1975-12

Application of the computer for real time encoding and decoding of cyclic block codes

Cetinilmaz, Nizamettin

https://hdl.handle.net/10945/20754

APPLICATION OF THE COMPUTER FOR REAL TIME ENCODING AND DECODING OF CYCLIC BLOCK CODES

Nizamettin Cetinyilmaz

NAVAL POSTGRADUATE SCHOOL
Monterey, California

THESIS

APPLICATION OF THE COMPUTER FOR REAL TIME
ENCODING AND DECODING OF CYCLIC BLOCK CODES

by

Nizamettin Cetinyilmaz

December 1975

Thesis Advisor: G. Marmont

Approved for public release; distribution unlimited.

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER	2. GOVT ACCESSION NO.	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) Application of the Computer for Real Time Encoding and Decoding of Cyclic Block Codes		5. TYPE OF REPORT & PERIOD COVERED Master's Thesis; December 1975
		6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s) Nizamettin Cetinyilmaz		8. CONTRACT OR GRANT NUMBER(s)
9. PERFORMING ORGANIZATION NAME AND ADDRESS Naval Postgraduate School Monterey, California 93940		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS
11. CONTROLLING OFFICE NAME AND ADDRESS Naval Postgraduate School Monterey, California 93940		12. REPORT DATE December 1975
		13. NUMBER OF PAGES 85
14. MONITORING AGENCY NAME & ADDRESS(if different from Controlling Office) Naval Postgraduate School Monterey, California 93940		15. SECURITY CLASS. (of this report) Unclassified
		15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

** Approved for public release; distribution unlimited.

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

This thesis is concerned with cyclic block codes which can be used for the detection and correction of errors in a transmitted message which are produced by various types of noise. Computer programs were developed and used for the actual encoding and decoding process. Advantages of using the computer as against using various types of dedicated hardware is demonstrated. Two different methods of decoding are presented: the minimum distance decoder

and the syndrome method decoder. Pseudo random noise sequences were also generated by computer program and used to simulate noise disturbance of the encoded transmitted message. Codes of several rates and with varying degrees of simulate channel noise were studied and compared with respect to the probability of error. It is shown how the methods developed in this thesis can materially help in choosing the 'best' code for a given noisy channel, consonant with other specified parameters for message transmission.

Application of the Computer for Real Time Encoding and Decoding of Cyclic Block Codes

by

Nizamettin Cetinyilmaz
Lieutenant, Turkish Navy
B.S., Naval Postgraduate School, 1974

Submitted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE IN ELECTRICAL ENGINEERING

from the

NAVAL POSTGRADUATE SCHOOL
December 1975

ABSTRACT

This thesis is concerned with cyclic block codes which can be used for the detection and correction of errors in a transmitted message which are produced by various types of noise. Computer programs were developed and used for the actual encoding and decoding process. Advantages of using the computer as against using various types of dedicated hardware is demonstrated. Two different methods of decoding are presented: the minimum distance decoder and the syndrome method decoder. Pseudo random noise sequences were also generated by computer program and used to simulate noise disturbance of the encoded transmitted message. Codes of several rates and with varying degrees of simulated channel noise were studied and compared with respect to the probability of error. It is shown how the methods developed in this thesis can materially help in choosing the 'best' code for a given noisy channel, consonant with other specified parameters for message transmission.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

I. INTRODUCTION

After the appearance of Shannon's classic papers in 1948 and 1949, a great deal of research has been devoted to the problem of designing efficient schemes by which information can be coded for reliable transmission across channels which are corrupted by noise. The channel is described statistically by giving a probability distribution over the set of all possible outputs for each permissible input.

In Shannon's model, a randomly generated message produced by a source of information is 'encoded,' that is each possible message that the source can produce is associated with a signal belonging to a specific set. It is the encoded message which is actually transmitted. When the transmitted encoded message is received, a 'decoding' operation is performed, that is, a decision is made as to the identity of the particular signal transmitted. The main objective is to increase the elements of any set to be transmitted, and at the same time decrease the probability of error at the output of the decoder. How well one can do these things depends essentially on the properties of the channel.

The establishment of digital technology provided a powerful way of utilization in satellite communication, data transfer between computers and in military applications.

Encoding and decoding operations were done by a minicomputer (DEC PDP - 11/40), channel noise was simulated by

a computer program. The results were obtained from the computer program close to actual world binomial distribution. The codes investigated were members of a type known as cyclic codes.

II. BACKGROUND

In a communication channel, noise and disturbances modifying the signal create errors, a simple way to reduce uncertainty at the receiver due to errors is to simply transmit the message two or more times, a much more efficient way of providing means for detection and correction of errors involves the use of error correcting codes (controlled redundancy).

Controlled redundancy or error correction coding is commonly divided into two main groups: (1) block codes (2) convolutional codes. Convolutional codes are decoded by a statistical procedure due to it's continuous (bit by bit) nature. On the other hand to decode the block codes, a whole word (block) has to be received.

A block diagram of a digital communication system is shown in Figure 1. The information source provides a message or sequence of messages to be communicated to the receiving terminal. Message may be of various types (1) sequence of letters as in a telegraph or teletype system, (2) an analog time function as in radio or telephone, (3) a function of time and two space coordinates as in black and white television, (4) several functions of several variables as in color television, etc. Since the purpose of the source encoder is to present the information source output by a sequence of binary digits, one of the major questions of concern is to determine how many binary digits per unit time are required

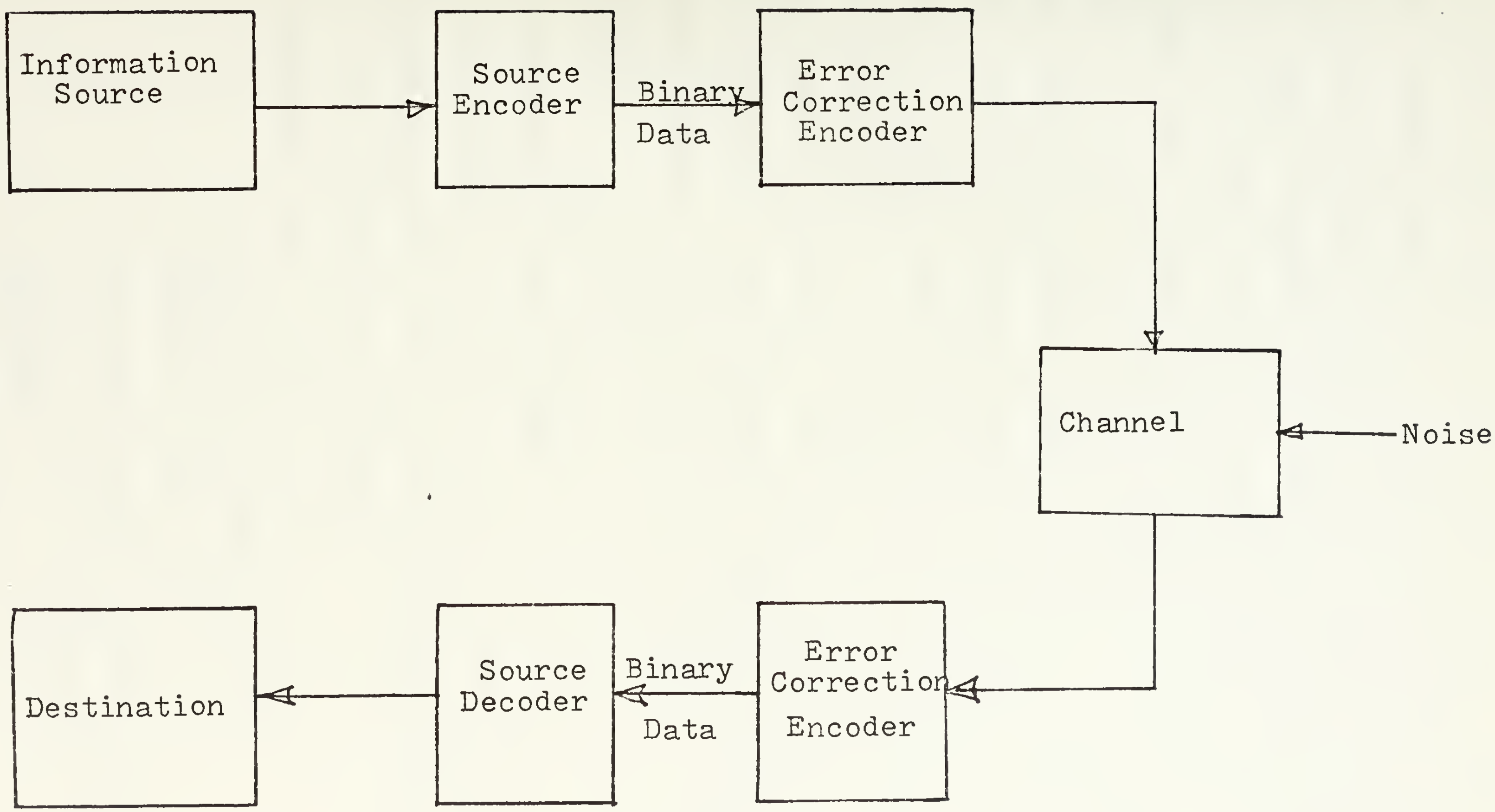


Figure 1. Digital Communication System

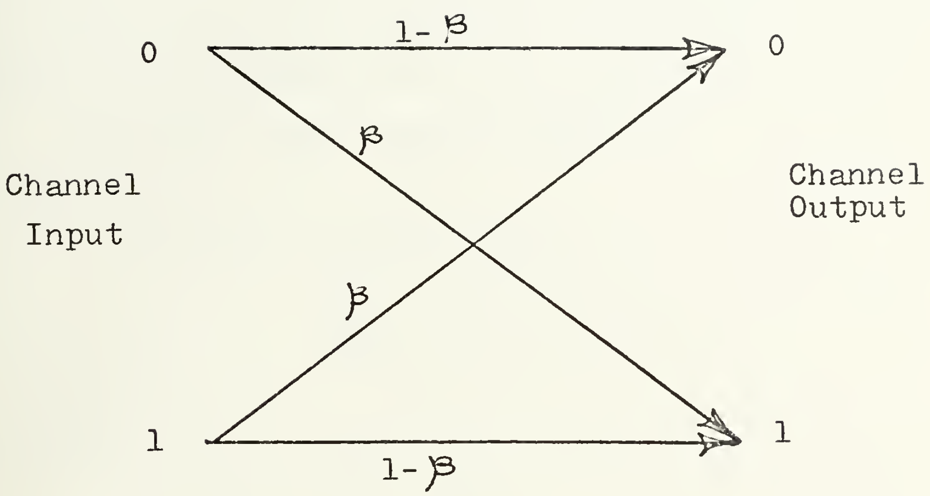
to represent the output of any given source. The error correction encoder used in this thesis is a cyclic encoder which is a type of block encoding system. Channel is merely a medium used to transmit the signal from transmitter to receiver. It may be a pair of wires, a coaxial cable, free space, a beam of light, etc. In any kind of channel the signal may be perturbed by noise. The channel modeled in this thesis is a binary symmetric channel, which is shown in Figure 2. The error correction decoder performs the inverse operation of that done by the channel encoder, and in addition corrects the errors altering the message to the extend of that the errors can be corrected. The source decoder does the inverse operation of the source encoder, changing the data to the original signal. Destination is the person or thing for whom the message is intended.

A. PRINCIPALS OF BLOCK CODES

As pointed out earlier, coding and decoding systems are implemented by with the aid of minicomputer (DEC PDP - 11/40). Only binary codes were considered.

Notations used in this thesis:

- k = Number of information bits
- m = Number of check bits
- n = Total word length in bits ($n=m+k$)
- e = Maximum number of errors can be corrected in one word
- R = Data rate ($R=k/m$)
- β = Binary symmetric channel (BSC) parameter



$$P(0/1) = P(1/0) = \beta$$

Figure 2. Binary Symmetric Channel

X = Source encoder output

W = Error correction encoder output (code word)

[T] = Characteristic matrix of the code

G(X) = Generator polynomial

H(X) = Check polynomial

d = Hamming distance between code words

Z = Noise (as a word)

In order to correct e-tuple or less errors in one word, there are two inequalities to be satisfied.

a. Hammings lower bound inequality

$$2^k \leqslant \frac{2^n}{\sum_{i=0}^{e} \binom{n}{i}}$$

or equivalently

$$2^m \geqslant \sum_{i=0}^{e} \binom{n}{i}$$

Hammings lower bound inequality is necessary but not sufficient for constructibility on an e-tuple error correction code.

b. Varsharmov - Gilbert - Sacks condition (upper bound)

$$2^m > \sum_{i=0}^{2e-1} \binom{n-1}{i}$$

This condition is sufficient but not necessary.

These two bound's box in the number of check digits, m , required for a block code, where each word consists of n digits.

The code rate is the ratio of the message digits per word k , divided by n , since k equals n-m, it is obvious that increasing the number of check digits decreases the data rate, on the other hand increasing the number of check digits decreases the number of uncorrectible errors, therefore for a given signal to noise power ratio to desire to keep the data rate high, it is in conflict with the desire to minimize errors. One is than faced to with the task of making an engineering compromise.

B. DESCRIPTION OF CYCLIC CODES

In this thesis a special class of block codes known as cyclic codes are described. This kind of codes have two special advantages over ordinary block codes:

(1) Encoding operation is easy to instrument

(2) A large amount of mathematical structure in the code makes it possible to find various simple decoding algorithms.

Let $X=(x_1,x_2,x_3, \ldots\ldots\ldots,x_k)$ be an arbitrary sequence of information digits with each $x_i, 1 \leqslant i \leqslant k$ an element of a Galois field (GF 2) (which is 0 or 1). An (n,k) code is a code in which the code words $W=(w_1,w_2,w_3,\ldots\ldots\ldots,w_n)$ corresponding to each W is a sequence of $n > k$ letters, generated by the rule

$$W_n = \sum_{i=1}^{k} x_i \, b_{i,n}$$

Where the elements $b_{i,n}$ are arbitrary chosen elements of GF(2) the additions and multiplication are operations in GF(2)

A set containing at least two members that is closed under two operations (called 'addition' and 'multiplication') is called a field. Roughly speaking, a field is a set in which we can do addition, subtraction, multiplication, and division without leaving the set. The field I of matric polynomials p(I) of degree $\leqslant q - 1$ has 2^q elements; I is called the Galois field of order 2^q, written as GF (2^q). The type of block codes can be represented

$$b_{n-1}w^{n-1} + b_{n-2}w^{n-2} + b_{n-3}w^{n-3} + \ldots\ldots + b_1 w + b_o$$

if any right or left cyclic shifts of this word is another word, and any linear combinations of such code words is another code word, the code is called a cyclic code (name of the cyclic code comes from the cyclic shifts of words to get another code word).

In the binary case multiplication of any code word by positive powers of two (left cyclic shift) is another code word, conversely multiplication by negative powers of two (right cyclic shift) is also another code word (in modulo 2).

Generation of any code word can be realized by a k-stage feedback shift register or m-stage feedback shift register.

C. CYCLIC ENCODING

1. <u>k-stage feedback shift register</u> (Figure 3) :

This type of encoder has binary storage cells $F_0, F_1, F_2, \ldots, F_{k-1}$ switches $g_0, g_1, g_2, \ldots, g_{k-1}$ if $g_i = 1$ the corresponding switch is closed, if $g_i = 0$ the switch is open, the device also includes a modulo 2 adder. The system is controlled by a clock pulse. At $t = 0$ the binary message to be encoded is put into cells of the register. At each clock pulse the contents of F_i are shifted to F_{i-1} and the new number in F_{k-1} is

$$g_o x_o + g_1 x_1 + g_2 x_2 + \cdots\cdots + g_{k-1} x_{,-1}$$

where $x_0, x_1, x_2, \ldots, x_{k-1}$ is the message word to be encoded and x_i is the contents of register cell F_i.

