BLOCK CODES CAPABLE OF CORRECTING BOTH SYNCHRONIZATION ERRORS AND ADDITIVE ERRORS

(同期誤りと加法的誤りの両者を訂正する符号に関する研究)

IKUO IIZUKA

JANUARY, 1980

ABSTRACT

The purpose of this thesis is to provide block codes for reliable transmission over the binary channels with synchronization errors such as timing errors, deletion and insertion errors, or resulting synchronization slippage besides the additive burst errors.

In Chapter 1, introducing the basic concept of block coding for a noisy channel, the causes of synchronization error which entirely degrade the performance of the block codes are discussed from two points of view. The outlines of two approaches for correcting the synchronization slippage and the originated timing error are described.

In Chapter 2, coset codes capable of correcting a synchronization slippage, which is due to timing errors occurred in the previously received words, and additive multiple burst errors is discussed. After the historical review in this approach, the required condition of the coset codes is described. Then the bounds on the required number of check symbols are obtained in order to evaluate the performance of the proposed codes.

In Chapter 3, the historical review is given and the heuristic approach of constructing the timing-error correcting codes is discussed. Then a new method of constructing block codes capable of correcting a deletion error and/or an additive burst error is presented, where the code is constructed by interleaving a single burst-error correcting code and a coset code generated by a low rate

cyclic code. It is shown that, first, the proposed code is capable of correcting either a deletion error or an additive burst error in a received word, and second, it is also capable of correcting a deletion error imbedded within an additive burst error.

In Chapter 4, extending the concept of the code presented in Chapter 3, the improved interleaved codes which are capable of not only correcting an arbitrary number of deletion errors, but also correcting deletion or insertion errors in a received word are presented. First, we show that the code is capable of correcting a successive deletion error and/or an additive burst error. Then we show it can correct a successive insertion error and/or an additive burst error, and finally, we show it is also capable of correcting a successive deletion or insertion error imbedded within an additive burst error. The redundancy newly added for timing-error correction is discussed.

In the final chapter, the principal results achieved in this research are summarized.

ACKNOWLEDGMENTS

This work was pursued throughout the graduate studies at Department of Communication Engineering, Faculty of Engineering, Osaka University.

The author would like to express his heartful gratitude to Professor T. Namekawa, this dissertation supervisor, for his sincere instruction and encouragement. The author is gratefull to the Professors at the Graduate School for their zealous instructions in the field of the general communication engineering.

The author wishes to express his sincere gratitude to Associate Professor M. Kasahara whose cordial discussions and many suggestions have enhanced this work exceedingly.

The author would like to thank Assistant Professor M. Sato for his valuable discussions. In addition, the author thanks all members of Namekawa Laboratory for their kind discussions and cooperations.

Last but not least, the author is gratefull to Professor S. Tazaki at Ehime University for his recommendation to this field and encouragement during the cource of this studies.

TABLE OF CONTENTS

CHAPTER Page

LIST OF FIGURES

Figure Page

CHAPTER 1

INTRODUCTION

1.1 Communication System Model and Block Codes

Figure 1.1 shows the block diagram of a general communication system model. Information symbols from the source-encoder output can be encoded for reliable data transmission over the discrete channel between the channel-encoder and the channel-decoder. The general class of codes for which the enormous amount of researches has been done is a class whose members have fixed length, i.e., *block codes*. For a block code the channel-encoder partitions the continuous sequence of information symbols into k-symbol blocks, and then encodes these blocks independently according to the coding rule to be used. Each block of information symbols is encoded to an n-tuple of discrete channel symbols, a *code word*. The quantity n is referred to as the *code length*, where $n>k$ obviously holds. The code word is transmitted and disturbed by channel errors and decoded at the receiving end.

1.2 Synchronization Problems

In order to decode a block code the channel-decoder should be able to identify the first symbol, the second symbol,$\cdots$, the (n-1)st symbol and the nth symbol of the received word in a correct order. The problem discussed in this paper is to construct block codes that are capable of correcting the deletion or insertion errors, so called *timing errors* collectively. These errors can be considered occurring in the following situations [1],[2];

(i) The discrete channel of Figure 1.1, in general, relies on the assumption that an oscillator in the demodulator is kept locked in phase and frequency with an oscillator in the modulator. However, during periods when the waveform-channel is subjected to a noise burst, phase lock is usually lost and the demodulator must make symbol decision without synchronization. As a result, timing errors occur.

(ii) An independent clocking technique is so far considered to be a promising technique. In the communication system using the independent clocking technique, each node has its own accurate frequency clock. However, since the clocks operate independently they will differ slightly in timing, even if they are as accurate as atomic clocks. Such difference in timing will cause artificial deletion or insertion errors in a sequence of the received symbols, even if there exists no channel error.

1.3 Research Outline

There are two basic approaches to recover synchronization when timing errors have occurred in a received word, resulting *synchronization slippage* in the subsequently received word. One of the approaches is to correct the received word itself corrupted by timing errors. The other is to recover synchronization slippage of the subsequently received words,

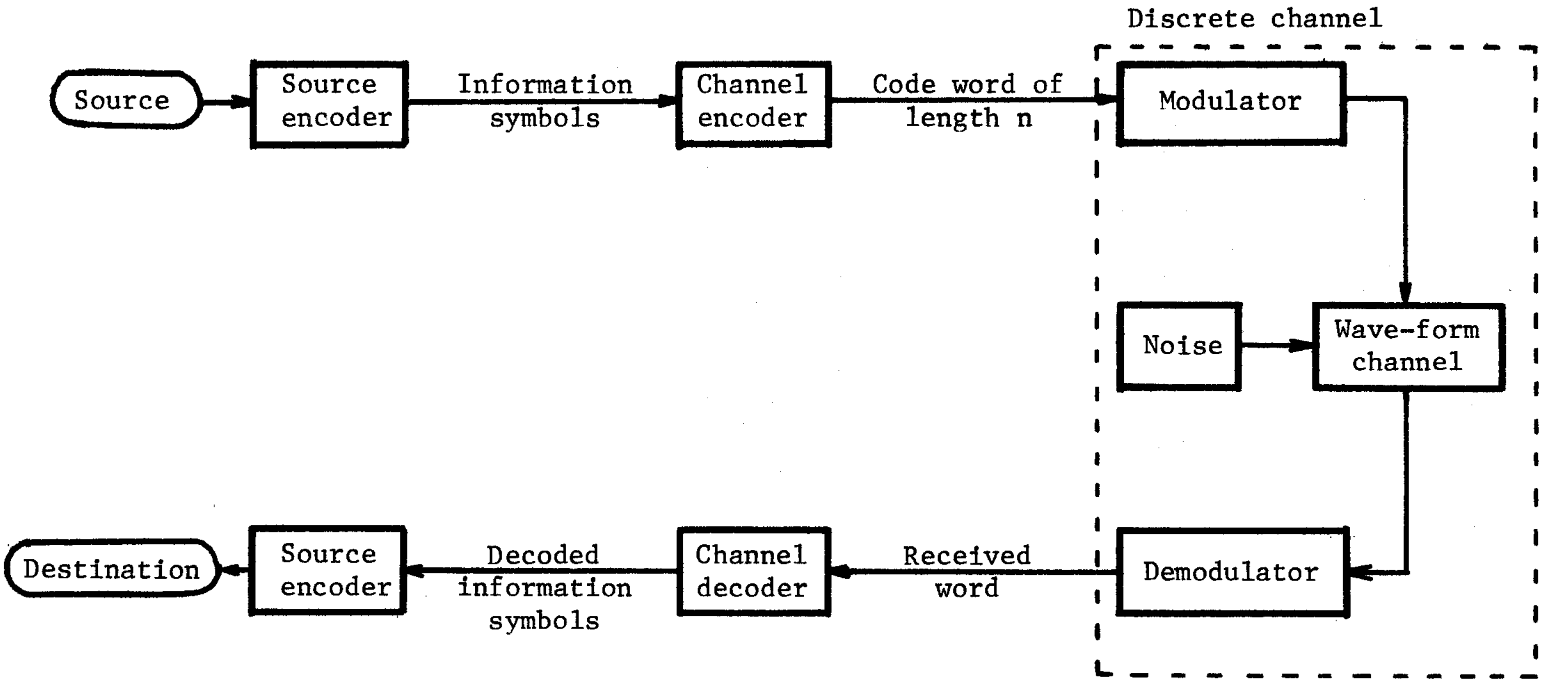


Figure 1.1 Block diagram of a digital communication system.

remaining the causal word undecoded.

In Chapter 2, the latter approach which uses coset codes, generated by cyclic codes, capable of correcting synchronization slippage besides additive burst errors are discussed, since the latter is simpler than the former and provides the fundamental concepts of the next chapters. The condition required for the cyclic code to correct synchronization slippage and additive burst errors is presented along with the optimal pattern of the coset leader and also the bounds on the required check symbols is derived.

In Chapter 3, the former approach using interleaved codes on the basis of two algebraic codes is discussed. It is shown that the proposed codes are capable of correcting a deletion error of one symbol imbedded within an additive burst error.

Extending the ideas of Chapter 3, improved interleaved codes, which are capable of not only correcting an arbitrary number of deletion errors but also correcting deletions or insertion errors imbedded within an additive burst error in a received word, are presented in Chapter 4.

In the final chapter, the principal results achieved in the research are summarized.

CHAPTER 2

COSET CODES[3],[4]

2.1 Introduction

Algebraic codes have been developed in great detail because of their beautiful algebraic structure and simplicity of systematic implementation. This class of block codes is very powerful for reliable transmission provided that the word-synchronization is maintained. These codes, however, turn out to be useless when word-synchronization is lost, or in other words synchronization slippage occurs. For the correction of synchronization slippage we can use the various kinds of code such as comma codes, comma-free codes and coset codes. But a comma code can be only useful in synchronization-error channel where relatively few additive errors occur in any received word. A comma-free code relies on the assumption that there exists no additive error in any received word. A coset code is constructed from a cyclic code by adding a fixed sequence of symbols by modulo 2 to every code word, where the fixed sequence is not a code word of the cyclic code.
Thus the coset code has the advantages of the easy implementation due to its simple algebraic structure.

Stiffler[5] has first shown that a coset code can be designed to be capable of correcting either synchronization slippage or additive errors in any received word. Tong[6] has also proposed the interesting coset codes. However, these coset codes rely on the assumption that both synchronization slippage and additive errors do not occur concurrently in any received word. Redinbo and Wintz[7] have designed coset codes, based on the concept of minimum distance of the Hamming metric, capable of correcting both synchronization slippage and random additive errors concurrently existing in any received word.

In this chapter, we present the required condition of a burst error correcting cyclic code and the optimal pattern of coset leader, where they constitute the proposed coset code capable of correcting both synchronization slippage and multiple additive burst errors in any received word. Obtaining the minimum redundancy of the ideal coset codes, we evaluate the proposed coset codes and compare with some of the previously known codes. Historically, the bounds on the required number of check symbols for the original burst-error correcting codes were first derived by Reiger[8, pp.110]. The bounds on the multiple burst-error correcting codes were recently derived by Belelli, Bianciardi and Cappellini[9]. But these bounds are derived assuming that no end-around burst error exists, which, however, appears as a correctable burst error for any cyclic code.

2.2 Preliminaries

Let us represent a cyclic (n,k) code $\mathcal{C}$ as follows;

$$\mathcal{C} = \{X(z)=\sum_{i=o}^{n-1} \xi_i z^i \mid X(z)\equiv 0,\ z^n-1\equiv 0 \bmod G(z)\} \qquad (2.1)$$

where $\xi_i \in GF(2)$, $i=1,2,\cdots,n-1$, $G(z)$ is the generator polynomial of degree $n-k$ and ξ_{n-1} is the first symbol to be transmitted.

<u>Definition 2.1</u> A *synchronization slippage of* s *bits* occurs when the counter at the channel-decoder counts s bits more that it should.

As shown in Figure 2.1, a received word $Y(z)$ corrupted by a synchronization silppage of $s(>0)$ bits can be represented by

$$Y(z)=z^{-s}X(z)-z^{-s}X_L(z)+z^{n-s}X_L'(z); \quad s>0 \tag{2.2}$$

where $X_L(z)=\sum_{i=o}^{s-1}\xi_i z^i$ and $X_L'(z)=\sum_{i=o}^{s-1}\xi_i' z^i$, which is the suffix of the code word transmitted just prior to $X(z)$. Using the well-known property of cyclic codes, $z^n-1\equiv 0 \bmod G(z)$, the equation (2.2) is given by

$$Y(z)=z^{-s}X(z)+z^{-s}X_E(z) \tag{2.3}$$

where

$$X_E(z)=-X_L(z)+X_L'(z)$$

Noting that equation (2.3) also holds even when s takes on a negative value except that we have the relation $X_E(z)=X_U''(z)+X_U(z)$ $=\sum_{i=n-s}^{n-1}(\xi_i''+\xi_i)z^i$ where $X_U''(z)$ is the prefix of the code word following $X(z)$, we only assume that $s>0$ in what follows for simplicity.

Definition 2.2 An *end-around burst error* occurs when the burst error starts from the (n-j)th position and ends with the ith position where $i<n-j$.

Definition 2.3 Two polynomials $Z_j(z)=\sum_{i=a_j}^{b_j} e_{j,i}z^i$ with $e_{j,a_j}=e_{j,b_j}=1$, j=1 and 2, *overlap by* b_1-a_2+1 when the relation, $a_1\leq a_2\leq b_1\leq b_2$, holds.

For example, $Z_1(z)=z^{10}+e_{1,11}z^{11}+z^{12}$ and $Z_2(z)=z^{11}+e_{2,12}z^{12}+z^{13}$ are overlapped by 2 regardless of the values of $e_{1,11}$ and $e_{2,12}$.

A coset code $\mathcal{C}_C$ is generated by adding a fixed polynomial R(z) to the cyclic code $\mathcal{C}$ as follows;

$$\mathcal{C}_C \triangleq \{A(z)=X(z)+R(z) \mid X(z)\in\mathcal{C} \text{ and } R(z)\notin\mathcal{C}\} \qquad (2.4)$$

Therefore, in decoding of the code $\mathcal{C}_C$, R(z) should be subtracted from the received word B(z) before decoding of the underlying cyclic code $\mathcal{C}$ in the ordinary way. For the received word B(z) corrupted by a synchronization slippage of s bits, we have

$$Y(z)\triangleq B(z)-R(z)=z^{-s}X(z)+z^{-s}X_E(z)+z^{-s}R(z)-R(z) \qquad (2.5)$$

from (2.3). Since $z^{-s}X(z)$ is divisible by G(z) for any s, we delete the term $z^{-s}X(z)$ in the expression in (2.5) and call the resulting polynomial an *error polynomial* E(z) in what follows. Then from (2.5), E(z) can be represented by

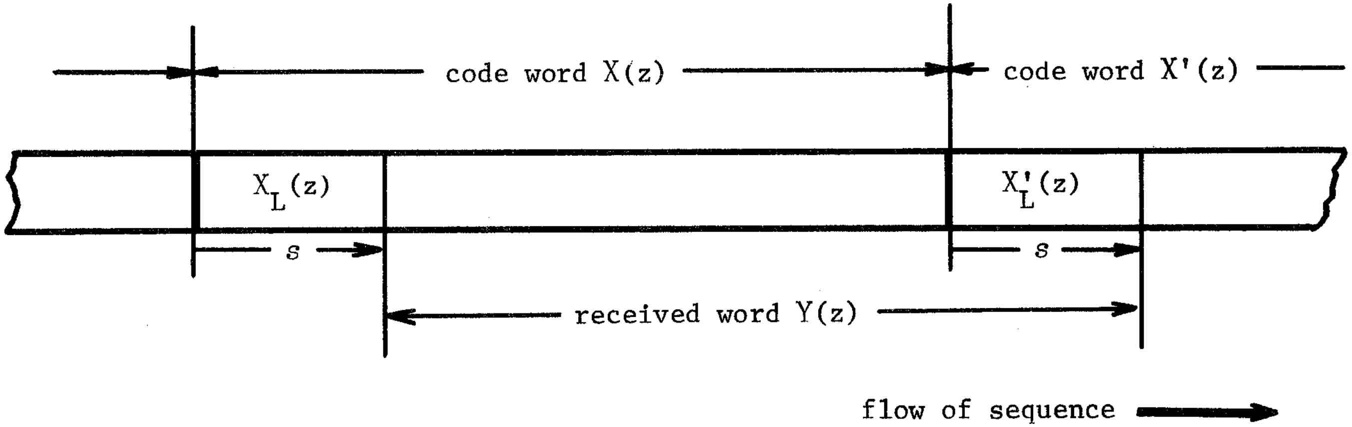


Figure 2.1 Received word when a synchronization slippage of $s(>0)$ bits occurs.

$$E(z)=(z^{-s}-1)R(z)+z^{-s}X_E(z) \qquad \text{mod } G(z) \tag{2.6}$$

<u>Example</u> (Tong's coset codes[6])

When $R(z)=1+z^{n-1}$ is used, $E(z)$ is given by

$$E(z)=(z^{-s}-1)(1+z^{n-1})+z^{-s}X_E(z)$$

$$=-1+(z^{n-s-1}+z^{n-s}(1+X_E(z)-z^{s-1})) \tag{2.7}$$

Equation (2.7) shows that $E(z)$ is an end-around burst error started from the (n-s-1)st position and ended with the 0th position whenever a synchronization slippage of s bits occurs. Therefore, (i) if an additive burst error of length B or less does not overlap the end-around burst error due to the synchronization slippage, and (ii) if an additive burst error of length B or less does not occur as an end-around burst error when no synchronization slippage occurs, then by using a cyclic code capable of detecting an end-around burst error of length S+2 and double additive burst errors of length B or less, both a synchronization slippage of s ($|s|\leqq S$) bits and an additive burst error of length B or less can be corrected.

But obviously the above assumptions (i) and (ii) cannot always be applied to general situations. Thus we should choose an appropriate polynomial $R(z)$ so as to be able to correct both a synchronization slippage and any single additive burst error, and simultaneously we should clarify the required condition on the underlying cyclic code $\mathcal{C}$.

2.3 Correction of Both Synchronization Slippage and Additive Burst Errors

2.3.1 Decoding Algorithm

Before describing a new class of coset codes in detail which is capable of correcting both a synchronization slippage and additive burst errors, we first give the decoding algorithm for the codes.

Decoding Algorithm

Step 1: For the received word $B(z)$, obtain $Y(z)=B(z)-R(z)$ and detect errors if any in $Y(z)$. If no error is detected, i.e., $Y(z)/G(z)=0$, the decoding is completed. Otherwise, go to Step 2.

Step 2: Estimating a burst error $\hat{Z}_B(z)$ of length B or less to offset the burst error $Z_B(z)$, obtain $Y(z)=B(z)-R(z)-\hat{Z}_B(z)$ and go to Step 1. If all of the possible $\hat{Z}_B(z)$ have been assumed, resulting in non-zero residue in $Y(z)/G(z)$, then decide that a synchronization slippage and/or an additive burst error have occurred and go to Step 3.

Step 3: Estimating a synchronization slippage of $\hat{s}(|\hat{s}|\leq S)$ bits to recover the synchronization slippage of s bits, shift the received sequence by $\hat{s}$ bits and go to Step 1.

The precise decoding algorithm is shown in Figure 2.2.

