DOCUMENT RESUME

ED 218 084 SE 037 855

AUTHOR Cohen, Simon; Sherman, Gary J.
TITLE Aspects of Coding. Applications of Linear Algebra to Communication and Information Science. [and] A Double-Error Correcting Code. Applications of Algebra to Information Theory. Modules and Monographs in Undergraduate Mathematics and Its Applications Project. UMAP Units 336 and 337.
INSTITUTION Education Development Center, Inc., Newton, Mass.
SPONS AGENCY National Science Foundation, Washington, D.C.
PUB DATE 79
GRANT SED-76-19615-A02
NOTE 57p.

EDRS PRICE MF01 Plus Postage. PC Not Available from EDRS.
DESCRIPTORS *College Mathematics; Higher Education; *Information Science; *Information Theory; Instructional Materials; *Learning Modules; *Mathematical Applications; *Problem Solving; Supplementary Reading Materials; Tests
IDENTIFIERS Coding; *Coding Theory

ABSTRACT

These two modules cover aspects of the coding process and algebraic coding theory. The first unit defines coding as a branch of information and communication science, which draws extensively upon many diverse mathematical fields, primarily abstract and linear algebra, number theory, probability and statistics, and combinatorial theory. Aspects of the process, including linear codes and error correction, are discussed. The second unit focuses on the construction of a double-error correcting code. Both units contain problems and answers to these exercises. The second unit contains a model exam and concludes with an answer key to this test. (MP)

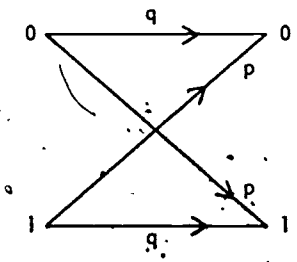
umap

UNIT 336

MODULES AND MONOGRAPHS IN UNDERGRADUATE MATHEMATICS AND ITS APPLICATIONS PROJECT

ASPECTS OF CODING

by Simon Cohen



The Binary Symmetric Channel

APPLICATIONS OF LINEAR ALGEBRA TO COMMUNICATION AND INFORMATION SCIENCE

edc/umap/55chapel st./newton, mass. 02160

ASPECTS OF CODING

by

Simon Cohen
Department of Mathematics
New Jersey Institute of Technology
Newark, New Jersey 07102

TABLE OF CONTENTS

Intermodular Description Sheet: UMAP Unit 336

Title: ASPECTS OF CODING

Author: Simon Cohen
Department of Mathematics
New Jersey Institute of Technology
Newark, New Jersey 07102

Review Stage/Date: III 4/14/79

Classification: APPL LIN ALG/COMMUNICATIONS & INF SCI

Suggested Support Materials:

References: See Section 4 of text.

Prerequisite Skills:
1. From linear algebra: vector spaces over arbitrary fields, basis, matrix algebra, rank.
2. From group theory: coset decomposition, permutation groups.
3. From probability: elementary concepts.

Output Skills:
1. To acquaint the undergraduate student with some of the elementary mathematical facets of coding.

Other Related Units:
A Double-Error Correcting Code *(Unit 337)*
Error Correcting Codes: I *(Unit 346)*

MODULES AND MONOGRAPHS IN UNDERGRADUATE
MATHEMATICS AND ITS APPLICATIONS PROJECT (UMAP)

The goal of UMAP is to develop, through a community of users and developers, a system of instructional modules in undergraduate mathematics and its applications which may be used to supplement existing courses and from which complete courses may eventually be built.

The Project is guided by a National Steering Committee of mathematicians, scientists and educators. UMAP is funded by a grant from the National Science Foundation to Education Development Center, Inc., a publicly supported, nonprofit corporation engaged in educational research in the U.S. and abroad.

PROJECT STAFF

Ross L. Finney	Director
Solomon Garfunkel	Associate Director/Consortium Coordinator
Felicia DeMay	Associate Director for Administration
Barbara Kelczewski	Coordinator for Materials Production
Dianne Lally	Project Secretary
Paula M. Santillo	Administrative Assistant
Carol Forray	Production Assistant
Zachary Zevitas	Order Processor

NATIONAL STEERING COMMITTEE

W.T. Martin	MIT (Chairman)
Steven J. Brams	New York University
Llayron Clarkson	Texas Southern University
Ernest J. Henley	University of Houston
William U. Hogan	Harvard University
Donald A. Larson	SUNY at Buffalo
William F. Lucas	Cornell University
R. Duncan Luce	Harvard University
George Miller	Nassau Community College
Frederick Mosteller	Harvard University
Walter E. Sears	University of Michigan Press
George Springer	Indiana University
Arnold A. Strassenburg	SUNY at Stony Brook
Alfred B. Willcox	Mathematical Association of America

The Project would like to thank Ralph E. Walde of Trinity College and Jack Robertson of Washington State University for their reviews and all others who assisted in the production of this unit.

This material was prepared with the support of National Science Foundation Grant No. SED76-19615 A02. Recommendations expressed are those of the author and do not necessarily reflect the views of the NSF, nor of the National Steering Committee.

ASPECTS OF CODING

1. INTRODUCTION

1.1 What is Coding?

Coding is a branch of information and communication science. It draws extensively upon many diverse mathematical fields, primarily abstract and linear algebra, number theory, probability and statistics, and combinatorial theory. If you are a frustrated applications-oriented algebraist seeking a "real world" outlet for your knowledge, coding just might be the answer.

Coding can be thought of as a sort of reverse shorthand. By means of shorthand one is able, by omitting certain letters from words, to reduce transcription time. However, this saving in time is counterbalanced by the fact that one is more likely to misread a shorthand word than an English language word. In coding, information, in the form of blocks of binary k-tuples, is transmitted over a noisy channel. The noisiness of the channel presents the possibility that the received k-tuple may differ from the transmitted one. However, by lengthening each k-tuple to be transmitted by adding digits to it, channel distortion is made less likely. Of course, these additional digits increase transmission time. A central problem of coding is to find the most efficient way of adding these so-called *check digits*.

1.2 The Coding Process

The processes involved in coding are illustrated in the flow chart in Figure 1. The binary encoder receives the discrete outputs of a communications device and associates a binary k-tuple (k a predetermined positive integer) with each of them. These outputs may be in the form of human speech, high frequency radio waves, numerical data, or in a host of other forms. The k-tuples so formed are called *messages*. The channel encoder receives a message from the binary encoder and, by adding n-k binary digits to it, forms a binary n-tuple (n also a predetermined positive integer, greater than k). The set of all n-tuples formed in this manner is called a *code* and the n-tuples in the code are called *codewords*. The channel is the medium of transmission (e.g., telephone lines, high frequency radio links, space communication links). We assume that the channel is noisy (i.e., "what goes in is not necessarily what comes out").

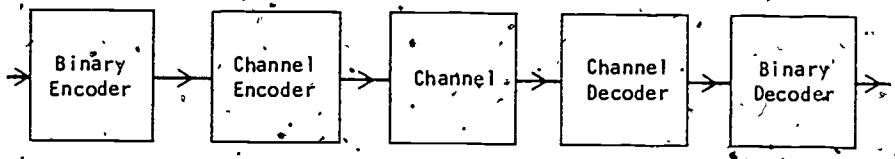


Figure 1. The coding process.

The coding sequence is thus: An output enters the binary encoder where it "becomes" a message which in turn "becomes" a codeword. The channel (possibly) perturbs this codeword into another binary n-tuple and transmits this (possibly) perturbed n-tuple to the channel decoder. Upon receipt of this (possibly) perturbed n-tuple, the channel decoder, which has knowledge of all possible codewords, attempts to determine which codeword in fact entered the channel. The channel decoder then sends its decision (a codeword) to the binary decoder which, by simply reversing the procedure of the channel encoder, determines the message contained in the codeword received. If the channel decoder makes the correct decision, the message leaving the binary encoder is identical to the message leaving the binary decoder.

1.3 The Channel

There are several mathematical models for the channel. We shall deal exclusively with the Binary Symmetric Channel (BSC). A schematic diagram of this channel appears in Figure 2.

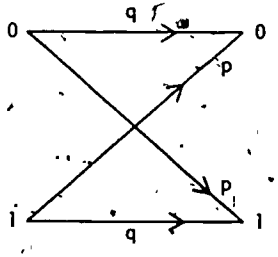


Figure 2. The Binary Symmetric Channel.

With the BSC a given bit (0 or 1) has probability q of being transmitted unaltered and each has probability $p = 1 - q$ of being changed into the other. The BSC assumes that errors occur randomly and independently of one another. We shall stipulate that $p < q$ (actually, to be considered a "good" BSC p has to be on the order of 10^{-5}).

1.4 Decoding

The channel decoder assumes all codewords are equally likely to have been transmitted and makes its decisions according to the *principle of maximum likelihood:*

> A received n-tuple r is decoded into that codeword c which differs from r in the least number of places.*

It is now time for an example.

Example 1. Suppose we have a communications device with four outputs, a, b, c, d, and choose k = 2 so that each of them is represented by a binary 2-tuple, say $a \leftrightarrow 00$, $b \leftrightarrow 10$, $c \leftrightarrow 01$, $d \leftrightarrow 11$. Without channel encoding a message would have to be transmitted through the BSC without error in order to be correctly interpreted. The probability of this happening is q^2. Let us see what happens if we add two check digits to each message. (We are taking n = 4. Precisely how these check digits are chosen is discussed later and is, of course, of crucial importance, but not for the purposes of this example.) We then obtain the four codewords 0000, 1001, 0111, 1110. Now, using maximum likelihood decoding, we make up the following *decoding table.*

0000	1001	0111	1110
0100	1101	0011	1010
0010	1011	0101	1100
0001	1000	0110	1111

At the top of each column appear the codewords. The other 4-tuples appearing in a column are those 4-tuples which differ from the codeword at the top of the column in fewer places than they differ from the other codewords. (Ties have been broken by assuming an error has occurred in position 4.) All 16 binary 4-tuples appear in the table and if a 4-tuple r is received at the channel decoder it is decoded into the codeword c at the top of its column.

