ON THE DECOMPOSITION OF CYCLIC CODES INTO CYCLIC CLASSES

by

Paul Emile Allard, MSc.

Submitted to the Department of Electrical Engineering, of the University of Ottawa, in partial fulfilment of the requirements for the degree of Doctor of Philosophy

Department of Electrical Engineering,

Faculty of Science and Engineering,

The University of Ottawa,

Ottawa, Canada.

January 1973

ABSTRACT

In this thesis, the problem of partitioning a cyclic code into equivalence classes and obtaining a representative element from every class is examined. The equivalence classes are defined by that property of cyclic codes which renders them invariant under the permutation group which cyclically shifts the coordinates of the codewords. The problem of counting cyclic classes in a cyclic code is solved, as well as that of obtaining a representative codeword from every cyclic class. Knowledge of cyclic class representatives is then used in dealing with the problem of binary cyclic code synchronization.

ACKNOWLEDGEMENTS

The author wishes to gratefully acknowledge the assistance provided by his supervisors Prof. S.G.S. Shiva of the University of Ottawa and Dr. S.E. Tavares of Queen's University, Kingston, Ontario. Their help, encouragement and dedication were important factors in making this work both pleasant and profitable. To his friend Dr. G. Seguin of the Royal Military College, Kingston, a note of thanks for the many stimulating discussions and useful suggestions. Thanks are also due to his colleagues at the Communications Research Centre in Ottawa for providing that invaluable research atmosphere and to Mrs. Bev Lake for typing the manuscript.

The author is grateful, for financial assistance, to the Department of Communications and to the National Research Council of Canada under Grants A-3371 and A-7781.

LIST OF SPECIAL SYMBOLS

$a \in A$	the element a belongs to the set A.
$A \subset B$	the set A is contained in B.
g.c.d.	greatest common divisor.
l.c.m.	least common multiple.
μ	Möbius function.
ϕ	Euler function.
$[y]$	integer part of y.
$\cup$	union symbol.
$\cap$	intersection symbol.
$<x>$	cyclic group generated by x.
GF(q)	Galois Field of q elements.
GF[x]	polynomial ring over GF(q).
M(i)	minimal ideal generated by $(x^n-1)/h_i(x)$.
$M^*(i)$	non-zero elements of M(i).
$E_i(x)$	idempotent element of $M^*(i)$.
$\forall a \in S\|$	for all elements a belonging to S such that ...
r[G(x),d]	degree of g.c.d. $(x^d-1, G(x))$.
W(c)	number of non-zero coordinates of the vector c.

TABLE OF CONTENTS

CHAPTER 1

1. INTRODUCTION

The theory of error correcting codes has its origins in the now celebrated work of Shannon[1] who first demonstrated the possibility of achieving error free transmission of messages by introducing a certain redundancy in the data stream. This work which established the existence of a family of codes having the property that the probability of error would tend to zero as the lengths of the codes approached infinity while maintaining a non-zero rate of information transmission, has stimulated the efforts of researchers for a quarter century in the search for such a family of codes. The fruits borne by this research can be measured by the ever increasing number of applications of coding to various channels such as HF, VHF, satellite and deep space[47].

Historically, two schools of thought have emerged yielding different but not unrelated approaches to the solution of the problem of code construction. These may be broadly classified as the block coding and non-block coding approaches. In the block coding approach, a block of k information digits is encoded into a block of n digits, $n > k$, and transmitted over the channel, the redundancy in the block being dependent only on the information digits present in that particular block. In the non-block or convolutional coding approach, the block structure is lost, a continuous processing of information takes place with the check digits intersperced amongst the information bits[2]. The convolutional codes have been the subject of numerous studies[3,4,5,6] and

demonstrated to be an effective means of combatting channel noise[47]. The present work however is concerned solely with that subclass of block codes called cyclic codes[7].

The study of cyclic codes as such originated with Prange[8], although earlier authors made use of them under a different guise[9,29,48]. The real history of cyclic codes begins however with the work of Bose-Chaudhuri[11] and Hocquenhem[12], who provided the first constructive algorithm for generating a class of cyclic codes with a guaranteed minimum distance. Other codes such as the Hamming[9], Reed-Muller[29], Reed-Solomon[38], were subsequently shown to be special cases of the Bose-Chaudhuri-Hocquenhem (BCH) codes[15], so that these codes have been by far the most extensively studied class of cyclic codes. The decoding algorithm for the BCH codes was first given by Peterson[14] and later refined by Berlekamp[15], Chien[49] and Massey[50]. The richness of the algebraic structure of the cyclic codes as well as their relative ease of implementation have been the important factors that have contributed to their study and use. Although to this day there exist no known class of cyclic codes that satisfy the noisy-channel theorem, it is perhaps significant that the first constructive method for generating a class of asymptotically good block codes relied heavily on the knowledge gained in the study of cyclic codes[51].

Now the cyclic codes are by definition invariant under that group of permutations which consist in cyclically shifting the coordinates of codewords[15]. The equivalence classes so obtained are called cycles and a problem that naturally arises is that of generating

one representative of each cycle in the code. This problem was first considered by MacWilliams[16] and later by Goethals[17] who outlined a technique for finding cycle representatives in minimal ideal codes. More recently, Scholtz and Welch[59] gave a systematic procedure for obtaining a complete set of cycle representatives for the case where the cyclic code is the entire vector space of binary n-tuples. In this thesis we are concerned with developing an algorithm for finding all the cycle representatives in any cyclic code. Some applications are given by Reed[20] and Massey[21] for finding sequences with special correlation properties, and by MacWilliams[16] in connection with obtaining the weight distribution of cyclic codes. In this thesis, we give an application of the results obtained to the problem of synchronization of cyclic codes.

The greater part of the second chapter can be considered introductory in that we are concerned with establishing those coding and algebraic concepts required for further development of the thesis. Here we draw heavily upon the excellent works on coding theory of Peterson[7] and Berlekamp[15]. The importance of the BCH codes to the coding literature is recognized by devoting a number of sections (2-3 to 2-7) to a discussion of some of their properties. The better known classes of BCH codes such as the Hamming, Reed-Muller, Reed-Solomon are also briefly mentioned. In Section 2-8, we introduced a number of decomposition theorems for cyclic codes. The results derived in this section provide the necessary tools for the solution of the problem of cyclic representatives.

In Chapter 3 we enter into the subject matter of the thesis as such by deriving a number of results associated with the allied problem of counting cycles in cyclic codes. The last section of this chapter is concerned with finding cycle representatives in minimal ideal codes and serves as a link to the next chapter where this problem is dealt with at length.

Chapter 4 contains the complete solution to the problem of cycle representatives and is divided into three sections. The first contains a technique for obtaining the cycle representatives of those cyclic codes whose parity check polynomial is of a special type. The approach here taken follows the lines of that of Seguin[22] who considered the case for codes of length $n = 2^s-1$. The material of this section also provides additional tools of great value when we consider the synchronization aspect of cyclic codes in Chapter 5. The results of the second section are more general in that no restrictions are placed on the parity check polynomial, however at the expense of greater computational complexity. The results should be compared to those of Goethals[17] and MacWilliams[16] in that they represent an extension of their work. In the last section of the third chapter, a synthesis of the techniques introduced is made to yield a practical computational algorithm.

In Chapter 5, applications of the theory of cycle representatives to the synchronization problem associated with cyclic codes are considered. The use of cyclic codes requires obtaining proper frame synchronization at the receiver before decoding can begin. The possibility of frame synchronization loss has led various authors to propose a number

of techniques for synchronization recovery. Of these we mention the works of Bose and Caldwell[24], Weldon[52], Tong[42], Mandelbaum[44,46] and Tavares and Fukada[23,43]. It will be shown in that chapter that the natural applications of the theory of cycle representatives is in the generalization of the synchronization techniques of Tavares and Fukada[23], as well as that of Bose-Caldwell[24] and Weldon[52]. The generalization so obtained results in a greater insight into the synchronization process as well as an increase in the rate at which information can be transmitted over a channel susceptible to loss of synchronization.

Finally in Chapter 6, we include a number of concluding remarks followed by some tables which compare the known synchronization techniques for various conditions of slip and additive errors.

Before going on to Chapter 2, we may mention that some of the results of this thesis have already been reported elsewhere[18,19].

CHAPTER 2

CYCLIC CODES

2-1 Linear Block Codes

In this chapter, the basic definitions and theorems relating to cyclic codes and relevant to the present work will be established. First it is to be mentioned that cyclic codes are a subclass of the general class of linear block codes[7]. An (n,k) linear block code is defined as a subspace of dimension k of the vector space of n-tuples over a Galois Field of q elements, GF(q), q a power of some prime p. It is then possible to associate with such a code, a set of k linear independent vectors which form the basis of the subspace and hence a matrix G called the generator matrix, whose row space is the (n,k) code. Also there exists another matrix H called the parity check matrix of the code whose rows are linearly independent and orthogonal to the rows of G. The rows of H then generate an (n,n-k) subspace called the dual space or simply the dual code. If the generator matrix is reduced to a normalized form G' of the type $G' = (I_k, P)$ where I_k is a kxk identity matrix and P is a k x (n-k) matrix, we have the following relation between G' and H.

Theorem 2-1-1

If the generator matrix of an (n,k) code is of the form $G' = (I_k, P)$, then there exists a parity check matrix of the form $H = (-P^T, I_{n-k})$, where P^T is the transpose of P and I_{n-k} is a (n-k)x(n-k) identity matrix.

Proof

Since the sum of the ranks of G' and H is clearly n, and noting that $H(G')^T = 0$, it follows that the row space of G' and H are dual of one another.

Q.E.D.

In most applications it is useful to have the generator matrix G in the form G'. A codeword is then obtained by taking k information digits, considered as a row vector J and forming the matrix product

$$C = J\,G'.$$

The first k digits of the codeword C are then equal to J and the code is said to be systematic. We note that all linear codes can be rendered systematic by proper row-columns operations on G for reduction to its normalized form G'. The codes generated by G and G' are said to be equivalent in that they are column permutations of each other.

We define the weight of a vector C, denoted as W(C), to be the number of non-zero coordinates. The Hamming distance between two codewords C_1 and C_2, denoted as $D(C_1,C_2)$, is then defined as the number of coordinates in which C_1 and C_2 differ, so that we have

$$D(C_1,C_2) = W(C_1-C_2).$$

It then follows that for a linear code, the distance between any two codewords is equal to the weight of some other codeword because of the closure property of vector spaces. The minimum distance of a linear code is then equal to the minimum weight vector in the code. The im-

portance of the concept of minimum distance arises from the fact that for a binary symmetric channel, (BSC), the error correcting capability of the code is related to its minimum distance according to the relation

$$t = \left[\frac{(d-1)}{2}\right],$$

where t is the maximum number of errors the code can correct, and d is its minimum distance[7].

Assuming then that in the process of transmission of the code-word C, an error pattern ζ generated by the channel is added to C. The received vector will then be of the form

$$R = C + \zeta.$$

The decoder then calculates the "syndrome" of R defined as

$$S = HR^T = HC^T + H\zeta^T = H\zeta^T,$$

since C being a vector in the row-space of G', $HC^T = 0$, as implied by theorem 2-1-1. The problem then is to determine the most likely error pattern that could give rise to S. For the BSC, the most likely error pattern ζ' is the one having minimum weight and such that $S = H(\zeta')^T$. It is not difficult to show that if $\zeta' \neq \zeta''$ and $W(\zeta'), W(\zeta'') \leq (d-1)/2$, then

$$S' = H(\zeta')^T \neq S'' = H(\zeta'')^T,$$

so that the syndrome S' uniquely characterizes the error pattern ζ'.

Although conceptually simple, a decoder for block codes can be extremly difficult to implement due to the larger number of possible error patterns. Highly structured block codes such as the cyclic codes considerably simplify the decoding problem[15].

Now the relation $HC^T = 0$ implies a linear dependence amongst certain columns of H, so that the minimum distance of a linear code is related to the smallest set of columns of H that are linearly dependent. We state the following[7]:

Theorem 2-1-2

A linear block code has minimum distance d, if and only if every set of (d-1) columns or less of its parity check matrix are linearly independent.

We now proceed to investigate a certain class of block codes called cylic codes, and how these can be so constructed so as to satisfy the conditions of the above theorem.

2-2 Cyclic Codes and Ideals

Cyclic codes are by far the most extensively studied class of linear block codes. Their high degree of algebraic structure has considerably simplified the encoding-decoding problem usually associated with other block codes[7,15]. In this section we develop a number of basic results on cyclic codes, and we relate these to the general matrix formulation of block codes of Section 2-1.

A code V is said to be cyclic if for all codewords of the form $C = (c_0,c_1,c_2,\ldots,c_{n-1})$ in V, $C' = (c_{n-1},c_0,c_1,\ldots,c_{n-2})$ is also in V.

Note that C' is obtained by cyclically shifting C one unit to the right. It is convenient at this point to associate with every vector $C = (c_0,c_1,c_2,\ldots,c_{n-1})$ the polynomial $C(x) = c_0 + c_1x + c_2x^2 + \ldots + c_{n-1}\,x^{n-1}$. Furthermore, all such polynomials will be considered elements of the polynomial ring GF[x] modulo x^n-1. Also all codewords (vectors), are defined over GF(q). Recalling the definitions of Rings and Ideals[25], we have

Theorem 2-2-1

In the ring of polynomials modulo x^n-1, a code is a cyclic subspace if and only if it is an ideal.

Proof

If the subspace is an ideal and C(x) one of its elements, then

$$xC(x) = c_{n-1} + c_0x + c_1x^2 \ldots + c_{n-2}x^{n-1}$$

modulo x^n-1, is also an element of the ideal and hence the subspace is cyclic. Conversely if the subspace is cyclic then t(x)C(x) modulo x^n-1 is also an element of the subspace for all t(x) in the ring GF[x] mod. x^n-1, and the subspace is an ideal.

Q.E.D.

The importance of the above theorem stems from the fact that every cyclic code in the ring of polynomials modulo (x^n-1) is a principal ideal[7] and therefore there exists a polynomial G(x), a divisor of (x^n-1) of degree (n-k) which divides every codeword. G(x) is termed the generator polynomial and generates a cyclic code of dimension k. Every codeword is then of the form

$$C(x) = J(x)G(x),$$

where $J(x)$ is any polynomial of degree $(k-1)$ or less. The generator matrix corresponding to $G(x) = g_0 + g_1x + \ldots + g_{n-k}x^{n-k}$ is

$$G = \begin{bmatrix} & & & & & \longleftarrow (k-1) \text{ zeros} \longrightarrow & & & & \\ g_0 & g_1 & g_2 & \cdots & g_{n-k} & 0 & 0 & 0 & \ldots & 0 \\ 0 & g_0 & g_1 & \cdots & g_{n-k-1} & g_{n-k} & 0 & 0 & \ldots & 0 \\ 0 & 0 & g_0 & \cdots & g_{n-k-2} & g_{n-k-1} & g_{n-k} & 0 & \ldots & 0 \\ \multicolumn{10}{c}{\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots} \\ 0 & 0 & 0 & \ldots & 0 & 0 & 0 & g_0 & \cdots & g_{n-k} \end{bmatrix}$$

An element $h(x)$ is said to be in the null space of the ideal generated by $G(x)$ if and only if

$$h(x)C_i(x) = 0 \text{ modulo } (x^n-1),$$

for every element $C_i(x)$ in the code generated by $G(x)$. The element $h(x)$ and $C_i(x)$ are said to be orthogonal, and the null space of $G(x)$ has then dimension $(n-k)$. In matrix notation, the parity check matrix of G can be obtained by noting that the coefficients of the polynomial $H(x)C_i(x) = 0$ mod. (x^n-1), where $C_i(x) = c_0 + c_1x + c_2x^2 + \ldots + c_{n-1}x^{n-1}$ is a codeword in the cyclic space generated by $G(x)$ and $H(x) = (x^{n-1})/G(x) = h_0 + h_1x + h_2x + \ldots + h_{n-1}x^{n-1}$, are the first row in the matrix product

$$\begin{bmatrix} h_0 & h_1 & h_2 & \cdots & h_{n-1} \\ h_{n-1} & h_0 & h_1 & \cdots & h_{n-2} \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ h_1 & h_2 & h_3 & \cdots & h_0 \end{bmatrix} \begin{bmatrix} c_0 & c_1 & c_2 & \cdots & c_{n-1} \\ c_{n-1} & c_0 & c_1 & \cdots & c_{n-2} \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ c_1 & c_2 & c_3 & \cdots & c_0 \end{bmatrix} = HC = CH = 0.$$

Taking the transpose on both sides, it follows that

$$\begin{bmatrix} h_0 & h_{n-1} & h_{n-2} & \cdots & h_1 \\ h_1 & h_0 & h_{n-1} & \cdots & h_2 \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ h_{n-1} & h_{n-2} & h_{n-3} & \cdots & h_0 \end{bmatrix} \begin{bmatrix} c_0 \\ c_1 \\ \vdots \\ c_{n-1} \end{bmatrix} = HC_i^T = 0$$

Only the first (n-k) rows of the H matrix need be taken since these are linearly independent and the dimension of the null space of G is (n-k).

