INFORMATION TO USERS

This manuscript has been reproduced from the microfilm master. UMI films the text directly from the original or copy submitted. Thus, some thesis and dissertation copies are in typewriter face, while others may be from any type of computer printer.

The quality of this reproduction is dependent upon the quality of the copy submitted. Broken or indistinct print, colored or poor quality illustrations and photographs, print bleedthrough, substandard margins, and improper alignment can adversely affect reproduction.

In the unlikely event that the author did not send UMI a complete manuscript and there are missing pages, these will be noted. Also, if unauthorized copyright material had to be removed, a note will indicate the deletion.

Oversize materials (e.g., maps, drawings, charts) are reproduced by sectioning the original, beginning at the upper left-hand corner and continuing from left to right in equal sections with small overlaps.

ProQuest Information and Learning
300 North Zeeb Road, Ann Arbor, MI 48106-1346 USA
800-521-0600

SOURCE ENCODING WITH NEGACYCLIC CODES

by Jean-Pierre Dion

Thesis presented to the School of Graduate Studies
in partial fulfilment of the requirements for the
degree of Ph.D. in Electrical Engineering

UNIVERSITY OF OTTAWA
OTTAWA, CANADA, 1977

ABSTRACT

Following a brief review of existing source coding techniques, negacyclic codes are defined and a joint source and channel coding scheme making use of these codes is introduced. The distortionless case and cases where distortion is allowed, are examined from the point of view of data compression. An orthogonal variation on the original scheme is also introduced. Discrete memoryless sources are assumed throughout, and the metric used in all cases is the Lee distance.

TABLE OF CONTENTS

LIST OF ILLUSTRATIONS iv

LIST OF TABLES vi

ACKNOWLEDGEMENTS vii

Chapter

I. INTRODUCTION 1

II. A REVIEW OF SOURCE CODING TECHNIQUES . . . 18

1. Introduction 18
2. The Standard Array 20
3. The Noiseless Coding Theorem 23
4. Entropy Coding 27
   a. Shannon-Fano Source Coding . . 27
   b. Huffman Source Coding 32
5. Syndrome Source Coding 38
6. Source Coding with Distortion 41

III. NEGACYCLIC CODES 44

1. Introduction 44
2. Definition of Negacyclic Codes . . . 44
3. Parallels between Negacyclic and Cyclic Codes 55

IV. BLOCK SOURCE CODING WITH NEGACYCLIC CODES . 60

1. Introduction 60
2. Extension of Binary Cyclic Code Theorems to Negacyclic Codes . . . 63
3. A Joint Source and Channel Encoding Scheme 81
   a. Introduction 81
   b. Distortionless Encoding Scheme 82
      (i) Average Length of the Encoded Sequence 88
   c. Encoding with Distortion . . . 96
      (i) Schemes for the Introduction of Distortion 96
         Case (a) 98
         Case (b) 99
         Case (c) 101

Chapter
IV. Continued

(ii) Average Length of the Encoded Sequence, Cases with Distortion 103
(iii) Decrease in Average Lengths of the Encoded Sequence . 108
(iv) Average Distortion per Digit 112
Case (a) 112
Cases (b) and (c) . . . 114
4. Orthogonal Distortionless Encoding Scheme 115
5. The Rate-Distortion Function for Non-binary Sources 119

V. CONCLUDING REMARKS 139

. . . . . .

REFERENCES 142

BIBLIOGRAPHY 149

VITA 152

LIST OF ILLUSTRATIONS

Figure		
1.	Block diagram of a communications or memory storage system model	3
2.	A typical rate distortion function R(D) for a binary source with equiprobable symbols	9
3.	Tree obtained by Huffman source coding of the 27 messages of Example 2. (The message probabilities are given in Table 3.)	35
4.	Block diagram of joint source and channel encoding system	86
5.	Flowchart of second error-detection step for decoder of Figure 4	87
6.	Normalized average length of the encoded sequence σ for the distortionless case, with the distribution of Equation (3). The upper curve corresponds to the approximate expression	93
7.	Normalized average length of the encoded sequence σ for the distortionless case, with the distribution $\Pi_i = \Pi_0 \lambda^i$	97
8.	Normalized average length of the encoded sequence σ for case (a), with $t_0 = t - 1$	105
9.	Normalized average length of the encoded sequence σ for cases (b) and (c), with Ω as parameter	107
10.	Normalized change in average length from the distortionless case for case (a), with $t_0 = t - 1$	111

Figure

11. Normalized change in average length from the distortionless case for cases (b) and (c), with Ω as parameter 113

12. Rate distortion function of a 3-letter source, with $p_1 = p_3 = \frac{1-p_2}{2}$, for $p_2 > 0.45$ 133

13. Source entropy as a function of p_2 for a 3-letter source with $p_1 = p_3 = \frac{1-p_2}{2}$ 134

LIST OF TABLES

Table

1. Examples of Different Types of Source Codes 26

2. Shannon-Fano Source Code for a 3-Symbol Coding Alphabet 30

3. Huffman Source Code for a 3-Symbol Coding Alphabet 36

4. Example Results for the Three Weight-Limiting Schemes 102

ACKNOWLEDGEMENTS

The author wishes to express his gratitude to his supervisor, Professor S. G. S. Shiva, for his encouragement and guidance throughout the course of this research.

Thanks are also due to various members of the staff of the Electrical Engineering Department for their unfailing encouragement.

Special thanks are due Mrs. T. Goldberger for her meticulous typing of the thesis.

The author is also grateful to the Communications Research Centre, the National Research Council and the Province of Ontario for their financial support.

CHAPTER I

INTRODUCTION

In recent years the demand for efficient and reliable transmission of digital information has been accelerated by the increasing use of digital computing equipment and the rising need for long range communications [1]. Also, more and more analog signals are being digitally coded, since digital signal representation offers many advantages. These include efficient signal regeneration so that in long range transmission signal quality is nearly independent of distance. A further advantage is offered through the possibility of combining transmission and switching functions. Digital signals also allow a uniform format for representing different types of signals and for use on different transmission media (present and future). The possibility of using digital techniques for further storage and processing of the signals is one more factor in their favour. Digital representation of signals is also used for solving difficult transmission problems, such as the transmission of video data from space probes. A final advantage is the ease with which messages can be encrypted in secrecy systems [2],[3].

Communications systems [4] can generally be grouped into one of the following two classes: those requiring a

very high fidelity of reproduction of the transmitted information, and those which can tolerate a certain amount of distortion in the received information. An example of the first class would be a computer communication link, while a television link would belong to the second class.

In this thesis a general model of a communications or memory storage system is first presented. This model is information-theoretic in that the modem is considered to be part of the channel [5]. Figure 1 is a block diagram of such a system. The model is quite general in the sense that not all coder/decoder pairs need necessarily appear. The source generates the information which is ultimately transmitted to the sink or user. The different components of the system model are described next.

Referring to Figure 1, the source encoder is the component which categorizes the different possible output states of the source into a lesser number of states or categories. These constitute the source encoder's output. The source encoder is used to remove redundancy from the source output. This reduces the amount of data that must be transmitted to the user for transmission of a given message. Redundancy reduction results in data compression, and, all other things being equal, a reduction in transmission rate. Thus a lower capacity channel can be used, so that data compression is equivalent to bandwidth compression. It may be noted at this point that many expressions have been used to describe the function of the

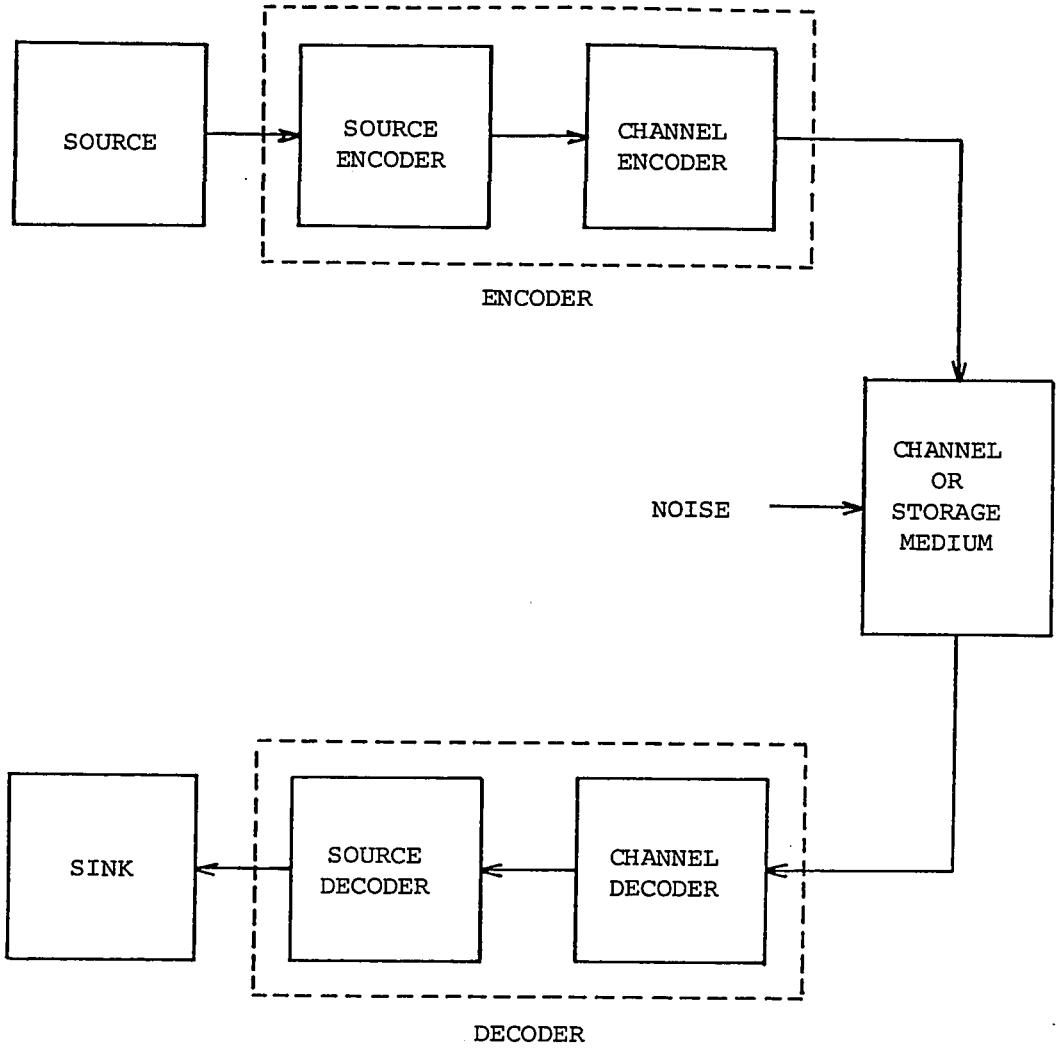

Figure 1:- Block diagram of a communications or memory storage system model.

source coder. Those encountered most often are: bandwidth reduction, bandwidth compression, data compression, data reduction, redundancy reduction and redundancy removal [6].

The source encoding step may introduce distortion, but need not. When distortion is allowed, there generally exists a trade-off between the allowed distortion and the amount of data compression. The main purpose of this thesis is to study certain source coding techniques and the corresponding compression-distortion trade-offs.

Referring again to Figure 1, the channel encoder serves to introduce some structured redundancy to the output of the source encoder. This is done on the basis of a one-to-one assignment rule, to produce a channel code word. The purpose of the added redundancy is to enable the channel decoder to correct all of the channel errors the code is designed to correct.

The channel is a spatial link between the source and the sink. For example, it can be a cable made up of many parallel twisted pairs of wire connecting a computer to one of its peripherals. Another example would be that of a microwave link between an earth station and a communications satellite. The channel can also be a storage medium such as magnetic tape, drum or disc, core or semiconductor memory, etc. In the case of a storage medium, storing data in memory is equivalent to transmission, and subsequent retrieval of that data, to reception.

Whether a spatial link or a storage medium, however, the channel has the capability of corrupting the information being transmitted or stored.

The errors caused by the channel can be detected or corrected by the channel decoder, as long as the number of errors per code word does not exceed the error-detecting or error-correcting capability of the code. The channel decoder's function is to categorize the many possible channel outputs into a generally much smaller number of channel decoder outputs. From this point of view, the channel decoder's function is analogous to that of the source encoder.

Finally, the channel decoder output is the input to the source decoder. It reverses the assignment made in the source encoder, and so its output is an approximation of the original source output. This serves as the input to the sink. The source decoder output is identical to the source encoder input if the source encoding scheme is distortionless and all channel errors are within the channel codec's error-correcting capability. If the scheme is not distortionless, or if the number of channel errors exceeds the channel code's capability, then the source decoder's output will not be identical to the source encoder's input. It is not generally possible to determine at the receiver which portion of the difference is due to the channel errors, and which is due to the source encoding scheme. For this reason, it is usual to assume that all channel errors are corrected by the channel decoder. Thus the

portion of Figure 1 made up of the channel encoder, channel and channel decoder can be replaced by a line joining the source encoder to the source decoder, so that any difference between the source encoder input and the source decoder output is due solely to the distortion introduced by the source encoding scheme.

It should be noted that the channel encoder and the source decoder, which must perform one-to-one assignments, are relatively simple compared to the source encoder and channel decoder, which must categorize their respective inputs according to some categorization rule.

Referring to the second paragraph in this chapter, for communications systems requiring a high degree of fidelity, efforts have mainly been directed toward achieving the result promised by Shannon's coding theorem [7]. The theorem states that the probability of transmission error can be made arbitrarily small as long as the transmission rate is less than the channel capacity. The rate can be reduced to accomodate the channel capacity by adding redundancy to the information prior to its transmission, and removing it at the receiver. This corresponds to channel encoding and decoding respectively. To facilitate the introduction and removal of the redundancy, it is incorporated in a structured fashion such that a given information sequence always results in the identical redundancy being added to it by the channel encoder.

There have been two fundamentally different approaches to this problem [8]. The first approach is the use of block coding, in which the channel encoder partitions the continuous sequence of information digits into k-symbol sections or blocks. It then operates on these blocks independently according to the particular coding rule being implemented, and transmits n symbols for each k-symbol block, with $k < n$. These n symbols are transmitted over the channel, and the channel decoder at the receiver decodes each n-symbol block independently into a block of k symbols. Block codes will be defined at the end of this chapter.

In this thesis, block codes are the main concern. However, for the sake of completeness, the second type of codes is mentioned. Tree codes, in contrast to block codes, do not break up the information sequence into blocks which are then coded independently. Rather, the tree channel coder produces an n_0-tuple which is dependent on the latest k_0 ($k_0 < n_0$) information symbols and the previous information symbols. Thus, the k_0-tuples are not encoded independently. Convolutional codes are an important class of tree codes, since their implementation is less complex than that of other tree codes.

In regard to block codes, those which have been most studied are the cyclic codes, possessing a rich algebraic structure. Originally, this was also due to their ease of implementation, since they could be generated

using shift registers instead of a look-up table. (This is also true of some non-cyclic codes.) With the development of faster, more inexpensive, higher capacity and higher density memories, a table look-up can be a viable alternative to generating the code words as needed using shift registers. However, the length of a code implemented using a table look-up is severely limited.

The low-fidelity communications system mentioned in the second paragraph of this chapter will now be considered. In this type of system, an approximation of the source output is to be transmitted. Generally, the better the approximation, the greater the amount of symbols that must be transmitted. Hence, the portion of the source output to be transmitted and the amount of tolerable distortion are of primary interest.

Any communications or memory storage system of this type [5] has associated with it a quantity C and a function R(D), where C is the capacity of the channel and R(D) is the rate distortion function of the source-user pair. Figure 2 illustrates a typical rate-distortion function, that of the discrete memoryless source (DMS) for p(0) = p(1). From Shannon [9], it is known that transmission of information with, at most, distortion D, is possible if and only if C > R(D).

A fundamental quantity associated with a source, as will be seen later, is its entropy H [5],[10]. Entropy can be interpreted as its effective rate of information

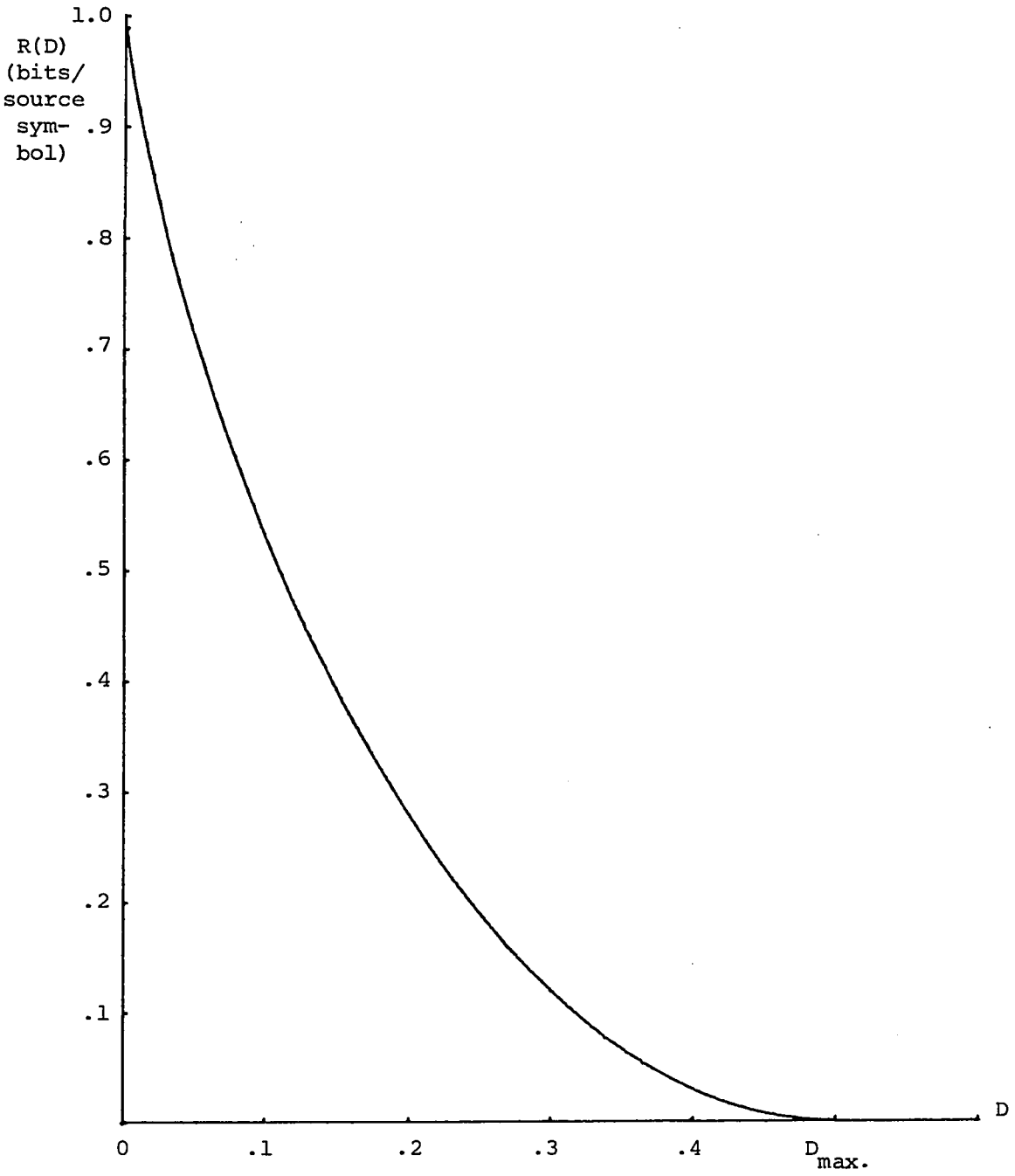

Figure 2:- A typical rate distortion function R(D) for a binary source with equiprobable symbols.

production without distortion. When $D > 0$, the effective rate of information production is reduced, since the source encoder will be categorizing more distinct outputs of the source into the same source encoder category (or output). Hence as D increases, the rate $R(D)$ decreases, and one may transmit over a channel with a reduced capacity C. Alternatively, if D is allowed to increase, then for a given channel capacity C, the output from a higher-entropy source can be accomodated on the channel.

It should be noted at this point that while the source and channel encoders on the one hand, and the channel and source decoders on the other, are usually represented as being separate, they need not necessarily be distinct, and can actually be implemented as a generalized encoder and decoder, respectively, as will be seen in Chapter IV.

Since the characteristics of the source determine to a great extent the source-encoding scheme to be used, it is appropriate at this point to make a closer examination of information sources themselves.

There are two basic types of sources: discrete sources, which produce inherently digital outputs, and analog sources, whose outputs are continuous functions. Examples of the former would be computer output ports, and teletype and radar transmitters. Examples of the latter would include telephone, broadcast AM and FM transmitters, as well as image orthicon (video) tubes. While this thesis is exclusively concerned with discrete sources, some

remarks concerning the conversion of analog source outputs into a discrete format will be made for the sake of completeness.

One of the most-used methods for converting analog source outputs to a digital format is first examined: pulse code modulation (PCM).

As is well known [2], discretizing analog sources involves sampling and quantizing of a band-limited analog signal. The number of samples per second (the Nyquist rate [11],[12]), must be greater than, or equal to, twice the bandwidth of the analog signal, if it extends to the zero frequency. On the other hand, if the lowest frequency in the signal is f_1, then the minimum sampling rate is given by $2(f_1 + \text{bandwidth})/M$, where M is the largest integer not exceeding $(f_1 + \text{bandwidth})/(\text{bandwidth})$. In regard to sampling approximations, it is useful to note that no distortion is introduced by finite sampling as against ideal instantaneous sampling. The sampling time Δt appears only as a multiplicative constant in the reconstructed signal.

Quantization can be either uniform or nonuniform. While uniform quantization can be implemented simply, it has some serious drawbacks. Since the quantization error power associated with the reproduced signal is independent of the magnitude of the signal itself, it affects the lower amplitudes more than it does the higher amplitudes. Thus, the distortion due to quantization will tend to be

relatively greater when the signal level is low. Also, from an information-theoretic point of view, uniform quantization does not maximize the entropy when the levels are not uniformly distributed.

In nonuniform quantization [13], a set of uniformly-spaced levels is converted into a set of nonuniformly-spaced levels according to what is usually referred to as a companding (compression-expansion) law. Among the available companding laws, the ideal one for minimizing noise is the logarithmic law since it yields a constant signal-to-noise ratio. Other companding laws exist which do not perform as well as the logarithmic law, but are simpler to implement.

Three basic considerations govern the choice of a companding law. The first lies in choosing between minimizing noise or maximizing information, since the companding laws differ in both cases. The disagreement between the two laws increases with the departure of the source output distribution from the uniform distribution. (Companding laws maximize the information content by making the quantizer outputs equiprobable.) The second involves knowledge of the types of analog signals involved. The third consideration is that of the number of quantization levels. For example, if the number of quantization levels is small, the use of nonuniform quantizers will not significantly alter the performance of a quantizer.

Max [14] has presented an algorithm for the design of quantizers which minimize the quantization noise for any

given number of quantization levels. However, O'Neal [15] has pointed out that Wood, as well as Gish and Pierce, have shown that quantizers designed using the Max algorithm are not optimum when entropy coding is used to further encode the quantizer output. On the other hand, a quantizer with equally-spaced levels has been found to be very nearly optimum when used with entropy coding. (Entropy coding is described in Chapter II.)

Besides pulse code modulation, there exist many techniques for discretizing analog sources. Adaptive PCM (APCM), differential pulse code modulation (DPCM), and adaptive DPCM (ADPCM), as well as delta modulation (DM), adaptive DM (ADM), continuous DM (CDM) and digitally-controlled DM (DCDM) are examples of the different types. Jayant [3] gives a general treatment of these different techniques, as well as an extensive bibliography. A discussion of all these techniques is beyond the scope of this thesis.

At this point it is appropriate to mention that the concept of a rate distortion function does not in general apply to continuous sources, since the entropy of unbounded continuous sources can be considered to be infinite. Since R(D) is generally equal to the source entropy when D = 0, this implies that for analog sources, the R(D) function becomes unbounded as D approaches zero.

As mentioned previously, discrete sources are of main interest in this thesis. They can be defined as

producing letters or messages taken from a finite alphabet $A = \{a_1, a_2, \ldots, a_N\}$. Associated with A is a probability set $P_A = \{p_1, p_2, \ldots, p_N\}$, where $p_i > 0$ and $\sum_{i=1}^{N} P_i = 1$. Any analog source can be discretized using one of the methods listed above. At this point, it is natural to examine the reasons for the emphasis being placed on discrete sources.

Discrete sources are gaining in both numbers and importance because of the advantages of digital transmission over analog, as well as the increasing availability of more powerful digital integrated circuit processing blocks, such as microprocessors. The design simplicity inherent in digital circuitry and the widening use of computers in handling all kinds of data processing and storing functions has also contributed to the increase in importance of discrete sources.

Unless otherwise indicated, the sources mentioned in this thesis will be assumed to be discrete memoryless sources [16]. In other words, the successive letters generated by a source are independent and identically distributed random variables. Of course, many other types of sources exist, such as Markov sources and composite sources, but these are beyond the scope of this study.

Many sources generate information containing a high proportion of redundancy. An example will illustrate this. Consider the ternary alphabet A = {0, 1, 2}, with the

probability set $P_A = \{\frac{2}{3}, \frac{1}{6}, \frac{1}{6}\}$. The entropy* H(P) of this source is then given by $H(P) = \frac{2}{3}\log_2 \frac{3}{2} + \frac{1}{6}\log_2 6 + \frac{1}{6}\log_2 6$ $\simeq$ 1.25 bits per digit. This means that, on the average, the information content is only 1.25 bits per ternary digit. If the three letters of the alphabet were equiprobable, then the entropy of the source would be $\log_2 3$ ($\simeq$1.58). This illustrates the case of maximum entropy, making the most efficient use of each digit in an information-theoretic sense. For the non-equiprobable case, then, the information content per ternary digit is less than $\log_2 3$. Since physically it is impossible to transmit a fraction of a digit, there is a redundancy of $\log_2 3 - H(P) \simeq 0.33$ bits per ternary digit. For example, in a block of 158 digits put out by the source, only 158 - 33 (= 125) digits carry information, while the remaining 33 digits are redundant.

This redundancy is unfortunately unstructured and consequently cannot be used in an advantageous way. On the other hand it is worthwhile reducing the redundancy. Hence, to transmit the same amount of information, a smaller number of digits per second can be transmitted, thereby reducing the bandwidth. Equivalently, in a given time, more information can be transmitted without requiring extra bandwidth. Hence, if to satisfy a given criterion it is necessary to transmit a k-tuple instead of an n-tuple (k < n)

---

*Note: This is using the logarithmic measure of information capacity. For other measures of information, see Hyvärinen [17].

put out by a source, then a compression ratio of $\frac{n}{k}$ is said to have been achieved.

Before concluding this introduction with an outline of the thesis, some definitions which will be used later in the thesis are given.

An n-tuple over a certain alphabet A is a sequence of n symbols taken from the alphabet. There are thus altogether N^n distinct n-tuples, where N is the number of elements in the alphabet.

A code can now be defined. Suppose each of the N^k possible distinct k-tuples made up of symbols drawn from A is converted into an n-tuple according to a coding rule. The set of N^k distinct n-tuples obtained thereby is said to be an (n,k) code. The coding rule used, and hence the values of the code parameters used depends on what the code is expected to achieve. The n-tuple, called a code word, implies that the code has a code length or block length n.

In order to average the effects of noise over large numbers of symbols and thus increase their effectiveness, codes are usually chosen to be long. This causes implementation problems since usually it is not feasible to store all the code words. Hence the interest in codes which can be conveniently generated as needed, instead of stored.

It should be noted that, although this thesis is mainly concerned with data compression, or, equivalently bandwidth compression, the emphasis is on the use of block

source coding techniques as opposed to the use of transform or other techniques. Hence the derived results are also useful in assessing the probability of decoding error for the codes under consideration.

An outline of the thesis can now be given.

Chapter II will contain a definition of the standard array and cosets, followed by a brief review of the better-known source coding procedures.

In Chapter III, negacyclic codes and their main features are examined. Then various cyclic binary coding theorems relevant to source encoding are extended to the negacyclic case in Chapter IV. This is followed by the description of a joint source and channel encoding scheme. Finally, the rate distortion function for a non-binary source relevant to negacyclic codes will be determined.

A few concluding remarks will be made in Chapter V.

Before beginning Chapter II, it may be mentioned that some of the results of this thesis have already been reported elsewhere [18],[19].

CHAPTER II

A REVIEW OF SOURCE CODING TECHNIQUES

1. Introduction

Source coding schemes [20] can be classified as either producing a constant data compression ratio or one which is dependent on the instantaneous output of the source. Source encoders of the first kind are said to be "transparent" in the sense that, for a fixed source encoder input rate, they produce a fixed output rate. They are also referred to as fixed-length to fixed-length coding schemes. This is highly important, since in this case the output of the encoder will be compatible with fixed-rate modems.

