

GALLAGER

INFORMATION  
THEORY  
and  
RELIABLE  
COMMUNICATION

---

INFORMATION THEORY and  
RELIABLE COMMUNICATION

Q  
360  
.G3  
C.1

WILEY

ROBERT G. GALLAGER

# INFORMATION THEORY AND RELIABLE COMMUNICATION

*Robert G. Gallager*

Massachusetts Institute of Technology

**JOHN WILEY & SONS**

New York • Chichester • Brisbane • Toronto • Singapore

30 29 28 27 26 25 24 23 22

Copyright © 1968 by John Wiley & Sons, Inc.

All rights reserved.

Reproduction or translation of any part of this work beyond that permitted by Sections 107 or 108 of the 1976 United States Copyright Act without the permission of the copyright owner is unlawful. Requests for permission or further information should be addressed to the Permissions Department, John Wiley & Sons, Inc.

ISBN W-471-29048-3

Library of Congress Catalog Card Number: 68-26850  
Printed in the United States of America

## PREFACE

This book is designed primarily for use as a first-year graduate text in information theory, suitable for both engineers and mathematicians. It is assumed that the reader has some understanding of freshman calculus and elementary probability, and in the later chapters some introductory random process theory. Unfortunately there is one more requirement that is harder to meet. The reader must have a reasonable level of mathematical maturity and capability for abstract thought. The major results of the theory are quite subtle and abstract and must sometimes be arrived at by what appears to be rather devious routes. Fortunately, recent simplifications in the theory have made the major results more accessible than in the past.

Because of the subtlety and abstractness of the subject, it is necessary to be more rigorous than is usual in engineering. I have attempted to soften this wherever possible by preceding the proof of difficult theorems both with some explanation of why the theorem is important and with an intuitive explanation of why it is true. An attempt has also been made to provide the simplest and most elementary proof of each theorem, and many of the proofs here are new. I have carefully avoided the rather obnoxious practice in many elementary textbooks of quoting obscure mathematical theorems in the middle of a proof to make it come out right.

There are a number of reasons for the stress on proving theorems here. One of the main reasons is that the engineer who attempts to apply the theory will rapidly find that engineering problems are rarely solved by applying theorems to them. The theorems seldom apply exactly and one must understand the proofs to see whether the theorems provide any insight into the problem. Another reason for the stress on proofs is that the techniques used in the proofs are often more useful in doing new research in the area than the theorems themselves. A final reason for emphasizing the precise statement of results and careful proofs is that the text has been designed as an integral part of a course in information theory rather than as the whole course. Philosophy, intuitive understanding, examples, and applications, for example, are better developed in the give and take of a classroom, whereas precise statements and details are better presented in the permanent record

of a textbook. Enough of the intuition has been presented here for the instructor and the independent student, but added classroom stress is needed for the beginning graduate student.

A large number of exercises and problems are given at the end of the text. These range from simple numerical examples to significant generalizations of the theory. There are relatively few examples worked out in the text, and the student who needs examples should pause frequently in his reading to work out some of the simpler exercises at the end of the book.

There are a number of ways to organize the material here into a one-semester course. Chapter 1 should always be read first (and probably last also). After this, my own preference is to cover the following sections in order: 2.1–2.4, 3.1–3.4, 4.1–4.5, 5.1–5.6, 6.1–6.5, and finally either 6.8–6.9 or 6.6–6.7 or 8.1–8.3. Another possibility, for students who have some background in random processes, is to start with Sections 8.1 and 8.2 and then to proceed with the previous outline, using the white Gaussian noise channel as an example throughout. Another possibility, for students with a strong practical motivation, is to start with Chapter 6 (omitting Section 6.2), then to cover Sections 5.1 to 5.5, then 6.2, then Chapters 2 and 4 and Sections 8.1 and 8.2. Other possible course outlines can be made up with the help of the following table of prerequisites.

Table of Prerequisites

| Sections | Prerequisites | Sections | Prerequisites         |
|----------|---------------|----------|-----------------------|
| 2.1–2.3  | None          | 6.2      | 5.6, 6.1              |
| 2.4–2.5  | 2.1–2.3       | 6.3–6.7  | 6.1                   |
| 3.1      | 2.1–2.3       | 6.8      | 6.1                   |
| 3.2–3.4  | 2.1–2.3       | 6.9      | 5.1–5.5, 6.1–6.5, 6.8 |
| 3.5–3.6  | 3.1–3.4       | 6.10     | 6.1–6.5, 6.8          |
| 4.1–4.5  | 2.1–2.3       | 7.1–7.5  | 2.1–2.5, 4.3, 5.6–5.7 |
| 4.6      | 3.6, 4.1–4.5  | 8.1–8.2  | None                  |
| 5.1–5.5  | None          | 8.3–8.6  | 7.1–7.5, 8.1–8.2      |
| 5.6–5.8  | 4.4, 5.1–5.5  | 9.1–9.6  | 4.1–4.5               |
| 5.9      | 4.6, 5.1–5.6  | 9.7      | 8.5, 9.1–9.5          |
| 6.1      | None          | 9.8      | 3.5, 9.1–9.5          |

As a general rule, the latter topics in each chapter are more difficult and are presented in a more terse manner than the earlier topics. They are included primarily for the benefit of advanced students and workers in the field, although most of them can be covered in a second semester. Instructors are

cautioned not to spend too much time on Chapter 3, particularly in a one-semester course. The material in Sections 4.1–4.5, 5.1–5.6, and 6.1–6.5 is simpler and far more significant than that of Sections 3.5–3.6, even though it may be less familiar to some instructors.

I apologize to the many authors of significant papers in information theory whom I neglected to cite. I tried to list the references that I found useful in the preparation of this book along with references for selected advanced material. Many papers of historical significance were neglected, and the authors cited are not necessarily the ones who have made the greatest contributions to the field.

Robert G. Gallager



## **ACKNOWLEDGMENTS**

I am grateful for the patient support of the Research Laboratory of Electronics and of the Electrical Engineering Department at MIT while this book was being written. The work at the Research Laboratory of Electronics was supported by the National Aeronautics and Space Administration under Grant NSG-334. I am particularly grateful to R. M. Fano who stimulated my early interest in information theory and to whom I owe much of my conceptual understanding of the subject. This text was started over four years ago with the original idea of making it a revision, under joint authorship, of *The Transmission of Information* by R. M. Fano. As the years passed and the text grew and changed, it became obvious that it was a totally different book. However, my debt to *The Transmission of Information* is obvious to anyone familiar with both books.

I am also very grateful to P. Elias, J. M. Wozencraft, and C. E. Shannon for their ideas and teachings, which I have used liberally here. Another debt is owed to the many students who have taken the information theory course at MIT and who have made candid comments about the many experiments in different ways of presenting the material here. Finally I am indebted to the many colleagues who have been very generous in providing detailed criticisms of different parts of the manuscript. J. L. Massey has been particularly helpful in this respect. Also, G. D. Forney, H. Yudkin, A. Wyner, P. Elias, R. Kahn, R. S. Kennedy, J. Max, J. Pinkston, E. Berlekamp, A. Kohlenberg, I. Jacobs, D. Sakrison, T. Kailath, L. Seidman, and F. Preparata have all made a number of criticisms that significantly improved the manuscript.

R. G. G.



## CONTENTS

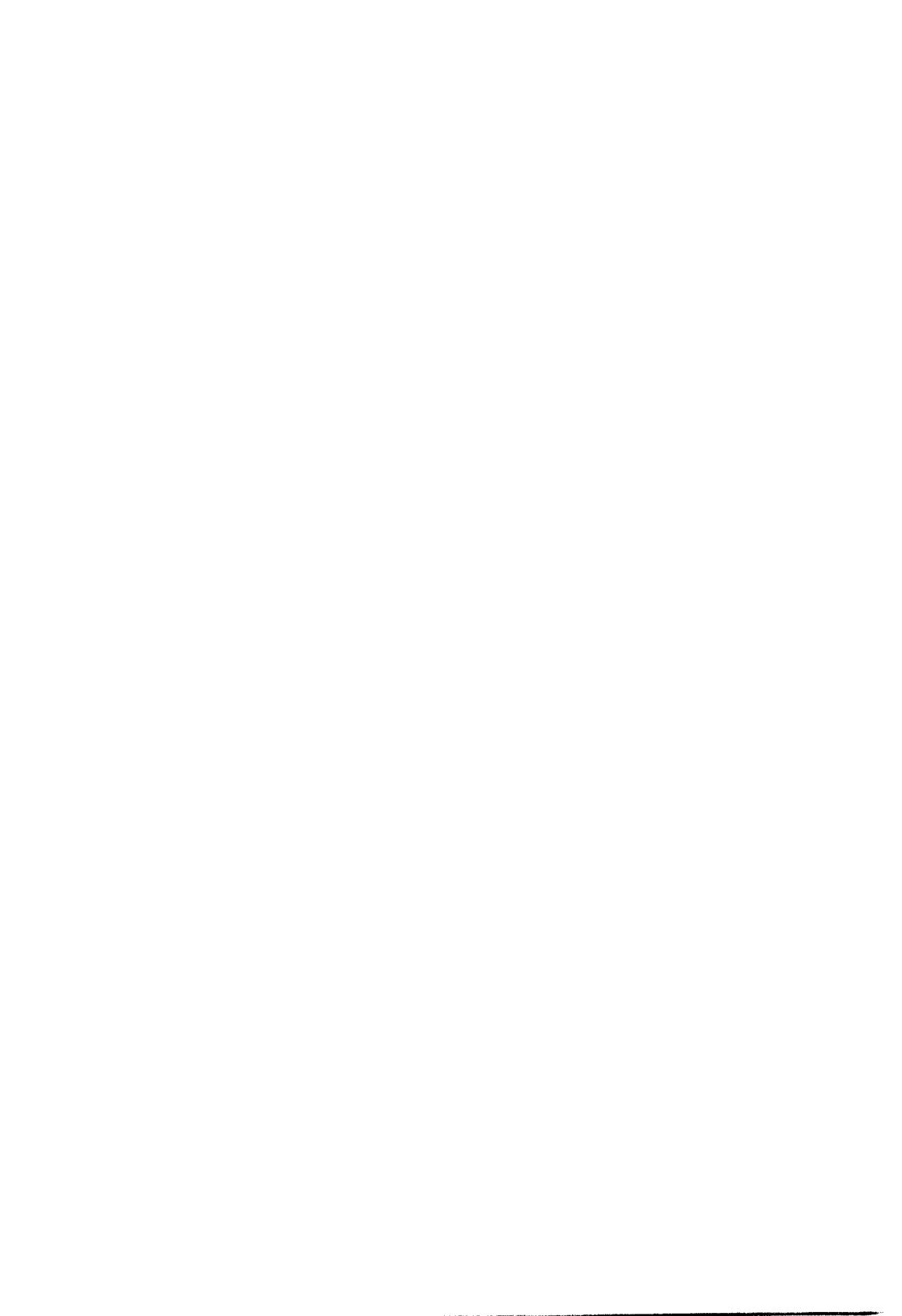
|   |           |
|---|-----------|
| <b>1 Communication Systems and Information Theory</b>           | <b>1</b>  |
| 1.1 Introduction  | 1         |
| 1.2 Source Models and Source Coding                             | 4         |
| 1.3 Channel Models and Channel Coding                           | 6         |
| Historical Notes and References                                 | 12        |
| <b>2 A Measure of Information</b>                               | <b>13</b> |
| 2.1 Discrete Probability: Review and Notation                   | 13        |
| 2.2 Definition of Mutual Information                            | 16        |
| 2.3 Average Mutual Information and Entropy                      | 23        |
| 2.4 Probability and Mutual Information for Continuous Ensembles | 27        |
| 2.5 Mutual Information for Arbitrary Ensembles                  | 33        |
| Summary and Conclusions   | 37        |
| Historical Notes and References                                 | 37        |
| <b>3 Coding for Discrete Sources</b>                            | <b>38</b> |
| 3.1 Fixed-Length Codes  | 39        |
| 3.2 Variable-Length Code Words                                  | 43        |
| 3.3 A Source Coding Theorem                                     | 50        |
| 3.4 An Optimum Variable-Length Encoding Procedure               | 52        |
| 3.5 Discrete Stationary Sources                                 | 56        |
| 3.6 Markov Sources  | 63        |
| Summary and Conclusions   | 69        |
| Historical Notes and References                                 | 70        |
| <b>4 Discrete Memoryless Channels and Capacity</b>              | <b>71</b> |
| 4.1 Classification of Channels                                  | 71        |
| 4.2 Discrete Memoryless Channels                                | 73        |

|          |   |            |
|----------|---|------------|
| 4.3      | The Converse to the Coding Theorem                          | 76         |
| 4.4      | Convex Functions  | 82         |
| 4.5      | Finding Channel Capacity for a Discrete Memoryless Channel  | 91         |
| 4.6      | Discrete Channels with Memory                               | 97         |
|          | Indecomposable Channels                                     | 105        |
|          | Summary and Conclusions                                     | 111        |
|          | Historical Notes and References                             | 111        |
|          | Appendix 4A   | 112        |
| <b>5</b> | <b>The Noisy-Channel Coding Theorem</b>                     | <b>116</b> |
| 5.1      | Block Codes   | 116        |
| 5.2      | Decoding Block Codes  | 120        |
| 5.3      | Error Probability for Two Code Words                        | 122        |
| 5.4      | The Generalized Chebyshev Inequality and the Chernoff Bound | 126        |
| 5.5      | Randomly Chosen Code Words                                  | 131        |
| 5.6      | Many Code Words—The Coding Theorem                          | 135        |
|          | Properties of the Random Coding Exponent                    | 141        |
| 5.7      | Error Probability for an Expurgated Ensemble of Codes       | 150        |
| 5.8      | Lower Bounds to Error Probability                           | 157        |
|          | Block Error Probability at Rates above Capacity             | 173        |
| 5.9      | The Coding Theorem for Finite-State Channels                | 176        |
|          | State Known at Receiver                                     | 182        |
|          | Summary and Conclusions                                     | 187        |
|          | Historical Notes and References                             | 188        |
|          | Appendix 5A   | 188        |
|          | Appendix 5B   | 193        |
| <b>6</b> | <b>Techniques for Coding and Decoding</b>                   | <b>196</b> |
| 6.1      | Parity-Check Codes  | 196        |
|          | Generator Matrices  | 199        |
|          | Parity-Check Matrices for Systematic Parity-Check Codes     | 200        |
|          | Decoding Tables   | 202        |
|          | Hamming Codes   | 203        |
| 6.2      | The Coding Theorem for Parity-Check Codes                   | 206        |

|          |  |            |
|----------|--|------------|
| 6.3      | Group Theory   | 209        |
|          | Subgroups  | 210        |
|          | Cyclic Subgroups   | 211        |
| 6.4      | Fields and Polynomials                                     | 213        |
|          | Polynomials  | 214        |
| 6.5      | Cyclic Codes   | 219        |
| 6.6      | Galois Fields  | 225        |
|          | Maximal Length Codes and Hamming Codes                     | 230        |
|          | Existence of Galois Fields                                 | 235        |
| 6.7      | BCH Codes  | 238        |
|          | Iterative Algorithm for Finding $\sigma(D)$                | 245        |
| 6.8      | Convolutional Codes and Threshold Decoding                 | 258        |
| 6.9      | Sequential Decoding  | 263        |
|          | Computation for Sequential Decoding                        | 273        |
|          | Error Probability for Sequential Decoding                  | 280        |
| 6.10     | Coding for Burst Noise Channels                            | 286        |
|          | Cyclic Codes   | 291        |
|          | Convolutional Codes  | 297        |
|          | Summary and Conclusions                                    | 305        |
|          | Historical Notes and References                            | 305        |
|          | Appendix 6A  | 306        |
|          | Appendix 6B  | 309        |
| <b>7</b> | <b>Memoryless Channels with Discrete Time</b>              | <b>316</b> |
| 7.1      | Introduction   | 316        |
| 7.2      | Unconstrained Inputs                                       | 318        |
| 7.3      | Constrained Inputs   | 323        |
| 7.4      | Additive Noise and Additive Gaussian Noise                 | 333        |
|          | Additive Gaussian Noise with an Energy Constrained Input   | 335        |
| 7.5      | Parallel Additive Gaussian Noise Channels                  | 343        |
|          | Summary and Conclusions                                    | 354        |
|          | Historical Notes and References                            | 354        |
| <b>8</b> | <b>Waveform Channels</b>                                   | <b>355</b> |
| 8.1      | Orthonormal Expansions of Signals and White Gaussian Noise | 355        |
|          | Gaussian Random Processes                                  | 362        |
|          | Mutual Information for Continuous-Time Channels            | 369        |

|          |   |            |
|----------|---|------------|
| 8.2      | White Gaussian Noise and Orthogonal Signals   | 371        |
|          | Error Probability for Two Code Words  | 374        |
|          | Error Probability for Orthogonal Code Words   | 379        |
| 8.3      | Heuristic Treatment of Capacity for Channels with Additive Gaussian Noise and Bandwidth Constraints | 383        |
| 8.4      | Representation of Linear Filters and Nonwhite Noise   | 390        |
|          | Filtered Noise and the Karhunen-Loeve Expansion   | 398        |
|          | Low-Pass Ideal Filters  | 402        |
| 8.5      | Additive Gaussian Noise Channels with an Input Constrained in Power and Frequency                   | 407        |
| 8.6      | Fading Dispersive Channels  | 431        |
|          | Summary and Conclusions   | 439        |
|          | Historical Notes and References   | 440        |
| <b>9</b> | <b>Source Coding with a Fidelity Criterion</b>  | <b>442</b> |
| 9.1      | Introduction  | 442        |
| 9.2      | Discrete Memoryless Sources and Single-Letter Distortion Measures                                   | 443        |
| 9.3      | The Coding Theorem for Sources with a Fidelity Criterion  | 451        |
| 9.4      | Calculation of $R(d^*)$   | 457        |
| 9.5      | The Converse to the Noisy-Channel Coding Theorem Revisited  | 465        |
| 9.6      | Discrete-Time Sources with Continuous Amplitudes  | 470        |
| 9.7      | Gaussian Sources with Square Difference Distortion Gaussian Random-Process Sources                  | 475        |
| 9.8      | Discrete Ergodic Sources  | 490        |
|          | Summary and Conclusions   | 500        |
|          | Historical Notes and References   | 501        |
|          | Exercises and Problems  | 503        |
|          | References and Selected Reading   | 569        |
|          | Glossary of Symbols   | 578        |
|          | <b>Index</b>  | <b>581</b> |

*Information Theory and Reliable Communication*



## *Chapter 1*

### COMMUNICATION SYSTEMS AND INFORMATION THEORY

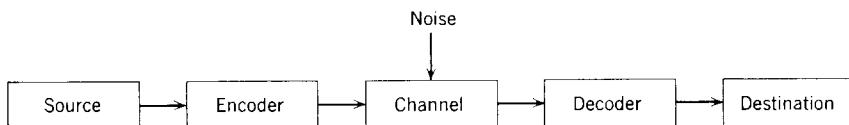
#### **1.1 Introduction**

Communication theory deals primarily with systems for transmitting information or data from one point to another. A rather general block diagram for visualizing the behavior of such systems is given in Figure 1.1.1. The source output in Figure 1.1.1 might represent, for example, a voice waveform, a sequence of binary digits from a magnetic tape, the output of a set of sensors in a space probe, a sensory input to a biological organism, or a target in a radar system. The channel might represent, for example, a telephone line, a high frequency radio link, a space communication link, a storage medium, or a biological organism (for the case where the source output is a sensory input to that organism). The channel is usually subject to various types of noise disturbances, which on a telephone line, for example, might take the form of a time-varying frequency response, crosstalk from other lines, thermal noise, and impulsive switching noise. The encoder in Figure 1.1.1 represents any processing of the source output performed prior to transmission. The processing might include, for example, any combination of modulation, data reduction, and insertion of redundancy to combat the channel noise. The decoder represents the processing of the channel output with the objective of producing at the destination an acceptable replica of (or response to) the source output.

In the early 1940's, C. E. Shannon (1948) developed a mathematical theory, called information theory, for dealing with the more fundamental aspects of communication systems. The distinguishing characteristics of this theory are, first, a great emphasis on probability theory and, second, a primary concern with the encoder and decoder, both in terms of their functional roles and in terms of the existence (or nonexistence) of encoders and decoders that achieve a given level of performance. In the past 20 years, information theory has been made more precise, has been extended, and has

been brought to the point where it is being applied in practical communication systems. Our purpose in this book is to present this theory, both bringing out its logical cohesion and indicating where and how it can be applied.

As in any mathematical theory, the theory deals only with mathematical models and not with physical sources and physical channels. One would think, therefore, that the appropriate way to begin the development of the theory would be with a discussion of how to construct appropriate mathematical models for physical sources and channels. This, however, is not the way that theories are constructed, primarily because physical reality is rarely simple enough to be precisely modeled by mathematically tractable



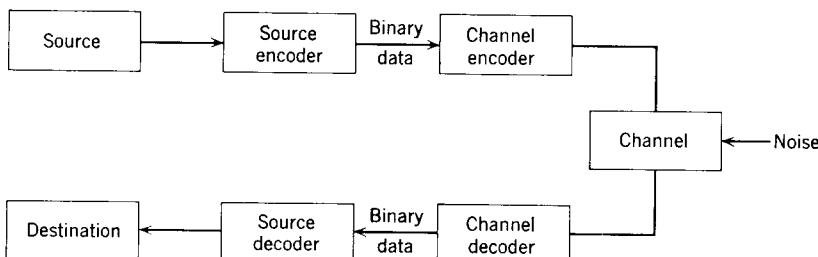
**Figure 1.1.1. Block diagram of communication system.**

models. Our procedure here will be rather to start by studying the simplest classes of mathematical models of sources and channels, using the insight and the results gained to study progressively more complicated classes of models. Naturally, the choice of classes of models to study will be influenced and motivated by the more important aspects of real sources and channels, but our view of what aspects are important will be modified by the theoretical results. Finally, after understanding the theory, we shall find it useful in the study of real communication systems in two ways. First, it will provide a framework within which to construct detailed models of real sources and channels. Second, and more important, the relationships established by the theory provide an indication of the types of tradeoffs that exist in constructing encoders and decoders for given systems. While the above comments apply to almost any mathematical theory, they are particularly necessary here because quite an extensive theory must be developed before the more important implications for the design of communication systems will become apparent.

In order to further simplify our study of source models and channel models, it is helpful to partly isolate the effect of the source in a communication system from that of the channel. This can be done by breaking the encoder and decoder of Figure 1.1.1 each into two parts as shown in Figure 1.1.2. The purpose of the source encoder is to represent the source output by a sequence of binary digits and one of the major questions of concern is to determine how many binary digits per unit time are required to represent the output of any given source model. The purpose of the channel encoder

and decoder is to allow the binary data sequences to be reliably reproduced at the output of the channel decoder, and one of the major questions of concern here is if and how this can be done.

It is not obvious, of course, whether restricting the encoder and decoder to the form of Figure 1.1.2 imposes any fundamental limitations on the performance of the communication system. One of the most important results of the theory, however, is that under very broad conditions no such limitations are imposed (this does not say, however, that an encoder and decoder of the form in Figure 1.1.2 is always the most economical way to achieve a given performance).



**Figure 1.1.2. Block diagram of communication system with encoder and decoder each split in two parts.**

From a practical standpoint, the splitting of encoder and decoder in Figure 1.1.2 is particularly convenient since it makes the design of the channel encoder and decoder virtually independent of the source encoder and decoder, using binary data as an interface. This, of course, facilitates the use of different sources on the same channel.

In the next two sections, we shall briefly describe the classes of source models and channel models to be studied in later chapters and the encoding and decoding of these sources and channels. Since the emphasis in information theory is primarily on this encoding and decoding, it should be clear that the theory is not equally applicable to all communication situations. For example, if the source is a radar target, there is no opportunity for encoding the source output (unless we want to look at the choice of radar signals as a form of encoding), and thus we cannot expect the theory to produce any more than peripheral insight. Similarly, if the source output is a sensory input to a biological organism, we might consider the organism to be a combination of encoding, channel, and decoding, but we have no control over the encoding and decoding, and it is not at all clear that this is the most fruitful model for such studies of a biological organism. Thus, again, information theory might provide some insight into the behavior of such organisms but it can certainly not be regarded as a magical key for understanding.

## 1.2 Source Models and Source Coding

We now briefly describe the mathematical models of sources that the theory will deal with. Naturally, these models will be presented more carefully in subsequent chapters. All source models in information theory are random process (or random sequence) models. Discrete memoryless sources constitute the simplest class of source models. These are sources for which the output is a sequence (in time) of letters, each letter being a selection from some fixed alphabet consisting of, say, the letters  $a_1, a_2, \dots, a_K$ . The letters in the source output sequence are random statistically independent selections from the alphabet, the selection being made according to some fixed probability assignment  $Q(a_1), \dots, Q(a_K)$ .

| Method 1             | Method 2              |
|----------------------|-----------------------|
| $a_1 \rightarrow 00$ | $a_1 \rightarrow 0$   |
| $a_2 \rightarrow 01$ | $a_2 \rightarrow 10$  |
| $a_3 \rightarrow 10$ | $a_3 \rightarrow 110$ |
| $a_4 \rightarrow 11$ | $a_4 \rightarrow 111$ |

*Figure 1.2.1. Two ways of converting a four-letter alphabet into binary digits.*

It undoubtedly seems rather peculiar at first to model physical sources, which presumably produce meaningful information, by a random process model. The following example will help to clarify the reason for this. Suppose that a measurement is to be performed repeatedly and that the result of each measurement is one of the four events  $a_1, a_2, a_3$ , or  $a_4$ . Suppose that this sequence of measurements is to be stored in binary form and suppose that two ways of performing the conversion to binary digits have been proposed, as indicated in Figure 1.2.1.

In the first method above, two binary digits are required to represent each source digit, whereas in the second method, a variable number is required. If it is known that  $a_1$  will be the result of the great majority of the measurements, then method 2 will allow a long sequence of measurements to be stored with many fewer binary digits than method 1. In Chapter 3, methods for encoding the output of a discrete source into binary data will be discussed in detail. The important point here is that the relative effectiveness of the two methods in Figure 1.2.1 depends critically upon the frequency of occurrence of the different events, and that this is incorporated into a mathematical model of the source by assigning probabilities to the set of source letters. More familiar, but more complicated, examples of the same type are given by shorthand, where short symbols are used for commonly occurring words, and in Morse code, where short sequences of dots and dashes are assigned to common letters and longer sequences to uncommon letters.

Closely related to the encoding of a source output into binary data is the measure of information (or uncertainty) of the letters of a source alphabet, which will be discussed in Chapter 2. If the  $k$ th letter of the source alphabet has probability  $Q(a_k)$ , then the self-information of that letter (measured in bits) is defined as  $I(a_k) \triangleq -\log_2 Q(a_k)$ . From an intuitive standpoint, as will be seen in more detail in Chapter 2, this technical definition has many of the same qualities as the nontechnical meaning of information. In particular, if  $Q(a_k) = 1$ , then  $I(a_k) = 0$ , corresponding to the fact that the occurrence of  $a_k$  is not at all informative since it had to occur. Similarly, the smaller the probability of  $a_k$ , the larger its self-information. On the other hand, it is not hard to see that this technical definition of information also lacks some qualities of the nontechnical meaning. For example, no matter how unlikely an event is, we do not consider it informative (in the non-technical sense) unless it happens to interest us. This does not mean that there is something inadequate about the definition of self-information; the usefulness of a definition in a theory comes from the insight that it provides and the theorems that it simplifies. The definition here turns out to be useful in the theory primarily because it does separate out the notion of unexpectedness in information from that of interest or meaning.

*The average value of self-information over the letters of the alphabet is a particularly important quantity known as the entropy of a source letter, and it is given by*

$$\sum_{k=1}^K -Q(a_k) \log_2 Q(a_k).$$

The major significance of the entropy of a source letter comes from the source coding theorem which is treated in Chapter 3. This states that, if  $H$  is the entropy of a source letter for a discrete memoryless source, then the sequence of source outputs cannot be represented by a binary sequence using fewer than  $H$  binary digits per source digit on the average, but it can be represented by a binary sequence using as close to  $H$  binary digits per source digit on the average as desired. Some feeling for this result may be obtained by noting that, if for some integer  $L$ , a source has an alphabet of  $2^L$  equally likely letters, then the entropy of a source letter is  $L$  bits. On the other hand, if we observe that there are  $2^L$  different sequences of  $L$  binary digits, then we see that each of these sequences can be assigned to a different letter of the source alphabet, thus representing the output of the source by  $L$  binary digits per source digit. This example also goes a long way towards showing why a logarithm appears in the definition of self-information and entropy.

The entropy of a source is also frequently given in the units of bits per second. If, for a discrete memoryless source, the entropy of a source letter

is  $H$ , and if the source produces one letter each  $\tau_s$  seconds, then the entropy in bits per second is just  $H/\tau_s$ , and the source coding theorem indicates that the source output can be represented by a binary sequence of arbitrarily close to  $H/\tau_s$  binary digits per second.

As a more complicated class of source models, we shall consider discrete sources with memory in which successive letters from the source are statistically dependent. In Section 3.5, the entropy for these sources (in bits per digit or bits per second) is defined in an analogous but more complicated way and the source coding theorem is shown to apply if the source is ergodic.

Finally, in Chapter 9, we shall consider nondiscrete sources. The most familiar example of a nondiscrete source is one where the source output is a random process. When we attempt to encode a random process into a binary sequence, the situation is conceptually very different from the encoding of a discrete source. A random process can be encoded into binary data, for example, by sampling the random waveform, then quantizing the samples, and then encoding the quantized samples into binary data. The difference between this and the binary encoding discussed previously is that the sample waveform cannot be precisely reconstructed from the binary sequence, and thus such an encoding must be evaluated both in terms of the number of binary digits required per second and some measure of the distortion between the source waveform and the waveform reconstructed from the binary digits. In Chapter 9 we shall treat the problem of finding the minimum number of binary digits per second required to encode a source output so that the average distortion between the source output and a replica constructed from the binary sequence is within a given level. The major point here is that a nondiscrete source can be encoded with distortion into a binary sequence and that the required number of binary digits per unit time depends on the permissible distortion.

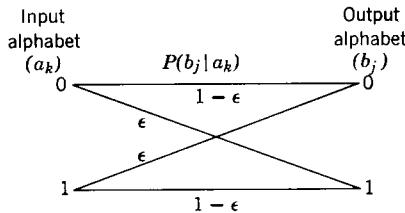
### **1.3 Channel Models and Channel Coding**

In order to specify a mathematical model for a channel, we shall specify first the set of possible inputs to the channel, second, the set of possible outputs, and third, for each input, a probability measure on the set of outputs. Discrete memoryless channels constitute the simplest class of channel models and are defined as follows: the input is a sequence of letters from a finite alphabet, say  $a_1, \dots, a_K$ , and the output is a sequence of letters from the same or a different alphabet, say  $b_1, \dots, b_J$ . Finally, each letter in the output sequence is statistically dependent only on the letter in the corresponding position of the input sequence and is determined by a fixed conditional probability assignment  $P(b_j | a_k)$  defined for each letter  $a_k$  in the input alphabet and each letter  $b_j$  in the output alphabet. For example, the binary symmetric channel (see Figure 1.3.1) is a discrete memoryless channel

with binary input and output sequences where each digit in the input sequence is reproduced correctly at the channel output with some fixed probability  $1 - \epsilon$  and is altered by noise into the opposite digit with probability  $\epsilon$ . In general, for discrete memoryless channels, the transition probability assignment tells us everything that we have to know about how the noise combines with the channel input to produce the channel output. We shall describe later how discrete memoryless channels relate to physical channels.

A much broader class of channels, which we shall call discrete channels with memory, is the class where the input and output are again sequences of letters from finite alphabets but where each letter in the output sequence can depend statistically on more than just the corresponding letter in the input sequence.

Another class of channel models which bears a more immediate resemblance to physical channels is the class where the set of inputs and set of outputs are each a set of time functions (that is, waveforms), and for each input



*Figure 1.3.1. Binary symmetric channel.*

waveform the output is a random process. A particular model in this class which is of great theoretical and practical importance (particularly in space communication) is the additive white Gaussian noise channel. The set of inputs for this model is the set of time functions with a given upper limit on power and the output is the sum of the input plus white Gaussian noise. When using this model for a physical channel with attenuation, we naturally take the input in the model to be the input to the physical channel as attenuated by the channel.

When transmitting binary data over a channel in the above class, it is often convenient to separate the channel encoder and channel decoder each into two parts, as shown in Figure 1.3.2. The output from the discrete channel encoder in Figure 1.3.2 is a sequence of letters from a finite alphabet, say  $a_1, \dots, a_K$ . These letters are produced at some fixed rate in time, say one letter each  $\tau_c$  seconds. In each interval of  $\tau_c$  seconds, the digital data modulator (DDM) produces one of a fixed set of waveforms, say  $s_1(t), s_2(t), \dots, s_K(t)$ , each of duration  $\tau_c$ . The particular waveform produced is determined by the letter entering the DDM in that interval,  $a_1$  causing  $s_1(t)$ ,

$a_2$  causing  $s_2(t)$ , and so forth. Thus the entire waveform input to the channel has the form

$$\sum_n s_{i_n}(t - n\tau_c)$$

where the sequence  $i_n$ ,  $n = \dots, -1, 0, 1, \dots$  is determined by the corresponding inputs to the DDM.

The digital data demodulator (DDD) takes the received waveform from the channel and converts it into a sequence of letters from a finite alphabet, say  $b_1, \dots, b_J$ , producing letters again at a rate of one letter each  $\tau_c$  seconds.

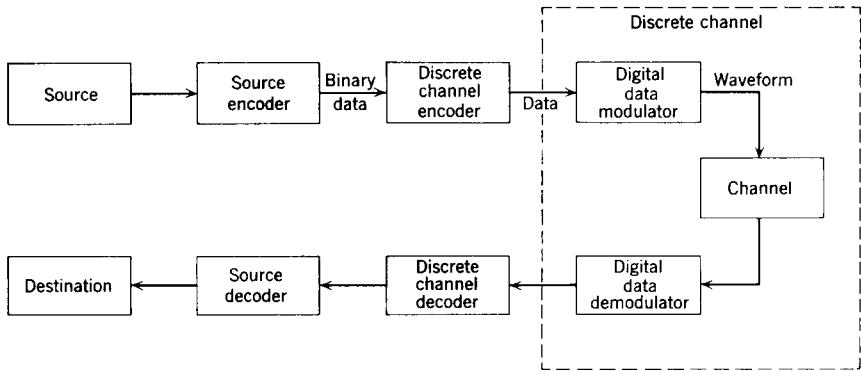


Figure 1.3.2. Representation of waveform channel as discrete channel.

In the simplest case, each letter from the DDD will be a decision (perhaps incorrect) on what letter entered the DDM in the corresponding time interval, and in this case the alphabet  $b_1, \dots, b_J$  will be the same as the alphabet at the input to the DDM. In more sophisticated cases, the output from the DDD will also contain information about how reliable the decision is, and in this case the output alphabet for the DDD will be larger than the input alphabet to the DDM.

It can be seen from Figure 1.3.2 that the combination of DDM, waveform channel, and DDD can together be considered as a discrete channel, and it is this that gives discrete channels their importance as models of physical channels. If the noise is independent between successive intervals of  $\tau_c$  seconds, as will be the case for additive white Gaussian noise, then the above discrete channel will also be memoryless.

By discussing encoding and decoding for discrete channels as a class, we shall first find out something about the discrete channel encoder and decoder in Figure 1.3.2, and second, we shall be able to use these results to say something about how a DDM and DDD should be designed in such a system.

One of the most important parameters of a channel is its capacity. In

Chapter 4 we define and show how to calculate capacity for a broad class of discrete channels, and in Chapters 7 and 8, the treatment is extended to nondiscrete channels. Capacity is defined using an information measure similar to that used in discussing sources and the capacity can be interpreted as the maximum average amount of information (in bits per second) that can be transmitted over the channel. It turns out that the capacity of a non-discrete channel can be approached arbitrarily closely by the capacity of a discrete channel made up of an appropriately chosen digital data modulator and digital data demodulator combined with the nondiscrete channel.

The significance of the capacity of a channel comes primarily from the noisy-channel coding theorem and its converse. In imprecise terms, this coding theorem states that, for a broad class of channels, if the channel has capacity  $C$  bits per second and if binary data enters the channel encoder (see Figure 1.1.2) at a rate (in binary digits per second)  $R < C$ , then by appropriate design of the encoder and decoder, it is possible to reproduce the binary digits at the decoder output with a probability of error as small as desired. This result is precisely stated and proved in Chapter 5 for discrete channels and in Chapters 7 and 8 for nondiscrete channels. The far-reaching significance of this theorem will be discussed later in this section, but not much intuitive plausibility can be given until Chapter 5. If we combine this result with the source coding theorem referred to in the last section, we find that, if a discrete source has an entropy (in bits per second) less than  $C$ , then the source output can be recreated at the destination with arbitrarily small error probability through the use of appropriate coding and decoding. Similarly, for a nondiscrete source, if  $R$  is the minimum number of binary digits per second required to reproduce the source output within a given level of average distortion, and if  $R < C$ , then the source output can be transmitted over the channel and reproduced within that level of distortion.

The converse to the coding theorem is stated and proved in varying degrees of generality in Chapters 4, 7, and 8. In imprecise terms, it states that if the entropy of a discrete source, in bits per second, is greater than  $C$ , then, independent of the encoding and decoding used in transmitting the source output over the channel, the error probability in recreating the source output at the destination cannot be less than some positive number which depends on the source and on  $C$ . Also, as shown in Chapter 9, if  $R$  is the minimum number of binary digits per second required to reproduce a source within a given level of average distortion, and if  $R > C$ , then, independent of the encoding and decoding, the source output cannot be transmitted over the channel and reproduced within that given average level of distortion.

The most surprising and important of the above results is the noisy channel coding theorem which we now discuss in greater detail. Suppose that we want to transmit data over a discrete channel and that the channel accepts

an input letter once each  $\tau_c$  seconds. Suppose also that binary data are entering the channel encoder at a rate of  $R$  binary digits per second. Let us consider a particular kind of channel encoder, called a block encoder, which operates in the following way: the encoder accumulates the binary digits at the encoder input for some fixed period of  $T$  seconds, where  $T$  is a design parameter of the encoder. During this period,  $TR$  binary digits enter the encoder (for simplicity, we here ignore the difficulty that  $TR$  might not be an integer). We can visualize the encoder as containing a list of all  $2^{TR}$  possible sequences of  $TR$  binary digits and containing alongside each of these sequences a code word consisting of a sequence of  $N = T/\tau_c$  channel input letters. Upon receiving a particular sequence of  $TR$  binary digits, the encoder finds that

| Binary input<br>sequences to<br>encoder |   | Code word<br>outputs from<br>encoder |
|---|---|--------------------------------------|
| <b>00</b>                               | → | $a_1a_1a_1$                          |
| <b>01</b>                               | → | $a_2a_3a_1$                          |
| <b>10</b>                               | → | $a_3a_1a_2$                          |
| <b>11</b>                               | → | $a_1a_2a_3$                          |

Figure 1.3.3. Example of discrete channel encoder,  
 $TR = 2, N = 3$ .

sequence in the list and transmits over the channel the corresponding code word in the list. It takes  $T$  seconds to transmit the  $N$  letter code word over the channel and, by that time, another sequence of  $TR$  binary digits has entered the encoder, and the transmission of the next code word begins. A simple example of such an encoder is given in Figure 1.3.3. For that example, if the binary sequence **0011** ··· enters the encoder, the **00** is the encoder input in the first  $T$ -second interval and at the end of this interval the code word  $a_1a_1a_1$  is formed and transmitted in the second  $T$ -second interval. Similarly, **11** is the encoder input in the second  $T$ -second time interval and the corresponding code word, transmitted in the third time interval, is  $a_1a_2a_3$ .

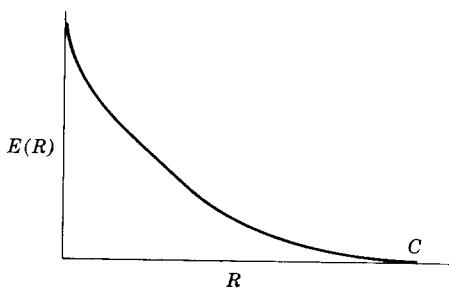
The decoder for such a block encoder works in a similar way. The decoder accumulates  $N$  received digits from the channel corresponding to a transmitted code word and makes a decision (perhaps incorrect) concerning the corresponding  $TR$  binary digits that entered the encoder. This decision making can be considered as being built into the decoder by means of a list of all possible received sequences of  $N$  digits, and corresponding to each of these sequences the appropriate sequence of  $TR$  binary digits.

For a given discrete channel and a given rate  $R$  of binary digits per second entering the encoder, we are free to choose first  $T$  (or, equivalently,  $N = T/\tau_c$ ), second the set of  $2^{TR}$  code words, and third the decision rule. The

probability of error in the decoded binary data, the complexity of the system, and the delay in decoding all depend on these choices. In Chapter 5 the following relationship is established between the parameter  $T$  and the probability,  $P_e$ , of decoding a block of  $TR$  binary digits incorrectly: it is shown for a broad class of channels that it is possible to choose the  $2^{TR}$  code words and the decision rule in such a way that

$$P_e \leq \exp [-TE(R)]$$

The function  $E(R)$  is a function of  $R$  (the number of binary digits per second entering the encoder) and depends upon the channel model but is independent of  $T$ . It is shown that  $E(R)$  is decreasing with  $R$  but is positive for all  $R$  less than channel capacity (see Figure 1.3.4). It turns out that the above bound on  $P_e$  is quite tight and it is not unreasonable to interpret  $\exp [-TE(R)]$  as an estimate of the minimum probability of error (over all choices of the



*Figure 1.3.4. Sketch of function  $E(R)$  for a typical channel model.*

$2^{TR}$  code words and all decision rules) that can be achieved using a block encoder with the constraint time  $T$ . Thus, to make  $P_e$  small, it is necessary to choose  $T$  large and the closer  $R$  is to  $C$ , the larger  $T$  must be.

