1-1-1962

Some Useful Coding Techniques for Binary Communication Systems

J. C. Hancock
Purdue University

J. L. Holsinger
Purdue University

TR-EE 62-1
CONTRACT NO. AF 33(616)-8283

PURDUE UNIVERSITY
SCHOOL OF ELECTRICAL ENGINEERING

Some Useful Coding Techniques for Binary Communication Systems

J. C. Hancock, Principal Investigator

J. L. Holsinger

Communications Sciences Laboratory

January 2, 1962
Lafayette, Indiana

RESEARCH PROJECT PRF 2906
CONTRACT NO. AF 33(616)-8283
FOR
UNITED STATES AIR FORCE
AERONAUTICAL SYSTEMS DIVISION
WRIGHT-PATTERSON AIR FORCE BASE
OHIO

TR-EE 62-1

SOME USEFUL CODING TECHNIQUES
FOR
BINARY COMMUNICATION SYSTEMS

J. C. Hancock, Principal Investigator

J. L. Holsinger

SCHOOL OF ELECTRICAL ENGINEERING

PURDUE UNIVERSITY

Lafayette, Indiana

JANUARY 1962

UNITED STATES AIR FORCE

AERONAUTICAL SYSTEMS DIVISION

WRIGHT-PATTERSON AIR FORCE BASE

OHIO

PREFACE

On June 2, 1961 the Communication Sciences Laboratory, School of Electrical Engineering, Purdue University, was awarded USAF Contract No. 33(616)-8283. This contract is administered under the Aeronautical Systems Division, Wright-Patterson Air Force Base, Ohio by Mr. B. W. Russell.

During the initial phases of this program it seemed desirable to classify and unify the various coding techniques as they related to digital communication systems. This report represents the results of this brief study and represents a minor phase of the overall program.

ABSTRACT

An introduction to coding theory and a discussion of specific coding techniques are given as applied to digital communication systems.

The place of coding in a communication system is illustrated and the various approaches to coding are discussed. The information theory concepts required are presented along with the First and Second Fundamental Theorems of Shannon. The relation between Shannon's theorems and coding for the noisy and noiseless channel is discussed. For the noiseless channel the techniques of Shannon, Fano, Huffman, Gilbert-More, Karp and others are discussed. For the noisy channel the techniques of Hamming, Slepian, Elias, Cowell, Bose-Chaudhuri, Reed-Muller, Fire, and Wozencraft are presented. The relationships between the various codes are given and the advantages and disadvantages of each indicated. Numerous examples illustrating the use of the codes are given and areas of further research outlined.

TABLE OF CONTENTS

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

The classic work of Shannon (1,2), followed by that of Feinstein, Khinchin, Fano, Elias, and others laid the foundation for the modern field of Information Theory. In his original work Shannon proved an important theorem giving a promise of information transmission capabilities previously considered impossible. Loosely stated the theorm is as follows: If information* is transmitted over a noisy channel at a rate less than the channel capacity it is possible to encode the transmitted message in a manner such that it may be received with an arbitrarily small error rate.

Unfortunately Shannon's proof of this theoremis an existence proof, and does not give any information about how the encoding is to be accomplished in practice. The severity of this problem is readily apparent when it is realized that today, more then a decade after Shannon's original work, communication systems still do not operate at an information rate or an error rate even close to that theoretically possible.

At present a large amount of work is being done in an attempt to devise coding techniques that will allow this situation to be improved. Present results (3,4) indicate that within a few years it will be possible to operate a communication system at an information rate near the channel capacity with an error rate in the range of one error per day to possibly one error per several hundred years.

Because of this it is increasingly important that more people become aware of the basic concepts involved in coding.

1.1 Purpose and Structure of the Report

To a person working in the field of coding theory the names Golay, Hamming, Slepian, Shannon, Fano, Elias, Bose-Chaudhuri, Huffman, Gilbert-Moore, Wozencraft

---

* Here as in the next few paragraphs, terms such as information, information rate, capacity, coding, and others should be given their intuitive meaning until more precise definitions are given.

and Reed-Muller bring to mind several approaches to the solution of the coding problem. Unfortunately this same preponderance of names gives rise to considerable confusion in the minds of those not acquainted with the coding field and in addition points out the lack of a unified approach to the determination of optimum codes. It is the aim of this report to alleviate some of this confusion by presenting, with a minimum of proofs and analyses, the various better known coding schemes and showing how they are related. Thus, this report will be tutorial in nature and will, hopefully, provide a more unified picture of the field of coding than can presently be obtained from the literature.

As is always the case in a tutorial presentation, an assumption must be made concerning the background of the reader. In this report it will be assumed that the reader is familiar with the representation of discrete messages by binary numbers and with the basic concepts of discrete probability theory. References (5,6,7) provide an introduction to discrete probability theory for those lacking this background.

The construction of this report is briefly as follows: First, a discussion of a general communication system will be given, pointing out how coding fits into the complete system. Next, several concepts from Information Theory will be presented. This will involve precise definitions of terms such as information and channel capacity that are required for a study of coding theory. Thirdly, coding techniques for the noiseless binary channel will be discussed. Fourthly, the major portion of the report will discuss coding techniques for the noisy binary channel. In each case numerous examples will be given to illustrate the material discussed.

1.2 A General Communication System

A conventional communication system is illustrated in Fig. 1. Here an information source supplies a message to the transmitter. The transmitter converts the message to a form suitable for transmission over the channel. (For an RF channel this usually involves modulating some property of a carrier with the message signal.

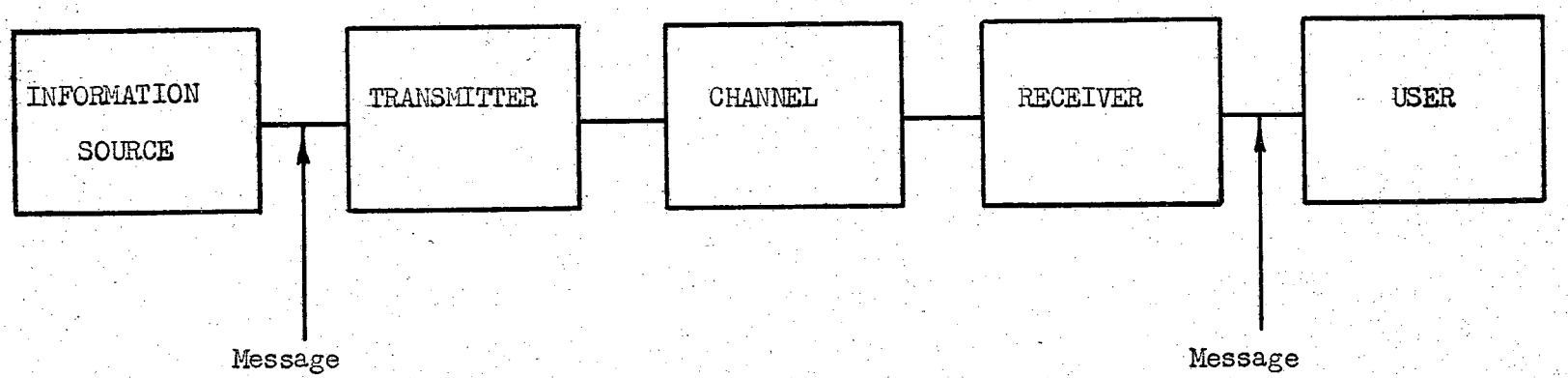


Fig. 1 - A Communication System

For a wire circuit it could involve nothing more than direct transmission of the message.) At the channel output there is, in general, a noisy, distorted replica of the transmitted message. The receiver operates on this signal converting it into a form suitable for the user. It is usually desired that the message supplied to the user be as nearly identical, in some sense, to the source message as is possible.

In general, a communication system may be either continuous, discrete or both. An example of a continuous system is a conventional AM system used for transmitting voice information. A teletype system represents a discrete system while a system for a transmitting TV signals by pulse-code-modulation (PCM) represents a combined discrete and continuous system. Fig. 2 illustrates the basic differences between the signals involved in each of these systems.

At present there has been essentially no work done in coding for continuous systems although Shannon's work applies to these as well as discrete systems. Because of this, this report will be concerned only with the discrete case and, due to its widespread use, only the binary form of this. Thus, the information source of Fig. 1 will now be considered to produce a sequence of 0's and 1's which represent the message to be transmitted. It is the function of the transmitter, channel, and receiver to accept these binary digits (binits), to reproduce them with as few errors as possible at the receiver, and to supply the results to the user.

The numerous details involved in designing a transmitter and receiver to operate with a specified channel and to produce a minimum error rate are of no interest to the coding theorist. For this reason the transmitter, channel, and receiver are usually considered as a "black box" which accepts 0's and 1's at its input and reproduces these, with an occasional error, at its output. This simplified model is illustrated in Fig. 3(a). In this illustration P_o is the transitional probability that a transmitted 0 will be received as a 1. For example if $P_o = 0.1$ this model implies that for every 100 0's transmitted there will be, on the average,

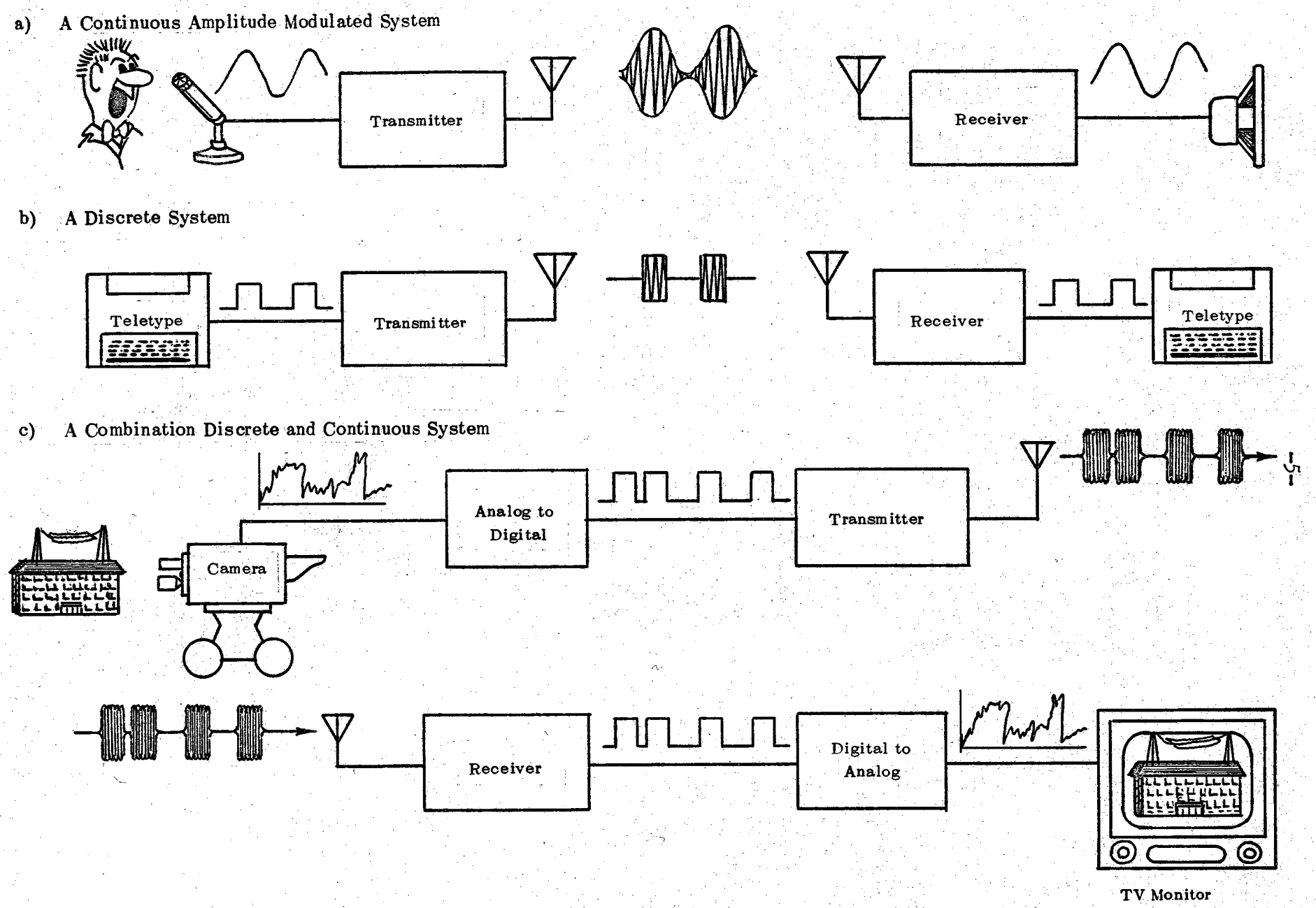
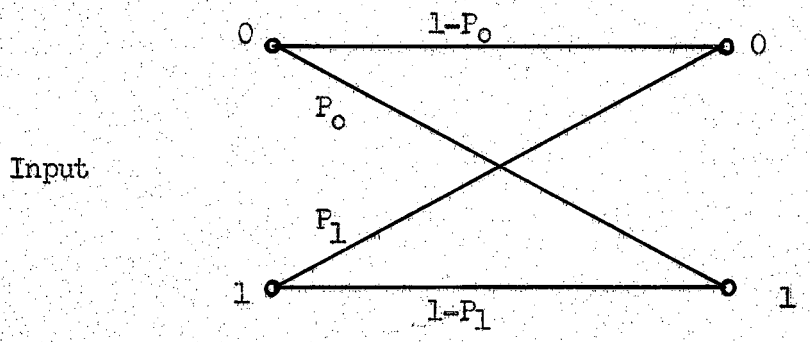


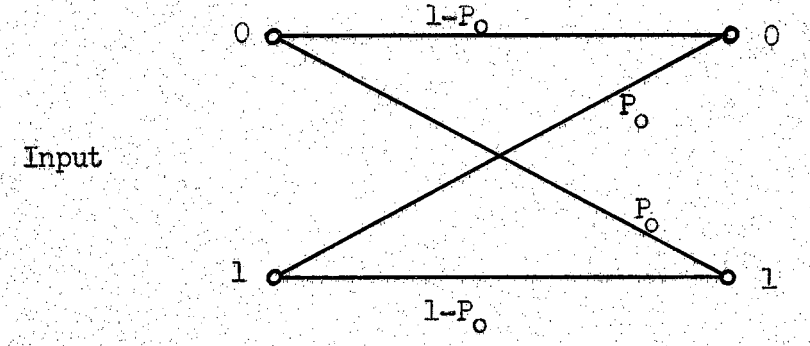
Fig. 2 – Types of Communication Systems

a)



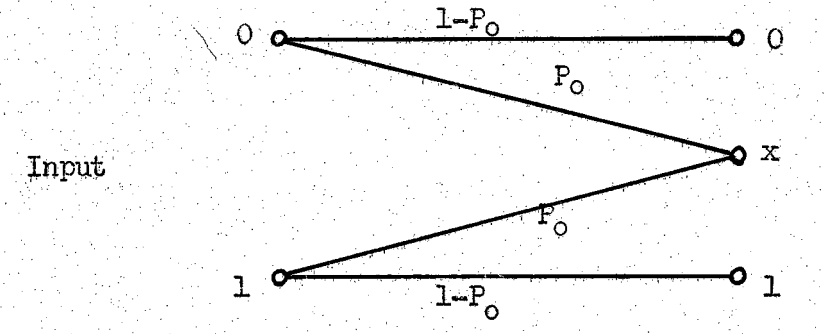
General Binary Channel

b)



Binary Symmetric Channel — BSC

c)



Binary Erasure Channel - BEC

Fig. 3 - Binary Channel Models

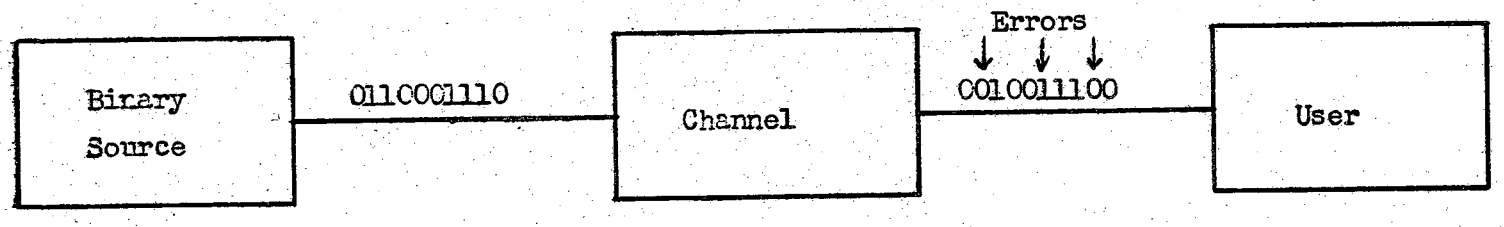
ten of these received erroneously as 1's. The remaining 0's will be received correctly. Similarily P_1 is the transitional probability that a transmitted 1 is received as a 0. These transitional probabilities provide all the information required to determine the effectiveness of a particular code. Because of this all further discussion in the report will be based on the model of Fig. 3(a) or on the closely related models of Fig. 3(b) and 3(c).

The binary communication system resulting from these simplifications is shown in Fig. 4(a). Here a source produces a sequence of binary digits that are transmitted over the channel. Due to noise on the channel some of these are received in error and thus represent erroneous information. In general this incorrect information can not be tolerated and some means of eliminating the errors must be found. This can be accomplished in one of the following three ways:

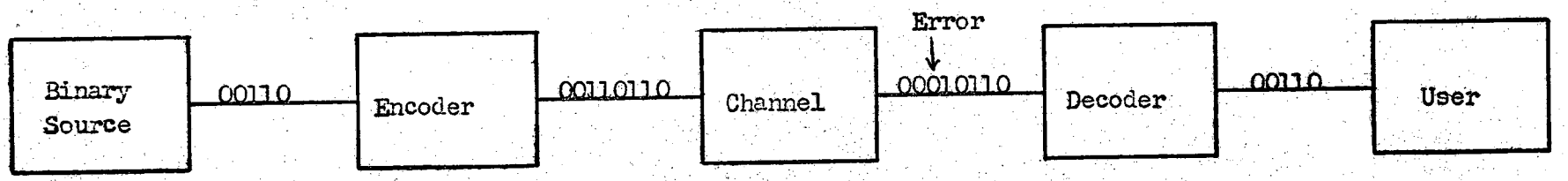
1. Use Error Correcting codes that correct an error before the message is presented to the user. In general this involves the periodic insertion of so-called "check digits" into the sequence of message digits that are to be transmitted. Proper use of these check digits at the receiver allows the most probable transmission errors to be corrected. Fig. 4(b) illustrates a system using this technique. Note that additional equipment, usually a small special purpose digital computer, is required at both the transmitting and receiving terminals to perform the encoding and decoding operations.

2. Use Error Detecting Codes. These codes provide only for the detection of errors and as such are normally of use only when provision is made for the retransmission of incorrectly received messages. This requires the use of a feedback channel from receiver to transmitter which is not always available. However, recent results (3) indicate that this approach offers the greatest hope for attaining the information and error rates theoretically possible.

3. Use Error Correcting and Detecting codes. With some codes it is



A Binary Communication System



A Binary Communication System with Coding for Error Correction

Fig. 4 - A Binary Communication System

possible to correct some of the received errors and to detect, but not correct, additional errors. When a feedback channel is present these detected, but uncorrected, erroneous messages are retransmitted. The advantage of these codes, as compared to error detection only codes, lies in the reduced capacity required for the feedback channel.

With these techniques it is possible, in principle, to obtain an arbitrarily small error rate provided only that P_0 and P_1 are less then 1/2 and that information is transmitted at a rate below the channel capacity. In practice, an increasingly large amount of coding equipment is necessary as the required error rate is decreased. This means that in most situations a compromise must be made between equipment cost and allowable error rate. At present there is considerable effort being expended to discover codes that require less equipment for a specified error rate. To date the most promising of the new approaches appears to be that of sequential coding (4) discovered by Wozencraft of MIT and that of error detection coding with feedback (3).

In summary, the following concepts from this section should be emphasized.

1. This report will be concerned only with binary communication systems and the coding techniques for these. The binary information source will be considered to produce a sequence of 0's and 1's that represents the information to be transmitted and the transmitter-channel-receiver will be represented by one of the models of Fig. 3.

2. Due to noise on the channel some of the transmitted 0's will be received as 1's and vice versa. This effect is included in the models of Fig. 3 through the probabilities P_0 and P_1.

3. Use of suitable encoding techniques at the transmitting station and decoding techniques at the receiving station allow these errors to be reduced to an arbitrarily low value.

4. There are essentially two coding methods that can be used to approach

this low error rate, namely, error correcting codes or error detecting codes plus a feedback channel for retransmission.

It should be noted that although error detecting and correcting codes form a major portion of the field of coding theory a second type of coding, used with a noise free channel, is also of considerable importance and will be discussed later.

1.3 Information theory concepts.

Up to this point the terms information, channel capacity, information rate, etc. have been used in an intuitive manner. To be able to discuss further the concepts of coding and the benefits to be gained from coding it is necessary that these terms be defined in a precise manner.