If the compression ratio is dependent on the source output, then the source encoder output rate will vary with time. In order to convert the variable-rate output of the encoder to fixed-rate data compatible with fixed-rate modems, some buffering will be required. If the variation of the encoder output rate is high, the amount of buffer storage required may be extensive.

Because of these problems, much attention has been given to fixed-rate source encoding techniques, and in particular, block-to-block encoding schemes, wherein a block

of n source output digits is compressed into a block of m source coder output digits with $m < n$, resulting in a compression ratio of $\frac{n}{m}$. Distortion is generally produced using this technique. Hence a scheme must be suitably designed so as to minimize the effects of this distortion on the eventual user.

Another consideration of major importance is that of source coder complexity. (This should not be confused with source coder instrumentability, as discussed in Berger [21].) This complexity can be expressed in terms of storage and computational requirements, these requirements being generally interchangeable, at least to some extent. In many applications [22] for which data compression is highly attractive, such as remote telemetry, source encoder complexity must be a minimum because of weight/space/power or other such limitations. The source decoder, however, can be more complex. This requirement of asymmetric complexity can be a very restrictive constraint, since the source encoder would naturally tend to be much more complex than the source decoder, and the source decoder simpler, as outlined in Chapter I.

There are many possible approaches to describing existing source coding schemes, depending on the primary characteristics to be considered. The approach here will loosely follow that of Tavares [23].

The discussion will begin with a description of the standard array, since it will be subsequently required to

explain some source coding schemes.

Distortionless coding schemes will be discussed next, followed by coding schemes where distortion is allowed. The study of source encoding with respect to a fidelity criterion, also known as rate distortion theory, serves as the theoretical basis for data compression [24]. Prior to this, source coding had generally consisted of entropy coding, examples of which are Shannon-Fano and Huffman coding. Since these schemes are distortionless, the concept of distortion was not required.

Unless otherwise stated, the discussion will apply to discrete memoryless sources.

2. The Standard Array [25]

The standard array for an (n,k) binary linear code is a partitioning of 2^n n-tuples into a 2^{n-k} by 2^k array.

The rows of the array are called cosets, and the leftmost n-tuple of each row is called a coset leader. The coset leaders are chosen to be of minimum Hamming weight. Some important properties of the standard array are:

(i) Every n-tuple appears once and only once in the array.

(ii) All 2^k n-tuples of a coset have the same syndrome.

(iii) The syndromes for different cosets are different.

A syndrome is an (n - k)-tuple which is zero when either a block is received without error or a block is transformed into another code word by channel errors. Syndromes are defined in Peterson [26].

The standard array for a given (n,k) binary group code can be used as a decoding table in the following fashion.

Let v_i be a transmitted code vector. Then if $v_i' = v_i + e_i$, where e_i is an error n-tuple, v_i' will be decoded correctly if and only if the error pattern e_i is a coset leader. If, say, e_i is the jth coset leader, then v_i' will match up with the array element located in the jth row, ith column, and will be decoded correctly as v_i. If e_i is not a coset leader, then the received vector will be decoded incorrectly. Hence the 2^{n-k} coset leaders are referred to as the correctable error patterns for the (n,k) code. These are chosen as the most likely error patterns to occur for the channel under consideration, in order to minimize the probability of decoding error.

For example, for the binary symmetric channel, the lowest Hamming weight error patterns are the most likely to occur, so that minimum weight n-tuples are chosen for the coset leaders.

The standard array is thus a table listing every code vector and the sums resulting from the addition of every code word with every correctable error pattern.

Some useful definitions relating to maximum coset leader weight can now be stated [27]. A t-error-correcting

linear code is said to be perfect if it has n-tuples of weight t or less as coset leaders, and none of greater weight.

A quasi-perfect code, on the other hand, is a t-error correcting linear code which has all n-tuples of weight t or less as coset leaders, some of weight (t + 1), and none of greater weight.

Another useful concept is that of code optimality. A binary group code is said to be optimum for the binary symmetric channel if its probability of error is as small as for any group code with the same length and the same rate.

Examples of perfect codes are the Golay (23,12) binary group code and the Hamming $(2^m - 1, 2^m - 1 - m)$ binary group code [28].

Examples of quasi-perfect codes are the 2-error-correcting BCH codes and the Golay (24,12) code. The Hamming codes and the 2-error-correcting BCH codes are also optimum.

The decoding scheme just described, using the standard array, requires the storage of the standard array and searching among 2^{n-k} array elements to find the one matching the received vector. However, properties (i) and (ii) of the standard array can be used to simplify the decoding scheme. Since there are 2^{n-k} distinct correctable error patterns for an (n,k) binary group code, and since each syndrome of an (n,k) code is a unique (n - k)-tuple, there exists a

one-to-one correspondence between a correctable error pattern (or coset leader) and the syndrome. So a decoding table consisting of the 2^{n-k} correctable error patterns and their corresponding syndromes can be formed. Hence, decoding of a received vector would consist of, first, calculating its syndrome and matching this with a syndrome in the decoding table. Then the error pattern corresponding to that syndrome is subtracted from the received n-tuple to yield the transmitted code vector. This scheme is known as table look-up decoding, and is much more rapid than using the standard array, since it involves a search among 2^{n-k} syndromes. However, for large values of (n-k), this method will not be practical. Hence the interest in other decoding schemes which may be more complex to implement but require little storage.

In this thesis, a noiseless channel is assumed, so that the problem of error correction will not be considered. In the next section, the Noiseless Coding Theorem is presented. It sets a lower bound for the average length of a coded source message.

3. The Noiseless Coding Theorem [29]

A discrete memoryless source which puts out messages $\{x_1, x_2, \ldots, x_M\}$ with probabilities $\{p_1, p_2, \ldots, p_M\}$ is assumed. To transmit each message x_i, it must be represented by a finite sequence of symbols, called a code word, chosen from the set of code characters

$\{a_1, a_2, \ldots, a_D\}$. The collection of all such words or vectors is called a code, and all code words are assumed to be distinct. As will be seen below, some restrictions must be placed on code word assignments so that no ambiguous code character sequences will be transmitted. A noiseless channel is further assumed, so that whatever code word is transmitted, it will be received without error. Hence, for a fixed transmission rate (in terms of code characters per unit time), efficient communication involves transmitting as few code characters as possible per message x_i. Thus, the average code word length, $\bar{L} = \sum_{i=1}^{M} p_i n_i$, is to be minimized, where n_i is the length of the code word corresponding to the message x_i.

Before the Noiseless Coding Theorem can be presented, some definitions are required in order to explain restrictions on code word assignments. The first concerns ambiguity. A code is said to be uniquely decipherable (or decodable) if every finite sequence of code characters corresponds to at most one message.

Let A be a finite sequence of code characters (i.e. a code word) for some code V. Let B and C also be code words of V. Then if B can be written as a concatenation of A and C, A is said to be a prefix of the sequence B. This leads to the definition of an instantaneous code being one in which no code word is a prefix of another word.

It is also true that, while every instantaneous code is uniquely decodable, the converse is not true.

Example 1

Table 1 below lists examples of non-uniquely decodable, instantaneous, and uniquely decodable but not instantaneous, binary codes.

Theorem 1
(The Noiseless Coding Theorem)

If $\bar{L} = \sum_{i=1}^{M} p_i n_i$ is the average code word length of a uniquely decipherable code for a given source X, then $\bar{L} \geq H(X)/\log D$, with equality iff $p_i = D^{-n_i}$ for $i = 1, 2, \ldots, M$.

It should be noted that for the case where $D = 2$, the above reduces to: $\bar{L} \geq H(X)$.

A lower bound for the average length of the encoded sequence is thus established and any code that achieves this lower bound is referred to as an absolutely optimal code.

The next theorem establishes the existence of a code whose average code word length $\bar{L}$ can be brought within one digit of the lower bound set by Theorem 1.

Theorem 2

Given that X is a random variable describing the output of a source with entropy $H(X)$, there exists a base D instantaneous code whose average code word length $\bar{L}$ satisfies

$$\frac{H(X)}{\log D} \leq \bar{L} < \frac{H(X)}{\log D} + 1$$

TABLE 1

EXAMPLES OF DIFFERENT TYPES OF SOURCE CODES

Source Message	Code Word (Code A)	Code Word (Code B)	Code Word (Code C)
x_1	0	0	0
x_2	11	11	01
x_3	111	101	011
x_4	1	100	111
Code Characteristic	Not uniquely decodable	Instantaneous (uniquely decodable)	Uniquely decodable (not instantaneous)

While the Noiseless Coding Theorem has set a lower bound on $\bar{L}$, it does not establish a construction rule for codes which will achieve, or at least approach, the lower bound. However [30], it is clear that in order to construct codes with $\bar{L}$ reasonably close to $H(X)/\log D$, the different code characters or symbols must occur with equal probability at each position of the code word. Also, the probabilities of occurrence of the code symbols at each position in the code words should be independent of all preceding symbols.

With this in mind, the next section presents two distortionless source code construction schemes.

4. Entropy Coding

Entropy coding [15] is a term used to describe various methods of coding the output of sources with unequal message probabilities so as to achieve a minimum average code word length. The best known of these are the Shannon-Fano and Huffman source coding schemes, both of which are distortionless.

a. Shannon-Fano Source Coding [30]

This source code construction scheme will yield absolutely optimal codes only when the message probabilities can all be expressed as negative powers of the number of code symbols.

The procedure for constructing code words is as follows. Given an arbitrary code alphabet made up of

D symbols, the message ensemble must be successively subdivided into D equiprobable groups. The first position of all the code words for the first group will contain the first code symbol, the first position of all the code words for the second group will contain the second code symbol, and so on until the Dth group. In the Dth group, code words will have the Dth code symbol in their first position.

Clearly, the subdivision can only be carried out exactly if all message probabilities can be expressed as negative powers of D. Otherwise, the groups will not be exactly equiprobable.

The next step consists in subdividing each of the D equiprobable groups into D equiprobable sub-groups. Then the first code symbol is assigned to the second position in all the code words of the first sub-group, the second code symbol to the second position in all the code words of the second sub-group, and so on until the Dth sub-group, in which all code words will have the Dth code symbol in the second position.

This process of subdivision is continued until all the sub-groups contain exactly one message.

The Shannon-Fano coding method is illustrated in the following example.

Example 2

The message ensemble $X = \{x_1, x_2, \ldots, x_{27}\}$ with the corresponding message probabilities $P = \{p_1, p_2, \ldots, p_{27}\}$

as given in Table 2 below is considered. These messages are to be encoded using the D code symbols {-1, 0, 1}, so that D = 3.

X is first subdivided into three equiprobable groups, the first consisting of x_1, x_2, x_3, the second of x_4 to x_{12}, the third of x_{13} to x_{27}. Thus the first position of the code words for the first group will contain a -1, the first position of the code words for the second group will contain 0, and that of the code words of the third group, -1, as shown in Table 2.

The next step is to subdivide each of these three groups into three equiprobable sub-groups. The three sub-groups of the first group then consist of $\{x_1\}$, $\{x_2\}$ and $\{x_3\}$ respectively. The three sub-groups of the second group contain $\{x_4, x_5, x_6\}$, $\{x_7, x_8, x_9\}$ and $\{x_{10}, x_{11}, x_{12}\}$ respectively, and the three sub-groups of the third group, $\{x_{13}, x_{14}, x_{15}\}$, $\{x_{16}, x_{17}, x_{18}\}$, $\{x_{19}, x_{20}, x_{21}, \cdots, x_{27}\}$ respectively.

This process of subdivision and assignment of code symbols is repeated until all sub-groups consist of only one message, yielding the set of code words of length 2, 3 and 4 illustrated in Table 2. It is clear that for each subdivision, each code symbol occurs equiprobably. Thus $\overline{L}$, the average length of the code words, is given by

$$\overline{L} = 3\left(\frac{1}{9}\right)\ (2) + 15\left(\frac{1}{27}\right)\ (3) + 9\left(\frac{1}{81}\right)\ (4) = 2.778$$

while $\frac{H(x)}{\log_2 D}$, the entropy of the source ensemble, is given by

TABLE 2

SHANNON-FANO SOURCE CODE FOR A 3-SYMBOL CODING ALPHABET

Message	Message Probabilities	Subdivisions 1st	2nd	3rd	4th	Code Word-Symbols Drawn from {-1,0,1}
x_1	$\frac{1}{9}$	}				-1 -1
x_2	$\frac{1}{9}$	}				-1 0
x_3	$\frac{1}{9}$	}				-1 1
x_4	$\frac{1}{27}$	}	}			0 -1 -1
x_5	$\frac{1}{27}$	}	}			0 -1 0
x_6	$\frac{1}{27}$	}	}			0 -1 1
x_7	$\frac{1}{27}$	}	}			0 0 -1
x_8	$\frac{1}{27}$	}	}			0 0 0
x_9	$\frac{1}{27}$	}	}			0 0 1
x_{10}	$\frac{1}{27}$	}	}			0 1 -1
x_{11}	$\frac{1}{27}$	}	}			0 1 0
x_{12}	$\frac{1}{27}$	}	}			0 1 1
x_{13}	$\frac{1}{27}$	}	}			1 -1 -1
x_{14}	$\frac{1}{27}$	}	}			1 -1 0
x_{15}	$\frac{1}{27}$	}	}			1 -1 1
x_{16}	$\frac{1}{27}$	}	}			1 0 -1
x_{17}	$\frac{1}{27}$	}	}			1 0 0
x_{18}	$\frac{1}{27}$	}	}			1 0 1
x_{19}	$\frac{1}{81}$	}	}	}		1 1 -1 -1
x_{20}	$\frac{1}{81}$	}	}	}		1 1 -1 0
x_{21}	$\frac{1}{81}$	}	}	}		1 1 -1 1
x_{22}	$\frac{1}{81}$	}	}	}		1 1 0 -1
x_{23}	$\frac{1}{81}$	}	}	}		1 1 0 0
x_{24}	$\frac{1}{81}$	}	}	}		1 1 0 1
x_{25}	$\frac{1}{81}$	}	}	}		1 1 1 -1
x_{26}	$\frac{1}{81}$	}	}	}		1 1 1 0
x_{27}	$\frac{1}{81}$	}	}	}		1 1 1 1

$$\frac{H(x)}{\log_2 D} = \frac{-3[\frac{1}{9}\log_2(\frac{1}{9})] - 15[\frac{1}{27}\log_2(\frac{1}{27})] - 9[\frac{1}{81}\log_2(\frac{1}{81})]}{\log_2 3} = 2.778$$

The compression ratio CR achieved with Shannon-Fano encoding can easily be computed once $\bar{L}$ is known. This is best illustrated by computation of the CR for Example 2. Since $M = 2^3$, then all original messages $\{x_1, x_2, \ldots, x_{27}\}$ can be represented by a block of 3 symbols taken from the set of D code symbols $\{-1,0,1\}$. This would then be the length of all messages prior to source encoding. Thus the CR would be given by

$$CR = \frac{\text{block length prior to encoding}}{\bar{L}} = \frac{3}{2.778} = 1.08$$

Clearly, the maximum value for CR will occur whenever $\bar{L}$ is a minimum, but from the Noiseless Coding Theorem, the source entropy is a lower bound on $\bar{L}$, so that for Example 2,

$$CR_{max} = \frac{\text{block length prior to encoding}}{\text{source entropy}} = \frac{3}{2.778} = 1.08$$

This is to be expected, since the code of Example 2 is absolutely optimal.

Thus, since the average code word length $\bar{L}$ has achieved the lower bound set by the Noiseless Coding Theorem, the code in this example is absolutely optimal. However, this source coding procedure does not guarantee an absolutely optimal source code, or even an optimal code.

In the next section, a source coding technique is presented which does in fact yield an optimal source code, in that no other code will have a smaller average number of symbols per message. Thus the code will always be optimum, but not necessarily absolutely optimal.

b. Huffman Source Coding

Huffman [31] discovered a systematic procedure which always yields an optimum code, in the sense that no other code has a smaller average code word length for the given message ensemble.

Briefly, the coding procedure is as follows. Given an ensemble X of M messages $\{x_1, x_2, \ldots, x_M\}$ occurring with probabilities $P = \{p_1, p_2, \ldots, p_M\}$, each message is to be expressed as a code word made up of elements drawn from an alphabet of D symbols. The messages will be grouped together m_0 messages at a time, where m_0 is the integer satisfying the two requirements

$$2 \leqslant m_0 \leqslant D$$

$$\frac{M - m_0}{D-1} = \text{positive integer.}$$

The m_0 least probable messages are grouped together, and the sum of m_0 message probabilities is used as the probability for this grouped message.

A reduced ensemble of messages is now constructed in order of non-increasing probability, with the grouped

message acting as a single message in the reduced ensemble.

New "grouped" messages are then formed by taking the D least probable messages of the reduced ensemble, and using this as a new group to form a new auxiliary ensemble.

This is repeated successively until the auxiliary ensemble consists of but D grouped messages, which leads to a final auxiliary ensemble consisting of one message with a unit probability.

The steps briefly described above are equivalent to forming a tree of which the original messages are the terminal nodes. If, to each branch of the tree, a code symbol is assigned, then a code word for every message can be written by beginning at the root and following the branches to the terminal node associated with a particular message.

The Huffman source coding procedure is illustrated in the following example.

Example 3

Given the M = 27 messages of Example 2, an optimal code is to be constructed using the D code symbols $\{-1,0,1\}$.

The 27 messages are first arranged in order of non-increasing probability. Since M = 27, and D = 3, $m_0 = 3$. So the first auxiliary message ensemble is formed by grouping the three least probable messages, and assigning to the group message a probability equal to the sum of the

probabilities of the three constituent messages. The auxiliary ensemble is then reordered so that it will be in nonincreasing order.

Since D = 3, in the next step the three least probable messages are again grouped in the auxiliary ensemble, assigning the group message a probability as in the previous step, and reordering the auxiliary ensemble. This is continued 11 times, until only one message is left, occurring with a unit probability. This is illustrated in Figure 3. Starting at the root of the graph of Figure 3 and working up the tree to the original messages, the code words can be constructed. In this case, the set of code words resulting from Huffman encoding is given in Table 3. From this table, it is clear that the average code word length is:

$$\bar{L} = 3(\frac{1}{9})\ (2) + 15(\frac{1}{27})\ (3) + 9(\frac{1}{81})\ (4) = 2.778$$

From Example 2, it is known that $\frac{H(X)}{\log_2 3}$ for the original message ensemble is equal to 2.778, so that this code is absolutely optimal. This is to be expected, since the Huffman encoding procedure always produces an optimum code, and the code constructed using the Shannon-Fano encoding technique was absolutely optimal.

The compression ratio achieved using Huffman coding can easily be computed once $\bar{L}$ is known.

As for the case of Shannon-Fano coding, the CR of Example 3 would be given by:

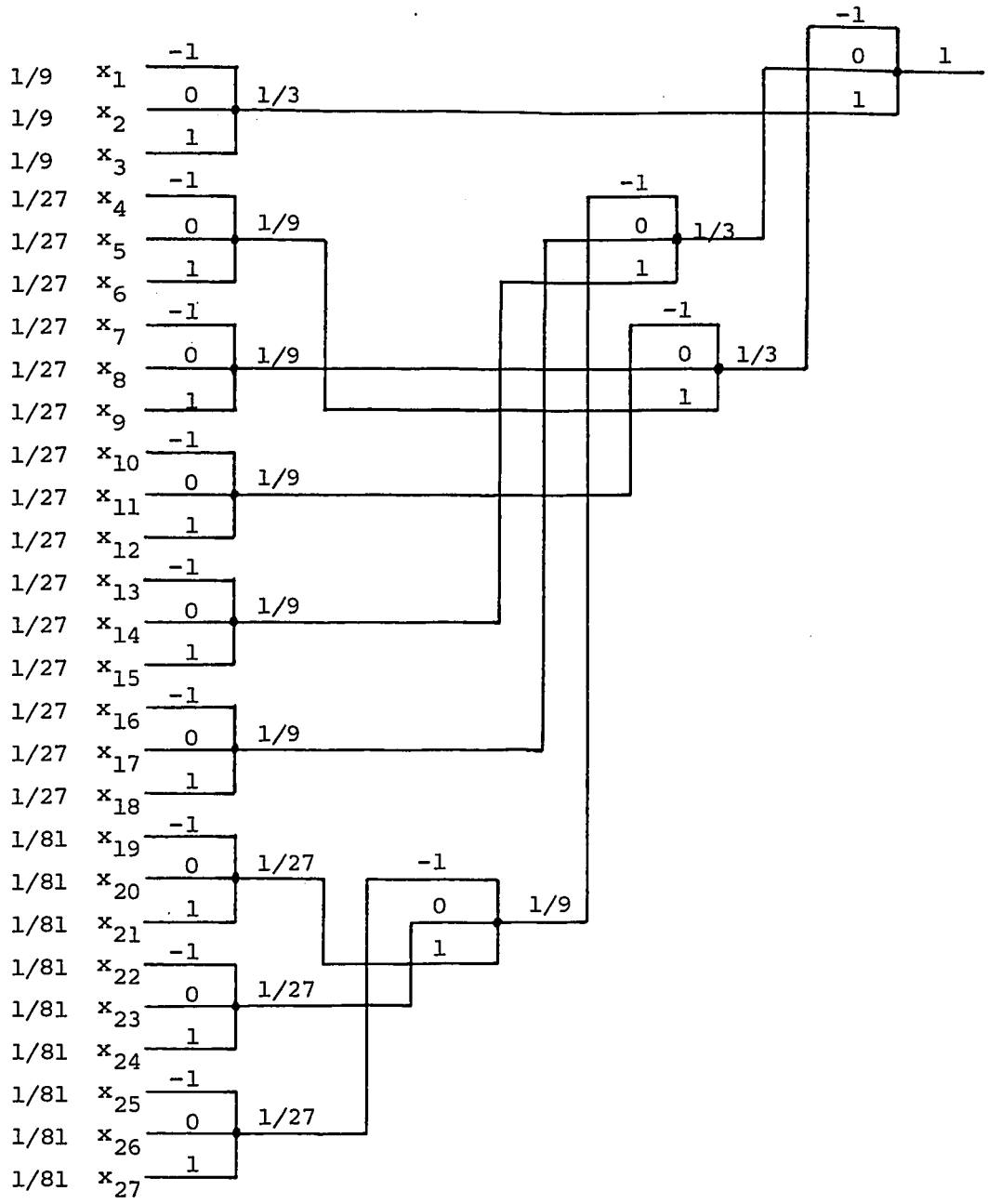

Figure 3:- Tree obtained by Huffman source coding of the 27 messages of Example 2. (The message probabilities are given in Table 3.)

TABLE 3

HUFFMAN SOURCE CODE FOR A 3-SYMBOL CODING ALPHABET

Message	Message Probability	Code Word
x_1	$\frac{1}{9}$	1 -1
x_2	$\frac{1}{9}$	1 0
x_3	$\frac{1}{9}$	1 1
x_4	$\frac{1}{27}$	-1 1 -1
x_5	$\frac{1}{27}$	-1 1 0
x_6	$\frac{1}{27}$	-1 1 1
x_7	$\frac{1}{27}$	-1 0 -1
x_8	$\frac{1}{27}$	-1 0 0
x_9	$\frac{1}{27}$	-1 0 1
x_{10}	$\frac{1}{27}$	-1 -1 -1
x_{11}	$\frac{1}{27}$	-1 -1 0
x_{12}	$\frac{1}{27}$	-1 -1 1
x_{13}	$\frac{1}{27}$	0 1 -1
x_{14}	$\frac{1}{27}$	0 1 0
x_{15}	$\frac{1}{27}$	0 1 1
x_{16}	$\frac{1}{27}$	0 0 -1
x_{17}	$\frac{1}{27}$	0 0 0
x_{18}	$\frac{1}{27}$	0 0 1
x_{19}	$\frac{1}{81}$	0 -1 1 -1
x_{20}	$\frac{1}{81}$	0 -1 1 0
x_{21}	$\frac{1}{81}$	0 -1 1 1
x_{22}	$\frac{1}{81}$	0 -1 0 -1
x_{23}	$\frac{1}{81}$	0 -1 0 0
x_{24}	$\frac{1}{81}$	0 -1 0 1
x_{25}	$\frac{1}{81}$	0 -1 -1 -1
x_{26}	$\frac{1}{81}$	0 -1 -1 0
x_{27}	$\frac{1}{81}$	0 -1 -1 1

$$CR = \frac{\text{block length prior to encoding}}{\bar{L}} = \frac{3}{2.778} = 1.08$$

As for Example 2,

$$CR_{max} = \frac{\text{block length prior to encoding}}{\text{source entropy}} = \frac{3}{2.778} = 1.08$$

Again, this is to be expected, since in Example 2, it was shown that $\bar{L}$ was equal to the source entropy.

Both Shannon-Fano and Huffman [32] codes have non-uniform sequence lengths and exhibit the prefix property, i.e., no sequence can be obtained by concatenating terms of one sequence with another.

Shannon-Fano coding is very efficient for large alphabets but can be significantly inferior to Huffman coding for short alphabets.

Shannon-Fano and Huffman coding are both examples of schemes in which fixed-length message blocks are converted to variable length code words. This type of scheme is susceptible to loss of synchronization and buffer overflow. However important these considerations may be in practice, they are beyond the scope of this thesis and will not be considered.

Another block to variable-length distortionless coding technique is Elias' sequential coding scheme [23]. While it does not possess the prefix property, it is uniquely decodable and can be more easily implemented than either of the two schemes just described.

In the next section, two variable-length to fixed-length (or block) distortionless encoding schemes are considered. They are syndrome encoding (which includes run-length coding) and fixed-weight coding.

5. Syndrome Source Coding [23]

As seen in Section 2, there exists a one-to-one relationship between the set of correctable errors for a code and the set of syndromes of the code. Hence any message (or source) block which is equivalent to the sum of an all-zero code word and a correctable error pattern can be represented by the syndrome for that error pattern, and the syndrome transmitted instead of the message block. This technique is only useful where the source symbol probabilities are highly unequal.

There are two variations of this method. The first variation is described by using as an illustration the case of a binary discrete memoryless source with $p(0) >> p(1)$. The source output is first partitioned into n-tuples, where n and k are parameters of an (n,k) binary group code. If there are t-1 or less 1's in the n-tuple, then the (n-k)-tuple (or syndrome) representing that error pattern is transmitted, thus giving a CR of $(\frac{n}{n-k})$.

If there are, however, t or more 1's in a source n-tuple, enough zeros are appended to the portion of the n-tuple containing the t 1's to form an n-tuple of weight t, which has of course an (n-k)-tuple as syndrome. The

rest of the 1's in this n-tuple are then used to begin a new n-tuple, formed as above. Thus the CR for these blocks will be less than $\frac{n}{n-k}$, and may be less than unity. At the receiver, the decoder discards the appended zeros whenever an n-tuple of weight t is decoded. If the weight of the decoded n-tuple is (t-1) or less, the entire n-tuple is retained.

One special case of syndrome encoding is known as run-length encoding. This discussion is confined to the binary case, but can be extended, with suitable modifications, to non-binary sources. For a binary source, the appearance of a run or string of zeros will be terminated by a one, and vice-versa, so that a string of length j corresponds to (j-1) zeros terminated by a one. Therefore, a run of length zero corresponds to the occurrence of a one. If $p(0) >> p(1)$, it is more efficient to code the run lengths of zeros as opposed to the zeros in the runs.

If the probability distribution of the different run lengths are known, then a Huffman code can be used to efficiently encode the runs. However, this code, while optimal for the given probability distribution, will not be optimal for a different distribution.

Thus, a suboptimal code which is much less sensitive to the run length distribution is required. One approach would be to represent each possible run length by a unique fixed-length code word. This is inefficient, since the code words would be too long.

A second approach involves representing all possible runs up to a certain maximum length by a non-zero k-tuple. Since there are 2^k k-tuples, they could represent runs from length 1 to length $(2^k - 1)$. The problem in this scheme lies in the choice of k. If it is chosen too small, it will not be able to represent the longer runs, while if chosen too large, it will be inefficient in representing the shorter runs. This problem may be reduced by letting the k-tuple of all zeros represent an unterminated run of 2^k zeros, so that runs longer than $2^k - 1$ can be represented by a succession of k-tuples.

Still another scheme [33] has been to classify the runs as type 1 or type 2. Type 1 runs are those terminating with a one, while type 2 runs are unterminated runs of zeros of differing lengths. These differing lengths are all multiples of the shortest unterminated run.

It has been shown [34] that run-length coding for a discrete memoryless source is a special case of syndrome source coding using a $(2^m - 1, 2^m - 1 - m)$ single-error correcting Hamming code. The maximum run length that can be encoded using this scheme is $2^m - 1$.

There are many other types of distortionless source coding schemes, such as variations on syndrome coding, fixed weight coding and transition image coding. The latter is relevant to discrete sources with memory, but still requires the use of another source coding scheme once the transition sequence has been obtained.