The operation of a feedback shift register can be described by a matrix equation. If x_i is the number stored in F_i before the clock pulse and if x'_i is the number stored in the same register after the clock pulse, the contents of the register cells after the first clock pulse becomes;

$$X'_0 = X_1$$

$$X'_1 = X_2$$

$$\vdots$$

$$X'_{k-2} = X_{k-1}$$

$$X'_{k-1} = g_0 X_0 + g_1 X_1 + \cdots\cdots + g_{k-1} X_{k-1}$$

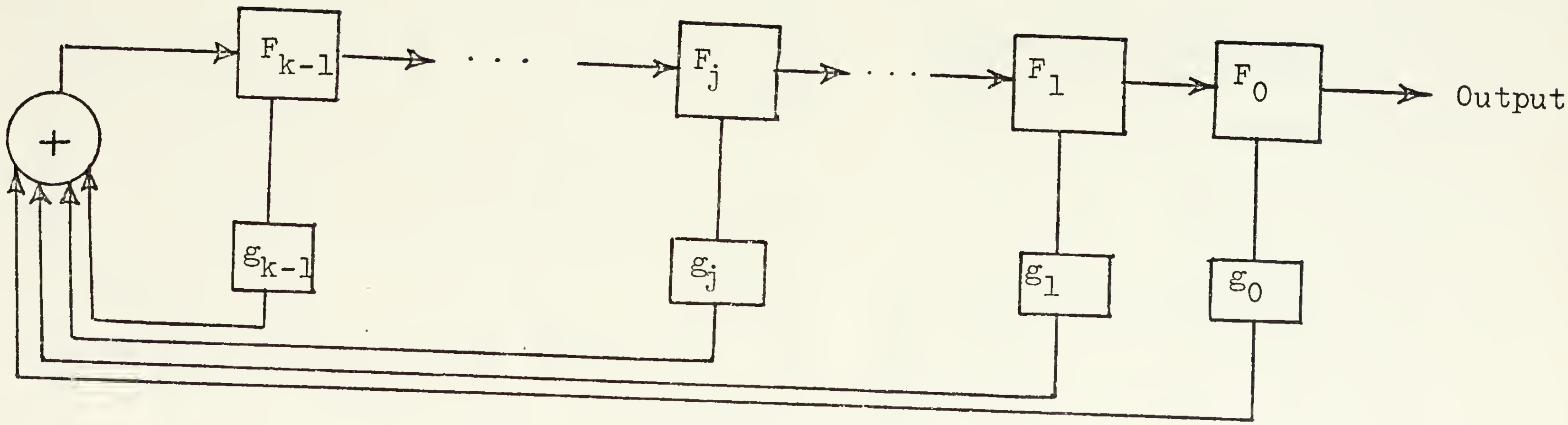


Figure 3. k-Stage Feedback Shift Register.

In matrix form;

$$[X'] = [T][X]$$

where

$$X = \begin{bmatrix} X_0 \\ X_1 \\ \cdot \\ \cdot \\ X_{k-1} \end{bmatrix} \qquad X = \begin{bmatrix} X_0' \\ X_1 \\ \cdot \\ \cdot \\ X_{k-1} \end{bmatrix} \qquad T = \begin{bmatrix} 0 & 1 & 0 & 0 & . & . & . & . & 0 \\ 0 & 0 & 1 & 0 & . & . & . & . & 0 \\ 0 & 0 & 0 & 1 & . & . & . & . & 0 \\ . & . & . & . & . & . & . & . & . \\ g_0 & g_1 & g_2 & & . & . & . & . & g_{k-1} \end{bmatrix}$$

input word characteristic matrix

Any initial message put into the shift register cells F_0, F_1, . . . , F_{k-1} (unless all zeros) will repeat itself after n-1 clock pulses. At this point ($t = 0$ or $t = n$) a new message to be encoded is put into shift register cells, at each clock pulse the contents of shift register cell F_0 will be taken as a encoded word bit, as one can see, the first k bits out of the shift register will be the actual message bits (information bits). At $t = n-1$ clock pulse the last encoded bit will be out of the shift register cell F_0, the last n-k bits out of the shift register cell F_0 will be the check digits. After $(n-1)^{th}$ clock pulse the contents of shift register cells repeat, n defines the code length and it is called the period of the shift register.

The characteristic polynomial of the $[T]$ matrix (characteristic matrix) is defined by;

$$\emptyset(x) = \left|T - I\,x\right| = x^k + g_{k-1}x^{k-1} + g_{k-2}x^{k-2} + \ldots + g_1 x + g_0$$

The generator polynomial of the code G(x), in general $\emptyset(x) \neq G(x)$, which is characteristic polynomial $\emptyset(x)$ is higher in degree than generator polynomial G(x). However if characteristic matrix $[T]$ is in the form of given above, characteristic polynomial $\emptyset(x)$ is equal to generator polynomial G(x). Given a characteristic matrix $[T]$, G(x) can be used to define the code uniquely. Example 1: If characteristic polynomial is chosen as

$$G(X) = X^4+X+1$$

which is $G(X)=X^4+g_3X^3+g_2X^2+g_1X+1$

$g_3 = g_2 = 0$ switches are open, k-stage feedback shift register becomes as shown in Figure 4.

At t=0 if the message 0001 put in shift registers the contents of shift registers will follow the period shown in Figure 5. Since the period of the characteristic matrix or the characteristic polynomial is fifteen, $G(X) = X^4+X+1$ generates the code with four information digits (power of characteristic polynomial is four) and eleven check digits. The given characteristic polynomial has the period $2^4-1 = 15$, therefore it is called the maximum period polynomial or irreducible polynomial. Since the period of generator polynomial is fifteen it divides the polynomial $X^{15}+1$ (in modulo 2).

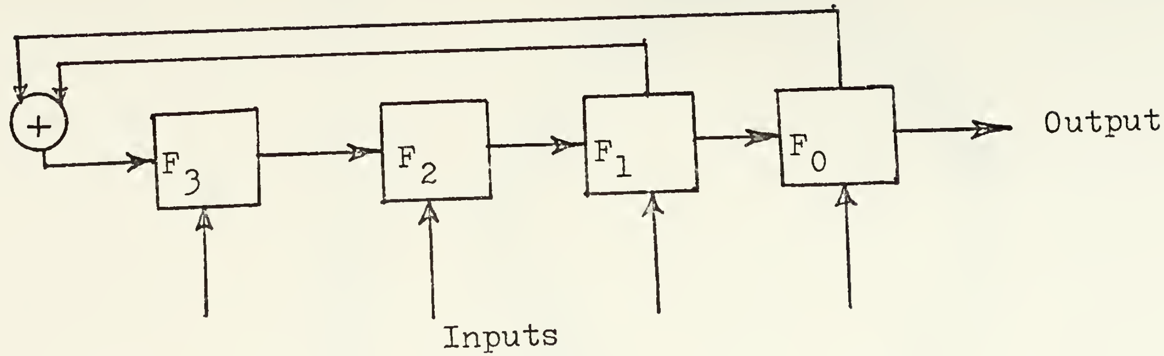
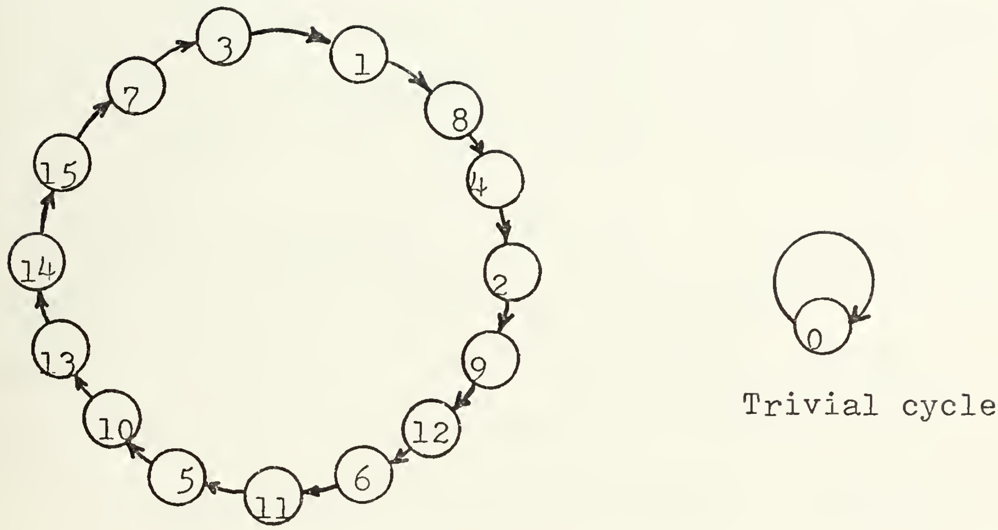


Figure 4. k-Stage Encoder of the Characteristic Polynomial
$G(x) = x^4+x+1$



Numbers in the circles are the decimal representation of what is in the shift registers at $t = t_i$.

Figure 5. Cycle set of generator Polynomial $G(x) = x^4+x+1$.

The check polynomial of the same code is defined by:

$$H(X) = \frac{X^{15}+1}{G(X)} = X^{11}+X^8+X^7+X^5+X^3+X^2+X+1$$

The coefficients form a code word, namely 000100110101111. The code generated by the given polynomial is a (15,4) code. The code is cyclic, therefore any cyclic shift of the check polynomial is another code word and any linear combinations of code words is also another code word. Since the generator polynomial G(x) is irreducible, fifteen cyclic shifts of the check polynomial is a code word and the code alphabet has $2^4 = 16$ words (including the zero word). To represent the code alphabet 15 cyclic shifts of the coefficients of the check polynomial defines the all non zero alphabet letters.

Example 2: Let the generator polynomial be chosen as

$$G(x) = (x^4+x+1)\ (x^4+x^3+x^2+x+1) = x^8+x^7+x^6+x^4+1$$

The k-stage shift register becomes as shown in Figure 6. If at $t = 0$ the message 00000001 is put in shift register cells, the contents of the register will follow the period shown in Figure 7. The generator polynomial has the degree eight, therefore number of code words are $2^8 = 256$ (including the zero word), and the period of the polynomial is 15. This polynomial (being a reducible polynomial) has (255/15) = 17 non trivial cycle sets and one trivial zero cycle (the zero word). Since the period of the polynomial is 15 (the word length), G(x) divides $x^{15}+1$ (in modulo 2) the check polynomial of the code is

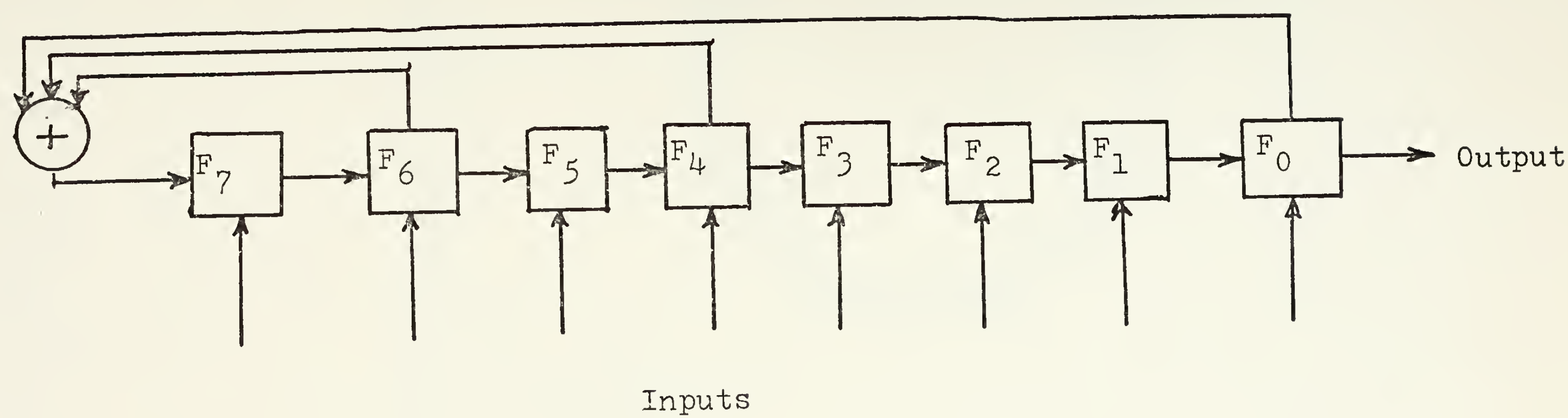
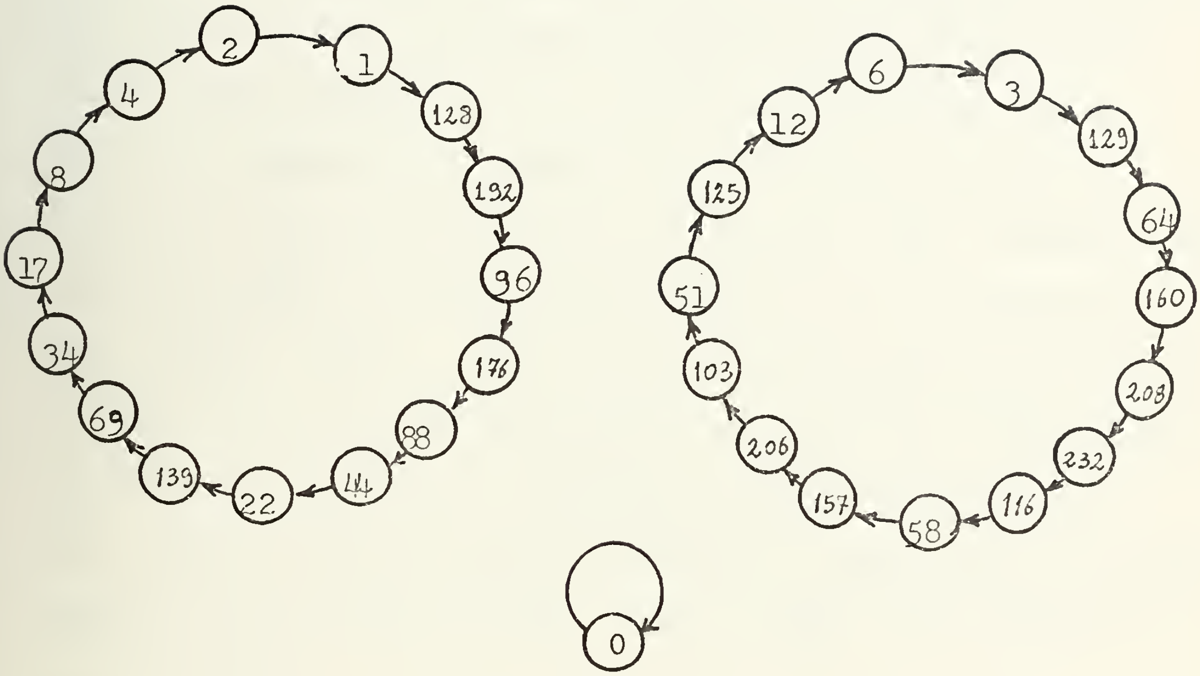


Figure 6. k-Stage Shift Register For Generator Polynomial
$G(x) = x^8+x^7+x^6+x^4+1$



Trivial Cycle

Numbers in the circles are the decimal representation of what is in the shift registers of $t = t_i$

Figure 7. The two cycle sets out of 17 of the generator polynomial $G(x) = x^8+x^7+x^6+x^4+1$

$$H(x) = \frac{x^{15}+1}{G(x)} = x^7+x^6+x^4+1$$

The coefficients form a code word, namely 000000011010001. Any cyclic shifts of this code word is another code word, but by simply shifting it one can get only fifteen different code words. The code alphabet has $2^{15}-1 = 255$ (excluding the zero word) words, the other code words can be obtained by linear combinations of the 15 cyclic shifted code words. Since the generator polynomial given in this example is reducible, there is more than one maximum cycle (actually all the cycles have the same cycle length of 15, excluding the zero trivial cycle). The code generated by this polynomial is a (15,8) code. In the case where the number of check digits less than the number of information digits, choosing the shift register based upon the number of check digits (m-stage feedback shift register[Ref. 5 page 225]) will simplify the encoder design.

2. Computer application of encoder:

Since the whole encoding operations were done with the aid of a computer (DEC PDP - 11/40), this section describes how easy it is to implement encoding operations with a computer. Encoder program used in this thesis just incorporates a matrix multiplication of the message word by the generator matrix described below.