Consider the intuitive concept of information. Imagine that a coin is to be tossed. If the coin is biased so that it is certain to come up heads then, intuitively, it would seem that no information would be gained by tossing the coin. Similar reasoning follows if the coin is certain to come up tails. However, when either heads or tails may occur some information may be obtained by tossing the coin. Thus the conclusion to be reached is that information can be obtained from the occurrance of an event only when the probability of that event occurring is less than 1. Extending this reasoning it seems reasonable to require than any measure of information be such that the information associated with an event increase as the probability of the event decreases. This reasoning plus other mathematical requirements (pp 80-82, Ref. 8) leads to the following definition of information, commonly called entropy or uncertainty.

$$\mathcal{H} = -\log_2 P(X) \quad \text{bits/event} \tag{1}$$

Here and throughout the report, all logarithms are to the base 2 unless otherwise noted.

As an example consider the tossing of a biased coin where the probability of obtaining a head is 1/4. From Eq. (1) the information, or entropy associated with

obtaining a head is H = -log 1/4 = log 4 = 2 bits. Similarily the uncertainty associated with a tail is log 4/3 = 0.415 bits. Since different entropies are associated with a head and a tail it is more meaningful to speak of the average uncertainty associated with the toss of the coin. The average uncertainty is just the uncertainty associated with a head times the probability that a head is obtained plus the uncertainty associated with a tail times the probability of a tail. Letting the probability of tails = P(T) and the probability of heads = P(H) the above statement becomes

$$H = -P(T) \log P(T) - P(H) \log P(H) \text{ bits/toss} \qquad (2)$$

Note the use of H to denote the average entropy as contrasted to $\mathcal{H}$ which represents an individual entropy. For the above example Eq. (2) shows that the entropy associated with tossing the biased coin is

$$H = -1/4 \log 1/4 - 3/4 \log 3/4 = .811 \text{ bit/toss.}$$

Observe that H as given by Eq. (2) is maximum for P(T) = P(H) = 1/2. This is intuitively satisfying since this represents a condition of maximum uncertainly about the outcome of the toss of a coin.

Next, consider a binary source that produces a sequence of 0's and 1's. Before each digit is produced there is a certain probability that it will be a 1, denoted P(1), and a corresponding probability that it will be a 0, denoted P(0). Since it is assumed that either a 0 or a 1 must be produced, the relation P(1) + P(0) = 1 must hold. Analogous to Eq. (2) the amount of information produced this source is defined to be

$$H(X) = -P(0) \log P(0) - P(1) \log P(1) \text{ bit/binit} \qquad (3)$$

or, since $P(0) + P(1) = 1$

$$H(X) = -P(0) \log P(0) - [1-P(0)] \log [1-P(0)] \text{ bit/binit} \qquad (4)$$

Here the notation H(X) rather than just H is used so that the entropies associated with a source, H(X), can be differentiated from the entropy at the user, H(Y). The reason for this distinction will become clear as the discussion progresses.

Fig. 5 illustrates this entropy function for values of P(0) between zero and unity. Observe that H(X) is maximum when a 0 and a 1 are equiprobable and is zero when either a 0 or a 1 is certain.

This definition gives the amount of information produced when a single binary digit (binit) is produced. Thus if the source generates 1 binit/sec. it produces information at a rate of H(X) bits/sec. Likewise if m binits/sec are produced the information rate of the source is mH(X) bits/sec. With this definition it is possible to specify unambiguously the amount of information produced by a binary source.

A useful generalization of Eq. (4) can be obtained by considering a discrete source that can produce any one of m symbols each with a specified probability. Denoting the m symbols by X_1, X_2 - - - X_m and the corresponding probabilities by $P(X_1)$, $P(X_2)$, - - - $P(X_m)$ the entropy of such a source is defined to be

$$H(X) = -\sum_{i=1}^{M} P(X_i) \log P(X_i) \text{ bits/symbol} \qquad (5)$$

Example 1.2-1

Consider a discrete information source that produces the following symbols with the probabilities indicated

A	1/2	D	1/16
B	1/4	E	1/32
C	1/8	F	1/32

For this source the average entropy per symbol is

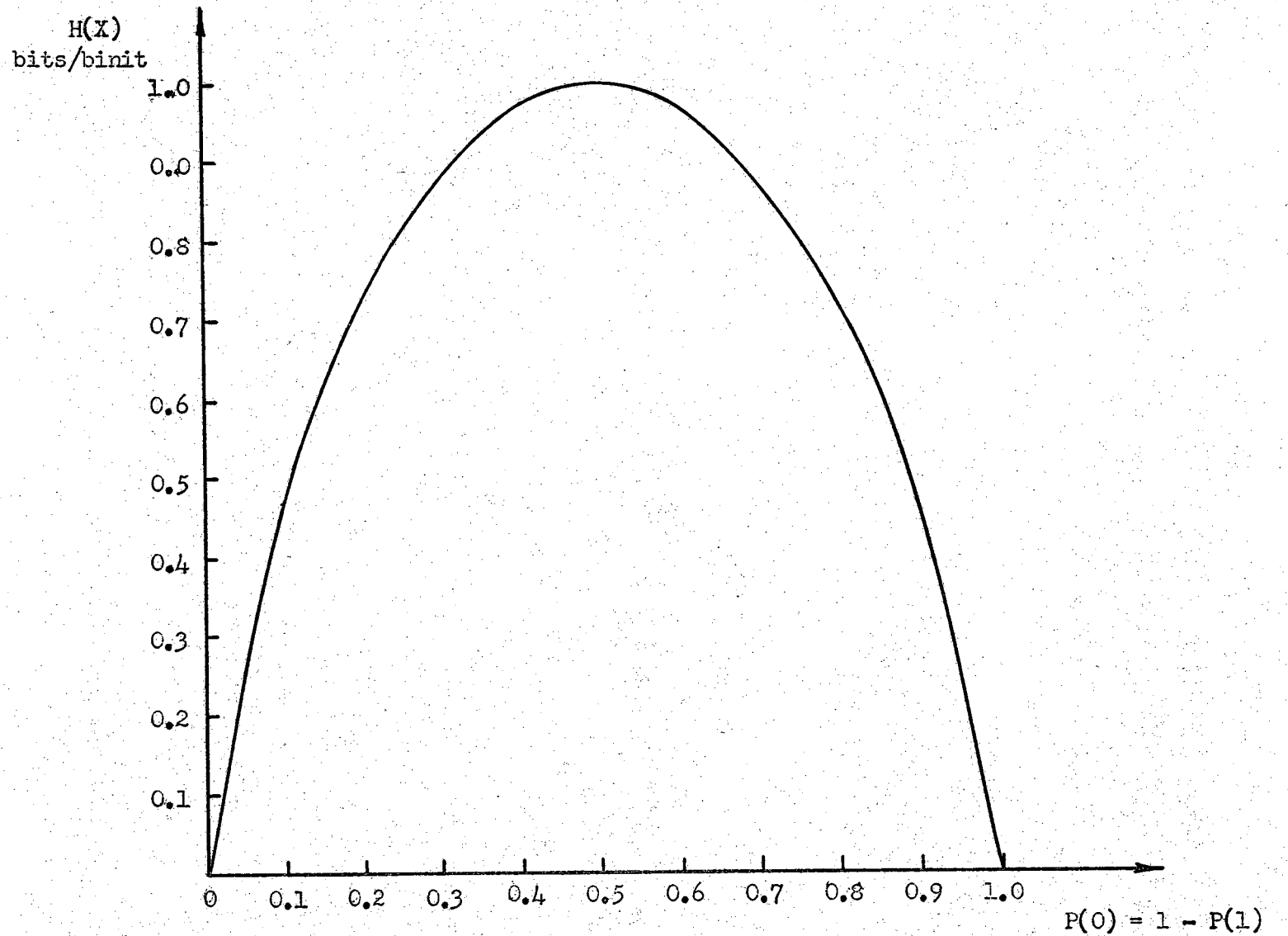


Fig. 5 - Entropy of a Binary Source

$$H(X) = -1/2 \log 1/2 - 1/4 \log 1/4 - - - - - 1/32 \log 1/32$$

$$= 1\ 15/16 \text{ bits/symbol}$$

It can be shown (pp 82-89, Ref. 8) that the entropy of a source is maximum when all symbols are equiprobable. Thus for this case

$$\text{Max } H(X) = -1/6 \log 1/6 - - - - - 1/6 \log 1/6$$

$$= \log 6 = 2.58 \text{ bits/symbol}$$

From this example it can be seen that more information is produced, on a per symbol basis, when a larger number of symbols are possible.

Denoting the symbols supplied to the user by Y, the information supplied to the user is defined in a manner analogous to that of the source, i.e.,

$$H(Y) = -P(Y=0) \log P(Y=0) - P(Y=1) \log P(Y=1) \text{ bits/binit} \qquad (6)$$

The relations between P(Y), the channel probabilities, and P(X) can be determined from Fig. 3(a) and are as follows:

$$P(Y=0) = P(X=0)(1-P_o) + P(X=1)\ P_1$$

$$P(Y-1) = P(X=0)\ P_o + P(X=1)\ (1-P_1) \qquad (7)$$

In certain cases (for example $P(X=0) = P(X=1)$ and $P_o = P_1$) H(Y) and H(X) are equal numerically; however, in general this is not true.

Referring again to the channel model of Fig. 3(a) note that the various probabilities (P_o, P_1, $1-P_o$, $1-P_1$) indicated are actually transitional, or conditional, probabilities, i.e. P_o represents the probability of receiving a 1 given that a 0 is transmitted, P_1 represents the probability of receiving a 1 given that a 0 was transmitted etc.

Thus in a more consistent notation the relations are

$$
\begin{aligned}
P_o &= P(Y=1|X=0) \triangleq^{*} P(Y_1|X_o) \\
P_1 &= P(Y=0|X=1) \triangleq P(Y_o|X_1) \\
1-P_o &= P(Y=0|X=0) \triangleq P(Y_o|X_o) \\
1-P_1 &= P(Y=1|X=1) \triangleq P(Y_1|X_1)
\end{aligned}
\tag{8}
$$

With these definitions three additional entropies associated with the source and user may be defined as follows:

$$H(Y|X) = -\sum_{i=1}^{2}\sum_{j=1}^{2} P(X_i,Y_j) \log P(Y_i|X_j) \tag{9}$$

$$H(X|Y) = -\sum_{i=1}^{2}\sum_{j=1}^{2} P(X_i,Y_j) \log P(X_i|Y_j) \tag{10}$$

$$H(X,Y) = -\sum_{i=1}^{2}\sum_{j=1}^{2} P(X_i,Y_j) \log P(X_i,Y_j) \tag{11}$$

The justification of these entropies on an intuitive basis is not as straight forward as it was for the source and user entropies, H(X) and H(Y). However, some feeling for the meaning of these may be obtained in the following manner. Associated with the occurrence of a specified event at both the transmitter and the receiver (for example the event a 1 is transmitted and a 1 is received) there is a definite probability which depends upon the source probability, $P(X_i)$, and the transitional probability $P(Y_j|X_i)$ which is

$$P(X_i,Y_j) = P(Y_j|X_i)\, P(X_i) \tag{12}$$

From the earlier discussion it appears reasonable to define the information

---

* The symbol $\triangleq$ is to be read: "is defined as" or, "is, by definition, equal to".

associated with this event as

$$\mathcal{H} = -\log P(X_i, Y_j) \text{ bits/occurrence of } X_i \text{ and } Y_j$$

Taking the average of this information over all possible values of X and Y gives the expression of Eq. (10) for the average entropy associated with the joint occurrence of a source and user symbol.

In a similar manner the conditional entropies of Eqs. (9) and (10) are the average of the individual entropies $-\log P(Y_j|X_i)$ and $-\log P(X_i|Y_j)$ respectively. The conditional entropy $H(Y|X)$ can be considered as the uncertainty concerning the received symbol Y when it is known that X has been transmitted. For a nosieless channel Y would be uniquely determined by X and $H(Y|X)$ would be zero. Likewise $H(X|Y)$ is the average uncertainty concerning the symbol transmitted, X, where the received symbol, Y, is known. For a noiseless channel $H(X|Y)$ is also zero.

For convenience the five entropies associated with a source-user combination are listed below along with their appropriate interpretations.

$H(X)$ - A measure of the average uncertainty of the symbols produced by the source in terms of bits/source symbol.

$H(Y)$ - A measure of the average uncertainty associated with the received symbols in the terms of bits/received symbol.

$H(X,Y)$ - The average uncertainty associated with the transmission and reception of a symbol in terms of bits/symbol pair.

$H(Y|X)$ - The uncertainty concerning the received symbol when the transmitted symbol is known.

$H(X|Y)$ - The equivocation of the channel which is a measure of the uncertainty concerning the source when the received symbol is known.

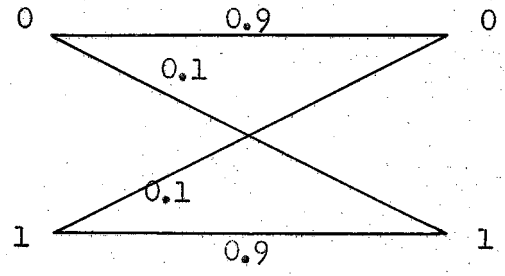
It is readily shown (pp 101-102, Ref. 8) that the following relationships exist between the various entropies.

$$H(X,Y) = H(X) + H(Y|X) = H(Y) + H(X|Y) \tag{13}$$

$$H(X) \geq H(X|Y) \qquad H(Y) \geq H(Y|X) \tag{14}$$

Example 1.2-2

A source produces 0's and 1's with the probabilities $P(0) = 1/4$, $P(1) = 3/4$. The channel transitional probabilities are given by



Determine the various entropies associated with this system.

$H(X) = -1/4 \log 1/4 - 3/4 \log 3/4 = 0.811$ bit/binit

$P(Y_0) = 1/4 \times 0.9 + 3/4 \times 0.1 = 0.300$
$P(Y_1) = 1/4 \times 0.1 + 3/4 \times 0.9 = 0.700$

$H(Y) = -0.3 \log 0.3 - 0.7 \log 0.7 = 0.881$ bit/binit

$$H(Y|X) = -\frac{0.9}{4} \log 0.9 - \frac{0.1}{4} \log 0.1 - \frac{0.9x3}{4} \log 0.9 - \frac{0.9x3}{4} \log 0.1$$

$$= -0.9 \log 0.9 - 0.1 \log 0.1$$

$H(Y|X) = 0.469$ bit/binit

$P(X_1,Y_1) = P(Y_1|X_1)\, P(X_1) = 0.9 \times 0.75 = .675$

$P(X_1,Y_0) = P(Y_0|X_1)\, P(X_1) = 0.1 \times 0.75 = .075$

$P(X_0,Y_1) = P(Y_1|X_0)\, P(X_0) = 0.1 \times 0.25 = .025$

$P(X_0,Y_0) = P(Y_0|X_0)\, P(X_0) = 0.9 \times .25 = .225$

$$H(X,Y) = -.675 \log .675 - .225 \log .225 - .075 \log .075 - .025 \log .025$$

$= 1.280$ bits/ pair of binits

$$P(X_1|Y_1) = \frac{P(X_1,Y_1)}{P(Y_1)} = \frac{.675}{.7} = .965$$

$$P(X_1|Y_o) = \frac{P(X_1,Y_o)}{P(Y_o)} = \frac{.075}{.3} = .25$$

$$P(X_1|Y_o) = \frac{P(X_o,Y_1)}{P(Y_1)} = \frac{.025}{.7} = .035$$

$$P(X_o|Y_o) = \frac{P(X_o,Y_o)}{P(Y_o)} = \frac{.225}{.3} = .75$$

$$H(X|Y) = -.765 \log .965 - .075 \log .25 - .025 \log .035 - .225 \log .75$$

$$= 0.399 \text{ bit/binit}$$

Note that

$$H(X,Y) = H(Y) + H(X|Y) = 0.881 + 0.399 = 1.280 \text{ bit}$$
$$= H(X) + H(Y|X) = 0.811 + 0.469 = 1.280 \text{ bit}$$

$$H(X) = .811 \geq H(X|Y) = 0.399$$

$$H(Y) = .881 \geq H(Y|X) = 0.469$$

Note that in this example the uncertainty, H(Y), at the user is greater than that, H(X) supplied to the channel. It should be emphasized that this increase in extropy does not mean that useful information is gained on the channel but only that the noise has introduced additional uncertainty into the received symbols. The following discussion concerning the actual information transmitted through the channel will further clarify this point.

One additional concept, that of mutual information or transinformation is required before proceeding further. Mutual information is a measure of the amount of information transmitted through channel and is defined, for the binary channel as

$$I(X,Y) = \sum_{i=1}^{2} \sum_{j=1}^{2} P(X_i,Y_j) \log \frac{P(X_i,Y_j)}{P(X_i)P(Y_j)} \qquad (15)$$

Straight forward algebraic manipulations show that

$$\begin{aligned} I(X,Y) &= H(X) - H(X|Y) && \text{bits/binit} \\ &= H(Y) - H(Y|X) && \text{bits/binit} \\ &= H(X) + H(Y) - H(X,Y) && \text{bits/binit} \end{aligned} \tag{16}$$

For a noiseless channel $H(X) = H(Y)$, $H(X|Y) = H(Y|X) = 0$ and the information transmitted through the channel is equal to the source information $H(X)$. Conversely, when the noise on the channel is so great that $P(Y_1|X_0) = P(Y_0|X_1) = 1/2$ then $H(X) = H(X|Y)$ and the information transmitted through the channel is zero. Since these are intuitively satisfying results this appears to be reasonable definition for the amount of information transmitted through a channel.

Considering the results of Example 1.2-2 above observe that the information transmitted through the channel is

$$\begin{aligned} I(X,Y) &= H(X) - H(X|Y) = 0.811 - 0.399 = .412 \text{ bit/binit} \\ &= H(Y) - H(Y|X) = 0.881 - .469 = 0.412 \text{ bit/binit} \\ &= H(X) + H(Y) - H(X,Y) = .811 + .881 - 1.280 = .412 \text{ bit/binit} \end{aligned}$$

Thus, due to noise on the channel, the information $I(X,Y)$, transmitted through the channel is considerably less than that supplied to it.

With these precise definitions for the amount of information supplied by a source and the amount of information transmitted by a channel it is now possible to define precisely what is meant by channel capacity. According to Shannon (1) the capacity of a discrete, memoryless, channel is given by

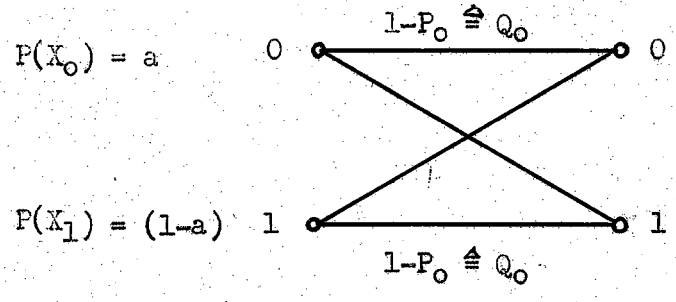
$$\begin{aligned} C = \max I(X,Y) &= \max \left[H(X) - H(X|Y)\right] \\ &= \max \left[H(Y) - H(Y|X)\right] \end{aligned} \tag{17}$$

where the maximization is with respect to the source probabilities $P(X)$. This definition is completely general, applying to all discrete channels and even to continuous channels when the various entropies are properly defined. The work in this report will be concerned with only the binary symmetric channel (BSC)

of Fig. 3(b) and the binary erasure channel (BEC) of Fig. 3(c). The calculation of the capacity for these channels is straightforward and is illustrated below.

Example 1.2-3

Determine the capacity of the BSC shown below



For this

$$-H(Y|X) = aP_o \log P_o + (1-a) P_o \log P_o + aQ_o \log Q_o + (1-a) Q_o \log Q_o$$

$$= P_o \log P_o + Q_o \log Q_o$$

thus

$$C = \max \left[H(Y) + P_o \log P_o + Q_o \log Q_o \right]$$

From previous results H(Y) is maximum when

$$P(Y_0) = P(Y_1) = 1/2$$

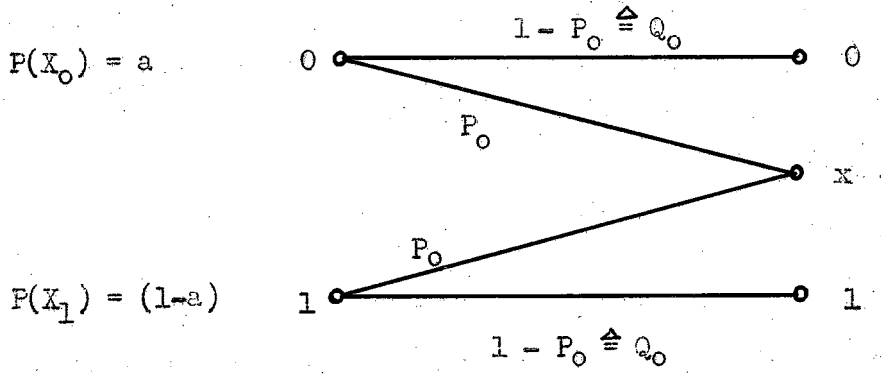
and has a value of unity. Thus, for the BSC

$$C = 1 + P_o \log P_o + Q_o \log Q_o \text{ bits/binit} \qquad (18)$$

Since $P(Y_0) = a\, Q_o + (1-a)\, P_o = 1/2$, the source probability that will transmit information over the channel at this rate is $a = 1/2$.