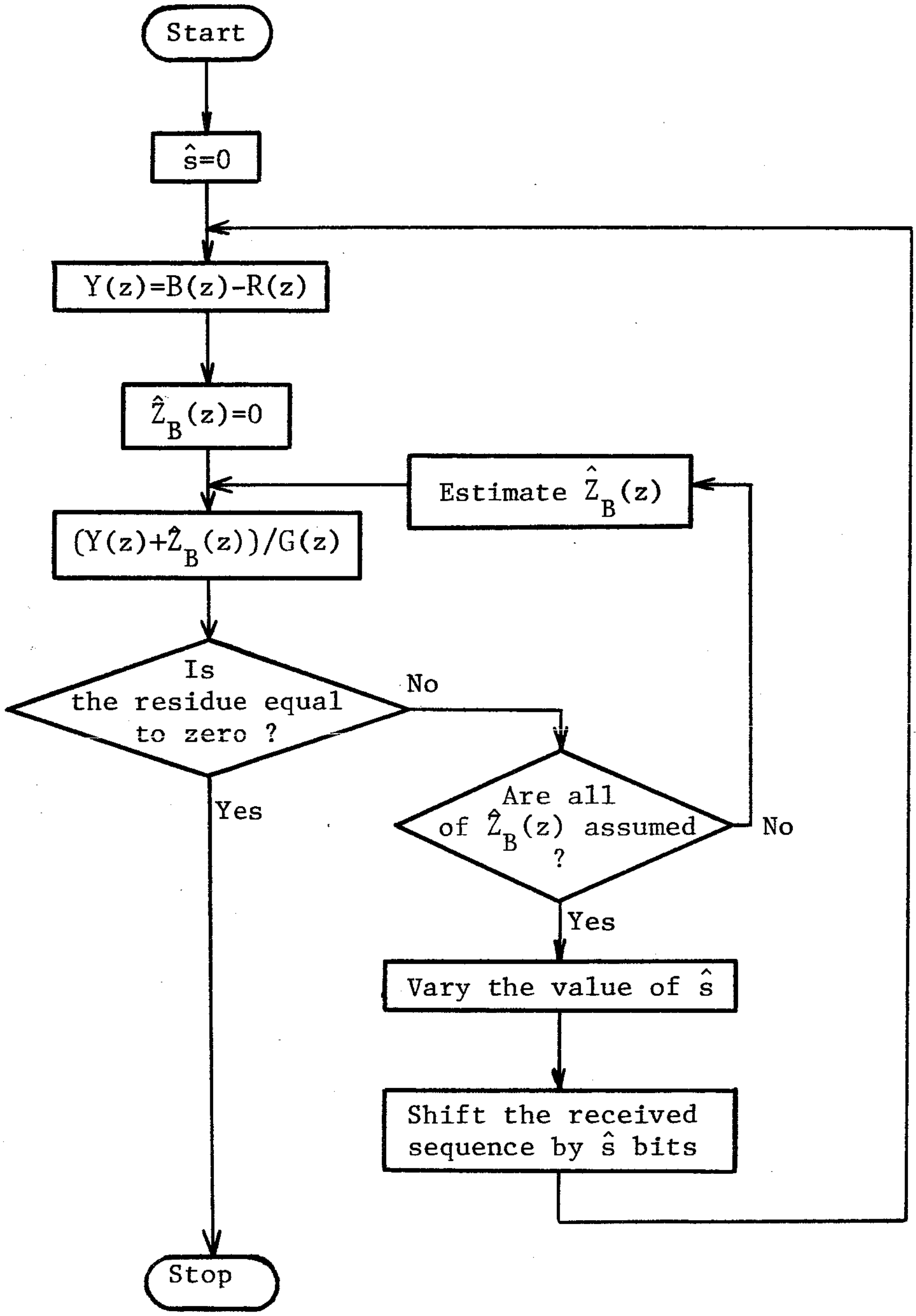


Figure 2.2 Decoding algorithm for proposed coset codes.

2.3.2 Synchronization Slippage and An Additive Burst Error

Consider a coset code capable of correcting both a synchronization slippage of $s(|s|\leq S)$ bits and an additive burst error of length B or less and let

$$R(z)=\sum_{i=o}^{L-1} r_i z^i; \quad L<n/2 \tag{2.8}$$

When a synchronization slippage of s bits and a burst error $Z_B(z)$ of length B or less occur, the error polynomial E(z) in the decoding process can be represented by

$$E(z)=(z^{-s}-1)R(z)+z^{-s}X_E(z)+Z_B(z)-\hat{Z}_B(z) \tag{2.9}$$

The two typical error-patterns of the error polynomial are illustrated in Figure 2.3, where $Z_L(z)$ is the suffix of $(z^{-s}-1)R(z)$ and $X_E'(z)$ is the sum of the prefix of $(z^{-s}-1)R(z)$ and $X_E(z)$. From this, we see that the two conditions given below should be satisfied for the coset code in order to correct both the synchronization slippage and the burst error $Z_B(z)$.

<u>Condition 1</u> The underlying cyclic code is able to detect double burst errors of length B or less and a burst error of length L+S or less.

<u>Condition 2</u> For any $Z_B(z)$ and $\hat{Z}_B(z)$, certain errors are detected whenever a synchronization slippage exists.

Let us derive the optimal pattern of R(z) in the sense that the underlying cyclic code requires the minimum number of check symbols for a given capability. From Condition 1, we should

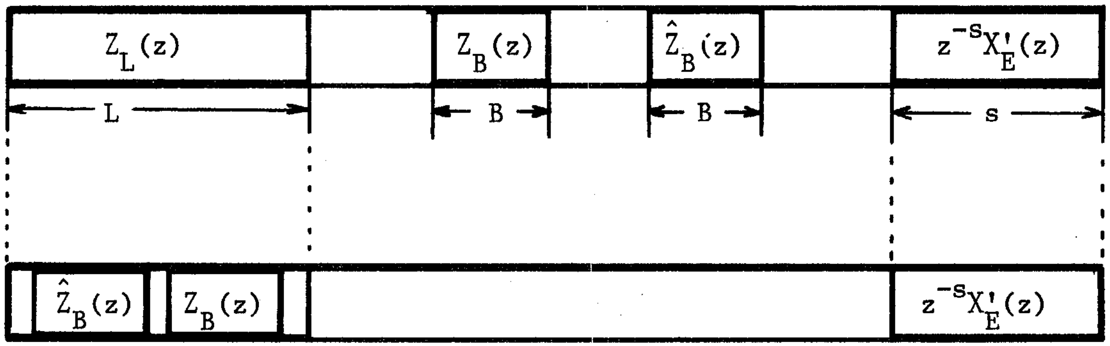


Figure 2.3 Errors when a synchronization slippage of s bits and a burst error occur simultaneously.

choose L as small as possible so as to make the number of check symbols small. On the other hand, L should be greater than 2B since Condition 2 should hold even if $Z_B(z)$ and $\hat{Z}_B(z)$ overlap $Z_L(z)$ and even if $X'_E(z)=0$. Thus, the optimal L is given by

$$L=2B+1 \tag{2.10}$$

For this value of L, there may occur three cases where $Z_B(z)$ and $\hat{Z}_B(z)$ overlap $Z_L(z)$, leaving at least one symbol of $Z_L(z)$ non-overlapped as shown in Figure 2.4. We then have

$$Z_L(z)=\sum_{i=o}^{2B-s}(-r_i+r_{i+s})z^i-\sum_{i=2B-s+1}^{2B}r_iz^i$$

$$=1+z^B+z^{2B}+\sum_{i=1,i\neq B}^{2B-1}r'_iz^i \bmod 2 \text{ for any } s(0<s\leqq S) \tag{2.11}$$

which implies that $r_i=1$ for i=0, B and 2B and $r_{s+B}=r_s=0$ for $s\leqq S<B$. We then have $r_i=0$ for $i=1,2,\cdots, B-1, B+1,\cdots, 2B-1$. Thus, the polynomial R(z) is given by

$$R(z)=1+z^B+z^{2B} \tag{2.12}$$

Furthermore, since the decoder does not know the direction of slippage in general, the relation $|s+\hat{s}|\leqq S$ should hold. Then, the maximum allowable amount of synchronization slippage in any direction is given by

$$S=[(B-1)/2] \tag{2.13}$$

where [a] denotes the largest integer less than or equal to a. For this R(z) the burst error $(z^{-s}-1)R(z)+z^{-s}X_E(z)$ in (2.9) due to the synchronization slippage can be partitioned into double burst

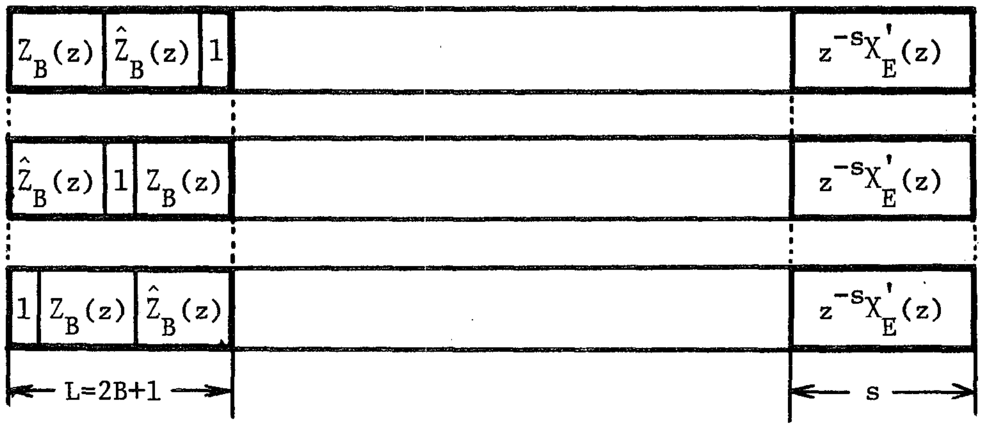


Figure 2.4 Three cases of errors $Z_B(z)$ and $\hat{Z}_B(z)$ overlapping $Z_L(z)$.

errors, $Z_1(z)$ and $Z_2(z)$, as follows;

$$Z_1(z)=z^{-s}X'_E(z)+1+z^{B-s} \tag{2.14}$$

and

$$Z_2(z)=z^{B}+z^{2B-s}+z^{2B} \tag{2.15}$$

These are of length B+1 or less and of length B+1, respectively. Let us summarize the above result in the following theorem.

<u>Theorem 2.1</u> A coset code $\mathcal{C}_C$, generated by a cyclic code $\mathcal{C}$ capable of detecting double burst errors of length B or less and another double burst errors of length B+1 or less by adding the sequence $R(z)=1+z^{B}+z^{2B}$, can correct both a synchronization slippage of $s(|s|\leq[(B-1)/2])$ bits and an additive burst error of length B or less.

2.3.3 Synchronization Slippage and Multiple Additive Burst Errors

Generalizing the coset code specified in Theorem 2.1, we can easily obtain the following theorem.

<u>Theorem 2.2</u> A coset code $\mathcal{C}_C$, generated by a cyclic code $\mathcal{C}$ capable of detecting 2M-ple burst errors of length B or less and another (M+1+[(M+1)/3])-ple burst errors of length B+1 or less, by adding the sequence

$$R(z)=\sum_{i=o}^{2M} z^{iB}, \tag{2.16}$$

can correct both a synchronization slippage of $s(|s|\leq[(B-1)/2])$

bits and M-ple burst errors of length B or less. ___________

2.3.4 Coset Codes Generated by Shortened Cyclic Codes

A shortened cyclic $(n-l_s,k)$ code $\mathcal{C}_S$ is derived from the cyclic (n,k) code $\mathcal{C}$ can be represented by

$$\mathcal{C}_S = \{X(z) = \sum_{i=o}^{n-l_s-1} \xi_i z^i \mid X(z) \equiv 0,\ z^n - 1 \equiv 0 \mod G(z)\} \qquad (2.17)$$

For the coset code $\mathcal{C}_{CS}$ given by

$$\mathcal{C}_{CS} = \{A(z) = X(z) + R(z) \mid X(z) \in \mathcal{C}_S \text{ and } R(z) = 1 + z^B + z^{2B}\} \qquad (2.18)$$

a synchronization slippage of s bits gives the following error polynomial;

$$E(z) = (z^{-s}-1)R(z) - z^{-s}X_L(z) + z^{n-l_s-s}X_L'(z) \qquad (2.19)$$

As shown in Figure 2.5, the second and the third terms in (2.19) appear as burst errors overlapped or non-overlapped, depending upon the relative magnitude of s and l_s. Thus, we have the following theorem.

Theorem 2.3 A coset code $\mathcal{C}_{CS}$, generated by a shortened cyclic code $\mathcal{C}_S$ capable of detecting triple burst errors of length B or less and another double burst errors of length B+1 or less by adding the sequence $R(z)=1+z^B+z^{2B}$, can correct both a synchronization slippage of $s(|s| \leq [(B-1)/2])$ bits and an additive burst error of length B or less. ___________

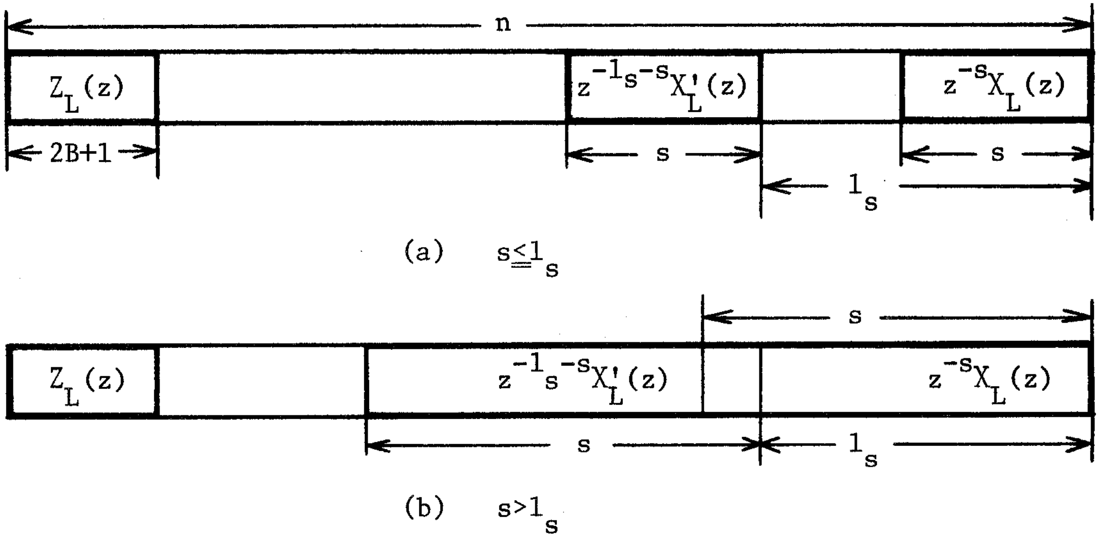


Figure 2.5 Errors for coset code generated by shortened cyclic code when a synchronization slippage of s bits occurs.

2.4 Bounds for Coset Codes Generated by Cyclic Codes

Let us derive the lower bounds on the redundancy required for the underlying cyclic codes of coset codes which are capable of correcting both a synchronization slippage of $s(|s|\leq S)$ bits and M-ple burst errors of length B or less. We use the fact that the total number of the possible syndrome patterns of any error correcting code should be greater than or equal to the total number of the possible error-patterns which can be corrected by the syndrome decoding.

<u>Definition 2.4</u> Any one of burst errors such as

$$Q(z)=1+\sum_{i=1}^{b-2} e_i z^i + z^{b-1} \quad \text{for } e_i \in GF(2) \text{ and } b=1,\cdots,B \tag{2.20}$$

is called a *burst-error block* of length B.

We then have the following lemma.

<u>Lemma 2.1</u> The total number of the possible error-patterns of a burst-error block of length B is 2^{B-1}.

The following lemma states an inherent property of any coset code generated by the cyclic code.

<u>Lemma 2.2</u> The total number of the possible error-patterns of the coset code C_C given by (2.4) is 2^s, when a synchronization slippage of s bits occurs.

(proof) When a synchronization slippage of s bits occurs, the resulting error can be represented by (2.6) as follows;

$$E(z)=(z^{-s}-1)R(z)+z^{-s}X_E(z) \tag{2.6}$$

Since R(z) is a fixed polynomial, the first term of (2.6) takes on a fixed error polynomial for each value of s. The second term takes on one of the 2^s possible error polynomials depending on the components of the transmitted code words. Therefore, the total number of the error-patterns of (2.6) is 1×2^s, completing the proof. Q.E.D.

We also have the following lemma.

<u>Lemma 2.3</u> For non-negative integers v_i, $i=1,2,\cdots, m$ and N the relation

$$\sum_{v_m=0}^{v'_m} \sum_{v_{m-1}=0}^{v'_{m-1}} \cdots \sum_{v_2=0}^{v'_2} \sum_{v_1=0}^{v'_1} 1 = \binom{N+m}{m} \tag{2.21}$$

holds, where

$$\begin{cases} v'_m = N \\ v'_j = N - \sum_{i=j+1}^{m} v_i \; ; \quad 1 \leq j \leq m-1 \end{cases} \tag{2.22}$$

(proof) Let us use the induction on m. Equation (2.21) is clearly valid for m=1. Assume that (2.21) is valid for m=m'-1.

$$\sum_{v_{m'-1}=0}^{v'_{m'-1}} \sum_{v_{m'-2}=0}^{v'_{m'-2}} \cdots \sum_{v_2=0}^{v'_2} (N+1-\sum_{i=2}^{m'-1} v_i) = \binom{N+m'-1}{m'-1} \tag{2.23}$$

Then, the left hand side of (2.21) can be represented by

$$\sum_{v_m=0}^{v'_m} \sum_{v_{m-1}=0}^{v'_{m-1}} \cdots \sum_{v_1=0}^{v'_2} 1 = \sum_{v_m=0}^{v'_m} \{ \sum_{v_{m-1}=0}^{v'_{m-1}} \cdots \sum_{v_2=0}^{v'_2} (N+1-\sum_{i=2}^{m-1} v_i - v_m)\}$$

$$= \sum_{v_m=o}^{v'_m} \{ \sum_{v_{m-1}=o}^{v'_{m-1}} \cdots \sum_{v_2=o}^{v'_2} (N'+1-\sum_{i=2}^{m-1} v_i)\} \qquad (2.24)$$

where $N'=N-v_m$. Using (2.23), we have

$$\sum_{v_m=o}^{v'_m} \sum_{v_{m-1}=o}^{v'_{m-1}} \cdots \sum_{v_1=o}^{v'_2} 1 = \sum_{v_m=o}^{v'_m} \binom{N'+m-1}{m-1} = \sum_{v_m=o}^{v'_m} \binom{N-v_m+m-1}{m-1} \qquad (2.25)$$

By using the well-known formula;

$$\sum_{r=1}^{n} \prod_{s=o}^{k} (r+s) = \frac{1}{k+2} \cdot \frac{(n+k+1)!}{(n-1)!} \qquad (2.26)$$

equation (2.25) reduces to $\binom{N+m}{m}$, completing the proof. Q.E.D.

In addition to the synchronization slippage, let us assume that the multiple additive burst errors of m burst-error blocks $Q_i(z)$, $i=1,2,\cdots, m$, each of length B, occur as shown in Figure 2.6. Note that the fixed error-pattern $(z^{-s}-1)R(z)$ of (2.6) is omitted in this figure. Letting v_i denote the interval between $Q_i(z)$ and $Q_{i+1}(z)$, the error polynomial $E_{SB}(z)$ due to the synchronization slippage and multiple additive burst errors can be represented as follows;

$$E_{SB}(z)=z^{L(1)}Q_m(z)+z^{L(2)}Q_{m-1}(z)+\cdots+z^{L(m)}Q_1(z)+z^{-s}X_E(z) + (z^{-s}-1)R(z) \qquad (2.27)$$

where

$$L(i)=(i-1)B+\sum_{j=o}^{i-1} v_{m-j} \qquad (2.28)$$

Let us enumerate the possible error-patterns represented by (2.27) for fixed values of s and m, by dividing them into the following

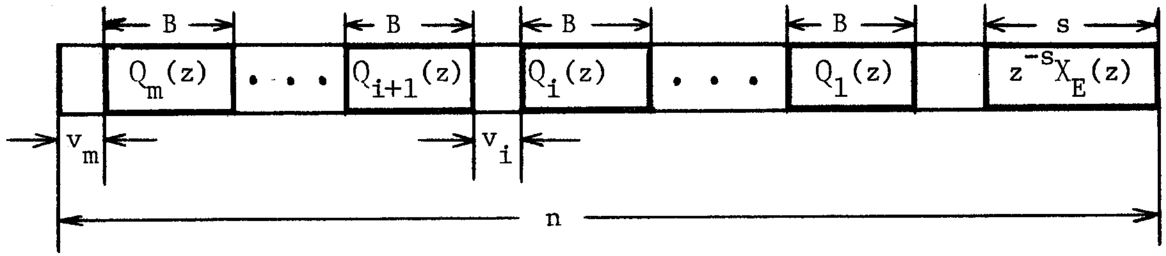


Figure 2.6 Error for coset code generated by cyclic code when a synchronization slippage of s bits and m-ple burst errors occur simultaneously.

three cases.

(i) *In case of* $Q_1(z)$ *which is non-overlapping* $z^{-s}X_E(z)$

The total number of the possible error-patterns in an array of the burst-error blocks is $\prod_{i=1}^{m} 2^{B-1}$ from Lemma 2.1. For each of the error-patterns of the burst-error blocks, $z^{-s}X_E(z)$ takes on one of the 2^s possible error-patterns from Lemma 2.2. Thus, $E_{SB}(z)$ takes on one of the $2^{(B-1)m+s}$ possible error-patterns for an array of the burst-error blocks. Let us derive the total number of the possible arrays of the burst-error blocks.

