

ESD RECORD COPY

ESD-TR-65-87

RETURN TO
SCIENTIFIC & TECHNICAL INFORMATION DIVISION
(ESTI), BUILDING 1211

TM-04113

COPY NR. _____ OF _____ COPIES

ERROR CONTROL FOR DIGITAL DATA
TRANSMISSION OVER TELEPHONE NETWORKS

TECHNICAL REPORT NO. ESD-TR-65-87

ESTI PROCESSED

MAY 1965

☐ DDC TAB ☐ PROJ OFFICER☐ ACCESSION MASTER FILE

D. R. O'Neil

DATE _____

ESN

ESTI CONTROL NR. **AL** 46220

Prepared for

CY NR. _____ OF _____ CYS

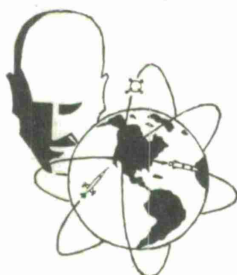
DEPUTY FOR COMMUNICATIONS SYSTEMS MANAGEMENT

ELECTRONIC SYSTEMS DIVISION

AIR FORCE SYSTEMS COMMAND

UNITED STATES AIR FORCE

L. G. Hanscom Field, Bedford, Massachusetts



Project No. 456.1

Prepared by

THE MITRE CORPORATION
Bedford, Massachusetts
Contract AF19(628)-2390

ADD 61 6678

Copies available at Office of Technical Services,
Department of Commerce.

Qualified requesters may obtain copies from DDC.
Orders will be expedited if placed through the librarian
or other person designated to request documents
from DDC.

When US Government drawings, specifications, or
other data are used for any purpose other than a
definitely related government procurement operation,
the government thereby incurs no responsibility
nor any obligation whatsoever; and the fact that the
government may have formulated, furnished, or in
any way supplied the said drawings, specifications,
or other data is not to be regarded by implication
or otherwise, as in any manner licensing the holder
or any other person or corporation, or conveying
any rights or permission to manufacture, use, or sell
any patented invention that may in any way be related
thereto.

Do not return this copy. Retain or destroy.

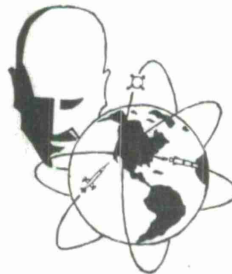
ERROR CONTROL FOR DIGITAL DATA
TRANSMISSION OVER TELEPHONE NETWORKS

TECHNICAL REPORT NO. ESD-TR-65-87

MAY 1965

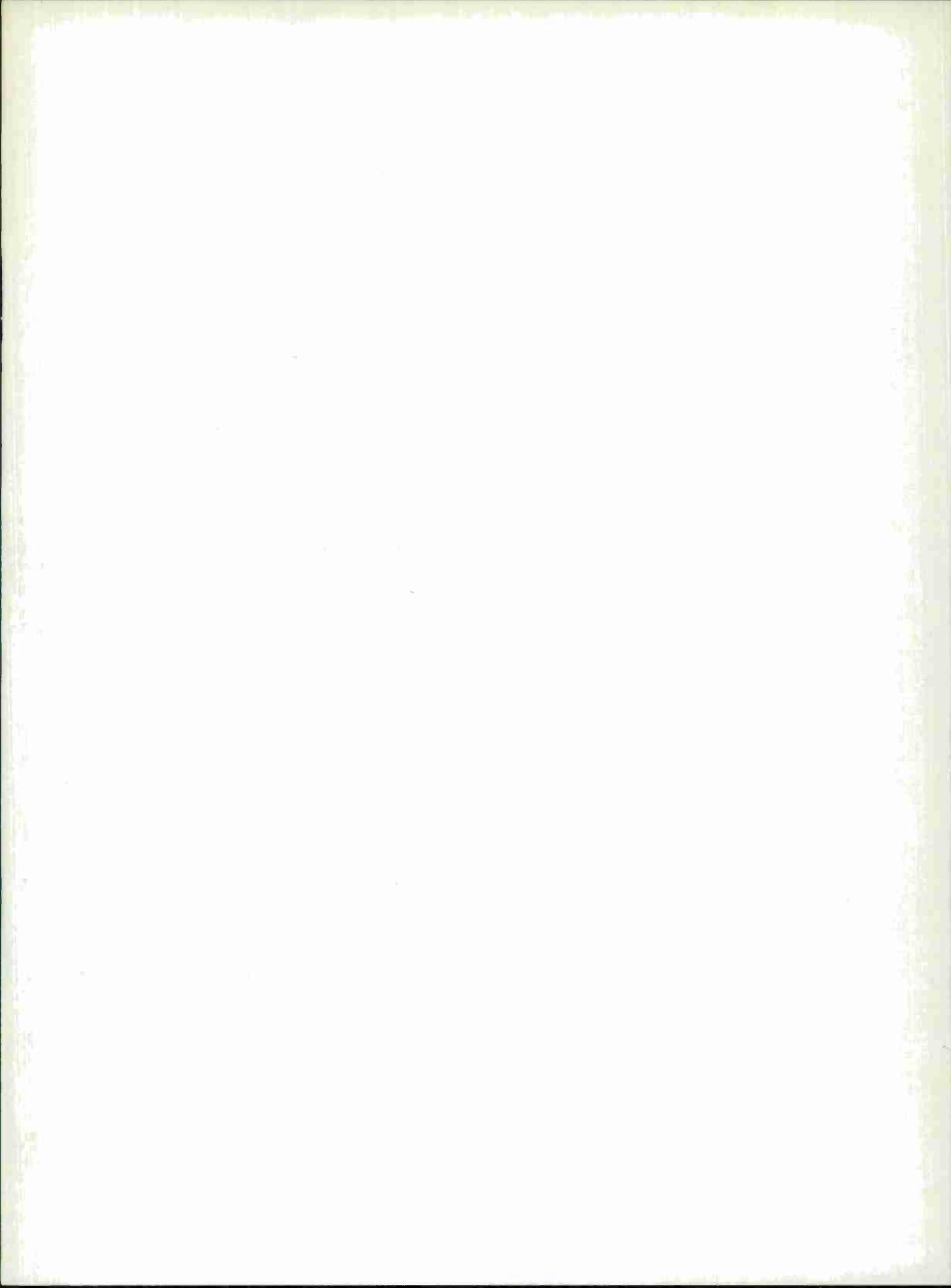
D. R. O'Neil

Prepared for
DEPUTY FOR COMMUNICATIONS SYSTEMS MANAGEMENT
ELECTRONIC SYSTEMS DIVISION
AIR FORCE SYSTEMS COMMAND
UNITED STATES AIR FORCE
L. G. Hanscom Field, Bedford, Massachusetts



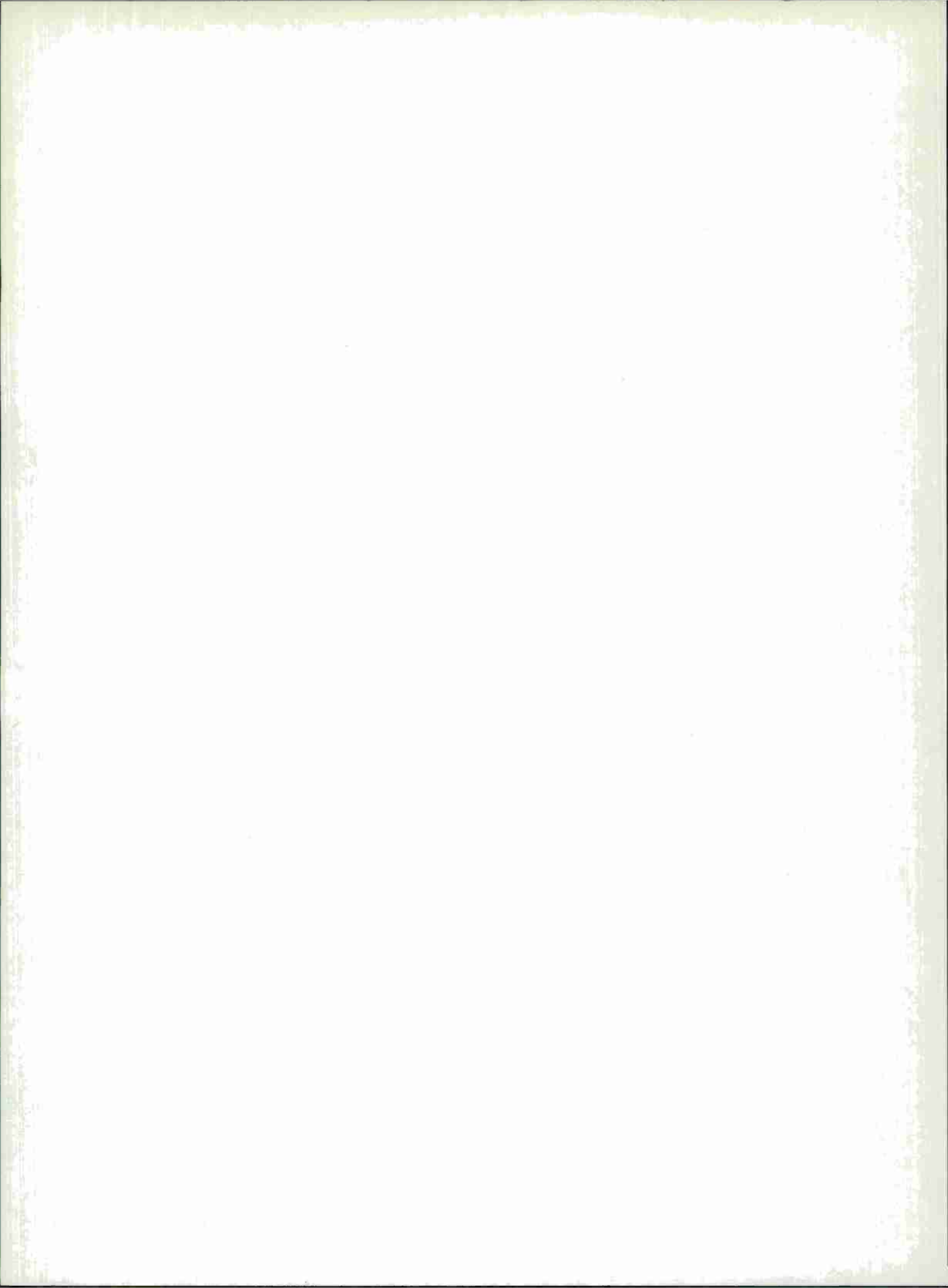
Project No. 456.1

Prepared by
THE MITRE CORPORATION
Bedford, Massachusetts
Contract AF19(628)-2390



FOREWORD

The author wishes to express his appreciation to the following members of The MITRE Corporation without whose assistance this report would not have been possible: Mr. Roger Walen, for his aid in providing a great deal of the information necessary in this study; Mr. Ronald Steeves, for his aid in analyzing the error statistics; and Mr. William Wood, for his many helpful suggestions, both during the study and in the preparation of this report.



ERROR CONTROL FOR DIGITAL DATA
TRANSMISSION OVER TELEPHONE NETWORKS

ABSTRACT

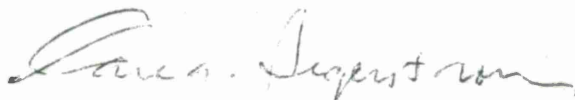
This report presents the results of a study of error control techniques applicable to binary digital data transmission over commercial telephone networks. The investigation consisted of a study of error control algorithms, a compilation of the error statistics for digital data on telephone lines, an evaluation of the performance of the error control techniques when applied to these error statistics, and a survey of the state-of-the-art in the hardware development of error control devices.