Example 1.2-4

Determine the capacity for the BEC shown



For a BEC the symbols received as x are ignored. Thus

$$H(Y|X) = - aQ_o \log Q_o - (1-a)\, Q_o \log Q_o$$

$$= - Q_o \log Q_o$$

Since $H(Y|X)$ is independent of $P(X)$ it is necessary only to maximize $H(Y)$.

As before $H(Y)$ is maximum for equiprobable symbols. The corresponding entropy is

$$\text{Max } H(Y) = - 1/2\, Q_o \log 1/2\, Q_o - 1/2\, Q_o \log 1/2\, Q_o$$

$$= - Q_o \log 1/2\, Q_o$$

$$C = Q_o \left[\log Q_o - \log 1/2\, Q_o\right] = Q_o \text{bits/binit} \qquad (19)$$

It is an interesting property of the BEC that for $Q_o > 0.23$ the channel capacity is greater than that for the BSC. In addition all digits received as a 0 or a 1 are correct. Because of this it is in some cases easier to use error correcting codes with the BEC than with the BSC.

With this material as background it is now possible to give the first and second fundamental theorems of Shannon. It is because of the communication possibilities promised by these theorems, and the fact that both theorms are based upon the assumption of appropriate coding techniques, that the current interest exists in the field of coding theory.

In the first of Shannon's theorems channel capacity is considered on a per second basis rather than on a per binit basis. If m binits/sec can be transmitted over a channel its capacity, on a per second basis, is

$$C' = mC \text{ bits/sec} \tag{20}$$

Shannons First Fundamental Theorem applies to a discrete, noiseless, (i.e. the probability of an error is zero) memoryless channel and is as follows.

Theorem - Let a source have an entropy of H(X) bits/source symbol and a channel a capacity of C' bits/sec. Then it is possible to encode the output of the source in such a way as to transmit over the channel at an average rate arbitrarily close to C'/H source symbols per second. It is not possible to transmit at an average rate greater than C'/H.

Considering the symbols of Example 1.2-1 and assuming a binary channel with m = 1 binit/sec. this theorm states that the source symbols, A, B, C, D, E, F, can be encoded into binary digits in such a manner that they can be transmitted over the channel at a rate up to but not exceeding 16 source symbols per 31 second.

The importance of this is made clear by noting that since there are 6 symbols to be represented a 3 digit code would appear to be necessary. This would allow transmission of only 1 symbol/3 seconds which is considerably less than that indicated by Shannon's theorem. Optimum coding techniques for this situation have been developed and will be presented later in this report.

Shannon's second fundamental theorem, given earlier in a less precise form, applies to a noisy, memoryless channel and is as follows:

Theorem - Let a binary source have an entropy of H(X) bits/binit and channel a capacity of C bits/binit. Then if $H(X) < C$ there exists a coding scheme such that the output of the source may be transmitted over the channel with arbitrarily small error rate. This is not possible if

$H(X) > C$.

The importance of this theorem lies in the fact that it was previously thought that a reduction in the error rate could be accomplished only through a corresponding reduction in the information rate. Thus as the error rate approached zero so would the information rate. Shannon's theorem states that this is not true provided that proper encoding techniques can be obtained.

The determination of these techniques represents the major effort in coding research at the present time.

CHAPTER II

CODING FOR THE NOISELESS CHANNEL

2.1 Introduction

Before starting a discussion of coding the following definitions, pertaining to a noiseless channel, are given.

Source symbol - - One of n possible symbols produced by a message source.

Alphabet - - A list of all n allowable source symbols.

Message - - A finite sequence of source symbols.

Encoding or enciphering - - By definition this operation occurs at the transmitter and is a procedure for associating the source symbols with a corresponding set of binary digits in a one-to-one manner.

Decoding or deciphering - - This operation occurs at the receiver and corresponds to the inverse of encoding, i.e., it is a procedure whereby the received set of binary digits are associated with the original source symbols.

Coding - - A general term including both the operation of encoding and that of decoding.

Code word - - The binary number assigned to a source symbol. This may be composed of one or more binary digits.

Length of a Code word - - The number of binits in a code word.

Optimum Code - - A code having the maximum possible efficiency for a given set of source symbols and probabilities.

The capacity, C', of a noiseless, binary channel is given by Eqs. (18) and (20) (with $P_o = 0$) as a m bits/sec. Shannon's first theorem states that the symbols from a source having an entropy of H(X) bits/source symbol can be encoded in such a manner that they can be transmitted over this channel at a rate up to but not exceeding $\frac{C'}{H(X)}$ source symbols/sec. Thus if a binary source with

$P(0) = P(1) = 1/2$ is considered the entropy is $H(X) = 1$ bit/source symbol and the symbols can be transmitted at a rate of m source symbols per second. Obviously this represents straight foreward transmission of binits each having maximum possible entropy and no coding is possible or necessary. However, if the source probabilities are such that $H(X) = 0.25$ bit/source symbol the theorem states that 4 m source symbols, or binits, can be transmitted over the channel each second. Since the channel can transmit only m binits per second coding is obviously required for this case. The following example illustrates encoding for this situation.

Example 2.1-1

Assume that a coin is to be tossed 100 times at the rate of one toss/sec and that the results (heads=1 or tails = 0) are to be transmitted, in order, over a noiseless binary channel. If a fair coin is considered the probabilities will be $P(0) = P(1) = 1/2$ giving rise to a source entropy, on a per second basis, of

$$H'(X) = -1/2 \log 1/2 - 1/2 \log 1/2 = 1 \text{ bit/sec.}$$

Since the capacity of a noiseless binary channel transmitting 1 binit/sec is 1 bit/sec., the entropy of the information supplied to this channel is equal to the channel capacity and coding is not required.

Next consider the case where the coin is biased so that the probabilities are $P(0) = 0.05$ and $P(1) = 0.95$.

Under this condition the source entropy is

$$H'(X) = -0.05 \log 0.15 - 0.95 \log 0.95$$
$$= 0.286 \text{ bit/sec.}$$

Direct transmission of these symbols results in an information input to

the channel of approximately one-fourth its capacity. Shannon's first theoremstates that this situation can be improved by suitable coding. Ideally this coding would lead either to the transmission of nearly 4 times as many source symbols per second over the same channel or the transmission of the same number of source symbols per second over a channel having a capacity of approximately one-fourth that of the original channel. In either case this would represent a considerable increase in the channel utilization. To demonstrate the improvement possible with a relatively simple code consider the technique of transmitting only the positions in which a 0 occurs and assuming that all other positions are 1's. Since there are 100 positions to be represented, a seven digit binary code is required. Thus if a 0 occurs in positions 7, 25, 63, 75 and 92 the code sequence to be transmitted is

0000111 0011001 0111111 1001011 1011100

If this experiment were repeated a large number of times the average number of 0's appearing would be 5 and the average code length would be 35 binits. Thus a channel operating at a rate of 0.35 binit/sec can be used to transmit the coded message as compared to a rate of 1 binit/sec required for the uncoded message.

Since no information is gained or lost in the encoding process the information associated with one binit in the original sequence must be the same as that associated with 0.35 binit in the code sequence. This gives an entropy for the code binits of

$$H_c(X) = \frac{0.286 \text{ bits/source binit}}{0.35 \text{ code binit/source binit}}$$

$$= 0.817 \text{ bits/code binit}$$

A convenience measure of the efficiency of a coding procedure is the ratio of the average information per code binit to the maximum possible

information per code binit. The efficiency, η_c, is thus

$$\eta_c = \frac{H_c(X) \text{ bits/code binit}}{1 \text{ bit/code binit}} \times 100 = 100\, H_c(X)\,\% \qquad (21)$$

$$= 81.7\%$$

Later discussion of more sophisticated techniques will show that in general efficiencies of greater than 95% are readily obtained.

Note that η_c is equally well a measure of the efficiency of channel utilization since it is equivalent to the ratio of the actual rate of information transmission to the maximum possible rate of information transmission.

This example illustrates encoding for a binary source. In many case this binary source would have been obtained by assigning binary digits to each of the n (n an integer > 2) possible messages of the original message source. An example of this would be the transmission of English text by assigning 5 binits to each letter of the alphabet. In general this would not result in binary sequences for which $H_c(X) = 1$ bit/binit and therefore coding of the binary source would be required. This two step encoding procedure is rather pointless since it should be possible to encode the original message in such a manner that $H_c(X) \simeq 1$ bit/binit. The following example illustrates this point.

Example 2.1-2

Consider the source of Example 1.2-1. For this source the symbols and their probabilities were

A	1/2	D	1/16
B	1/4	E	1/32
C	1/8	F	1/32

These six symbols are to be transmitted over a binary channel and therefore must be represented by binary digits. When assigning binits to

these symbols it should be realized that if fewer digits are assigned to the symbols having the greatest probability then, with care, the average-number of binits per source symbol will be less than when an equal number of binits are assigned to each source symbol.

With this in mind consider the following code.

A	0	D	110
B	10	E	11110
C	110	F	11111

The average number of binits, $\overline{L}$, required for this code is

$$\overline{L} = 1x1/2 + 2x1/4 + 3x1/8 + 4x1/16 + 5x1/32 + 5x1/32$$

$$= 1\ 15/16 \text{ binits/source symbol}$$

The entropy of this source was previously found to be

$$H(X) = 1\ 15/16 \text{ bits/source symbol}$$

The entropy of the code digits is given by

$$H_c(X) = \frac{H(X)}{\overline{L}} = 1\ 15/16 \frac{\text{bit}}{\text{source symbol}} \times \frac{1 \text{ source symbol}}{1\ 15/16 \text{ binits}}$$

$$= 1 \text{ bit/binit}$$

Since the maximum entropy of a binit is 1 bit/binit, the coding efficiency is

$$\eta_c = \frac{1 \text{ bit/binit}}{1 \text{ bit/binit}} \times 100 = 100\%$$

Note that

$$P(0) = \frac{\text{average number of zeros}}{\text{average number binits}}$$

$$= \frac{1x1/2 + 1x1/4 + 1x1/8 + 1x1/16 + 1x1/32}{1\ 15/16} = 31/32 \times 16/31 = 1/2$$

Likewise

$$P(1) = \frac{1x1/4 + 2x1/8 + 3x1/16 + 4x1/32 + 5x1/32}{1\ 15/16} = 31/32 \times 16/31 = 1/2$$

or, equally well,

$$P(1) = 1-P(0) = 1/2$$

This illustrates that for this case the code binits are equiprobable and independent.

At this point a question mught be raised concerning the reason for assigning such a large number of binits to some of the symbols in the above example. For example why not assign the code words in the following, apparently much more efficient, way?

A	0	D	01
B	1	E	10
C	00	F	11

This gives an average length of

$$\bar{L} = 1x1/2 + 1x1/4 + 2x1/8 + 2x1/16 + 2x1/32 + 2x1/32$$

$$= 1\ 1/4 \text{ binit/source symbol}$$

$$H_c(X) = \frac{H(X)}{\bar{L}} = \frac{1\ 15/16}{1\ 1/4} = \frac{31}{20} \text{ bits/binit}$$

Since the maximum possible entropy for binary digits is 1 bit/binit it is obvious that a falacy in this coding scheme must exist and indeed one does. This is readily demonstrated by considering the code sequences that would be transmitted for the message symbols A C F E D B. These sequences are as follows:

a) original code 011011111111110111010

b) alternate code 0001110011

Imagine that these sequences have been received and are to be decoded. The decoding procedure consists of checking the first digit to see if it corres-

ponds to a source symbol. If it does not then the first two symbols are considered. This procedure is continued until a group of digits are recognized that correspond to a source symbol. The symbol is then recorded and the procedure repeated for the following digits in exactly the same manner.

The following illustrates this procedure for the original code.

digits checked	result
0	A
1	meaningless
11	meaningless
110	C
1	meaningless
11	meaningless
111	meaningless
1111	meaningless
11111	F
1	meaningless
1111	meaningless
11110	E
1	meaningless
11	meaningless
111	meaningless
1110	D
1	meaningless
10	B

Thus the transmitted message is

A C F E D B

When this procedure is applied to the alternate code the following sequences

are obtained as possible transmitted messages.

AAA BBB AA BB

AAA F E D B

C D F B C F

A C F E D B

Because of this, unambiguous decoding is not possible for the alternate coding scheme and no useful information can be transmitted. It is for this reason that an inconsistent value was found for the entropy of the alternate code digit.

Since it is usually desirable to obtain codes having a maximum average length it is well to reconsider the above situation in an attempt to determine why one code failed and the other did not. Consider the siutation that would exist if the alternate code were transmitted with a space between each of the code words. For this case the code sequences would be

0 00 11 10 01 1

Obviously no ambiguity exists with this code and the transmitted message is directly obtained as ACFEDB. Note, however, that this spacing of code words was not required for the original code. Thus the property required for unambiguous decoding is that the code words can be transmitted in sequence without intervening spaces. A code having this property is described as being uniquely decipherable. Further consideration will show that the alternate code does not have this property due to the fact that some of the code words can be obtained from others by adding a digit. For example the code word for C is obtained from the code word for A by adding a 0. Observe that this situation does not exist in the original code, i.e. no code word is the prefix of another code word. It is this property, called the prefix property, that determines whether or not a particular code is uniquely decipherable.

From this discussion the two requirements for an optimum code should be

clear.

1. The code must be uniquely decipherable i.e., each code word must have the prefix property. This condition also insures that $H_c(X) < 1$ bit/binit. (A proof of this statement is given in Ref. 8, pp 148-151). Because of this, η_c, as defined above, can never exceed 100%.

2. The average length of a code word should be as small as possible consistent with the above requirement.

For the sake of completeness it should be noted that it is possible to conceive of codes that do not have the prefix property but are still uniquely decipherable. For example consider the code

A 1

B 10

C 100

Application of the above decoding procedure to any sequence of these code words shows this to be a uniquely decipherable code. However, there appears to be no general method for determining such codes and in addition no known codes of this type have a higher efficiency than codes having the prefix property. Thus all codes discussed in the following sections of this chapter will have the prefix property.

The following sections will discuss, in a more or less chronological order, the various better known techniques used in coding for the noiseless channel. Since Huffman encoding represents the optimum (in the source of giving maximum efficiency) coding procedure it may seem superfluous to describe some of the other non-optimum techniques. To delete these, however, would be to defeat the purpose of this report.

2.2 Shannon-Fano Encoding

The Shannon-Fano encoding procedure (9) appears to have been the first constructive procedure for determining codes having the prefix property and as

such represents a logical starting point in the discussion of specific coding techniques.

In essence the procedure is a technique for assigning binary digits to source symbols in such a manner that the number of binits assigned is inversely proportional to the probability of the corresponding message symbol. The procedure consists of listing the source symbols in the order of nonincreasing probability and then dividing this group of symbols into two new groups having approximately equal probabilities. A 0 is assigned as the first digit of the code words in one group and a 1 is assigned to the first digit in the other group. This subdivision process is then repeated until groups are obtained that contain only one source symbol each. The resulting code will in all cases have the prefix property although it will not always have maximum efficiency. The following examples illustrate this procedure.

Example 2.2-1

Apply the Shannon-Fano encoding procedure to the following source.

Symbol	A	B	C	D	E	F
Probability	1/2	1/4	1/8	1/16	1/32	1/32

Step 1. List the symbols in the order of nonincreasing probability.

Step 2. Divide the list into two groups having probabilities that are as nearly equal as possible.

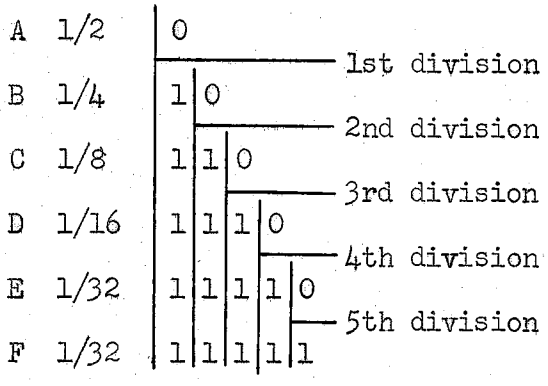
A 1/2 } total prob. = 1/2

B 1/4 }
C 1/8 }
D 1/16 } total prob. = 1/2
E 1/32 }
F 1/32 }

Although probabilities of exactly 1/2 are obtained for this problem in general this will not be possible.

Step 3. Assign a 0 as the first digit in the code word for the first group and a 1 as the first digit in the code word for the second group.

Step 4. Repeat the division and assigning of digits process until single symbol groups are obtained.



Observe that this code has the prefix property and from Example 2.1-2 its efficiency is 100%. Thus this is an optimum code and no other procedure can yield a better code. This situation, however, is not typical of Shannon-Fano encoding and occurs in this example only because of the particular source probabilities used. In fact the following proof shows that 100% efficiency is possible only when

$$P(X_i) = 2^{-n_i} \tag{22}$$

where n_i is the number of letters in the i th code word. Note that this relation exists in the above example.

Taking the logarithm of both sides of Eq. (21), multiplying by $P(X_i)$, and summing over all i yields

$$-\sum_{i=1}^{N} P(X_i) \log P(X_i) = \sum_{i=1}^{N} P(X_i)\, n_i \tag{23}$$

The right hand side of this expression is the average length of the code

words, $\bar{L}$, while the left hand side is the entropy of the source symbols, H(X). Thus $H(X) = \bar{L}$ and $\eta_c = \frac{H(X)}{\bar{L}} \times 100 = 100\%$. Any other sets of probabilities will lead to the condition

$$P(X_i) > 2^{-n_i}$$

causing $\bar{L}$ to be greater than H(X) and η_c to be less then 100%.

Example 2.2-2

Determine the Shannon-Fano code for the following source.

Source Symbol	X_1	X_2	X_3	X_4	X_5	X_6	X_7	X_8	X_9	X_{10}
Probability	.3	.2	.2	.1	.05	.05	.03	.03	.02	.02

Applying the procedure demonstrated above yields

X_1	.3	0 0	
X_2	.2	0 1	2nd division
X_3	.2	1 0 0	1st "
X_4	.1	1 0 1	4th "
X_5	.05	1 1 0 0	3rd "
X_6	.05	1 1 0 1	6th "
X_7	.03	1 1 1 0 0	5th "
X_8	.03	1 1 1 0 1	8th "
X_9	.02	1 1 1 1 0	7th "
X_{10}	.02	1 1 1 1 1	9th "

$$\bar{L} = 0.6 + 0.4 + 0.6 + 0.3 + 0.2 + 0.2 + 0.15 + 0.15 + 0.1 + 0.1 = 2.8 \text{ binits/source symbol}$$

$$H(X) = -(0.3 \log 0.3 + 0.4 \log 0.2 + 0.1 \log 0.1 + 0.1 \log 0.005 + 0.06 \log 0.03 + 0.04 \log 0.02)$$

$$= 2.743 \text{ bit/source symbol}$$

$$\eta_c = \frac{H(X)}{\bar{L}} \times 100 = \frac{2.742}{2.8} \times 100 = 97.9\%$$

Although this efficiency is quite high it will be shown later that this code is not optimum i.e., it is possible to obtain a higher efficiency with another coding technique. Obviously any improvement will be small.

Example 2.2-3

Apply the Shannon-Fano encoding procedure to the binary source of Example 2.1-1 for the case of P(0) = 0.05.

The procedure for encoding a binary source consists of grouping the source binits into groups of two or more binits and considering these groups as new source symbols. Binary code words are then assigned to these symbols in the usual manner.

Consider the case of using 2 binits per group. The possible sequences two binits in length are 00, 01, 10, 11. Assuming successive binits to be indpendent, the corresponding probabilities of these sequences are

Sequence	Probability
00	.05 x .05 = .0025
01	.05 x .95 = .0475
10	.95 x .05 = .0475
11	.95 x .95 = .9025

The Shannon Fano code for these sequences is thus

Sequence	Code word
00	0
01	10
10	110
11	111

$\bar{L}$ = 1 x .9025 + 2 x .0475 + 3 x .0475 + 3 x .0025

= 1.1475 binit/sequence

Since the source binits are considered in pairs the average entropy per pair is twice the entropy for a single binit. Thus, from example 2.1-1,

$H(X) = 2 \times 0.286 = 0.572$ bits/source symbol

$$\eta_c = \frac{0.572}{1.1475} \times 100 = 49.7\%$$

This example illustrates that encoding groups of two source binits can reduce the required channel capacity from 1 bit/sec to 0.5875 bit/sec. An even greater reduction in channel capacity can be obtained by encoding larger groups of source symbols. This is illustrated in the following example.

Example 2.2-4

A source produces two independent symbols, A and B, with the probability $P(A) = 1/16$, $P(B) = 15/16$.

It is desired to encode these so as to obtain a coding efficiency greater than 70%.