(i-1) If we let $v_2=v_3=\cdots=v_m=0$, the variable v_1 can takes on an integer between 0 and n-mB-s, i.e., the total number of the possible arrays is n-mB-s+1.

(i-2) If we let $v_3=v_4=\cdots=v_m=0$ and $Q_2(z)$ be shifted by v_2, the variable v_1 can take on an integer between 0 and n-mB-s $-v_2$. Thus, the total number of the possible arrays is

$\sum_{v_2=o}^{N_m}(N_m+1-v_2)$ where N_m=n-mB-s.

(i-3) Inducing this enumeration up to $Q_m(z)$, we obtain the total number of the possible arrays of the burst-error blocks as follows;

$$\sum_{v_m=o}^{v'_m}\ \sum_{v_{m-1}=o}^{v'_{m-1}}\cdots\sum_{v_2=o}^{v'_2}(N_m+1-\sum_{j=2}^{m}v_j) = \binom{N_m+m}{m} \qquad (2.29)$$

where

$$\begin{cases} v'_m = N_m = n-mB-s \\ v'_j = N_m - \sum_{i=j+1}^{m} v_i \ ; \quad 2 \leqq j \leqq m-1 \end{cases} \tag{2.30}$$

and we have used Lemma 2.3 in (2.29).

Therefore, letting $N_1(B,m,s)$ denote the total number of the possible error-patterns of the error polynomial $E_{SB}(z)$ with $Q_1(z)$ which is non-overlapping $z^{-s}X_E(z)$, we have

$$N_1(B,m,s) = 2^{(B-1)m+s}\binom{N_m+m}{m} \tag{2.31}$$

(ii) *In case of $Q_1(z)$ which is overlapping $z^{-s}X_E(z)$*

In this case, introducing a new parameter h which denotes the length of the overlap between $Q_1(z)$ and $z^{-s}X_E(z)$ instead of v_1, we consider for convenience sake that $z^{-s}X_E(z)$ starts from not the (n-s-1)st position but from the (n-s-1-h)th position so that m-ple burst errors exist. Then the total number of the possible error-patterns for an array of the burst-error blocks is $2^{(B-1)m+s-h}$. On the other hand, the total number of the possible arrays of the burst-error blocks $Q_2(z)$, $Q_3(z)$, $\cdots$, $Q_m(z)$ is given by

$$\sum_{v_m=o}^{v'_m+h} \sum_{v_{m-1}=o}^{v'_{m-1}+h} \cdots \sum_{v_3=o}^{v'_3+h} (N_m+1+h-\sum_{j=3}^{m} v_j) = \binom{N_m+h+m-1}{m-1} \tag{2.32}$$

Note that the range of h should be $1 \leqq h < B$ when $s+1 \geqq B$, and $1 \leqq h \leqq S$ when $s+1<B$ so that m-ple burst errors exist. Letting $N_2(B,m,s)$ denote the total number of the possible error-patterns of this

case, we have

$$N_2(B,m,s) = 2^{(B-1)m+s} \sum_{h=1}^{H} 2^{-h} \binom{N_m+h+m-1}{m-1} \qquad (2.33)$$

where

$$H = \begin{cases} B-1 \ ; & s+1 \geqq B \\ s \ ; & s+1 < B \end{cases} \qquad (2.34)$$

(iii) *In case of $Q_1(z)$ which appears as an end-around burst-error block*

Since any cyclic code can correct an end-around burst error, we should enumerate the case where $s+1<B$ and $Q_1(z)$ appears as an end-around burst-error block (A burst-error block is called an *end-around burst-error block* if it takes on positions starting from a higher position and ending with a lower position. Note that a burst error of the burst-error block does not always appear as an end-around burst error.). Let h' denote the length of the suffix of the end-around burst-error block $Q_1(z)$. Then the variable h' should satisfy the relation $1 \leqq h' \leqq B-s-1$ and $h' \leqq v_m$ so that m-ple burst errors exist. The total number of the possible arrays of the burst blocks $Q_2(z)$, $Q_3(z)$, $\cdots$, $Q_m(z)$ is given by

$$\sum_{v_m=h'}^{v'_m} \ \sum_{v_{m-1}=o}^{v'_{m-1}+h} \cdots \sum_{v_3=o}^{v'_3+h} (N_m+1+h-\sum_{j=3}^{m} v_j) = \binom{N_m+s+m-1}{m-1} \qquad (2.35)$$

where $h=h'+s$. Taking acount of $z^{-s}X_E(z)$ which is totally overlapped by $Q_1(z)$, we obtain the total number of the possible error-patterns in this case as follows;

$$N_3(B,m,s)=2^{(B-1)m}\sum_{h'=1}^{B-s-1}\binom{N_m+s+m-1}{m-1}$$

$$=(B-s-1)2^{(B-1)m}\binom{N_m+s+m-1}{m-1}; \quad s+1<B \tag{2.36a}$$

and

$$N_3(B,m,s)=0 \ ; \ s+1\geq B \tag{2.36b}$$

Finally let us summarize the above results in the following theorem.

<u>Theorem 2.4</u> An underlying cyclic (n,k) code of a coset code capable of correcting both a synchronization slippage of s ($|s|\leq S$) bits and M-ple additive burst errors of length B or less has a redundancy given by

$$n-k\geq \log_2 N(B,M,S) \tag{2.37}$$

where

$$N(B,M,S)=4(2^S-1)+\sum_{m=1}^{M}\{N_1(B,m,0)+N_3(B,m,0)\}$$
$$+2\sum_{m=1}^{M}\sum_{s=1}^{S}\sum_{i=1}^{3}N_i(B,m,s) \tag{2.38}$$

This bound on redundancy is general in the following sense.

(a) For S=0, it gives the bound on redundancy of a cyclic code capable of correcting M-ple burst errors of length B or less. By Benelli, Bianciardi and Cappellini [9], a bound on redundancy of a multiple-burst-error correcting code was derived. But this bound does not take acount of the end-around burst errors which can usually be corrected by the cyclic code, very large class of linear codes.

(b) For S=0 and M=2, our bound is tighter than the Reiger bound[8, pp.110] for a single burst-error correcting code.

(c) For S=0 and B=1, our bound coincides the Hamming bound.

2.5 Concluding Remarks

Let us compare the efficiency of the codes capable of correcting both a synchronization slippage of $s(|s| \leq B/2)$ bits and an additive burst error of length B or less with four class of codes, i.e., ideal coset codes whose number of check symbols meets the bound, the coset codes derived in Section 2.3, Tong's coset codes[6] and comma codes. In Figure 2.7, some of the upper bounds on the information rate are shown for the code length n=2000.

(i) From the bound of (2.38), the ideal coset code requires approximately B check symbols in order to correct the synchronization slippage, while the coset code derived in Section 2.3 requires approximately 2B check symbols. This is due to the fact that the latter code relies on the assumption that a systematic decoding method is used and that it does not take acount of the fact that $(z^{-s}-1)R(z)$ is the low density and with the fixed pattern.

(ii) Tong's coset code does not require check symbols for the correction of the synchronization slippage on the condition that the synchronization slippage and an additive burst error do not occur concurrently and the additive burst error does not occur as an end-around burst error. But the weak condition that the

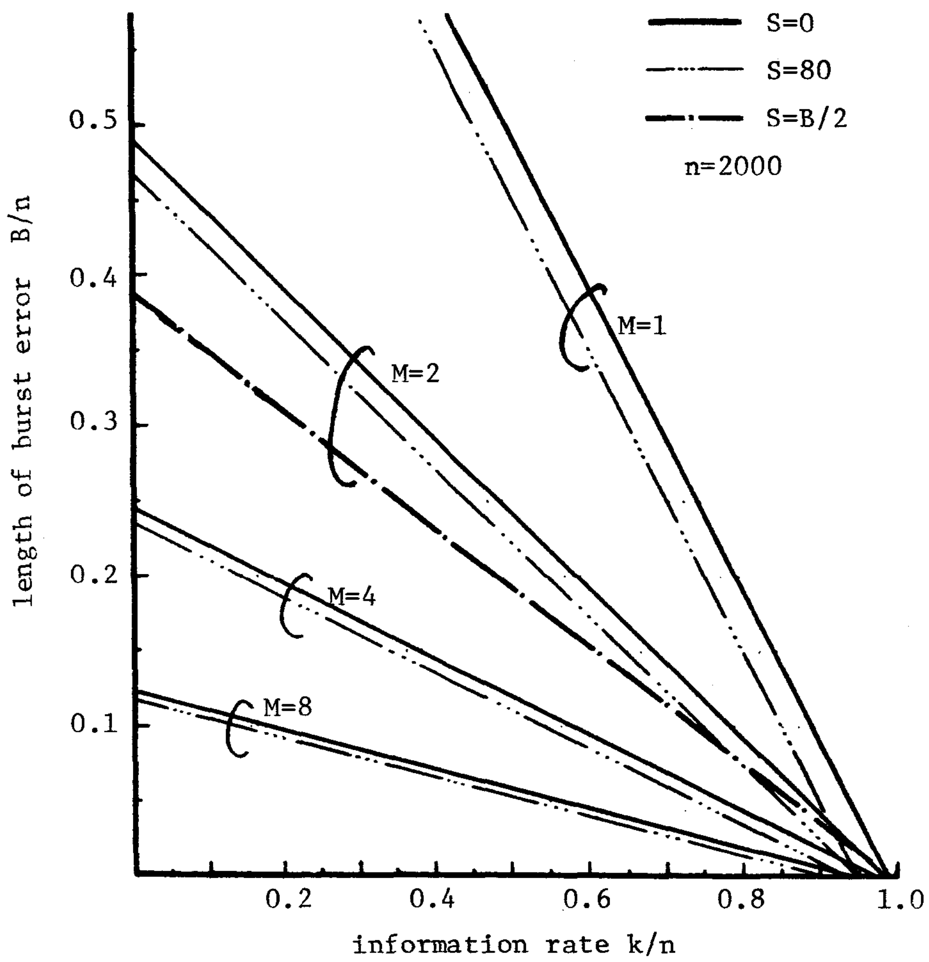


Figure 2.7 Upper bounds on information rate for various coset codes.

synchronization slippage and the additive burst error occur concurrently but not overlap each other, Tong's coset code does require B check symbols more. Therefore, we can conclude that the coset code derived in Section 2.3 further requires B check symbols more in order to release the weak condition.

(iii) A comma code, generated by inserting an autocorrelation function fetween code words of a burst-error correcting code, requires the autocorrelation function of length 4B. Therefore, this code is inferior to the coset code derived in Section 2.3.

The proposed coset code may be difficult to construct in practical use since it is hard to find the ideal multiple burst-error correcting code, but the principle of the codes will provide the fundamental concepts of the following chapters.

CHAPTER 3

INTERLEAVED CODES[10]

3.1 Introduction

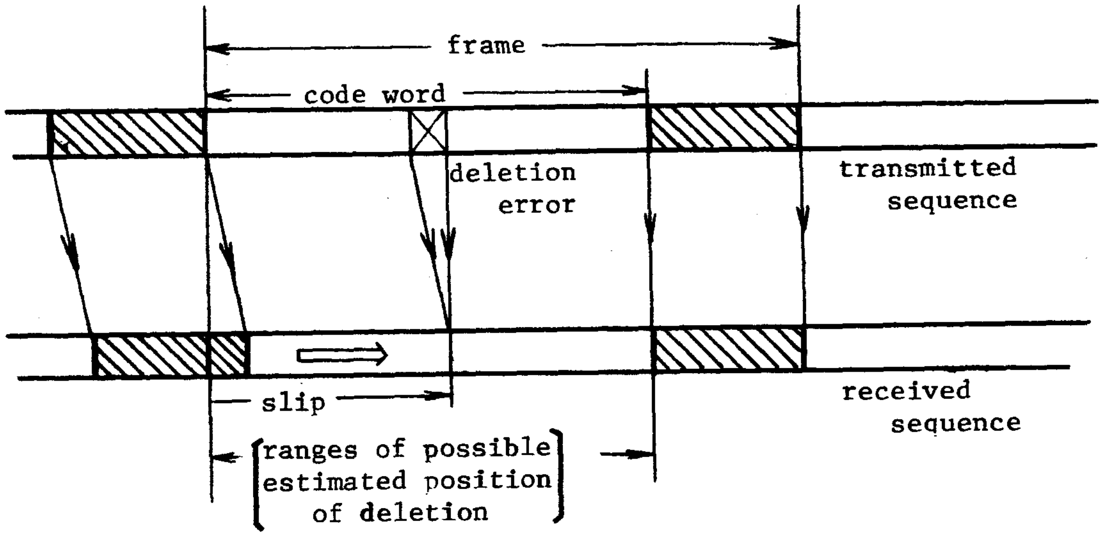
In the previous chapter we have constructed the coset codes which are capable of correcting both a synchronization slippage and additive burst errors. But the synchronization slippage is in general, due to the timing errors such as deletion or insertion errors occurred in the preceding code words. Since these codes have no capability of correcting original timing errors, a message of the received word corrupted by the timing errors may be totally lost. Therefore, it is desirable to construct timing-error correcting codes from not only practical but also information-theoretical standpoints. Sellers[11] and Ullman[12] constructed block codes capable of correcting a single deletion or insertion error, provided that no additive error exists in the received word. The codes are constructed by inserting the special sequence into an additive-error correcting code at fixed periodic intervals. Tanaka and Kasai[13] investigated the block codes capable of correcting timing errors besides additive errors based on the Levenshtein metric under the condition that both types of error do not exist in a received word. Greene[1] described a decoding algorithm, for any single burst-error correcting code, which can be used for correcting a few deletions within a single burst error.

This code should be properly synchronized at the start of the received word and has a possibility of false correction. Thus it is desirable to construct a code capable of correcting timing errors occurring at any position within a received word with possible additive errors.

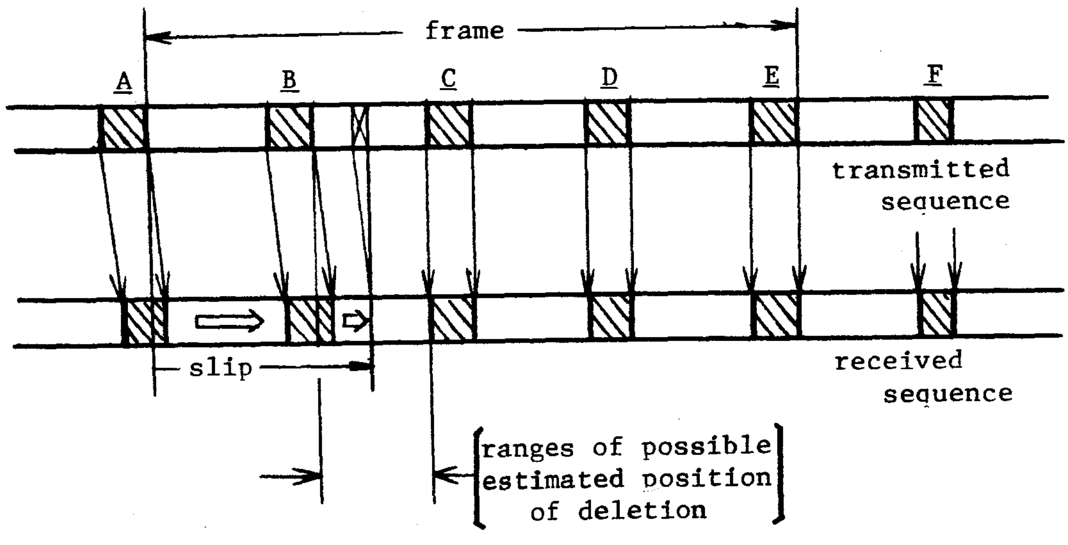
In this chapter, we present a new class of block codes capable of correcting both a deletion error and an additive burst error in order to provide the basic concept of timing- and additive-error correcting codes which are to be presented in the next chapter.

3.2 Heuristic Approach

In channels, the symbols can be deleted from or inserted into a code word, forcing the decoder misframe this received word and the subsequent words. One method of recovering synchronization in a communication system with such a channel, which we call *Method I*, requires the use of a special synchronization sequence (syn-sequence) which is inserted between code words (see Figure 3.1 (a)). This sequence may be one with the ideal autocorrelation function such as a Barker sequence[16, pp.465] or one with a fixed pattern[11]. In this case, even though the occurrence of a deletion error is detected by the syn-sequences, the position where the deletion error occurs is uncertain. This is due to the fact that the possible deleted position ranges over the entire code word as illustrated in Figure 3.1(a). Accordingly, a deletion error



(a) Method I



(b) Method II

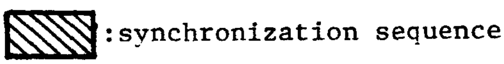


Figure 3.1 Ranges of possible estimated position of deletion due to two types of aligning synchronization sequences.

occurred at the central position of a code word causes a slippage in the remaining parts of the code word, resulting in a burst error of length approximately one half of the code length. Since the number of check symbols required to correct a single burst error of length b or less is approximately 2b from the argument of the section 2.4, we should use a very low rate code, which is impractical.

In order to improve the capability in estimating a deleted position, let us consider the other method, which we call *Method II*, where the special syn-sequences are inserted into an additive-error correcting code at fixed periodic intervals (see Figure 3.1(b)). When a deletion error occurs between the syn-sequences $\underline{B}$ and $\underline{C}$, a slippage of 1 bit is detected in the syn-sequences $\underline{A}$ and $\underline{B}$, and no slippage is detected in the syn-sequences $\underline{C}$, $\underline{D}$ and $\underline{E}$. Therefore, we can assume that a deletion error has occurred in the subblocks between the syn-sequences $\underline{B}$ and $\underline{C}$. Since the possible deleted position ranges over the region between the syn-sequences $\underline{B}$ and $\underline{C}$, we can correct the deletion error by using a block code capable of correcting an additive burst error of length less than or equal to the length of the subblocks.

However, Method II is inferior to Method I with respect to the capability of distinguishing additive errors from deletion errors, since the length of each syn-sequence of the former is less than that of the latter when the total length of syn-sequence per word is made equal. In what follows, we shall present a new method of using an appropriate coset code generated by the cyclic code as the syn-sequences of Method II.

3.3 Preliminaries

Let us consider a binary cascaded channel which consists of two channels as shown in Figure 3.2. The first is the channel with additive errors and the second, with deletion errors. Let us represent a code word consisting of the N input symbols to the cascaded channel as $A=(\alpha_0, \alpha_1, \cdots, \alpha_{N-1})$ or $A(z)=\sum_{i=o}^{N-1} \alpha_i z^i$ where $\alpha_i \in GF(2)$, $0 \leq i \leq N-1$. Similarly, let us represent the output word from the cascaded channel as $B=(\beta_0, \beta_1, \cdots, \beta_{N-1})$ or $B(z)=\sum_{j=o}^{N-1} \beta_j z^j$ where $\beta_j \in GF(2)$, $0 \leq j \leq N-1$. We assume that α_{N-1} is the first symbol of the code word being transmitted and that the successive code words are transmitted in a continuous manner.

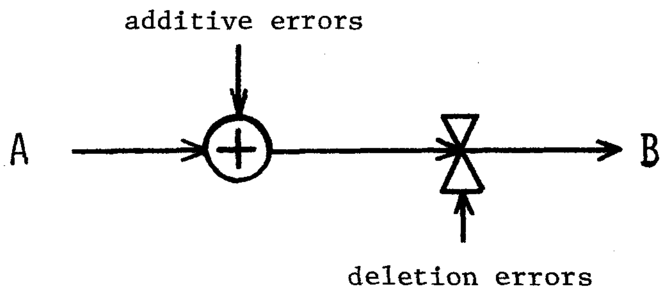


Figure 3.2 Cascaded channel with additive errors and deletion errors.

