SYNCHRONIZATION RECOVERY IN BINARY CYCLIC CODES

by

Paul Moreau

Submitted to the Department of Electrical Engineering in partial fulfillment of the requirements for the degree

of

Master of Applied Science

Department of Electrical Engineering
Faculty of Sciences and Engineering
University of Ottawa
Ottawa, Ontario

July, 1972

ABSTRACT

In the past years there has been a growing interest in the field of synchronization of binary codes and in particular binary cyclic codes. This was initiated as a result of an increase in demand for large data telecommunication and space satellite communication networks. This thesis critically reviews the different existing synchronization techniques now available. It describes the encoding and decoding procedures of each technique while discussing their respective advantages and disadvantages. The primary aim of this thesis is to provide a guide to the selection of the most suitable synchronization technique for a given system.

ACKNOWLEDGEMENT

The author would like to thank Professor Shiva for his useful comments and suggestions.

TABLE OF CONTENTS

INTRODUCTION

The basic concern of a communication system is the efficient and reliable transmission of data from one point to another. This demand has been accelerated during the past few years by the increasing use of digital data processors, by the extensive use of telephone and satellite communication networks, i.e. by the rising need for long range communications. The communication engineer is therefore faced with two major problems when designing a communication system. First, the rate of transmission should be as high as possible; second, the system should be able to handle errors occurring during the transmission of information. Since the engineer is working with a communication system which is composed of discrete components (i.e. integrated circuits (IC's), transistors, etc.) it is very likely that errors may occur during the transmission of data.

A Communication System

In digital data communication systems and digital computers the coding of information is done in binary digits

"0" or "1". A communication system can be very complex, but nevertheless can be described using the block diagram shown in Fig. 1.1.1.

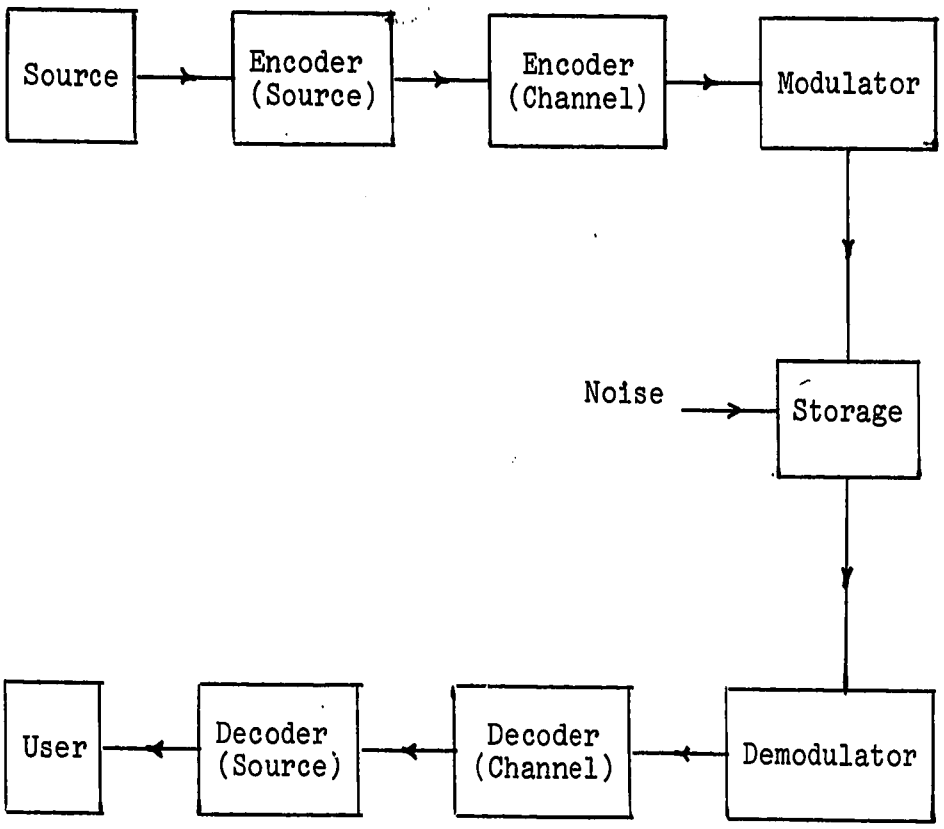


Fig. 1.1.1 Block Diagram of a Typical Communication System

The elements of this system will now be described briefly:

Source This element could be a digital computer or simply a person talking through a telephone. The output of the source could be discrete symbols (or letters) or a continuous waveform.

Channel This is a medium over which signals containing useful information are transmitted. Some examples of transmission channels are space communication links, telephone lines.

Encoder (Source) This element takes the source output and associates to each of these outputs a sequence of binary digits, i.e., a sequence of 0's and 1's. It should be noted that for efficiency reasons the output of the source encoder (i.e., the number of binary digits per unit of time necessary to represent the source output) be small. Also, one should be able to reconstruct the source symbol from the sequence generated by the source encoder.

Channel Encoder From the sequence generated by the source encoder a longer binary sequence is generated by the channel encoder and this sequence is called the code word.

Modulator Since the binary digits are not suitable for transmission over a physical channel, they are encoded by the modulator into one of two (in the binary case) physical waveforms.

Demodulator This element determines whether a "1" or a "0" was transmitted upon receiving the waveforms initially generated by the modulator. The output of the demodulator is a sequence of bits of length equal to the input sequence of the modulator.

Channel Decoder The purpose of this element is to correct errors which have occurred in the transmitted sequence. This may be done by using the rules of channel encoding and the channel characteristics. The different types of errors will be described later in the introduction.

Source Decoder Assuming that the channel decoder has corrected all the errors, then this element simply transforms the input sequence into the sent sequence symbol and delivers it to the user.

Types of Errors

There are basically two types of errors that can occur in a channel and which the decoder must be able to

detect and correct. They are the additive and synchronization errors.

Definition 1.1.1 - Additive Errors

In an ideal binary transmission system the occurrence of a "1" or a "0" at the transmitted end should be accompanied by an occurrence of a "1" or a "0" at the receiving end. If due to noise in the channel, a transmitted "1" is incorrectly interpreted (or decoded) as a "0" or a "0" is incorrectly interpreted as a "1", then we say that an additive error has occurred during which the data was transmitted. For example, suppose the sequence 1001110 is transmitted and the sequence 1000110 is received. It can be seen that an additive error has occurred in the fourth position of the transmitted sequence.

In the above example we considered the binary case where only two elements were used (i.e., "0", "1"). Unless specified, all the definitions, theorems, and results described in this thesis are considered in the binary case only. Nevertheless, it should be said here that these results can be also extended to the nonbinary case.

One of the best known classes of additive error-correcting codes is the class of cyclic codes. Since most of the codes described in this thesis are cyclic, it would be advantageous to discuss them at this point.

Definition 1.1.2 - Cyclic Codes

If V is a binary code of length n with elements $V_1, V_2, V_3 \ldots V_n$, we say that V is cyclic if for any $V_i = (a_0, \ldots a_{n-1})$ belonging to V, $(a_{n-1}\ a_0 \ldots a_{n-2})$ also belongs to V. We say that $(a_{n-1}\ a_0 \ldots a_{n-2})$ is a right cyclic shift of V_i and similarly $(a_1 a_2 \ldots a_{n-1} a_0)$ is a left cyclic shift of V_i. For example, suppose V_1 = (1 0 0 1 1 1 0); then (0 0 1 1 1 0 1) and (0 1 0 0 1 1 1) belongs to V. The class of BOSE - CHAUDHURI - HOCQUENGHEN or BCH codes, is one of the best known classes of additive error-correcting codes. Though very well suited for the correction of additive errors, cyclic codes are particularly vulnerable to synchronization errors.

Definition 1.1.3 - Synchronization Errors

In a coded system not all possible combinations of symbols ("0" and "1" for the binary case) can be transmitted; rather a family of code words is developed and only these words are transmitted through the channel. These words have a determined length which we will denote as n.

Generally these words are transmitted end to end in a continuous sequence. To maintain synchronism, the decoder counts off n bits at a time and frames them to obtain a single word. Now suppose that for some reason bits are lost or gained in a sequence or the receiver counts incorrectly, then the receiver misframes the incoming sequence and we say that a synchronization error (slip) has occurred. Therefore, the receiver must know when each new symbol, when each new word, and when each new frame begins. To explain this, consider three words, A, B, C, which belong to our dictionary,

where $A = (a_0, a_1, \ldots a_{n-1})$

$B = (b_0, b_1, \ldots b_{n-1})$

$C = (c_0, c_1, \ldots c_{n-1})$

all of length n.

In the binary case, a_i, b_i, c_i for i, j, k = 0, n-1 will be "0" or "1". Suppose these sequences are transmitted end to end, we obtain

$$a_0\ a_1 \ldots a_{n-1},\ b_0\ b_1 \ldots b_{n-1},\ c_0\ c_1 \ldots c_{n-1}$$

Now suppose that the decoder misframes the correct sequence $(b_0\ b_1 \ldots b_{n-1})$ by 1 bit; we will obtain

$$a_0\ a_1 \ldots a_{n-2} \underbrace{\left[a_{n-1},\ b_0\ b_1 \ldots b_{n-2}\right]}_{\text{received sequence}} b_{n-1},\ c_0\ c_1 \ldots c_{n-2}\ c_{n-1}$$

It can be seen that the receiver is framing part of the word A and part of the word B. Generally, if s bits are framed from A, then we say that a left slip of s bits has occurred. Similarly if the receiver had framed r bit of the word C, then we would say that a right slip of r bits has occurred.

For the purpose of this thesis it is convenient to define two types of channels, namely Type I, and Type II. In a Type I channel, it is assumed that additive and synchronization errors do not occur simultaneously. In a Type II channel, additive and synchronization errors occur simultaneously.

Synchronization Techniques

Many techniques have been developed to correct synchronization errors in both Type I and Type II channels. These techniques may be divided into two classes, namely, those that alter the word length and those that do not. Among the latter techniques we find the Subset[29], Coset[10] and Mandelbaum[14] techniques. Among the former are the "extended cyclic codes" developed by Bose and Caldwell[8] and later generalized by Weldon[9] (called the BCW technique). Tong[5] and Shiva and Seguin[15] have also examined techniques that change the word length.

In this thesis we introduce a different technique for the correction of synchronization errors. This technique can be applied to all cyclic additive-error correcting codes (BCH), and can simultaneously correct the occurrence of both additive and synchronization errors. This technique is based on the BCW technique since the encoding procedure for both techniques are almost identical. It will be shown that under certain conditions we succeed in obtaining a better rate than the BCW technique. This technique is also similar to the Mandelbaum technique.

In Chapter I we introduce some necessary basic definitions regarding cyclic error-correcting codes. The problem of synchronization is introduced and a mathematical formulation is given.

In Chapter II a survey is made of existing techniques. The techniques described are the BCW, Coset, Subset (developed by Tavares and Fukada), Mandelbaum, (also Mandelbaum's hybrid technique), Tong, and Shiva and Seguin techniques.

In Chapter III we introduce a different technique. The detecting and correcting capabilities of this new technique are proven. A comparison is made between this technique and those described in Chapter II - i.e.,

specifically the rates are compared.

In Chapter IV some conclusions are made.

CHAPTER I

CYCLIC CODES AND SYNCHRONIZATION

1.1 Properties of Binary Cyclic Codes

It is a well known fact that the transmission of data through a communication system is done more efficiently by transmitting the information as a binary bit pattern. This sequence could represent a command to which a satellite will respond. Using some form of data modulation one can achieve a probability of error as small as 10^{-7}. Sometimes, especially in a satellite communication system the power needed to achieve this reliability is too expensive and demands a transponder (receiver-transmitter) of too great a size and weight. Under these conditions one must go to some form of error-correcting codes. It is known that one can achieve, by properly encoding the information bits, probability of errors in the order of 10^{-10} as compared to 10^{-7} when the information is not encoded. This is a factor of 10^{-3}.

For the purpose of this thesis we will consider a very important class of error-correcting codes called the

binary cyclic codes. The term binary indicates that all words are represented by only two symbols "0" and a "1". The properties of binary cyclic codes have been extensively studied [1, 2, 3] and only a brief review will be given in this section.

To begin with, we will introduce a special format to represent a binary sequence, called the polynomial form. For example, the sequence

$$1\ 1\ 0\ 0\ 1\ 0\ 0$$

can be represented by the polynomial

$$P(x) = 1.1 + 1.x + 0.x^2 + 0.x^3 + 1.x^4 + 0.x^5 + 0.x^6$$

giving

$$P(x) = 1 + x + x^4$$

where the components of the sequence are treated as coefficients of a polynomial. It will become evident that such a representation is very useful when working with binary cyclic codes and also makes the implementation of encoding and decoding techniques possible.

Binary cyclic codes are said to be linear block codes with one additional property. Linear block codes are

characterized by the fact that all the code words have the same number of symbols (i.e. same block length) and constitute a linear vector space. This simply means that the addition (i.e. using the polynomial form) of one code word with another gives a third code word belonging to the same dictionary or code.

The additional property which makes binary cyclic codes so attractive is that a cyclic shift of any code word is again a code word in the same dictionary. For example, suppose a code word is represented by the following sequence

$$a_0\ a_1\ a_2\ a_3 \ldots . a_{n-1} \qquad (1.1.1)$$

where $a_i = 0, 1$, $(i = 0, n-1)$ and n is the length of the code. The polynomial form can be written as

$$a_0 x^0 + a_1 x + a_2 x^2 + a_3 x^3 + \ldots . + a_{n-1} x^{n-1} \qquad (1.1.2)$$

where the degree of the polynomial is (n-1). A cyclic shift of one symbol to the left in (1.1.1) can be represented by

$$a_{n-1}\ a_0\ a_1\ a_2\ a_3 \ldots . a_{n-2}$$

or

$$a_{n-1} + a_0 x + a_1 x^2 + a_2 x^3 + a_3 x^4 + \ldots + a_{n-2} x^{n-1}$$

which can be obtained from (1.1.2) by multiplying by x and assuming that $x^n = 1$. This means that the powers of a polynomial representing a code word are to be taken module n which is the remainder of a division by n.