1. Encoding of single symbols

Symbol	Probability	code word
A	1/16	0
B	15/16	1

$\bar{L} = 1$

$H(X) = -(1/16 \log 1/16 + 15/16 \log 15/16)$

$= 0.337$ bit/source symbol

$\eta_c = 33.7\%$

2. Encoding of pairs of symbols

Symbol	Probability	code word
BB	225/256	0
AB	15/256	10
BA	15/256	110
AA	1/256	111

$$\bar{L} = 1/2 \left(\frac{225}{256} + \frac{30}{256} + \frac{45}{256} + \frac{3}{256} \right) = 0.592 \text{ binit/source symbol}$$

$$\eta_c = \frac{H(X)}{\bar{L}} \quad \frac{0.337}{0.592} \times 100 = 56.5\%$$

3. Encoding of 3 symbols

Symbol	Probability	code word
BBB	$(15/16)^3$	0
BBA	$(15/16)^2(1/16)$	100
BAB	$(15/16)^2(1/16)$	101
ABB	$(15/16)^2(1/16)$	110
BAA	$(15/16)(1/16)^2$	11100
ABA	$(15/16)(1/16)^2$	11101
AAB	$(15/16)(1/16)^2$	11110
AAA	$(1/16)^3$	11111

$$\bar{L} = 1/3 \left(\frac{3375 + 9 \times 225 + 46 \times 5}{4096} \right)$$

$$= 0.458 \text{ binit/source symbol}$$

$$\eta_c = \frac{0.337}{0.458} \times 100 = 73.3\%$$

This illustrates that encoding larger groups of source symbols yields more efficient codes. Actually it is possible to obtain an η_c arbitrarily close to 100% by encoding suitably large groups of symbols. However, for this example the efficiency increases so slowly for groups greater than 4 or 5 binits in length that the increased cost of the encoding equipment would, in most situations, more than offset the increase in coding efficiency.

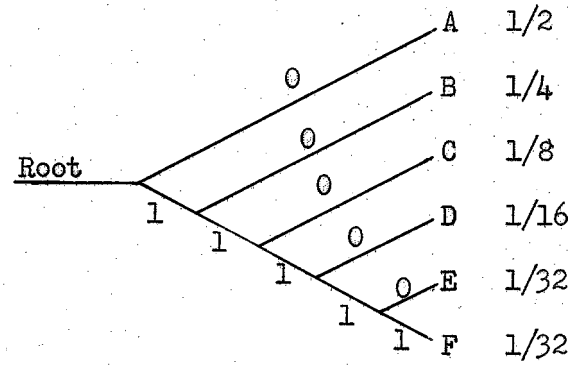
A second fundamental limitation of any coding scheme can be observed in this example. Note that as larger groups of source symbols are encoded there is an increasing amount of delay between the time that a source symbol is pro-

duced and the time that its code word is transmitted. Thus, depending upon the particular application, there may also be a limit on the size of the groups that may be encoded due to a limitation on the allowable delay time.

An alternate method for constructing Shannon-Fano codes consists of using a coding tree. When using the coding tree it is desired that the probabilities of symbols whose branches meet at a node point be as nearly equal as possible. The use of the code tree is best demonstrated by giving the code tree for some of the previously derived codes.

Example 2.2-5

Determine the Shannon-Fano code for the source of Example 2.2-1 by means of the code tree.

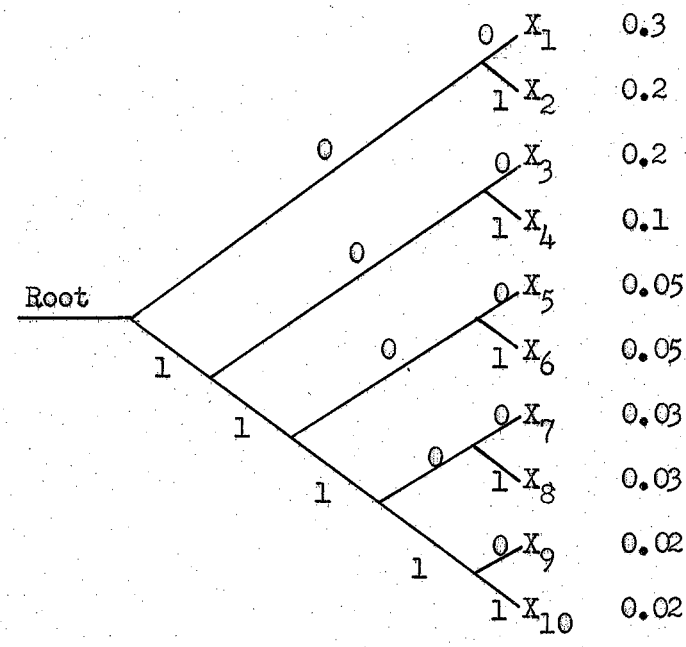


Note that at each node point the symbol probabilities are equal. When this condition exists a 100% efficient code is obtained.

The code words are obtained by starting at the root and progressing via the branches to the desired symbol noting the 0's and 1's that are encountered on each branch. Thus the code word for C is 110. The resulting code words derived from this tree are the same as those of Example 2.2-1.

Example 2.2-6

Draw the code tree for the source of Example 2.2-2



In this case the symbol probabilities are not equal at each node point and a 100% efficient code is not obtained. The code symbols derived from this tree correspond to those found in Example 2.2-2.

2.3 Shannon's Binary Encoding

Shannon's binary encoding procedure (pp. 402-403 Ref. 1) is primarily of theoretical interest since it has no practical advantages over other techniques and often has a lower efficiency. It is presented here for the sake of completeness and because it allows a simple verification of Shannon's first fundamental theorem.

The procedure is based upon determining code words that have the prefix property and satisfy the following relation.

$$2^{-n_i+1} > P(X_i) \geq 2^{-n_i} \tag{24}$$

where, as before, n_i is the number of binits in the i th code word. The code words are determined in the following manner.

1. List the symbol probabilities in nonincreasing order and let these be denoted by $P(X_1)$, $P(X_2)$ - - - $P(X_n)$ where

$$P(X_1) \geq P(X_2) \geq - - \geq P(X_n)$$

2. Calculate the numbers

$$P_k = \sum_{i=1}^{k-1} P(X_i) \qquad k = 2, 3, - - -, n$$

$$P_1 = 0$$

3. For the k th symbol write P_k as a binary number* of n_k binits where

---

* In the binary representation of a number less than unity the binary digit weights are

$$.2^0, 2^{-1}, 2^{-2}, 2^{-3}, - - - -$$

Thus, for example,

$$0.8 = .2^{-1} + 2^{-2} + 0{:}2^{-3}\ 0{:}2^{-4} + 2^{-5}+$$

$$= .11001 - - -$$

n_k is an integer satisfying Eq. (24). The resulting binary number is the k th code word.

It can be shown (p402, Ref. 1) that these code words have the prefix property. The following example illustrates this procedure.

Example 2.3-1

Determine the Shannon binary code for the following source and compare it to the corresponding Shannon-Fano code.

Source Symbol	X_1	X_2	X_3	X_4	X_5	X_6	X_7	X_8
Probability	0.4	0.1	0.01	0.2	0.01	0.03	0.2	0.05

1. List probabilities in nonincreasing order.

X_1	X_4	X_7	X_2	X_8	X_6	X_3	X_5
0.4	0.2	0.2	0.1	0.05	0.03	0.01	0.01

2. Calculate $P_{k's}$

$P_1 = 0 \quad = .0000 - -$

$P_2 = 0.4 \quad = .0110 - -$

$P_3 = 0.6 \quad = .1001 - -$

$P_4 = 0.8 \quad = .1100 - -$

$P_5 = 0.9 \quad = .11100 - -$

$P_6 = 0.95 \quad = .111100 - -$

$P_7 = 0.98 \quad = .1111101 - -$

$P_8 = 0.99 \quad = .1111110 - -$

3. Determine n_k's and code symbols

k	n_k	Source Symbol	Code word
1	2	X_1	00
2	3	X_2	1100
3	3	X_3	1111101
4	4	X_4	011
5	5	X_5	1111110
6	6	X_6	111100
7	7	X_7	100
8	7	X_8	11100

For this code

$$\bar{L} = 0.8 + 0.6 + 0.6 + 0.4 + 0.25 + 0.18 + 0.07 + 0.07$$

$$= 2.97 \text{ binits/source symbol}$$

$$H(X) = -(0.4 \log 0.4 + 0.4 \log 0.2 + 0.1 \log 0.1 + 0.05 \log 0.05 + 0.03 \log 0.03 + 0.02 \log 0.02)$$

$$= 2.29 \text{ bits/source symbol}$$

Thus the entropy of the code digits is

$$H_c(X) = \frac{H(X)}{\bar{L}} = \frac{2.29}{2.97} = 0.77 \text{ bit/binit}$$

and

$$\eta_c = 77\%$$

It is readily determined that the Shannon-Fano code for this source is as follows.

X_1 0

X_2 1 1 1 0

X_3 1 1 1 1 1 1 0

X_4 1 0

X_5 1 1 1 1 1 1 1

X_6 1 1 1 1 1 0

X_7 1 1 0

X_8 1 1 1 1 0

For this code

$\bar{L}$ = 2.37 binit/source symbol

η_c = 96.8%

This illustrates the fact that in general Shannon's binary encoding is less efficient than other methods.

The theoretical importance of this coding technique lies in the fact that the condition imposed by Eq. (24) allows bounds to be determined for the average code length $\bar{L}$.

These bounds are readily determined in the following manner. Taking the logarithm of Eq. (24), multiplying by $P(X_i)$ and summing over all i yields

$$\sum_{i=1}^{n} P(X_i)\, n_i \geq -\sum_{i=1}^{n} P(X_i) \log P(X_i) > \sum_{i=1}^{n} P(X_i)\,(n_i - 1) \qquad (25)$$

Noting that

$$\sum_{i=1}^{n} P(X_i)\, n_i = \bar{L}$$

$$-\sum_{i=1}^{n} P(X_i) \log P(X_i) = H(X)$$

$$\sum_{i=1}^{n} P(X_i)\,(n_i-1) = \sum_{i=1}^{n} P(X_i)\,n - \sum_{i=1}^{n} P(X_i) = \bar{L} - 1$$

allows Eq. (25) to be written as

$$\bar{L} \geq H(X) > \bar{L} - 1$$

which can be rearranged to give

$$H(X) + 1 > \bar{L} \geq H(X) \tag{26}$$

Here $H(X)$ is the average entropy per encoded symbol and $\bar{L}$ is the average number of binits per encoded symbol. If it is assumed that the encoded symbols represent groups of N independent source symbols the relation between the entropy of the encoded symbols and that of the source symbol is

$$H(X) = N\,H_s(X) \text{ bits/encoded symbol}$$

where $H_s(X)$ denotes the entropy associated with a single source symbol. Similarly

$$\bar{L} = N\,\bar{L}_s \qquad \text{binits/encoded symbol}$$

Using these relations Eq. (26) can be rewritten as

$$N\,H_s(X) + 1 > N\,\bar{L}_s \geq N\,H_s(X)$$

or

$$H_s(X) + \frac{1}{N} > \bar{L}_s \geq H_s(X) \tag{27}$$

It is this relation that justifies consideration of Shannon's binary encoding procedure. Observe that as larger groups of source symbols are encoded, the average number of code binits per source symbol approaches the entropy of the source symbols. However, the condition $\bar{L}_s = H_s(X)$ is exactly that required to obtain 100% coding efficiency and the transmission of information at the

channel capacity. Thus the limiting form (as N becomes infinite) of Eq. (27) demonstrates that an encoding procedure can be determined for the noiseless channel that will allow the transmission of information at a rate arbitrarily close to the channel capacity.

This statement is exactly that of Shannon's first fundamental theorem and as such demonstrates the importance of Shannon's binary encoding procedure. It should be emphasized, however, that this result does not imply that a more efficient code can not be obtained for a given value of N. The above example illustrates that one can.

2.4 Huffman Encoding

The Huffman encoding procedure (10) is a systematic method for determining optimum codes in the sense that no other codes having the prefix property and a higher efficiency can be determined.

This procedure is slightly more complex than those previously discussed and is as follows:

1. List the symbols to be encoded in the order of nonincreasing probability.
2. Group the two least probable symbols together and consider these as a single new symbol whose probability is the sum of the individual probabilities.
3. Form a new list of symbols containing the remaining original symbols and the new symbol. List these in the order of nonincreasing probability also.
4. Group the two least probable symbols of this list forming a second new symbol whose probability is the sum of the individual probabilities.
5. Repeat the regrouping and relisting process until a one element group having a probability of unity is obtained.
6. Assign code binits to the original symbols according to the position occupied by the symbol in the various subgroups that were formed.

The following examples illustrate systematic procedures for carrying out these steps.

Example 2.4-1

Determine the Huffman code for the following set of source symbols.

Symbols	A	B	C	D	E
Probability	1/2	1/6	1/6	1/12	1/12

Proceeding with steps one through four above gives.

A	1/2	1/2
B	1/6	1/6
C	1/6	1/6
D	1/12	1/6
E	1/12	

The exact location of the resulting symbol is unimportant as long as no symbols having a greater probability are below it in the list.

Continuing with step five gives, as a complete result,

A	1/2	1/2	1/2	1/2	1
B	1/6	1/6	1/3	1/2	
C	1/6	1/6	1/6		
D	1/12	1/6			
E	1/12				

The code binits are determined for each symbol by assigning a 0 to the code word each time the symbol, or a sub-group containing the symbol, is the lower element in a subgroup and a 1 when it is the upper element.

For example, the locations of C, or a subgroup containing C, in the above columns are as follows.

location:	not included	upper	upper	lower
code symbol:	-	1	1	0

The uniquely decipherable code words are obtained by writing these binits in the reverse order.

The code words for this source are thus

A 1

B 0 0

C 0 1 1

D 0 1 0 1

E 0 1 0 0

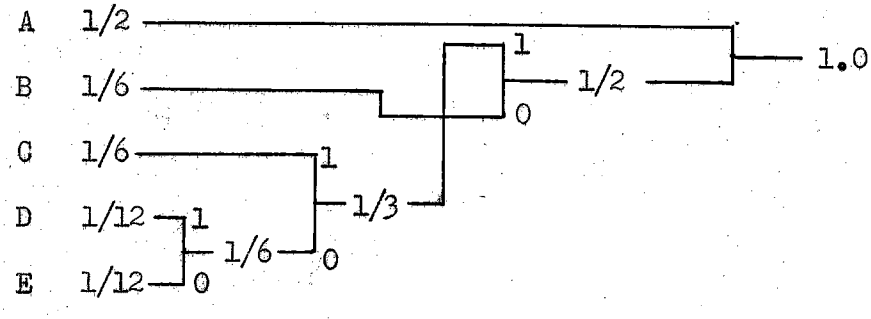
$\bar{L} = 1/2 + 2/6 + 3/6 + 4/12 + 4/12 = 2$ binits/source symbol

$$H(X) = -\sum_{i=1}^{5} P(X_i) \log P(X_i) = 1.959 \text{ bits/source symbol}$$

$$H_c(X) = \frac{H(X)}{\bar{L}} = 0.979 \text{ bits/binit}$$

$\eta_c = 97.9\%$

An alternate procedure for carrying out Huffman encoding that is similar to the coding tree for Shannon-Fano encoding, has been given by Fano (pp. 75 Ref. 11). Applied to this problem it gives the following result.



The determination of the code word from this graph is essentially the same as above, namely, proceed from the symbol via the most direct path to the terminal

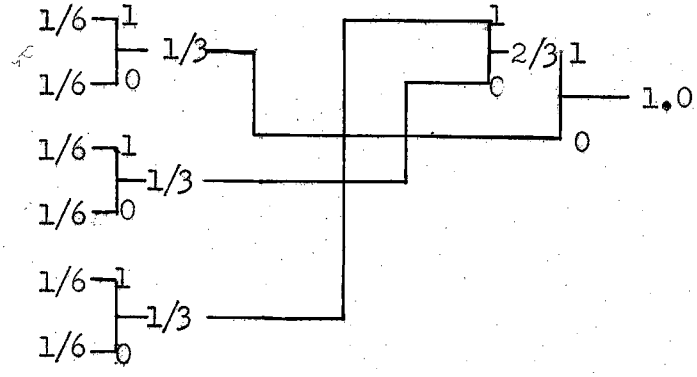
point noting the 0's and 1's encountered. The code words are these digits in reverse order and are the same as those above.

Example 2.4-2

Apply the Huffman encoding procedure to the following symbols.

Symbol	X_1	X_2	X_3	X_4	X_5	X_6
Probability	1/6	1/6	1/6	1/6	1/6	1/6

Code word	Symbol
0 1	X_1
0 0	X_2
1 0 1	X_3
1 0 0	X_4
1 1 1	X_5
1 1 0	X_6



$\bar{L}$ = 1/3 + 1/3 + 1/2 + 1/2 + 1/2 + 1/2 = 2 2/3 binit/symbol

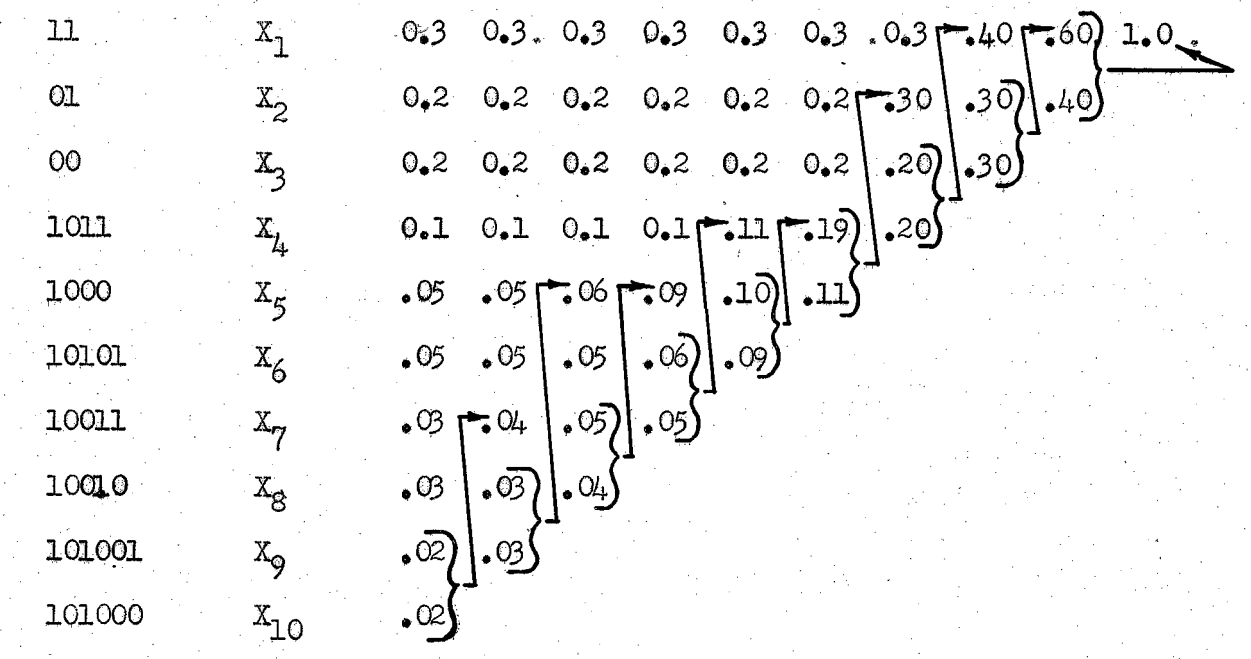
H(X) = log 6 = 2.58 bits/symbol

$H_c(X) = \frac{2.58}{2.67}$ = 0.967 bit/binit

η_c = 96.7%

Example 2.4-3

Determine the Huffman code for the source of Example 2.2-2.



$$\bar{L} = 0.6 + 0.4 + 0.4 + 0.4 + 0.2 + 0.25 + 0.15 + 0.15 + 0.12 + 0.12$$

$$= 2.79 \text{ binits/source symbol}$$

$$H(X) = 2.743 \text{ bits/source symbol}$$

$$\eta_c = \frac{2.743}{2.79} \times 100 = 98.5\%$$

Note that this gives an efficiency slightly higher than that for the Shannon-Fano code previously considered.

2.5 Additional Techniques for the Noiseless channel

In the previous discussions it has been assumed that the code having the greatest efficiency, for a given source, is the best code. This is a valid assumption when the cost, in time or money, involved in transmitting a 0 is the same as that for a 1. For this condition, the total cost involved in transmitting a message is minimized when the coding efficiency is maximized. However, when the code symbols have unequal cost the maximization of η_c, as

as previous defined, does not give the least cost encoding. Under this condition the Huffman procedure is no longer optimum and other techniques must be considered. Blackman (12) and Marcus (13) have considered this problem giving results that are extensions of the Shannon-Fano and Huffman procedures. Their methods, however, do not necessarily give minimum cost encoding. A recent article by Karp (14) describes such a technique which involves the use of digital computer. Because of the complexity of this procedure reference should be made to the article for specific details.

An additional situation in which Huffman encoding can not be used occurs when the encoding is to be done in such a manner that the alphabetical order of the source symbols is maintained in the code words. This might occur, for example, when English text is to be encoded for storage in a computer memory. Gilbert and Moore (15) have developed a technique for encoding such sources. When applied to the English alphabet this technique results in an average code word length of 4.1978 binits/letter as compared to the minimum possible (Huffman code) of 4.1195 binits/letter. The procedure for determining these codes, however, is considerably more complex than that for the Huffman code.