In Chapter 6 we shall discuss ways of implementing channel encoders and decoders. It is difficult to make simple statements about either the complexity or the error probability of such devices. Roughly, however, it is not hard to see that the complexity increases with the constraint time  $T$  (in the best techniques, approximately linearly with  $T$ ), that  $P_e$  decreases with  $T$  for fixed  $R$ , and that  $T$  must increase with  $R$  to achieve a fixed value of  $P_e$ . Thus, roughly, there is a tradeoff between complexity, rate, and error probability. The closer  $R$  is to capacity and the lower  $P_e$  is, the greater the required encoder and decoder complexity is.

In view of the above tradeoffs, we can see more clearly the practical advantages of the separation of encoder and decoder in Figure 1.3.2 into two

parts. In recent years, the cost of digital logic has been steadily decreasing whereas no such revolution has occurred with analog hardware. Thus it is desirable in a complex system to put as much of the complexity as possible in the digital part of the system. This is not to say, of course, that completely analog communication systems are outmoded, but simply that there are many advantages to a primarily digital system that did not exist ten years ago.

### **Historical Notes and References**

Much of modern communication theory stems from the works of Shannon (1948), Wiener (1949), and Kotel'nikov (1947). All of these men recognized clearly the fundamental role of noise in limiting the performance of communication systems and also the desirability of modeling both signal and noise as random processes. Wiener was interested in finding the best linear filter to separate the signal from additive noise with a prescribed delay and his work had an important influence on subsequent research in modulation theory. Also Wiener's interest in reception with negative delay (that is, prediction) along with Kolmogorov's (1941) work on prediction in the absence of noise have had an important impact on control theory. Similarly, Kotel'nikov was interested in the detection and estimation of signals at the receiver. While his work is not as widely known and used in the United States as it should be, it provides considerable insight into both analog modulation and digital data modulation.

Shannon's work had a much more digital flavor than the others, and more important, focused jointly on the encoder and decoder. Because of this joint emphasis and the freedom from restrictions to particular types of receiver structures, Shannon's theory provides the most general conceptual framework known within which to study efficient and reliable communication.

For theoretically solid introductory texts on communication theory, see Wozencraft and Jacobs (1965) or Sakrison (1968).

## *Chapter 2*

### A MEASURE OF INFORMATION

The concepts of information and communication in our civilization are far too broad and pervading to expect any quantitative measure of information to apply universally. As explained in the last chapter, however, there are many communication situations, particularly those involving transmission and processing of data, in which the information (or data) and the channel are appropriately represented by probabilistic models. The measures of information to be defined in this chapter are appropriate to these probabilistic situations, and the question as to how appropriate these measures are generally revolves around the question of the appropriateness of the probabilistic model.

#### **2.1 Discrete Probability: Review and Notation**

We may visualize a probabilistic model as an experiment with an outcome chosen from a set of possible alternatives with a probability measure on the alternatives. The set of possible alternatives is called the sample space, each alternative being an element of the sample space. For a discrete set of alternatives, a probability measure simply involves the assignment of a probability to each alternative. The probabilities are of course nonnegative and sum to one. A sample space and its probability measure will be called an ensemble\* and will be denoted by a capital letter: the outcome will be denoted by the same letter, lower case. For an ensemble  $U$  with a sample space  $\{a_1, a_2, \dots, a_K\}$ , the probability that the outcome  $u$  will be a particular element  $a_k$  of the sample space will be denoted by  $P_U(a_k)$ . The probability that the outcome will be an arbitrary element  $u$  is denoted  $P_U(u)$ . In this expression, the subscript  $U$  is used as a reminder of which ensemble is under consideration

\* In most of the mathematical literature, what we call an ensemble here is called a probability space.

and the argument  $u$  is used as a variable that takes on values from the sample space. When no confusion can arise, the subscript will be omitted.

As an example, the ensemble  $U$  might represent the output of a source at a given time where the source alphabet is the set of letters  $\{a_1, \dots, a_K\}$  and  $P_U(a_k)$  is the probability that the output will be letter  $a_k$ .

We shall usually be concerned with experiments having a number of outcomes rather than a single outcome. For example, we might be interested in a sequence of source letters, or in the input and output to a channel, or in a sequence of inputs and outputs to a channel.

Suppose that we denote the outcomes by  $x$  and  $y$  in a two-outcome experiment and suppose that  $x$  is a selection from the set of alternatives  $a_1, \dots, a_K$  and  $y$  is a selection from the set of alternatives  $b_1, \dots, b_J$ . The set  $\{a_1, \dots, a_K\}$  is called the sample space for  $X$ , the set  $\{b_1, \dots, b_J\}$  is called the sample space for  $Y$ , and the set of pairs  $\{a_k, b_j\}$ ,  $1 \leq k \leq K$ ,  $1 \leq j \leq J$  is called the joint sample space. A probability measure on the joint space is given by the joint probability  $P_{XY}(a_k, b_j)$ , defined for  $1 \leq k \leq K$ ,  $1 \leq j \leq J$ . The combination of a joint sample space and probability measure for outcomes  $x$  and  $y$  is called a joint  $XY$  ensemble.

Within an ensemble or a joint ensemble, an event is by definition a subset of elements in the sample space. For a discrete ensemble, the probability of an event is the sum of the probabilities of the elements in the sample space comprising that event. In the  $XY$  ensemble under discussion, the event that  $x$  takes on a particular value  $a_k$  corresponds to the subset of pairs  $\{a_k, b_1; a_k, b_2; \dots; a_k, b_J\}$ . Thus the probability of this event is

$$P_X(a_k) = \sum_{j=1}^J P_{XY}(a_k, b_j) \quad (2.1.1)$$

In more abbreviated notation, this is written

$$P(x) = \sum_y P(x, y) \quad (2.1.2)$$

Likewise, the probability of a given  $y$  outcome is

$$P(y) = \sum_x P(x, y) \quad (2.1.3)$$

Considerable caution is needed with the notation  $P(x)$  and  $P(y)$  in (2.1.2) and (2.1.3). The  $x$  and  $y$  are doing double duty, both indicating which outcome is under consideration and acting as variables. In particular, if the  $x$  and  $y$  sample spaces are the same, we cannot substitute elements of the sample space for  $x$  or  $y$  without causing ambiguity.

If  $P_X(a_k) > 0$ , the conditional probability that outcome  $y$  is  $b_j$ , given that outcome  $x$  is  $a_k$ , is defined as

$$P_{Y|X}(b_j | a_k) = \frac{P_{XY}(a_k, b_j)}{P_X(a_k)} \quad (2.1.4)$$

In abbreviated notation, this is

$$P(y|x) = P(x,y)/P(x) \quad (2.1.5)$$

Likewise

$$P(x|y) = P(x,y)/P(y) \quad (2.1.6)$$

The events  $x = a_k$  and  $y = b_j$  are defined to be statistically independent if

$$P_{XY}(a_k, b_j) = P_X(a_k)P_Y(b_j) \quad (2.1.7)$$

If  $P_X(a_k) > 0$ , this is equivalent to

$$P_{Y|X}(b_j | a_k) = P_Y(b_j) \quad (2.1.8)$$

so that the conditioning does not alter the probability that  $y = b_j$ . The ensembles  $X$  and  $Y$  are statistically independent if (2.1.7) is satisfied for all pairs  $a_k b_j$  in the joint sample space.

Next consider an experiment with many outcomes, say  $u_1, u_2, \dots, u_N$ , each selected from a set of possible alternatives. The set of possible alternatives for outcome  $u_n$  is called the sample space for  $U_n$ ,  $1 \leq n \leq N$ , and the set of alternatives for the sequence  $u_1, \dots, u_N$  is called the joint sample space for the experiment. For discrete sample spaces, a probability measure is specified by the joint probability  $P_{U_1 \dots U_N}(u_1, \dots, u_N)$  defined for each possible sequence of alternatives for  $u_1 \dots u_N$  in the argument. The combination of joint sample space and joint probability assignment is called the joint ensemble  $U_1 \dots U_N$ .

The probability assignments for individual outcomes and combinations of outcomes is determined from  $P(u_1 \dots u_N)$  as in the two-outcome case. For example,

$$P_{U_n}(u_n) = \sum_{u_1} \dots \sum_{\substack{u_i \\ i \neq n}} \dots \sum_{u_N} P(u_1, \dots, u_N) \quad (2.1.9)$$

where the summation is over all alternatives for each outcome other than the  $n$ th. Likewise

$$P_{U_n U_m}(u_n, u_m) = \sum_{u_1} \dots \sum_{\substack{u_i \\ i \neq n}} \dots \sum_{\substack{u_N \\ i \neq m}} P(u_1, \dots, u_N) \quad (2.1.10)$$

The ensembles  $U_1, U_2, \dots, U_N$  are statistically independent if for all  $u_1, \dots, u_N$ ,

$$P_{U_1 \dots U_N}(u_1, \dots, u_N) = \prod_{n=1}^N P_{U_n}(u_n) \quad (2.1.11)$$

As an example of this notation, consider a sequence of  $N$  letters from a source with a binary alphabet, **0** or **1**. The sample space for each  $u_n$  is the set **{0,1}**. The joint sample space for the experiment is the set of all  $2^N$  sequences of  $N$  binary digits. The probability measure assigns a probability

to each of these sequences. In the special case where the source letters are statistically independent, the probability assignment has the form in (2.1.11). If the  $N$  letters are identically distributed, say with  $P_{U_n}(1) = q$  and  $P_{U_n}(0) = 1 - q$ , then the probability of a sequence depends only on the number of 1's in the sequence. The probability of each of the

$$\binom{N}{j} = \frac{N!}{j!(N-j)!}$$

sequences containing  $j$  1's and  $N - j$  0's is then  $q^j(1 - q)^{N-j}$ .

## 2.2 Definition of Mutual Information

Let  $\{a_1, \dots, a_K\}$  be the  $X$  sample space and  $\{b_1, \dots, b_J\}$  be the  $Y$  sample space in an  $XY$  joint ensemble with the probability assignment  $P_{XY}(a_k, b_j)$ . For example,  $x$  might be interpreted as being the input letter into a noisy discrete channel and  $y$  as being the output. We want a quantitative measure of how much the occurrence of a particular alternative, say  $b_j$ , in the  $Y$  ensemble tells us about the possibility of some alternative, say  $a_k$ , in the  $X$  ensemble. In probabilistic terms, the occurrence of  $y = b_j$  changes the probability of  $x = a_k$  from the a priori probability  $P_X(a_k)$  to the a posteriori probability  $P_{X|Y}(a_k | b_j)$ . The quantitative measure of this which turns out to be useful is the logarithm of the ratio of a posteriori to a priori probability. This gives us the following fundamental definition: *the information provided about the event  $x = a_k$  by the occurrence of the event  $y = b_j$  is*

$$I_{X;Y}(a_k; b_j) = \log \frac{P_{X|Y}(a_k | b_j)}{P_X(a_k)} \quad (2.2.1)$$

The base of the logarithm in this definition determines the numerical scale used to measure information. The most common bases are 2 and  $e$ . For base 2 logarithms, the numerical value of (2.2.1) is called the number of *bits* (binary digits) of information, and for natural logarithms, the numerical value of (2.2.1) is the number of *nats* (natural units) of information. Thus the number of nats is the number of bits times  $\ln 2 = 0.693$ . Since most of the theory and results are valid for any logarithm base, we shall specify the base only where necessary.

If we interchange the role of  $x$  and  $y$  in (2.2.1), we find that the information provided about the event  $y = b_j$  by the occurrence of  $x = a_k$  is

$$I_{Y;X}(b_j; a_k) = \log \frac{P_{Y|X}(b_j | a_k)}{P_Y(b_j)} \quad (2.2.2)$$

We now show, by using the definition of conditional probabilities, that the right-hand sides of (2.2.1) and (2.2.2) are identical. Because of this symmetry,

$I_{X;Y}(a_k; b_j)$  is called the *mutual information* between the events  $x = a_k$  and  $y = b_j$ .

$$\begin{aligned} I_{Y;X}(b_j; a_k) &= \log \frac{P_{XY}(a_k, b_j)}{P_X(a_k)P_Y(b_j)} \\ &= \log \frac{P_{X|Y}(a_k | b_j)}{P_X(a_k)} = I_{X;Y}(a_k; b_j) \end{aligned} \quad (2.2.3)$$

When no confusion can result, we shall abbreviate the notation to indicate the information provided about a particular  $x$  event by a particular  $y$  event by

$$I(x; y) = \log \frac{P(x | y)}{P(x)} \quad (2.2.4)$$

The full justification for the definition of information in (2.2.1) will become clear only as the theory unfolds. However, the following example will provide some intuitive appreciation.

**Example 2.1.** The channel in Figure 2.2.1 is called a binary symmetric channel. With probability  $1 - \epsilon$ , the output letter is a faithful replica of the input, and with probability  $\epsilon$ , it is the opposite of the input letter.

Assuming equally likely inputs,  $P_X(a_1) = P_X(a_2) = \frac{1}{2}$ , the joint probabilities are given by

$$\begin{aligned} P_{XY}(a_1, b_1) &= P_{XY}(a_2, b_2) = \frac{1 - \epsilon}{2} \\ P_{XY}(a_1, b_2) &= P_{XY}(a_2, b_1) = \frac{\epsilon}{2} \end{aligned}$$

Observing from this that the output letters are equally probable, we get

$$\begin{aligned} P_{X|Y}(a_1 | b_1) &= P_{X|Y}(a_2 | b_2) = 1 - \epsilon \\ P_{X|Y}(a_1 | b_2) &= P_{X|Y}(a_2 | b_1) = \epsilon \end{aligned} \quad (2.2.5)$$

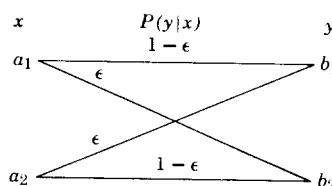


Figure 2.2.1. Binary symmetric channel.

The mutual information is then

$$\begin{aligned} I_{X;Y}(a_1; b_1) &= I_{X;Y}(a_2; b_2) = \log 2(1 - \epsilon) \\ I_{X;Y}(a_1; b_2) &= I_{X;Y}(a_2; b_1) = \log 2\epsilon \end{aligned} \quad (2.2.6)$$

For  $\epsilon = 0$  in Figure 2.2.1, the channel is noiseless, the output specifying the input with certainty. For  $\epsilon = \frac{1}{2}$ , the channel is completely noisy, the input and output being statistically independent. Now assume that  $\epsilon$  is a small number much less than  $\frac{1}{2}$  and assume that  $x = a_1$ , and  $y = b_1$ . At the channel output, the reception of letter  $b_1$  makes it highly likely that  $a_1$  was transmitted and we see that (2.2.6) asserts that the information provided by  $y = b_1$  about  $x = a_1$  is indeed positive in this case. For  $\epsilon = 0$ , this information is 1 bit, corresponding to the fact that  $y = b_1$ , specifies with certainty to the receiver which binary alternative was transmitted. As  $\epsilon$  gets larger, this mutual information decreases, corresponding to the increasing lack of certainty at the receiver that it was  $x = a_1$  that was transmitted.

Consider next the case where  $x = a_2$  is transmitted and  $y = b_1$  is received. The information given by (2.2.6) is in this case negative (for  $\epsilon < \frac{1}{2}$ ), corresponding to the fact that the reception of  $b_1$  is misleading, giving the receiver some degree of certainty that  $x = a_1$  rather than  $a_2$  was transmitted. We shall see in a later example how some subsequent positive information can correct the false impression at the receiver caused by an initial negative information. It is interesting to observe that as  $\epsilon$  approaches 0 this negative information approaches  $-\infty$ , corresponding to the receiver not only being misled, but misled with absolute certainty. Fortunately, if  $\epsilon = 0$ , this event cannot occur.

It can be seen from the definition in (2.2.1) that mutual information is a random variable, that is, a numerical function of the elements in the sample space. Mutual information is a rather unconventional kind of random variable since its value depends upon the probability measure, but it can be treated in the same way as any other random variable. In particular, mutual information has a mean value, a variance, moments of all orders, and a moment-generating function. The mean value, which is called the average mutual information and denoted by  $I(X; Y)$ , is given by\*

$$I(X; Y) = \sum_{k=1}^K \sum_{j=1}^J P_{XY}(a_k, b_j) \log \frac{P_{X|Y}(a_k | b_j)}{P_X(a_k)} \quad (2.2.7)$$

In abbreviated notation, this is

$$I(X; Y) = \sum_x \sum_y P(x, y) \log \frac{P(x | y)}{P(x)} \quad (2.2.8)$$

\* In this expression, and throughout the book, we take  $0 \log 0$  to be 0. This corresponds to the limit of  $W \log W$  as  $W$  approaches 0 from above.

It is seen that the average mutual information is a function only of the  $XY$  ensemble, whereas the random variable mutual information is also a function of the particular  $x$  and  $y$  outcomes. In Example 2.1, the mutual information takes on the value  $\log 2(1 - \epsilon)$  with probability  $1 - \epsilon$  and the value  $\log 2\epsilon$  with probability  $\epsilon$ . The average mutual information is then given by  $(1 - \epsilon) \log 2(1 - \epsilon) + \epsilon \log 2\epsilon$ .

An interesting special case of mutual information is the case when the occurrence of a given  $y$  outcome, say  $y = b_j$ , uniquely specifies the  $x$  outcome to be a given element  $a_k$ . In this instance,  $P_{X|Y}(a_k | b_j) = 1$  and

$$I_{X;Y}(a_k; b_j) = \log \frac{1}{P_X(a_k)} \quad (2.2.9)$$

Since this is the mutual information required to specify  $x = a_k$ , it is defined to be the self-information of the event  $x = a_k$  and it is denoted

$$I_X(a_k) = \log \frac{1}{P_X(a_k)} \quad (2.2.10)$$

In abbreviated notation, this is  $I(x) = -\log P(x)$ . The self-information of the event  $x = a_k$  is clearly a function only of the  $X$  ensemble. The self-information of  $x = a_k$  is always nonnegative and increases with decreasing  $P_X(a_k)$ . It can be interpreted either as the a priori uncertainty of the event  $x = a_k$  or the information required to resolve this uncertainty. Self-information appears at first to be a simpler concept than mutual information since it is defined in terms of a single rather than a joint ensemble. We have defined mutual information first partly because it generalizes naturally to nondiscrete sample spaces while self-information does not, and partly because an intuitive understanding of self-information as the resolution of uncertainty is virtually impossible in terms of a single ensemble. Many of the attempts in the literature to give heuristic interpretations to self-information in terms of a single ensemble have led to considerable confusion. In particular, in the context of a single ensemble, it is difficult to see why information and uncertainty should not be inversely related rather than being two different ways of looking at the same thing.

**Example 2.2.** Consider an ensemble  $X$  for which the sample space is the set of all binary sequences of a given length  $m$ . Assume that the sequences are all equally likely, so that there are  $2^m$  elements in the sample space, each with probability  $2^{-m}$ . The self-information of any given outcome is then

$$I(x) = -\log P(x) = \log 2^m = m \text{ bits} \quad (2.2.11)$$

It is intuitively satisfying that it takes  $m$  bits of self-information to specify a sequence of  $m$  binary digits and this example brings out clearly the reason for the appearance of a logarithm in the measures of information.

In a joint  $XY$  ensemble, we define the conditional self-information of an event  $x = a_k$ , given the occurrence of  $y = b_j$ , as

$$I_{X|Y}(a_k | b_j) = \log \frac{1}{P_{X|Y}(a_k | b_j)} \quad (2.2.12)$$

More simply,

$$I(x | y) = -\log P(x | y)$$

This is the self-information of the event  $x = a_k$  in an ensemble conditioned by  $y = b_j$ . It can be interpreted as the information that must be supplied to an observer to specify  $x = a_k$  after the observer has observed the occurrence of  $y = b_j$ . Combining definitions (2.2.1), (2.2.10), and (2.2.12), we obtain

$$I(x; y) = I(x) - I(x | y) \quad (2.2.13)$$

In words, the information provided about an  $x$  outcome by a  $y$  outcome is the self-information required to specify the  $x$  outcome less the uncertainty in that  $x$  outcome given  $y$ .

Self-information is a random variable just as mutual information is. *The entropy of the ensemble is defined to be the average value of the self-information and is given by*

$$H(X) = \sum_{k=1}^K P_X(a_k) \log \frac{1}{P_X(a_k)} \quad (2.2.14)$$

$$= - \sum_x P(x) \log P(x) \quad (2.2.15)$$

There is little reason for the use of the letter  $H$  here except that it is almost universally used in information theory. The entropy of an ensemble is closely related to the entropy used in statistical thermodynamics and, in fact, is that entropy (aside from an additive constant) if we interpret the set of  $a_k$  as a set of vanishingly small equal volume elements of phase space.\* Fortunately, information entropy is a much simpler concept than thermodynamic entropy.

Conditional self-information is also a random variable over the joint ensemble  $XY$  and has an average given by

$$\begin{aligned} H(X | Y) &= \sum_{x,y} P(x,y) I(x | y) \\ &= - \sum_{x,y} P(x,y) \log P(x | y) \end{aligned} \quad (2.2.16)$$

This is interpreted as the average information (over  $x$  and  $y$ ) required to specify  $x$  after  $y$  is known.

\* See, for example, R. C. Tolman, *The Principles of Statistical Mechanics*, p. 168 (Tolman's  $\bar{H}$  is the negative of the entropy).

If (2.2.13) is averaged over the  $XY$  ensemble, we find that the average mutual information between  $x$  and  $y$  is the difference between the entropy of  $X$  and the conditional entropy of  $X$  given  $Y$ .

$$I(X; Y) = H(X) - H(X | Y) \quad (2.2.17)$$

This equation shows that we can interpret  $I(X; Y)$  as the average amount of uncertainty in  $X$  resolved by the observation of the outcome in the  $Y$  ensemble,  $H(X | Y)$  being the average remaining uncertainty in  $X$  after the observation.

We can obtain some additional relations between self and mutual informations by considering a joint ensemble  $XY$  to be a single ensemble whose elements are the  $xy$  pairs of the joint sample space. The self-information of an  $x,y$  pair is then

$$I(x,y) = -\log P(x,y) \quad (2.2.18)$$

Since  $P(x,y) = P(x)P(y | x) = P(y)P(x | y)$ , we obtain

$$I(x,y) = I(x) + I(y | x) = I(y) + I(x | y) \quad (2.2.19)$$

The mutual information can also be written in terms of  $I(x,y)$  as

$$I(x;y) = I(x) + I(y) - I(x,y) \quad (2.2.20)$$

Averaging these expressions over the joint  $XY$  ensemble, we obtain

$$H(XY) = H(X) + H(Y | X) = H(Y) + H(X | Y) \quad (2.2.21)$$

$$I(X; Y) = H(X) + H(Y) - H(XY) \quad (2.2.22)$$

Next, let  $u_1, \dots, u_N$  be the outcomes in a joint ensemble  $U_1 \dots U_N$ . The conditional mutual information between  $u_1$  and  $u_2$  given  $u_3$  is defined, consistently with (2.2.1), as

$$I(u_1; u_2 | u_3) = \log \frac{P(u_1 | u_2, u_3)}{P(u_1 | u_3)} \quad (2.2.23)$$

$$= I(u_1 | u_3) - I(u_1 | u_2, u_3) \quad (2.2.24)$$

The average conditional mutual information is then given by

$$I(U_1; U_2 | U_3) = \sum_{u_1} \sum_{u_2} \sum_{u_3} P(u_1, u_2, u_3) \log \frac{P(u_1 | u_2, u_3)}{P(u_1 | u_3)} \quad (2.2.25)$$

$$= H(U_1 | U_3) - H(U_1 | U_2 U_3) \quad (2.2.26)$$

We could now develop an unlimited number of relations between conditional and unconditional mutual and self informations by using joint outcomes in place of single outcomes in these quantities. One relation of particular interest is that the mutual information provided about a particular outcome  $u_1$  by a particular pair of outcomes  $u_2 u_3$  is equal to the information

provided about  $u_1$  by  $u_2$  plus that provided about  $u_1$  by  $u_3$  conditioned on  $u_2$ . To see this, we have

$$\begin{aligned} I(u_1; u_2) + I(u_1; u_3 | u_2) &= \log \frac{P(u_1 | u_2)}{P(u_1)} + \log \frac{P(u_1 | u_2, u_3)}{P(u_1 | u_2)} \\ &= \log \frac{P(u_1 | u_2, u_3)}{P(u_1)} = I(u_1; u_2, u_3) \end{aligned} \quad (2.2.27)$$

A second relationship, following from the chain rule of probability,

$$P(u_1, u_2, \dots, u_N) = P(u_1)P(u_2 | u_1) \cdots P(u_N | u_1, \dots, u_{N-1})$$

is

$$I(u_1, u_2, \dots, u_N) = I(u_1) + I(u_2 | u_1) + \cdots + I(u_N | u_1, \dots, u_{N-1}) \quad (2.2.28)$$

Averaging (2.2.27) and (2.2.28) over the joint ensemble, we obtain

$$I(U_1; U_2, U_3) = I(U_1; U_2) + I(U_1; U_3 | U_2) \quad (2.2.29)^*$$

$$H(U_1 U_2 \cdots U_N) = H(U_1) + H(U_2 | U_1) + \cdots + H(U_N | U_1 \cdots U_{N-1}) \quad (2.2.30)$$

**Example 2.3.** Consider the channel of Figure 2.2.1 again, but consider using it three times in succession so that the input is a sequence  $x_1 x_2 x_3$  of three binary digits and the output is a sequence  $y_1 y_2 y_3$  of three binary digits. Suppose also that we constrain the input to be a triple repetition of the same digit, using the sequence  $a_1 a_1 a_1$  with probability  $\frac{1}{2}$  and the sequence  $a_2 a_2 a_2$  with probability  $\frac{1}{2}$ . Finally, assume that the channel acts independently on each digit or, in other words, that

$$P(y_1 y_2 y_3 | x_1 x_2 x_3) = P(y_1 | x_1)P(y_2 | x_2)P(y_3 | x_3) \quad (2.2.31)$$

We shall analyze the mutual information when the sequence  $a_1 a_1 a_1$  is sent and the sequence  $b_2 b_1 b_1$  is received. We shall see that the first output provides negative information about the input but that the next two outputs provide enough positive information to overcome this initial confusion. As in (2.2.6), we have

$$I_{X_1; Y_1}(a_1; b_2) = \log 2\epsilon \quad (2.2.32)$$

$$\begin{aligned} I_{X_1; Y_2|Y_1}(a_1; b_1 | b_2) &= \log \frac{P_{X_1|Y_1 Y_2}(a_1 | b_2 b_1)}{P_{X_1|Y_1}(a_1 | b_2)} \\ &= \log \frac{\frac{1}{2}}{\epsilon} = -\log 2\epsilon \end{aligned} \quad (2.2.33)$$

\* All equations and theorems marked with an asterisk in this section and the next are also valid for nondiscrete ensembles (see Sections 2.4 and 2.5).

We see that the conditional information provided by the second output exactly counterbalances the negative information on the first output. This is intuitively satisfying since, after the reception of  $b_2 b_1$ , the receiver is just as uncertain of the input as it was initially. The conditional information provided by the third received digit is

$$I_{X_1; Y_3|Y_1 Y_2}(a_1; b_1 | b_2 b_1) = \log 2(1 - \epsilon) \quad (2.2.34)$$

The total information provided by the three received digits about the input is then positive, corresponding to the a posteriori probability of the input  $a_1 a_1 a_1$  being larger than the a priori probability of  $a_1 a_1 a_1$ .

## 2.3 Average Mutual Information and Entropy

In this section, we derive a number of inequalities concerning entropy and average mutual information.

**Theorem 2.3.1.** Let  $X$  be an ensemble with a sample space consisting of  $K$  elements. Then

$$H(X) \leq \log K \quad (2.3.1)$$

with equality if and only if the elements are all equally probable.

*Proof.* This theorem and a number of subsequent inequalities can be proven by use of the inequality

$$\begin{aligned} \ln z &< z - 1; & z > 0, z \neq 1 \\ \ln z &= z - 1; & z = 1 \end{aligned} \quad (2.3.2)$$

This is sketched in Figure 2.3.1 and is verified analytically by noting that the difference  $\ln z - (z - 1)$  has a negative second derivative and a stationary point at  $z = 1$ .

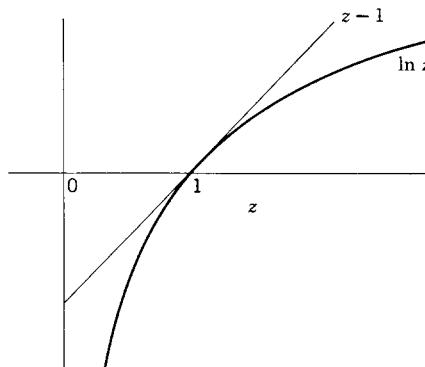


Figure 2.3.1. Sketch of  $\ln z$  and  $z - 1$ .

We now show that  $H(X) - \log K \leq 0$ .

$$\begin{aligned} H(X) - \log K &= \sum_x P(x) \log \frac{1}{P(x)} - \sum_x P(x) \log K \\ &= (\log e) \sum_x P(x) \ln \frac{1}{KP(x)} \end{aligned} \quad (2.3.3)$$

Considering the sum to be only over those  $x$  for which  $P(x) > 0$ , we can apply (2.3.2) to each term, obtaining

$$\begin{aligned} H(X) - \log K &\leq (\log e) \sum_x P(x) \left[ \frac{1}{KP(x)} - 1 \right] \\ &= \log e \left[ \sum_x \frac{1}{K} - \sum_x P(x) \right] \leq 0 \end{aligned} \quad (2.3.4)$$

The last inequality follows since the sum over  $x$  involves, at most,  $K$  terms. Both inequalities are equalities if and only if  $1/[KP(x)] = 1$  for all  $x$ ; this is equivalent to the elements being equiprobable. |

Since the entropy of an ensemble is maximized when the elements are equiprobable, we might surmise that the entropy of an ensemble is increased whenever the probability of one element is incrementally increased at the expense of some more probable element; this result is proven in Problem 2.15.

In the next theorem, we show that even though mutual information as a random variable can be negative, the average mutual information is always nonnegative.

**Theorem 2.3.2.\*** Let  $XY$  be a discrete joint ensemble. The average mutual information between  $X$  and  $Y$  satisfies

$$I(X;Y) \geq 0 \quad (2.3.5)$$

with equality if and only if  $X$  and  $Y$  are statistically independent.

---

*Proof.* We show that  $-I(X;Y) \leq 0$ .

$$-I(X;Y) = (\log e) \sum_{x,y} P(x,y) \ln \frac{P(x)}{P(x|y)} \quad (2.3.6)$$

Consider the sum in (2.3.6) to be over only those  $xy$  for which  $P(x,y) > 0$ . For these terms,  $P(x) > 0$ ,  $P(x|y) > 0$ , and (2.3.2) can be applied to each term.

$$-I(X;Y) \leq (\log e) \sum_{x,y} P(x,y) \left[ \frac{P(x)}{P(x|y)} - 1 \right] \quad (2.3.7)$$

$$= (\log e) \left[ \sum_{x,y} P(x)P(y) - \sum_{x,y} P(x,y) \right] \leq 0 \quad (2.3.8)$$

Equation 2.3.7 is satisfied with equality if and only if  $P(x) = P(x \mid y)$  whenever  $P(x,y) > 0$ . Since the sum in (2.3.8) is over only those  $xy$  pairs for which  $P(x,y) > 0$ , (2.3.8) is satisfied with equality if and only if  $P(x)P(y) = 0$  when  $P(x,y) = 0$ . Thus both inequalities are satisfied with equality, and consequently  $I(X;Y) = 0$ , if and only if  $X$  and  $Y$  are statistically independent. |

As an immediate consequence of this theorem, we can use the relationship  $I(X;Y) = H(X) - H(X \mid Y)$  to obtain

$$H(X) \geq H(X \mid Y) \quad (2.3.9)$$

with equality if and only if  $X$  and  $Y$  are statistically independent. Thus any conditioning on an ensemble can only reduce the entropy of the ensemble. It is important to note that (2.3.9) involves an averaging over both the  $X$  and  $Y$  ensembles. The quantity

$$-\sum_x P(x \mid y) \log P(x \mid y)$$

can be either larger or smaller than  $H(X)$  (see Problem 2.16).

Applying (2.3.9) to each term of (2.2.30), letting  $U_n$  play the role of  $X$  and  $U_1 \cdots U_{n-1}$  the role of  $Y$ , we have

$$H(U_1 \cdots U_N) \leq \sum_{n=1}^N H(U_n) \quad (2.3.10)$$

with equality if and only if the ensembles are statistically independent.

**Theorem 2.3.3.\*** Let  $XYZ$  be a discrete joint ensemble. Then

$$I(X;Y \mid Z) \geq 0 \quad (2.3.11)$$

with equality if and only if, conditional on each  $z$ ,  $X$  and  $Y$  are statistically independent; that is, if

$$P(x, y \mid z) = P(x \mid z)P(y \mid z) \quad (2.3.12)$$

for each element in the joint sample space for which  $P(z) > 0$ .

---

*Proof.* Repeat the steps of the proof of Theorem 2.3.2, adding in the conditioning on  $z$ . |

Combining (2.3.11) and (2.2.26), we see that

$$H(X \mid Z) \geq H(X \mid ZY) \quad (2.3.13)$$

with equality if and only if (2.3.12) is satisfied.

The situation in which  $I(X; Y \mid Z) = 0$  has a number of interesting interpretations. We can visualize the situation as a pair of channels in cascade as

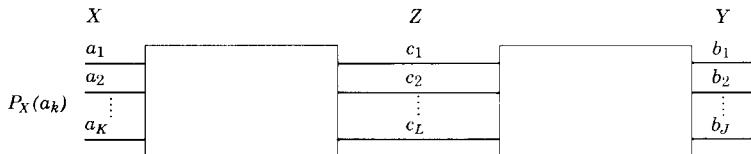
shown in Figure 2.3.2. The  $X$  ensemble is the input to the first channel, the  $Z$  ensemble is both the output from the first channel and the input to the second channel, and the  $Y$  ensemble is the output from the second channel. We assume that the output of the second channel depends statistically only on the input to the second channel; that is, that

$$P(y | z) = P(y | z, x); \quad \text{all } x, y, z \quad \text{with} \quad P(z, x) > 0 \quad (2.3.14)$$

Multiplying both sides by  $P(x | z)$ , we obtain (2.3.12), so that

$$I(X; Y | Z) = 0 \quad (2.3.15)^*$$

For such a pair of cascaded channels, it is reasonable to expect the average mutual information between  $X$  and  $Y$  to be no greater than that through



**Figure 2.3.2. Cascaded channels.**

either channel separately. We now show that this is indeed the case. From (2.2.29), we have both of the following equations.

$$I(X; YZ) = I(X; Y) + I(X; Z | Y) \quad (2.3.16)^*$$

$$= I(X; Z) + I(X; Y | Z) \quad (2.3.17)^*$$

Equating the right-hand sides and using (2.3.15), we have

$$I(X; Z) = I(X; Y) + I(X; Z | Y) \quad (2.3.18)^*$$

From (2.3.11),  $I(X; Z | Y) \geq 0$  and, thus, (2.3.15) implies that

$$I(X; Z) \geq I(X; Y) \quad (2.3.19a)^*$$

From the symmetry of (2.3.12) between  $X$  and  $Y$ , it also follows that

$$I(Y; Z) \geq I(X; Y) \quad (2.3.19b)^*$$

Writing out (2.3.19a) in terms of entropies, we have

$$\begin{aligned} H(X) - H(X | Z) &\geq H(X) - H(X | Y) \\ H(X | Z) &\leq H(X | Y) \end{aligned} \quad (2.3.20)$$

The average uncertainty  $H(X | Z)$  about the input of a channel given the output is called the *equivocation* on the channel, and thus (2.3.20) yields the

intuitively satisfying result that this uncertainty or equivocation can never decrease as we go further from the input on a sequence of cascaded channels.

Equations 2.3.19 and 2.3.20 become somewhat more surprising if we interpret the second box in Figure 2.3.2 as a data processor, processing the output of the first box which is now the channel. Whether this processing on the ensemble  $Z$  is deterministic or probabilistic, it can never decrease the equivocation about  $X$  nor increase the mutual information about  $X$ . This does not mean that we should never process the output of a channel and, in fact, processing is usually necessary to make any use of the output of the channel. Instead, it means that average mutual information must be interpreted as an average measure of available statistical data rather than in terms of the usefulness of the presentation. This result will be discussed in more detail in Chapter 4.

## 2.4 Probability and Mutual Information for Continuous Ensembles

Consider an ensemble  $X$  where the outcome  $x$  is a selection from the sample space consisting of the set of real numbers. A probability measure on this sample space is most easily given in terms of the distribution function,

$$F_X(x_1) = \Pr[x \leq x_1] \quad (2.4.1)$$

For each real number  $x_1$ ,  $F_X(x_1)$  gives the probability that the outcome  $x$  will be less than or equal to  $x_1$ . The probability that the outcome  $x$  lies in an interval  $x_1 < x \leq x_2$  is then given by

$$\Pr[x_1 < x \leq x_2] = F_X(x_2) - F_X(x_1) \quad (2.4.2)$$

Since the probability of any event must be nonnegative, (2.4.2) implies that  $F_X(x_1)$  is a nondecreasing function of  $x_1$ . Excluding the possibility of infinite outcomes,  $F_X(x_1)$  climbs from 0 at  $x_1 = -\infty$  to 1 at  $x_1 = +\infty$ .

The probability density of  $X$  (if it exists) is given by

$$p_X(x_1) = \frac{dF_X(x_1)}{dx_1} = \lim_{\Delta \rightarrow 0} \frac{F_X(x_1) - F_X(x_1 - \Delta)}{\Delta} \quad (2.4.3)$$

$$= \lim_{\Delta \rightarrow 0} \frac{\Pr[x_1 - \Delta < x \leq x_1]}{\Delta} \quad (2.4.4)$$

Thus  $p_X(x_1)$  is the density of probability per unit length on the  $x$  axis. A probability density is nonnegative, can be greater than one, but its integral from  $-\infty$  to  $+\infty$  must be one. Observe from (2.4.4) that if  $p_X(x_1)$  is finite, then the probability is zero that the outcome  $x$  takes on exactly the value  $x_1$ . If the outcome  $x$  takes on the value  $x_1$  with nonzero probability, it is often convenient to consider  $p_X$  as containing an impulse of magnitude  $\Pr(x = x_1)$  at the point  $x_1$ .

Next let us consider a joint ensemble,  $XY$ , in which both the  $x$  outcome and the  $y$  outcome are selections from the sample space consisting of the set of real numbers. A probability measure on the joint sample space can be given in terms of the joint probability distribution function,

$$F_{XY}(x_1, y_1) = \Pr[x \leq x_1, y \leq y_1] \quad (2.4.5)$$

This is a nondecreasing function of two variables, and for each pair of values,  $x_1$  and  $y_1$ , it gives the probability that the  $x$  outcome is less than or equal to  $x_1$  and the  $y$  outcome is less than or equal to  $y_1$ . The distribution functions of  $X$  and  $Y$  are given in terms of the joint distribution function by

$$F_X(x_1) = F_{XY}(x_1, \infty) \quad (2.4.6)$$

$$F_Y(y_1) = F_{XY}(\infty, y_1) \quad (2.4.7)$$

The joint probability density of  $X$  and  $Y$  (if it exists) is given by

$$p_{XY}(x_1, y_1) = \frac{\partial^2 F_{XY}(x_1, y_1)}{\partial x_1 \partial y_1} \quad (2.4.8)$$

The function  $p_{XY}$  is a probability density per unit area in the  $xy$  plane and the probability that the joint outcome  $x, y$  falls in a particular region of the plane is given by the integral of  $p_{XY}$  over that region.

The individual probability densities, defined by (2.4.3), are also given by

$$p_X(x_1) = \int_{-\infty}^{\infty} p_{XY}(x_1, y_1) dy_1 \quad (2.4.9)$$

$$p_Y(y_1) = \int_{-\infty}^{\infty} p_{XY}(x_1, y_1) dx_1 \quad (2.4.10)$$

If  $p_X(x_1)$  is nonzero, the conditional probability density of  $Y$  given  $X$  is given by

$$p_{Y|X}(y_1 | x_1) = \frac{p_{XY}(x_1, y_1)}{p_X(x_1)} \quad (2.4.11)$$

This is the probability density per unit length of the  $y$  outcome at the value  $y_1$  given that the  $x$  outcome has the value  $x_1$ . Likewise

$$p_{X|Y}(x_1 | y_1) = \frac{p_{XY}(x_1, y_1)}{p_Y(y_1)} \quad (2.4.12)$$

As with discrete ensembles, we shall often omit the subscripts on probability densities where no confusion can arise. When we do this, it must be remembered that, for example,  $p(x)$  is the probability density of the outcome  $x$  and is not necessarily the same function as  $p(y)$ , the probability density of the outcome  $y$ .

For joint ensembles with more than two outcomes, the joint distribution function and the various joint, single, and conditional probability densities are defined in the analogous way.

We can now define mutual information for a continuous joint ensemble. Let the joint ensemble  $XY$  have  $X$  and  $Y$  sample spaces each consisting of the set of real numbers and have a joint probability density  $p_{XY}(x_1, y_1)$ . The mutual information between the  $x$  outcome having a value  $x_1$  and the  $y$  outcome having a value  $y_1$  is defined as

$$I_{X;Y}(x_1; y_1) = \log \frac{p_{XY}(x_1, y_1)}{p_X(x_1)p_Y(y_1)} \quad (2.4.13)$$

In our abbreviated notation, this becomes

$$I(x; y) = \log \frac{p(x, y)}{p(x)p(y)} \quad (2.4.14)$$

Using (2.4.11) and (2.4.12), this can also be expressed as

$$I(x; y) = \log \frac{p(x | y)}{p(x)} = \log \frac{p(y | x)}{p(y)} \quad (2.4.15)$$

The similarity between the definition of information here and that for discrete ensembles is pleasing from a mnemonic standpoint, but provides no real motivation for the definition. In order to provide this motivation, think of quantizing the  $x$  axis into intervals of length  $\Delta$  and quantizing the  $y$  axis into intervals of length  $\delta$ . The quantized outcomes then form a discrete ensemble and the mutual information between an  $x$  interval from  $x_1 - \Delta$  to  $x_1$  and a  $y$  interval from  $y_1 - \delta$  to  $y_1$  is given by

$$\log \frac{\Pr[x_1 - \Delta < x \leq x_1, y_1 - \delta < y \leq y_1]}{\Pr[x_1 - \Delta < x \leq x_1]\Pr[y_1 - \delta < y \leq y_1]} \quad (2.4.16)$$

Dividing numerator and denominator by  $\Delta \delta$ , we obtain

$$\log \frac{\frac{1}{\Delta \delta} \Pr[x_1 - \Delta < x \leq x_1, y_1 - \delta < y \leq y_1]}{(1/\Delta)\Pr[x_1 - \Delta < x \leq x_1](1/\delta)\Pr[y_1 - \delta < y \leq y_1]} \quad (2.4.17)$$

Passing to the limit as  $\Delta$  and  $\delta$  go to zero, this becomes  $I_{X;Y}(x_1; y_1)$  as defined above.