A binary cyclic code of length n can be written as the product of a polynomial g(x) which divides $1 + x^n$ and an information polynomial I(x). The polynomial g(x) is called the generator polynomial. If we assume that V(x) is a code word (in the polynomial form) then we can say that

$$V(x) = g(x)\, I(x) \qquad (1.1.3)$$

where V(x) is the code word and has degree $\leq (n-1)$

I(x) is the information polynomial and has degree $\leq (k-1)$

g(x) is the generator polynomial and has degree $(n-k)$

and k is the number of information bits in every code word.

The generator polynomial is found from the factors of $1+x^n$ such that

$$1+x^n = g(x)\, h(x) \qquad (1.1.4)$$

where $h(x)$ is called the parity check polynomial. Therefore if we want to generate a code of length n capable of transmitting k information bits we choose a polynomial $g(x)$ of degree $(n-k)$ which is a factor of $1+x^n$. For example the polynomial $1+x^7$ can be factored as

$$1+x^7 = (1+x+x^3)\,(1+x^2+x^3)\,(1+x)$$

from which we can choose the generator polynomial as $g(x) = 1+x+x^3$ to generate the code. We call this a (7,4) binary cyclic code.

The code is generated by the product $g(x)I(x)$. Since $I(x)$ must have degree at most equal to 3 it can form $2^4=16$ distinct polynomials and hence the code will contain 16 words.

Error Detection and Correction

One important class of binary cyclic codes are the (n,k,t) BCH codes where t denotes the number of additive errors that the code can correct. For the purpose of this thesis we will consider only BCH codes. Error-correcting

techniques for BCH codes have been studied and explained by many authors[1,2] and will not be described in this thesis.

1.2 Synchronization

In this section we analyze the different ways possible to represent a received sequence which contains synchronization errors and give a mathematical description.

To begin, we will analyze mathematically what happens to a transmitted code word which has suffered synchronization loss during transmission in the case where the length is not altered before transmission, in the case where the length of the code word is increased, and finally in the case where the length is shortened. Also described is the case where we restrict certain information bits. In Chapter II we will see that the different techniques now existing use at least one of these configurations.

Case 1. The length is not changed

It was shown in section 1.1 that a binary (n,k) cyclic code V(x) can be expressed as

$$V(x) = g(x)\, I(x)$$

where g(x) is the generator polynomial of degree (n-k)

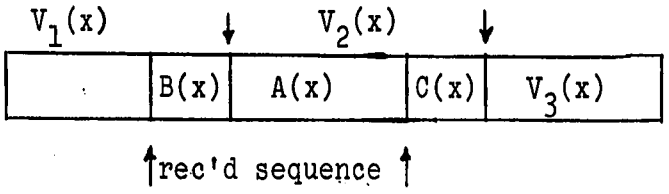
and I(x) is the information polynomial of degree $\leq$(k-1).

Assume that three code words $V_1(x)$, $V_2(x)$, $V_3(x)$ are transmitted end to end through the channel. The transmitted sequence will be

$V_1(x)$	$V_2(x)$	$V_3(x)$

where each code word have length equal to n.

Now suppose the decoder miscounts while receiving $V_2(x)$ and slips to the left by s bits. The received sequence will be



It can be seen that the received sequence contains part of $V_1(x)$ and part of $V_2(x)$. If we assume that $V_2(x) = a_0 + a_1x + a_2x^2 + a_3x^2 + \ldots + a_{n-1}x^{n-1}$ and $B(x) = b_0 + b_1x + b_2x^2 + \ldots + b_{s-1}x^{s-1}$

where B(x) is an arbitrary polynomial of degree $\leq$s-1.

The received polynomial denoted by R(x) can be written as

$$R(x) = b_0 + b_1x + \dots b_{s-1}x^{s-1} + a_0x^{s+1} + \dots + a_{n-s-1}x^{n-1}$$

This can be written as

$$R(x) = b_0 + b_1x + \dots + b_{s-1}x^{s-1} + x^s \left[a_0 + a_1x + \dots + a_{n-s-1}x^{n-s-1}\right] + x^s \left[a_{n-s}x^{n-s} + \dots + a_{n-1}x^{n-1}\right] + x^s \left[a_{n-s}x^{n-s} + \dots + a_{n-1}x^{n-1}\right]$$

which becomes

$$R(x) = x^s \left[a_0 + a_1x + \dots + a_{n-1}x^{n-1}\right] + \left[b_0 + b_1x + \dots + b_{s-1}x^{s-1}\right] + x^n \left[a_{n-s} + a_{n-s-1}x + \dots + a_{n-1}x^{s-1}\right] \quad (1.2.1)$$

Since $x^n = 1$, we can treat the above expression as being $R(x) = x^s V_2(x) + U_s(x)$ (1.2.1) where $U_s(x)$ is an arbitrary polynomial of degree $\leq s-1$.

Similarly for a right slip of magnitude r, the received sequence will be

$$R(x) = x^{-r} V_2(x) + x^{n-r} U_r(x) \qquad (1.2.2)$$

where $U_r(x)$ is an arbitrary polynomial of degree $\leq r-1$. It can be seen from 1.2.1 and 1.2.2 that the received vector can be treated as the sum of a code word (cyclic shift of the transmitted code word) and an error with a maximum weight equal to s for a left slip and r for a right slip. In the remainder of this section we will assume that r=s.

If during transmission the word was corrupted with some additive error polynomial E(x) of weight e, then the received sequence will be from 1.2.1

$$R(x) = x^{s} V_2(x) + U_s(x) + E_1(x) \qquad (1.2.3)$$

for a left slip. For a right slip the sequence will be from 1.2.2

$$R(x) = x^{-s} V_2(x) + x^{n-s} U_s(x) + E_2(x) \qquad (1.2.4)$$

where $E_1(x)$ and $E_2(x)$ have the same weight as E(x). Since we can assume that $U_s(x)$ is also an additive error, then the total weight of the error in 1.2.3 and 1.2.4 is equal maximally to s+e.

Case 2. The Length is Increased

In Case 1 it was shown that the received sequence was not cyclic but contained an extra term which we can treat as an additive error. A technique [8] can be used which will ensure that even if a synchronization error has occurred, the received sequence will be a cyclic shift of the transmitted sequence. The technique increases the length of the sequence by 2s bits (where s is the maximum slip) before transmission by annexing in a fixed manner bits at the beginning and at the end of the sequence. That is, the first s bits of the sequence are placed at the end and the last s bits are placed at the beginning. For example, consider a cyclic code V(x) of lengths n=7 which can be represented by the sequence

$$U_0\ U_1\ U_2\ U_3\ U_4\ U_5\ U_6$$

If we increase the length (assume slip s=1) using the technique described above, we obtain the sequence (of length 7+2 = 9)

$$U_6\ U_0\ U_1\ U_2\ U_3\ U_4\ U_5\ U_6\ U_0$$

Suppose that during transmission of this sequence a slip has occurred. The received sequence will be

$$x\ U_6\ U_0\ U_1\ U_2\ U_3\ U_4\ U_5\ U_6$$

where x is arbitrarily 0 or 1 from another word. We decrease the length (equal to 9) to its original size (n=7) by removing the first and last s bits (in this example s=1). The truncated sequence will be

$$U_6\ U_0\ U_1\ U_2\ U_3\ U_4\ U_5$$

which is simply a cyclic shift of the original code word. Similarly for a right slip (s=1) the truncated sequence would have been

$$U_1\ U_2\ U_3\ U_4\ U_5\ U_6\ U_0$$

which is also a cyclic shift of V(x).

Case 3. The Length of the Code is Shortened

Assume that every code word is shortened by 2s+1 digits where s is the maximum allowable slip. If we consider three code words A(x), B(x), and C(x) each of length $n_1 = n-2s-1$ then we have that

$$A(x) = a_0 + a_1x + a_2x^2 + \ldots + a_{n_1-1}x^{n_1-1}$$

$$B(x) = b_0 + b_1x + b_2x^2 + \ldots + b_{n_1-1}x^{n_1-1}$$

and $$C(x) = c_0 + c_1x + c_2x^2 + \ldots + c_{n_1-1}x^{n_1-1}$$

Suppose A(x), B(x), and C(x) are transmitted end to end, and a left slip of magnitude $i \leq s = 1$ occurs as shown in the following sequence

$$a_0a_1a_2\ldots\left[a_{n_1-1},\ b_0b_1b_2\ldots b_{n_1-2}\right]\ b_{n_1-1},\ c_0c_1c_2\ldots c_{n_1-1}$$

↑ received sequence ↑

The received sequence is

$$a_{n_1-1}\ b_0b_1b_2\ldots b_{n_1-2}$$

which can be written as

$$a_{n_1-1} + b_0x + b_1x^2 + b_2x^3 + \ldots + b_{n_1-2}x^{n_1-1} \qquad (1.2.5)$$

The polynomial (1.2.5) can be written as

$$a_{n_1-1} + x\left[b_0 + b_1x + b_2x^2 + \ldots + b_{n_1-2}x^{n_1-2}\right] + x\left[b_{n_1-1}x^{n_1-1}\right] + x\left[b_{n_1-1}x^{n_1-1}\right] \qquad (1.2.6)$$

In (1.2.6) the first term represents a polynomial of weight equal to 1 occurring in the first position and the last term is a polynomial of weight equal to 1 occurring in the last position. The second and third terms describes a cyclic shift of B(x). This means that (1.2.6) can be written as

$$A_1(x) + x\,B(x) + x^{n_1}B_1(x) \qquad (1.2.7)$$

More generally for a slip of maximum magnitude s the received polynomial will be

$$A_1(x) + x^{s}B(x) + x^{n-s}B_1(x) \qquad (1.2.8)$$

where $A_1(x)$ and $B_1(x)$ are arbitrary polynomials of weight $\leq s$.

Similarly it can be shown that when a right slip of magnitude s occurs the received sequence will be

$$x^{-s}B(x) + x^{n-i}B(x) + x^{n_1-i}C_1(x) \qquad (1.2.9)$$

where $B_1(x)$ and $C_1(x)$ are polynomials of weight $\leq s$.

Case 4. Restriction of Information Bits

In this case we will show that in the presence of slip error it is possible to ensure that the received sequence is a cyclic shift of the transmitted sequence without altering the length of the code word. This can be done by assuming that every transmitted code word starts and terminates with s or more zeros. For example, if we transmit two code words, A(x), B(x), each of length n=7, and suppose the slip s=1, the transmitted sequence will be

$$0a_1a_2a_3a_4a_50,\ 0b_1b_2b_3b_4b_50,$$

where $A(x) = 0 + a_1x + a_2x^2 + a_3x^3 + a_4x^4 + a_5x^5 + 0.x^6$

and $B(x) = 0 + b_1x + b_2x^2 + b_3x^3 + b_4x^4 + b_5x^5 + 0.x^6$

If a slip s=1 to the left had occurred when the decoder was receiving B(x), the sequence would be

$$0\ 0\ b_1\ b_2\ b_3\ b_4\ b_5$$

which is a cyclic shift of B(x). Similarly for a right slip the received sequence would be

$$b_1\ b_2\ b_3\ b_4\ b_5\ 0\ 0$$

which is also a cyclic shift.

It will be shown in Chapter II that to detect and correct a synchronization error one must restrict one more information bit. This means that 2s+1 information bits would be reserved so that synchronization errors can be corrected. Since the received sequence is simply a cyclic shift of the original sequence, then the decoder can correct any additive errors that occurred during transmission.

In summary, Cases 2 and 4 do not necessitate that the error-correcting capability of the code be increased in order to correct synchronization errors. On the other hand, in Cases 1 and 3, some additive-errors may be created when slip errors occur and must be taken into account when choosing the code to be used.

CHAPTER II

EXISTING TECHNIQUES

2.1 Bose-Caldwell Technique

This technique was discovered by Bose and Caldwell[8] and applied only to BCH codes. Later Weldon[9] generalized this technique to apply to any (n,k) binary cyclic codes where n is the length and k is the number of information bits in the code word.

The philosophy behind the BCW technique can be described using Figure 2.1.1.

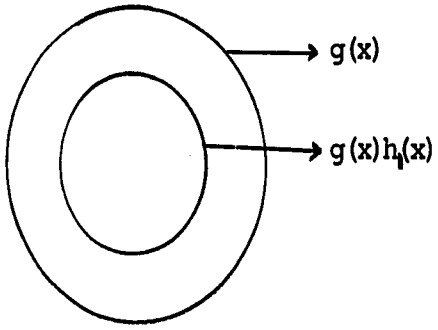


Fig. 2.1.1

The larger circle represents all code words generated by g(x) and the inner circle represents all code words generated by g(x) $h_1(x)$ where $h_1(x)$ is a factor of the parity check

polynomial h(x) and has exponent u. This means that $h_1(x)$ divides $1+x^u$ but not $1+x^i$ for $i<u$. If we choose only those code words which are generated by g(x) but not by g(x) $h_1(x)$ then this code (following a special encoding procedure) can be used to correct slip errors. This can be achieved by adding to the code words generated by g(x) $h_1(x)$ a polynomial c(x). For example, let c(x) = g(x). Therefore, a particular code word V(x) will look like the following:

$$V(x) = g(x)\, h_1(x)\, I(x) + g(x) \qquad (2.1.1)$$

where I(x) is an information polynomial of degree $<(k-m)$ given that m is the degree of $h_1(x)$. The encoding is completed by increasing the length of (2.1.1) as described in (Chapter I, Case 2). This will assume that the cyclic property of the code word will be retained even if slip errors occur during transmission.

Next we will show how the slip correcting capability of the code is dependent on the exponent u of the auxilliary polynomial $h_1(x)$. Suppose that when V(x) is transmitted a left slip error of magnitude i occurs. After truncation the result is simply a cyclic shift of V(x) given by

$$x^i V(x) = x^i g(x) \left[I(x) h_1(x) + 1 \right] \qquad (2.1.2)$$

For the moment we assume the absence of additive errors. From (2.1.2), after dividing by g(x), we obtain that

$$x_i\left[I(x)h_1(x) + 1\right] \bmod h_1(x) = x^i \bmod h_1(x) \qquad (2.1.3)$$

The expression $x^i \bmod h_1(x)$ is called the slip syndrome. We must prove that the slip syndrome is distinct for all values of slip given by $-r \leq i \leq s$ where r and s are the maximum right and left slip that the code can correct. The slip syndromes are distinct if

$$x^s \bmod h_1(x) \neq x^r \bmod h_1(x) \qquad (2.1.4)$$

where $r \neq s$. Expression (2.1.4) can be written as

$$(x^s - x^r) \bmod h_1(x) \neq 0$$

or

$$x^s(1 + x^{r-s}) \bmod h_1(x) \neq 0 \qquad (2.1.5)$$

Since $h_1(x)$ does not divide x^s, (2.1.5) is true only if $r+s<u$ where u is the exponent of $h_1(x)$. Suppose that $r=s$ then the restriction on $h_1(x)$ is that its exponent u be greater than 2s.