Have we actually increased the probability of correctly interpreting a message by resorting to this procedure of adding check digits? A codeword will be correctly decoded if and only if it or any 4-tuple appearing in its column are received. Each of the noncodewords appearing in a column differs from the codeword at the top in exactly one of the positions 2, 3, or 4. Thus if a given codeword is transmitted, the probability of it or a 4-tuple appearing in its column being received is

*In the event of "ties" additional decoding criteria must be given.

$q^4 + 3pq^3$ which can be shown (see Exercise 3) to be greater than q^2 (the non-encoded probability) as long as $p < q$.

Exercises

1. Form a decoding table for the code consisting of the four codewords 11000, 00110, 10011, 01101.
2. For the code in Exercise 1, if the codeword 11000 is transmitted, what is the probability that it will be correctly decoded?
3. Show that $q^4 + 3pq^3 > q^2$ if $p \leq q$.

1.5 Shannon's Theorem

With regard to Example 1 another question arises, namely "Can we do better?" That is, can we somehow alter our code (to obtain a new and different code) so as to increase the probability of correct decoding? In dealing with this question we shall focus upon the effects of changing k and/or n and not upon the actual binary digits in our messages and codewords. It seems plausible that by fixing k and increasing n, say by repeating the message a sufficient number of times, we can obtain a code with probability of correct decoding as close to 1 as we desire. However, this method of repetition has the serious drawback of greatly increasing transmission time and decreasing the *code rate* k/n. Generally speaking, in forming "good" codes we seek to maximize both the code rate and the probability of correct decoding. By virtue of a remarkable theorem we can state categorically that such codes do indeed exist.

Theorem 1 (Shannon's Fundamental Theorem of Coding). Let K be the capacity* of a given BSC. Given any real numbers R and ε, where $0 < R < K$ and $\varepsilon > 0$, there exists a code with code rate > R and probability of correct decoding $> 1 - \varepsilon$.

*Capacity is a positive number associated with a BSC and is a function of the probability p only. In fact, we have $K = 1 - p \log_2 p - (1-p) \log_2 (1-p)$.

The proof of this theorem establishes existence nonconstructively. This is very unfortunate (fortunate?) for it forces us into an in-depth study of coding in order to produce desirable codes. This is precisely what we initiate in the next chapter.

2. LINEAR CODES

2.1 The Code Concept Refined

Until now the only way we have of describing a code is as a subset of binary n-tuples. As there is not much one can say about or do with arbitrary sets of n-tuples, we shall have to restrict our concept of a code somewhat. This restriction will, however, pay great dividends in the quality of the results obtained.

Let V_n be the vector space of binary n-tuples over GF(2). The field GF(2) is the two-element (0 and 1) field of binary arithmetic, with $0 + 0 = 1 + 1 = 0$, $1 + 0 = 0 + 1 = 1$, $0 \cdot 0 = 1 \cdot 0 = 0 \cdot 1 = 0$, $1 \cdot 1 = 1$. Thus, the elements of V_n are n-tuples of zeros and ones from this field.

> An *(n,k) linear code* is a k-dimensional subspace of V_n.

In what follows the adjective linear will be omitted and we shall simply speak of the (n,k) code C or, simply, the code C.

Exercises

4. How many codewords are there in an (n,k) code?
5. Show that the three 5-tuples (1,0,1,0,0), (1,0,0,1,1), (1,0,1,1,1) constitute a linearly independent subset of V_5.
6. Find the codewords in the (5,3) code which has the vectors in Exercise 5 as a basis.

2.2 Hamming Weight and Hamming Distance

Let x be a vector (n-tuple) in V_n. The *Hamming weight*, w(x), of x is the number of 1's appearing among the coordinates of x. From the definitions of binary and vector addition we see that the only coordinate positions where x+y has a 1 are those positions where x or y (but not both) have a 1. Therefore

(1) $w(x+y) \leq w(x) + w(y)$.

The *Hamming distance*, d(x,y), between two n-tuples x and y is the number of positions in which they differ. Note that

(2) $d(x,y) = w(x-y)$

and d is a metric. That is

(3) (i) $d(x,y) \geq 0$, with equality holding if and only if $x = y$.

(ii) $d(x,y) = d(y,x)$.

(iii) $d(x,y) \leq d(x,z) + d(z,y)$.

Exercises

7. Use Equation (2) to establish the fact that Hamming distance satisfies Equations (i)-(iii) in (3) above, so that it is a metric.

2.3 Decoding Revisited

Viewing V_n as a group (with respect to n-tuple addition), and an (n,k) code C as a subgroup, we may form the coset decomposition of V_n with respect to C.

(4) $$V_n = \bigcup_{i=1}^{2^{n-k}} (a_i + C), \qquad \text{(disjoint)}.$$

where the coset representatives a_i are chosen to be n-tuples of minimum weight in their cosets. Letting $A = \{a_i\}$ and $C = \{c_i\}$, it follows directly from Equation (4) that

(5) $$V_n = \bigcup_{i=1}^{2^k} (c_i + A). \qquad \text{(disjoint)}$$

In terms of Hamming distance, maximum likelihood decoding reads as follows: A received n-tuple r is decoded into that codeword which minimizes $\{d(c,r)/c \in C\}$. As a consequence of Equation (5), for any received n-tuple r there exists a unique codeword c_j and an a_k in A such that $r = c_j + a_k$. Then for any c in C we have

(6) $$c-r = c - (c_j + a_k) = a_k + (c - c_j) \in a_k + C \Rightarrow$$
$$\Rightarrow d(c,r) = w(c-r) \geq w(a_k) = w(r-c_j) = d(c_j,r)^*$$

which tells us that r should be decoded as c_j.

From the result above we can deduce that the rows of the decoding table for C are the cosets appearing in Equation (4). This observation allows for the simple construction of the table. The algorithm is as follows: As is standard, the first row consists of the codewords themselves. Of the remaining n-tuples, one of minimum weight is chosen and placed under the zero codeword. Each of the remaining n-tuples in the second row is the sum of the codeword immediately above it and this minimum-weight n-tuple (see the decoding table in Example 1). Now once again an n-tuple of minimum weight is selected from the remaining n-tuples, placed under the zero codeword, and the third row is filled out as was the second. The procedure is repeated until all n-tuples are exhausted.

Exercises

8. Form a decoding table for the (5,3) code of Exercise 6.

2.4 The Generator and Parity-check Matrices

A k x n matrix whose rows are basis vectors for an (n,k) code C is called a *generator matrix*. Writing down

---

*Note that $a_k = -a_k$.

such a matrix is actually a compact way of specifying a code, for knowing its k rows enables us to determine all the 2^k codewords in the code. (This is a significant observation, for there are codes in use with k greater than fifty.)

The subspace of V_n orthogonal to C is denoted by $C^\perp$ and called the *dual code* of C. That is

$$C^\perp = \{x \in V_n \mid x \cdot y = 0 \text{ for all } y \in C\},$$

where $x \cdot y$ is the usual dot product of n-tuples. Since the dimension of $C^\perp$ is n-k, $C^\perp$ is an (n,n-k) code.*

Let H be an (n-k) x n generator matrix for $C^\perp$. Then

(7) $x \in C$ if and only if $xH^T = 0$.

Letting $x = (x_1, x_2, \ldots, x_n)$, and denoting by h_{ij} the entry in the ith row and jth column of H, we may rewrite Equation (7) equivalently as

(8) $x \in C$ if and only if $\sum_{j=1}^{n} x_j h_{ij} = 0$ for $i = 1, \ldots, n-k$.

Since we are working over GF(2), Equation (8) says x is a codeword if and only if the number of integers j for which both x_j and h_{ij} are 1 is *even* for each $i = 1, \ldots, n-k$. For this reason we call H a *parity-check matrix* for C.

Suppose G is a generator matrix for an (n,k) code C. The reduced echelon form of G also serves as a generator matrix for C. By permuting certain columns of the reduced echelon form of G we obtain a $k \times n$ matrix of the form $G' = [I_k P]$, where I_k is the $k \times k$ identity matrix and P is a $k \times (n-k)$ matrix (call the permutation involved ρ). The matrix G' can be thought of as a generator matrix for an (n,k) code C'. It can easily be verified that the rank

*A proof that $\dim C^\perp = n-k$ may be based upon the observation that $C^\perp$ can be identified as the solution space of a system of k linearly independent equations in n unknowns.

n-k matrix $H' = [P^T I_{n-k}]$ satisfies the equation

$$G'H'^T = 0,$$

from which we conclude that H' is a parity-check matrix for C'. Applying ρ^{-1} to the columns of H' will produce a parity-check matrix H for C.

Example 2. The matrix

$$G = \begin{bmatrix} 1 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 1 & 1 \\ 1 & 0 & 1 & 1 & 1 \end{bmatrix}$$

is a generator matrix for the (5,3) code of Exercise 6. The reduced echelon form of G is

$$E = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 \end{bmatrix}.$$

By applying the permutation

$$\rho = (243)$$

(written in cycle form) to the columns of E, we obtain the matrix

$$G' = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 \end{bmatrix},$$

which is of the form $[I_3 P]$. Then

$$H' = [P^T I_2] = \begin{bmatrix} 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 \end{bmatrix}$$

and applying

$$\rho^{-1} = (234)$$

to the columns of this matrix we get

$$H = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 \end{bmatrix}.$$

2.5 Systematic Codes

Codes like the code C' of the preceding section, which have a generator matrix of the form G', are called *systematic codes*. The study of systematic codes is greatly facilitated by the simple nature of their generator matrices. For let $r'_1, \ldots, r'_k$ be the rows of G'. Then $x \in C'$ if and only if there exist scalars (binary digits) $a_1, \ldots, a_k$ such that

$$x = \sum_1^k a_i r'_i.$$

Upon expanding this sum we get

$x \in C'$ if and only if $x = (a_1, \ldots, a_k, a_{k+1}, \ldots, a_n)$

(9) where $a_j = \sum_{i=1}^{k} a_i g'_{ij}$ for $j = k+1, \ldots, n$ and

g'_{ij} is the row i-column j entry in G'.