All of the source-encoding techniques discussed so far have been distortionless, i.e., the source output is reproduced exactly at the receiver, given a noiseless channel. However, if some distortion can be tolerated, then the compression ratio can be greater. In the next section, some of the source coding techniques which allow distortion are discussed.

6. Source Coding with Distortion

As already stated, in block coding for channel error control, a k-digit information sequence is converted into an n-digit sequence. In the so-called systematic coding, the (n-k) check digits are determined according to some coding rule and are simply appended to the k information digits. One way to make use of this in source coding is as follows.

Let $a_0, a_1, a_2, \ldots, a_{n-1}$ be an n-tuple put out by the source. Instead of transmitting all of the n digits, only the first k are transmitted, so that a CR of $\frac{n}{k}$ is achieved. At the receiver, the (n-k) check digits are computed and appended to the k digits to get, say, $a_0, a_1, a_2, \ldots, a_{k-1}, b_k, b_{k+1}, \ldots, b_{n-1}$. Thus the source and received n-tuples differ in at most the (n-k) last positions, and the distortion is confined to those digits. The difference between the two sequences is measured in some metric and divided by n to obtain the average distortion per digit.

In the case of binary sources, the maximum possible distortion using systematic coding is thus (n-k), and the minimum is, of course, zero. Hence the average distortion per digit will be given by $\frac{(n-k)}{2n}$, because of the linearity of the code.

Instead of using systematic codes, nonsystematic codes can be used in the same way as above. If a set of k-tuples is coded into a set of n-tuples, this set is called a code, and every n-tuple is a word of the code. In the case of nonsystematic codes, the n-tuples are made up of k scrambled information digits and (n-k) scrambled check digits. Hence, given the n-tuple $a_0, a_1, \ldots, a_{n-1}$ emerging from the source, the codeword nearest to it with respect to some metric is chosen. This can be done using the standard array. Assume the chosen n-tuple is $a_0, a_1, \ldots, a_{n-1}$. Then the k-tuple corresponding to this code word is determined and transmitted and $b_0, b_1, \ldots, b_{n-1}$ is reconstructed at the receiver. Then the difference (or distance) between $a_0, a_1, \ldots, a_{n-1}$ and $b_0, b_1, \ldots, b_{n-1}$ will be the distortion for that block, and it will be distributed among the n digits of the n-tuple.

If a perfect code is used, then because of the discussion in Section 2, it is clear that the maximum distortion would be $\frac{t}{n}$, where t is the random-error-correcting capability of the code, while a quasi-perfect code would have a maximum distortion of $\frac{(t+1)}{n}$.

Also, if Hamming $(2^m - 1, 2^m - 1 - m)$ perfect codes are used, the resulting compression ratio will be given by

$$CR = \frac{2^m - 1}{2^m - 1 - m}$$

Similarly, for the 2-error-correcting BCH codes, which are quasi-perfect, the compression ratio would be

$$CR = \frac{2^m - 1}{2^m - 1 - 2m}$$

The techniques which apply to block codes can usually be extended to tree codes, with appropriate modifications [35].

Many other nonblock channel coding techniques have been successfully applied to source coding. One such technique is known as sliding-block source coding [36]. It includes as special cases many other nonblock algorithms, including quantization schemes, tree code schemes, trellis code schemes, convolutional source coders with sequential channel decoders used as source decoders, etc.

This very brief review of some of the better-known source coding techniques is incomplete. More information on source coding can be obtained by consulting Wilkins and Wintz's bibliography on data compression [37] and the Special Issue on Redundancy Reduction, Proceedings IEEE, Vol. 5., no. 3, March 1967, as well as Tavares [23].

CHAPTER III

NEGACYCLIC CODES

1. Introduction [38],[39],[40]

While there is a natural division between block and tree codes, linear block codes can themselves be further classified as to their cyclic or noncyclic nature.

Negacyclic codes [41], as their name might suggest, are closely related to cyclic error-correcting codes. In this chapter, negacyclic codes will first be defined, their features explained and their parallels with cyclic codes pointed out. Then some relevant binary coding theorems will be extended to the non-binary case. This will be followed by the description of a novel joint source and channel encoding scheme, and the derivation of some of the scheme's parameters. Finally, the rate distortion function of a non-binary source relevant to negacyclic codes is determined.

2. Definition of Negacyclic Codes [40],[41]

Since the weight function relevant to negacyclic codes is the Lee weight, the discussion begins with its definition.

The Lee weight of a sequence $B = b_0, b_1, b_2, \ldots, b_{n-1}$ over GF(p), where p is an odd prime, is the sum of

the Lee weights of the b_i's. The Lee weight ω_i of b_i is $|b_i|$ if $0 \leq b_i \leq \frac{p-1}{2}$ and $|p - b_i|$ if $\frac{p-1}{2} < b_i \leq p - 1$. This should be contrasted with the Hamming weight ω_H of B, which is defined as the sum of the Hamming weights of the b_i's. The Hamming weight of b_i is 1 if $b_i \neq 0$ and is 0 if $b_i = 0$. Thus the Hamming weight is a measure of the number of nonzero digits in a sequence. This is illustrated in the following example.

Example 4

(a) Let B = 0 1 -1 1 0 1 1 0 1 -1 0 1, with p = 3. Then the Lee weight of B is 8, and its Hamming weight is also 8, since it contains 8 nonzero digits, all of which are ±1.

(b) Let B = 0 1 3 2 1 1 0 0 1 3 1 1, with p = 5. Then the Lee weight of B is 12, while its Hamming weight is 9.

It can be remarked at this point that from the definition of the Lee weight, a sequence $B = b_0, b_1, b_2 \ldots, b_{n-1}$ over GF(p), where p is an odd prime, can be treated equivalently as being over the set
$S = \{-\frac{(p-1)}{2}, -\frac{(p-3)}{2}, \ldots, -2, -1, 0, 1, 2, \ldots, \frac{p-3}{2}, \frac{p-1}{2}\}$.
In all further discussions, B will be assumed to be over S.

The Hamming and Lee metrics are not the only error measures which have been defined. A general definition of error metrics is given in Golomb [42].

A weight function, or metric, must assign low weights to likely error words, and high weights to unlikely error words. Thus, the choice of weight function is dependent on the modulation scheme used in a given communications system, which along with the signal set to be used, is, in general, dependent on the channel characteristics. The Hamming metric is particularly suitable for orthogonal modulation schemes, since it can be shown that the probability of occurrence of an error word depends only on the number of nonzero digits in the error word, and not on the values of those digits. However, it is not a suitable weight function for other modulation schemes, such as phase modulation, where the probability of obtaining a small phase error is much greater than that of obtaining a large phase error. The Lee weight function, however, is a metric which is well-suited to phase modulation schemes.

It is interesting to note that for a system using amplitude modulation to transmit information via a channel containing additive Gaussian noise, neither the Hamming nor the Lee metric is particularly suited. The Lee metric does, however, provide a much better approximation than the Hamming metric, if the number of symbols in the alphabet is large.

The codes which make use of the Lee metric are the negacyclic codes invented by Berlekamp. These can now be defined.

A negacyclic code over GF(p), where p is an odd prime, and of length n, where n is a nonmultiple of p, is the set of multiples of a generator polynomial g(x) which divides $X^n + 1$ over GF(p).

A few remarks can be made at this point. First, it can be noted that a cyclic code of length n is the set of multiples of a generator polynomial g(x) which divides $X^n - 1$. Secondly, since the metric used for negacyclic codes is the Lee metric, the code will be considered to be over the set S, rather than GF(p), from this point on.

A negacyclic shift B' of an n-tuple $B = b_0, b_1, b_2, \ldots, b_{n-1}$ over S is the sequence $B' = -b_{n-1}, b_0, b_1, \ldots, b_{n-2}$. It should be noted that only the b_{n-1} term of the n-tuple is negated. All other digits are merely shifted. Since g(x) divides $X^n + 1$, if B belongs to a negacyclic code, then B' will also be a code word. B and all of its distinct negacyclic shifts constitutes a negacyclic class, while B and all of its distinct cyclic shifts forms a cyclic class.

It is of interest to note that a cyclic shift of an n-tuple $B = b_0, b_1, b_2, \ldots, b_{n-1}$ is the sequence $B' = b_{n-1}, b_0, b_1, \ldots, b_{n-2}$, which is parallel to that of the negacyclic shift, but without the negation of the b_{n-1} term. This explains how the name of negacyclic codes arose.

Before giving examples of cyclic and negacyclic classes, a theorem relevant to negacyclic classes is presented.

Theorem 3 [43]

The number of distinct elements in a negacyclic class of a (n,k) negacyclic code V can only be 2n or a factor of 2n.

Example 5

(a) Let B = 0 1 1 1 1 1 0 0 1 1 1 1, with n = 12. Then B and its (n-1) cyclic shifts are:

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

(b) Let B = 0 1 -2 2 1 1 0 0 1 -2 1 1, with n = 12. It should be noted that this is the same sequence as in Example 4, except that in this case, B is over S = {0, ±1, ±2} instead of over GF(p). Then B and its (2n - 1) negacyclic shifts are:

0	1	-2	2	1	1	0	0	1	-2	1	1
-1	0	1	-2	2	1	1	0	0	1	-2	1
-1	-1	0	1	-2	2	1	1	0	0	1	-2
2	-1	-1	0	1	-2	2	1	1	0	0	1
-1	2	-1	-1	0	1	-2	2	1	1	0	0
0	-1	2	-1	-1	0	1	-2	2	1	1	0
0	0	-1	2	-1	-1	0	1	-2	2	1	1
-1	0	0	-1	2	-1	-1	0	1	-2	2	1
-1	-1	0	0	-1	2	-1	-1	0	1	-2	2
-2	-1	-1	0	0	-1	2	-1	-1	0	1	-2
2	-2	-1	-1	0	0	-1	2	-1	-1	0	1
-1	2	-2	-1	-1	0	0	-1	2	-1	-1	0
0	-1	2	-2	-1	-1	0	0	-1	2	-1	-1
1	0	-1	2	-2	-1	-1	0	0	-1	2	-1
1	1	0	-1	2	-2	-1	-1	0	0	-1	2
-2	1	1	0	-1	2	-2	-1	-1	0	0	-1
1	-2	1	1	0	-1	2	-2	-1	-1	0	0
0	1	-2	1	1	0	-1	2	-2	-1	-1	0
0	0	1	-2	1	1	0	-1	2	-2	-1	-1
1	0	0	1	-2	1	1	0	-1	2	-2	-1
1	1	0	0	1	-2	1	1	0	-1	2	-2
2	1	1	0	0	1	-2	1	1	0	-1	2
-2	2	1	1	0	0	1	-2	1	1	0	-1
1	-2	2	1	1	0	0	1	-2	1	1	0

Following the usage in the case of cyclic codes, a negacyclic code of block length $n = \frac{p^r - 1}{2}$, where r is any

positive integer, is said to be a primitive negacyclic code. On the other hand, if the block length is a divisior of $n = \frac{p^r - 1}{2}$, then the code is said to be a nonprimitive negacyclic code.

The error-correcting capability of negacyclic codes can now be investigated. Before introducing the relevant theorem, it is useful in passing to note that, since

$$X^{2n} - 1 = (X^n + 1)(X^n - 1)$$

the roots of $X^n + 1$ are the roots of $X^{2n} - 1$ which are not roots of $X^n - 1$. Also, if α is a primitive root of $X^{2n} - 1$, then all odd powers of α will be roots of $X^n + 1$, while all even powers of α will be roots of $X^n - 1$ [41]. In addition to this, if $2n = p^r - 1$, then every irreducible factor of $X^{2n} - 1$ over GF(p) has degree r or less. Hence every irreducible factor of $X^n - 1$ and $X^n + 1$ must also have degree r or less.

$$\text{Since } h(X) = \frac{X^n + 1}{g(X)}$$

then deg[h(X)] + deg[g(X)] = n, where h(X) and g(X) are the parity-check and generator polynomials, respectively, and every irreducible factor of h(X) and g(X) must have degree r or less.

The most important result concerning the error-correcting capability of negacyclic codes is contained in the following known theorem [41].

Theorem 4

> With α as a primitive root of $X^{2n} - 1$, if the roots of the generator polynomial of a negacyclic code over S includes α, α^3, α^5, . . ., α^{2t-1}, where $2t - 1 < p$, then the code can correct all error patterns of Lee weight t or less.

Recalling the rule of construction of BCH codes, it is clear that a parallel exists between BCH codes and negacyclic codes. However, BCH codes do not have the restriction that $t \leqslant \frac{p - 1}{2}$.

This theorem provides a rule for constructing g(X), the generator polynomial of the negacyclic code, requiring only α and its odd powers to α^{2t-1}. Clearly, α must be a root of $X^n + 1$, since it has order 2n. Hence, it is necessary to factor $X^n + 1$ into irreducible factors in order to obtain g(X). This has been done for p = 5 to 13, with r = 2 and 3 [44].

The information rate of negacyclic codes can now be examined. Since every irreducible factor of $X^n + 1$ has degree r or less, because $n = \frac{p^r - 1}{2}$, the degree of the generator polynomial is $\leqslant rt$. Since $t \leqslant \frac{p - 1}{2}$, then

$$\deg [g(X)] \leqslant r(\frac{p - 1}{2})$$

and the information rate will be given by [45]

$$\frac{k}{n} = \frac{n - (n-k)}{n} \geqslant \frac{n - rt}{n} \geqslant 1 - \frac{r(p - 1)}{2(p^r - 1)} \quad (2)$$

which reduces to

$$\frac{k}{n} \geq 1 - \frac{r(p-1)}{p^r - 1}$$

Taking $t = \frac{p-1}{2}$ to achieve the maximum error-correction capability, the expression for information rate can be expressed as

$$\frac{k}{n} \geq 1 - \frac{2rt}{(2t+1)^r - 1}$$

For the case of $r = 2$, this reduces to

$$\frac{k}{n} \geq \frac{p-1}{p+1}$$

which gives a lower bound on the information rate of an (n,k) $r = 2$ primitive negacyclic code.

It should also be pointed out that negacyclic codes do not have any particular advantage over binary cyclic codes for burst error correction, since the significant variable for burst error correction is the length of the burst, and not its weight.

At this point, it is useful to examine the different error patterns which are correctable by a negacyclic code V for which $t = 2$ and $p = 5$. This implies that $t \leq \frac{p-1}{2}$, and for equality, $t = 2$. Hence V can correct any error pattern of two or less errors whose total Lee weight does not exceed 2. These would include all patterns of the form $b_i X^i$, where $|b_i| = 0$, 1 or 2 and of the form $b_i X^i + b_j X^j$,

where $|b_i| + |b_j| \leq 2$.

A negacyclic code can now be constructed. This is illustrated in the following example.

Example 6

Let $n = \frac{p^r - 1}{2} = 12$, so that $p = 5$, $r = 2$, $k = n - rt = 8$, and $t = \frac{p - 1}{2} = 2$. Then $GF(5^2)$ can be constructed using $X^2 + X + 2$, a primitive polynomial of degree $r = 2$. This yields the following:

$GF(5^2)$ in terms of α, a root of $X^2 + X + 2$:

$$\alpha^0 = 1 + 0$$
$$\alpha^1 = 0 + \alpha$$
$$\alpha^2 = -2 - \alpha$$
$$\alpha^3 = +2 - \alpha$$
$$\alpha^4 = +2 - 2\alpha$$
$$\alpha^5 = -1 - \alpha$$
$$\alpha^6 = +2 + 0$$
$$\alpha^7 = 0 + 2\alpha$$
$$\alpha^8 = 1 - 2\alpha$$
$$\alpha^9 = -1 - 2\alpha$$
$$\alpha^{10} = -1 + \alpha$$
$$\alpha^{11} = -2 - 2\alpha$$
$$\alpha^{12} = -1 + 0$$
$$\alpha^{13} = 0 - \alpha$$
$$\alpha^{14} = +2 + \alpha$$
$$\alpha^{15} = -2 + \alpha$$

$$\alpha^{16} = -2 + 2\alpha$$
$$\alpha^{17} = 1 + \alpha$$
$$\alpha^{18} = -2 + 0$$
$$\alpha^{19} = 0 - 2\alpha$$
$$\alpha^{20} = -1 + 2\alpha$$
$$\alpha^{21} = 1 + 2\alpha$$
$$\alpha^{22} = 1 - \alpha$$
$$\alpha^{23} = 2 + 2\alpha$$
$$\alpha^{24} = 1 + 0 = \alpha^{0}$$

Now g(X) must have as roots all odd powers of α up to α^{2t-1}, which is α^3 in this case. Since α is a root of $X^2 + X + 2$, another irreducible polynomial of degree ≤ 2 which has α^3 as a root is required. From the GF(5^2) just constructed, it is clear that $\alpha^6 = (\alpha^3)^2 = 2$, is such a root, so that $X^2 - 2$ is a polynomial of degree 2 which has α^3 as a root. Hence,

$$g(X) = (X^2 + X + 2)(X^2 - 2)$$
$$= X^4 + X^3 - 2X + 1$$

contains roots α and α^3 and will generate a (12,8) 2-error correcting primitive negacyclic code.

The information rate for this code would be

$$\frac{k}{n} = \frac{p - 1}{p + 1} = \frac{2}{3}, \text{ since } r = 2.$$

Only primitive negacyclic codes with $n = \frac{p^r - 1}{2}$, have been covered. However, there are also nonprimitive negacyclic codes. While the better negacyclic codes, in the sense of having a relatively large error-correcting capability and a relatively small number of check digits, usually occur in primitive block lengths, there are also some moderately good nonprimitive negacyclic codes.

If a negacyclic code V has length n, this means that g(X) divides $1 + X^n$. Hence g(X) must also divide $1 - X^{2n}$. Therefore $2n = p^r - 1$, or $n = \frac{p^r - 1}{2}$, for primitive negacyclic codes. Nonprimitive negacyclic codes are codes for which $n \neq \frac{p^r - 1}{2}$.

In the next section, the close parallels between negacyclic and cyclic codes are studied.

3. Parallels between Negacyclic and Cyclic Codes

It is now clear that there is a great deal of similarity between cyclic codes and negacyclic codes. Some of the more obvious similarities are:

(i) The generator polynomial of a cyclic code divides $X^n - 1$, while that of a negacyclic code divides $X^n + 1$.

(ii) The construction technique for negacyclic codes as outlined in Berlekamp [41] and Theorems III and IV follows closely that of BCH codes, which are cyclic codes.

(iii) While cyclic codes can be decomposed into cyclic classes, negacyclic codes can be decomposed into negacyclic classes.

(iv) While binary cyclic codes can be implemented using shift registers with appropriate feedback connections, negacyclic codes could be implemented using either multi-level shift registers, or, what would be more realistic, through hardware or software simulation of a multi-level shift register.

(v) Negacyclic codes can be decoded using the BCH decoding procedure [41] with appropriate modifications, and are, under appropriate restrictions, 1-step permutation decodable [46].

(vi) Since majority logic decoding is one of the simplest decoding techniques known, and since most majority-logic decodable codes found so far are cyclic, one would naturally wish to find negacyclic codes, or codes derived from them, which are majority-logic decodable. However, no general results seem available, though one subset of a (12,8) primitive negacyclic code with t = 2 is majority-logic decodable. The decoding procedure for this code will be illustrated by way of an example [47].

Example 7

Let the generator polynomial of a primitive (12,8) t = 2 negacyclic code be

$$g(X) = (X^2 - 2X + 3)(X^2 + 2) = X^4 - 2X^3 + X + 1$$

Let V be the (11,4) shortened subset of the negacyclic code, generated by

$$\begin{aligned} g'(X) &= g(X)(1 - X + 2X^3) \\ &= 1 - X^2 - X^5 + X^6 + 2X^7 \end{aligned}$$

Since V is negacyclic, every word V(X) of V may be expressed in the form

$$\begin{aligned} V(X) &= g(X)m(X) \\ &= m_0 + m_1X + (m_2 - m_0)X^2 + (m_3 - m_1)X^3 - m_2X^4 \\ &\quad - (m_0 + m_3)X^5 + (m_0 - m_1)X^6 + (xm_0 + m_1 - m_2)X^7 \\ &\quad + (2m_1 + m_2 - m_3)X^8 + (2m_2 + m_3)X^9 + 2m_3X^{10} \end{aligned}$$

Any received polynomial R(X) can be expressed as $R(X) = r_0 + r_1 + r_2X^2 + \ldots + r_{10}X^{10}$.

In order to decode R(X), the corresponding coefficients of V(X) and R(X) must be equated, yielding,

$$
\begin{aligned}
r_0 &= m_0 \\
r_1 &= m_1 \\
r_2 &= -m_0 + m_2 \\
r_3 &= -m_1 + m_3 \\
r_4 &= -m_2 \\
r_5 &= -m_0 - m_3 \\
r_6 &= m_0 - m_1 \\
r_7 &= 2m_0 + m_1 - m_2 \\
r_8 &= 2m_1 + m_2 - m_3 \\
r_9 &= 2m_2 + m_3 \\
r_{10} &= 2m_3
\end{aligned}
\qquad (1)
$$

where it should be noted that the above equations are independent of m(X).

The above equations can be combined to yield

$$
\begin{aligned}
2m_0 &= 2r_0 \\
2m_0 &= 2r_2 - 2r_4 \\
2m_0 &= 2r_1 + 2r_6 \\
2m_0 &= -2r_5 - r_{10} \\
2m_0 &= r_7 + 2r_8 + 2r_9
\end{aligned}
$$

In the set of equations (1) above, m_0 is estimated by a majority vote. In other words, m_0 is considered to be correct if it agrees with at least 3 out of 5 equations in the set above.

After obtaining m_0, $m_0 g(X)$ can be subtracted from R(X), and the sum can be shifted to the left by one

position so that

$$R(X)' = r_0' + r_1'X + r_2'X^2t \ . \ . \ . \ tr'_{n-2}X^{n-2}$$

Hence, m_1 can now be estimated by combining equations from (1) and using r_j' instead of r_j.

This process can be repeated for all m_i's in $m(X)$, and so it is concluded that the (11,4) shortened subset of the negacyclic code is 1-step majority logic decodable.

In the following chapter, it will be seen that many of the theorems for binary cyclic codes can be extended to negacyclic codes. This will be used to derive some expressions for the average distortion of some negacyclic codes used for source coding. This is followed by the description of a joint source and channel coding scheme making use of negacyclic codes.

CHAPTER IV

BLOCK SOURCE CODING WITH NEGACYCLIC CODES

1. Introduction

As seen in previous chapters, source encoding is used to reduce redundancy in source outputs. Channel coding, on the other hand, concerns itself with adding structured redundancy in such a way as to make error detection and/or correction possible at the receiver, in order to combat channel noise.

The rate-distortion function R(D), introduced in Chapter I, furnishes, for a given fidelity criterion, a lower bound for the rate of a code for a given distortion D. Plotting the intersection of D and rate for a given code on the same graph as the rate-distortion function of the source under consideration yields both a qualitative and quantitative description of the effectiveness of a source code. Hence, relative and absolute comparisons of the source encoding effectiveness of block codes requires knowledge of D, the average distortion per digit for the code being considered.

First, however, one type of source encoding scheme covered in Chapter II will be extended to negacyclic codes. The method is a fixed-length to fixed-length block source coding scheme using negacyclic codes, to code a source

output over GF(p), or, equivalently, over $S = \{0, \pm 1, \pm 2, \ldots, \pm \frac{p-1}{2}\}$. The scheme is implemented as follows.

The source output is partitioned into blocks of n digits. For each block, the code word nearest to it in the sense of Lee distance is found. Then the n-digit block is replaced with the k information digits relevant to this code word.

The resulting compression ratio CR resulting from this source coding scheme is then given by $\frac{n}{k}$.

The average distortion per digit [48] D for group codes is given by

$$D = \frac{\sum_{\omega=1}^{n} K_\omega \omega}{p^{n-k} n}$$

where K_ω is the number of cosets of the code with leaders of Lee weight ω. p^{n-k} is the total number of cosets.

At this point a result which may be of some interest is derived. It regards the number of n-tuples, over S, having any Lee weight, where $S = \{0, \pm 1, \pm 2, \ldots, \pm \frac{p-1}{2}\}$.

Let N_t represent the number of n-tuples with Lee weight $t = \frac{p-1}{2}$. Then it is readily observed that $N_0 = 1$ and that $N_1 = 2n$, since there can be a +1 or -1 in any one of the n positions. As regards t = 2, then $N_2 = 2n + \binom{n}{2} 2^2$, where 2n is the number of single errors of Lee weight 2 and $\binom{n}{2} 2^2$ is the number of 2-error patterns, each error digit in the n-tuple having a Lee weight of unity. From the

preceding discussion, it is clear that the number N_t depends on the number of ways t can be expressed as the sum of j nonzero numbers drawn from S. The weight t can be expressed as the sum of j numbers drawn from the above set in $\binom{t-1}{j-1}$ ways if the signs are not taken into account.

Since the sign of any of the j digits can be positive or negative, a 2^j factor must be added to this expression. On the other hand, the j integers can occur in an n-tuple in $\binom{n}{j}$ ways, so that this factor must also be included. This discussion leads to a theorem.

Theorem 5

The number N_t of n-tuples, over S, which have Lee weight t ($t > 0$), is given by

$$N_t = 2\binom{n}{1}\binom{t-1}{0} + 2^2\binom{n}{2}\binom{t-1}{1} + 2^3\binom{n}{3}\binom{t-1}{2} + \cdots + 2^t\binom{n}{t}\binom{t-1}{t-1}$$

This result is illustrated in the following example.

Example 8

For t = 1. the expression obtained from Theorem 5 yields $N_t = 2\binom{n}{1}\binom{t-1}{0} = 2n$.

Similarly, for t = 2,

$$N_t = 2\binom{n}{1}\binom{1}{0} + 2^2\binom{n}{2}\binom{1}{1} = 2n + 2^2\binom{n}{2}.$$

Again, for t = 3,

$$N_t = 2\binom{n}{1}\binom{2}{0} + 2^2\binom{n}{2}\binom{2}{1} + 2^3\binom{n}{3}\binom{2}{2}$$

$$= 2n + 8\binom{n}{2} + 8\binom{n}{3}$$

and so on.

In the next section, some results on binary cyclic codes will be extended to negacyclic codes.

From this point on, it should be noted that unless otherwise specified, Lee weight is meant whenever weight is mentioned.

2. Extension of Binary Cyclic Code Theorems to Negacyclic Codes

Since the maximal-length cyclic code is a constant-weight code [49], a parallel result in the negacyclic case would be expected. As is known [50], this is indeed the case. The derivation is included here for the sake of completeness.

By a maximal-length negacyclic code is meant the $(n = \frac{p^r - 1}{2}, k = r)$ code generated by $g(X) = \frac{X^n + 1}{h(X)}$, where $h(X)$ is an irreducible polynomial over GF(p), of degree r and exponent 2n, i.e., $h(X)$ divides $X^{2n} - 1$, but not $X^{\nu} - 1$ for any $\nu < 2n$.

The code has $p^k - 1 = p^r - 1 = 2n$ nonzero words. Since $r = k$, this means, from Theorem 3, that the code has one negacyclic class. Since all the elements of a negacyclic class are negacyclic shifts of each other, they must necessarily have the same weight. Hence, this code is a constant weight code.

This weight must now be established. If $B(X)$ represents a 2n-digit maximal-length sequence, then $B(X)$ can be expressed as follows:

$$B(X) = \frac{X^{2n} - 1}{h(X)} \; C(X)$$

where $C(X)$ has degree $k - 1 = r - 1$ or less. From this it follows that

$$B(X) = g(X)(X^n - 1)\; C(X) \qquad (2)$$

since $$h(X)g(X) = X^n + 1$$

Expanding equation (2) yields

$$B(X) = -g(X)\; C(X) + X^n g(X)\; C(X)$$

where it is noted that $g(X)\; C(X)$ has degree $(n - 1)$ or less, and that $g(X)\; C(X)$, which belongs to the code, has one-half the weight of $B(X)$. Since $B(X)$ is a maximal-length sequence, it contains every nonzero element of GF(p) $\frac{p^r}{p} = p^{r-1}$ times. This implies that the weight of every nonzero word of code V is given by

$$p^{r-1}[1 + 2 + 3 + \ldots + \tfrac{p-1}{2}]$$

or, equivalently,

$$p^{r-1}\,\frac{(\frac{p-1}{2})(\frac{p+1}{2})}{2} = \frac{p^{r-1}(p^2-1)}{8}$$

This result is restated in the following theorem.

Theorem 6

> The maximal-length negacyclic code over $S = \{0, \pm1, \pm2, \ldots, \pm\frac{p-1}{2}\}$ has a constant Lee weight of $\frac{p^{r-1}(p^2-1)}{8}$.

From this theorem, the following corollary is directly obtained.