It can be seen that $x^s V(x)$ in (2.1.1) is a code word generated by g(x) and hence has the same error

correcting capability as the parent code. This can be shown by assuming that an error E(x) of weight e has occurred during transmission in (2.1.1) giving after truncation

$$x^i V(x) + E(x)$$

giving an error syndrome

$$(x^i V(x) + E(x)) \bmod g(x) = E(x) \bmod g(x)$$

which can be corrected using some known procedure.

The decoding procedure can be summarized in the following steps:

1. Receive the incoming sequence and truncate.
2. Correct all additive errors using g(x).
3. Find the slip syndrome using $h_1(x)$ and if necessary correct for slip errors.
4. Remove c(x) (c(x) = g(x) for example).

Step three is possible because all slip syndromes are distinct, even for the case of no slip errors. The results obtained from the BCW technique can be summarized in the following theorem.

Theorem 2.1.1

Given an (n,k) cyclic code with a parity check polynomial of degree m and exponent u, this code can be extended to form a (n+2s, k-m) code that can correct the simultaneous occurrence of a slip of s digits and any additive error pattern that the parent code can correct, where $2s<u$. The rate obtained using the BCW technique is $\frac{k-m}{n+2s}$ where m is the degree of $h_1(x)$ and s is the maximum allowable slip.

For example consider the (7,4) cyclic code generated by

$$g(x) = 1 + x + x^3$$

where $1 + x^7 = (1 + x + x^3)(1 + x)(1 + x^2 + x^3)$

and $h(x) = (1 + x)(1 + x^2 + x^3)$

Suppose we choose $h_1(x) = 1 + x^2 + x^3$. It can be shown that $1 + x^2 + x^3$ has exponent 7. Therefore, if all code words have the form

$$V(x) = g(x)\left[I(x)\, h_1(x) + 1\right]$$

then this code has $2s<7$ and can correct a slip error less than or equal to 3. Also it can simultaneously correct up to 1 additive-error.

Modified BCW Technique

The BCW technique can be modified so that the length need not be increased. It was shown that the reason for increasing the length was to preserve the additive error correcting capability of the code even during synchronization errors. This can also be achieved by assuring that all code words end and start with s zeros. The transmitted code word will be of the form

$$V(x) = g(x)\, h_1(x)\, I(x) + g(x)$$

and will look like the following

```
|<-s->|          |<-s->|
,00....0xxxxxxxxx00....0, 00....0xxxxxxxxx00....0,
|<----------n---------->|
```

where "x" denotes an arbitrary "1" or "0".

It can be seen that a slip to the left or to the right will result in a cyclic shift of the code word. Hence, any additive errors that occur during transmission can be corrected. Also the same BCW decoder can be used to find the slip error. The rate now becomes

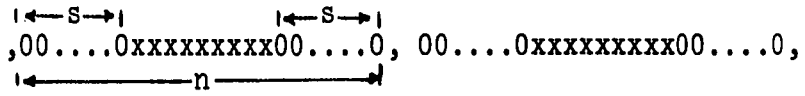
$$\frac{R-m-2s}{n}$$

Modified BCW Technique

The BCW technique can be modified so that the length need not be increased. It was shown that the reason for increasing the length was to preserve the additive error correcting capability of the code even during synchronization errors. This can also be achieved by assuring that all code words end and start with s zeros. The transmitted code word will be of the form

$$V(x) = g(x)\, h_1(x)\, I(x) + g(x)$$

and will look like the following



where "x" denotes an arbitrary "1" or "0".

It can be seen that a slip to the left or to the right will result in a cyclic shift of the code word. Hence, any additive errors that occur during transmission can be corrected. Also the same BCW decoder can be used to find the slip error. The rate now becomes

$$\frac{R-m-2s}{n}$$

which is always slightly lower than the rate of the BCW technique given as

$$\frac{k-m}{n+2s}$$

Conclusions

The interesting thing about the BCW technique is that the amount of slip that the code can correct is solely dependent on the exponent and not on the degree of $h_1(x)$. The BCW technique would be particularly attractive for the cases where $h_1(x)$ can be chosen among the factors of $h(x)$ for minimal degree and suitable value of exponent depending on the maximum slip value. The BCW technique would be best suited for codes of large length and cases of large slip errors. For example, for $n=31$, $h_1(x)$ has exponent 31 (capable of correcting up to $s=15$) and has a degree=5. But on the other hand, the increase in length does have a predominant influence on the rate as the magnitude of the slip to be corrected increases.

2.2 Subset Codes

The subset codes are very similar to the BCW codes in the way in which the information bits are restrained.

This means that the parity check polynomial $h_1(x)$ has degree m and exponent u and the exponent must be such that u>2s where s is the magnitude of the slip.

The technique of the subset code consists in choosing a subset of an (n,k) cyclic code which will be generated by g(x) but not by g(x) $h_1(x)$ where $h_1(x)$ is the parity check polynomial. In Fig. 2.2.1 all code words belonging to the subset code belong to the subset V_2. This can be achieved by adding to any code word generated by g(x) $h_1(x)$ another code word generated by g(x) but not

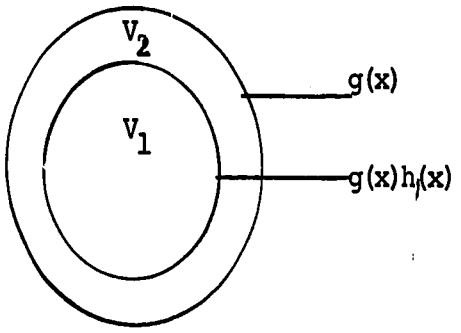


Fig. 2.2.1

by g(x) $h_1(x)$. For example, suppose $V_1(x)$, generated by g(x) $h_1(x)$, has the form $V_1(x) = g(x)\, h_1(x)\, I(x)$ where I(x) is an information polynomial of degree <k-m. If we add g(x) to $V_1(x)$ we obtain $V_1(x) + g(x) = V(x) = g(x)[h_1(x)I(x)+1]$. We can see that V(x) can be treated as a code word not belonging to the subclass generated by g(x) $h_1(x)$. Also it should be noted that unlike the BCW procedure no redundant bits are added to the subset code before transmission.

Suppose $V(x) = g(x)\ (h_1(x)\ I(x) + 1)$ is sent through the channel and a left slip of magnitude s occurs. The received vector will have the following representation

$$x^s\ V(x) + u_s(x) + E(x) \qquad (2.2.1)$$

where $u_s(x)$ is an auxilliary polynomial of weight s or less

and $E(x)$ is an additive error with weight e.

If we assume that the parent code has a t- additive error correcting capability where $t \geq s + e$, then $(u_s(x) + E(x))$ can be treated and corrected as if they were a simple additive-error. After correction (2.2.1) becomes $x^s V(x)$ which has a slip error syndrome equal to $\left(\frac{x^s V(x)}{g(x)}\right)$ modulo $h_1(x)$. We can determine the magnitude and direction of s if the slip syndromes are distinct for all $-r \leq i,\ j \leq s$ where r and s are the maximum right and left slip respectively. This means that for any $-r \leq i,\ j \leq s$ we have that

$$x^i \text{ modulo } h_1(x) \neq x^j \text{ modulo } h_1(x)$$

or

$$\left(x^i(1 + x^{j-i})\right) \text{ modulo } h_1(x) \neq 0 \qquad (2.2.2)$$

Since $h_1(x)$ has exponent u, 2.2.2 is true of $u>2s$ (assuming $s=r=s$). This simply means that for any $-s\leq i\leq s$ any cyclic code $x^iV(x)$ will not belong to V_1 of Fig. 2.2.1 and therefore the slip can be detected and corrected.

The following theorems are concerned with the ability of the subset codes to detect and correct slip in noiseless, Type I, and Type II channels.

Theorem 2.2.1

Given an (n,k) cyclic code whose parity check polynomial has degree m and exponent u, then there exists an (n,k-m) subset code with $u>s$ that can detect s or less digits of slip in a noiseless channel provided $s\leq n-k$.

Theorem 2.2.2

Given an (n,k) cyclic code whose parity check polynomial has degree m and exponent u, then there exists an (n,k-m) subset code with $u>2s$ that can correct s or less digits of slip in a noiseless channel provided $s\leq[(n-k)/2]$*.

Theorem 2.2.3

Given an (n,k) cyclic code whose parity check

---

* $[x]$ *means the integral part of* x.

polynomial has degree m and exponent u, then there exists an (n,k-m) subset code having $u \geq d$ that can detect any combination of s digits of slip and e additive errors provided $e+s \leq d-1$.

Theorem 2.2.4

Given an (n,k) cyclic code whose parity check polynomial has degree m and exponent u, and a Type I channel, then there exists an (n,k-m) subset code that can correct s or less digits of slip and any additive error pattern that the parent cyclic code can correct, provided that $s \leq [(d-1)/2]$.

Theorem 2.2.5

Given a t-error-correcting (n,k) cyclic code whose parity check polynomial has a factor of degree m and exponent u, then there exists an (n,k-m) subset code with $u > t$ that can correct any additive error pattern that the parent can correct in the absence of slip and in addition can detect the presence of a slip of s digits in the presence of e additive errors if $e+s \leq t$.

Theorem 2.2.6

Given an (n,k) cyclic code with an error-correcting

capability of t errors or less, with a parity check polynomial of degree m and exponent u, and a Type II channel, then there exists an $(n,k-m)$ subset code with $u>2s$ that can correct any combination of e additive errors and s digits of slip if $e+s\leq t$.

The proof of these theorems are based on the fact that the parity check polynomial is chosen among the factors of $h(x)$ where $h(x) = 1 + x^n/g(x)$ and has an exponent u which means that it divides $1 + x^u$ but no $1 + x^i$ for $i<u$. Also the auxilliary polynomial $u_s(x)$ or $x^{n-r}u_r(x)$ from (2.2.1) and (2.2.2) are treated as additive errors and are corrected accordingly.

The decoding procedure can be described in the following steps:

Step 1 Receive the incoming sequences and correct for any additive error pattern $(u_s(x) + E(x)$ or $(x^{n-r}u_r(x) + E(x))$.

Step 2 Find the slip syndrome using the parity check matrix.

Step 3 From the slip syndrome find the magnitude and direction of the slip and shift accordingly.

Conclusion

This decoding procedure is simple and requires little computer memory storage if the slip is small. When the slip is large the decoder must retrieve the slip syndrome using a search process which could be time consuming and expensive in storage devices. A decoding procedure would be needed therefore which would find the magnitude and direction of the slip with minimum storage.

The proof for the subset code technique is very similar to the one presented by Weldon[9]. The only difference in the subset code technique is that the polynomial $u_s(x)$ which is a result of the slip error, is corrected using the additive error-correcting capability of the code. After the correction is made the resultant vector is the same as if we had used the BCW technique. We also pay the price of reduced rate because the subset code must correct more errors than it really has to.

Rate of Subset Codes - In Type II Channels

The rate of an (n,k,t) subset code is given by

$$R_{sub} = \frac{k-m}{n} \qquad (2.2.3)$$

where m is the degree of the auxilliary polynomial $h_1(x)$. Also the parent code is t error-correcting such that $t \geq x+e$ where s is the maximum slip.

Because the subset code must correct an additional number of additive errors (i.e. s) to be able to determine the slip error, the rate will decrease rapidly as the slip or the number of additive errors increases.

In Table 2.2.1, it can be seen that for n=31, s=3, e=1 the rate is only equal to .129 as compared to the BCW rate of .514. Even if the subset code can correct 5 additive errors when the no slip error occurs the rate is much too small to recommend its use. For n=127 the rate is reasonably high for small values of slip but does not achieve better rates than the BCW technique.

TABLE 2.2.1

s	e	Rate Subset	eo	Rate BCW	eo
n=31					
1	1	.452	2	.576	1
1	2	.290	3	.424	2
1	3	.129	5	.273	3
2	1	.290	3	.543	1
2	2	.129	5	.400	2
2	3	.129	5	.257	3
3	1	.129	5	.514	1
3	2	.129	5	.378	2
3	3	-	7	.243	3
n=127					
1	1	.835	2	.876	1
1	2	.780	3	.822	2
2	1	.780	3	.863	1
2	2	.724	4	.809	2
3	1	.724	4	.850	1
3	2	.669	5	.797	2
4	1	.669	5	.837	1
4	2	.614	6	.785	2
5	1	.614	6	.825	1
5	2	.559	7	.774	2

s: slip
e: additive errors
eo: errors that code can correct when s=0

2.3 Coset Code Techniques

The coset codes have been the subject of many authors for their use in correcting slip errors in noiseless, Type I, and Type II channels. Even if the coset code technique is very well understood, only recently has a simple and efficient decoding technique been introduced[17]. Before this, one had to calculate the syndrome of the received word using the generator polynomial g(x) and compare it to all possible stored syndromes to determine the slip error. Needless to say that this decoding scheme required a computer for storage and computation which is in itself very expensive and time consuming. Because of the importance of this new decoding technique introduced by Lewis and Fukada[17] we will first of all describe briefly the old decoding scheme with its limitations and then describe the new decoding technique in detail.

Definition 2.3.1 - Coset Code

Basically, a coset code is formed by adding a fixed polynomial to a given (n,k) binary cyclic code before transmission. As a result a word V(x) belonging to the coset code can be written as

$$V(x) = [I(x)g(x) + C(x)] \qquad (2.3.1)$$

where $V(x)$ is the word belonging to the coset

$g(x)$ is the generator polynomial of the cyclic code

$I(x)$ is an arbitrary polynomial of degree $\leq k-1$

and $C(x)$ is the polynomial used to form the coset.