Thus the first k coordinates (the message digits) of a codeword in C' can be chosen arbitrarily while the remaining n-k coordinates (the check digits) are linear combinations of these message digits.

Example 3. If $C = (1,0,1,C_4,C_5)$ is to be a codeword in the systematic code C' generated by the matrix G' of Example 2, then we must have

$$C_4 = C_1 g'_{14} + C_2 g'_{24} + C_3 g'_{34} = 1 \cdot 0 + 0 \cdot 0 + 1 \cdot 0 = 0.$$

$$C_5 = C_1 g'_{15} + C_2 g'_{25} + C_3 g'_{35} = 1 \cdot 0 + 0 \cdot 0 + 1 \cdot 1 = 1.$$

Exercises

9. What relationship exists between the codewords of C and C'?

10. Find the matrices G', H', and H for the (6,3) code with generator matrix

$$G = \begin{bmatrix} 0 & 1 & 1 & 1 & 1 & 0 \\ 1 & 0 & 1 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 \end{bmatrix}.$$

11. Find the check digits for the codeword with message digits 111 in the (6,3) code of Exercise 10.

---

3. ERROR CORRECTION

3.1 A Criterion for Code Quality

Suppose that the channel is sufficiently reliable for the channel decoder to assume that at most t errors (i.e., t alterations, t a positive integer) can occur in a transmitted codeword. Can we then be one hundred percent certain that the decoder will decode correctly? If the answer is yes, for every codeword in the code, we say the code is *t error-correcting*. More precisely, a code C is t error-correcting if the closed balls

$$\{\bar{S}(c,t);\ c \in C\},$$

where

$$\bar{S}(c,t) = \{x \in V_n / d(x,c) \leq t\},$$

are pairwise disjoint.

In view of the fact that $w(x) = d(x,0)$, $d(x,y) = w(x-y)$, and a code is a subspace (so that x-y is in the code whenever x and y are), we may conclude that W, the minimum weight of all nonzero codewords, is equal to D, the minimum distance between different codewords. It seems intuitively clear that for a code to be t error-correcting, the codewords have to be "sufficiently far apart." Just how far apart is revealed in the next theorem.

Theorem 2. A code is t error-correcting if $D \geq 2t + 1$.

Proof: Suppose not. Then there exist two codewords, c_1 and c_2, such that the closed balls $\bar{S}(c_1,t)$ and $\bar{S}(c_2,t)$ have an n-tuple, call it r, in common. Then

(10) $$d(c_1,c_2) \leq d(c_1,r) + d(r,c_2) \leq t + t < 2t + 1.$$

But,

(11) $$d(c_1,c_2) = w(c_1-c_2) \geq W = D \geq 2t + 1,$$

so we have arrived at a contradiction and the theorem is proven.

Exercises

12. Establish the converse of Theorem 2. That is, show a code is t error-correcting only if $D \geq 2t + 1$.

13. Comment on the error-correcting capabilities of the code in Exercise 6.

3.2 Error Correction and the Parity-check Matrix

An extremely elegant and simple characterization of t error-correcting codes may be expressed in terms of the parity-check matrix.

Theorem 3. Let H be a parity-check matrix for a code C. If every subset of 2t columns of H is linearly independent, then C is t-error-correcting.

This result is an immediate consequence of Theorem 2 and the following lemma.

Lemma 1. If C has a codeword c of weight w, then some w columns of H are linearly dependent.

Proof: Let $c = (c_1, \ldots, c_n)$. Now $w(c) = w$ means that exactly w of the $\{c_i\}$ are 1 (which we may assume to be, without loss of generality, coordinates 1, 2, ..., w). Now $c \in C$ means

(12) $$cH^T = 0.$$

Denoting the columns of H by $h_1, \ldots, h_n$, Equation (12) may be rewritten equivalently as

(13) $$\sum_i^n c_i h_i = 0$$

or, using the values of the c_i,

(14) $$\sum_1^w h_i = 0.$$

Equation (14) says the first w columns of H are linearly dependent and the lemma is proven.

3.3 Hamming Codes

We put Theorem 3 to immediate use. Let m be a positive integer and let H be the $m \times (2^m-1)$ matrix whose columns are the binary representations of the integers 1, ..., 2^m-1 respectively. For m = 3,

$$H = \begin{pmatrix} 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 & 1 & 0 & 1 \end{pmatrix}.$$

Since H contains the m columns of the identity matrix I_m, rank H = m and H can serve as a parity-check matrix for a $(2^m-1, 2^m-1-m)$ code C, a type of code referred to as a *Hamming code*. Since two nonzero binary n-tuples are linearly dependent if and only if they are identical, every pair of columns of H are linearly independent. Hence, by virtue of Theorem 3, *Hamming codes are single error-correcting.*

Moreover, with the aid of the matrix H the decoder can easily correct any single error. For suppose a single error occurs (say in position i) in the transmitted codeword c in C, so that the channel decoder receives an n-tuple r which differs from c only in position i. Then, since $cH^T = 0$,

$$rH^T = (r-c)H^T = (0, \ldots 1_i, \ldots 0)H^T = h_i,$$

(the ith column of H, i.e., the binary representation of i), thus enabling the decoder to determine the position of the error.

Exercises

14. Find a parity-check matrix for the (15,11) Hamming code.

15. Find a generator matrix for the (7,4) Hamming code.

16. Working with the (7,4) Hamming code and assuming no more than one error occurs during transmission, what codeword was transmitted if (0,0,1,1,1,1,1) was received?

3.4 Perfect Codes

Recall that a code is t error-correcting if the closed balls $\{\bar{S}(c,t)\}$ of radius t with centers at the codewords are pairwise disjoint. If, furthermore, these balls fill the space (i.e., their union is all of V_n), the code is said to be *perfect*. To show that Hamming codes are perfect, we may use the following counting argument. Let $n = 2^m-1$ and let C be the (n,n-m) Hamming code. Now $\#V_n = 2^n$. Each of the closed balls $\bar{S}(c,1)$ contains n+1 n-tuples. Since $\#C = 2^k$, where $k = 2^m-1-m = n-m$, and the balls are pairwise disjoint, the union of the balls contains $2^{n-m}(n+1)$ n-tuples. A little arithmetic will show that $2^n = 2^{n-m}(n+1)$ when $n = 2^m-1$.

3.5 The Baseball Pool Problem

Hamming codes can be used to supply a simple (albeit partial) solution to a rather intriguing problem in combinatorics, the Baseball Pool Problem: "On a given day n baseball games are to be played. If a single bet is defined as picking the winner in each of the n games, what is the minimum number of bets one has to make to guarantee choosing at least n-1 winners?" Clearly, 2^{n-1} bets are sufficient to guarantee at least n-1 winners. We solve the problem for n of the form 2^m-1. By denoting a home team victory by 1 and a home team defeat by 0, each of the 2^n possible bets can be associated with a binary n-tuple (i.e., a vector in V_n). Noting that Hamming codes are single error-correcting and perfect, the only bets we need place to guarantee ourselves at least n-1 winners are the 2^{n-m} n-tuples in the (n,n-m) Hamming code. Thus, when $n = 2^m-1$, we have reduced the sufficiency number from 2^{n-1} to 2^{n-m}.

Exercises

17. Show that 2^{n-m} is indeed a minimum.

4. REFERENCES

Needless to say, we have only scratched the surface of coding in this unit. We have concentrated upon the linear algebra aspects of the subject while hardly speaking at all about the large role played by modern algebra. If this module has wetted your appetite, you may wish to consult the texts listed below.

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

Blake, I.F., and Mullin, R.C., The Mathematical Theory of Coding, Academic Press, New York, 1975.

Lin, S., An Introduction to Error-Correcting Codes, Prentice-Hall, Englewood Cliffs, 1970.

MacWilliams, F.J., and Sloane, N.J.A., The Theory of Error-Correcting Codes, North-Holland, New York, 1978.

Peterson, W.W., and Weldon, E.J., Error-Correcting Codes, MIT Press, Cambridge, 1972.

Van Lint, J.H., Coding Theory, Springer-Verlag, New York, 1973.

5. ANSWERS TO EXERCISES

1.

11000	00110	10011	01101
11001	00110	10010	01100
11010	00100	10001	01111
11100	00010	10111	01001
10000	01110	11011	00101
01000	10110	00011	11101
11110	00000	01011	10101
01010	10100	11111	00001

The 5-tuples below the codewords and above the dotted line differ from the codeword at the top in one position and are uniquely decodeable. Those 5-tuples below the dotted line differ in two positions from two or more codewords and are *not* uniquely decodeable using only the principle of maximum likelihood.

2. 11000 will be decoded correctly if and only if a 5-tuple in its column is received. One of these 5-tuples differs from 11000 in zero positions, five differ in one position, and two differ in two positions. Hence the probability of correct decoding is $q^5 + 5pq^4 + 2p^2q^3$.