Corollary 1

> The maximal-length negacyclic code over S corrects any single error pattern of Lee weight
>
> $$\left\lfloor \left(\frac{\frac{p^{r-1}(p^2-1)}{8} - 1}{2} \right) \right\rfloor$$
>
> or less, where $\lfloor x \rfloor$ represents the largest integer contained in the positive number x.

The corollary demonstrates that the condition $t \leqslant \frac{p-1}{2}$ in Theorem 4 is a sufficient rather than a necessary condition. This condition is required, however, in order to make use of Berlekamp's decoding algorithm.

It is known [51] that the binary Hamming code, of length $2^m - 1$, has $\frac{(2^m - 1)(2^m - 2)}{6}$ words of weight 3. A parallel result for the negacyclic code of length $n = \frac{p^r - 1}{2}$ for $p = 3$ is proven next.

The known theorem which follows is a convenient starting point [52].

Theorem 7

The negacyclic code V with $n = \frac{p^r - 1}{2}$ and $t = 1$ is perfect.

It is noted that V has $k = n - r$. Since $p = 3$, every word $V_3(X)$ of V with Lee weight 3 can be expressed in the form

$$V_3(X) = a_u X^u + a_v X^v + a_w X^w$$

where a_u, a_v and a_w are chosen from the set $\{-1, 1\}$. This implies, in view of the definition of a negacyclic class, the following lemma.

Lemma 1

Every negacyclic class with words of Lee weight 3 in V has exactly 3 words of the form $1 + a_v X^v + a_w X^w$.

From Theorem 6 it is clear that, using V as one coset, the set of 3^r n-tuples can be completely partitioned into 1 coset with the leader of zero weight and 2n cosets

with leaders of Lee weight 1. Since V has minimum Lee distance of d = 3, all of the $2^2\binom{n}{2} = 2n(n-1)$ n-tuples of Lee weight 2 (see Theorem 5) are contained in the cosets with leaders of Lee weight 1. This means, since V is negacyclic and all of the n-tuples of Lee weight 1 have been used as coset leaders, that the number of n-tuples of Lee weight 2 contained in each coset with leader of Lee weight 1 is $\frac{2n(n-1)}{2n} = n - 1$. This result is repeated in the following lemma.

Lemma 2

Every coset of V with a leader of Lee weight 1 contains n - 1 n-tuples of Lee weight 2.

Next, the coset with leader X^0 can be considered, without loss of generality. This leader can produce an n-tuple of Lee weight 2 only with a code word of the form $1 + a_v X^v + a_w X^w$. This means, in view of Lemma 2, that V contains n - 1 words of the form $1 + a_v X^v + a_w X^w$. Applying Lemma 1, the following lemma is obtained.

Lemma 3

V has $\frac{n-1}{3}$ negacyclic classes containing words of Lee weight 3.

This lemma yields, in view of Theorem 3, the following theorem.

Theorem 8

The negacyclic code with $n = \frac{3^r - 1}{2}$ and $t = 1$ contains $\frac{2n(n-1)}{3}$ words of Lee weight 3.

From Theorem 7, it also follows that the code has 1 coset with the leader of weight 0 and 2n cosets with leaders of weight 1, so that the average distortion D per digit, since this is a block code, is given by

$$D = \frac{(0)(1) + (1)(2n)}{2^{n-k_n}} = \frac{2n}{p^r n} = \frac{2n}{(2n+1)n} = \frac{1}{n + \frac{1}{2}}$$

This result is restated in the following corollary.

Corollary 2

The negacyclic code with $n = \frac{p^r - 1}{2}$ and $t = 1$ has a compression ratio of $\frac{n}{n-r} = \frac{p^r - 1}{p^r - 1 - 2r}$ and an average distortion per digit of $\frac{1}{n + \frac{1}{2}}$.

This is illustrated in the following example.

Example 9

Let V be a (4,2) $t = 1$ negacyclic code over $S = \{0, \pm 1\}$. Then the CR obtained for this code would be $\frac{n}{n-r} = \frac{4}{4-2} = 2$, and the average distortion per digit D would be $\frac{1}{n + \frac{1}{2}} = \frac{1}{4.5} \simeq 0.222$.

Similarly, if V is a (13,10), t = 1 negacyclic code over S = {0, ±1}, then the CR = $\frac{n}{n - r}$ = 1.3, and $D = \frac{1}{n + \frac{1}{2}} \simeq .0741$.

It is known [53] that the binary shortened-by-1-digit Hamming code of length $2^m - 2$ is quasiperfect. Now parallel results for the case of negacyclic codes over GF(3) will be proven.

Let V be the ($n = \frac{3^r - 1}{2}$, $k = n - r$) negacyclic code with t = 1. Let V' be the (n - 1, k - 1) shortened-by-1-digit code obtained from V.

Since p = 3, all of the words of Lee weight 3 in V are of the form $a_u X^u + a_v X^v + a_w X^w$, where a_u, a_v and a_w, as before, are taken from the set {-1, 1}. This implies that every negacyclic class containing words of Lee weight 3 in V has 6 words ending with a nonzero digit. Since, from Lemma 3, there are $\frac{n - 1}{3}$ such negacyclic classes, V contains in all $6(\frac{n - 1}{3}) = 2(n - 1)$ words of Lee weight 3 ending in a nonzero digit. From Theorem 8, this means that V' contains $\frac{2n(n - 1)}{3} - 2(n - 1) = \frac{2(n - 1)(n - 3)}{3}$ words of Lee weight 3.

This result is repeated in the following lemma.

Lemma 4

If V' is the (n - 1, k - 1) shortened-by-1-digit code obtained from the ($n = \frac{3^r - 1}{2}$, $k = n - r$) t = 1 negacyclic code, then V'

contains $\frac{2(n-1)(n-3)}{3}$ words of Lee weight 3.

Now the cosets of V' can be considered. Since V' has $t = 1$, then $1 + 2(n-1) = 2n - 1$ cosets can be readily accounted for, namely, 1 coset with leader of weight zero, and $2(n-1)$ cosets with leaders of weight 1. This means that 2 cosets are unaccounted for. From Lemma 4, it is known that, in all the words of weight 3 contained in V', there are $\frac{2(n-1)(n-3)}{3}(3) = 2(n-1)(n-3)$ nonzero digits. This means, since all the $(n-1)$-tuples of Lee weight 1 are used as coset leaders, that there are in all $2(n-1)(n-3)$ $(n-1)$-tuples of Lee weight 2 in the cosets of V' with leaders of weight 1. Since there are in all $2^2\binom{n-1}{2} = 2(n-1)(n-2)$ $(n-1)$-tuples of Lee weight 2, the following lemma can be concluded.

Lemma 5

If V' is the $(n-1, k-1)$ shortened-by-1-digit code obtained from the $(n = \frac{3^r - 1}{2}, k = n - r)$ $t = 1$ negacyclic code, then V' has $2(n-1)$ $(n-1)$-tuples of Lee weight 2 available to construct cosets of V' with leaders of weight 2.

Now an $(n-1)$-tuple v_2 of Lee weight 2 can only produce an $(n-1)$-tuple of Lee weight 2 with words of V'

that have Lee weight 3 or 4.

Without loss of generality, v_2 = 1 1 0 0 . . . 0 can be considered. Any (n - 1)-tuple of Lee weight 3 which, with v_2, can produce an (n - 1)-tuple of Lee weight 2, has to be of the form 1 -1 x x x . . . x or -1 1 x x x . . . x, where only one x is nonzero. Since the mutual Lee distance between these two possibilities has to be at least 3, since t = 1, a little reflection reveals that the number of words in V' which can produce, with v_2, an (n - 1)-tuple of Lee weight 2, is 2. This result is restated in the following lemma.

Lemma 6

> If V' is the (n - 1, k - 1) shortened-by-1-digit code obtained from the ($n = \frac{3^r - 1}{2}$, k = n - r) t = 1 negacyclic code, then in a coset of V' with a leader of Lee weight 2, the number of (n - 1)-tuples of Lee weight 2, due to codewords of Lee weight 3, is 2.

There now remains the investigation of the number of (n - 1)-tuples, of Lee weight 2, that can exist in a coset of V' with leader of Lee weight 2, because of code words of Lee weight 4. This number is the same as the number γ_ν of possible ν-tuples such that all have Lee weight 4, and have the form, say, of 1 1 x x . . . x with only 2 nonzero x's, the mutual Lee distance being at least 3.

For $\nu = 3$, γ_3 is clearly zero. For $\nu = 4$, $\gamma_4 = 1$, since

1 1 1 1, 1 1 -1 1, 1 1 1 -1 or 1 1 -1 -1 can be chosen. Suppose, without loss of generality, that

1 1 1 1 is chosen.

For $\nu = 5$, $\gamma_5 = 3$ since, without loss of generality,

1 1 1 1 0, 1 1 -1 0 1, 1 1 0 -1 -1

or

1 1 1 1 0, 1 1 -1 0 -1, 1 1 0 -1 1

can be chosen.

Suppose the former three words, again, without loss of generality, are chosen.

For $\nu = 6$, $\gamma_6 = 3$, since one more word cannot be added.

For $\nu = 7$, $\gamma_7 = 4$, since

1 1 1 1 0 0 0, 1 1 -1 0 1 0 0,

1 1 0 -1 -1 0 0, 1 1 0 0 0 1 1

or

1 1 1 1 0 0 0, 1 1 -1 0 1 0 0,

1 1 0 -1 -1 0 0, 1 1 0 0 0 1 -1

or

1 1 1 1 0 0 0, 1 1 -1 0 1 0 0,

1 1 0 -1 -1 0 0, 1 1 0 0 0 -1 1

or

1 1 1 1 0 0 0, 1 1 -1 0 1 0 0,

1 1 0 -1 -1 0 0, 1 1 0 0 0 -1 -1

can be used.

Continuing this line of reasoning yields the following lemma.

Lemma 7

> The number γ_ν of ν-tuples over GF(3) of Lee weight 4, having a mutual distance of at least 3, and matching in two positions, is given by $\gamma_\nu = 3(\alpha - 1)$, for $\nu = 3\alpha$ and $\nu = 3\alpha - 1$ and is given by $\gamma_\nu = 3(\alpha - 1) - 2$ for $\nu = 3\alpha - 2$, where α is a positive integer.

From Lemma 7, the following lemma can be directly obtained.

Lemma 8

> If V' is the (n - 1, k - 1) shortened-by-1-digit code described in Lemma 4, the number of (n - 1)-tuples of Lee weight 2 that can exist in a coset of V' with leader of Lee weight 2, because of words of Lee weight 4 in V', is at most n - 4.

From Lemmas 6 and 8, it is clear that a leader of Lee weight 2 can produce at most n - 4 + 2 = n - 2 (n - 1)-tuples of Lee weight 2 in a coset of V'. This means, if the leader itself is included, that there can be

at most n - 1 (n - 1)-tuples of Lee weight 2 in a coset of V' with leader of weight 2. In view of Lemma 5, this then implies that there are 2 cosets of V' with leaders of Lee weight 2.

To summarize the discussion up to this point, all of the $p^r = 3^r$ cosets of V' have been accounted for. In addition to V' itself, there are 2(n - 1) cosets with leaders of Lee weight 1 and 2 cosets with leaders of Lee weight 2. Since V' has t = 1, a further conclusion can be drawn.

Theorem 9

> The (n - 1, k - 1) code V' obtained by shortening the ($n = \frac{3^r - 1}{2}$, k = n - r) negacyclic code with t = 1 is quasiperfect.

Since the code of Theorem 9 has 1 coset with the leader of weight 0, 2(n - 1) cosets with leaders of Lee weight 1 and 2 cosets with leaders of Lee weight 2, it follows that the code gives an average distortion per digit of

$$D = \frac{(0)(1) + (1)(2)(n-1) + (2)(2)}{2^{n-1-k+1}(n-1)} = \frac{2n - 2 + 4}{2^{n-k}(n-1)}$$

$$= \frac{2n - 2 + 4}{3^r(n-1)} = \frac{2n + 2}{(2n+1)(n-1)} = \frac{n+1}{(n+\frac{1}{2})(n-1)}$$

This result is restated in the following corollary.

Corollary 3

The $(n - 1, k - 1)$ code obtained by shortening the $(n = \frac{3^r - 1}{2}, k = n - r)$ negacyclic code with $t = 1$ has a compression ratio of $\frac{n - 1}{k - 1}$ and an average distortion per digit of $\frac{n + 1}{(n + \frac{1}{2})(n - 1)}$.

These results are illustrated in the following example.

Example 10

Let V be the (3,1) code obtained by shortening the (4,2) $t = 1$ negacyclic code over $S = \{0, \pm 1\}$. Then the CR for V is given by $\frac{n - 1}{k - 1} = \frac{3}{1} = 3$, and the average distortion per digit D is given by $\frac{(n + 1)}{(n + \frac{1}{2})(n - 1)} \simeq 0.296$.

Similarly, if V is the (12,9) code obtained by shortening the (13,10) $t = 1$ negacyclic code over $S = \{0, \pm 1\}$, then the CR $= \frac{12}{9} = 1.333$ and D is given by $\frac{(n + 1)}{(n + \frac{1}{2})(n - 1)} \simeq .0864$.

It is known that the code obtained by interleaving two binary Hamming codes is quasiperfect [54]. A similar result for negacyclic codes will now be established.

Let V_1 be the $(n_1 = \frac{p^{r_1} - 1}{2}, k_1 = n_1 - r_1)$ primitive negacyclic code with $t = 1$ and V_2 the $(n_2 = \frac{p^{r_2} - 1}{2}, k_2 = n_2 - r_2)$ primitive negacyclic code with $t = 1$. Let V be the $(n_1 + n_2, k_1 + k_2)$ code obtained by interleaving

V_1 and V_2.

Since V can correct any pattern which, on deinterleaving, becomes 2 error patterns correctable by V_1 and V_2, respectively, the number of error patterns correctable by V is given by

$$1 + 2(n_1 + n_2) + 2n_1 2n_2 = 1 + 2\left(\frac{p^{r_1} - 1}{2} + \frac{p^{r_2} - 1}{2}\right) + 4\left(\frac{p^{r_1} - 1}{2}\right)\left(\frac{p^{r_2} - 1}{2}\right) = p^{r_1 + r_2}$$

since $2(n_1 + n_2)$ is the number of single errors of Lee weight 1 and $2n_1 2n_2$ the number of two-error patterns of Lee weight 2.

Since $p^{r_1 + r_2}$ is equal to the number of syndromes of V, it can be concluded that coset leaders of Lee weight > 2 are not required, and since V has t = 1, the following conclusion is arrived at:

Theorem 10

> The code obtained by interleaving two primitive negacyclic codes, each having t = 1, is quasiperfect.

Hence, an expression for the average distortion per digit D for the interleaved primitive negacyclic codes of Theorem 10 can be obtained as follows:

$$D = \frac{(0)(1) + (1)2(n_1 + n_2) + (2)(2n_1 2n_2)}{2^{n_1 + n_2 - k_1 - k_2}(n_1 + n_2)}$$

$$= \frac{2(n_1 + n_2) + 8n_1n_2}{p^{r_1 + r_2}(n_1 + n_2)} = \frac{2(n_1 + n_2) + 8n_1n_2}{(2n_1 + 1)(2n_2 + 1)(n_1+n_2)}$$

$$= \left(\frac{n_1}{n_1 + n_2}\right) D_1 + \left(\frac{n_2}{n_1 + n_2}\right) D_2$$

where $D_1 = \frac{1}{n_1 + \frac{1}{2}}$ and $D_2 = \frac{1}{n_2 + \frac{1}{2}}$, since D_1 and D_2 are, from Corollary 2, the average distortions per digit of the component codes.

This result is restated in the following corollary.

Corollary 4

The code obtained by interleaving two primitive negacyclic codes with respective lengths $n_1 = \frac{p^{r_1} - 1}{2}$ and $n_2 = \frac{p^{r_2} - 1}{2}$ and with $t = 1$, has a compression ratio given by $\frac{n_1 + n_2}{n_1 - r_1 + n_2 - r_2}$ and an average distortion per digit of $\left(\frac{n_1}{n_1 + n_2}\right) D_1 + \left(\frac{n_2}{n_1 + n_2}\right) D_2$, where D_1 and D_2 are the average distortions per digit for the respective component codes.

Example 11

Let V be the (17,12) code obtained by interleaving the $V_1 = (4,2)$ $t = 1$ and $V_2 = (13,10)$ $t = 1$ primitive

negacyclic codes. Then the compression ratio of V is given by

$$CR = \frac{n_1 + n_2}{n_1 - r_1 + n_2 - r_2} = \frac{4 + 13}{(4 - 2) + (13 - 10)}$$

$$= 3.4$$

The average distortion per digit of V_1 is given by $D_1 = \frac{1}{n_1 + \frac{1}{2}} \simeq 0.222$, while $D_2 = \frac{1}{n_2 + \frac{1}{2}} \simeq .0741$ so that the average distortion per digit of V is given by

$$D = \left(\frac{n_1}{n_1 + n_2}\right) D_1 + \left(\frac{n_2}{n_1 + n_2}\right) D_2 = .0522 + .0567 = .1089.$$

Finally, the code obtained by interleaving two codes of Theorem 9 can be considered. Let V_1 be the $(n_1 - 1, k_1 - 1)$ code obtained by shortening the $(n_1 = \frac{3^{r1} - 1}{2}, k_1 = n_1 - r_1)$ negacyclic code with t = 1 and V_2 the $(n_2 - 1, k_2 - 1)$ code obtained by shortening the $(n_2 = \frac{3^{r2} - 1}{2}, k_2 = n_2 - r_2)$ negacyclic code with t = 1. Let V be the $(n_1 + n_2 - 2, k_1 + k_2 - 2)$ code obtained by interleaving V_1 and V_2.

Let K_{ij} be the number of cosets of V_i with coset leaders of Lee weight j, i = 1, 2. It is clear from Theorem 9 that $K_{i0}=1$, $K_{i1} = 2(n_i - 1)$, $K_{i2} = 2$. Since V can correct any error pattern which, on deinterleaving, reduces to coset leaders of V_1 and V_2, it follows that $K_0 = 1$, $K_1 = 2(n_1 - 1 + n_2 - 1)$,

$K_2 = 2(n_1 - 1)\ 2(n_2 - 1) + 2 + 2$, $K_3 = 2(n_2 - 1) + 2(n_1 - 1)$, $K_4 = 2(2)$, where K_ω is the number of cosets of V with leaders of Lee weight ω.

Therefore, V produces an average distortion per digit given by,

$$D = \frac{(0)(1) + (1)2(n_1-1 + n_2-1) + (2)[4 + 4(n_1-1)(n_2-1)] + (3)[2(n_2-1) + 2(n_1-1)] + (4)(4)}{3^{r_1+r_2}\,(n_1-1 + n_2-1)}$$

$$= \frac{2(n_1+n_2-2) + 8[1 + (n_1-1)(n_2-1)] + 6(n_2-1 + n_1-1) + 16}{(2n_1+1)(2n_2+1)(n_1 + n_2 - 2)}$$

$$= \frac{2(n_1+n_2-2) + 8(-n_1-n_2+n_1n_2+2) + 6(n_1+n_2-2) + 16}{(2n_1+1)(2n_2+1)(n_1 + n_2 - 2)}$$

$$= \frac{(n_1+n_2-2)(2 - 8 + 6) + 8n_1n_2 + 16}{(2n_1+1)(2n_2+1)(n_1 + n_2 - 2)}$$

$$= \frac{8n_1n_2 + 16}{(2n_1+1)(2n_2+1)(n_1 + n_2 - 2)}$$

This result is restated in the following corollary.

Corollary 5

If V_1 is the code obtained by shortening the $(n_1 = \frac{p^{r_1} - 1}{2},\ k_1 = n_1 - r_1)$ negacyclic code with $t = 1$ by 1 digit and V_2 is the code obtained by shortening the $(n_2 = \frac{p^{r_2} - 1}{2},\ k_2 = n_2 - r_2)$ negacyclic code with $t = 1$ by

1 digit, then the code produced by interleaving V_1 and V_2 will have a compression ratio given by $\frac{n_1 - r_1 + n_2 - r_2}{n_1 + n_2 - 2}$ and an average distortion per digit given by $D = \frac{8n_1n_2 + 16}{(2n_1+1)(2n_2+1)(n_1+n_2-2)}$.

Example 12

Let V_1 be the (3,1) code obtained by shortening the (4,2) $t = 1$ negacyclic code over $\{0, \pm 1\}$ and let V_2 be the (12,9) code obtained by shortening the (13,10) $t = 1$ negacyclic code over $S = \{0, \pm 1\}$. Further let V be the $(n_1 + n_2, n_1 - r_1 + n_2 - r_2)$ code produced by interleaving V_1 and V_2. Then the compression ratio of V will be given by

$$CR = \frac{n_1 - r_1 + n_2 - r_2}{n_1 + n_2 - 2} \simeq 0.8$$

so that there is expansion instead of compression. The average distortion per digit of V will be given by

$$D = \frac{8n_1n_2 + 16}{(2n_1 + 1)(2n_2 + 1)(n_1 + n_2 - 2)} \simeq .1185$$

In the last section, it was shown that negacyclic codes have many properties in common with cyclic codes, and some expressions for the average distortion per digit were obtained. In the next section, a joint source and channel encoding scheme will be presented.

3. A Joint Source and Channel Encoding Scheme

a. Introduction

Hellman [22] has suggested jointly encoding both the source and the channel using a convolutional encoder and a sequential decoder. His goal was, principally, to develop a distortionless coding method which would increase source entropy and provide channel encoding using a simple encoder, even if it resulted in greatly increased decoder complexity. As discussed in Chapter I, this is unusual. The source encoder is inherently the more complex element of the source encoder-decoder pair. Hellman's systen makes use of a binary error-propagating (convolutional) code and is both distortionless and optimal for a discrete memoryless source and channel, for a rate less than the channel capacity. His scheme can be extended to sources with memory, at the cost of increased complexity, and does not require knowledge of the source statistics at the encoder.

In this section, another type of joint source and channel encoding is presented, based on a variation of syndrome encoding. Using negacyclic codes, it results in data compression as well as an error-detection capability. The main feature of the technique is that, while reducing the redundancy under appropriate conditions, it also has a built-in error-detecting capability with respect to the channel. Hence it actually is equivalent to a source and channel coder. If the average length $\overline{L}$ after coding is

allowed to be greater than n, then the technique can be used purely as an error-detecting scheme.

The discussion is begun by introducing the basic distortionless source encoding scheme.

b. Distortionless Encoding Scheme

Let V be a negacyclic code over $S=\{0, \pm 1, \ldots, \pm\frac{p-1}{2}\}$ with block length n and error-correcting capability t. Its syndromes will then have length rt.

With reference to V, the source output is partitioned into blocks of n digits. Let B be such a block with weight ω. Then the maximum possible value of ω is nt.

If $\omega = 0$, then B can be replaced with the (rt)-tuple of all zeros, the syndrome for the case of no error.

If $\omega \neq 0$, then ω can be expressed as

$$\omega = mt + \rho, \quad 0 < \rho \leq t,$$

where m and ρ are integers. Clearly, the maximum value m can assume is (n - 1).

Now B can be treated as the sum of (m + 1) n-tuples $B_1, B_2, \ldots, B_{m+1}$ such that the (m + 1)-by-n matrix B, in which the rows are $B_1, B_2, \ldots, B_{m+1}$, satisfies the following properties:

(i) Every row of B has a weight of t except the (m + 1) st row which may have a Lee weight < t, but greater than zero.

(ii) Every column of B has at most 2 nonzero entries.

(iii) The Lee weight of every column of B is t or less.

(iv) If a column has two nonzero entries, then they are adjacent and are of the same sign.

There is another property of B, stemming directly from the first four, which will be mentioned here, as it will be important in the error-detection phase:

(v) All elements to the right of the jth element in the (i - 1) st row ($i \geq 2$) must be zero if the jth element in the ith row of B is nonzero.

Example 13

The case where n = 12 and S = {0, ±1, ±2}, is considered so that p = 5, t = 2 and r = 2. Suppose B = 0 1 -2 2 1 1 0 0 1 -2 1 1, which has a weight of 12. Then $\omega = mt + \rho = 12$ so that m = 5 and $\rho = 2$. This results in

$$B = \begin{bmatrix} 0 & 1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -2 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \end{bmatrix}$$

Henceforth, for purposes of comparison and unless otherwise noted, all examples and graphs in the remainder of this chapter will make use of the aforementioned negacyclic code and, in most cases, the same output sequence B.

Returning to the discussion, the matrix B could be transmitted row by row. This would be extremely wasteful, however, since, by construction, each row of the matrix is restricted to a maximum Lee weight of t, and so any row can be considered to be an error vector corresponding to a correctable error for a t-error-correcting negacyclic code. Then, since any correctable error vector can be represented by its syndrome, an rt-tuple, each row of B can be replaced by its corresponding syndrome, which can then be transmitted.

Thus, if σ_i represents the syndrome of B_i, the ith row of the matrix B, then the block B can be replaced with the encoded sequence $\sigma = \sigma_1, \sigma_2, \ldots, \sigma_{m+1}$, which, it is noted, is in general (m + 1)rt digits long. Hence, in Example 13 above, the sequence B of length 12 would be replaced by a sequence σ 24 digits in length.

The sequence σ of syndromes is then transmitted over the channel. At the receiver, σ' is received, where $\sigma' = \sigma_1', \sigma_2', \ldots, \sigma'_{m+1}$. Then, corresponding to each syndrome σ_i', the error pattern B_i' is obtained and the matrix B' in which the ith row is B_i' is formed.

If B' satisfies all of the properties satisfied by B, then the rows of B' are added to obtain B' which, it is decided, was the sequence put out by the source. If not,

either a retransmission request is transmitted back, or the block is abandoned. Figure 4 is a block diagram illustrating this source encoding system, while Figure 5 is a flow chart of the logic involved in detecting transmission errors in this system.

Let L be the length of the transmitted sequence σ, L a multiple of rt. Then since the scheme possesses an error-detection capability, data compression is assured if $L \leqslant n$. Otherwise L must be $> n$.

It should be pointed out at this point that this system cannot in general be optimal, since the system is constrained to transmitting sequences with lengths which are multiples of rt. Hence, even a unit increase in source sequence weight can increase the transmitted sequence length by rt digits.

The error detection occurs in two stages. In the first, the weight of the n-tuple corresponding to the received syndrome is determined. If it is less than t and the corresponding n-tuple is not the last row of the matrix B', then an error has been detected. If no error is detected, then B' is formed and is tested for the presence of errors. This is illustrated in Figure 5.

In the following sections, expressions for the average length of the encoded sequences for the distortionless case and those cases where distortion is allowed are derived. This provides the groundwork necessary for the calculation of the average distortion per digit for the

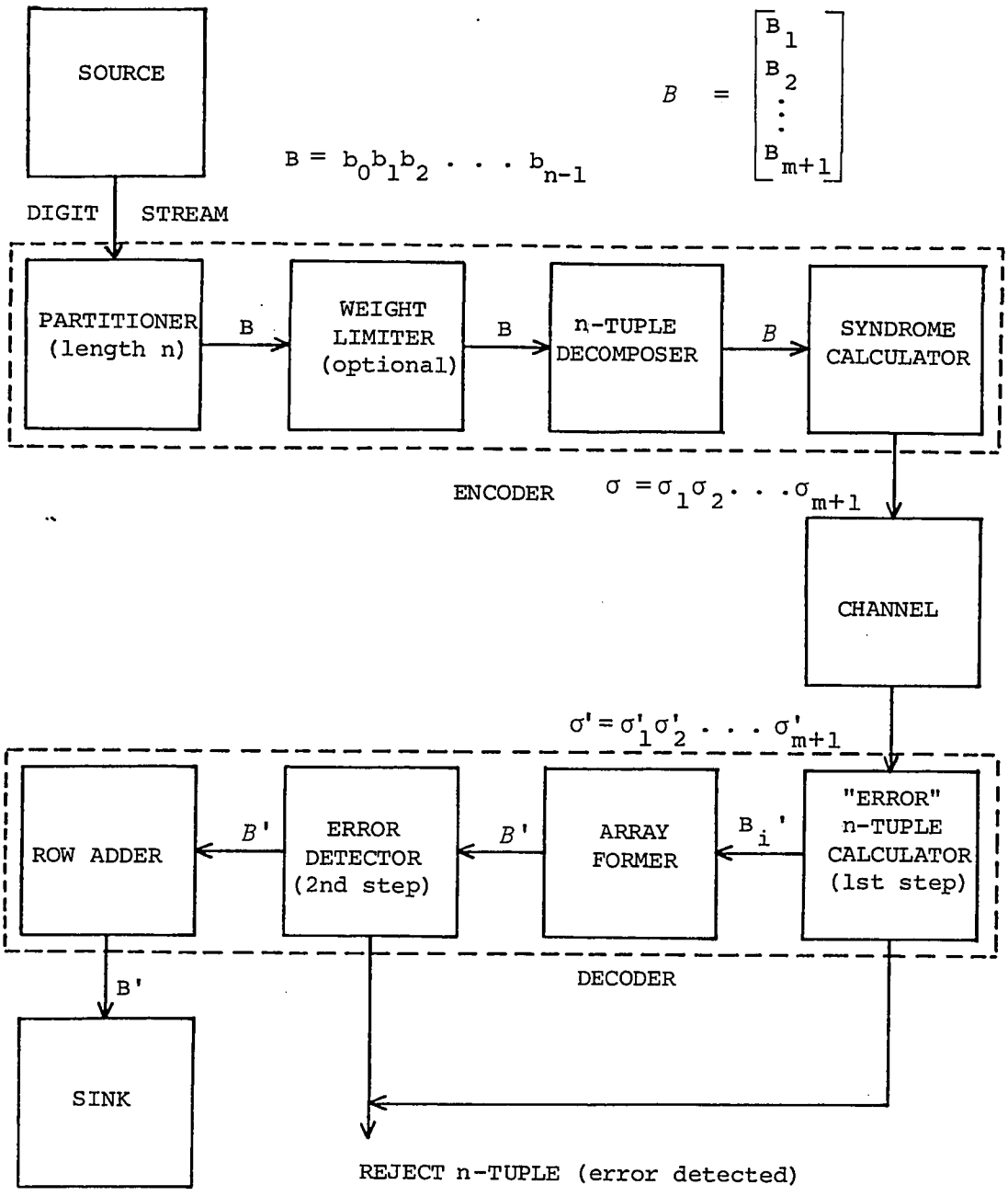

Figure 4:- Block diagram of joint source and channel encoding system.