We shall denote the output word corrupted by a deletion error of $\alpha_{\lambda+1}$, $-1 \leqq \lambda \leqq N-2$, with no additive error by the vector A_λ or polynomial $A(z|\lambda)$ as follows;

$$B(z)=A(z|\lambda)= \alpha'_{N-1}+\sum_{j=1}^{\lambda+1} \alpha_{j-1}z^j + \sum_{j=\lambda+2}^{N-1} \alpha_j z^j \tag{3.1}$$

where α'_{N-1} is the first symbol of the subsequently transmitted code word following A. Let $A(z|\lambda;\hat{\lambda})$, $1 \leqq \hat{\lambda} \leqq N$, represent the word that is formed by inserting a symbol $u_{\hat{\lambda}-1} \in GF(2)$ between the $(\hat{\lambda}-1)$st and the $\hat{\lambda}$th positions of the received word $B(z)=A(z|\lambda)$. In what follows, the term *position* means an exponent of z in any polynomial. Let $B(z|\hat{\lambda})$ be defined as follows;

$$B(z|\hat{\lambda}) \triangleq A(z|\lambda;\hat{\lambda}) \tag{3.2}$$

For $\hat{\lambda}$, $1 \leqq \hat{\lambda} \leqq \lambda+2$, $B(z|\hat{\lambda})$ can be represented by

$$B(z|\hat{\lambda})=\sum_{j=o}^{\hat{\lambda}-2} \alpha_j z^j + u_{\hat{\lambda}-1} z^{\hat{\lambda}-1} + \sum_{j=\hat{\lambda}}^{\lambda+1} \alpha_{j-1} z^j + \sum_{j=\lambda+2}^{N-1} \alpha_j z^j \tag{3.3}$$

Let us define a subset C_X of a cyclic (n_x,k_x) code which is capable of detecting a burst error of length n_x-k_x-f or less, or of correcting a burst error of length f or less as follows;

$$C_X \triangleq \{X=(\xi_0, \xi_1, \cdots, \xi_{n_x-1}) | X(z)=\sum_{i=o}^{n_x-1} \xi_i z^i \equiv 0 \bmod G_X(z)\} - \{ \mathbf{0}^{n_x}, \mathbf{1}^{n_x}\} \tag{3.4}$$

where $\xi_i \in GF(2)$, $0 \leqq i \leqq n_x-1$, and $G_x(z)$ is the generator polynomial of the cyclic (n_x,k_x) code which will be specified later. The $\mathbf{0}^{n_x}$ and $\mathbf{1}^{n_x}$ are the n_x-dimensional all 0's and all 1's vectors, respectively. Let us also define a block code $\mathcal{C}_U$ which is capable of correcting a burst error of length g or less as follows;

$$\mathcal{C}_U \triangleq \{U=(\mu_0, \mu_1, \cdots, \mu_{m_u-1})\} \tag{3.5}$$

where $\mu_j \in GF(2)$, $0 \leqq j \leqq m_u-1$. We assume that code lengths n_x and m_u are given by

$$n_x = fT \qquad \text{for some integer } f \geqq k_x \tag{3.6}$$

and

$$m_u = gT \qquad \text{for any integer } g \tag{3.7}$$

respectively, where T is a positive integer whose role will be apparent later.

Let us divide each code word of the code $\mathcal{C}_X$ into T subblocks of length f. The subblocks can be represented by

$$\boldsymbol{\xi}_t = (\xi_{tf}, \xi_{tf+1}, \cdots, \xi_{tf+f-1}); \qquad 0 \leqq t \leqq T-1 \tag{3.8}$$

Using these f-dimensional vectors, we define two kinds of (f+1)-dimensional vectors as follows;

$$x_t^* = (\xi_{tf-1}, \boldsymbol{\xi}_t); \qquad 2 \leqq t \leqq T-1 \tag{3.9}$$

and

$$x_t^{**} = (\xi_{tf}, \boldsymbol{\xi}_t) + r_0; \qquad t=0 \text{ and } 1 \tag{3.10}$$

where

$$r_0 = (0, 1, 0^{f-1}) \tag{3.11}$$

In the above equations and in what follows, each suffix of the components is evaluated by modulo n_x. The vector r_0 is used for generating the coset code on the basis of the code $\mathcal{C}_X$.

Similarly, let us divide each code word of the code $\mathcal{C}_U$ into T subblocks of length g. The subblocks can be represented by

$$\boldsymbol{\mu}_t = (\mu_{tg}, \mu_{tg+1}, \cdots, \mu_{tg+g-1}); \qquad 0 \leq t \leq T-1 \tag{3.12}$$

Using the vectors x_t^*, x_t^{**} and $\boldsymbol{\mu}_t$, let us construct a new class of the interleaved codes as follows;

$$\mathcal{C}_{0_T} \triangleq \{A=(x_0^{**}, \boldsymbol{\mu}_0, x_1^{**}, \boldsymbol{\mu}_1, x_2^*, \boldsymbol{\mu}_2, \cdots, x_{T-1}^*, \boldsymbol{\mu}_{T-1})\} \tag{3.13}$$

where the length of the code $\mathcal{C}_{0_T}$ is given by

$$N = n_x + m_u + T \tag{3.14}$$

Note that x_t^{**} is used only for t=0 and 1 in constructing the code $\mathcal{C}_{0_T}$. The construction of the code $\mathcal{C}_{0_T}$ is illustrated in Figure 3.3.

From the received word B, we can de-interleave the following two vectors Y and V corresponding to $X \in \mathcal{C}_X$ and $U \in \mathcal{C}_U$, respectively;

$$Y = [B + R_0]W_{0_X} \tag{3.15}$$

and

$$V = B\, W_{0_U} \tag{3.16}$$

where

$$R_0 = (r_0, 0^g, r_0, 0^g, 0^{(f+g+1)(T-2)}) \tag{3.17}$$

and W_{0_X} is the N by n_x matrix

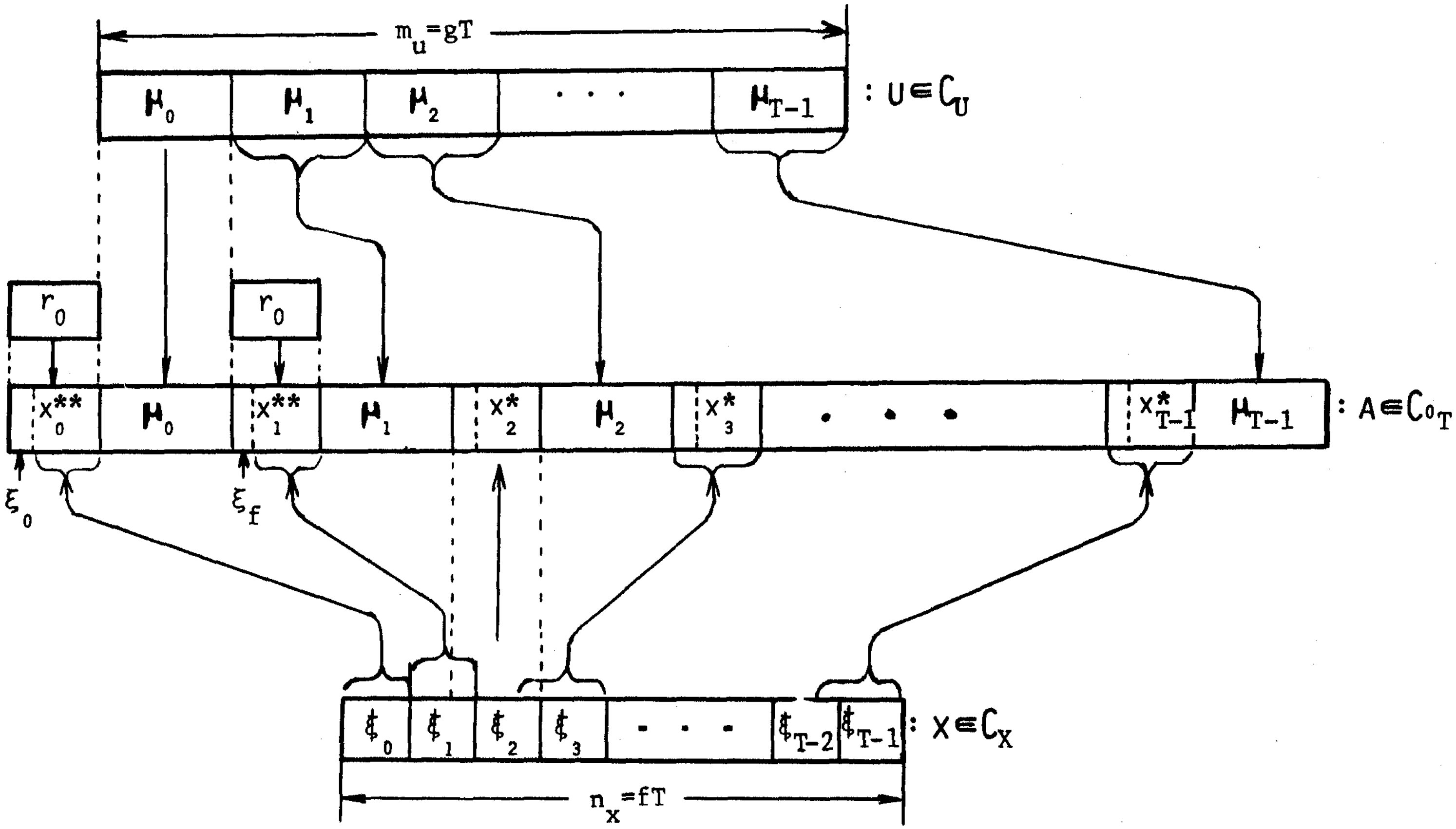
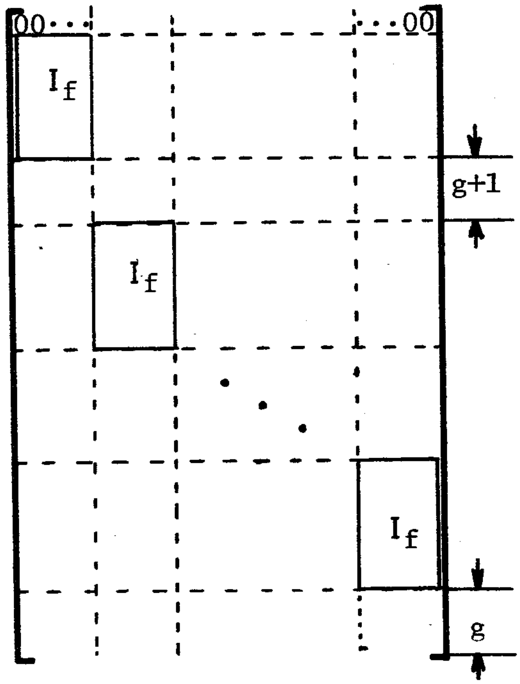


Figure 3.3 Construction of code C_{0_T}.

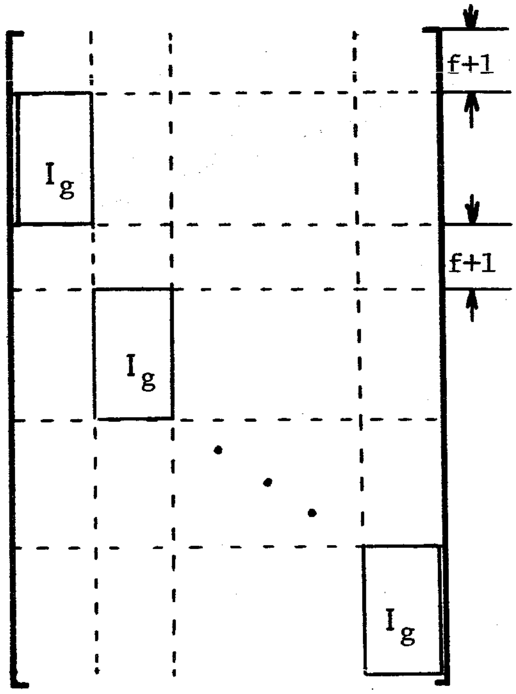
W_{0_X} =



(3.18)

and W_{0_U} is the N by m_u matrix;

W_{0_U} =



(3.19)

In (3.18) and (3.19), I_j is the j by j identity matrix and the blank areas in the matrices indicate zero values.

3.4 Capabilities of The Code C_{0_T}

3.4.1 Correction of Either A Deletion Error or An Additive Burst Error

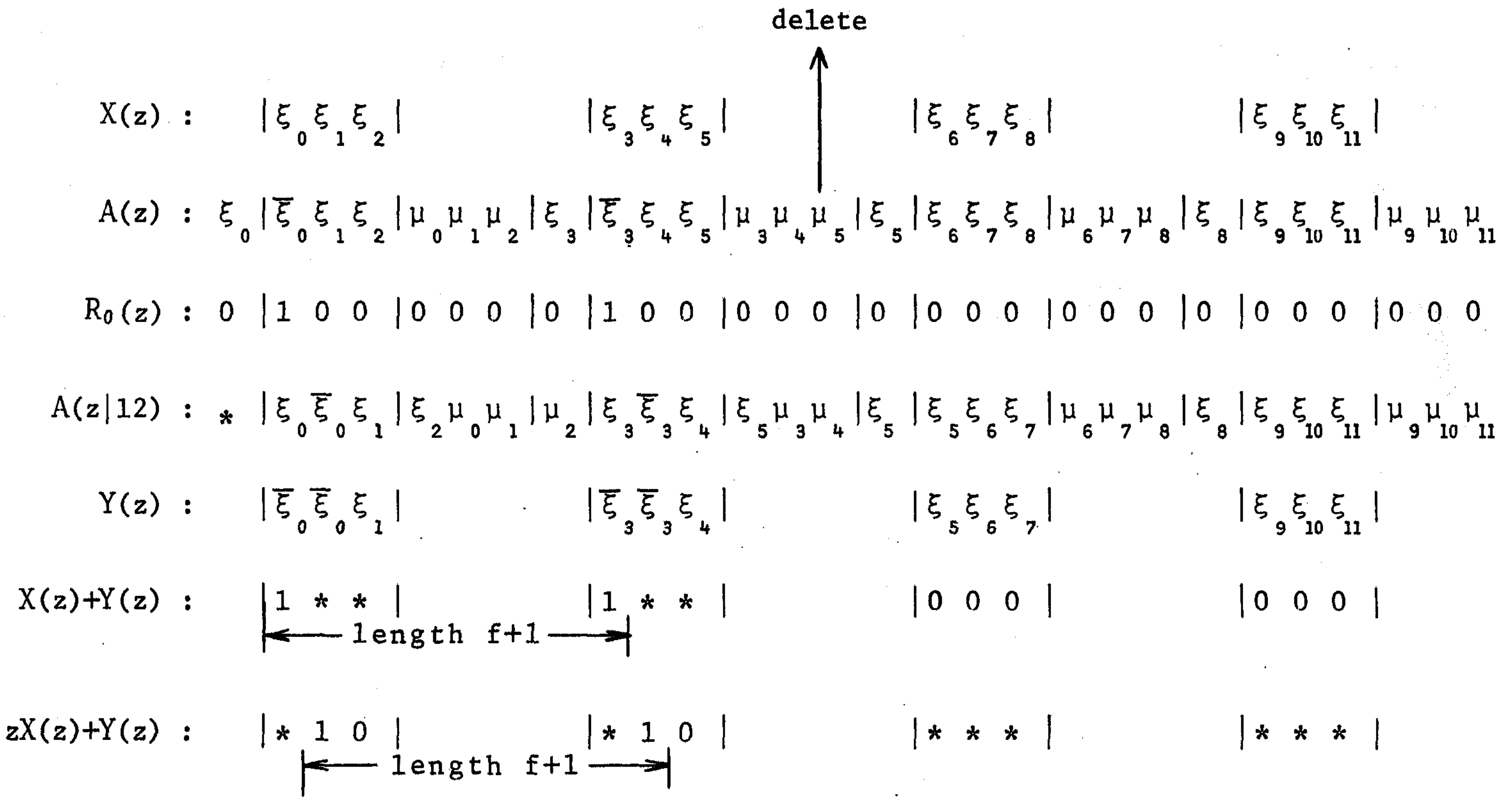
<u>Definition 3.1</u> Consider a set of the possible burst errors due to a deletion error in a received word. We refer to the burst error of the minimum length in the set as *pseudo burst error*, and use the symbol ρ to denote the length.

The pseudo burst error will play an important role in the deletion-error correction as we shall see soon.

When the received word B is corrupted by a deletion error, the vector R_0 and the coset leaders[16, pp.463] r_0 in X_0^{**} and X_1^{**} (together with the suffixes of X_t^{*}, $t=2,3,\cdots, T-1$) cause a pseudo burst error of length f+1 or more which is not correctable but detectable. Figure 3.4 shows an example of this type of error. We see that the pseudo burst error of length at least f+1=4 due to the deletion error exists in either $X(z)$ or $zX(z)$. Thus in decoding we can exploit this property to decide whether a deletion error has occurred or not, provided that the length of any additive burst error in Y is at most f.

In the following we shall give an outline of the basic decoding algorithm whose complete version is given in APPENDIX I.

When the de-interleaved word Y from the received word B cannot be decoded by assuming that any additive burst error of length f or less occurred, the decoder then assumes that a deletion error has occurred. Then the decoder inserts a symbol $u_{\hat{\lambda}-1}$(0 or 1) at



$Y=[A_{12}+R_0]W_{0_X}$, f=3, g=3, T=4 and (*)=0 or 1.

Figure 3.4 Example of pseudo burst error due to a deletion error.

each position, $\hat{\lambda}$=f+g+3, ···, N, and tries to correct a burst error. When no error is detected or a burst-error correction is successful, both decoding of Y and correction of the deletion error of B are completed. Finally de-interleaving V from B with no deletion error, the decoder can decode V by assuming that an additive burst error might have occurred.

We shall give the following theorem.

Theorem 3.1 The code C_{0_T} can correct either a deletion error or an additive burst error of length b=f+g+1 or less, provided that the relation $T \geqq 6$ holds.

Before giving the proof of Theorem 3.1, we shall give the following lemmas and properties.

Lemma 3.1 If $B = A_\lambda$ for λ, $f+g+2 \leqq \lambda+1 \leqq N-1$, the length of the pseudo burst error of $Y = [B + R_0]W_{0_X}$ is given by

$$f+1 \leqq \rho \leqq \left[\frac{n_X+f+2}{2}\right] \tag{3.20}$$

(proof) Consider the received word $Y=[A_\lambda+R_0]W_{0_X}$, λ=f+g+1, whose components ξ_f of X is deleted and to which R_0 is added erroneously as illustrated in Figure 3.5(a). This word can be represented by

$$Y(z)=\xi_0+\sum_{i=1}^{f-1}\xi_{i-1}z^i+\sum_{i=f}^{n_X-1}\xi_i z^i+1+z+z^f \tag{3.21}$$

The pseudo burst error is then given by

$$E(z)=X(z)+Y(z)=1+z+z^f+\sum_{i=1}^{f-1}(\xi_i+\xi_{i-1})z^i \tag{3.22}$$

X(z): $|\xi_0\xi_1\cdots\xi_{f-1}|$ $|\xi_f\xi_{f+1}\cdots\xi_{2f-1}|$ $|\boldsymbol{\xi}_2|$ $|\boldsymbol{\xi}_{T-1}|$

A(z): $\xi_0|\overline{\xi}_0\xi_1\cdots\xi_{f-1}|\ \boldsymbol{\mu}_0\ |\ \xi_f|\overline{\xi}_f\xi_{f+1}\cdots\xi_{2f-1}|\ \boldsymbol{\mu}_1\ |\ x_2^*\ |\ \boldsymbol{\mu}_2\ |\cdots\cdots|\ x_{T-1}^*\ |\ \boldsymbol{\mu}_{T-1}$

A(z|f+g+1): $*\ |\xi_0\overline{\xi}_0\cdots\xi_{f-2}|\xi_{f-1}\ \boldsymbol{\mu}_0\ |\xi_f\xi_{f+1}\cdots\xi_{2f-1}|\ \boldsymbol{\mu}_1\ |\ x_2^*\ |\ \boldsymbol{\mu}_2\ |\cdots\cdots|\ x_{T-1}^*\ |\ \boldsymbol{\mu}_{T-1}$

Y(z): $|\overline{\xi}_0\overline{\xi}_0\cdots\xi_{f-2}|$ $|\overline{\xi}_f\xi_{f+1}\cdots\xi_{2f-1}|$ $|\boldsymbol{\xi}_2|$ $\cdots$ $|\boldsymbol{\xi}_{T-1}|$

X(z)+Y(z): $|1\ *\cdots*\ |$ $|1\ 0\cdots 0\ |$ $|\ \mathbf{0}^f\ |$ $\cdots\ |\ \mathbf{0}^f\ |$

$\leftarrow$ pseudo burst error $\rightarrow$

Figure 3.5(a) Pseudo burst error due to deletion of ξ_f.

Equation (3.22) shows that there exists a burst error of length f+1 in Y for any $X \in C_X$. As λ, $\lambda > f+g+2$, increases, the length of the pseudo burst error in the suffix of the code word, clearly, does not decrease. If a deletion error occurs at a higher position, it is required to assume that the suffix of Y is in the correct phase and the prefix is shifted by 1 bit in the lower direction so that it is reduced to the pseudo burst error. That is, the pseudo burst error appears as an end-around burst error, $zX(z)+Y(z)$, as illustrated in Figure 3.5(b). Thus a part of the pseudo burst error, of length at least f+1, appears in the subblocks ξ_0 and ξ_1. From the above argument, we can conclude that the maximum length of the pseudo burst error of A_λ, $f+g+2 \leq \lambda+1 \leq N-1$, is given by letting the length of the pseudo burst error in the suffix be equal to that of the end-around burst error as shown in Figure 3.6. We then have

$$\max \rho = \min_{\zeta} [\ \zeta+2,\ n_X-(\zeta+2)+f+2\ ] = \left[\frac{n_X+f+2}{2}\right] \tag{3.23}$$

Q.E.D.