As with discrete ensembles, mutual information is a random variable, and has the average value

$$I(X; Y) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} p(x, y) \log \frac{p(x, y)}{p(x)p(y)} dx dy \quad (2.4.18)$$

In a slightly more general situation, suppose that the  $X$  sample space is the set of  $n$ -dimensional real-valued vectors and the  $Y$  sample space is the set of  $m$ -dimensional real-valued vectors. If  $p_{XY}(x_1, y_1)$  is the joint probability density of  $XY$  over the joint  $(n + m)$ -dimensional sample space and  $p_X(x_1)$  and  $p_Y(y_1)$  are the probability densities over the  $X$  space and  $Y$  space respectively, then  $I_{X;Y}(x_1; y_1)$  is again defined by (2.4.13) and can again be interpreted as a limit when each dimension of the joint space is quantized more and more finely. The average mutual information  $I(X; Y)$  is given by (2.4.18), where the integration is now over the joint  $(n + m)$ -dimensional space.

Next let  $x$ ,  $y$ , and  $z$  be outcomes with real-valued, finite-dimensional sample spaces and let  $p(x, y, z)$  be their joint probability density. The conditional mutual information between  $x$  and  $y$  given  $z$  is defined as

$$I(x; y | z) = \log \frac{p(x, y | z)}{p(x | z)p(y | z)} \quad (2.4.19)$$

This has the same interpretation as  $I(x; y)$  as a limit of finer and finer quantization on the  $x$ ,  $y$ , and  $z$  axes. The average conditional mutual information is given by

$$I(X; Y | Z) = \iiint p(x, y, z) \log \frac{p(x, y | z)}{p(x | z)p(y | z)} dx dy dz \quad (2.4.20)$$

Using these definitions, all of the theorems and equations with asterisks in Sections 2.2 and 2.3 follow immediately, using the same proofs and derivations as found there.

When discussing discrete ensembles, it was obvious that the average mutual information had nothing to do with the names or labels attached to the elements in the individual sample spaces. This invariance to labeling is also a property of average mutual information on continuous ensembles, although it is less obvious. To see this, consider an  $XZ$  joint ensemble with outcomes  $x$  and  $z$ , and let  $y$  be a transformation of the  $z$  outcome,  $y = f(z)$ . We can represent this situation by Figure 2.3.2, where  $x$  is the input to a channel,  $z$  the output, and  $y$  a transformation of  $z$ . As in the analysis, there,\*  $I(X; Y | Z) = 0$ , and consequently, as in (2.3.19),

$$I(X; Z) \geq I(X; Y) \quad (2.4.21)$$

Next, suppose that  $y$  is a reversible transformation of  $z$ , so that  $z = f^{-1}(y)$ . We can then consider  $y$  as the channel output and  $z$  as the transformed output, yielding

$$I(X; Y) \geq I(X; Z) \quad (2.4.22)$$

\* There are some minor mathematical problems here. Since  $y$  is uniquely specified by  $z$ , the joint probability density  $p(x, y, z)$  will have impulse functions in it. Since this is a special case of the more general ensembles to be discussed later, we shall ignore these mathematical details for the time being.

Combining these equations, we have  $I(X;Y) = I(X;Z)$  and, consequently, the average mutual information between two ensembles is invariant to any reversible transformation of one of the outcomes. The same argument of course can be applied independently to any reversible transformation of the other outcome.

Let us next consider whether a meaningful definition of self-information can be made for a continuous ensemble. Let  $X$  be an ensemble with a real-valued outcome  $x$  and a finite probability density  $p(x)$ . Let the  $x$  axis be quantized into intervals of length  $\Delta$ , so that the self-information of an interval from  $x_1 - \Delta$  to  $x_1$  is

$$\log \frac{1}{\Pr[x_1 - \Delta < x \leq x_1]} \quad (2.4.23)$$

In the limit as  $\Delta$  approaches 0,  $\Pr[x_1 - \Delta < x \leq x_1]$  approaches  $\Delta p_X(x_1)$ , which approaches 0. Thus the self-information of an interval approaches  $\infty$  as the length of the interval approaches 0. This result is not surprising if we think of representing real numbers by their decimal expansions. Since an infinite sequence of decimal digits is required to exactly specify an arbitrary real number, we would expect the self-information to be infinite. The difficulty here lies in demanding an exact specification of a real number. From a physical standpoint, we are always satisfied with an approximate specification, but any appropriate generalization of the concept of self-information must involve the kind of approximation desired. This problem will be treated from a fundamental standpoint in Chapter 9, but we shall use the term self-information only on discrete ensembles.

For the purposes of calculating and manipulating various average mutual informations and conditional mutual informations, it is often useful to define the entropy of a continuous ensemble. *If an ensemble  $X$  has a probability density  $p(x)$ , we define the entropy of  $X$  by*

$$H(X) = \int_{-\infty}^{\infty} p(x) \log \frac{1}{p(x)} dx \quad (2.4.24)$$

*Likewise, conditional entropy is defined by*

$$H(X | Y) = \int_{-\infty}^{\infty} \int p(x,y) \log \frac{1}{p(x|y)} dx dy \quad (2.4.25)$$

Using these definitions, we have, as in (2.2.17) and (2.2.22),

$$I(X;Y) = H(X) - H(X | Y) \quad (2.4.26)$$

$$= H(Y) - H(Y | X) \quad (2.4.27)$$

$$= H(X) + H(Y) - H(YX) \quad (2.4.28)$$

These entropies are not necessarily positive, not necessarily finite, not invariant to transformations of the outcomes, and not interpretable as average self informations.

**Example 4.** The following example of the preceding definitions will be useful later in dealing with additive Gaussian noise channels. Let the input  $x$  to a channel be a zero mean Gaussian random variable with probability density

$$p(x) = \frac{1}{\sqrt{2\pi E}} \exp\left(-\frac{x^2}{2E}\right) \quad (2.4.29)$$

The parameter  $E$  is the mean square value or “energy” of the input. Suppose that the output of the channel,  $y$ , is the sum of the input and an independent zero mean Gaussian random variable of variance  $\sigma^2$ . The conditional probability density of the output given the input is then

$$p(y | x) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left[-\frac{(y-x)^2}{2\sigma^2}\right] \quad (2.4.30)$$

That is, given  $x$ ,  $y$  has a Gaussian distribution of variance  $\sigma^2$  centered around  $x$ . The joint probability density  $p(x,y)$  is given by  $p(x)p(y/x)$  and the joint ensemble  $XY$  is fully specified. It is most convenient to calculate the average mutual information  $I(X;Y)$  from (2.4.27).

$$H(Y | X) = - \int p(x) \int p(y | x) \log p(y | x) dy dx \quad (2.4.31)$$

$$= \int p(x) \int p(y | x) \left[ \log \sqrt{2\pi\sigma^2} + \frac{(y-x)^2}{2\sigma^2} \log e \right] dy dx$$

$$= \int p(x) [\log \sqrt{2\pi\sigma^2} + \frac{1}{2} \log e] dx \quad (2.4.32)$$

$$= \frac{1}{2} \log 2\pi e \sigma^2 \quad (2.4.33)$$

In (2.4.32), we have used the fact that  $\int p(y/x)(y-x)^2 dy$  is simply the variance of the conditional distribution, or  $\sigma^2$ .

We next observe that the channel output is the sum of two independent Gaussian random variables and is thus Gaussian\* with variance  $E + \sigma^2$ ,

$$p(y) = \frac{1}{\sqrt{2\pi(E + \sigma^2)}} \exp\left[-\frac{y^2}{2(E + \sigma^2)}\right] \quad (2.4.34)$$

Calculating  $H(Y)$  in the same way as  $H(Y | X)$ , we obtain

$$H(Y) = \frac{1}{2} \log 2\pi e (E + \sigma^2) \quad (2.4.35)$$

$$I(X;Y) = H(Y) - H(Y | X) = \frac{1}{2} \log \left(1 + \frac{E}{\sigma^2}\right) \quad (2.4.36)$$

\* See Problem 2.22.

We observe that, as  $\sigma^2$  approaches 0, the output  $y$  approximates the input  $x$  more and more exactly and  $I(X; Y)$  approaches  $\infty$ . This is to be expected since we have already concluded that the self-information of any given sample value of  $x$  should be infinite.

We shall often be interested in joint ensembles for which some of the outcomes are discrete and some continuous. The easiest way to specify a probability measure on such ensembles is to specify the joint probability of the discrete outcomes taking on each possible joint alternative and to specify the conditional joint probability density on the continuous outcomes conditional on each joint alternative for the discrete outcomes. For example, if outcome  $x$  has the sample space  $(a_1, \dots, a_K)$  and outcome  $y$  has the set of real numbers as its sample space, we specify  $P_X(a_k)$  for  $1 \leq k \leq K$  and  $p_{Y|X}(y_1 | a_k)$  for all real numbers  $y_1$  and  $1 \leq k \leq K$ . The unconditional probability density for outcome  $y$  is then

$$p_Y(y) = \sum_{k=1}^K P_X(a_k) p_{Y|X}(y_1 | a_k) \quad (2.4.37)$$

The conditional probability of an  $x$  alternative given a  $y$  alternative,  $y_1$ , for which  $p_Y(y_1) > 0$  is

$$P_{X|Y}(a_k | y_1) = \frac{P_X(a_k) p_{Y|X}(y_1 | a_k)}{p_Y(y_1)} \quad (2.4.38)$$

The mutual information and average mutual information between  $x$  and  $y$  is given by

$$I_{X;Y}(a_k; y_1) = \log \frac{P_{X|Y}(a_k | y_1)}{P_X(a_k)} = \log \frac{p_{Y|X}(y_1 | a_k)}{p_Y(y_1)} \quad (2.4.39)$$

$$I(X; Y) = \sum_{k=1}^K \int_{y_1=-\infty}^{\infty} P_X(a_k) p_{Y|X}(y_1 | a_k) \log \frac{p_{Y|X}(y_1 | a_k)}{p_Y(y_1)} dy_1 \quad (2.4.40)$$

Conditional mutual information is defined in the analogous way. All of the relationships with asterisks in Sections 2.2 and 2.3 clearly hold for these mixed discrete and continuous ensembles.

## 2.5 Mutual Information for Arbitrary Ensembles

The previously discussed discrete ensembles and continuous ensembles with probability densities appear to be adequate to treat virtually all of the problems of engineering interest in information theory, particularly if we employ some judicious limiting operations to treat more general cases. However, in order to state general theorems precisely without a plethora of special cases, a more abstract point of view is often desirable. A detailed treatment of such a point of view requires measure theory and is beyond the

scope of this text.\* In this section, we shall briefly develop those results for the general case which can be understood without a measure theoretic background. These results will be used in Chapter 7 but are needed only in the treatment of channels which cannot be described in terms of well-behaved probability densities. The major point to be made in this section is that the theorems and equations marked with asterisks in Section 2.2 and 2.3 are valid in general.

In terms of measure theory, an ensemble  $X$  is specified by a sample space, a set of events, each of which is a subset of elements of the sample space, and a probability measure on the set of events. The set of events has the properties that any finite or countable union or intersection of a set of events is another event and that the complement of any event is another event. The probability measure has the properties that each event has a nonnegative probability, the entire sample space has the probability one, and the probability of any disjoint finite or countable union of events is equal to the sum of the probabilities of the individual events. For all practical purposes,† any subset of elements that we might wish to consider is an event and has a probability.

A joint ensemble  $XY$  (or  $X_1 \dots X_n$ ) is specified in the same way. The elements of the joint sample space are  $x, y$  pairs, and the events are subsets of the joint sample space. There is the additional restriction, however, that if  $A$  is an event in the  $X$  sample space and  $B$  is an event in the  $Y$  sample space, then the joint subset  $AB$  corresponding to  $x$  in  $A$ ,  $y$  in  $B$  is an event in the joint sample space. The individual probability measures  $P_X$  and  $P_Y$  on the individual ensembles are defined in terms of the joint probability measure  $P_{XY}$ . For example, if  $B$  is taken as the entire  $Y$  sample space, then

$$P_X(A) = P_{XY}(AB) \quad \text{for each event } A$$

In order to define the average mutual information between two ensembles, we must first discuss partitioning an ensemble. *A partition  $X_p$  of an ensemble  $X$  is defined as a finite collection  $(A_1, A_2, \dots, A_K)$ ,  $K \geq 1$ , of mutually exclusive events whose union is the entire sample space.* Physically, a partition can be interpreted as a rule for quantizing the outcome of an experiment. We can consider  $X_p$  to be a discrete ensemble with elements  $A_1, \dots, A_K$  and probabilities  $P_X(A_1), \dots, P_X(A_K)$ . Given a joint ensemble  $XY$ , we

\* See Pinsker (1964) for a development of this point of view. The translator's notes, by Feinstein, provide proofs of a number of Russian results in this area that are not widely available in translation.

† Mathematically, however, even for such a simple ensemble as the unit interval with a uniform probability density, it can be shown that there exist pathological sets of points which are not events [see Halmos (1950), p. 67; what we call an event is there called a measurable set]. A probability cannot be assigned to these nonevents without violating the axioms of probability.

can partition the  $X$  space into  $A_1, \dots, A_K$  and the  $Y$  space into  $B_1, \dots, B_J$  to obtain a joint discrete ensemble  $X_p Y_p$ . The joint probabilities are of course given by  $P_{XY}(A_k B_j)$ . The average mutual information between two ensembles  $XY$  is now defined as

$$I(X; Y) = \sup I(X_p; Y_p) \quad (2.5.1)$$

$$I(X_p; Y_p) = \sum_{k,j} P_{XY}(A_k B_j) \log \frac{P_{XY}(A_k B_j)}{P_X(A_k)P_Y(B_j)} \quad (2.5.2)$$

where the supremum is taken over all partitions of the  $X$  ensemble and all partitions of the  $Y$  ensemble.

We now show that if an  $X$  partition or  $Y$  partition is further subpartitioned,  $I(X_p; Y_p)$  cannot decrease. To see this, consider Figure 2.5.1. In this figure,  $Y_{p_2}$  is a subpartition of  $Y_{p_1}$  in the sense that the event  $B_1$  in  $Y_{p_1}$  is sub-

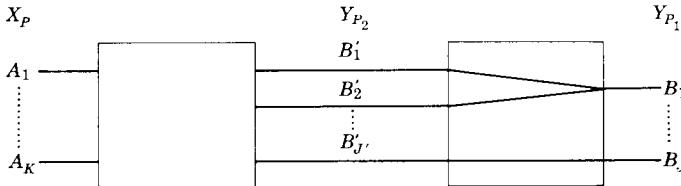


Figure 2.5.1. Effect of subpartitioning.

partitioned into  $B'_1$  and  $B'_2$  in  $Y_{p_2}$ . We have already seen in our analysis of Figure 2.3.2, however, that  $I(X_p; Y_{p_2}) \geq I(X_p; Y_{p_1})$ , and the same argument clearly applies no matter how the events in partition 1 are subpartitioned. The same argument applies to  $X$  by interchanging the roles of  $X$  and  $Y$ .

From the above result, we see that  $I(X; Y)$  can be interpreted as a limit of an appropriate sequence of finer and finer partitions, and thus for discrete ensembles and ensembles with well-behaved probability densities, the definition of (2.5.1) reduces to the definition already given.

Since  $I(X_p; Y_p)$  has already been shown to be nonnegative, it follows from (2.5.1) that  $I(X; Y)$  is nonnegative. Furthermore, the  $X$  and  $Y$  ensembles are statistically independent iff all partitioned ensembles are statistically independent, and thus  $I(X; Y) = 0$  iff  $X$  and  $Y$  are statistically independent. This proves theorem 2.3.2 for the general case.

For a joint ensemble  $XYZ$ , we similarly define

$$I(X; YZ) = \sup I(X_p; Y_p Z_p) \quad (2.5.3)$$

where the supremum is over all partitions of the  $X$  space, all partitions of the  $Y$  space, and all partitions of the  $Z$  space. There is a subtle problem arising

out of this definition. Do we obtain the same result if we interpret  $YZ$  as a single ensemble rather than as a joint ensemble? In other words, if we partition the joint  $YZ$  space rather than separately partition the  $Y$  space and the  $Z$  space, does the supremum change? Fortunately, Dobrushin's\* (1959) theorem shows that we get the same result both ways.

*Finally, average conditional mutual information is defined by*

$$I(X; Z | Y) = I(X; YZ) - I(X; Y) \quad (2.5.4)$$

This definition is not quite as general as our definition of  $I(X; Y)$ , the difficulty being that  $I(X; Z | Y)$  is undefined if both  $I(X; YZ)$  and  $I(X; Y)$  are infinite. Pinsker (1964) gives a more general measure theoretic definition, but we shall have no need for it here. See Problem 2.27 for an example showing why  $I(X; Z | Y)$  cannot reasonably be defined as  $\sup I(X_p; Z_p | Y_p)$ .

Definition (2.5.4) also leaves some doubt about whether the relation

$$I(X; Z | Y) = I(Z; X | Y) \quad (2.5.5)$$

s satisfied. To see that (2.5.5) is satisfied whenever the quantities are defined, let  $\epsilon > 0$  be an arbitrary number and choose a partition for which

$$I(X; YZ) - \epsilon \leq I(X_p; Y_p Z_p) \leq I(X; YZ)$$

$$I(X; Y) - \epsilon \leq I(X_p; Y_p) \leq I(X; Y)$$

$$I(Z; YX) - \epsilon \leq I(Z_p; Y_p X_p) \leq I(Z; YX)$$

$$I(Z; Y) - \epsilon \leq I(Z_p; Y_p) \leq I(Z; Y)$$

Clearly four different partitions can be chosen, one satisfying each of the above relations, and any partition that is a subpartition of each of those four satisfies all four equations together. From (2.5.4), such a partition will satisfy

$$|I(X; Z | Y) - I(X_p; Z_p | Y_p)| \leq \epsilon \quad (2.5.6)$$

$$|I(Z; X | Y) - I(Z_p; X_p | Y_p)| \leq \epsilon \quad (2.5.7)$$

On the other hand, since  $I(X_p; Z_p | Y_p) = I(Z_p; X_p | Y_p)$ , it follows that

$$|I(X; Z | Y) - I(Z; X | Y)| \leq 2\epsilon \quad (2.5.8)$$

Since  $\epsilon > 0$  can be chosen arbitrarily small, this establishes (2.5.5).

As an example of how to use these definitions, and also as a demonstration that  $I(X; Y)$  can be infinite, consider the ensemble where  $x$  is evenly distributed on the unit interval and  $y = x$  with probability 1. Partitioning the  $X$

\* See Pinsker (1964), p. 9 and p. 21 for a statement and proof of this theorem. We assume here that the  $YZ$  space and its events form the cartesian product of the individual spaces and their events [see Halmos (1950), p. 140].

and  $Y$  spaces each into  $K$  equal-length intervals, we see that  $I(X_p; Y_p) = \log K$ . Since  $K$  can be made arbitrarily large, we see that  $I(X; Y) = \infty$ .

We now turn to the problem of defining mutual information as a random variable for arbitrary ensembles. This definition can only be given in measure theoretic terms, but fortunately we shall have no subsequent need for it and present it only as an interesting sidelight. *Given an XY ensemble with joint probability measure  $P_{XY}$  and individual measures  $P_X$  and  $P_Y$ , we define the product probability measure  $P_{X \times Y}$  as the probability measure that would exist on the joint space if  $x$  and  $y$  were statistically independent and had the individual measures  $P_X$  and  $P_Y$ . The mutual information  $I(x; y)$  between a point  $x$  in the  $X$  ensemble and a point  $y$  in the  $Y$  ensemble is the log of the Radon-Nikodym derivative at  $x, y$ , of the joint probability measure  $P_{XY}$  with respect to the product probability measure  $P_{X \times Y}$ .* Gelfand and Yaglom (1958) have shown that if there is an event  $E$  for which  $P_{XY}(E) > 0$  and  $P_{X \times Y}(E) = 0$ , then  $I(X; Y) = \infty$ , otherwise  $I(X; Y)$  [as defined by (2.5.1)] is equal to the expectation of  $I(x; y)$ .

In the previous example, the joint probability measure is concentrated on the line  $x = y$  and the product probability measure is uniform over the unit square. Thus, if  $E$  is taken as the event  $x = y$ , we have  $P_{XY}(E) = 1$  and  $P_{X \times Y}(E) = 0$ , thus validating the Gelfand-Yaglom result for at least this one example.

## Summary and Conclusions

In this chapter, we have defined mutual and self-information and developed a number of properties of information which will be useful in subsequent chapters. The properties are for the most part what we would expect of a reasonable mathematical definition of information, but the real justification for the definitions will come in the subsequent chapters. The suggestiveness of the word "information" makes it easy to build up intuitive understanding of the concepts here, but we must be careful not to confuse this intuition with the mathematics. Ideally, our intuition should suggest new mathematical results, and the conditions required to prove these mathematical results should in turn sharpen our intuition.

## Historical Notes and References

For supplementary reading, Feller (1950) is an excellent text in probability theory. The material contained in this chapter is also contained, for the most part, in Fano (1961), Abramson (1963), and Ash (1965). Most of the results and concepts here (as in the other chapters) are due to Shannon (1948), whose original treatise is still highly recommended reading. Pinsker (1964) treats the topics in Section 2.5 in a more advanced and complete manner.

## *Chapter 3*

### CODING FOR DISCRETE SOURCES

In this chapter we shall be concerned with encoding the output of a discrete information source into a sequence of letters from a given code alphabet. We wish to choose the encoding rules in such a way that the source sequence can be retrieved from the encoded sequence, at least with high probability, and also in such a way that the number of code letters required per source letter is as small as possible. We shall see that the minimum number of binary code letters per source letter required to represent the source output is given by the entropy of the source.

As pointed out in Section 1.2, the output of a discrete source model is a random sequence of letters from a discrete alphabet. A discrete source model is appropriate for dealing with physical sources which produce discrete data and also for dealing with waveform sources for which the output has been converted to discrete data by some means such as sampling and quantizing. In Chapter 9, we shall give a fundamental treatment of the problem of encoding a waveform source into discrete data subject to a limitation on average distortion. In this chapter, we deal only with the problem of encoding (and decoding) mathematical models of discrete sources specified as random sequences. The construction of a mathematical model for a physical source is more difficult and is an art which depends on detailed insight into the source and its use; such model making cannot be treated intelligently in the abstract.

We shall assume then, that each unit of time the source produces one of a finite set of source letters, say  $a_1, \dots, a_K$ . We shall also assume, initially, that these source letters are produced with a fixed set of probabilities,  $P(a_1), \dots, P(a_K)$ , and that successive letters are statistically independent. Such sources are called *discrete memoryless sources*.<sup>\*</sup> This assumption of

\* More generally, the letters of a discrete source may be chosen from a countably infinite alphabet,  $a_1, a_2, \dots$ . We shall distinguish the finite alphabet case from the countable alphabet case by specifying the alphabet size in the former case.

statistical independence, or no memory, is somewhat unrealistic for most data sources. On the other hand, this assumption allows the important concepts of source coding to be developed without the mathematical complications introduced by statistical dependence.

In many practical forms of source coding, such as Morse code and shorthand, we assign short code words to commonly occurring letters or messages and longer code words to less frequent letters or messages. For example, in Morse code, the common letter *e* has the code word · and the uncommon letter *q* has · · -- as its code word. Such codes, in which different code words contain different numbers of code symbols, are called *variable-length codes*. If a source produces letters at a fixed rate in time and if it is necessary to transmit the encoded digits at a fixed rate in time, then a variable-length code leads to waiting line problems. Whenever the source emits an uncommon letter, a long code word is generated and the waiting line lengthens. Conversely, common letters generate short code words, decreasing the waiting line.

From a practical standpoint, it is usually desirable to avoid these waiting-line problems by using a fixed-length code, that is, a code in which each code word has the same length. The number of code words in these practical codes is generally quite small (for example, 32 words for teletype). We shall begin by discussing fixed-length codes in the next section, but our interest will be in codes of very long length. Such codes have little practical significance, but bring out clearly some of the deeper significance of self-information and entropy.

### 3.1 Fixed-Length Codes

Let  $\mathbf{u}_L = (u_1, u_2, \dots, u_L)$  denote a sequence of  $L$  consecutive letters from a discrete source. Each letter is a selection from the alphabet  $a_1, \dots, a_K$  and thus there are  $K^L$  different sequences of length  $L$  that might be emitted from the source. Suppose that we wish to encode such sequences into fixed-length code words. If the code alphabet contains  $D$  symbols and if the length of each code word is  $N$ , then there are  $D^N$  different sequences of code letters available as code words. Thus, if we wish to provide a separate code word for each source sequence (as we must if we want to retrieve every possible source sequence from its code word), we must have

$$\frac{N}{L} \geq \frac{\log K}{\log D} \quad (3.1.1)$$

Thus, for fixed-length codes, if we always wish to be able to decode the source sequence from the code word, we need at least  $\log K / \log D$  code letters per source letter. For example, in teletype transmission, the source has an alphabet of  $K = 32$  symbols (the 26 English letters plus 6 special

symbols). Encoding single source digits ( $L = 1$ ) into binary code letters ( $D = 2$ ), we need  $N = 5$  binary digits per source symbol to satisfy (3.1.1).

If we wish to use fewer than  $\log K/\log D$  code letters per source symbol, then we clearly must relax our insistence on *always* being able to decode the source sequence from the code sequence. In what follows, we shall assign code words only to a subset of the source sequences of length  $L$ . We shall see that by making  $L$  sufficiently large, we can make the probability of getting a source sequence for which no code word is provided arbitrarily small and at the same time make the number of code letters per source symbol as close to  $H(U)/\log D$  as we wish.

For a memoryless source, the probability of a given sequence  $\mathbf{u}_L = (u_1, \dots, u_L)$  of  $L$  source letters is the product of the probabilities of the individual source letters,

$$\Pr(\mathbf{u}_L) = \prod_{i=1}^L P(u_i)$$

In this equation, each letter  $u_i$  is a selection from the alphabet  $a_1, \dots, a_K$  and  $P(u_i)$  is the probability of that selection. For example, if a source has a two-letter alphabet,  $a_1$  and  $a_2$ , with  $P(a_1) = 0.7$  and  $P(a_2) = 0.3$ , then the probability of the sequence  $\mathbf{u}_3 = (u_1, u_2, u_3)$  for  $u_1 = a_2$ ,  $u_2 = a_1$ ,  $u_3 = a_1$ , is  $0.3 \times 0.7 \times 0.7 = 0.147$ .

The self-information of the sequence  $\mathbf{u}_L$  is given by

$$\begin{aligned} I(\mathbf{u}_L) &= -\log \Pr(\mathbf{u}_L) = -\log \prod_{i=1}^L P(u_i) \\ &= \sum_{i=1}^L -\log P(u_i) = \sum_{i=1}^L I(u_i) \end{aligned} \quad (3.1.2)$$

Each letter  $u_i$  is a statistically independent selection from the same source and therefore (3.1.2) states that  $I(\mathbf{u}_L)$  is the sum of  $L$  independent identically distributed variables. Since the average value of each of the random variables  $I(u_i)$  is the entropy of the source  $H(U)$ , the law of large numbers tells us that if  $L$  is large,  $[I(\mathbf{u}_L)]/L$  will be, with high probability, close to  $H(U)$ .

$$\frac{I(\mathbf{u}_L)}{L} \approx ? H(U) \quad (3.1.3)$$

The symbol  $\approx ?$  in (3.1.3) is used both to indicate approximate equality and to emphasize that we do not wish to be precise yet about what we mean by approximate equality. Let us first explore the consequences of (3.1.3) and then return to supply the necessary mathematical precision. Rearranging (3.1.3), we obtain

$$-\log_2 \Pr(\mathbf{u}_L) \approx ? LH(U) \quad (3.1.4)$$

$$\Pr(\mathbf{u}_L) \approx ? 2^{-LH(U)} \quad (3.1.5)$$

Here and in the rest of this chapter, all entropies will be in bits, that is, defined with base 2 logarithms. From (3.1.5), the probability of any typical long sequence of length  $L$  from the source is in some sense about equal to  $2^{-LH(U)}$  and, consequently, the number of such typical sequences  $M_T$  should be about

$$M_T \approx ? 2^{LH(U)} \quad (3.1.6)$$

If we wish to provide binary code words for all these typical sequences, we observe that there are  $2^N$  different binary sequences of length  $N$ , and, therefore, we require  $N$  to be approximately  $LH(U)$  to represent all the typical sequences from the source.

The preceding heuristic argument has given us three different interpretations of source entropy: one in terms of the probability of typical long source sequences, one in terms of the number of typical long source sequences, and one in terms of the number of binary digits required to represent the typical source sequences. These heuristic ideas are very useful in obtaining a simple picture of how sources behave, and the ideas extend easily to sources with statistical dependence between successive letters. However, before building on such ideas, it is necessary to supply the necessary precision.

We have seen that  $I(\mathbf{u}_L)$  is the sum of  $L$  independent, identically distributed random variables, each of which has a finite expectation,  $H(U)$ . The weak law of large numbers\* then states that for any  $\delta > 0$ , there is an  $\epsilon(L, \delta) > 0$  such that

$$\Pr\left[\left|\frac{I(\mathbf{u}_L)}{L} - H(U)\right| > \delta\right] \leq \epsilon(L, \delta) \quad (3.1.7)$$

and

$$\lim_{L \rightarrow \infty} \epsilon(L, \delta) = 0 \quad (3.1.8)$$

In words, this says that the probability that the sample mean,  $I(\mathbf{u}_L)/L$ , differs from  $H(U)$  by more than any fixed increment  $\delta$  goes to 0 with increasing  $L$ . For a given  $\delta$  and  $L$ , let  $T$  be the set of sequences,  $\mathbf{u}_L$ , for which

$$\left|\frac{I(\mathbf{u}_L)}{L} - H(U)\right| \leq \delta; \quad \mathbf{u}_L \in T \quad (3.1.9)$$

This is the set of typical sequences referred to before and, from (3.1.7), we have

$$\Pr(T) \geq 1 - \epsilon(L, \delta) \quad (3.1.10)$$

Rearranging (3.1.9), we now have, for  $\mathbf{u}_L \in T$ ,

$$L[H(U) - \delta] \leq I(\mathbf{u}_L) \leq L[H(U) + \delta] \quad (3.1.11)$$

$$2^{-L[H(U)-\delta]} \geq \Pr(\mathbf{u}_L) \geq 2^{-L[H(U)+\delta]} \quad (3.1.12)$$

\* See Problem 2.4 or, alternatively, see any elementary text on probability theory. For a countably infinite alphabet, we must assume that  $H(U)$  is finite. The finite variance assumption in Problem 2.4 is not necessary (see Feller, (1950) Section 10.2).

We can bound the number  $M_T$  of sequences in  $T$  by observing that

$$1 \geq \Pr(T) \geq M_T \min_{\mathbf{u}_L \in T} \Pr(\mathbf{u}_L)$$

Using (3.1.12) as a lower bound on  $\Pr(\mathbf{u}_L)$  for  $\mathbf{u}_L$  in  $T$ , this becomes

$$M_T \leq 2^{L[H(U)+\delta]} \quad (3.1.13)$$

Likewise, using (3.1.10),

$$1 - \epsilon(L, \delta) \leq \Pr(T) \leq M_T \max_{\mathbf{u}_L \in T} \Pr(\mathbf{u}_L)$$

Using (3.1.12) to upper bound  $\Pr(\mathbf{u}_L)$  for  $\mathbf{u}_L$  in  $T$ , this becomes

$$M_T \geq [1 - \epsilon(L, \delta)] 2^{L[H(U)-\delta]} \quad (3.1.14)$$

Equations 3.1.12 to 3.1.14 are precise statements of (3.1.5) and (3.1.6).

Next, suppose that we wish to encode the source sequences  $\mathbf{u}_L$  into codeword sequences of length  $N$  from a code alphabet of  $D$  letters. We shall map only one message sequence into each code sequence and thus there frequently will be a set of message sequences for which no code word has been provided. Define  $P_e$ , the probability of error, as the overall probability of this set for which code words have not been provided. Let us first choose  $N$  to satisfy

$$N \log D \geq L[H(U) + \delta] \quad (3.1.15)$$

Then, from (3.1.13), the total number of code words,  $D^N$ , is larger than  $M_T$  and we can provide code words for all  $\mathbf{u}_L$  in  $T$ .

If we use this strategy, then

$$P_e \leq \epsilon(L, \delta) \quad (3.1.16)$$

Then, if we let  $L$  become arbitrarily large, simultaneously increasing  $N$  to satisfy  $N/L \geq [H(U) + \delta]/\log D$ , we see from (3.1.8) and (3.1.16) that  $P_e$  approaches 0 for any  $\delta > 0$ . We next show that, if  $N/L$  is kept at any fixed amount below  $H(U)/\log D$ , the probability of error must approach 1 as  $L$  approaches  $\infty$ . Let us choose  $N$  to satisfy

$$N \log D \leq L[H(U) - 2\delta] \quad (3.1.17)$$

Thus the number of code words,  $D^N$ , is at most  $2^{L[H(U)-2\delta]}$ . Since every  $\mathbf{u}_L$  in  $T$  has a probability of, at most,  $2^{-L[H(U)-\delta]}$ , the overall probability of the sequences in  $T$  for which we can provide code words is, at most,

$$2^{-L[H(U)-\delta]} \times 2^{L[H(U)-2\delta]} = 2^{-L\delta}$$

We might also wish to provide code words for some of the source sequences outside of the set  $T$ , particularly those with large probability. On the other hand, the total probability of the sequences outside of  $T$  is, at most,  $\epsilon(L, \delta)$ .

Thus the probability of the set of all sequences, in  $T$  or out, for which code words can be provided if (3.1.17) is satisfied is upper bounded by

$$1 - P_e \leq \epsilon(L, \delta) + 2^{-L\delta} \quad (3.1.18)$$

Thus, as  $L \rightarrow \infty$ , with  $N/L \leq [H(U) - 2\delta]/\log D$ ,  $P_e$  must approach 1 for any  $\delta > 0$ . We can summarize these results in the following fundamental theorem.

**Theorem 3.1.1 (Source Coding Theorem).** Let a discrete memoryless source have finite entropy  $H(U)$  and consider a coding from sequences of  $L$  source letters into sequences of  $N$  code letters from a code alphabet of size  $D$ . Only one source sequence can be assigned to each code sequence and we let  $P_e$  be the probability of occurrence of a source sequence for which no code sequence has been provided. Then, for any  $\delta > 0$ , if

$$N/L \geq [H(U) + \delta]/\log D \quad (3.1.19)$$

$P_e$  can be made arbitrarily small by making  $L$  sufficiently large. Conversely, if

$$N/L \leq [H(U) - \delta]/\log D \quad (3.1.20)$$

then  $P_e$  must become arbitrarily close to 1 as  $L$  is made sufficiently large.

---

Theorem 3.1.1 has a very simple and useful heuristic interpretation. Since the code words are simply another representation of the probable source sequences, they should have essentially the same entropy as the source sequences. On the other hand, we saw in Chapter 2 that  $\log D$  is the largest entropy per letter that can be associated with a sequence of letters from an alphabet of size  $D$ ; this entropy occurs when the letters are equiprobable and statistically independent. Theorem 3.1.1 thus states that we can code letters from an arbitrary discrete memoryless source in such a way that the code letters have essentially their maximum entropy.

In interpreting this theorem, we can think of  $L$  as being the total number of messages that come out of the source during its lifetime, and think of  $N$  as being the total number of code letters that we are willing to use in representing the source. Then we can use any strategy that we wish in performing the coding and the theorem states that  $H(U)/\log D$  is the smallest number of code letters per source letter that we can use and still represent the source with high probability. The fact that such fixed length to fixed length encodings of very long sequences are rather impractical is immaterial in this general interpretation.

### 3.2 Variable-Length Code Words

Suppose that a discrete memoryless source  $U$  has a  $K$  letter alphabet  $a_1, \dots, a_K$  with probabilities  $P(a_1), \dots, P(a_K)$ . Each source letter is to be

represented by a code word consisting of a sequence of letters from a prescribed code alphabet. Denote the number of different symbols in the code alphabet by  $D$  and denote the number of letters in the code word corresponding to  $a_k$  by  $n_k$ . Later, we shall consider the source letters to be sequences of letters from a simpler source, thus generalizing the above situation considerably.

In what follows, we shall be interested primarily in  $\bar{n}$ , the average number of code letters per source letter  $u$ .

$$\bar{n} = \sum_{k=1}^K P(a_k)n_k \quad (3.2.1)$$

From the law of large numbers, if we encode a very long sequence of source letters by the above encoding scheme, then the number of code letters per source letter will be close to  $\bar{n}$  with high probability.

| Source Letters | $P(k_k)$ | Code I | Code II | Code III | Code IV |
|----------------|----------|--------|---------|----------|---------|
| $a_1$          | 0.5      | 0      | 0       | 0        | 0       |
| $a_2$          | 0.25     | 0      | 1       | 10       | 01      |
| $a_3$          | 0.125    | 1      | 00      | 110      | 011     |
| $a_4$          | 0.125    | 10     | 11      | 111      | 0111    |

Figure 3.2.1.

Before investigating how small  $\bar{n}$  can be made, let us consider some restrictions on variable-length codes which are illustrated in Figure 3.2.1.

Observe that code I has the unfortunate property that letters  $a_1$  and  $a_2$  are both coded into the same code word, 0. Thus this code word cannot be uniquely decoded into the source letter that gave rise to it. Since such codes cannot represent the letters from a source, we exclude them from further consideration.

Code II in Figure 3.2.1 suffers from the same defect as code I, although in a more subtle way. If the sequence  $a_1a_1$  is emitted from the source, it will be encoded into 00, which is the same as the code word for  $a_3$ . This will present no problem as far as decoding is concerned if some sort of spacing or separation is provided between successive code words. On the other hand, if such spacing is available, it should be considered as a separate symbol,  $s$ , in the code alphabet, and we should list the code words in code II as 0s, 1s, 00s, and 11s. By explicitly denoting spacing when required, we can consider such codes merely as special cases of codes with no spacing. For this reason, we exclude such codes from further special consideration.

We have observed that neither code I nor code II in Figure 3.2.1 can be used to represent the source since neither is uniquely decodable. This leads

us to the following definition: *A code is uniquely decodable if for each source sequence of finite length, the sequence of code letters corresponding to that source sequence is different from the sequence of code letters corresponding to any other source sequence.*

The above definition does not immediately suggest any way of determining whether or not a code is uniquely decodable.\* We shall be primarily interested, however, in a special class of codes which satisfy a restriction known as the prefix condition; these codes are easily shown to be uniquely decodable.

In order to define the prefix condition, let the  $k$ th code word in a code be represented by  $\mathbf{x}_k = (x_{k,1}, \dots, x_{k,n_k})$ , where  $x_{k,1}, \dots, x_{k,n_k}$  denote the individual code letters making up the code word. Any sequence made up of an initial part of  $\mathbf{x}_k$ , that is,  $x_{k,1}, \dots, x_{k,i}$  for some  $i \leq n_k$  is called a prefix of  $\mathbf{x}_k$ . *A prefix condition code is defined as a code in which no code word is the prefix of any other code word.*

In Figure 3.2.1, we see that code I is not a prefix condition code since **1**, the code word for  $a_3$ , is a prefix of **10**, the code word for  $a_4$ . Also, if we look carefully at the definition of a prefix, we see that **0**, the code word for  $a_1$ , is a prefix of **0**, the code word for  $a_2$ . In other words, any code with two or more code words the same is not a prefix condition code. It should be verified by the reader that codes II and IV in Figure 3.2.1 are not prefix condition codes but that code III is.

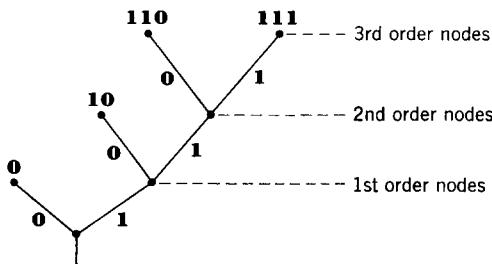
In order to decode a sequence of code words generated from a prefix condition code, we simply start at the beginning and decode one word at a time. When we get to the end of a code word, we know it is the end since that code word is not the prefix of any other code word. Thus we can uniquely decode any sequence of code letters corresponding to a sequence of source letters, and any code satisfying the prefix condition is proven to be a uniquely decodable code. For example, the source sequence  $a_1a_4a_2a_1$  is coded by code III of Figure 3.2.1 into **0111100**. Since the first letter in the coded sequence is **0**, and this corresponds to  $a_1$  and not the initial part of any other sequence, we decode  $a_1$ , leaving the coded sequence **111100**. Neither **1** nor **11** correspond to any code word, but **111** does, and is decoded into  $a_4$ , leaving **100**. Next, the **10** is decoded into  $a_2$ , leaving only **0**, which is decoded into  $a_1$ .

Not every uniquely decodable code satisfies the prefix condition. To see this, consider code IV in Figure 3.2.1. Here every code word is a prefix of every longer code word. On the other hand, unique decoding is trivial since the symbol **0** always indicates the beginning of a new code word. Prefix condition codes are distinguished from other uniquely decodable codes, however, by the end of a code word always being recognizable so that decoding can be accomplished without the delay of observing subsequent

\* Sardinas and Patterson (1953) have devised a test for unique decodability. See Problem 3.4 for a simple development and proof of this test.

code words. For this reason, prefix condition codes are sometimes called *instantaneous* codes.