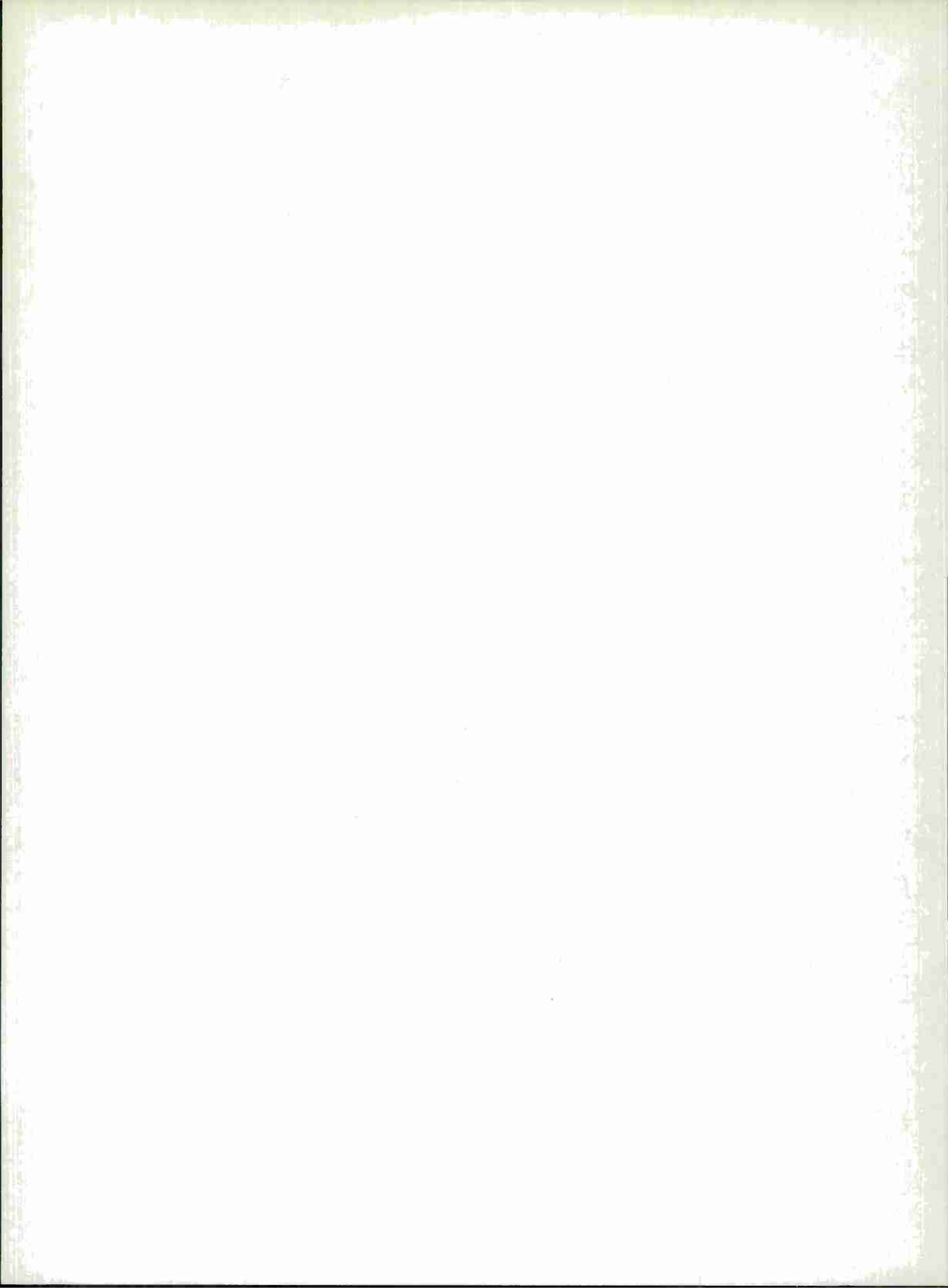
The main objectives of this study have been a determination of the performance to be expected from these error control algorithms when applied to the actual error statistics of common carrier voice bandwidth communication channels and the feasibility of implementing these techniques.

An additional purpose of this report is to provide communication engineers and managers with an introduction to the important considerations for selection and evaluation of error control techniques.

REVIEW AND APPROVAL

This technical documentary report has been reviewed and is approved.

for 
DONALD W. ROBERTS
Colonel, USAF
Deputy for Communications System



CONTENTS

	<u>Page</u>
SECTION I INTRODUCTION	1
SECTION II DESCRIPTION OF CODING ALGORITHMS	2
SIMPLE PARITY CHECK TECHNIQUES	2
Character Parity	3
Block Parity	4
Binary Count Parity Check	7
Other Parity Check Techniques	8
POLYNOMIAL (BLOCK CYCLIC) CODES	9
Hamming Codes	10
Bose-Chaudhuri Codes	14
Fire Codes	20
Other Block Codes	20
Implementation of the Block Cyclic Codes	21
CONVOLUTIONAL CODES	21
SECTION III TELEPHONE LINE ERROR STATISTICS	29
BUIC MODEM TEST STATISTICS	29
COMPARISON WITH OTHER ERROR STATISTICS	31
SECTION IV APPLICATION OF CODING TECHNIQUES	35
ERROR DETECTION AND RETRANSMISSION	35
Error Detection with Parity Check Techniques	36
Error Detection with Polynomial Codes	36

CONTENTS (Cont.)

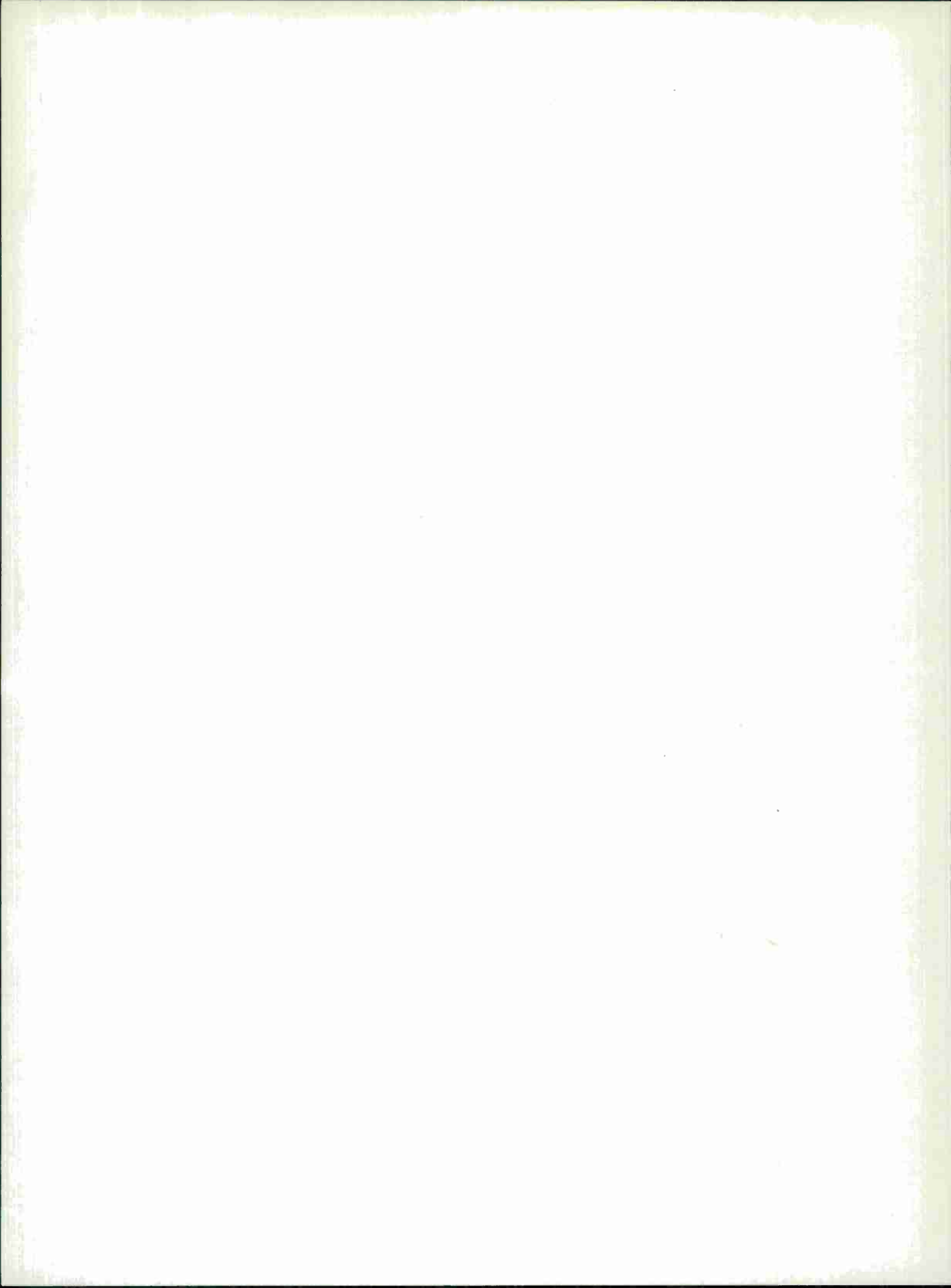
	<u>Page</u>
SECTION IV FORWARD ERROR CORRECTION	38
SUMMARY OF CODE PERFORMANCES	40
SECTION V IMPLEMENTATION OF CODING TECHNIQUES	41
ERROR DETECTING CODING DEVICES	41
FORWARD ERROR CORRECTING DEVICES	42
SECTION VI CONCLUSIONS	44
SECTION VII REFERENCES	46

LIST OF ILLUSTRATIONS

<u>Figure Number</u>		<u>Page</u>
1	Block Parity Checking Matrix	4
2	Failure of Block Parity	5
3	Convolution Encoder	23
4	Convolution Threshold Decoder	24
5	Undetected Bit Error Rates versus Efficiency for Some Cyclic and Block Codes When Applied to Statistics Obtained from BUIC Tests	39

LIST OF TABLES

<u>Table Number</u>		<u>Page</u>
I	Character Parity	3
II	Block Parity	6
III	Binary Count Parity	7
IV	Spiral Block Parity	9
V	Hamming Code	12
VI	Hamming Correction	13
VII	Block Cyclic Code	16
VIII	Bose-Chaudhuri Codes	18
IX	Convolutional Codes	26
X	Diffuse Convolutional Codes	28
XI	Summary of Test Statistics	31
XII	Block Error Statistics Including Dropouts	32
XIII	Burst Lengths of Blocks in Error	33
XIV	Error Correcting Code Test Results	42



SECTION I

INTRODUCTION

Despite major improvements in the quality of digital data modems and transmission facilities, the resulting data error rates are still not adequate for satisfying the stringent specifications which are being set forth for some communication systems. The purpose of this report is to describe the feasibility of reducing this error rate by the application of error control techniques to digital data transmission over telephone voice communication channels.

This study was undertaken to satisfy a need for a simplified explanation of error control techniques, and a need to know how this technology can be expected to perform against real error distributions.

The obvious reason for coding is to achieve a bit error rate which would be impossible otherwise. In addition, it is conceivable that an overall cost reduction could be realized by combining the use of error control with modems and channels that are not the best available or with reduced transmitter powers.

Section II of this report describes the basic coding theory and capabilities of several coding techniques which are most applicable to digital data transmission on the telephone system. Section III discusses the error statistics used for the evaluation of the various coding techniques. Section IV presents the results of the application of the coding algorithms to the collected data and compares the performances of these codes. Section V briefly discusses the state-of-the-art hardware development for implementing these codes.

This report is designed to serve as a guide to communication system engineers in determining which of the available error control techniques are best suited to a particular system, and in the writing of specifications to incorporate these techniques.

SECTION II

DESCRIPTION OF CODING ALGORITHMS

This section describes several error control techniques, all of which are for application to binary digital traffic. There are two basic philosophies to be considered: forward error correction and detection/correction with retransmission (commonly called ARQ). All these techniques require that additional bits be added to the information bits which make up the message. These additional bits are variously termed "parity," "check," or "redundant" bits. Basically, the only difference between the various coding techniques is the manner in which these parity check bits are generated and in their capabilities for handling errors. Thus, there are two criteria for evaluating these codes. The first is the capability to control (detect and/or correct) the errors. The second is the basic efficiency, or rate, of each code, which for the purpose of this report is defined as the ratio of the number of information bits to the total number of bits in a message. When calculating the efficiency of error detection and retransmission schemes, the number of bits rejected due to a block in error and the number of bits necessary for retransmission signaling must be taken into account. This efficiency, the ratio of total information bits delivered to total bits transmitted, is referred to as "throughput."

