE Infor. Theory, Vol. IT-16, pp. 717-719, November 1970.

[25] Preparata, F.P., "Weight and Distance Structure of Nordstrom-Robinson Quadratic Code", IEEE Infor. Control, Vol. 12, pp. 466-473, 1968.

[26] Pless, V., "Information to the Theory of Error-Correcting Codes", IRE Infor. Theory, Wiley, New York, 1982.

[27] Lee, C.Y., "Some Properties of Non-Binary Error-Correcting Codes", IRE Infor. Theory, Vol. IT-4, pp. 77-82, 1958.

[28] Assmus, E.F., and Mattson, H.F., "Coding and Combinatories", SIAM Review, Vol. 16, pp. 349-388, July 1974.

[29] Prange, E., "The Use of Information Sets in Decoding Cyclic Codes", IRE Infor. Theory, Vol. IT-8, pp. S-5-9, September 1962.

[30] Massey, J.L., "Shift-Register Synthesis and BCH Decoding", IEEE Infor. Theory, Vol. IT-15, pp. 122-127, January 1969.

[31] Reed, I.S., and Solomon, G., "Polynomial Codes over Certain Finite Fields", J. Soc. Indust. Appl. Math., Vol. 8, pp. 300-304, June 1960.

[32] Bhargava, V.K., "Some Results on Quasi-Cyclic Codes", Ph.D. Thesis, Queen's University, Kingston, April 1974.

[33] Bose, R.C., and Ray-Chaudhuri, D.K., "On a Class of Error-Correct-

ing Binary Group Codes", IEEE Infor. Control, Vol. 3, pp. 68-79, March 1960.

[34] Bose, R.C., and Ray-Chaudhuri, D.K., "Further Results on Error-Correcting Binary Group Codes", IEEE Infor. Control, Vol. 3, pp. 279-290, September 1960.

[35] Hocquenghem, A., "Codes Correcteurs d'erreurs", Chiffres, 2 , pp. 147-156, 1959.

[36] Forney, G.D., "Concatenated Codes", MIT Press, Cambridge, Mass., 1966.

[37] Goppa, V.D., "A New Class of Linear Codes", Probl. Peredachi Infor., Vol. 6, pp. 24-30, September 1970.

[38] Berlekamp, E.R., "Goppa Codes", IEEE Infor. Theory, Vol. IT-19, pp. 590-592, September 1973.

[39] Tzeng, K.K., and Zimmerman, K., "On Extending Goppa Codes to Cyclic Codes", IEEE Infor. Theory, Vol. IT-21, pp. 713-716, November 1975.

[40] Bhargava, V.K., "Forward Error Correcting Schemes for Digital Communications", IEEE Communications Magazine, pp. 11-19, January 1983.

[41] Townsend, R.L., and Weldon, E.J., "Self-Orthogonal Quasi-Cyclic Codes", IEEE Infor. Theory, Vol. IT-13, pp. 183-195, April 1967.

[42] Chen C.L., "Some Results on Algebraically Structured Error-Correcting Codes", Ph.D., Thesis, University of Hawaii, Hawaii, 1969.

[43] Karlin, M., "New Binary Coding Results by Circulants", IEEE Infor.

Theory, Vol. IT-15, pp. 81-92, January 1969.

[44] Benyamin-Seeyar, A., Shiva, S.G.S., Bhargava, V.K., and Kanetkar, S.V., "On the Error-Trapping Decoding Technique for Binary Cyclic Codes", IEEE International Symp. Infor. Theory, St-Jovite, Canada, Setpember 26-30, 1983.

[45] Benyamin-Seeyar, A., Shiva, S.G.S., and Bhargava, V.K., "Capability of Error-Trapping Technique in Decoding Cyclic Codes", Under Revision by IEEE Infor. Theory.

[46] Shiva, S.G.S., Benyamin-Seeyar, A., and Bhargava, V.K., "On the Permutation-Decodability of Triple-Error-Correcting Codes", Proc. 21st Annual Allerton Conf., Illinois University, October 5-7, 1983.

[47] Meggitt, J.E., "Error Correcting Codes and Their Implementation", IRE Infor. Theory, Vol. IT-7, pp. 232-244, October 1961.

[48] Rudolph, L.D., and Mitchell, M.E., "Implementation of Decoders for Cyclic Codes", IEEE Infor. Theory, Vol. IT-10, pp. 259-260, July 1964.

[49] MacWilliams, F.J., "Permutation Decoding of Systematic Codes", Bell System Tech. J., Vol. 43, pp. 485-505, January 1964.

[50] Kasami, T., "A Decoding Procedure for Multiple-Error-Correcting Cyclic Codes", IEEE Infor. Theory, Vol. IT-10, pp. 134-138, April 1964.

[51] Wei, V.K., "An Error-Trapping Decoder for Nonbinary Cyclic Codes", Vol. IT-30, pp. 538-541, May 1984.

[52] Gordon, D.M., "Minimal Permutation Sets for Decoding the Binary Golay Codes", IEEE Infor. Theory, Vol. IT-28, pp. 541-543, May 1982.

[53] Wolfmann, J., "A Permutation Decoding of the (24,12,8) Golay Code", IEEE Infor. Theory, Vol. IT-29, pp. 748-750, September 1983.

[54] Shiva, S.G.S., Fung, K.C., and Tan, H.S.Y., "On Permutation Decoding of Binary Cyclic Double-Error-Correcting Codes of Certain Lengths", IEEE Infor. Theory, Vol. IT-16, pp. 641-64[illegible] September 1970.

[55] Shiva, S.G.S., and Fung, K.C., "Permutation Decoding of Certain Triple-Error-Correcting Binary Codes", IEEE Infor. Theory, Vol. IT-8, pp. 444-446, May 1972.

[56] Yip, P.W., "Permutation Decodable Cyclic Codes", M.Sc., Thesis, University of Ottawa, Ottawa, May 1974.

[57] Yip, P.W., Shiva, S.G.S., and Cohen, E.L., "Permutation Decodable Binary Cyclic Codes", Electronic Letters, Vol. 10, pp. 467-468, October 1974.

APPENDIX

COMMENT ON WORST CASE ANALYSIS

In Chapter 3, we have introduced four different types of patterns of an error pattern (EP), in R_{i+1}-domain, for $i \geq 0$. From these four types of patterns Table 3-1 to 3-3 have been obtained, resulting in the derivations of the equations, for the worst case analysis.

In addition to the four different types of patterns, for the worst case analysis, one may think of some other kinds of patterns such as

0 x x x 0

or

x 0 0 0 x

denoted by δ-type, and any other combination of error positions of this type. This pattern can not be considered within the worst case patterns. Take δ-type pattern in R_1-domain as an example in Table 3-1: it will correspond to three gaps in R_0-domain; two of those gaps should be of gap-lengths $g_1 \leq \frac{k_e-2}{2}$, and the third gap of length $g_2 \leq k_e-1$. So, it is clear that this type of patterns in comparison with the types of patterns given in Table 3-1 for the derivation of Equation (3.7) cannot be considered as the worst case.