When the received word B is corrupted by a deletion error with no additive error, i.e., $B = A_\lambda$, the pseudo burst error of $Y = [B + R_0]W_{0_X}$ can be corrected by letting $\hat{\lambda}=\lambda+2$ and inserting a proper symbol $u_{\hat{\lambda}-1}$ between the $(\hat{\lambda}-1)$st and the $\hat{\lambda}$th positions of A_λ. But as we shall see later, the value of $\hat{\lambda}$ at which the pseudo burst error is corrected is not necessarily unique, when there is a consecutive run of 0's or 1's in the part of the code word where

$$X(z):\quad |\xi_0\xi_1\cdots\xi_{f-1}|\qquad |\xi_f\xi_{f+1}\cdots\xi_{2f-1}|\qquad |\xi_{(T-2)f}\cdots\xi_{(T-1)f-1}|\qquad |\xi_{(T-1)f}\cdots\xi_{Tf-1}|$$

$$A(z):\xi_0|\overline{\xi}_0\xi_1\cdots\xi_{f-1}|\mu_0|\xi_f\ |\overline{\xi}_f\xi_{f+1}\cdots\xi_{2f-1}|\cdots|\xi_{(T-2)f}\cdots\xi_{(T-1)f-1}|\mu_{T-2}\xi_{(T-1)f-1}|\xi_{(T-1)f}\cdots\xi_{Tf-1}|\mu_{T-1}$$

$$A(z|Tf-3):\ *|\xi_0\overline{\xi}_0\cdots\xi_{f-2}|\xi_{f-1}\mu_0|\xi_f\overline{\xi}_f\cdots\xi_{2f-2}|\cdots|\xi_{(T-2)f-1}\cdots\xi_{(T-1)f-2}|\xi_{(T-1)f-1}\mu_{T-2}|\xi_{(T-1)f-1}\cdots\xi_{Tf-3}\xi_{Tf-1}|\mu_{T-1}$$

$$Y(z):\quad |\overline{\xi}_0\overline{\xi}_0\cdots\xi_{f-2}|\qquad |\overline{\xi}_f\overline{\xi}_f\cdots\xi_{2f-2}|\cdots|\xi_{(T-2)f-1}\cdots\xi_{(T-1)f-2}|\qquad |\xi_{(T-1)f-1}\cdots\xi_{Tf-3}\xi_{Tf-1}|$$

$$zX(z)+Y(z):\quad |*\ 1\ 0\cdots 0|\qquad |*\ 10\cdots 0|\cdots|0\cdots\cdots 0|\qquad |0\cdots 0\ \ *\ \ |$$

—pseudo burst error → | ... | ←

Figure 3.5(b) Pseudo burst error due to deletion of ξ_{Tf-2}.

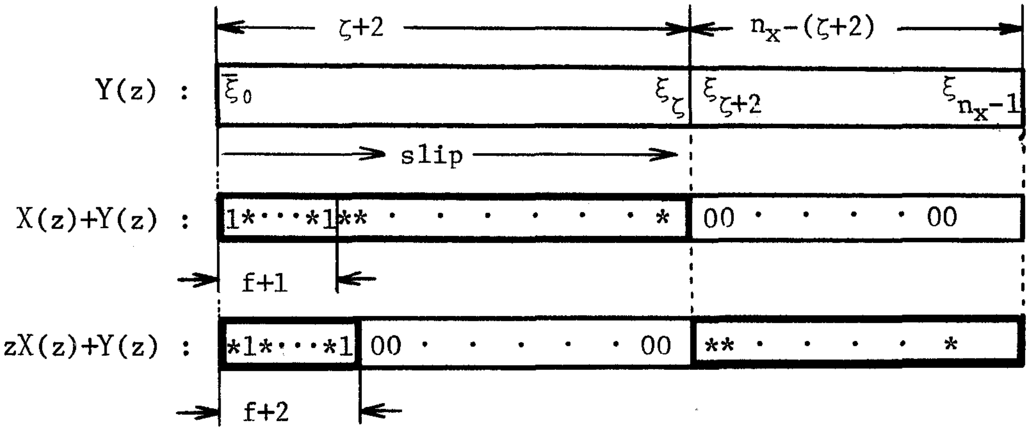


Figure 3.6 Two ways of taking pseudo burst error.

the deletion error occurs. In this case, even if the decoding of Y from $B_{\hat{\lambda}}$ with the insertion at $\hat{\lambda}=\lambda+2$ is successful, there may remain a partial slippage in $V=B_{\hat{\lambda}}W_{0_U}$ which has at present no assurance of being corrected. In this respect we shall give the following lemma.

<u>Lemma 3.2</u> The amount of the deviation from the position of deletion to the position where the correction is made is given by

$$\lambda-\hat{\lambda} \leqq f+g+2 \tag{3.24}$$

provided $f \geqq k_X$.

We shall present the fundamental properties of cyclic codes before giving the proof of Lemma 3.2.

<u>Property 3.1</u> In any cyclic (n,k) code the length of the consecutive run of 0's is at most k-1 in any code word except the all 0's word.

(proof) In any cyclic code the existence of the code word with k or more consecutive run of 0's implies the existence of the code word of degree n-k-1 or less, which is less than the degree of the generator polynomial, yielding the contradiction. Q.E.D.

<u>Property 3.2</u> In any cyclic (n,k) code the length of the consecutive run of 1's is at most k in any code word except the all 1's word.

(proof) Suppose that there exists a code word with k+1 or more consecutive run of 1's. This word can be represented by

$$F(z)=\sum_{i=o}^{J-1}\phi_i z^i+\sum_{i=J}^{J+k} z^i+\sum_{i=J+k+1}^{n-1}\phi_i z^i \quad \text{for any integer } J \tag{3.25}$$

where $\phi_i \in GF(2)$. The cyclically shifted version, $zF(z)$, is also a code word. Then $F(z)+zF(z)$ is also a code word and is given by

$$F(z)+zF(z) = \sum_{i=o}^{J}(\phi_i+\phi_{i-1})z^i + \sum_{i=J+1}^{J+k}(1+1)z^i + \sum_{i=J+k+1}^{n-1}(\phi_i+\phi_{i-1}) \tag{3.26}$$

which has a k or more consecutive run of 0's, contradicting Property 3.1. Q.E.D.

(proof of Lemma 3.2) Consider a deletion error occurred in the range $[t(f+g+1), (t+1)(f+g+1)]$, $1 \leqq t \leqq T-1$, and an insertion of $u_{\hat{\lambda}-1}$ between the $(\hat{\lambda}-1)$st and the $\hat{\lambda}$th position of B. Let the parameter θ, which indicates the position of the deleted symbol $\xi_{\theta+1}$ of X, be given by

$$\theta = \lambda-(g+1)t-1 \tag{3.27a}$$

when the deletion error occurred in x_t^*, or

$$\theta = ft+f-2 \tag{3.27b}$$

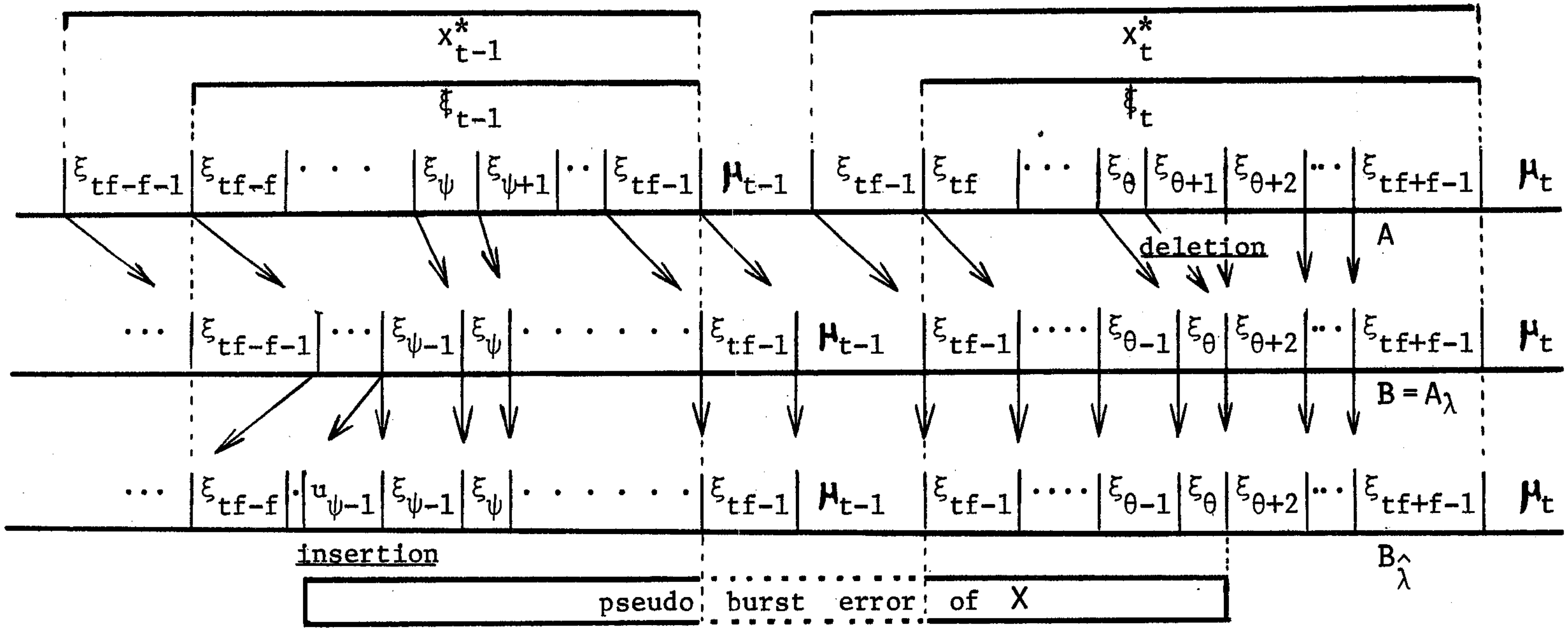
when it occurred in μ_t. Then from Figure 3.7 we see that the error polynomial of $Y(z)$ is represented by

$$E(z) = (u_{\psi-1}+\xi_{\psi-1})z^{\psi-1} + \sum_{i=\psi}^{\phi+1}(\xi_i+\xi_{i-1})z^i \; ; \; \hat{\lambda} \leqq \lambda+2 \tag{3.28}$$

where

$$\hat{\lambda} = \psi+1+(g+1)[\psi/f] \tag{3.29}$$

Thus in the case where $E(z)=0$ holds for some $X(z) \in C_X$, the relation $u_{\psi-1} = \xi_{\psi-1} = \xi_\psi = \cdots = \xi_\phi = \xi_{\phi+1}$ holds. From Property 3.2, the length of any consecutive run of 1's of $X \in C_X$ is



$\theta = \lambda - (g+1)t - 1$

$\psi = \hat{\lambda} - (g+1)[\psi/f] - 1$

Figure 3.7 Pseudo burst error due to erroneous insertion for deletion error.

at most k_X, i.e., $\theta+1-(\psi-2)=\theta-\psi+3 \leqq k_X$. Since the correct insertion is $\psi=\theta+2$, the upper-bound on the amount of the deviation due to the erroneous estimation is given by $\theta-\psi+2 \leqq k_X-1$. In the worst case, $\theta-\psi+2=k_X-1$, we have the following relation for $\theta \in [tf, tf+f-2]$

$$(t-1)f < tf-k_X+3 \leqq \psi \tag{3.30}$$

The left inequality of (3.30) comes from the restriction $f \geqq k_X$ in (3.6). Equality in (3.30) implies that the insertion, by which the decoding of Y is completed for the first time, may be made in the subblock ξ_{t-1}. Since the subblock μ_{t-1} is located between the subblocks ξ_{t-1} and X_t^*, the amount of the deviation due to the erroneous estimation in a code word of C_{0_T} is the sum of the length of the sequence cosisting of μ_{t-1} and one symbol ξ_{tf-1} of X_t^* and the length k_X-1, yielding (3.24). Q.E.D.

(proof of Theorem 3.1) By the assumption we can exclude the case that both an additive burst error and a deletion error occur concurrently in a code word. For an additive burst error of length $f+g+1$ or less, we can correct it by the capability of the codes C_X and C_U. For a deletion error, let us consider two cases depending upon the deleted position .

(i) For λ, $0 \leqq \lambda+1 \leqq f+g+1$, there exists a pseudo burst error of length f or less in $Y=[A_\lambda+R_0]W_{0_X}$, which is correctable as an additive burst error. The de-interleaved word $V=A_\lambda W_{0_U}$ can be successfully decoded by assuming that an additive burst error has occurred.

(ii) For λ, $f+g+2 \leqq \lambda+1 \leqq N-1$, there exists a pseudo burst error of length at least $f+1$ and at most $\left\lceil \frac{n_X+f+2}{2} \right\rceil$ in $Y=[A_\lambda+R_0]W_{0_X}$ from Lemma 3.1. When the following inequality is satisfied.

$$n_X - k_X - f \geqq n_X - 2f \geqq \frac{n_X+f+2}{2} \geqq \left\lceil \frac{n_X+f+2}{2} \right\rceil \qquad (3.31)$$

i.e., $n_X/f = T \geqq 6$, the code C_X can successfully detect the pseudo burst error, distinguishing a deletion error from an additive burst error. After the insertion by which the decoding of Y is completed for the first time, there may exist a burst error of length g or less in V from Lemma 3.2. Since the code C_U can correct the burst error of length g or less, we can successfully decode the received word B. Q.E.D.

3.4.2 Correction of A Deletion Error Imbedded within An Additive Burst Error

Consider a deletion error imbedded within an additive burst error. In this case there may occur the following problems.

Problem 1 The length of a pseudo burst error due to a deletion error may be decreased by an additive burst error.

···1 0 0 1 1 1 0 1 0 1 1 0···;A - - - -transmitted sequence

····0 0 0 0 0 1 1 1 1 1 0 0···;E - - - -additive burst error

···1 0 0 1 1 0 1 0 1 0 1 0···;A+E - - -output sequence of the first channel

······1 0 0 1 1 0 1 0 1 1 0··· $(A+E)_\lambda$ - -output sequence of the second channel

·····1 0 1 0 0 0 0 0 0 0 0···;$A+(A+E)_\lambda$ -error sequence

(pseudo) burst error

Figure 3.8 An example of related Problem 1.

<u>Problem 2</u> An insertion, by which the decoding of Y is completed for the first time, may be made at a lower position by f bits (f+g bits with respect to the received word B) beyond the range (3.24) in Lemma 3.2. because the additive burst-error correction in addition to the deletion-error correction is performed in this case even if there exists no additive error.

For these reasons we must modify the basic decoding algorithm described in Section 3.4.1 as follows;

M1 : Try to insert at

$$\hat{\lambda}_t = t(f+g+1)+1; \quad t=1,2,\cdots, T-1 \tag{3.32}$$

M2-1: Detect errors of Y for each insertion.

M2-2: If errors are detected in Y for all insertions, try to correct a burst error of length f or less for each insertion.

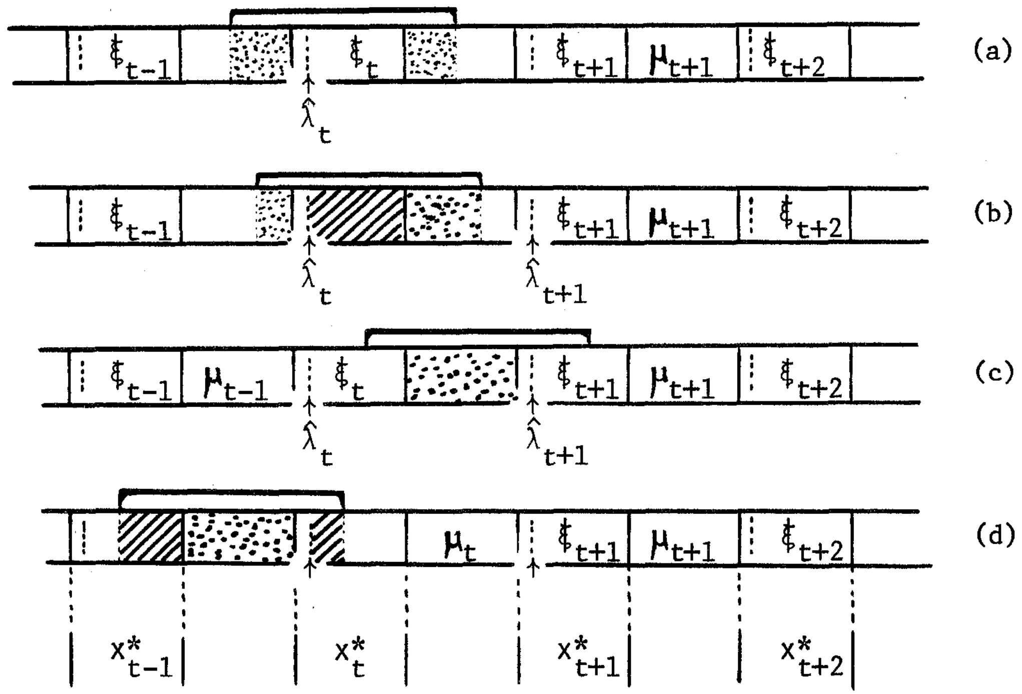
In APPENDIX I, the complete decoding algorithm is given, taking acount of these modifications.

First, we show that the code C_{0_T} is able to detect a deletion error, even when an additive burst error exists. For a deletion error imbedded within an additive burst error of length f+g+1 or less, the length of a pseudo burst error is varied by ±f bits at most. Therefore, the following relation;

$$n_X - k_X - f \geqq n_X - 2f \geqq \frac{n_X + f + 2}{2} + f \geqq \left[\frac{n_X + f + 2}{2}\right] + f \qquad (3.33)$$

i.e., $n_X/f = T \geqq 8$ should hold so that the code C_X can detect the pseudo burst error successfully. On the other hand, the pseudo burst error of length f or less can occur when additive errors occur over the subblocks ξ_0 and ξ_1, and a deletion error occurs, which causes those errors in ξ_1 no error by the slippage, but leaves errors in ξ_0 due to the additive errors and the slippage. Therefore, when a burst-error correction of Y is made successfully assuming that there exists a burst error in only the subblock ξ_0, we should decode V assuming that errors occur not only in the subblock μ_0 but also in μ_1.

Second, we show that the code C_{0_T} is able to correct a deletion error imbedded within an additive burst error of length f+g+1 or less. These errors can be classified into four types of error according to the relative position of the additive burst error with respect to the final position of the insertion as shown in Figure 3.9. In the case (a), there exists no error in the subblock ξ_t



:additive burst error of length f+g+1 with deletion error.

:errors in X. :errors in U.

Figure 3.9 Insertion for deletion error imbedded within an additive burst error.

but a slippage of 1 bit in the subblocks ξ_i, $i<t$, so that the decoding of Y is completed only by the insertion at $\hat{\lambda}_t$ in the detection mode of M2-1. After the decoding of Y there may exist a burst error in the subblocks μ_{t-1} and μ_t.
In the case (b), there exist errors in ξ_t and a slippage of 1 bit in ξ_i, $i<t$, so that errors are detected by the proper insertion at $\hat{\lambda}_t$. If errors are detected by the next insertion at $\hat{\lambda}_{t+1}$, then errors are detected for all the remaining insertions, resulting in the successful transition to the correction mode of M2-2. Thus the errors in ξ_t as well as the slippage are corrected by the insertion at $\hat{\lambda}_t$. Otherwise, by the final insertion at $\hat{\lambda}_{t'}$, there may exist a burst error in $\mu_{t'-2}$ and $\mu_{t'-1}$ where $t'=t+1$.
Similarly as we see in the cases (c) and (d), there may exist a burst error in μ_{t-1} and μ_t by the final insertion at $\hat{\lambda}_t$.
Therefore, the code C_U should correct a burst error over the subblocks μ_{t-2}, μ_{t-1} and μ_t when the final insertion at $\hat{\lambda}_t$ is made, i.e., consequently it should be capable of correcting a burst error of length 3g or less.