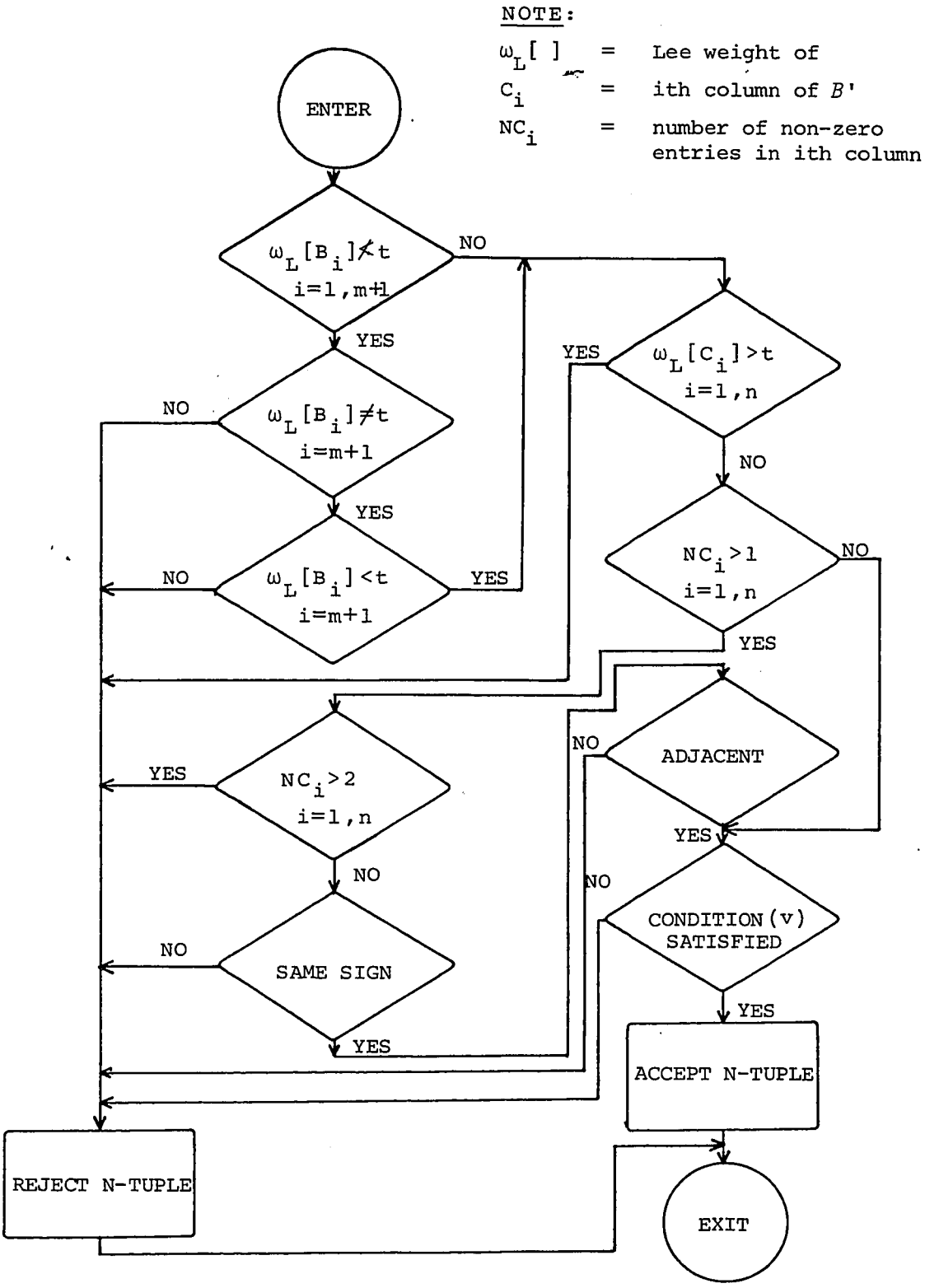

Figure 5:- Flowchart of second error-detection step for decoder of Figure 4.

cases where distortion is allowed.

(i) Average Length of the Encoded Sequence

Let P_ω represent the probability that the source n-tuple B has Lee weight ω. Since the encoded sequence of length L depends on the number of times t divides ω, a new set of probabilities $Q = Q_1, Q_2, \ldots, Q_n$ can be defined where

$$\left.\begin{aligned} Q_1 &= P_0 + P_1 + \ldots + P_t \\ Q_2 &= P_{t+1} + P_{t+2} + \ldots + P_{2t} \\ Q_3 &= P_{2t+1} + P_{2t+2} + \ldots + P_{3t} \\ &\vdots \\ Q_n &= P_{(n-1)t+1} + P_{(n-1)t+2} + \ldots P_{nt} \end{aligned}\right\} \quad (3)$$

It can be noted that $Q_1 = \text{Prob}\{0 < \omega < t\}$ and so includes the case of the all-zero source n-tuple, while $Q_j = \text{Prob}\{(j-1)t + 1 \leq \omega \leq jt\}$ for $j = 2, 3, 4, \ldots n$. Since every n-tuple of weight t is replaced by its syndrome, an rt-tuple, then the average length $\bar{L}$ of the encoded sequence σ is given by

$$\begin{aligned} \bar{L} &= Q_1 rt + Q_2\, 2rt + Q_2\, 2rt + Q_3\, 3rt + \ldots + Q_n\, nrt \\ &= rt\,(Q_1 + 2Q_2 + 3Q_3 + \ldots + nQ_n) \end{aligned}$$

Two observations can be made at this point. The first is that L, the length of the encoded sequence σ, must always be a multiple of rt, the syndrome length. However, $\bar{L}$, the average length of the encoded sequence σ, can be anything > rt, since under no circumstances can a source n-tuple be encoded as anything shorter than one rt-tuple. Hence in all expressions for $\bar{L}$, the minimum value must be rt. The second observation is that $\frac{\bar{L}}{n}$ is nothing but the average compression ratio CR.

The expression for $\bar{L}$ can now be normalized with respect to n. This yields

$$\frac{\bar{L}}{n} = \left(\frac{rt}{n}\right) \left(Q_1 + 2Q_2 + 3Q_3 + \ldots + nQ_n\right)$$

And so for a given distribution Q, $\frac{\bar{L}}{n}$ can be computed. If the distribution Q is appropriate, then $\frac{\bar{L}}{n}$ will be ≤ 1.

For example, if $Q_i = Q_1 \lambda^{i-1}$, with $0 < \lambda < 1$, then

$$\frac{\bar{L}}{n} = \left(\frac{rt}{n}\right) \left(Q_1 + 2Q_1\lambda + 3Q_1\lambda^2 + \ldots + nQ_1\lambda^{n-1}\right)$$

$$\frac{\bar{L}}{n} = \left(\frac{rt}{n}\right)Q_1\left(1 + 2\lambda + 3\lambda^2 + \ldots + n\lambda^{n-1}\right) \qquad (4)$$

Now, $1 + 2\lambda + 3\lambda^2 + \ldots + n\lambda^{n-1}$ can be decomposed into a sum of n geometric series, or portions thereof. This yields

$$1 + 2\lambda + 3\lambda^2 + \ldots + n\lambda^{n-1} = 1 + \lambda + \lambda^2 + \ldots + \lambda^{n-1}$$
$$+ \lambda + \lambda^2 + \ldots + \lambda^{n-1}$$
$$+ \lambda^2 + \ldots + \lambda^{n-1}$$
$$\vdots$$
$$+ \lambda^{n-1}$$

so that the first n terms of the decomposition sum to $\frac{1 - \lambda^n}{1 - \lambda}$, the next (n - 1) terms sum to $(\frac{1 - \lambda^n}{1 - \lambda}) - 1$, the next (n - 2) terms, to $\frac{1 - \lambda^n}{1 - \lambda} - (1 + \lambda)$, and so on, to the last term, which sums to $\frac{1 - \lambda^n}{1 - \lambda} - (1 + \lambda + \lambda^2 + \ldots + \lambda^{n-2})$.

Hence, $1 + 2\lambda + 3\lambda^2 + \ldots + n\lambda^{n-1}$

$$= \left[\frac{1 - \lambda^n}{1 - \lambda}\right] + \left[\frac{1 - \lambda^n}{1 - \lambda} - (\frac{1 - \lambda}{1 - \lambda})\right] + \left[\frac{1 - \lambda^n}{1 - \lambda} - (\frac{1 - \lambda^2}{1 - \lambda})\right]$$
$$+ \left[\frac{1 - \lambda^n}{1 - \lambda} - (\frac{1 - \lambda^3}{1 - \lambda})\right] + \ldots + \left[\frac{1 - \lambda^n}{1 - \lambda} - (\frac{1 - \lambda^{n-1}}{1 - \lambda})\right]$$

$$= n(\frac{1 - \lambda^n}{1 - \lambda}) - (\frac{1 - \lambda}{1 - \lambda} + \frac{1 - \lambda^2}{1 - \lambda} + \frac{1 - \lambda^3}{1 - \lambda} + \ldots + \frac{1 - \lambda^{n-1}}{1 - \lambda})$$

$$= \frac{n(1 - \lambda^n)}{1 - \lambda} - \frac{1}{1 - \lambda}\left[(n - 1) - (\lambda + \lambda^2 + \ldots + \lambda^{n-1})\right]$$

$$= \frac{n(1 - \lambda^n)}{1 - \lambda} - \frac{1}{1 - \lambda}\left[(n - 1) - (\frac{1 - \lambda^n}{1 - \lambda} - 1)\right]$$

$$= \frac{1}{1 - \lambda}\left[n(1 - \lambda^n) - (n - 1) + (\frac{1 - \lambda^n}{1 - \lambda}) - 1\right]$$

$$= \frac{1}{1 - \lambda}\left[\frac{1 - \lambda^n}{1 - \lambda} - n\lambda^n\right]$$

Equation (4) can thus be rewritten as:

$$\frac{\bar{L}}{n} = \left(\frac{rt}{n}\right)\left(\frac{Q_1}{1-\lambda}\right)\left(\frac{1-\lambda^n}{1-\lambda} - n\lambda^n\right)$$

Now, since $Q_i = Q_1\lambda^{i-1}$, then

$$Q_1 + Q_2 + Q_3 + \ldots + Q_n = Q_1[1 + \lambda + \lambda^2 + \ldots + \lambda^{n-1}]$$

$$= Q_1\left(\frac{1-\lambda^n}{1-\lambda}\right) = 1, \quad \text{and so}$$

$$Q_1 = \frac{1-\lambda}{1-\lambda^n}$$

then

$$\frac{\bar{L}}{n} = \left(\frac{rt}{n}\right)\left(\frac{1}{1-\lambda^n}\right)\left(\frac{1-\lambda^2}{1-\lambda} - n\lambda^n\right)$$

$$= \left(\frac{rt}{n}\right)\left(\frac{1}{1-\lambda} - \frac{n\lambda^n}{1-\lambda^n}\right)$$

and so

$$\frac{\bar{L}}{n} = \left(\frac{rt}{n}\right)\left(\frac{1}{1-\lambda} - \frac{n\lambda^n}{1-\lambda^n}\right) = \frac{rt}{n(1-\lambda)} - \frac{rt\lambda^n}{1-\lambda^n} \quad (5)$$

Since $0 < \lambda < 1$, then λ^n is usually $<< 1$, so that $\frac{\lambda^n rt}{1-\lambda^n}$ will be small. For example, for the 2-error-correcting negacyclic code of length 12, $n = 12$, $r = t = 2$, so that $\frac{\lambda^n rt}{1-\lambda^n} = 1.57$ for $\lambda = 0.9$, .295 for $\lambda = .8$, .0561 for $\lambda = .7$ and 9.77×10^{-4} for $\lambda = .5$.

Also, since $\frac{rt\lambda^n}{1-\lambda^n}$ is positive for $0 < \lambda < 1$, it serves to reduce $\frac{\bar{L}}{n}$. Hence the approximation

$$\frac{\bar{L}}{n} \simeq \frac{rt}{n(1-\lambda)}$$

is pessimistic for large λ, the approximation becoming

closer to equation (5) as λ decreases.

Hence, for small λ,

$$\frac{\overline{L}}{n} = \frac{rt}{n(1-\lambda)} \qquad (6)$$

The case where $\frac{\overline{L}}{n} < 1$ is of primary importance. For the distribution $Q_i = Q_1 \lambda^{i-1}$ above, this occurs for $\lambda \leqslant 1 - \frac{rt}{n}$. Also, as noted before, even if B has zero weight, corresponding to a source n-tuple consisting of only zeros, one rt-tuple must be transmitted, so that $\frac{\overline{L}}{n}$ cannot be less than $\frac{rt}{n}$.

Figure 6 is a graph of equations (5) and (6) for the (n = 12, k = 8) 2-error-correcting negacyclic code of Example 13, for the source probability distribution defined by equation (3).The lower curve corresponds to equation (5) and the upper, to equation (6) , the approximate expression for $\frac{\overline{L}}{n}$. For this code, r = t = 2, and hence, from the previous discussion, $\frac{\overline{L}}{n}$ should be unity or less for $\lambda \leqslant 0.666$. This is borne out in the figure. It is also clear that $\frac{\overline{L}}{n}$ is nowhere less than $\frac{rt}{n}$, which is 0.333 in this case. This is in agreement with the bound on $\frac{\overline{L}}{n}$ mentioned above.

Hence, for $\lambda \leqslant 0.666$, there has been some data compression, since every time *B'* does not satisfy all of those properties attributed to *B*, an error situation has been detected.

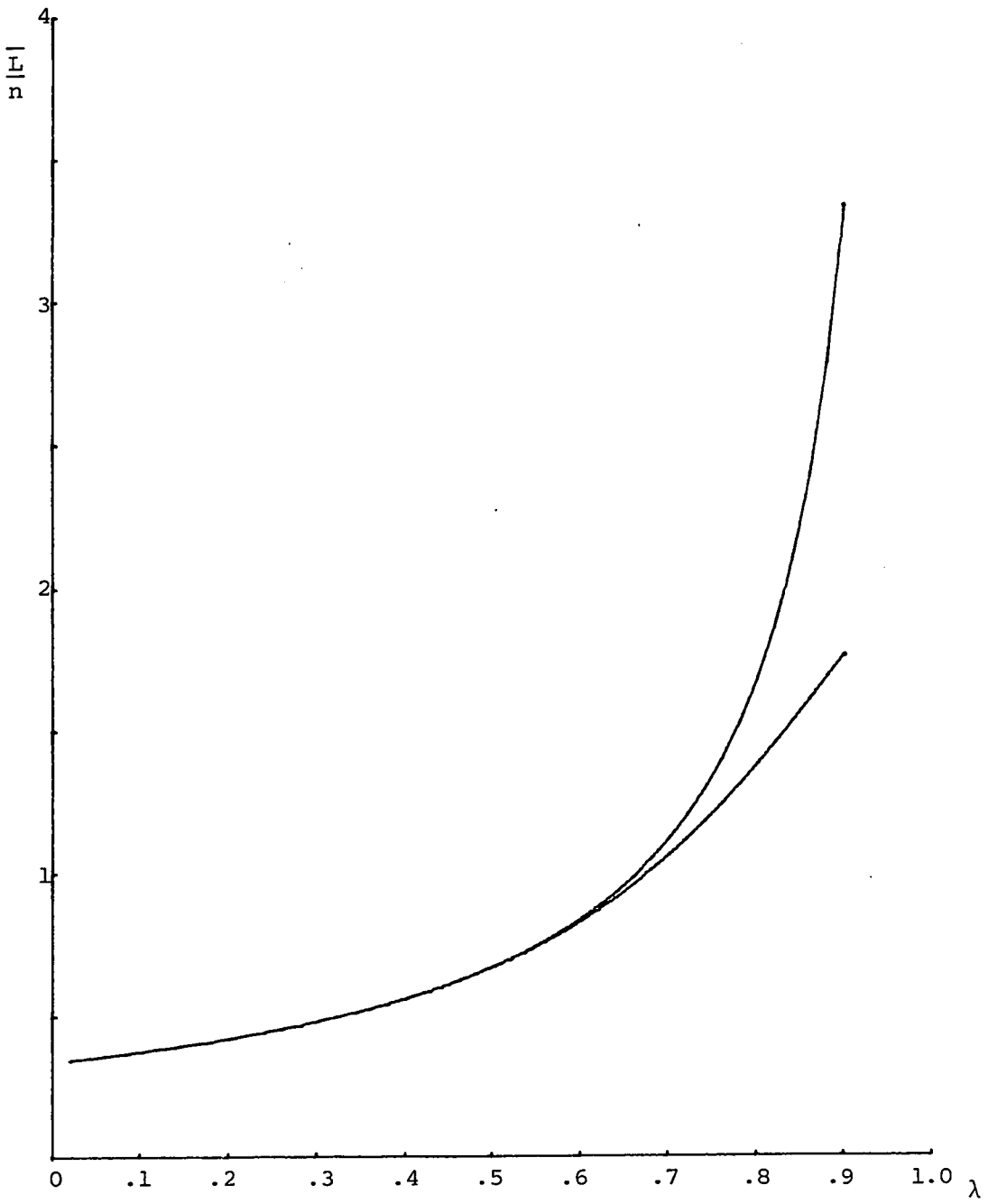

Figure 6:- Normalized average length of the encoded sequence σ for the distortionless case, with the distribution of Equation (3). The upper curve corresponds to the approximate expression.

This distortionless source coding scheme will then, for appropriate distributions Q, have both error-detection and data-compression capabilities. However, if the distribution Q is such that $\bar{L}$ works out to be greater than n, distortion can be introduced to lower the value of $\bar{L}$ down to n.

Next, an expression for $\bar{L}$ is derived, for a different source probability distribution.

The probability distribution Q defined above made use of P_ω, the probability that the source n-tuple has a weight ω. Since in general a given n-tuple weight can be achieved in many different ways, this is very different from the probability of occurrence of a single digit of a given weight in a source n-tuple. As will become clear in the next section, one type of distribution is more convenient for calculations involving certain types of weight-limiting schemes, while the other distribution is more convenient for the other.

Let Π_i be the probability that the source output is a +i or -i.

Clearly, then, the average weight $\bar{\omega}$ of a source n-tuple is given by

$$\bar{\omega} = (\Pi_0 n)(0) + (\Pi_1 n)(1) + (\Pi_2 n)(2) + \ldots + (\Pi_t n)(t)$$

$$= n(\Pi_1 + 2\Pi_2 + 3\Pi_3 + \ldots + t\Pi_t)$$

As in section 3b,

$$\bar{\omega} = mt + \rho, \qquad 0 < \rho < t$$

Then B will contain $(m + 1)$ rows, and so the average length $\bar{L}$ of σ is given by

$$\bar{L} = (m + 1)rt$$

$$= \left\langle \frac{\bar{\omega}}{t} \right\rangle rt$$

$$= \left\langle \frac{n(\Pi_1 + 2\Pi_2 + 3\Pi_3 + \ldots + t\Pi_t)}{t} \right\rangle rt \qquad (7)$$

And so, as in the previous derivation, the average length $\bar{L}$ of the encoded sequence σ is dependent only on the negacyclic code used and on the source probability distribution.

For example, if $\Pi_i = \Pi_0 \lambda^i$, then (7) becomes

$$\bar{L} = \left\langle \frac{n(\Pi_0\lambda + 2\Pi_0\lambda^2 + \ldots + t\Pi_0\lambda^t)}{t} \right\rangle rt$$

$$= \left\langle \frac{n\Pi_0\lambda(1 + 2\lambda + 3\lambda^2 + \ldots + t\lambda^{t-1})}{t} \right\rangle rt$$

$$= \left\langle \frac{n\Pi_0\lambda[(\frac{1}{1 - \lambda})\ (\frac{1 - \lambda^t}{1 - \lambda} - t\lambda^t)]}{t} \right\rangle rt$$

and since $\sum_{i=1}^{t} \Pi_i = 1$, $\Pi_0 = \frac{1 - \lambda}{1 - \lambda^{t+1}}$ and so

$$\bar{L} = \left\langle \frac{n\lambda\left(\frac{1-\lambda}{1-\lambda^{t+1}}\right)\left[\left(\frac{1}{1-\lambda}\right)\left(\frac{1-\lambda^{t}}{1-\lambda} - t\lambda^{t}\right)\right]}{t} \right\rangle_{rt}$$

$$\bar{L} = \left\langle \left(\frac{n}{t}\right)\left(\frac{\lambda}{1-\lambda^{t+1}}\right)\left(\frac{1-\lambda^{t}}{1-\lambda} - t\lambda^{t}\right) \right\rangle_{rt} \qquad (8)$$

As an example, the (12,8) 2-error-correcting negacyclic code is considered, with $\lambda = 0.5$. Then $\bar{L} = 16$ as against $n = 12$.

Figure 7 is a graph of equation (8), normalized by division by n, for the average length of an encoded sequence as a function of λ.

c. Encoding With Distortion

(i) Schemes for the Introduction of Distortion:

Distortion can be introduced in many ways, with different effects. Considering the expressions obtained for $\bar{L}$ in the previous section, the obvious techniques are to either reduce the weight of large-weight digits, or limit the total weight of a source n-tuple, or a combination of these two techniques. Each of these alternatives will be discussed next and expressions for the average length of the encoded sequence for the different types of distortion will be derived.

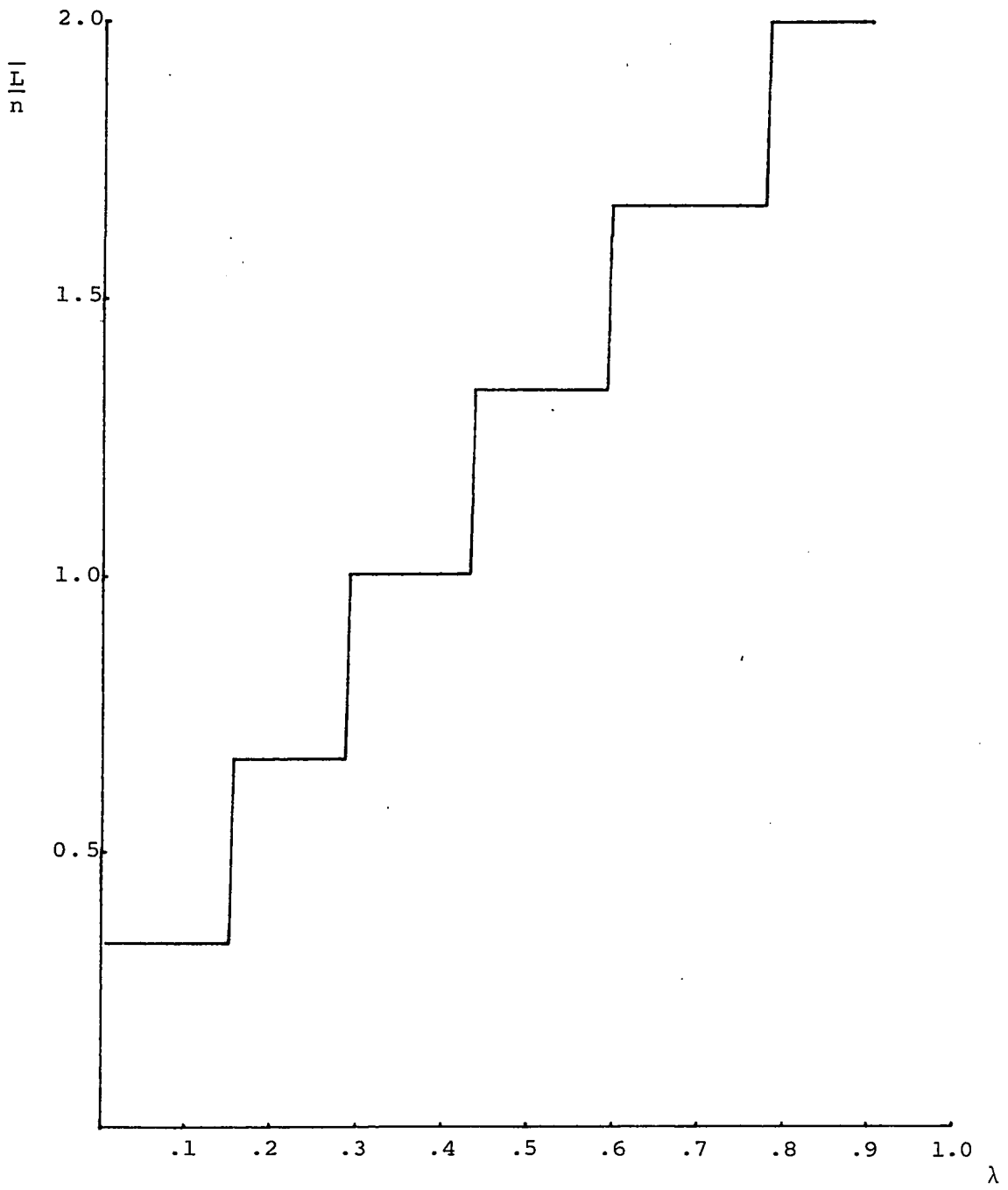

Figure 7:- Normalized average length of the encoded sequence σ for the distortionless case, with the distribution $\Pi_i = \Pi_0 \lambda^1$.

Case (a):

The source output can be limited such that any source digit greater than a given weight is clipped to the allowable maximum weight. Assuming that $t_0 < t$ is the maximum allowable source digit weight, then the resulting n-tuple will be over the set $S_0 = \{0, \pm 1, \pm 2, \ldots \pm t_0\}$, with a resulting maximum possible weight of nt_0. Hence, only the large-weight digits will be affected, so spreading out the distortion. If no digits have a weight $> t_0$, then the source n-tuple will be encoded and transmitted unchanged. This technique is illustrated in the following example.

Example 14

For the (12,8) 2-error-correcting negacyclic code, $S = \{0, \pm 1, \pm 2\}$ and $t = 2$, so that if
B = 0 1 -2 2 1 1 0 0 1 -2 1 1, of weight 12, then limiting to $t_0 = t - 1$ produces
B = 0 1 -1 1 1 1 0 0 1 -1 1 1, the sequence which would be encoded and transmitted. The clipped B has a weight of 9, with a distortion of 3 distributed as follows: 1 unit each of distortion in the 3rd, 4th and 10th digits of B. Thus the matrix B for the clipped n-tuple would contain 5 rows, as opposed to 6 rows for the unclipped n-tuple.

Letting $t_0 = t - 1$, an expression for the average distortion per digit introduced by the use of this scheme can be easily derived.

Clearly, if the source output is limited to t - 1, then every time a +t or -t occurs, one unit of distortion will be introduced in that digit. Hence the total distortion in an n-tuple will be equal to the number of times a ±t occurs in the n-tuple. In any given n-tuple, the average number of ±t's will be $n\Pi_t$, and so will the distortion. Thus the average distortion per digit is $\frac{n\Pi t}{n} = \Pi t$, for $t_0 = t - 1$.

It is clear that since the source n-tuple is decomposed into n-tuples of weight t, every reduction of the source n-tuple's weight by t will result in one less row in *B*. Hence, unless there are frequently t or more ±t's per source n-tuple, there will be little appreciable decrease in $\overline{L}$.

Case (b):

An improvement over the previous scheme would be to limit the weight of the source n-tuple to some value, say Ω, where Ω is less than nt, the maximum block weight. This has the effect of imposing an upper limit on $\overline{L}$, and always results in a fixed-weight n-tuple, if the source block has a weight of Ω or more. If the source n-tuple has a weight < Ω, it is encoded and transmitted unchanged. This technique is illustrated in the following example.