The sequence represented by (2.3.1) is transmitted through the channel. At the receiver end the decoder immediately subtracts (or adds in the binary case) the polynomial $C(x)$ from the received word. This will result in a sequence which will be used to identify a right or left slip or no-slip errors at all. Once the slip error has been corrected the information can be retrieved. For the case of "no-slip" error, adding the polynomial $C(x)$ simply gives back the original generated code word $I(x)$ $g(x)$. Now suppose a left slip has occurred of magnitude s. From Chapter I the received polynomial will be

$$R(x) = x^{s}V(x) + U_{s}(x)$$

or

$$x^{s}[I(x)g(x) + C(x)] + U_{s}(x)$$

Adding $C(x)$ gives

$$R(x) + C(x) = x^{s}I(x)g(x) + C(x)(1 + x^{s}) + U_{s}(x) \qquad (2.3.2)$$

Similarly for a right slip of magnitude r we have that

$$R(x) + C(x) = x^{n-r}I(x)g(x) + C(x)(1 + x^{n-r}) + x^{n-r}U_r(x) \quad (2.3.3)$$

The problem now consists in choosing a suitable C(x) such that the slip error in (2.3.2) or (2.3.3) can be detected and corrected.

For the noiseless case a suitable choice for C(x) is $(1 + x^{n-1})$. This can be interpreted as introducing errors in the first and last position of the code word. Using this C(x) will guarantee that not both terms will be cancelled by $U_s(x)$ in (2.3.2) or $x^{n-r}U_r(x)$ in (2.3.2) for some left or right slip of magnitude s and r respectively. This means that (2.3.2) or (2.3.3) will never be a code word. If all slip errors are to be identified using syndromes, then these syndromes must be distinct. This means that

$$[x^sV(x) + c(x)\ (1 + x^s) + U_s(x)] \bmod g(x) \neq$$

$$[x^{-r}V(x) + C(x)\ (1 + x^{-r}) + x^{n-r}U_r(x)] \quad (2.3.4)$$

Substituting for $C(x) = 1 + x^{n-1}$ and bring all terms to the left hand side we obtain for s=r=s

$$[x^s + x^{n-s-1} + U_s(x) + x^{n-s}U_s(x)] \bmod g(x) \neq 0 \quad (2.3.5)$$

where $U^1_s(x) = U_r(x)$

The weight of $[x^s + x^{n-s-1} + U_s(x) + x^{n-s}U^1_s(x)]$ is equal to 2s + 2 and hence the expression (2.3.5) is true only if

$$\underline{2s + 2 < n-k}$$

or

$$\left[s \leq \frac{n-k-2}{2}\right] \qquad (2.3.6)$$

since no burst of length $\leq n-k$ is a code word. The decoder must also distinguish between two left slips and two right slips. The proof is not given since the bound on s is larger than (n-k-2)/2 and hence need not be considered.

For a Type I channel a suitable choice for C(x) is

$$C(x) = \sum_{i=0}^{[e/2]} x^{i(s+1)} + (e - 2[e/2])\, x^{n-1}$$

where e is the number of additive errors the code can correct in the absence of slip

and s is the maximum slip (right or left) the code can correct in the absence of additive errors.

The bound on s is given by

$$S = \text{Min}\,[\text{Max}(s_1, s_2),\ s_3]$$

where $s_1 = d - 2(e + 1)$, $d = 2t + 1$

$$s_2 = [(3d - s)/2] - 3e - [e/2]$$

$$s_3 = [(n-R-e + [e/2] - 1) / ([e/2] + 2)]$$

where [y] still denotes the integral part of y.

Finally for a Type II channel a suitable choice of C(x) is

$$C(x) = x^{n-1} + \sum_{i=0}^{e} x^{i^{(2s+1)}}$$

where e and s are the additive and slip errors the decoder can correct simultaneously.

The bound on s is

$$s = \min [(d-4e-s)/2, (n-e-2)/2(e+1)]$$

It is not felt necessary to give a complete proof of the bounds for s in Type I and II channels. The interested reader can consult Tavares[27].

We are mainly concerned in this section with the new decoding procedure for the coset codes[17]. It should be noted that the decoding procedure applies only to noiseless and Type I channels. For the sake of completness a

simple decoding procedure will also be described for the Type II channel.

Decoding Procedure

This decoding procedure was introduced by Lewis and Fukada[17]. It consists in computing the product $h(x)R(x)$, modulo $1 + x^n$ where $h(x)$ is the parity polynomial given by $h(x) = 1 + x^n/g(x)$ and $R(x)$ is the received word.

For the sake of simplicity the noiseless case is considered where we have shown that a suitable choice for $C(x)$ is $1 + x^{n-1}$ and the bound on s is $[(n-k-2)/2]$. In the presence of a left slip of magnitude s the received polynomial $R(x)$ can be written from (2.3.5) as

$$R(x) = I(x)g(x) + x^s + x^{n+s-1} + U_s(x) \qquad (2.3.7)$$

where $U_s(x)$ has been defined earlier in this section. The expression (2.3.7) can be treated as a cyclic code word $I(x)g(x)$ with an error $x^s + x^{n-s-1} + U_s(x)$. Since $U_s(x)$ has degree $\leq s-1$ we can also treat $x^s + x^{n-s-1} + U_s(x)$ as a burst error of length $\leq s-1$ occurring in the first $s+1$ positions. Similarly, for a right slip the error $x^{-R} + x^{n-R-1} + x^{n-R} U_R(x)$ assumes the form of a burst of length $\leq (R+1)$ and occurring in the last $(R+1)$ positions.

If we take the case of a left slip it can be seen from (2.3.7) that the magnitude of $\underline{s}$ must be found and also the presumed burst error $x^s + x^{s-1} + U_s(x)$ must be corrected.

(1) Burst Error Correction

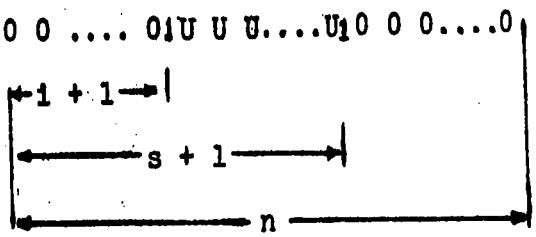
If we multiply R(x) by the parity check polynomial h(x) and take modulo $1 + x^n$ we obtain

$$R(x)h(x), \text{ mod } 1 + x^n = \left\{x^s I(x)g(x)h(x) + (x^{s-1} + x^s + U_s(x))h(x)\right\}, \text{ modulo } 1 + x^n \quad (2.3.8)$$

Since $g(x)h(x) = 1 + x^n$, the first term vanishes. We therefore obtain

$$R(x)h(x), \text{ mod } 1 + x^n = (x^{s-1} + x^s + U_s(x))h(x), \text{ mod } 1 + x^n$$

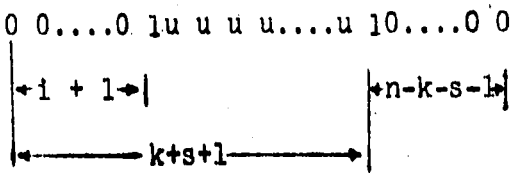
Since $s \leq [(n-k-2)/2] < \deg g(x)$, the degree of $(x^{s-1} + x^s + U_s(x))h(x)$ is always smaller than $\underline{n}$ and therefore we do not have to take modulo $1 + x^n$. Suppose $\underline{i}$ is the power of the first non-zero coefficient of the burst error $(x^{s-1} + x^s + U_s(x))$. This can be shown in the following way:



Also we assume that

$$x^i B(x) = (x^{s-1} + x^s + U_s(x))$$

where B(x) has degree (s-i). The product B(x)h(x) will have the following form:



where k is the degree of h(x). It can be seen that the position of the first 1 from the left denotes the start of the burst error $x^i B(x)$, the position of last 1 to the right followed by n-k-s-1 zeros, giving the end of the burst $x^i B(x)$, can be determined since the position of the last 1 is k+s+1. The start and end of the burst error is removed and then the procedure is restarted until $x^i B(x)$ has been removed from R(x).

(2) We must also find the magnitude s of the slip. This is easily done since the sum k+s+1 can be determined from the position of the last 1, from which s can be determined.

Example Consider the (15,11,1) BCH cyclic code generated by $g(x) = 1 + x + x^4$ and with parity check polynomial $h(x) = 1 + x^n/g(x) = 1 + x + x^2 + x^3 + x^5 + x^7 + x^8 + x^{11}$.

We also choose $C(x) = 1 + x^{14}$.

Suppose we transmit two code words $A_1(x)$ and $B_1(x)$. The coset code words are then

$$A_1(x) = 1 + x + x^4 + C(x) = x + x^4 + x^{14}$$

$$B_1(x) = x + x^2 + x^5 + C(x) = 1 + x + x^2 + x^5 + x^{14}$$

Assuming $A_1(x)$ and $B_1(x)$ are transmitted end to end, the transmitted sequence will be

0 1 0 0 1 0 0 0 0 0 0 0 0 0 [1, 1 1 1 0 0 1 0 0 0 0 0 0 0 0] 1

Suppose a left slip has occurred and the sequence in bracket is received such that

$$R(x) = 1 + x + x^2 + x^3 + x^6$$

The product $R(x)h(x)$, mod $1 + x^n$ gives

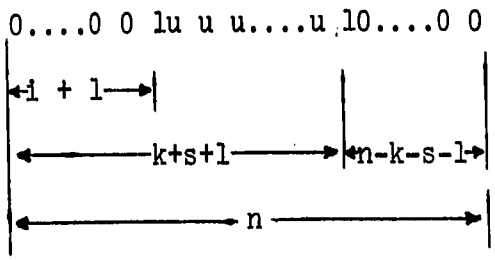
$$1 + x^4 + x^5 + x^6 + x^7 + x^9 + x^{11} + x^{12}$$

which will be represented as

1 0 0 0 1 1 1 1 0 1 0 1 1 0 0 (2.3.9)

The last 1 in (2.3.9) occurs in the 13th position and is followed by two zeros. Therefore, the magnitude of the slip s is 13-k-1 = 1. The burst error can be corrected by noting that the first position is a 1 and hence denotes the start of the burst error. Since s=1 we have that $x^{i}B(x) = 1 + x$. R(x) becomes $x^2 + x^3 + x^6$ and shifting one position to the right we obtain $x^{-1}R(x) = x + x^2 + x^5$ which is the transmitted code word. Since the burst length was equal to 2 the procedure did not have to be repeated.

This technique can be easily implemented using shift registers, nor gates, and adders. It can be seen from the product representation



that one must simply look for the sequence 1 0....0 0 occurring in the last (n-k) positions to find the slip error. The correction of the presumed burst error is done very easily.

If no slip error has occurred then the product sequence has in its last (n-k) position the sequence

1 0....0 1. This could be used to check if correct synchronization exists. For example

$$(1 + x^{14})\,(1 + x + x^2 + x^3 + x^5 + x^7 + x^8 + x^{11}) \bmod 1 + x^{15}$$

$$= (0\ 0\ 0\ 1\ 1\ 1\ 1\ 0\ 1\ 0\ 1\ 1\ 0\ 0\ 1)$$

We can see that in the last n-k = 4 position we have the sequence 1 0 0 1 which means that proper synchronization existed.

The same procedure can be used for correcting a right slip of magnitude r. The only difference is that one looks for a sequence which looks like 0 0 0....1 in the last (n-k) position. The number of 0's is equal to (n-k-r-1) from which r can be found. Also the presumed burst error can be corrected.

The same procedure can be applied to Type I channels. Since additive and burst errors do not occur simultaneously one must identify which error is present. If a slip error has occurred, then one can use the same decoding procedure as for the noiseless case. This differentiation can be done by using

$$C(x) = 1 + x^{s+1} + x^{2(s+1)} + \ldots . x\, x^{[e/2]\,(s+1)}$$

$$+ (e - 2[e/2])\, x^{n-1} \qquad (2.3.9)$$

where s is the maximum slip

and e is the number of additive errors the cyclic code can correct.

The first step in the decoding procedure is to add C(x) to the received word and compute the syndrome using g(x). If V(x) is a code word of the form I(x)g(x) it is transmitted as V(x) + C(x). If an additive error E(x) of weight smaller or equal to e occurs the received word will be

$$R(x) = V(x) + E(x) + C(x)$$

Removing C(x) and taking modulo g(x) we obtain

$$(R(x) + C(x)) \text{ modulo } g(x) = E(x), \text{ mod } g(x)$$

where E(x) has a weight at most e.

On the other hand, if a slip $i \leq s$ occurs then the received word after removing C(x) will be

$$R(x) + C(x) = x^{i}V(x) + C(x)\,(1 + x^{i}) + U_{s}(x)$$

The syndrome is $(C(x)\ (1 + x^i) + U_s(x))$, mod $g(x)$. From (2.3.9) it can be seen that $C(x)$ will always have $\underline{e}$ or $\underline{e+1}$ terms depending on if $\underline{e}$ is odd or even. This means that $C(x)\ (1 + x^i) + U_s(x)$ will have at least a weight of $\underline{e+1}$ and at most a weight of $\underline{e+s+1}$. Since the code is $\underline{e}$ error-correcting the decoder will detect errors greater than $\underline{e}$ and less than $(n-k)$, and hence assume that a slip error has occurred. If additive errors did occur the decoder determines the errors using some error-correcting procedure. On the other hand, if a slip error is present the decoder adds $\underline{C(x)}$ again to the word giving

$$R(x) = x^i V(s) + x^i C(x) + U_s(x)$$

As for the noiseless case the product $R(x)h(x)$ is

$$R(x)h(x), \text{ mod } 1 + x^n =$$

$$(x^i C(x) + U_s(x))\ h(x), \text{ mod } 1 + x^n \qquad (2.3.10)$$

where $$C(x) = 1 + x^{s+1} + x^{2(s+1)} + \ldots + x^{[e/2](s+1)} + (e - 2(e/2))\ x^{n-1} \qquad (2.3.11)$$

When e is odd the degree of $C(x)$ is $n-1$ and when e is even the degree of $C(x)$ is $[e/2]\ (s+1)$. When a left slip occurs of magnitude $s_1 \leq s$, the degree of $x^s C(x)$ is

$[e/2]\,(s+1) + s_1$ and hence the degree of (2.3.10) is $[e/2]\,(s+1) + s_1 + k$ which would look like the following sequence (α is a "0" or "1")

$$\underset{\alpha\ \alpha\ \alpha\ldots\alpha}{x^0}\ \underset{1}{x^{[e/2](s+1)+s+k}}\ \underset{0\ldots0\ 0\ 0}{x^{n-1}}$$

characterized by a "1" in the $\{[e/2](s+1)+s+k+1\}$ position followed by $(n-1)$ zeros where $l = [(e/2)](s+1) + s + k + 1$.