Now we give the following property of burst-error correcting codes (This is based on the erasure correction capability of the burst-error correcting code. See also Berlekamp [15, pp.393].).

<u>Property 3.3</u> A code which can detect a single burst error of length 3g or less is capable of correcting a single burst error of length 3g or less, provided that the position of the burst error is known.

In what follows, let $\mathcal{C}o_{TB}$ denote the code $\mathcal{C}o_T$ if the number of check symbols of the code $\mathcal{C}_U$ is 3g or more. Thus we have the following theorem.

<u>Theorem 3.2</u> The code $\mathcal{C}o_{TB}$ can correct a deletion error imbedded within an additive burst error of length b=f+g+1 or less, provided that the relation $T \geq 8$ holds.

3.5 Conclusion

Using the concepts of an interleaved method, Method II, discussed in Section 3.2, we have constructed the interleaved code, first, $\mathcal{C}o_T$ capable of correcting either a deletion error or an additive burst error, and second, $\mathcal{C}o_{TB}$ capable of correcting a deletion error imbedded within an additive burst error.

CHAPTER 4

EXTENDED INTERLEAVED CODES[14]

4.1 Introduction

In this chapter we shall present the extended interleaved code $\mathcal{C}_T$. We shall show that the new code $\mathcal{C}_T$ can be constructed for correcting an arbitrary number of deletion errors as well as for correcting either deletion or insertion errors in a received word.

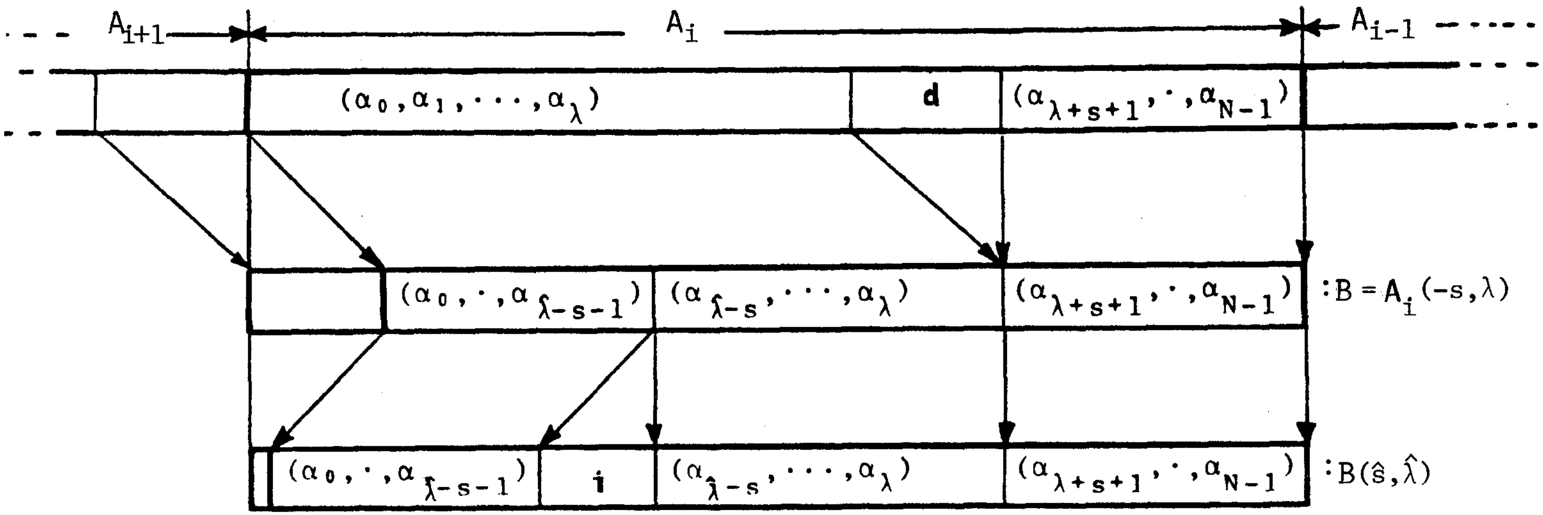
In Sections 4.2 and 4.3, we shall treat only deletion errors, and in Section 4.4 we shall show that the proposed code can also correct insertion errors. In Section 4.5 we shall show that the redundancy newly added for correcting deletion or insertion errors besides additive errors can be made sufficiently small by reinforcing the capabilities of the both codes $\mathcal{C}_X$ and $\mathcal{C}_U$.

4.2 Definitions

A successive deletion-error of s bits on a vector A causes a slippage of s bits in the suffix of the received word as illustrated in Figure 4.1.

<u>Definition 4.1</u> When successive s symbols, $\alpha_{\lambda+1}$, $\alpha_{\lambda+2}$, $\cdots$, $\alpha_{\lambda+s-1}$ and $\alpha_{\lambda+s}$ are deleted in A, we refer to this event as *successive deletion-error* of s bits, or *SD(s) error* in abbreviated notation.

We shall denote the output word of the second channel, provided



d $= (\alpha_{\lambda+1}, \cdots, \alpha_{\lambda+s})$: deleted sequence

i $= (u_{\hat{\lambda}-\hat{s}}, \cdots, u_{\hat{\lambda}-1})$: inserting sequence

Figure 4.1 Received word corrupted by SD(s) error and insertion on it.

that no additive error occurs, by the vector $A(-s,\lambda)$ or polynomial $A(z|-s,\lambda)$ as follows;

$$B(z) = A(z|-s,\lambda) = \sum_{i=o}^{s-1} \alpha'_{i+N-s} z^i + \sum_{i=s}^{\lambda+s} \alpha_{i-s} z^i + \sum_{i=\lambda+s+1}^{N-1} \alpha_i z^i \tag{4.1}$$

where s is a positive integer, λ, $-1 \leq \lambda \leq N-s-1$, represents a position where the deletions begin, and $(\alpha'_{N-s}, \alpha'_{N-s+1}, \cdots, \alpha'_{N-1})$ is a prefix of the subsequently transmitted code word following A. Let $A(z|-s,\lambda;\hat{s},\hat{\lambda})$, $\hat{s}>0$ and $1 \leq \hat{\lambda} \leq N$, represent the word that is formed by inserting a sequence $(u_{\hat{\lambda}-\hat{s}}, \cdots, u_{\hat{\lambda}-1})$ of length $\hat{s}$ between the $(\hat{\lambda}-1)$st and the $\hat{\lambda}$th positions of the received word $B(z)=A(z|-s,\lambda)$ where $u_j \in GF(2)$, $\hat{\lambda}-\hat{s} \leq j \leq \hat{\lambda}-1$. Let $B(z|\hat{s},\hat{\lambda})$ be defined as follows;

$$B(z|\hat{s},\hat{\lambda}) \triangleq A(z|-s,\lambda;\hat{s},\hat{\lambda}) \tag{4.2}$$

When $\hat{s}$ and $\hat{\lambda}$ satisfy $0<\hat{s} \leq s$ and $s \leq \hat{\lambda} \leq \lambda+s+1$, $B(z|\hat{s}.\hat{\lambda})$ is given by

$$B(z|\hat{s},\hat{\lambda}) = \sum_{i=o}^{s-\hat{s}-1} \alpha'_{i+N-s+\hat{s}} z^i + \sum_{i=s-\hat{s}}^{\hat{\lambda}-\hat{s}-1} \alpha_{i-s+\hat{s}} z^i + \sum_{i=\hat{\lambda}-\hat{s}}^{\hat{\lambda}-1} u_i z^i + \sum_{i=\hat{\lambda}}^{\lambda+s} \alpha_{i-s} z^i + \sum_{i=\lambda+s+1}^{N-1} \alpha_i z^i \tag{4.3}$$

Figure 4.1 also depicts this situation.

Let us define the interleaved code $\mathcal{C}_T$ by replacing the subblocks x_t^* and x_t^{**} of the code $\mathcal{C}_{0_T}$ defined in (3.13) with the subblocks ξ_t^* and ξ_t^{**}, respectively, which are given by

$$\xi_t^* = (\xi_{tf-D}, \xi_{tf-D+1}, \cdots, \xi_{tf-1}, \xi_t, \xi_{tf+f}, \xi_{tf+f+1}, \cdots, \xi_{tf+f+D-1}); \quad 2 \leq t \leq T-1 \tag{4.4}$$

and

$$\xi_t^{**} = (\xi_{tf}^D, \xi_t, \xi_{tf+f}^D) + r; \quad t=0, 1 \tag{4.5}$$

where the vectors ξ_t, $0 \leq t \leq T-1$, are given in (3.8), the ξ_{tf}^D and ξ_{tf+f}^D are the D-dimensional all ξ_{tf}'s and all ξ_{tf+f}'s vectors, respectively, and

$$r = (0^D, 1, 0^{f-2}, 1, 0^D) \tag{4.6}$$

Thus we define the code C_T as follows;

$$C_T \triangleq \{A = (\xi_0^{**}, \mu_0, \xi_1^{**}, \mu_1, \xi_2^*, \mu_2, \cdots, \xi_{T-1}^*, \mu_{T-1})\} \tag{4.7}$$

where the length of the code C_T is given by

$$N = n_x + m_u + 2DT \tag{4.8}$$

Note that $(\xi_{tf-D}, \xi_{tf-D+1}, \cdots, \xi_{tf-1})$ and $(\xi_{tf+f}, \xi_{tf+f+1}, \cdots$ $\cdots, \xi_{tf+f+D-1})$ in (4.4) are the prefix of length D of ξ_{t-1} and the suffix of length D of ξ_{t+1}, respectively. Therefore, they are doubly transmitted for $t \geq 2$. The construction of the code C_T is illustrated in Figure 4.2.

The relationship between the lth component of $A \in C_T$ and the ith component of $X \in C_X$ is given by

$$l = i + D + (g + 2D)[i/f] \triangleq L_X(i) \tag{4.9}$$

Similarly, the relationship between the lth component of A and the jth component of $U \in C_U$ is given by

$$l = j + (f + 2D) \cdot (1 + [j/g]) \triangleq L_U(j) \tag{4.10}$$

From the received word B, we can de-interleave the following two vectors Y and V corresponding to $X \in C_X$ and $U \in C_U$, respectively;

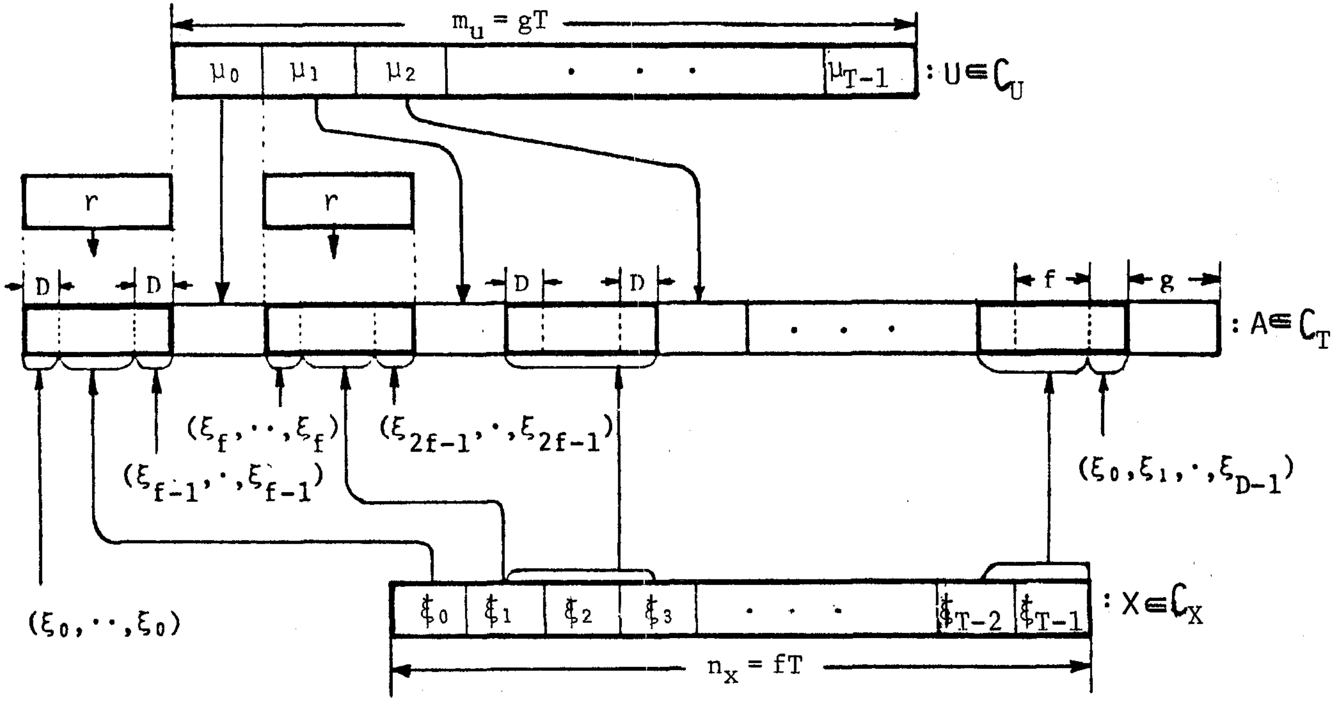


Figure 4.2 Construction of code $\mathcal{C}_T$.

$$Y = [B + R]W_X \tag{4.11}$$

and

$$V = B\,W_U \tag{4.12}$$

where W_X is the N by n_X matrix;

$$W_X = \| \delta_{l l'} \| , \quad l' = L_X(i) \; ; \quad 0 \leq i \leq n_X - 1, \; 0 \leq l \leq N-1 \tag{4.13}$$

and W_U is the N by m_u matrix;

$$W_U = \| \delta_{l l''} \| , \quad l'' = L_U(j) ; \quad 0 \leq j \leq m_u - 1 \tag{4.14}$$

In (4.13) and (4.14), δ_{ij} is the Kronecker delta and R is given by

$$R = (r, 0^g, r, 0^g, 0^{(f+g+2D)\cdot(T-2)}) \tag{4.15}$$

The decoding algorithm of the code C_T is the same that of Section 3.4.2, precisely given in APPENDIX I, except that the decoder inserts a sequence of the predetermind pattern of length $\hat{s}$ at each inserting position. The complete version of the decoding algorithm is given in APPENDIX II.

4.3 Capabilities of The Code C_T

4.3.1 Correction of Either An SD error or An Additive Burst Error

The next lemma assures that the code C_T can distinguish additive errors from deletion errors.

<u>Lemma 4.1</u> If $B = A(-s,\lambda)$ for s, $0 < s \leq D$, and λ, $f+g+3D \leq \lambda + s \leq N-1$, the length of the pseudo burst error of $Y = [B+R]W_X$ is given by

$$f + 1 \leq \rho \leq \left\lceil \frac{T+2}{2} \right\rceil f \tag{4.16}$$

In the proof, for convenience' sake we shall use a transmitted code word A instead of a received word B having an SD error which occurred in ξ_1 or higher subblocks than ξ_1^{**}.

(proof) Consider a received word $Y=[A(-s,\lambda)+R]W_X$, $0<s\leqq D$, and $\lambda=L_X(\zeta)$, whose components $\xi_{\zeta+1}, \xi_{\zeta+2}, \cdots, \xi_{\zeta+s}$ of X are deleted and to which R is added erroneously. This word can be represented by

$$Y(z) = \sum_{i=o}^{\zeta+s} \xi_{i-s} z^i + \sum_{i=\zeta+s+1}^{n_X-1} \xi_i z^i + \sum_{i=o}^{s-1} (\xi_0 + \xi_{i-s}) z^i$$

$$+ \sum_{i=f}^{f+s-1} (\xi_f + \xi_{i-s}) z^i + (1 + z^s + z^{f-1})\cdot(1 + z^f); \quad 0 < s \leqq D \tag{4.17}$$

As shown in Figure 4.3, the pseudo burst error can be classified into two types of error depending upon the position where the deletion error starts, i.e.,

$$E(z) = X(z) + Y(z) = (1 + z^f) + \sum_{i=1,\neq f}^{2f-1} q_i z^i + \sum_{i=2f}^{\zeta+s} (\xi_i + \xi_{i-s}) z^i \tag{4.18a}$$

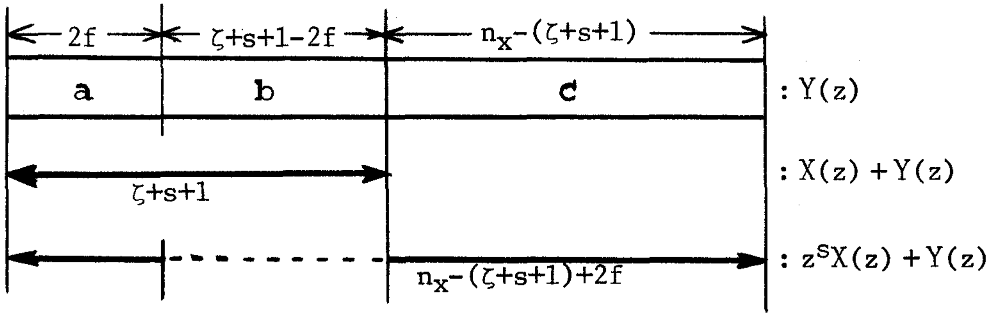
or

$$E(z) = z^s X(z) + Y(z) = (z^s + z^{f-1})\cdot(1 + z^f) + \sum_{\substack{i=o \\ i\neq s,f-1,f+s}}^{2f-2} q_i z^i$$

$$+ \sum_{i=\zeta+s+1}^{n_X-1} (\xi_i + \xi_{i-s}) z^i \tag{4.18b}$$

where q_i=0 or 1 for any i.

Equations (4.18a) and (4.18b) show that there always exists in Y



a: Region where a burst error of length at least f+1 definitely occurs as a part of pseudo burst error.

$\mathbf{b} = (\xi_{2f-s}, \xi_{2f-s+1}, \ldots, \xi_{\zeta})$

$\mathbf{c} = (\xi_{\zeta+s+1}, \xi_{\zeta+s+2}, \ldots, \xi_{n_x-1})$

Figure 4.3 Pseudo burst error due to SD(s) error of $\lambda=L_X(\zeta)$.

a burst error whose length is at least f+1 and at most ζ+s+1 or $n_x-(\zeta+s+1)+2f$ (See also Figure 4.3.). From Definition 3.1, we see that the length of the pseudo burst error for any pair of ζ and s is upper-bounded by

$$\rho \leqq \min\{\zeta+s+1,\ n_x-(\zeta+s+1)+2f\} \tag{4.19}$$

The maximum ρ with respect to ζ and s is given by equating the two terms in the braces of (4.19) as follows;

$$\max_{\zeta,s} \rho = \left[\frac{n_x+2f}{2}\right] = \left[\frac{T+2}{2}\right] f \tag{4.20}$$

Q.E.D.

We shall give the following lemma on cyclic code before giving Lemma 4.3.

<u>Lemma 4.2</u> Assume that $H(z)=(1+z^n)/G(z)$ can be factored into $H(z)=\prod_{i=1}^{j} h_i(z)$, $1\leqq j\leqq k$, where each factor $h_i(z)$ is an irreducible polynomial of degree k or less over GF(2) whose roots in an extension field have order e_i. Then for any $X(z)\in \mathcal{C}$ and s, $0<s\leqq D$

$$X(z)+z^sX(z)\neq 0 \quad (\mathrm{mod}\ 1+z^n) \tag{4.21}$$

where

$$D<\min\{f,\ e_1,\cdots,\ e_j\} \tag{4.22}$$

(proof) Suppose that $X(z)+z^sX(z)=0$ (mod $1+z^n$). Letting X(z) be represented by P(z)G(z) where P(z) is some polynomial of degree at most k-1, we have the following relation for some polynomial Q(z)

$$(1+z^s)P(z)G(z) = (1+z^n)Q(z) \tag{4.23}$$

Dividing both sides of (4.23) by G(z), we have

$$(1+z^s)P(z) = H(z)Q(z) \tag{4.24}$$

Since $H(z)$ is the polynomial of degree k, it cannot divide $P(z)$ whose degree is at most k-1. On the other hand, there is no common factor between $H(z)$ and $(1+z^s)$, $0<s\leqq D$, by the assumption. Therefore, $H(z)$ cannot divide $(1+z^s)P(z)$. This is a contradiction, completing the proof. Q.E.D.