Example 15

For the (12,8) 2-error-correcting negacyclic code, $S = \{0, \pm 1, \pm 2\}$, and $t = 2$, so that if $B = 0\ \ 1\ -2\ \ 2\ \ 1\ \ 1\ \ 0\ \ 0\ \ 1\ -2\ \ 1\ \ 1$, which has a weight of 12. If $\Omega = 8$, limiting using case (b) would produce $B = 0\ \ 1\ -2\ \ 2\ \ 1\ \ 1\ \ 0\ \ 0\ \ 1\ \ 0\ \ 0\ \ 0$, the sequence which would be encoded and transmitted. The matrix B would thus contain 4 rows, as opposed to 6 rows for the unclipped n-tuple. The distortion in this case would be 4, concentrated in the last 3 positions, whereas in Example 14, the clipped n-tuple had a weight of 9, thus producing a B of 5 rows, with a distortion of 3, spread over the n-tuple.

This technique has one obvious advantage, in that Ω can always be made small enough so that $\overline{L}$ will be unity or less, at the cost of increasing distortion. The limiting case is for $\Omega = 1$, since one rt-tuple will be produced whether the n-tuple weight is 0 or 1.

While case (a) is not very effective in reducing $\overline{L}$, it does not concentrate the distortion to the same extent as the method of case (b). On the other hand, even though the second technique can very effectively reduce $\overline{L}$, it does concentrate the distortion. Both techniques limit the weight of the clipped n-tuple to some maximum value but this choice is more flexible in using the method of case (b). The next scheme to be discussed will combine the desirable features of both of these techniques.

Case (c):

From the point of view of distributing the distortion over the block as well as limiting the block weight to Ω, it is desirable to combine cases (a) and (b). Thus all digits would be limited to a maximum weight of t_0, and the block weight would be limited to Ω. This combined scheme is illustrated in the following example.

Example 16

For the (12,8) 2-error-correcting negacyclic code considered in Examples 14 and 15, if
B = 0 1 -2 2 1 1 0 0 1 -2 1 1, combined weight limiting with $t_0 = 1$ and $\Omega = 8$ will produce
B = 0 1 -1 1 1 1 0 0 1 -1 1 0, of weight 8. The distortion in this case is 4, distributed as follows:
1 unit of distortion in the 3rd, 4th, 10th and 12th digits of B. The matrix *B* would then consist of 4 rows.

Table 4 below allows a comparison of the results obtained for the examples used to illustrate schemes (a), (b) and (c).

In the next section, expressions for the average lengths of the encoded sequences are derived for the three weight-limiting techniques introduced in this section.

TABLE 4

EXAMPLE RESULTS FOR THE THREE

WEIGHT-LIMITING SCHEMES

Weight-Limiting Scheme	Block Weight Before Limiting	Block Weight After Limiting ($t_0=1, \Omega=8$)	Total Distortion in Block	Type of Distortion	Relative Decrease in Number of Rows of B
Case (a)	12	9	3	distributed	$\frac{1}{6}$
Case (b)	12	8	4	concentrated	$\frac{1}{3}$
Case (c)	12	8	4	distributed	$\frac{1}{3}$

(ii) Average Length of the Encoded Sequence, Cases with Distortion

Parallel to the distortionless case, equations will next be derived for the average length $\bar{L}$ of the encoded sequence corresponding to each of the three weight-limiting schemes described above.

Case (a):

The average weight $\bar{\omega}$ of a source n-tuple for the distortionless case is given by

$$\bar{\omega} = n(\Pi_1 + 2\Pi_2 + 3\Pi_3 + \ldots + t\Pi_t)$$

Since all digits of weight $> t_0$ are clipped to t_0, this equation becomes

$$\bar{\omega} = [\Pi_1 + 2\Pi_2 + 3\Pi_3 + \ldots + (t_0 - 1)\Pi_{t_0-1} + t_0(\Pi_{t_0} + \Pi_{t_0+1} + \ldots + \Pi_t)]$$

Hence $\bar{L}_a$, the average length of the encoded sequence for scheme (i) is given by

$$\bar{L}_a = \frac{n[\Pi_1+2\Pi_2+ \ldots (t_0-1)\Pi_{t_0-1}+t_0(\Pi_{t_0}+\Pi_{t_0+1}+ \ldots \Pi_t)]}{t} > rt \quad (9)$$

Given some source probability distribution, $\bar{L}_a$ can be computed for a given t_0.

For example, if $\Pi_i = \Pi_0 \lambda^i$, then (9) becomes

$$\bar{\omega} = n\Pi_0 [(\lambda + 2\lambda^2 + 3\lambda^3 + \ldots + (t_0-1)\lambda^{t_0-1} + (t_0)(\lambda^{t_0} + \lambda^{t_0+1} + \ldots + \lambda^t)]$$

$$= n\Pi_0 \{ [\frac{\lambda}{1-\lambda}(\frac{1-\lambda^{t_0-1}}{1-\lambda} - (t_0-1)\lambda^{t_0-1})] + t_0\lambda^{t_0}(1+\lambda+\lambda^2+\ldots\lambda^{t-t_0}) \}$$

$$= n\Pi_0 [(\frac{\lambda}{1-\lambda})(\frac{1-\lambda^{t_0-1}}{1-\lambda}) - (t_0-1)\lambda^{t_0-1} + t_0\lambda^{t_0}(\frac{1-\lambda^{t-t_0+1}}{1-\lambda})]$$

$$\therefore \bar{L}_a = \frac{n\Pi_0 [\frac{\lambda}{1-\lambda}(\frac{1-\lambda^{t_0-1}}{1-\lambda}) - (t_0-1)\lambda^{t_0-1} + t_0\lambda^{t_0}(\frac{1-\lambda^{t-t_0+1}}{1-\lambda})]}{t} > rt$$

Since $\Pi_0 = \frac{1-\lambda}{1-\lambda^{t+1}}$, this can be written as

$$\bar{L}_a = <(\frac{n}{t})(\frac{1-\lambda}{1-\lambda^{t+1}})[\frac{\lambda}{1-\lambda}(\frac{1-\lambda^{t_0-1}}{1-\lambda}) - (t_0-1)\lambda^{t_0-1}$$

$$+ t_0\lambda^{t_0}(\frac{1-\lambda^{t-t_0+1}}{1-\lambda})] > rt$$

For a given negacyclic code, a graph of $\bar{L}_a$ as a function of λ can be drawn, with t_0 as a parameter. This has been done in Figure 8 for the (12,8) t=2 negacyclic code.

Cases (b) and (c):

The average length $\bar{L}$ of the encoded sequence σ for the distortionless case is given by

$$\bar{L} = rt(Q_1 + 2Q_2 + 3Q_3 + \ldots + nQ_n)$$

However, in the case of limiting the block weight, no n-tuple will have a weight greater than Ω, so that the above

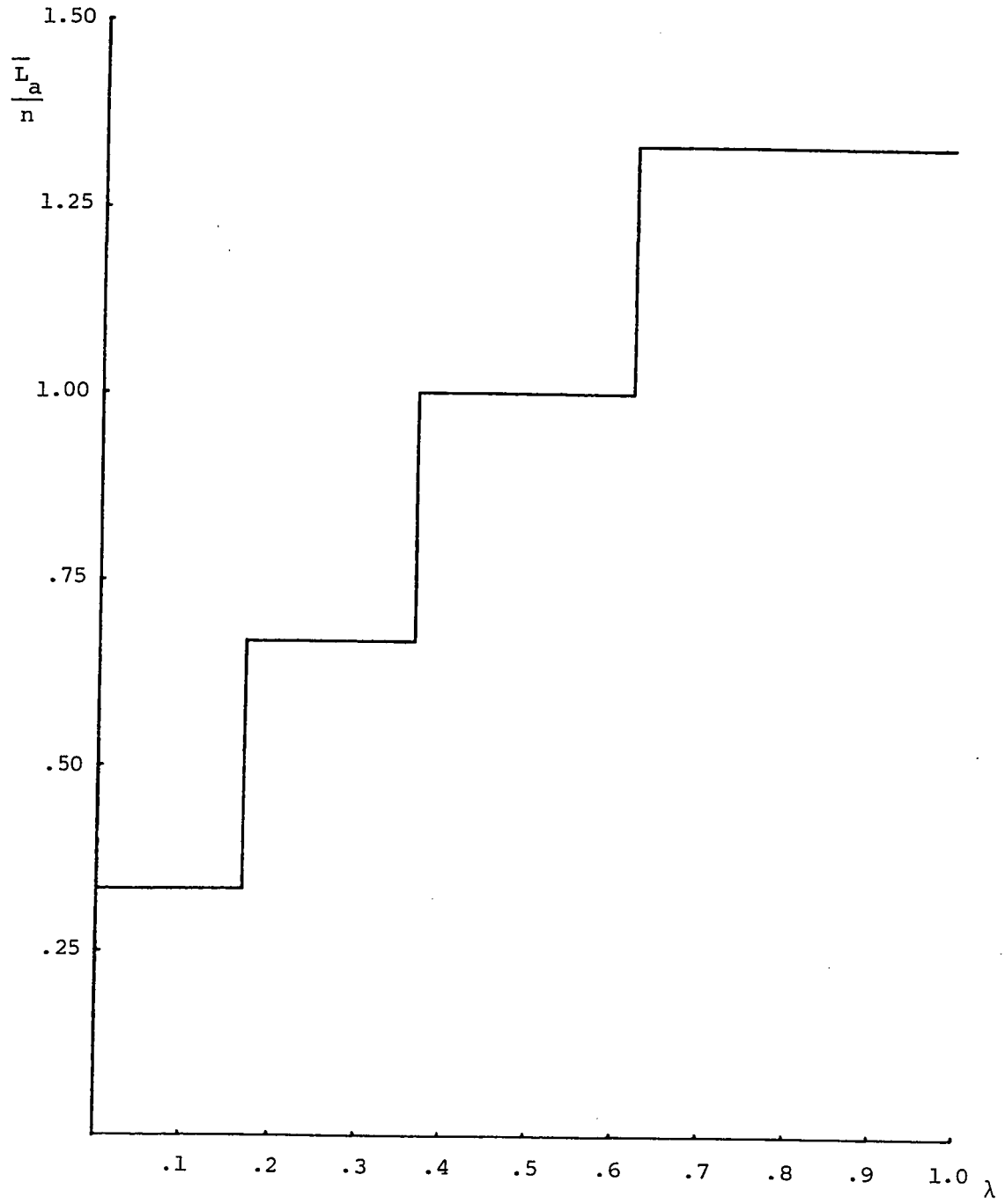

Figure 8:- Normalized average length of the encoded sequence σ for case (a), with $t_0 = t - 1$.

equation reduces to

$$\bar{L}_b = rt(Q_1 + 2Q_2 + 3Q_3 + \ldots + \nu Q_\nu) \qquad (10)$$

where $\nu = \langle \frac{\Omega}{t} \rangle$

Equation (10) reduces to the distortionless case for $\nu = \langle \frac{nt}{n} \rangle$.

For the distribution $Q_i = Q_1 \lambda^{i-1}$, this becomes

$$\bar{L}_b = rtQ_1(1 + 2\lambda + 3\lambda^2 + \ldots + \nu\lambda^{\nu-1}) = rtQ_1 (\frac{1}{1-\lambda}) (\frac{1-\lambda^\nu}{1-\lambda} - \nu\lambda^\nu)$$

and since $Q_1 = \frac{1-\lambda}{1-\lambda^n}$,

$$\bar{L}_b = rt(\frac{1-\lambda}{1-\lambda^n}) (\frac{1}{1-\lambda}) (\frac{1-\lambda^\nu}{1-\lambda} - \nu\lambda^\nu) = (\frac{rt}{1-\lambda^n}) (\frac{1-\lambda^\nu}{1-\lambda} - \nu\lambda^\nu) \qquad (11)$$

It is clear that equation (11) also describes $\bar{L}_c$, the average length of the encoded sequence σ for case (c) so that

$$\frac{\bar{L}_c}{n} = \frac{\bar{L}_b}{n} = (\frac{rt}{n}) (\frac{1}{1-\lambda^n}) (\frac{1-\lambda^\nu}{1-\lambda} - \nu\lambda^\nu)$$

Figure 9 is a graph of equation (11) for the probability distribution defined by equation (3), with Ω as parameter, for the (12,8) 2-error-correcting negacyclic code. It should be noted that for the case where $\Omega = nt = 24$, there is no limiting, and so this reduces to the distortionless case.

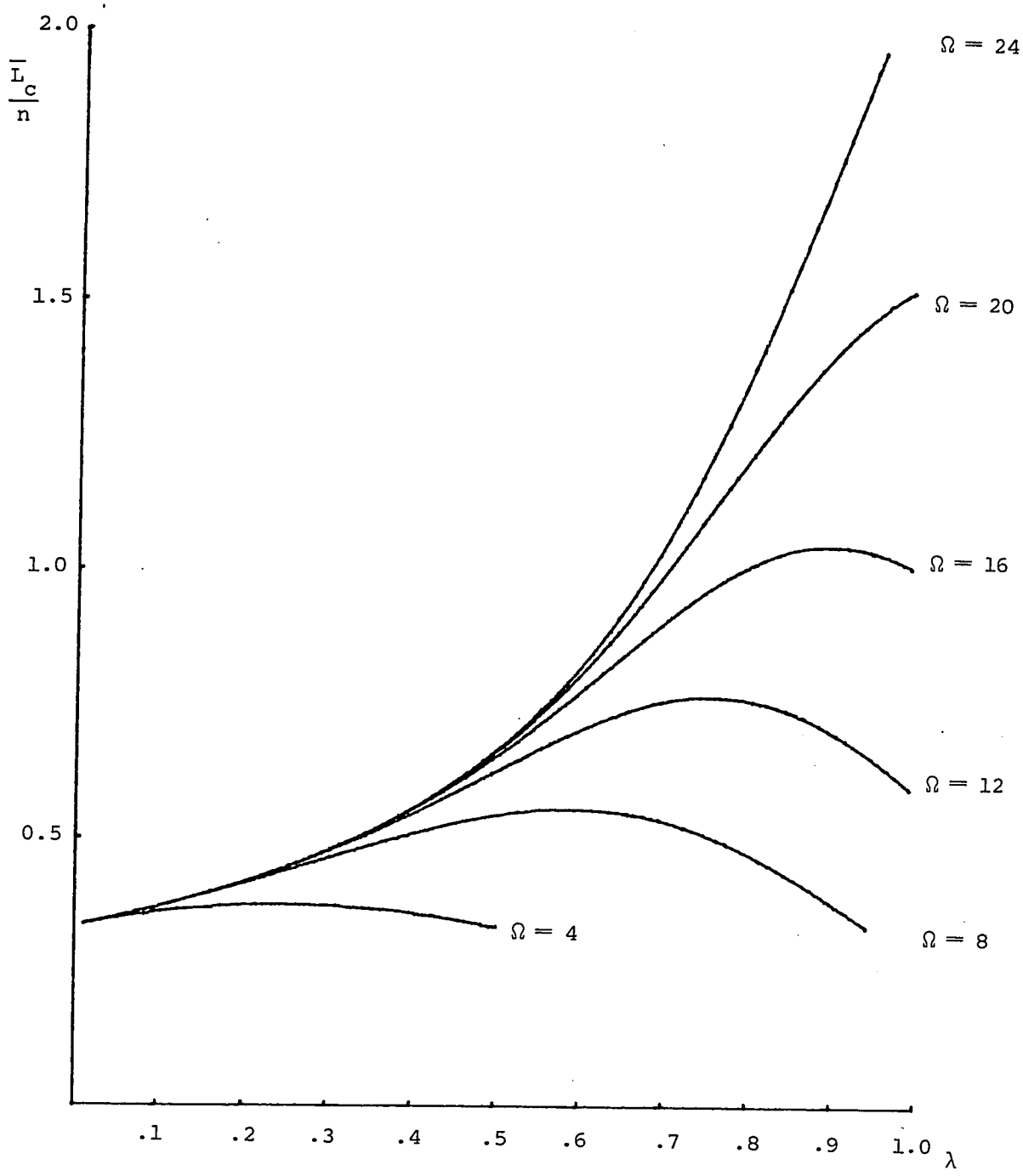

Figure 9:- Normalized average length of the encoded sequence σ for cases (b) and (c), with Ω as parameter.

One other comment can be made here. Since $\frac{\bar{L}_c}{n}$ cannot be less than $(\frac{rt}{n})$, λ must be constrained such that $\frac{\bar{L}_c}{n}$ min $> (\frac{rt}{n})$, which implies that

$$(\frac{1}{1 - \lambda^n})(\frac{1 - \lambda^\nu}{1 - \lambda} - \nu\lambda^\nu) \geq 1$$

must be satisfied.

(iii) Decrease in Average Length of the Encoded Sequence

In this section, expressions for Δ, the normalized decrease in average length, from the distortionless case for cases (a), (b) and (c) are derived, and curves are drawn.

Case (a):

From equation (8), then Δ_a, the normalized decrease in average length of the encoded sequence σ is given by

$$\Delta_a = \frac{\bar{L}}{n} - \frac{\bar{L}_a}{n} = \{<(\frac{n}{t})(\frac{\lambda}{1 - \lambda^{t+1}})(\frac{1-\lambda^t}{1-\lambda} - t\lambda^t)> \frac{rt}{n}\}$$

$$- \{<(\frac{n}{t})(\frac{1-\lambda}{1 - \lambda^{t+1}})[\frac{\lambda}{1-\lambda}(\frac{1 - \lambda^{t_0-1}}{1-\lambda}) - (t_0-1)\lambda^{t_0-1}$$

$$+ t_0\lambda^{t_0}(\frac{1 - \lambda^{t - t_0+1}}{1-\lambda})]> \frac{rt}{n}\}$$

$$= \frac{rt}{n}\ \{<(\frac{n}{t})\ (\frac{1}{1-\lambda^{t+1}})\ [\lambda\ (\frac{1-\lambda^{t}}{1-\lambda}\ -\ t\lambda^{t})\ -\ (1-\lambda)\ [\frac{\lambda}{1-\lambda}\ (\frac{1\ -\lambda^{t_0-1}}{1-\lambda})$$

$$-\ (t_0-1)\lambda^{t_0-1}\ +\ t_0\lambda^{t_0}\ (\frac{1-\lambda^{t-t_0+1}}{1-\lambda})]\ ]\}$$

Let $A = \frac{1}{1-\lambda^{t+1}}$. Then

$$\Delta_a = \frac{rt}{n}\{A(\frac{\lambda-\lambda^{t+1}}{1-\lambda}\ -\ t\lambda^{t+1}\ -\ (1-\lambda)\ (\frac{\lambda}{1-\lambda})\ (\frac{1-\lambda^{t_0-1}}{1-\lambda}) + (1-\lambda)\ (t_0-1)\lambda^{t_0-1}$$

$$-\ (1-\lambda)\ t_0\lambda^{t_0}\ (\frac{1\ -\ \lambda^{t-t_0+1}}{1-\lambda})\ \}$$

$$\Delta_a\ =\ \frac{rt}{n}\ <(\frac{n}{t})\ (\frac{1}{1-\ \lambda^{t+1}})\ [\frac{\lambda-\lambda^{t+1}}{1-\lambda}\ -\ t\lambda^{t+1}\ -\ \lambda(\frac{1-\lambda^{t_0-1}}{1-\lambda})$$

$$+(1-\lambda)\ (t_0-1)\lambda^{t_0-1}\ -\ t_0\lambda^{t_0}\ (1-\ \lambda^{t-t_0+1})\ ]>$$

$$=\ \frac{rt}{n}\ <(\frac{n}{t})\ \frac{1}{1-\lambda^{t+1}}\ [\frac{\lambda-\lambda^{t+1}}{1-\lambda}\ -\ (\frac{\lambda+\lambda^{t_0}}{1-\lambda})-\ t\lambda^{t+1}\ +\ (1-\lambda)\ (t_0-1)\lambda^{t_0-1}$$

$$-\ t_0\lambda^{t_0}\ +\ t_0\lambda^{t-t_0+1+t_0}\ ]>$$

$$=\ \frac{rt}{n}\ <(\frac{n}{t})\ \frac{1}{1-\lambda^{t+1}}\ [\frac{\lambda-\lambda-\lambda^{t+1}\ +\lambda^{t_0}}{1-\lambda}\ -\ t\lambda^{t+1}\ +\ (t_0-1)\lambda^{t_0-1}$$

$$-\lambda(t_0-1)\lambda^{t_0-1}\ -\ t_0\lambda^{t_0}\ +\ t_0\lambda^{t+1}\ ]>$$

$$=\ \frac{rt}{n}\ <(\frac{n}{t})\ \frac{1}{1-\lambda^{t+1}}\ [\frac{\lambda^{t_0}-\lambda^{t+1}}{1-\lambda}\ -\ t\lambda^{t+1}\ +\ t_0\lambda^{t_0-1}-\lambda^{t_0-1}\ -\ t_0\lambda^{t_0}$$

$$+\ \lambda^{t_0}\ -\ t_0\lambda^{t_0}\ +\ t_0\lambda^{t+1}\ ]>$$

$$=\ \frac{rt}{n}\ <(\frac{n}{t})\ (\frac{1}{1\ -\lambda^{t+1}})[\frac{\lambda^{t_0}-\ \lambda^{t+1}}{1-\lambda}\ +\ (t_0-t)\lambda^{t+1}\ +\ (t_0-1)\lambda^{t_0-1}$$

$$+ (1-2t_0)\lambda^{t_0}]>$$

And so

$$\Delta_a = \frac{rt}{n} <(\frac{n}{t}) \frac{1}{1-\lambda^{t+1}} [\frac{\lambda^{t_0}-\lambda^{t+1}}{1-\lambda} + (t_0-1)\lambda^{t_0-1} + (1-2t_0)\lambda^{t_0}$$

$$- (t-t_0)\lambda^{t+1}]> \qquad (12)$$

Figure 10 is a graph of equation (12) for the (12,8) t=2 negacyclic code, for the case $t_0 = t-1$.

Case (b):

$$\Delta_c = \Delta_b = \frac{\bar{L}}{n} - \frac{\bar{L}_b}{n} = [(\frac{rt}{n})(\frac{1}{1-\lambda} - \frac{n\lambda^n}{1-\lambda^n})]$$

$$-[\frac{rt}{n} (\frac{1}{1-\lambda^n})(\frac{1-\lambda^\nu}{1-\lambda} - \nu\lambda^\nu)]$$

$$= \frac{rt}{n} [(\frac{1}{1-\lambda} - \frac{n\lambda^n}{1-\lambda^n}) - (\frac{1}{1-\lambda^n})(\frac{1-\lambda^\nu}{1-\lambda} - \nu\lambda^\nu)]$$

$$= \frac{rt}{n} [\frac{1}{1-\lambda} - \frac{n\lambda^n}{1-\lambda^n} - (\frac{1}{1-\lambda^n}) (\frac{1-\lambda^\nu}{1-\lambda}) + \frac{1}{1-\lambda^n} (-\nu\lambda^\nu)]$$

$$= \frac{rt}{n} [\frac{1}{1-\lambda} - \frac{n\lambda^n}{1-\lambda^n} - \frac{\nu\lambda^\nu}{1-\lambda^n} - (\frac{1}{1-\lambda^n})(\frac{1-\lambda^\nu}{1-\lambda})]$$

$$= \frac{rt}{n} [\frac{1}{1-\lambda} - (\frac{n\lambda^n - \nu\lambda^\nu}{1-\lambda^n}) - \frac{1}{1-\lambda^n} (\frac{1}{1-\lambda}) + (\frac{\lambda^\nu}{1-\lambda^n}) \frac{1}{1-\lambda}]$$

$$= \frac{rt}{n} [\frac{1}{1-\lambda} (1 - \frac{1}{1-\lambda^n} + \frac{\lambda^\nu}{1-\lambda^n}) - (\frac{n\lambda^n - \nu\lambda^\nu}{1-\lambda^n})]$$

And so

$$\Delta_b = \frac{rt}{n} [\frac{1}{1-\lambda} (1 - \frac{(1-\lambda^\nu)}{1-\lambda^n}) - (\frac{n\lambda^n - \nu\lambda^\nu}{1-\lambda^n})] \qquad (13)$$

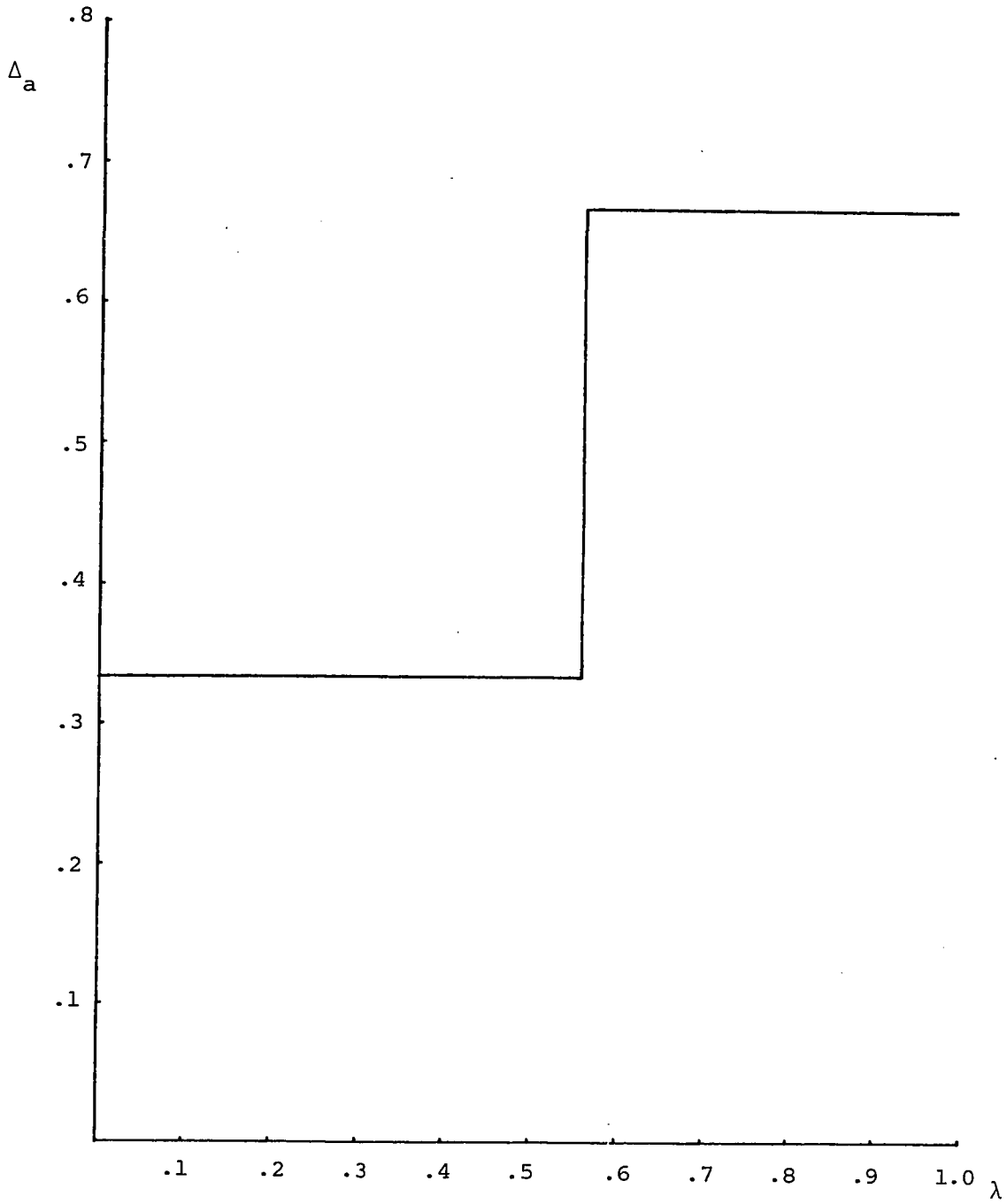

Figure 10:- Normalized change in average length from the distortionless case for case (a), with $t_0 = t - 1$.

Figure 11 is a graph of equation (13) for the (12,8) t=2 negacyclic code, with Ω as parameter.

(iv) Average Distortion per Digit

Let Π_i represent the probability that the source puts out +i or -i as defined in section 3b. Expressions for the total average distortion per block D' and the average distortion per digit D, for the three cases which introduce distortion will be derived next.