The coefficients of the check polynomial H(x) is a code word and any cyclic shifts of this coefficients is another code

word. One can define the generator matrix as one whose rows are code words. Such a generator matrix is in the form of:

$$G_{k,n} = \begin{bmatrix} 1\ 0\ 0\ 0 \ldots & h_{0,k-1} & h_{0,k-2} & \cdots & h_{0,0} \\ 0\ 1\ 0\ 0 & h_{1,k-1} & h_{1,k-2} & \cdots & h_{1,0} \\ 0\ 0\ 1\ 0 \ldots & h_{2,k-1} & h_{2,k-2} & \cdots & h_{2,0} \\ \cdot & \cdot & \cdot & & \cdot \\ \cdot & \cdot & \cdot & & \cdot \\ 0\ 0\ 0\ 0 \ldots & 1 & h_{k-1} & \cdots & h_0 \end{bmatrix}$$

Any source encoder output message when multiplied by this matrix, gives the encoded word as a result. If the encoded word is defined by $[W]_{1,n}$ and the input message to the error correction encoder is defined by a $[X]_{1,k}$ matrix

$$[W]_{1,n} = [X]_{1,k} [G]_{k,n}$$

This is the easy and fast way to encode the messages. By changing the rows of the generator matrix $[G]_{k,n}$, as for a different code, the encoder will be changed to one for the different cyclic code. To change the rows of the generator matrix $[G]_{k,n}$ one has to define the coefficients of the check polynomial $H(x)$ for the new code, this is easy to do for any given code. Example 3: Generator matrix of the code (15,4)

$$[G]_{k,n} = \begin{bmatrix} 1&0&0&0&1&0&0&1&1&0&1&0&1&1&1 \\ 0&1&0&0&1&1&0&1&0&1&1&1&1&0&0 \\ 0&0&1&0&0&1&1&0&1&0&1&1&1&1&0 \\ 0&0&0&1&0&0&1&1&0&1&0&1&1&1&1 \end{bmatrix}$$

Let $[X]_{1,k} = [0\ 0\ 1\ 1]$

After multiplication by the generator matrix, the encoded word becomes

$$[W]_{1,n} = [0\ 0\ 1\ 1\ 0\ 1\ 0\ 1\ 1\ 1\ 1\ 0\ 0\ 0\ 1]$$

this is the modulo 2 addition of last 2 rows of the generator matrix $[G]_{k,n}$.

The encoder program described by the flow chart shown in Figure 8 performs the matrix multiplication of the message word by the generator matrix. The operations used in the flow chart involved the following notations:

MOV	=	move
CLR	=	clear
ASL	=	arithmetically shift left
BCC	=	branch if carry is clear
XOR	=	exclusive or
SOB	=	substruct one and branch if the result is not equal to zero
RO	=	register # 0
R1	=	register #1 etc.
(RO)	=	contents of register # 0
(w)	=	contents of address w
HLT	=	halt
(RO)+	=	increment contents of register # 0 (RO) by two

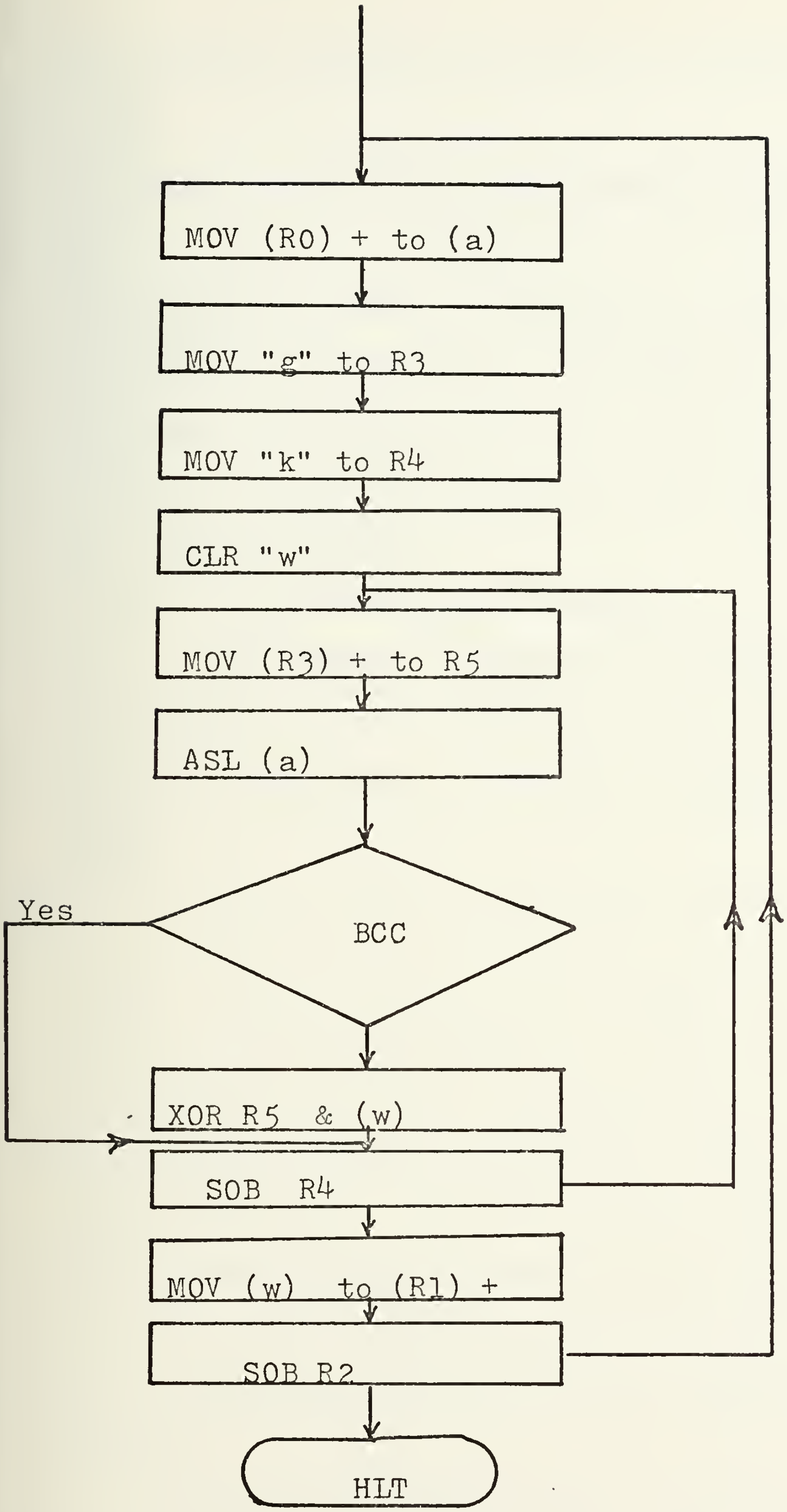


Figure 8. Cyclic Encoder Flow Chart.

The message digits are stored in a block in the form of ASCII code.

The starting address of information message to be encodes is in R0.

The starting address of encoded word is in R1.

The number of information characters to be encoded is in R2.

The starting address of rows of generator matrix is in address(g).

k is the number of information message bits.

D. CYCLIC DECODER

1. Minimum Distance Decoder

The Hamming distance is defined by the minimum number of different digits between 2 code words.

Example: The Hamming distance between the following 2 code words

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

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

is 8. If any combinations of $(\frac{d-1}{2})$ or less erros occur in a received code word, the distance of this perturbed code word to the original transmitted word is less than the other original alphabet letters. For the code above if three or less errors occur in one word the distance of this noisy word to the actual transmitted word is three or less but the distance to the other code words is five or greater.

Example:

original transmitted word	: 0 0 1 0 0 1 1 0 1 0 1 1 1 1 0
error sequence	: 0 0 1 0 0 0 1 0 0 0 0 0 1 0 0
received word	: 0 0 0 0 0 1 0 0 1 0 1 1 0 1 0

The distance (d_i) between this received word and some of the other words is as follows:

Code alphabet:	000100110101111	001001101011110	000000000000000
received word:			
000001001011010	$d_1=9$	$d_2=3$	$d_3=5$

Using the fixed properties of irreducible polynomial codes, if the received word is not in the alphabet set, the decoder takes the code word which has a distance to this received word which is equal to or less than d/2, as a decoded word.

For the code given by the coefficients of the

$H(x) = x^{11}+x^8+x^7+x^5+x^3+x^2+x+1$, the Hamming distance is 8. One expects any combinations of three or less errors will be decoded correctly. Due to Varsharmov - Gilbert-Sacks condition (upper bound) for the (15,4) code $e = 4$ does not satisfy the inequality. In experiments it is found that out of 1365 different combinations of 4 errors 926 four errors can be corrected. Minimum distance decoder for any irreducible polynomial can be constructed as shown in Figure 9.

2. Operation of Minimum Distance Decoder

Starting from time $t = 0$ the received word is fed bit by bit to the shift register A. Register C has the coefficients of the check polynomial H(x) in binary representation, register D contains all zeros. At time $t = n-1$ register A will have the whole word $[w+z]_{1,n}$. Where $[w]_{1,n}$ is the original

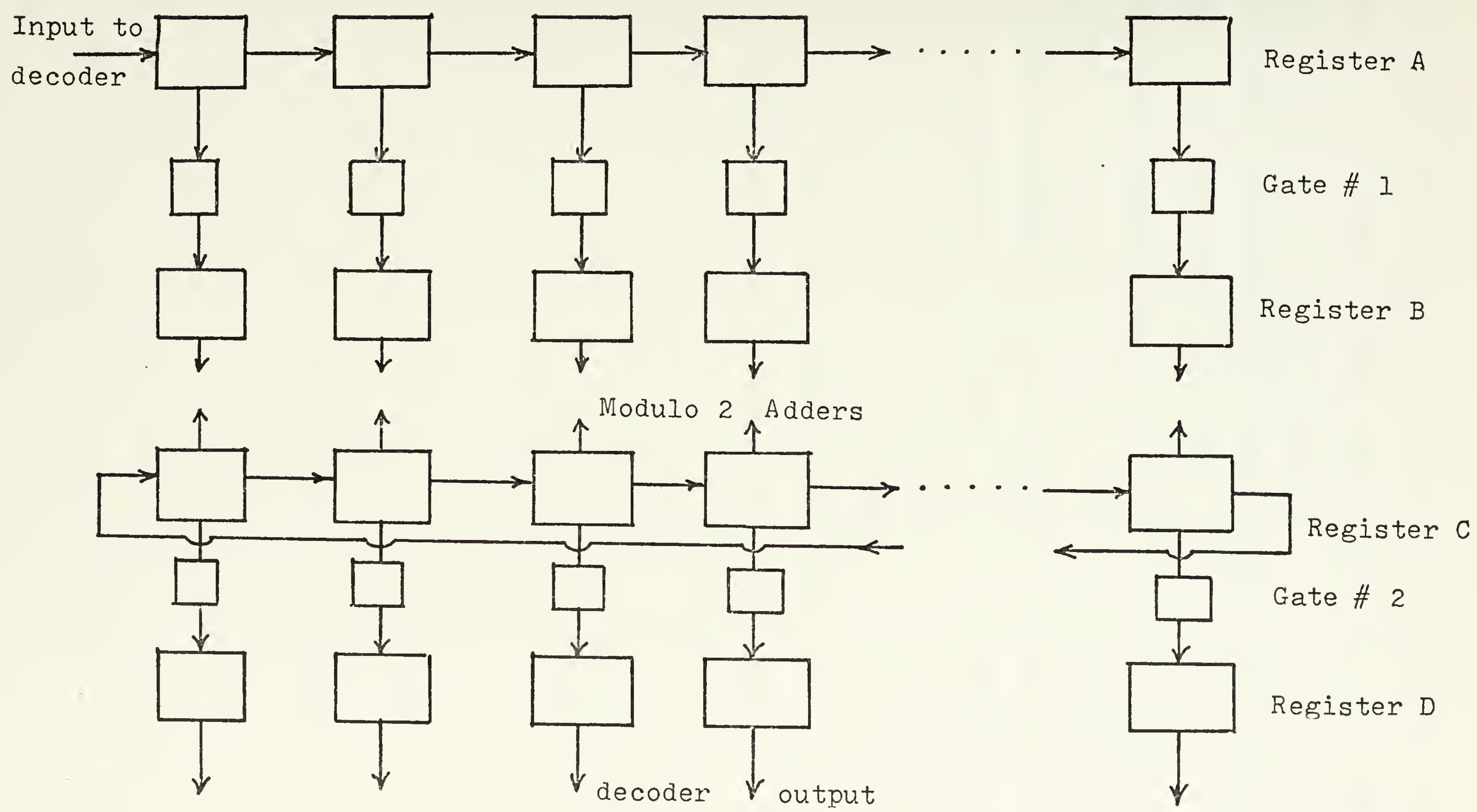


Figure 9. Minimum Distance Decoder

transmitted word $[z]_{1,n}$ is the noise due to the channel. After time $t = n-1$ gate #1 opens and the contents of register A enters register B. Since register C contains the coefficients of the check polynomial and every cyclic shift of this polynomial is another code word, the corresponding bits of register B and C are added (modulo 2). If the resulting number of ones after addition in modulo 2 is greater than d/2 (indicating the Hamming distance between registers B and C is greater than d/2), the contents of register C will be shifted right and the corresponding bits of registers B and C are added together again to check if the resulted number of ones are less than or equal to d/2. Note that as each clock pulse shifts register C a new digit of the next word to be decoded is shifted into register A therefore after n shifts of register C, register A will contain the complete code word, thus the decoding process is continuous. If during any one of the checks between registers B and C, the resulting number of ones is equal to or less than d/2, gate #2 opens and contents of register C will be transferred to register D as a corrected word. However, if for any of the n clock pulses none of the additions result in d/2 or less number of ones, the cleared contents of register D (the zero word) will be taken as the corrected word. At $t = n-1$ register A has the next received word $[w + z]_{1,n}$, gate #1 opens and contents of register A enters the register B, register D is cleared and another cycle begins. As a property of cyclic codes if a received word has a distance less than d/2 to the original transmitted word (code alphabet)

it can not be simultaneously that closer to another code word of the code alphabet. After one word is decoded, there is no need to continue the modulo 2 additions between registers B and C until the next received word has been completely shifted into register A.

When a reducible polynomial is used for the generator polynomial, such as $G(x) = x^8+x^7+x^5+x^4+1 = (x^4+x+1)(x^4+x^3+x^2+x+1)$ and with check polynomial $H(x) = x^7+x^6+x^4+1$, all the cyclic shifts of the coefficients of the check polynomial H(x) is not enough to represent all of the possible code alphabet letters. Since the given (15,8) code cited above has $2^8-1 = 255$ possible words (excluding the zero word), and since all the periods of the generator polynomial have length 15 (except the zero trivial cycle) one needs (255/15)=17 register C's (as described in Figure 9). With the different code words belong to the different cycle sets shown in Figure 7. Because as a property of cyclic codes, one code can be defined as a linear combinations of others and because the rank of generator matrix $[G]_{k,n}$ is 8, it can be shown the number of register C's can be reduced to 8, instead of 17.

The flow chart shown in Figure 10 performs the modulo 2 addition between registers C and B to check the distance between original word and the received word to see if the distance is equal to or less than d/2. When the condition is met the contents of R3 is taken as the decoded word.