SIMPLE PARITY CHECK TECHNIQUES

Although some of the coding schemes discussed in this section could be utilized for forward error correction, their correcting ability is extremely limited. Therefore, these techniques find practical application only for error detection, with correction by retransmission.

Character Parity

The simplest parity check scheme involves dividing the binary data stream into groups and summing the digits modulo two without carry.¹ A parity check bit is then added to make the sum, including the check bit, either odd (odd parity) or even (even parity). As an example, consider the three 7-bit characters in Table I.

Table I

Character Parity				
	Parity		Parity	
	<u>Odd Parity</u>	<u>Check Bit</u>	<u>Even Parity</u>	<u>Check Bit</u>
m_1	1 1 0 1 1 0 1	0	1 1 0 1 1 0 1	1
m_2	0 0 0 0 1 1 0	1	0 0 0 0 1 1 0	0
m_3	1 0 1 0 1 0 1	1	1 0 1 0 1 0 1	0

The 7-bit information character is now transmitted as an 8-bit character. Both the odd parity scheme and the even parity scheme are capable of detecting an odd number of errors in the 8-bit character, but will not detect an even number of errors. This technique can, of course, be utilized for characters of any length. The efficiency, or rate, for the example shown is 7/8 or 0.875.

¹Modulo two addition without carry:

$0 + 0 = 0$	$0 + 1 = 1$
$1 + 0 = 0$	$1 + 1 = 0$

Block Parity

This technique, also referred to as horizontal and vertical parity checking or character and block parity checking, consists of arranging the data into an $(n \text{ by } m)$ rectangular matrix (n bits per character and m characters per block) and then applying the simple parity scheme described previously to each column and each row of the matrix resulting in an $(n + 1) \text{ by } (m + 1)$ matrix (see Figure 1). This coding scheme could now be used for correcting a single bit error within a block, since a single bit error can be uniquely located. However, this technique is utilized mainly for error detection. All odd numbers of errors within the block will be detected. In addition, most of the arrangements of the even number of errors will also be detected.

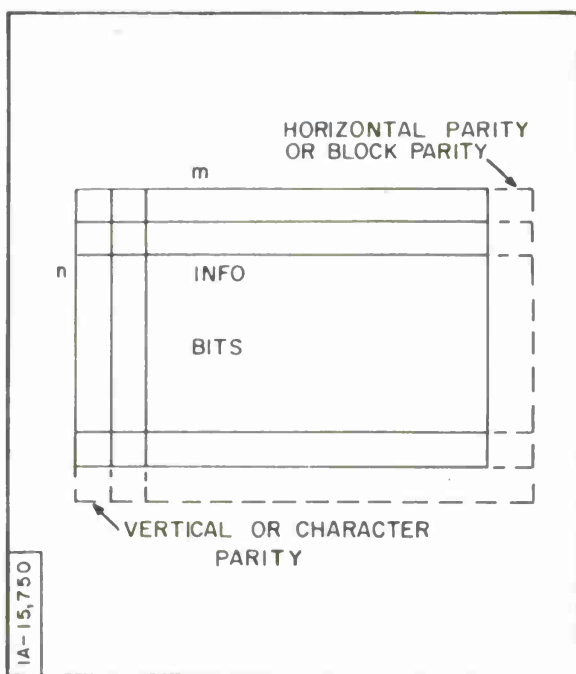


Figure 1. Block Parity Checking Matrix

The simplest example of where this technique would fail to detect is the occurrence of four errors arranged so that there are two errors in each of two rows and two errors in each of two columns as shown in Figure 2.

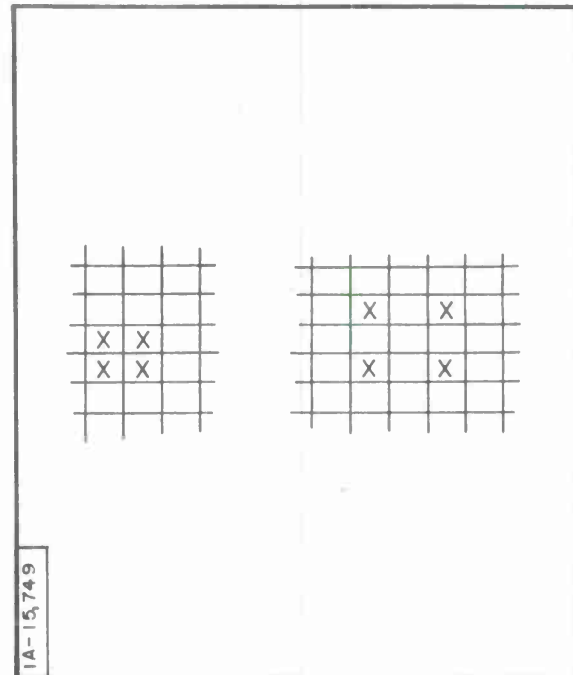


Figure 2. Failure of Block Parity

An excellent analysis of this type of error detection technique has been presented in the literature. Some of the important results of that analysis are included in this report. Table II presents the probability of accepting a block in error and the corresponding undetected bit error rate for blocks with character dimension $(n + 1) = 8$, as a function of block size $(m + 1)$ and initial bit error probability (p) .

The block parity technique can be used for a block of any desirable dimension. The results presented in Table II are for 8-bit characters because these are extensively used in two data transmission languages (Fieldata and ASC II). The eighth bit for each of these is a parity bit.

Table II
Block Parity

Block Size [(n + 1) x (m + 1)]	Basic Efficiency	Line Error Rate	Prob. of Accepting a Block in Error	Undetected Bit Error Rate
[8 x 20]	0.831	10^{-2}	5.6×10^{-5}	1.4×10^{-6}
		10^{-3}	5.35×10^{-9}	1.3×10^{-10}
		10^{-4}	5.19×10^{-13}	1.3×10^{-14}
		10^{-5}	5.32×10^{-17}	1.3×10^{-18}
[8 x 40]	0.853	10^{-2}	2.29×10^{-4}	2.9×10^{-6}
		10^{-3}	2.19×10^{-8}	2.7×10^{-10}
		10^{-4}	2.19×10^{-12}	2.7×10^{-14}
		10^{-5}	2.19×10^{-16}	2.7×10^{-18}
[8 x 60]	0.860	10^{-2}	5.29×10^{-4}	4.3×10^{-6}
		10^{-3}	4.97×10^{-8}	4.1×10^{-10}
		10^{-4}	4.97×10^{-12}	4.1×10^{-14}
		10^{-5}	4.96×10^{-18}	4.1×10^{-18}
[8 x 80]	0.864	10^{-2}	9.19×10^{-4}	5.7×10^{-6}
		10^{-3}	8.88×10^{-4}	5.5×10^{-10}
		10^{-4}	8.84×10^{-12}	5.5×10^{-14}
		10^{-5}	8.84×10^{-16}	5.5×10^{-18}
[8 x 100]	0.866	10^{-2}	1.44×10^{-3}	7.2×10^{-6}
		10^{-3}	1.39×10^{-7}	6.9×10^{-10}
		10^{-4}	1.39×10^{-11}	6.9×10^{-14}
		10^{-5}	1.33×10^{-15}	6.9×10^{-18}

The basic efficiency of the block parity scheme is

$$\frac{n \times m}{(n + 1) \times (m + 1)}$$

Binary Count Parity Check

This technique is similar to the block parity check in that data is again grouped into some convenient character size (n) and then a number of these characters (m) are arranged into a block as shown in Table III.

Table III
Binary Count Parity

		n = 5		
		1 0 1 1 0		
		1 1 0 1 0		
m = 6		0 1 1 0 1		
		1 0 1 1 0		
		1 0 1 1 1		
		0 1 1 1 1		
Binary Sum		1 1 1 1 0 0 1		
5-bit complement		0 0 1 1 0	(check) character	

A binary sum is then taken of the block, and a check character is formed by taking the complement of the first n lowest order bits in the sum. At this receiver, the summing process is repeated on the m characters, and this sum is added to the complement generated at the transmit end. In the absence of any errors in transmission, the 5-state counter would contain all ones (11111).

The presence of any zeros would indicate an error. This type of error detection would be particularly effective against errors which are limited to one type, either ones changing to zeros or zeros to ones, but not both.

Even limiting the consideration of errors to those of one type, it is obvious that because the check character is limited to only n bits, as few as two errors in the highest order bit column would cause this technique to fail. Two methods are available for obviating this difficulty. The first would be to allow the checking character to include all the bits of the binary sum. This would cause a decrease in code efficiency by increasing the redundancy. A better technique consists of changing the cycle of the n -stage binary counter from 2^n to 2^{n-1} . It can be shown.^[2] that this technique would require 2^{n-1} errors of a single type in any given column before it would fail to detect. The coding efficiency of this technique is

$$\frac{n \times m}{n \times (m + 1)}$$

Other Parity Check Techniques

Many variations of these coding techniques have been presented in the literature. These include "stacking" many matrices, each of which employ vertical and horizontal parity checking and generating parity checks in a third dimension. This results in an additional matrix which consists entirely of parity check bits.

Another variation is spiral parity checking, which is most easily explained by referring to Table IV, utilizing odd parity for both the character parity and the spiral block parity. The spiral block parity checks are formed by following the arrows. For the sake of clarity, only two of the six spiral parity bits are shown.

Table IV

Spiral Block Parity

Vertical Character Parity →	1	0	1	1	0	1	0	1	1	1	→	1	Spiral Block Parity Checks
	1	1	1	0	0	1	0	1	1	0	→	()	
	0	1	0	1	1	0	1	1	0	0	→	()	
	0	0	1	0	1	0	0	0	1	1	→	1	
	1	0	1	0	0	1	1	0	1	0	→	()	
	0	1	1	1	1	0	1	0	1	1	→	()	

This technique has the same error probabilities as those presented for the regular horizontal and vertical parity check technique. The two methods differ only in the patterns of error that they will detect.

POLYNOMIAL (BLOCK CYCLIC) CODES

This class of codes is characterized by a high degree of algebraic structure, in which each code can be mathematically described in terms of a generator polynomial. [3] Any group of binary digits can be expressed as a polynomial, with the binary digits as the coefficients of this polynomial. For example, the message 1 0 0 1 1 0 1 1 can be represented by the polynomial $M(x) = x^7 + x^4 + x^3 + x + 1$ (i. e., $1 \cdot x^7 + 0 \cdot x^6 + 0 \cdot x^5 + 1 \cdot x^4 + 1 \cdot x^3 + 0 \cdot x^2 + 1 \cdot x^1 + 1 \cdot x^0$). The polynomial codes are (n, k) block codes, where blocks of k information bits are mapped into blocks of n bits by appending $(n - k)$ redundant bits. Representing the block of k information bits by the polynomial $I(x)$, the encoding process consists of multiplying this polynomial by a factor x^{n-k} and dividing it by $P(x)$,

a generator polynomial. The remainder, $R(x)$, is then the polynomial representing the $(n - k)$ check bits which are appended to the information bits to form the message polynomial $M(x)$. It can be shown that the resulting message polynomial $M(x)$ is evenly divisible by $P(x)$. At the decoder, the received message is divided by $P(x)$. If a non-zero remainder exists, then errors have occurred. If the code is to be used for error correction, each correctable pattern of errors would need a different remainder which would be associated with the pattern. Complete discussions of the coding algorithm, including the factors to be considered in choosing the generator polynomial, $P(x)$, are available in the literature.^[3, 4, 5]

The term cyclic is derived from the fact that a cyclic shift of any code word is itself a code word. For example, if 1011010 is a code word, then 0101011 is also a code word. This cyclic, algebraic structure of these codes gives rise to a relatively simple method for implementing them. The generator polynomial associated with each code gives rise to a fundamental block size and fundamental values for the number of check bits within a block. However, it is possible to shorten any of these codes by reducing the total number of bits within a block while maintaining a constant number of check bits. These shortened cyclic codes are then $(n - i, k - i)$ block codes.

Not all of the codes to be described in this section satisfy the description given above for cyclic codes. However, these codes are equivalent in performance to the cyclic codes and like cyclic codes are easily implemented.

Hamming Codes

The Hamming codes are among those which are not truly polynomial codes and are not cyclic. However, they are included here because many of the concepts involved in cyclic coding can easily be explained by referring to these codes.