Similarly for a right slip of magnitude r we have that

$$R(x) = x^{n-r}I(x)g(x) + x^{n-r}C(x) + x^{n-r}V_r(x)$$

If we analyse the product $R(x)h(s)$, $1+x^n$, the term $\underline{x^{n-r}V_r(x)}$ will affect at most the terms in the first k positions and the last r positions. The term $\underline{x^{n-r}(e-2[e/2])x^{n-1}}$ or $\underline{x^{n-r-1}(e-2[e/2])}$ will affect terms in the first k positions and terms in the last r positions. We are therefore interested in the product

$$x^{-r}\left\{1 + x^{s+1} + x^{2(s+1)} +\ldots+ x^{[e/2](s+1)}\right\} h(x),\ \text{mod}\ 1 + x^n$$

It can be seen that the degree of this expression is $[e/2]\,(s+1)+k-r$. Since only the last r terms are affected, a 1 in the $[e/2](s+1)+k-r+1$ position is present. It should also

be noted that for a right slip a 1 does exist in the position n-r+1 or n-r depending on if e is even or odd. If the [e/2](s+1) +k+s+1 position denoting a left slip and the [e/2](s+1)+k-r+1 position denoting a right slip never coincides with positions n-r+1 or n-r then slip errors can be detected and corrected. The bound on the slip error is

$$s \leq \left[\frac{n-k-1-e + [e/2]}{2 + [e/2]}\right]$$

or $[e/2]s + 2s \leq n-k-1-e + [e/2]$

or $[e/2]s - [e/2] + e + s + k \leq n-s-1$

or $[e/2](s+1) + k + s \leq n-s-1$ when e is even

and $[e/2](s+1) + k + s \leq n-s-2$ when e is odd

Since $s_1 \leq s$ we have that

$$\begin{aligned}[e/2](s+1) + k + s_1 &\leq [e/2](s+1) + k + s \\ &\leq n-s-1 \\ &\leq n-r \text{ when e is even}\end{aligned}$$

and

$$\begin{aligned}[e/2](s+1) + k + s_1 &\leq [e/2](s+1) + k + s \\ &\leq n-r-2 \\ &\leq n-r \text{ when e is odd}\end{aligned}$$

For the case of a right slip the condition is always satisfied since $-s \leq r < 0$. This means that for some slip $0 < |s_1| \leq s$, s_1 can be determined by finding the highest non-zero term x^{α} after the first $\underline{k}$ bits and before the last s+1 positions. This non-zero term x^{α} where $\alpha = [e/2](s+1) + k + s_1$ determines the slip $s_1 = \alpha - [e/2](s+1) - k$.

Example: Consider the (15,7,2) BCH code and assume e=2. The bound on s is

$$s \leq \frac{15-7-1-2+1}{2+1} \pm \frac{6}{3} = 2$$

and $c(x) = 1 + x^3$. It can be shown that

$g(x) = 1 + x^4 + x^6 + x^7 + x^8$

and $h(x) = 1 + x^4 + x^6 + x^7$

Suppose we transmit

$$V_1(x) = g(x) + C(x) = x^2 + x^4 + x^6 + x^7 + x^8$$

$$V_2(x) = g(x) + C(x) = 1 + x + x^2 + x^5 + x^7 + x^8 + x^9$$

end to end:

1 1 1 0 0 1 0 1 1 1 0 0 0 [0 0, 0 0 1 0 1 0 1 1 1
0 0 0 0] 0 0,

Suppose we receive the sequence in bracket where

$$R(x) = x^4 + x^6 + x^8 + x^9 + x^{10}$$

Adding C(x) to R(x) we obtain

$$R(x) + C(x) = 1 + x^2 + x^4 + x^6 + x^8 + x^9 + x^{10}$$

The error syndrome (R(x) + C(x)) mod g(x) is $1 + x^2 + x^3 + x^5$ which can be shown not to be a syndrome defining an error E(x) of weight 2. This means that a slip error has occurred. We add C(x) to obtain R(x) again. The product R(x) h(x) is

$$R(x)\ h(x),\ \text{mod}\ 1 + x^{15} = x^2 + x^5 + x^6 + x^8 + x^{11} + x^{12}$$

or $0\ 0\ 1\ 0\ 0\ 1\ 1\ 0\ 1\ 0\ 0\ 1\ 1\ 0\ 0$

From the above sequence the high power term is x^{12} in the interval. Hence

$$s_1 = 11 - [e/2](s+1) - k$$

or $$s_1 = 12 - 10 = 2$$

Hence, the magnitude of the slip is 2. Shifting R(x) two places to the right and removing C(x) gives $1 + x^4 + x^6 + x^7 + x^8 = g(x)$ which is the correct code word.

Rate of Coset Code in Type 1 Channels

The rate depends on the number of additive errors e to be corrected from which the maximum slip s can be determined. Table 2.3.1 shows the rates for different values of n, e, and s.

Type II channels

In Type II channels, the additive and slip errors occur simultaneously. Because of this and due to the nature of a coset code, the decoder cannot differentiate immediately between an additive and a slip error. A suitable choice for C(x) is

$$C(x) = 1 + x^{2s+1} + x^{2(2s+1)} + \ldots + x^{e(2s+1)} + V^{n-1} \quad (2.3.11)$$

where e is the maximum number of additive errors

and s is the maximum slip.

Suppose V(x) + C(x) is transmitted where V(x) is a code word. If a slip of magnitude is occurs, the received word after adding C(x) is

$$R(x) + C(x) = x^{i}V(x) + C(x)\ (1 + x^{i}) + U_{i}(x) + E(x) \quad (2.3.12)$$

Now (2.3.11) has always e+1 terms, E(x) has at most e terms, and $U_i(x)$ has at most s terms. Therefore, the maximum weight of $C(x)(1 + x^{i}) + U_i(x) + E(x)$ is 2e + s + 1. This means that the term can be reconstructed using a t = 2e + s + 1 error-correcting code. From the reconstructed term the slip $i \leq s$ can be determined. This will not be shown here since it is felt that the rate of this technique (because it must correct at least 2e + s + 1 errors) decreases very rapidly as the slip s increases. This can be seen in Table 2.3.2 where the rate is calculated for different values of n, e, and s. It can therefore be concluded that the coset code technique is inefficient in Type II channels.

TABLE 2.3.2

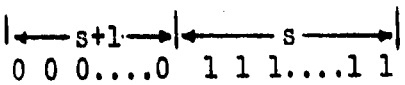
s	e	Rate Coset
n=31		
1	1	.355
1	2	.194
1	3	-
2	1	.355
2	2	.194
2	3	-
3	1	.194
3	2	-
n=127		
1	1	.780
1	2	.669
1	3	.559
1	4	.504
1	5	.394
2	1	.724
2	2	.614
2	3	.559
2	4	.449
2	5	.394
3	1	.669
3	2	.559
3	3	.504
3	4	.394
3	5	.339

2.4(a) Mandelbaum Technique

Among all the present synchronization error-correcting techniques, the Mandelbaum's technique is the one that achieves the highest rate (for a certain range of slip). The encoding and decoding procedures are very simple and straightforward.

Encoding Technique

Suppose we consider the case where the magnitude of the right slip is equal to the magnitude of the left slip and suppose the slip is equal to s. The encoding technique consists in generating a subset of the parent code such that all code words in that subset start with the following bit pattern

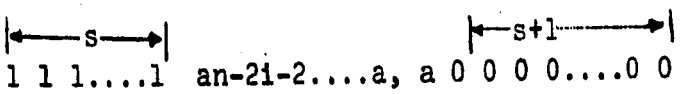


The prefix consists of (s+1) 0's and s 1's. For example, suppose we have two code words A(x) and B(x) transmitted end to end; we would have the sequence

0 0....0 1 1 1....1 an-2i-2....a 0, 0 0....0 1 1 1....1 bn-2i-2....b 0

Considering the fact that the codes are cyclic, each code

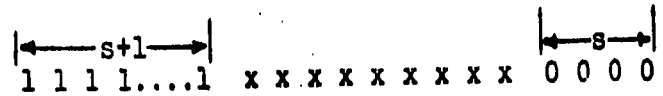
word can be redefined as



(2.4.1)

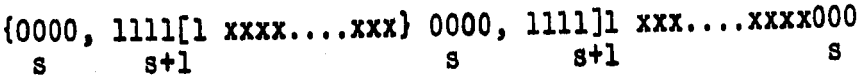
Therefore a code word can be identified as being at the point between s+1 zeros on the left and s ones on the right.

It was said [14] that the first and last code words in the sequence will not have the appearance of (2.4.1) and are assumed to be in synchronism. We can remove this restriction by modifying Mandelbaum's technique slightly. It will also simplify the decoding technique. We modify the Mandelbaum technique by assuming that every code word starts with s+1 ones and terminate with s zeros. A code word will look like



where "x" is "1" or "0".

To show that we can detect and correct a slip to the right or to the left we consider three code words transmitted end to end



where "x" again denotes a ("0" or "1").

Case 1

A right slip of magnitude s has occurred and the received sequence in brackets is

$$1\ x\ x\ x....x\ x\ x\ \underset{s}{0\ 0\ 0\ 0}\ \underset{s}{1\ 1\ 1\ 1} \qquad (2.4.2)$$

Case 2

A left slip of magnitude s has occurred and the received sequence in brackets is

$$0\ 0\ 0\ 0\ \ 1\ 1\ 1\ 1\ 1\ \ x\ x\ x....x\ x\ x \qquad (2.4.3)$$

Case 3

No slip has occurred. The received sequence is

$$1\ 1\ 1\ 1\ 1\ \ x\ x\ x\ x....x\ x\ x\ \ 0\ 0\ 0 \qquad (2.4.4)$$

It can be seen that a right slip can be identified simply by noting that the received sequence starts with a one and terminates with a one. This is also true for all right slips of magnitude smaller than s. A left slip of magnitude s can be identified by noting that the sequence (2.4.3) starts with a zero. This is also true for all left slip smaller than s. Finally a "no slip" condition can be identified by noting that the sequence (4) starts with a one and terminates with a zero. Therefore, the decoding

procedure is very simple, and also the first and last code words do not necessarily have to be in synchronism.

The received sequence which contains a slip error (right or left) is again a code word and any additive errors can be corrected prior to checking for a synchronization error.

Given an (n,k,t) cyclic binary code, the rate obtained using the Mandelbaum technique is

$$\frac{k-2s-1}{n}$$

Figure 2.4.1 shows the block diagram of a possible decoding scheme.

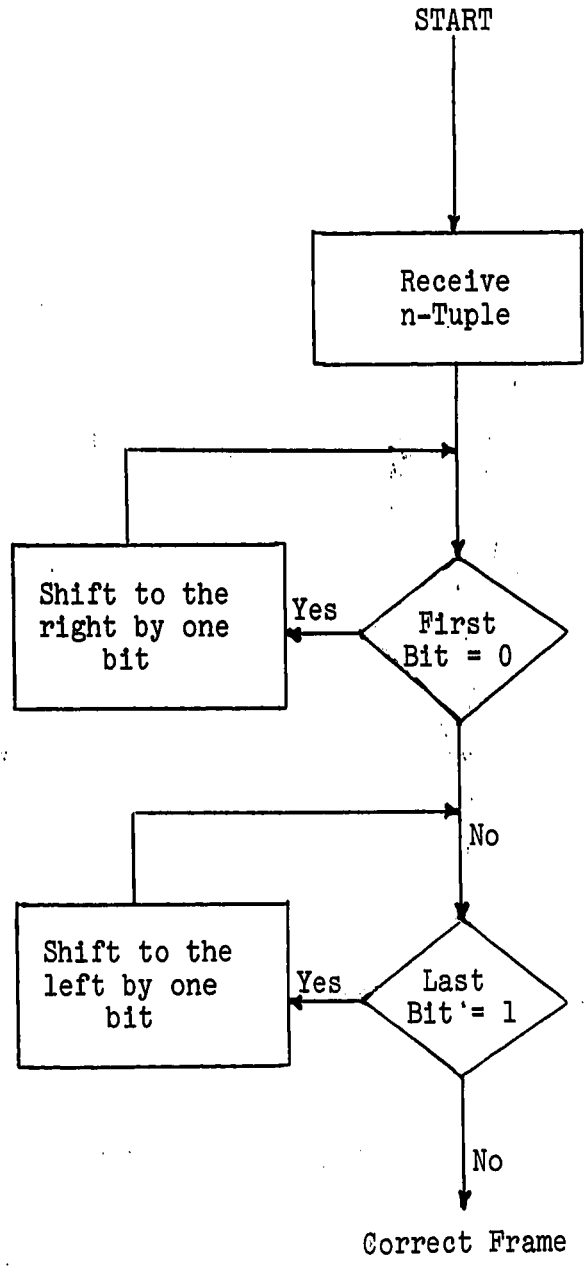


Fig. 2.4.1 Possible Decoding Scheme

2.4(b) Hybrid Synchronization Codes

It was shown in Sections 2.1 and 2.2 that some synchronization error-correcting capability can be incorporated in a code by adding some redundant bits before transmission and by adding additional bits in the information stream. It was also shown that the exponent of the auxilliary polynomial used gave the range of slip which can be corrected. Mandelbaum[13] has proposed a hybrid technique which can increase the slip connecting capability of the BCW and the subset code techniques.

It was shown in (Chapter II, Section 1, 2) that to correct the slip error one simply finds

$$x^s \bmod H_1(x)$$

where s is the slip.

Assuming that the exponent of the auxilliary polynomial is u then

$$(x^{s+mu}) \bmod H(x) = x^s \bmod H(x)$$

Since $x^s[x^{mu} - 1] \bmod H(x) = 0$ this is true since H(x) does not divide x^s but divides $(x^{mu}-1)$. This means that if

we choose a constant pattern of the form $0^z\ 1^z$ where each bit is separated by u-1 information bits. then the slip error-correcting capability of the code will be increased. For example, assume that there exists the following pattern within the information digits of a code word

, 0 xxxxxx 0 xxxxxx 0 xxxxxx 1 xxxxxx 1 xxxxxx 1 xxxxxx,
↑

and assume that a pointer indicates the position of the first "1" (before transmission). Here we assume that u=7 (hence each pattern is separated by u-1=6 information bits). We can form the following pattern of constants

		PATTERN		
LEFT	NO SLIP	0 0 0	1	1 1 x x
	P<u	x x 0 0 0	1	1 1 x x
	u<P≤2u	x x x 0 0	0	1 1 1 x
	2u<P≤3u	x x x x 0	0	0 1 1 1
RIGHT	P<u	x x 0 0 0	1	1 1 x x
	u<P≤2u	x 0 0 0 1	1	1 x x x
	2u<P≤3u	0 0 0 1 1	1	x x x x

For example, if $s=3u-4 = 17$ it can be seen that

$$x^{17} \bmod H(x) = x^3 \bmod H(x)$$

from which we would obtain the pattern

$$x\ x\ x\ x\ 0\ 0\ 0\ 1\ 1\ 1$$
$$\uparrow$$

which immediately indicates that the sequence should be shifted to the right by $2u = 14$ bits to obtain the correct sequence. The following theorem is due to Mandelbaum.