3. Since $p < q$, $1/2 < q < 1$. Then

$$q^4 + 3pq^3 - q^2 = q^4 + 3(1-q)q^3 - q^2 = -2q^4 + 3q^3 - q^2$$
$$= q^2(-2q^2+3q-1) = 2q^2(1/2-q)(q-1) > 0.$$

4. Since there are only two scalars, there are 2^k codewords in an (n,k) code.

5. For three nonzero vectors in V_n to be linearly dependent, we must have either
   (i) two or more of the vectors are equal, or
   (ii) one of the vectors is the sum of the other two.
   Since neither of these condtions is true for the three vectors given, the vectors are linearly independent.

6. We find the seven nonzero codewords by forming all possible sums of the basis vectors taken 1, 2, and 3 at a time. The codewords are 00000, 10100, 10011, 10111, 00100, 00011, 00111, 10000.

7. As (i) and (ii) of Equation (3) are obvious, we shall only prove (iii)

$$d(x,y) = w(x-y) = w(x-z+z-y) \leq w(x-z) + w(z-y) = d(x,z) + d(z,y).$$

8. As in Exercise 1, the table below is not the only possible one.

00000	10100	10011	10111	00100	00011	00111	10000
01000	11000	11011	11111	01100	01011	01111	11000
00010	10110	10001	10101	00110	00001	00101	10010
01010	11110	11001	11101	01110	01001	01101	11010

9. The codewords of C' can be obtained by applying ρ to the codewords of C.

10. The reduced echelon form of G is

$$E = \begin{pmatrix} 1 & 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 \end{pmatrix}.$$

By applying the permutation $\rho = (34)$ to the columns of E, we obtain

$$G' = \begin{pmatrix} 1 & 0 & 0 & 1 & 1 & 1 \\ 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 \end{pmatrix},$$

which is of the form $[I_3 P]$. Then

$$H' = [P^T I_3] = \begin{pmatrix} 1 & 1 & 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 \\ 1 & 1 & 1 & 0 & 0 & 1 \end{pmatrix}$$

and applying $\rho^{-1} (= \rho)$ to the columns of this matrix we get

$$N = \begin{pmatrix} 1 & 1 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 & 1 & 0 \\ 1 & 1 & 0 & 1 & 0 & 1 \end{pmatrix}.$$

11. Let C' be the systematic code generated by the matrix G' of Exercise 10. Then $c' = (1,1,1,c'_4,c'_5,c'_6)$ is the codeword of C' with message digits 111. Furthermore,

$$c'_4 = c'_1g'_{14} + c'_2g'_{24} + c'_3g'_{34} = 1\cdot1 + 1\cdot1 + 1\cdot0 = 0$$

$$c'_5 = c'_1g'_{15} + c'_2g'_{25} + c'_3g'_{35} = 1\cdot1 + 1\cdot0 + 1\cdot1 = 0$$

$$c'_6 = c'_1g'_{16} + c'_2g'_{26} + c'_3g'_{36} = 1\cdot1 + 1\cdot1 + 1\cdot1 = 1$$

Thus $c' = (1,1,1,0,0,1)$ and the codeword in C with message digits 111 is, using the idea in Exercise 9 and the fact that $\rho^{-1} = (34),(1,1,0,1,0,1)$.

12. Suppose $D < 2t + 1$. Then there exist two codewords x and y such that $d(x,y) < 2t + 1$. Let $r \in V_n$ be obtained from x by changing m of the digits in which x and y differ, where $\frac{1}{2}d(x,y) \leq m \leq t$. Then

$$d(x,r) = m \leq t \text{ and } d(y,r) \leq \tfrac{1}{2}d(x,y) \leq t \Rightarrow$$

the code is not t error-correcting.

13. The minimum weight of the code is 1 so that it has extremely poor error-correcting capabilities. In fact a single error in the codeword 00000 will produce the codeword 00100.

14. $$H = \begin{pmatrix} 0&0&0&0&0&0&0&1&1&1&1&1&1&1&1 \\ 0&0&0&1&1&1&1&0&0&0&0&1&1&1&1 \\ 0&1&1&0&0&1&1&0&0&1&1&0&0&1&1 \\ 1&0&1&0&1&0&1&0&1&0&1&0&1&0&1 \end{pmatrix}.$$

15. As a parity-check matrix for a code C may be thought of as a generator matrix for the code $C^{\perp}$, and a parity-check matrix for $C^{\perp}$ regarded as a generator matrix for C, all we need do is apply the same technique as in Example 2 with the roles of G and H reversed. Doing so, we find

$$G = \begin{pmatrix} 1&1&1&0&0&0&0 \\ 1&0&0&1&1&0&0 \\ 0&1&0&1&0&1&0 \\ 1&1&0&1&0&0&1 \end{pmatrix}$$ is a generator matrix for the (7,4) Hamming code.

16. $rH^T = (0011111)\begin{pmatrix} 0&0&1 \\ 0&1&0 \\ 0&1&1 \\ 1&0&0 \\ 1&0&1 \\ 1&1&0 \\ 1&1&1 \end{pmatrix} = (011)$, which is the binary representation of 3. Thus there is a single error in position 3 and the codeword transmitted was (0,0,0,1,1,1,1).

17. Suppose A is a collection of bets (i.e., a subset of V_n) which guarantee at least n-1 winners. Then $V_n = \bigcup_{a \in A} \{\bar{S}(a,1)\} \Rightarrow$

$$\Rightarrow \#V_n = \#\left(\bigcup_{a \in A} \{\bar{S}(a,1)\}\right) \leq \sum_{a \in A} \#\{\bar{S}(a,1)\}$$

$$= (\#A)(n+1) \Rightarrow \#A \geq 2^{n-m}.$$

STUDENT FORM 1

Request for Help

Return to:
EDC/UMAP
55 Chapel St.
Newton, MA 02160

Student: If you have trouble with a specific part of this unit, please fill out this form and take it to your instructor for assistance. The information you give will help the author to revise the unit.

Your Name_____ Unit No._____

Page_____
- ○ Upper
- ○ Middle
- ○ Lower

OR

Section_____
Paragraph_____

OR

Model Exam
Problem No._____
Text
Problem No._____

Description of Difficulty: (Please be specific)

Instructor: Please indicate your resolution of the difficulty in this box.

○ Corrected errors in materials. List corrections here:

○ Gave student better explanation, example, or procedure than in unit. Give brief outline of your addition here:

○ Assisted student in acquiring general learning and problem-solving skills (not using examples from this unit.)

Instructor's Signature_____

Please use reverse if necessary.

STUDENT FORM 2

Unit Questionnaire

Return to:
EDC/UMAP
55 Chapel St.
Newton, MA 02160

Name_____ Unit No._____ Date_____

Institution_____ Course No._____

Check the choice for each question that comes closest to your personal opinion.

1. How useful was the amount of detail in the unit?

   ____Not enough detail to understand the unit
   ____Unit would have been clearer with more detail
   ____Appropriate amount of detail
   ____Unit was occasionally too detailed, but this was not distracting
   ____Too much detail; I was often distracted

2. How helpful were the problem answers?

   ____Sample solutions were too brief; I could not do the intermediate steps
   ____Sufficient information was given to solve the problems
   ____Sample solutions were too detailed; I didn't need them

3. Except for fulfilling the prerequisites, how much did you use other sources (for example, instructor, friends, or other books) in order to understand the unit?

   ____A Lot ____Somewhat ____A Little ____Not at all

4. How long was this unit in comparison to the amount of time you generally spend on a lesson (lecture and homework assignment) in a typical math or science course?

Much Longer	Somewhat Longer	About the Same	Somewhat Shorter	Much Shorter
____	____	____	____	____

5. Were any of the following parts of the unit confusing or distracting? (Check as many as apply.)

   ____Prerequisites
   ____Statement of skills and concepts (objectives)
   ____Paragraph headings
   ____Examples
   ____Special Assistance Supplement (if present)
   ____Other; please explain_____

6. Were any of the following parts of the unit particularly helpful? (Check as many as apply.)

   ____Prerequisites
   ____Statement of skills and concepts (objectives)
   ____Examples
   ____Problems
   ____Paragraph headings
   ____Table of Contents
   ____Special Assistance Supplement (if present)
   ____Other; please explain_____

Please describe anything in the unit that you did not particularly like.

Please describe anything that you found particularly helpful. (Please use the back of this sheet if you need more space.)

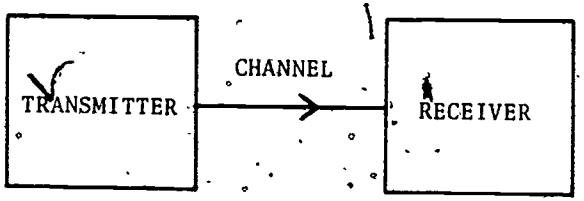
umap

UNIT 337

MODULES AND MONOGRAPHS IN UNDERGRADUATE MATHEMATICS AND ITS APPLICATIONS PROJECT

A DOUBLE-ERROR CORRECTING CODE

by Gary J. Sherman



APPLICATIONS OF ALGEBRA TO INFORMATION THEORY

edc/umap /55 chapel st./ newton, mass. 02160

A DOUBLE-ERROR CORRECTING CODE

by

Gary J. Sherman
Department of Mathematics
Rose-Hulman Institute of Technology
Terre Haute, Indiana 47803

TABLE OF CONTENTS

Intermodular Description Sheet: UMAP Unit 337

Title: A DOUBLE-ERROR CORRECTING CODE

Author: Gary J. Sherman
Department of Mathematics
Rose-Hulman Institute of Technology
Terre-Haute, IN 47803

Review Stage/Date: III 7/16/79

Classification: APPL ALG/INFORMATION THEORY

Prerequisite Skills:
1. Elementary linear algebra (matrix notation, arithmetic of vectors in finite vector spaces over Z_2, multiplication and division of polynomials).

Output Skills:
1. To see a brief non-technical introduction to applied algebra.
2. To provide motivation to learn abstract algebra.