A convenient graphical representation of a set of code words satisfying the prefix condition can be obtained by representing each code word by a terminal node in a tree. The tree representing the code words of code III in Figure 3.2.1 is shown in Figure 3.2.2. Starting from the root of the tree, the two branches leading to the first-order nodes correspond to the choice between **0** and **1** as the first letter of the code words. Similarly, the two branches stemming from the right-hand first-order node correspond to the choice between **0** and **1** for the second letter of the code word if the first letter is **1**; the same representation applies to the other branches. The successive digits of each code word can be thought of as providing the instructions for climbing from the root of the tree to the terminal node representing the desired source letter. A tree can also be used to represent the code words of a code that does



*Figure 3.2.2. Tree representing code words of code III in Figure 3.2.1.*

not satisfy the prefix condition, but in this case, some intermediate nodes in the tree will correspond to code words.

Next, we turn our attention to the problem of choosing a prefix condition code in such a way as to minimize  $\bar{n}$ . First, we treat the problem in an heuristic way, then we prove some general theorems about the code-word lengths, and finally we give an algorithm for constructing a code that minimizes  $\bar{n}$ .

For a code satisfying the prefix condition, we can visualize a receiver observing a sequence of code letters and tracing its way up a tree as in Figure 3.2.2 to decode a source message. At each node in this tree, the next code digit provides information about which branch to take. It can be seen that the sum of the mutual informations at the successive nodes leading to a given terminal node is just the self-information of the associated source digit. Thus, to achieve a small  $\bar{n}$ , it is desirable to achieve a large average mutual information at each of the intermediate nodes in the tree. This, in turn, suggests trying to choose the code words in such a way that each branch

rising from a node in the associated tree is equiprobable. Comparing the tree in Figure 3.2.2 with the source letter probabilities for code III in Figure 3.2.1, we see that each branch in the tree is taken with probability  $\frac{1}{2}$ . For this example,  $\bar{n} = 1.75$  and  $H(U) = 1.75$ . In a long sequence of, say,  $L$  source letters,  $\bar{n}$  is with high probability close to the number of code letters per source letter. Thus, if a code existed with  $\bar{n} < H(U)$ , we could, for large  $L$ , encode most source sequences of length  $L$  with fewer than  $H(U)$  code letters per source letter, in violation of Theorem 3.1.1. It follows that 1.75 is the minimum possible value for  $\bar{n}$  for this code. This is hardly surprising since we have been able to construct the code so that each code letter contains exactly 1 bit of information about the source output.

This example suggests an obvious generalization to constructing prefix condition codes for a general set of source letters. First, break the set of letters into  $D$  subsets making the probability of each subset as close to  $1/D$  as possible. Assign a different initial code letter to each of these subsets. Then break up each subset into  $D$  approximately equiprobable groups, and assign a second letter according to this division. Continue until each group contains only one source letter. The resulting code clearly satisfies the prefix condition. This procedure does not necessarily minimize  $\bar{n}$  since achieving a large average self-information on one code letter might force a poor choice on succeeding code letters. Notice that, if this division can be accomplished with exactly equiprobable groups at each point, then the source letter probabilities and code-word lengths must be related by

$$P(a_k) = D^{-n_k} \quad (3.2.2)$$

Before going more deeply into the minimization of the average length of a set of code words, we shall investigate the constraints between the code-word lengths of a prefix condition code.

**Theorem 3.2.1 (Kraft (1949) Inequality).** If the integers  $n_1, n_2, \dots, n_K$  satisfy the inequality

$$\sum_{k=1}^K D^{-n_k} \leq 1 \quad (3.2.3)$$

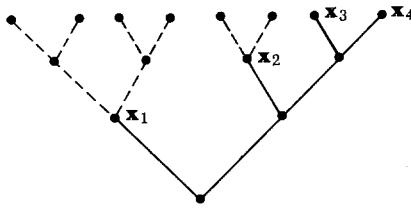
then a prefix condition code of alphabet size  $D$  exists with these integers as code-word lengths. Conversely, the lengths of every prefix condition code satisfy (3.2.3). (*Note.* The theorem *does not* say that any code whose lengths satisfy (3.2.3) is a prefix condition code. For example, the set of binary code words 0, 00, 11 satisfies (3.2.3), but not the prefix condition. The theorem *does* say that *some* prefix condition code exists with those lengths, for example, 0, 10, and 11.)

---

*Proof.* Define a *full tree* of order  $n$  and alphabet size  $D$  as a tree containing  $D^n$  terminal nodes of order  $n$  with  $D$  nodes of order  $i$  stemming from each

node of order  $i - 1$  for each  $i$ ,  $1 \leq i \leq n$ . Observe that a fraction  $D^{-1}$  of the nodes of each order  $i \geq 1$  stem from each of the  $D$  nodes of order one. Likewise, a fraction  $D^{-2}$  of the nodes of each order  $i \geq 2$  stem from each of the  $D^2$  nodes of order 2 and a fraction  $D^{-i}$  of the nodes of each order greater than or equal to  $i$  stem from each of the  $D^i$  nodes of order  $i$ .

Now let  $n_1, \dots, n_K$  satisfy (3.2.3). We shall show how to construct a prefix condition code with these lengths by starting with a full tree of order  $n$  equal to the largest of the  $n_k$  and assigning various nodes in this full tree as terminal nodes in the code tree. Thus, when the construction is completed, the code tree will be imbedded in the full tree as shown in Figure 3.2.3. To simplify notation, assume that the  $n_k$  are arranged in increasing order,



**Figure 3.2.3.** Binary code tree (solid) imbedded in full tree (dashed) of order 3.

$n_1 \leq n_2 \leq \dots \leq n_K$ . Pick any node of order  $n_1$ , say  $\mathbf{x}_1$ , in the full tree as the first terminal node in the code tree. All nodes on the full tree of each order greater than or equal to  $n_1$  are still available for use as terminal nodes in the code tree except for the fraction  $D^{-n_1}$  that stem from node  $\mathbf{x}_1$ . Next, pick any available node of order  $n_2$ , say  $\mathbf{x}_2$ , as the next terminal node in the code tree. All nodes in the full tree of each order greater than or equal to  $n_2$  are still available except for the fraction  $D^{-n_1} + D^{-n_2}$  that stem from either  $\mathbf{x}_1$  or  $\mathbf{x}_2$ . Continuing in this way, after the assignment of the  $k$ th terminal node in the code tree, all nodes in the full tree of each order greater than or equal to  $n_k$  are still available except for the fraction

$$\sum_{i=1}^k D^{-n_i}$$

stemming from  $\mathbf{x}_1$  to  $\mathbf{x}_k$ . From (3.2.3), this fraction is always strictly less than 1 for  $k < K$ , and thus there is always a node available to be assigned as the next terminal node.

To prove the second part of the theorem, notice that the code tree corresponding to any prefix condition code can be imbedded in a full tree whose order is the largest of the code-word lengths. A terminal node of order  $n_k$  in the code tree has stemming from it a fraction  $D^{-n_k}$  of the terminal nodes

in the full tree. Since the sets of terminal nodes in the full tree stemming from different terminal nodes in the code tree are disjoint, these fractions can sum to at most 1, yielding (3.2.3). |

We next prove a theorem about the code-word lengths for uniquely decodable codes which will justify our emphasis on codes satisfying the prefix condition.

**Theorem 3.2.2.\*** Let a code have code-word lengths  $n_1, \dots, n_K$  and have  $D$  symbols in the code alphabet. If the code is uniquely decodable, then the Kraft inequality, (3.2.3), must be satisfied.

*Proof.* Let  $L$  be an arbitrary positive integer, and consider the identity

$$\left( \sum_{k=1}^K D^{-n_k} \right)^L = \sum_{k_1=1}^K \sum_{k_2=1}^K \cdots \sum_{k_L=1}^K D^{-[n_{k_1} + n_{k_2} + \cdots + n_{k_L}]} \quad (3.2.4)$$

Observe that there is a distinct term on the right-hand side of (3.2.4) corresponding to each possible sequence of  $L$  code words. Furthermore, the quantity  $n_{k_1} + n_{k_2} + \cdots + n_{k_L}$  gives the total length in code letters of the corresponding sequence of code words. Thus, if we let  $A_i$  be the number of sequences of  $L$  code words having a total length of  $i$  code letters, we can rewrite (3.2.4) as

$$\left[ \sum_{k=1}^K D^{-n_k} \right]^L = \sum_{i=1}^{Ln_{\max}} A_i D^{-i} \quad (3.2.5)$$

where  $n_{\max}$  is the largest of the  $n_k$ .

If the code is uniquely decodable, all code word sequences with a length of  $i$  code letters are distinct and thus  $A_i \leq D^i$ . Substituting this inequality into (3.2.5), we have

$$\left[ \sum_{k=1}^K D^{-n_k} \right]^L \leq \sum_{i=1}^{Ln_{\max}} 1 = Ln_{\max} \quad (3.2.6)$$

$$\sum_{k=1}^K D^{-n_k} \leq (Ln_{\max})^{1/L} \quad (3.2.7)$$

Equation 3.2.7 must be valid for all positive integers  $L$ , and taking the limit as  $L \rightarrow \infty$ , we get (3.2.3), completing the proof. |

Since the code-word lengths for any uniquely decodable code satisfy (3.2.3), and since we can construct a prefix condition code for any set of lengths satisfying (3.2.3), any uniquely decodable code can be replaced by a prefix condition code without changing any of the code-word lengths. Thus the subsequent theorems concerning average code-word length apply equally to uniquely decodable codes and the subclass of prefix condition codes.

\* This theorem is due to McMillan (1956). The proof given is due to Karush (1961).

### 3.3 A Source Coding Theorem

**Theorem 3.3.1.** Given a finite source ensemble  $U$  with entropy  $H(U)$  and given a code alphabet of  $D$  symbols, it is possible to assign code words to the source letters in such a way that the prefix condition is satisfied and the average length of the code words,  $\bar{n}$ , satisfies

$$\bar{n} < \frac{H(U)}{\log D} + 1 \quad (3.3.1)$$

Furthermore, for any uniquely decodable set of code words,

$$\bar{n} \geq \frac{H(U)}{\log D} \quad (3.3.2)$$


---

*Proof.* First we establish (3.3.2) by showing that

$$H(U) - \bar{n} \log D \leq 0 \quad (3.3.3)$$

Let  $P(a_1), \dots, P(a_K)$  be the probabilities of the source letters and let  $n_1, \dots, n_K$  be the code-word lengths.

$$H(U) - \bar{n} \log D = \sum_{k=1}^K P(a_k) \log \frac{1}{P(a_k)} - \sum_{k=1}^K P(a_k) n_k \log D \quad (3.3.4)$$

Taking  $-n_k$  inside the logarithm and combining terms, we have

$$H(U) - \bar{n} \log D = \sum_{k=1}^K P(a_k) \log \frac{D^{-n_k}}{P(a_k)} \quad (3.3.5)$$

Using the inequality  $\log Z \leq (Z - 1) \log e$  for  $Z > 0$ ,

$$H(U) - \bar{n} \log D \leq (\log e) \left[ \sum_{k=1}^K D^{-n_k} - \sum_{k=1}^K P(a_k) \right] \leq 0 \quad (3.3.6)$$

The last inequality in (3.3.6) follows from the Kraft inequality, (3.2.3), which is valid for any uniquely decodable code. This verifies (3.3.2). Notice that equality holds in (3.3.2) if and only if

$$P(a_k) = D^{-n_k}; \quad 1 \leq k \leq K \quad (3.3.7)$$

This is the previously derived condition for each code letter to have the maximum entropy, (3.2.2).

Next, we show how to choose a code satisfying (3.3.1). If the code-word lengths did not have to be integers, we could simply choose the  $n_k$  to satisfy (3.3.7). We can approximately satisfy (3.3.7), however, by choosing  $n_k$  to be the integer satisfying

$$D^{-n_k} \leq P(a_k) < D^{-n_k+1}; \quad 1 \leq k \leq K \quad (3.3.8)$$

Summing (3.3.8) over  $k$ , the left-hand inequality becomes the Kraft inequality, (3.2.3), and a prefix condition code exists with these lengths. Taking the logarithm of the right-hand side of (3.3.8), we obtain

$$\begin{aligned} \log P(a_k) &< (-n_k + 1) \log D \\ n_k &< \frac{-\log P(a_k)}{\log D} + 1 \end{aligned} \quad (3.3.9)$$

Multiplying (3.3.9) by  $P(a_k)$  and summing over  $k$ , we obtain (3.3.1), completing the proof. |

We can achieve stronger results if, instead of providing code words for individual source letters, we provide code words directly for sequences of  $L$  source letters.

**Theorem 3.3.2.** Given a discrete memoryless source  $U$  with entropy  $H(U)$ , and given a code alphabet of  $D$  symbols, it is possible to assign code words to sequences of  $L$  source letters in such a way that the prefix condition is satisfied and the average length of the code words per source letter,  $\bar{n}$ , satisfies

$$\frac{H(U)}{\log D} \leq \bar{n} < \frac{H(U)}{\log D} + \frac{1}{L} \quad (3.3.10)$$

Furthermore, the left-hand inequality must be satisfied for any uniquely decodable set of code words.

---

*Proof.* Consider the product ensemble of sequences of  $L$  source letters. The entropy of the product ensemble is  $LH(U)$  and the average length of the code words is  $L\bar{n}$ , using  $\bar{n}$  as the average length per source letter. Then Theorem 3.3.1 states that the minimum achievable  $\bar{n}L$ , assigning a variable-length code word to each sequence of  $L$  source letters, satisfies

$$\frac{LH(U)}{\log D} \leq \bar{n}L < \frac{LH(U)}{\log D} + 1$$

Dividing by  $L$ , the theorem is proved. |

Theorem 3.3.2 is very similar to Theorem 3.1.1. If we take  $L$  arbitrarily large and then apply the law of large numbers to a long string of sequences each of  $L$  source letters, we see that Theorem 3.3.2 implies the first part of Theorem 3.1.1. Theorem 3.3.2 is a little stronger than Theorem 3.1.1, however, since it suggests a relatively simple encoding technique and also suggests an alternative to making occasional errors, namely, occasional long delays.

### 3.4 An Optimum Variable-Length Encoding Procedure

In this section, we shall give a constructive procedure, discovered by D. A. Huffman (1952), for finding an optimum set of code words to represent a given set of messages. By optimum, we mean that no other uniquely decodable set of code words has a smaller average code-word length than the given set. The set of lengths given by (3.3.6) does not usually minimize  $\bar{n}$ , even though it achieves the bound of Theorem 3.3.1. We consider binary codes first and then generalize to an arbitrary code alphabet.

Let the source letters,  $a_1, \dots, a_K$  have probabilities  $P(a_1), \dots, P(a_K)$ , and assume for simplicity of notation that the letters are ordered so that  $P(a_1) \geq P(a_2) \geq \dots \geq P(a_K)$ . Let  $\mathbf{x}_1, \dots, \mathbf{x}_K$  be a set of binary code words for this source and let  $n_1, \dots, n_K$  be the code-word lengths. The code words  $\mathbf{x}_1, \dots, \mathbf{x}_K$  corresponding to an optimum code, in general, will not be unique and often the lengths  $n_1, \dots, n_K$  will not be unique (see Problem 3.13). In what follows, we shall impose some conditions that must be satisfied by at least one optimum code, and then show how to construct the code using those conditions. We restrict our attention to prefix condition codes since any set of lengths achievable in a uniquely decodable code is achievable in a prefix condition code.

---

**LEMMA 1:** For any given source with  $K \geq 2$  letters, an optimum binary code exists in which the two least likely code words,  $\mathbf{x}_K$  and  $\mathbf{x}_{K-1}$ , have the same length and differ in only the last digit,  $\mathbf{x}_K$  ending in a **1** and  $\mathbf{x}_{K-1}$  in a **0**.

---

*Proof.* First observe that, for at least one optimum code,  $n_K$  is greater than or equal to each of the other word lengths. To see this, consider a code in which  $n_K < n_i$  for some  $i$ . If the code words  $\mathbf{x}_i$  and  $\mathbf{x}_K$  are interchanged, then the change in  $\bar{n}$  is

$$\begin{aligned}\Delta &= P(a_i)n_K + P(a_K)n_i - P(a_i)n_i - P(a_K)n_K \\ &= [P(a_i) - P(a_K)][n_K - n_i] \leq 0\end{aligned}\tag{3.4.1}$$

Thus any code can be modified to make  $n_K$  the maximum length without increasing  $\bar{n}$ . Next observe that, in any optimum code for which  $n_K$  is the largest length, there must be another code word differing from  $\mathbf{x}_K$  in only the last digit, for otherwise the last digit of  $\mathbf{x}_K$  could be dropped without violating the prefix condition. Finally, if  $\mathbf{x}_i$  is the code word differing from  $\mathbf{x}_K$  in only one position, we must have  $n_i \geq n_{K-1}$ , and by the argument in (3.4.1),  $\mathbf{x}_i$  and  $\mathbf{x}_{K-1}$  can be interchanged without increasing  $\bar{n}$ . Thus there is an optimum code in which  $\mathbf{x}_K$  and  $\mathbf{x}_{K-1}$  differ in only the last digit. These words can be interchanged, if necessary, to make  $\mathbf{x}_K$  end in a **1**. |

With this lemma, we have reduced the problem of constructing an optimum

code to that of constructing  $\mathbf{x}_1, \dots, \mathbf{x}_{K-2}$  and finding the first  $n_K - 1$  digits of  $\mathbf{x}_K$ . Now define the reduced ensemble  $U'$  as an ensemble consisting of letters  $a'_1, a'_2, \dots, a'_{K-1}$  with the probabilities

$$\Pr(a'_k) = \begin{cases} P(a_k); & k \leq K-2 \\ P(a_{K-1}) + P(a_K); & k = K-1 \end{cases} \quad (3.4.2)$$

Any prefix condition code for  $U'$  can be changed into a corresponding prefix condition code for  $U$  simply by adding a terminal **0** to  $\mathbf{x}'_{K-1}$  to generate  $\mathbf{x}_{K-1}$  and adding a terminal **1** to  $\mathbf{x}'_{K-1}$  to generate  $\mathbf{x}_K$ . This sets up a one to one correspondence between the set of prefix condition codes for  $U'$  and the set of those prefix condition codes for  $U$  in which  $\mathbf{x}_K$  and  $\mathbf{x}_{K-1}$  differ only in the terminal digit, **1** for  $\mathbf{x}_K$  and **0** for  $\mathbf{x}_{K-1}$ .

**LEMMA 2:** If a prefix condition code is optimum for  $U'$ , then the corresponding prefix condition code for  $U$  is optimum.

---

*Proof.* The lengths  $n'_k$  of the code words for  $U'$  are related to the lengths  $n_k$  of the corresponding code for  $U$  by

$$n_k = \begin{cases} n'_k; & k \leq K-2 \\ n'_{K-1} + 1; & k = K-1, \quad k = K \end{cases} \quad (3.4.3)$$

Thus the average lengths  $\bar{n}'$  and  $\bar{n}$  are related by

$$\begin{aligned} \bar{n} &= \sum_{k=1}^K P(a_k)n_k = \sum_{k=1}^{K-2} P(a_k)n'_k + [P(a_{K-1}) + P(a_K)](n'_{K-1} + 1) \\ &= \sum_{k=1}^{K-2} \Pr(a'_k)n'_k + \Pr(a'_{K-1})(n'_{K-1} + 1) \\ &= \bar{n}' + \Pr(a'_{K-1}) \end{aligned} \quad (3.4.4)$$

Since  $\bar{n}$  and  $\bar{n}'$  differ by a fixed amount independent of the code for  $U'$ , we can minimize  $\bar{n}$  over the class of codes where  $\mathbf{x}_K$  and  $\mathbf{x}_{K-1}$  differ in only the last digit by minimizing  $\bar{n}'$ . But by Lemma 1, a member of this class minimizes  $\bar{n}$  over all prefix condition codes. |

The problem of finding an optimum code has now been reduced to the problem of finding an optimum code for an ensemble with one fewer message. But now the reduced ensemble can have its least two probable messages grouped together and a further reduced ensemble generated. Continuing, we eventually reach the point where we have an ensemble of only 2 messages, and then an optimum code is clearly generated by assigning **1** to one message and **0** to the other.

A systematic procedure for carrying out the above operations is demonstrated in Figure 3.4.1. First, we tie together the two least probable messages,

$a_4$  and  $a_5$  in this case, assigning **0** as the last digit for  $a_4$  and **1** as the last digit for  $a_5$ . Adding the probabilities for  $a_4$  and  $a_5$ , we find that the two least likely messages in the reduced ensemble are  $a_3$  and  $a_4'$ . On the next stage, the two least likely messages are  $a_1$  and  $a_2$ , and at this point only two messages remain. Looking at the resulting figure, we see that we have constructed a code tree for  $U$ , starting at the outermost branches and working down to the trunk. The code words are read off the tree from right to left.

There is a complication that arises in extending this procedure to non-binary codes. Lemma 1 still applies for a nonbinary code alphabet. Lemma 2 does not apply, however, the problem being whether more code words than just  $\mathbf{x}_{K-1}$  should differ from  $\mathbf{x}_K$  in the last digit.

| Code Word  | Message | $\Pr(a_k)$ |  |
|------------|---------|------------|--|
| <b>00</b>  | $a_1$   | 0.3        |  |
| <b>01</b>  | $a_2$   | 0.25       |  |
| <b>10</b>  | $a_3$   | 0.25       |  |
| <b>110</b> | $a_4$   | 0.10       |  |
| <b>111</b> | $a_5$   | 0.10       |  |

Figure 3.4.1. Huffman coding procedure.

Define a complete code tree as a finite code tree in which each intermediate node has  $D$  nodes of the next higher order stemming from it. It can be seen from the proof of the Kraft inequality that a complete code tree is one where the Kraft inequality is satisfied with equality.

LEMMA 3: The number of terminal nodes in a complete code tree with alphabet size  $D$  must be of the form  $D + m(D - 1)$  for some integer  $m$ .

*Proof.* The smallest complete tree of alphabet size  $D$  contains  $D$  terminal nodes. If one of these terminal nodes is converted into an intermediate node,  $D$  new terminal nodes are generated and one is lost for a net gain of  $D - 1$ . Since any complete tree can be built up by such successive conversions of terminal nodes into intermediate nodes, and since each such conversion increases the number of nodes by  $D - 1$ , the final number of nodes must have the form  $D + m(D - 1)$ . |

We shall now adopt the convention of regarding every code tree as a complete tree with perhaps some number  $B$  of unused terminal nodes added to complete the tree. It is clear that, for an optimum code, all the unused terminal nodes must have the same length as the longest code word. Also, by interchanging used and unused terminal nodes, all the unused terminal

nodes can be arranged to differ in only the last digit. Thus, an optimum code tree must have, at most,  $D - 2$  unused terminal nodes since, if  $D - 1$  unused nodes were grouped together, the accompanying code word could be shortened without violating the prefix condition. Since the number of code words plus the number of unused nodes has the form  $D + m(D - 1)$ , this uniquely specifies the number of unused nodes in the complete tree. For example, if  $D = 3$ , every complete tree has an odd number of nodes. If  $K$  is even, then the number of unused terminal nodes,  $B$ , for an optimum code is 1, and if  $K > 2$  is odd,  $B = 0$ .

For the more formula-oriented reader, we can obtain an explicit expression for  $B$  by observing that  $B + K = m(D - 1) + D$ . Equivalently,  $K - 2 = m(D - 1) + (D - 2 - B)$ . For an optimum code  $0 \leq B \leq D - 2$ , and

| Code Word | Message | $\Pr(a_k)$ |   |
|-----------|---------|------------|---|
| 0         | $a_1$   | 0.4        | 0 |
| 1         | $a_2$   | 0.3        | 1 |
| 20        | $a_3$   | 0.2        | 0 |
| 21        | $a_4$   | 0.05       | 1 |
| 220       | $a_3$   | 0.03       | 0 |
| 221       | $a_6$   | 0.02       | 1 |

Figure 3.4.2. Huffman coding:  $D = 3$ .

therefore  $0 \leq D - 2 - B \leq D - 2$ . Thus  $D - 2 - B$  is the remainder upon dividing  $K - 2$  by  $D - 1$ , denoted  $R_{D-1}(K - 2)$ . Thus

$$B = D - 2 - R_{D-1}(K - 2)$$

By the same argument as in Lemma 1, an optimum code exists where the  $B$  unused nodes and the  $D - B$  least probable code word nodes differ in only the last digit. Thus the first step in the decoding procedure is to group together the least probable  $D - B = 2 + R_{D-1}(K - 2)$  nodes.

After this initial step, the construction of an optimum code proceeds as before. We form a reduced ensemble by combining the probabilities of the code words grouped together previously. It is easy to verify that the number of messages in the reduced ensemble is of the form  $D + m(D - 1)$ , and we make the  $D$  least likely of them differ in only the last digit. Proceeding in this way, we eventually have a reduced ensemble of  $D$  messages for which an optimum code is obvious. Figure 3.4.2 illustrates the construction for  $D = 3$ . Since  $K = 6$ , the number of messages grouped together initially is  $2 + R_2(6 - 2) = 2$ .

### 3.5 Discrete Stationary Sources

The previous sections have been restricted to discrete memoryless sources. In this section, we consider the effect of statistical dependence between source letters.

Let  $\mathbf{u} = (\dots, u_{-1}, u_0, u_1, \dots)$  denote a sequence of letters produced by the source, where each letter  $u_i$  is a selection from a discrete alphabet. A complete probabilistic description of the source is given by the probabilities  $\Pr(u_{j+1}, u_{j+2}, \dots, u_{j+L})$ , specified for all sequence lengths  $L$ , all starting points  $j$ , and all sequences  $u_{j+1}, \dots, u_{j+L}$ . A source, in this generality, is nothing more and nothing less than an arbitrary discrete random process.

A discrete source is defined to be *stationary* if its probabilistic description is independent of a time origin. More formally, a source is stationary if

$$P_{U_1 U_2 \dots U_L}(u'_1, u'_2, \dots, u'_L) = P_{U_{j+1} U_{j+2} \dots U_{j+L}}(u'_1, \dots, u'_L) \quad (3.5.1)$$

for all lengths  $L$ , integers  $j$ , and sequences  $u'_1, \dots, u'_L$ . In words, the probability of the source producing a sequence  $\mathbf{u}'$  in the interval 1 to  $L$  is the same as the probability of its producing that sequence in the interval  $j + 1$  to  $j + L$ .

A discrete source is said to be periodic if (3.5.1) is satisfied for all  $j$  that are multiples of some integer  $m > 1$ . The period is the smallest  $m$  satisfying this test. If we consider  $m$ -tuples of letters from a periodic source of period  $m$  as “super letters” in a larger alphabet, then the sequence of super letters is stationary. For this reason, we shall consider only stationary sources in what follows.

Let  $\mathbf{u}_L = (u_1, \dots, u_L)$  be a sequence of  $L$  letters from a discrete stationary source and let  $U_1 U_2 \dots, U_L$  be the joint ensemble for  $\mathbf{u}_L$ . We now wish to define the entropy per source letter of a discrete stationary source. There are two approaches that we might take and, fortunately, they both give the same result. The first approach is to define the entropy per letter in a sequence of  $L$  letters as

$$H_L(U) = \frac{1}{L} H(U_1 U_2 \cdots U_L) = \overline{\frac{1}{L} \log \frac{1}{\Pr(\mathbf{u}_L)}} \quad (3.5.2)$$

We could then define the entropy per letter of the source as the limit of  $H_L(U)$  as  $L \rightarrow \infty$ . The second approach is to define the conditional entropy of the  $L$ th letter in a sequence given the first  $L - 1$  letters,  $H(U_L | U_1 \cdots U_{L-1})$ , and then define the entropy per letter of the source as

$$\lim_{L \rightarrow \infty} H(U_L | U_1 \cdots U_{L-1}).$$

The following theorem asserts, among other things, that both these limits exist and are equal.

**Theorem 3.5.1.** For a discrete stationary source with  $H_1(U) < \infty$ , we have

- (a)  $H(U_L | U_1 \cdots U_{L-1})$  is nonincreasing with  $L$
- (b)  $H_L(U) \geq H(U_L | U_1 \cdots U_{L-1})$  (3.5.3)
- (c)  $H_L(U)$  is nonincreasing with  $L$

$$(d) \lim_{L \rightarrow \infty} H_L(U) = \lim_{L \rightarrow \infty} H(U_L | U_1 \cdots U_{L-1}) \quad (3.5.4)$$


---

*Proof.* Using first the fact that conditioning cannot increase entropy (2.3.13), and then the stationarity of the source, we have, for  $L \geq 2$

$$\begin{aligned} H(U_L | U_1 \cdots U_{L-1}) &\leq H(U_L | U_2 \cdots U_{L-1}) \\ &= H(U_{L-1} | U_1 \cdots U_{L-2}) \end{aligned} \quad (3.5.5)$$

This establishes part (a).

Using the chain rule (2.2.30) to expand  $H_L(U)$ ,

$$H_L(U) = \frac{1}{L} [H(U_1) + H(U_2 | U_1) + \cdots + H(U_L | U_1 \cdots U_{L-1})] \quad (3.5.6)$$

From part (a), the last entropy term in (3.5.6) is a lower bound to each of the  $L$  entropy terms. Applying this bound, we have part (b).

Using the definition of  $H_L(U)$ , we have

$$\begin{aligned} H_L(U) &= \frac{1}{L} H(U_1 \cdots U_{L-1}) + \frac{1}{L} H(U_L | U_1 \cdots U_{L-1}) \\ &\leq \frac{L-1}{L} H_{L-1}(U) + \frac{1}{L} H_L(U) \end{aligned} \quad (3.5.7)$$

Rearranging terms, we have  $H_L(U) \leq H_{L-1}(U)$ , establishing part (c).

Since  $H_L(U)$  and  $H(U_L | U_1 \cdots U_{L-1})$  are both nonnegative and non-increasing with  $L$ , both limits must exist. Define  $H_\infty$  as  $\lim_{L \rightarrow \infty} H_L(U)$ . Using the chain rule again, we have

$$\begin{aligned} H_{L+j}(U) &= \frac{1}{L+j} H(U_1 \cdots U_{L-1}) + \frac{1}{L+j} [H(U_L | U_1 \cdots U_{L-1}) \\ &\quad + H(U_{L+1} | U_1 \cdots U_L) + \cdots + H(U_{L+j} | U_1 \cdots U_{L+j-1})] \\ &\leq \frac{1}{L+j} H(U_1 \cdots U_{L-1}) + \frac{j+1}{L+j} H(U_L | U_1 \cdots U_{L-1}) \end{aligned} \quad (3.5.8)$$

We have used the fact that the first term within brackets is an upper bound to each of the other terms. Taking the limit of (3.5.8) as  $j \rightarrow \infty$ .

$$H_\infty(U) \leq H(U_L | U_1 \cdots U_{L-1}) \quad (3.5.9)$$

Since (3.5.9) is valid for all  $L$ , we have

$$H_\infty(U) \leq \lim_{L \rightarrow \infty} (U_L | U_1 \cdots U_{L-1}) \quad (3.5.10)$$

Equations 3.5.10 and 3.5.3 together establish (3.5.4), completing the proof. |

**Theorem 3.5.2. (Variable-Length Source Coding Theorem).** Let  $H_L(U)$  be the entropy per letter in a sequence of length  $L$  for a discrete source of alphabet size  $K$ . Given a code alphabet of  $D$  symbols, it is possible to encode sequences of  $L$  source letters into a prefix condition code in such a way that the average number of code letters per source letter,  $\bar{n}$ , satisfies

$$\frac{H_L(U)}{\log D} \leq \bar{n} < \frac{H_L(U)}{\log D} + \frac{1}{L} \quad (3.5.11)$$

Furthermore, the left-hand inequality must be satisfied for any uniquely decodable set of code words for the sequences of  $L$  source letters. Finally, for any  $\delta > 0$ , if the source is stationary it is possible to choose  $L$  large enough so that  $\bar{n}$  satisfies

$$\frac{H_\infty(U)}{\log D} \leq \bar{n} < \frac{H_\infty(U)}{\log D} + \delta \quad (3.5.12)$$

and  $\bar{n}$  can never violate the left-hand inequality for any uniquely decodable code.

---

*Proof.* The proof of (3.5.11) is the same as that of Theorem 3.3.2, except that  $LH_L(U)$  is the entropy of a sequence of  $L$  source letters rather than  $LH(U)$ . Taking the limit in (3.5.11) as  $L \rightarrow \infty$ ,  $H_L(U)$  approaches  $H_\infty(U)$  and  $1/L$  approaches 0, establishing (3.5.12). |

In our discussion of discrete memoryless sources, our interest in  $\bar{n}$  was motivated by the law of large numbers, which asserted that the number of code letters per source letter in a long sequence of code words approaches  $\bar{n}$ . The following example shows that this limiting behavior need not hold for arbitrary discrete stationary sources. Suppose that a source with the alphabet  $(a_1, a_2, a_3)$  has two modes of behavior, each occurring with probability  $1/2$ . In the first mode the source produces an infinite sequence of repetitions of  $a_1$ . In the second mode, the source produces an infinite sequence of statistically independent, equiprobable selections of the letters  $a_2$  and  $a_3$ . If we encode sequences of  $L$  source letters into a binary code, it is not hard to see that  $\bar{n}$  is minimized by mapping the sequence of  $a_1$ 's into a single binary digit and mapping each of the  $2^L$  sequences of  $a_2$ 's and  $a_3$ 's into code words of length  $L + 1$ . Since the mode of the source never changes, either all code words in a sequence will be of length 1 or all will be of length  $L + 1$ . For such sources, neither  $\bar{n}$  nor the entropy are quantities of any great significance.

Sources that cannot be separated into different persisting modes of behavior are known as ergodic sources. To define ergodicity carefully, let  $\mathbf{u} = \dots, u_{-1}, u_0, u_1, \dots$  be an infinite length sequence of source letters and let  $T^l\mathbf{u}$  denote the sequence  $\mathbf{u}$  shifted in time by  $l$  positions. That is, if we denote  $T^l\mathbf{u}$  by  $\mathbf{u}'$  we have

$$u_n' = u_{n+l}; \quad -\infty < n < \infty$$

Likewise, if  $S$  is a set of infinite length sequences of source letters,  $T^lS$  denotes the same set shifted by  $l$  positions. That is, if  $\mathbf{u}' = T^l\mathbf{u}$ , then  $\mathbf{u}'$  is in the set  $T^lS$  iff  $\mathbf{u}$  is in  $S$ . A set of sequences is said to be *invariant* if  $TS = S$ . It can easily be seen that the set of all sequences from a discrete source is invariant, and also that for any  $\mathbf{u}$ , the set  $\dots, T^{-1}\mathbf{u}, \mathbf{u}, T\mathbf{u}, T^2\mathbf{u}, \dots$  is invariant. A discrete stationary source is defined to be *ergodic* if every measurable, invariant set of sequences has either probability one or probability zero. It can be seen that, in the previous example, the set of sequences in each of the modes of behavior referred to are invariant sets, each with probability  $\frac{1}{2}$ . Thus that source is nonergodic.

The above definition, although elegant, is sometimes difficult to work with and does not bring out the intuitive concept of ergodicity. An equivalent definition is as follows. Let  $f_n(\mathbf{u})$  be a function of the infinite length source sequence  $\mathbf{u}$  which depends only on a finite sequence,  $u_1, \dots, u_n$ , of the source letters. Then a discrete stationary source is *ergodic* iff for all  $n \geq 1$  and all  $f_n(\mathbf{u})$  for which  $\overline{|f_n(\mathbf{u})|} < \infty$ , we have, for all source sequences  $\mathbf{u}$  except a set of probability 0,

$$\lim_{L \rightarrow \infty} \frac{1}{L} \sum_{l=1}^L f_n(T^l\mathbf{u}) = \overline{f_n(\mathbf{u})} \quad (3.5.13)$$

The class of functions in the definition can be expanded to all measurable functions  $f(\mathbf{u})$  for which  $\overline{|f(\mathbf{u})|} < \infty$ , or can be restricted to the special class of functions  $f_{\mathbf{u}_n'}(\mathbf{u})$  where  $\mathbf{u}_n'$  is a fixed sequence  $u_1', \dots, u_n'$  of letters and

$$f_{\mathbf{u}_n'}(\mathbf{u}) = \begin{cases} 1 & \text{if } u_1 = u_1', u_2 = u_2', \dots, u_n = u_n' \\ 0 & \text{otherwise} \end{cases} \quad (3.5.14)$$

For a proof of the equivalence of these definitions, see Khinchin (1957), pp. 49–54, and for yet another equivalent definition, see Wolfowitz (1961), Lemma 10.3.1.

The definition of (3.5.13) is particularly important since it is the result that we shall need for ergodic sources. What it says is that the law of large numbers applies to ergodic sources. Alternatively, it says that the time average, averaged in time over any sample source output (except a set of zero probability), is equal to the ensemble average  $\overline{f_n(\mathbf{u})}$ . Since  $f_{\mathbf{u}_n'}(\mathbf{u})$  is just the

probability of sequence  $\mathbf{u}_n'$ , (3.5.14) states that the relative frequency of occurrence of  $\mathbf{u}_n'$  in a very long source sequence will be approximately equal to the probability of  $\mathbf{u}'$ .

Unfortunately, not even ergodicity quite implies that the number of code letters per source letter in a variable length code approaches  $\bar{n}$ . If we encode  $L$  source letters at a time and let  $n(u_1, \dots, u_L)$  be the length of a code word, then the time average number of code letters per source letter is given by

$$\lim_{J \rightarrow \infty} \frac{1}{JL} \sum_{j=0}^{J-1} n(u_{Lj+1}, \dots, u_{Lj+L}) \quad (3.5.15)$$

See Problem 3.21 for an example of an ergodic source where this average, as a random variable, takes on different values with nonzero probability. The difficulty is that (3.5.15) is not a time average in the same sense as (3.5.13) since it is defined in terms of shifts of  $L$  letters at a time rather than shifts of one letter at a time.

Fortunately, Theorem 3.1.1 does apply to arbitrary ergodic sources. The major difficulty in proving this lies in establishing a law of large numbers for self-information; that is, in showing that  $I(\mathbf{u}_L)/L$  is, with high probability, close to  $H_\infty(U)$  for large  $L$ . This law of large numbers is of considerable mathematical and information theoretic interest and we now state it as a theorem.

**Theorem 3.5.3 (McMillan (1953)).** Let a discrete stationary ergodic source have  $H_1(U) < \infty$ .

For arbitrary  $\epsilon > 0$ ,  $\delta > 0$ , there exists an integer  $L_o(\epsilon, \delta)$  (which depends upon the source) such that for all  $L \geq L_o(\epsilon, \delta)$

$$\Pr \left[ \left| \frac{I(\mathbf{u}_L)}{L} - H_\infty(U) \right| > \delta \right] < \epsilon \quad (3.5.16)$$


---

Before proving the theorem, we shall develop some necessary notation and prove two lemmas. We observe that

$$\frac{I(\mathbf{u}_L)}{L} = \frac{1}{L} \sum_{l=1}^L I(u_l \mid u_1, \dots, u_{l-1}) \quad (3.5.17)$$

Notice that the right-hand side of (3.5.17) closely resembles a time average as given in (3.5.13). The difference is that each self-information term in (3.5.17) depends on a different number of previous source digits. The point of the proof is to show that this dependence dies out sufficiently quickly as  $l$  becomes large. Let  $P(\mathbf{u}_L) = P(u_1, \dots, u_L)$  denote the probability assignment on a sequence of  $L$  source digits, and define  $Q_m(\mathbf{u}_L)$ , for any integer  $1 \leq m \leq L$  by

$$Q_m(\mathbf{u}_L) = P(\mathbf{u}_m) \prod_{l=m+1}^L P(u_l \mid u_{l-1}, \dots, u_{l-m}) \quad (3.5.18)$$

In other words,  $Q_m(\mathbf{u}_L)$  is an approximation to the probability measure on the source, taking into account only statistical dependencies  $m$  letters into the past. Notice that

$$\sum_{\mathbf{u}_L} Q_m(\mathbf{u}_L) = 1$$

This can be seen by summing first over  $\mathbf{u}_L$ , then  $\mathbf{u}_{L-1}$ , and so forth to  $\mathbf{u}_1$ .

LEMMA 1. For a discrete stationary ergodic source with  $H_1(U) < \infty$ , and for arbitrary  $m \geq 1$ ,

$$\lim_{L \rightarrow \infty} \frac{1}{L} \log Q_m(\mathbf{u}_L) = -H(U_{m+1} | U_m \dots U_1) \quad (3.5.19)$$

with probability 1.