CHAPTER 3

CODING FOR THE NOISY CHANNEL

3.1 Introduction

The previous chapter considered coding for the noiseless channel. The techniques discussed represent methods for approaching the information transmission rate given by Shannon's first fundamental theorem. In most practical situations, however, the entire channel will not be noise free and the second fundamental theorem must be applied. The coding techniques discussed in this chapter represent various approaches to the realization of the information and error rates given by this second theorem.

Unfortunately there is at present no single technique, analogous to the Huffman procedure for the noiseless channel, that gives a maximum information rate and a minimum error rate. There are instead a number of procedures, each having their own advantages and disadvantages, that have been proposed as a solution to this problem. The better known and more readily explained of these techniques will be discussed in this chapter. It should be emphasized, however, that a large amount of work remains to be done in this area since the techniques presented all represent essentially trial-and-error solutions to the coding problem.

In the previous chapter it was shown that the output of a discrete source could be encoded into binary digits (binits) in such a manner that the resulting probabilities for a 0 and a 1 were as nearly equal as desired. Thus this chapter can consider, without loss of generality, only a binary source for which the symbol probabilities are equal. The block diagram resulting from this approach is given in Fig. 6.

If a sequence of 0's and 1's are transmitted over a noisy binary channel some will be received in error. Since the source binits are assumed to be pro-

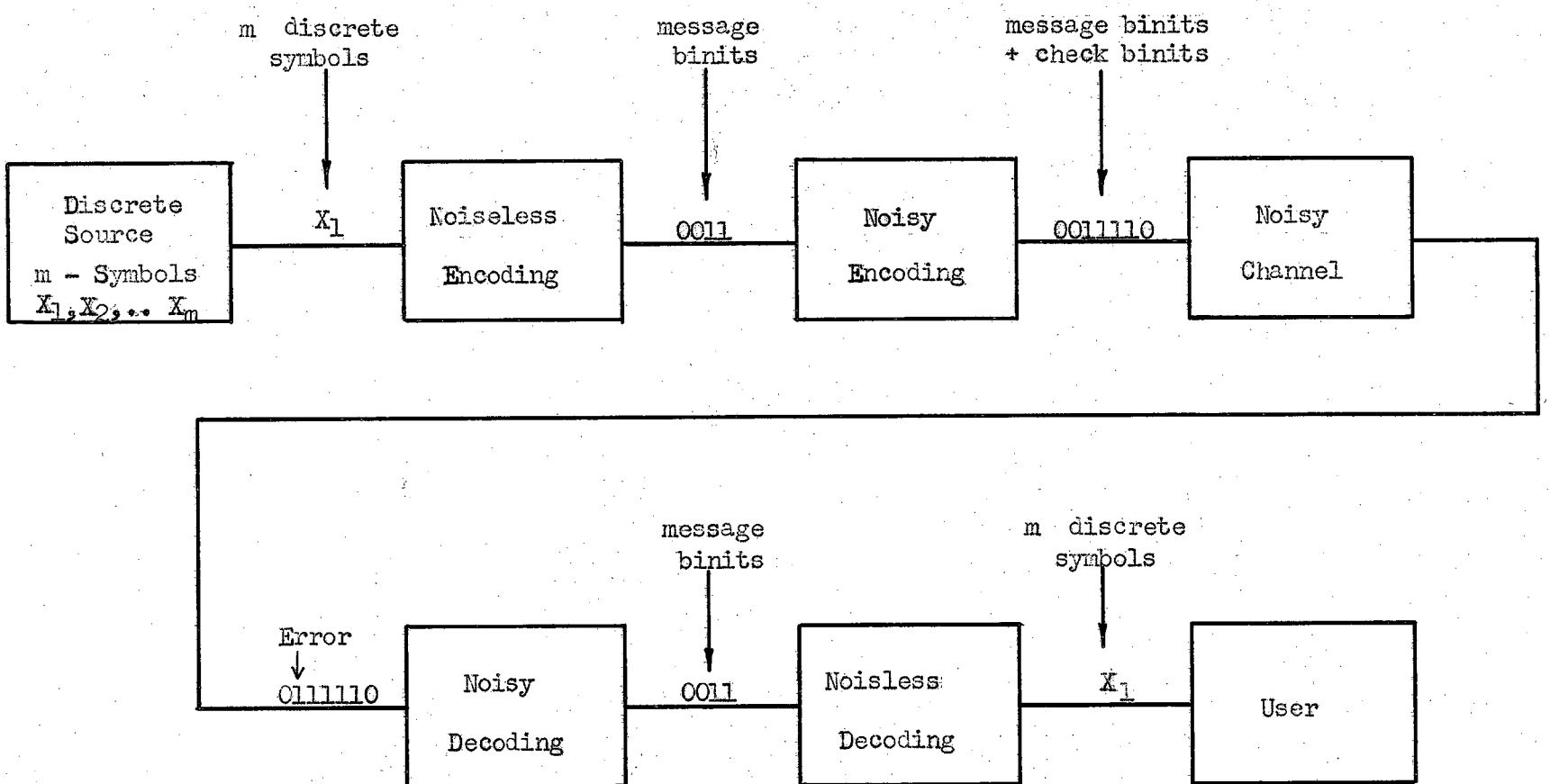


Fig. 6 - Coding For The Noisy Channel

duced independently, all sequences of binits will be equiprobable and there will be no way in which the erroneous digits can be detected.

One method of alleviating this problem is to transmit each message digit an odd number of times and to select at the receiver the digit occurring most often in each group. For example, assume the sequence to be transmitted is 00101110 and that three digits are to be transmitted for each message digit. The transmitted sequence is thus

000 000 111 000 111 111 111 000

Assume the transmission errors are such that the received sequence is

100 010 011 000 101 110 111 010

Assigning to each successive group of three binits the symbol appearing most often in the group yields the original transmitted sequence. Thus this technique gives error free transmission when only one error occurs within an individual group. Note, however, that to obtain this improvement in error rate it has been necessary to reduce the rate of transmitting message digits by a factor of one-third. Proper selection of the redundant digits allows more efficient error correction than that illustrated. However, the selection of these binits in an optimum manner represents the major problem in coding for the noisy channel.

This example illustrates an important general characteristic of coding for the noisy channel, namely, to be able to detect and/or correct an error at the receiver it is necessary that redundant binits be inserted into the message at the transmitter. In the above example the second and third binits in each group are redundant since they are uniquely determined by the first binit. These redundant digits contain no information. Their effect is thus to reduce the average information, or entropy, per binit of the transmitted sequence. Because of this it can be stated that for error detection and/or correction to be

possible it is necessary that the average entropy of the transmitted binits be less than 1 bit/binit. This should not be too surprising since Shannon's second fundamental theorem states that the average entropy per binit for the digits supplied to the channel must be less than the channel capacity, C, if error free transmission is to be theoretically possible.

A second important characteristic of coding for the noisy channel is the encoding of groups of message digits. In most codes groups of, say, m, message binits are encoded by inserting k redundant digits to give a code word of $n = m + k$ binits. Such codes in which all code words are of equal length are commonly called block codes. In the above example $m = 1$, $k = 2$ and $n = 3$.

In summary, the two important properties of codes for use with a noisy channel are as follows.

1. The probability of error for a received code word. If the coding is to be of value this must be less than the probability of error for the message sequence when it is transmitted without coding.
2. The ratio of message binits, m, to total binits, n, in a code word. This ratio can never exceed, C, the channel capacity, but should be close to it for efficient transmission. At present, few codes approach this ideal while simultaneously giving a low probability of error.

Before proceeding to the discussion of specific coding techniques the following definitions pertaining to coding for the noisy channel are given.

Memoryless channel - A channel in which the probability of error for a received binit is independent of the occurrence of previous errors.

Parity check digits - A more descriptive term that means essentially the same as the term redundant digits used above.

Code word - A sequence of n binits composed of both message digits and parity check, or simply, check digits.

Length of a code word - The number of binits in a code word. Usually all

code words in a given code are of equal length.

Weight of a code word - The number of 1's in the word.

Block code - Any code in which all code words are of equal length, n.

Group - A collection of elements or symbols having a specific mathematical property. This term will be defined more precisely in section 3.3.2 and is given here only to indicate that it now has a specific mathematical definition.

Group code - A binary code in which the code words have the group property.

Systematic code - An n binit block code in which m digits are information digits and k = n - m digits are parity check digits.

Linear code - A mathematical term for n-ary (binary, trinary, etc.) codes having a specific property. For binary codes the terms linear code and group code are synonymous.

3.2 Hamming Codes

The error detecting and error correcting codes discovered by Hamming (15) represent the first useful coding techniques for the noisy memoryless channel. The work of Hamming is best considered in four parts as follows.

1. Coding to provide for single error detection, i.e., SED codes.
2. Coding to provide for single error correction, i.e., SEC codes.
3. Coding which allows single error correction plus double error detection, i.e., SEC-DED codes.
4. Certain conditions required of code words to obtain higher orders of detectability and correctability.

All of the following results are based upon the assumption of a binary source with equiprobable symbols, a binary symmetric channel (BSC) with $P_o < 1/2$, and the use of equal length code words.

3.2.1 SED Codes

Hamming's SED codes for n binit code words are readily determined in the following manner: In the first n-1 positions are placed message digits. In the n th position a 0 or a 1 is placed so that there is an even number of 1's in the

total code word. The resulting code word allows single error detection (actually odd error detection) since any single (odd) error would result in an odd number of 1's in the received code word. Observe that all even errors go undetected.

Since the ratio of message digits to total digits is $m/n = 1 - 1/n$ it might appear desirable to make n as large as possible so as to obtain the maximum transmission of message digits. However, as n increases the probability of two or more errors, and thus an undetected error, increases. Thus when a maximum probability of an undetected error is specified there is an upper limit on n.

Example 3.2.1-1

For a BSC in which $P_o = 10^{-2}$ determine the value of n for a Hamming SED code that will make the probability of an undetected error approximately 10^{-3}. Compare this to the probability of an undetected error without coding.

For the SED code the probability of an undetected error, P(UDE), in a code word is simply the probability that an even number of errors will occur. Thus

$$P(UDE) = P(2 \text{ errors}) + P(4 \text{ errors}) + - -$$

For a BSC the probability of a particular set of two errors out of n transmitted digits is $P_o^{\,2} (1-P_o)^{n-2}$. There are a combination of n digits taken 2 at a time, $\binom{n}{2}$*, different ways in which two errors can occur. Thus the total probability of two errors in n digits is

$$\binom{n}{2} P_o^{\,2} (1-P_o)^{n-2}$$

Similar reasoning follows for 4, 6, 8 - - errors. Thus

---

* $\binom{n}{r} = \dfrac{n!}{r!\,(n-r)!}$

$$P(UDE) = \sum_{\substack{i=2 \\ i\text{-even}}}^{n} \binom{n}{i} P_o^{\,i} (1-P_o)^{n-i}$$

Use of the binomial expansion (pp 51-52, Ref. 8) allows this to be written as

$$P(UDE) = 1/2 \left[1-2(1-P_o)^n + (1-2P_o)^n\right] \qquad (28)$$

or

$$P(UDE) = 1/2 \left[1-2(.99)^m + (.98)^n\right]$$

Substituting values of n gives

n	P(UDE)
3	.00057
4	.00112
5	.00229

Thus a value of n=4 meets the specified error probability.

Without coding, an undetected error will occur whenever a message is received incorrectly. Thus

$$\begin{aligned} P(UDE) &= 1-P(\text{no errors}) \\ &= 1-(1-P_o)^n \\ &= 1-(.99)^4 \\ &= 0.03936 \end{aligned}$$

The probability of an undetected error has thus been reduced by a factor of more than 30 while reducing the information rate by only 25%.

The type of check used above to determine the presence of a single error is called a parity check and will be used throughout the discussion of coding for the noisy channel. The above discussion used an even parity check. Had an

odd check been used, the n th digit would have been chosen so as to make an odd number of digits in the code word. This report will use only even parity checks. It should be noted that the parity check need not always involve a check over all of the message digits but may check only a portion of these. The codes of the following sections illustrate this point.

3.2.2 SEC Codes

Hamming's SEC code allows the correction of any single error that occurs within a particular code word. However, when two or more errors occur this procedure can cause additional errors to be created in the decoding process. Thus it is necessary that these codes be used only in siutations where the probability of two or more errors is negligibly small.

The construction of SEC code proceeds by first assigning m of the n binits in a code word to be information digits. For a given n, m will be considered to be fixed. The specific location of three digits will be determined later. The remaining k = n-m positions are assigned to be parity check digits. The values of the check digits will be determined in the encoding operation by even parity checks over the selected information places. The following discussion will determine how these parity checks are to be made.

Consider the situation in which a code word has been received either with or without a single error. Assuming the parity check rules to be known, they can be applied in order with the condition that for each time the parity check assigns the value observed in the corresponding check position a 0 will be recorded while a 1 will be recorded when the two values disagree. Since there are k check digits, a sequence of k 0's and 1's will be obtained. When this sequence is written from right to left it can be considered as a binary number. This number is called the checking number and shall be required to give the position of any single error in the code word. The zero value of this number shall mean that no error has occurred. Since the code words are n binits long the checking number must be

capable of specifying n + 1 different events. The relation between the number of check digits, k, and n is thus

$$2^k \qquad n + 1$$

n	maximum m	k = n-m
1	0	1
2	0	2
3	1	2
4	1	3
5	2	3
6	3	3
7	4	3
8	4	4
9	5	4
10	6	4
11	7	4
12	8	4

TABLE I

With this result the values of Table I may be determined. This table gives, for a specified n, the maximum number of message digits that can be used while retaining the capability for correcting single errors.

Although it appears from this table that more information can be transmitted by using larger values of n it should be remembered that the probability of two or more errors also increases with n. Thus an upper bound on n will also exist for SEC codes when the maximum probability of error is specified.

It is now necessary to determine the parity check rules that will allow the operation described to be obtained. The digits of the checking number are to be obtained by applying the parity check rules in order and recording, from right to left, the resulting sequence of 0's and 1's. Since the checking number is to give the position of any single error in a code word, any position in the code having a 1 on the right side of its binary representation must cause the first parity check to fail. The binary representations of the various positions are as follows.

Position	Binary representation
1	0001
2	0010
3	0011

4	0100
5	0101
6	0110
7	0111
8	1000
9	1001

Observe the right hand binit is a 1 for all odd positions. Thus the first parity check must be over positions 1, 3, 5, 7, 9 - - -. Similar reasoning indicates that the second parity check should be over all positions having a 1 in the second digit from the right. From above these are 2, 3, 6, 7, 10, 11, - - -. Likewise the positions for the third parity check are 4, 5, 6, 7, 12, 13, 14, 15 etc.

These results indicate the positions to be checks in each of the successive parity checks. It remains to determine exactly where in the n binit sequence the k parity check digits should be placed. Observe that by placing the check digits in positions 1, 2, 4, 8 etc. each check digit will be involved in only one of the parity check operations determined above. Although this condition is not required to obtain the SEC property, it greatly simplifies the decoding procedure. Thus these positions will be used. Table II summarizes these results.

Parity Check No.	Location of Check Digit	Positions Checked
1	1	1,3,5,7,9,11---
2	2	2,3,6,7,10,11---
3	4	4,5,6,7,12,13,14,15 ---
4	8	8,9,10,11,12,13,14,15,24,25---

TABLE II

Example 3.2.2-1

As an illustration of the preceeding results consider the Hamming SEC code of n=7. Table I shows that there are 4 message digits and 3 check digits per code word. Table II shows that the first parity check is over positions 1, 3, 5, 7 and determines the value for the digit in position 1; the second parity check is over positions 2, 3, 6, 7 and determines the value in the second position; while the third check is over 4, 5, 6, 7 and determines the value in position 4. The information positions in this code are thus 3, 5, 6, 7 allowing a total of 2^4 = 16 different code words. As an example of the application of these check rules assume that the digits in positions 3, 5, 6, 7 are 1, 0, 1, 1 respectively. The first parity check rule thus requires that a 0 be placed in position 1. Likewise, the second and third parity check rules require a 1 and a 0 in positions 2 and 4 respectively. The resulting code word is 0110011. Table III givesthe code words when all 16 possible message sequences are considered.

To demonstrate the error correcting capability of this code assume that code word 6 has been received as

0110101

Applying the first parity check to positions 1, 3, 5, 7 indicates that the digit in position 1 should be a 1. Since the received digit is a 0 the first digit of the checking number is 1. Similarly the second parity check predicts a 0 for position 2 which disagrees with the received digit. Thus a 1 is written to the left of the 1 obtained above.

Finally, the third check predicts a 0 for position 4 which agrees with the received value. The resulting check number is thus

0 1 1

which correctly indicates that position 3 is in error.

To demonstrate the effect of 2 errors consider the situation in which

code word 9 is received as

1 1 1 1 1 0 0

Applying the parity check rules gives the checking number

0 0 1

Thus the decoder would change the digit in position 1 to a 0 causing a new error to be created. This demonstrates that the probability of two or more errors should be negligibly small when a SEC code is used.

Code Word	Letter Position						
	1	2	3	4	5	6	7
1	0	0	0	0	0	0	0
2	1	1	0	1	0	0	1
3	1	1	0	1	0	1	0
4	1	0	0	0	0	1	1
5	1	0	0	1	1	0	0
6	0	1	0	0	1	0	1
7	1	1	0	0	1	1	0
8	0	0	0	1	1	1	1
9	1	1	1	0	0	0	0
10	0	0	1	1	0	0	1
11	1	0	1	1	0	1	0
12	0	1	1	0	0	1	1
13	0	1	1	1	1	0	0
14	1	0	1	0	1	0	1
15	0	0	1	0	1	1	0
16	1	1	1	1	1	1	1

TABLE III

Since a SEC code is used to reduce the probability of a code word being received in error it is useful to determine the amount by which this probability is reduced. For this code, the probability of correct reception is the probability that either no errors or a single error occurs. From the results of example 3.2.1-1 this is given by

$$P(\text{no error}) = (1-P_o)^n + \binom{n}{1} P_o (1-P_o)^{n-1}$$

$$= (1-P_o)^n + n P_o(1-P_o)^{n-1}$$

Since

$$P(\text{no error}) = 1 - P(\text{error})$$

$$\triangleq 1 - P_e$$

the desired result is

$$P_e = 1 - (1-P_o)^n - n P_o(1-P_o)^{n-1}$$

Without coding, m digits would be transmitted in each word.

The corresponding probability of error is thus

$$P_e = 1 - (1-P_o)^m$$

Considering specific values of n = 7, m = 4, and $P_o = 10^{-2}$ gives the following results.

With coding

$$P_e = 1 - (.99)^7 - 7(.99)^6 = 0.00195$$

Without coding

$$P_e = 1 - (.99)^4 = 0.03936$$

Thus, SEC coding has reduced the probability of an uncorrected error by a factor of 15 while reducing the information rate by less than a half (the rate with coding is essentially 4/7 bits/binit giving a reduction of 43%).

Observe here that a penality has been paid to obtain the error correcting capability. In example 3.2.1-1 a reduction in the probability of an undetected error of 30 times was obtained for only a 25% reduction in the information rate.

The primary advantage of the SEC codes as compared to the SED codes of the last section lies in the fact that SEC codes correct instead of only detecting the most probable of the received errors. Thus in situations where message retransmission is not possible SEC codes can be used to improve the reliability of transmission. However, a penality is paid for this capability since fewer message digits can be transmitted in each code word. Because of this the SED codes can be of value when a feedback channel is available. In the following section a code is discussed which has both error detecting and error correcting capabilities.

3.2.3 SEC-DED codes

In some cases where a low capacity feedback channel is present it might be advantageous to correct the most probable single errors by means of a SEC code and to provide for message retransmission via the feedback channel when more than a single error occurs. Hamming has suggested such a code which is obtained from the SEC code by simply adding an additional digit that is an even parity check over all previous digits. For the code words of Table III this involves adding an 8th column having the following digits

0
0
1
1

1
1
0
0

1
1
0
0

0
0
1
1

The operation of the SEC-DED code is best explained by considering several cases.

1. No errors occur. In this case all parity checks are satisfied. For example if the sequence

1 0 1 1 0 1 0 0

is received the check number is found to be 0 0 0. Since an even number of 1's are present the last parity check is also satisfied. Thus when the last parity check is satisfied and the checking number is zero it is concluded that no errors have occurred. (Actually this is not completely true since the errors could be such as to change one code word into another. The probability of this occurring, however, is considerably less than the corresponding probability of a single or double error.)

2. A single error occurs. For this situation the last parity check will fail. The resulting checking number will indicate the position of an error with a zero indicating an error in the last check position. For example, assume that the sequence

0 0 0 0 1 1 1 0

is received. The checking number is found to be 100 and the last check fails. Thus the error is in position 4.

3. Two errors occur. In this situation the last parity check is satisfied but a checking number is obtained. This indicates that two errors have occurred but gives no information regarding their location. Thus, if the received sequence is

1 0 0 0 1 0 1 1

the last check is satisfied and the checking number is 0 1 1. However, the errors occurred in positions 4 and 5 and the checking number is of no use.