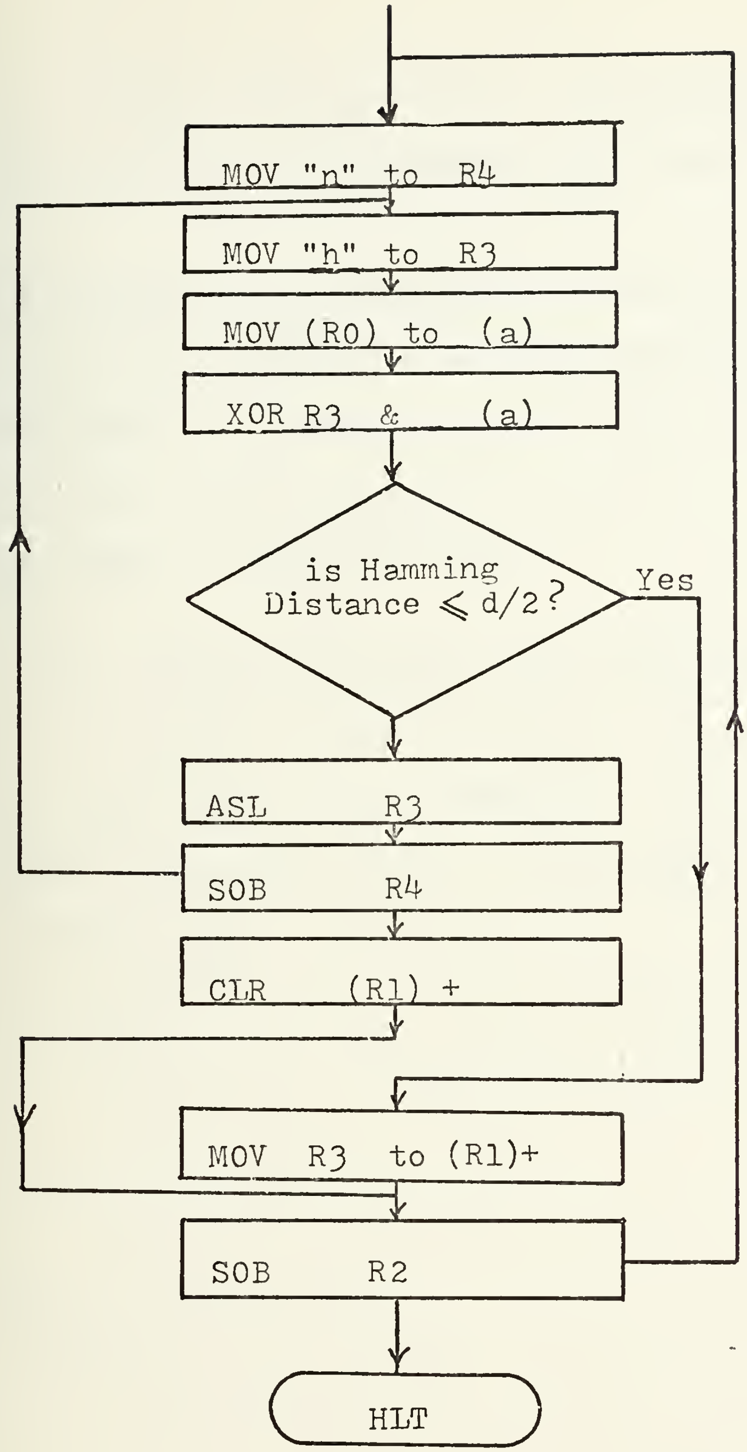


Figure 10. Minimum distance decoder flow chart (for irreducible polynomial).

The notation used for the minimum distance decoder is the same used for the encoder flow chart.

The starting address of received word is in R0

The starting address of decoded information word is in R1

The number of received messages is in R2

The parity check polynomial is in (h)

The number of word bits is n

3. Computer decoding using the syndrome method

Another decoding system is achieved by using a decoding table stored in the computer's memory. In computer application of encoder section, the generator matrix for any code is defined by:

$$[G]_{k,n} = \begin{bmatrix} 1 & 0 & 0 & 0 & \cdots & 0 & h_{1,k-1} & h_{1,k-2} & \cdots & h_{1,0} \\ 0 & 1 & 0 & 0 & \cdots & 0 & h_{2,k-1} & h_{2,k-2} & \cdots & h_{2,0} \\ & & \cdot & & & \cdot & & \cdot & & \cdot \\ & & \cdot & & & \cdot & & \cdot & & \cdot \\ & & \cdot & & & \cdot & & \cdot & & \cdot \\ & & \cdot & & & \cdot & & \cdot & & \cdot \\ & & \cdot & & & \cdot & & \cdot & & \cdot \\ 0 & 0 & 0 & 0 & \cdots & 1 & h_{k-1} & h_{k-2} & \cdots & h_0 \end{bmatrix}$$

The check matrix of the same code can be represented as:

$$
[H]_{n,m} = \begin{bmatrix}
h_{1,k-1} & h_{1,k-2} & \cdots & h_{1,0} \\
h_{2,k-1} & h_{2,k-2} & \cdots & h_{2,0} \\
\cdot & \cdot & \cdot & \cdot \\
\cdot & \cdot & \cdot & \cdot \\
\cdot & \cdot & \cdot & \cdot \\
\cdot & \cdot & \cdot & \cdot \\
h_{k-1} & h_{k-2} & \cdot & h_0 \\
1 & 0 & \cdot & 0 \\
0 & 1 & 0 & 0 \\
\cdot & \cdot & \cdot & \cdot \\
\cdot & \cdot & \cdot & \cdot \\
0 & 0 & 0 & 1
\end{bmatrix}
$$

Matrix multiplication of any original code word by the check matrix will result in a $[0]_{1,m}$ matrix

$$[w]_{1,n}\,[H]_{n,m} = [0]_{1,m}$$

However if any error present in the received code word the result will not be the $[0]_{1,m}$ matrix,

$$[w + z]_{1,n}[H]_{n,m} = [w]_{1,n}[H]_{n,m} + [z]_{1,n}[H]_{n,m}$$

$$= \quad [0]_{1,m} = [z]_{1,n}[H]_{n,m} = [S]_{1,m}$$

The matrix $[S]_{1,m}$ is called the syndrome. For every error pattern $[z]_{1,n}$ has a unique syndrome $[S]_{1,m}$. By simply listing the correctible error patterns and versus their syndromes in table stored in the computer, one can find the error pattern readily after the syndrome has been found by

matrix multiplication of received by check matrix. The error pattern when exclusive OR'ed with the received word yield to most probable transmitted word.

$$[w + z]_{1,n} + [z]_{1,n} = [w]_{1,n} \quad \text{(modulo 2)}$$

Restated step by step:

(1) List all of the possible syndromes $[S]_{1,m}$ and error patterns $[Z]_{1,n}$ (or the other name is 'corrector') due to given syndrome

(2) Multiply the received word by check matrix to obtain the syndrome.

(3) From decoding list, get the corrector due to obtained syndrome

(4) Add this corrector to the received word in order to correct the errors

Example: The alphabet of code generated by polynomial

$$G(X) = X^4+X+1$$

is given by the $[W]_{n,n}$ matrix (one cycle of register C)

$$
[W]_{n,n} = \begin{bmatrix}
1&0&0&0&1&0&0&1&1&0&1&0&1&1&1\\
0&1&0&0&1&1&0&1&0&1&1&1&1&0&0\\
0&0&1&0&0&1&1&0&1&0&1&1&1&1&0\\
0&0&0&1&0&0&1&1&0&1&0&1&1&1&1\\
1&1&0&0&0&1&0&0&1&1&0&1&0&1&1\\
1&0&1&0&1&1&1&1&0&0&0&1&0&0&1\\
1&0&0&1&1&0&1&0&1&1&1&1&0&0&0\\
0&1&1&0&1&0&1&1&1&1&0&0&0&1&0\\
0&1&0&1&1&1&1&0&0&0&1&0&0&1&1\\
0&0&1&1&0&1&0&1&1&1&1&0&0&0&1\\
1&1&1&0&0&0&1&0&0&1&1&0&1&0&1\\
1&1&0&1&0&1&1&1&1&0&0&0&1&0&0\\
1&0&1&1&1&1&0&0&0&1&0&0&1&1&0\\
0&1&1&1&1&0&0&0&1&0&0&1&1&0&1\\
1&1&1&1&0&0&0&1&0&0&1&1&0&1&0
\end{bmatrix}
$$

$\longleftarrow k \longrightarrow \longleftarrow\!\!-\!\!-\!\!-\!\!-\; m \;-\!\!-\!\!-\!\!-\!\!\longrightarrow$

The last "m" columns of $[W]_{n,n}$ matrix gives the syndrome of corresponding corrector of first "k" columns (error pattern).

Example:

Error pattern $[Z]_{1,n}$	Syndrome $[S]_{1,m}$
1 0 0 0 0 0 0 0 0 0 0 0 0 0 0	1 0 0 1 1 0 1 0 1 1 1
1 0 0 0 1 0 0 0 0 0 0 0 0 0 0	0 0 0 1 1 0 1 0 1 1 1
1 0 0 0 0 0 0 0 0 1 1 0 0 0 0	1 0 0 1 1 1 0 0 1 1 1

As one sees any additional errors in the check digits yield a corresponding digit changed in the syndrome, this property of syndromes make the decoding table easier and shorter.

To make up the decoding table, first list the error patterns $[z]_{1,n}$ according to their weight using the $[w]_{n,n}$ matrix and list the corresponding syndromes. Since one only needs to

correct the errors in the information digits, it is enough to add the first k digits of error pattern $[z]_{1,n}$ to corresponding digits of received word.

4. <u>Syndrome method decoder flow chart</u> (Figure 11)

First syndromes and error patterns (correctors) assumed listed in memory. The program multiples the received word by check matrix $[H]_{n,m}$ and result is the syndrome $[S]_{1,m}$ matrix from this syndrome, listed error pattern are obtained and added to received word for correction (in modulo 2)

The starting address of received word is in R0

The starting address of decoded information words is in R1

The number of received message is in R2

The starting address of syndromes is in memory address "a"

The starting address of correctors is in memory address "b"

The starting address of the rows of parity check matrix $[H]_{n,m}$ is in memory address "h"

"n" is the number of code word bits.

Additional notation used is given below

TST = test address

BEQ = branch if equal to

COMP = compare two addresses

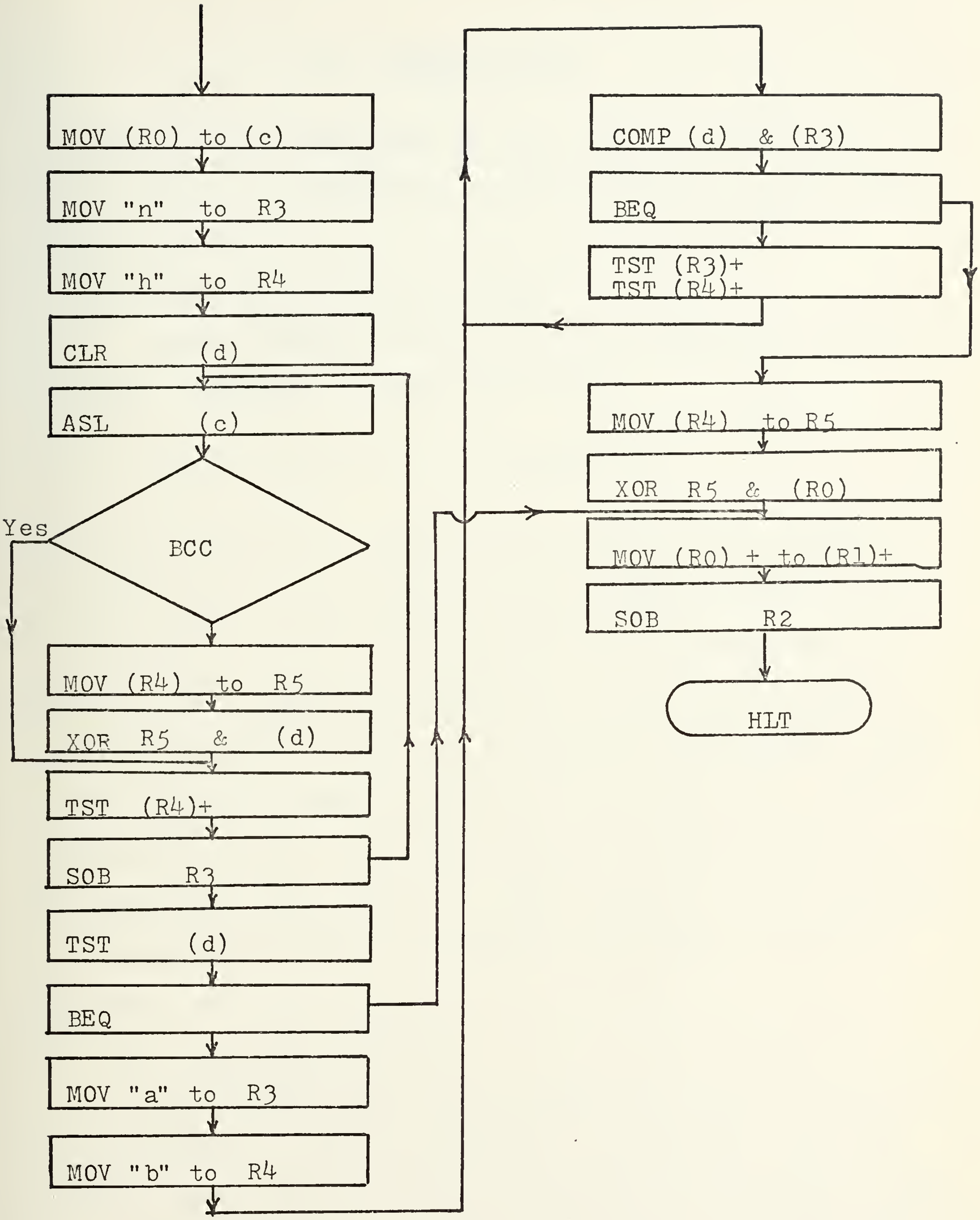


Figure 11. Syndrome Method Decoder Flow Chart.

III. CHANNEL NOISE

Channel noise simulation by the computer can be described in two parts, (1) generation of random numbers (2) generation of noise sequence.

(1) Generation of random numbers

Random numbers generated in this program were obtained by using the Lehmer congruential method.

$$x_{n+1} = a\, x_n + b \qquad (\text{modulo } T_0)$$

Letting a = 257, b = 3, the starting number x_0 is chosen as a prime number and changed for each sequence, $T_0 = 2^{16}$ (all in decimal). Therefore the period of the random number sequence is 2^{16}. After the generation of the random numbers, every k^{th} of them is taken as a selected random number, where K is called the indexing factor which we shall see, related to the channel β in the binary symmetric channel (Figure 2). The selected random number is taken to specify the address of a random number field, and a marker '1' is put in to this address.

(2) Generation of noise

The markers put in the random number field were taken to designate the ones in a sequence of zeros and ones, the sequence has a one to one correspondence to the random number field. The resulting sequence is exclusive OR'ed with successive words of the encoded message, thereby simulating the

introduction of error bits. Word corresponding to carriage return was given noise immunity, but in the probability of error calculations, the number of carriage returns were subtracted from the number of inputs.

Figures from 12 to 16 represent the actual and binomial distribution of errors due to the indexing factors K used. The β's is taken as a probability of an error (or a 1) and is calculated by counting the number of ones out of 153000 bits (in decimal). It is found that the simulated noise sequences more or less closely follow the binomial distribution except for K equal to powers of 2. It is helpful to point out that for any binary symmetric channel, β is very closely related to signal to noise ratio (S/N).

The channel used in this thesis is a memoryless binary symmetric channel. Memoryless channel is the one which noise doesn't depend upon the previous-in time - value). Binary symmetric channel is the one which the probability of bit zero to change the bit one is equal to the probability of bit one to change the bit zero. Table I represents some indexing factors K versus binary symmetric channel β's.