These codes are a class of single error correction (SEC) or single error correcting - double error detecting (SEC - DED) codes developed in 1950 by Dr. R. W. Hamming.^[6] He proved that a single bit error within a block of (n) binary bits consisting of (k) information bits and (m = n - k) check bits could be corrected if the following relationship held:

$$2^m \geq m + k + 1 \quad . \quad (1)$$

Consider, as an example, information messages consisting of four bits each (k = 4). The minimum value of m which satisfies the above relationship is m = 3. Therefore, there exists a (7, 4) code which is capable of correcting single errors. If the messages were 5 bits each, the minimum value of m would be 4, which results in a (9, 5) SEC code. However, 4 check bits are capable of correcting single errors in blocks up to 15-bits long, which corresponds to message lengths of 11 bits. When the equality of Equation (1) is met, the codes are referred to as optimum codes, and the block length is a maximum for a given number of check bits.

Using the (7, 4) code as an example, there are $2^k = 2^4 = 16$ possible messages with four information bits. These are shown in Table V. The coded form of these messages employing the Hamming algorithm is also shown, with the parity bits and information bits labeled accordingly. Because the intent of this report is to consider the practical capabilities of the coding techniques and not the coding algorithms themselves, no attempt is made to describe the logic equations used in deriving the individual parity bits. However, the reader interested in this theory, can refer to the literature.^[3, 6, 7]

There are now 16 possible 7-bit words that can be transmitted. The individual bit values of each of these words differ in at least three bit positions from the values of any other word. The number of bit positions in which two words

differ is defined as the Hamming distance or the distance between the words. As a corollary, the minimum distance which can exist between any two code words in a group is referred to as the minimum distance of this code. Thus, the minimum distance of the code considered is three, and this minimum distance for an (n, k) block code determines its error correction or error detection capabilities. In order for any block cyclic code to correct t or less errors in a block of size n , a minimum distance between code words of $(2t + 1)$ is necessary.^[3]

Table V
Hamming Code

	<u>I</u>	<u>I</u>	<u>I</u>	<u>I</u>	<u>P</u>	<u>P</u>	<u>I</u>	<u>P</u>	<u>I</u>	<u>I</u>	<u>I</u>
m_1	0	0	0	0	0	0	0	0	0	0	0
m_2	0	0	0	1	1	1	0	1	0	0	1
m_3	0	0	1	0	0	1	0	1	0	1	0
m_4	0	0	1	1	1	0	0	0	0	1	1
m_5	0	1	0	0	1	0	0	1	1	0	0
m_6	0	1	0	1	0	1	0	0	1	0	1
m_7	0	1	1	0	1	1	0	0	1	1	0
m_8	0	1	1	1	0	0	0	1	1	1	1
m_9	1	0	0	0	1	1	1	0	0	0	0
m_{10}	1	0	0	1	0	0	1	1	0	0	1
m_{11}	1	0	1	0	1	0	1	1	0	1	0
m_{12}	1	0	1	1	0	1	1	0	0	1	1
m_{13}	1	1	0	0	0	1	1	1	1	0	0
m_{14}	1	1	0	1	1	0	1	0	1	0	1
m_{15}	1	1	1	0	0	0	1	0	1	1	0
m_{16}	1	1	1	1	1	1	1	1	1	1	1
	(a)				(b)						

For error detection only, to detect D or less errors, the minimum distance is $(D + 1)$. To simultaneously correct t errors and detect $D > t$ errors, the minimum distance is $(t + D + 1)$.

Returning to the $(7, 4)$ SEC code in the example, because there are $\binom{7}{1} = 7$ ways for a single error to change a given word (where $\binom{n}{t}$ is the number of ways of getting t errors in a block of n bits), each of the 16 possible receiver choices must have $\binom{7}{1} = 7$ other words associated with it, which, if received, will be interpreted as that word (see Table VI). When two errors occur within a 7-bit word, the receiver can no longer properly correct because a double error in any word will be understood as a single error in some other word. For example, a double error which changes m_4 from 1000011 to 1010001 would be interpreted as a single error which changes m_{14} from 1010101 to 1010001.

Table VI

Hamming Correction

m_1	m_4	m_{14}	m_{16}
0000000	1000011	1010101	1111111
1000000	0000011	0010101	0111111
0100000	1100011	1110101	1011111
0010000	1010011	1000101	1101111
0001000	1001011	1011101	1110111
0000100	1000111	1010001	1111011
0000010	1000001	1010111	1111101
0000001	1000010	1010100	1111110

If the function of a Hamming code is changed from SEC to one of error detection only, it will be capable of detecting all single or double errors which occur in a block. In addition, since only 2^k possible words can be correct

words, and there are 2^n possible words that can be received, the probability of failing to detect any word in error may be approximated as:

$$\frac{2^k}{2^n} = \frac{1}{2^{n-k}} \quad (2)$$

The basic coding efficiency or rate of this code is $(n - k)/n$.

If the Hamming (7, 4) code was used for error detection only, it would detect all blocks that have either one or two errors and the probability of failing to detect a block in error would be $1/2^3 = 1/8 = 0.125$. The basic efficiency would be $4/7 = 0.571$.

If the Hamming (15, 11) code was used for error detection only, it would again detect all blocks with one or two errors, but now the probability of failing to detect a block with more than two errors would be $1/2^4 = 1/16 = 0.0625$. The basic efficiency would be $11/15 = 0.733$.

The Hamming (SEC-DED) codes are derived by adding an additional parity bit to the 7-bit code word developed for the previous example. This increases the minimum distance of the code to four.

Bose-Chaudhuri Codes

This class of codes is a generalization of the Hamming algorithm. They provide for correcting multiple errors in a block. These codes are polynomial (n, k) block, cyclic codes which correct t errors in a block of length $n = 2^m - 1$ with the number of check digits $(n - k) \leq mt$. When implemented for error correction, as in the Hamming codes, there are 2^k possible choices that the decoding process has for interpreting the n bit block received, and there are 2^n possible n bit sequences that could be received when the signal has been perturbed by noise. If the code is designed for t or less error correction, then there are

$$\binom{n}{1} + \binom{n}{2} + \dots + \binom{n}{t}$$

ways in which t or less errors may alter a given n bit code word. The receiver must assign to each of the 2^k possible words

$$\binom{n}{1} + \binom{n}{2} + \dots + \binom{n}{t}$$

other words, each of which differ in t or less bit positions from a given word and thus can be interpreted as that word.

This requires a minimum distance between code words of $(2t + 1)$, which can be realized with these codes. If the function of these codes is changed to error detection only, all blocks with $D = 2t$ or less errors will be detected, since to detect a block with D errors requires a minimum distance of $(D + 1)$. In addition, the probability of failing to detect blocks with more than D errors can again be approximated as $2^k/2^n = 1/2^{n-k}$.

Because of the cyclic nature of these codes, an additional error detecting ability is available which is not present in Hamming algorithm. It has been shown^[3] that in an (n, k) cyclic code, no burst of errors of length $(n - k)$ or less can change one code word into another. For the purpose of this report, the length of a burst will be defined as the total number of bits between, and including, the first error in a block and the last error in that block, where the bits between these errors may or may not be in error. This ability can be illustrated by referring to Table VII, which shows the same 16 information words used in the previous example, now encoded as a $(7, 4)$ block cyclic code with information and parity bits labeled accordingly. Insert a burst of length $7 - 4 = 3$ or less in any word and note that it does not change into one of the other words.

Table VII
Block Cyclic Code

	P	P	P	I	I	I	I
0 0 0 0	0	0	0	0	0	0	0
0 0 0 1	0	1	1	0	0	0	1
0 0 1 0	1	1	0	0	0	1	0
0 0 1 1	1	0	1	0	0	1	1
0 1 0 0	1	1	1	0	1	0	0
0 1 0 1	1	0	0	0	1	0	1
0 1 1 0	0	0	1	0	1	1	0
0 1 1 1	0	1	0	0	1	1	1
1 0 0 0	1	0	1	1	0	0	0
1 0 0 1	1	1	0	1	0	0	1
1 0 1 0	0	1	1	1	0	1	0
1 0 1 1	0	0	0	1	0	1	1
1 1 0 0	0	1	0	1	1	0	0
1 1 0 1	0	0	1	1	1	0	1
1 1 1 0	1	0	0	1	1	1	0
1 1 1 1	1	1	1	1	1	1	1

In terms of burst errors, the probability of not detecting blocks with bursts greater than $(n - k)$ is $2^{-(n - k - 1)}$ if the burst length (b) is equal to $(n - k + 1)$ and $2^{-(n - k)}$ if $b > (n - k + 1)$.

It is also possible to use these codes for burst error correction; however, for this case, the relationship between the burst length corrected, and the number of parity bits is not well defined. It has been shown that, in order for an (n, k) block code to correct bursts of length b or less, at least $2b$ parity check

symbols are necessary.^[3] In general, the burst length that these codes can correct lies in the range of $1/3$ to $1/2$ times $(n - k)$, the number of check bits.

These codes can also be modified, so that the full capability of error correction is not utilized. This excess capability can then be used for simultaneous error detection for errors in excess of those corrected. For example, a (31, 11) Bose-Chaudhuri code has a minimum distance of 11. It could, therefore, be implemented to correct up to five or less errors in the 31-bit block. It could also be used for error detection only, in which 10 or less errors could be detected for certain, with the probability of failing to detect blocks with more than 10 errors equal to

$$\frac{1}{2^{20}} \approx 1 \times 10^{-6}.$$

If, however, this code is used to correct only single and double errors, it can still detect all blocks with six or less errors. The probability of failing to detect a block with more than six errors would now be approximately

$$\begin{aligned} \frac{2^k \left[\binom{31}{0} + \binom{31}{1} + \binom{31}{2} \right]}{2^n} &= \frac{2^k (1 + 31 + 465)}{2^n} \\ &= \frac{497}{2^{n-k}} = \frac{497}{2^{20}} \approx 5 \times 10^{-4}. \end{aligned} \quad (3)$$

To provide an insight into the capabilities of the Bose-Chaudhuri codes, some are shown in Table VIII, with the minimum distance, the maximum number of errors corrected, the maximum number of errors detected for sure, the probability of failing to detect blocks with more than this amount of errors, the burst size that is detectable, the probability of failing to detect blocks with longer bursts and the basic efficiency, or rate, of each code.

Table VIII

Bose-Chaudhuri Codes

Code	d	t	D	Pn. d	b	Qn. d.	Basic Efficiency
(7, 4)	3	1	2	0.125	3	0.156	0.571
(15, 11)	3	1	2	0.0625	4	0.068	0.733
(15, 7)	5	2	4	0.39×10^{-2}	8	0.45×10^{-2}	0.467
(23, 12)	7	3	6	0.49×10^{-3}	11	0.53×10^{-3}	0.522
(31, 26)	3	1	2	0.031	5	0.032	0.839
(31, 21)	5	2	4	0.98×10^{-3}	10	0.10×10^{-2}	0.677
(31, 16)	7	3	6	0.31×10^{-4}	15	0.32×10^{-4}	0.516
(31, 11)	11	5	10	0.95×10^{-6}	20	0.10×10^{-5}	0.355
(63, 57)	3	1	2	0.016	6	0.016	0.905
(63, 51)	5	2	4	0.24×10^{-3}	12	0.25×10^{-3}	0.810
(63, 45)	7	3	6	0.38×10^{-5}	18	0.47×10^{-5}	0.714
(63, 39)	9	4	8	0.60×10^{-7}	24	0.61×10^{-7}	0.619
.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.
(127, 120)	3	1	2	0.78×10^{-2}	7	0.79×10^{-2}	0.945
(127, 113)	5	2	4	0.61×10^{-4}	14	0.62×10^{-4}	0.890
(127, 106)	7	3	6	0.48×10^{-6}	21	0.48×10^{-6}	0.835
(127, 99)	9	4	8	0.37×10^{-8}	28	0.38×10^{-8}	0.780
(127, 92)	11	5	10	0.29×10^{-10}	35	0.29×10^{-10}	0.724
.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.

where d = minimum distance

t = number of random errors correctable

D = number of random errors detectable

Pn. d. = Approx. prob. of not detecting blocks
with more than D errors

b = burst size detectable

Qn. d. = Approx. prob. of not detecting blocks
with longer bursts.