Theorem

An (n,k) cyclic code can be converted to an $(n,k-R_s)$ subset code or an $(m+2s, k-Rs)$ extended cyclic code if there exists a polynomial having degree m and exponent u that divides $h(x)$. This code will correct a slippage of s or less digits where

$$s = uz-1$$
$$R_s = m + 2z$$

for any positive integer z.

The rate of the hybrid technique will be greater only if

$$k-m-2z > k-m^1$$

where $m^1 \geq m$

or $$m+2z < m^1$$

or $$z < (m^1-m)/2$$

This means that the hybrid technique can be used only in cases where the length n is large and the factors of $1 + x^n$ are far apart in value. For example, if $n = 127$, we have that $m = m^1 = 7$. Hence z must be smaller than $m^1-m/2 = 0$ showing that the hybrid technique cannot be used or gives the same rate as the BCW or the subset code technique.

It can also be said that the hybrid technique cannot be used for codes with prime lengths. This is obvious since all the factors of $h(x)$, where $h(x) = 1 + x^n/g(x)$ have degree m.

2.5 Tong's Shortened Codes

This technique differs from the other known synchronization error-correcting codes in that the length of the code is shortened depending on the maximum value of the slip that the code must correct.

Tong's technique consists in adding a polynomial F(x) to every code word before transmission and transmitting a shortened code. Suppose the length of the shortened code is n and the length of the parent code is equal to n^1. The polynomial F(x) is chosen such that

$$\Big(F(x)\Big) \bmod g(x) = (x^n) \bmod g(x) \qquad (2.5.1)$$

where g(x) is the generator polynomial.

From (2.5.1) we see that adding F(x) to the code word is the same as if we added the error-syndrome of an error equal to x^n. Suppose $V_2(x) = g(x)I_2(x) + F(x)$ is a transmitted word of length n, and a slip of magnitude s occurs. The received word R(x) is (from Chapter 1, Section 2 - Case 3)

$$R(x) = x^s\Big[g(x)I_2(x) + F(x)\Big] + A(x) + x^nB(x) \qquad (2.5.2)$$

where A(x) and B(x) are polynomials of degree $\leq s-1$.

If we add F(x) to (2.5.2) and calculate the syndrome with g(x) we obtain

$$s(x) = \left[R(x) + F(x)\right], \text{ mod } g(x) = F(x)\,(1 + x^s), \text{ mod } g(x)$$

$$+ \left[A(x) + x^n B(x)\right], \text{ mod } g(x) \qquad (2.5.3)$$

Since we choose F(x) such that $F(x), \text{ mod } g(x) = x^n, \text{ mod } g(x)$ we have that

$$F(x)\,(1 + x^s), \text{ mod } g(x) = (x^{s+n} + x^n), \text{ mod } g(x) \qquad (2.5.4)$$

Substituting (2.5.4) in (2.5.2) we see that the maximum weight that $\left\{(x^{s+n} + x^n), \text{ mod } g(x) + A(x) + x^n B(x)\right\}$ can have is 2s+1. If the code word can correct at least 2s+1 errors or more, then errors in positions from 1 to s, from n+1 to n+s and at the position n+s+1 will be corrected or shown to occur. Since the bit in the (n+s+1) position was never transmitted it identifies a slip error. Similarly for a right slip of magnitude r the received word will be

$$R(x) = x^{-r}[g(x)\,I(x) + F(x)] + x^{n-r}C(x) + x^{n^1-r}D(x) \qquad (2.5.5)$$

Adding F(x) and taking modulo g(x) we obtain

$$\Big(R(x) + F(x)\Big) \bmod g(x) = \Big\{F(x)\ (1+x^{-r}) + x^{n-r}C(x) + x^{n^1-r}D(x)\Big\} \bmod g(x)$$

where C(x) and D(x) are of degree $\leq$s-1.

Since $(F(x)) \bmod g(x) = (x^n) \bmod g(x)$ we have that

$$F(x)\ (1 + x^{-r}) = (x^n + x^{-r}) \bmod g(x) \qquad (2.5.6)$$

This means that if the decoder can correct up to 2r+1 errors it will show that errors occurred at positions from n-r+1 to n, from n^1-r+1 to n^1, and at position (n+1). Since only n bits were transmitted it identifies the occurrence of a slip error. Assuming that r=s we must have that the position of the error x^{n+s} indicating a left slip should be such that

$$n + s < n + s + 1 < n^1 - s + 1$$

and the position of the error x^n indicating a right slip should be such that

$$n + s < n < n^1 - s + 1$$

from which we obtain the following condition

$$n^1 - n \geq 2s + 1 \qquad (2.5.7)$$

This means that to correct a slip error of magnitude s the code must correct at least (2s+1) additive errors and must be shortened by at least (2s+1) bits. The rules for a decoding are shown in the following steps:

Step 1 If the decoder indicates an error in location n+1 but no errors in positions greater than n+1 and smaller or equal to n^1-r+1 then the decoder concludes that a right slip has occurred. One can correct for slippage by simply shifting to the left until synchronization is restored.

Step 2 If the decoder indicates an error in position n+B+1 where $1 \leq B \leq s$ then the decoder concludes that a left slip has occurred and corrects it accordingly.

Rate of Shortened Codes

Suppose we choose an (n,k) binary cyclic code with a t-additive error-correcting capability. Assume that the maximum slip (right or left) that the decoder must

correct is s. Also assume that t = 2s+1+e where e is the number of additive errors that the code can correct in the presence of a slip. The length of the shortened code will be

$$n = n^1 - 2s - 1$$

Therefore the rate is

$$\frac{k-2s-1}{n^1-2s-1}$$

It is clear that this technique can be used only when the slip is small. As the slip increases the rate decreases drastically as shown in Table 2.5.1.

Conclusions

It was shown that the Tong's shortened codes can be used only when the chosen (n,k) code has an error-correcting capability of at least 2s+1. It was also shown that the rate decreases rapidly as the slip increases.

The main reason why the rate decreases so rapidly is that the code must correct up to 2s bits only because the received word is not cyclic after a slip has occurred. We can modify this technique by assuming that all shortened code words start and terminate with at least s zeros.

TABLE 2.5.1

s	e	Rate	e_0
n=31			
1	1	.286	5
1	2	.286	5
1	3	.107	7
2	1	-	
2	2	-	
n=127			
1	1	.774	4
1	2	.718	5
1	3	.661	6
1	4	.605	7
1	5	.548	9
2	1	.656	6
2	2	.598	7
2	3	.541	9
2	4	.541	9
2	5	.484	10
3	1	.533	9
3	2	.533	9
4	1	.466	10
4	2	.407	11
5	1	.336	13
5	2	.336	13
6	1	.263	14
6	2	.202	15

This will guarantee that the received word which has suffered slip error (right or left) will again be cyclic.

We are faced with one major problem. That is, how can we guarantee that all transmitted code words start and finish with at least s zeros, given that F(x) is added before transmission. To solve this problem we propose the following. To begin with, choose all code words of length n^1 (length of parent code) which terminate with 4s+1 zeros. Delete the last 2s+1 zeros and then add the vector F(x) to the code word. Before transmission shift the sequence s bits to the left. This will guarantee that the transmitted word starts and ends with s zeros. For example, consider the (15,11,1) BCH code and let the maximum slip s = 1. Suppose V(x) belongs to the parent code and terminates with 4s+1 = 5 zeros. We have, therefore, that V(x) can be represented by the sequence

$$a_0\ a_1\ a_2\ a_3\ a_4\ a_5\ a_6\ a_7\ a_8\ a_9\ 00000$$

Adding F(x) we obtain

$$V(x) + F(x) = C_0\ C_1\ C_2\ C_3\ a_4\ a_5\ a_6\ a_7\ a_8\ a_9\ 00000$$

where C_0, C_1, C_2, C_3 depend on the error syndrome $(F(x)) \bmod g(x)$.

Deleting the last 3 zeros and shifting to the left by 1 bit we obtain

$$x[V(x) + F(x)] = 0\ C_0\ C_1\ C_2\ C_3\ a_4\ a_5\ a_6\ a_7\ a_8\ a_9\ 0$$

$$\text{or } V^1(x) + F^1(x) = 0\ b_0\ b_1\ b_2\ b_3\ b_4\ b_5\ b_6\ b_7\ b_8\ b_9\ 0$$

where $V^1(x) = x\ V(x)$

$F^1(x) = x\ F(x)$

Generally for a slip s we have

$$x^s[V(x) + F(x)] = \underbrace{0\ 0....0}_{s}\ b_0\ b_1\ b_2\ b_3....b_{n-1}\ \underbrace{0\ 0\ 0....0}_{s}$$

and let

$V^1(x) = x^s V(x)$

$F^1(x) = x^s F(x)$

Now suppose we transmit $V^1(x) + F^1(x)$. Since the words start and terminate with s zeros we have that, for a left slip of magnitude $i \leq s$, the received vector will be

$$R(x) = x^i[V^1(x) + F^1(x)]$$

and for a right slip of magnitude i≤s the received vector will be

$$R(x) = x^{-i}[V^1(x) + F^1(x)]$$

and lastly for no slip errors the received vector will be

$$R(x) = [V^1(x) + F^1(x)]$$

Substituting $V^1(x) = x^s V(x)$

and $F^1(x) = x^s F^1(x)$

we obtain

(1) <u>For Left Slip</u> $R(x) = x^{s+i}V(x) + x^{s+i}F(x)$

(2) <u>For Right Slip</u> $R(x) = x^{s-i}V(x) + x^{s-i}F(x)$

(3) <u>For No Slip</u> $R(x) = x^s V(x) + x^s F(x)$

The error syndromes of the above equations are

(1) <u>For Left Slip</u> $(x^{s+i}F(x)) \bmod g(x)$

(2) <u>For Right Slip</u> $(x^{s-1}F(x)) \bmod g(x)$

(3) <u>For No Slip</u> $(x^s F(x)) \bmod g(x)$

Given $F(x) = [x^n] \bmod g(x)$, we obtain the following error syndromes

(1) For Left Slip $[x^{n+s+i}] \bmod g(x)$

(2) For Right Slip $[x^{n+s-1}] \bmod g(x)$

(3) For No Slip $[x^{n+s}] \bmod g(x)$

Therefore, if the decoder finds an error in position (n+s+i+1) we conclude that a left slip of magnitude i has occurred and correct it by shifting to the right by i bits.

If the decoder finds an error in position n+s-i we conclude that a right slip of magnitude i has cocurred and we correct it accordingly.

If the decoder finds an error in position n+s+1 we conclude that no slip error has occurred.

After shifting right or left depending on the slip error, we add F(x) to recover the original transmitted code word.

Example

Consider the (15,11,1) BCH cyclic code and assume that the maximum slip is s=1. The generator polynomial is

$$g(x) = 1 + x + x^4$$

Suppose we want to transmit the code word s(x) = 1 1 0 0 1 0 0 0 0 0 0 0 0 0 0. Since s=1 we have that the shortened code will have a length n=15-3=12. Since the code word s(x) terminates with at least 4s+1=5 zeros, it is a valid code word. We choose F(x) such that

$$\begin{aligned} F(x) &= (x^n) \bmod g(x) \\ &= (x^{12}) \bmod g(x) \\ &= 1\ 1\ 1\ 1\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0 \end{aligned}$$

Adding F(x) to g(x) we have

$$F(x) + g(x) = 0\ 0\ 1\ 1\ 1\ 0\ 0\ 0\ 0\ 0\ 0\ 0$$

Shifting to the left by s=1 bit we obtain

$$0\ 0\ 0\ 1\ 1\ 1\ 0\ 0\ 0\ 0\ 0\ 0$$

Now suppose the transmitted sequences are

$$[0,\ 0\ 0\ 0\ 1\ 1\ 1\ 0\ 0\ 0\ 0\ 0]0,\ 0$$

and a left slip occurs. The received sequenc will be

$$0\ 0\ 0\ 0\ 1\ 1\ 1\ 0\ 0\ 0\ 0\ 0$$

or in polynomial representation

$$x^4 + x^5 + x^6$$

The error syndrome $(x^4 + x^5 + x^6)$ mod $g(x)$ is equal to $1 + x^3$ which is equal to the syndrome of an error occurring in the first position.

Hence, an error has been found in the position $n+s+i+1 = 15$ from which we conclude that a left slip of magnitude $i=1$ has occurred. Shifting by 1 bit we obtain

0 0 0 1 1 1 0 0 0 0 0 0

Shifting again by $s=1$ bit we obtain

0 0 1 1 1 0 0 0 0 0 0 0

Adding F(x) = 1 1 1 1 0 0 0 0 0 0 0 0

we obtain 1 1 0 0 1 0 0 0 0 0 0 0

or $1 + x + x^4$ which was the transmitted code word.

Similarly for a right slip of magnitude $s=1$, the received sequence will be

0 0 1 1 1 0 0 0 0 0 0 0

or in the polynomial representation

$$x^2 + x^3 + x^4$$

The error syndrome $(x^2 + x^3 + x^4)$ mod g(x) is equal to $1 + x + x^2 + x^3$ which is the syndrome of an error occurring in the 13^{th} position. Hence an error has been found in position n + s - i + 1 = 13 from which we conclude that a right slip of magnitude i=1 has occurred. Since the word has been initially shifted by s=1 bits we need not make any corrective action in the case (in general we would simply have to shift the sequence to the right by s-i bits). Adding F(x) we obtain

1 1 0 0 1 0 0 0 0 0 0

or $1 + x + x^4$ which was the transmitted code word.