It is known that (x^n-1) factors into irreducible factors over GF(q), whose roots are all the n^{th} roots of unity[26] in some extension field of GF(q). Since the generator polynomial G(x) of a cyclic code will divide (x^n-1), we can now specify the cyclic code by the roots of unity that its generator polynomial G(x) contains. This is the approach that will be taken in the construction of cyclic codes. In the following discussion we will assume that the reader is familiar with the basic elements of Finite Field theory[55,56].

2-3 The BCH Codes

No other class of cyclic block codes has been extensively studied as the Bose-Chaudhuri-Hocquenhem (BCH) codes[15,36]. This class of multiple error correcting codes contains as subclasses many other previously known

classes of cyclic codes and is the largest known class of cyclic random error correcting codes. Their construction is dependent upon the following lemma, concerning the Vandermonde determinant[7], which we state without proof.

Lemma 2-3-1

Let $x_1, x_2, x_3, \ldots, x_n$ be a set of indeterminates. The determinant

$$\begin{bmatrix} 1 & x_1 & x_1^2 & x_1^3 & \cdots & x_1^{n-1} \\ 1 & x_2 & x_2^2 & x_2^3 & \cdots & x_2^{n-1} \\ 1 & x_3 & x_3^2 & x_3^3 & \cdots & x_3^{n-1} \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ 1 & x_n & x_n^2 & x_n^3 & \cdots & x_n^{n-1} \end{bmatrix}$$

can be expanded as $\Pi(x_j - x_i)$.

$$1 \leq i \leq j \leq n$$

If we let α denote any element of order n in an extension field $GF(q^S)$ and we consider the code whose generator polynomial contains the roots $(\alpha, \alpha^2, \alpha^3, \alpha^4, \ldots, \alpha^{d-1})$, then all codewords C(x) whose coordinates are in GF(q), must satisfy the following set of equations:

$$\begin{bmatrix} 1 & \alpha & \alpha^2 & \alpha^3 & \cdots & \alpha^{n-1} \\ 1 & \alpha^2 & (\alpha^2)^2 & (\alpha^2)^3 & \cdots & (\alpha^2)^{n-1} \\ 1 & \alpha^3 & (\alpha^3)^2 & (\alpha^3)^3 & \cdots & (\alpha^3)^{n-1} \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ 1 & \alpha^{d-1} & (\alpha^{d-1})^2 & (\alpha^{d-1})^3 & \cdots & (\alpha^{d-1})^{n-1} \end{bmatrix} \begin{bmatrix} c_0 \\ c_1 \\ c_2 \\ \vdots \\ c_{n-1} \end{bmatrix} = HC^T = 0.$$

From the above lemma, it is seen that every set of $(d_{BCH}-1)$ columns of H will be linearly independent, since their determinant is of the Vandermonde type and hence the weight of C(x), $W(C(x) \geq d_{BCH}$, which we refer to as the designed distance of the code. If $m_i(x)$ denotes the minimal polynomial[7] of α^i, we are then assured by the above construction that the code whose generator polynomial is

$$G(x) = \text{l.c.m.}(m_1(x), m_2(x), \ldots, m_{d_{BCH}-1}(x)),$$

where l.c.m. refers to the least common multiple, has minimum distance at least d_{BCH}. The length of the code is n and its dimension is n-deg (G(x)). A BCH code is said to be primitive if it has length q^s-1 and non-primitive otherwise. A partial listing of binary primitive and non-primitive BCH codes is contained in [27].

We note that d_{BCH} is a lower bound on the actual minimum distance of the code d, which may in fact be greater than d_{BCH}[54]. A case of interest is that of a binary cyclic code whose designed distance d_{BCH} divide n. Assuming then $d_{BCH}.b = n$, because α is a n^{th} root of unity of order n over GF(q) and the fact that we can write

$$(1 + x^b)(1 + x^b + x^{2b} + \ldots + x^{(d_{BCH}-1)b}) = 1 + x^n,$$

the polynomial $(1 + x^n)/(1 + x^b)$ will be a codeword and hence $d_{BCH} = d$.

We now review briefly some of the better known subclasses of the BCH codes.

2-4 Reed-Solomon (RS) Codes[38]

If we refer to GF(q) as the symbol field and $GF(q^s)$ as the locator field, then the RS codes are a class of non-binary BCH codes with the property that the locator field and symbol field are identical, i.e., when s = 1. The coordinates of the codewords will then be from GF(q), with the length of the codewords equal to q - 1. If α is a primitive element in GF(q), the generator polynomial of an RS code of designed distance d_{BCH} is

$$G(x) = (x-\alpha)(x-\alpha^2)(x-\alpha^3)\ldots(x-\alpha^{d_{BCH}-1}).$$

Since the weight of G(x) is d_{BCH}, the distance, length and number of information symbols are related as

$$d = n - k - 1.$$

Such codes are said to be optimal. RS codes are very powerful for correcting multiple burst. They are also sometimes used as the outer code in the construction of concatenated codes[28].

2-5 Reed-Muller (RM) codes[29]

The equivalence of the RM codes to a subclass of the BCH codes was first established by Kasami, Lin and Peterson[30] who later generalized these to the non-binary case[13]. Their construction for the binary case is as follows:

$$G(x) = \Pi (x-\alpha^j), \quad 0 \leq j < 2^s - 1 , \; 0 \leq W(j) < s - r ,$$

where W(j) denotes the sum of the digits in the binary expansion of j and α is a primitive element of GF(2^s). The r^{th} order RM code is obtained by annexing the all one vector of length 2^s to the generator matrix G. We note that the code obtained is equivalent to a RM code in the sense that the generator matrix is a column permutation of the matrix of a RM code. Since for $j = 1,2,\ldots,2^{s-r}$, $w(j) \leq s-r$, the r^{th} order RM code is seen to be a subcode of the BCH codes of designed distance $d_{BCH} = 2^{s-r}$ whose generator matrix has been modified by annexing the all one vector of length 2^s.

2-6 Hamming Codes[9]

The Hamming codes are the simplest of the BCH codes. If α is a primitive element of GF(2^s), the generator polynomial is

$$G(x) = m_1(x).$$

These codes are single error correcting codes of length $n = 2^s-1$, and $k = 2^s-s-1$. Further, these codes are extremely well understood in that their weight distribution is completely known[31]. They are also perfect codes in that they satisfy the sphere-packing bound[15] with equality.

2-7 M-Sequence Codes[10]

This class of primitive BCH codes has for generator polynomial

$$G(x) = (x^n-1)/m_1(x),$$

where $n = 2^s-1$, and $m_1(x)$ is the minimal polynomial of α, a primitive root of a $GF(2^s)$. The 2^s-1 non-zero codewords all have weight 2^{s-1}. The codewords are also referred to as Maximal-Length-Shift-Register-Sequences. This code is also the dual of an Hamming code.

Other well known classes of cyclic codes include the Projective Geometry codes and the Euclidean Geometry codes[32]. These codes are easily decodable but deteriorate much faster than BCH codes in the sense of comparing achievable rate as a function of minimum distance. Other cyclic codes are the cyclic product codes[33], the Fire codes for burst error correction[34] and the quadratic residue codes[15] which are considered as hopeful candidates as a class of cyclic codes that would meet the requirements of the noisy-channel theorem[40].

In the preceeding sections we have presented some of the properties of cyclic codes and discussed briefly a number of the better known classes of the BCH codes. Our presentation has been purposely brief, for in the present work we are not so much concerned with classes of cyclic codes as with that property possessed by cyclic codes which leaves them invariant under a cyclic shift of their coordinates. Before considering this aspect further, we establish a number of results on the algebraic structure of cyclic codes.

2-8 On the Algebraic Structure of Cyclic Codes

In this section, we derive some known results on the algebraic structure of cyclic codes which will be of value in the solution of the problem of cycle representatives which we consider in the next chapter.

We let G(x) denote the generator polynomial of an (n,k) binary cyclic code, n odd, having as parity check polynomial $(x^n-1)/G(x) = H(x) = h_1(x)h_2(x) \ldots h_r(x)$, where $h_i(x)$, i = 1,2,3,...,r are the irreducible factors of degrees m_i. We denote by M(i) the code generated by $(x^n-1)/h_i(x)$.

Theorem 2-8-1

The elements of M(i) form a field of order 2^{m_i}.

Proof

Since M(i) is an ideal in a commutative ring, we need only to prove the existence of a multiplicative inverse and idempotent. Now $G(x)^{2^{m_i}} = G(x)$, hence $G(x)^{2^{m_i}-1}$ acts as an idempotent. Also since M(i) does not contain any proper sub-ideal, all elements in M(i) will be of the form a(x)G(x), with g.c.d. $(a(x), h_i(x)) = 1$. Therefore there exists an element b(x) such that $a(x)b(x) = 1$ modulo $h_i(x)$, and $b(x)G(x)^{2^{m_i}-2}$ is the inverse of a(x)G(x). Q.E.D.

Corollary 2-8-1-1

M(i) is isomorphic to the field of polynomials modulo $h_i(x)$.

Proof

All finite fields of the same order are isomorphic[26] so that if f(x) is an element of the field of polynomials modulo $h_i(x)$, the isomorphism is established by mapping f(x) into $f(x)E_i(x)$, where $E_i(x)$ is the idempotent of M(i).

We note that if a(x) is a primitive element in the field of polynomials modulo $h_i(x)$, then $a(x)E_i(x)$ is a primitive element of M(i). M(i) is said to be an irreducible code because it does not contain any proper cyclic sub-code. The fact that such codes have the structure of a field will considerably simplify the problem of obtaining their cycle representatives.

If we let S' and S" be any two sets of elements, and we take S' + S" to mean the set obtained by forming all possible sum of elements from S' and S", we have the following decomposition theorem for cyclic codes.

Theorem 2-8-2

Let $G(x) = (x^n-1)/H(x) = (x^n-1)/h_1(x)h_2(x)\ldots h_r(x)$ generate an (n,k) cyclic code V_0, n odd. Then

$$V_0 = M(1) + M(2) + \ldots + M(r) = \sum_{i=1}^{r} M(i).$$

Proof

Clearly $\sum_{i=1}^{r} M(i) \subset V_0$. Noting that the minimal ideals M(i) and M(j), $i \neq j$, are orthogonal, all the elements in $\sum_{i=1}^{r} M(i)$ must be distinct. The result then follows upon noting that

$$\left|\sum_{i=1}^{r} M(i)\right| = \prod_{i=1}^{r} \left|M(i)\right| = \left|V_0\right|. \qquad \text{Q.E.D.}$$

This result, first used by MacWilliams[16], in the study of cyclic codes, will be of great value in the solution of the problem of cycle representatives in cyclic codes. The complete solution however also depends on some properties related to the multiplicative structure of cyclic codes[39], which we now establish. We denote by $M^*(i)$ the non-zero elements of $M(i)$.

Definition 2-8-1

Let V_0 be the ideal generated by $(x^n-1)/H(x) = (x^n-1)/h_1(x)\ldots h_r(x)$. Also, let $M^*(i_1,i_2,i_3,\ldots,i_s) = M^*(i_1) + M^*(i_2) + \ldots + M^*(i_s)$, where $M(i_j)$, $j = 1,2,\ldots,s$ are minimal ideals in V_0 and $s \leq r$.

We note that the set obtained according to the above definition will consist of elements all of the form $\sum_{j=1}^{s} m_{i_j}(x)$, where $m_{i_j}(x)$ is a non-zero element of $M(i_j)$, $j = 1,2,\ldots,s$. The following theorem further characterizes this set.

Theorem 2-8-3

$M^*(i_1,i_2,i_3,\ldots,i_s)$ is a group under multiplication modulo (x^n-1) of order $\prod_{j=1}^{s} |M^*(i_j)|$.

Proof

First we note that $M^*(i_j)$ being the set of non-zero elements of the field $M(i_j)$ is therefore a cyclic group under multiplication[25]. Furthermore because of the orthogonality of $M^*(i)$ and $M^*(j)$, if $\sum_{j=1}^{s} m_{i_j}(x)$ and $\sum_{j=1}^{s} m'_{i_j}(x)$ are elements of $M^*(i_1,i_2,\ldots,i_s)$,

(a) $\left[\sum_{j=1}^{s} m_{i_j}(x)\right]\left[\sum_{j=1}^{s} m'_{i_j}(x)\right] \varepsilon\ M^*(i_1,i_2,\ldots,i_s)$,

(b) $\left[\sum_{j=1}^{s} E_{i_j}(x)\right]$ acts as idempotent,

(c) $\left[\sum_{j=1}^{s} m^{-1}_{i_j}(x)\right]$ is the inverse of $\left[\sum_{j=1}^{s} m_{i_j}(x)\right]$.

This group is then Abelian and of order $\prod_{j=1}^{s} |M(i_j)|$.

Q.E.D.

We note in passing, that if we let $h_1(x),h_2(x),\ldots,h_L(x)$ denote all the irreducible factors of (x^n-1) over GF(2), the group $M^*(1,2,\ldots,L)$ is then the set of units of the ring GF[x]mod.(x^n-1), since all of its elements will be relatively prime to (x^n-1). Further $|M^*(1,2,\ldots,L)|/n$, is the number of invertible circulants[40], a circulant being a nxn matrix whose i^{th} row is obtained from the $(i-1)^{th}$ row by cyclically shifting the latter one place to the right for i = 2,3,...,n.

We now establish the fact that these groups can act as the elements of a partition of the code V_0. $M^*(0,0,\ldots,0)$ will be taken to mean that multiplicative group consisting only of the all zero-element.

Theorem 2-8-4

Let M(1),M(2),...,M(s) be all the minimal ideals in a binary code V_0, of odd length n. Then the set of groups

$$M^*(\delta_1 1,\delta_2 2,\ldots,\delta_s s) = \delta_1 M^*(1) + \delta_2 M^*(2) + \ldots + \delta_s M^*(s),$$

define a partition on V_0 for the 2^s possible choices of $\delta_i = 1$ or 0, $i = 1,2,\ldots,s$.

Proof

The proof follows readily from the fact that the above theorem is merely another reformulation of theorem 2-8-2, since adding the all-zero vector to $M^*(i)$ yields $M(i)$, $i = 1,2,\ldots,s$.

Q.E.D.

In this chapter, we have established those coding and algebraic concepts needed for the further development of the subject matter of the thesis. In Section 2-1, we briefly reviewed some fundamentals of linear block codes. In Section 2-2, we related cyclic codes to the general context of linear block codes, followed in Section 2-3 by a discussion on the major class of cyclic codes, the BCH codes. The next four sections occupied us with a short review of some other well known BCH codes. Finally in Section 2-8, we established a number of basic results on the algebraic structure of the cyclic codes. Amongst these, the decomposition theorems 2-8-2 and 2-8-4 are key theorems and will prove of considerable value in the next chapters.

We now establish some preliminary results associated with the problem of cycle representatives.

CHAPTER 3

CYCLES AND CYCLIC CODES

3-1 Counting Cycles in Cyclic Codes

In the preceeding chapter a number of properties of cyclic codes were established. Of particular interest to the present section is their property of invariance under a cyclic shifting of the coordinates of the codewords. There exists then a cyclic permutation group that partitions a code into equivalence classes called cycles[35], hence the problem of obtaining one element from every equivalence class in the code. In this section, we derive a number of results allied to the problem of cycle representatives and related to the counting of cycles in cyclic codes. First however we establish the following:

Definition 3-1-1

Let $c(x) = c_0 + c_1x + c_2x^2 + \ldots + c_{n-1}x^{n-1}$, then $x^ic(x) = c_{n-i} + c_{n-i-1}x + \ldots + c_{n-i-1}x^{n-1}$ is the i^{th} cyclic shift of $c(x)$, $0 \leq i \leq n-1$.

We now formally define the terms cycle and cycle representative.

Definition 3-1-2

A cycle is a set containing an n-tuple and all its distinct cyclic shifts. A cycle representative is any element from a cycle.

Definition 3-1-3

The order of a cycle is the number of elements in the cycle. The cyclic order of an element is the order of the cycle to which it belongs.

We also require the following well known result on the Möbius μ function.

Definition 3-1-4

The Möbius μ function is a function from the non-negative integers into (-1,0,1) defined by

$$\mu(0) = 0,$$

$$\mu(1) = 1,$$

$$\mu(n) = 0 \text{ if } p^2 \text{ divides } n \text{ for some prime } p,$$

$$\mu(n) = (-1)^k \text{ if } n = p_1 p_2 p_3 \cdots p_k,$$

where $(p_1 p_2 p_3 \cdots p_k)$ is a set of distinct primes.

Lemma 3-1-1 (Möbius inversion formula[26])

If, for $m > 0$, $g(m)$ and $F(m)$ are arithmetic functions defined by

$$g(m) = \sum_{d|m} F(d),$$

for all divisors d of m, then

$$F(m) = \sum_{d|m} g(d)\mu(m/d) = \sum_{d|m} g(m/d)\mu(d),$$

where μ is the Möbius function.

Before attacking the problem of finding cycle representatives for all the cycles in a cyclic code, it is clearly useful if we could count the number of cycles of order e in a cyclic code generated by $G(x) = (x^n-1)/H(x)$. Our first result specifies the order a cycle may have in cyclic codes of odd length n.

Theorem 3-1-1

In a cyclic code of length n, the order of a cycle divides n.