*Proof.* From (3.5.18).

$$\frac{1}{L} \log Q_m(\mathbf{u}_L) = \frac{1}{L} \log P(\mathbf{u}_m) + \frac{1}{L} \sum_{l=m+1}^L \log P(u_l | u_{l-1}, \dots, u_{l-m}) \quad (3.5.20)$$

Since  $\log P(\mathbf{u}_m)$  is independent of  $L$  and finite with probability 1,

$$\lim_{L \rightarrow \infty} \frac{1}{L} \log P(\mathbf{u}_m) = 0 \quad (3.5.21)$$

with probability 1. Also, since  $\log P(u_l | u_{l-1}, \dots, u_{l-m})$  is a function of a sequence of  $m$  letters and has the finite expectation  $-H(U_{m+1} | U_m, \dots, U_1)$ , and since the source is ergodic, (3.5.13) gives us

$$\lim_{L \rightarrow \infty} \frac{1}{L-m} \sum_{l=m+1}^L \log P(u_l | u_{l-1}, \dots, u_{l-m}) = -H(U_{m+1} | U_m \dots U_1) \quad (3.5.22)$$

with probability 1. Finally, since  $m$  is fixed, the limit in (3.5.22) is unaffected by replacing  $1/(L-m)$  with  $1/L$ . Thus combining (3.5.20) to (3.5.22), we obtain (3.5.19). |

LEMMA 2. For a discrete, stationary, ergodic source with  $H_1(U) < \infty$ , for arbitrary  $\epsilon > 0$ ,  $\delta > 0$ , and for sufficiently large  $m$ , and any  $L > m$ ,

$$\Pr\left\{\left|\frac{1}{L} \log Q_m(\mathbf{u}_L) - \frac{1}{L} \log P(\mathbf{u}_L)\right| > \epsilon\right\} \leq \delta \quad (3.5.23)$$

*Proof.*

$$\begin{aligned} \Pr\left\{\left|\frac{1}{L} \log Q_m(\mathbf{u}_L) - \frac{1}{L} \log P(\mathbf{u}_L)\right| > \epsilon\right\} &= \Pr\left\{\left|\log \frac{Q_m(\mathbf{u}_L)}{P(\mathbf{u}_L)}\right| > L\epsilon\right\} \\ &\leq \frac{1}{L\epsilon} \overline{\left|\log \frac{Q_m(\mathbf{u}_L)}{P(\mathbf{u}_L)}\right|} \end{aligned} \quad (3.5.24)$$

where in (3.5.24) we have applied the Chebyshev inequality\* to the non-negative random variable

$$\left| \log \frac{Q_m(\mathbf{u}_L)}{P(\mathbf{u}_L)} \right|.$$

Now for any number  $y$ , let  $[y]_+$  denote the “positive part” of  $y$ ,

$$[y]_+ = \begin{cases} y; & y \geq 0 \\ 0; & y < 0 \end{cases} \quad (3.5.25)$$

It is then easy to verify, by considering positive and negative  $y$  separately, that

$$|y| = 2[y]_+ - y \quad (3.5.26)$$

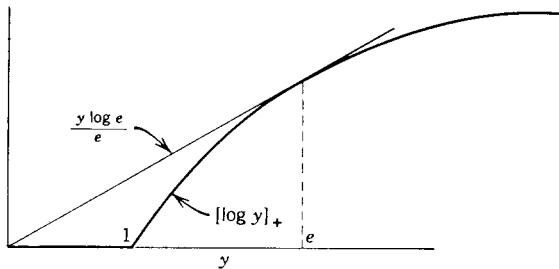


Figure 3.5.1.

It can be also be seen from Figure 3.5.1 that, for  $y \geq 0$ ,

$$[\log y]_+ \leq \frac{y \log e}{e} \quad (3.5.27)$$

Combining (3.5.26) and (3.5.27),

$$\left| \log \frac{Q_m(\mathbf{u}_L)}{P(\mathbf{u}_L)} \right| \leq \frac{2 \log e}{e} \left[ \frac{Q_m(\mathbf{u}_L)}{P(\mathbf{u}_L)} \right] - \log \frac{Q_m(\mathbf{u}_L)}{P(\mathbf{u}_L)} \quad (3.5.28)$$

The right-hand terms in (3.5.28) are evaluated as

$$\left[ \frac{Q_m(\mathbf{u}_L)}{P(\mathbf{u}_L)} \right] = \sum_{\mathbf{u}_L} P(\mathbf{u}_L) \frac{Q_m(\mathbf{u}_L)}{P(\mathbf{u}_L)} = 1 \quad (3.5.29)$$

$$-\log \frac{Q_m(\mathbf{u}_L)}{P(\mathbf{u}_L)} = (m)H_m(U) + (L-m)H(U_{m+1} | U_m \dots U_1) - LH_L(U) \quad (3.5.30)$$

\* See (5.4.5) for a derivation of the Chebyshev inequality.

Substituting (3.5.28) to (3.5.30) into (3.5.24), we obtain

$$\begin{aligned} \Pr\left[\left|\frac{1}{L} \log Q_m(\mathbf{u}_L) - \frac{1}{L} \log P(\mathbf{u}_L)\right| > \epsilon\right] &\leq \frac{1}{\epsilon} \left[ \frac{2 \log e}{Le} + \frac{m}{L} H_m(U) \right] \\ &\quad + \left(1 - \frac{m}{L}\right) H(U_{m+1} \mid U_m \cdots U_1) - H_L(U) \end{aligned} \quad (3.5.31)$$

Finally, we observe from Theorem 3.5.1 that the term in brackets on the right-hand side of (3.5.31) with  $L > m$  goes to zero with increasing  $m$ , uniformly with  $L > m$ . Thus, for any fixed  $\epsilon$ , the right-hand side is smaller than  $\delta$  for large enough  $m$ . |

*Proof of Theorem.* For given  $\epsilon > 0$ ,  $\delta > 0$ , pick  $m$  large enough so that the right-hand side of (3.5.31) is less than  $\delta$  for all  $L > m$  and also large enough that

$$|H(U_{m+1} \mid U_m \cdots U_1) - H_\infty(U)| < \epsilon \quad (3.5.32)$$

Then pick  $L_o > m$  large enough so that

$$\Pr\left[\left|\frac{1}{L} \log Q_m(\mathbf{u}_L) + H(U_{m+1} \mid U_m \cdots U_1)\right| > \epsilon\right] \leq \delta \quad (3.5.33)$$

for all  $L \geq L_o$ . This is possible from Lemma 1. Combining (3.5.31) to (3.5.33), we have, for  $L > L_o$ ,

$$\Pr\left[\left|\frac{1}{L} \log P(\mathbf{u}_L) + H_\infty(U)\right| > 3\epsilon\right] \leq 2\delta \quad (3.5.34)$$

To see this, observe that if neither the event on the left of (3.5.31) nor the event on the left of (3.5.33) occurs, then the event in (3.5.34) cannot occur. Thus the probability of the event in (3.5.34) is at most the sum of the probabilities of the events in (3.5.31) and (3.5.33). Since  $\epsilon > 0$  and  $\delta > 0$  are arbitrary, this is equivalent to (3.5.16), completing the proof. |

Theorem 3.1.1, using  $H_\infty(U)$  in place of  $H(U)$ , now follows as before, using Theorem 3.5.3 in place of (3.1.7).

### 3.6 Markov Sources

Some of the ideas of the last section can be brought out more clearly by discussing a special class of sources known as Markov sources. Such sources are characterized by a set of states, denoted by the integers  $1, \dots, J$ , and a source letter alphabet, denoted as usual by  $a_1, \dots, a_K$ . Each unit of time the source emits a letter and assumes a new state. The source sequence is denoted by  $\mathbf{u} = (u_1, u_2, \dots)$  and the state sequence by  $\mathbf{s} = (s_1, s_2, \dots)$ .

Let  $Q_{ji}$  denote the conditional probability of entering state  $i$  given that the previous state was  $j$ ,

$$Q_{ji} = \Pr(s_i = i \mid s_{i-1} = j) \quad (3.6.1)$$

Assume that the probability of entering a state depends only on the past state,

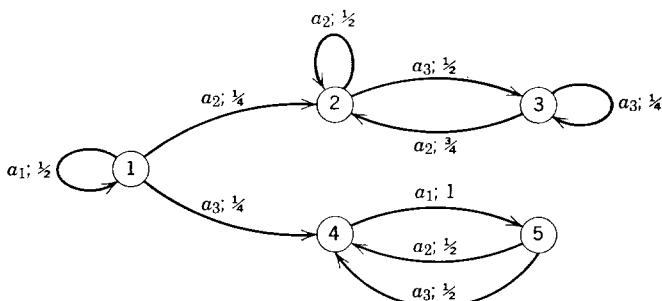
$$\Pr(s_l \mid s_{l-1}, s_{l-2}, \dots) = \Pr(s_l \mid s_{l-1}) \quad (3.6.2)$$

A random sequence of states obeying (3.6.1) and (3.6.2) is called a finite homogeneous Markov chain.

Let  $P_j(a_k)$  denote the probability that letter  $a_k$  is emitted when the source is in state  $j$ , and assume that this probability depends only on the current state.

$$P_j(a_k) = \Pr[u_i = a_k \mid s_i = j] \quad (3.6.3)$$

$$\Pr(u_i \mid s_i) = \Pr(u_i \mid s_i, u_{i-1}, s_{i-1}, \dots) \quad (3.6.4)$$



**Figure 3.6.1. Markov source; branches labeled by output  $a_k$  and probability  $P_j(a_k)$ .**

Finally, assume that the state of the source is uniquely determined by the previous state and previous letter.

Figure 3.6.1 illustrates the operation of a Markov source. The nodes correspond to states and the directed branches correspond to source letters and to transitions between states. Each branch directed away from a given node must correspond to a different letter to ensure that the new state is determined by the previous state and letter.

The state of a Markov source can be thought of as representing the effect of the history of the source on the next letter. For example, a stationary source for which each output depends statistically only on the  $l$  previous output letters can be represented as a Markov source, with a state corresponding to each possible sequence of  $l$  letters. From this example, it should be clear that most well-behaved stationary sources can at least be approximated by Markov sources.

It is instructive to consider modeling English text as a Markov source. Shannon (1948) gives sample outputs of such models, first letting letters

depend on the previous one or two letters and then letting words depend on the previous word with the appropriate probabilities. A part of this last sample follows:

The head and in frontal attack on an English writer that the character of this point is therefore another method. . . .

While this sample is clearly jibberish, it closely resembles meaningful text. We cannot carry attempts to model text as a random process too far, however. Clearly there are radical differences between the text in a medical dictionary and that in a child's first reader. Because of such differences, the source coding theorem does not apply to English text and we cannot precisely define its entropy. On the other hand, by using some of the statistical dependencies in English, we can represent text more efficiently than without making use of these dependencies.

We now summarize without proof some properties of finite homogeneous Markov chains.\* A state  $s$  is called *transient* if there is some state that can be reached from  $s$  in one or more transitions but from which it is impossible ever to return to  $s$ . For example, in Figure 3.6.1, state 1 is a transient state. A set of states is called *irreducible* if, from each state in the set, no state outside of the set can be reached and every state within the set can be reached in one or more transitions. For example, states 2 and 3 constitute an irreducible set and so do states 4 and 5.

The states in any finite homogeneous Markov chain can be uniquely separated into one or more irreducible sets of states and a set (perhaps empty) of transient states. With probability 1, the chain eventually gets to one of the irreducible sets and, of course, remains there.

Starting from any state  $s$  in an irreducible set, the number of transitions required for the first return to  $s$  is a random variable known as the recurrence time for  $s$ . The *period* of an irreducible set of states is the largest integer  $m$  such that all possible recurrence times for states in the set are multiples of  $m$ . For example, the period of the set of states 2 and 3 in Figure 3.6.1 is 1 since the recurrence time for each state can be any positive integer. The period of states 4 and 5 is 2 since the recurrence time must be 2 for each state. If an irreducible set has a period  $m \geq 2$ , it is called *periodic*. If  $m = 1$ , the set is called *ergodic*.

An ergodic set of states,  $E$ , has a set of limiting probabilities,  $q(j)$ , associated with it, given by the solution to the equations

$$\sum_{j \in E} q(j)Q_{ji} = q(i); \quad i \text{ in } E \quad (3.6.5)$$

$$\sum_{j \in E} q(j) = 1 \quad (3.6.6)$$

\* See Feller (1950) or Cox and Miller (1965), for example, for proofs.

Furthermore, for  $i$  and  $j$  in  $E$ ,

$$\lim_{l \rightarrow \infty} \Pr(s_l = i \mid s_1 = j) = q(i) \quad (3.6.7)$$

where the limit is approached exponentially in  $l$ .

It can be seen that the probabilities in (3.6.1) to (3.6.4) do not completely describe a source. What is needed is a statement of when the source starts and what the initial state probability assignment is. If the source is started in a given ergodic set of states infinitely far in the past, then

$$\Pr(s_l = i) = q(i); \quad \text{all } l \quad (3.6.8)$$

and the source is stationary and ergodic, according to the definitions in the last section. On the other hand, if the source is started at a given finite time in a given state, it is technically not stationary and ergodic since there is no past and since there is an initial transient in the probabilities.

The problem of transients is much more serious for periodic sets of states than for ergodic sets. For a periodic set of states with period  $m$ , there are  $m$  possible phases corresponding to whether a given state can occur only at the times  $\dots, -m, 0, m, \dots$  or at times  $\dots, -m+1, 1, m+1, \dots$ , or, and so forth, up to times  $\dots, -m+(m-1), m-1, m+(m-1), \dots$ . If the source is started infinitely far into the past in a given phase, then the resulting random state sequence is periodic, as defined in the last section. If the source is started infinitely far in the past with each phase equiprobable, then the resulting random state sequence satisfies (3.5.13) and is thus ergodic.

We now investigate the entropy of a Markov source. The entropy of a source output letter at a given time, given the current state of the source, is

$$H(U \mid s = j) = - \sum_{k=1}^K P_j(a_k) \log P_j(a_k) \quad (3.6.9)$$

Next, we find the entropy of a source output, given a particular state at some point in the past and given the intervening source outputs.

#### LEMMA

$$\begin{aligned} H(U_l \mid U_{l-1} U_{l-2} \cdots U_1, s_1 = j) \\ = \sum_{i=1}^J \Pr(s_l = i \mid s_1 = j) H(U \mid s = i) \end{aligned} \quad (3.6.10)$$


---

*Proof.* From the definition of a Markov source,  $s_2$ , the state at time 2, is uniquely specified by  $s_1$  and  $u_1$ , the previous state and output letter. Likewise,  $s_3$  is uniquely specified by  $s_2$  and  $u_2$ , and thus by  $u_2, u_1$ , and  $s_1$ . Proceeding by induction, we see that, for any positive  $l$ ,  $s_l$  is uniquely specified by  $u_1, \dots, u_{l-1}$ , and  $s_1$ . Thus

$$\Pr(u_l \mid u_1, \dots, u_{l-1}, s_1) = \Pr(u_l \mid s_l, u_1, \dots, u_{l-1}, s_1)$$

where  $s_l$  is taken as the state specified by  $u_1, \dots, u_{l-1}$  and  $s_1$ . Since  $u_l$  depends only on  $s_l$  (see 3.6.4), this becomes

$$\Pr(u_l \mid u_1, \dots, u_{l-1}, s_1) = \Pr(u_l \mid s_l) \quad (3.6.11)$$

Taking the logarithm of both sides of (3.6.11) and averaging over  $u_1, \dots, u_l$  and  $s_l$ , we have

$$\begin{aligned} \sum_{u_1, \dots, u_l, s_l} \Pr(u_1, \dots, u_l, s_l \mid s_1) \log \Pr(u_l \mid u_1, \dots, u_{l-1}, s_1) \\ = \sum_{u_1, \dots, u_l, s_l} \Pr(u_1, \dots, u_{l-1}, s_l \mid s_1) \Pr(u_l \mid s_l) \log \Pr(u_l \mid s_l) \end{aligned}$$

Summing the right-hand side over  $u_1, \dots, u_{l-1}$ , we have (3.6.10). |

Notice that the proof of this lemma depends critically upon the assumption that the state of the source is specified by the previous state and output letter. Calculating the left-hand side of (3.6.10) for sources not satisfying this condition is surprisingly tricky and difficult.\*

Any given probability assignment on the state  $s_1$  determines a probability assignment on the states at all future times and we can then average (3.6.9) over  $s_1$  to obtain

$$H(U_l \mid U_{l-1} \cdots U_1 S_1) = \sum_{i=1}^J \Pr(s_l = i) H(U \mid s = i) \quad (3.6.12)$$

For a stationary ergodic Markov source,  $\Pr(s_l = i)$  is independent of  $l$ , and from (3.6.8),

$$H(U_l \mid U_{l-1} \cdots U_1 S_1) = \sum_i q(i) H(U \mid s = i); \quad \text{all } l \geq 1. \quad (3.6.13)$$

Next, we consider the entropy per letter of a sequence of source letters given the initial state.

$$\frac{1}{L} H(U_1 \cdots U_L \mid S_1) = \frac{1}{L} \sum_{l=1}^L H(U_l \mid U_{l-1} \cdots U_1 S_1) \quad (3.6.14)$$

From (3.6.12), this is

$$\frac{1}{L} H(U_1 \cdots U_L \mid S_1) = \sum_{i=1}^J q_{(1,L)}(i) H(U \mid s = i) \quad (3.6.15)$$

where

$$q_{(1,L)}(i) = \frac{1}{L} \sum_{l=1}^L \Pr(s_l = i) \quad (3.6.16)$$

We see that  $q_{(1,L)}(i)$  is just the time average probability of being in state  $i$ .

\* Blackwell (1957) has considered this problem in the limit as  $L \rightarrow \infty$  and has shown that it can be "reduced" to solving a difficult integral equation.

For a stationary ergodic Markov source,  $q_{(1,L)}(i)$  is just  $q(i)$  as given by (3.6.5) and (3.6.6) and we have

$$\frac{1}{L} H(U_1 \cdots U_L \mid S_1) = \sum_{i=1}^J q(i) H(U \mid s = i) \quad (3.6.17)$$

In the limit as  $L \rightarrow \infty$ , we can define

$$q_{(1,\infty)}(i) = \lim_{L \rightarrow \infty} \frac{1}{L} \sum_{l=1}^L \Pr(s_l = i) \quad (3.6.18)$$

This limit will always exist, although in general it will depend upon the probability assignment on  $s_1$ . For example, in Figure 3.6.1, it clearly makes a difference whether  $s_1$  is given as state 2 with probability 1 or as state 4 with probability 1. On the other hand, for a Markov chain containing only one irreducible set of states,  $q_{(1,\infty)}(i)$  is independent of the initial probability assignment. With this definition of  $q_{(1,\infty)}$ , we have

$$\lim_{L \rightarrow \infty} \frac{1}{L} H(U_1 \cdots U_L \mid S_1) = \sum_{i=1}^J q_{(1,\infty)}(i) H(U \mid s = i) \quad (3.6.19)$$

We next consider the unconditional entropy per letter of the source sequence. We have

$$H(U_1 \cdots U_L) = I(S_1; U_1 \cdots U_L) + H(U_1 \cdots U_L \mid S_1)$$

The average mutual information above is bounded between 0 and  $\log J$ , and, therefore,

$$\lim_{L \rightarrow \infty} \frac{1}{L} H(U_1 \cdots U_L) = \lim_{L \rightarrow \infty} \frac{1}{L} H(U_1 \cdots U_L \mid S_1) \quad (3.6.20)$$

Defining the left-hand side of (3.6.20) as  $H_\infty(U)$ , (3.6.19) yields

$$H_\infty(U) = \sum_{i=1}^J q_{(1,\infty)}(i) H(U \mid s = i) \quad (3.6.21)$$

The following theorem summarizes our results.

**Theorem 3.6.1.** The entropy per letter of a Markov source is given by (3.6.21), where  $q_{(1,\infty)}(i)$  is given by (3.6.18), and  $H(U \mid s = i)$  is given by (3.6.9). If the Markov chain contains no more than one irreducible set of states, then  $q_{(1,\infty)}(i)$  is independent of the probability assignment on  $s_1$ , and if that irreducible set is ergodic,  $q_{(1,\infty)}(i) = q(i)$  as given by (3.6.5) and (3.6.6).

---

The discussion of variable-length source coding for discrete stationary sources applies directly to Markov sources, but there is a simplification possible for the Markov source. In (3.5.11), the average number of code

letters per source letter, encoding  $L$  source letters at a time, satisfied  $\bar{n} \geq H_L(U)/\log D$ . To get  $\bar{n}$  close to  $H_\infty(U)/\log D$ , it might be necessary to make  $L$  quite large. For stationary ergodic Markov sources, we shall see that, by using the state information and encoding  $L$  source letters at a time, we can achieve an  $\bar{n}$  satisfying

$$\frac{H_\infty(U)}{\log D} \leq \bar{n} < \frac{H_\infty(U)}{\log D} + \frac{1}{L} \quad (3.6.21a)$$

To achieve this result, we use a different code for each initial state. The length of the code word corresponding to sequence  $\mathbf{u} = (u_1, \dots, u_L)$  and  $s_1 = j$  can be chosen as in (3.3.6) to satisfy

$$D^{-n_j(\mathbf{u})} \leq \Pr(\mathbf{u} \mid s_1 = j) < D^{-n_j(\mathbf{u})+1} \quad (3.6.22)$$

As in Theorem 3.3.1, these lengths satisfy the Kraft inequality for each initial state, and the average code-word length  $\bar{n}_j L$  for a given initial state  $j$  satisfies

$$\frac{H(U_1 \cdots U_L \mid s_1 = j)}{\log D} \leq \bar{n}_j L < \frac{H(U_1 \cdots U_L \mid s_1 = j)}{\log D} + 1 \quad (3.6.23)$$

Averaging over the states and dividing by  $L$ , we obtain

$$\frac{H(U_1 \cdots U_L \mid S_1)}{L \log D} \leq \bar{n} < \frac{H(U_1 \cdots U_L \mid S_1)}{L \log D} + \frac{1}{L} \quad (3.6.24)$$

Using (3.6.17), this reduces to (3.6.21).

## Summary and Conclusions

We have had three objectives in this chapter: first, to bring out the significance of entropy; second, to develop some familiarity with manipulating sequences of events; and, third, to learn how to encode sources with the minimum average number of code letters per source letter. We found that the entropy  $H(U)$  of a discrete memoryless source could be interpreted in terms of sequences of  $L$  source letters for large  $L$ . Roughly,  $H(U)$  is interpreted as  $1/L$  times the logarithm of the number of “typical” source sequences, and minus  $1/L$  times the logarithm of the probability of a typical source sequence. Also,  $H(U)$  gives the minimum number of code letters per source letter,  $\bar{n}$ , required to represent the source output. For a fixed-length code, such an  $\bar{n}$  can be achieved only at the expense of a nonzero (but vanishing with  $L$ ) probability of getting a source sequence with no unique assigned code word. For a variable-length code, such an  $\bar{n}$  involves waiting time problems if source digits arrive at the encoder at a fixed rate and encoded digits are to leave the encoder at a fixed rate.

In Chapter 9, a much more general source coding problem will be considered where the source is not to be perfectly represented but only represented within a given fidelity criterion.

Another important source coding problem that we have not touched upon is how to generate codes with certain constraints on them. For example, there is a considerable body of literature on self-synchronizing codes. These are codes for which an infinite encoded sequence must be decodable starting in the middle of the sequence.

It should be remembered that, for most real sources, the central problems do not appear to be the information theoretic problems of representing the source, but the more subjective problems of determining what is worth representing in the output of the sources.

### **Historical Notes and References**

Alternate treatments of the topics here (particularly in the first four sections) are to be found in Abramson (1963), Fano (1961), Ash (1965), and Shannon (1948). The source coding theorem, Theorem 3.1.1, is due to Shannon (1948), as are most of the concepts in this chapter. Shannon also established the theorem for the case where the source letters have unequal durations and where the source is Markov. As pointed out in the text, the optimum code construction procedure in Section 2.4 is due to Huffman (1952). Source coding with the restriction of self-synchronization has been treated by Golomb, Gordon, and Welch (1958), Kendall and Reed (1962), Eastman (1965), Scholtz (1966), and others.

Theorem 3.5.3 is due to McMillan (1953) and is often called the AEP property of ergodic sources. McMillan proved  $L_1$  convergence, which is slightly stronger than the convergence established here, but his theorem is restricted to a source with a finite alphabet and his proof is far more complicated than that given here. Breiman (1957) subsequently proved convergence in probability for an ergodic source with a finite alphabet. Ott (1962) has given a procedure for encoding a more general type of Markov source than that discussed here, in which the source state need not be uniquely determined by the previous state and previous source letter.

## *Chapter 4*

### DISCRETE MEMORYLESS CHANNELS AND CAPACITY

In the preceding chapter we discussed the problem of representing the output of an information source by means of letters from a code alphabet. In so doing, we found a number of interpretations of self-information and entropy and also found a number of clear, simple results concerning source coding. We found that the difficulties in applying the theory lay not in its complexity but in the difficulty of representing real information sources by reasonable probabilistic models. In this and the succeeding chapters, we shall discuss transmitting information over noisy channels. In so doing, we shall acquire more insight into the nature of mutual information and find some deep and significant results concerning coding for noisy channels. We shall find that these results are not as simple as those for source coding, but are of far greater practical significance. This significance stems from the fact that reasonably simple and useful probabilistic models can be constructed for many real communication channels, and that applying the theory to these models gives us nontrivial insight into the problems of designing communication systems.

#### **4.1 Classification of Channels**

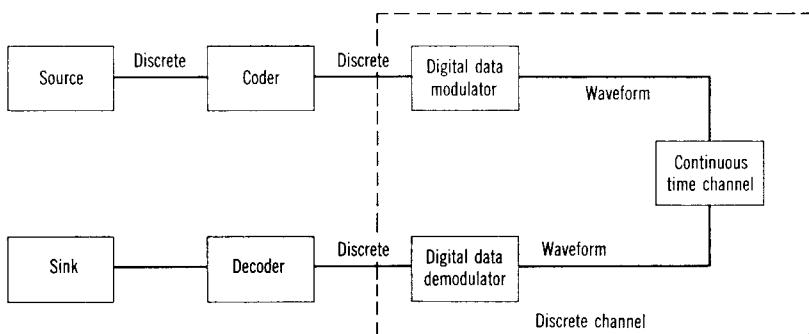
A transmission channel can be specified in terms of the set of inputs available at the input terminal, the set of outputs available at the output terminal, and for each input the probability measure on the output events conditional on that input.

The first kind of channel to be considered is the discrete memoryless channel. This is a channel for which the input and output are each sequences of letters from finite alphabets and for which the output letter at a given time depends statistically only on the corresponding input letter.

Another kind of channel, to be considered in Chapter 7, is the continuous amplitude, discrete-time memoryless channel. Here the input and output

are sequences of letters from alphabets consisting of the set of real numbers (or, more generally, real-valued vectors), and again the output letter at a given time depends statistically only on the input at the corresponding time. We can also consider channels in which the input is discrete and the output is continuous or vice versa, but these will turn out to be trivial variations.

Yet another kind of channel, to be considered in Chapter 8, is the continuous time channel in which the input and output are waveforms. Which of the above channel models to use in describing a given communication link is often a matter of choice. For example, in Figure 4.1.1, we can consider the



**Figure 4.1.1. Choice of channel model.**

channel to be either discrete or continuous in time. If we are interested primarily in the coder and decoder in Figure 4.1.1, then it is convenient to consider the digital data modulator and demodulator as being part of the channel, in which case the channel is discrete. On the other hand, if we are interested in the design of both the coder and the digital data modulator, or if we wish to consider both as one device, then the appropriate channel to consider is continuous.

In the last section of this chapter, we shall consider discrete channels with memory; these are channels in which the output at a given time depends statistically both on the current input and on prior inputs and outputs. Memory in discrete-channel models arises from a number of effects on real communication channels. One obvious effect is intersymbol interference caused by filtering in the channel. This causes one output symbol to be statistically related to several inputs. Another effect is fading. There is a temptation to think of fading channels as time-varying channels without memory. This viewpoint is somewhat misleading, however, since fading is usually best modeled as a statistical phenomenon and must be reflected in our probabilities of outputs given inputs. For example, in binary communication over a channel fading slowly relative to the bit rate, the errors will tend

to be bunched together by the fading rather than being statistically independent of each other. Thus the channel has memory in the sense that it remembers when it is performing poorly and tends to continue performing poorly for a certain duration of time.

## 4.2 Discrete Memoryless Channels

Consider a discrete memoryless channel (DMC) whose input alphabet  $X$  consists of the  $K$  integers  $0, 1, \dots, K - 1$  and whose output alphabet  $Y$  consists of the  $J$  integers  $0, 1, \dots, J - 1$ . Using integers for input and output letters simplifies our notation somewhat in what follows and also reemphasizes the fact that the names given to input and output letters are of no concern whatsoever.

The channel is specified by a transition probability assignment,  $P(j|k)$ , given for  $0 \leq j \leq J - 1$  and  $0 \leq k \leq K - 1$ . By definition,  $P(j|k)$  is the probability of receiving integer  $j$  given that integer  $k$  is the channel input.

We shall represent a sequence of  $N$  input letters to the channel as  $\mathbf{x} = (x_1, \dots, x_n, \dots, x_N)$

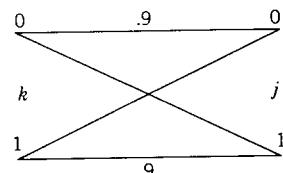
where each  $x_n$ ,  $1 \leq n \leq N$ , takes on values from the input alphabet, 0 to  $K - 1$ . Similarly, we represent the corresponding sequence of output letters as  $\mathbf{y} = (y_1, \dots, y_N)$  where each  $y_n$  takes on values from the output alphabet 0 to  $J - 1$ . The probability assignment on  $y_n$  conditioned on  $x_n$  is given by the previously described transition probability assignment,  $P(y_n|x_n)$ .

Since the channel is memoryless, each output letter in the sequence depends only on the corresponding input, and the probability of an output sequence  $\mathbf{y} = (y_1, \dots, y_N)$  given an input sequence  $\mathbf{x} = (x_1, \dots, x_N)$  is given by

$$P_N(\mathbf{y}|\mathbf{x}) = \prod_{n=1}^N P(y_n|x_n) \quad (4.2.1)$$

More formally, a channel is memoryless if there is a transition probability assignment,  $P(j|k)$ , such that (4.2.1) is satisfied for all  $N$ , all  $\mathbf{y} = (y_1, \dots, y_N)$  and all  $\mathbf{x} = (x_1, \dots, x_N)$ .

As an example of the notation, consider the binary symmetric channel in Figure 4.2.1. The transition probability assignment is given in Figure 4.2.1 as  $P(0/0) = 0.9$ ,  $P(1/0) = 0.1$ ,  $P(0/1) = 0.1$ ,  $P(1/1) = 0.9$ . For sequences of length 2, (4.2.1) yields  $P_2(00/00) = (0.9) \cdot (0.9) = 0.81$ ,  $P_2(10/00) = (0.1) \cdot (0.9) = 0.09$ , and so on.  $P_2(00/00)$  is simply the probability of receiving two zeros given that two zeros are transmitted.



*Figure 4.2.1. Transition probabilities for a binary symmetric channel.*

Observe that, in specifying a channel, nothing has been said about the way the inputs are to be used. If we assign a probability measure to the input integers, letting  $Q(k)$  be the probability of using integer  $k$ , then the average mutual information between input and output is

$$I(X;Y) = \sum_{k=0}^{K-1} \sum_{j=0}^{J-1} Q(k)P(j/k) \log \frac{P(j/k)}{\sum_{i=0}^{K-1} Q(i)P(j/i)} \quad (4.2.2)$$

We have written out the probability of receiving integer  $j$  as

$$\sum_i Q(i)P(j/i)$$

to emphasize that it is a function both of the input assignment and the channel transition probabilities.

Since the relative frequencies of the channel input letters can be adjusted by an encoder, it should not be surprising that the maximum of  $I(X;Y)$  over the input probabilities is a quantity of information theoretic significance. *The capacity  $C$  of a discrete memoryless channel (DMC) is defined as the largest average mutual information  $I(X;Y)$  that can be transmitted over the channel in one use, maximized over all input probability assignments.*

$$C \triangleq \max_{Q(0), \dots, Q(K-1)} \sum_{k,j} Q(k)P(j/k) \log \frac{P(j/k)}{\sum_i Q(i)P(j/i)} \quad (4.2.3)$$

Observe that, whereas  $I(X;Y)$  is a function of both the channel and the input assignment,  $C$  is a function only of the channel. The calculation of  $C$  involves a maximization over  $K$  variables with both inequality constraints,  $Q(k) \geq 0$ , and an equality constraint,  $\sum Q(k) = 1$ . The maximum value must exist since the function is continuous and the maximization is over a closed bounded region of vector space.\* We shall return to the computational problem of finding capacity in Sections 4.4 and 4.5.

The major significance of capacity for a DMC comes from the coding theorem, which states that data can be transmitted reliably over the channel at any rate below capacity. Notice that the surprising thing about the coding theorem is the word “reliable”. It is obvious that information can be transmitted at capacity, simply by using the appropriate input probability assignment. The coding theorem will be treated in the next chapter. We now show that capacity has the interpretation of being the maximum average mutual

\* See any text on mathematical analysis, for example, Buck (1956), p. 41. The reason for even questioning the existence of a maximum can be seen from attempting to maximize the function  $x^2$  over the open interval  $0 < x < 2$ . The function has no maximum since it gets arbitrarily close to 4 but never reaches 4.

information per letter that can be transmitted for a sequence of inputs and outputs. We then prove the converse to the coding theorem, that is, that reliable transmission is not possible at source rates above channel capacity.

**Theorem 4.2.1.** Let  $Q_N(\mathbf{x})$  be an arbitrary joint probability assignment on sequences of  $N$  inputs for a DMC. Let  $\mathbf{X}^N$  and  $\mathbf{Y}^N$  represent the ensembles of input and output sequences, and let  $X_1, \dots, X_N, Y_1, \dots, Y_N$  represent the ensembles corresponding to individual letters. Then

$$I(\mathbf{X}^N; \mathbf{Y}^N) \leq \sum_{n=1}^N I(X_n; Y_n) \quad (4.2.4)$$

$$I(\mathbf{X}^N; \mathbf{Y}^N) \leq NC \quad (4.2.5)$$

Equality holds in (4.2.4) if the inputs are statistically independent and in (4.2.5) if the inputs are independent and have the probability assignment determined by (4.2.3).

---

*Proof.*

$$I(\mathbf{X}^N; \mathbf{Y}^N) = H(\mathbf{Y}^N) - H(\mathbf{Y}^N | \mathbf{X}^N) \quad (4.2.6)$$

$$H(\mathbf{Y}^N | \mathbf{X}^N) = \sum_{\mathbf{x}, \mathbf{y}} Q_N(\mathbf{x}) P_N(\mathbf{y} | \mathbf{x}) \log \frac{1}{P_N(\mathbf{y} | \mathbf{x})} \quad (4.2.7)$$

Since the channel is memoryless, (4.2.1) applies, yielding

$$H(\mathbf{Y}^N | \mathbf{X}^N) = \sum_{\mathbf{x}, \mathbf{y}} Q_N(\mathbf{x}) P_N(\mathbf{y} | \mathbf{x}) \sum_{n=1}^N \log \frac{1}{P(y_n | x_n)} \quad (4.2.8)$$

Log  $[1/P(y_n | x_n)]$  is a random variable, and the right side of (4.2.8) is the average of a sum of  $N$  random variables. This is equal to the sum of the averages whether the inputs are statistically independent or not. But the average of  $\log [1/P(y_n | x_n)]$  is  $H(Y_n | X_n)$ , so that

$$H(\mathbf{Y}^N | \mathbf{X}^N) = \sum_{n=1}^N H(Y_n | X_n) \quad (4.2.9)$$

This reduces one term in (4.2.6) to a sum of entropies; we next treat the other term,  $H(\mathbf{Y}^N)$ . From (2.3.10),

$$H(\mathbf{Y}^N) \leq \sum_{n=1}^N H(Y_n) \quad (4.2.10)$$

Substituting (4.2.9) and (4.2.10) into (4.2.6), we get

$$I(\mathbf{X}^N; \mathbf{Y}^N) \leq \sum_{n=1}^N [H(Y_n) - H(Y_n | X_n)] \quad (4.2.11)$$

from which (4.2.4) follows.

Equality occurs in (4.2.10), and consequently in (4.2.4), if and only if the output letters are statistically independent. If the input letters are statistically independent, satisfying

$$Q_N(\mathbf{x}) = \prod_n Q_{X_n}(x_n),$$

then the joint probability is

$$\prod_n Q_{X_n}(x_n)P(y_n/x_n).$$

and the statistical independence of the outputs follows.

The definition of  $C$  implies that  $I(X_n; Y_n) \leq C$  for each  $n$ , and thus (4.2.5) follows from (4.2.4). Furthermore, if the inputs are statistically independent and chosen to maximize each  $I(X_n; Y_n)$ , then  $I(X_n; Y_n) = C$  for each  $n$ , and (4.2.5) is satisfied with equality. |

It should not be concluded from this theorem that statistical dependence between inputs is something to be avoided. In fact, all of the coding techniques to be discussed later provide ways of introducing statistical dependence between the input letters, and some such dependence is generally necessary in order to achieve reliable transmission.

### 4.3 The Converse to the Coding Theorem

Up to this point, we have been discussing sources and channels in terms of various entropies and average mutual informations. In most data transmission systems, however, the mutual information is of less interest than the probability that the source letters are incorrectly reproduced at the destination. This error probability is the subject of the major theorem of information theory, the coding theorem. For a broad class of sources and channels, this theorem states that if the source entropy per unit time is less than the channel capacity per unit time, then the error probability can be reduced to any desired level by using a sufficiently complex encoder and decoder. In this section, we are interested in the converse result; if the source entropy is greater than capacity, arbitrarily small error probability cannot be achieved.

Let us first consider a sequence  $\mathbf{u} = (u_1, \dots, u_L)$  of  $L$  letters from the discrete source in Figure 4.3.1. The entropy per letter for the sequence of  $L$  letters is defined by

$$H_L(U) = \frac{H(\mathbf{U}^L)}{L} = -\frac{1}{L} \sum_{\mathbf{u}} \Pr(\mathbf{u}) \log \Pr(\mathbf{u}) \quad (4.3.1)$$

It was shown, in Theorem 3.5.1, that for a stationary source,  $H_L(U)$  is non-increasing with  $L$  and approaches a limit,  $H_\infty(U)$ , as  $L \rightarrow \infty$ . For a discrete memoryless source, of course,  $H_L(U) = H(U)$  for all  $L$ .

Suppose that the output from the decoder is a sequence,  $\mathbf{v} = (v_1, \dots, v_L)$  of letters from the same alphabet as the source. For any given source, coder,

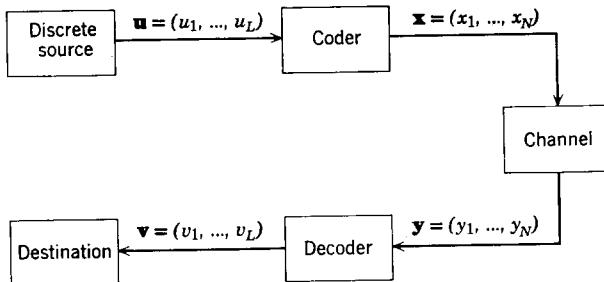


Figure 4.3.1. A communication system.

channel, and decoder, there is a joint probability measure\* on the set of possible input sequences  $\mathbf{u}$  and the set of possible output sequences  $\mathbf{v}$ .

The objective of the communication system is to have the sequence  $\mathbf{v}$  reproduce the sequence  $\mathbf{u}$ . If  $u_l \neq v_l$ , then an error has occurred on the  $l$ th digit of the transmission. The probability of such an error,  $P_{e,l}$  is specified by the joint ensemble  $\mathbf{U}^L \mathbf{V}^L$ . The average error probability  $\langle P_e \rangle$  over the sequence of  $L$  digits, is defined as

$$\langle P_e \rangle = \frac{1}{L} \sum_{l=1}^L P_{e,l} \quad (4.3.2)$$

The expected numbers of errors in the sequence is  $L\langle P_e \rangle$ . In later chapters, we shall often deal with the probability of one or more errors in a sequence of  $L$ . Here, however, we are interested in showing that reliable communication is impossible if the source entropy is larger than capacity. Thus, even if we showed that the probability of error in a sequence is 1 for large  $L$ , this would only guarantee one digit error out of  $L$ , or  $\langle P_e \rangle \geq 1/L$ . Thus, to show that  $\langle P_e \rangle$  is bounded away from 0 in the limit as  $L \rightarrow \infty$ , we must consider  $\langle P_e \rangle$  directly rather than the sequence error probability.

We begin by finding a relationship between  $\langle P_e \rangle$ ,  $H_L(U)$  and  $I(\mathbf{U}^L; \mathbf{V}^L)$  for the special case when  $L = 1$ . We then extend the result to arbitrary  $L$ , and finally relate  $I(\mathbf{U}^L; \mathbf{V}^L)$  to the capacity of the channel.

**Theorem 4.3.1.** Let  $U$ ,  $V$  be a joint ensemble in which the  $U$  and  $V$  sample spaces each contain the same  $M$  elements,  $a_1, \dots, a_M$ . Let  $P_e$  be the probability that the  $u$  and  $v$  outcomes are different,

$$P_e = \sum_u \sum_{v \neq u} P(u, v) \quad (4.3.3)$$

\* We shall see, in Section 4.6, that for channels with memory this statement requires some interpretation. For the time being, however, we shall assume the existence of such a probability measure.

Then

$$P_e \log (M - 1) + \mathcal{H}(P_e) \geq H(U | V) \quad (4.3.4)$$

where

$$\mathcal{H}(P_e) = -P_e \log P_e - (1 - P_e) \log (1 - P_e) \quad (4.3.5)$$

*Discussion.* The function  $P_e \log (M - 1) + \mathcal{H}(P_e)$  is sketched in Figure 4.3.2. The theorem says that if  $H(U | V)$  has a given value, plotted on the ordinate of Figure 4.3.2, then  $P_e$  must be greater than or equal to the corresponding abscissa. Since  $H(U | V) = H(U) - I(U; V)$ , the theorem bounds  $P_e$  in terms of the excess of the source entropy over the average mutual

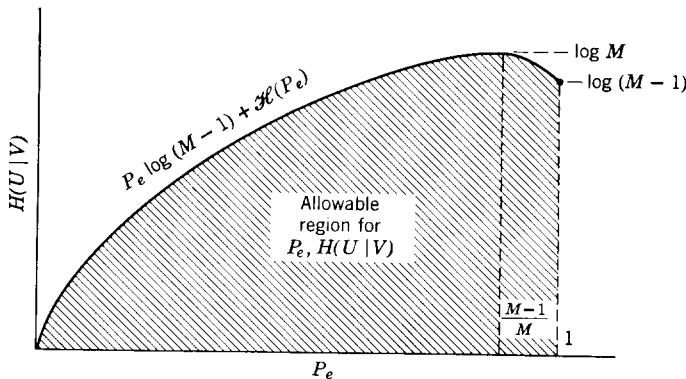


Figure 4.3.2. Interpretation of Theorem 4.3.1.

information. From an heuristic standpoint, it is easy to see why (4.3.4) is true. The average uncertainty in  $u$  given  $v$  can be broken into two terms: first, the uncertainty as to whether or not an error is made given  $v$ ; and, second, in those cases when an error is made, the uncertainty as to which input was transmitted. The first term is upper bounded by  $\mathcal{H}(P_e)$ , the second term by  $P_e$  times the maximum uncertainty given an error. Since the uncertainty given an error is among  $M - 1$  alternatives, the second term is upper bounded by  $P_e \log (M - 1)$ .