A. FLOW CHART DESCRIPTION OF NOISE PROGRAM (Figure 17)

This program describes the generation of random numbers to put into random number field by the Lehmer congruential method. After this program one can combine the markers according to it's word length (code length). Additional notation used in the flow chart:

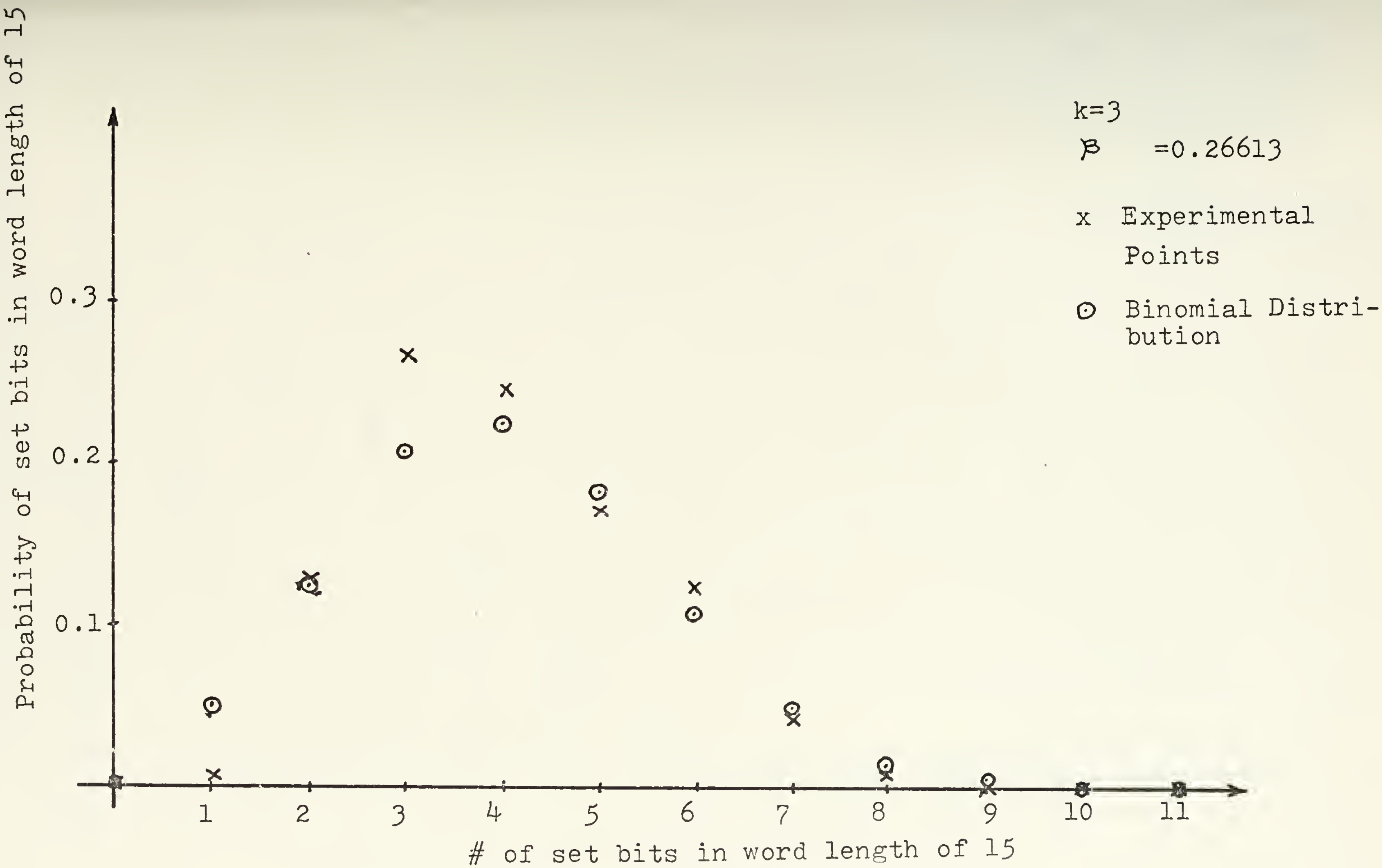


Figure 12. Simulated Noise (K=3, β = 0.26613) vs. Binomial Distribution.

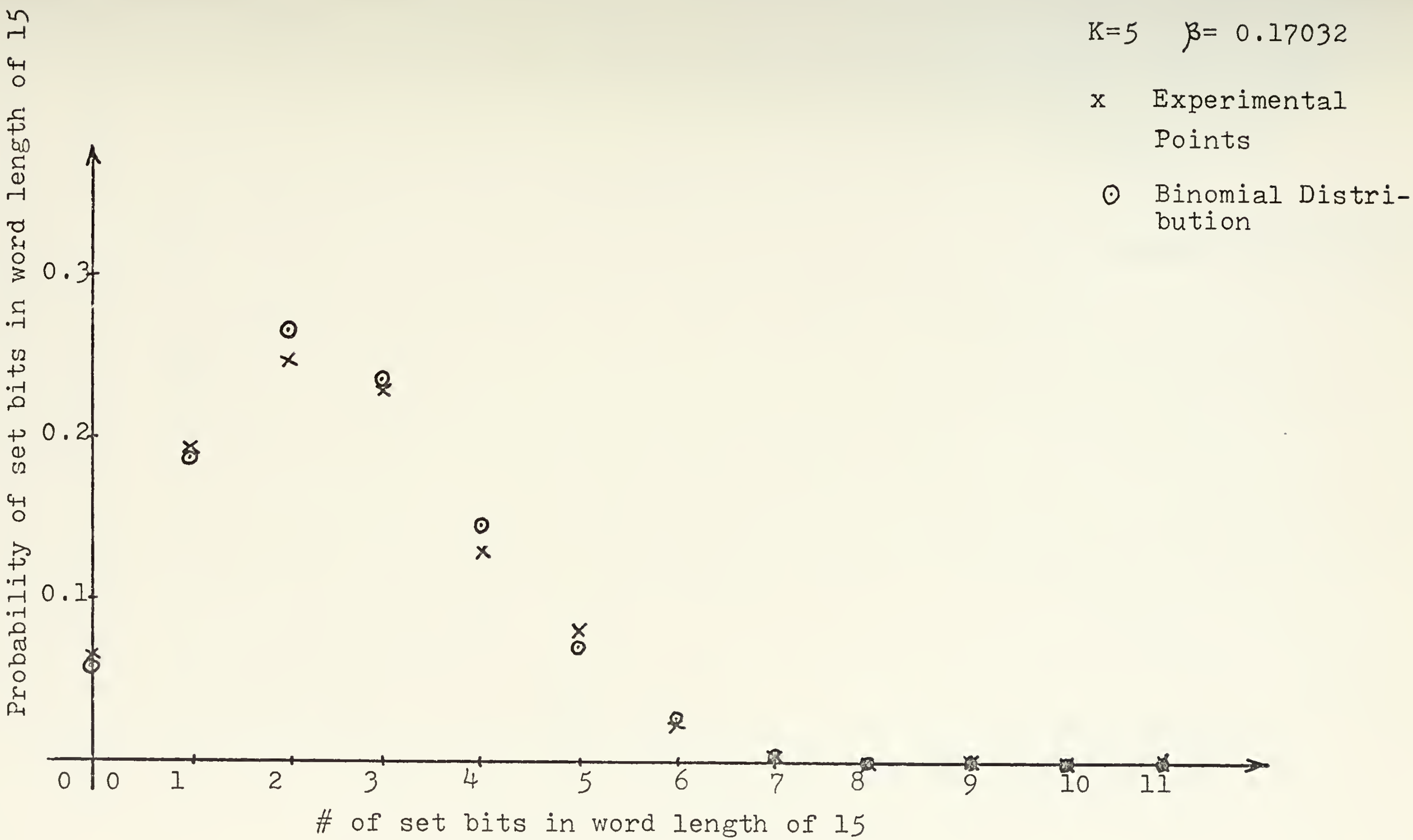


Figure 13. Simulated noise (K=5, β=0.17032) vs. Binomial Distribution.

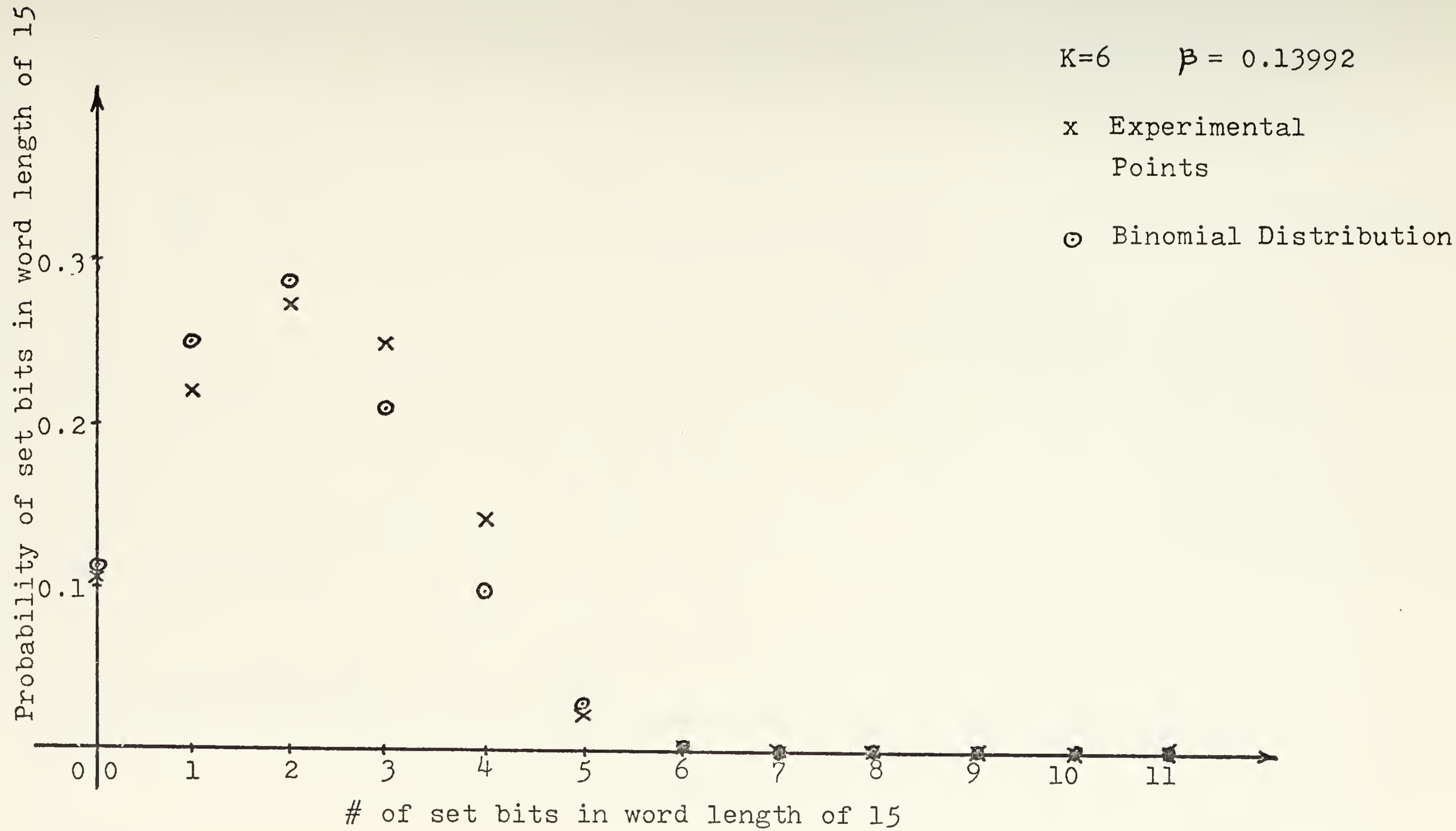


Figure 14. Simulated Noise (K=6, β =0.13992) vs. Binomial Distribution.

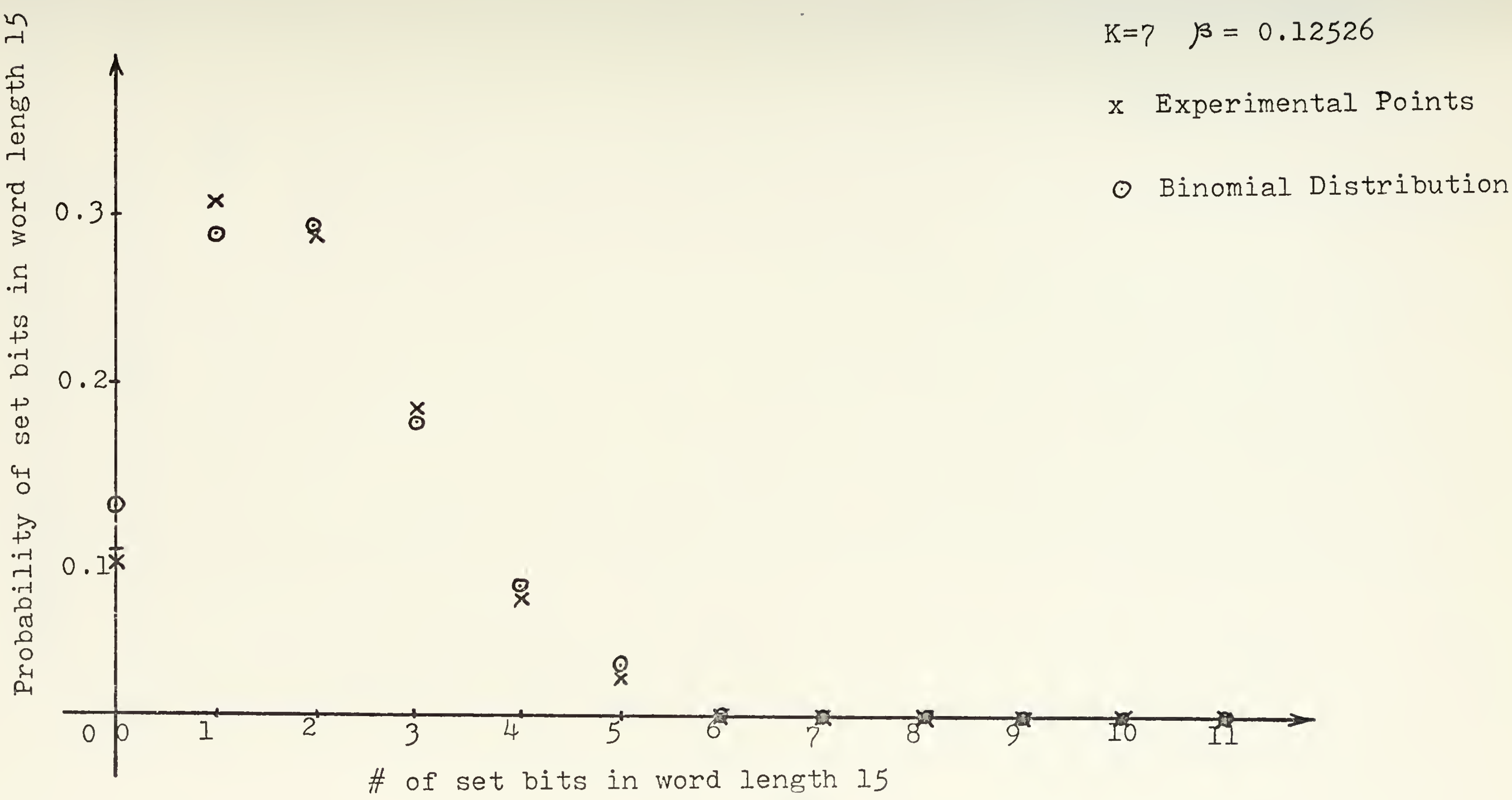


Figure 15. Simulated Noise (K=7, β=0.12526) Vs. Binomial Distribution.