Proof

Let c(x) be a non-zero element of a cycle of order e. We then have

$$x^e c(x) = c(x) \text{ and } x^n c(x) = c(x) \text{ mod.}(x^n-1).$$

If $n = qe + r$, where $0 \leq r < e$, we can then write

$$x^n c(x) = x^{qe+r} c(x) = x^r c(x) = c(x) \text{ mod. } (x^n-1).$$

Since c(x) has order e, r = 0, and hence n = qe.

Q.E.D.

From the preceeding theorem, it is seen that whenever the length n is prime, all the non-zero elements in a cyclic code will have cyclic order 1 or n. However if n is not a prime, it may happen that a cyclic code will contain cycles of order less than n. We now derive an expression for counting the number of cycles of order e, when $e|n$, which belong to a binary cyclic code of odd length n. We require the following definition:

Definition 3-1-5

A polynomial $h(x)$ is said to have exponent e, if and only if $h(x)|(x^e-1)$ and $h(x)\nmid(x^j-1)$ for all $j < e$.

Theorem 3-1-2

Let $G(x)H(x) = (x^n-1)$. The number of cycles of order e, $e|n$, belonging to the binary cyclic code generated by $G(x)$ is

$$N(e) = 1/e \sum_{d|e} 2^{d-r[G(x),d]}\mu(e/d) = 1/e \sum_{d|e} 2^{r[H(x),d]}\mu(e/d).$$

$r[G(x),d]$ = degree of g.c.d. $((x^d-1),G(x))$,

$r[H(x),d]$ = degree of g.c.d. $((x^d-1),H(x))$,

and μ is the Möbius function.

Proof

The set of elements having cyclic order e form a vector space of dimension e, generated by the vector $1 + x^e + x^{2e} + \ldots + x^{(n/e-1)e}$ and its (e-1) cyclic shifts. Since all the elements having cyclic order e belong to the cyclic code generated by $1 + x^e + x^{2e} + \ldots + x^{(n/e-1)e}$, they must then be all of the form $p(x)\,(1 + x^e + x^{2e} + \ldots + x^{(n/e-1)e})$ where the degree of $p(x) < e$. Letting g.c.d.$((x^e-1),G(x)) = d(x) = r[G(x),e]$, we can find a polynomial $t(x)$, a divisor of (x^n-1), such that $t(x)G(x) = (1 + x^e + x^{2e} + \ldots + x^{(n/e-1)e})d(x)$, and the code contains a subspace of dimension e - degree $(d(x))$ whose elements all have cyclic order d, $d|e$. If $B(d)$ is the number of elements of that subspace having cyclic order d, then

$$eN(e) = B(e) = 2^{e-r[G(x),e]} - \sum_{d|e,e\neq d} B(d)$$

$$\sum_{d|e} dN(d) = \sum_{d|e} B(d) = 2^{e-r[G(x),e]}$$

The result of the theorem then follows upon using the Möbius inversion formula of lemma 2-4-1 and noting that $d-r[G(x),d] = r[H(x),d]$. Q.E.D.

Example 3-1-1

Consider the (63,45,3) BCH code. The generator polynomial of such a code is the l.c.m. of the minimal polynomials of the roots $(\alpha,\alpha^2, \alpha^3, \alpha^4, \alpha^5, \alpha^6)$ where α is a primitive element of a $GF(2^6)$. We wish to specify the number of elements of order e in the code. Since, $G(x) = (1 + x + x^6)(1 + x + x^2 + x^4 + x^6)(1 + x + x^2 + x^5 + x^6)$, there are $2^{63-18} = 2^{45}$ codewords, and because degree $(G(x))$ < degree $(H(x))$, it is easier to use the form

$$eN(e) = B(e) = \sum_{d|e} 2^{d-r[G(x),d]}\mu(e/d).$$

We then have,

$B(1) = N(1) = 2^{1-0}\mu(1) = 2,$

$B(3) = 3N(3) = 2^{3-0}\mu(1) + 2^{1-0}\mu(3) = 6,$

$B(7) = 7N(7) = 2^{7-0}\mu(1) + 2^{1-0}\mu(7) = 126,$

$B(9) = 9N(9) = 2^{9-0}\mu(1) + 2^{3-0}\mu(3) + 2^{1-0}\mu(9) = 504,$

$B(21) = 21N(21) = 2^{21-6}\mu(1) + 2^{7-0}\mu(3) + 2^{3-0}\mu(7) + 2^{-10}\mu(21) = 32,634,$

$B(63) = 63N(63) = 2^{63-18}\mu(1) + 2^{21-6}\mu(3) + 2^{9-0}\mu(7) + 2^{7-0}\mu(9)$

$+ 2^{3-0}\mu(21) + 2^{1-0}\mu(63) = 35,184,372,055,560.$

A number of corollaries to theorem 3-1-2 follows upon examining special cases for the generator polynomial G(x).

Corollary 3-1-2-1

The number of cycles of order e, in the entire space of n-tuples is

$$N(e) = (1/e) \sum_{d|e} 2^d \mu(e/d).$$

Proof

This result follows upon letting G(x) = 1 in the expression for N(e) of theorem 3-1-2.

Upon substitution of e = n, the corollary is seen to be the expression derived by Golomb[37] for the maximum number of words in a comma-free code of length n. This expression also gives the number of irreducible polynomials of degree e[15]. If we now let $G(x) = x^n - 1$ in theorem 3-1-2, we have the next result.

For all divisors d of e,

$$\sum_{d|e} \mu(e/d) = 1 \text{ if } e = 1,$$
$$= 0 \text{ if } e > 1.$$

If we are interested only in the number of distinct cycles belonging to the code, the next theorem is used. First, however we need the following:

Definition 3-1-6

Let $n = p_1^{r_1}p_2^{r_2}\ldots p_s^{r_s}$ be the prime factorization of $n > 0$. Then

$$\phi(n) = n(1-1/p_1)(1-1/p_2)\ldots(1-1/p_s)$$

$\phi(n)$ is called the Euler function[26], and it gives the number of positive integers smaller than n that are relatively prime to n.

Lemma 3-1-2

Let S be a cyclic permutation group of a set A, such that $|S| = n$. The number of equivalence classes into which the set S is divided by the equivalence relation induced by S is

$$M = 1/n \sum_{d|n} Y(s^d)\phi(n/d),$$

where the summation is over all the divisors d of n, $Y(s^d)$ being the number of elements in A that are left invariant under the permutation s^d an element of S. ϕ is the Euler function.

Theorem 3-1-3

Let $G(x)H(x) = (x^n-1)$, n odd. The number of distinct cycles belonging to the binary cyclic code generated by G(x) is

$$M = 1/n \sum_{d|n} 2^{d-r[G(x),d]}\phi(n/d) = 1/n \sum_{d|n} 2^{r[H(x),d]}\phi(n/d).$$

$r[G(x),d]$ = degree of g.c.d. $((x^d-1),G(x))$,

$r[H(x),d]$ = degree of g.c.d. $((x^d-1),H(x))$,

ϕ is the Euler function.

Proof

The proof follows from the above lemma upon noting that the number of elements in the code generated by G(x) that are left invariant when cyclically shifted d times is

$$2^{d-r[G(x),d]} = 2^{r[H(x),d]}. \qquad \text{Q.E.D.}$$

Example 3-1-2

Considering the (63,45,3) BCH code of example 3-1-1, we wish to determine the number of distinct cycles present in the code. From example 3-1-1

$$M = \sum_{e|n} N(e) = 558,482,097,752.$$

Using theorem 3-1-3,

$$M = 1/63(2\phi(63) + 2^3\phi(21) + 2^7\phi(9) + 2^9\phi(7) + 2^{21-6}\phi(3) + 2^{63-18}\phi(1)$$
$$= 558,482,097,752.$$

Again considering the special cases of G(x) = 1, and $G(x) = x^n-1$, the following corollaries result.

Corollary 3-1-3-1

The number of distinct cycles in the entire space of n-tuples is

$$M = 1/n \sum_{d|n} 2^d\phi(n/d)$$

Proof

This expression follows when substituting $G(x) = 1$ in theorem 3-1-3.

Corollary 3-1-3-2

$$\sum_{d|n} \phi(n/d) = n$$

Proof

This result follows upon letting $G(x) = x^n-1$ in the expression of theorem 3-1-3 and noting that the code generated by x^n-1 contains only the all-zero codeword.

We note that in all the expressions dealing with the number of cycles in cyclic codes, nothing has been said of the weight of the cycles. The following can be said when considering the entire space of n-tuples.

Theorem 3-1-4

In the space of 2^n binary n-tuples, the number of distinct cycle representatives of weight i and cyclic order e, $e|n$, is the coefficient of x^i in the expansion of

$$R_e(x) = 1/e \sum_{d|e} \mu(e/d)\left[1 + x^{n/d}\right]^d = 1/e \sum_{d|e} \mu(d) \left[1 + x^d\right]^{n/d},$$

where μ is the Möbius function.

Proof

The number of n-tuples of weight i that are left invariant by e cyclic shifts is given by the coefficient of x^i in the polynomial $(1 + x^{n/e})^e$, or

$$\sum_{d|e} dR_d(x) = \left(1 + x^{n/e}\right)^e.$$

Using the Möbius inversion formula yields the desired result.

Q.E.D.

The number of distinct cycles of order e, $e|n$, is $R_e(1)$ which is the expression of corollary 3-1-2-1. The number of distinct cyclic classes of order e, for n = 3 to 22 is given in Table 1.

Example 3-1-3

For n = 15, we wish to determine the number of cycle representatives of weight i and cyclic order e = 1,3,5,15.

$$R_1(x) = \mu(1)(1 + x^{15}) = 1 + x^{15}.$$

$$R_3(x) = 1/3\ (\mu(3)(1 + x^{15}) + \mu(1)\ (1 + x^5)^3) = x^5 + x^{10}.$$

$$R_5(x) = 1/5\ (\mu(5)(1 + x^{15}), + \mu(1)(1 + x^3)^5) = x + 2x + 2x + x^{12},$$

$$R_{15}(x) = 1/15\ (\mu(15)(1 + x^{15}) + \mu(5)(1 + x^5)^3 + \mu(3)(1 + x^3) + \mu(1)(1 + x)^{15}) = x + 7x^2 + 30x^3 + 91x^4 + 200x^5 + 33x^6 + 429x^7 + 429x^8 + 333x^9 + 200x^{10} + 91x^{11} + 30x^{12} + 7x^{13} + x^{14}.$$

If we are interested only in counting the number of cycles of weight i, we proceed as follows:

Theorem 3-1-5

In the space of 2^n binary n-tuples, the number of distinct cycle representatives of weight i, is the coefficient of x^i in the expansion of

$$P(x) = 1/n \sum_{d|n} (1 + x^d)^{n/d} \phi(d) = 1/n \sum_{d|n} (1 + x^{n/d})^d\ \phi(n/d).$$

where ϕ is the Euler function.

Proof

The proof follows from lemma 3-1-2 upon noting that the number of elements that are left invariant by d cyclic shifts and have weight i, is given by the coefficient of x^i in the polynomial

$$(1 + x^{n/d})^d. \qquad \text{Q.E.D.}$$

We note that the number of distinct cycles belonging to the vector space of 2^n n-tuples is given by P(1) which is also the expression of corollary 3-1-3-1.

Example 3-1-4

For n = 15, we wish to determine the number of cycles of weight i.

$$P(x) = 1/15\ (\phi(1)(1 + x)^{15} + \phi(3)(1 + x^3)^5 + \phi(5)(1 + x^5)^3 +$$

$$\phi(15)(1 + x^{15})) = 1 + x + 7x^2 + 31x^3 + 91x^4 + 201x^5 + 335x^6$$

$$+ 429x^7 + 429x^8 + 335x^9 + 201x^{10} + 91x^{11} + 31x^{12} + 7x^{13} + x^{14} + x^{15}.$$

We note also that $P(x) = \sum_{e|n} R_e(x)$.

In the last two theorems, the weight of the cycles in the entire space of binary n-tuples has been fully specified. The generalization of these theorems to any cyclic code is indeed a very difficult problem and except for special cyclic codes very few results have been obtained[15,57,58]. Knowledge of the weight distribution of codes is essential to the calculation of the probability of error at the receiver assuming a knowledge of the statistics of the channel. The theory of

cycle representatives finds application to the problem of the weight distribution by reducing the amount of labor involved[16] in such computations.

Having derived in this section a number of results related to the counting of cycles, we consider in the next section the problem of generating cycle representatives in irreducible codes.

3-2 Cycle Representatives in Irreducible Codes

The problem of generating a complete set of cycle representatives for irreducible codes was first solved by Goethals[17] using a result due to Nili[41]. For such a code M(i), it is a simple matter to show that all the non-zero elements will have as cyclic order e_i, the exponent of $h_i(x)$. The solution of the problem then lies in corollary 2-8-1-1 which establishes the isomorphism between M(i) and the field of polynomials mod.(x^n-1).

Lemma 3-2-1

The elements of $M^*(i)$ have cyclic order e_i, the exponent of $h_i(x)$.

Proof

Let $t_i'(x)$ be any element in $M^*(i)$. Let e be the least positive integer such that $x^e t_i'(x) = t_i'(x)$ mod.(x^n-1). Then $t_i'(x)(x^e-1) = 0$ mod.(x^n-1) and $h_i(x)$ will divide (x^e-1). It then follows that $e = e_i$, from the definition of the exponent of $h_i(x)$.

Q.E.D.

Theorem 3-2-1

Let a(x) be a primitive element in the field of polynomials modulo the irreducible polynomial $h_i(x)$. If m_i and e_i are the degree and exponent of $h_i(x)$, the expression

$$(a(x)E_i(x))^j, \quad j = 1,2,\ldots,(2^{m_i}-1)/e_i \tag{3-2-1}$$

where $E_i(x)$ is the idempotent of M(i), generates all the cycle representatives of M(i).

Proof

By the above lemma, all of the elements of $M^*(i)$ have cyclic order e_i. Further, in that there are $(2^{m_i}-1)/e_i$ cycles, we need only prove that the elements obtained through Eqn. (3-2-1) belong to distinct cycles. Assume then a value of $j_1 < (2^{m_i}-1)/e_i$ such that

$$(a(x)E_i(x))^{j_1} = x^c a(x)E_i(x), \quad 0 \leq c < e_i.$$

Now since all the elements in $M^*(i)$ have cyclic order e_i, we have

$$(a(x)E_i(x))^{j_1 e_i} = (a(x)E_i(x))^{e_i}$$

in contradiction of the assumption that a(x) is primitive, since under the isomorphism of corollary 2-8-1-1 we would have

$$(a(x))^{j_1 e_i} = (a(x))^{e_i}, \text{ with } j_1 e_i < 2^{m_i}-1.$$

Q.E.D.

The above theorem is of fundamental importance in the development of the theory of cycle representatives. When used in conjunction with the decomposition method of theorem 2-8-2, the possibility arises of obtaining a complete set of cycle representatives for any cyclic code. Goethals recognized this although he did not extend his result to the more general case[17]. In fact the problem becomes more complicated if we allow for the possibility of having cycles of different order.

In the next chapter, we will seek to generalize the result of theorem 3-2-1, so that we may be able to obtain a complete set of cycle representatives for any cyclic code. Because of the fact that some cyclic codes may have cycles of varying order, the decomposition theorem 2-8-4 will prove to be the key to the solution of the general case.

CHAPTER 4

CYCLE REPRESENTATIVES IN CYCLIC CODES

We now concern ourselves with the problem of obtaining a representative from all the cycles of any cyclic code. The results presented in this chapter have been reported elsewhere[18,19] and will be shown to yield all the cycle representatives of a cyclic code. The chapter is divided into three sections. In the first section, results are obtained which are applicable to those codes of length n whose parity check polynomial can be completely factored into irreducible polynomials of exponent n, this we refer to as the special case. Many of the results of this section will also find application to the problem of cyclic code synchronization which is to be considered in the next chapter. In the second section, no restrictions are placed on the factors of the parity check polynomial, this we called the general case. Finally in the third section, a practical computational algorithm is presented which incorporates the results of the first two sections. The results are derived for binary codes of odd length n, the extension to cyclic codes defined over a GF(q) follows a parallel development.

4-1 The Special Case

Let V_0 be a binary cyclic code of odd length n, generated by $G(x) = (x^n-1)/H(x)$. We let $h_i(x)$, $i = 1,2,\ldots,r$, be the distinct irreducible factors of $H(x)$ of degrees m_i and exponents e_i respectively. Also, let V_i be the subcode generated by the polynomial $G(x)h_1(x)\ldots h_i(x)$.

Definition 4-1-1

The complement of V_i in V_{i-1} is the set

$$(V_{i-1}-V_i) = \{m(x) | m(x) \in V_{i-1}, m(x) \notin V_i\}.$$

Theorem 4-1-1

The set $\{(V_0-V_1),(V_1-V_2),\ldots,(V_{r-1}-V_r)\}$ defines a partition on V_0^*, the non-zero elements of V_0.