*Proof.* We can write  $H(U | V)$  as a sum of two terms, one involving the  $u, v$  pairs that yield errors,  $u \neq v$ , and the other involving the  $u, v$  pairs for which  $u = v$ .

$$H(U | V) = \sum_v \sum_{u \neq v} P(u, v) \log \frac{1}{P(u | v)} + \sum_{v, u=v} P(u, v) \log \frac{1}{P(u | v)} \quad (4.3.6)$$

Using (4.3.3) for  $P_e$ , the difference of the two sides of (4.3.4) is

$$\begin{aligned} H(U \mid V) - P_e \log(M - 1) - \mathcal{H}(P_e) &= \sum_v \sum_{u \neq v} P(u,v) \log \frac{P_e}{(M - 1)P(u \mid v)} \\ &\quad + \sum_{v, u=v} P(u,v) \log \frac{1 - P_e}{P(u \mid v)} \end{aligned} \quad (4.3.7)$$

Using the inequality  $\log z \leq (\log e)(z - 1)$ , the right-hand side of (4.3.7) is less than or equal to

$$(\log e) \left\{ \sum_v \sum_{u \neq v} P(u,v) \left[ \frac{P_e}{(M - 1)P(u \mid v)} - 1 \right] + \sum_{v, u=v} P(u,v) \left[ \frac{1 - P_e}{P(u \mid v)} - 1 \right] \right\} \quad (4.3.8)$$

$$= (\log e) \left[ \frac{P_e}{M - 1} \sum_v \sum_{u \neq v} P(v) - \sum_v \sum_{u \neq v} P(u,v) + (1 - P_e) \sum_v P(v) - \sum_{v, u=v} P(u,v) \right] \quad (4.3.9)$$

$$= (\log e)[P_e - P_e + (1 - P_e) - (1 - P_e)] = 0 \quad | \quad (4.3.10)$$

**Theorem 4.3.2.** Let  $\mathbf{U}^L, \mathbf{V}^L$  be a joint ensemble of sequences  $\mathbf{u} = (u_1, \dots, u_L)$  and  $\mathbf{v} = (v_1, \dots, v_L)$  in which each  $u_l$  and  $v_l$  sample space contains the same  $M$  elements,  $a_1, \dots, a_M$ . Let  $\langle P_e \rangle$  be defined by (4.3.2). Then

$$\langle P_e \rangle \log(M - 1) + \mathcal{H}(\langle P_e \rangle) \geq \frac{1}{L} H(\mathbf{U}^L \mid \mathbf{V}^L) \quad (4.3.11)$$


---

*Proof.* Using the chain rule on the joint ensemble  $\mathbf{U}^L = U_1 U_2 \cdots U_L$  (see Equation 2.2.30), we have

$$\begin{aligned} H(\mathbf{U}^L \mid \mathbf{V}^L) &= H(U_1 \mid \mathbf{V}^L) + H(U_2 \mid U_1 \mathbf{V}^L) + \cdots \\ &\quad + H(U_L \mid \mathbf{V}^L U_1 \cdots U_{L-1}) \end{aligned} \quad (4.3.12)$$

$$\leq \sum_{l=1}^L H(U_l \mid V_l) \quad (4.3.13)$$

Equation 4.3.13 follows from the general inequality  $H(X \mid Z) \leq H(X \mid ZY)$  (see Equation 2.3.13).

Applying Theorem 4.3.1 to each term in (4.3.13), we have

$$H(\mathbf{U}^L \mid \mathbf{V}^L) \leq \sum_{l=1}^L [P_{e,l} \log(M - 1) + \mathcal{H}(P_{e,l})] \quad (4.3.14)$$

$$\frac{1}{L} H(\mathbf{U}^L \mid \mathbf{V}^L) \leq \langle P_e \rangle \log(M - 1) + \frac{1}{L} \sum_{l=1}^L \mathcal{H}(P_{e,l}) \quad (4.3.15)$$



To complete the proof, we must show that

$$\frac{1}{L} \sum_{l=1}^L \mathcal{H}(P_{e,l}) \leq \mathcal{H}(\langle P_e \rangle) \quad (4.3.16)$$

This can be shown by means of the inequality  $\log z \leq (\log e)(z - 1)$ . We shall not go through the details, however, since (4.3.16) is also a simple consequence of the convexity properties of entropy to be discussed in the next section. |

At this point, we have bounded the error probability associated with a source in terms of the equivocation  $H(\mathbf{U}^L | \mathbf{V}^L)$ . We next bring the channel into the picture.

*A source sequence  $\mathbf{u} = (u_1, \dots, u_L)$  is defined to be connected to a destination through a sequence of  $N$  channel uses if the joint ensemble  $\mathbf{U}^L \mathbf{X}^N \mathbf{Y}^N \mathbf{V}^L$  (corresponding to source output  $\mathbf{u}$ , channel input  $\mathbf{x} = (x_1, \dots, x_N)$ , channel output  $\mathbf{y} = (y_1, \dots, y_N)$ , and decoded output  $\mathbf{v} = (v_1, \dots, v_L)$  has the properties that  $\mathbf{y}$  is conditionally independent of  $\mathbf{u}$  given  $\mathbf{x}$ , and  $\mathbf{v}$  is conditionally independent of  $\mathbf{u}$  and  $\mathbf{x}$  given  $\mathbf{y}$ .*

For discrete ensembles, these conditions assert that  $P(\mathbf{y} | \mathbf{xu}) = P(\mathbf{y} | \mathbf{x})$  and  $P(\mathbf{v} | \mathbf{y}, \mathbf{x}, \mathbf{u}) = P(\mathbf{v} | \mathbf{y})$ . In general, as indicated by Figure 4.3.1, they assert that the channel output depends statistically on the source sequence only through the channel input, and the decoded output  $\mathbf{v}$  depends on  $\mathbf{u}$  and  $\mathbf{x}$  only through the channel output  $\mathbf{y}$ . In other words, the conditions are a mathematical way of stating that there is no subsidiary “hidden” channel passing information about  $\mathbf{u}$  to the decoder.

If the source has memory, this definition is less innocuous than it appears. If successive blocks of  $L$  source digits are transmitted to the destination, a decoder could be constructed which made use of one block of received letters in decoding the next. The above definition rules out such decoders, but we shall see later that this problem disappears when we take the limit  $L \rightarrow \infty$ .

**Theorem 4.3.3 (Data-Processing Theorem).** Let a source sequence  $\mathbf{u} = (u_1, \dots, u_L)$  be connected to a destination through a sequence of  $N$  channel uses. Then

$$I(\mathbf{U}^L; \mathbf{V}^L) \leq I(\mathbf{X}^N; \mathbf{Y}^N) \quad (4.3.17)$$

where  $I(\mathbf{U}^L; \mathbf{V}^L)$  is the average mutual information between source sequence  $\mathbf{u} = (u_1, \dots, u_L)$  and decoded output sequence  $\mathbf{v} = (v_1, \dots, v_L)$ , and  $I(\mathbf{X}^N; \mathbf{Y}^N)$  is the average mutual information on the  $N$  uses of the channel.

*Proof.* The first condition of the above definition yields  $I(\mathbf{U}^L; \mathbf{Y}^N | \mathbf{X}^N) = 0$  from Theorem 2.3.3. From (2.3.19), we then have

$$I(\mathbf{U}^L; \mathbf{Y}^N) \leq I(\mathbf{X}^N; \mathbf{Y}^N) \quad (4.3.18)$$

The second condition of the definition yields  $I(\mathbf{U}^L; \mathbf{V}^L | \mathbf{Y}^N) = 0$ , and using (2.3.19) again

$$I(\mathbf{U}^L; \mathbf{V}^L) \leq I(\mathbf{U}^L; \mathbf{Y}^N) \quad (4.3.19)$$

Combining (4.3.18) and (4.3.19), we have (4.3.17). |

We can now combine the previous two theorems to yield

$$\begin{aligned} \langle P_e \rangle \log(M - 1) + \mathcal{H}(\langle P_e \rangle) &\geq \frac{1}{L} H(\mathbf{U}^L | \mathbf{V}^L) = H_L(U) - \frac{1}{L} I(\mathbf{U}^L; \mathbf{V}^L) \\ &\geq H_L(U) - \frac{1}{L} I(\mathbf{X}^N; \mathbf{Y}^N) \end{aligned} \quad (4.3.20)$$

where  $H_L(U) = (1/L) H(\mathbf{U}^L)$ .

If the channel is a DMC, we have  $I(\mathbf{X}^N; \mathbf{Y}^N) \leq NC$ , yielding

$$\langle P_e \rangle \log(M - 1) + \mathcal{H}(\langle P_e \rangle) \geq H_L(U) - \frac{N}{L} C \quad (4.3.21)$$

We can now relate  $N$  and  $L$  by the time interval between each source letter,  $\tau_s$ , and the time interval between each channel letter,  $\tau_c$ . We assume that the number of channel uses allowed is given by  $N = \lfloor L\tau_s/\tau_c \rfloor$  where by the notation  $\lfloor x \rfloor$  we mean the largest integer less than or equal to  $L\tau_s/\tau_c$ .

**Theorem 4.3.4 (Converse to the Coding Theorem).** Let a discrete stationary source with an alphabet size of  $M$  have entropy  $H_\infty(U) = \lim_{L \rightarrow \infty} H_L(U)$  and produce letters at a rate of one letter each  $\tau_s$  seconds. Let a discrete memoryless channel have capacity  $C$  and be used at a rate of one letter each  $\tau_c$  seconds. Let a source sequence of length  $L$  be connected to a destination through a sequence of  $N$  channel uses where  $N = \lfloor L\tau_s/\tau_c \rfloor$ . Then for any  $L$ , the error probability per source digit,  $\langle P_e \rangle$ , satisfies

$$\langle P_e \rangle \log(M - 1) + \mathcal{H}(\langle P_e \rangle) \geq H_\infty(U) - \frac{\tau_s}{\tau_c} C \quad (4.3.22)$$

---

*Proof.* From Theorem 3.5.1,  $H_\infty(U) \leq H_L(U)$ , and (4.3.22) follows immediately from (4.3.21). |

The appropriate interpretation of the above theorem is to consider  $L\tau_s$  as the total time over which transmission takes place. Within this total time, the coder can involve fixed-length or variable-length source coding and block

or nonblock coding for the channel. No matter what coding or data processing is done, the average error probability per source digit must satisfy (4.3.22), and is thus bounded away from zero if  $H_\infty(U)$  (the source rate) is greater than  $(\tau_s/\tau_c)C$  (the channel capacity per source digit). It should be observed that the theorem says nothing about the individual error probabilities,  $P_{e,l}$ . By appropriate design of the coder and decoder, we can make  $P_{e,l}$  small for some values of  $l$  and large for others.

While the theorem, as stated here, applies only to discrete memoryless channels, we see that that restriction was used only in going from (4.3.20) to (4.3.21). In order to apply the theorem to more general channels, we must find a way to define a joint  $\mathbf{X}^N\mathbf{Y}^N$  ensemble, and we must define  $C$  so that, in the limit as  $N \rightarrow \infty$ ,  $C \rightarrow (1/N)I(\mathbf{X}^N; \mathbf{Y}^N)$ . We shall consider this problem in Section 4.6.

In the special case for which the channel is noiseless, with a  $D$  letter input and output alphabet, (4.3.22) is simply a converse to the source coding theorem. It differs from (3.1.20) in that it bounds the error probability per digit rather than just the block error probability.

It can be seen that the bound in (4.3.22) is quite weak when the source alphabet size  $M$  is large. To show that this weakness is unavoidable, we shall show how to construct a source for which  $H_\infty(U)$  is arbitrarily large and  $\langle P_e \rangle > 0$  is arbitrarily small even for  $C = 0$ . Let the source be memoryless and have the alphabet  $a_1, \dots, a_M$ . Let  $\epsilon$  be an arbitrarily small positive number and let  $P(a_1) = 1 - \epsilon$  and  $P(a_m) = \epsilon/(M - 1)$  for  $m = 2, \dots, M$ . Then, if the decoder decodes each digit as  $a_1$ , errors will occur only when the source produces letters other than  $a_1$ . Thus the error probability,  $\langle P_e \rangle$ , is equal to  $\epsilon$ . On the other hand,

$$H_\infty(U) = (1 - \epsilon) \log \frac{1}{1 - \epsilon} + (M - 1) \frac{\epsilon}{M - 1} \log \frac{M - 1}{\epsilon} \quad (4.3.23)$$

For any  $\epsilon > 0$ , we can make  $H_\infty(U)$  as large as desired by making  $M$  sufficiently large. It can be seen that this “communication system” satisfies (4.3.22) with equality if  $C = 0$ .

#### 4.4 Convex Functions

In this section, we shall return to the problem of calculating the capacity of a discrete memoryless channel. As can be seen from (4.2.3), this involves maximizing a nonlinear function of many variables with both inequality and equality constraints. This maximization is greatly simplified by a property of mutual information known as convexity.\* We pause here for a brief

\* See, for example, Blackwell and Girshick. *Theory of Games and Statistical Decisions*. Chapter 2 (Wiley, New York, 1954) for a more complete discussion of convexity.

discussion of convexity which will be useful both in this problem and a number of similar problems later in the text.

Let  $\alpha = (\alpha_1, \dots, \alpha_K)$  be a  $K$  dimensional vector with real-valued components defined in a region  $R$  of vector space. We define a region  $R$  to be convex if for each vector  $\alpha$  in  $R$  and each vector  $\beta$  in  $R$ , the vector  $\theta\alpha + (1 - \theta)\beta$  is in  $R$  for  $0 \leq \theta \leq 1$ . Geometrically, as  $\theta$  goes from 0 to 1,  $\theta\alpha +$

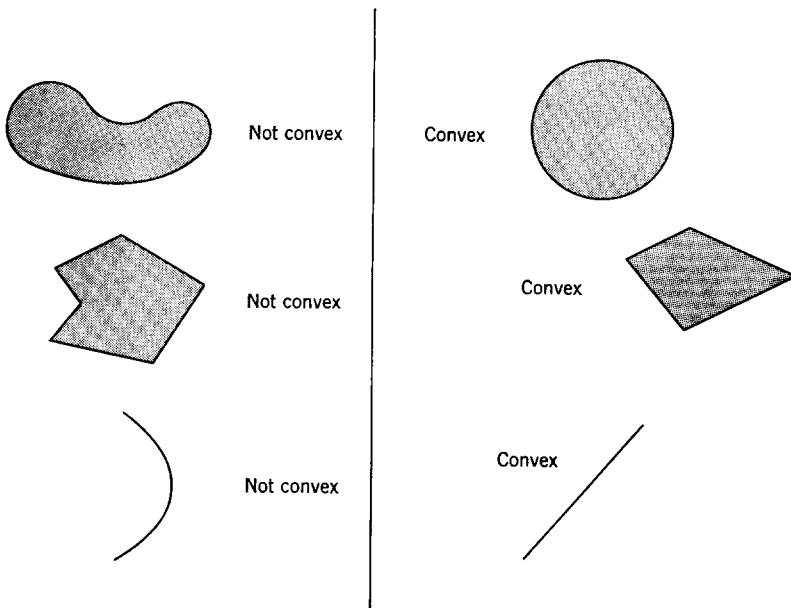


Figure 4.4.1. Examples of convex regions for two-dimensional vectors.

$(1 - \theta)\beta$  traces out a straight line from  $\beta$  to  $\alpha$ . Thus a region is convex if, for each pair of points in the region, the straight line between those points stays in the region (see Figure 4.4.1).

A very useful example of a convex region for our purposes is the region of probability vectors. A vector is defined to be a probability vector if its components are all nonnegative and sum to 1. To show that this region is convex, let  $\alpha$  and  $\beta$  be probability vectors and let  $\gamma = \theta\alpha + (1 - \theta)\beta$  for  $0 \leq \theta \leq 1$ . Then

$$\gamma_k = \theta\alpha_k + (1 - \theta)\beta_k \quad (4.4.1)$$

Thus  $\gamma_k \geq 0$ , and also

$$\sum_{k=1}^K \gamma_k = \theta \sum_{k=1}^K \alpha_k + (1 - \theta) \sum_{k=1}^K \beta_k = 1 \quad (4.4.2)$$

Thus  $\gamma$  is a probability vector and the region of probability vectors is convex.

*A real-valued function  $f$  of a vector is defined to be convex  $\cap$  (read convex cap) over a convex region  $R$  of vector space if, for all  $\alpha$  in  $R$ ,  $\beta$  in  $R$ , and  $\theta$ ,  $0 < \theta < 1$ , the function satisfies*

$$\theta f(\alpha) + (1 - \theta)f(\beta) \leq f[\theta\alpha + (1 - \theta)\beta] \quad (4.4.3)$$

*If the inequality is reversed for all such  $\alpha$ ,  $\beta$  and  $\theta$ ,  $f(\alpha)$  is convex  $\cup$  (read convex cup). If the inequality can be replaced with strict inequality,  $f(\alpha)$  is strictly convex  $\cap$  or convex  $\cup$ .\**

Figure 4.4.2 sketches the two sides of (4.4.3) as a function of  $\theta$ . It can be seen that the geometric interpretation of (4.4.3) is that every chord connecting two points on the surface representing the function must lie beneath (or on)

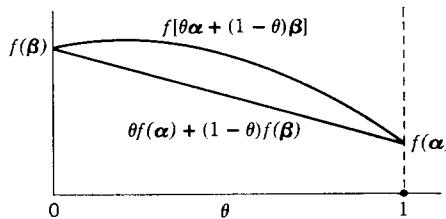


Figure 4.4.2. A convex  $\cap$  function.

the surface. The reason for restricting the definition to a convex region is to guarantee that the vector on the right-hand side of (4.4.3) is in  $R$ .

It is immediately seen from (4.4.3) that if  $f(\alpha)$  is convex  $\cap$ , then  $-f(\alpha)$  is convex  $\cup$  and vice versa. Thus, we shall deal only with convex  $\cap$  functions since the results can be easily applied to convex  $\cup$  functions.

The following properties of convex functions are frequently useful and are listed here together for convenience.

(1) If  $f_1(\alpha), \dots, f_L(\alpha)$  are convex  $\cap$  functions and if  $c_1, \dots, c_L$  are positive numbers, then

$$\sum_l c_l f_l(\alpha)$$

is convex  $\cap$  with strict convexity if any of the  $f_l(\alpha)$  are strictly convex.

(2) For a one-dimensional vector, if

$$d^2 f(\alpha)/d\alpha^2 \leq 0 \quad (4.4.4)$$

\* In the mathematical literature, a convex  $\cap$  function is usually called concave and a convex  $\cup$  function convex. That notation is avoided here since most people find the distinction very difficult to keep straight. In a recent poll among ten people who thought they knew the distinction, eight got convex and concave confused.

everywhere in an interval, then  $f(\alpha)$  is convex  $\cap$  in that interval with strict convexity if (4.4.4) is true with strict inequality.

(3) If  $(\alpha_1, \dots, \alpha_L)$  is a set of vectors all in a region where  $f(\alpha)$  is convex  $\cap$ , and if  $(\theta_1, \dots, \theta_L)$  is a set of probabilities (that is,  $\theta_i \geq 0$ ,  $\sum \theta_i = 1$ ), then

$$\sum_{i=1}^L \theta_i f(\alpha_i) \leq f\left[\sum_{i=1}^L \theta_i \alpha_i\right] \quad (4.4.5)$$

Considering  $\alpha$  as a discrete random vector and using an overline bar to indicate expectation, this is equivalent to

$$\overline{f(\alpha)} \leq f(\bar{\alpha}) \quad (4.4.6)$$

Property 1 follows immediately from substituting  $\sum c_i f_i(\alpha)$  into the equation defining convexity, (4.4.3). Property 2 is almost obvious geometrically (see Figure 4.4.2), and is proven in Problem 4.9. Property 3, in geometric terms, states that the portion of the “plane” joining the points  $f(\alpha_1), \dots, f(\alpha_L)$  lies beneath (or on) the surface generated by  $f(\alpha)$  between those points; see Problem 4.9 for a proof.

In terms of these properties, it is easy to see that the entropy of an ensemble,

$$-\sum_k P(a_k) \log P(a_k),$$

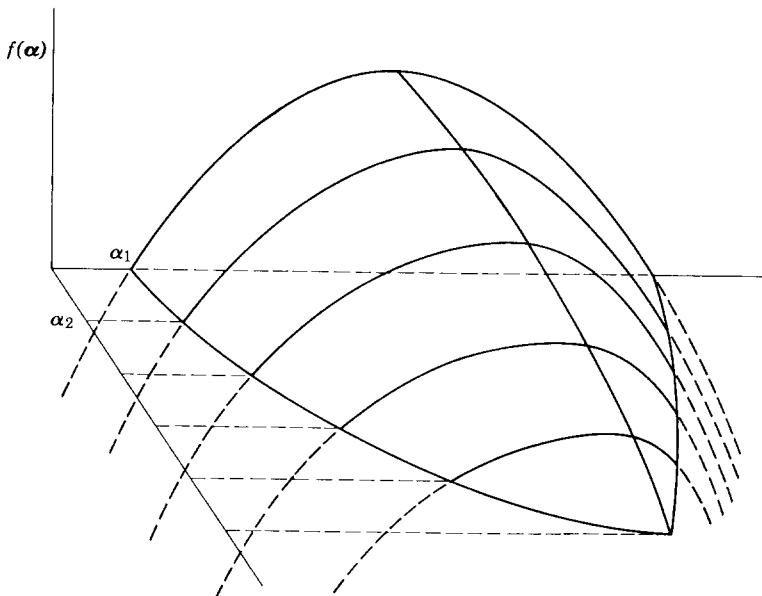
is a strictly convex  $\cap$  function of its constituent probabilities. Property 2 shows that  $-P(a_k) \log P(a_k)$  is strictly convex  $\cap$  in  $P(a_k)$ , and property 1 shows that the sum is strictly convex  $\cap$ . Equation 4.3.16, which we left unproved in the last section, follows from (4.4.6) and the convexity of entropy.

For our purposes, the major reason for discussing convexity is that convex  $\cap$  functions are relatively easy to maximize over a convex region. In order to see this, we shall first consider some examples from a nonrigorous standpoint. First, suppose that  $f(\alpha)$  is a convex  $\cap$  function in the region  $R$  where  $\alpha_k \geq 0$ ,  $1 \leq k \leq K$ . We might reasonably attempt to maximize  $f(\alpha)$  over  $R$  by finding a stationary point of  $f(\alpha)$ ; that is, an  $\alpha$  for which

$$\frac{\partial f(\alpha)}{\partial \alpha_k} = 0; \quad 1 \leq k \leq K \quad (4.4.7)$$

The set of equations above might have no solution, and if there are solutions, they might not satisfy the constraints  $\alpha_k \geq 0$ . We now show, however, that if there is an  $\alpha$  satisfying both (4.4.7) and the constraints, then that  $\alpha$  maximizes the function. To see this, we shall assume that the result is false: that is, that there is some vector  $\beta$  in the region for which  $f(\beta) > f(\alpha)$ . The chord joining  $f(\beta)$  and  $f(\alpha)$  is then increasing from  $f(\alpha)$  to  $f(\beta)$ . As seen from Figure 4.4.2, the rate of increase of the function at  $\alpha$  in the direction of  $\beta$  is at least as large as that of the chord. Thus  $\alpha$  cannot be a stationary point of  $f$  and we have arrived at a contradiction.

Next, consider the possibility that the maximum of  $f(\alpha)$  over  $R$  occurs on the boundary of the region, that is, where one or more of the components of  $\alpha$  are zero. In that case, as shown in Figure 4.4.3, we should not be surprised if the function is strictly decreasing with respect to variations *into* the region, that is, with respect to variations of the zero-valued components of  $\alpha$ . On the other hand, if  $f$  is differentiable, we still expect the maximum to be at a



*Figure 4.4.3. Sketch of convex function with maximum over  $\alpha_1 \geq 0$ ,  $\alpha_2 \geq 0$  at  $\alpha_1 = 0$ .*

stationary point with respect to variation of the nonzero components of  $\alpha$ . This suggests replacing (4.4.7) with the equations

$$\frac{\partial f(\alpha)}{\partial \alpha_k} = 0; \quad \text{all } k \text{ such that } \alpha_k > 0 \quad (4.4.8)$$

$$\frac{\partial f(\alpha)}{\partial \alpha_k} \leq 0; \quad \text{all } k \text{ such that } \alpha_k = 0 \quad (4.4.9)$$

How to solve these equations is of no concern to us at the moment. What is important is that if an  $\alpha$  in the region satisfies (4.4.8) and (4.4.9), it maximizes  $f$ , and, conversely, if  $f$  is differentiable and has a maximum in the region at  $\alpha$ , then (4.4.8) and (4.4.9) are satisfied. We shall see how to prove these statements shortly.

As a second example, suppose that we want to maximize a convex function,  $f(\alpha)$ , over the region where  $\alpha$  is a probability vector, that is, where the components of  $\alpha$  are nonnegative and sum to 1. The constraint  $\sum \alpha_k = 1$  suggests using a Lagrange multiplier, which implies maximizing  $f(\alpha) - \lambda \sum \alpha_k$  over the region where the components are nonnegative, and then choosing  $\lambda$  so that the maximum occurs at  $\sum \alpha_k = 1$ . Applying (4.4.8) and (4.4.9) to the function  $f(\alpha) - \lambda \sum \alpha_k$ , we obtain

$$\frac{\partial f(\alpha)}{\partial \alpha_k} = \lambda; \quad \text{all } k \text{ such that } \alpha_k > 0 \quad (4.4.10)$$

$$\frac{\partial f(\alpha)}{\partial \alpha_k} \leq \lambda; \quad \text{all } k \text{ such that } \alpha_k = 0 \quad (4.4.11)$$

We shall see that (4.4.10) and (4.4.11) are indeed necessary and sufficient conditions on the probability vector  $\alpha$  that maximizes a convex differentiable function  $f$ . In other words, if a probability vector  $\alpha$  satisfies (4.4.10) and (4.4.11) for some value of  $\lambda$ , that  $\alpha$  maximizes  $f$  over the region, and conversely if  $\alpha$  maximizes  $f$  over the region, (4.4.10) and (4.4.11) are satisfied for some  $\lambda$ . We now turn to a proof of this result.

**Theorem 4.4.1.** Let  $f(\alpha)$  be a convex function of  $\alpha = (\alpha_1, \dots, \alpha_k)$  over the region  $R$  when  $\alpha$  is a probability vector. Assume that the partial derivatives,  $\partial f(\alpha)/\partial \alpha_k$  are defined and continuous over the region  $R$  with the possible exception that  $\lim_{\alpha_k \rightarrow 0} \partial f(\alpha)/\partial \alpha_k$  may be  $+\infty$ . Then (4.4.10) and (4.4.11) are necessary and sufficient conditions on a probability vector  $\alpha$  to maximize  $f$  over the region  $R$ .

*Proof (Sufficiency).* Assume that (4.4.10) and (4.4.11) are satisfied for some  $\lambda$  and some probability vector  $\alpha$ . We shall show that, for any other probability vector  $\beta$ ,  $f(\beta) - f(\alpha) \leq 0$ , thus establishing that  $\alpha$  maximizes  $f$ . From the definition of convexity,

$$\theta f(\beta) + (1 - \theta)f(\alpha) \leq f[\theta\beta + (1 - \theta)\alpha]; \quad 0 < \theta < 1 \quad (4.4.12)$$

Rearranging terms (4.4.12) becomes

$$f(\beta) - f(\alpha) \leq \frac{f[\theta\beta + (1 - \theta)\alpha] - f(\alpha)}{\theta} \quad (4.4.13)$$

Since (4.4.13) is valid for all  $\theta$ ,  $0 < \theta < 1$ , we can pass to the limit, obtaining

$$f(\beta) - f(\alpha) \leq \left. \frac{df[\theta\beta + (1 - \theta)\alpha]}{d\theta} \right|_{\theta=0} \quad (4.4.14)$$

Carrying out the differentiation, we obtain

$$f(\beta) - f(\alpha) \leq \sum_k \frac{\partial f(\alpha)}{\partial \alpha_k} (\beta_k - \alpha_k) \quad (4.4.15)$$

The existence of the derivative in (4.4.14) and the equivalence of (4.4.14) and (4.4.15) are guaranteed by the continuity of the partial derivatives. This continuity is given by hypothesis since (4.4.10) and (4.4.11) rule out the exceptional case where  $\partial f / \partial \alpha_k = +\infty$ . Observe now that

$$\frac{\partial f(\alpha)}{\partial \alpha_k} (\beta_k - \alpha_k) \leq \lambda (\beta_k - \alpha_k) \quad (4.4.16)$$

This follows from (4.4.10) if  $\alpha_k > 0$ . If  $\alpha_k = 0$ , we have  $\beta_k - \alpha_k \geq 0$ , and (4.4.12) follows from (4.4.11).

Substituting (4.4.16) into (4.4.15), we have

$$f(\beta) - f(\alpha) \leq \lambda \left[ \sum_k \beta_k - \sum_k \alpha_k \right] \quad (4.4.17)$$

Since  $\beta$  and  $\alpha$  are probability vectors, we have  $f(\beta) - f(\alpha) \leq 0$  for each  $\beta$  in the region.

*Necessity.* Let  $\alpha$  maximize  $f$  over the region and assume, for the moment that the partial derivatives are continuous at  $\alpha$ . Since  $\alpha$  maximizes  $f$ , we have

$$f[\theta \beta + (1 - \theta)\alpha] - f(\alpha) \leq 0 \quad (4.4.18)$$

for any probability vector  $\beta$  and any  $\theta$ ,  $0 < \theta < 1$ . Dividing by  $\theta$  and taking the limit as  $\theta \rightarrow 0$ , we have

$$\frac{df[\theta \beta + (1 - \theta)\alpha]}{d\theta} \Big|_{\theta=0} \leq 0 \quad (4.4.19)$$

$$\sum_k \frac{\partial f(\alpha)}{\partial \alpha_k} (\beta_k - \alpha_k) \leq 0 \quad (4.4.20)$$

At least one component of  $\alpha$  is strictly positive, and we assume, for simplicity of notation, that  $\alpha_1 > 0$ . Let  $\mathbf{i}_k$  be a unit vector with a one in the  $k$ th position and zeros elsewhere and choose  $\beta$  as  $\alpha + \epsilon \mathbf{i}_k - \epsilon \mathbf{i}_1$ . Since  $\alpha_1 > 0$ ,  $\beta$  is a probability vector for  $0 \leq \epsilon \leq \alpha_1$ . Substituting this  $\beta$  in (4.4.20), we have

$$\epsilon \frac{\partial f(\alpha)}{\partial \alpha_k} - \epsilon \frac{\partial f(\alpha)}{\partial \alpha_1} \leq 0 \quad (4.4.21)$$

$$\frac{\partial f(\alpha)}{\partial \alpha_k} \leq \frac{\partial f(\alpha)}{\partial \alpha_1} \quad (4.4.22)$$

If  $\alpha_k > 0$ ,  $\epsilon$  can also be chosen negative, in which case the inequality in (4.4.22) is reversed, yielding

$$\frac{\partial f(\alpha)}{\partial \alpha_k} = \frac{\partial f(\alpha)}{\partial \alpha_1}; \quad \alpha_k > 0 \quad (4.4.23)$$

Finally, choosing  $\lambda$  as  $\partial f(\alpha)/\partial \alpha_1$ , (4.4.22) and (4.4.23) become equivalent to (4.4.10) and (4.4.11). To complete the proof, we shall consider an  $\alpha$  for which  $\partial f(\alpha)/\partial \alpha_k = +\infty$  for some  $k$  and show that such an  $\alpha$  cannot maximize  $f$ . Assume, for simplicity of notation, that  $\alpha_1 > 0$ .

$$\begin{aligned} \frac{f(\alpha + \epsilon \mathbf{i}_k - \epsilon \mathbf{i}_1) - f(\alpha)}{\epsilon} &= \frac{f(\alpha + \epsilon \mathbf{i}_k - \epsilon \mathbf{i}_1) - f(\alpha + \epsilon \mathbf{i}_k)}{\epsilon} \\ &\quad + \frac{f(\alpha + \epsilon \mathbf{i}_k) - f(\alpha)}{\epsilon} \end{aligned} \quad (4.4.24)$$

In the limit as  $\epsilon \rightarrow 0$ , the first term on the right-hand side of (4.4.24) remains bounded because of the continuity of  $\partial f/\partial \alpha_1$ . The second term blows up, so that the left side of (4.4.24) is positive for sufficiently small  $\epsilon$ . This, however, shows that  $\alpha$  does not maximize  $f$ , completing the proof. |

As one might guess from the discussion of convexity in the last section, we are about to show that mutual information is a convex  $\cap$  function of the input probabilities.

**Theorem 4.4.2.** Let a discrete memoryless channel with  $K$  inputs and  $J$  outputs have transition probabilities  $P(j \mid k)$ ,  $0 \leq j \leq J - 1$ ,  $0 \leq k \leq K - 1$ . Let  $\mathbf{Q} = [Q(0), \dots, Q(K - 1)]$  represent an arbitrary input probability assignment to the channel.

Then

$$I(X; Y) = \sum_{j,k} Q(k)P(j \mid k) \log \frac{P(j \mid k)}{\sum_i Q(i)P(j \mid i)} \quad (4.4.25)$$

is a convex  $\cap$  function of  $\mathbf{Q}$ .

*Proof.* Let  $\mathbf{Q}_0$  and  $\mathbf{Q}_1$  be arbitrary probability vectors for the channel input and let  $I_0$  and  $I_1$  be the associated average mutual informations. Let  $\theta$  be an arbitrary number,  $0 < \theta < 1$ , let  $\mathbf{Q} = \theta \mathbf{Q}_0 + (1 - \theta) \mathbf{Q}_1$ , and let  $I$  be the average mutual information for the input probability assignment  $\mathbf{Q}$ . We must show that

$$\theta I_0 + (1 - \theta) I_1 \leq I \quad (4.4.26)$$

We can, if we wish, consider  $\mathbf{Q}_0$  and  $\mathbf{Q}_1$  to be conditional probabilities conditional on a binary valued random outcome  $z$  (see Figure 4.4.4).

$$Q_0(k) = Q_{X|Z}(k \mid 0); \quad Q_1(k) = Q_{X|Z}(k \mid 1) \quad (4.4.27)$$

We choose  $P_Z(0) = \theta$ ,  $P_Z(1) = 1 - \theta$ , and (as indicated in Figure 4.4.4)  $P(y | x, z) = P(y | x)$ . In terms of this triple ensemble, the left-hand side of (4.4.26) is  $I(X; Y | Z)$  and the right-hand side is  $I(X; Y)$ .

As in the cascaded channels discussed in Section 2.3.  $z$  and  $y$  are conditionally independent, given  $x$ . Thus, as in (2.3.15),

$$I(Y; Z | X) = 0 \quad (4.4.28)$$

We also have, as in (2.3.16) and (2.3.17),

$$I(Y; ZX) = I(Y; Z) + I(Y; X | Z) \quad (4.4.29)$$

$$= I(Y; X) + I(Y; Z | X) \quad (4.4.30)$$

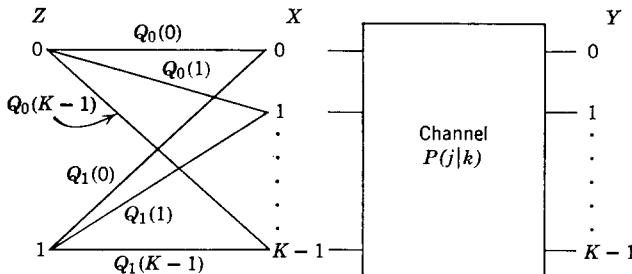


Figure 4.4.4.

Equating the right-hand sides of (4.4.29) and (4.4.30), and using (4.4.28), we have

$$I(Y; Z) + I(Y; X | Z) = I(Y; X) \quad (4.4.31)$$

$$I(Y; X | Z) \leq I(Y; X) \quad (4.4.32)$$

$$I(X; Y | Z) \leq I(X; Y) \quad (4.4.33)$$

Since this is equivalent to (4.4.26), the proof is complete. |

An alternative, more direct, proof is sketched in Problem 4.16. The proof here has the advantage of great generality, applying to virtually arbitrary channels. Before using this result in the calculation of channel capacity, we shall prove a closely related theorem that is useful both in the study of unknown channels and in Chapter 9.

**Theorem 4.4.3.** Consider  $I(X; Y)$  in (4.4.25) as a function of the transition probabilities  $P(j | k)$  and the input assignment  $Q(k)$ . For a fixed input assignment,  $I(X; Y)$  is a convex  $\cup$  function of the set of transition probabilities (notice that this is convex  $\cup$  rather than convex  $\cap$  as in Theorem 4.4.2).

*Proof.* Let  $P_0(j|k)$  and  $P_1(j|k)$ ,  $0 \leq k \leq K-1$ ,  $0 \leq j \leq J-1$ , be two arbitrary sets of transition probabilities, and let  $P(j|k) = \theta P_0(j|k) + (1-\theta)P_1(j|k)$  for an arbitrary  $\theta$ ,  $0 < \theta < 1$ . Let  $I_0$ ,  $I_1$ , and  $I$  be the average mutual informations for these sets of transition probabilities. We must show that

$$\theta I_0 + (1-\theta)I_1 \geq I \quad (4.4.34)$$

As in the last theorem, we can consider  $P_0$  and  $P_1$  as conditional on a binary variable  $z$ ,

$$P_0(j|k) = P_{Y|XZ}(j|k, 0); \quad P_1(j|k) = P_{Y|XZ}(j|k, 1) \quad (4.4.35)$$

Letting  $P_Z(0) = \theta$ ,  $P_Z(1) = 1 - \theta$ , and defining  $z$  to be statistically independent of  $x$ , the left-hand side of (4.4.34) is  $I(X; Y|Z)$  and the right-hand side is  $I(X; Y)$ . Proceeding as in the last theorem, we have

$$I(X; YZ) = I(X; Z) + I(X; Y|Z) \quad (4.4.36)$$

$$= I(X; Y) + I(X; Z|Y) \quad (4.4.37)$$

Since  $x$  and  $z$  are statistically independent,  $I(X; Z) = 0$  yielding

$$I(X; Y|Z) = I(X; Y) + I(Z; X|Y) \quad (4.4.38)$$

$$I(X; Y|Z) \geq I(X; Y) \quad (4.4.39)$$

This is equivalent to (4.4.34), completing the proof. |

#### 4.5 Finding Channel Capacity for a Discrete Memoryless Channel

**Theorem 4.5.1.** A set of necessary and sufficient conditions on an input probability vector  $\mathbf{Q} = [Q(0), \dots, Q(K-1)]$  to achieve capacity on a discrete memoryless channel with transition probabilities  $P(j|k)$  is that for some number  $C$ ,

$$I(x = k; Y) = C; \quad \text{all } k \text{ with } Q(k) > 0 \quad (4.5.1)$$

$$I(x = k; Y) \leq C; \quad \text{all } k \text{ with } Q(k) = 0 \quad (4.5.2)$$

in which  $I(x = k; Y)$  is the mutual information for input  $k$  averaged over the outputs,

$$I(x = k; Y) = \sum_j P(j|k) \log \frac{P(j|k)}{\sum_i Q(i)P(j|i)} \quad (4.5.3)$$

Furthermore, the number  $C$  is the capacity of the channel.

*Proof.* We wish to maximize

$$I(X; Y) = \sum_{k,j} Q(k)P(j|k) \log \frac{P(j|k)}{\sum_i Q(i)P(j|i)} \quad (4.5.4)$$

over all choices of  $\mathbf{Q}$ . Taking partial derivatives,\*

$$\frac{\partial I(X;Y)}{\partial Q(k)} = I(x = k; Y) - \log e \quad (4.5.5)$$

We can apply Theorem 4.4.1 to the maximization since  $I(X;Y)$  is convex  $\cap$  in  $\mathbf{Q}$  and the partial derivatives satisfy the appropriate continuity conditions. Thus necessary and sufficient conditions on  $\mathbf{Q}$  to maximize  $I(X;Y)$  are

$$\frac{\partial I(X;Y)}{\partial Q(k)} = \lambda; \quad Q(k) > 0 \quad (4.5.6)$$

$$\leq \lambda; \quad Q(k) = 0 \quad (4.5.7)$$

Using (4.5.5), and letting  $C = \lambda + \log e$ , we have (4.5.1) and (4.5.2). Multiplying both sides of (4.5.1) by  $Q(k)$ , and summing over the  $k$  for which  $Q(k) > 0$ , we get the maximum value of  $I(X;Y)$  on the left and the constant  $C$  on the right, establishing that  $C$  is indeed channel capacity. |

Theorem 4.5.1 has a simple intuitive interpretation. If one input has a larger mutual information associated with it than another, then we should be able to increase the average information by using the larger mutual information more often. Any such change, however, will change the mutual information of each input, and thus, after enough change, all inputs will yield the same information, except perhaps a few inputs that are so poor that their probabilities are reduced to zero.

Despite the elegance of Theorem 4.5.1, there is a real question of how to use it in actually evaluating channel capacity. In what follows, we shall give a number of ways to actually find capacity. First, we start with the examples in Figure 4.5.1.