4. More than two errors occur. In this case no useful information is obtained and if the number of errors is odd (so that the last check is satisfied) it is possible that the resulting checking number will cause an additional error to be created.

For the SEC-DED code the probability that a received code word is either correct or known to be incorrect is simply the probability that either 0, 1 or 2 errors have occurred. Thus the probability, P_e, of receiving an erroneous word and not knowing that it is incorrect is

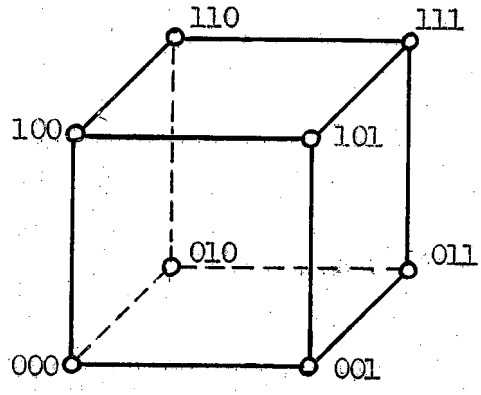
$$P_e = 1 - P(\text{no errors}) - P(1 \text{ error}) - P(2 \text{ errors})$$

$$= 1 - (1 - P_o)^{n+1} - (n + 1) P_o (1 - P_o)^n - \frac{n(n+1)}{2} P_o^2 (1 - P_o)^{n-1}$$

For the values considered previously, i.e., $n = 7$, $m = 4$ and $P_o = 10^{-2}$ this gives a P_e of less than 10^{-4}. Thus the use of a SEC-DED code has reduced the probability of an undetected error by approximately 500 times while causing a reduction in the information rate of 50%.

3.2.4 Code Requirements for Larger values of Detecting and Correcting Capability

In his article (15) Hamming introduced a geometrical model that allows some conditions to be specified for codes that are to either detect or correct more than two errors. This model consists of identifying the sequences of 0's and 1's in each code word with a point in n-dimensional space. For large values of n this ia a rather abstract concept that is of value primarily to the mathematician. However, the case for $n = 3$ can be readily considered and illustrates the basic concept. For $n = 3$ the $2^3 = 8$ possible code words can be associated with the points of a 3-dimensional cube in the following manner.



Let the distance between any two of these points, say X and Y, be D(X,Y). From the above figure it is clear that the distance between any two points is equal to the number of digits in which the two corresponding code words differ. Thus, for example, the distance between the parts 101 and 010 is 2. This corresponds to the number of edges of the cube that must be traversed in going from one point to the other.

Using this concept it is apparent that the effect of an error in the transmission of a code word is to move the code point to a new location. Thus if all points are used as code words the occurrence of an error can not be detected. However, if code words having a minimum distance of 2 units are chosen a single error will cause a code point to be moved in only one coordinate to a point that is not defined as a code word. This allows a single error to be detected. From the above model one such set of symbols would be

0 0 0

0 1 1

1 0 1

1 1 0

or, equally well.

0 0 1

0 1 0

1 0 0

1 0 0

Observe now that if the minimum distance between code words is at least 3 units then any single error will leave the displaced point nearer to the correct point than to any other code point. This means that any single error can be corrected. This can be generalized to larger minimum distances with the following results.

Minimum Distance	Resulting code
1	No error detection or correction possible
2	Single error detection (SED)
3	Single error correction (SEC)
4	Single error correction - double error detection(SEC-DED)
5	Double error correction (DEC) or, Single error correction - triple error detection (SEC-TED) or, quadruple error detection (QED)
6	Double error correction - triple error detection (DEC-TEC) or, etc.

The procedures discussed in sections 3.2.1, 3.2.2 and 3.2.3 are merely specific techniques for determining code words having a minimum distance of 2, 3, and 4 respectively. Thus all the code words of Table III will be observed to have a minimum distance of at least 3 units.

The determination of a set of code words which is as large as possible while maintaining a specified minimum distance represents an unsolved problem for distances greater than 4 units. These results, however, give conditions that must be met by any coding scheme that may be devised.

3.3 Slepian Group Codes

3.3.1 Introduction

The work of Slepian (16), which is a generalization of results obtained

earlier by Hamming and Reed-Muller (17), represents a major contribution to the field of coding theory. In essence Slepian showed the Hamming and Reed-Muller codes to a subclass of a larger class of codes called group codes. The group codes have several special features of practical interest. In particular, (1) the encoding scheme is relatively simple to instrument due to the placement of the check digits in the last k positions of the code word; (2) the decoder - a maximum likelihood detector-is the best possible theoretically (i.e. it gives the lowest possible P_e for a given code) and is reasonably easy to instrument; and (3) in many cases of practical interest the codes are the best possible theoretically (i.e., no other code of any type which is composed of the same number of equal length n-binit code words has a lower P_e).

The Slepian group codes do not, however, allow transmission at a rate near the channel capacity with an arbitrarly small error rate. Since Elias (17) has shown that such codes do exist it is clear that additional work remains to be done. At present nearly all of the block codes being studied are a subclass of the general group codes discussed by Slepian.

As with the Hamming codes, all discussion of the Slepian codes is based upon the assumption of a memoryless binary symmetric channel (BSC) with $P_o < 1/2$ and equiprobable binary source symbols.

The following section will discuss the mathematical properties that are required for an understanding of subsequent work.

3.3.2 Definition and Properties of a Group.

The following discussion of the definition and properties of a group is not as rigorous nor as complete as that given by mathematicians. The information presented, however, will allow the fundamental properties of group codes to be understood.

In terms of binary words (i.e., sequences of n binary digits) a group is defined as follows:

Definition: A collection of binary words is said to form a group if the product* of any two words is also a member of this collection and if the collection contains the identity element (this element, I, is defined to be the all-zero n binit sequence).

From this definition it is clear that the 2^n possible sequences of n binits form a group since the product of any number of the sequences is another sequence. This group is denoted by B_n and contains 2^n words or elements.

Other groups having less than 2^n elements can also be found from these binary words. For example the words

0 0 0

1 0 0

0 0 1

1 0 1

form a group since the product of any number of the words is also contained in the group. (Note that any word multiplied by itself yields the identity element). Groups of this type are contained in the larger group B_n and are defined to be a subgroup of B_n. The group codes investigated by Slepian are in this category.

3.3.3 Definition of a Group Code

An n-place group code is defined to be a collection of 2^m $(m<n)$ n binit code words that form a group as defined above. Since the group B_n contains all 2^n possible sequences of n binits, all n-place group codes are subgroups of B_n.

---

* The product of two binary words is defined as follows. Let $A = a_1, a_2, a_3, - - - - a_n$ and $B = b_1, b_2, - - - b_n$ be two n-digit binary words. Then the product AB is defined as

$$AB = a_1 + b_1, a_2 + b_2, - - - a_n + b_n$$

where + denotes addition modulo 2, i.e., $0 + 0 = 1 + 1 = 0$, $0 + 1 = 1 + 0 = 1$. Thus if A = 011000 and B = 110110, AB = 101110.

For simplicity in subsequent discussion such codes will be deonoted as (n,m)-codes.

Slepian has shown (pp 219-221, Ref. 16) that there are exactly

$$N(n,m) = \frac{(2^n-2^0)\,(2^n-2^1)\,(2^n-2^2)\,\cdots\,(2^n-2^{m-1})}{(2^m-2^0)\,(2^m-2^1)\,-\,-\,-\,-\,-\,-\,(2^m-2^{m-1})} \tag{29}$$

different subgroups of B_n having 2^m elements or words and thus N(n,m) possible (n,m)-codes. Some values of N(n,m) are given in Table IV below.

n	m			
	1	2	3	4
3	7	7	1	0
4	15	35	15	1
5	31	155	155	31
6	63	651	1395	651
7	127	2667	11,811	11,811
8	255	10,795	97,155	200,787

TABLE IV

Observe that as n and m become large the number of possible subgroups increases rapidly. Since the Slepian group codes are to be selected from these subgroups, the problem of choosing the best code for a given n and m becomes quite difficult for large n and m.

Example 3.3.3-1

For n = 3, m = 2 Table IV shows that there are N(3,2) = 7 possible (3,2)-codes. Trial and error methods show that these are as follows:

	(3,2)-Code No.						
	1	2	3	4	5	6	7
	000	000	000	000	000	000	000
Code	001	011	001	010	001	010	011
Words	010	101	100	100	110	101	100
	011	110	101	110	111	111	111

TABLE V

The determination of these codes for larger values of n and m is not a simple problem.

Assuming that a (n,m)-code has been chosen and the 2^m words determined, information is transmitted with this code by selecting blocks of m message digits and associating these in a one-to-one manner with the 2^m code words. Then as each block of m message digits is received, the corresponding block of n code digits is transmitted over the channel. Due to noise on the channel some of the digits in the received code word will be in error. The next problem is thus concerned with the method for correcting these errors using the known property that the transmitted words formed a group.

3.3.4 Detection of Group Codes

It has been stated that the transmitted code words form a subgroup of B_n. This means that only 2^m of the possible 2^n n-binit sequences are transmitted. However, due to noise on the channel it is possible to receive any of the 2^n n-binit sequences. Thus, the detection process must involve associating a number of received words with each of the transmitted words in such a manner that the probability of error is minimized. Slepian has shown (pp 222-223, Ref. 16) that the optimum detection method (i.e., it gives the least probability of error) is as described in the following paragraphs.

Let the words of a specific (n,m)-code be A = I = 000 - - 0 (I is the identity

element), A_2, A_3, - - - A_u, where $u = 2^m$. The group B_n (i.e. the collection of all 2^n possible received words) can be developed from this subgroup as shown below

$$B_n = \begin{array}{ccccc} I & A_2 & A_3 & \cdots & A_u \\ S_2 & S_2 A_2 & S_2 A_3 & \cdots & S_2 A_u \\ S_3 & S_3 A_2 & S_3 A_3 & \cdots & S_3 A_u \\ \vdots & & & & \\ S_v & S_v A_2 & S_v A_3 & \cdots & S_v A_u \end{array}$$

where $u = 2^m$, $v = 2^{n-m}$, and $S_i A_j$ is the product of n-binit sequences as previously defined. Observe that there are $u = 2^n$ elements, or words, in this array. It can be shown (pp 17, Ref. 19) that this array contains every element of B_n once and only once if the words S_2, S_3 - - S_v are chosen in the following manner: For S_2 choose any code word not contained in the first row, for S_3, any word not contained in the first two rows, etc. The various rows, other than the first, in this array are called cosets and the first word, i.e., S_2, S_3 -- S_v, in each row is called a coset leader.

It can also be shown (p 436, Ref. 8) that if a coset leader is replaced by any element in the coset, the same coset will result. Thus, the two collections of words

$$S_i, \quad S_i A_2, \quad S_i A_3, \quad - -, \quad S_i A_u$$

and

$$S_i A_k, \quad (S_i A_k) A_2, \quad (S_i A_k) A_3, \quad - - (S_i A_k) A_u$$

are the same. (Note that this does not imply that words in the same position of each coset are identical but only that the same words are contained somewhere in

each coset).

The weight, W_{ij}, of an element in the above array is defined to be the number of 1's in the n-binit word located in the i th row and the j th column. With this definition and in view of the preceeding paragraph it is possible to rearrange the array for B_n so that the coset leaders will have the minimum weight in each coset. Such an array is defined to be a standard array.

Example 3.3.4-1

A standard array for B_4 when developed according to the specific (4,2)-code 0000, 1100, 0011, 1111 is as follows

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

In the last row any of the elements could have been chosen as coset leaders since all are of equal weight 2. In the third row either 0100 or 1000 could have been used, while either 0001 or 0010 could have been used in the second row. It should be clear that many such standard arrays could be obtained by choosing different coset leaders having the same weights.

The detection scheme for a group code used with a BSC is now as follows: When a word, say A_j, is transmitted, the received word can be any element in B_n. If the received word lies in column i of the standard array the detector will indicate that A_i has been transmitted. For example, the array of Example 3.3.4-1 shows that the received word 0111 will be produced by the detector as 0011, 0110 will be produced as 0000, etc. Since any word in a standard array is at least as close to the code word at the top of its column as it is to any other transmitted code word (pp 222-223, Ref. 16) this detection scheme represents maximum likeli-

hood detection, i.e., the detected symbol is the one most likely to have been transmitted. It will be shown later that, for a given group code, this scheme gives the lowest possible probability of error, i.e., no other method has a greater average probability that the transmitted word be correctly produced by the detector.

Observe that this detection scheme requires a knowledge of all 2^n possible received words. This means that detection equipment requirements will grow exponentially with increasing code length. Since in many practical situations large code words are required this represents a serious limitation. A later section of this report will discuss an alternate mathod for obtaining maximum likelihood detection that does not have this characteristic.

3.3.5 Probability of Error for Group Codes

Let an arbitrary code word that is to be transmitted over a BSC be denoted by A and the resulting received word by T. Note that each of these words are n-plane binary sequences. The digits of T differ from those of A only in the positions where an error occurred due to noise on the channel. Thus, a new word, N can be defined as N = AT which will have a 1 in each position in which the digits of A and T differ, i.e., in each position in which an error occurred. This word, is also an element of B_n and serves as a record of the noise on the channel during the transmission. (For example, if A = 1010010 and T = 1110110 then N = AT = 0100100 indicating that an error occurred in positions 2 and 5.) From previous results it is known that the probability of N being any particular element of B_n is

$$P_o^{\,w} (1 - P_o)^{n-w}$$

where w is the weight of N.

Consider now the case of transmitting with a particular (n,m)-code and assume that the standard array for this code is known at the receiver. If the

maximum likelihood detection scheme is used, a transmitted letter, A_i, will be produced without error if and only if the received word is of the form S_jA_i, i.e., the received word must lie in the column of the standard array having A_i as its head. Thus there will be no error only if the noise on the channel represented by N, is one of the coset leaders. In view of this the probability of correct detection, $1 - P_e$, is just the sum of the probabilities that N is a coset leader, i.e.,

$$1 - P_e = \sum_{i=o}^{v} P_o^{w_i} (1 - P_o)^{n-w_i} \tag{30}$$

where w_i is the weight of S_i. Since, for a fixed n, the term

$$P_o^{w_i} (1- P_o)^{n-w_i}$$

is minimum when w_i is minimum ($P_o < 1/2$) and since the coset leaders of a standard array have minimum weight the probability of correct detection given by Eq. (30) is as large as possible. Thus, as previously indicated, maximum likelihood detection gives, for a particular code, the greatest average probability of correct detection. However, for a specified n and m there are N(n,m) possible group codes and this result tells nothing about which of these will have minimum P_e. This problem is considered in a later section.

Example 3.3.5-1

For the (4,2) code of Example 3.3.4-1 the probability of correct detection is

$$1-P_e = (1-P_o)^4 + P_o(1-P_o)^3 + P_o(1-P_o)^3 + P_o^2(1-P_o)^2$$

Assuming $P_o = 10^{-2}$ gives

$$P_e = 1-(.99)^4 + 0.02\ (.99)^3 + 10^{-4}(.99)^2$$

$$= .01985$$

Without coding

$$P_e = 1 - (1-P_o)^m = 1-(.99)^2$$

$$= .0199$$

Thus, in this example nothing has been gained by coding. In fact a loss is involved since the information rate with coding is only 50% of that without coding. This illustrates that coding can not be used indiscriminately to obtain a reduction in P_e.

In general such a situation would be remedied by encoding larger blocks of message digits.

3.3.6 Generation of Group Codes by Parity Checks

An encoding method has been suggested in Section 3.3.3 in which the $2^m(n,m)$-code words are listed in a code book along with the 2^m possible m binit sequences. The sequence of binits from the message source is then divided into blocks of m binits and the corresponding code word determined from the code book. The resulting m binit code word is transmitted over the channel. This procedure suffers from the fact that 2^{m+1} words must be stored in the encoding device. Thus, storage requirement will increase exponentially with increasing message block lengths. A simpler encoding procedure, giving rise to only a linear increase in equipment requirements, involves the generation of code words by suitable parity checks over the message digits in a manner similar to the Hamming procedure. Two concepts are required before this approach can be discussed: that of a systematic code and that of equivalence.

In a systematic code the digits in any word can be divided into two classes:

(1) the information digits (there are m of these in a (n,m)-code), and (2) the check digits (n-m=k in number for the (n,m) code). All words in the code have the same information digit locations and the same check digit locations. The m information locations may be occupied by any of the 2^m m-digit binary sequences. The digits in the check locations are determined by fixed parity checks over prescribed combinations of the information digits. Thus the Hamming codes are one example of systematic codes.

Example 3.3.6-1

Consider the (4,2)-code given by 0000, 1100, 0011, 1111. Assume that positions 2 and 3 are to be information positions. Appropriate parity checks over these positions will give the digits in position 1 and 4. Denoting a code word by X_1, X_2, X_3, X_4, the parity check rules can be determined by solving for the unknown constants in the following equations.

$$X_1 = A_1X_2 \oplus A_2X_3 \qquad \text{(a)}$$

$$X_4 = A_3X_2 \oplus A_4X_3 \qquad \text{(b)}$$

Substituting values from the second and third code words above gives the following simultaneous equations

$$\begin{aligned} 1 &= A_1 \cdot 1 \oplus A_2 \cdot 0 \\ 0 &= A_1 \cdot 0 \oplus A_2 \cdot 1 \end{aligned} \qquad \text{(a-1)}$$

$$\begin{aligned} 0 &= A_3 \cdot 1 \oplus A_4 \cdot 0 \\ 1 &= A_3 \cdot 0 \oplus A_4 \cdot 1 \end{aligned} \qquad \text{(b-1)}$$

Simultaneous solution of Eqs. (2-1) and (b-1) gives $A_1 = A_4 = 1$, $A_2 = A_3 = 0$. Thus the parity check rules are

$X_1 = X_2$

$X_4 = X_3$

If instead the information positions where chosen to be 1 and 3 the above procedure would yield for the parity check rules

$X_2 = X_1$

$X_4 = X_3$

Note that for this code positions 1 and 2 or 3 and 4 can not be used for information since only 2 numbers appear in each position, i.e. either 00 or 11.

Example 3.3.6-2

Consider the (5,3)-code 00000, 10001, 01011, 00111, 11010, 10110, 01100, 11101 and choose the information positions to be positions 1, 2, and 3. The general parity check equations to be solved are

$$X_4 = A_1X_1 \oplus A_2X_2 \oplus A_3X_3 \qquad (c)$$

$$X_5 = A_4X_1 \oplus A_5X_2 \oplus A_6X_3 \qquad (d)$$

Using the second, fifth, and seventh code words gives, for the simultaneous equations,

$$0 = A_1 \cdot 1 \oplus A_2 \cdot 0 \oplus A_3 \cdot 0$$

$$1 = A_1 \cdot 1 \oplus A_2 \cdot 1 \oplus A_3 \cdot 0 \qquad (c\text{-}1)$$

$$0 = A_1 \cdot 0 \oplus A_2 \cdot 1 \oplus A_3 \cdot 1$$

$$1 = A_4 \cdot 1 \oplus A_5 \cdot 0 \oplus A_6 \cdot 0$$

$$0 = A_4 \cdot 1 \oplus A_5 \cdot 1 \oplus A_6 \cdot 0 \qquad (d\text{-}1)$$

$$0 = A_4 \cdot 0 \oplus A_5 \cdot 1 \oplus A_6 \cdot 1$$

Simultaneous solution of these gives for the parity check equations

$$X_4 = X_2 \oplus X_3$$
$$X_5 = X_1 \oplus X_2 \oplus X_3$$

Two group codes are defined to be <u>equivalent</u> if one can be obtained from the other by permuting the digit locations. Thus in Example 3.3.3-1 code numbers 1, 3, 4 are equivalent; code numbers 5, 6, 7 are equivalent; and code number 2 is in a class by itself. In conjunction with this Slepian gives the following important results: (p 210, Ref. 16)

1. Every group code is a systematic code and vice versa.

2. Every (n,m)- code is equivalent to a (n,m)- code in which the first m places are information digits and in which the last n-m = k places are determined by parity checks over the first m places.

Because of these results it is now necessary to consider only (n,m)-codes in which the first m digits are information digits. The general expression for the k check digits then becomes

$$X_i = \sum_{j=1}^{m} \gamma_{ij} X_j \qquad i = m+1, - - - n \tag{31}$$

Here the summation is modulo 2 with the multiplication rules being 0·1 = 1·0 = 0·0 = 0, 1·1 = 1. The km values for γ_{ij} may be either 0 or 1 and define the particular (n,m)-code being used.

Using these results, group codes will now be specified by giving the parity check rules rather than by listing all 2^m code words. The encoding operation will then be performed by applying these check rules to blocks of m information digits in the order specified. The k check digits thus obtained will be added to the m information digits and the resulting n binit sequence transmitted as the code word.

Example 3.3.6-3

Consider the (6,3)-code. Suitable parity check rules are given by Slepian (Table III, Ref. 16) as

$X_4 = X_1 \oplus X_2$

$X_5 = X_1 \oplus X_3$

$X_6 = X_2 \oplus X_3$

Since m = 3, there are $2^3 = 8$ words in the (6,3)-code which are as follows.