Finally for the case where no synchronization error occurs the received sequence is

0 0 0 1 1 1 0 0 0 0 0

or $x^3 + x^4 + x^5$

The error syndrome $(x^3 + x^4 + x^5)$ mod g(x) is equal to $1 + x^2 + x^3$ which indicates an error in the 14^{th} position

from which we conclude that no slip has occurred. We shift the sequence to the right by s=1 bit and add F(x) to obtain

1 1 0 0 1 0 0 0 0 0 0 0

or $1 + x + x^4$ which is the desired code word.

The rate is equal to k-us-1/n^1-2s-1 = .500. On the other hand, if we had used Tong's technique we would have needed the (15,5,3) BCH code. The rate would have been .167. Therefore, it can be seen that the modified Tong's shortened code technique can achieve better rates than the Tong technique.

As another example, suppose we choose the (127,85,6) BCH code. Assume that the maximum slip is 2 and that e=1, n=127-5 = 122. If we used the Tong shortened code technique, the rate will be

$$\frac{k-2s-1}{n-2s-1} = \frac{85-5}{127-5} = .656$$

and since nearly all the error-correcting capability is used to correct slip, the number of additive errors e that the code can correct is only 1.

On the other hand, if we used the modified shortened code technique we can do with only a 2 error-correcting code and not a 6 error-correcting code. We could choose the (127,113,2) BCH code. The rate would be

$$\frac{k-4s-1}{n-2s-1} = \frac{113-9}{122} = \frac{104}{122} = .852$$

If we compare this result to the BCW and Mandelbaum techniques we obtain

n=127	Modified	BCW	Mandelbaum
s=2	Tong's Shortened Code		
e=1	.852	.856	.900

It can be seen that the rates are about equal in value.

2.6 Shiva and Seguin Results

In this section we will analyse the Shiva and Seguin results. We will first discuss the case where the length of the code is shortened before transmission and then analyse a slight modification where the length is unaltered.

Case 1 - Code is Shortened

In this first case we start with an (n,k) binary cyclic code denoted by V that has the following error-correcting capabilities: (assume s is the maximum slip)

(1) corrects a burst of length s or less in the first 2s positions;

(2) corrects a burst of length s or less always occurring in the last 2s positions;

(3) corrects errors occurring anywhere belonging to the set of errors that the parent code (n,k) can correct.

We shorten the code words by 2s and assume they all start with a 1. For example, assume that the length of the parent code n is equal to 15 and the maximum slip is s=2. We will have that all transmitted code words look like

1 x x x x x x x x x x

where "x" can be 0 or 1, and the length $n_1 = n-2s = 11$.

Suppose three code words are transmitted end to end. We obtain

z z [z z, 1 x{x x x x x x x]x x, 1 y}y y y y y y y y y

where "y" + "z" can be 0 or 1.

For a left slip (sequence in brackets) we obtain

z z 1 x x x x x x x (2.6.1)

and for a right slip of magnitude s=2 we obtain the sequence in [ ] as

x x x x x x x x x x 1 y (2.6.2)

We know that when a left slip occurs we can expect a burst error of length $\leq s$ in the first s positions and another burst of length $\leq s$ starting in position n_1+1 where $n_1=n=2s$. Similarly for a right slip we can expect two burst errors in positions from n_1-s+1 to n.

To correct the slip error we must be able to locate the start of the code word denoted by the "1".

One way is to guarantee that this "1" is always in the first 2s+1 positions. We can do this by shifting (2.6.1) and (2.6.2) by s bits to the left. We obtain for the case of left and right slip respectively

$$x\ x\ z\ z\ 1\ x\ x\ x\ x\ x\ x \qquad (2.6.3)$$

and

$$1\ y\ x\ x\ x\ x\ x\ x\ x\ x\ x \qquad (2.6.4)$$

where x,y,z are again 0 or 1.

This means that for a left slip we can now expect burst errors of length $\leq$s in the first 2s positions and in the last 2s positions. Similarly for a right slip we can expect burst errors in the first 2s positions and last 2s positions.

Now since the parent code can correct these errors, (2.6.3) and (2.6.4) will look like

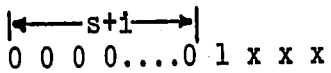
$$0\ 0\ 0\ 0\ 1\ x\ x\ x\ x\ x\ x\ x\ x\ x$$

and

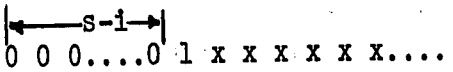
$$1\ x\ x\ x\ x\ x\ x\ x\ x\ x\ 0\ 0\ 0\ 0$$

More generally, if we assume that a slip i$\leq$s has occurred, the first 2s+1 positions of any received code word, after

shifting to the left by s bits, will be for the cases of left and right slip:



and



Therefore, to recover the correct code word we simply shift to the right until the first bit is a "1".

To summarize the decoding procedures, we simply shift the received sequence (shortened sequence) by s bits to the left, correct for any burst and additive errors using the error-correcting capability of the parent code and then shift to the right until the first bit is a "1".

If we treat all burst errors resulting from a slip error as additive errors then the parent code must correct $t = 2s + e$ errors where s is the maximum slip and e is the number of additive errors that can occur during transmission.

The rate of this shortened code technique is equal to

$$\frac{(k-2s-1)}{(n-2s)}$$

Case 2 - Modified Technique

In this case we assume again that all code words start with a "1" and terminate with at least 2s zeros, and that the parent code can correct a burst of length s occurring in the first 2s positions. In this case we do not shorten the code word. The decoding procedure is along the same lines as in Case 1. The only difference is that we now expect at most only one burst of length $\leq s$ in the first position resulting from a slip error and not two burst-errors. This means that the parent code need only correct (if we treat the burst error as additive errors) t=s+e errors.

The rate of this modified technique is

$$\left[\frac{k-2s-1}{n}\right]$$

Conclusion

In Section 2.5 of this chapter we analysed Tong's shortened code technique where the rate was given as

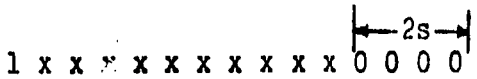
$$\frac{k-2s-1}{n-2s-1}$$

assuming the parent code can correct at least t = 2s+1+e errors. The rate obtained using the Shiva and Sequin result is

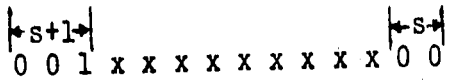
$$\frac{k-2s-1}{n-2s}$$

but the parent code must correct at least t = 2s+e errors. Therefore, the Shiva and Seguin technique has a better rate than the Tong technique. It should be noted that both techniques have low rates when the slip is large.

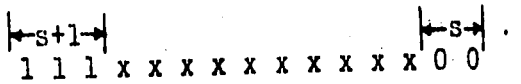
The modified technique (Case 2) gives a better rate than the shortened technique (Case 1). This is because the parent code must correct only s additional errors and not 2s additional errors as in Case 1. Nevertheless, if we compare this modified technique with that of Mandelbaum it can be seen that they are very similar. In the modified technique we would simply have to bring s zeros to the front of the code words to obtain the same results as in Mandelbaum's technique. For example, if n=15 and s=2, a code word would look like



If we bring s zeros to the front we have



In Mandelbaum's technique the code word would look like



.

It can be seen that the two sequences are very similar.

CHAPTER III

A FURTHER DISCUSSION OF BCW TECHNIQUE

In this chapter we present a different technique for the correction of sysnchronization errors in the presence of additive errors. This technique is similar to the BCW technique in that we increase the length of the code word before transmission. This procedure guarantees that the received word will be cyclic in nature even in the presence of slip errors. The main difference between the two techniques is the form of the auxiliary polynomial (i.e. P(x) in BCW technique). Also unlike the BCW technique no coset is added to the code word before transmission. To begin with we will analyse the auxiliary polynomial which will be denoted by C(x). Unlike the BCW technique where the auxiliary polynomial was conveniently a factor of the parity check polynomial H(x) and therefore divided $1+x^n$ we assume in this discussion that C(x) and H(x) are relatively prime. This can be denoted as

$$(C(x),H(x)) = 1 \qquad (3.1.1)$$

where $H(x) = (1+x^n)/g(x)$

Also instead of adding a coset to every code word before transmission (as in BCW technique) we assume that every code word ends with a "1". This assures us that every transmitted code word has degree (n-1).

Using the above restriction we obtain the following results:

Result 1

Consider an (n,k) cyclic code with generator polynomial g(x). Suppose we consider a subset generated by g(x)C(x) where C(x) and the parity polynomial H(x) (i.e. $H(x) = 1 + x^n/g(x)$) are relatively prime and each code word ends with a '1', we have that a received word will belong to the suset only if it is in synchronization.

Proof

Assume that a left slip of magnitude s has occurred during transmission. The received code word R(x) will look like

$$R(x) = x^s V(x) = A(x)(1+x^n) + B(x) \qquad (3.1.2)$$
$$= x^s g(x)C(x)I_1(x) + A(x)H(x)g(x) + g(x)I_2(x)$$

where $R(x)$ is the received word

$V(x)$ is the transmitted code word

$A(x)$ is a polynomial of degree $\leq s-1$

$B(x)$ is a polynomial of degree $\leq n-1$.

and $I_1(x)$ and $I_2(x)$ are information polynomials of degree k.

Suppose we divide (3.1.2) by $g(x)$ and take modulo $C(x)$ giving,

$$R(x), \text{mod } C(x) = A(x)H(x), \text{mod } C(x) + I_2(x), \text{mod } C(x)$$
$$= x^s C(x) I_1(x), \text{mod } C(x) \qquad (3.1.3)$$

Since $x^s C(x) I_1(x)$ has $C(x)$ as a factor we have that,

$$x^s C(x) I_1(x) \text{ mod } C(x) = 0$$

Since $C(x)$ and $H(x)$ are relatively prime, the expression $A(x)H(x), \text{mod } C(x)$ is different from zero only if the degree of $A(x)$ is smaller than the degree of $C(x)$. If this is true then

$$I_2(x), \text{ mod } C(x) \neq 0$$

which proves Result 1. It can be seen from (3.1.3) that if $A(x)$ is equal to zero then

$$I_2(x), \text{ mod } C(x) = 0$$

which means that the received word contained no synchronization error. Also since each code word ends with a "1", this guarantees that a left slip of magnitude s or less the degree of A(x) will always be less than or equal to s-1.

Result 1 does not hold for right slips since there is no way of predicting the degree of A(x) in the presence of right slip. To resolve this difficulty, we shift the received sequence to the left by s bits where s is said to be the maximum allowable slip error. This means that the received sequence will always be checked for left slip since we have eliminated the case of a right slip.

We now examine what happens when we shift to the left (multiply by x^s) the received sequence.

Case 1: A right slip of magnitude r ($r \leq s$). The received vector will be

$$R(x) = x^{-r} V(x) \qquad (3.1.4)$$

Multiplying by x^s or shifting (3.1.4) to the left by s bits we obtain

$$x^s R(x) = x^{s-r} V(x)$$

or $$x^{s-} V(x) = A(x)\ (1+x^n) + B(x)$$

where A(x) has degree $\leq (s-r)-1$.

Case 2: A left slip of magnitude L($L \leq s$). The received vector will be

$$R(x) = x^L V(x)$$

Similarly if we shift to the left by s bits we get

$$x^s R(x) = x^{s+L} V(x)$$

or $$x^{s+L} V(x) = A(x)\ (1+x^n) + B(x) \qquad (3.1.5)$$

From (3.1.5) we see that the maximum degree of A(x) is (2s-1). From Result 1 we must have that the degree of C(x) be greater than (2s-1). Therefore if

$$(\deg C(x) = 2s) \qquad (3.1.6)$$

we can determine the magnitude of the slip. It should be noted that we do not have to determine the direction since

$$x^s R(x) = x^{s-r} V(x)$$

or $$x^{s-} V(x) = A(x)\,(1+x^n) + B(x)$$

where A(x) has degree $\leq (s-r)-1$.

Case 2: A left slip of magnitude L($L \leq s$). The received vector will be

$$R(x) = x^L V(x)$$

Similarly if we shift to the left by s bits we get

$$x^s R(x) = x^{s+L} V(x)$$

or $$x^{s+L} V(x) = A(x)\,(1+x^n) + B(x) \qquad (3.1.5)$$

From (3.1.5) we see that the maximum degree of A(x) is (2s-1). From Result 1 we must have that the degree of C(x) be greater than (2s-1). Therefore if

$$(\deg C(x) = 2s) \qquad (3.1.6)$$

we can determine the magnitude of the slip. It should be noted that we do not have to determine the direction since

it is always to the left. Of course, if the result from (3.1.3) is equal to zero, then it can be assumed that no slip errors has occurred. In the BCW technique, all additive errors can be corrected prior to checking for slip errors. This is true since the resultant code word, even under slip error, is still a cyclic shift of the transmitted code word.

Rate of this Different Technique

The rate of this different technique is

$$\frac{k-2s-1}{n+2s} \tag{3.1.7}$$

comparatively to BCW's rate which is equal to $\left(\frac{k-m}{m+2s}\right)$ (m is the degree of the auxiliary polynomial). This means that result 1 will yield a better rate as compared to the BCW technique when

$$k-2s-1 > k-m$$

or

$$s < m-1/2 \tag{3.1.8}$$

For example, if we consider the (31,26,1) BCH code, the degree of the auxiliary polynomial is equal to 5. This

means that for

$$s < s-1/2 = 2$$

result 1 yeilds a better rate than the BCW technique.

One disadvantage in using the BCW or the subset code technique lies in the fact that for prime lengths different from those that can be expressed as $n=2^q-1$, which are called maximal lengths, the auxiliary polynomial used often leads to a trivial case. For example, suppose the encoder calls for a code of length equal to 17. It can be shown that since 17 is a prime factor of $2^8-1 = 3x5x17$, the generator and auxiliary polynomial must have order equal to 2^8-1 and have each degree of 8. This will result in the code having only 2 words which is a trivial case. On the other hand, if we use result 1 and if we assume that we want to correct a left or right slip equal to 2, the number of words in our code will be $2^{k-2s} = 2^{9-5} = 16$.

As another example if one wants a code of length 131, we must use a generator and an auxiliary polynomial of degree 60 which leads again to a trivial case. Using result 1 and assuming a maximum slip (right or left) of 2 bits the number of words is equal to $2^{k-2s-1} = 2^{60-5} = 2^{55}$.