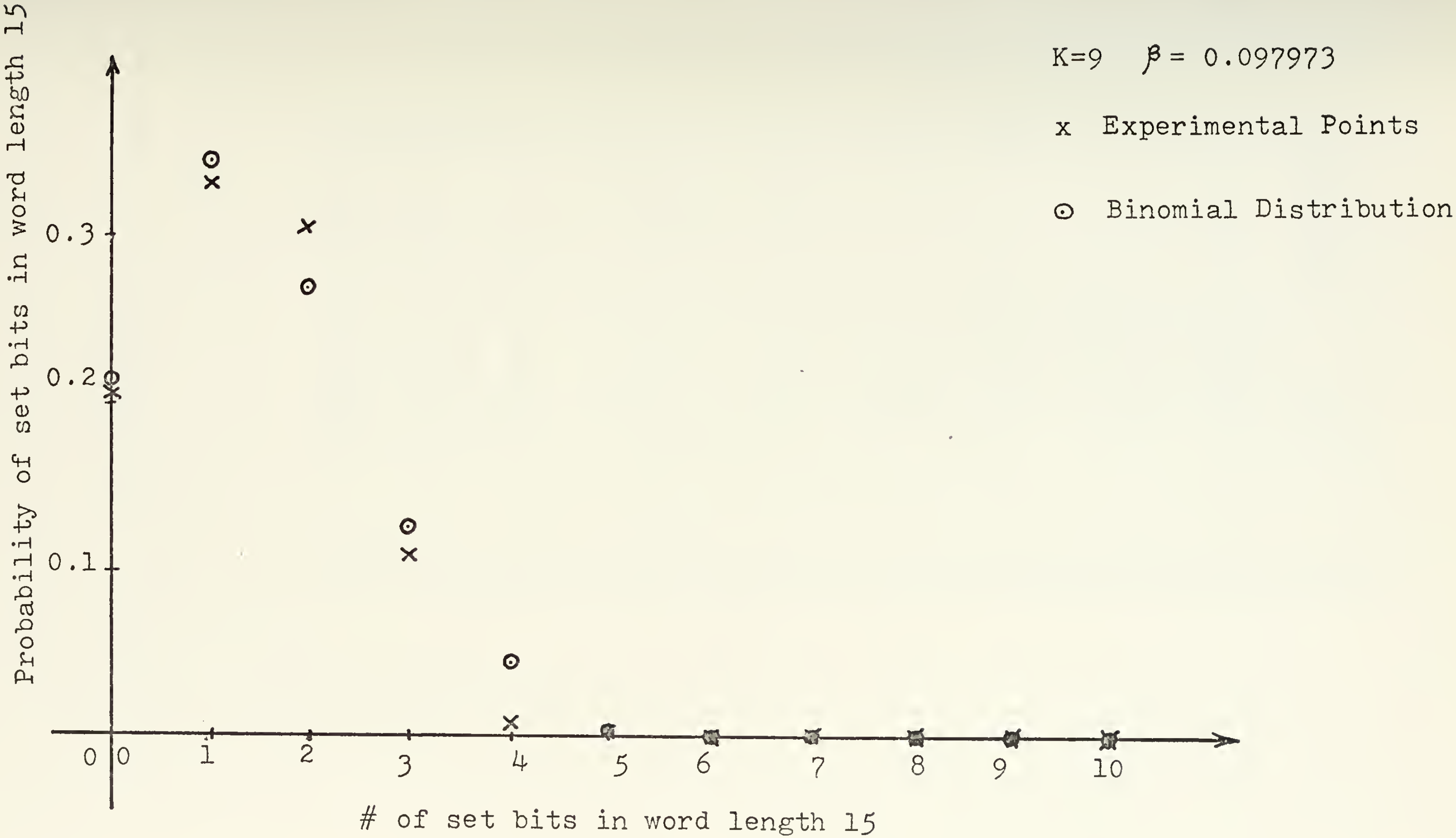


Figure 16. Simulated Noise (K=9 β = 0.097973) vs. Binomial Distribution.

Indexing Factor(k)	Channel β
3	0.26613
5	0.1709
6	0.13992
7	0.12521
9	0.09797
10	0.0945
11	0.0859
12	0.07506
13	0.07050
15	0.063242
18	0.052187
21	0.04561
24	0.03855
29	0.032961
36	0.026429
41	0.023132

Table I. Indexing Factor K vs. Channel β.

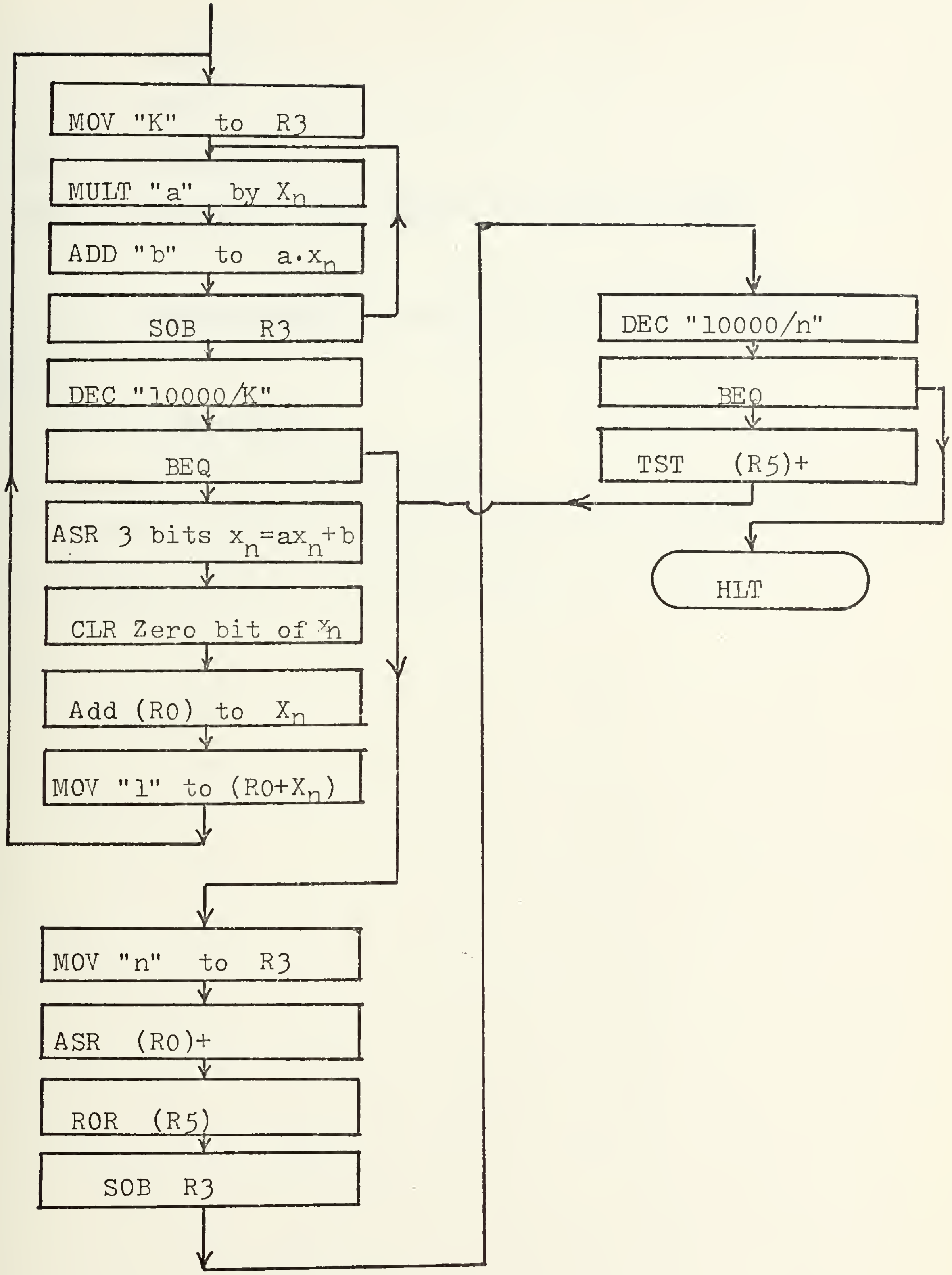


Figure 17. Flow Chart of the Simulated Noise Program.

MULT = multiply

ADD = add

DEC = decrement

BEQ = branch if the result is equal to zero

K is the indexing factor (defines β for binary symmetric channel

n is the word length

x_0 is the starting prime number

The starting address of random number field is in R0

The starting address of noise field is in R5

IV. BEST CODE DETERMINATION

The noisy channel theorem [Ref. 1]: Let a discrete channel have the capacity C bits/sec. and a discrete source has the entropy per second H. If $H < C$ there exists a coding system such that the output of the source can be transmitted over the channel with an arbitrarily small frequency of errors. If $H > C$, it is possible to encode the source so that the equivocation is less than $H-C+\epsilon$ where ϵ is arbitrarily small. There is no method of encoding which gives an equivocation less than H-C. The discrete source entropy for long messages consisting of discrete symbols is given by

$$H(x) = - \sum_{i=1}^{n} p_i \log p_i$$

where p_i is the probability of occurrence of a given symbol, in the above formula it is assumed that the each new symbol is independent of the proceeding ones. In the situation where the symbols are transmitted over a noisy channel a given symbol x_i may be received as y_i. Shannon's measure of equivocation or uncertainity at the receiver is to what was actually transmitted is defined as:

$$H(x/y) = \sum \sum P(x_i, y_i) \log P(x_i/y_i)$$

For the binary symmetric channel where the β is a conditional probability of an error being made in the channel

$$H(x/y) = - (\beta \log \beta + (1-\beta) \log (1-\beta))$$

Then the channel capacity

$$C = H(x) - H(x/y),$$

maximized for H(x).

In the following discussion the probability of error will be used instead of equivocation, the two concepts were closely related but the probability of error is more convenient.

For clearity β and P(e) will be defined here as follows:

$\beta = P(0/1) = P(1/0)$ for the channel

$$P(e) = \frac{\text{number of wrong decoded words}}{\text{total number of words}}$$

note that in the case of ASCII characters where each character is represented by 8 binary digits, less than the total number of digits representing the ASCII character may be coded into a single word or on the other hand one or more ASCII characters plus a fraction of a character might be coded into a word.

A 'best code' means one that least probability of error for any given channel β and highest rate R=k/n. The error correction ability of the code can be derived from the Varsharmov - Gilbert - Sacks condition

$$2^m > \sum_{i=0}^{2e-1} \binom{n-1}{i}$$

and closely related to rate R of the code. After definition of the code rate R, and word length n one can find the number of correctible e-tuple errors from Varsharmov - Gilbert -

Sacks condition. The theoretical value of probability of error is given by [Ref. 3]:

$$p(e) = 1 - \left[\sum_{i=0}^{e} N_i \quad \beta^{i}(1-\beta)^{n-i}\right.$$

where N_i is the number of correctible e-tuple errors, where e_i=0,1,2,.., up to the maximum number of correctible errors per word.

The Hamming distance (d) as defined earlier is the minimum distance between code words. If d happens to be even and the maximum value of e is given by (d-1)/2, this will yield a fraction. Then number of maximum e_i-tuple errors is given by [Ref. 4].

$$\frac{\text{number of correctible } \; d/2 \text{ errors}}{\text{total number of } \; d/2 \text{ erros}} = 1 - \frac{\frac{u(u+1)}{2}}{\binom{n}{d/2}}$$

where $u = \dfrac{d!}{\left(\frac{d}{2}\right)! \left(\frac{d}{2}\right)!}$

Reduction in the probability of error, keeping the channel constant, results also in a reduction of the code rate. By working backward, for any given probability of error and word length, (for a given channel β), from the Varsharmov - Gilbert - Sacks condition and the theoretical value of probability of error equation, one can find the information length and code rate. Figure 21 to be described in the conclusion section relates β, rate R and the probability of error.

V. RESULTS

Three different code rates vs. different channel β's were examined in this thesis. To get the probability of error, approximately 40000 words were mixed with noise for each given binary symmetric channel β. For a (15,4) code (Rate R = 4/15), two different decoding systems (Minimum distance decoder and syndrome method decoder) and two different generator polynomials G(x) were used to find the probability of error.

Table II represents the probability of error vs. channel β's of the code (15,4) for two different generator polynomials and two different decoding systems. Probability of errors for those systems and for different generator polynomials for given channel β's are in the limits of $\pm$ 1% difference. This means that for any given code rate, the minimum distance decoder and the syndrome method decoder gives the same probability of error. Furthermore using other generator polynomials of the same rate does not change the probability of error. Figure 18 shows the three dimensional representation of the (15,4) code.

Figure 19 shows the three dimensional representation of the (15,8) code using the syndrome method decoder. As one sees the shape of the P(e) vs. β curve is a S-shaped. As β increases, P(e) approaches 1.0 as a limit.

Channel β	P(e) $G(X)=x^4+x+1$ Min. Distance Decoder	P(e) $G(X)=x^4+x^3+1$ Min. Distance Decoder	P(e) $G(X)=X^4+x+1$ Syndrome Decoder
0.07050	5.4480×10^{-3}		
0.09797	2.9176×10^{-2}	3.176×10^{-2}	2.655×10^{-2}
0.12426	6.2425×10^{-2}	5.795×10^{-2}	5.817×10^{-2}
0.13992	1.2542×10^{-1}	1.0479×10^{-1}	1.1442×10^{-1}
0.1709	1.8780×10^{-1}	1.885×10^{-1}	1.778×10^{-1}
0.26613	4.9052×10^{-1}	4.878×10^{-1}	4.8309×10^{-1}

Table II. P(e) vs. Channel β for the code (15,4).

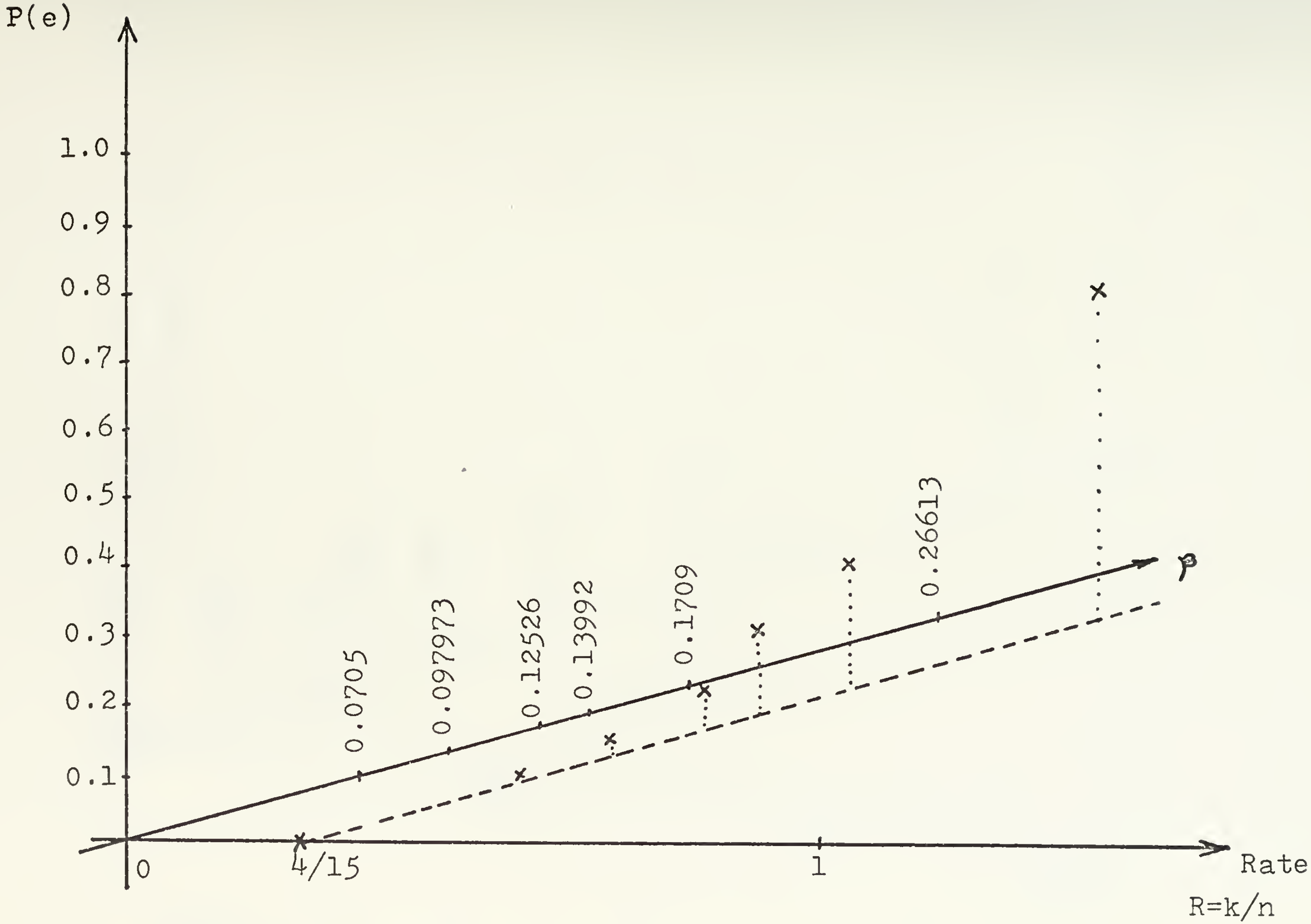


Figure 18. P(e) vs. Channel β for the code (15,4)
$G(X) = x^4+x+1$.