Other Related Units:

Aspects of Coding *(Unit 336)*
Error Correcting Codes I *(Unit 346)*

The Project would like to thank Martin E. Flashman and Jack M. Robertson for their reviews, and all others who assisted in the production of this unit.

This material was prepared with the support of National Science Foundation Grant No. SED76-19615 A02. Recommendations expressed are those of the author and do not necessarily reflect the views of the NSF, nor of the National Steering Committee.

A DOUBLE-ERROR CORRECTING CODE

1. INTRODUCTION

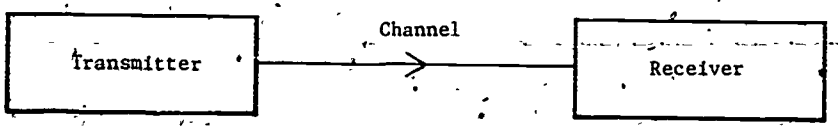
Algebraic coding theory originated in the late 1940's in the attempt to solve the problem of transmitting an electronic message through a noisy channel. From its beginning, as a hybrid of algebraic and probabilistic results, the theory has developed, while shedding light on the original engineering problem, to the point where it is being applied in other areas of mathematics (e.g., group theory and combinatorics).

This module will provide you with a brief introduction to algebraic coding theory via an example: we will construct a double-error correcting code. Only elementary algebraic techniques will be used. The prerequisites are an elementary linear algebra course and some facility for manipulating polynomials.

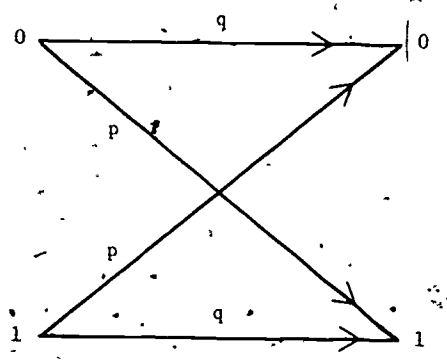
If, after reading this module, you would like to learn more algebraic coding theory, then The Theory of Error Correcting Codes I and II by F.J. MacWilliams and N.J.A. Sloane is the book to see.

2. THE PROBLEM

The following simple model is appropriate for the study of many communication problems.



For example, the model could represent a satellite transmitting radio signals to a station on earth or the transmission of telephone signals along a cable. For our purposes we assume that the transmitter can produce and send two symbols, 0 and 1, along the channel to the receiver. In practice all channels are noisy; i.e., occasionally, a 1 is transmitted and a 0 is received, or a 0 is transmitted and a 1 is received. We assume that there is a fixed nonzero probability, p, of incorrect transmission. The probability of correct transmission is q = 1-p. Such a channel is called a binary symmetric channel.



Let's be specific. Suppose you transmit a message, $m = m_1 m_2 \cdots m_{15}$, consisting of a sequence of fifteen symbols chosen from {0,1} through the channel to the receiver. If $r = r_1 r_2 \cdots r_{15}$ is received can the transmitted message be recovered by the receiver? Since p > 0, r could differ from m in up to fifteen places. From the receiver's point of view any one of 2^{15} possible messages could have been transmitted! However, if p is close to zero (a reliable channel) he or she would not expect many errors in transmission; i.e., the message should be near the received word in the sense that they do not differ in many places. Throughout this discussion we will assume p is so close to zero that more than two errors in transmission are unlikely. (If you know some probability: the probability of two or fewer

errors in transmission is

$$q^{15} + 15pq^{14} + \binom{15}{2}p^2q^{13}$$

which approaches one as p approaches zero.)

For example, if r = 111111111111111, then the message is likely to be among those words containing two or fewer 0's. Even with this restriction, the receiver has $\binom{15}{0} + \binom{15}{1} + \binom{15}{2} = 121$ messages to choose from. Given r, the solution to the receiver's dilemma is obvious. Prior to transmission it should have been agreed that only one message with two or fewer 0's is a candidate for transmission; say m = 110111111111111. Given this restriction the receiver could conclude

(i) that errors were made in transmission—probably a single error in the third bit,

(ii) that the message was (probably) m = 110111111111111.

The problem: Can a set, C, of message words be chosen from the set of 2^{15} binary fifteen-tuples so that the occurrence of two or fewer errors in transmission can be detected and corrected by the receiver? We will refer to such a set as a *code* and to its elements as *codewords*.

Some obvious choices for C are {000000000000000}, {111111111111111} and {000000000000000,111111111111111}. But, while these choices for C satisfy our error detection and correction requirements, they do not enable us to transmit much information. Ideally C should contain as many codewords as possible. However, since there are $2^{2^{15}}-1$ non-empty subsets of binary fifteen-tuples (more than a billion) one cannot rummage through them at random looking for a large code. We must restrict our attention to sets of binary fifteen-tuples which possess some sort of regular structure. Our approach is algebraic.

3. CODEWORDS AS VECTORS

We denote the set of binary fifteen-tuples by Z_2^{15} where Z_2 is the field with two elements, 0 and 1, and operations, + and ·, given by the table below.

TABLE I

Addition and Multiplication in Z_2

+	0	1
0	0	1
1	1	0

·	0	1
0	0	0
1	0	1

Using the addition of Z_2 to add elements of Z_2^{15} componentwise

$$\begin{array}{r} 1\,1\,0\,1\,1\,1\,0\,0\,1\,0\,1\,1\,1\,0\,1 \\ +\;0\,1\,1\,0\,1\,0\,1\,1\,1\,1\,0\,1\,0\,1\,1 \\ \hline 1\,0\,1\,1\,0\,1\,1\,1\,0\,1\,1\,0\,1\,1\,0 \end{array}$$

enables us to view Z_2^{15} as a vector space of dimension 15 over (the finite field) Z_2. This observation is useful for the following reasons.

(i) The introduction of errors by the channel can be described algebraically. If m is transmitted and r is received, then we can write

(1) $\quad r = m + e,$

where $e = e_1e_2\cdots e_{15}$ is called the *error word* and is defined as follows:

$$e_i = \begin{cases} 0 & \text{if } r_i = m_i \\ 1 & \text{if } r_i \neq m_i \end{cases}$$

Notice that Equation (1) is equivalent to

(2) $\quad r + e = m$

since e + e = 0 (remember, 1 + 1 = 0). Thus, finding e and adding to r yields the transmitted message.

Problem 1. Compute e if m = 101010101010101 and r = 101010111010100.

(ii) Subspaces of Z_2^{15} are obvious candidates to serve as structured sets of message words—codes. Better yet, they are easy to construct! We can take $C \subseteq Z_2^{15}$ to be the set of solutions to the equation

(3) $\quad Hm^t = 0;$

where H is some $n \times 15$ binary matrix and m^t is the transpose of $m = m_1 m_2 \cdots m_{15}$. The matrix H is called the *parity-check matrix* of the code C it determines. (In the literature, H is called a *Hamming matrix*, and C a *Hamming code*.)

(iii) The receiver can use the algebraic descriptions of C and e to advantage. It follows from (2) and (3) that

$$Hr^t = H(m+e)^t = H(m^t+e^t) = Hm^t + He^t = He^t.$$

Case 1. $Hr^t \neq 0$. This means that an error (or errors) occurred in transmission since $r \notin C$. Moreover the error-word is among the solutions to the nonhomogeneous equation $He^t = Hr^t$.

Case 2. $Hr^t = 0$. This means that r is a codeword. Either transmission was error-free or the error-word satisfies $He^t = 0$; i.e., the error-word is a codeword.

These remarks illustrate the importance of the vector Hr^t to the error detection and correction process. We will refer to Hr^t as the *syndrome* of r and denote it by s.

4. A SINGLE-ERROR CORRECTING CODE

To use the observations in Section 3 we need to specify a parity-check matrix. Let's try

$$H_1 = \begin{bmatrix} 1&1&1&1&1&1&1&1&1&1&1&1&0&0&0 \\ 1&0&1&0&1&0&1&0&1&0&1&0&0&1&0 \\ 0&1&0&1&0&1&0&1&0&1&0&1&0&0&1 \end{bmatrix}.$$

The code, C_1, is the set of solutions to $H_1 m^t = 0$ which is equivalent to the system:

$$m_1 + m_2 + m_3 + m_4 + m_5 + m_6 + m_7 + m_8 + m_9 + m_{10} + m_{11} + m_{12} = m_{13}$$

$$m_1 + m_3 + m_5 + m_7 + m_9 + m_{11} = m_{14}$$

$$m_2 + m_4 + m_6 + m_8 + m_{10} + m_{12} = m_{15}.$$

These equations are called the *parity-check equations* of C_1. As written, they imply that $m_1, m_2, \cdots, m_{12}$ may be chosen freely from 0,1 as long as m_{13}, m_{14}, and m_{15} are chosen to be the appropriate sums. (You might think of $m_1, m_2, \cdots, m_{12}$ as information bits and m_{13}, m_{14}, and m_{15} as check bits.) This observation (or the number of columns of H_1 minus the row rank of H_1) implies that the dimension of C_1 is 12 and $|C_1| = 2^{12}$.

Problem 2. List a few of the codewords.

With this code you can transmit a lot of messages. How does the receiver fare? Let's suppose you transmitted m and he received r = 111111111110000. Since $H_1 r^t = \begin{bmatrix} 1 \\ 0 \\ 1 \end{bmatrix}$, at least one error occurred in transmission and the error-word is among the solutions to $H_1 e^t = \begin{bmatrix} 1 \\ 0 \\ 1 \end{bmatrix}$, which is equivalent to the nonhomogeneous system

$$e_1 + e_2 + e_3 + e_4 + e_5 + e_6 + e_7 + e_8 + e_9 + e_{10} + e_{11} + e_{12} + e_{13} = 1$$

$$e_1 + e_3 + e_5 + e_7 + e_9 + e_{11} + e_{14} = 0$$

$$e_2 + e_4 + e_6 + e_8 + e_{10} + e_{12} + e_{15} = 1.$$

The receiver might solve this system and look for error-words in the solution set which have 1's in only one or two positions. A shortcoming of this approach is that after solving the system the receiver must examine 2^{12} words. This is bound to be time consuming (expensive)!