Proof

We need to show that

a) $V_0^* = \bigcup_{i=1}^{r} (V_{i-1}-V_i)$,

b) $(V_{i_1-1}-V_{i_1}) \cap (V_{i_2-1}-V_{i_2}) = \theta$, if $i_1 \neq i_2$,

where θ is the null set, $\cup$ and $\cap$ being the union and intersection symbols respectively.

a) It is clear that $\bigcup_{i=1}^{r} (V_{i-1}-V_i) \subset V_0^*$, also if $m(x) \in V_0^*$, then g.c.d. $(x^n-1, m(x)) = G(x)h_1(x)\ldots h_{i_1}(x)$, $1 \leq i_1 < r$, so that $m(x) \in (V_{i_1-1}-V_{i_1})$, hence $V_0^* \subset \bigcup_{i=1}^{r} (V_{i-1}-V_i)$ and therefore,

$$V_0^* = \bigcup_{i=1}^{r} (V_{i-1}-V_i).$$

b) because the g.c.d. of x^n-1 and an element $m(x)$ of V_0^* is unique, if $m(x) \in (V_{i_1-1}-V_{i_1})$, then $m(x) \notin (V_{i_2-1}-V_{i_2})$

if $i_1 \neq i_2$, so that condition b) is satisfied.

Q.E.D.

The method for finding the cycle representatives as outlined in this section, will consist in finding the cycle representatives of the sets $(V_{i-1}-V_i)$, $i = 1,2,\ldots,r$. We therefore proceed to examine further the algebraic structure of the sets $(V_{i-1}-V_i)$.

Theorem 4-1-2

Let $h_i(x)t_i(x) = x^n-1$. The non-zero elements of the code M(i) generated by $t_i(x)$, belong to $(V_{i-1}-V_i)$.

Proof

Let $t_i'(x)$ be a non-zero element of the code generated by $t_i(x)$. Since $G(x)h_1(x)h_2(x)\ldots h_{i-1}(x)$ divide $t_i'(x)$, then $t_i'(x) \in V_{i-1}$. Also since $t_i'(x)h_i(x) = 0$ modulo x^n-1, $t_i'(x) \notin V_i$.

Q.E.D.

Definition 4-1-2

Let $r(x) \in V_{i-1}$, a coset of V_i in V_{i-1} is defined as the set

$$(r(x) + V_i) = \{(r(x) + V_i(x), \forall\ V_i(x) \in V_i) .$$

Theorem 4-1-3

If $t_i'(x)$ and $t_i''(x)$ are non-zero elements of the code M(i), generated by $(x^n-1)/h_i(x) = t_i(x)$, such that $t_i'(x) \neq t_i''(x)$, then the sets $(t_i'(x) + V_i)$ and $(t_i''(x) + V_i)$ are subsets of $(V_{i-1}-V_i)$ and non-intersecting.

Proof

By theorem 4-1-2 the elements $t_i'(x)$ and $t_i''(x)$ belong to $(V_{i-1}-V_i)$ so that the sets $(t_i'(x) + V_i)$ and $(t_i''(x) + V_i)$ also belong to $(V_{i-1}-V_i)$. Let us then assume that there exists two distinct elements of V_i, $v_i'(x)$ and $v_i''(x)$ such that $t_i'(x) + v_i'(x) = t_i''(x) + v_i''(x)$, then $t_i'(x) + t_i''(x) = v_i'(x) + v_i''(x)$. Since $h_i(x)$ divides $(v_i'(x) + v_i''(x))$, then $h_i(x)$ must also divide $(t_i'(x) + t_i''(x))$. This in turn implies that $v_i'(x) = v_i''(x)$ and $t_i'(x) = t_i''(x)$, in contradiction of our assumption, so that the cosets are distinct.

Q.E.D.

Theorem 4-1-4

The set $(V_{i-1}-V_i)$ is the union of all the distinct cosets of the form $(t_i(x)a(x) + V_i)$, where $t_i(x)a(x)$ is an element of the code M(i) generated by $t_i(x) = (x^n-1)/h_i(x)$ and $a(x)$ is any non-zero polynomial of degree less than m_i, the degree of $h_i(x)$.

Proof

Let the code V_{i-1} contain 2^k elements. The code V_i will then contain 2^{k-m_i} elements and $(V_{i-1}-V_i)$ will contain $2^k-2^{k-m_i}$ elements. From theorem 4-1-3, all the non-zero elements of the code M(i) generate distinct cosets of V_i in V_{i-1}. Furthermore every such coset contains 2^{k-m_i} belonging to $(V_{i-1}-V_i)$. The result of the theorem follows upon noting that the total number of elements in these cosets is

$$(2^{m_i}-1)2^{k-m_i} = 2^k - 2^{k-m_i} = |(V_{i-1}-V_i)|$$

Q.E.D.

We have so far shown that the elements of $(V_{i-1}-V_i)$ could be obtained by generating cosets of V_i, using as coset leader all the non-zero elements of the code having as generator polynomial $(x^n-1)/h_i(x)$. We now wish to determine the cyclic order of the elements of $(V_{i-1}-V_i)$.

Theorem 4-1-5

Let e_i be the exponent of the irreducible polynomial $h_i(x)$. If $m(x) \in (V_{i-1}-V_i)$, then $m(x)$ has cyclic order ℓe_i, where ℓ is a positive integer such that ℓe_i divides n.

Proof

Let e be the cyclic order of an element $m(x)$ belonging to $(V_{i-1}-V_i)$. By theorem 4-1-4, we can write $m(x) = t_i'(x) + v_i(x)$, where $t_i'(x)$ belongs to M(i) and $v_i(x)$ to V_i. Because the spaces V_i and M(i) are orthogonal, $e = \ell e_i$ by lemma 3-2-1, and $\ell e_i | n$ by theorem 3-1-1.

Q.E.D.

We note that the number of elements of order ℓe_i in $(V_{i-1}-V_i)$ can be found by an application of theorem 3-1-2. The number of elements of order ℓe_i in the code V_i is found and subtracted from the number of elements of order ℓe_i in the code V_{i-1}. The following theorems answer the question as to the distribution of those elements having cyclic order ℓe_i amongst the various cosets of $(V_{i-1}-V_i)$.

Theorem 4-1-6

Let $m(x) \in (V_{i-1}-V_i)$ and have cyclic order ℓe_i. The elements $m(x), x^{e_i}m(x), x^{2e_i}m(x), \ldots, x^{(\ell-1)e_i}m(x)$ all belong to the same coset of V_i in V_{i-1}.

Proof

By theorem 4-1-4, we can express $m(x)$ as $m(x) = t_i'(x) + v_i(x)$, where $t_i'(x)$ belongs to $M(i)$, $v_i(x)$ to the space V_i. Since $t_i'(x)$ has cyclic order e_i and $m(x)$ ℓe_i, it is clear that

$$m^{qe_i}(x) = x^{qe_i} t_i'(x) + x^{qe_i} v_i(x) = t_i'(x) + x^{qe_i} v_i(x)$$

belong to the coset $(t_i'(x) + V_i)$ for $q = 0,1,2,\ldots,(\ell-1)$.

Theorem 4-1-7

Let $m(x) \varepsilon (V_{i-1}-V_i)$ and have cyclic order ℓe_i. The elements $m(x), xm(x), x^2m(x), x^3m(x), \ldots, x^{(e_i-1)}m(x)$ all belong to distinct cosets of V_i in V_{i-1} that are cyclic shifts of each other.

Proof

By theorem 4-1-4, we can express $m(x)$ as $m(x) = t_i'(x) + v_i(x)$, where $t_i'(x)$ belongs to $M(i)$ and $v_i(x)$ to the space V_i. Also, since all the elements in $(V_{i-1}-V_i)$ by theorem 4-1-5 must have a cyclic order which is a multiple of e_i, the cyclic order of $t_i'(x)$, the n-tuples $m(x), xm(x), x^2m(x), \ldots, x^{(e_i-1)}m(x)$ will belong to distinct cosets whose coset leaders are of the form $x^j t_i'(x)$, $0 \leq j < e_i$, and hence are cyclic shifts of one another.

Q.E.D.

In the last two theorems, we have answered the question of the distribution of the elements of a cycle in $(V_{i-1}-V_i)$. We now look into the problem of generating representatives of the cycles of $(V_{i-1}-V_i)$.

The complexity of this problem is greatly reduced if we know that all the cycles in $(V_{i-1}-V_i)$ have order n. In this case we have the following:

Theorem 4-1-8

Let $t_i(x) = (x^n-1)/h_i(x)$ generate the code M(i) whose elements have cyclic order n. If $(a(x)E_i(x))^j$, $j = 1,2,\ldots,(2^{m_i}-1)/n$ denote representatives from all the cycles in M(i), where m_i is the degree of $h_i(x)$, then

$$\{(a(x)E_i(x))^j + V_i\}\ ,\quad j = 1,2,3,\ldots,(2^{m_i}-1)/n$$

generates representatives of all the cycles in $(V_{i-1}-V_i)$.

Proof

From theorem 4-1-4, all the elements of $(V_{i-1}-V_i)$ are contained in the proper cosets of V_i in V_{i-1}, and having as cosets leaders the non-zero codewords of M(i). Now since $h_i(x)$ has exponent n, from theorem 4-1-5 and 4-1-7, all the elements of a coset must belong to distinct cycles. Also, as a further result of theorem 4-1-7, no two cosets whose coset leaders belong to distinct cycles in M(i) may contain elements of the same cycle in $(V_{i-1}-V_i)$. Finally by theorem 3-2-1, $(a(x)E_i(x))^j$, $j = 1,2,\ldots,(2^{m_i}-1)/n$, generates representatives from all of the cycles in M(i). Q.E.D.

In summary, what has so far been shown, is that any (n,k) cyclic code can be partitioned into subsets of the form $(V_{i-1}-V_i)$ (theorem 4-1-1).

Furthermore, $(V_{i-1}-V_i)$ can be expressed as the union of the proper cosets of V_i in V_{i-1}, having as coset leaders the non-zero elements of the code generated by $(x^n-1)/h_i(x)$ (theorem 4-1-4). We then went on to show that the elements of $(V_{i-1}-V_i)$ must have as cyclic order a multiple of e_i, the exponent of $h_i(x)$ (theorem 4-1-5). In theorems 4-1-6 and 4-1-7, we established the distribution of the elements belonging to the same cycle in the various cosets partitioning $(V_{i-1}-V_i)$. Finally, in theorem 4-1-8 a method for obtaining all the cycles of $(V_{i-1}-V_i)$ was given for the special case $e_i = n$. We note in passing that theorem 4-1-8 is a generalization of a result of Seguin[22] who considered the case for those cyclic codes of length $n = 2^q-1$. We now apply the results to a particular example.

Example 4-1-1

We wish to determine a set of cycle representatives for a (21,9) cyclic code having as generator polynomial,

$$G(x) = (1 + x)(1 + x + x^2)(1 + x^2 + x^3)(1 + x^2 + x^4 + x^5 + x^6),$$

and as parity check polynomial,

$$H(x) = (1 + x + x^2 + x^4 + x^6)(1 + x + x^3) = h_1(x)h_2(x).$$

We also have $G(x)H(x) = x^{21} + 1$, and

degree of $h_1(x) = 6$, exponent of $h_1(x) = 21$,

degree of $h_2(x) = 3$, exponent of $h_2(x) = 7$.

Step 1 - Finding cycle representatives in (V_0-V_1)

a) $|(V_0-V_1)| = 2^9-2^3 = 512 - 8 = 504$,

b) by theorem 4-1-5, all the elements in (V_0-V_1) have cyclic order 21,

c) we find a representative of the cycles in the code generated by $(x^{21} + 1)/h_1(x) = 1 + x + x^3 + x^6 + x^7 + x^{10} + x^{13} + x^{15}$.

These are:

$$t_1(x) = 1 + x + x^3 + x^6 + x^7 + x^{10} + x^{13} + x^{15}$$

$$t_2(x) = 1 + x^2 + x^3 + x^4 + x^6 + x^8 + x^{10} + x^{11} + x^{13} + x^{14} + x^{15} + x^{16}$$

$$t_3(x) = 1 + x + x^4 + x^5 + x^6 + x^7 + x^8 + x^9 + x^{11} + x^{12} + x^{16} + x^{20}$$

Every cycle contains 21 elements which accounts for the 63-non-zero elements in the code generated by $t_1(x)$.

c) The cycle representatives of (V_0-V_1) are then given by the cosets $(t_1(x) + V_1),(t_2(x) + V_1),(t_3(x) + V_1)$, where V_1 is the code generated by $G(x)(1 + x + x^2 + x^4 + x^6)$.

$$(t_1(x) + V_1) =$$

$$(1 + x + x^2 + x^3 + x^4 + x^5 + x^7 + x^9 + x^{10} + x^{11} + x^{12} + x^{15} + x^{16} + x^{18} + x^{19} + x^{20},$$

$$1 + x^4 + x^5 + x^6 + x^7 + x^8 + x^{11} + x^{12} + x^{13} + x^{17} + x^{18} + x^{19},$$

$$x + x^2 + x^4 + x^6 + x^9 + x^{11} + x^{13} + x^{14} + x^{15} + x^{16} + x^{17} + x^{18},$$

$$x^2 + x^3 + x^5 + x^6 + x^8 + x^9 + x^{10} + x^{12} + x^{13} + x^{14} + x^{16} + x^{19},$$

$$1 + x + x^3 + x^6 + x^7 + x^{10} + x^{13} + x^{15},$$

$$1 + x^2 + x^7 + x^8 + x^9 + x^{16} + x^{17} + x^{20},$$

$$x + x^5 + x^{12} + x^{14} + x^{15} + x^{17} + x^{19} + x^{20},$$

$$x^3 + x^4 + x^8 + x^{10} + x^{11} + x^{14} + x^{18} + x^{20}).$$

$$(t_2(x) + V_1) =$$

$$(1 + x^2 + x^3 + x^4 + x^6 + x^8 + x^{10} + x^{11} + x^{13} + x^{14} + x^{15} + x^{16},$$

$$1 + x + x^2 + x^5 + x^6 + x^{12} + x^{13} + x^{14} + x^{16} + x^{17} + x^{18} + x^{19},$$

$$1 + x^3 + x^5 + x^8 + x^9 + x^{10} + x^{12} + x^{14} + x^{15} + x^{18} + x^{19} + x^{20},$$

$$x^2 + x^4 + x^5 + x^7 + x^8 + x^{11} + x^{12} + x^{15} + x^{16} + x^{17} + x^{19} + x^{20},$$

$$x + x^3 + x^4 + x^5 + x^6 + x^7 + x^9 + x^{10} + x^{11} + x^{12} + x^{13} + x^{19},$$

$$1 + x + x^4 + x^9 + x^{11} + x^{14} + x^{17} + x^{20},$$

$$x^6 + x^7 + x^8 + x^9 + x^{13} + x^{15} + x^{17} + x^{18},$$

$$x + x^2 + x^3 + x^7 + x^{10} + x^{16} + x^{18} + x^{20}).$$

$(t_3(x) + V_1)$ =

$1 + x + x^4 + x^5 + x^6 + x^7 + x^8 + x^9 + x^{11} + x^{12} + x^{16} + x^{20}$,

$1 + x^3 + x^6 + x^7 + x^9 + x^{10} + x^{15} + x^{16} + x^{17} + x^{18} + x^{19} + x^{20}$,

$1 + x^2 + x^3 + x^4 + x^5 + x^7 + x^{10} + x^{11} + x^{12} + x^{13} + x^{15} + x^{17}$,

$x + x^2 + x^3 + x^5 + x^6 + x^8 + x^{10} + x^{12} + x^{14} + x^{17} + x^{18} + x^{20}$,

$x + x^3 + x^4 + x^8 + x^9 + x^{10} + x^{11} + x^{13} + x^{14} + x^{16} + x^{17} + x^{19}$,

$1 + x + x^2 + x^7 + x^8 + x^{13} + x^{18} + x^{19}$,

$x^5 + x^9 + x^{12} + x^{13} + x^{14} + x^{15} + x^{16} + x^{18}$,

$x^2 + x^4 + x^6 + x^{11} + x^{14} + x^{15} + x^{19} + x^{20})$.

Step 2 - Finding the cycles of $(V_1\text{-}V_2)$

a) $|(V_1\text{-}V_2)| = 2^3 - 2^0 = 7$,

b) by theorem 4-1-5 all the cycles in $(V_1\text{-}V_2)$ have order 7, hence there is only one cycle of which a representative is

$1 + x^2 + x^3 + x^4 + x^7 + x^9 + x^{10} + x^{11} + x^{14} + x^{16} + x^{17} + x^{18}$.

The weight distribution of the code is then:

280 elements have weight 12

210 elements have weight 8

21 elements have weight 16

1 element has weight 0 for a total of 512 elements.

4-2 The General Case

In the previous section, the problem of finding all the cycle representatives of a cycle code was solved for all cyclic codes whose

parity check polynomial factors into irreducible polynomials all having exponent n (theorem 4-1-8). To obtain a set of cycle representatives for those codes that have cycles of order less than n, we introduce another method. In fact, the entire cyclic class decomposition can be based on this approach since it will generate cycles of order n. Essentially the idea is to view a cyclic code as the direct sum of its minimal ideals as discussed in theorem 2-8-2. Goethals[17] has used this method in the case where the cyclic code is itself a minimal ideal, and he also suggested that the method could be extended to non-minimal ideals. However, we will show in this section that such an extension requires a further partitioning of the set $(V_{i-1}-V_i)$.