It should be made clear that even if we wanted to use the BCW technique for correction of slip errors of very small magnitude, we would still have to use an auxiliary polynomial which leads to some trivial cases. Table 3.1.1 shows the result of a computer program which determined the non-repeated prime factors of 2^i-1 where i was varied from 2 to 65. For example, the distinct prime factors that belong to $2^{59}-1$ are 67, 83, 101, 107, and 457. Because the table would be extremely large we discarded all prime factors greater than 500. From Table 3.1 it can be seen that for many cases one could not use the BCW technique or the Subset technique. The "Max Slip" (column 2) is the maximum slip value for which result 1 is better than the BCW technique. The value m (column 3) is the degree of the auxiliary polynomial for the BCW or subset code.

Leaf 100 omitted in page numbering.

LENGTH	MAX SLIP	
N	s	m
3	0	2
5	1	4
7	1	3
11	4	10
13	5	12
17	3	8
19	8	18
23	5	11
29	13	28
31	2	5
37	17	36
41	9	20
43	6	14
47	11	23
53	25	52
59	28	58
61	28	58
67	29	59
71	17	35
73	4	9
79	19	39
83	29	59
89	5	11
97	23	48
101	29	59
103	25	51
107	29	59
109	17	36
113	13	28
127	3	7
131	29	60
137	29	60
139	29	60
149	29	60
151	7	15
157	25	52
163	29	60
167	29	60
173	29	60
179	29	60
181	27	56
191	29	60
193	29	60

Table 3.1.1(a)

LENGTH n	SLIP s	n m
197	29	60
199	29	60
211	29	60
223	18	37
227	29	60
229	29	60
233	14	29
239	29	60
241	11	24
251	24	50
257	7	16
263	30	61
269	30	61
271	30	61
277	30	61
281	30	61
283	29	60
293	30	61
307	30	61
311	29	60
313	28	57
317	30	61
331	14	30
337	10	21
347	30	61
349	30	61
353	30	61
359	28	57
367	30	61
373	30	61
379	30	61
383	30	61
389	30	61
397	21	44
401	30	61
409	29	60
419	30	61
421	30	61
431	21	43
433	30	61
439	30	61
443	30	61
449	30	61
457	29	59
461	30	61

Table 3.1.1(b)

LENGTH	SLIP	
n	s	m
463	30	61
467	30	61
479	30	61
487	30	61
491	30	61
499	30	61
503	30	61
509	30	61
521	30	62
523	30	62
541	29	60
547	30	62
557	30	62
563	30	62
569	30	62
571	30	62
577	30	62
587	30	62
593	29	59
599	30	62
601	12	25
607	30	62
613	28	58
617	30	62
619	30	62
631	22	45
641	30	62
643	30	61
647	30	62
653	30	62
659	30	62
661	30	62
673	23	48
677	30	62
683	10	22
691	30	62
701	30	62
709	30	61
719	30	62
727	30	62

Table 3.1.1(c)

LENGTH	SLIP	
M	S	M
733	30	62
739	30	62
743	30	62
751	30	62
757	30	62
761	30	62
769	30	62
773	28	58
787	30	62
797	30	62
809	30	62
811	30	62
821	30	62
823	30	62
827	30	62
829	30	62
839	30	62
853	30	62
857	30	62
859	30	61
863	30	61
877	30	62
881	27	55
883	30	62
887	28	58
907	30	62
911	30	62
919	30	62
929	30	62
937	30	62
941	30	62
947	30	62
953	30	62
967	30	62
971	30	62
977	30	62
983	30	62
991	30	62
997	30	62

Table 3.1.1(d)

CHAPTER IV

CONCLUSIONS

In this thesis we have described the existing techniques which can be used for the correction of synchronization error. In Chapter II, the encoding and decoding procedures of these techniques were described and their rates determined. In Chapter III a new technique was introduced. To compare the rate of this new technique to those discussed in Chapter II we use Table 4.1.1. It can be seen that the rate obtained using the Mandelbaum technique will always be greater than the rate obtained in the case of result 1. This is because the extended length $n+2s$ decreases the rate of result 1. On the other hand, if we use result 1 then we can achieve better rates than the BCW technique for a certain range of slip. The techniques in Table 4.1.1 which necessitates an increase in the error-correcting capability simply to correct slip error, achieves lower rates than those which do not. This is shown in Table 4.1.2 which gives the rate for a certain range of slip s and errors e in Type II channels. We assume in Table 4.1.2 that the length is $n=127$. It can be seen that by using result 1 we can achieve better rates than the Coset, Tong, Shiva and Seguin, and Subset techniques. On the other hand, it is

never better than Mandelbaum, but greater than the BCW rate up to a slip of 3.

It was thought at first that using the auxiliary polynomial C(x) in result 1 could in some way increase the error-correcting capability of the code. Special interest was placed upon the possibility of the code to correct burst errors. Unfortunately it was discovered that at present no additional error correcting could be included by choosing a particular auxiliary polynomial C(x). Nevertheless the subject should be analysed and studied thoroughly in the future.

TYPE II CHANNELS

Magnitude of Slip s
Magnitude of Errors e

	Length*	# Errors code must correct	# Information k ***
BCW	n+2s	e	k-m
Subset	n	e+s	k-m
Coset	n	2e+s+1	k
Mandelbaum	n	e	k-2s-1
Hybrid	n+2s	e	k-m-2z**
Tong (shortened)	n-2s-1	2s+e+1	k-2s-1
Tong (mod)	n-2s-1	e	k-4s-1
Shiva & Seguin	n	e+s	k-2s-1
Shiva & Seguin (short)	n-2s	2s+e	k-2s-1
Result 1	n+2s	e	k-2s-1

* Assume n the length of parent code
** z = s+1/p (p is the exponent)
*** n depends on the number of error the parent code must be capable to correct

Table 4.1.1

n 127

s	e	SUBSET $\hat{e}_0$		COSET e_0		MANDEL e_0		BCW e_0		TONG e_0		SHIVA e_0		Res.1 e_0	
1	1	.83								.7					
1	2	.78								.71					
1	3	.724	4	.559	3					.661	6	.756			
1	4	.669	5	.504	4	.756	4	.713	4	.605	7	.701	5	.74	
1	5	.614	6	.394	5	.701	5	.659	5	.548	9	.646	6	.685	5
2	1	.700	3	.724	1	.906	1	.863	1	.656	6	.795	3	.874	1
2	2	.724	4	.614	2	.850	2	.809	2	.598	7	.740	4	.819	2
2	3	.669	5	.559	3	.795	3	.756	3	.541	9	.685	5	.764	3
2	4	.614	6	.449	4	.740	4	.702	4	.541	9	.630	6	.709	4
2	5	.559	7	.394	5	.685	5	.649	5	.484	10	.575	7	.654	5
3	1	.724	4	.669	1	.890	1	.850	1	.533	9	.724	4	.843	1
3	2	.669	5	.559	2	.835	2	.797	2	.533	9	.669	5	.787	2
3	3	.614	6	.504	3	.780	3	.744	3	.475	10	.614	6	.732	3
3	4	.559	7	.394	4	.724	4	.692	4	.417	11	.559	7	.677	4
3	5	.504	9	.339	5	.669	5	.639	5	.358	13	.504	9	.622	5
4	1	.669	5	.614	1	.874	1	.837	1	.466	10	.654	5	.811	1
4	2	.614	6	.559	2	.819	2	.785	2	.407	11	.598	6	.756	2
4	3	.559	7	.449	3	.764	3	.733	3	.347	13	.543	7	.701	3
4	4	.504	9	.394	4	.709	4	.681	4	.347	13	.488	9	.646	4
4	5	.504	9	.283	5	.654	5	.630	5	.288	14	.488	9	.591	5
5	1	.614	6	.559	1	.858	1	.825	1	.336	13	.583	6	.780	1
5	2	.559	7	.504	2	.803	2	.774	2	.336	13	.528	7	.724	2
5	3	.504	9	.394	3	.748	3	.723	3	.276	14	.472	9	.669	3
5	4	.504	9	.339	4	.693	4	.672	4	.216	15	.472	9	.614	4
5	5	.449	10	.228	5	.638	5	.620	5	.155	21	.417	10	.559	5
6	1	.559	7	.559	1	.843	1	.813	1	.263	14	.512	7	.748	1
6	2	.504	9	.449	2	.787	2	.763	2	.202	15	.457	9	.693	2
6	3	.504	9	.394	3	.732	3	.712	3	.140	21	.457	9	.638	3
6	4	.449	10	.283	4	.677	4	.662	4	.140	21	.402	10	.583	4
6	5	.394	11	.228	5	.622	5	.612	5	.140	21	.346	11	.528	5
7	1	.504	9	.504	1	.827	1	.801	1	.125	21	.441			1
7	2	.504	9	.394	2	.772	2	.752	2	.125	21	.441			2
7	3	.449	10	.339	3	.717	3	.702	3	.125	21	.386			3
7	4	.394	11	.228	4	.661	4	.652	4	.125	21	.331			4
7	5	.339	13	.228	5	.606	5	.603	5	.125	21	.276			5
8	1	.504	9	.449	1	.811	1	.790	1	.109	21	.425	9	.685	1
8	2	.449	10	.394	2	.756	2	.741	2	.109	21	.370	10	.630	2
8	3	.394	11	.283	3	.701	3	.692	3	.109	21	.315	11	.575	3
8	4	.339	13	.228	4	.646	4	.643	4	.109	21	.260	13	.52	
8	5	.339	13	.228	5				5	.045	23	.26			

Table 4.1.2

REFERENCES

1. Berlekamp, E.R. Algebraic Coding Theory. McGraw-Hill, New York, 1968.

2. Peterson, W.W. Error-Correcting Codes. M.I.T. Press, Cambridge Mass., 1971.

3. Ash, R.B. Information Theory. Interscience Publishers, New York, 1965.

4. Stiffler, J.J. Theory of Synchronous Communications. Prentice-Hall, Inc., Englewood Cliffs, New Jersey, 1971.

5. Tong, S.Y. Synchronization Recovery Techniques for Binary Cyclic Codes. Bell System Technical Journal, Vol. 45, pp. 561-596, 1966.

6. Tavares, S.E. and M. Fukada. Matrix Approach to Synchronization Recovery for Binary Cyclic Codes. IEEE Trans. on Inf. Theo., Vol. 15, pp. 93-101, 1969.

7. Tavares, S.E. A Study of Synchronization Techniques for Binary Cyclic Codes. Ph.D. Thesis, Dept. of Elect. Eng., McGill University, Montreal, Canada, 1968.

8. Bose, R.C. and J.G. Caldwell. Synchronizable Error-Correcting Codes. Inf. and Control, Vol. 10, pp. 616-630, 1967.

9. Weldon, E.J. Jr. A Note on Synchronization Recovery with Extended Cyclic Codes. Inf. and Control, Vol. 13, pp. 354-356, 1968.

10. Tavares, S.E. and M. Fukada. Synchronization of Cyclic Codes in the Presence of Burst Errors. Inf. and Control, Vol. 14, 1969, pp. 423-441.

11. Tavares, S.E. and M. Fukada. Further Results on the Synchronization of Binary Cyclic Codes. IEEE Trans. Inf . Theory, IT-16, p. 238, 1970.

12. Levy, J.E. Self-Synchronizing Codes Derived from Binary Cyclic Codes. IEEE Trans. Inf. Theory, IT-12, p. 286, 1966.

13. Mandelbaum, D. Some Hybrid Methods for Synchronization of Cyclic Codes. Inform. Control, Vol. 19, pp. 93-103, 1971.

14. Mandelbaum, D. A Note on Synchronizable Error-Correcting Codes. Inform. and Control, Vol. 13, p. 429, 1968.

15. Shiva, S.G.S. and G. Seguin. Synchronizable Error-Correcting Binary Codes. IEEE Trans. Inf. Theory, IT-16, p. 241, 1970.

16. Tong, S.Y. Correction of Synchronization Errors with Burst-Error-Correcting Cyclic Codes. IEEE Trans. Inf. Theory, IT-15, p. 106, 1969.

17. Lewis, D.J. and M. Fukada. A Note on Burst-Error Correction Using the Check Polynomial. Department of Electrical Engineering, McGill University, Montreal, 1970.

18. Tavares, S.E. and S.G.S. Shiva and P.E. Allard. Some Recent Results on Cyclic Codes. Presented at the 1970 EIC Annual and General Professional Meeting, Ottawa, September 16-18.

19. Stiffler, J.J. Synchronization of Telemetry Codes. IEEE Trans. Space Elec. and Telemetry, SET-8, p. 112, 1962.

20. Golomb, S.W. and B. Gordon. Codes with Bounded Synchronization Delay. Inform. and Control, 8, p. 355, 1965.

21. Schaltz, R.A. Codes with Synchronization Capability. IEEE Trans. Inf. Theory, IT-12, p. 135, 1966.

22. Hatcher, J.R. On a Family of Error-Correcting and Synchronizable Codes. IEEE Trans. Inf. Theory, IT-15, p. 620, 1969.

23. Gillbert, E.N. Synchronization of Binary Messages. IRE Trans. Inf. Theory, IT-6, p. 470, 1960.

24. Ullman, J.D. Near-Optimal, Single Synchronization - Error-Correcting Codes. IEEE Trans. Inf. Theory, IT-12, p. 718, 1966.

25. Ullman, J.D. On the Capabilities of Codes of Correct Synchronization Errors. IEEE Trans. Inf. Theory, IT-13, p. 95, 1967.

26. Solomon, G. Self-Synchronizing Reed-Solomon Codes. IEEE Trans. on Inf. Theory, IT-14, p. 608, 1968.

27. Galabi, L. and W.E. Hartnett. A Family of Codes for the Correction of Substitution and Synchronization Errors. IEEE Trans. Inf. Theory, IT-15, p. 102, 1969.

28. Hachett, G.M. Jr. Synchronization of Cyclic Codes. First Ann. Conf. Inform. Sci. Systems, Princeton, N.J., 1967.

29. Tavares, S.E. and M. Fukada. Synchronization of a Class of Codes Derived from Cyclic Codes. Information and Control, Vol. 16, No. 2, April 1970.

30. Mandelbaum, D. Synchronization of Codes by Means of Kautz's Fibonacci Encoding. Information and Control, Vol. 18, No. 2, March 1972.

VITAE

Name: Paul Moreau

Born: Hearst, Ontario

School: Secondary University of Ottawa High School

University University of Ottawa, Ba.Sc. 1969.