Table VIII

(Concluded)

Code	d	t	D	Pn. d.	b	Qn. d.	Basic Efficiency
(255, 247)	3	1	2	0.39×10^{-2}	8	0.39×10^{-2}	0.969
(255, 239)	5	2	4	0.15×10^{-4}	16	0.15×10^{-4}	0.937
(255, 231)	7	3	6	0.60×10^{-7}	24	0.60×10^{-7}	0.906
(255, 223)	9	4	8	0.23×10^{-9}	32	0.23×10^{-9}	0.875
(255, 215)	11	5	10	0.91×10^{-12}	40	0.91×10^{-12}	0.843
.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.
(511, 502)	3	1	2	0.20×10^{-2}	9	0.20×10^{-2}	0.982
(511, 493)	5	2	4	0.38×10^{-5}	18	0.38×10^{-5}	0.965
(511, 484)	7	3	6	0.75×10^{-8}	27	0.75×10^{-8}	0.947
(511, 475)	9	4	8	0.15×10^{-10}	36	0.15×10^{-10}	0.930
(511, 466)	11	5	10	0.28×10^{-13}	45	0.28×10^{-13}	0.912
(511, 457)	13	6	12	.	54	.	.
(511, 448)	15	7	14	.	63	.	.
.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.
(1023, 1013)	3	1	2	0.98×10^{-3}	10	0.98×10^{-3}	0.990
(1023, 1003)	5	2	4	0.95×10^{-6}	20	0.95×10^{-6}	0.980
(1023, 993)	7	3	6	0.93×10^{-9}	30	0.93×10^{-9}	0.971
(1023, 983)	9	4	8	0.93×10^{-12}	40	0.91×10^{-12}	0.961
.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.

Fire Codes [3, 8]

This group of block cyclic codes are burst error detecting and correcting codes discovered by P. Fire in 1959. In general, these codes have the following characteristics. With a block size $n = c(2^m - 1)$, a burst of size b can be corrected, and a burst of size $d > b$ can simultaneously be detected if $m \geq b$ and $c \geq (b + d - 1)$. The number of parity check bits necessary is $(c + m)$. For detection alone, these codes can detect any combination of two bursts if the length of the shorter burst is $\leq m$ and the sum of the lengths is $\leq (c + 1)$. They can also detect any single burst of length $\leq (c + m)$, the number of parity checks. For error correction only, to correct bursts of length b or less, both c and m must be $\geq b$.

As mentioned previously, these codes can be modified to fit any block size m , by shortening the code. For example, a Fire code can be generated which has $m = 5$ and $c = 9$. The length of this code is $n = 9(2^5 - 1) = 279$ bits, of which $(m + c) = 14$ are parity check bits and it is capable of correcting burst lengths up to 5 bits long. If, however, it was desired to encode the messages into blocks consisting of 200 information bits, this code could be shortened by simply omitting 65 of the 265 information bits that can be included in this code. This results in a (214, 200) block code capable of correcting bursts up to 5 bits long.

Other Block Codes

There are many other binary codes of the (n, k) block type which have been discussed in the technical literature. However, since all of the codes either closely approximate, or are special cases of those previously described, they are not discussed in any detail. Among these additional algorithms are the Reed-Muller codes [3] and burst correcting codes developed by both Abramson [9] and Reiger. [10] The capabilities of the Golay (23, 12) code have been included as a special case of the Bose-Chaudhuri codes listed in Table VIII.

Implementation of the Block Cyclic Codes

One of the most attractive features of these block cyclic codes is the ease with which they may be implemented. The encoding can be accomplished with a feedback shift register of $(n - k)$ stages and a small number of modulo two adders (i. e. , EXCLUSIVE-OR gates). The k information bits are shifted through this register and simultaneously transmitted through the channel. When the k information bits have been transmitted through the register, the $(n - k)$ stage register then contains the $(n - k)$ parity bits associated with the particular message transmitted. The contents of this register are then transmitted to complete the n bit block. For error detection, the decoder is identical to the encoder. The entire n bit message is shifted through the register. If no error has occurred, the register contains $(n - k)$ zeros. The presence of any one indicates the block had errors. For error correction, the same encoder is again used, but the decoder becomes much more complex, requiring a memory capability. For this and other reasons to be discussed in later sections, these codes are utilized mainly for error detection purposes.

CONVOLUTIONAL CODES

The main difference between convolutional codes (also referred to as recurrent codes) and those previously described is the lack of block structure. In this case, the parity check bits are generated and interleaved with the information bits in a continuous processing of the information bits through a shift register. Since a completely acceptable theoretical derivation of these codes has not been developed, the concepts involved in this type of coding will be explained by referring to examples. This coding technique can be applied for both random and burst error correction.

Two different techniques for decoding these recurrent codes exist. One of these is the sequential decoding technique developed by Wozencraft.^[11] In this technique, one information symbol is decoded at a time by comparing a sequence of received symbols to each sequence of two sets of possible transmitted sequences, one starting with a zero and the other starting with a one. The distance (as defined on page 12) between the received sequence and the possible transmitted sequences then determines the set in which the received sequence belongs, and thus determines the value of the first symbol of the received sequence. This symbol is then discarded from the received sequence and a new last symbol read in, and the process is repeated.

When the received sequence has been so corrupted by noise that a satisfactory decoding decision cannot be made, then a decoding failure can be noted and a retransmission can be initiated. For sequential decoding, it can be shown that the probability of an uncorrected error occurring approaches zero exponentially as the length of the computation increases linearly.^[11] A disadvantage of this scheme is the variation in the time required to decode a symbol, which depends on the amount of errors present.^[12]

A sequential decoding device has been developed, but, it is presently attractive only for experimental purposes due to its complexity.

The second decoding technique for these codes is threshold decoding, a bit-by-bit majority rule technique developed by Massey.^[13] Although this scheme is less effective than sequential techniques, ease of implementation has led to its use in the development of new devices. The examples of convolutional codes discussed in this report are concerned with this decoding technique.

A description of the operation of a convolutional encoding and decoding device based on a 0.5 rate code follows. Consider the convolutional encoder

illustrated in Figure 3. A continuous stream of information bits is fed into the shift register and simultaneously into the communication channel and to a parity check generator. This is a modulo two adder which sums the current information bit with the previous third, fourth and fifth information bits and puts out a parity check bit.

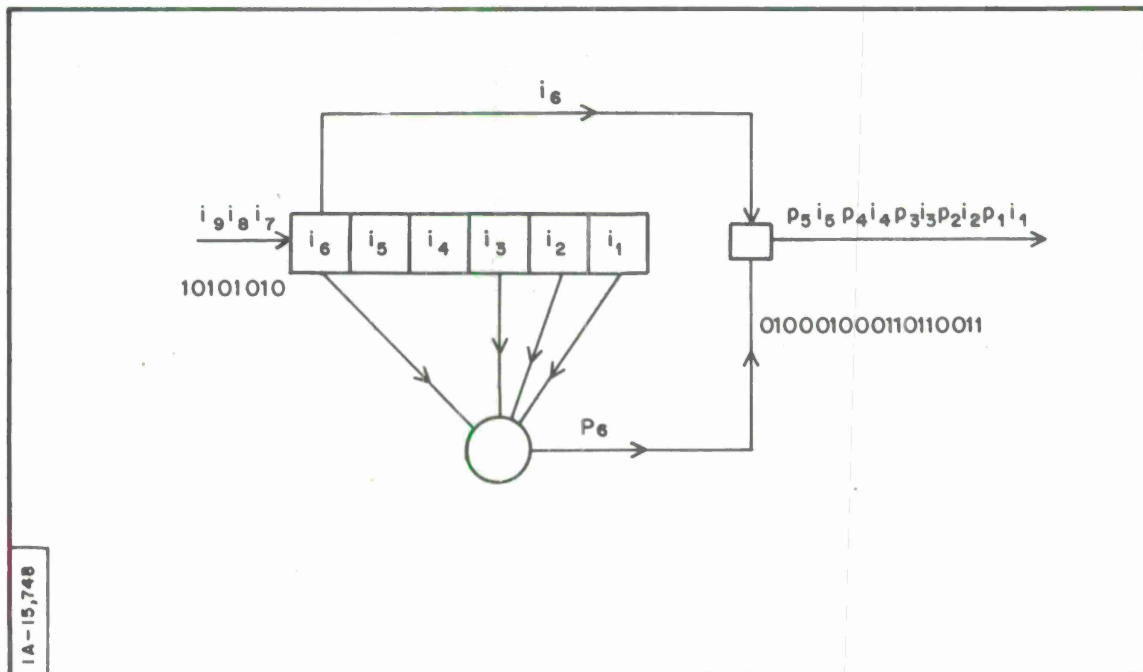


Figure 3. Convolutional Encoder

This parity check bit is then transmitted immediately after the associated information bit. At the instant illustrated in the diagram, i_6 is just entering the channel and p_6 is being determined. In the next bit interval, i_1 will be shifted out of the register and discarded. As an example of this technique, consider the 9-bit message 101010101 shifted into the encoder from right to

left. Assuming that the shift register initially contained all zeros, the encoded form of this message reading from right to left would be

i_9	i_8	i_7	i_6	i_5	i_4	i_3	i_2	i_1											
0	1	0	0	0	1	0	0	0	1	1	0	1	1	0	0	1	1		
p_9	p_8	p_7	p_6	p_5	p_4	p_3	p_2	p_1											

At the receiver, the data stream is separated into the original stream of information bits and a stream of parity bits. The information bits are fed into a shift register which is identical to that used for encoding. However, now the modulo two adder has, as an additional input, the received parity check bits. Thus, if there have been no errors in transmission, the output of this adder will be a parity check bit added to itself (i. e., zero). The presence of a one in the output of the adder represents an error in transmission. In order to correct errors by threshold decoding, the output of this adder is fed into another shift register as shown in Figure 4.

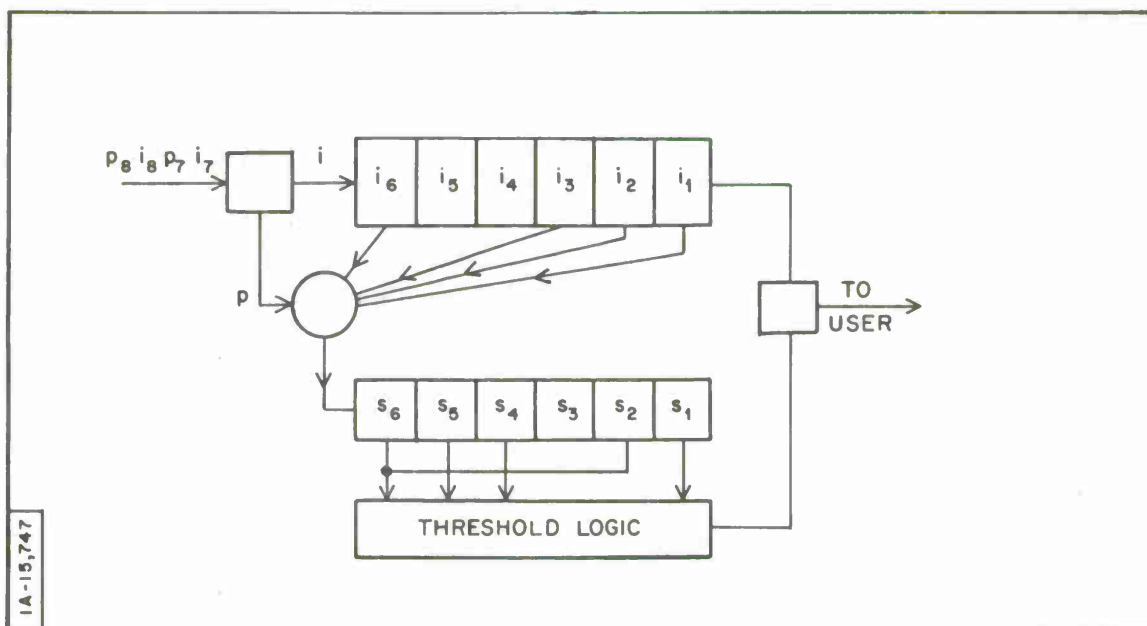


Figure 4. Convolutional Threshold Decoder

Because the input to this register will be zero except when errors occur, the contents of these registers will be the errors that have occurred. For example, consider the moment illustrated in Figure 4. The first six of the interleaved information and parity bits have already entered the decoder. The contents of the six stages of the shift registers are then given by the following relationships:

$$S_1 = E(i_1) + E(p_1),$$

$$S_4 = E(i_1) + E(i_4) + E(p_4),$$

$$S_5 = E(i_1) + E(i_2) + E(i_5) + E(p_5),$$

$$S_2 + S_6 = E(i_1) + E(i_3) + E(i_6) + E(p_2) + E(p_6).$$

where $E(i_j)$ or $E(p_j)$ is a one only if i_j or p_j was in error, and is zero otherwise.