A more efficient approach to error identification is to view the syndrome of r, $\begin{bmatrix}1\\0\\1\end{bmatrix} = H_1 e^t$, as a linear combination of the columns of H_1:

$$e_1\begin{bmatrix}1\\1\\0\end{bmatrix} + e_2\begin{bmatrix}1\\0\\1\end{bmatrix} + e_3\begin{bmatrix}1\\1\\0\end{bmatrix} + e_4\begin{bmatrix}1\\0\\1\end{bmatrix} + e_5\begin{bmatrix}1\\1\\0\end{bmatrix} + e_6\begin{bmatrix}1\\0\\1\end{bmatrix} + e_7\begin{bmatrix}1\\1\\0\end{bmatrix} +$$
$$+ e_8\begin{bmatrix}1\\0\\1\end{bmatrix} + e_9\begin{bmatrix}1\\1\\0\end{bmatrix} + e_{10}\begin{bmatrix}1\\0\\1\end{bmatrix} + e_{11}\begin{bmatrix}1\\1\\0\end{bmatrix} + e_{12}\begin{bmatrix}1\\0\\1\end{bmatrix} + e_{13}\begin{bmatrix}1\\0\\0\end{bmatrix} +$$
$$+ e_{14}\begin{bmatrix}0\\1\\0\end{bmatrix} + e_{15}\begin{bmatrix}0\\0\\1\end{bmatrix} = \begin{bmatrix}1\\0\\1\end{bmatrix}.$$

The error-words with a single 1 are easy to find: the 1 must occur in a position corresponding to a column which is equal to the syndrome. All such error-words, and the corresponding messages, are listed in the following array,

r	+	e	=	m
111111111110000	+	010000000000000	=	101111111110000
		000100000000000		111011111110000
		000001000000000		111110111110000
		000000010000000		111111101110000
		000000000100000		111111111010000
		000000000001000		111111111111000

which illustrates the receiver's problem with this code. He has no way of knowing which one of the six equally likely error-words actually occurred.

Problem 3. If r = 111111111110000 and two errors occurred in transmission, what was m?

To avoid this difficulty the columns of the parity-check matrix should be distinct. This is impossible using a 3×15 parity-check matrix since only eight binary three-tuples exist. Let's try a 4×15 parity-check matrix, say

$$H_2 = \begin{bmatrix}0&0&0&0&0&0&0&1&1&1&1&1&1&1&1\\0&0&0&1&1&1&1&0&0&0&0&1&1&1&1\\0&1&1&0&0&1&1&0&0&1&1&0&0&1&1\\1&0&1&0&1&0&1&0&1&0&1&0&1&0&1\end{bmatrix}.$$

Problem 4. Why is it a bad idea to have a column of 0's in a parity-check matrix?

The dimension of C_2 is eleven so there are 2^{11} codewords; fewer than in C_1, but the receiver can now detect and correct any single error in transmission. Indeed, a single error occurs in the ith position if, and only if,

$$s = H_2 r^t = H_2 e^t = c_i,$$

where c_i denotes the ith column of H_2.

Example 1. If r = 110000111111101, then transmission was incorrect since

$$H_2 r^t = \begin{bmatrix}1\\0\\1\\0\end{bmatrix} \neq 0.$$

Assuming a single error occurred, it was in the tenth position since the syndrome is the tenth column of H_2. Thus

$$m = 110000111111101 + 000000000100000$$
$$= 110000111011101.$$

Note that each column of H_2 can be viewed as the binary representation of the column number. Thus $H_2 r^t = \begin{bmatrix}1\\0\\1\\0\end{bmatrix}$ implies the error (assuming a single error) occurred in the tenth position. The receiver simply changes the tenth bit of r to recover the message.

Problem 5. Show that if r = 111100101001101, then transmission was incorrect. Assume a single error occurred, and recover the message. Show that more than one double error can give the same syndrome as r.

---

5. A DOUBLE-ERROR-CORRECTING CODE

Are H_2 and C_2 useful on double errors? If two errors occur in transmission, say in the ith and jth positions, then

$$s = H_2 r^t = H_2 e^t = c_i + c_j.$$

(i) Detection? Since the columns of H_2 are distinct and nonzero, $s = c_i + c_j \neq 0$, which implies transmission was incorrect.

(ii) Correction? Not hardly! The columns corresponding to the error locations are not uniquely determined by the syndrome (see Problem 5). This is because $c_i + c_j = s$ is one (vector) equation in two unknowns.

Another equation in c_i and c_j would be helpful. If we continue to think of parity-check matrices as matrices of columns, then an 8×15 matrix of the form

$$H_3 = \left[\begin{array}{c|ccc} c_1 & c_2 & \cdots & c_{15} \\ \hline f(c_1) & f(c_2) & \cdots & f(c_{15}) \end{array}\right],$$

where c_m is the mth column of H_2 and $f(c_m)$ is some c_n, provides a second equation in c_i and c_j. Specifically,

$$H_3 r^t = \left[\begin{array}{c} c_i \\ \hline f(c_i) \end{array}\right] + \left[\begin{array}{c} c_j \\ \hline f(c_j) \end{array}\right] = \left[\begin{array}{c} s_1 \\ \hline s_2 \end{array}\right]$$

implies

$$\begin{aligned} c_i + c_j &= s_1, \\ f(c_i) + f(c_j) &= s_2, \end{aligned} \tag{4}$$

where the eight-component syndrome $H_3 r^t$ is written as two four-component syndromes.

Whether (4) can be solved uniquely for c_i and c_j depends on how the function f is defined. Given our ability to add columns, and multiply them by 0 or 1, about the only algebraic choice for f is linear:

$$f(c_m) = bc_m + c_k,$$

where $b \in Z_2$ and c_k is a four-component binary column. With this definition of f, the second equation in (4) is

$$\begin{aligned} (bc_i + c_k) + (bc_j + c_k) &= s_2, \\ b(c_i + c_j) + (c_k + c_k) &= s_2, \\ b(c_i + c_j) &= s_2. \end{aligned}$$

Either choice of b leads to a redundant second equation ($c_i + c_j = s_2$ or $0 = s_2$) and provides no help in determining c_i and c_j.

Something nonlinear is required. Maybe $f(c_m) = (c_m)^2$ will do, if we can invent a multiplication procedure for four-component columns. The key is to associate each four-component column with a binary polynomial of degree three or less. This identification process

$$\begin{bmatrix}0\\1\\1\\0\end{bmatrix}^t = 0\ 1\ 1\ 0 \leftrightarrow x^2 + x$$

$$\begin{bmatrix}1\\0\\1\\1\end{bmatrix}^t = 1\ 0\ 1\ 1 \leftrightarrow x^3 + x + 1$$

preserves *column addition*.

$$0110 + 1011 = 1101 \leftrightarrow x^3 + x^2 + 1 = (x^2 + x) + (x^3 + x + 1),$$

and makes *column multiplication* possible

$$\begin{aligned}(0110) \cdot (1011) &\leftrightarrow (x^2 + x)(x^3 + x + 1) \\ &= x^5 + x^4 + x^3 + x^2 + x^2 + x \\ &= x^5 + x^4 + x^3 + x \leftrightarrow 111010,\end{aligned} \tag{5}$$

although closure is not guaranteed, as you can see. This deficiency is not fatal. We can identify $x^5 + x^4 + x^3 + x$ with a polynomial of degree at most three, namely its remainder upon division by a (previously agreed upon) fourth degree polynomial, say $p(x) = x^4 + x^3 + 1$. Since

$$\begin{array}{r|l} & x \\ \hline x^4 + x^3 + 1 & x^5 + x^4 + x^3 + x \\ & \underline{x^5 + x^4 \qquad + x} \\ & x^3 \end{array} \tag{6}$$

$x^5 + x^4 + x^3 + x$ is identified with x^3. We denote this identification by $x^5 + x^4 + x^2 + x \equiv x^3$ and say $x^5 + x^4 + x^3 + x$ is congruent to x^3 modulo $x^4 + x^3 + 1$. It follows from (5) and (6) that

$$(0110) \cdot (1011) = 1000;$$

that is,

$$c_6 \cdot c_{11} = c_8.$$

Problem 6. Which column of H_2 serves as the multiplicative identity under the multiplication just introduced? Show that $c_{12} \cdot c_2 = c_1$ and $c_{13} \cdot c_{12} = c_{10}$.

Although we will not prove it, the columns of H_2, together with the zero column, form a finite field with respect to the addition and multiplication we have just introduced. Less precisely, we can manipulate the columns algebraically as if they were real numbers. For example, it follows from the results of Problem 6 that

$$c_{13}/c_2 = c_{13} \cdot c_{12} = c_{10}$$

since c_{12} is the multiplicative inverse of c_2. The following table will enable us to multiply and divide columns modulo $x^4 + x^3 + 1$ quite easily.

TABLE 2

Column Multiplication for the Matrix H_2 of Section 4

(7)	$(c_2)^2 = c_4$	$(c_2)^7 = c_7$	$(c_2)^{12} = c_3$
	$(c_2)^3 = c_8$	$(c_2)^8 = c_{14}$	$(c_2)^{13} = c_6$
	$(c_2)^4 = c_9$	$(c_2)^9 = c_5$	$(c_2)^{14} = c_{12}$
	$(c_2)^5 = c_{11}$	$(c_2)^{10} = c_{10}$	$(c_2)^{15} = c_1$
	$(c_2)^6 = c_{15}$	$(c_2)^{11} = c_{13}$	

Problem 7. Verify that $(c_2)^5 = c_{11}$ and determine the multiplicative inverse of each c_i, $1 \leq i \leq 15$.