<u>Lemma 4.3</u> In any code word of $\mathcal{C}_X$ under the same condition of Lemma 4.2, if any consecutive components of length $f(\geqq k_X)$ starting from the ζth position, $0\leqq\zeta\leqq n_X-f-1$, is substituted by the corresponding components starting from the ζth position of the cyclically shifted version of the code word by s bits, then there exists at least a burst error of length 1 in the code word where $0<s\leqq D$

(proof) The word mentioned above can be expressed as follows;

$$F(z) = X(z) + \sum_{i=\zeta}^{\zeta+f-1} (\xi_i + \xi_{i-s}) z^i; \quad 0 \leqq \zeta \leqq n_X - k_X - 1 \tag{4.25}$$

where $X(z)$ is a code word of $\mathcal{C}_X$. We then have the error $E(z)$ as follows;

$$E(z) = F(z) + X(z) = \sum_{i=\zeta}^{\zeta+f-1} (\xi_i + \xi_{i-s}) z^i \tag{4.26}$$

Thus the error $E(z)$ is proved to be the consecutive components of length f existing in another code word $X(z)+z^sX(z) \in \mathcal{C}_X$ from Lemma 4.2. Since the code word $X(z)+z^sX(z)$ has a consecutive run of at most k_X-1 0's from Property 3.1, there exists at least a burst error of length 1 in $X(z)$, completing the proof. Q.E.D.

Lemma 4.4 Consider an insertion of length s at the $\hat{\lambda}(t)$th position of the received word $B=A(-s,\lambda)$ where the positional index $\hat{\lambda}(t)$ is given by

$$\hat{\lambda}(t) = t(f+g+2D)+D \tag{4.27}$$

Using the function $\hat{\lambda}(t)$, the location of the SD(s) error is given by

$$\hat{\lambda}(t_0) \leqq \lambda + s \leqq \hat{\lambda}(t_0+1) - 1; \quad 2 \leqq t_0 \leqq T \tag{4.28}$$

Then, for all $\hat{s}$, $\hat{s} \neq s$ and $0<\hat{s} \leqq D$, and $2 \leqq t \leqq t_0$, there exists a pseudo burst error of length at least f+1 and at most n_X-2f, provided that $T \geqq 8$.

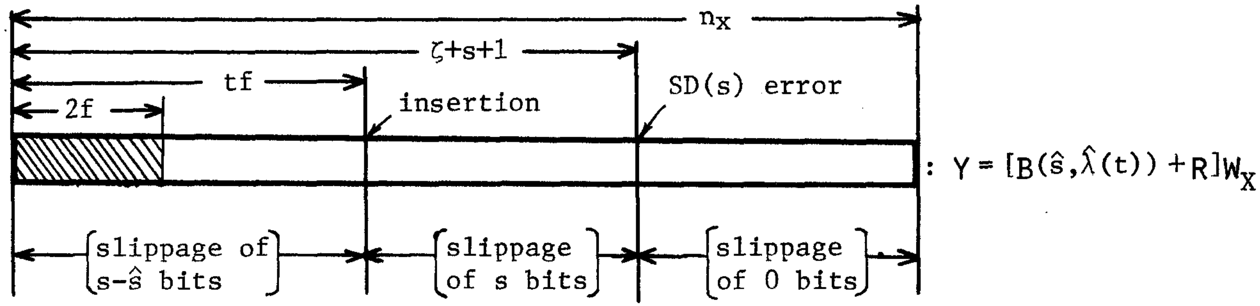
(proof) If $s-\hat{s}>0$, there exists a pseudo burst error of length at least f+1 in $\mathfrak{t}_0$ and $\mathfrak{t}_1$ from Lemma 4.1. Even if $s-\hat{s}<0$, we can show that there exists a pseudo burst error of length at least f+1 in $\mathfrak{t}_0$ and $\mathfrak{t}_1$ as shown in the proof of Lemma 4.1. In these cases, the pseudo burst error assumes one of the three types of burst error corresponding to $X(z)$, $z^s X(z)$ and $z^{s-\hat{s}} X(z)$ (See Figure 4.4.). The length ρ is then given by

$$\rho \leqq \min\{\zeta+s+1,\ n_X-(t-2)f,\ n_X-(\zeta+s+1-tf)\} \tag{4.29}$$

where $\zeta+1$ denotes the starting position of the SD(s) error. The maximization with respect to ζ, s and t is given by equating the three terms in the braces of (4.29) as follows;

$$\max_{\zeta,s,t} \rho = \left[\frac{2T+2}{3}\right] f \leqq (T-2)f \tag{4.30}$$

This relation implies that $T \geqq 8$. Q.E.D.



: region with burst error of length at least f+1.

Figure 4.4 Pseudo burst error by insertion of $\hat{s} \neq s$ at $\hat{\lambda}(t)$.

Now we have the following theorem.

<u>Theorem 4.1</u> The code $\mathcal{C}_X$ can correct either an SD(s) error, $s \leqq D$, or an additive burst error of length $b=f+g+2D$ or less, provided that the relations (4.22) and $T \geqq 8$ hold.

In the proof of Theorem 4.1, we shall need the following decoding rules.

<u>Rule 1</u> Each insertion of lengths $\hat{s}$, $\hat{s}=1,2,\cdots,D$ starts at $\hat{\lambda}(1)$ and the insertions are made at $\hat{\lambda}(t)$ where t increases by 1 up to T-1.

<u>Rule 2</u> Burst-error correction of length f is made only in $\mathfrak{k}_{t-1}$ and $\mathfrak{k}_t$.

(proof) *-detectability of SD error-* The length of an additive burst error in Y is limited to be at most f by the assumption. On the other hand, from Lemma 4.1 the length of a pseudo burst error caused by an SD error is at least f+1 for λ, $f+g+3D \leqq \lambda+s \leqq N-1$. This implies that the code $\mathcal{C}_X$ can detect an SD error in Y or B, provided that the following relation holds;

$$\left[\frac{T+2}{2}\right] f \leqq \frac{T+2}{2} f \leqq fT - 2f \leqq fT - k_X - f \tag{4.31}$$

i.e., $T \geqq 6$. Note that for an SD error at λ, $\lambda+s \leqq f+g+3D-1$, we can correct it simply as an additive burst error.

-correctability of SD error- For an SD(s) error that satisfies the relation (4.28), we fail to decode $Y=[B(\hat{s},\hat{\lambda}(t))+R]W_X$ by any insertion $\hat{s} \neq s$ and $t \leqq t_0$ from Lemma 4.4. For an insertion of $\hat{s}=s$ and $t \leqq t_0$, the slipped subblocks $\mathfrak{k}_0$, $\mathfrak{k}_1, \cdots, \mathfrak{k}_{t-2}$ and $\mathfrak{k}_{t-1}$ are

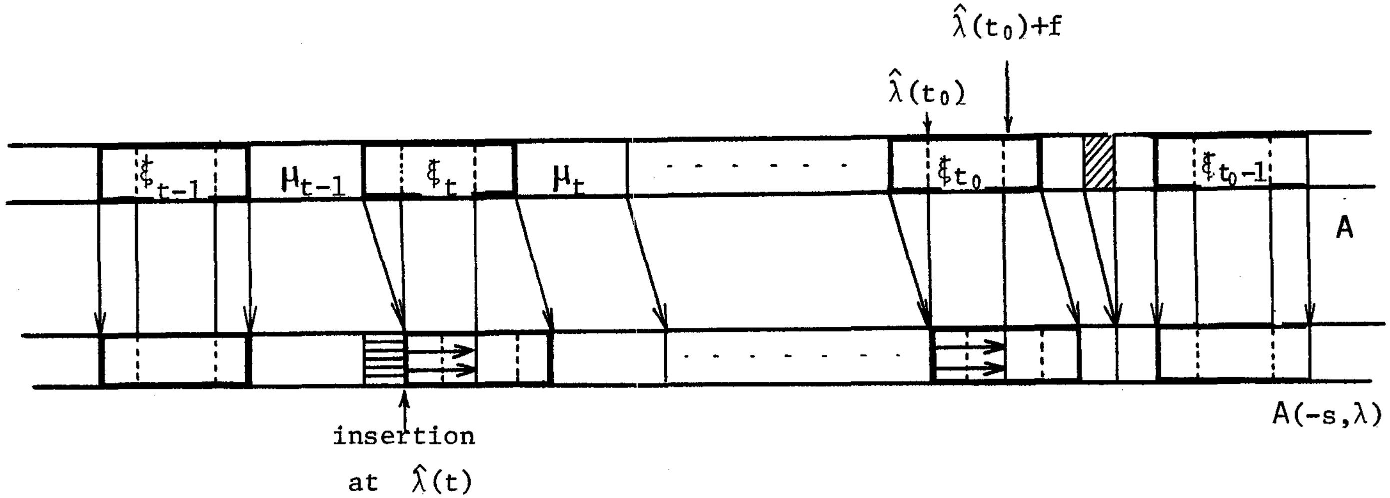
restored to their original positions. Therefore, the remaining slippage is in the subblocks ξ_t, $\xi_{t+1}, \cdots, \xi_{t_0-1}$ and with a part of ξ_{t_0} in case of $\hat{\lambda}(t_0) \leqq \lambda+s \leqq \hat{\lambda}(t_0)+f-2$, or with a whole of ξ_{t_0} in case of $\hat{\lambda}(t_0)+f-1 \leqq \lambda+s \leqq \hat{\lambda}(t_0+1)-1$ (See Figure 4.5.). Since each slipped subblock of length f has at least a single error from Lemma 4.3, using Rule 2, we can decode Y successfully by the insertion at $t=t_0-1$ when there is no error in ξ_{t_0}, or at $t=t_0$ when there are errors in ξ_{t_0} in case of $\hat{\lambda}(t_0) \leqq \lambda+s \leqq \hat{\lambda}(t_0)+f-2$. In case of $\hat{\lambda}(t_0)+f-1 \leqq \lambda+s \leqq \hat{\lambda}(t_0+1)-1$, we can decode Y successfully by the insertion at $t=t_0$, since all the components of ξ_{t_0} are slipped.

Also for an SD(s) error of $\hat{\lambda}(1) \leqq \lambda+s \leqq \hat{\lambda}(2)-1$, we fail to decode Y by any insertion of $\hat{s}<s$ from (4.18a). We can decode Y successfully by the insertion of $\hat{s}=s$ at $\hat{\lambda}(1)$ since the remaining slippage is only in ξ_1.

In decoding of $V=B(\hat{s}=s, \hat{\lambda}(t=t_0-1 \text{ or } t_0))W_U$, it is sufficient to correct a pseudo burst error of length g or less in μ_t. Q.E.D.

4.3.2 Correction of An SDB Error

<u>Definition 4.2</u> When an additive burst error of length b or less occurs in the first channel and a successive deletion error of s bits occurs within that burst error in the second channel, we refer to such errors as *successive deletion error of s bits imbedded within the burst error of length b or less*, or as *SDB(s,b) error* in abbreviated notation.



: SD(s) error of $\hat{\lambda}(t_0)+f-1 \leqq \lambda + s \leqq \hat{\lambda}(t_0+1) - 1$.

: Inserted region of length $\hat{s}$.

: Slipped region of Y by s bits.

Figure 4.5 Insertion of $\hat{s}=s$ at $\hat{\lambda}(t)$.

Consequently, the additive burst error of length b or less is reduced to that of length b-s or less due to successive deletion error of s bits.

We show that the code C_T is capable of not only correcting either an SD error or an additive burst error but also correcting an SDB error.

Lemma 4.5 For any SDB(s,b) error, $s \leqq D$ and $b=f+g+2D$, occurred in the higher positions than $f+g+3D-1$, the length of pseudo burst error in $Y=[B+R]W_X$ is given by

$$f+1 \leqq \rho \leqq \frac{T+3}{2} f \tag{4.32}$$

(proof) Consider an additive burst error of length $b=f+g+2D$ in B, which is reduced to a burst error of length at most f in a subblock ξ_{t_0} or in two adjacent subblocks ξ_{t_0} and ξ_{t_0+1} depending upon the starting position of the additive burst error of length b. An SD(s) error occurred within the additive burst error of length b, i.e., an SDB(s,b) error, causes the slippage of s bits in the suffix followed by the additive burst error in $Y=[B+R]W_X$.

From the assumption that the SDB(s,b) error occurs in the higher positions than $f+g+3D-1$, there always exists a pseudo burst error of length at least $f+1$ in the suffix of Y as shown in the proof of Lemma 4.1.

Letting ζ denote the starting position of the additive burst error in Y (See Figure 4.6.), we can obtain the upper-bound on length ρ as follows;

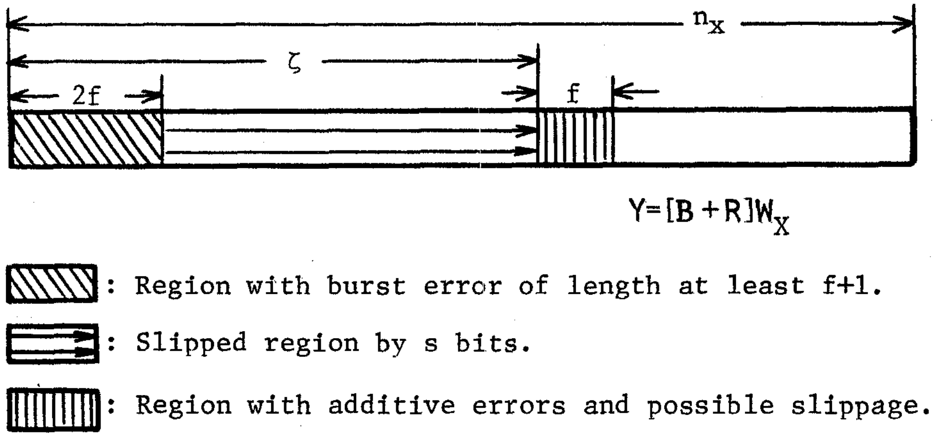


Figure 4.6 Pseudo burst error of Y caused by SDB(s,b) error, b=f+g+2D.

$$\max_{\zeta} \rho \leqq \max_{\zeta} \{ \min(\zeta + f,\ n_X - (\zeta - 2f)) \} \leqq \frac{n_X + 3f}{2} \qquad (4.33)$$

Q.E.D.

In decoding of B corrupted by an SDB error, we shall need the following decoding rule in addition to Rule 1 and Rule 2.

Rule 3 After decoding of Y by the insertion at $\hat{\lambda}(t)$, burst-error correction of length 3g is made in μ_{t-1}, μ_t and μ_{t+1}. When a burst-error correction in ξ_0 with no insertion is successful, then decide that a burst error has occurred in μ_0 and μ_1.

In the following, as in Chapter 3, we denote the code C_T by C_{TB} if the code C_U is capable of detecting a single burst error of length 3g or less.

Theorem 4.2 The code C_{TB} can correct an SDB(s,b) error, $s \leqq D$ and $b=f+g+2D$, provided that the relations (4.22) and $T \geqq 9$ hold.

(proof) *-Detectability of deletion errors-* From Lemma 4.5, we see that the code C_{TB} can detect an SDB(s,b) error, provided that the following relation holds;

$$\frac{n_X + 3f}{2} \leqq n_X - 2f \leqq n_X - k_X - f \qquad (4.34)$$

i.e., $T \geqq 7$.

-Correctability of SDB error- By combining the proofs of Lemmas 4.4 and 4.5, the upper-bound on length ρ is given by

$$\rho \leqq \min_{\zeta, t} \{ \zeta + f,\ n_X - (t-2)f,\ n_X - (\zeta - tf) \} \leqq \frac{2n_X + 3f}{3} \qquad (4.35)$$

for any insertion of $\hat{s}$, $0 < \hat{s} \leqq D$, and t, $\hat{\lambda}(t) \leqq \zeta$ where ζ denotes the starting position of the additive burst error as shown in Figure 4.6.

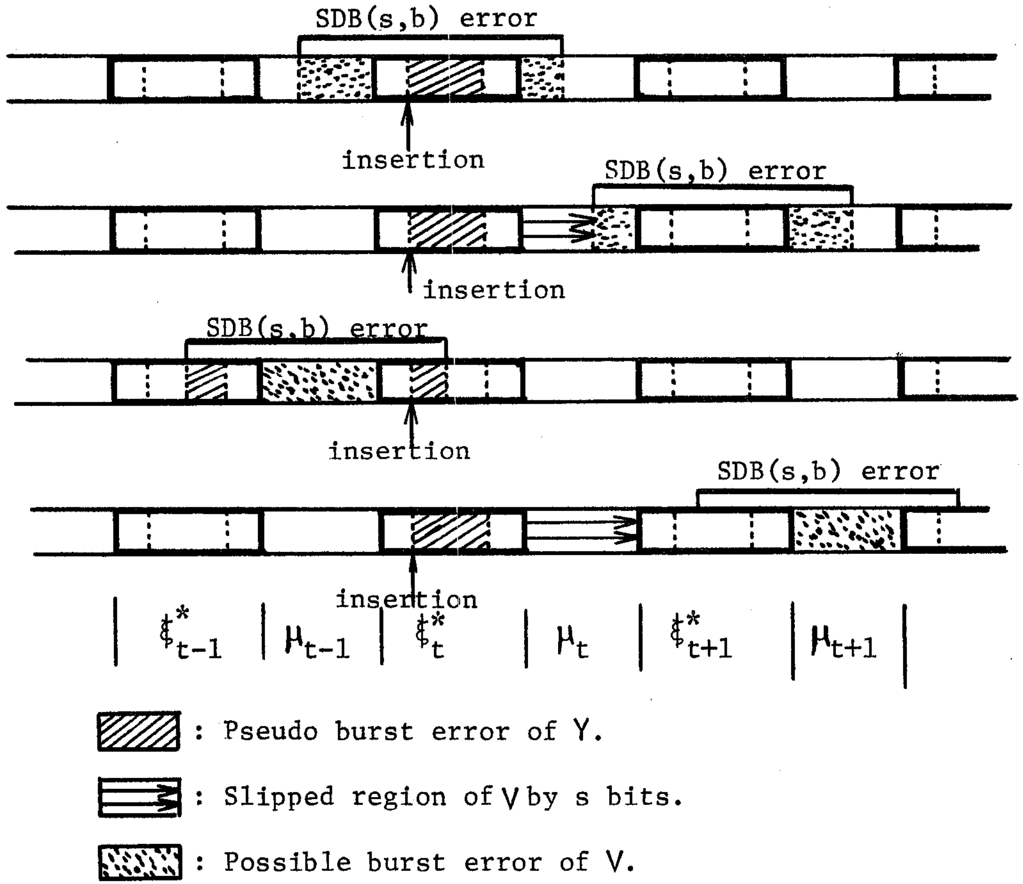


Figure 4.7 Final insertions of $\hat{s}$=s at $\hat{\lambda}$(t) for four types of SDB(s,b) error, b=f+g+2D.

From (4.35) the length ρ is upper-bounded by (T-2)f, provided that $T \geqq 9$.

After decoding of Y by an insertion of $\hat{s}=s$ at $\hat{\lambda}(t)$, there are exactly four cases of the remaining burst error in $V=B(s,\hat{\lambda}(t))W_U$ as shown in Figure 4.7. For the SDB error including $\xi^*_{t_0}$, there may be no error in ξ_{t_0} by the slippage and/or additive errors in spite of possible errors existing in μ_{t_0-1} and μ_{t_0}. Thus there may exist a burst error of length at most 2g in the three subblocks μ_{t-1}, μ_t and μ_{t+1} after decoding of Y by the insertion at $\hat{\lambda}(t)$. From Property 3.3 and Rule 3, we can decode V successfully, completing the proof. Q.E.D.

4.3.3 Correction of Either An SDB Error or SIB Error

The code C_T can also correct a successive insertion error of s bits (or an *SI(s) error* in abbreviated notation). This can be shown by simply translating deletions into insertions in the previous sections. Note that a deletion of $\hat{s}$ at $\hat{\lambda}(t)$ for an SI(s) error is made by deleting $\beta_{\hat{\lambda}(t)-\hat{s}}$, $\beta_{\hat{\lambda}(t)-\hat{s}+1}, \cdots, \beta_{\hat{\lambda}(t)-2}$ and $\beta_{\hat{\lambda}(t)-1}$ of $B=A(+s,\lambda)$. But in this case an inserted sequence on the channel may appear in Y as a burst error. Especially an SI(s) error imbedded within an additive burst error of length b=f+g+2D or less (or an *SIB(s,b) error* in abbreviated notation) incurs an additive burst error of length f+s in Y. For a received word B corrupted by an SIB(s,b) error, $s \leqq D$ and b=f+g+2D, the length of pseudo burst error in $Y=[B+R]W_X$ is given by

$$f+1 \leqq \rho \leqq \frac{n_x+3f+s}{2} \tag{4.36}$$

Also by any deletion of $\hat{s}$ at $\hat{\lambda}(t)$, the length of pseudo burst error in $Y=[B(-\hat{s},\hat{\lambda}(t))+R]W_X$ is given by

$$f+1 \leqq \rho \leqq \frac{2n_x+3f+s}{3} \tag{4.37}$$

We see that from (4.36) the relation $T \geqq 8$ assures that the code C_{TB} can detect the SIB error and from (4.37) the relation $T \geqq 10$ assures that the code C_{TB} can correct the SIB error. Thus we have the following theorem corresponding to Theorem 4.2.