Case (a):

The total average distortion per source block D_a' for this case is given by the sum of the products of the probability of a digit being greater than t_0 by the Lee distance between that digit and t_0, over the source block. Hence,

$$D_a' = n[\ \Pi_{t_0+1}(t_0+1 - t_0) + \Pi_{t_0+2}(t_0+2-t_0) + \ . \ . \ .$$

$$+ \Pi_{t-1}(t-1-t_0) + \Pi_t(t-t_0)]$$

Let $\delta = t-t_0$. Then the average distortion D_a per digit is given by

$$D_a = \Pi_{t_0+1}(1) + \Pi_{t_0+2}(2) + \ . \ . \ . + \Pi_{t-1}(\delta-1) + \Pi_t(\delta)$$

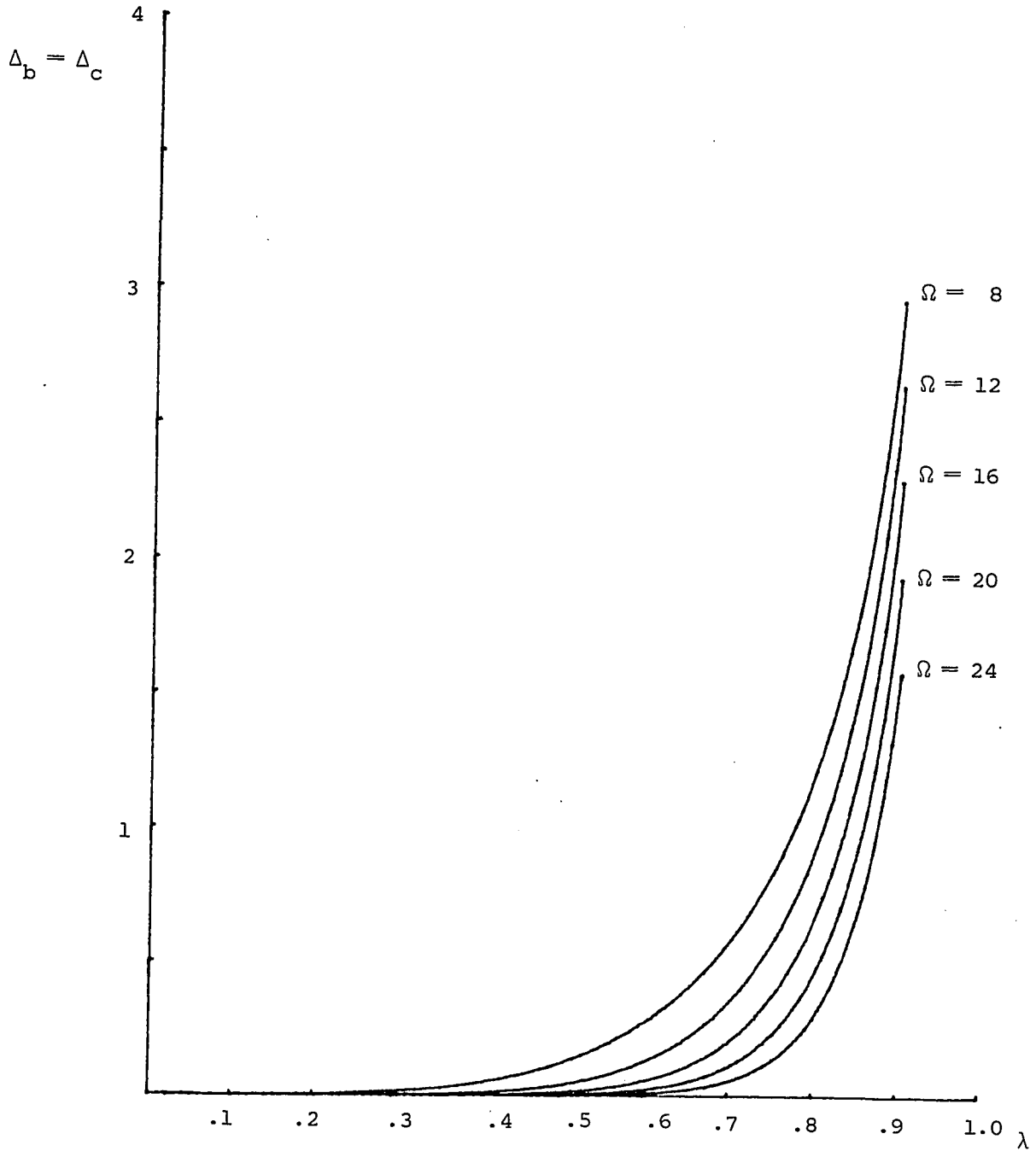

Figure 11:- Normalized change in average length from the distortionless case for cases (b) and (c), with Ω as parameter.

For a given source probability distribution, code, and t_0, D_a can be computed. Let $\Pi_i = \Pi_0 \lambda^i$. Then $\Pi_0 = \frac{1-\lambda}{1-\lambda^{t+1}}$ and so

$$D_a = \Pi_0 \lambda^{t_0+1}(1) + \Pi_0 \lambda^{t_0+2}(2) + \ldots + \Pi_0 \lambda^{t-1}(\delta-1)$$

$$+ \Pi_0 \lambda^{t}(\delta)$$

$$= \Pi_0 \lambda^{t}[1 + 2\lambda + 3\lambda^2 + \ldots + \delta\lambda^{\delta-1}]$$

$$= \Pi_0 \frac{\lambda^{t_0+1}}{1-\lambda} [\frac{1 - \lambda^{\delta}}{1-\lambda} - \delta\lambda^{\delta}]$$

which yields

$$D_a = \frac{\Pi_0 \lambda^{t_0+1}}{1-\lambda} [\frac{1 - \lambda^{t-t_0}}{1-\lambda} - (t - t_0)\lambda^{t-t_0}]$$

Cases (b) and (c):

The derivation of an expression for D_b' consists simply in finding the difference between the average weight $\bar{\omega}$ without any limiting of the block weight and the average weight with block weight limiting. Clearly, the weight with block weight limiting $\leq \Omega$. If the source n-tuple weight is less than Ω, there will be no weight limiting, and thus, no distortion.

The average weight $\bar{\omega}$ of a source block is given by:

$$\bar{\omega} = n[\Pi_0(0) + \Pi_1(1) + \Pi_2(2) + \ldots + \Pi_t(t)]$$

$$= n[\Pi_1 + 2\Pi_2 + 3\Pi_3 + \ldots + t\Pi_t]$$

Therefore the average distortion D per digit for cases (b) and (c) is given by:

$$D_b = D_c = \Pi_1 + 2\Pi_2 + 3\Pi_3 + \ldots + t\Pi_t - \frac{\Omega}{n} \qquad (14)$$

In the next section, a variation on the joint source and channel encoding scheme presented in section 3 will be introduced. Referred to as an orthogonal joint source and channel encoding scheme, it also results in data compression as well as an error-detection capability.

As before, the distortionless source encoding scheme will be considered first.

4. Orthogonal Distortionless Encoding Scheme

As in section 3, let V be a t-error-correcting negacyclic code of length n over $S = \{0, \pm 1, \pm 2, \ldots \pm\frac{p-1}{2}\}$, with syndromes of length rt.

With reference to V, the source output can be partitioned into blocks of n digits. Let B be such a block, with m nonzero digits. Then the maximum value of m is n.

If m = 0, then B is replaced with the rt-tuple of all zeros, the syndrome for the case of no error.

Now B can be treated as the sum of m n-tuples $B_1, B_2, \ldots, B_m$ such that the m-by-n matrix B_0, in which the rows are $B_1, B_2, \ldots, B_m$, satisfies the following properties.

(i) Every row of B_0 has exactly one nonzero digit.

(ii) Every column of B_0 has at most one nonzero element.

(iii) All elements to the right of the jth element in the (i-1)th row ($i \geq 2$) of B_0 must be zero if the jth element in the ith row of B_0 is nonzero.

Other properties which stem directly from (i) and (ii) above are:

(iv) The weight of every row of B_0 is greater than 0, but $\leq t$.

(v) The weight of every column of B_0 is, at most, t.

It is now clear why this scheme is referred to as being orthogonal. The decomposition of a source n-tuple produces a matrix B_0 which is orthogonal in the sense that no row or column contains more than one nonzero entry.

The decomposition of a source output sequence B into the matrix B_0 is illustrated in the following example.

Example 17

The (12,8) negacyclic code with t = r = 2 is the relevant code in this example.

Suppose B = 0 1 -2 2 1 1 0 0 1 -2 1 1 is a source output sequence. It contains 9 nonzero digits, and has a weight of 12.

Then m = 9, so that

$$B_0 = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -2 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

This should be compared with Example 13 for the original scheme, where, for the same source n-tuple B, B consists of 6 rows.

A little thought will make it clear that the orthogonal scheme, for the case where all nonzero digits (or all except the last nonzero digit) have a weight of t, will yield the same size decomposition matrix as the nonorthogonal scheme. This will also be true if B is an n-tuple of weight zero. In all other cases, B_0 will contain more rows than B, for a given source n-tuple. Hence, the orthogonal scheme will, on the average, be less efficient

in its encoding. It would, however, be expected to have a better error-detection capability than the nonorthogonal scheme, since more checks can be made on the matrix.

As in the nonorthogonal scheme, the matrix B_0 could be transmitted row by row, but since by construction each row of B_0 must have a weight < t, each row can be considered to be an error vector corresponding to a correctable error for a t-error-correcting negacyclic code. Then, since any correctable error pattern can be represented by its syndrome, an rt-tuple, each row of B_0 can be replaced by its corresponding syndrome, which can then be transmitted.

Thus, if σ_i represents the syndrome of B_i, the ith row of matrix B_0, then the block B can be replaced by the encoded sequence $\sigma = \sigma_1\sigma_2 \ldots \sigma_m$, which it is noted, is mrt digits long. Hence, for the previous example, the sequence B of length 12 would be replaced by an encoded sequence σ 36 digits in length. In the corresponding example for the nonorthogonal case, the encoded sequence σ was 24 digits long for the same source n-tuple B.

The sequence σ is then transmitted over the channel. At the receiving end, σ' is received, where $\sigma' = \sigma_1' \ \sigma_2' \ \ldots \ \sigma_m'$. Corresponding to each syndrome σ_i', the error-pattern B_i' is obtained and the matrix B_0' is formed in which the ith row is B_i'.

If B_0' satisfies all of the properties satisfied by B_0, then the rows of B_0' are added to obtain B', which,

it is decided, is the sequence put out by the source. If not, either a request for retransmission is transmitted back to the source or the block is abandoned.

As in the nonorthogonal scheme, data compression is assured if the length L of the encoded sequence σ is $< n$. As in the nonorthogonal case, this scheme will not in general be optimal.

In the orthogonal scheme, as in the nonorthogonal scheme, the error detection also occurs in two stages. In the first, the weight of the n-tuple corresponding to the received syndrome is determined, and if it is less than t, and is not the last row of B_0, an error has been detected. If no error is detected, then B_0 can be formed and all its properties verified in order to check for errors.

5. The Rate Distortion Function for Nonbinary Sources [54],[55]

A basic question which may be asked of any communications system is the following: if a user is willing to accept an average amount of distortion D per digit in the received (or stored) messages, then what is the least average number of bits required to transmit the message? The rate distortion function R(D) gives an answer to this question, at least in theory.

R(D) is the effective rate at which the source produces information subject to the constraint that the user can tolerate an average distortion of D. The rate at

which a source produces information subject to a requirement of perfect reproduction is known as the entropy of the source.

Of course, this discussion is meaningless unless the meaning of distortion is first clarified. Exactly what is meant by distortion, and how can it be measured? Conceptually, distortion can be thought of as the amount of "difference" between the source symbol and the received symbol. However, a description of this "difference" and a message which is both physically meaningful and analytically tractable is usually difficult to achieve.

Knowledge of the distortion measure is not enough, however. Computation of the rate distortion function also requires knowledge of the source statistics, which are not always available.

For a given discrete memoryless source and a single-letter fidelity criterion, the expected value of the single-letter distortion measure will depend on the channel conditional probabilities. However, only those probability assignments referred to as D-admissible, for which the expected value of the single-letter distortion measure $\leqslant D$, are considered.

The distortion measure is usually specified as a distortion matrix, the elements of which must be non-negative.

The average mutual information will also depend on the probability assignments. Hence, for a fixed D,

the rate-distortion function is defined as

$$R(D) = \min_{Q \in Q_D} I(Q)$$

where Q_D is the set of all D-admissible conditional probabilities and I(Q) is the average mutual information for the conditional probability set Q.

It can be shown [56] that R(D) is positive for $0 \leq D < D_{max}$, and vanishes for $D \geq D_{max}$. R(D) is also a continuous, monotonic decreasing, convex downward function in the interval $[0, D_{max}]$. The slope of R(D) is a continuous function of D in the interval $[0, D_{max})$, but may be discontinuous at the point $D = D_{max}$. The slope of R(D) may also be constant over a range of values of D.

Obtaining the rate distortion function involves minimizing the average mutual information by an appropriate choice of conditional probabilities, subject to the following constraints:

(i) All conditional probabilities are non-negative.

(ii) The sum of all conditional probabilities is unity.

(iii) The distortion weighted by the output probability distribution is equal to D.

If the constraints are ignored, a solution can be obtained by the use of Lagrange multipliers. However, the solution will generally be valid only for small values of D.

If the conditional probability distribution that yielded this solution is non-negative, then the complete problem has been solved.

The general (unconstrained) solution is given by generating points on the R(D) curve, with s and q_i as parameters:

$$D = \sum_{ji} \lambda_j q_i e^{2\rho_{ji}} \rho_{ji} \tag{15}$$

and

$$R = sD + \sum_j p_j \log \lambda_j \tag{16}$$

where $$\lambda_j = \frac{1}{\sum_i q_i e^{q\rho_{ji}}}$$

p_j = probability of occurrence of jth source symbol.

q_i = a parameter (probability of occurrence of ith output (or reproducing) symbol).

s = a parameter (the slope of R(D) at a given point).

ρ_{ji} = entry of jth row and ith column of the distortion matrix.

Using these two equations, D and R can be obtained for given values of q_i and s.

If the unconstrained solution described by (15) and (16) results in one or more $q_i \leq 0$ for some $s (s > -\infty)$, then all

symbols for which $q_i = 0$ can be dropped from the reproducing alphabet without affecting the solution.

Hence the reproducing alphabet can be partitioned into two subsets, the boundary set B and the interior set V. Then, for every value of s, $i \varepsilon B$ if $q_i = 0$ is optimum, and $i \varepsilon V$ otherwise. For $i \varepsilon V$, the solution described by equations (15) and (16) is still valid.

In general, evaluation of the rate distortion function cannot be done analytically. However, many rate distortion functions have been generated using computers. Arimoto [57] and Blahut [58] have all proposed algorithms for the computation of R(D). Another approach has been to derive the Shannon lower bound $R_L(D)$ to R(D) and then to find conditions on source statistics and distortion measure for the existence of a D* such that the bound is identical to R(D) for all values of $D \varepsilon [0, D^*]$, [59],[60].

However, there are many cases of interest for which an analytic expression can be found. In the case to be considered here, the source is both discrete and memoryless, and the source alphabet is identical to the output alphabet, so that the distortion matrix will be square. The input alphabet is made up of the elements of $S = \{0, \pm 1, \pm 2, \ldots, \pm\frac{p-1}{2}\}$, which are relevant to negacyclic codes. The distortion measure is taken to be the absolute value of the difference, i.e., the Lee distance, between the input and output vectors.

The notation and approach used in the example are close to that of Berger [56], and in the same context, it is noted that all logarithms are to be to the base e, unless otherwise indicated. However, logarithms to the base 2 are usually used when graphing, to simplify comparison with other results.

The example follows.

Example 18

Let the single-letter distortion measure be described by the distortion matrix ρ whose entries are $\rho_{ij} = |i-j|$. The matrix ρ will be p-by-p, since the distortion can vary between 0 and (p-1). Hence,

$$\rho = \begin{bmatrix} 0 & 1 & 2 & \cdots & p-1 \\ 1 & 0 & 1 & \cdots & p-2 \\ 2 & 1 & 0 & \cdots & p-3 \\ \cdot & \cdot & & & \\ \cdot & & \cdot & & \\ \cdot & & & \cdot & \\ p-1 & p-2 & p-3 & \cdots & 0 \end{bmatrix}$$

Let matrix $A = (a_{ij}) = (e^{s\rho_{ij}})$, so that

$$A = \begin{bmatrix} 1 & \beta & \beta^2 & \beta^3 & \cdots & \beta^{p-1} \\ \beta & 1 & \beta & \beta^2 & \cdots & \beta^{p-2} \\ \beta^2 & \beta & 1 & \beta & \cdots & \beta^{p-3} \\ & & & \vdots & & \\ \beta^{p-1} & \beta^{p-2} & \beta^{p-3} & \beta^{p-4} & \cdots & 1 \end{bmatrix}$$

where $\beta = e^s$.

While the problem of evaluating R(D) for the type of distortion matrix under consideration has been solved, the case of p = 3 will be worked out. This is the simplest case which is meaningful for negacyclic codes. The inverse A^{-1} of A is known to be given by: [61]

$$A^{-1} = \frac{1}{1-\beta^2} \begin{bmatrix} 1 & -\beta & 0 & 0 & \cdots & 0 \\ -\beta & 1+\beta^2 & -\beta & 0 & \cdots & 0 \\ 0 & -\beta & 1+\beta^2 & -\beta & \cdots & 0 \\ & & & \vdots & & \\ 0 & 0 & 0 & \cdots & -\beta & 1+\beta^2 & -\beta \\ 0 & 0 & 0 & \cdots & 0 & -\beta & 1 \end{bmatrix}$$

For $p = 3$, the expressions for ρ and A^{-1} reduce to

$$\rho = \begin{bmatrix} 0 & 1 & 2 \\ 1 & 0 & 1 \\ 2 & 1 & 0 \end{bmatrix}$$

$$A^{-1} = \frac{1}{1-\beta^2} \begin{bmatrix} 1 & -\beta & 0 \\ -\beta & 1+\beta^2 & -\beta \\ 0 & -\beta & 1 \end{bmatrix}$$

Due to symmetry, $A^T = A$, and so,

$$(A^T)^{-1} = A^{-1} = \frac{1}{(1-\beta^2)} \begin{bmatrix} 1 & -\beta & 0 \\ -\beta & 1+\beta^2 & -\beta \\ 0 & -\beta & 1 \end{bmatrix}$$

Now $u_i = \lambda_i p_i$, where p_i is the probability of occurrence of the ith source symbol. The u_i's are computed using

$$\bar{u} = (A^T)^{-1} \bar{1}$$

where $\bar{u}$ and $\bar{1}$ are column vectors.

In this case,

$$\begin{bmatrix} u_1 \\ u_2 \\ u_3 \end{bmatrix} = (A^T)^{-1} \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} = \frac{1}{1-\beta^2} \begin{bmatrix} 1-\beta \\ (1-\beta)^2 \\ 1-\beta \end{bmatrix} = \frac{1}{1+\beta} \begin{bmatrix} 1 \\ 1-\beta \\ 1 \end{bmatrix} = \begin{bmatrix} \frac{1}{1+\beta} \\ \frac{1-\beta}{1+\beta} \\ \frac{1}{1+\beta} \end{bmatrix}$$

Given the u_i's, the v_i's are computed using

$$v_i = \frac{p_i}{u_i}$$

which yields, in this case,

$$v_1 = p_1(1+\beta)$$

$$v_2 = p_2(\frac{1+\beta}{1-\beta})$$

$$v_3 = p_3(1+\beta)$$

and v_1, v_2, v_3 make up $\overline{v}$, a column vector.

Next, $\overline{q}$, the column vector of channel output probabilities, given A^{-1} and $\overline{v}$, can be evaluated using

$$\overline{q} = A^{-1}\overline{v}$$

so that

$$\begin{bmatrix} q_1 \\ q_2 \\ q_3 \end{bmatrix} = A^{-1} \begin{bmatrix} v_1 \\ v_2 \\ v_3 \end{bmatrix} = \frac{1}{1-\beta^2} \begin{bmatrix} 1 & -\beta & 0 \\ -\beta & 1+\beta^2 & -\beta \\ 0 & -\beta & 1 \end{bmatrix} (1+\beta) \begin{bmatrix} p_1 \\ \frac{p_2}{1-\beta} \\ p_3 \end{bmatrix}$$

or

$$\begin{bmatrix} q_1 \\ q_2 \\ q_3 \end{bmatrix} = \frac{1}{1-\beta} \begin{bmatrix} p_1 - p_2\left(\frac{\beta}{1-\beta}\right) \\ -p_1\beta + \frac{p_2(1+\beta^2)}{1-\beta} - p_3\beta \\ -p_2\left(\frac{\beta}{1-\beta}\right) + p_3 \end{bmatrix} \quad (17)$$

Now p_1, p_2 and p_3 are the probabilities of the source putting out a -1, a 0, or a +1, respectively. If the source is assumed to put out -1 and +1 with the same probability, $p_1 = p_3$ and so $p_2 = 1 - 2p_1 = 1 - 2p_3$. Equivalently, $p_1 = p_3 = \frac{1 - p_2}{2}$.

Equation (17) then becomes

$$\begin{bmatrix} q_1 \\ q_2 \\ q_3 \end{bmatrix} = \frac{1}{(1-\beta)} \begin{bmatrix} \frac{1-p_2}{2} - p_2\left(\frac{\beta}{1-\beta}\right) \\ -(1-p_2)\beta + \frac{p_2(1+\beta^2)}{1-\beta} \\ -p_2\left(\frac{\beta}{1-\beta}\right) + \frac{1-p_2}{2} \end{bmatrix}$$

$$= \frac{1}{1-\beta} \begin{bmatrix} \frac{1}{2} + p_2\left(-\frac{1}{2} - \frac{\beta}{1-\beta}\right) \\ -\beta + p_2\left(\beta + \frac{1+\beta^2}{1-\beta}\right) \\ \frac{1}{2} + p_2\left(\frac{-\beta}{1-\beta} - \frac{1}{2}\right) \end{bmatrix} = \frac{1}{1-\beta} \begin{bmatrix} \frac{1}{2} - \frac{p_2}{2}\left(\frac{1+\beta}{1-\beta}\right) \\ -\beta + p_2\left(\frac{1+\beta}{1-\beta}\right) \\ \frac{1}{2} - \frac{p_2}{2}\left(\frac{1+\beta}{1-\beta}\right) \end{bmatrix}$$

Since

$$\lambda_i = \frac{u_i}{p_i} = \frac{1}{v_i} ,$$

$$\lambda_1 = \frac{1}{v_1} = \frac{1}{p_1(1+\beta)} = \frac{2}{(1-p_2)(1+\beta)}$$

$$\lambda_2 = \frac{1}{v_2} = \frac{(1-\beta)}{p_2(1+\beta)}$$

$$\lambda_3 = \frac{1}{v_3} = \frac{1}{p_3(1+\beta)} = \frac{2}{(1-p_2)(1+\beta)}$$

Given this, D can be determined using

$$D = \sum_{i,j} \lambda_i p_i q_j \beta^{\rho_{ij}} \rho_{ij}$$

$$= \lambda_1 p_1 [q_1 \beta^{\rho_{11}} \rho_{11} + q_2 \beta^{\rho_{12}} \rho_{12} + q_3 \beta^{\rho_{13}} \rho_{13}]$$

$$+ \lambda_2 p_2 [q_1 \beta^{\rho_{21}} \rho_{21} + q_2 \beta^{\rho_{22}} \rho_{22} + q_3 \beta^{\rho_{23}} \rho_{23}]$$

$$+ \lambda_3 p_3 [q_1 \beta^{\rho_{31}} \rho_{31} + q_2 \beta^{\rho_{32}} \rho_{32} + q_3 \beta^{\rho_{33}} \rho_{33}]$$

Since all elements on the main diagonal of ρ are zero, this reduces to

$$D = \lambda_1 p_1 [q_2 \beta + 2 q_3 \beta^2]$$

$$+ \lambda_2 p_2 [q_1 \beta + q_3 \beta]$$

$$+ \lambda_3 p_3 [2 q_1 \beta^2 + q_2 \beta]$$

Since $\lambda_1 p_1 = \lambda_3 p_3 = \frac{1}{1+\beta}$, D becomes

$$D = \lambda_1 p_1 [2q_1\beta^2+2q_2\beta+2q_3\beta^2] + \lambda_2 p_2 [2q_1\beta]$$

and as $q_1 = q_3 = \frac{1-q_2}{2}$,

$$D = \lambda_1 p_1 [4q_1\beta^2+2q_2\beta] + \lambda_2 p_2 [2q_1\beta]$$

$$= \lambda_1 (\frac{1-p_2}{2}) [4\beta^2 (\frac{1-q_2}{2}) + 2q_2\beta] + \lambda_2 p_2 [2\beta (\frac{1-q_2}{2})]$$

$$= \lambda_1 (1-p_2) [\beta^2 (1-q_2) + q_2\beta] + \lambda_2 p_2 [\beta (1-q_2)]$$

Substituting for the calculated values of λ_i, the equation becomes

$$D = \frac{1-p_2}{p_1(1+\beta)} [\beta^2 (1-q_2)+q_2\beta] + \frac{(1-\beta)}{p_2(1+\beta)} p_2 [\beta (1-q_2)]$$

$$= \frac{2}{1+\beta}[\beta^2 (1-q_2)+q_2\beta] + \frac{1-\beta}{1+\beta} [\beta (1-q_2)]$$

$$= q_2 [\frac{2\beta}{1+\beta} (\beta^2+\beta) - (\frac{1-\beta}{1+\beta})\beta] + \frac{2\beta^2}{1+\beta} + \frac{\beta(1-\beta)}{1+\beta}$$

$$= q_2 [\frac{2\beta}{1+\beta} (1-\beta) - (\frac{1-\beta}{1+\beta}) \beta] + \frac{\beta+\beta^2}{1+\beta}$$

$$= q_2 \frac{\beta(1-\beta)}{1+\beta} + \beta$$

Since $q_2 = \frac{1}{1-\beta}\ [-\beta+p_2(\frac{1+\beta}{1-\beta})]$, this becomes

$$D = \frac{1}{1-\beta}\ \frac{\beta(1-\beta)}{1+\beta}[p_2(\frac{1+\beta}{1-\beta})\ -\beta]\ +\ \beta$$

$$= \frac{\beta}{1+\beta}\ [p_2(\frac{1+\beta}{1-\beta})\ -\beta]\ +\ \beta$$

$$= \frac{\beta p_2}{1-\beta}\ -\ \frac{\beta^2}{1+\beta}\ +\ \beta$$

And so

$$D = \frac{p_2}{1-\beta}\ +\ \frac{\beta}{1+\beta} \qquad (18)$$

which is the expression for the distortion. It should be noted that D depends on s, since $\beta = e^s$.

From equation (17),

$$R(D) = (\log_e\beta)D + p_1\log\lambda_1 + p_2\log\lambda_2 + p_3\log\lambda_3$$

$$= (\log_e\beta)D + 2p_1\log\lambda_1 + p_2\log\lambda_2$$

$$= (\log_e\beta)D + (1-p_2)\log[\frac{1}{\frac{1-p_2}{2}(1+\beta)}] + p_2\log[\frac{1-\beta}{p_2(1+\beta)}]$$

$$= (\log_e\beta)D + \log[\frac{2}{(1-p_2)(1+\beta)}] + p_2\log[\frac{(1-\beta)}{p_2(1+\beta)}\ \frac{(1-p_2)(1+\beta)}{2}]$$

And so,

$$R(D) = (\log_e\beta)D + \log[\frac{2}{(1-p_2)(1+\beta)}] + p_2\log[\frac{(1-\beta)(1-p_2)}{2p_2}] \qquad (19)$$

is the required rate distortion function. Hence, for different values of p_2 and β, a graph of R(D) can be drawn, with some restrictions, as will be seen below. Figure 12 illustrates this rate distortion function for different values of p_2.

It is clear from the figure that R(0) is the maximum of the R(D) function. R(0), however, is only equal to the source entropy if $p_1 = p_2 = p_3$. The closest approach illustrated in Figure 12 is for $p_1 = p_3 = 0.275$ and $p_2 = 0.45$, which yields an R(0) only slightly less than the source entropy, which is equal to 1.585.

Figure 13 is a graph of the entropy of this source as a function of p_2. Two things can be noted on this figure. The first is that since $p_1 = p_3 = \frac{1-p_2}{2}$, the entropy is unity when $p_2 = 0$, since this reduces to the 2-symbol case, each symbol having a probability of $\frac{1}{2}$. The second point is that the entropy is a maximum for $p_1 = p_2 = p_3 = \frac{1}{3}$, as expected.