Let G(x) be the generator polynomial of an (n,k) binary cyclic code of odd length, having as parity check polynomial $H(x) = (x^n-1)/G(x) = h_1(x)h_2(x)\ldots h_r(x)$, where $h_i(x)$, $i = 1,2,3,\ldots,r$, are irreducible polynomials of degree m_i and exponent e_i. The following definitions are made by way of introducing the necessary notation.

$M^*(i)$ is the set of all non-zero elements of the minimal ideal M(i), generated by the polynomial $(x^n-1)/h_i(x)$ and having $E_i(x)$ as idempotent.

$\langle xE_i(x)\rangle$ is the cyclic group generated by $xE_i(x)$ and having e_i elements.

$a(x)E_i(x)$ is a generator of the factor group $M^*(i)/\langle xE_i(x)\rangle$.

V_i is the cyclic subcode having $G(x)h_1(x)\ldots h_i(x)$ as generator. With the above notation it is possible to re-write theorem 4-1-8 as follows:

Theorem 4-2-1

If $e_i = n$, the set

$$V_i + M^*(i)/\langle xE_i(x)\rangle,$$

consist of all the distinct cycle representatives of $(V_{i-1}-V_i)$.

Proof

This follows immediately upon noting that the set $M^*(i)/\langle xE_i(x)\rangle$ is the set of cycle representatives of the code generated by $(x^n-1)/h_i(x)$.

Q.E.D.

For a code whose parity check polynomial consists entirely of factors having exponent n, it is possible by repeated application of the above theorem to obtain all the cycles of the code. For those cases where $e_i \neq n$, the problem of finding the cycles of $(V_{i-1}-V_i)$ requires a further partitioning on the code.

As shown in Section 2-8, the set of elements consisting of the sum

$$M^*(i_1) + M^*(i_2) + \ldots + M^*(i_s) = M^*(i_1,i_2,\ldots,i_s)$$

is a group under multiplication modulo (x^n-1), of order

$$|M^*(i_1)||M^*(i_2)|\ldots|M^*(i_s)|.$$

Furthermore, it is a simple matter to show that all its cycles have period given by l.c.m. $(e_{i_1},e_{i_2},\ldots,e_{i_s})$. The technique as outlined in this section, will involve the partitioning of the set $(V_{i-1}-V_i)$ into such multiplicative groups and obtaining the cycle representatives of the

groups separately. The following theorem and its corollary characterizes those groups belonging to $(V_{i-1}-V_i)$.

Theorem 4-2-2

Let the codes V_{i-1} and V_i be generated by the polynomials $G(x)h_1(x)\ldots h_{i-1}(x)$ and $G(x)h_1(x)\ldots h_{i-1}(x)h_i(x)$ respectively.

$$(V_{i-1}-V_i) = M^*(i) + V_i.$$

Proof

Since by theorem 2-8-2, every ideal can be expressed as the sum of its minimal ideals, we must have

$$V_i = M(i+1) + M(i+2) + \ldots + M(r), \tag{4.1}$$

and

$$V_{i-1} = M(i) + M(i+1) + \ldots + M(r).$$

From the definition of $(V_{i-1}-V_i)$, it must be that $M^*(i) + V_i$ belongs to $(V_{i-1}-V_i)$. However, by the fact that $|M^*(i) + V_i| = (2^{m_i}-1)|V_i| = 2^{m_i}|V_i| - |V_i| = |V_{i-1}| - |V_i| = |(V_{i-1}-V_i)|$, it follows that $(V_{i-1}-V_i) = M^*(i) + V_i$. Q.E.D.

It is not hard to show that (4.1) can be re-written as

$$V_i = U\,(\delta_{i+1} M^*(i+1) + \delta_{i+2} M^*(i+2) + \ldots + \delta_r M^*(r)) \tag{4.2}$$

where δ_{i+j}, $j = 1,2,\ldots,r-i$, can be either 1 or 0, and U is the usual symbol for the union operation. However, we can now interpret (2.2) as equating V_i to the union of 2^{r-i} non-intersecting multiplicative groups, corresponding to the 2^{r-i} possible choices for $(\delta_{i+1},\delta_{i+2},\ldots,\delta_r)$.

For the purpose of simplifying the notation, we write

$$\delta_{i+1} M^*(i+1) + \delta_{i+2} M^*(i+2) + \ldots + \delta_r M^*(r) = M^*(i_1,i_2,\ldots,i_{(r-i)}),$$

where

$$i_j = i + j \quad \text{if } \delta_{i+j} = 1$$

$$= 0 \qquad \text{if } \delta_{i+j} = 0, \text{ for } j = 1,2,\ldots,(r-i).$$

To avoid any ambiguity, $M^*(0,0,\ldots,0)$ is defined as the all-zero codeword, and e_0 its period is 1. We have

Corollary 4-2-1

$(V_{i-1}-V_i) = U\, M^*(i,i_1,i_2,\ldots,i_{(r-i)})$, for the 2^{r-i} possible choices of $(i_1,i_2,i_3,\ldots,i_{(r-i)})$.

Proof

By theorem 4-2-2, $(V_{i-1}-V_i) = M^*(i) + V_i$. Furthermore since

$$V_i = U\, M^*(i_1,i_2,\ldots,i_{(r-i)})$$

for the 2^{r-i} possible choices of $(i_1,i_2,\ldots,i_{(r-i)})$, the corollary follows upon noting that

$$M^*(i) + M^*(i_1,i_2,i_3,\ldots,i_{(r-i)}) = M^*(i,i_1,i_2,\ldots,i_{(r-i)}).$$

Having characterized those groups which define a partition on $(V_{i-1}-V_i)$, we now derive an expression for their cycle representatives. Let $c_1(x),c_2(x),\ldots,c_N(x)$, be the cycle representatives of the group $M^*(i_1,i_2,\ldots,i_{(r-i)})$ all having order e. The following theorem enables us to obtain the cycle representatives of the group $M^*(i) + M^*(i_1,i_2,\ldots, i_{(r-i)})$.

Theorem 4-2-3

The cycle representatives of the group $M^*(i) + M^*(i_1,i_2,\ldots,i_{(r-i)})$, are given by

$$x^{\ell}c_k(x) + (a(x)E_i(x))^j, \tag{4.3}$$

$a(x)\ E_i(x)$ is a generator of $M^*(i)/\langle xE_i(x)\rangle$,

$j = 1,2,3,\ldots,(2^{m_i}-1)/e_i$,

$k = 1,2,3,\ldots,N$,

$\ell = 0,1,2,\ldots,(g.c.d.(e,e_i) - 1)$.

Proof

The number of elements obtained from (4.3) is

$$|M^*(i_1,i_2,\ldots,i_{(r-i)})||M^*(i)|g.c.d.(e,e_i)/ee_i =$$

$$|M^*(i,i_1,i_2,\ldots,i_{(r-i)})|/l.c.m.(e_i,e_{i_1},e_{i_2},\ldots,e_{i_{(r-i)}}).$$

which corresponds to the number of cycles in the group $M^*(i) + M^*(i_1,i_2,\ldots,i_{(r-i)})$. We need therefore only to prove that all the elements, obtained through (4.3), in fact belong to distinct cycles. We shall prove this by contradiction.

Let the two distinct elements $(a(x)E_i(x))^{j_1} + x^{\ell_1}c_{s_1}(x)$ and $(a(x)E_i(x))^{j_2} + x^{\ell_2}c_{s_2}(x)$, obtained through (4.3), belong to the same cycle. This means that there exists a t, such that

$$x^t(a(x)E_i(x))^{j_1} + x^{t+\ell_1}c_{s_1}(x) = (a(x)E_i(x))^{j_2} + x^{\ell_2}c_{s_2}(x) \qquad (4.4)$$

However, because of the orthogonality of the minimal ideals (4.4) cannot be valid if $j_1 \neq j_2$ and/or $s_1 \neq s_2$. Therefore (4.4) reduces to

$$x^t(a(x)E_i(x))^{j} + x^{t+\ell_1}c_{s}(x) = (a(x)E_i(x))^{j} + x^{\ell_2}c_{s}(x) \qquad (4.5)$$

Further, since $a(x)E_i(x)c_s(x) = 0$ modulo (x^n-1), we must have $t = qe_i$. Using this fact in (4.5), we get

$$c_s(x)(x^{qe_i+\ell_1-\ell_2} - 1) = 0.$$

Here, because $c_s(x)(x^e-1) = 0$, we must have $qe_i + \ell_1 - \ell_2 = be$. But g.c.d.$(e,e_i)$ divides e and g.c.d.(e,e_i) also divides e_i, so that it follows that g.c.d.(e,e_i) also divides $|\ell_1 - \ell_2|$. Here $|\ell_1 - \ell_2| <$ g.c.d.(e,e_i), since $0 \leq \ell <$ g.c.d.(e,e_i). Thus ℓ_1 has to be equal to ℓ_2. This implies that the two elements chosen were the same and this is a contradiction, hence the theorem. Q.E.D.

Theorem 4-2-1 is in fact a special case of theorems 4-2-2, 4-2-3 and corollary 4-2-2, when $e_i = n$. Furthermore, from theorem 4-2-3 and corollary 4-2-2, it is seen that when $\text{l.c.m.}(e_i, e_{i_1}, e_{i_2}, \ldots, e_{i_{(r-i)}}) = e_i e_{i_1} e_{i_2} \ldots e_{i_{(r-i)}}$, the cycles of $(V_{i-1}-V_i)$ are given simply by

$$M^*(i)/\langle xE_i(x)\rangle + \delta_{i+1} M^*(i+1)/\langle xE_{i+1}(x)\rangle \ldots + \delta_r M^*(r)/\langle xE_r(x)\rangle,$$

where the δ's are either 1 or 0.

4-3 An Algorithm for Finding Cycle Representatives

In this section, we present an algorithm for obtaining the cycle representatives of a cyclic code, based on the preceeding results. Let the code be generated by the polynomial $G(x) = (x^n-1)/H(x)$, and $H(x) = h_1(x)h_2(x)\ldots h_r(x)$ be the prime factorization of $H(x)$. We proceed to obtain representatives of the cycles in the sets $(V_{i-1}-V_i)$, $i = 1,2,\ldots,r$, as follows.

If the exponent of $h_i(x)$ is n, theorem 4-2-1 is applied. If not, the cycle representatives of $(V_{i-1}-V_i)$ must be found through an application of theorems 4-2-2 and 4-2-3. Now theorem 4-2-3, as stated, is clearly iterative, so that an expression for the cycle representatives of the groups belonging to $(V_{i-1}-V_i)$ can be obtained in terms of the primitive elements of the minimal ideals of the code V_i, and that of the group $M^*(i)$. By way of example, let $M^*(i,i_1,i_2)$ belong to $(V_{i-1}-V_i)$. From theorem 4-2-3, $(a_{i_1}(x)E_{i_1}(x))^{j_1} + x^{\ell_1}(a_{i_2}(x)E_{i_2}(x))^{j_2}$ defines all the cycles of $M^*(i_1,i_2)$, where $j_1 = 1,2,\ldots,(2^{m_{i_1}}-1)/e_{i_1}$,

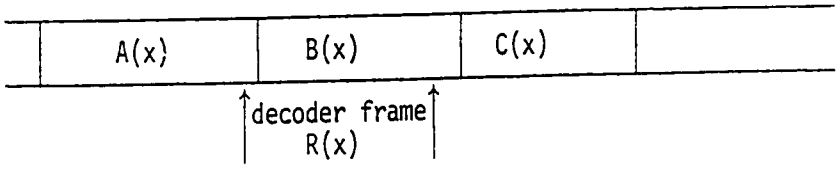
$j_2 = 1,2,\ldots,(2^{m_{i_2}}-1)/e_{i_2}$ and $\ell_1 = 0,1,2,\ldots,(\text{g.c.d.}(e_{i_1},e_{i_2}) - 1)$. Again applying theorem 4-2-3, the cycles of $M^*(i,i_1,i_2)$ are given by $(a(x)E_i(x))^{j_i} + x\,(a(x)E_{i_1}(x))^{j_1} + x^{\ell+\ell_1}\,(a_{i_2}(x)E_{i_2}(x))^{j_2}$ where j_1, ℓ_1 and j_2 are as previously defined, and $j_i = 1,2,\ldots,(2^{m_i}-1)/e_i$, $\ell = 0,1,2,\ldots(\text{g.c.d.}(e_i,e_{i_1},e_{i_2}) - 1)$. In this manner, all the cycles of the code may be obtained.

We now consider some applications of the theory of cycle representatives to the problem of cyclic code synchronization.

CHAPTER 5

APPLICATIONS TO THE SYNCHRONIZATION OF CYCLIC CODES

In the process of channel encoding using an (n,k) block code, an information sequence is partitioned into subsequences of k bits and encoded or mapped into sequences of n bits which are then transmitted over the channel. For the receiver to decode correctly, a knowledge of the beginning of each block in the received data stream is required. However, it may happen that the receiver will assume the wrong bit position for the start of a block resulting in a slip error or synchronization loss. Since we will only be considering the synchronization problem associated with cyclic codes, the following mathematical description of slip error will be employed. We let A(x),B(x),C(x) be codewords that are transmitted end to end as illustrated in the figure below.



At the receiver, because of a synchronization loss, the n-tuple R(x) is framed by the decoder. If we assume a left slip of s bits, we can then express R(x) as

$$R(x) = x^{s}B(x) + U_{s}(x) \quad \text{modulo } (x^{n}-1),$$

where $U_{s}(x)$ is an arbitrary polynomial of degree less than s which arises

from the s higher terms of A(x) which are framed and the s higher terms of B(x) that are not framed. Similarly for a right slip of r bits we would have

$$R'(x) = x^{-r}B(x) + U_{-r}(x) \quad \text{modulo } (x^n-1).$$

Here, $U_{-r}(x)$ is an arbitrary polynomial of degree less than r which arises from the r lower terms of C(x) that are framed and the r lower terms of B(x) that are not framed. Also, if we assume a noisy channel, then for a left slip we would have,

$$R(x) = x^{s}B(x) + U_{s}(x) + \zeta(x),$$

where $\zeta(x)$ is an additive error pattern. A solution to the synchronization problem will involve finding a technique that will enable the decoder to extract the codeword B(x) from the framed n-tuple R(x).

The chapter is divided into three sections. In the first section, we discuss briefly some of the better known and more powerful synchronization recovery techniques. The second and third sections of the chapter are concerned with the application of the theory of cycle representatives to the synchronization problem associated with the use of cyclic codes. Specifically, knowledge of cycle representatives is used to generalize the Subset Code technique of Tavares and Fukada[23] as well as the synchronization technique of Bose, Caldwell and Weldon[24,52]. The resulting generalization considerably simplifies the interpretation of the synchronization process and leads to more efficient codes, from the point of view of information rate, capable of synchronization recovery.

Finally, tables are included which compare the various synchronization techniques under varying conditions of slip and additive errors.

5-1 A Discussion of Synchronization Techniques

The following discussion is limited to some of the better known synchronization techniques that apply to cyclic codes. These, may be divided into two broad classes, namely, those that alter the length of the code, and those that do not. Among the former we will discuss the "extended cyclic codes" developed by Bose and Caldwell[24] and later generalized by Weldon[54], hereafter called the BCW technique. Tong[42], examined a scheme based on shortening cyclic codes. In a shortened cyclic code, some of the information bits are preassigned (usually set to zero) and are not transmitted. Knowledge of this fact can then be used at the decoder to recover synchronization. In contrast, there exists a number of techniques that do not alter the length of the code. Amongst these, we will discuss briefly the coset code technique[43,53,42], Mandelbaum's techniques[44,46] and the subset code technique[23]. The results obtained apply to three different types of channels, namely, the noiseless channel where slip error may occur, a so-called Type 1 channel where both slip and additive error may occur but not simultaneously, and a Type 2 channel (or noisy channel) where slip and additive error can occur simultaneously. In this section we content ourselves with merely quoting some of the more important results for the purpose of comparison, the proofs may be found in the given references. Finally, it will be readily noted that the advantages of one technique over

another are not always obvious, the choice of any one method for frame synchronization would depend on a number of factors such as constraints on the word length, characteristics of the channel, soft or hardware implementation, etc., and this should be kept in mind when examining the tables at the end of the chapter.

Mandelbaum's Technique[46]

This technique which does not alter the length of the codewords is extremely simple and very effective. To illustrate, let us assume that the vector $(a_0,a_1,a_2,\ldots,a_{n-1})$ is a codeword from an (n,k) cyclic code. Further, let the digits $a_0,a_1,a_2,\ldots,a_{k-1}$ represent the k information digits in the codeword. Let us then restrict the first 2S+1 digits as shown below

$$\underset{\longleftarrow S+1 \longrightarrow}{0,0,\ldots,0,} \quad \underset{\longleftarrow S \longrightarrow}{1,1,\ldots,1} \quad a_{2S+1},\ldots,a_{k-1}, \quad 2S+1 \leq k.$$

Before transmission, codewords are shifted (S+1) digits to the left. Using such a code, Mandelbaum proved the following:

Theorem 5-1-1

Given a t error-correcting (n,k) cyclic code, there exists an (n,k-2S-1) code that can correct the simultaneous occurrence of t or less additive errors and S or less digits of slip.