Code word	Information Digits	Parity check Digits
1	000	000
2	001	011
3	010	101
4	011	110
5	100	110
6	101	101
7	110	011
8	111	000

3.3.7 Detection of Group Codes by Parity Checks

The detection method presented in section 3.3.4 is analogous to the code book encoding described above, i.e. the standard array lists all possible received words and assigns each of these to a transmitted word. As mentioned previously the disadvantage of this method lies in the fact that storage space for 2^n words must be provided at the decoder. Slepian has described a method for obtaining maximum likelihood detection by means of parity checks over the received code

words. This approach eliminates the need for storing all possible code words and results in a simplification of the detection equipment.

Consider the standard array for the group B_n which has been developed about a specific (n,m)-code. This code is assumed to have a set of parity check rules in the form of Eq. (31). For any word in the array, say T, these parity checks may be applied. The check digit resulting from the i th parity check may or may not agree with the digit in the i th position of T. If it does T satisfies the i th parity check and a 0 is recorded. Otherwise T fails the i th check and a 1 is recorded. Proceeding in this manner with all k parity checks results in a k digit binary sequence which is defined to be R(T), the parity check sequence of T. (In determining R(T) the digits are to be written from left to right as the parity checks are applied in order, starting with the check for position m + 1.) For example, using the parity check rules of Example 3.3.6-3 shows R(101001) = 100 since $X_1 + X_2 = 1 \neq X_4$, $X_1 + X_3 = 0 = X_5$ and $X_2 + X_3 = 1 = X_6$. Obviously, R(T) can be determined for any word in the array. Using this definition of R(T) Slepian (pp 224-225, Ref. 16) has proved the following theorem.

Theorem: Let $I, A_2, A_3, - - - A_u$, be a (n,m)-code and consider B_n to be developed in a standard array about this code. Let R(T) be the parity check sequence for a word T which has been formed in accordance with the parity check rules of the specified code. Then $R(T_1) = R(T_2)$ if and only if T_1 and T_2 lie in the same row of the standard array.

Example 3.3.7-1

Consider the (4,2)-code shown below

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

The parity check rules for this code can be shown to be $X_3 = X_1$, $X_4 = X_1 \oplus X_2$. Every word in the second row fails the first parity check (for the digit in position 3) and satisfies the second check. The parity check sequence is thus 10. In like manner the parity check sequence for row 3 is 01 and for row 4 is 11. By definition, all words in the first row satisfy the parity checks giving a parity check sequence of 00. The following relations can thus be established between the coset leaders and the parity check sequences.

$00 \rightarrow S_1 = 0000$

$10 \rightarrow S_2 = 0010$

$01 \rightarrow S_3 = 0100$

$11 \rightarrow S_4 = 1000$

Maximum likelihood detection can now be obtained in the following manner. When a word T is received it is subjected to the k parity checks of the code being used. This gives a parity check sequence R(T) which places T in a definite coset and identifies the coset leader, say S_i. The product S_iT is formed (S_iT is the word that would be at the head of the column containing T in the standard array) and produced as the detector output. The probability of error for the detected word, P_e, is as given by Eq. (30).

Using this detection scheme only $(2^m + 2^k - 1)$ words, plus the parity check rules, must be stored by the detector. For large n and m this represents a considerable reduction from the 2^n words required for the original scheme.

Example 3.3.7-2

Assume that the word 0001 of the (4,2)-code of Example 3.3.7-1 has been received. The parity check rules are $X_3 = X_1$, $X_4 = X_1 \oplus X_2$ giving a parity check sequence of 01. From above, this sequence corresponds to $S_3 = 0100$. The detected word is thus (0100) (0001) = 0101.

3.3.8 Determination of Groups Codes Having Minimum Probability of Error

The discussion to this point has assumed that the code words, or the parity check rules, for a given (n,m)-code where known. However, it was previously indicated that there are N(n,m) possible (n,m)-codes from which to choose. Since these codes are used for error correction it is reasonable to require that the (n,m)-code selected have a minimum P_e when compared to all other codes having the same n and m. These considerations give rise to the following questions.

1. Which of the N(n,m) different subgroups of B_n give a (n,m)-code, having a minimum P_e?

2. What is the value of the minimum P_e?

Unfortunately, the answers to these questions are not known for general values of n and m. Slepian has, however, determined the answers for several specific values.

His results are presented in Tables II and III of Reference 16 and in Tables T-4 and T-5 of Reference 8. These results will be discussed in this section. Additional details should be obtained from the references cited.

Eq. (30) gives the probability, $1 - P_e$, of correctly detecting a transmitted word as

$$1 - P_e = \sum_{i=o}^{v} P_o^{w_i} (1 - P_o)^{n-w_i} \qquad (32)$$

It will be recalled that $P_o^{w_i} (1 - P_o)^{n-w_i}$ is the probability that a coset leader will be of weight w_i while having a specific configuration. In general there will be several, say α_i, coset leaders having a weight w_i. Grouping these together allows Eq. (32) to be written as

$$1 - P_e = \sum_{i=o}^{n} \alpha_i P_o^{w_i} (1 - P_o)^{n-w_i} \qquad (33)$$

Since there are $2^{n-m} = v$ coset leader the relation

$$\sum_{i=o}^{n} \alpha_i = v$$

must hold for any (n,m)-code. The maximum possible number of coset leaders having a weight w_i is the number of ways in which n digits can be divided into two collections of w_i 1's and $(n-w_i)$ 0's. Thus

$$\alpha_i \leq \binom{n}{w_i} = \frac{n!}{w_i!\,(n-w_i)!} \tag{34}$$

In a previous discussion a new word N was defined as N = AT, where A represents the transmitted code word and T the received word. It was shown that N was a record of the errors on the channel during the transmission of A and that correct detection was obtained only when N was one of the coset leaders. Thus, the 1's in a coset leader indicate the position, and the weight of a coset leader indicates the number of transmission errors that can occur without causing a detection error. The α_i's defined above thus give the number of i-fold errors that can be corrected by a given (n,m)-code.

Tables II and T-4 of the cited references give values of α_i for the best (i.e. they have the minimum possible P_e) (n,m)-codes for values of k = 2, 3, - - - n - 1, and n = 4, - - - 10. (These references use Q_1 instead of P_e. Here $P_e = 1 - Q_1$.) The binomial coefficients, $\binom{n}{w_i}$, of Eq. (34) representing the maximum possible number of i-fold errors, are also listed for comparison with the α_i's.

Example 3.3.8-1

For m = 4 and n = 7 Table II, Ref. 16, shows that all 7 single and none of the 21 double or 25 triple errors will be corrected. The corresponding probability of error, as given by this table is

$$P_e = 1 - (1 - P_o)^7 - 7\,P_o(1 - P_o)^6$$

Note that this is the same expression as determined for the Hamming SEC code of Example 3.3.3-1. Since Hamming codes are a subgroup of the Slepian group codes and since the above (7,4)-code corrects all single errors this means that the two codes are equivalent.

If instead n = 10 is used, Table II, Ref. 16, shows that all 10 single, 39 of the possible 45 double, 14 of the possible 120 triple, and none of the possible 210 quadruple errors will be corrected. The resulting P_e is

$$P_e = 1 - (1 - P_o)^{10} - 10\,P_o\,(1 - P_o)^9 - 39\,P_o^{\,2}\,(1 - P_o)^8 - 14\,P_o^{\,3}\,(1 - P_o)^7$$

In addition to knowing the minimum possible P_e for a given n and m it is also necessary to know the parity check rules that will allow the corresponding best code words to be generated. These rules are given in Table III and T-5 of Ref. 16 and 8 respectively. The use of these tables is best explained by an example.

Example 3.3.8-2

For the (7,4)-code considered above Table III (Ref. 16) gives the parity check rules as

5 1 3 4

6 1 2 4

7 1 2 3

In terms of previous notations this becomes

$X_5 = X_1 \oplus X_3 \oplus X_4$

$X_6 = X_1 \oplus X_2 \oplus X_4$

$X_7 = X_1 \oplus X_2 \oplus X_3$

Thus, if a particular 4 binit message sequence is 1100 the corresponding code word is 1100 X_5 X_6 X_7 where

$X_5 = 1 \oplus 0 \oplus 0 = 1$

$X_6 = 1 \oplus 1 \oplus 0 = 0$

$X_7 = 1 \oplus 1 \oplus 0 = 0$

Slepian makes the following observation about the best codes given by Table III, Ref. 16.

1. The (n,m)-code best for a particular value of P_o is best for all values of P_o, $0 \leq P_o \leq 1/2$.

2. Not all best (n,m)-codes have the greatest possible minimum distance between nearest words.

3. If a (n,m)-code corrects all errors equal to or less than j and no errors greater than j + 1, then there exists no 2^m word, n digit code of any type that is better than the (n,m)-code listed. Such codes are defined to be optimum codes. Note that all optimum codes are best codes but that best codes are not necessarily optimum. For example, of the best codes listed in Table II, Ref. 16, the (11,3)-code is not optimum while the (8,2)-code is optimum.

3.4 Elias's Iterative Coding

At the present time, the iterative encoding and decoding techniques presented by Elias (20), (21), (22) represent the only practical method for obtaining an arbitrarly small error rate without using a feedback channel. The procedure is conceptually quite simple and may be used with either the BSC or the binary erasure channel (BEC). The following discussion will illustrate the operation for the BEC. Similar results using the Hamming SEC-DED code, or any other systematic code, are obtained for the BSC (20).

Consider a BEC as given in Fig. 3-(c). This channel model differs from the BSC in that the decision process at the receiver is modified so as to produce an erasure symbol, x, instead of an erroneous symbol. In practice this would in-

volve the use of two decision levels instead of the single level used with a BSC. This difference allows error correction to be obtained with the BEC by using a single parity check encoding procedure equivalent to the Hamming SED code.

The error correction feature is obtained by dividing the input sequence of 0's and 1's into blocks of n - 1 binits. In the n th position of each block is placed a digit resulting from an even parity check over all previous n-1 digits. This sequence of n binits is transmitted over the BEC. At the receiver there is a probability, P_o, that a given digit will be received as an erasure. If only one binit in a single block is received as an erasure the missing digits can be re-inserted by performing an even parity check over the remaining digits, i.e., if an even number of 1's remain the erased symbol was a 0 while if an odd number remains the erased symbol was a 1. For example assume the following blocks (n = 8) were received.

0 1 1 1 0 X 1 0

1 0 1 0 0 1 X X

1 1 0 0 X 1 0 0

0 0 0 1 1 1 0 X

In the first block the erased symbol must have been a 0 since an even number of 1's remain. Likewise the erased symbol must have been a 1 in the third and fourth blocks. No correction is possible in the second block since the erased symbols could have been either 0 1 or 1 0.

It is clear that this procedure reduces the average number of erasures remaining in a block. The amount of this reduction is determined as follows: Before correction the probability of exactly z erasures is

$$P\ (z \text{ erasures}) = \binom{n}{z} P_o^{\ z} (1 - P_o)^{n-z}$$

The average, or expected, value, $\overline{X}$, of a discrete random variable, X, is

given by

$$\bar{X} = \sum_i X_i \, P(X_i)$$

Thus the average number of erasures before correction is

$$\bar{z} = \sum_{z=1}^{n} z \binom{n}{z} P_o^{\,z} (1 - P_o)^{n-z} \qquad (35)$$

$$= n P_o \qquad \text{(Series No. 194. Ref. 23)}$$

The average number of erasures after correction, $\bar{z}'$, is given by

$$\bar{z}' = \sum_{z=2}^{n} z \binom{n}{z} P_o^{\,z} (1 - P_o)^{z}$$

$$= \bar{z} - n P_o (1 - P_o)^{n-1}$$

$$= n P_o \left[1 - (1 - P_o)^{n-1} \right] \qquad (36)$$

The average number of erasures is thus reduced by a factor of $\left[1 - (1 - P_o)^{n-1}\right]$ when the first order correction procedure is used.

Example 3.4-1

If $n = 7$ and $P_o = 10^{-2}$ the resulting values are

$$\bar{z} = 7 \cdot 10^{-2} = 0.07$$

$$\bar{z}' = 0.07 \left[1 - (.99)^6 \right]$$

$$= 0.00409$$

Thus, compared to the situation with no coding, the average number of erasures has been reduced by a factor of 15 while reducing the information

rate by only 12.5%. This compares quite favorably with the Hamming SEC code of Example 3.2.2-1.

Elias's iteration technique suggests that the average number of erasures can be further reduced by periodically transmitting blocks of n binits that are second order parity checks over the digits in the preceeding $n_1 - 1$ blocks. In this manner correction may be made for most of the double erasures. This procedure is best explained by means of the following example.

Example 3.4-2

Assume that blocks of 8 digits are to be transmitted and that every 8th block is to contain the second order parity check digits. Let the input to the encoder be the following 7 sequence of 7 binits each.

0111101

1111000

1100101

1101110

0011111

0110010

0001110

The first order check digits are obtained by an even parity check over the digits in each row and are placed at the end of the corresponding row. The second order check digits are obtained by an even parity check over the digits in each column and are placed at the bottom of the corresponding column. Applied to the above sequences this results in the following array.

0111101 1
1111000 0
1100101 0
1101110 1
0011111 1
0110010 1
0001110 1

} 1st order check digits

1101101 1

2nd order check digits

The code words to be transmitted corresponds to the rows in this array. At the receiver the words, containing the erasures, are placed in a similar array. All single erasures are then corrected by parity checks over the rows in this array. Additional erasures are corrected by checks over the columns in the array. Elias (22) has shown that the average number of erasures, $\bar{z}''$, remaining after this second order correction is

$$\bar{z}'' = n_1 P_1 \left[1 - (1 - P_1)^{n_1 - 1} \right]$$

where $P_1 = \frac{\bar{z}'}{n} = P_o \left[1 - (1 - P_o)^{n-1} \right]$

and

$n_1 - 1$ = the number of digits checked by the second order parity check.

For the values of Example 3.4-1 this gives

$$P_1 = 0.00409 \times 1/7 = 0.000584$$

$$\bar{z}'' = 0.00409 \left[1 - (.999426)^7 \right]$$

$$\simeq 0.00409 \times 0.00464 \cong 0.000019$$

In this case the average information rate at the channel input is 49/64 bits/binit. This represents a descrease of 12.5% from the rate for single order correction and a 23.5% reduction from the rate with no correction. Corresponding to these reductions the average number of remaining erasures has been reduced by a factor 200 times from the single correction value and by a factor of 3500 from the no correction value.

Elias has shown (20) that this iteration procedure can be continued by means of 3rd, 4th, - - - order parity checks and that in the limit the average number of erasures will approach zero while the information rate remains at a usable non-zero value.

Thus, using this method, it is possible to make the erasure probability as small as desirable if the receiver is willing to wait until a sufficiently high order parity check has been received. A unique feature of this technique is the fact that the erasure probability can be controlled at the receiver without changing the transmitted code words.

3.5 Use of Group Codes in Feedback Communication Systems

Previous discussions have indicated that when a feedback channel (i.e. a communication link from the receiver to the transmitter) is present it is possible to use error detecting codes and to request retransmission of erroneous words via this channel. When possible, this approach has the advantage of requiring less coding equipment while simultaneously giving a high information rate and a lower error rate. It should be emphasized, however, that this method does not exceed the information rate given by the second fundamental theorem but only provides a practical means of more closely approaching this rate while maintaining a low error rate.

Many investigators (3) (24) (25) (26) (27) (28) (29) (30) have analyzed the characteristics of systems using a feedback channel. However, to quote Peterson (Ref. 19)

"The efficient use of feedback in error control has not received the attention it deserves in coding theory. Certainly feedback can greatly simplify error correction. Yet there is a definite limit to the efficiency of a simple error-detection and retransmission system, for short error-detection codes cannot efficiently detect errors, while if extremely long codes are used retransmission must be performed too frequently. Little is known about the use of the feedback channel in any more sophisticated way."

Thus, the method presented in this section is not to be considered as the ultimate answer in coding for the feedback channel. Instead it represents one approach that illustrates the use of group codes for error detection.

Cowell (30) has investigated the use of group codes in a feedback system in which the group property is used to correct some errors in the conventional manner (as described in Sec. 3.3.4) and to detect additional errors. When an error is detected a request is sent, via the feedback channel, for a retrnasmission of the erroneous code word. The procedure for accomplishing this is as follows: First, a (n,m)-code is assumed and the array (not necessarily in standard form) for the group B_n is developed about this code. The resulting $2^{n-m} = v$ coset leaders are then divided into two sets one of which contains the identity element, I. Let S be the set containing I. Also, let I, A_1, A_2 - - - A_u, ($u = 2^m$) represent the code words of the (n,m)-code selected. The decoding operation is then performed by expressing the received word, T, as the product of a transmitted word, A, and a noise word, N, i.e., T = AN. (This noise word is the same as previously discussed and is a record of the errors during the transmission of A.) If the word N is contained in the set S the received word is decoded as A; otherwise the transmitter is requested, via the feedback channel, to retransmit the code word. Thus, this decoder corrects all error patterns that give noise words contained in S and requests retransmission when the noise word is not contained in S. If S contains all v coset leaders and these are of minimum weight this corresponds to the maximum likelihood detection previously discussed. Conversely, if S contains only the identity element retransmission occurs whenever the received word is not a code word.

Using this decoding scheme Cowell (30) has shown that the probability, $1 - P_e$, of a word being correctly decoded (this includes the case of correct decoding after numerous retransmissions) is given by

$$1 - P_e = \frac{D}{1 - \Theta} \tag{37}$$

where

$$D = \sum_{N} P_o^{w(N)} (1 - P_o)^{n - w(N)}$$

$$1 - \Theta = \sum_{N} \sum_{A} P_o^{w(NA)} (1 - P_o)^{n - w(NA)}$$

Here the summations are over all noise words, N, contained in the set S and over all code words, A, contained in the (n,m)-code. The following example illustrates this procedure.

Example 3.5-1

Consider the following (5,2)-code

00000, 01110, 10101, 11011.

A suitable standard array for this code is as follows

00000	01110	10101	11011
00001	01111	10100	11010
00010	01100	10111	11001
00100	01010	10001	11111
01000	00110	11101	10011
10000	11110	00101	01011
00011	01101	10110	11000
10010	11100	00111	01001

When S contains all coset leaders of weight 0 or 1 any received word

lying in the first 6 rows of the array will be decoded as one of the code words. Similarly a received word lying in either of the last two rows will cause a request for retransmission. For this situation the summation for D is over the first 6 coset leaders in the array. Thus

$$D = (1 - P_o)^5 + 5 P_o (1 - P_o)^4$$

$$= (1 + 4 P_o) (1 - P_o)^4$$

Likewise the double summation for $1 - \Theta$ is over all words in the first 6 rows of the array. This gives

$$1 - \Theta = (1 - P_o)^5 + 5 P_o(1 - P_o)^4 + 6 P_o^2 (1 - P_o)^3$$

$$+ 6 P_o^3 (1 - P_o)^2 + 5 P_o^4 (1 - P_o) + P_o^5$$

$$= \sum_{i=0}^{5} \binom{5}{i} P_o^i (1 - P_o)^{5-i} - 4\left[P_o^2(1 - P_o)^3 + P_o^3 (1 - P_o)^2\right]$$

$$= 1 - 4 P_o^2(1 - P_o)^2$$

where the last step follows from the binomial expansion. The final expression is thus

$$1 - P_e = \frac{(1 + 4P_o) (1 - P_o)^4}{1 - 4 P_o^2 (1 - P_o)^2} \tag{38}$$

If, instead, S contains only I the expressionsbecome

$$D = (1 - P_o)^5$$

$$1 - \Theta = (1 - P_o)^5 + 2 P_o^3 (1 - P_o)^2 + P_o^4 (1 - P_o)$$

$$1 - P_e = \frac{1}{1 + 2P_o^3 (1 - P_o)^{-3} + P_o^4(1 - P_o)^{-4}} \tag{39}$$

Likewise, if S contains all coset leaders the expressions are

$$D = (1 - P_o)^5 + 5 P_o (1 - P_o)^4 + 2 P_o^2 (1 - P_o)^3$$

$$1 - \Theta = 1$$

$$1 - P_e = D \qquad (40)$$

which agrees with the value given by Slepian (Table II, ref. 16) for maximum likelihood detection.

It is instructive to compare, numerically, the cases for S containing only I and for S containing all coset leaders. Assume $P_o = 10^{-2}$. Then from Eq. (39) the probability of error using retransmission only (i.e., S = I) is

$$P_e = 1 - \frac{1}{1 + 2 \cdot 10^{-6} (.99)^{-3} + 10^{-8} (.99)^{-4}}$$

$$= 2.165 \times 10^{-7}$$

Conversely, when no retransmission is used (i.e., S contains all coset leaders) Eq. (40) gives

$$P_e = 1 - (.99)^5 + 5 \cdot 10^{-2} (.99)^4 + 2 \cdot 10^{-4} (.99)^3$$

$$= 7.36 \times 10^{-4}$$

Thus when the code is used only for error detection (with error correction being obtained by retransmitting the erroneous word) the probability of error is reduced by a factor of approximately 3400 times.