By now you have probably guessed that choosing $p(x) = x^4 + x^3 + 1$ to get multiplicative closure was no accident. It wasn't. A discussion of the properties of $x^4 + x^3 + 1$ which guarantee such a nice column algebra can

be found in The Theory of Error-Correcting Codes I by MacWilliams and Sloane.

The requirement $f(c_m) = (c_m)^2$ is legitimate now that we have invented a way to square columns, but it is not useful. The second equation in (4) becomes

$$(c_i)^2 + (c_j)^2 = s_2,$$

which is redundant, since

$$\begin{aligned}(s_1)^2 &= (c_i + c_j)^2 \\ &= (c_i)^2 + c_j \cdot c_i + c_i \cdot c_j + (c_j)^2 \\ &= (c_i)^2 + (c_i \cdot c_j + c_i \cdot c_j) + (c_j)^2 \quad \text{(Remember, 1+1=0.)} \\ &= (c_i)^2 + (c_j)^2 \\ &= s_2.\end{aligned}$$

That is, the second equation is the square of the first.

Pressing on, we try $f(c_m) = (c_m)^3$. The associated system is

$$\begin{aligned}c_i + c_j &= s_1, \\ (c_i)^3 + (c_j)^3 &= s_2.\end{aligned} \tag{8}$$

Notice that

$$\begin{aligned}s_2 &= (c_i)^3 + (c_j)^3 \\ &= (c_i + c_j)\left[(c_i)^2 + c_i \cdot c_j + (c_j)^2\right] \\ &= s_1 \cdot \left[c_i \cdot c_j + (c_i)^2 + (c_j)^2\right] \\ &= s_1 \cdot \left[c_i \cdot c_j + (c_i + c_j)^2\right] \\ &= s_1 \cdot \left[c_i \cdot c_j + (s_1)^2\right].\end{aligned}$$

Since $s_1 \neq 0$,

$$s_2/s_1 = c_i \cdot c_j + (s_1)^2,$$

so that

$$c_i \cdot c_j = (s_1)^2 + s_2/s_1,$$

which implies

$$c_j = \frac{(s_1)^2 + s_2/s_1}{c_i}.$$

Substituting for c_j in $c_i + c_j = s_1$ yields

$$c_i + \frac{(s_1)^2 + s_2/s_1}{c_i} = s_1,$$

which is equivalent to

$$(c_i)^2 + s_1 \cdot c_i + \left[(s_1)^2 + s_2/s_1\right] = 0.$$

Similarly,

$$(c_j)^2 + s_1 \cdot c_j + \left[(s_1)^2 + s_2/s_1\right] = 0.$$

Thus the columns c_i and c_j associated with errors in the ith and jth positions must satisfy the quadratic equation

$$z^2 + s_1 \cdot z + \left[(s_1)^2 + s_2/s_1\right] = 0.$$

If only one error occurs, say in the ith position, then $c_i = s_1$ and $(s_1)^3 = s_2$.

We have solved the problem posed in Section 2. The code, C_3, is the set of solutions to $H_3 m^t = 0$, where

$$H_3 = \left[\begin{array}{c|ccc} & c_1 & c_2 & \cdots & c_{15} \\ \hline & (c_1)^3 & (c_2)^3 & \cdots & (c_{15})^3 \end{array}\right]$$

$$= \left[\begin{array}{ccccccccccccccc} 0&0&0&0&0&0&0&1&1&1&1&1&1&1&1 \\ 0&0&0&1&1&1&1&0&0&0&0&1&1&1&1 \\ 0&1&1&0&0&1&1&0&0&1&1&0&0&1&1 \\ 1&0&1&0&1&0&1&0&1&0&1&0&1&0&1 \\ \hline 0&1&1&1&0&0&1&0&0&0&0&0&1&0&1 \\ 0&0&1&1&0&1&1&1&0&0&0&0&0&1&0 \\ 0&0&1&1&1&0&1&0&1&0&0&1&0&0&0 \\ 1&0&1&1&1&1&1&1&1&1&1&1&0&1&0 \end{array}\right]$$

Problem 8. Determine the number of codewords in C_3.

Our instructions to the receiver are: upon receipt of r, first compute

$$H_3 r^t = \begin{bmatrix} s_1 \\ s_2 \end{bmatrix}.$$

Then:

(i) If $s_1 = s_2 = 0$, assume no errors occurred.

(ii) If $s_1 \neq 0$ and $s_2 = (s_1)^3$, assume a single error occurred in the ith position where $s_1 = c_i$.

(iii) If $s_1 \neq 0$ and $s_2 \neq (s_1)^3$, examine $\{c_k : 1 \leq k \leq 15\}$ for solutions to the quadratic equation

(9) $$z^2 + s_1 z + \left((s_1)^2 + s_2/s_1\right) = 0.$$

If two solutions, c_i and c_j, are found, assume errors occurred in the ith and jth positions. Otherwise, assume three errors occurred (and request retransmission).

(iv) If $s_1 = 0$ and $s_2 \neq 0$, assume three errors occurred (and request retransmission).

Example 2: Suppose r = 101110000110001. Computing $H_3 r^t$ yields

$$s_1 = c_1 + c_3 + c_4 + c_5 + c_{10} + c_{11} + c_{15} = \begin{bmatrix} 1 \\ 1 \\ 0 \\ 1 \end{bmatrix} = c_{13}$$

and

$$s_2 = (c_1)^3 + (c_3)^3 + (c_4)^3 + (c_5)^3 + (c_{10})^3 + (c_{15})^3 = \begin{bmatrix} 1 \\ 0 \\ 1 \\ 0 \end{bmatrix} = c_{10}.$$

Since $s_1 \neq 0$ and $(s_1)^3 = c_8 \neq s_2$, we are in case (iii) above. Equation (9) becomes

$$z^2 + c_{13} \cdot z + c_{11} = 0,$$

since

$$\begin{aligned} (s_1)^2 + (s_2)/s_1 &= (c_2)^{22} + (c_2)^{10}/(c_2)^{11} \\ &= (c_2)^{22} + (c_2)^{-1} \\ &= (c_2)^7 + (c_2)^{14} \\ &= c_7 + c_{12} \\ &= c_{11}. \end{aligned}$$

Testing for roots, we find that

$$(c_3)^2 + c_{13} \cdot c_3 + c_{11} = 0$$

and

$$(c_{14})^2 + c_{13} \cdot c_{14} + c_{11} = 0,$$

which imply errors occurred in the 3rd and 14th positions.

The message was (probably)

$$m = 100110000110011.$$

Problem 9. Apply the receiver's instructions to

$$r_1 = 1101000101100 10$$

$$r_2 = 0100001110 10000$$

$$r_3 = 1101000110000 10$$

$$r_4 = 1100000101000 11$$

and recover the corresponding message when possible.

Problem 10. Conjecture the form of a parity-check matrix, H, for a code that will detect and correct up to three errors in transmission.

---

6. PROBLEM SOLUTIONS

1. $r = m + e$ implies $m = r + e$

$$\begin{array}{r} m = 1010101010101 01 \\ + \quad r = 1010101110101 00 \\ \hline e = 0000000100000 01 \end{array}$$

2. $c_1 = 000000000000000$

$c_2 = 011011011101011$

$c_3 = 111111111111000$

$c_4 = 111000000000101$

3. $H_1 r^t = \begin{bmatrix} 1 \\ 0 \\ 1 \end{bmatrix}$. The only two columns of H_1 which sum to $\begin{bmatrix} 1 \\ 0 \\ 1 \end{bmatrix}$ are the 13th and 15th. Thus $c = 000000000000101$ and $m = r + e = 111111111110101$.

4. If the ith column is a column of zeros then a single error in the ith column will not be detected; i.e.,

$$Hr^t = Hm^t + He^t = 0 + 0 = 0 \text{ if } e = \underbrace{0 \cdots 0}_{\text{ith position}} 1\, 0 \cdots 0.$$

5. If a single error occurred it was in the 4th position since $H_2 r^t = c_4$. Therefore $m = 111000101001101$. Since $c_{11} + c_{15} = c_3 + c_7 = c_4$ errors in the 11th and 15th positions or in the 3rd and 7th positions give the same syndrome.

6. $c_1 \leftrightarrow 1$ is the multiplicative identity

$c_{12} \cdot c_2 = c_1$ since $c_{12} \leftrightarrow x^3 + x^2$, $c_2 \leftrightarrow x$, and $(x^3 + x^2)x = x^4 + x^3 \equiv 1 \leftrightarrow 1$

$$\begin{array}{r|l} & 1 \\ \hline x^4 + x^3 + 1 & x^4 + x^3 \\ & x^4 + x^3 + 1 \\ \hline & 1 \end{array}$$

$c_{13} \cdot c_{12} = c_{10}$ since $c_{13} \leftrightarrow x^3 + x^2 + 1$, $c_{12} \leftrightarrow x^3 + x^2$, and $(x^3 + x^2 + 1)(x^3 + x^2) = x^6 + x^4 + x^3 + x^2 \equiv x^3 + x \leftrightarrow c_{10}$

$$\begin{array}{r|l} & x^2 + x \\ \hline x^4 + x^3 + 1 & x^6 + x^4 + x^3 + x^2 \\ & x^6 + x^5 + x^2 \\ \hline & x^5 + x^4 + x^3 \\ & x^5 + x^4 + x \\ \hline & x^3 + x \end{array}$$

7. $(c_2)^5 \leftrightarrow x^5 \equiv x^3 + x + 1 \leftrightarrow c_{11}$

$$x^4 + x^3 + 1 \;\overline{)\; x^5}$$

$$x^5 + x^4 \qquad + x$$

$$x^4 \qquad + x$$

$$x^4 + x^3 \qquad + 1$$

$$x^3 + x + 1$$

$c_1 \cdot c_1 = c_4 \cdot c_6 = c_8 \cdot c_3 = c_9 \cdot c_{13} = c_{11} \cdot c_{10} = c_{15} \cdot c_5$
$= c_7 \cdot c_{14} = c_2 \cdot c_{12} = 1.$

8. Since the row rank of H_3 is 8 the dimension of C_3 is $15 - 8 = 7$. Therefore $|C_3| = 2^7$.

9. Compute $Hr_i^t = \begin{bmatrix} s_1 \\ s_2 \end{bmatrix}$.

r_1: $s_1 = 0$, $s_2 = c_5$ implies that at least three errors occurred. Ask for a retransmission.

r_2: $s_1 = c_{15}$, $s_2 = 0$, $(s_1)^3 \neq s_2$ implies that errors occurred in the 2nd and 13th positions since the roots of $z^2 + c_{15} \cdot z + (c_{15})^2 = z^2 + c_{15} \cdot z + c_3 = 0$ are c_2 and c_{13}.