The stages of the shift register examined by the threshold logic device are picked so that each logic equation formed will contain $E(i_j)$ once. At the same time every other $E(i_j)$ and $E(p_j)$ appears at most once in the entire set of equations. In the example under consideration, the decoder is attempting to determine the correct value of information bit i_1 . Assuming that there are two or less errors in the 11 bits which are contained in these equations, a decision on the value of $E(i_1)$ can be made by the following rule. If three or four of the sums shown equal one, then $E(i_1)$ will be considered a one and bit i_1 will be considered in error.

The logic circuit is implemented to correct i_1 as it leaves the register and simultaneously correct all the S stages which contain $E(i_1)$. This threshold decoding scheme can be shown to always give a correct decision as long as no more than two errors are present in any 12 consecutive bits (6 information bits and 6 parity bits). In addition, it can correct some of the error patterns

containing more than two errors. Since the decoder at any time is examining 6 information bits and 6 parity check bits, this code is said to have a constraint length of 12.

The principles illustrated in the previous example apply to codes with various constraint lengths and various rates. Some typical constraint lengths with the number of errors within each length that are guaranteed to be corrected are listed in Table IX.

Table IX
Convolutional Codes

Efficiency (Rate)	Constraint Length	Errors Corrected
0.5	12	2
0.5	24	3
0.5	44	4
0.5	68	5
0.5	104	6

Kohlenberg^[14] has devised a scheme for utilizing these codes for long burst error correction which he has labeled diffuse convolutional coding. This consists of taking a code such as just described and extending the constraint length over a long sequence of bits, only a few of which are utilized to compute the value of the parity check bit. For example, consider the rate one-half code which is capable of correcting bursts up to 168 bits long. This code requires an encoder consisting of 336 stages. For generating parity check bits, registers 1, 84, 168, and 336 are taped and added as in the previous example through the modulo two adder. The decoder again consists of an identical shift register for

regenerating the parity check bits. The shift register for storing the error bits is also 336 stages long. The logic equations used in this case are as follows:

$$S_1 = E(i_1) + E(p_1),$$

$$S_{169} = E(i_1) + E(i_{169}) + E(p_{169}),$$

$$S_{253} = E(i_1) + E(i_{85}) + E(i_{253}) + E(p_{253}),$$

$$S_{336} = E(i_1) + E(i_{84}) + E(i_{168}) + E(i_{336}) + E(p_{336}).$$

The very same threshold logic applies in this case. If either three or four of the S equations equal one, bit i_1 will be assumed to be in error. Again, this scheme will give a correct decision as long as no more than two of the bits in the above equations are in error. The constraint length of this code is 672 bits. A burst of errors less than 168 bits long (information bits plus parity bits) can affect at most only two of the bits associated with the four logic equations. Thus, this code is capable of correcting bursts up to 167 bits long, as long as the burst is preceded and followed by an error free guard space of 505 bits. However, it is possible to also correct bursts of this length if there are errors in the guard space, as long as the error pattern is such so that no more than two of the bits included in the logic equation at any time are in error.

These coding techniques can be extended to include various burst sizes and various coding rates. Some of these codes which have been developed are listed in Table X. The only limiting factor in the size of the burst that these codes can correct is the size of the registers necessary for implementation.

Table X

Diffuse Convolutional Codes

Code Rate	Burst Length Corrected	Constraint Length
1/2	168 n bits	672 n bits
2/3	85 n bits	~ 680 n bits
3/4	45 n bits	~ 540 n bits
where n = 1, 2, 3 - - -		

SECTION III

TELEPHONE LINE ERROR STATISTICS

In order to evaluate the operating performances of the various coding techniques available, it is of course necessary to consider the error statistics of the transmission channels. Although several mathematical models for error statistics on telephone circuits have been proposed,^[15, 16] they are still very limited in practical application. Therefore, the determination of coding performances still requires an analysis of the raw error statistics of the telephone channel.

BUIC MODEM TEST STATISTICS

During the past year, a series of tests were conducted at The MITRE Corporation for evaluating a digital data modem for application to the SAGE and BUIC Air Defense System. As a result of these tests,^[17] a large amount of error statistics for digital data transmission on the switched telephone networks were collected. Some of the details of these tests are briefly discussed.

The statistics were collected for a 2600 bit-per-second, 4-phase data modem over looped line circuits from Bedford, Mass., to various cities throughout the United States and back. Only one data input channel was utilized, so that the effective data rate was 1300 bits per second. Bit-by-bit error statistics were collected using the Lincoln Laboratory ADDER. In addition, chart recordings were made for correlating data errors to variations in the line signal, and error totals on a per minute basis were also recorded.

The overall error rate excluding signal dropouts was 1×10^{-5} . However, approximately 20 percent of the calls experienced some degree of loss of service

and the errors that occurred during these periods accounted for over 90 percent of the total number of errors recorded in these tests. The major cause of these dropouts is believed to be fades in the microwave carrier sections of the telephone network. It seems reasonable to assume that most long-haul telephone circuits will utilize microwave links and therefore the effect of the fades should be taken into account when evaluating coding techniques.

By including the errors occurring during these dropouts, the overall bit error rate was approximately 5×10^{-4} . This includes the measured error statistics for dropouts up to 83 seconds in length. In over 300 hours of data transmission studied, there were 12 occasions where the error bursts were greater than 20 seconds in duration. Removing these errors from the statistics results in an overall bit error rate of about 2×10^{-4} .

For the purpose of evaluating code performances, a representative sample of calls was selected for further processing. There were 69 calls of about 45 minutes each selected for this sample. The selection of calls to be used was somewhat arbitrary and was based on the following factors: availability of bit-by-bit error distributions from the ADDER program, and inclusion of error bursts of lengths ranging from 140 milliseconds (184 bits) to 1.7 seconds (2200 bits). Table XI presents the statistics for the complete test program and for the test calls used in this analysis. The averages for the calls used are better than the average for the complete program. Because the lines tested were looped to various cities throughout the country, they were, on the average, at least twice as long as the average long-haul communication link. This, plus the fact that the tested lines received no special treatment that might lead to improved data transmission, makes the choice of these improved statistics closer to what seems reasonable for an actual long-haul data circuit.

Table XI
Summary of Test Statistics

	All Calls	Study
Percentage of Calls With Dropouts	20	$\frac{11}{69} \approx 16$
Total Error Rate With Dropouts	2×10^{-4}	5.4×10^{-5}
Without Dropouts	1.1×10^{-5}	0.68×10^{-5}
Ratio of Dropout Errors to Total Errors	0.925	0.874

The raw data for the test calls was then grouped into blocks corresponding to some of the basic block sizes for the cyclic codes listed in Table VIII and the errors in each block were tabulated. These results are presented in Table XII and discussed in Section IV. Some of these same blocks were tabulated with respect to burst size within blocks in error (see Table XIII).

Because of limited storage in ADDER, it was not possible to obtain a bit-by-bit error distribution for bursts of errors greater than 64 bits. However, by examining the minute counters, and the chart recordings, a bit-by-bit error distribution could be synthesized which is believed to be very accurate.

COMPARISON WITH OTHER ERROR STATISTICS

To obtain an independent check on the results of this study, a literature survey was performed to gather additional information on telephone line error

Table XII
Block Error Statistics Including Dropouts

Total Bits: 2.51 x 10 ⁸																									
Total Bit Errors: 13,629																									
Bit Error Rate: 5.4 x 10 ⁻⁵																									
Block Size	Total Blocks	Total Blocks with Errors	Block Error Rate	Number of Blocks With This Many Errors																					
				1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	>20	
15	1.67 x 10 ⁷	2062	1.2 x 10 ⁻⁴	401	120	76	92	41	47	41	28	775	436	3	1	0	1								
23	1.03 x 10 ⁷	1545	1.5 x 10 ⁻⁴	363	102	55	60	31	31	40	28	21	9	9	9	225	244	172	79	66	0	1			
31	8.09 x 10 ⁶	1314	1.6 x 10 ⁻⁴	345	106	57	51	30	29	22	20	15	5	8	26	8	6	4	7	2	4	375	193	1	
63	3.99 x 10 ⁶	928	2.3 x 10 ⁻⁴	307	93	46	42	20	24	28	15	11	3	2	3	5	7	4	5	0	3	2	3	305	
127	1.98 x 10 ⁶	718	3.6 x 10 ⁻⁴	277	85	36	36	23	13	23	17	11	5	1	7	1	1	4	6	2	0	1		169	
255	9.85 x 10 ⁵	608	6.2 x 10 ⁻⁴	233	86	42	33	26	16	19	21	5	4	1	7	0	1	3	1	6	0	1	1	102	
511	4.92 x 10 ⁵	533	1.1 x 10 ⁻³	214	78	38	29	23	23	17	17	9	4	2	6	2	1	2	1	1	0	0	0	66	
Total Bits: 2.54 x 10 ⁻⁹																									
Total Bit Errors: 25,000																									
Bit Error Rate: 1 x 10 ⁻⁵																									
Block Error Statistics From Reference 20																									
127	2 x 10 ⁷	2118	1.1 x 10 ⁻⁴	402	297	169	106	85	85	85	148	42	21	30	30	30	30	30	26	26	26	26	26	398	

33

[illegible]

statistics. There have been a number of studies conducted to measure the error rates for digital data transmission on standard telephone circuits.^[17, 18, 19, 20]

These tests have utilized a number of different modems and have been conducted on a variety of different circuits ranging from wire and cable links of slightly over 100 miles to circuits using microwave links with a length of over 4000 miles. All of these reports indicate that the characteristics of the telephone circuits for data transmission can be generalized as consisting of relatively long periods of good transmission capabilities with bit error rates of 1×10^{-5} or better, separated by periods of extremely high error rates, the frequency and duration of which are a function of the circuit length and type of transmission facility. In reference 20, over 500 hours of data collected on a 1300-bits per second modem on both microwave and K-carrier telephone lines is presented in block sizes corresponding to three of the Bose-Chaudhuri basic block lengths. The results of this study for a block size of 127 are included in Tables XII and XIII. The values listed are only approximate, because they were extrapolated from distribution curves.

Based on comparison with these statistics, the statistics analyzed in Section II are believed to be a good approximation of what can be expected on 1000-to 3000-mile digital data links over standard telephone cable and microwave circuits.

SECTION IV

APPLICATION OF CODING TECHNIQUES

As mentioned in Section II, there are two basic functions that may be performed in order to overcome the effects of errors in digital data transmission, forward error correction and error detection with correction via retransmission. Various coding techniques for performing these functions have been described. It is now necessary to evaluate these techniques when they are applied to actual error patterns that have been found on telephone circuits.

ERROR DETECTION AND RETRANSMISSION

This type of correction scheme has as a basic requirement a two-way communicating ability between the data link terminals. In addition, it requires that the transmitter store every group of data that it transmits until it is sure that the data was received correctly.

There are various schemes which have been developed to perform the retransmission function of the error control. These range from systems which send either an accept or reject signal back to the transmitter after receiving each block, to systems which notify the receiver only when a block has been detected in error and only the errored block has to be transmitted. For the purpose of evaluating the effects of errors on the throughput (defined on page 2) of these codes, an idealized system is assumed where only the blocks in error are retransmitted and the total bits of any block detected in error is considered as redundant. In actual practice, the throughput of any of these techniques is less than those determined here because of such factors as signaling and synchronization bits and delay times in the communication link.