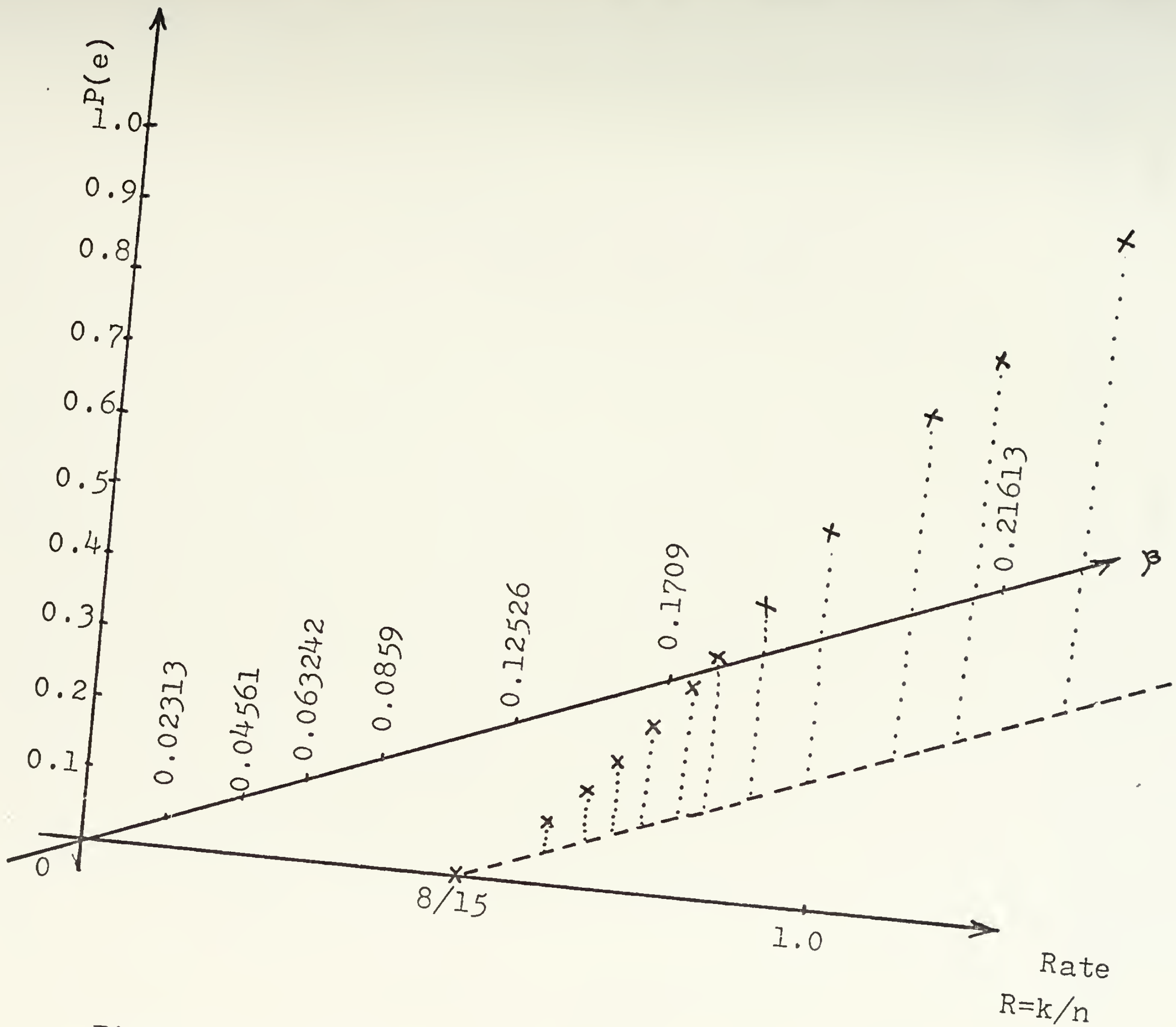


Figure 19. P(e) vs. Channel β for the code (15,8)
$G(X) = x^8+x^7+x^6+x^4+1$.

Figure 20 shows the three dimensional representation of the (21,16) code using the syndrome method decoder. The shape of the P(e) vs. β curve is also S-shaped, but the steepness of the curve is much greater than for the (15,8) P(e) vs. β curve.

Tables III and IV shows the probability of error versus channel β 's for the codes (15,8) and (21,16) respectively.

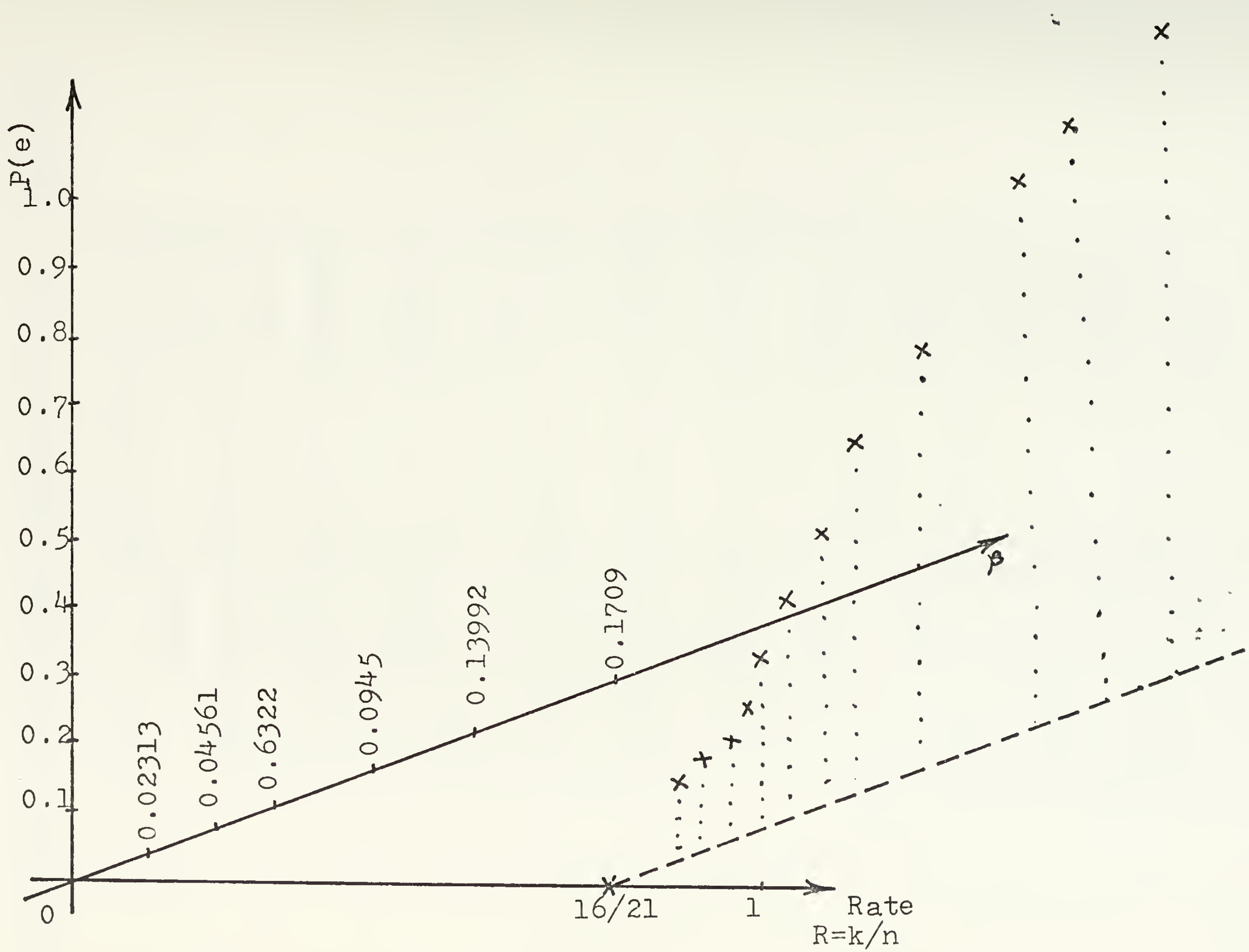


Figure 20. P(e) vs. Channel β for the code (21,16)
$G(X) = x^{16}+x^{15}+x^{14}+x^{13}+x^{12}+x^{10}+x^{8}+x^{5}+x^{4}+1$

Channel	P(e)
0.023132	0.0433
0.03855	0.07396
0.04561	0.09803
0.052187	0.13573
0.063242	0.17932
0.07050	0.20790
0.0859	0.27170
0.097973	0.35667
0.12526	0.48701
0.13992	0.54223
0.1709	0.6636
0.26613	0.90378

Table III. P(e) vs. Channel Beta for the Code (15,8).

Channel	P(e)
0.023132	0.11446
0.026429	0.13227
0.032961	0.14298
0.03855	0.18854
0.04561	0.25066
0.05218	0.31940
0.06324	0.39899
0.07506	0.51879
0.0945	0.61530
0.12526	0.80895
0.13992	0.86483
0.1409	0.94532

Table IV. P(e) vs. Channel Beta for the code (21,16).

VI. DISCUSSION AND CONCLUSIONS

Two different type of decoder (syndrome method and minimum distance decoder) discussed in this thesis give the same probability of error for the same channel β. For the irreducible polynomial code the minimum distance decoder is faster than the syndrome method decoder. For the (15,4) code the number of syndromes are $S(x) = 2^{15-4} = 2^{11} = 2048$ with the same number of correctors (error patterns). After multiplication of the received word by the parity check matrix, the result (syndrome $[S]_{1,m}$) will be checked if it is equal one of the 2048 syndromes previously listed in the computer memory in order to find the corrector. But for the same irreducible polynomial code (15,4) there is only one maximum cycle set; therefore the received word will be checked if it is in distance d/2 or less to these 15 words, and so is much shorter in time than the syndrome method decoder. On the other hand, for the reducible polynomial code (21,16), there are $2^{21-16} = 2^5 = 32$ syndromes and the same number of correctors (error patterns). After multiplication of the received word by the parity check matrix H(x), the result (syndrome $[S]_{1,m}$) will be checked if it is equal to one of the 32 syndromes previously listed in the computer memory, to find the corrector. But for the same reducible polynomial there are $2^{16}/21 = 3121$ maximum cycles length of 21 (excluding zero trivial cycle). Therefore there are $3121 \times 21 = 65536$

words to be checked if the received word is in d/2 distance apart from those words, which is much longer in time than the syndrome method decoder. As a result one can predict for any given code (n,k) from the number of syndromes and number of maximum cycles for the minimum distance decoder which method is shorter in time.

For any given code (n,k), sometimes there is more than one generator polynomials, then the probability of error results do not depend upon the polynomial being used for any specific word length and information length (See tabulated results for (15,4) code in Table II).

The shape of the P(e) vs. channel β curve for any given code rate is an S-shaped curve: the greater the code rate the steeper the S-shaped curve. Figure 21 combines the calculated probability of errors for the 3 different codes, were investigated. For any given channel β and permissible probability of error one can obtain the maximum code rate from the given figure above.

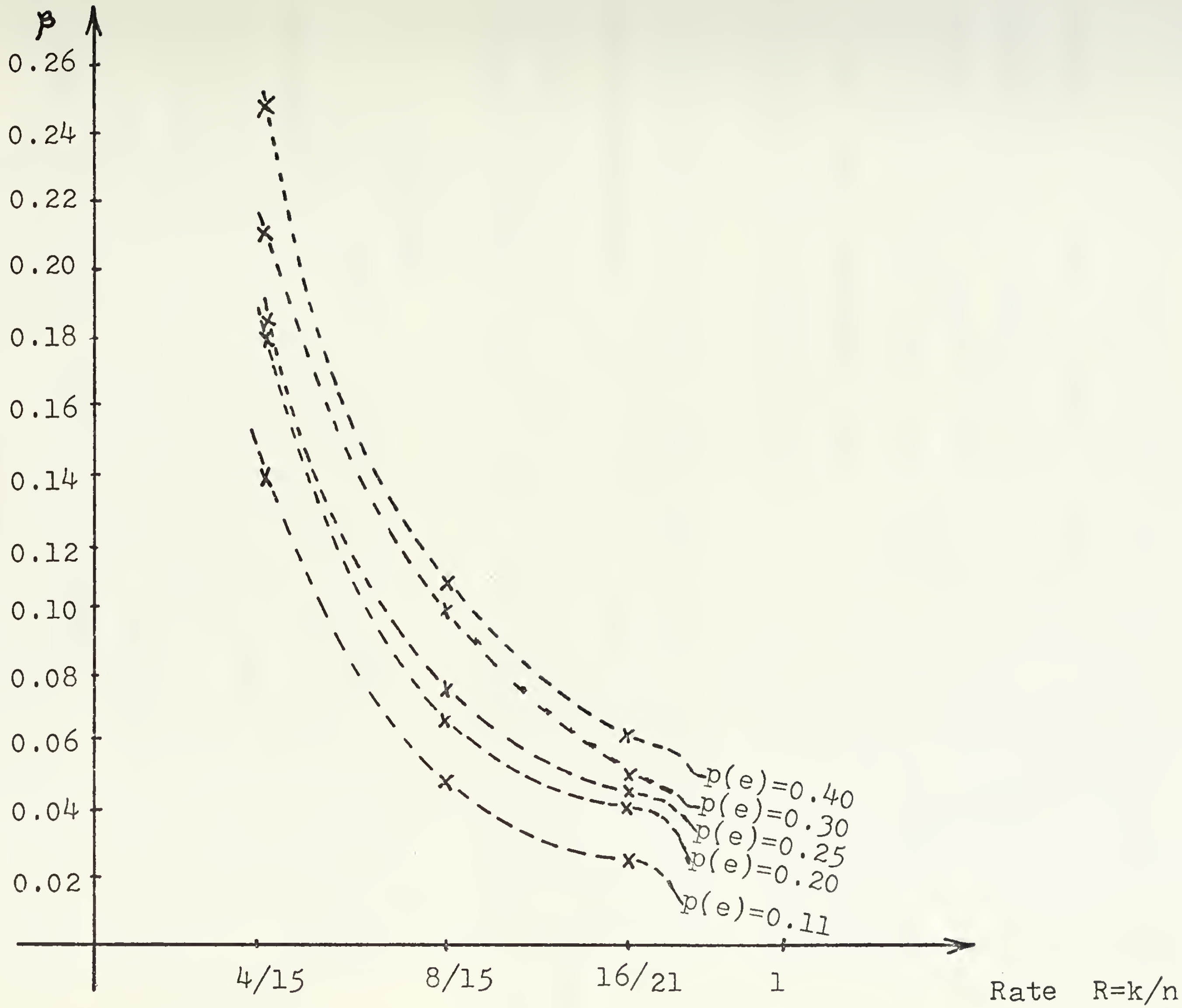


Figure 21. Rate R vs. β for any P(e).

APPENDIX A

Program flow:

1. Noise program; First generates random numbers and put a marker "1" due to random number in the random number field between memory addresses 57000 - 77776. Second combines the markers in the random number field according to word length and puts the resulted noise between memory addresses 32000-32400. Noise program starts at address 10000, end of the program at the address 10212.

2. Input program; Takes the input messages and puts addresses between 51000-52000 in sequential order. Location (50100) counts the number of input messages. Input program starts at address 20000, end of the program at address 20122.

3. Encoder;

A. Encoder for the (15,4) code; Takes the input messages between addresses 51000-52000, encodes and puts between addresses 52000-54000. Generator matrix is in addresses between 50200-50206, location (50140) is used for ASL, (50142) is used for encoding operations. Program starts at address 20124, end of the program is at 20240.

B. Encoder for the (15,8) code; Takes the input messages between addresses 51000-52000, encodes and puts between addresses 52000-54000. Generator matrix is in addresses between 20610-20626, location (50104) is used for encoding,

(50102) is used for ASL operations. Program starts at address 20124, end of the program is at 20204.