Cowell (p 169, Ref. 30) has show that this result is true in general. Thus when a group code may be used for either error correction, error detection (error correction via a feedback channel is assumed), or for both, the minimum P_e will be obtained when the code is used only for error detection. This is intuitively

satisfying since for this case the probability of retransmitting code words is maximum thus introducing a maximum amount of redundancy into the transmitted sequences.

Due to the pronounced improvement in P_e obtained with error-detection-only operation a question arises as to the amount by which this type of operation reduces the information rate. A convenient means of specifying this reduction is to define the coding efficiency, N_c, as

$$N_c = \frac{\text{m = number of message digits per code word}}{\text{average number of digits transmitted until a word is decoded}}$$

This ratio, when expressed as a decimal gives the information rate at the channel input. Thus, when no retransmission is used (i.e., the code is used only for error correction) the input information rate is

$$N_c = \frac{m}{n} \tag{41}$$

When the code is used for both correction and detection (or detection only) Cowell has shown that the coding efficiency is given by

$$N_c = \frac{m\,(1 - \Theta)}{n + L\,\Theta}$$

where n, m, and Θ are as previously defined and L represents the number of digits that are lost whenever a retransmission occurs (i.e., digits required to re-establish synchronization, digits lost because of an in interleaved transmission pattern, etc.).

In determining L the digits of a retransmitted code word are <u>not</u> included. Thus the value of L depends directly upon the communication system and only indirectly, if at all, upon the (n,m)-code.

Example 3.5-2

Assume that the (5,2) code of Example 3.5-1 is used for error detection only, that L = n, and that $P_o = 10^{-2}$. For this situation the coding efficiency is

$$N_c = \frac{m}{n} \cdot \frac{1 - \theta}{1 + \theta}$$

$$= \frac{2}{5} \cdot \frac{(.99)^5 + 2 \cdot 10^{-6}(.99)^2 + 10^{-8}(.99)}{2 - (.99)^5 - 2 \cdot 10^{-6}(.99)^2 - 10^{-8}(.99)}$$

$$= 0.4 \times \frac{0.951}{1.048}$$

$$= 0.363$$

Thus the use of the (5,2)-code for error detection only (as compared to its use for error correction only) causes a reduction in the information rate of approximately 10% while giving a reduction in error rate of 3400 times. This result illustrates the fact that in general the use of a feedback system will allow a greatly reduced error rate for a given information rate and code word length. However, very little work has been done in determining optimum codes for this operation and little is known about the maximum possible improvement that can be obtained. At present this area appears to offer the greatest potential for determining practical techniques that will allow the rates of the second fundamental theorem to be approached and as such is an area worthy of much additional research.

3.6 Additional Techniques for Noisy Channel

The coding techniques presented in the preceeding four sections were chosen primarily for two reasons: (1) they represent some of the most basic and better known of present techniques; and (2) they are relatively easy to explain. This section will present some of the more advanced techniques. These will not be

discussed in detail, however, since they require a knowledge of modern algebra with a strong emphasis on matrix theory.

3.6.1 Bose-Chaudhuri Codes

Bose-Chaudhuri codes (31) (32) (Chapt. 9, Ref. 19) represent a generalization of Hamming codes in that a specific procedure is given for constructing a set of code words when the amount of error correctability is specified. Petersen (p 165, Ref. 19) gives the following theorem regarding these codes:

"For any m and t ($mt < n$) there is a Bose-Chaudhuri code of length 2^m-1 which corrects all combinations of t or fewer errors and has no more than mt parity check digits."

Thus, in contrast to the Hamming (which could be constructed only for a capability up to SEC-DED) and the Slepian (which must be constructed by some type of a search through N(n,m)possibilities) codes, the Bose-Chaudhuri codes can be constructed for any n, m, and t provided the relations of the above theorem are satisfied. However, there is at present no general information concerning P_e for these codes.

The Bose-Chaudhuri codes are related to the Slepian codes in that they are a subgroup of cyclic* codes which are in turn a subgroup of the general class of group codes. The decoding procedure, however, differs considerably from that for the Slepian codes (33).

3.6.2 Reed-Muller Codes

As indicated previously, the Reed-Muller codes (17) are a subclass of the group codes considered by Slepian. They differ from the Slepian codes in that a

---

* A cyclic code is a special group code in which a cyclic shift of any code word is another code word. For example if 10110000 is a word of a cyclic code then 01011000, 00101l0, etc. must also be code words.

specific procedure is available for determining a set of code words when the following relations are satisfied:

$$n = 2^t$$

$$m = 1 + \binom{t}{1} + \cdots + \binom{t}{r}$$

$$n - m = 1 + \binom{t}{1} + 1 - \binom{t}{t-r-1}$$

where t is the maximum distance between code words and r is the "order of the code. For example, if t = 4 and r = 2 the Reed-Muller code would have n = 16, m = 11 and, from section 3.2.4, would be a SEC-DED code. The generation of the code words for a Reed-Muller code involves the use of vector algebra and therefore will not be discussed. The primary advantage of Reed-Muller codes lies in the realitive ease with which decoding equipment can be constructed. Some work has been done at the M.I.T. Lincoln Laboratory (34) in the construction of an encoder and decoder for a Reed-Muller code with n = 128, m = 64.

3.6.3 Fire Codes

The Fire Codes (Sections 10.1 and 10.2, Ref. 19) are designed to detect and/or correct errors that occur in a single burst within a code word (i.e., the errors do not occur independently but instead occur in several consecutive digits). Other codes such as the Reed-Solomon codes (Sections 9.3 and 10.7, Ref. 19) can correct more than one burst of errors.

The conditions under which a Fire Code can be constructed are as follows:

n = least common multiple (LCM) of $(2^t - 1)$ and $(b + d - 1)$

where b = length (in binits) of burst to be corrected

d = length (in binits) of burst to be detected

t = an integer $\geq b$

$n - m = k = t + b + d - 1$

When used for detection alone such a code can detect a single burst of

length no greater than k binits. When used for both detection and correction it will correct any single burst of length b or less and detect any single burst of length d or less.

Example 3.6.3-1

For m = 5, b = 5, d = 7 the length of the Fire Code is given by

n = LCM of 31 and 11 = 341

Thus

k = 5 + 5 + 7 - 1 = 16

and

m = 325

This code will correct a burst of 5 errors and detect a burst of 7 errors. Observe the high ratio of m/n for this code. This is a characteristic of codes for burst error detection and correction and is not possible with codes for independent errors.

Details concerning the construction of Fire Codes should be obtained from Ref. 19, Section 10.1.

3.6.4 Wozencraft's Sequential Coding

All of the codes previously discussed have been block codes. All block codes have the fault that as n is increased (in an attempt to obtain a greater m/n ratio and a lower P_e) the delay between the time a symbol is produced at the source and the time it is decoded at the receiver also increases. Thus in many situations a maximum allowable delay places an upper bound upon the length of any block code that might be used. This in turn limits the information rate and P_e that may be obtained. A practical method for circumventing this problem could offer a considerable potential for more closely approaching the rates of the second fundamental theorem.

The sequential encoding and decoding technique discovered by Wozencraft (4)

(35) represents such a method.

Since the crux of this method lies in the decoding operation only this will be considered.

In essence, sequential decoding is accomplished by decoding one received information digit at a time. The procedure is as follows: The actual received sequence is compared with all possible transmitted sequences starting with a 0 and also with all possible transmitted sequences starting with a 1. Unless a large number of errors have occurred the actual received sequence will differ from all but one sequence in one of these sets by such a large amount that it can be concluded that the sequence for which the difference is a minimum represents the transmitted sequence. In this manner the first information digit is determined. It is then recorded and deleted from the sequence. The comparison procedure is then repeated to determine the next information digit, etc.

With this procedure the delay between a generated symbol and a decoded symbol is greatly reduced for a given P_e. A second advantage of this method lies in the fact that decoding equipment requirements grow approximately as the square of the effective code length while many block decoding schemes involve equipment requirements that grow exponentially with increasing code length.

At the present time, sequential coding represents what is probably the most sophisticated of all techniques and as such is one of the most difficult to understand. For the serious worker in this area Ref. 4 gies a thorough discussion of the details involved.

3.7 Relationship Between the Coding Techniques Discussed in this Report.

It is often difficult for a newcomersto the field of coding theory to establish just exactly where the numerous coding techniques fit into the overall picture. The block diagram of Fig. 7 has been prepared to provide such a picture. Starting at the top, the general area of the study of coding techniques is indicated. This area can be divided into essentially two groups: (1) those systems

that use binary symbols, and (2) all non-binary (or n-ary) systems. Because of their widespread use, this report has considered only binary systems. Proceeding with binary systems, there are within this group two further divisions, namely, coding for the noiseless channel and coding for the noisy channel. From this further breakdown are indicated between equal and non-equal cost symbols, etc. Finally, the various coding techniques are indicated under the appropriate blocks.

For purposes of comparison, Fig. 8 lists some of the advantage and disadvantages of the various codes.

3.8 Conclusion

The coding techniques presented in this report represent some useful and practical methods of coding for both the noisy and noiseless channel. The noiseless procedures of Huffman, Gilbert-Moore and Karp represent optimum (i.e. they give maximum efficiency) procedures for the noiseless channel and as such may be used essentially without qualification. However, the noisy procedures that have been presented do not have this desirable characteristic. Instead, these procedures represent some of the less mathematical, and thus more readily explained, better known procedures. In many cases these procedures are well known simply because they represent the first work in a particular area and not because they are the best possible techniques. Thus any practical application of noisy coding should be preceeded by further investigation into some of the later and more advanced techniques. Elias (pp 342-343, Ref. 36) gives an excellent discussion of some additional factors and methods that should be considered.

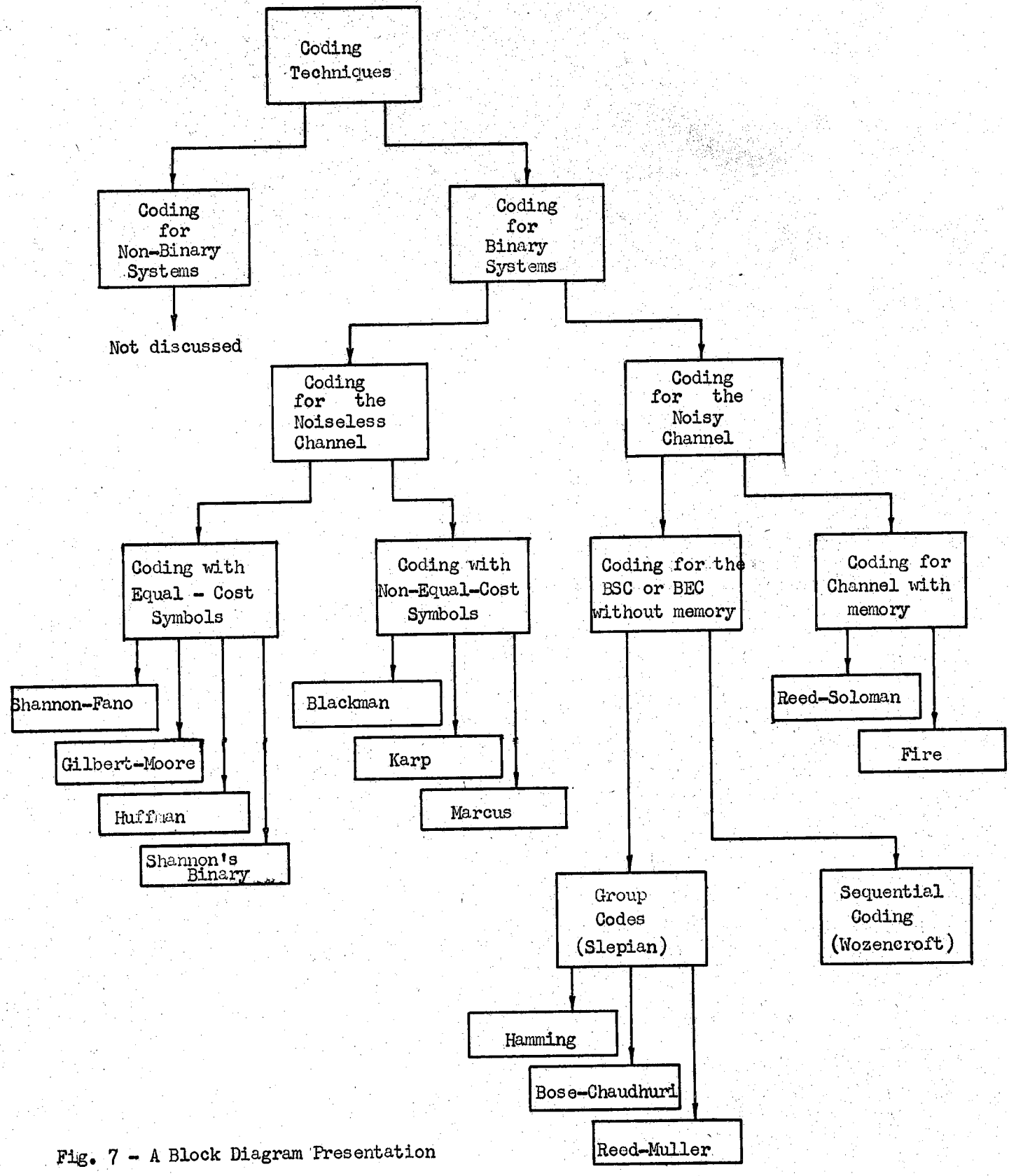


Fig. 7 - A Block Diagram Presentation of the Relationship Between Various Coding Techniques

Coding Techniques	Advantages	Disadvantages
Hamming Codes	1. Are conceptually the most simple codes. 2. Are reasonably easy to instrument.	1. No procedure for constructing codes having a minimum distance greater than 4. 2. Information rate is small due to the restriction on word length imposed by the requirements that the probability of higher order errors be negligibly small.
Slepian Group Codes	1. In some cases are best possible codes. 2. The encoding scheme is relatively easy to instrument. 3. The decoding scheme is the best possible theoretically and is relatively easy to instrument.	1. The procedure for determining parity check rules involves a search through a large number of possible codes. Because of this codes for n greater than 12 have not been determined. 2. The procedure for determining coset leaders used in decoding is involved for large n.
Elias's Iterative Coding	1. Allows error rate to be made arbitrarily small while giving a useful information rate. 2. For moderate P_e requirements the decoding is relatively simple. 3. When used with a BEC the encoding is extremely simple.	1. Both encoder and decoder have large storage requirements when P_e must be small. 2. Transmission at channel capacity is not possible while simultaneously obtaining arbitrarily small P_e.
Bose-Chaudhuri Codes	1. Provide an explicit procedure for constructing codes having a specified minimum distance between code words	1. Procedure is applicable only for code word lengths of $2P - 1$, $p = 1, 2, 3, - \ - .$

Fig. 8 - Some Advantages and Disadvantages of Various Noisy Coding Techniques

Coding Techniques	Advantages	Disadvantages
Reed-Muller Codes	1. Decoding procedure is relatively simple to instrument. 2. Provides an explicit procedure for constructing codes having a specified minimum disatance between code words.	1. Conceptually quite complex. 2. Procedure applies only for code lengths of 2^p, p = 2, 3, 4, - -.
Fire Codes	1. Can correct errors occuring in bursts with fewer check digits than with codes designed for independent errors. 2. Are relatively easy to instrument. 3. Can be used for simultaneous detection and correction.	1. Can correct only a single burst of errors within a given code word. 2. Require a knowledge of modern algebra to understand.
Sequential Coding	1. Offers possibility of obtaining small P_e without the excessive delay time of block codes. 2. Decoding equipment grows slowly with effective block length as compared to block decoding.	1. Operation extremely difficult to analyze. 2. No information available on P_e.
Feedback Systems	1. Reduces equipment complexity for a given information and error rate. 2. Allows the use of error-detection-only codes with are easier to instrument.	1. Requires a feedback channel.

Fig. 8 - - (cont.)

BIBLIOGRAPHY

1. C. E. Shannon, "A Mathematical Theory of Communication," Bell System Technical Journel, Vol. 27, 1948, pp. 379-423 and pp. 623-656.

2. C. E. Shannon, "Communication in The Presence of Noise," Proc. I.R.E. Vol. 37, 1949, pp. 10-21.

3. B. Reiffen, W. G. Schmidt and H. L. Yudkin, "The Design of an Error-Free Transmission System for Telephone Circuits," Communication and Electronics, July, 1961, pp. 224-230.

4. J. M. Wozencraft and B. Reiffen, "Sequential Decoding," Technology Press and Wiley, 1961.

5. J. C. Hancock, "An Introduction to the Principles of Communication Theory," McGraw-Hill, 1961, pp. 78-89.

6. Laning and Battin, "Random Processes in Automatic Control," McGraw-Hill, 1956, Chapter 2.

7. Y. W. Lee, "Statistical Theory of Communication," Wiley, 1960, Chapter 3.

8. F. M. Reza, "An Introduction to Information Theory," McGraw-Hill, 1961.

9. R. M. Fano, "The Transmission of Information," Technical Report 65, M.I.T. Cambridge, Mass., March 17, 1949.

10. D. A. Huffman, "A Method for the Construction of Minimum Redundance Codes," Proc. IRE, Sept. 1952, pp. 1098-1101.

11. R. M. Fano, "Transmission of Information," M.I.T. Press and Wiley, 1961.

12. N. M. Blackman, "Minimum-Cost Encoding of Information," IRE Trans. on Information Theory, Vol. IT-3, pp. 139-149.

13. R. S. Marcus, "Discrete Noiseless Coding," M.S. thesis in electrical engineering, M.I.T., Cambridge, 1957.

14. R. M. Karp, "Minimum-Redundance Coding for the Discrete Noiseless Channel," IRE Trans. on Information Theory, Vol. IT-7, pp. 27-38.

15. R. W. Hamming, Error Detecting and Error Correcting Codes," Bell System Technical Journal, Vol. 29, 1950, pp. 147-160.

16. D. Slepian, "A Class of Binary Signaling Alphabets," Bell System Technical Journal, Vol. 35, 1956, pp. 203-234.

17. I. S. Reed, "A Class of Multiple-Error-Correcting Codes and the Decoding Scheme," IRE Transactions on Information Theory, PGIT - 4, 1954, pp. 38-49.

18. P. Elias, "Coding for Noisy Channels," IRE Convention Record, Part 4, 1955, pp. 37-46.

19. W. W. Peterson, "Error-Correcting Codes," M.I.T. Press and Wiley, 1961.

20. P.Elias, "Error-Free Coding," IRE Transaction on Information Theory, PGIT-4 September, 1954, pp. 29-37.

21. E. J. Baghdady, "Lectures on Communication System Theory," McGraw-Hill, 1961, Chapter 13.

22. Grabbe, Ramo, and Wooldridge, "Handbook of Automation, Computation, and Control," John Wiley & Sons, 1958, Vol. 1, pp. 16-35 to 16-38.

23. L. B. W. Jolley, "Summation of Series," Dover Publications, Inc., 1961.

24. S. S. L.Chang, "The Theory of Information Feedback Systems," IRE Transactions on Information Theory, Vol. IT - 2, Sept. 1956, pp. 29-40.

25. B. Harris, A. Hauptschein, and L. S. Schwartz, "Optimum Decision Feedback Circuits," IRE Convention Record, Part 2, 1957, pp. 3-10.

26. B. Harris, A. Hauptschein, K. C. Morgan, and L. S. Schwartz, "Binary Decision Feedback Systems for Maintaining Reliability Under Conditions of Varying Field Strength," Proceeding of the National Electronics Conference, Vol. 13, 1957, pp. 126-140.

27. W. B. Bishop and B. L. Buchanan, "Message Redundance versus Feedback for Reducing Message Uncertainty," IRE Convention Record, Part 2, 1957, pp. 33-39.

28. B. Harris and K. C. Morgan, "Binary Symmetric Decision Feedback Systems," Communication and Electronics, No. 38, 1958, pp. 436-443.

29. J. J. Metzner, "Binary Relay Communication with Decision Feedback," IRE Convention Record, Part 4, 1959, pp. 112-119.

30. W. R. Cowell, "Thus Use of Group Codes in Error Detection and Message Retransmission," IRE Transactions on Information Theory, Vol. IT - 7, July,

31. R. C.Bose and D. K. Ray-Chaudhuri, "On a Class of Error Correcting Binary Group Codes," Information and Control, Vol. 3, No. 1, March 1960, pp. 68-79.

32. R. C. Bose and D. K. Ray-Chaudhuri, "Further Results on Error Correcting Binary Group Codes," Ibid., Vol. 3, No. 3, Sept. 1960, pp. 279-290.

33. W. W. Peterson, "Encoding and Error-Correction Procedures for the Bose-Chaudhuri Codes," IRE Transactions on Information Theory, Vol. IT-6, Sept., 1960, pp. 459-470.

34. K. E. Perry, "An Error-Correcting Encoder and Decoder for Phone Line Data," IRE Wescon Convention Record, Part 4, Aug., 1959, pp. 21-26.

35. J. M. Wozencraft, "Sequential Decoding for Reliable Communication," IRE National Convention Record," Part 2, Mar., 1957, pp. 11-23.

36. E. J. Baghdady, "Lectures on Communication System Theory," McGraw-Hill, 1961.