The proof for the theorem is based on the fact that a shift of S or less digits at the receiver will still yield a valid codeword. If no more than t errors have occurred in the framed n-tuple, then once these have been removed the sequence of 1's and 0's may be located and

synchronization restored. Note that the price paid to correct for S digits of slip error is 2S+1 digits.

Fibonacci Codes[44]

This technique which is also due to Mandelbaum is similar to his first technique in that frame synchronization is dependent on the recognition of a string of 1's in a framed n-tuple free of additive errors. In this method, all codewords will have the first s+q+2 information digits in the form

$$\underbrace{1,1,\ldots,1}_{\longleftarrow S+1 \longrightarrow}\quad 0,d_1,d_2,\ldots,d_q,\quad a_{q-s-2},a_{q-s-3},\ldots,a_{n-1},$$

where a string of (s+1) ones is followed by a zero which in turn is followed by the Fibonacci sequence $d_1,d_2,\ldots,d_q$. The Fibonacci sequence (d_i), i = 1,2,...,q, has the property that it contains no string of (s+1) ones or more, and is a transformation on (q-1) information digits with a redundancy of 1 digit. An (n,k) cyclic code in which the first s+q+2 information digits are of the above form will be called an F(n,k,s,q) Fibonacci code. If all the codewords of an F(n,k,s,q) code are shifted by h = [(s+2)/2]* digits to the left before transmission, it is not too difficult to see that a slip of s or less digits at the receiver can be detected, by the recognition of the string of s ones in the framed n-tuple free of additive errors.

The following theorems describe the performance of an F(n,k,s,q) code.

---

* $[y]$ *represents the integral part of* y.

A-Slip in a noisy channel

Theorem 5-1-2

An F(n,k,s,q) code can detect any combination of S digits of slip and t additive errors if $t + S - h < d$ when $S - h \geq 0$, or if $t < d$ when $S - h < 0$, where d is the distance of the cyclic code and $S < q+1$.

Theorem 5-1-3

An F(n,k,s,q) code can detect $S = \min(q+1, [(d-1)/2] - t - h)$ or less digits of slip when t additive errors occur simultaneously. If no slip occurs, the code can correct (d-1)/2 additive errors.

Theorem 5-1-4

An F(n,k,s,q) code can correct any combination of t additive error and S or less digits of slip if $t + (S-h) \leq [(d-1)/2]$ where S is bounded by

$$S = [(q-1)/2] - 1 \quad \text{if } q \text{ is odd,}$$

$$S = [(q-2)/2] - 1 \quad \text{if } q \text{ is even,}$$

provided that $S - h \leq [(d-1)/2]$. The additive noise correction is that of the original (n,k) code.

B - Slip in type 1 channel

Theorem 5-1-5

In a type 1 channel, an F(n,k,s,q) code can correct slip of

$$S = [(q-1)/2] - 1 \quad \text{if } q \text{ is odd,}$$

$$S = [(q-2)/2] - 1 \quad \text{if } q \text{ is even,}$$

provided that $S - h \leq [(d-1)/2]$. The additive noise correction is that of the original (n,k) code.

The coset code technique[42, 43, 53]

The coset code technique involves transmitting a suitable coset of the selected (n,k) cyclic code, and as such the only additional operation involved at the transmitter is that of adding a fixed n-tuple

to every codeword. The choice of the coset leader is dependent upon the channel and whether it is desired to detect or correct the errors at the receiver. The following theorems summarize the known ability of coset codes to handle slip error for a variety of channels.

A-slip in a noiseless channel.

Theorem 5-1-6

Given any (n,k) cyclic code, there exists a coset code such that the decoder can determine both the magnitude and direction of any slip not exceeding (n-k-2)/2, by examining only the syndrome of the received n-tuple.

The coset leader chosen in the above theorem is $c(x) = 1 + x^{n-1}$. The following theorems describe the performance of coset codes over a type 1 channel.

B-slip in a type 1 channel.

Theorem 5-1-7

Given an (n,k) cyclic code and a type 1 channel, the coset generated by $c(x) = 1$ can detect a slip of n-k-1 or less digits, or any additive error pattern that the original (n,k) code could detect.

Theorem 5-1-8

Given an (n,k) cyclic code and a type 1 channel, the coset code generated by

$$c(x) = \sum_{i=0}^{t/2} x^{i(S-1)} + (t-2[t/2])x^{n-1}$$

can correct t or less additive errors or S or less digits of slip, where S is given by

$$S = \min(\max(S_1, S_2), S_3), \text{ with}$$
$$S_1 = d - 2(t-1)$$
$$S_2 = [(3d-5)/2] - 3t - [t/2]$$
$$S_3 = [(n-k-t - [t/2] - 1)/([t/2] + 2)].$$

For a type 2 channel, the next two theorems describe the ability of coset codes to handle slip and additive errors.

C-slip in a noisy channel.

Theorem 5-1-9

Given any (n,k) cyclic code, there exists a coset code that can detect the simultaneous occurrence of t or less additive errors and S or less bits of slip in any received n-tuple, where S is given by

$$S = \min[\max(S_1, S_2), S_4]$$

where S_1 and S_2 are as defined in theorem 5-1-8, and

$$S_4 = [(n-5 - t/2 - 1)/(t/2 - 1)].$$

Theorem 5-1-10

Given any (n,k) cyclic code, the coset code having as coset leader

$$c(x) = x^{n-1} + \sum_{i=0}^{t} x^{i(2S+1)}$$

can correct both t or less additive errors, and S or less bits of slip even when they occur simultaneously in every received n-tuple, where S is given by

$$S = \min([d-4t-3/2],\ [n-t-2/2(t-1)]).$$

We now discuss two techniques that have special relevance to this thesis.

The BCW technique[24, 52]

This is one of the techniques that alters the length of the code-words of an (n,k) cyclic code. To illustrate how this is done, let $G(x) = (x^n-1)/H(x)$ be the generator polynomial of the code. Further, let h(x) be an irreducible factor of H(x) having exponent e, and $(a_0,a_1,a_2,\ldots,a_{n-1})$ a codeword of the form

$$a(x) = (h(x)i(x) + 1)G(x),$$

where i(x) is an arbitrary polynomial of deg. < k-m, m being the degree of h(x). The above word is then extended as

$$(a_{n-S},\ldots,a_{n-1},a_0,a_1,\ldots,a_{n-1},a_0,a_1,\ldots,a_{S-1}).$$

The following theorems describes the error-correcting capability of the BCW technique.

A - Slip in Noisy Channel

Theorem 5-1-11

An (n,k) cyclic code whose parity check polynomial has a factor of degree m and exponent e can be extended to form an (n+2S,k-m) code that can correct the simultaneous occurrence of S or less digits of slip and any additive error pattern that the original cyclic code could correct, where e > 2S.

If we are interested only in detecting a loss of synchronization, we have

Theorem 5-1-12

Given an (n,k) cyclic code whose parity check polynomial has a factor of degree m and exponent e, there exists an (n+2S,k-m) extended code that can correct any additive error pattern the original cyclic code could correct, and in addition detect the simultaneous occurrence of S or less digits of slip, where e > S.

In the BCW technique, codewords are so formed, that a slip of S or less digits will permit the decoder to frame a word from the original cyclic code. It will be shown in the next section that the BCW technique also allows for choosing one element from a number of cycles in the code and that it is this property that allows the decoder to regain synchronization.

The subset code technique[23]

This technique was inspired by the BCW technique and in fact could be considered a special case of the latter. The essential differ-

ence being that in the subset code technique no extra symbols are added on at the end of the codewords. As in the BCW case, every codeword is of the form

$$c(x) = (h(x)i(x) + 1)G(x).$$

Here again, G(x) is the generator polynomial of the original (n,k) cyclic code, and h(x) is an irreducible factor of H(x), the parity check polynomial, of degree m and exponent e. The information polynomial i(x) is arbitrary and of degree less than k-m. We have a subcode of the original code, and the following theorems summarizes the ability of such codes to handle slip and additive errors.

A - Slip in a noiseless channel.

Theorem 5-1-13

Given an (n,k) cyclic code whose parity check polynomial has a factor of degree m and exponent e, there exists an (n,k-m) subset code having e > S that can detect S or less digits of slip if the channel has no additive errors, provided $S \leq n-k$.

Theorem 5-1-14

Given an (n,k) cyclic code whose parity check polynomial has a factor of degree m and exponent e, there exists an (n,k-m) subset code having e > 2S that can correct S or less digits of slip if the channel has no additive errors, provided $S \leq [(n-k)/2]$.

B - Slip in a type 1 channel

Theorem 5-1-15

Given an (n,k) cyclic code whose parity check polynomial has a factor of degree m and exponent e, and a type 1 channel, there exists a (n,k-m) subset code with $e > 2S$ that can correct S or less digits of slip and any additive error pattern that the original cyclic code could correct, provided that $S \leq [(d-1)/2]$, where d is the minimum distance of the (n,k) cyclic code.

C - Slip in a noisy channel

Theorem 5-1-16

Given an (n,k) cyclic code whose parity check polynomial has a factor of degree m and exponent e, there exists an (n,k-m) subset code having $e \geq d$ that can detect any combination of s digits of slip and t additive errors if $t + s \leq d-1$.

Theorem 5-1-17

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

Theorem 5-1-18

Given a t-error-correcting (n,k) cyclic code whose parity check polynomial has a factor of degree m and exponent e, there exists an

an (n,k-m) subset code with e > 2S that can correct any combination of b additive errors and s digits of slip if b + s $\leq$ t.

Some other synchronization techniques

Tong, who carried out some of the early work in the area of coset codes has also studied schemes based on shortening cyclic codes[42]. In a shortened cyclic code, some of the information bits are preassigned (usually set to zero) and are not transmitted. Knowledge of this fact can then be used by the decoder. Shiva and Seguin[45] have also proposed synchronization recovery schemes. They present a scheme that shortens the codewords, and another, which they call a "modified technique", that does not alter the word length. The modified technique technique appears to be the more efficient of the two, being more efficient than the coset codes for correcting slip in a type 2 channel but less efficient than Mandelbaum's technique (which it resembles) or the subset codes. Some of their results appear in the tables.

We note that all the theorems quoted in this section have been used in the compilation of the tables appearing at the end of the chapter. Although we have offered no proofs, these may be found in the references. These theorems also provide a mean of comparison for the various schemes discussed, complementing the data in the tables.

We now consider the application of the theory of cycle representatives to the synchronization recovery problem.

5-2 Generalized Subset Codes

Tavares and Fukada[23] introduced a class of synchronizable error correcting codes which they called subset codes. Every codeword of such a code is of the form

$$c(x) = (h(x)i(x) + 1)G(x), \tag{5-1}$$

where $G(x)$ is the generator polynomial of an (n,k) cyclic code, $h(x)$ is an irreducible factor of $(x^n-1)/G(x)$ of degree m and exponent e and $i(x)$ is any information polynomial of degree less than $k-m$. They then proceeded to show how subset codes can be used to correct for slip and additive errors.

In this section, the concept of subset codes will be generalized to include a larger number of codewords than that obtained from Eqn. (5-1) without losing any of their slip and additive error-correcting capability. These codes will be called generalized subset codes (GSC) and will be defined over GF(2).

The basic idea underlying the construction of the GSC codes lies in the recognition of the fact that codewords obtained through (5-1) belong to (V_0-V_1) and hence to a coset of V_1 in V_0 (theorem 4-1-4). It will be shown that there exist many such cosets whose union gives rise to a code which also possesses the synchronization property of the subset codes.

The codewords of a GSC are of the form

$$w(x) = \left\{G(x)h_1(x)i(x) + t_1^j(x)\right\} x^{(aS + 1)i}, \tag{5-2}$$

where G(x) is the generator polynomial of an (n,k) cyclic code, $h_1(x)$ is an irreducible factor of $(x^n-1)/G(x)$ of degree m_1 and exponent e_1, $t_1(x)$ is a primitive element of M(1), the minimal ideal code generated by $(x^n-1)/h_1(x)$ and i(x) is an information polynomial of degree less than $k-m_1$. Also, $j = 1,2,\ldots,(2^{m_1}-1)/e_1$, $i = 0,1,2,\ldots,([e_1/(aS+1)] - 1)$, a = 1 for detection purposes only and a = 2 for correction purposes. As usual [y] signifies the integer part of y and S is the maximum number of slip errors to be detected or corrected.

Clearly, all the words given by (5-2) belong to (V_0-V_1) by theorem 4-1-4. Also since all the elements of (V_0-V_1) have as cyclic order a multiple of e_1 (theorem 4-1-5), the above construction assures us that the elements chosen from the same cycle will be separated by at least (aS+1) shifts (theorems 4-1-6 and 4-1-7) which is the necessary condition for detection or correction of S or less slip errors. Furthermore, it is seen that the elements obtained from (5-2) represent the union of a large number of cosets of V_1 in V_0. Equation 5-2 is then a generalization of Eqn. 5-1 and as such the resulting code will be more efficient, in the sense that it will contain a larger number of codewords. We now establish the synchronization capability of the GSC codes.

Slip in a noiseless channel

The first theorem concerns the ability of a GSC to detect slip in a noiseless channel.

Theorem 5-2-1

Given an (n,k) cyclic code whose parity check polynomial has an irreducible factor of degree m_1 and exponent e_1, there exists a GSC that

can detect S or less bits of slip in a noiseless channel, where $S \leq \min(n-k, e_1-1)$. The rate of the GSC is

$$(k-m_1 + \log_2(2^{m_1}-1) - \log_2(e_1) + \log_2[e_1/(S+1)])/n.$$

Proof

The words in the GSC are all of the form of Eqn. (5-2) with $a = 1$. Assuming a left slip, from the previous description of slip error we can express the framed n-tuple as

$$x^s w(x) + U_s(x) \tag{5-3}$$

where $U_s(x)$ is a polynomial of degree at most s-1, and $s \leq S$. First we compute the syndrome of (5-3) modulo G(x) which is $U_s(x)$ modulo G(x). Since $s \leq n-k$, $U_s(x)$ is not a codeword and a non-zero syndrome is obtained if $U_s(x) \neq 0$. If $U_s(x) = 0$, we multiply (5-3) by $E_1(x)$, the idempotent of M(1), and reduce modulo x^n-1. This yields

$$x^{i(S+1)+s}\, t_1^j(x) \tag{5-4}$$

A slip will be detected if the slip syndrome is different from the slip syndrome when there is no slip. Now these are all of the form

$$x^{i_1(S+1)}\, t_1^{j_1}(x). \tag{5-5}$$

Clearly (5-4) is different from (5-5) if $|s| \leq |S| \leq e_1-1$. To obtain the expression for the rate, we need only to note that the number of words of

the form (5-2) with a = 1, is

$$(2^{k-m_1})((2^{m_1}-1)/e_1)[e_1/(S+1)],$$

and the rate is the $\log_2$ of this expression divided by n.

Q.E.D.

Although the proof of theorem 5-2-1 was given for the case of a left slip, the symmetry of the situation makes the proof valid also for right slip situations and henceforth all proofs will be given assuming left slip conditions. We now examine the ability of GSC codes to correct slip in a noiseless channel.

Theorem 5-2-2

Given an (n,k) cyclic code whose parity check polynomial has an irreducible factor of degree m_1 and exponent e_1, there exists a GSC that can correct S or less bits of slip in a noiseless channel, where $S \leq \min((n-k)/2,(e_1-1)/2)$. The rate of the GSC is

$$(k-m_1 + \log_2(2^{m_1}-1) - \log_2(e_1) + \log_2[e_1/(2S+1)])/n.$$

Proof

The codewords are all of the form of Eqn. (5-2) with a = 2. Assuming a left slip, the framed n-tuple has the form

$$x^S w(x) + U_S(x) \tag{5-7}$$

where $U_s(x)$ is a polynomial of at most degree s-1 and $s \leq S$. First the decoder computes the syndrome of (5-7) modulo G(x). Now $U_s(x)$ can be determined from the syndrome obtained since $U_s(x) \neq U_r(x)$ modulo G(x) for all distinct $U_s(x)$ and $U_r(x)$ if $|s|, |r| \leq (n-k)/2$. Once $U_s(x)$ has been determined, it is added onto (5-7) which we then multiply by $E_1(x)$ to obtain

$$x^{i(2S+1)+s} t_1^j(x) \tag{5-8}$$

the slip syndrome. If $s \leq S \leq (e_1-1)/2$, then s can be uniquely determined from (5-8) since all zero slip syndromes are of the form

$$x^{i_1(2S+1)} t_1^{j_1}(x).$$

The expression for the rate is obtained by counting the number of words of the form (5-2) with a = 2. This number is

$$(2^{k-m_1})((2^{m_1}-1)/e_1)[(e_1/(2S+1)] \tag{5-9}$$

and the rate is obtained by taking the $\log_2$ of (5-9) and dividing by n. Q.E.D.

In Table 2 the slip error detecting and correcting capability of a GSC of length 127 is compared to that of the subset codes (theorems 5-1-13 and 5-1-14), and the coset codes (theorems 5-1-6 and 5-1-7). It is seen that theorems 5-2-1 and 5-2-2 are a generalization of theorems 5-1-13 and 5-1-14. The result has been that the rate has gone from $(k-m_1)/n$, to $(k-m_1 + p)/n$, where p is always a positive quantity, with-

out any loss of synchronization ability. The close relationship between cycle representatives and the synchronization problem has also been shown. We now investigate the performance of a GSC for a type 1 channel.