$m_2 = 000000111010100$.

r_3: $s_1 = c_8$, $s_2 = c_5$, $(s_1)^3 = s_2$ implies that an error occurred in the 8th position. $m_3 = 110100001000010$.

r_4: $s_1 = 0$, $s_2 = 0$ implies that $m_4 = r_4$.

10. This is a difficult problem, and a complete solution requires the development of the finite field theory behind our choice of $p(x) = x^4 + x^3 + 1$. Such a development is beyond the scope of this module (see The Theory of Error Correcting Codes I and II, by MacWilliams and Sloane). Still, an educated guess is possible. Indeed, our discussion at the beginning of Section 5 suggests that we might add four rows to H_3 to correct the third error. The

matrix we obtain, say H, would be a 12×15 matrix and the syndrome of a received word could be written

$$\begin{bmatrix} s_1 \\ \hline s_2 \\ \hline s_3 \end{bmatrix},$$

where each s_i is a binary four-component column vector. Thus,

$$\left[\begin{array}{ccccc} c_1 & c_2 & \cdots & c_{15} \\ \hline c_1^3 & c_2^3 & \cdots & c_{15}^3 \\ \hline g(c_1) & g(c_2) & \cdots & g(c_{15}) \end{array}\right],$$

and

$$\begin{aligned} c_i + c_j + c_k &= s_1 \\ (*) \qquad c_i^3 + c_j^3 + c_k^3 &= s_2 \\ g(c_i) + g(c_j) + g(c_k) &= s_3, \end{aligned}$$

where errors occur in the ith, jth and kth positions, and the function g is to be determined such that (*) can be solved uniquely for c_i, c_j and c_k. How should we define g? Since $g(c_i) = c_i$, $g(c_i) = c_i^2$ and $g(c_i) = c_i^3$ won't do, let's try $g(c_i) = c_i^4$. Then

$$\begin{aligned} s_3 &= c_i^4 + c_j^4 + c_k^4 \\ &= (c_i^2)^2 + (c_j^2)^2 + (c_k^2)^2 \\ &= (c_i^2 + c_j^2)^2 + (c_k^2)^2 \\ &= (c_i^2 + c_j^2 + c_k^2)^2 \\ &= [(c_i + c_j)^2 + c_k^2]^2 \end{aligned}$$

$$= [(c_i + c_j^2 + c_k)^2]^2$$

$$= (c_i + c_j + c_k)^4$$

$$= s_1^4.$$

That is, the first and third equations in (*) are redundant. Our next guess is $g(c_i) = c_i^5$. Hard as you may try, you won't be able to find any redundancy in the system.

$$c_i + c_j + c_k = s_1$$

$$c_i^3 + c_j^3 + c_k^3 = s_2$$

$$c_i^5 + c_j^5 + c_k^5 = s_3.$$

Conjecture:

$$\begin{bmatrix} c_1 & c_2 & \cdots & c_{15} \\ c_1^3 & c_2^3 & \cdots & c_{15}^3 \\ c_1^5 & c_2^5 & \cdots & c_{15}^5 \end{bmatrix}.$$

7. MODEL EXAM

1. The binary matrix

$$H = \begin{bmatrix} 1 & 1 & 1 & 1 & 0 \\ 0 & 1 & 1 & 1 & 1 \\ 1 & 0 & 1 & 0 & 1 \end{bmatrix}$$

determines a code C contained in Z_2^5.

   a. Compute the syndrome of r = 10111.

   b. Is r a codeword?

   c. How many codewords are there in C?

   d. List all of the codewords in C.

   e. If r is received, what is the most likely message?

   f. The code C can detect all single errors. Can it correct all single errors? Why?

2. Solve the equation $x^6 + x^4 + x^2 + 1 \equiv p(x)$ (modulo $x^4 + x^3 + 1$), where p(x) is a binary polynomial of degree at most three.

3. This problem refers to the code C_3 of Section 5. Determine, if possible, the message if r = 100100110010110 is received.

4. Is it possible to apply the results of the module to a binary symmetric channel for which $p > \frac{1}{2}$?

8. MODEL EXAM SOLUTIONS

1. a. $\begin{bmatrix}1\\1\\1\end{bmatrix}$.

b. No.

c. $2^{5-3} = 2^2 = 4$.

d. $\begin{bmatrix}1&1&1&1&0\\0&1&1&1&1\\1&0&1&0&1\end{bmatrix}\begin{bmatrix}m_1\\m_2\\m_3\\m_4\\m_5\end{bmatrix} = \begin{bmatrix}0\\0\\0\end{bmatrix}$ implies $m_1 = m_5$, $m_2 = m_4 + m_5$, and $m_3 = 0$.

Therefore,

$C = \{00000, 01010, 11001, 10011\}$.

e. 10011.

f. No. If 00010 is received, then 00000 and 01010 are equally likely to have been sent.

2. $p(x) = x + 1$, since

$$
\begin{array}{r|l}
x^4 + x^3 + 1 & x^2 + x \\
\hline
x^6 + x^4 + x^2 + 1 & \\
x^6 + x^5 + x^2 & \\
\hline
x^5 + x^4 + 1 & \\
x^5 + x^4 + x & \\
\hline
x + 1. &
\end{array}
$$

3. $H_3 r^t = \begin{bmatrix}0\\0\\1\\0\\\hline 1\\0\\0\\0\end{bmatrix} \neq 0$ implies that at least one error occurred in transmission.

Since $s_1 = \begin{bmatrix}0\\0\\1\\0\end{bmatrix} = c_2$, and $s_2 = \begin{bmatrix}1\\0\\0\\0\end{bmatrix} = c_8 = c_2^3$,

we conclude that a single error occurred in the second position. That is, the message was 1101001100101l0.

4. Yes. If a 0 is received, rewrite it as a 1. If a 1 is received, rewrite it as a 0.

STUDENT FORM 1

Request for Help

Return to:
EDC/UMAP
55 Chapel St.
Newton, MA 02160

Student: If you have trouble with a specific part of this unit, please fill out this form and take it to your instructor for assistance. The information you give will help the author to revise the unit.

Your Name _____ Unit No. _____

Page _____
- ○ Upper
- ○ Middle
- ○ Lower

OR

Section _____
Paragraph _____

OR

Model Exam
Problem No. _____
Text
Problem No. _____

Description of Difficulty: (Please be specific)

Instructor: Please indicate your resolution of the difficulty in this box.

○ Corrected errors in materials. List corrections here:

○ Gave student better explanation, example, or procedure than in unit. Give brief outline of your addition here:

○ Assisted student in acquiring general learning and problem-solving skills (not using examples from this unit.)

Instructor's Signature _____

Please use reverse if necessary.

STUDENT FORM 2

Unit Questionnaire

Return to:
EDC/UMAP
55 Chapel St.
Newton, MA 02160

Name_____ Unit No._____ Date_____

Institution_____ Course No._____

Check the choice for each question that comes closest to your personal opinion.

1. How useful was the amount of detail in the unit?

   ____Not enough detail to understand the unit
   ____Unit would have been clearer with more detail
   ____Appropriate amount of detail
   ____Unit was occasionally too detailed, but this was not distracting
   ____Too much detail; I was often distracted

2. How helpful were the problem answers?

   ____Sample solutions were too brief; I could not do the intermediate steps
   ____Sufficient information was given to solve the problems
   ____Sample solutions were too detailed; I didn't need them

3. Except for fulfilling the prerequisites, how much did you use other sources (for example, instructor, friends, or other books) in order to understand the unit?

   ____A Lot ____Somewhat ____A Little ____Not at all

4. How long was this unit in comparison to the amount of time you generally spend on a lesson (lecture and homework assignment) in a typical math or science course?

Much Longer	Somewhat Longer	About the Same	Somewhat Shorter	Much Shorter
____	____	____	____	____

5. Were any of the following parts of the unit confusing or distracting? (Check as many as apply.)

   ____Prerequisites
   ____Statement of skills and concepts (objectives)
   ____Paragraph headings
   ____Examples
   ____Special Assistance Supplement (if present)
   ____Other, please explain_____

6. Were any of the following parts of the unit particularly helpful? (Check as many as apply.)

   ____Prerequisites
   ____Statement of skills and concepts (objectives)
   ____Examples
   ____Problems
   ____Paragraph headings
   ____Table of Contents
   ____Special Assistance Supplement (if present)
   ____Other, please explain_____

Please describe anything in the unit that you did not particularly like.

Please describe anything that you found particularly helpful. (Please use the back of this sheet if you need more space.)