For the unconstrained solution for R(D) to be valid, it is required that

$$q_1 = q_3 = \frac{1}{2} - \frac{p_2}{2}\left(\frac{1+\beta}{1-\beta}\right)$$

and

$$q_2 = -\beta + p_2\left(\frac{1+\beta}{1-\beta}\right)$$

be non-negative. From the requirement that

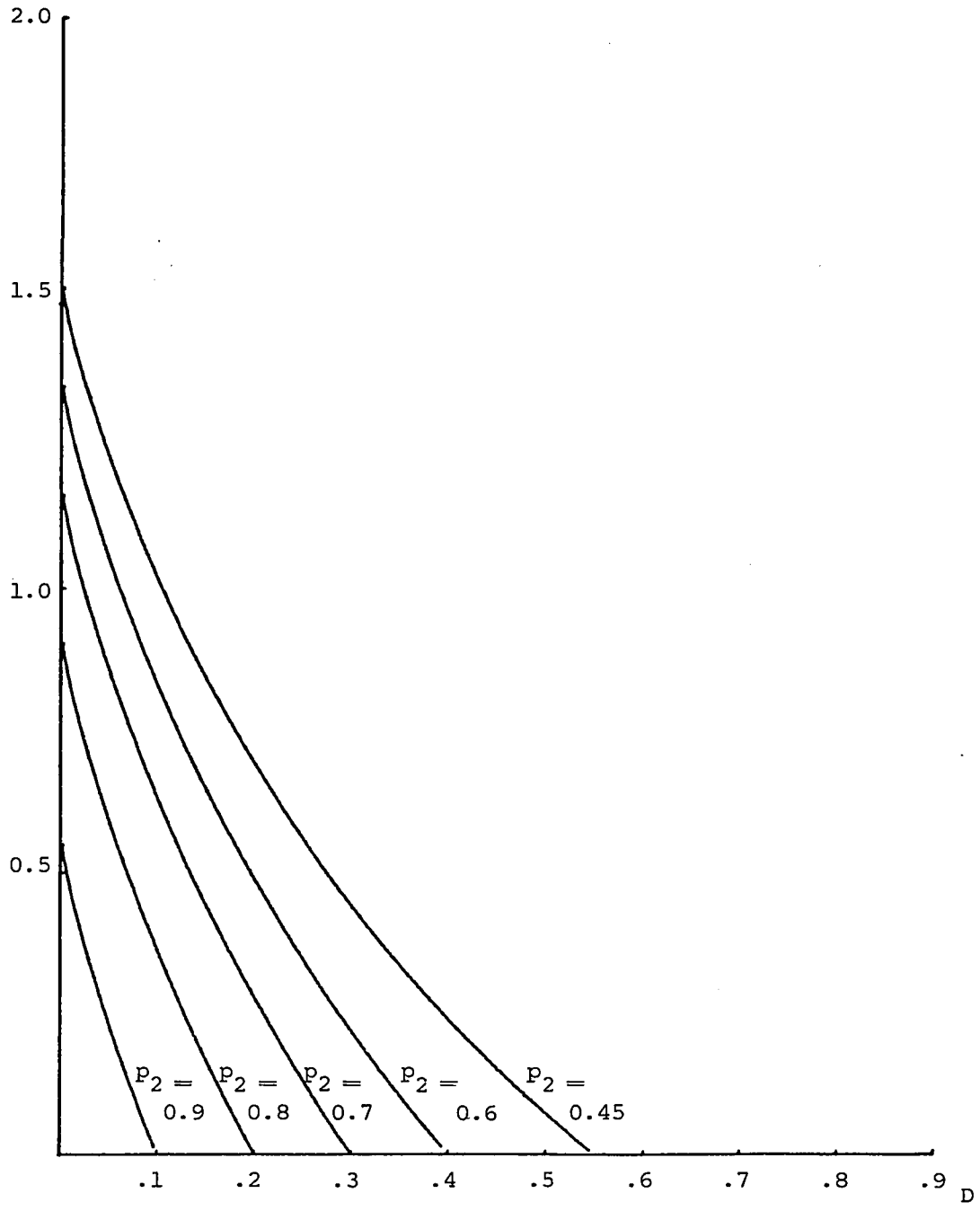

Figure 12:- Rate distortion function of a 3-letter source, with $p_1 = p_3 = \frac{1-p_2}{2}$, for $p_2 \geq 0.45$.

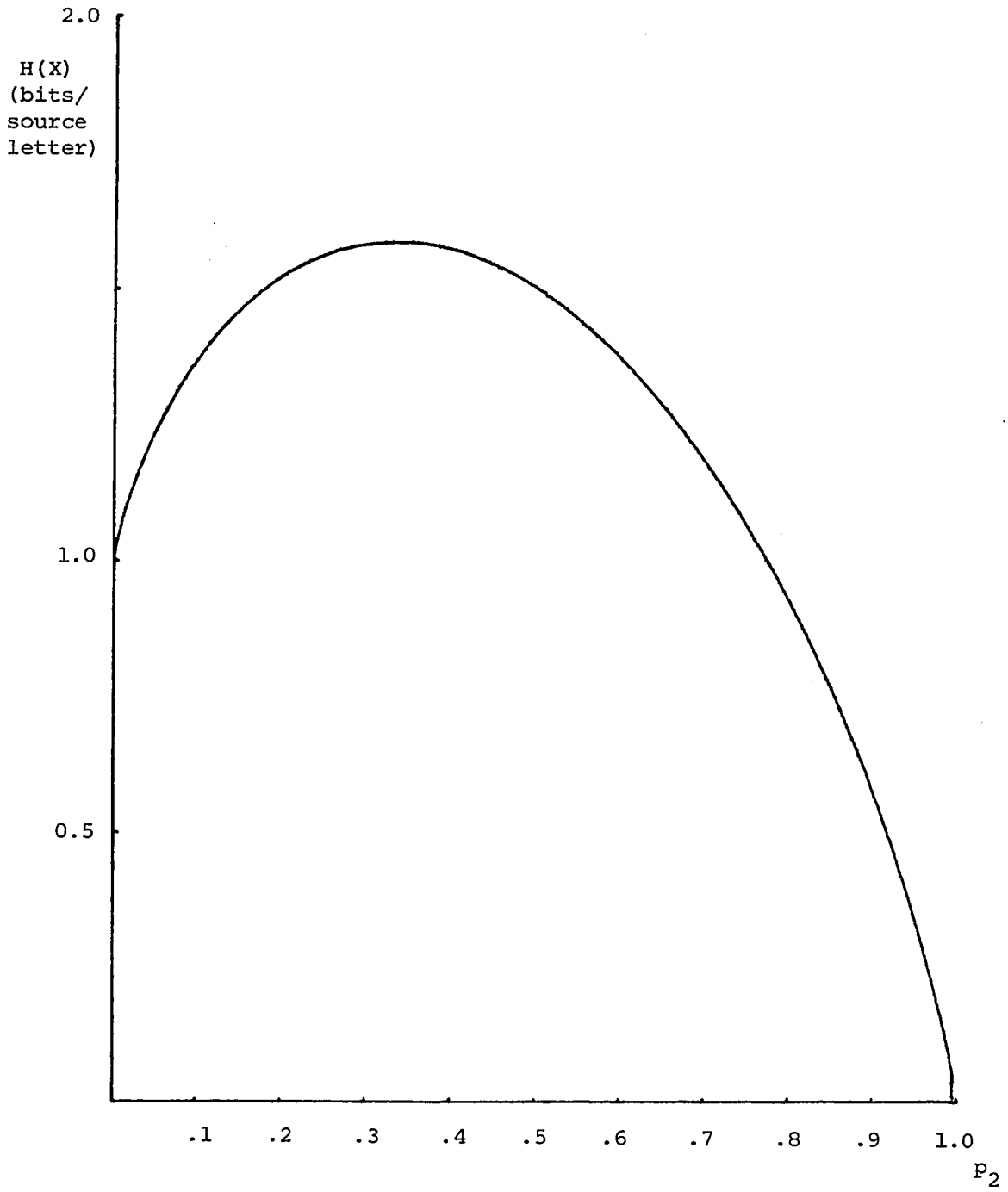

Figure 13:- Source entropy as a function of p_2 for a 3-letter source with $p_1 = p_3 = \frac{1 - p_2}{2}$.

$$q_1 = q_3 \geq 0,$$

$$p_2 \leq \frac{1 - \beta}{1 + \beta}$$

Similarly, since $q_2 \geq 0$,

$$p_2 \geq \beta\left(\frac{1 - \beta}{1 + \beta}\right)$$

Equivalently, these two expressions can be written as

$$p_2 \leq \frac{1 - \beta}{1 + \beta}$$

$$p_2 \geq \frac{\beta(1 - \beta)}{1 + \beta}$$

or

$$\beta \leq \frac{1 - p_2}{1 + p_2}$$

$$\frac{\beta(1 - \beta)}{1 + \beta} \leq p_2$$

Let $$\frac{1 - p_2}{1 + p_2} \leq p_2$$

which results in

$$1 - p_2 \leq p_2 + {p_2}^2$$

which can be written as

$${p_2}^2 + 2p_2 - 1 \geq 0$$

or

$$\left(p_2 + \frac{2 + \sqrt{2^2 + 4}}{2}\right)\left(p_2 + \frac{2 - \sqrt{2^2 + 4}}{2}\right) \geq 0$$

or

$$(p_2 + 1 + \sqrt{2})\left(p_2 + \frac{2 - \sqrt{2}}{2}\right) \geq 0$$

or, equivalently,

$$(p_2 + 2.414)(p_2 - 0.414) \geq 0$$

which yields

$$p_2 \geq 0.414$$

Then,

$$\frac{\beta(1 - \beta)}{1 + \beta} \leq \beta \leq \frac{1 - p_2}{1 + p_2} \leq p_2$$

Thus, $p_2 \geq \sqrt{2} - 1 \simeq 0.414$ is a sufficient condition for q_1, q_2 and q_3 to be all non-negative. For a given $p_2 \geq 0.414$, $\beta \leq \frac{1 - p_2}{1 + p_2}$. As a verification, if $p_2 \geq 0.414$,

$$D \leq \left(\frac{1 - p_2}{1 + p_2}\right) p_2 \left[\frac{1}{1 - \frac{(1 - p_2)}{1 + p_2}}\right] + \frac{(1 - p_2)}{1 + p_2} \left[\frac{1}{1 + \frac{(1 - p_2)}{1 + p_2}}\right]$$

$$= \left(\frac{1 - p_2}{1 + p_2}\right) p_2 \frac{(1 + p_2)}{2p_2} + \left(\frac{1 - p_2}{1 + p_2}\right)\left(\frac{1 + p_2}{2}\right)$$

$$= \frac{1 - p_2}{2} + \frac{1 - p_2}{2} = 1 - p_2 = D_{max}.$$

If $\beta = \frac{1 - p_2}{1 + p_2}$ and $D = 1 - p_2$, then

$$R(D_{max}) = (1 - p_2)\log_e\left(\frac{1 - p_2}{1 + p_2}\right)$$

$$+ \log\left[\frac{2}{(1 - p_2)\left(1 + \frac{1 - p_2}{1 + p_2}\right)}\right]$$

$$+ p_2 \log\left[\frac{\left(1 - \frac{1 - p_2}{1 + p_2}\right)(1 - p_2)}{2p_2}\right]$$

$$= (1 - p_2)\log_e\left(\frac{1 - p_2}{1 + p_2}\right) + \log\left(\frac{1 + p_2}{1 - p_2}\right) + p_2\log\left(\frac{1 - p_2}{1 + p_2}\right)$$

$$= (1 - p_2)\log_e\left(\frac{1 - p_2}{1 + p_2}\right) + (p_2 - 1)\log\left(\frac{1 - p_2}{1 + p_2}\right)$$

$$= 0$$

This result is expected, since R(D) should vanish at $D = D_{max}$.

It has been shown that the unconstrained solution is valid for $p_2 \geq \sqrt{2} - 1$. A condition will now be derived which ensures that $\beta \leq p_2$.

Since

$$\beta \leq \frac{1 - p_2}{1 + p_2}$$

and

$$\frac{\beta(1 - \beta)}{1 + \beta} \leq p_2$$

then if p_2 is sufficiently small, and if

$$\frac{\beta(1 - \beta)}{1 + \beta} \leq p_2,$$

then

$$\beta \leq \frac{1 - p_2}{1 + p_2}$$

since $\frac{1 - p_2}{1 + p_2}$ is very nearly unity.

Since $\frac{\beta(1-\beta)}{1+\beta} \leq p_2$

$$\beta^2 - \beta(1 - p_2) + p_2 \geq 0$$

or

$$\beta \leq \frac{(1 - p_2)}{2} \left[1 - \sqrt{1 - \frac{4p_2}{(1-p_2)^2}}\right]$$

Now, if

$$\frac{1 - p_2}{2} \left[1 - \sqrt{1 - \frac{4p_2}{(1-p_2)^2}}\right] \leq p_2$$

then

$$1 - \sqrt{1 - \frac{4p_2}{(1-p_2)^2}} \leq \frac{2p_2}{1 - p_2}$$

is a sufficient condition for $\beta \leq p_2$.

Hence, the solution described by equations (18) and (19), for $p_2 \geq \sqrt{2} - 1$, is the rate distortion function of the source. For $p_2 < \sqrt{2} - 1$, this solution will only be valid for a certain portion of the R(D) curve, up to a distortion D*. For $D > D^*$, a different expression for R(D) will generally be obtained, and it is not always possible to obtain an analytic solution.

It can also be pointed out that the relevance of this rate distortion function is dependent on the relevance of the distortion measure ρ. If it correctly reflects source behaviour, then it does meaningfully describe the rate-distortion trade-off for the source.

CHAPTER V

CONCLUDING REMARKS

In this thesis, the use of negacyclic codes in various source encoding schemes has been examined.

The desirability of digital communications links was first established in the introduction. However, since digital modulation schemes usually require more bandwidth than equivalent analog schemes, a modulation technique requiring a minimum bandwidth, such as phase shift keying (PSK), is usually used. In the same vein, a (source) coding scheme tailored to the output of a class of sources is used to achieve data (or bandwidth) compression. This reduces the number of bits to be transmitted (or stored) as compared to the uncoded source output. The source coded data is, however, more vulnerable to channel errors, since it contains less redundancy. Hence, in general, source coding will be followed by channel coding to reduce vulnerability to channel (or storage medium) errors.

The applicability of negacyclic codes to the source encoding of certain classes of sources, especially those with nonbinary outputs, was next demonstrated by achieving two major objectives. The first consisted in extending some theorems relevant to binary codes to negacyclic codes.

These results were used in deriving expressions for the average distortion of source coding schemes using negacyclic codes. Secondly, a scheme providing both data compression and an error-detection capability was introduced. This could be accomplished in a distortionless manner for sources with appropriate statistics. For others, distortion was allowed in order to achieve data compression. An orthogonal extension of this joint coding scheme was only briefly introduced, since its efficiency is inherently lower than that of the original scheme.

Finally, an example of the derivation of a rate distortion function for a 3-symbol negacyclic code was given. The applicability of the rate distortion function is, in this case as in all others, almost wholly a matter of the relevance of the distortion measure [63]. If the distortion measure is appropriate, then an optimal approximation of the source will result, when the rate of the source exceeds that of the channel. Its main application is thus to serve as a yardstick for the evaluation of source coding schemes.

Some general remarks relevant to negacyclic codes can be made at this point. The first is that, since the number of code symbols for negacyclic codes is always an odd prime, they cannot be represented efficiently in binary digits. Multi-level logic, such as that implemented using I^2L technology, would be more efficient at storing and/or processing negacyclic code symbols. The inefficiency

caused by using binary logic circuits can clearly be minimized by choosing an odd prime very close to a power of 2. Multi-level logic, on the other hand, is more efficient in terms of LSI circuit conductor utilization, packing density, and power utilization.

A second point is that negacyclic codes have an advantage over binary codes in having a Lee distance which is twice the Hamming distance guaranteed by the BCH bound [62].

Further investigation of some aspects of this work might yield interesting results. The first would involve comparing the results obtained for negacyclic codes to those for non-binary cyclic codes. This would be more relevant than comparisons with binary cyclic codes. Next, a more detailed study of the joint source and channel encoding scheme might yield an increase in its efficiency. The large number of 0's in the decomposition matrix indicated that this might be possible.

Another interesting aspect would involve comparison of the cost and complexity of implementing a negacyclic code codec with that for a non-binary cyclic code. This could be implemented in either binary or multi-level logic, or software-wise. Taking into consideration their respective source encoding capabilities, this would provide a realistic yardstick for comparing the complexity-data compression trade-offs of negacyclic codes with respect to non-binary cyclic codes.

Finally, in an effort to determine their relevance, actual non-binary source statistics should be compared to the distributions assumed in the dissertation.

REFERENCES

[1] Lin, S., 1970, "An Introduction to Error-Correcting Codes," Prentice-Hall, Englewood Cliffs, N.J., 1.

[2] Cattermole, K. W., 1969, "Principles of Pulse Code Modulation," American Elsevier, N.Y., 27-29.

[3] Jayant, N. S., 1974, "Digital Coding of Speech Waveforms: PCM, DPCM and DM Quantizers," Proc. IEEE, 611-632.

[4] Goblick, T. J., Jr., 1962, "Coding for a Discrete Information Source with a Distortion Measure," Ph.D. dissertation, Dept. of Elec. Eng., M.I.T., Cambridge, Mass., 1-5.

[5] Berger, T., 1971, "Rate Distortion Theory," Prentice-Hall, Englewood Cliffs, N.J., 1-11.

[6] Duan, J. R., and Wintz, P. A., 1974, "Information-Preserving Coding for Multispectral Scanner Data," Technical Report TR-EE 74-15, School of Elec. Eng., Purdue University, 2.

[7] Shannon, C. E., 1948, "A Mathematical Theory of Communication," Bell System Technical Journal, Vol. 27, 379-423, 623-656.

[8] Peterson, W. W., and Weldon, E. J., Jr., 1971, "Error-Correcting Codes," 2nd edition, M.I.T. Press, Cambridge, Mass., p. 5.

[9] Shannon, C. E., 1959, "Coding Theorems for a Discrete Source with a Fidelity Criterion," IRE Nat'l. Conv. Rec., part 4, 142-163.

[10] Ash, R., 1965, "Information Theory," Wiley-Interscience, 5-12.

[11] Nyquist, H., 1924, "Certain Factors Affecting Telegraph Speed," Bell System Technical Journal, Vol. 3, 324-346.

[12] Nyquist, H., 1928, "Certain Topics in Telegraph Transmission Theory," Trans. AIEE, Vol. 46, 617-644.

[13] Cattermole, 1969, ch. 3.

[14] Max, J., 1960, "Quantizing for Minimum Distortion," IRE Trans. Inform. Theory, Vol. IT-6, 7-12.

[15] O'Neal, J. B., Jr., 1971, "Entropy Coding in Speech and Television Differential PCM Systems," IEEE Trans. Inform. Theory, Vol. IT-17, Corr., 758-760.

[16] Berger, 1971, p. 20.

[17] Hyvärinen, L. P., 1970, "Information Theory for Systems Engineers," Springer-Verlag, Heidelberg, 18-20.

[18] Dion, J-P., and Shiva, S. G. S., 1977, "A Joint Source and Channel Encoding Scheme," Int. J. Electronics, accepted for publication.

[19] Dion, J-P., and Shiva, S. G. S., 1977, "Some Results on Negacyclic Codes," Int. J. Electronics, accepted for publication.

[20] Turner, L. F., 1976, "Application of Data Compression to an Experimental 9.6 kbits/s Adaptive p.c.m. Digital Speech System," Proc. IEE, Vol. 123, no. 2, 109-114.

[21] Berger, 1971, p. 199.

[22] Hellman, M. E., 1974, "Joint Source and Channel Encoding," Proc. of the 7th Hawaii Int'l. Conference on Systems Science, 46-48.

[23] Tavares, S., 1972, "Redundancy Removal in Binary Sources," Interim Report prepared for the Dept. of Communications, Communications Research Center, Ottawa, Canada.

[24] Viterbi, A. J., 1973, "Information Theory in the Sixties," IEEE Trans. Inform. Theory, Vol., IT-19, no. 3, 257-262.

[25] Lin, 1970, pp. 47-54.

[26] Peterson and Weldon, 1972, pp. 52-56.

[27] Peterson and Weldon, 1972, p. 89.

[28] Ibid., pp. 117-122.

[29] Ash, 1965, pp. 27-40.

[30] Fano, R. M., 1961, "Transmission of Information," M.I.T. Press and Wiley, N.Y., 62-66.

[31] Huffman, D. A., 1952, "A Method for the Construction of Minimum-Redundancy Codes," Proc. IRE, 1098-1101.

[32] Shaft, P. D., 1974, "A Source Encoding Algorithm for Quantized Data," IEEE Trans. on Comm., 867-869.

[33] Bradley, S. D., 1969, "Optimizing a Scheme for Run Length Encoding," Proc. IEEE, Corres., Jan. 1969, 108-109.

[34] Allard, P. E., and Bridgewater, A. W., 1972, "A Source Encoding Technique Using Algebraic Codes," Canadian Computer Conference, Montréal, Proceedings, 423-201 to 423-213.

[35] Jelinek, F., 1969, "Tree Encoding of Memoryless Time-Discrete Sources with a Fidelity Criterion," IEEE Trans. Inform. Theory, Vol. IT-15, 584-590.

[36] Gray, R. M., 1975, "Sliding-Block Source Coding," IEEE Trans. Inform. Theory, Vol. IT-21, 357-368.

[37] Wilkins, L. C., and Wintz, P. A., 1971, "Bibliography on Data Compression, Picture Properties, and Picture Coding," IEEE Trans. Inform. Theory, Vol. IT-17, no. 2, 180-197.

[38] Peterson and Weldon, 1972, p. 206.

[39] Lin, 1970, p. 58.

[40] Van Chinh, C., 1974, "A Study of Negacyclic Codes," M.A. Sc. Thesis, Univ. of Ottawa, Ottawa, Canada, 1.

[41] Berlekamp, E. R., 1968, "Algebraic Coding Theory," McGraw-Hill, N.Y., 200-217.

[42] Golomb, S., 1969, "A General Formulation of Error Metrics," IEEE Trans. Inform. Theory, Corr., May 1969, 425-426.

[43] Van Chinh, 1974, pp. 25-26.

[44] Ibid., Appendix A.

[45] Ibid., pp. 22-23.

[46] Ibid., pp. 34-37.

[47] Ibid., pp. 41-46.

[48] Berger, 1971, pp. 204-205.

[49] Lin, 1970, pp. 151-152.

[50] Berlekamp, 1968, pp. 305-316.

[51] Ibid., p. 422.

[52] Ibid., p. 303.

[53] Peterson and Weldon, 1972, pp. 121-123.

[54] Berger, 1971, pp. 6-8.

[55] Rosenfeld, A. and Kak, A. C., 1976, "Digital Picture Processing," New York, Academic Press, 140-148.

[56] Berger, 1971, pp. 13-55.

[57] Arimoto, S., 1972, "An Algorithm for Computing the Capacity of Arbitrary Discrete Memoryless Channels," IEEE Trans. Inform. Theory, Vol. IT-18, 14-20.

[58] Blahut, R. E., 1972, "Computation of Channel Capacity and Rate-Distortion Functions," IEEE Trans. Inform. Theory, Vol. IT-18, 460-473.

[59] Tan. H. and Yao, K., 1975, "Rate Distortion Functions of Countably Infinite Alphabet Memoryless Sources," Information and Control, Vol. 27, 272-288.

[60] Jelinek, F., 1967, "Evaluation of Rate Distortion Functions for Low Distortions," Proc. IEEE, Vol. 55, 2067-2068.

[61] Berger, 1971, p. 61.

[62] Peterson and Weldon, 1972, pp. 307-308.

[63] Tretiak, O. J., 1974, Review of "Rate Distortion Theory: A Mathematical Basis for Data Compression," by T. Berger. Inform. and Control, Vol. 24, 92-95.

BIBLIOGRAPHY

Allard, P. E., and Bridgewater, A. W., 1972. "A Source Encoding Technique Using Algebraic Codes." Canadian Computer Conference, Montreal, Proceedings.

Arimoto, S., 1972. "An Algorithm for Computing the Capacity of Arbitrary Discrete Memoryless Channels." IEEE Trans. Inform. Theory, Vol. IT-18.

Ash, R., 1965. "Information Theory." Wiley-Interscience.

Berger, T., 1971. "Rate Distortion Theory." Englewood Cliffs, N.J.: Prentice Hall.

Berlekamp, E. R., 1968. "Algebraic Coding Theory." N.Y.: McGraw-Hill.

Blahut, R. E., 1972. "Computation of Channel Capacity and Rate-Distortion Functions." IEEE Trans. Inform Theory, Vol. IT-18.

Bradley, S. D., 1969. "Optimizing a Scheme for Run Length Encoding." Proc. IEEE, Corres., Jan. 1969.

Cattermole, K. W., 1969. "Principles of Pulse Code Modulation." N.Y.: American Elsevier.

Dion, J.-P., and Shiva, S. G. S., 1977. "A Joint Source and Channel Encoding Scheme." Int. J. Electronics, accepted for publication.

Dion, J.-P., and Shiva, S. G. S., 1977. "Some Results on Negacyclic Codes." Int. J. Electronics, accepted for publication.

Duan, J. R., and Wintz, P. A., 1974. "Information-Preserving Coding for Multispectral Scanner Data." Technical Report TR-EE 74-15, School of Elect. Eng., Purdue University.

Fano, R. A., 1961. "Transmission of Information." N.Y.: M.I.T. Press and Wiley.

Goblick, T. J., Jr., 1962. "Coding for a Discrete Information Source with a Distortion Measure." Ph.D. dissertation, Dept. of Elec. Eng., M.I.T., Cambridge, Mass.

Golomb, S., 1969. "A General Formulation of Error Metrics." IEEE Trans. Inform. Theory, Corr., May 1969.

Gray, R. M., 1975. "Sliding-Block Source Coding." IEEE Trans. Inform. Theory, Vol. IT-21.

Hellman, M. E., 1974. "Joint Source and Channel Encoding." Proc. of the 7th Hawaii Int'l. Conference on Systems Science.

Huffman, D. A., 1952. "A Method for the Construction of Minimum-Redundancy Codes." Proc. IRE.

Hyvärinen, L. P., 1970. "Information Theory for Systems Engineers." Heidelberg: Springer-Verlag.

Jayant, N. S., 1974. "Digital Coding of Speech Waveforms: PCM, DPCM and DM Quantizers." Proc. IEEE, Vol. 62.

Jelinek, F., 1967. "Evaluation of Rate Distortion Functions for Low Distortions." Proc. IEEE, Vol. 55.

Jelinek, F., 1969. "Tree Encoding of Memoryless Time-Discrete Sources with a Fidelity Criterion." IEEE Trans. Inform. Theory, Vol. IT-15.

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

Max, J., 1960. "Quantizing for Minimum Distortion." IRE Trans. Inform. Theory, Vol. IT-6.

Nyquist, H., 1924. "Certain Factors Affecting Telegraph Speed." Bell System Technical Journal, Vol. 3.

Nyquist, H., 1928. "Certain Topics in Telegraph Transmission Theory." Trans. AIEE, Vol. 46.

O'Neal, J. B., Jr., 1971. "Entropy Coding in Speech and Television Differential PCM Systems." IEEE Trans. Inform. Theory, Vol. IT-17.

Peterson, W. W., and Weldon, E. J., Jr., 1972. "Error-Correcting Codes." Cambridge, Mass.: M.I.T. Press, 2nd edition.

Rosenfeld, A., and Kak, A. C., 1976. "Digital Picture Processing." N.Y.: Academic Press.

Shaft, P. D., 1974. "A Source Encoding Algorithm for Quantized Data." IEEE Trans. on Comm.

Shannon, C. E., 1948. "A Mathematical Theory of Communication." Bell System Technical Journal, Vol. 27.

Shannon, C. E., 1959. "Coding Theorems for a Discrete Source with a Fidelity Criterion." IRE Nat'l. Conv. Rec., part 4.

Tan, H., and Yao, K., 1975. "Rate Distortion Functions of Countably Infinite Alphabet Memoryless Sources." Information and Control, Vol. 27.

Tavares, S., 1972. "Redundancy Removal in Binary Sources." Interim Report prepared for the Dept. of Communications, Communications Research Center, Ottawa, Canada.

Tretiak, O. J., 1974. Review of "Rate Distortion Theory: A Mathematical Basis for Data Compression," by T. Berger. Inform. and Control, Vol. 24.

Turner, L. F., 1976. "Application of Data Compression to an Experimental 9.6 kbits/s Adaptive p.c.m. Digital Speech System." Proc. IEE, Vol. 123.

Van Chinh, C., 1974. "A Study of Negacyclic Codes." M.A. Sc. Thesis, University of Ottawa, Ottawa, Canada.

Viterbi, A. J., 1973. "Information Theory in the Sixties." IEEE Trans. Inform. Theory, Vol. IT-19, no. 3.

Wilkins, L. C., and Wintz, P. A., 1972. "Bibliography on Data Compression, Picture Properties, and Picture Coding." IEEE Trans. Inform. Theory, Vol. IT-17, no. 2.

VITA

Name: Jean-Pierre Dion

Born: Ottawa, Canada; June 15, 1948.

Education:

Secondary: University of Ottawa High School
Ottawa

University: Department of Electrical Engineering
University of Ottawa
B.A. Sc. (E.E.), 1971.

Department of Electrical Engineering
University of Ottawa
M.A. Sc. (E.E.), 1974.