Slip in a type 1 channel

Following Tavares and Fukada[23] a type 1 channel is defined as one in which additive and slip error cannot occur simultaneously. We then have

Theorem 5-2-3

Given an (n,k) cyclic code having minimum distance d, whose parity check polynomial has a factor of degree m_1 and exponent e_1, and a type 1 channel, there exists a GSC which can correct $|s| \leq S$ bits of slip and any additive error pattern that the original cyclic code could correct, where $S = \min((d-1)/2,(e_1-1)/2)$. The rate of the GSC is

$$(k-m_1 + \log_2(2^{m_1}-1) - \log_2(e_1) + \log_2[e_1/(2S+1)])/n.$$

Proof

Let w(x) be a codeword as defined in (5-2) with a = 2. The decoder proceeds to calculate a syndrome modulo G(x) assuming that additive errors have occurred. If slip has occurred, then because $|s| \leq (d-1)/2$, $U_s(x)$ can be uniquely determined. After this first step, assuming a left slip, we are left with $x^S w(x)$. We then compute the slip syndrome by multiplying $x^S w(x)$ by $E_1(x)$, to obtain $x^{i(2S+1)+s}\, t_1^j(x)$. Since $|s| \leq |S| \leq (e_1-1)/2$, and the fact that all zero slip syndromes are of

the form $x^{i_1(2S+1)} t_1^{j_1}$ the slip s can be determined. The rate again follows from the number of words of the form of Eqn. (5-2) with a = 2.

Q.E.D.

In Table 3, the performance of a GSC code of length 127 is compared to that of a subset code (theorem 5-1-15), a coset code (theorem 5-1-8), and a Fibonacci code (theorem 5-1-5) for a type 1 channel. In this instance, the increase in rate of the GSC codes over the subset codes is given by the term $\log_2[e_1/(2S+1)]/n$ in the expression for the information rate of theorem 5-2-3.

Slip in noisy channels

The following theorem is concerned with the ability of a GSC to detect additive and slip errors when they may occur simultaneously.

Theorem 5-2-4

Given an (n,k) cyclic code with minimum distance d, whose parity check polynomial has a factor of degree m_1 and exponent e_1, there exists a GSC which can detect the simultaneous occurrence of $|s| \leq |S| \leq d-1$ slip errors and $d-|s|-1$ additive errors, where $S = \min(e_1-1, d-1)$. The rate of the GSC is

$$(k-m_1 + \log_2(2^{m_1}-1) - \log_2(e_1) + \log_2[e_1/(S+1)])/n.$$

Proof

Let w(x) be a codeword as defined by Eqn. (5-2) with a = 1. Assuming a left slip of s bits, the framed word can be expressed as

$$x^S w(x) + U_s(x) + \xi(x) \qquad (5\text{-}10)$$

where the weight of $\xi(x)$, the additive error pattern, is $d-|s|-1$ or less and $s \leq S$. Provided that $U_s(x) + \xi(x) \neq 0$, (5-10) will be a non-zero syndrome modulo $G(x)$ and an error will be detected. If $U_s(x) + \xi(x) = 0$, the decoder computes the slip syndrome by multiplying (5-10) by $E_1(x)$ to obtain $x^{i(S+1)+s}\, t_1^j(x)$. Now since $|s| \leq S \leq e_1-1$ and all zero slip syndromes being of the form $x^{i_1(S+1)}\, t_1^{j_1}(x)$ an error will be detected. The rate follows by noting the number of words of the form (5-2) with $a = 1$.

Q.E.D.

In Table 4, the simultaneous slip and additive error detecting capability of a GSC code of length 127 is compared to that of a subset code (theorem 5-1-16), a coset code (theorem 5-1-9), and a Fibonacci code (theorem 5-1-2) of the same length. The next two theorems describe the performance of the GSC codes when we desire to correct additive errors.

Theorem 5-2-5

Given an (n,k) cyclic code with minimum distance d, whose parity check polynomial has a factor of degree m_1 and exponent e_1, then there exists a GSC that can correct any t additive error pattern the original cyclic code could correct in the absence of slip, where $t = (d-1)/2$, and in addition detect the presence of $|s| \leq S$ bits of slip when slip(s) and additive error (b) occur simultaneously if $|s| + b \leq (d-1)/2$ and $S = \min(e_1-1, (d-1)/2)$. The rate of the GSC is

$$(k-m_1 + \log_2(2^{m_1}-1) - \log_2(e_1) + \log_2[e_1/(S+1)])/n.$$

Proof

All codewords are of the form given by Eqn. (5-2), with a = 1. The decoder regards all errors as additive errors and correct them as such. Assuming an additive error $\xi(x)$ and s bits of left slip to have occurred, the framed polynomial will be of the form

$$x^S w(x) + U_s(x) + \xi(x).$$

Let $|s| + W(\xi(x)) \leq (d-1)/2$, the decoder will correct $U_s(x) + \xi(x)$. The remaining polynomial is $x^S w(x)$ and the decoder now computes the slip syndrome by multiplying by $E_1(x)$ to obtain $x^{i(S+1)+s}\, t_1^j(x)$. Since $|s| \leq S \leq e_1-1$, the decoder can detect a slip, because all zero-slip syndromes are of the form $x^{i_1(S+1)}\, t_1^{j_1}(x)$. The rate expression is obtained by counting the number of words of the form of Eqn. (5-2) with a = 1.

Q.E.D.

In Table 5, the performance of a GSC of length 127 for the correction of additive errors and the detection of slip errors is compared to that of a subset code (theorem 5-1-17), a coset code[23] and a Fibonacci code (theorem 5-1-3) of the same length. In the next theorem, we wish to correct for both slip and additive errors.

Theorem 5-2-6

Given an (n,k) cyclic code with minimum distance d, whose parity check polynomial has an irreducible factor of degree m_1 and exponent e_1, there exists a GSC which can correct any combination of b additive errors and $|s| \leq S$ bits of slip if $b + |s| \leq (d-1)/2$ and $S = \min((d-1)/2, (e_1-1)/2)$.

The rate of the GSC is

$$(k-m_1 + \log_2(2^{m_1}-1) - \log_2(e_1) + \log_2[e_1/(2S+1)])/n.$$

Proof

The proof is along the same lines as that of theorem 5-2-5, however because $|s| \leq S \leq (e_1-1)/2$, the decoder can now correct for slip errors. Q.E.D.

In Table 6 the performance of a GSC of length 127 for the correction of both slip and additive errors is compared to that of a subset code (theorem 5-1-18), a coset code (theorem 5-1-10), and a Fibonacci code (theorem 5-1-4) of the same length. Also included in the table are Tong's shortened codes[42,23] and the modified Shiva and Seguin codes[45,23].

In this section, the concept of subset codes has been generalized resulting in more powerful codes as Tables 2 to 6 indicate. The GSC possess all the properties of the subset codes including adaptivity under various conditions of slip and additive errors, so that if no slip has occurred, the code retains its full error detecting/correcting capability.

The technique outlined in this section for synchronization recovery, consisted in generating elements of (V_0-V_1) that are separated by at least (aS+1) cyclic shifts from other elements in the same cycle. We note that the set of words in a GSC could still be further enlarged by considering all the cycles in the code and choosing elements in each cycle that are separated by at least (aS+1) cyclic shifts. For those cases where all the cycles have order n (excluding the all-zero codeword),

the number of words in the code would then be $(2^k-1)[n/(aS+1)]/n$, which yield a rate of

$$(\log_2(2^k-1) - \log_2(n) + \log_2[n/(aS+1)])/n,$$

this compares to the rates obtained in theorems 5-2-1 through 5-2-6 which are of the form

$$(k-m_1 + \log_2(2^{m_1}-1) - \log_2(n) + \log_2 n/(aS+1))/n.$$

For large value of m_1 the difference between these expressions approaches zero, so that in most instances the added complexity required to generate all of the cycles would not justify the small gains, in that most of the cycles are already in (V_0-V_1).

We now apply the theory of cycle representatives to another well known technique for cyclic code synchronization.

5-3 A Generalization of the BCW Technique

The BCW technique has already been discussed. We have seen that this method alters the length of the codewords while also constraining these to be of the form

$$w(x) = (h(x)i(x) + 1)G(x),$$

where G(x) is the generator polynomial of an (n,k) cyclic code, h(x) an irreducible polynomial of degree m and exponent n, and i(x) an information polynomial of degree less than k-m. Because the codewords are extended by 2S bits, any loss of synchronization of $|s| \leq S$ bits will cause the

receiver to frame an n-tuple that will be of the form $x^S w(x)$. It is then a simple matter to show that the slip syndrome, which is $x^S w(x)$ modulo $h(x)$, uniquely specifies the magnitude and direction of s. In this section, we wish to extend the BCW technique to include a larger set of words without sacrificing any of its slip error correcting/detecting capability. Starting with an (n,k) cyclic code having G(x) has its generator, we consider a subcode whose words are of the form

$$w(x) = \left\{G(x)h_1(x)i(x) + t_1^j(x)\right\} x^{i(aS+1)},$$

which is the form of Eqn. (5-2). Every such codeword will then be extended as described by the BCW technique, so that if $(a_0,a_1,a_2,\ldots,a_{n-1})$ is a codeword of the form of Eqn. (5-2), it is extended as

$$(a_{n-S},\ldots,a_{n-1},a_0,a_1,\ldots,a_{n-1},a_0,a_1,\ldots,a_{S-1}).$$

The following theorems describe the synchronization capabilities of these codes.

Theorem 5-3-1

Given an (n,k) cyclic code whose parity check polynomial has a factor of degree m_1 and exponent e_1, there exists a generalized BCW code that can correct any additive error pattern the original cyclic code could correct, and in addition detect the simultaneous occurrence of S or less bits of slip, where $S \leq e_1-1$. The rate of the code is

$$(k-m_1 + \log_2(2^{m_1}-1) - \log(e_1) + \log_2[e_1/(S+1)])/(n+2S).$$

Proof

Every codeword is of the form of Eqn. (5-2) with a = 1, and is then extended as according the BCW technique. Assuming a left slip to have occurred, the framed word can then be expressed as $x^S w(x) + \xi(x)$, where $\xi(x)$ is an additive error pattern and $|s| \leq S \leq e_1-1$. After correcting $\xi(x)$, we multiply by $E_1(x)$ to obtain $x^{i(S+1)+s}\, t_1^j(x)$, the slip syndrome. Now since all zero slip syndromes are of the form $x^{i_1(S+1)}\, t_1^{j_1}(x)$, and $|s| \leq S \leq e_1-1$, the slip can be detected. The expression for the rate follows upon noting the number of words of the form of Eqn. (5-2) with a = 1, and the fact that the length of the transmitted word is now n + 2S. Q.E.D.

In Table 7 the performance of the generalized BCW codes is compared to that of the subset codes (theorem 5-1-17), the BCW codes (theorem 5-1-12), the Fibonacci codes (theorem 5-1-3), and the GSC codes (theorem 5-2-5) for n = 127. In the next theorem, the performance of the generalized BCW codes for the correction of slip and additive errors is described.

Theorem 5-3-2

Given an (n,k) cyclic code whose parity check polynomial has a factor of degree m_1 and exponent e_1, there exists a generalized BCW code that can correct any additive error pattern the original code could correct, and in addition correct the simultaneous occurrence of S or less slip errors, where $S \leq (e_1-1)/2$. The rate of the code is

$$(k-m_1 + \log_2(2^{m_1}-1) - \log_2(e_1) + \log_2[e_1/(2S+1)])/(n+2S).$$

Proof

The proof is along the same lines as in the previous theorem, except now, those elements of the code which belong to the same cycle are separated by (2S+1) shifts, so that correction of slip is possible for all slips such that $|s| \leq S \leq (e_1-1)/2$.

Q.E.D.

The performance of the generalized BCW codes is illustrated in Table 8, where rate is computed as a function of slip and additive errors corrected for n = 127. The rate is also computed for the BCW codes (theorem 5-1-11), the Fibonacci codes (theorem 5-1-4), the Mandelbaum codes (theorem 5-1-1), the subset codes (theorem 5-1-18), and the GSC codes (theorem 5-2-6). It is seen from this table, that the most powerful techniques presently known are the Mandelbaum and the generalized BCW techniques, for slip and additive error correction.

In this chapter we have used our knowledge of cycle representatives to generalize two synchronization techniques, namely, the subset codes and the BCW codes. Recognizing that these techniques generated only a fraction of the total number of cycles in the code, we showed how these methods could be extended by using a subset of the code whose codewords are separated by (aS+1) cyclic shifts. The generalized code so obtained possessed all of the synchronization capabilities of the original codes. Also, the development of the theory was such so as to bring out the close relationship between the problems of cycle representatives and

that of the synchronization of cyclic codes. The end result has been to obtain more efficient synchronization techniques (from the point of view of achievable rate), and at the same time further our knowledge of the synchronization problem associated with cyclic codes.

CHAPTER 6

CONCLUDING REMARKS

The problem of finding the cyclic class representatives of a cyclic code has been examined and a solution provided for the binary case. The generalization over GF(q) follows parallel lines. The generation of the cycles depends however on obtaining a primitive element of the groups $M^*(i)$, $i = 1,2,\ldots,r$, a task that could involve considerable computational difficulties, and which must be taken into account when applying the results. In this context, an algorithm for obtaining primitive elements of finite fields would be of great value. The application of the theory of cycle representatives to the problem of the weight distribution in cyclic codes was briefly discussed and illustrated by means of an example. Here, another result that is helpful is the fact that since the ideal V_i is maximally contained in V_{i-1}, the cosets of V_i in V_{i-1} can be shown to have the structure of a field. The significance of this result to the weight distribution problem is in the fact that the cosets $(t_i'(x) + V_i)^{2^f}$ will have the same weight distribution for $f = 0,1,2,\ldots,(m_i-1)$, and hence if one is interested only in the weight spectrum, the number of elements that need be generated, can be further reduced.

The theory of cyclic representatives found a natural application to the problem of cyclic code synchronization. All of the synchronization techniques of Chapter 5 have relied on generating elements separated

by a definite number of cyclic shifts so that the theory of cycle representative is a basis for a unifying approach to this problem. This point was further emphasized in the generalization of the Subset Codes and BCW techniques presented in that chapter. There, the theory of cycle representatives was employed to extend these techniques so as to include a larger set of codewords while maintaining their synchronization capabilities. The performance of the generalized techniques was then compared to various other synchronization techniques in tables 2 to 8. This comparison was strictly on the basis of achievable rate under various conditions of slip and additive errors and it is obvious that such a comparison will suffer from a number of limitations. The first is that the tables fail to point out the major advantage of the GSC codes, namely, their adaptive capability. The GSC codes are so constructed that they are able to trade-off between their additive and slip error correcting capability so that in the absence of slip, the code increases its additive error correcting capability from b to b+S(theorem 5-2-6). Except for the subset and coset codes none of the other techniques are fully adaptive in the above sense. This feature is of special importance for those channels where once synchronization has been established it is likely to be maintained. The second limitation arises from practical considerations. Here from the point of view of cost/effectiveness, a technique such as that of Mandelbaum is highly competitive. However, for codes of maximal length ($n = 2^S-1$), it is seen from Eqn. (5-2) that $t_1(x)$ is in fact a maximal-length-sequence which is added to the codeword $G(x)h_1(x)i(x)$. In that m-sequences possess good correlation properties[10], it is con-

ceivable that the GSC and the generalized BCW codes could be practical alternatives to other synchronization techniques.

Apart from these limitations discussed above, the tables do serve the purpose of providing a comparison of the various synchronization techniques. From Table 8, we see that the Mandelbaum and the generalized BCW techniques achieve the highest rate for the various conditions of slip and additive errors. This can be explained from the fact that in these techniques, slip and additive errors can be treated independently. In the other techniques, the effect of slip can be to add an additive error pattern $U_S(x)$ to the codeword, as described in theorems 5-2-1 to 5-2-6 for the GSC codes. Hence, a more powerful code is required at the onset with a resultant lower rate. Theoretically, if slip and additive errors could be treated independently, a maximum of $\log_2(aS+1)$ bits of redundancy would be needed to maintain synchronization. So far all of the techniques investigated for cyclic code synchronization must make use of far more redundancy.

Although it will be conceded that many of the results derived in this thesis are presently only of theoretical interest, the results on synchronization, lend themselves quite readily to a software realization and could find application in that type of communication system in which the digital computer is an integral element. Finally, a possible extension of this work would be to a more general class of codes such as the Abelian Group Codes[60] for which a general weight preserving permutation group is defined. Such an extension would follow a similar development to the present work and might be of value in computing the error correcting capability of interesting new codes.