Error Detection with Parity Check Techniques

By referring to Table XII for block sizes of 15, 23, and 31, and noting that a simple single parity bit technique would detect all blocks with an odd number of errors, it is seen that about 65 percent of the blocks (i. e., characters) in error are detected, with the errors in these blocks accounting for about 60 percent of the total errors. This is about 2.5:1 reduction in the undetected errors. The basic efficiency of these three examples is 0.933, 0.957 and 0.968 respectively. Because the block error rate for each of these three blocks is about 1×10^{-4} , the effect on efficiency of the detected blocks in error which are discarded is negligible, (i. e., the throughput $\approx$ the efficiency).

The performance of the horizontal and vertical block parity techniques when applied to these statistics can be estimated by referring to Table II. For the bit error rate of 5.4×10^{-5} , the corresponding undetected block error rates for block sizes of 160, 320, 480, 640 and 800 are approximately 2.8×10^{-10} , 6×10^{-10} , 8.6×10^{-10} , 1.2×10^{-9} and 1.5×10^{-9} , respectively. Here again, since the block error rate varies from 1×10^{-4} to 1×10^{-3} , the effect of discarded blocks on the efficiency is negligible. Thus, for the idealized retransmission scheme assumed, the throughput is approximately equal to the basic efficiency. This error detection technique is presently used in the AUTODIN system using eighty 8-bit characters per block. An undetected character error rate of 1×10^{-7} is quoted for this system.^[21] This appears to be a very conservative estimate, based on a bit error probability of about 5×10^{-3} . For the statistics of Table XII, the undetected character error rate would be about 5×10^{-10} .

Error Detection with Polynomial Codes

The cyclic codes offer the most effective techniques for error detection. As described earlier, these codes can be implemented to not only detect, for

certain, all blocks with a given amount of errors or a given error burst, but also to detect, with a high probability, all blocks in error. This latter capability is the most important.

Upon investigating the block error statistics compiled in Tables XII and XIII, it becomes evident that the blocks detected for certain represent less than an order of magnitude improvement in the block error rate and only a small improvement in the overall bit error rate. For example, consider the (127, 106) Bose-Chaudhuri code, which is capable of detecting blocks in error that have up to 6 errors, and blocks which have bursts of errors up to 21 bits long. From Table XII, this results in 470 of the 718 blocks in error (i. e. , 65 percent) being detected. From Table XIII, the burst lengths of 6 or less have been included in the 470 having 6 or less errors. The only additional blocks detected for certain because of the burst detecting ability of the code are those which have bursts of lengths 7 to 21 and have more than 6 errors. It is not evident from the tables, but for the data analyzed, this accounted for only 43 additional blocks detected for certain. Therefore, about 71 percent of the blocks in error are detected for certain. However, this accounts for only 1259 of the 13,629 bit errors or only about 9 percent of the total bit errors. For the data presented in Table XII, this same code would result in certain detection of about 54 percent of the blocks in error which accounts for only about 11 percent of the total bit errors. If it is assumed that half of the blocks with burst lengths between 7 and 21 have more than 6 errors, then about 65 percent of the blocks in error will be detected, accounting for about 18 percent of the total bit errors.

The real value of these codes lies in their ability to detect with a high probability, any block in error. Considering the previous example, of the 205 remaining blocks in error accounting for 91 percent of the bit errors in one data sample and the 753 blocks accounting for 82 percent of the total bit errors in the second sample, the probability that these blocks will not be detected is

approximately $1/2^{21} \approx 0.5 \times 10^{-6}$. Therefore, the undetected block error rate is reduced from 3.6×10^{-4} to 5.2×10^{-11} for the first case and from 1.1×10^{-4} to 9.4×10^{-12} for the second sample.

Figure 5 shows the undetected bit error rates with the corresponding efficiencies for some of the Bose-Chaudhuri codes when applied to the statistics obtained from the BUIC Tests. These error rates have been calculated by assuming only those blocks with $2t$ or less errors have been detected for certain, while those with more than the $2t$ errors are detected with a probability of $(1 - 1/2^{n-k})$. Also included are the expected undetected error rates and efficiencies of some of the horizontal and vertical parity check blocks.

There are many factors involved when deciding what block code is the most optimum. Most of the factors will involve the characteristics of the particular data link under consideration such as message formatting, types of traffic, desired error rate and desired efficiency. As in Figure 5, the longer the block, the higher the efficiency and the lower the undetected bit error rate. However, there may often be many other factors which preclude the use of a long block.

FORWARD ERROR CORRECTION

Although the block cyclic codes possess an error correcting ability, the effect of these codes in reducing the error rate when employed in this fashion is almost negligible. As was pointed out in Section IV, the errors detected for certain represent only a small percentage of the total errors. Since the error correcting ability of these codes is limited to less than half of their ability to detect for certain, the reduction of the overall error rate when using these codes for error correction is even less than the reduction obtained with the blocks detected for certain. For example, consider the statistics for the block

size of 127 presented in Table XII. Even if an efficiency of about 50 percent (i. e. , 127, 64) was used, only those blocks with 10 or less errors could be corrected. This results in only 73 percent of the blocks in error corrected and a reduction of the bit error rate of only 10 percent. Fontaine and Gallagher^[20] in discussing the error correcting ability of the cyclic codes when applied to the statistics they collected for a block size of 511, point out that in order to get an order of magnitude improvement in the block error rate at least 161 parity checks would be necessary. They further point out that this would result in less than a 2:1 reduction in the bit error rate, and that the amount of equipment required would prohibit their use except for very unusual circumstances. From the error correcting ability of these codes listed in Table VIII and the statistics of Table XII, the same conclusions hold for any of the possible block cyclic codes.

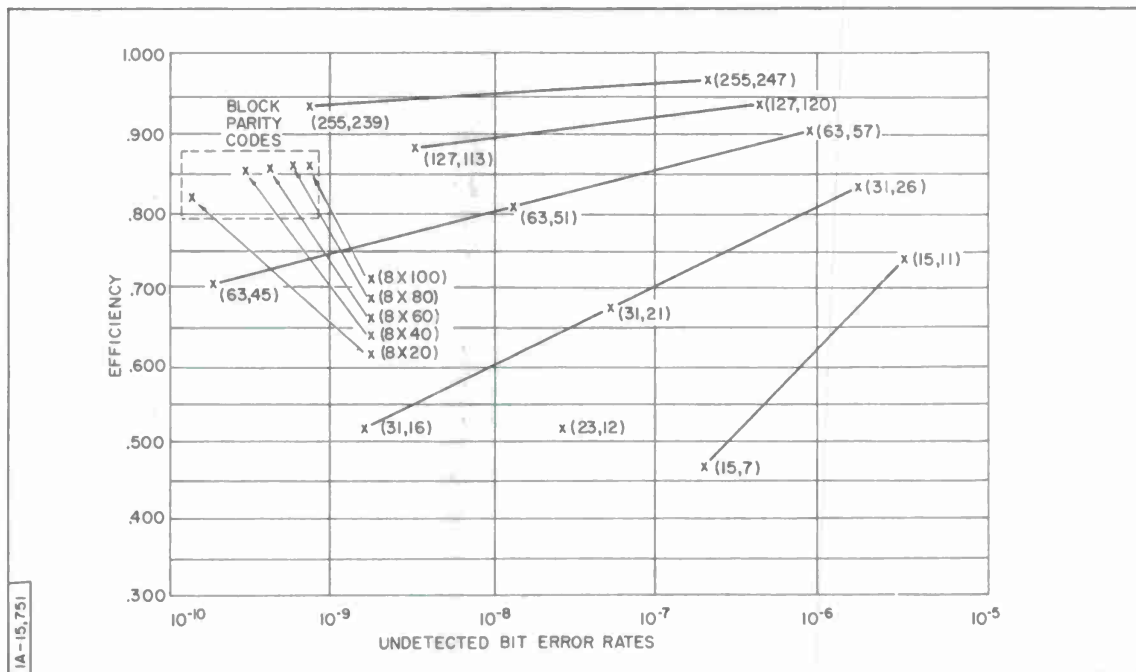


Figure 5. Undetected Bit Error Rates versus Efficiency for Some Cyclic and Block Codes When Applied to Statistics Obtained from BUIC Tests

The evaluation of the convolutional burst error correcting codes is not possible using the block error statistics of Table XII. This requires an analysis of the error statistics as a function of time. For example, in order to evaluate the error correcting ability of the code which is capable of correcting bursts of errors up to 168 bits long at 2400 bits per second, the raw error statistics must be analyzed to determine the relative occurrence of error bursts in excess of 70 milliseconds. For the error statistics collected on the switched network during the BUIC modem test, this occurred about once every 4 hours. When the considerations discussed in Section III are taken into account, it is estimated that the performance of these codes for a 1000- to 3000-mile digital data link over conventional telephone circuits will be an uncorrectable burst of errors about once every 24 hours.

SUMMARY OF CODE PERFORMANCES

It has been shown that for the statistics considered in this study, both the block parity codes and the block cyclic codes are capable of reducing bit error rates by three or four orders of magnitude with an efficiency of 80 percent or better, when used in conjunction with retransmission techniques. It has also been shown that these codes are not effective for forward error correction. The forward burst error correcting codes are capable of reducing the bit error rate by about two orders of magnitude with an efficiency of 50 percent.

SECTION V

IMPLEMENTATION OF CODING TECHNIQUES

Within the past few years, a number of devices for implementing these coding techniques have been developed. Most of this effort has been concerned with implementing the error detection codes with correction by retransmission.

ERROR DETECTING CODING DEVICES

As mentioned previously, the AUTODIN system utilizes the block parity technique in an ARQ system. Devices utilizing the block cyclic codes have also been developed and tested. Among these is a (31, 21) code that was tested at 2000-bits per second over the direct distance dialing network. [22] During the test, approximately 6.36×10^7 31-bit code words or 1.97×10^9 bits were transmitted. There were 29,731 blocks in error accounting for 62,002 bit errors for a bit error rate of 3.19×10^{-5} and a block error rate of 4.67×10^{-4} . Of the 29,731 blocks in error, only two went undetected.

Another device [23] which has been built utilizes the (255, 231) code. Laboratory tests simulating on-line conditions were performed in which over 3×10^6 blocks in error were correctly detected without any undetected erroneous words. Laboratory tests simulating high level random noise conditions were also performed. Over 5×10^6 blocks in error were detected without a single block in error going undetected. This same device was also tested for a total of 1515 hours on toll-grade telephone circuits in the New England area. During this time 6,707 blocks were detected in error with no undetected erroneous words.

A device for implementing a (1023, 993) cyclic code has been tested on commercial telephone circuits at a bit rate of 1200 bits per second. [24] The results of these tests are shown below in Table XIV.

Table XIV
Error Correcting Code Test Results

Loop	Blocks Transmitted	Blocks in Error	Undetected Blocks in Error
Fullerton, Calif. , to Los Angeles and return	10,000	180	0
Fullerton, Calif. , to St. Louis, Mo. , and return	10,000	250	0
Fullerton, Calif. , to New York City and return	20,000	400	0

An encoder which allows for automatically varying the redundancy of the blocks has also been developed. [25] This utilizes a fixed number of parity bits (i. e. , 20) and varies the block length from 40 bits (i. e. , rate = 0.5) to 260 bits (i. e. , rate = 0.92). No test results of this device have been made available.

FORWARD ERROR CORRECTING DEVICES

The only forward error correcting device which has been implemented and tested on telephone circuits is the diffuse convolutional code which corrects bursts

up to 168 bits long with a rate of 0.5. [26] A similar rate 0.5 device with a burst correcting capability of 1 second with a 4-second constraint length has also been implemented and tested. [27] However, these tests were performed on tropospheric circuits and are not discussed in this report. The tests on the device capable of correcting bursts up to 168 bits long were performed on a 107.6 mile telephone circuit at a transmission rate of 2400 bits per second. The tests were conducted for over 1137 hours during which time approximately 10^{10} bits were transmitted. There were four occasions when the device failed to correct errors. Two of these were complete line outages of more than 15 minutes. The remaining two bursts contained 406 and 449 errors respectively. Between these bursts, there were a total of 815 errors that were corrected. This device had an additional capability of detecting those times when it failed to correct. The results of this test have been summarized as follows: An uncorrectable burst of errors occurred on an average of once every 12 days and these bursts were detected. However, these tests were performed on a relatively short circuit which had an overall bit error rate of 1.63×10^{-7} which is at least two orders of magnitude better than what can be expected for 1000- to 3000-mile circuits.