For the binary symmetric channel (BSC) of Figure 4.5.1a, we use the symmetry to guess that capacity is achieved with  $Q(0) = Q(1) = \frac{1}{2}$ . Checking our guess from (4.5.1), we obtain  $I(x = 0; Y) = I(x = 1; Y) = 1 - \mathcal{H}(\epsilon)$  bits, where  $\mathcal{H}(\epsilon) = -\epsilon \log_2 \epsilon - (1 - \epsilon) \log_2 (1 - \epsilon)$  is the entropy of a binary variable with the probabilities  $\epsilon$  and  $(1 - \epsilon)$ . Since (4.5.1) is satisfied for both inputs, this  $\mathbf{Q}$  yields capacity and  $C = 1 - \mathcal{H}(\epsilon)$  bits. Thus one important use of the theorem is to provide a simple test for checking any hypothesis about the input probabilities that yield capacity. This also means that we can be as mathematically careless as we wish in finding a  $\mathbf{Q}$  that yields capacity since it is easy to check the result. We can find the capacity of the binary erasure channel (BEC) in Figure 4.5.1b in the same way. We

\* Notice that the sum in the denominator of the logarithm in (4.5.4) contains a term  $Q(k)P(j \mid k)$ ; this term gives rise to the  $\log e$  in (4.5.5).

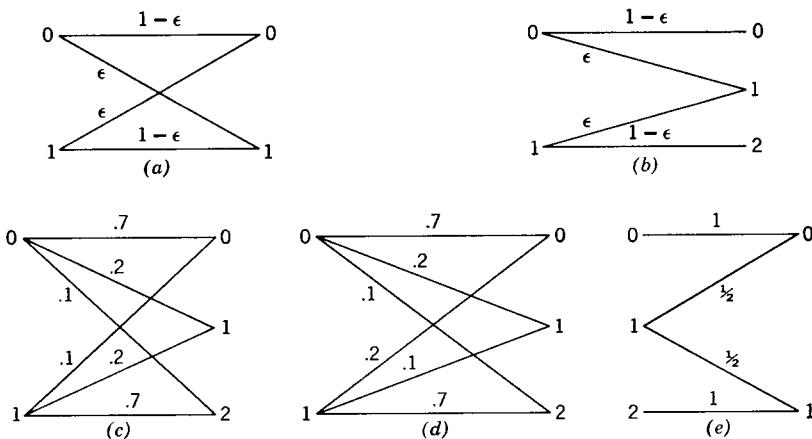


Figure 4.5.1.

guess  $Q(0) = Q(1) = \frac{1}{2}$  and verify that  $I(x = 0; Y) = I(x = 1; Y) = C = 1 - \epsilon$  bits. These capacities are sketched in Figure 4.5.2. In Figure 4.5.1c, we notice the same kind of symmetry and again verify that capacity is achieved with  $Q(0) = Q(1) = \frac{1}{2}$ . Figure 4.5.1d, while superficially similar to Figure 4.5.1c, is somehow less symmetric, and if we try  $Q(0) = Q(1) = \frac{1}{2}$ , we find that (4.5.1) is not satisfied and thus capacity is not achieved.

In defining what we mean by a symmetric channel, it is surprisingly awkward to include channels such as *c* and exclude channels such as *d*. In doing this, we observe that in *c*, there are two outputs, 0 and 2, which are similar and the other one is different. It is then not surprising that a general definition of symmetric channels will involve a partitioning of the set of outputs into subsets of similar outputs.

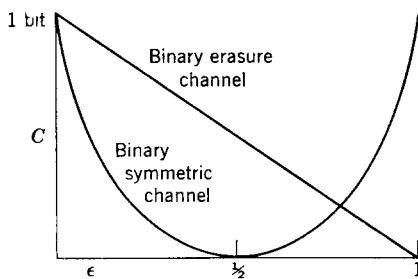


Figure 4.5.2. Capacity of binary symmetric and binary erasure channels.

A DMC is defined to be symmetric if the set of outputs can be partitioned into subsets in such a way that for each subset the matrix of transition probabilities (using inputs as rows and outputs of the subset as columns) has the property that each row is a permutation of each other row and each column (if more than 1) is a permutation of each other column. For example, partitioning the outputs of channel  $c$  in Figure 4.5.1 into (0,2) and 1, we get the matrices

$$\begin{array}{c}
 & & j \\
 & 0 & 2 \\
 \hline
 k & 0 & \boxed{\begin{matrix} 0.7 & 0.1 \\ 0.1 & 0.7 \end{matrix}} & \boxed{1} \\
 & 1 & \boxed{0.2} \\
 \hline
 \end{array}$$

Since each matrix above has the permutation property of the definition, the channel is symmetric.

**Theorem 4.5.2.** For a symmetric discrete memoryless channel, capacity is achieved by using the inputs with equal probability.

*Proof.* For equiprobable inputs, we have

$$I(x = k; Y) = \sum_{j=0}^{J-1} P(j | k) \log \frac{P(j | k)}{(1/K) \sum_i P(j | i)} \quad (4.5.8)$$

Within a partition of outputs, each column of the  $P(j | k)$  matrix is a permutation of each other column, and thus the output probability,

$$(1/K) \sum_i P(j | i),$$

is the same for all outputs in a partition. It follows that, within a partition of outputs, the matrix with elements  $P(j | k)I_{X;Y}(k; j)$  has the same permutation properties as  $P(j | k)$ , and thus the sum for these terms in (4.5.8) is the same for each input. Thus (4.5.1) is satisfied and capacity is achieved. |

Next, consider finding the capacity of channel  $e$  in Figure 4.5.1. From an intuitive viewpoint, input 1 is a poor choice for transmitting information, and we guess that capacity is achieved by  $Q(0) = Q(2) = \frac{1}{2}$ . Checking the guess, we find  $I(x = 0; Y) = I(x = 2; Y) = 1$  bit and  $I(x = 1; Y) = 0$ . Thus (4.5.1) and (4.5.2) are satisfied and our guess is proven to be correct.

Unfortunately, one might sometimes be interested in finding the capacity of a channel that is not symmetric and for which the optimum  $\mathbf{Q}$  cannot be guessed. The easiest procedure is to use a computer to find the maximum. Since the function is convex  $\cap$ , such a computer program simply involves

varying  $\mathbf{Q}$  to increase  $I(X;Y)$  until a local (and, therefore, global) maximum is reached.

If one rigidly insists, however, in finding a traditional solution to (4.5.1) and (4.5.2), an answer can sometimes be found as follows. First, assume all  $Q(k)$  to be nonzero. Then rewrite (4.5.1) in the form

$$\sum_{j=0}^{J-1} P(j \mid k) \log P(j \mid k) - \sum_j P(j \mid k) \log \omega(j) = C \quad (4.5.9)$$

in which the output probability  $\omega(j)$  is

$$\omega(j) = \sum_{k=0}^{K-1} Q(k)P(j \mid k) \quad (4.5.10)$$

Rewriting (4.5.9),

$$\sum_j P(j \mid k)[C + \log \omega(j)] = \sum_j P(j \mid k) \log P(j \mid k) \quad (4.5.11)$$

This gives us  $K$  linear equations in the  $J$  unknowns,  $(C + \log \omega(j))$ . If  $K = J$ , and if the  $P(j \mid k)$  matrix is nonsingular, these linear equations can be solved. If we let  $\beta_j = C + \log \omega(j)$  be the solutions, then we can find  $C$  from the restriction that  $\sum \omega(j) = 1$ , obtaining

$$C = \log_2 \sum_j 2^{\beta_j} \text{ bits} \quad (4.5.12)$$

where all logs are taken base 2. Unfortunately, we cannot be sure that  $C$ , as given by (4.5.12), is the channel capacity. We must first use  $C$  to calculate the  $\omega(j)$ , and then solve for the input probability,  $Q(k)$ , from (4.5.10). If each  $Q(k)$  is nonnegative, then the solution is correct; otherwise not.

If  $J > K$ , then the equations (4.5.11) will typically have a family of solutions, but only one will lead to a solution of (4.5.10). Finding  $C$  in this case involves solving a set of simultaneous nonlinear equations. Even if a solution is found, some of the  $Q(k)$  might be negative, voiding the solution.

This concludes our discussion of finding capacity. We can guess, use the channel symmetry, use a computer search, or solve the equations, but solving the equations is often a tedious if not impossible task. We conclude this section with several interesting corollaries to Theorems 4.5.1 and 4.4.2.

**COROLLARY 1.** For any input probability assignment that achieves capacity on a discrete memoryless channel, the output probabilities are all strictly positive. (Here we assume that each output can be reached from some input.)

*Proof.* Letting  $\omega(j)$  be the probability of output  $j$ , (4.5.2) becomes

$$\sum_{j=0}^{J-1} P(j \mid k) \log \frac{P(j \mid k)}{\omega(j)} \leq C; \quad \text{for } k \text{ such that } Q(k) = 0 \quad (4.5.13)$$

If some  $\omega(j) = 0$ , any input that reaches output  $j$  with nonzero transition probability must have  $Q(k) = 0$ . For such a  $k$ , the left side of (4.5.13) is infinite, establishing a contradiction. |

COROLLARY 2. The output probability vector that achieves capacity is unique. All input probability vectors with the proper zero components that give rise to that output vector also give rise to capacity. —————

*Proof.* Let  $\mathbf{Q}_0$  and  $\mathbf{Q}_1$  be any two input probability vectors that achieve capacity  $C$ . For  $\theta$  between 0 and 1, the input  $\theta\mathbf{Q}_0 + (1 - \theta)\mathbf{Q}_1$  also achieves capacity, since the convexity  $\cap$  of  $I(X; Y)$  shows that the average mutual information for  $\theta\mathbf{Q}_0 + (1 - \theta)\mathbf{Q}_1$  cannot be less than  $C$ . Using the same conditional probability notation,

$$Q_0(k) = Q_{X|Z}(k \mid 0) \quad \text{and} \quad Q_1(k) = Q_{X|Z}(k \mid 1),$$

as in the proof of Theorem 4.4.2, we see that  $I(X; Y) = I(X; Y \mid Z)$ . From (4.4.31), however, it follows that  $I(Y; Z) = 0$ . Thus  $y$  and  $z$  are statistically independent, and  $P_{Y|Z}(j \mid 0) = P_Y(j) = P_{Y|Z}(j \mid 1)$ . Thus the same output probability assignment corresponds to both  $\mathbf{Q}_0$  and  $\mathbf{Q}_1$  and this output probability vector is unique. Finally, since conditions (4.5.1) and (4.5.2) depend on  $Q(k)$  only through the output probabilities

$$\sum_k Q(k)P(j \mid k),$$

any  $\mathbf{Q}$  (with the proper zero components) giving rise to this output probability vector satisfies (4.5.1) and (4.5.2), thus yielding capacity. |

COROLLARY 3. Let  $m$  be the smallest number of inputs that can be used with nonzero probability to achieve capacity and let  $A$  be such a set of inputs. Then  $m \leq J$ , where  $J$  is the size of the output alphabet, and the input probability assignment on  $A$  to achieve capacity using only inputs in  $A$  is unique. —————

*Proof.* Let  $\omega = (\omega(0), \dots, \omega(J-1))$  be the output probability vector that achieves capacity. The input probabilities, restricted to be nonzero only on the set  $A$ , must satisfy

$$\sum_{k \in A} Q(k)P(j \mid k) = \omega(j); \quad 0 \leq j \leq J-1 \quad (4.5.14)$$

This is a set of  $J$  equations in  $m$  unknowns, and by assumption there is at least one solution [notice that any solution satisfies  $\sum Q(k) = 1$ ]. Assume that the solution is not unique, let  $\mathbf{Q}$  be a probability vector solution and let  $\mathbf{h}$  be a nonzero solution of the homogeneous equations. Then  $\mathbf{Q} + \theta\mathbf{h}$  satisfies (4.5.14). Now increase  $\theta$  from 0 until a component of  $\mathbf{Q} + \theta\mathbf{h}$  reaches 0. This can always be done since  $\sum h(k) = 0$  and  $\mathbf{h}$  has negative components. This  $\mathbf{Q} + \theta\mathbf{h}$  achieves capacity with  $m - 1$  nonzero components,

and the assumption of a nonunique solution is false. Since the solution is unique, the number of unknowns  $m$  is less than or equal to the number of equations,  $J$ . |

#### 4.6 Discrete Channels with Memory

For a discrete channel with memory, each letter in the output sequence depends statistically both on the corresponding input and on past inputs and outputs (we assume implicitly throughout that the channel is non-anticipatory; that is, for a given current input and given input output history, the current output is statistically independent of future inputs). Without loss of generality, the input and output sequence, up to a given point, can be considered as the *state* of the channel at that point. In these terms, the statistical behavior of the channel is described by a joint probability measure on the output letter and state at a given time conditional on the current input letter and the previous state.

In constructing mathematical models of physical channels with memory, it is frequently more desirable to consider some physically meaningful parameter (such as the fading level of a slowly fading transmission path) as the state of the channel. In such cases, the channel will still be described by a probability measure on output and state given input and previous state, but the state might not be determined by previous inputs and outputs.

For ease of analysis, we shall consider only discrete finite state channels; that is, channels with a finite set of possible states with a probability assignment that is independent of time. More specifically, a discrete finite state channel has an input sequence  $\mathbf{x} = \cdots x_{-1}, x_0, x_1, \dots$ , an output sequence  $\mathbf{y} = \cdots y_{-1}, y_0, y_1, \dots$ , and a state sequence  $\mathbf{s} = \cdots s_{-1}, s_0, s_1, \dots$ . Each input letter  $x_n$  is a selection from an alphabet  $\{0, 1, \dots, K - 1\}$ , each output letter  $y_n$  is a selection from an alphabet  $\{0, 1, \dots, J - 1\}$ , and each state  $s_n$  is a selection from a set  $\{0, 1, \dots, A - 1\}$ . The channel is described statistically by a conditional probability assignment  $P(y_n s_n | x_n s_{n-1})$ . This assignment is independent of  $n$  and we can consider  $P$  as simply a function of four variables, each variable taking on integer values,  $0 \leq y_n \leq J - 1$ ,  $0 \leq s_n \leq A - 1$ ,  $0 \leq s_{n-1} \leq A - 1$ ,  $0 \leq x_n \leq K - 1$ . We assume that, conditional on  $x_n$  and  $s_{n-1}$ , the pair  $y_n, s_n$  is statistically independent of all outputs, inputs, and states prior to  $y_n$ ,  $x_n$ , and  $s_{n-1}$ , respectively.

The following examples are special cases of finite state channels (FSC's) in which, conditional on  $x_n$  and  $s_{n-1}$ , there is statistical independence between  $y_n$  and  $s_n$ ; that is,  $P(y_n s_n | x_n s_{n-1}) = P(y_n | x_n s_{n-1})q(s_n | x_n s_{n-1})$ . We can represent  $q$  by a graph and  $P$  by the usual kind of line diagrams as shown below in Figures 4.6.1 and 4.6.2. On the graphs, the states are indicated by small circles. The directed branches indicate transitions from one state to another, with the number on each branch representing the probability of that transition. If the probability of the transition depends on  $x_n$ , the value of  $x_n$

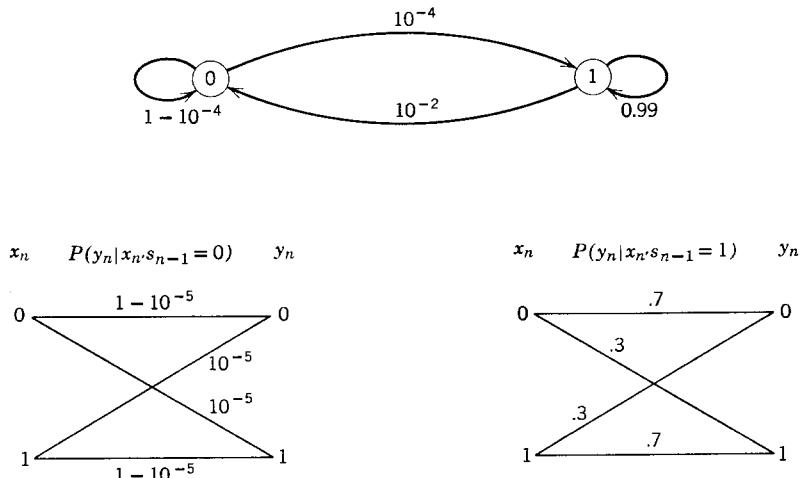


Figure 4.6.1. Finite state channel; Simple model of fading (bursty) channel.

corresponding to the probability is given in parentheses. For example, the uppermost branch in Figure 4.6.1 represents a transition from state 0 to state 1. The number on the branch,  $10^{-4}$ , is the conditional probability of going to state 1 given that the previous state is 0 and given either 0 or 1 for the current input. That is,  $q(s_n | x_n, s_{n-1}) = 10^{-4}$  for  $s_{n-1} = 0, s_n = 1$ . Likewise, the uppermost branch in Figure 4.6.2 indicates that, for this FSC,  $q(s_n | x_n, s_{n-1}) = 1$  for  $s_n = 1, s_{n-1} = 0, x_n = 1$ .

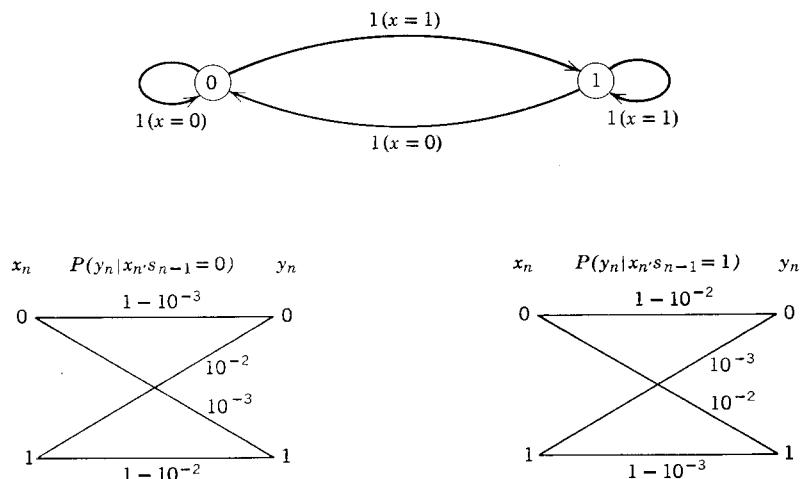


Figure 4.6.2. FSC; simple model of intersymbol interference.

We observe that the channel in Figure 4.6.1 has a tendency to persist in whatever state it is in, remaining in state 0 for a typical run of  $10^4$  digits and in state 1 for a typical run of 100 digits. This type of channel provides a conceptually simple (but not entirely adequate) model for slowly fading binary data links. Most of the time, the channel is in state 0, with virtually no errors between transmitted and received digits. Occasionally, the channel will enter state 1 (the faded state), and for a period of 100 digits or so, about  $\frac{3}{10}$  of the received channel digits will be in error. We can think of the channel as a BSC with a time-varying crossover probability, alternating between  $10^{-5}$  and 0.3. The state sequence is a Markov chain which determines this crossover probability. This is not an entirely satisfactory model of a fading binary data link partly because it is surprisingly difficult to analyze and partly because each real channel seems to require a different finite state model, usually of more than two states.

The channel in Figure 4.6.2 is a simple model of a channel with intersymbol interference. It can be seen that the state entered at any instant is the same as the input at that instant. Thus  $P(y_n | x_n s_{n-1})$  in this case gives the probability assignment on the current output conditional on the current and previous input. The probability of an error (that is,  $y_n \neq x_n$ ) is larger when  $x_n \neq x_{n-1}$  than when  $x_n = x_{n-1}$ . If we try to calculate the probability of an output given the current input, we find  $P(y_n | x_n) = Q(0)P(y_n | x_n, 0) + Q(1)P(y_n | x_n, 1)$  where  $Q$  is the probability assignment on  $x_{n-1}$ . It is important to notice that  $P(y_n | x_n)$  depends on the channel input assignment and thus is undefined in terms of the channel alone. This is a general characteristic of any finite state channel where the state sequence depends statistically on the input sequence. Thus, for this class of channels, not only is (4.6.1), that is,

$$P_N(\mathbf{y} | \mathbf{x}) = \prod_n P(y_n | x_n)$$

false, but also the probabilities appearing in the expression are undefined in terms of the channel alone.

We shall refer to channels such as Figure 4.6.1, where  $q(s_n | x_n, s_{n-1})$  is independent of  $x_n$ , as channels without intersymbol interference, and to channels such as Figure 4.6.2, where  $q(s_n | x_n, s_{n-1})$  takes on only the values 1 and 0 and depends on  $x_n$ , as channels with only intersymbol interference memory. In the first case, the memory is due to noise alone, and in the second case, to previous inputs alone. The general FSC, of course, is subject to both effects.

Since  $P_N(\mathbf{y} | \mathbf{x})$  is in general undefined for a FSC, we shall generally work with  $P_N(\mathbf{y}, s_N | \mathbf{x}, s_0)$ , the probability of a given output sequence  $\mathbf{y} = (y_1, \dots, y_N)$  and a final state  $s_N$  at time  $N$  given an input sequence  $\mathbf{x} = (x_1, \dots, x_N)$  and an initial state  $s_0$  at time 0. This quantity can be calculated

inductively from

$$P_N(\mathbf{y}, s_N \mid \mathbf{x}, s_0) = \sum_{s_{N-1}} P(y_N, s_N \mid x_N, s_{N-1}) P_{N-1}(\mathbf{y}_{N-1}, s_{N-1} \mid \mathbf{x}_{N-1}, s_0) \quad (4.6.1)$$

where  $\mathbf{x}_{N-1} = (x_1, \dots, x_{N-1})$  and  $\mathbf{y}_{N-1} = (y_1, \dots, y_{N-1})$ . The final state can be summed over to give

$$P_N(\mathbf{y} \mid \mathbf{x}, s_0) = \sum_{s_N} P_N(\mathbf{y}, s_N \mid \mathbf{x}, s_0) \quad (4.6.2)$$

The capacity of a FSC is a quantity that can be reasonably defined in several different ways. We shall give two definitions (which in general have different numerical values) and then show by some examples what the significance of each definition is. The lower capacity,  $\underline{C}$ , of a FSC is defined as

$$\underline{C} = \lim_{N \rightarrow \infty} \underline{C}_N \quad (4.6.3)$$

where

$$\underline{C}_N = \frac{1}{N} \max_{\mathbf{Q}_N} \min_{s_0} I_Q(\mathbf{X}^N; \mathbf{Y}^N \mid s_0) \quad (4.6.4)$$

$$I_Q(\mathbf{X}^N; \mathbf{Y}^N \mid s_0) = \sum_{\mathbf{x}} \sum_{\mathbf{y}} Q_N(\mathbf{x}) P_N(\mathbf{y} \mid \mathbf{x}, s_0) \log \frac{P_N(\mathbf{y} \mid \mathbf{x}, s_0)}{\sum_{\mathbf{x}'} Q_N(\mathbf{x}') P_N(\mathbf{y} \mid \mathbf{x}', s_0)} \quad (4.6.5)$$

Similarly, the upper capacity,  $\bar{C}$ , of a FSC is defined as

$$\bar{C} = \lim_{N \rightarrow \infty} \bar{C}_N \quad (4.6.6)$$

where

$$\bar{C}_N = \frac{1}{N} \max_{\mathbf{Q}_N} \max_{s_0} I_Q(\mathbf{X}^N; \mathbf{Y}^N \mid s_0) \quad (4.6.7)$$

The following theorem, which is proved in Appendix 4A, asserts the existence of these limits.

**Theorem 4.6.1.** For a finite state channel with  $A$  states,

$$\lim_{N \rightarrow \infty} \underline{C}_N = \sup_N \left[ \underline{C}_N - \frac{\log A}{N} \right] \quad (4.6.8)$$

$$\lim_{N \rightarrow \infty} \bar{C}_N = \inf_N \left[ \bar{C}_N + \frac{\log A}{N} \right] \quad (4.6.9)$$

It is clear from the definitions (4.6.4) and (4.6.7) that

$$\underline{C}_N \leq \bar{C}_N; \quad \text{all } N \quad (4.6.10)$$

It therefore follows immediately from the theorem that, for any  $N$ ,

$$-\frac{\log A}{N} + \underline{C}_N \leq \underline{C} \leq \bar{C} \leq \bar{C}_N + \frac{\log A}{N} \quad (4.6.11)$$

This relationship is useful in any attempt to calculate  $\underline{C}$  and  $\bar{C}$ , particularly when  $\underline{C} = \bar{C}$ , since it provides upper and lower bounds on the limit which become arbitrarily tight as  $N$  becomes large.

As a first example to understand the significance of  $\underline{C}$  and  $\bar{C}$ , consider Figure 4.6.3.

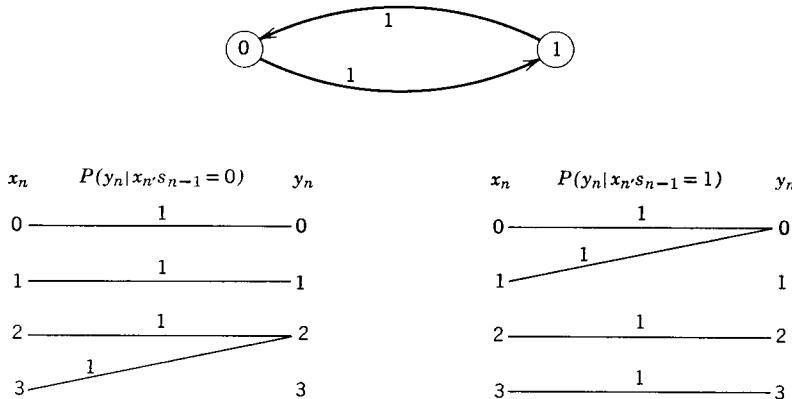


Figure 4.6.3. FSC; example of channel with ambiguous capacity.

If the initial state is  $s_0 = 0$ , it is not difficult to see (by considering the channel as a pair of parallel channels as in Problem 4.1 with one channel for each state) that the average mutual information is maximized by using statistically independent inputs, using  $Q(0) = Q(1) = Q(2) + Q(3) = \frac{1}{3}$  for the first and all odd-numbered inputs, and using  $Q(0) + Q(1) = Q(2) = Q(3) = \frac{1}{3}$  for the second and all even-numbered inputs. For this input distribution and starting state,  $I_Q(\mathbf{X}^N; \mathbf{Y}^N | s_0) = N \log 3$ . Similarly, if  $s_0 = 1$ , the average mutual information is maximized by reversing the above two single-letter input distributions. Thus  $\bar{C}_N = \log 3$  for all  $N$ . It can be seen, however, that to use the appropriate input distribution above, the initial state of the channel must be known at the transmitter, and thus  $\bar{C}$  in this example is the maximum average mutual information per digit that can be achieved if the transmitter can choose an input distribution to match the initial state.

If the initial state is unknown at the transmitter, then it is appropriate to choose an input distribution that yields a large average mutual information for each possible initial state. The lower capacity  $\underline{C}$  is the largest average

mutual information per letter than can be guaranteed for a fixed input distribution no matter what the starting state is. For this example, it can be shown that  $\underline{C} = \underline{C}_N = \frac{3}{2}$  bits for each  $N$  and that  $\underline{C}$  is achieved with statistically independent, equally likely inputs.

For an example such as this, there is no justification for saying that either  $\underline{C}$  or  $\bar{C}$  is the capacity of the channel. They simply apply to slightly different physical situations: one where some experimentation is possible to determine the phase of the state sequence, and one where such experimentation is not possible.

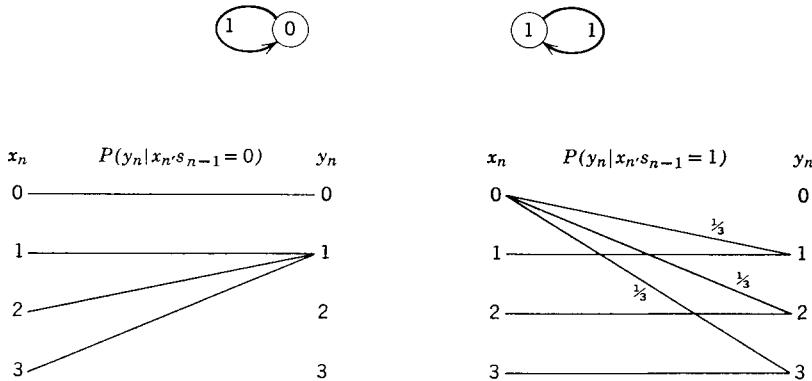


Figure 4.6.4.

In Figure 4.6.4, we have a different type of problem. The channel persists for all time in either state 0 or state 1. The capacity for the channel corresponding to state 0 is 1 bit, and the capacity of the channel corresponding to state 1 is  $\log_2 3$  bits. Then  $\bar{C} = \log_2 3$  bits. After a little calculation, it can be shown that  $\underline{C} \approx 0.965$  bits, achieved with independent inputs using the distribution  $Q(0) \approx 0.391$ ,  $Q(1) = Q(2) = Q(3) \approx 0.203$ . It is seen that  $\underline{C}$  is less than the capacity of either channel alone, necessitated by the fact that the same input distribution must give an average mutual information per digit of at least  $\underline{C}$  for each state.

Finally, in Figure 4.6.5, we have a “panic-button” channel. Input letter 2 is the panic button, and its use incapacitates the channel for all future time. We clearly have  $\bar{C} = 1$  bit and  $\underline{C} = 0$  bits.

The preceding examples were all somewhat pathological in the sense that the effect of the starting state did not die away with increasing time. We shall shortly define a class of FSC's where the effect of the starting state does die away with time, and we shall show for these channels that  $\bar{C} = \underline{C}$ . Before doing this, we establish two converses to the coding theorem for FSC's, one applying to  $\bar{C}$  and one to  $\underline{C}$ .

As in Section 4.3, we consider a discrete source of alphabet size  $M$  which produces letters at a rate of one letter each  $\tau_s$  seconds. The source sequence  $\mathbf{u}$ , after appropriate processing, is to be transmitted over a FSC and reproduced as a sequence  $\mathbf{v}$  at the destination. Let that channel be used once each  $\tau_c$  seconds and consider source sequences  $\mathbf{u} = (u_1, \dots, u_L)$  of arbitrary length  $L$  and channel sequences with a length given by the integer part of  $L\tau_s/\tau_c$ .

$$N = \lfloor L\tau_s/\tau_c \rfloor \quad (4.6.12)$$

Assume that the channel is in some initial state  $s_0$  and that data processors are placed between the source and channel and between the channel and

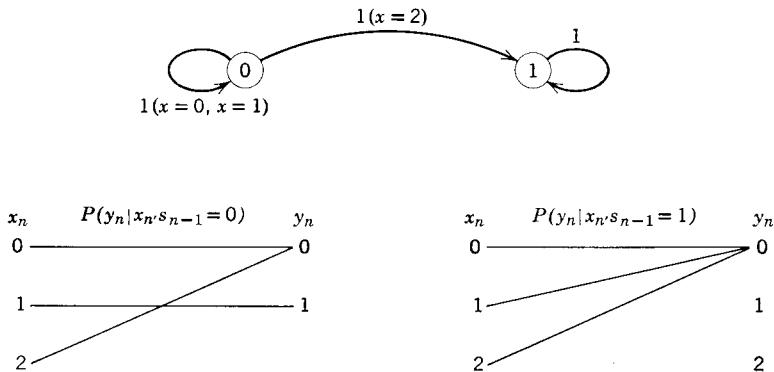


Figure 4.6.5. Panic-button channel.

destination. The source probabilities, the channel transition probabilities, the initial state, and the processors now determine a joint ensemble  $\mathbf{U}^L, \mathbf{X}^N, \mathbf{Y}^N, \mathbf{V}^L$ . This joint ensemble depends on the initial state,  $s_0$ , which for the time being we treat as being deterministic. We assume, as in Section 4.3, that the source sequence is *connected* to the destination through the  $N$  channel uses in the sense that, for all possible  $\mathbf{u}, \mathbf{x}, \mathbf{y}, \mathbf{v}$ , we have

$$P_N(\mathbf{y} \mid \mathbf{x}, s_0) = P_N(\mathbf{y} \mid \mathbf{x}, s_0, \mathbf{u}) \quad (4.6.13)$$

$$P(\mathbf{v} \mid \mathbf{y}, s_0) = P(\mathbf{v} \mid \mathbf{y}, s_0, \mathbf{x}, \mathbf{u}) \quad (4.6.14)$$

The data processing theorem then applies, giving us

$$I(\mathbf{U}^L; \mathbf{V}^L \mid s_0) \leq I(\mathbf{X}^N; \mathbf{Y}^N \mid s_0) \quad (4.6.15)$$

The  $s_0$  in (4.6.14) is merely inserted as a reminder that the joint ensemble under consideration depends on the given initial state.

Next, let  $\langle P_e(s_0) \rangle$  be the average error probability per source letter as given by (4.3.2). From Theorem 4.3.2, we have

$$\langle P_e(s_0) \rangle \log(M - 1) + \mathcal{H}(\langle P_e(s_0) \rangle) \geq \frac{1}{L} H(\mathbf{U}^L | \mathbf{V}^L s_0) \quad (4.6.16)$$

$$\geq \frac{1}{L} [H(\mathbf{U}^L | s_0) - I(\mathbf{U}^L; \mathbf{V}^L | s_0)] \quad (4.6.17)$$

$$\geq \frac{1}{L} [H(\mathbf{U}^L | s_0) - I(\mathbf{X}^N, \mathbf{Y}^N | s_0)] \quad (4.6.18)$$

We now make the further assumption that the source probabilities are independent of the initial state of the channel, and use Theorem 3.5.1 to obtain

$$\frac{1}{L} H(\mathbf{U}^L | s_0) \geq H_\infty(U) = \lim_{L \rightarrow \infty} \frac{1}{L} H(\mathbf{U}^L) \quad (4.6.19)$$

Substituting (4.6.19) and the definition of  $N$  [see (4.6.12)] into (4.6.18), we get

$$\langle P_e(s_0) \rangle \log(M - 1) + \mathcal{H}(\langle P_e(s_0) \rangle) \geq H_\infty(U) - \frac{\tau_s}{\tau_c N} I(\mathbf{X}^N; \mathbf{Y}^N | s_0) \quad (4.6.20)$$

We can now relate (4.6.20) to the upper and lower capacities of the channel. From the definition of  $\bar{C}_N$  in (4.6.7), we have

$$\frac{1}{N} I(\mathbf{X}^N; \mathbf{Y}^N | s_0) \leq \bar{C}_N \quad \text{for all } s_0 \quad (4.6.21)$$

Substituting (4.6.21) into (4.6.20) and going to the limit as  $L \rightarrow \infty, N \rightarrow \infty$ , we have, for all  $s_0$ ,

$$\langle P_e(s_0) \rangle \log(M - 1) + \mathcal{H}(\langle P_e(s_0) \rangle) \geq H_\infty(U) - \frac{\tau_s}{\tau_c} \bar{C} \quad (4.6.22)$$

It is important to notice that no assumption was made about  $\mathbf{X}^N$  being independent of  $s_0$  in the derivation of (4.6.22). Physically, this means that (4.6.22) is valid even if the input data processor knows the initial state of the channel and uses it in the mapping of source sequences into channel input sequences. Equation 4.6.22 is also valid whether or not the output data processor uses the initial state in mapping the sequences  $\mathbf{y}$  into the sequences  $\mathbf{v}$ .

To relate (4.6.20) to the lower capacity of the channel, we must make the additional assumption that  $\mathbf{X}^N$  is independent of the initial state  $s_0$ . Then,

from the definition of  $\underline{C}_N$ , we have, for some initial state  $s_0$ ,

$$\frac{1}{N} I(\mathbf{X}^N; \mathbf{Y}^N | s_0) \leq \underline{C}_N \quad (4.6.23)$$

Since  $\underline{C}_N \leq \underline{C} + (\log A)/N$  from Theorem 4.6.1, we have, for any  $N$  and some  $s_0$ ,

$$\langle P_e(s_0) \rangle \log(M - 1) + \mathcal{H}(\langle P_e(s_0) \rangle) \geq H_\infty(U) - \frac{\tau_s}{\tau_c} \left[ \underline{C} + \frac{\log A}{N} \right] \quad (4.6.24)$$

If there is a probability distribution  $q_0(s_0)$  on the initial states, then  $\langle P_e(s_0) \rangle$  can be averaged to get an overall error probability per source letter,

$$\langle P_e \rangle = \sum_{s_0} q_0(s_0) \langle P_e(s_0) \rangle$$

If  $q_{\min}$  is the smallest of these initial-state probabilities, then  $\langle P_e \rangle \geq q_{\min} \langle P_e(s_0) \rangle$  for each initial state, and (4.6.24) can be further bounded by

$$\frac{\langle P_e \rangle}{q_{\min}} \log(M - 1) + \mathcal{H}\left(\frac{\langle P_e \rangle}{q_{\min}}\right) \geq H_\infty(U) - \frac{\tau_s}{\tau_c} \left[ \underline{C} + \frac{\log A}{N} \right] \quad (4.6.25)$$

While (4.6.24) and (4.6.25) are not necessarily valid if the input data processor can make use of the initial state, they are valid whether or not the output data processor uses any knowledge of the state. We can summarize these results in the following theorem.

**Theorem 4.6.2.** Let a discrete stationary source have an alphabet size  $M$ , produce a letter each  $\tau_s$  seconds, and have a limiting entropy per letter of  $H_\infty(U)$ . Let a finite-state channel have an upper capacity  $\bar{C}$ , a lower capacity  $\underline{C}$ , and be used once each  $\tau_c$  seconds. In the limit as  $L \rightarrow \infty$ , if an  $L$  letter source sequence is connected to the destination by  $N = \lfloor L\tau_s/\tau_c \rfloor$  channel uses [that is, (4.6.13) and (4.6.14) are satisfied], then, independent of the initial channel state, the error probability per source digit must satisfy (4.6.22). If, in addition, the channel input ensemble  $\mathbf{X}^N$  is independent of the initial state, then for each  $N$  there is some initial state for which (4.6.24) is satisfied. If there is also a probability distribution on the initial state, with a smallest probability  $q_{\min}$ , then (4.6.25) is satisfied. 

---

### Indecomposable Channels

We now define a large class of channels, known as indecomposable FSC's, for which  $\underline{C} = \bar{C}$ . Roughly, an indecomposable FSC is a FSC for which the effect of the initial state dies away with time. More precisely, let

$$q_N(s_N | \mathbf{x}, s_0) = \sum_{\mathbf{y}} P_N(\mathbf{y}, s_N | \mathbf{x}, s_0).$$

A FSC is indecomposable if, for every  $\epsilon > 0$ , no matter how small, there exists an  $N_0$  such that for  $N \geq N_0$ ,

$$|q_N(s_N | \mathbf{x}, s_0) - q_N(s_N | \mathbf{x}, s_0')| \leq \epsilon \quad (4.6.26)$$

for all  $s_N$ ,  $\mathbf{x}$ ,  $s_0$ , and  $s_0'$ .

It should be verified by the reader that the channels of Figures 4.6.3 and 4.6.5 are not indecomposable. It can be seen that the channel of Figure 4.6.2 is indecomposable and, in fact, the left side of (4.6.26) is zero for all  $N \geq 1$ . We shall soon see that the channel of Figure 4.6.1 is also indecomposable.

For a fixed input sequence  $(x_1, x_2, \dots)$ , we can regard the state sequence  $(s_0, s_1, \dots)$  as a nonhomogeneous Markov chain. To avoid cluttering the notation in what follows, we shall suppress the dependence on the input sequence and, for example, use\*  $q(s_N | s_0)$  in place of  $q_N(s_N | \mathbf{x}, s_0)$ . We shall be interested in how  $q(s_N | s_0)$  depends on  $s_0$  for large  $N$ . As a measure of this dependence, define the distance  $d_N(s_0', s_0'')$  as

$$d_N(s_0', s_0'') = \sum_{s_N} |q(s_N | s_0') - q(s_N | s_0'')| \quad (4.6.27)$$

For  $N = 0$  and  $s_0' \neq s_0''$ , we take  $d_N(s_0', s_0'')$  to be 2.

The following lemma shows that, in terms of this distance measure, the dependence of  $s_N$  on  $s_0$  cannot increase with increasing  $N$ .

**LEMMA 4.6.1.** For any given input  $x_1, x_2, \dots$  and any given  $s_0', s_0''$ , the distance  $d_N(s_0', s_0'')$  as defined above is nonincreasing in  $N$ . 

---

*Proof.* For  $N \geq 1$ , we have

$$d_N(s_0', s_0'') = \sum_{s_N} |q(s_N | s_0') - q(s_N | s_0'')| \quad (4.6.28)$$

$$= \sum_{s_N} \left| \sum_{s_{N-1}} q(s_N | s_{N-1}) [q(s_{N-1} | s_0') - q(s_{N-1} | s_0'')] \right| \quad (4.6.29)$$

Upper bounding the magnitude of a sum by the sum of the magnitudes yields

$$d_N(s_0', s_0'') \leq \sum_{s_N} \sum_{s_{N-1}} q(s_N | s_{N-1}) |q(s_{N-1} | s_0') - q(s_{N-1} | s_0'')| \quad (4.6.30)$$

$$= \sum_{s_{N-1}} |q(s_{N-1} | s_0') - q(s_{N-1} | s_0'')| \quad (4.6.31)$$

$$= d_{N-1}(s_0', s_0'') \quad | \quad (4.6.32)$$

\* To be precise, this probability should be denoted

$$q_{s_N | x_1, \dots, x_N, s_0}(i | k_1, \dots, k_N, j)$$

the probability that  $s_N$  takes the value  $i$  given that  $\mathbf{x}$  is the sequence  $k_1, \dots, k_N$  and  $s_0$  is state  $j$ .

The next lemma now provides a condition under which  $d_N(s_0', s_0'')$  approaches 0 with increasing  $N$ .