4. Noise mixing sequence; Adds the noise between addresses 32000-32400 to the encoded messages between addresses 52000-54000 (in modulo 2). Carriage return has the noise immunity. Noise mixing sequence for the code (15,4) is between 20242-20332, for the code (15,8) is between 20206-20250.

5. Decoder;

A. Syndrome method decoder for the code (15,4); Takes the transmitted message mixed with noise from the addresses between 52000-54000, decodes, corrects the errors if they are correcible and puts the addresses between 56000-57000. Parity check matrix H(X) is between addresses 50210-50244, syndrome S(X) is between 50246-50304, corrector (error pattern) Z(X) is between 50306-50340.

B. Minimum distance decoder for the code (15,4); Takes the transmitted message mixed with noise from the addresses between 52000-54000, decodes, corrects the errors if they are correctible and puts the addresses between 56000-57000. Register C described in Figure 9 is in address (50104).

C. Syndrome method decoder for the code (15,8); Takes the transmitted message mixed with noise from the addresses between 52000-54000, decodes, corrects the errors if they are correctible and puts the addresses between 56000-57000. Parity check matrix H(X) is in addresses between 20630-20664, corrector (error pattern) Z(X) is in addresses between 20666-20774, syndrome S(X) is in addresses between 20776-21104.

6. Output program; Takes the decoded message from addresses 56000-57000 and writes it out, program is in addresses between 20704-21000 for the code (15,4) minimum distance decoder, 21506-21604 for the code (15,4) syndrome method decoder, 20406-20502 for the code (15,8) syndrome method decoder.

APPENDIX B

NOISE PROGRAM ADDRESS 10024 HAS THE STARTING RANDOM NUMBER ADDRESS 10030 HAS THE INDEXING FACTOR.

```
W
010000 /012700
010002 /032000
010004 /012701
010006 /001000
010010 /005020
010012 /077102
010014 /000240
010016 /012700
010020 /057000
010022 /012746
010024 /012705
010026 /012746
010030 /000030
010032 /011667
010034 /000026
010036 /012704
010040 /177304
010042 /012714
010044 /010000
010046 /012637
010050 /177300
010052 /011467
010054 /000030
010056 /012701
010060 /177316
010062 /012703
010064 /000030
010066 /012624
010070 /012714
010072 /000401
010074 /014446
010076 /062716
010100 /000003
010102 /077307
010104 /005327
010106 /000000
010110 /001414
010112 /011614
010114 /005044
010116 /012711
010120 /177775
010122 /005724
010124 /042714
010126 /000001
*
```

```
W
010130 /060014
010132 /012774
010134 /000001
010136 /000000
010140 /000750
010142 /005026
010144 /012700
010146 /057000
010150 /012701
010152 /032000
010154 /012702
010156 /000177
010160 /012703
010162 /000020
010164 /006220
010166 /006011
010170 /077303
010172 /005721
010174 /012703
010176 /000005
010200 /006220
010202 /006011
010204 /077303
010206 /005721
010210 /077215
010212 /000000
*
```

APPENDIX C

```
INPUT PROGRAM  ADDRESS 50100 HAS THE NUMBER OF
INPUT MESSAGES
```

The character @ defines the end of the program

```
W
020000 /012700
020002 /051000
020004 /005002
020006 /105737
020010 /177560
020012 /100375
020014 /113710
020016 /177562
020020 /122710
020022 /000300
020024 /001432
020026 /005202
020030 /105737
020032 /177564
020034 /100375
020036 /111037
020040 /177566
020042 /122720
020044 /000215
020046 /001401
020050 /000756
020052 /012701
020054 /000012
020056 /105737
020060 /177564
020062 /100375
020064 /112737
020066 /000200
020070 /177566
020072 /077107
020074 /105737
020076 /177564
020100 /100375
020102 /112737
020104 /000212
020106 /177566
020110 /000736
020112 /010237
020114 /050100
020116 /005000
020120 /005002
020122 /000000
*
```

APPENDIX D

```
ENCODER FOR THE CODE (15,4)
ROWS OF GENERATOR MATRIX G(X) IN ADDRESSES BETWEEN
50200 -50206


W
020124 /012700
020126 /051000
020130 /000240
020132 /000240
020134 /013702
020136 /050100
020140 /112037
020142 /050140
020144 /012703
020146 /000002
020150 /012704
020152 /000004
020154 /012705
020156 /050200
020160 /005037
020162 /050142
020164 /012501
020166 /106337
020170 /050140
020172 /103002
020174 /074137
020176 /050142
020200 /000240
020202 /077410
020204 /013737
020206 /050142
020210 /052000
020212 /005237
020214 /020210
020216 /005237
020220 /020210
020222 /077326
020224 /077233
020226 /012737
020230 /052000
020232 /020210
020234 /000000
*
```

ENCODER FOR THE CODE (15,8)

```
W
020124 /012700
020126 /051000
020130 /012701
020132 /052000
020134 /013702
020136 /050100
020140 /112037
020142 /050102
020144 /012703
020146 /020610
020150 /012704
020152 /000010
020154 /005037
020156 /050104
020160 /012305
020162 /106337
020164 /050102
020166 /103002
020170 /074537
020172 /050104
020174 /077407
020176 /013721
020200 /050104
020202 /077222
020204 /000240
*
```

APPENDIX E

NOISE MIXING SEQUENCE FOR THE CODE (15,4)

```
W
020242 /012700
020244 /052000
020246 /012701
020250 /032000
020252 /013702
020254 /050100
020256 /063702
020260 /050100
020262 /022027
020264 /104656
020266 /001005
020270 /022027
020272 /153610
020274 /001010
020276 /077207
020300 /000414
020302 /005740
020304 /011003
020306 /074311
020310 /012120
020312 /077215
020314 /000406
020316 /005740
020320 /005740
020322 /011003
020324 /074311
020326 /012120
020330 /077224
020332 /000000
*
```

NOISE MIXING SEQUENCE FOR THE CODE (15,8)

```
W
020206 /000240
020210 /000240
020212 /000240
020214 /000240
020216 /012700
020220 /052000
020222 /012701
020224 /032000
020226 /013702
020230 /050100
020232 /021027
020234 /106700
020236 /001402
020240 /012103
020242 /074310
020244 /005720
020246 /077207
020250 /000240
*
```

APPENDIX F

SYNDROME METHOD DECODER FOR THE CODE (15,4)

```
W
020334 /012700
020336 /052000
020340 /000240
020342 /000240
020344 /013737
020346 /050100
020350 /050102
020352 /012737
020354 /000002
020356 /050104
020360 /012702
020362 /000017
020364 /012703
020366 /050210
020370 /005037
020372 /050142
020374 /011037
020376 /050140
020400 /012304
020402 /006337
020404 /050140
020406 /103002
020410 /074437
020412 /050142
020414 /077207
020416 /022737
020420 /000000
020422 /050142
020424 /001002
020426 /000137
020430 /021310
020432 /023737
020434 /050246
020436 /050142
020440 /001002
020442 /000137
020444 /021274
020446 /012737
020450 /004000
020452 /050150
020454 /012737
020456 /000000
020460 /050152
020462 /013701
020464 /050150
*
```

```
W
020466 /012703
020470 /000013
020472 /012705
020474 /050250
020476 /012704
020500 /050306
020502 /012737
020504 /000016
020506 /050154
020510 /023715
020512 /050142
020514 /001002
020516 /000137
020520 /021304
020522 /013737
020524 /050152
020526 /050156
020530 /074137
020532 /050156
020534 /011502
020536 /074237
020540 /050156
020542 /023737
020544 /050156
020546 /050142
020550 /001002
020552 /000137
020554 /021304
020556 /005725
020560 /005724
020562 /005337
020564 /050154
020566 /003350
020570 /006201
020572 /077341
020574 /005337
020576 /020470
020600 /001406
020602 /000240
020604 /006237
020606 /050150
020610 /006237
020612 /050152
020614 /000722
*
```

```
   W
020616 /000240
020620 /000167
020622 /000000
020624 /012737
020626 /002000
020630 /020450
020632 /012737
020634 /004000
020636 /020456
020640 /012737
020642 /000012
020644 /020470
020646 /012737
020650 /050260
020652 /020474
020654 /012737
020656 /050316
020660 /020500
020662 /012737
020664 /000012
020666 /020504
020670 /012737
020672 /000404
020674 /020510
020676 /062737
020700 /000064
020702 /020622
020704 /000137
020706 /020446
020710 /012737
020712 /001000
020714 /020450
020716 /012737
020720 /006000
020722 /020456
020724 /012737
020726 /000011
020730 /020470
020732 /012737
020734 /050274
020736 /020474
020740 /012737
020742 /050332
020744 /020500
020746 /012737
020750 /000004
020752 /020504
020754 /062737
020756 /000056
020760 /020622
020762 /000137
020764 /020446
*
```

```
   W
020766 /012737
020770 /000400
020772 /020450
020774 /012737
020776 /005000
021000 /020456
021002 /012737
021004 /000010
021006 /020470
021010 /062737
021012 /000034
021014 /020622
021016 /000137
021020 /020446
021022 /012737
021024 /000100
021026 /020450
021030 /012737
021032 /004400
021034 /020456
021036 /012737
021040 /000006
021042 /020470
021044 /062737
021046 /000034
021050 /020622
021052 /000137
021054 /020446
021056 /012737
021060 /000040
021062 /020450
021064 /012737
021066 /004200
021070 /020456
021072 /012737
021074 /000005
021076 /020470
021100 /062737
021102 /000034
021104 /020622
021106 /000137
021110 /020446
021112 /012737
021114 /000020
021116 /020450
021120 /012737
021122 /004100
021124 /020456
*
```

```
   W
021400 /005337
021402 /050104
021404 /001405
021406 /110337
021410 /050350
021412 /005720
021414 /000137
021416 /020360
021420 /106303
021422 /106303
021424 /106303
021426 /106303
021430 /012702
021432 /000004
021434 /106303
021436 /106137
021440 /050350
021442 /077204
021444 /000412
021446 /005720
021450 /113737
021452 /050350
021454 /056000
021456 /005237
021460 /021454
021462 /000240
021464 /000240
021466 /000137
021470 /020352
021472 /005337
021474 /050102
021476 /002363
021500 /012737
021502 /056000
021504 /021454
*
```

MINIMUM DISTANCE DECODER FOR THE CODE (15, 4)

```
W
020456 /000240
020460 /000240
020462 /000240
020464 /012700
020466 /052000
020470 /013737
020472 /050100
020474 /050102
020476 /063737
020500 /050100
020502 /050102
020504 /013701
020506 /050104
020510 /012703
020512 /054000
020514 /012704
020516 /000017
020520 /005037
020522 /050116
020524 /011005
020526 /074105
020530 /012702
020532 /000017
020534 /006305
020536 /005537
020540 /050116
020542 /077204
020544 /022737
020546 /000004
020550 /050116
020552 /002010
020554 /006301
020556 /103402
020560 /077421
020562 /000407
020564 /062701
020566 /000002
020570 /077425
020572 /000403
020574 /010123
020576 /005720
020600 /000403
020602 /012723
020604 /000000
020606 /005720
020610 /162737
020612 /000001
020614 /050102
020616 /003336
020620 /000240
```

```
W
020622 /000240
020624 /000240
020626 /000240
020630 /000240
020632 /013702
020634 /050100
020636 /012700
020640 /054000
020642 /012701
020644 /056000
020646 /012705
020650 /000002
020652 /005004
020654 /012046
020656 /012046
020660 /012703
020662 /000004
020664 /006316
020666 /006104
020670 /077303
020672 /005726
020674 /077507
020676 /110421
020700 /077216
020702 /000000
*
```

SYNDROME METHOD DECODER FOR THE CODE (15,8)

```
W
020252 /012700
020254 /052000
020256 /012701
020260 /056000
020262 /013702
020264 /050100
020266 /011037
020270 /050102
020272 /012703
020274 /020630
020276 /012704
020300 /000017
020302 /005037
020304 /050104
020306 /012305
020310 /006337
020312 /050102
020314 /103002
020316 /074537
020320 /050104
020322 /077407
020324 /005737
020326 /050104
020330 /001421
020332 /012703
020334 /020776
020336 /012704
020340 /020666
020342 /012737
020344 /000044
020346 /050106
020350 /023723
020352 /050104
020354 /001405
020356 /005724
020360 /005337
020362 /050106
020364 /002371
020366 /000402
020370 /011405
020372 /074510
020374 /000310
020376 /111021
020400 /005720
020402 /077247
020404 /000240
*
```

APPENDIX G

OUTPUT PROGRAM

```
W
021510 /012700
021512 /056000
021514 /013702
021516 /050100
021520 /105737
021522 /177564
021524 /100375
021526 /111037
021530 /177566
021532 /122720
021534 /000215
021536 /001402
021540 /077211
021542 /000420
021544 /012701
021546 /000012
021550 /105737
021552 /177564
021554 /100375
021556 /112737
021560 /000200
021562 /177566
021564 /077107
021566 /105737
021570 /177564
021572 /100375
021574 /112737
021576 /000212
021600 /177566
021602 /077232
021604 /000000
*
```

APPENDIX H

The simple run:

```
.GE CODE1.SAV

.EXAMPLE FOR THE CODE (15,4) CHANNEL NOISE
HAS INDEXING FACTOR 9 THIS IS LOW LEVEL
NOISE FOR THE CODE .CARRIAGE RETURN HAS
THE NOISE IMMUNITY .TESTING FOR NOISSY CHANNEL.
```

Decoding with minimum distance decoder ;

```
EXAMPLE FOR THE CODE (15,4)CHANNL NOIYE
HAS INDEXING FACTOR 9 THIS IS LOW LEVEL
NOISE JOR THE COTE .CARRIAGE RETURN HAS
THE NOISE IMMUNITY .TESTMNG FOR NOISSY JHANNEL.
```

Without decoding ; Without error correction

```
QHA
PLED_R$THEGGDM<15D4!S@NEHON_IQE
HAN INDEXINF!FEKUOQ 8!THIC MS LK_"DMVE
FOYRU GO"THU SOTE.CERRIAGE REJUVFH@SXUH
NOSM$IMIUA\Y.VE[PINV G_R JOIRQX BHENEL.
```

LIST OF REFERENCES

1. Shannon and Weaver, The Mathematical Theory of Communication.

2. W. Wesley Peterson and E. J. Weldon, Jr., Error Correction Codes.

3. Robert B. Ash, Information Theory.

4. S.G.S. Shiva, "Some Results on Binary Codes with Equivalent Words," IEEE Transactions on Information Theory, March 1969, Volume IT-15, Number 2.

5. Robert G. Gallager, Information Theory and Reliable Communication.

INITIAL DISTRIBUTION LIST

		No. Copies
1.	Defense Documentation Center Cameron Station Alexandria, Virginia 22314	2
2.	Library, Code 0212 Naval Postgraduate School Monterey, California 93940	2
3.	Professor George Marmont, Code 52Ma Department of Electrical Engineering Naval Postgraduate School Monterey, California 93940	10
4.	Associate Professor R. W. Burton, Code 52Zn Department of Electrical Engineering Naval Postgraduate School Monterey, California 93940	1
5.	Istanbul Teknik Universitesi Elektrik Fakültesi Gümüssuyu, Istanbul, Turkey	1
6.	Boğazici Teknik Universitesi Elektrik Fakültesi Bebek, Istanbul, Turkey	1
7.	Ortadoğu Teknik Universitesi Elektrik-Elekronik Fakultesi Ankara, Turkey	1
8.	Derince Sinif Okullari Komutanligi Derince Kocaeli, Turkey	1
9.	Deniz Harb Okulu Komutanliği Heybeliada, Istanbul, Turkey	1
10.	Kayhan Elitaş Naval Postgraduate School Monterey, California 93940	1
11.	Nizamettin Cetinyilmaz Emirsultan Gelir Sokak No. 17 Bursa, Turkey	2

Thesis
C3385
c.1

164095

Cetinyilmaz

Application of the computer for real time encoding and decoding of cyclic block codes.