SECTION VI

CONCLUSIONS

As a result of this study, it is evident that techniques exist for reducing the bit error rates for digital data transmission over commercial telephone circuits to any desired degree of accuracy and efficiency using error detection and retransmission schemes. Because these circuits presently have relatively good error rates, the retransmission blocks in error will not seriously affect the throughput of data. For those circumstances where retransmission of data is not practical, there are available forward error correcting devices which can be expected to reduce the bit error rate by about two orders of magnitude.

The error detection and retransmission schemes can reduce the undetected bit error rate of even the poorer grades of data transmission circuits many orders of magnitude. It is believed that the estimate presented in Section IV for the performance of the forward error correcting codes is somewhat optimistic and would apply only to special dedicated links using the best available equipment.

It is also conceivable, that since many of the longer bursts of errors could be detected by other means (i. e., monitoring the received signal power), this could be utilized in conjunction with forward error correction to reduce the number of retransmissions. In general, the reduction of retransmissions by this technique would not be significant if the cyclic codes were used for this purpose. It would, however, greatly reduce the number of retransmissions necessary if it was used in conjunction with the convolutional burst error correcting codes. However, since this requires that a retransmission ability be available for use at any time, and since the error detection schemes provide such a high undetected bit error rate, this would appear to be applicable to only a very few special cases.

Finally it is to be stressed that this study concerned itself with statistics for conventional telephone circuits of wire-line and microwave links and that the performance of these circuits is such as to make retransmission techniques highly efficient. In situations where performance is not as good (e.g., HF radio and tropo circuits) no conclusions should be drawn from this study.

D. R. O'Neil
D. R. O'Neil

SECTION VII

REFERENCES

1. Kuhn, T. G. "Retransmission Error Control," IEEE Transactions on Communication Systems, June, 1963.
2. Steeneck, R. "The EDAC Solution to the Problem of Error Control in Telecommunications," a paper presented at the Scott-St. Louis Chapter of AFCEA, November 1, 1963.
3. Peterson, W. W. Error-Correcting Codes, MIT Press and John Wiley and Sons, Inc., New York, 1961.
4. Peterson, W. W. and Brown, D. T. "Cyclic Codes for Error Detection," Proceedings of the IRE, January, 1961.
5. Bose, R. C. and Chaudhuri, D. K. "A Class of Error-Correcting Binary Group Codes," Information and Control, March, 1960.
6. Hamming, R. W. "Error-Detecting and Error Correcting Codes," Bell System Technical Journal, April, 1950.
7. Regenauer, B. J. "Error-Detection and Correction Techniques," MITRE Technical Memorandum 3187, October, 1961.
8. Fire, P. A Class of Multiple-Error-Correcting Binary Codes for Non-Independent Errors, Technical Report No. 55, Stanford Electronics Laboratory, Stanford University, Stanford, California, April, 1959, (AD 219307).
9. Abramson, N. M. "A Class of Systematic Codes for Non-Independent Errors," IRE Transactions of Information Theory, December, 1959.
10. Reiger, S. H. "Codes for the Correction of 'Clustered' Errors," IRE Transactions on Information Theory, March, 1960.
11. Wozencraft, J. M. and Reiffen, B. Sequential Decoding, MIT Press and John Wiley and Sons, Inc., 1961.

REFERENCES (Cont.)

12. Perry, K. E. and Wozencraft, J. M. "SECO: A Self Regulating Error Correcting Coder-Decoder," IRE Transactions on Information Theory, September, 1962.
13. Massey, J. L. Threshold Decoding, Doctoral thesis presented at the Massachusetts Institute of Technology, September, 1962.
14. Codex Corporation Diffuse Coding for Data Transmission, Technical Bulletin #4, Cambridge, Mass.
15. Berger, J. M. and Mandelbrot "A New Model for Error Clustering in Telephone Circuits," IBM Journal of Research and Development, July, 1963.
16. Sussman, S. M. "Analysis of the Pareto Model for Error Statistics on Telephone Circuits," IEEE Transactions on Communications Systems, June, 1963.
17. Kelly, J. P. "Test of Data Transmission via Switched Message Networks for SAGE and BUIC," MITRE Technical Memorandum 3380, March, 1964.
18. Alexander, A. A., Gryb, R. M., and Nast, D. W. "Capabilities of the Telephone Network for Data Transmission," Bell System Technical Journal, May, 1960.
19. Morris, R. "Further Analysis of Errors Reported in 'Capabilities of the Telephone Network for Data Transmission,' " Bell System Technical Journal, July, 1962.
20. Fontaine, A. B. and Gallagher, R. G. "Error Statistics and Coding for Binary Transmission over Telephone Circuits," Proceedings of the IRE, June, 1961.
21. Wollner, L. J., "AUTODIN - A General Description," ESD-TDR-64-632, May, 1965.
22. Townsend, R. L. and Watts, R. N. "Effectiveness of Error Control in Data Communication over the Switched Telephone Network," Digest of Technical Papers, 1964 International Symposium on Global Communications, June, 1964.

REFERENCES (Cont.)

23. Schmidt, W. G. Automatic Error-Detection System for Toll-Grade Telephone Circuit Data Transmission, Lincoln Laboratory Report 25G-5, November, 1961.
24. Beckman Instruments, Inc. Reliable Data Transmission, a special report made available by Beckman Instrument, Inc., Systems Division, Fullerton, Calif.
25. Honeywell, Variable Redundancy Error-Control System, Document T-10/64, April, 1964, a special report made available by the Honeywell Military Products Group, Ordnance Division, Seattle, Washington.
26. Codex Corporation, Report of TD 680.5 Error Correcting Coder Test on a Phone Line Data Transmission System, Cambridge, Mass.
27. North Atlantic Teletype Engineering Study, ITT Communications Systems, Inc., Paramus, N. J., June 1964.

DOCUMENT CONTROL DATA - R&D

(Security classification of title, body of abstract and indexing annotation must be entered when the overall report is classified)

1. ORIGINATING ACTIVITY (Corporate author)		2a. REPORT SECURITY CLASSIFICATION	
The MITRE Corporation Bedford, Massachusetts		Unclassified	
		2b. GROUP	
3. REPORT TITLE			
Error Control for Digital Data Transmission Over Telephone Network			
4. DESCRIPTIVE NOTES (Type of report and inclusive dates)			
N/A			
5. AUTHOR(S) (Last name, first name, initial)			
O'Neil, Daniel R.			
6. REPORT DATE	7a. TOTAL NO. OF PAGES	7b. NO. OF REFS	
May 1965	54	27	
8a. CONTRACT OR GRANT NO.	9a. ORIGINATOR'S REPORT NUMBER(S)		
AF19(628)-2390	ESD-TR-65-87		
b. PROJECT NO.	9b. OTHER REPORT NO(S) (Any other numbers that may be assigned this report)		
456.1	TM-04113		
c.			
d.			
10. AVAILABILITY/LIMITATION NOTICES			
Qualified requestors may obtain from DDC. DDC release to OTS authorized.			
11. SUPPLEMENTARY NOTES		12. SPONSORING MILITARY ACTIVITY	
		Deputy For Communications Systems Management Electronic Systems Division L. G. Hanscom Field, Bedford, Massachusetts	
13. ABSTRACT			
This report presents the results of a study of error control techniques applicable to binary digital data transmission over commercial telephone networks. The investigation consisted of a study of error control algorithms, a compilation of the error statistics for digital data on telephone lines, an evaluation of the performance of the error control techniques when applied to these error statistics, and a survey of the state-of-the-art in the hardware development of error control devices. The main objectives of this study have been a determination of the performance to be expected from these error control algorithms when applied to the actual error statistics of common carrier voice bandwidth communication channels and the feasibility of implementing these techniques. An additional purpose of this report is to provide communication engineers and managers with an introduction to the important considerations for selection and evaluation of error control techniques.			

14. KEY WORDS	LINK A		LINK B		LINK C	
	ROLE	WT	ROLE	WT	ROLE	WT
Computers and Data Systems Data Transmission Systems Error Control						

INSTRUCTIONS

1. **ORIGINATING ACTIVITY:** Enter the name and address of the contractor, subcontractor, grantee, Department of Defense activity or other organization (*corporate author*) issuing the report.

2a. **REPORT SECURITY CLASSIFICATION:** Enter the overall security classification of the report. Indicate whether "Restricted Data" is included. Marking is to be in accordance with appropriate security regulations.

2b. **GROUP:** Automatic downgrading is specified in DoD Directive 5200.10 and Armed Forces Industrial Manual. Enter the group number. Also, when applicable, show that optional markings have been used for Group 3 and Group 4 as authorized.

3. **REPORT TITLE:** Enter the complete report title in all capital letters. Titles in all cases should be unclassified. If a meaningful title cannot be selected without classification, show title classification in all capitals in parenthesis immediately following the title.

4. **DESCRIPTIVE NOTES:** If appropriate, enter the type of report, e.g., interim, progress, summary, annual, or final. Give the inclusive dates when a specific reporting period is covered.

5. **AUTHOR(S):** Enter the name(s) of author(s) as shown on or in the report. Enter last name, first name, middle initial. If military, show rank and branch of service. The name of the principal author is an absolute minimum requirement.

6. **REPORT DATE:** Enter the date of the report as day, month, year; or month, year. If more than one date appears on the report, use date of publication.

7a. **TOTAL NUMBER OF PAGES:** The total page count should follow normal pagination procedures, i.e., enter the number of pages containing information.

7b. **NUMBER OF REFERENCES:** Enter the total number of references cited in the report.

8a. **CONTRACT OR GRANT NUMBER:** If appropriate, enter the applicable number of the contract or grant under which the report was written.

8b, 8c, & 8d. **PROJECT NUMBER:** Enter the appropriate military department identification, such as project number, subproject number, system numbers, task number, etc.

9a. **ORIGINATOR'S REPORT NUMBER(S):** Enter the official report number by which the document will be identified and controlled by the originating activity. This number must be unique to this report.

9b. **OTHER REPORT NUMBER(S):** If the report has been assigned any other report numbers (*either by the originator or by the sponsor*), also enter this number(s).

10. **AVAILABILITY/LIMITATION NOTICES:** Enter any limitations on further dissemination of the report, other than those

imposed by security classification, using standard statements such as:

- (1) "Qualified requesters may obtain copies of this report from DDC."
- (2) "Foreign announcement and dissemination of this report by DDC is not authorized."
- (3) "U. S. Government agencies may obtain copies of this report directly from DDC. Other qualified DDC users shall request through _____."
- (4) "U. S. military agencies may obtain copies of this report directly from DDC. Other qualified users shall request through _____."
- (5) "All distribution of this report is controlled. Qualified DDC users shall request through _____."

If the report has been furnished to the Office of Technical Services, Department of Commerce, for sale to the public, indicate this fact and enter the price, if known.

11. **SUPPLEMENTARY NOTES:** Use for additional explanatory notes.

12. **SPONSORING MILITARY ACTIVITY:** Enter the name of the departmental project office or laboratory sponsoring (*paying for*) the research and development. Include address.

13. **ABSTRACT:** Enter an abstract giving a brief and factual summary of the document indicative of the report, even though it may also appear elsewhere in the body of the technical report. If additional space is required, a continuation sheet shall be attached.

It is highly desirable that the abstract of classified reports be unclassified. Each paragraph of the abstract shall end with an indication of the military security classification of the information in the paragraph, represented as (TS), (S), (C), or (U).

There is no limitation on the length of the abstract. However, the suggested length is from 150 to 225 words.

14. **KEY WORDS:** Key words are technically meaningful terms or short phrases that characterize a report and may be used as index entries for cataloging the report. Key words must be selected so that no security classification is required. Identifiers, such as equipment model designation, trade name, military project code name, geographic location, may be used as key words but will be followed by an indication of technical context. The assignment of links, rules, and weights is optional.