TABLE 1

Cyclic Class Decomposition of the Ring $GF[x]/(x^n-1)$

n	
3	2(1), 2(3).
4	2(1), 1(2), 3(4).
5	2(1), 6(5).
6	2(1), 2(3), 9(6).
7	2(1), 18(7).
8	2(1), 1(2), 3(4), 30(8).
9	2(1), 2(3), 56(9).
10	2(1), 1(2), 6(5), 99(10).
11	2(1), 186(11).
13	2(1), 630(13).
14	2(1), 1(2), 18(7), 1161(14).
15	2(1), 2(3), 6(5), 2182(15).
16	2(1), 1(2), 3(4), 30(8), 4080(16).
17	2(1), 7710(17).
18	2(1), 1(2), 2(3), 9(6), 56(9), 14532(18).
19	2(1), 27594(19).
20	2(1), 3(4), 6(5), 99(10), 52377(20).
21	2(1), 2(3), 18(7), 99858(21).
22	2(1), 1(2), 186(11), 190557(22).

x(y) = x cycles of cyclic order y.

TABLE 2

Rate Comparison of Codes for Slip in Noiseless Channels

	THEOREMS 5-2-1 DETECTION			THEOREM 5-2-2 CORRECTION		
S	Subset Codes	Coset Codes	GSC Codes	Subset Codes	Coset Codes	GSC Codes
1	0.890	0.945	0.936	0.890	0.945	0.932
2	0.890	0.945	0.932	0.890	0.945	0.926
3	0.890	0.945	0.929	0.890	0.890	0.922
4	0.890	0.945	0.926	0.835	0.890	0.864
5	0.890	0.945	0.924	0.835	0.890	0.864
6	0.890	0.945	0.922	0.835	0.890	0.859
7	0.890	0.890	0.920	0.835	0.835	0.858
12	0.835	0.890	0.859	0.725	0.780	0.743
13	0.835	0.890	0.858	0.725	0.780	0.740
14	0.835	0.835	0.858	0.725	0.725	0.740

S slip error n = 127

TABLE 3

Rate Comparison of Codes for Slip in Type 1 Channels

S	b	Subset Codes	Coset Codes	Fibonacci Codes	GSC Codes
1	1	0.890	0.890	0.913	0.932
	2	0.835	0.835	0.859	0.877
	3	0.780	0.780	0.804	0.822
	4	0.725	0.725	0.747	0.767
	5	0.670	0.670	0.693	0.712
3	1	0.780	0.835	0.905	0.813
	2	0.780	0.780	0.850	0.813
	3	0.780	0.725	0.797	0.813
	4	0.725	0.670	0.739	0.758
	5	0.670	0.614	0.685	0.703
5	1	0.670	0.780	0.797	0.697
	2	0.670	0.725	0.797	0.697
	3	0.670	0.670	0.797	0.697
	4	0.670	0.614	0.739	0.697
	5	0.670	0.559	0.685	0.697
	6	0.614	0.559	0.630	0.641
	7	0.557	0.504	0.574	0.584

S slip error

b additive error

n = 127

REF: Theorem 5-2-3

TABLE 4

Rate Comparison of Codes for Slip in Noisy Channels

S	b	Subset Codes	Coset Codes	Fibonacci Codes	GSC Codes
1	1	0.890	0.890	0.913	0.937
	2	0.836	0.835	0.913	0.929
	3	0.835	0.780	0.859	0.882
	4	0.780	0.725	0.859	0.874
	5	0.780	0.670	0.804	0.827
3	1	0.835	0.835	0.905	0.874
	2	0.780	0.780	0.850	0.866
	3	0.780	0.725	0.850	0.819
	4	0.725	0.670	0.797	0.811
	5	0.725	0.614	0.797	0.764
5	1	0.780	0.780	0.850	0.814
	2	0.725	0.725	0.797	0.806
	3	0.725	0.670	0.797	0.759
	4	0.670	0.614	0.739	0.751
	5	0.670	0.559	0.739	0.704
	6	0.614	0.559	0.685	0.696
	7	0.614	0.504	0.685	0.649

S slip error

b additive error n = 127

Theorem 5-2-4

TABLE 5

Rate Comparison of Codes for Slip in Noisy Channels

S	b	Subset Codes	Coset Codes	Fibonacci Codes	GSC Codes
1	1	0.835	0.835	0.913	0.882
	2	0.780	0.725	0.859	0.827
	3	0.725	0.614	0.804	0.772
	4	0.670	0.559	0.747	0.717
	5	0.614	0.449	0.693	0.661
3	1	0.725	0.780	0.804	0.764
	2	0.670	0.670	0.747	0.709
	3	0.614	0.559	0.693	0.653
	4	0.559	0.504	0.637	0.596
	5	0.504	0.394	0.583	0.543
5	1	0.614	0.725	0.693	0.648
	2	0.559	0.614	0.637	0.591
	3	0.504	0.559	0.582	0.538
	4	0.504	0.449	0.528	0.538
	5	0.449	0.394	0.472	0.483
	6	0.394	0.284	0.418	0.428
	7	0.338	0.228	0.362	0.372

S slip error

b additive error n = 127

Theorem 5-2-5

TABLE 6

Rate Comparison of Codes for Correction of Slip in Noisy Channels

S	b	Coset Codes	Tong's Shortened Codes	Modified Shiva Seguin	Fibonacci Codes	Subset Codes	GSC Codes
1	1	0.780	0.774	0.867	0.913	0.835	0.877
	2	0.670	0.717	0.811	0.859	0.780	0.822
	3	0.559	0.660	0.755	0.804	0.725	0.767
	4	0.504	0.605	0.700	0.747	0.670	0.712
	5	0.394	0.549	0.645	0.693	0.614	0.656
3	1	0.670	0.533	0.725	0.804	0.725	0.757
	2	0.559	0.533	0.670	0.747	0.670	0.702
	3	0.504	0.475	0.614	0.693	0.614	0.646
	4	0.394	0.417	0.559	0.637	0.559	0.591
	5	0.338	0.358	0.504	0.582	0.504	0.536
5	1	0.559	0.336	0.583	0.693	0.614	0.641
	2	0.504	0.336	0.527	0.637	0.559	0.586
	3	0.394	0.276	0.473	0.582	0.504	0.531
	4	0.338	0.216	0.473	0.528	0.504	0.531
	5	0.228	0.155	0.417	0.472	0.449	0.476

S slip error

b additive error $\qquad n = 127$

Theorem 5-2-6

TABLE 7

Rate Comparison of Codes for Slip in Noisy Channels

S	b	Subset Codes	BCW Codes	Fibonacci Codes	GSC Codes	Generalized BCW Codes
1	1	0.835	0.877	0.913	0.882	0.923
	2	0.780	0.822	0.859	0.827	0.868
	3	0.725	0.767	0.804	0.772	0.813
	4	0.670	0.713	0.747	0.717	0.759
	5	0.614	0.659	0.693	0.661	0.705
3	1	0.725	0.850	0.804	0.764	0.887
	2	0.670	0.797	0.747	0.709	0.834
	3	0.614	0.745	0.693	0.653	0.782
	4	0.559	0.691	0.637	0.596	0.728
	5	0.504	0.639	0.582	0.543	0.676
5	1	0.614	0.825	0.693	0.648	0.857
	2	0.559	0.774	0.637	0.591	0.806
	3	0.504	0.725	0.582	0.538	0.757
	4	0.504	0.672	0.528	0.538	0.704
	5	0.449	0.620	0.472	0.583	0.652
	6	0.394	0.570	0.418	0.428	0.602
	7	0.338	0.518	0.362	0.372	0.550

S slip error

b additive error n = 127

Theorem 5-3-1

TABLE 8

Rate Comparison of Codes for Correction of Slip in Noisy Channels

S	b	Subset Codes	GSC Codes	Fibonacci Codes	Mandelbaum Codes	BCW Codes	Generalized BCW
1	1	0.835	0.877	0.913	0.922	0.877	0.918
	2	0.780	0.822	0.859	0.866	0.822	0.863
	3	0.725	0.767	0.804	0.811	0.767	0.808
	4	0.670	0.712	0.747	0.756	0.713	0.754
	5	0.614	0.656	0.693	0.700	0.659	0.700
3	1	0.725	0.757	0.804	0.890	0.850	0.881
	2	0.670	0.702	0.747	0.835	0.797	0.828
	3	0.614	0.646	0.693	0.779	0.745	0.776
	4	0.559	0.591	0.637	0.725	0.691	0.722
	5	0.504	0.536	0.582	0.670	0.639	0.670
5	1	0.614	0.641	0.693	0.858	0.825	0.850
	2	0.559	0.586	0.637	0.803	0.774	0.799
	3	0.504	0.531	0.582	0.748	0.725	0.750
	4	0.504	0.531	0.528	0.693	0.672	0.697
	5	0.449	0.476	0.472	0.638	0.620	0.645

S slip error

b additive error n = 127

Theorem 5-3-2

REFERENCES

1. Shannon, C.E. (1948), *A mathematical theory of communication,* Bell System Tech. J. 27, 379-423 and 623-656.

2. Elias, P. (1955), *Coding for noisy channels,* IRE Conv. Rec., pt. 4, 37-44.

3. Hagelbarger, D.W. (1959), *Recurrent Codes: Easily mechanized burst-correcting binary codes,* Bell System Tech. J. 38, 969-984.

4. Massey, J.L. (1963), *Threshold Decoding,* The M.I.T. Press Cambridge, Mass.

5. Wyner, A.D. and Ash, R.B. (1963), *Analysis of recurrent codes,* IEEE Trans. IT-9, 143-146.

6. Reddy, S.M. (1968), *A class of convolutional codes and a new decoding algorithm,* Ph.D. thesis, Electrical Engineering Department, University of Iowa, Iowa City.

7. Peterson, W.W. and Weldon, E.J. (1972), *Error-Correcting Codes,* 2nd Edition, MIT Press, Cambridge, Massachusetts.

8. Prange, E. (1957), *Cyclic error correcting codes in two symbols,* AFCRC-TN-57-103, Mass.

9. Hamming, R.W. (1950), *Error detecting and error correcting codes,* Bell System Tech. J. 29, 147-160.

10. Golomb, S.W. (1967), *Shift Register Sequences,* Holden-Day Inc.

11. Bose, R.C. and Ray-Chaudhuri, D.K. (1960), *On a class of error correcting binary group codes,* Inform. Control, 3, 68-79.

12. Hocquenghem, A. (1959), *Codes correcteur d'erreurs,* Chiffres (Paris), 2, 147-156.

13. Kasami, T., Lin, S. and Peterson, W.W. (1968), *New Generalization of the Reed-Muller codes - Part I: Primitive codes,* IEEE Trans. IT-14, 189-198.

14. Peterson, W.W. (1960), *Encoding and error-correcting procedures for the Bose-Chaudhuri codes,* IRE Trans. IT-6, 459-470.

15. Berlekamp, E.R. (1968), *Algebraic Coding, Theory,* McGraw-Hill Company, New York.

16. MacWilliams, F.J. (1965), *The structure and properties of binary cyclic alphabets,* Bell System Tech. J. 44, 303-332.

17. Goethals, J.M. (1966), *Analysis of weight distribution in binary cyclic codes,* IEEE Trans. IT-12-401-402.

18. Tavares, S.E., Allard, P.E. and Shiva, S.G.S. (1971), *On the decomposition of cyclic codes into cyclic classes,* Inform. Control 18, 342-354.

19. Allard, P.E., Shiva, S.G.S. and Tavares, S.E. (1972), *A note on the decomposition of cyclic codes into cyclic classes,* Inform. Control (to appear).

20. Reed, I.S. (1971), K^{th} *order near-orthogonal codes,* IEEE Trans. IT-15, 116-117.

21. Massey, J.L. (1970), *Sub-Baud coding and cyclic codes,* Int. Symp. on Inform. Theory, Noordwick, The Netherlands.

22. Seguin, G. (1970), *On the weight distribution of cyclic codes,* IEEE Trans. IT-16, 358.

23. Tavares, S.E. and Fukada, M. (1970), *Synchronization of a class of codes derived from cyclic codes,* Inform. Control 16, 153-166.

24. Bose, R.C. and Caldwell, J.C. (1967), *Synchronizable error correcting codes,* Inform. Control 10, 616-630.

25. Birkhoff and MacLane (1965), *A Survey of Modern Algebra,* 3rd ed., The MacMillan Company, New York.

26. Dean, R.A. (1966), *Elements of Abstract Algebra,* John Wiley and Sons, New York.

27. Lucky, R.W., Saltz, J. and Weldon, E.J. (1968), *Principles of Data Communications,* McGraw-Hill Book Company, San Francisco.

28. Forney, G.D. (1966), *Concatenated Codes,* The M.I.T. Press, Cambridge, Mass.

29. Reed, I.S. (1954), *A class of multiple error correcting codes and the decoding scheme,* IRE Trans. IT-4, 38-49.

30. Kasami, T., Lin, S. and Peterson, W.W. (1966), *Some results on cyclic codes which are invariant under the affine group,* AFCRL 66-622.

31. MacWilliams, F.J. (1963), *A theorem on the distribution of weights in a systematic code,* Bell System Tech. J. 42, 79-94.

32. Goethals, J.M. and Delsarte, P. (1968), *On a class of majority-logic decodable cyclic codes,* IEEE Trans. IT-14, 182-188.

33. Burton, H.O. and Weldon, E.J. (1965), *Cyclic product codes*, IEEE Trans. IT-11, 433-439.

34. Fire, P. (1959), *A class of multiple error-correcting codes for non-independent errors*, Sylvania Reconnaissance Systems Lab. Report RSL-E-2.

35. Liu, C.L. (1968), *Introduction to Combinatorial Mathematics*, McGraw-Hill Book Company.

36. Gorenstein, D.C. and Zierler, N. (1961), *A class of error correcting codes in p^m symbols*, J. Soc. Indus. Appl. Math. 9, 207-214.

37. Golomb, S.W., Gordon, B. and Welch, L.R. (1958), *Comma-free codes*, Canad. J. Math. 10, 202-209.

38. Reed, I.S. and Solomon, G. (1960), *Polynomial codes over certain finite fields*, J. Soc. Indus. Appl. Math. 8, 300-304.

39. Goethals, J.M. (1966), *Algebraic structure and weight distribution of binary cyclic codes*, Institute of Statistics Mineo Series No. 484, 4, Dept. of Statistics, Univ. of North Carolina at Chapel Hill.

40. Karlin, M. (1969), *New binary coding results by circulants*, IEEE Trans. IT-15, 81-92.

41. Nili, H. (1964), *Matrixschaltungen zur Codierung und Decodierung von Gruppen-Code*, Arch. Eleck. Ubertragung 18, 555-565.

42. Tong, S.Y. (1966), *Synchronization recovery techniques for binary cyclic codes*, Bell System TEch. J. 45, 561-596.

43. Tavares, S.E. and Fukada, M. (1969), *Matrix approach to synchronization recovery for binary cyclic codes*, IEEE Trans. IT-15, 93-101.

44. Mandelbaum, D.M. (1972), *Synchronization of codes by means of Kautz's Fibonacci encoding*, IEEE Trans. IT-18, 281-285.

45. Shiva, S.G.S. and Seguin, G. (1970), *Synchronizable error correcting binary codes*, IEEE Trans. IT-16,241.

46. Mandelbaum, D. (1968), *A note on synchronizable error correcting codes*, Inform. Control 13, 429-432.

47. *SPECIAL ISSUE ON ERROR CORRECTING CODES*, IEEE Trans. Comm. Techn. 19, October 1971.

48. Golay, M.J.E. (1949), *Notes on digital coding*, IRE 37, 657.

49. Chien, R.T. (1964), *Cyclic decoding procedures for BCH codes,* IEEE Trans. IT-10, 357-363.

50. Massey, J.L. (1968), *Feedback shift register synthesis and BCH decoding,* IEEE Trans. IT-15, 122-127.

51. Justensen, J. (1972), *A class of constructive asymptotically good algebraic codes,* IEEE Trans. IT-18, 652-656.

52. Weldon, E.J. (1968), *A note on synchronization recovery with extended cyclic codes,* Inform. Control 13, 354-356.

53. Tavares, S.E. and Fukada, M. (1969), *Further results on the synchronization of binary cyclic codes,* IEEE Trans. IT-16, 238-241.

54. Kasami, T. and Tokura, N. (1969), *Some remarks on BCH bounds and minimum weight of binary primitive BCH codes,* IEEE Trans. IT-15, 408-413.

55. Artin, E. (1966), *Galois Theory,* Notre Dame Mathematical Lectures Number 2, University of Notre Dame.

56. McCarthy, P.J. (1966), *Algebraic Extensions of Fields,* Blaisdell Publishing Company.

57. Kasami, T. (1971), *The weight enumerators for several classes of sub-codes of the 2nd order binary Reed-Muller codes,* Inform. Control. 18, 369-394.

58. Berlekamp, E.R. (1970), *The weight enumerators of the second order binary Reed-Muller codes,* Inform. Control 5, 485-500.

59. Scholtz, R.A. and Welch, L.R. (1970), *Mechanization of codes with bounded synchronization delays,* IEEE Trans. IT-16, 438-445.

60. MacWilliams, J.F. (1970), *Binary codes which are ideals in the group algebra of an Abelian Group,* Bell System Tech. J. 49, 987-1011.

V I T A E

Date of Birth	10 April 1942
Place of Birth	Pembroke, Ontario, Canada
Education:	
i) High School	Ecole Secondaire de Hull, Hull, Province de Quebec, Canada
ii) University	University of Ottawa, Ottawa, Canada
iii) Degrees	BSc (Physics) 1963 BaSc (Electrical Engineering) 1966 MSc (Electrical Engineering) 1969