Theorem 4.3 The code C_{TB} can correct an SIB(s,b) error, $s \leqq D$ and b=f+g+2D, provided that the relations (4.22) and $T \geqq 10$ hold.

Note that Theorems 4.1, 4.2 and 4.3 hold true only if the types of timing errors are limited to either deletion or insertion errors. However, when either deletion or insertion errors of amount less than or equal to D occur in a received word, the code C_{TB} can detect such errors. Thus it is easy to see that the upper-bound on the resultant amount of timing errors during the decoding process is given by D, if the amounts of deletion and insertion errors are limited to [D/2] or less, respectively. We then have the following theorem.

Theorem 4.4 The code C_{TB} can correct either ab SDB(s,b) error or an SIB(s,b) error, $s \leqq [D/2]$, b=f+g+2D, in a received word, provided that the relation (4.22) and $T \geqq 10$ hold.

4.4 Efficiency and Example

We use a maximal length code as the code $\mathcal{C}_X$ which satisfies the following relation;

$$n_X = 2^{k_X} - 1 = fT \quad \text{for } f \geqq k_X \text{ and } T \geqq 8 \tag{4.38}$$

In this case the polynomial $H(z)=(1+z^{n_X})/G_X(z)$ is the primitive polynomial of degree k_X, i.e., $j=1$ and $e_1=2^{k_X}-1=n_X$ in (4.22), which implies that $f<e_1=n_X$. Therefore, we can always choose $D=f-1$ for any maximal length code that satisfies (4.38).

We use an ideal burst-error correcting cyclic code in the sense that it meets the Reiger bound[15, p110] in equality, i.e., $m_u-k_u=2g$ as the (m_u,k_u) code $\mathcal{C}_U$. The coding rate of the code $\mathcal{C}_T$ is given by

$$R = (k_X + k_u)/N = \{k_X + g(T-2)\}/(k_X + g + 2D + 1)T \tag{4.39}$$

We can easily see that the following relation holds;

$$\lim_{g\to\infty} R = (T-2)/T \tag{4.40}$$

If the code $\mathcal{C}_T$ is used for a channel with SDB or SIB error, then the code $\mathcal{C}_U$ has the possibility of false correction because after a successful decoding of $\mathbf{Y}$ there may exist a burst error of length 2g or less in the three consecutive subblocks of $\mathbf{V}$ as stated in the proof of Theorem 4.2. This event occurs if no error exists in consecutive f components of $\mathbf{X}$ within an SDB or an SIB error. Thus the probability of false correction of the code $\mathcal{C}_U$ is given by 2^{-f} which is sufficiently small when $f \gg 1$ holds.

On the other hand, the code C_{TB} where the code C_U has 3g check symbols has no possibility of false correction and the coding rate in the limit of $g \to \infty$ is given by (T-3)/T.

Example

If we choose the shortest maximal length code as the code C_X which satisfies (4.38) and $T \geq 10$, there is the (255, 8) code whose generator polynomial is given by $G_X(z)=(1+z^{255})/H(z)$, $H(z)=1+z^2+z^3+z^4+z^8$. In this case we have f=17, T=15 and $D \leq 16$. If we use a shortened Fire code as the code C_U and if we let g=2f and the number of its check symbols be 3g, we can choose the generator polynomial as $G_U(z)=(1+z+z^2+z^{27}+z^{34})(1+z^{68})$. The length of the code C_U is then 510, although the natural length of the Fire code generated by $G_U(z)$ is about 1.2×10^{12}. We then have the code C_{TB} whose parameters are N=825, b=55 and $R \simeq 0.5$ for D=2, or N=1245, b=83, and $R \simeq 0.33$ for D=16.

4.5 Conclusion

We have constructed a new class of codes capable of correcting either deletion or insertion errors together with an additive burst error in a received word. One of our results shows that a maximal length code as the code C_X can serve not only to maintain bit-synchronization but also to transmit information, in a sharp contrast with a usual comma code which is used only as a comma in order to maintain word-synchronization. Furthermore, we have shown that the redundancy newly added for combatting with timing errors can be made sufficiently small by reinforcing the capabilities of the codes $\mathcal{C}_X$ and $\mathcal{C}_U$.

CHAPTER 5

CONCLUSION

We have constructed a new class of codes capable of correcting timing errors besides additive errors. In this chapter we shall summarize the principal results achieved in each chapter.

In Chapter 1, the motivation of this research was given.

In Chapter 2, we have derived the coset codes which are capable of correcting both a synchronization slippage and multiple additive burst errors concurrently occurred in a received word, and have derived the lower bounds on the redundancy of the coset codes.

In Chapter 3, using the interleaved codes, we have constructed a new class of codes having the property of generating a pseudo burst error by which the decoder can detect the occurrence of a deletion error.

In Chapter 4, we have improved the interleaved code derived in Chapter 3 so that it is capable of correcting an arbitrary number of successive deletion errors besides an additive burst error. We then have shown that the interleaved code has also the capability of correcting successive deletion or insertion errors imbedded within an additive burst error. The efficiency of the interleaved code has been given, which shows that the redundancy newly added for combatting with the timing errors can be made sufficiently small by reinforcing the capabilities of the codes $\mathcal{C}_X$ and $\mathcal{C}_U$ which constitute the interleaved code.

For the future communication network system using independent clocking technique, the coding theoretical approach to compensate the deviation in timing between each pair of the nodes and etc. should be more and more explored. Berore completing this dissertation, we sincerely wish the present work will be contributed for realizing such future communication network.

APPENDIX I

DECODING ALGORITHM OF THE CODE $\mathcal{C}_{0_T}$

Step I-1: De-interleave Y from B by $Y=[B+R_0]W_{0_X}$ and try to correct an additive burst error of length f or less. If no error is detected, or any burst-error correction is successful, go to Step I-9. Otherwise, go to Step I-2.

Step I-2: Assume that a deletion error has occurred and proceed to the deletion- and additive burst-error correction procedure of Step I-3 through Step I-8.

Step I-3: Set t=1.

Step I-4: Obtain $Y=[B_{\hat{\lambda}_t}+R_0]W_{0_X}$, $\hat{\lambda}_t=t(f+g+1)+1$, by inserting $u_{\hat{\lambda}_t-1}$ between the $(\hat{\lambda}_t-1)$st and the $\hat{\lambda}_t$th position of B, and then detect errors. If no error is detected, go to Step I-9. Otherwise, go to Step I-5.

Step I-5: If t=T-1, go to Step I-6. Otherwise, increase t by 1 and return to Step I-4.

Step I-6: Set t=1.

Step I-7: Obtain $Y=[B_{\hat{\lambda}_t}+R_0]W_{0_X}$ and then try to correct an additive burst error of length f or less in ξ_{t-1} and ξ_t. If a burst-error correction is successful, go to Step I-9. Otherwise, go to Step I-8.

Step I-8: Increase t by 1 and return to Step I-7.

Step I-9: When the transition from Step I-1 to this step is made, de-interleave V from B by $V=BW_{0_U}$ and decode V in an ordinary way. If a burst-error correction in ξ_0 is made, then assume that a burst error in μ_0 and μ_1 has occurred. When the transition from the deletion and additive burst-error correction procedure of Step I-2 through Step I-8 to this step is made, de-interleave V from $B_{\hat{\lambda}_t}$. Finally decode by trying to correct a a pseudo burst error of length 3g or less in μ_{t-1}, μ_t and μ_{t+1}.

Figure I.1 shows the flow-chart of the above decoding algorithm.

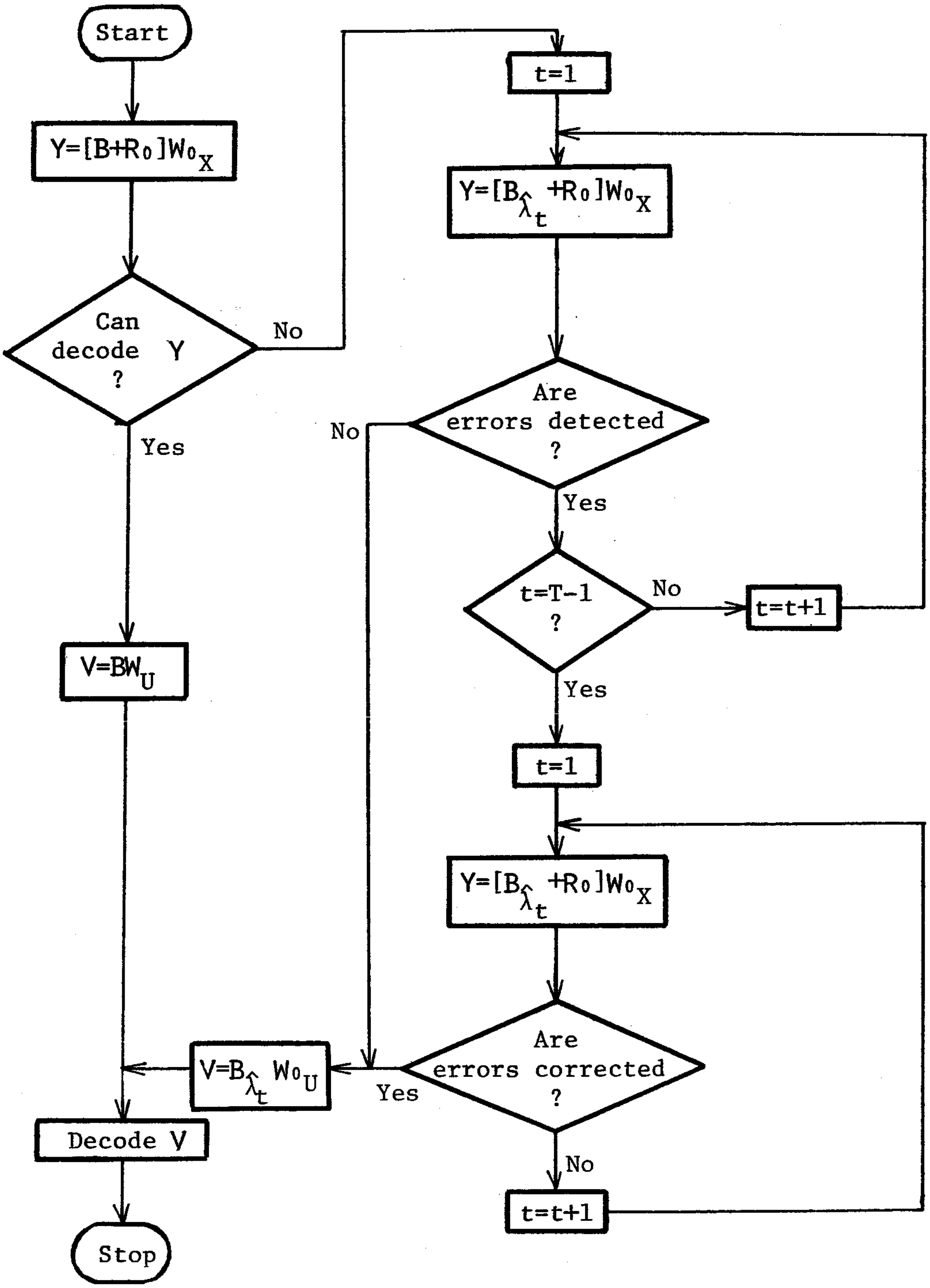


Figure I.1 Flow chart of decoding algorithm for code Co_T.

APPENDIX II

DECODING ALGORITHM OF THE CODE C_T

Step II-1: De-interleave Y from B by $Y=[B+R]W_X$ and try to correct an additive burst error. If no error is detected, or any burst-error correction of length f or less is successful, go to Step II-6. Otherwise, go to Step II-2.

Step II-2: Assume that timing errors have occurred and proceed to both the deletion- and the insertion-error correction procedures of Step II-3 through Step II-5.

Step II-3: Set $\hat{s}=1$ and $t=1$.

Step II-4: Obtain $Y=[B(+\hat{s},\hat{\lambda}(t))+R]W_X$, $\hat{\lambda}(t)=t(f+g+2D)+D$, by inserting a sequence of length $\hat{s}$ between the $(\hat{\lambda}(t)-1)$st and the $\hat{\lambda}(t)$th position of B, and then try to correct a burst error of length f or less in ξ_{t-1} and ξ_t. If a burst-error correction is successful, go to Step II-6. Otherwise, go to Step II-5.

At the same time, obtain $Y=[B(-\hat{s},\hat{\lambda}(t))+R]W_X$ by deleting the $(\hat{\lambda}(t)-\hat{s})$th through the $(\hat{\lambda}(t)-1)$st components of B, and then try to correct a burst error of length f or less in ξ_{t-1} and ξ_t. If a burst-error correction is successful, go to Step II-6. Otherwise, go to Step II-5.

Step II-5: If $\hat{s}=[D/2]$, increase t by 1, set $\hat{s}=1$ and return to Step II-4. Otherwise, increase $\hat{s}$ by 1 and return to Step II-4.

Step II-6: When the transition from Step II-1 to this step is made, de-interleave V from B by $V=BW_U$ and decode V in an ordinary way. If a burst-error correction in ξ_0 is made, then assume that a burst error in μ_0 and μ_1 has occurred.

When the transition to this step from the timing-error correction procedures of Step II-2 through Step II-5 is made, de-interleave V from $B(+\hat{s},\hat{\lambda}(t))$ or $B(-\hat{s},\hat{\lambda}(t))$ depending upon the transition from Step II-4. Finally decode V by trying to correct a pseudo burst error of length 3g or less in μ_{t-1}, μ_t and μ_{t+1}.

Figure II.1 shows the flow-chart of the above decoding algorithm.

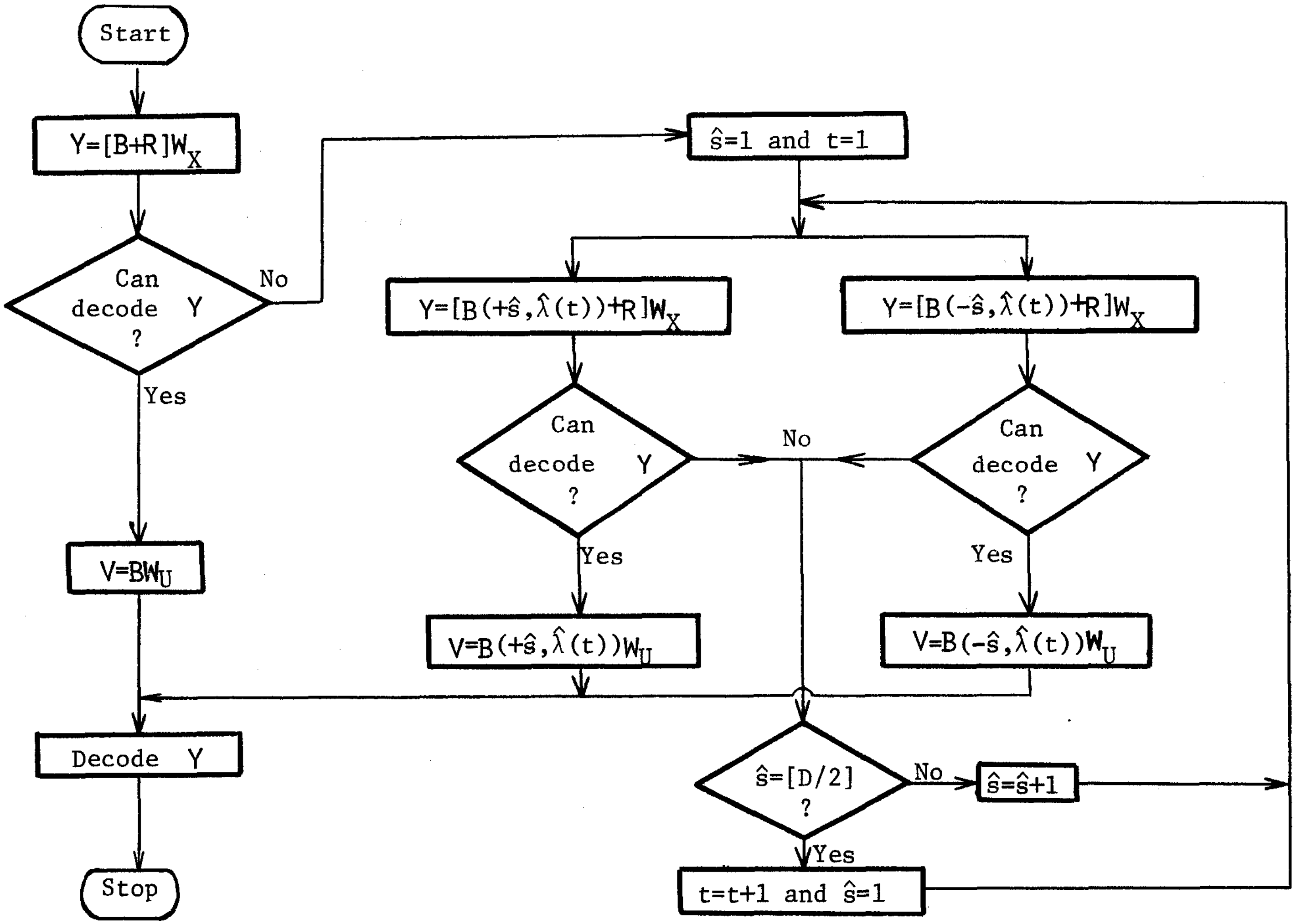


Figure II.1 Flow chart of decoding algorithm for code C_T

BIBLIOGRAPHY

[1] E.P.Greene,"Burst-trapping decoding for correction of both additive noise and deletion errors within cyclic block codes,"IEEE Trans.Inform. Theory, vol.IT-23, No.5, pp.618 -620, Sept. 1977.

[2] E.A.Harrington,"Synchronization techniques for various switching network topologies,"IEEE Trans.Communications, vol.COM-26. No.6, pp.925-932, June 1978.

[3] I.Iizuka, M.Sato, M.Kasahara and T.Namekawa,"Fundamental studies on coset codes using cyclic codes,"Trans. IECE Japan, vol.J60-A, No.7, pp.599-606, July 1977 (in Japanese).

[4] I.Iizuka, M.Sato, M.Kasahara and T.Namekawa,"Fundamental studies on coset codes using cyclic codes," rep. of professional research groups on communication systems, IECE, CS76-96, Aug. 1976 (in Japanese).

[5] J.J.Stiffler,"Comma-free error-correcting codes," IEEE, Trans. Inform. Theory, vol.IT-11, No.1, pp.107-112, Jan. 1965.

[6] S.Y.Tong,"Correction of synchronization errors with burst-error-correcting cyclic codes," IEEE Trans. Inform. Theory, vol.IT-15, No.1, pp.106-109, Jan. 1969.

[7] G.R.Redinbo and P.A.Wintz,"Codes for protection from synchronization loss and additive errors," Purdue Univ., Tech. Rep., Nov. 1969.

[8] W.W.Peterson and E.J.Weldon, *Error-correcting codes*, 2nd ed. Cambridge, Mass. :M.I.T. Press 1972.

[9] G.Benelli, C.Bianciardi and V.Cappellini,"Redundancy bounds for multiple-burst error-correcting codes," Elec. Letters, vol.13, No.13, 23rd June 1977.

[10] I.Iizuka, M.Kasahara and T.Namekawa,"A construction of block codes capable of correcting deletion and burst errors," Trans. IECE Japan, vol.J61-A, No.10, pp.1042-1049, Oct. 1978 (in Japanese).

[11] F.F.Sellers,"Bit loss and gain correcting codes," IRE Trans. Inform. Theory, vol.IT-8, pp.35-38, Jan. 1962.

[12] J.D.Ullman,"Near-optimal, single-synchronization error-correcting code," IEEE Trans. Inform. Theory, vol.IT-12, pp.418-424, Oct. 1966.

[13] E.Tanaka and T.Kasai,"Synchronization and substitution error-correcting codes for the Levenshtein metric," IEEE Trans. Inform. Theory, vol.IT-22, pp.156-162, March 1976.

[14] I.Iizuka, M.Kasahara and T.Namekawa,"Block codes capable of correcting both additive and timing errors," IEEE Trans. Inform. Theory (to be published).

[15] E.R.Berlekamp, *Algebraic coding theory*, McGraw-Hill, 1968.

[16] J.J.Stiffler, *Theory of synchronous communications*, Prentice-Hall, 1971.