LEMMA 4.6.2. Suppose that for some  $n > 0$ , some  $\delta > 0$ , and each  $N \geq 0$ , there is some choice of  $s_{N+n}$  such that

$$q(s_{N+n} | s_N) \geq \delta; \quad \text{all values of } s_N \quad (4.6.33)$$

Then  $d_N(s_0', s_0'')$  approaches 0 exponentially with  $N$  and is bounded by

$$d_N(s_0', s_0'') \leq 2(1 - \delta)^{(N/n)-1} \quad \text{_____}$$

*Proof.*

$$d_{N+n}(s_0', s_0'') = \sum_{s_{N+n}} |q(s_{N+n} | s_0') - q(s_{N+n} | s_0'')| \quad (4.6.34)$$

$$= \sum_{s_{N+n}} \left| \sum_{s_N} q(s_{N+n} | s_N) [q(s_N | s_0') - q(s_N | s_0'')] \right| \quad (4.6.35)$$

Define

$$a(s_{N+n}) = \min_{s_N} q(s_{N+n} | s_N) \quad (4.6.36)$$

Observing that

$$\sum_{s_N} a(s_{N+n}) [q(s_N | s_0') - q(s_N | s_0'')] = 0$$

we can rewrite (4.6.35) as

$$d_{N+n}(s_0', s_0'') = \sum_{s_{N+n}} \left| \sum_{s_N} [q(s_{N+n} | s_N) - a(s_{N+n})] [q(s_N | s_0') - q(s_N | s_0'')] \right| \quad (4.6.37)$$

Bounding the magnitude of a sum by the sum of the magnitudes and observing that  $q(s_{N+n} | s_N) - a(s_{N+n}) \geq 0$ , we obtain

$$d_{N+n}(s_0', s_0'') \leq \sum_{s_{N+n}} \sum_{s_N} [q(s_{N+n} | s_N) - a(s_{N+n})] |q(s_N | s_0') - q(s_N | s_0'')| \quad (4.6.38)$$

Summing over  $s_{N+n}$ , this becomes

$$\begin{aligned} d_{N+n}(s_0', s_0'') &\leq \left[ 1 - \sum_{s_{N+n}} a(s_{N+n}) \right] \sum_{s_N} |q(s_N | s_0') - q(s_N | s_0'')| \\ &= \left[ 1 - \sum_{s_{N+n}} a(s_{N+n}) \right] d_N(s_0', s_0'') \end{aligned} \quad (4.6.39)$$

By hypothesis,  $a(s_{N+n}) \geq \delta$  for at least one value of  $s_{N+n}$ , and thus

$$d_{N+n}(s_0', s_0'') \leq (1 - \delta) d_N(s_0', s_0'') \quad (4.6.40)$$

Applying this result for  $N = 0$ , then  $N = n$ , then  $N = 2n$ , and so forth, and recalling that  $d_0(s_0', s_0'') = 2$ , we obtain

$$d_{mn}(s_0', s_0'') \leq 2(1 - \delta)^m \quad (4.6.41)$$

Since  $d_N(s_0', s_0'')$  is nonincreasing in  $N$ , this completes the proof. |

The following theorem now provides a test for whether or not an FSC is indecomposable.

**Theorem 4.6.3.** A necessary and sufficient condition for an FSC to be indecomposable is that for some fixed  $n$  and each  $\mathbf{x}$ , there exists a choice for the  $n$ th state, say  $s_n$ , such that

$$q(s_n | \mathbf{x}, s_0) > 0 \quad \text{for all } s_0 \quad (4.6.42)$$

( $s_n$  above can depend upon  $\mathbf{x}$ ). Furthermore, if the channel is indecomposable,  $n$  above can always be taken as less than  $2^{A^2}$  where  $A$  is the number of channel states.

*Proof (Sufficiency).* If (4.6.42) is satisfied for some  $n$ , then since  $s_0'$  and  $\mathbf{x} = (x_1, \dots, x_n)$  can only take on a finite number of values, there is some  $\delta > 0$  such that

$$q(s_n | \mathbf{x}, s_0) \geq \delta \quad (4.6.43)$$

for all  $s_0$ , all  $\mathbf{x}$ , and some  $s_n$  depending on  $\mathbf{x}$ .

Also, since the channel probabilities are independent of time, we also have, for all  $N$  and some  $s_{N+n}$  depending on  $x_{N+1}, \dots, x_{N+n}$

$$q(s_{N+n} | x_{N+1}, \dots, x_{N+n}, s_N) > \delta \quad \text{for all } s_N \quad (4.6.44)$$

Thus the conditions of the previous lemma are satisfied, and

$$\sum_{s_N} |q(s_N | \mathbf{x}, s_0') - q(s_N | \mathbf{x}, s_0'')|$$

approaches zero exponentially as  $N \rightarrow \infty$ , uniformly in  $\mathbf{x}$ ,  $s_0'$ , and  $s_0''$ . Thus (4.6.26) is satisfied for large enough  $N$  and the channel is indecomposable.

*Necessity.* Pick  $\epsilon < 1/A$ , where  $A$  is the number of states, pick  $N$  large enough that (4.6.26) is satisfied, and for a given  $s_0$  and  $\mathbf{x}$ , pick  $s_N$  such that  $q(s_N | \mathbf{x}, s_0) \geq 1/A$ . Then from (4.6.26),  $q(s_N | \mathbf{x}, s_0') > 0$  for all  $s_0'$ , and the condition of the theorem is satisfied for  $n$  equal to this  $N$ .

*Prove that  $n < 2^{A^2}$ .* For a given  $n$ ,  $\mathbf{x}$ , define the connectivity matrix,

$$T_{n,\mathbf{x}}(s_0, s_n) = \begin{cases} 1; & q(s_n | \mathbf{x}, s_0) > 0 \\ 0; & q(s_n | \mathbf{x}, s_0) = 0 \end{cases} \quad (4.6.45)$$

This is an  $A$  by  $A$  matrix of 1's and 0's, with rows labeled by values of  $s_0$ ,

columns by values of  $s_n$ . A given entry is 1 if that  $s_n$  can be reached from that  $s_0$  with the given  $\mathbf{x}$ . Since

$$q(s_n \mid \mathbf{x}, s_0) = \sum q(s_n \mid x_n s_{n-1}) q(s_{n-1} \mid \mathbf{x}_{n-1}, s_0)$$

we can express  $T_{n,\mathbf{x}}(s_0, s_n)$  in terms of  $T_{n-1,\mathbf{x}}$  by

$$T_{n,\mathbf{x}}(s_0, s_n) = \begin{cases} 1; & T_{n-1,\mathbf{x}}(s_0, s_{n-1}) q(s_n \mid x_n s_{n-1}) > 0 \text{ for some } s_{n-1} \\ 0; & \text{otherwise} \end{cases} \quad (4.6.46)$$

Since there are only  $2^{A^2} - 1$  non-zero  $A$  by  $A$  matrices with binary entries, the sequence of matrices  $T_{n,\mathbf{x}}(s_0, s_n)$ ,  $n = 1, \dots, 2^{A^2}$  must contain two matrices that are the same, say for  $i < j \leq 2^{A^2}$ . If for this  $(x_1, \dots, x_j)$ , we choose  $x_{j+N} = x_{i+N}$  for all  $N \geq 1$ , then from (4.6.46), we see that  $T_{j+N,\mathbf{x}} = T_{i+N,\mathbf{x}}$  for all  $N \geq 1$ . For this choice of  $\mathbf{x}$ , if  $T_{n,\mathbf{x}}$  has no column of 1's for  $n \leq j$ , it will have no column of 1's for any larger  $n$ . But this means that, for this  $\mathbf{x}$ , there is no  $n, s_n$  for which  $q(s_n \mid \mathbf{x}, s_0) > 0$  for all  $s_0$ , and thus the channel is not indecomposable. Thus, for the channel to be indecomposable, for each  $\mathbf{x}$ ,  $T_{n,\mathbf{x}}$  must have a column of 1's for some  $n \leq 2^{A^2}$ . Finally, if  $T_{n,\mathbf{x}}$  has a column of 1's for some  $n$ , it follows from (4.6.46) that it has a column of 1's for all larger  $n$ , and thus, for an indecomposable channel there is some smallest  $n \leq 2^{A^2}$  for which  $T_{n,\mathbf{x}}$  has a column of 1's for all  $\mathbf{x}$ . |

For the channel of Figure 4.6.1, (4.6.54) is satisfied for  $n = 1$ , and thus that channel is indecomposable.

**Theorem 4.6.4.** For an indecomposable FSC,

$$\underline{C} = \bar{C} \quad (4.6.47)$$

---

*Proof.* For arbitrary  $N$ , let  $Q_N(\mathbf{x})$  and  $s_0'$  be the input distribution and initial state that maximize  $I_Q(\mathbf{X}^N; \mathbf{Y}^N \mid s_0)$  and let  $s_0''$  denote the initial state that minimizes  $I_Q$  for the same input distribution. Thus, by the definition of  $\bar{C}_N$  and  $\underline{C}_N$ ,

$$\bar{C}_N = \frac{1}{N} I_Q(\mathbf{X}^N; \mathbf{Y}^N \mid s_0') \quad (4.6.48)$$

$$\underline{C}_N \geq \frac{1}{N} I_Q(\mathbf{X}^N; \mathbf{Y}^N \mid s_0'') \quad (4.6.49)$$

Now let  $n + l = N$ , where  $n$  and  $l$  are positive integers. Let  $\mathbf{X}_1$  denote the ensemble of input sequences  $\mathbf{x}_1 = (x_1, \dots, x_n)$  and  $\mathbf{X}_2$  denote the ensemble of sequences  $\mathbf{x}_2 = (x_{n+1}, \dots, x_N)$ , according to the input assignment  $Q_N(\mathbf{x})$ . Similarly, let  $\mathbf{Y}_1$  and  $\mathbf{Y}_2$  denote the ensembles of outputs  $\mathbf{y}_1 = (y_1, \dots, y_n)$

and  $\mathbf{y}_2 = (y_{n+1}, \dots, y_N)$ , conditional on  $s_0'$ . We then have

$$\bar{C}_N = \frac{1}{N} [I(\mathbf{X}_1; \mathbf{Y}_1 \mathbf{Y}_2 | s_0') + I(\mathbf{X}_2; \mathbf{Y}_1 | \mathbf{X}_1, s_0') + I(\mathbf{X}_2; \mathbf{Y}_2 | \mathbf{X}_1 \mathbf{Y}_1, s_0')] \quad (4.6.50)$$

The probability of a joint element in the above ensembles, including the state at time  $n$ , can be expressed by

$$Q(\mathbf{x}_1)Q(\mathbf{x}_2 | \mathbf{x}_1)P_n(\mathbf{y}_1, s_n | \mathbf{x}_1, s_0')P_l(\mathbf{y}_2 | \mathbf{x}_2, s_n) \quad (4.6.51)$$

It can be seen from this that the second term on the right of (4.6.50) is zero. Also, the first term is upper bounded by  $n \log K$  since there are  $K^n$  elements in the  $\mathbf{X}_1$  ensemble. Finally, from Lemma 4A.2, the last term is changed by, at most,  $\log A$  by conditioning the mutual information on  $S_n$ . Thus

$$\bar{C}_N \leq \frac{1}{N} [n \log K + \log A + I(\mathbf{X}_2; \mathbf{Y}_2 | \mathbf{X}_1 \mathbf{Y}_1 S_n, s_0')] \quad (4.6.52)$$

Lower bounding  $\underline{C}_N$  in the same way, using  $s_0''$  for  $s_0'$ , and lower bounding the first term in (4.6.50) by 0, we obtain

$$\underline{C}_N \geq \frac{1}{N} [-\log A + I(\mathbf{X}_2; \mathbf{Y}_2 | \mathbf{X}_1 \mathbf{Y}_1 S_n, s_0'')] \quad (4.6.53)$$

Observing from (4.6.51) that the conditioning on  $\mathbf{Y}_1$  can be dropped in (4.6.52) and (4.6.53), we obtain

$$\begin{aligned} \bar{C}_N - \underline{C}_N &\leq \frac{1}{N} [n \log K + 2 \log A \\ &\quad + I(\mathbf{X}_2; \mathbf{Y}_2 | \mathbf{X}_1 S_n, s_0') - I(\mathbf{X}_2; \mathbf{Y}_2 | \mathbf{X}_1 S_n, s_0'')] \end{aligned} \quad (4.6.54)$$

$$\begin{aligned} &= \frac{1}{N} [n \log K + 2 \log A \\ &\quad + \sum_{\mathbf{x}_1} Q(\mathbf{x}_1) \sum_{s_N} [q(s_n | \mathbf{x}_1, s_0') - q(s_n | \mathbf{x}_1, s_0'')] I(\mathbf{X}_2; \mathbf{Y}_2 | s_n, \mathbf{x}_1)] \end{aligned} \quad (4.6.55)$$

We can further upper bound this by taking the magnitude of each term in the sum and by upper bounding  $I$  for each  $s_n$  and  $\mathbf{x}_1$  by  $l \log K$ . This yields

$$\bar{C}_N - \underline{C}_N \leq \frac{1}{N} [n \log K + 2 \log A + \bar{d}_n(s_0', s_0'') l \log K]$$

where  $\bar{d}_n$  is an upper bound on  $d_n(s_0', s_0'')$  valid for all  $\mathbf{x}_1$ . By definition of an indecomposable channel, for any  $\epsilon > 0$ , we can choose  $n$  so that  $\bar{d}_n < \epsilon$ . For this fixed  $n$ ,

$$\lim_{N \rightarrow \infty} \bar{C}_N - \underline{C}_N \leq \epsilon \log K \quad (4.6.56)$$

Since  $\epsilon > 0$  is arbitrary and  $\bar{C}_N \geq \underline{C}_N$ , this completes the proof.



While the indecomposability of a FSC is sufficient to show that  $\underline{C} = \bar{C}$ , it turns out that many decomposable FSC's also have  $\underline{C} = \bar{C}$ . Channels with only intersymbol interference memory are particularly easy to analyze in this respect. Suppose that such a channel is completely connected, that is, that each state can be reached from each other state by some finite sequence of inputs. Suppose further that there is some sequence of inputs that will drive the channel into some known state [that is,  $q(s_n | \mathbf{x}, s_0) = 1$  for all  $s_0$  for the given  $\mathbf{x}$ ]. Then there is also a finite input sequence that will drive the channel into any desired state, and from that point on,  $\bar{C}$  can be approached arbitrarily closely, so that  $\bar{C} = \underline{C}$ . It can be shown (see Problem 4.26) that, if the channel can be driven into a known state at all, it can be driven into such a state with, at most,  $2^A$  inputs.

### Summary and Conclusions

In this chapter we developed the notion of a probabilistic model for a communication channel. For discrete memoryless channel models and finite-state channel models, we defined the channel capacity in terms of maximum average mutual information. The major result of the chapter is the converse to the coding theorem which states that reliable communication is impossible over a channel if the rate of the source (that is, the source entropy in bits per unit time) exceeds the capacity of the channel (in bits per unit time). We introduced the theory of convex functions, which is a useful tool throughout information theory, and showed how to use this theory in finding the capacity of a discrete memoryless channel. For finite-state channels we introduced two meaningful definitions of capacity, but showed that they were the same for indecomposable channels.

### Historical Notes and References

The general structure and development in this chapter are due to Shannon (1948). The converse to the coding theorem, Theorem 4.3.4, is due to Gallager (1964), and is based on Theorem 4.3.1 which is due to Fano (1952). The observation that Theorem 4.3.4 applies to sources with memory as well as memoryless sources was made by Reiffen (1966). Theorem 4.4.1 is essentially a special case of a general convex programming result by Kuhn and Tucker (1951). Its application to channel capacity was discovered independently by Eisenberg (1962) and Gallager (1962), but the necessity of (4.5.1) in the theorem is due to Shannon (1948).

Part 1 of Theorem 4.6.1 and part 2 of Theorem 4.6.2 are due to Yudkin (1967), although Blackwell, Breiman, and Thomasian (1958) earlier established a weaker converse to the coding theorem for indecomposable FSC's. (The indecomposable channels of Blackwell, Breiman, and Thomasian

constitute the same class as the indecomposable channels discussed here. The reader can verify this, after reading the Blackwell, Breiman, and Thomasian paper, by observing that, if a Markov chain has a periodic set of states with period  $m$ , then the  $m$ th power of the matrix corresponds to a decomposable chain with at least  $m$  closed sets of states.) The last part of Theorem 4.6.3 is due to Thomasian (1963).

## APPENDIX 4A

We begin with two lemmas.

**LEMMA 1.** Let  $XYZS$  be a joint ensemble and let  $S$  contain  $A$  sample points. Then

$$|I(X; Y | ZS) - I(X; Y | Z)| \leq \log A \quad (4A.1)$$


---

*Proof.* We can expand  $I(XS; Y | Z)$  in the following ways:

$$I(XS; Y | Z) = I(X; Y | Z) + I(S; Y | ZX) \quad (4A.2)$$

$$= I(X; Y | ZS) + I(S; Y | Z) \quad (4A.3)$$

The final term in both (4A.2) and (4A.3) is nonnegative and upper bounded by  $H(S) \leq \log A$ . Thus, equating the right-hand sides of (4A.2) and (4A.3), we obtain (4A.1). |

**LEMMA 2.** Let  $a_N$ ,  $N = 1, 2, \dots$  be a bounded sequence of numbers and let

$$\bar{a} = \sup_N a_N \quad \text{and} \quad \underline{a} = \inf_N a_N$$

(By a bounded sequence, we mean that  $\bar{a} < \infty$  and  $\underline{a} > -\infty$ .) Assume that, for all  $n \geq 1$ , and all  $N > n$ ,

$$a_N \geq \frac{n}{N} a_n + \frac{N-n}{N} a_{N-n} \quad (4A.4)$$

Then

$$\lim_{N \rightarrow \infty} a_N = \bar{a} \quad (4A.5)$$

Conversely, if for all  $n \geq 1$  and  $N > n$ ,

$$a_N \leq \frac{n}{N} a_n + \frac{N-n}{N} a_{N-n} \quad (4A.6)$$

we have

$$\lim_{N \rightarrow \infty} a_N = \underline{a} \quad (4A.7)$$


---

*Proof.* Assume (4A.4) is valid and, for any  $\epsilon > 0$ , choose  $n$  to satisfy

$$a_n \geq \bar{a} - \epsilon \quad (4A.8)$$

Choosing  $N = 2n$ , (4A.4) becomes

$$a_{2n} \geq \frac{a_n}{2} + \frac{a_n}{2} \geq \bar{a} - \epsilon \quad (4A.9)$$

Similarly, choosing  $N = mn$  for any integer  $m \geq 2$ ,

$$a_{mn} \geq \frac{a_n}{m} + \frac{(m-1)a_{(m-1)n}}{m} \quad (4A.10)$$

Using induction, we assume that  $a_{(m-1)n} \geq \bar{a} - \epsilon$ , and (4A.10) then implies that  $a_{mn} \geq \bar{a} - \epsilon$ . Since the inductive hypothesis is true for  $m = 2, 3$ , we have

$$a_{mn} \geq \bar{a} - \epsilon, \quad \text{all } m \geq 1 \quad (4A.11)$$

Now, for any  $N > n$ , we can represent  $N$  as  $mn + j$  where  $0 \leq j \leq n - 1$ . Using  $j$  in place of  $n$  in (4A.4), we have

$$\begin{aligned} a_N &\geq \frac{j}{N} a_j + \frac{N-j}{N} a_{mn} = a_{mn} + (j/N)(a_j - a_{mn}) \\ &\geq \bar{a} - \epsilon + (n/N)(\underline{a} - \bar{a}) \end{aligned} \quad (4A.12)$$

It follows that, for all sufficiently large  $N$ ,  $a_N \geq \bar{a} - 2\epsilon$ . Since  $a_N \leq \bar{a}$ , and  $\epsilon$  is arbitrary, (4A.5) follows. Equation (4A.7) follows from (4A.6) by observing that (4A.6) implies that (4A.4) applies to the sequence  $-a_n$  and thus

$$\lim -a_n = \sup -a_n = -\inf a_n \mid \quad (4A.13)$$

*Proof of Theorem 4.6.1*

$$\left[ \lim_{N \rightarrow \infty} \underline{C}_N = \sup_N \left( \underline{C}_N - \frac{\log A}{N} \right) \right]$$

For arbitrary positive integers  $n$  and  $l$ , let  $\mathbf{Q}_n$  and  $\mathbf{Q}_l$  be input distributions that achieve  $\underline{C}_n$  and  $\underline{C}_l$ , respectively. Let  $N = n + l$  and choose  $\mathbf{Q}_N$  as

$$\mathbf{Q}_N(\mathbf{x}) = \mathbf{Q}_n(\mathbf{x}_1)\mathbf{Q}_l(\mathbf{x}_2) \quad (4A.14)$$

where  $\mathbf{x} = (x_1, \dots, x_N)$ ,  $\mathbf{x}_1 = (x_1, \dots, x_n)$ , and  $\mathbf{x}_2 = (x_{n+1}, \dots, x_N)$ . Let  $\mathbf{X}_1$  and  $\mathbf{X}_2$  be the ensembles of sequences  $\mathbf{x}_1$  and  $\mathbf{x}_2$  and let  $\mathbf{Y}_1$  and  $\mathbf{Y}_2$  be the corresponding output ensembles. Since  $\mathbf{Q}_N(\mathbf{x})$  is not necessarily the input distribution that achieves  $\underline{C}_N$ , we have

$$\begin{aligned} N\underline{C}_N &\geq \min_{s_0} I(\mathbf{X}_1 \mathbf{X}_2; \mathbf{Y}_1 \mathbf{Y}_2 \mid s_0) \\ &= \min_{s_0} [I(\mathbf{X}_1; \mathbf{Y}_1 \mathbf{Y}_2 \mid s_0) + I(\mathbf{X}_2; \mathbf{Y}_1 \mathbf{Y}_2 \mid \mathbf{X}_1, s_0)] \end{aligned} \quad (4A.15)$$

The first term on the right above is lower bounded by

$$I(\mathbf{X}_1; \mathbf{Y}_1 \mathbf{Y}_2 | s_0) \geq I(\mathbf{X}_1; \mathbf{Y}_1 | s_0) \geq n\underline{C}_n \quad (4A.16)$$

The last term in (4A.15) can be rearranged as

$$I(\mathbf{X}_2; \mathbf{Y}_1 \mathbf{Y}_2 | \mathbf{X}_1, s_0) = I(\mathbf{X}_2; \mathbf{Y}_1 \mathbf{Y}_2 \mathbf{X}_1 | s_0) - I(\mathbf{X}_2; \mathbf{X}_1) \geq I(\mathbf{X}_2; \mathbf{Y}_2 | s_0) \quad (4A.17)$$

where we have used the fact that  $\mathbf{X}_1$  and  $\mathbf{X}_2$  are statistically independent (see 4A.14). From Lemma 1,  $I(\mathbf{X}_2; \mathbf{Y}_2 | s_0)$  is lower bounded by  $I(\mathbf{X}_2; \mathbf{Y}_2 | S_n, s_0) - \log A$ . Finally,

$$\begin{aligned} I(\mathbf{X}_2; \mathbf{Y}_2 | S_n, s_0) &= \sum_{s_n} q(s_n | s_0) I(\mathbf{X}_2; \mathbf{Y}_2 | s_n) \\ &\geq \min_{s_n} I(\mathbf{X}_2; \mathbf{Y}_2 | s_n) = l\underline{C}_l \end{aligned} \quad (4A.18)$$

Using (4A.16) and (4A.18) in (4A.15) and observing that the bounds are independent of  $s_0$ , we have

$$N\underline{C}_N \geq n\underline{C}_n + l\underline{C}_l - \log A$$

or

$$N \left[ \underline{C}_N - \frac{\log A}{N} \right] \geq n \left[ \underline{C}_n - \frac{\log A}{n} \right] + l \left[ \underline{C}_l - \frac{\log A}{l} \right] \quad (4A.19)$$

Thus the sequence  $\underline{C}_N - (\log A)/N$  satisfies (4A.4) and, by Lemma 2, we have

$$\lim_{N \rightarrow \infty} \underline{C}_N = \lim_{N \rightarrow \infty} \left[ \underline{C}_N - \frac{\log A}{N} \right] = \sup_N \left[ \underline{C}_N - \frac{\log A}{N} \right] \quad (4A.20)$$

*Prove that*  $\lim_{N \rightarrow \infty} \bar{C}_N = \inf_N [\bar{C}_N + (\log A)/N]$ . Let  $N = n + l$  for arbitrary positive integers  $n, l$ , and let  $\mathbf{Q}_N$  and  $s_0$  be the input distribution and initial state that achieves  $\bar{C}_N$ . Let  $\mathbf{X}_1$  and  $\mathbf{X}_2$  be the resulting ensembles of input sequences  $\mathbf{x}_1 = (x_1, \dots, x_n)$  and  $\mathbf{x}_2 = (x_{n+1}, \dots, x_N)$ , and let  $\mathbf{Y}_1, \mathbf{Y}_2$  be the resulting output ensembles. Then

$$N\bar{C}_N = I(\mathbf{X}_1 \mathbf{X}_2; \mathbf{Y}_1 \mathbf{Y}_2 | s_0) \quad (4A.21)$$

$$= I(\mathbf{X}_1; \mathbf{Y}_1 | s_0) + I(\mathbf{X}_2; \mathbf{Y}_1 | \mathbf{X}_1, s_0) + I(\mathbf{X}_1 \mathbf{X}_2; \mathbf{Y}_2 | \mathbf{Y}_1, s_0) \quad (4A.22)$$

The probability assignment on this ensemble (including  $s_n$ , the state at time  $n$ ) can be expressed as

$$Q_n(\mathbf{x}_1) Q_l(\mathbf{x}_2 | \mathbf{x}_1) P_n(\mathbf{y}_1, s_n | \mathbf{x}_1, s_0) P_l(\mathbf{y}_2 | \mathbf{x}_2 s_n) \quad (4A.23)$$

It can be seen from this that, conditional on  $\mathbf{x}_1$  and  $s_0$ ,  $\mathbf{x}_2$  and  $\mathbf{y}_1$  are statistically independent. Thus, for the second term in (4A.22),

$$I(\mathbf{X}_2; \mathbf{Y}_1 | \mathbf{X}_1, s_0) = 0 \quad (4A.24)$$

Also, the first term in (4A.22) satisfies

$$I(\mathbf{X}_1; \mathbf{Y}_1 | s_0) \leq n\bar{C}_n \quad (4A.25)$$

Using Lemma 1, the final term of (4A.22) is upper bounded by

$$\begin{aligned}
 I(\mathbf{X}_1\mathbf{X}_2; \mathbf{Y}_2 | \mathbf{Y}_1, s_0) &\leq I(\mathbf{X}_1\mathbf{X}_2; \mathbf{Y}_2 | \mathbf{Y}_1\mathbf{S}_n, s_0) + \log A \\
 &= H(\mathbf{Y}_2 | \mathbf{Y}_1\mathbf{S}_n, s_0) - H(\mathbf{Y}_2 | \mathbf{X}_2\mathbf{S}_n, s_0) + \log A \\
 &\leq H(\mathbf{Y}_2 | \mathbf{S}_n, s_0) - H(\mathbf{Y}_2 | \mathbf{X}_2\mathbf{S}_n, s_0) + \log A \\
 &= \sum_{s_n} q(s_n | s_0) I(\mathbf{Y}_2; \mathbf{X}_2 | s_n) + \log A \\
 &\leq l\bar{C}_l + \log A
 \end{aligned} \tag{4A.26}$$

Substituting (4A.24), (4A.25), and (4A.26) into (4A.22), we have

$$N\bar{C}_N \leq n\bar{C}_n + l\bar{C}_l + \log A \tag{4A.27}$$

or

$$N\left[\bar{C}_N + \frac{\log A}{N}\right] \leq n\left[\bar{C}_n + \frac{\log A}{n}\right] + l\left[\bar{C}_l + \frac{\log A}{l}\right] \tag{4A.28}$$

From Lemma 2, it follows that

$$\lim_{N \rightarrow \infty} \bar{C}_N = \lim_{N \rightarrow \infty} \left[ \bar{C}_N + \frac{\log A}{N} \right] = \inf_N \left[ \bar{C}_N + \frac{\log A}{N} \right] \tag{4A.29}$$

## *Chapter 5*

### THE NOISY-CHANNEL CODING THEOREM

#### 5.1 Block Codes

In the last chapter, we showed that, if the information from a given source is to be transmitted over a given channel, and if the entropy of the source per unit time is *greater* than the channel capacity per unit time, then arbitrarily reliable reception of the source data is not possible. In this chapter, we shall show that if the source entropy is *less* than the capacity then, under certain conditions, arbitrarily reliable reception is possible.

Our approach here will be quite different from that in Chapter 4. There we were establishing a negative result; no matter how we encode or decode at rates above capacity, there is an unavoidable error probability. There, to avoid errors, we must reduce the data rate or improve the channel. To give a general proof of this, we had to avoid making any restrictions on the form of the encoder or decoder. In order to show that reliable transmission is possible at rates below capacity, however, we can place as many restrictions on the form of the encoder and decoder as we please. In fact, such restrictions often provide insight into techniques for achieving reliable transmission.

The first restriction that we shall make is to split the encoder and decoder into a source encoder and decoder and a channel encoder and decoder (see Figure 5.1.1). The source encoder transforms the source output into a stream of binary digits, the channel encoder transforms the binary data into channel input letters, the channel decoder attempts to transform the channel output into the original binary stream, and the source decoder attempts to recreate the original source stream. This separation has obvious advantages from a practical standpoint since the binary data provides a standard type of interface between sources and channels. From a conceptual standpoint, this separation is even more important since it separates the problem of communicating in noise from the problem of source representation. The source representation problem has already been treated in Chapter 3. Thus, for the

rest of this chapter (except in the problems), we shall ignore the source and assume that it has been converted to binary data.

Let us assume that binary digits enter the channel encoder at a rate of 1 binary digit each  $\tau_s$  seconds, say, and that the channel is discrete in time, transmitting one channel digit each  $\tau_c$  seconds, for instance. We shall restrict our attention in this chapter to *block encoders*. These are encoders that separate the incoming binary data stream into equal-length sequences of, say,  $L$  binary digits each. There are  $M = 2^L$  different binary sequences of length  $L$  and the encoder provides a *code word* for each. Each code word is a

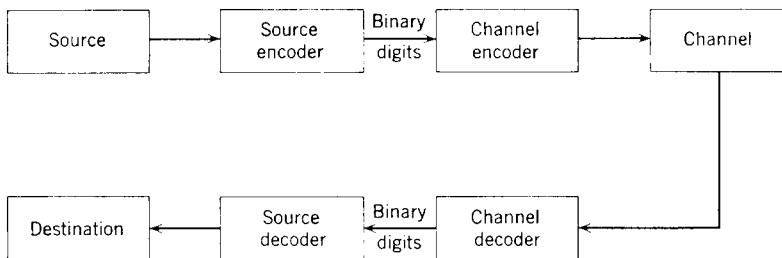


Figure 5.1.1.

sequence of a fixed number,  $N$ , of channel input letters. The number  $N$  is called the *block length* of the code and will be taken as the integer part of  $L\tau_s/\tau_c$ ,

$$N = \lfloor L\tau_s/\tau_c \rfloor \quad (5.1.1)$$

If  $L\tau_s/\tau_c$  is an integer, then the time required for  $L$  binary digits to enter the encoder is the same as the time required to transmit the code word of  $N$  channel digits. If  $L\tau_s/\tau_c$  is not an integer, then a “dummy digit” will occasionally have to be transmitted over the channel to keep the binary data stream and the channel sequence in synchronization with each other. At the receiver, a continuous stream of channel output digits is received. These are split into sequences of length  $N$  corresponding to the transmitted sequences of length  $N$ . The decoder guesses, on the basis of the received  $N$  sequence, what the corresponding  $L$  binary digits were. In practice, of course, there might be a problem in the receiver synchronizing itself to the transmitter, that is, in knowing when a block of  $N$  digits is starting. We shall ignore that problem here since, at this point, it would merely obscure the issues of coping with the channel noise.

We shall denote the  $M = 2^L$  code words corresponding to the  $2^L$  binary source sequences by  $\mathbf{x}_1 = (x_{1,1}, x_{1,2}, \dots, x_{1,N}), \dots, \mathbf{x}_m = (x_{m,1}, \dots, x_{m,N}), \dots, \mathbf{x}_M = (x_{M,1}, \dots, x_{M,N})$ . The correspondence between the integers 1 to  $M$

and the binary sequences is arbitrary and can be taken, for example, as the binary representation for the integers. Whatever this correspondence is, however, we assume that it is fixed and when the binary sequence corresponding to integer  $m$  enters the encoder, the code word  $\mathbf{x}_m$  is transmitted. Figure 5.1.2 gives an example of a block code of block length 5 for a channel with an input alphabet of three letters. If, for example, sequence (1,0) enters the encoder, the channel sequence (2,2,1,0,1) is transmitted on the channel. We assume here that  $\tau_s/\tau_c = \frac{5}{2}$  so that 5 channel digits can be transmitted in the time required for 2 binary digits to enter the encoder.

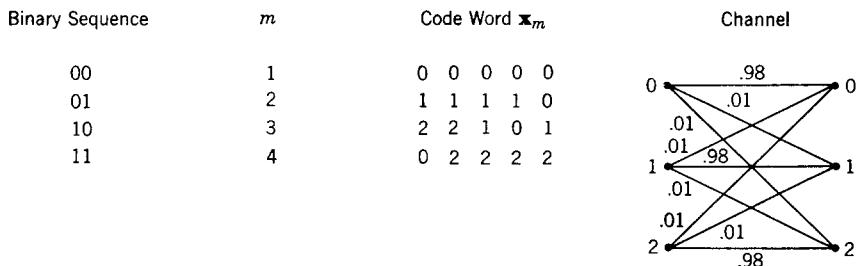


Figure 5.1.2. Code of block length  $N = 5$  with  $M = 4$  code words.

The rate  $R$  of a block code is defined as

$$R = \frac{\log M}{N} = \frac{L \log 2}{N} \quad (5.1.2)$$

If  $L\tau_s/\tau_c$  is an integer then, from (5.1.1), this reduces to  $R = (\tau_c/\tau_s) \log 2$ . Thus  $R$  in bits (that is, using  $\log_2$ ) is the number of binary digits entering the encoder per transmitted channel digit. When communication engineers speak of data rate, they generally mean the number of binary data digits per second that enter the transmitter. That is the same as the rate in bits that we are using here except that it is normalized to digits per second rather than digits per channel use. Throughout this chapter, we shall find it convenient to use natural logarithms, and  $R$  in natural units is  $(\ln M)/N$ . The conversion between  $R$  (in nats per digit) and the communication engineer's data rate (in binary digits per second) is thus

$$R = \text{data rate} \cdot \tau_c \ln 2 \quad (5.1.3)$$

It should be noted that  $R$  is not an entropy (although it can be interpreted as such if the binary source digits are independent and equally probable) and  $R$  is not, in general, the average mutual information for the channel.

If occasional "dummy digits" or other service bits must be transmitted on the channel to maintain synchronization, we shall still define  $R = (\ln M)/N$

but (5.1.3) will no longer be true exactly, the data rate being somewhat smaller than indicated by (5.1.3).

Let  $\mathbf{y} = (y_1, \dots, y_N)$  be the output sequence from the channel corresponding to a code word input. The function of the decoder is to use  $\mathbf{y}$  to guess which of the  $M = 2^L$  binary sequences entered the encoder. We say that a *block-decoding error* has occurred if the sequence guessed by the decoder differs from that entering the encoder. A block-decoding error implies that one or more errors have occurred in the sequence of  $L$  binary digits, but does not indicate how many. In most data transmission systems, we are interested both in how many errors occur and in how they are distributed. Neither digit nor block-error probabilities are completely adequate as measures of performance. For simplicity, block-error probability will be analyzed here. However, we shall see that the probability of error can be made so small for large  $N$  and  $L$  that the distinction between block and digit errors is of secondary importance.

Since we are interested in block errors, we can now drop the binary sequences from consideration and regard the encoding simply as a mapping from the integers 1 to  $M$  onto the code words  $\mathbf{x}_1$  to  $\mathbf{x}_M$ . If message  $m$  enters the encoder,  $\mathbf{x}_m$  is transmitted, and on the basis of the received sequence  $\mathbf{y}$ , the decoder produces an integer  $m'$ . An error occurs if  $m' \neq m$ .

The reason for restricting our attention here to block codes is not that they are better in any sense than other kinds of codes but simply that they are easier to treat. In the next chapter, we shall discuss an important class of nonblock codes called convolutional codes. These have some important advantages from the standpoint of implementation, but can be understood more thoroughly after understanding the behavior of block codes.

In attempting to construct good block codes, the major parameters of interest are the probability of a block decoding error, denoted  $P_e$ , the block length  $N$ , and the rate  $R$ . It should not be surprising that, if we decrease  $R$  (by slowing down the number of binary digits per second entering the encoder), then we can also decrease  $P_e$ . What is surprising is that, if  $R$  is less than the capacity of the channel, then we can hold  $R$  fixed and, by increasing  $N$ , find codes for which  $P_e$  becomes small exponentially with increasing  $N$ . This is the essence of the coding theorem to be proved in Section 5.6. There are, of course, prices to be paid by increasing the block length. One of these is in system delay. The first binary digit in a block of incoming data generally must be delayed by  $N\tau_c$  seconds before a code word can be formed, and then another  $N\tau_c$  seconds is necessary to transmit the code word before that binary digit can be decoded. Another problem is that the number of code words in a code is given by  $M = e^{NR}$  and, thus, we would expect the complexity of the encoder and decoder to grow rapidly with increasing  $N$ . This problem of complexity will be treated in Chapter 6, where we shall see that this growth

of complexity is much less severe than we might expect, but by no means insignificant. In most systems using coding, the problem of delay is far less significant than that of complexity.

## 5.2 Decoding Block Codes

A *minimum-error probability* decoding rule is a rule that minimizes the probability of decoding error for a given message ensemble, set of code words, and channel. Let  $P_N(\mathbf{y} \mid \mathbf{x}_m)$  be the probability of receiving a sequence  $\mathbf{y}$  given that the  $m$ th code word is transmitted. For a discrete memoryless channel, this is given in terms of the channel transition probabilities,  $P(y_n \mid x_{m,n})$ , by

$$P_N(\mathbf{y} \mid \mathbf{x}_m) = \prod_{n=1}^N P(y_n \mid x_{m,n}) \quad (5.2.1)$$

If the a priori message probabilities are  $\Pr(m)$ , then the a posteriori probability of message  $m$  conditioned on the received sequence  $\mathbf{y}$  is

$$\Pr(m \mid \mathbf{y}) = \frac{P_N(\mathbf{y} \mid \mathbf{x}_m)\Pr(m)}{\Pr(\mathbf{y})} \quad (5.2.2)$$

where

$$\Pr(\mathbf{y}) = \sum_{m=1}^M \Pr(m)P_N(\mathbf{y} \mid \mathbf{x}_m)$$

If the decoder decodes sequence  $\mathbf{y}$  into message  $m$ , the probability (given  $\mathbf{y}$ ) that the decoding is incorrect is  $1 - \Pr(m \mid \mathbf{y})$ . The decoder minimizes the probability of error by choosing  $m$  to maximize  $\Pr(m \mid \mathbf{y})$ . *Thus minimum-error probability decoding is defined by: decode the received sequence  $\mathbf{y}$  into an  $m'$  for which*

$$\Pr(m' \mid \mathbf{y}) \geq \Pr(m \mid \mathbf{y}) \quad \text{for all } m \neq m' \quad (5.2.3)$$

If, for a given  $\mathbf{y}$ ,  $\Pr(m \mid \mathbf{y})$  is maximized by several different values of  $m$ , it clearly makes no difference which of these is selected. Since the denominator in (5.2.2) is independent of  $m$ , an equivalent minimum-error probability rule is: decode  $\mathbf{y}$  into an  $m'$  for which

$$\Pr(m')P_N(\mathbf{y} \mid \mathbf{x}_{m'}) \geq \Pr(m)P_N(\mathbf{y} \mid \mathbf{x}_m); \quad \text{all } m \neq m' \quad (5.2.4)$$

*Maximum-likelihood decoding is an alternative type of decoding rule defined by: given  $\mathbf{y}$ , choose  $m'$  for which*

$$P_N(\mathbf{y} \mid \mathbf{x}_{m'}) \geq P_N(\mathbf{y} \mid \mathbf{x}_m); \quad \text{all } m \neq m' \quad (5.2.5)$$

The obvious advantage of maximum-likelihood decoding is that it can be used where the a priori message probabilities are unknown or unmeaningful. The name “maximum likelihood” is somewhat misleading since it does not

necessarily choose the message that is most likely given  $\mathbf{y}$ . Instead, it chooses the message for which the given  $\mathbf{y}$  is most likely, given  $m$  [compare (5.2.3) and (5.2.5)]. In the special case where the messages have equal a priori probabilities, it can be seen that (5.2.4) and (5.2.5) are equivalent and, for that case, maximum-likelihood decoding minimizes the error probability.

Another type of decoding rule, useful where unequal costs are associated with different kinds of errors, is *minimum-cost* decoding. Here  $\mathbf{y}$  is decoded into the  $m$  that minimizes the average cost (see Problem 5.1). This class of problems is considered from a more meaningful standpoint in Chapter 9.

Finally, in most practical applications of coding, it is necessary to choose decoding rules partly on the basis of ease of instrumentation; these problems will be discussed in the next chapter.

Thus far, we have been considering decoding rules as rules for guessing the message given the received channel sequence. If the noise is particularly bad, however, it is often desirable for the decoder to refuse to guess, in which case a *detected error* occurs. The capability to detect errors is particularly useful if the receiver can communicate back to the transmitter, for then the faulty blocks can be retransmitted.

*A decoding rule can now be formally defined as a mapping from the set  $\mathbf{Y}^N$  of channel output sequences into the set consisting of the  $M$  messages and the detected-error output.* We shall denote the set of sequences decoded into message  $m$  as  $Y_m$  and the complement of this set as  $Y_m^c$ . When the source produces message  $m$ , the code word  $\mathbf{x}_m$  is transmitted, and an error (either undetected or detected) occurs if the received sequence  $\mathbf{y}$  is in the set  $Y_m^c$ . Thus the probability of decoding error, given that message  $m$  is sent, is

$$P_{e,m} = \sum_{\mathbf{y} \in Y_m^c} P_N(\mathbf{y} \mid \mathbf{x}_m) \quad (5.2.6)$$

The overall probability of decoding error, if the messages have a priori probabilities  $\Pr(m)$ , is then given by

$$P_e = \sum_{m=1}^M \Pr(m) P_{e,m} \quad (5.2.7)$$

As an example, consider the code given in Figure 5.2.1. There are two code words,  $\mathbf{x}_1 = (0,0,0)$  and  $\mathbf{x}_2 = (1,1,1)$ . The decoding rule (which can be seen to be maximum likelihood for the BSC) is to decode each of the sequences  $(0,0,0)$ ,  $(0,0,1)$ ,  $(0,1,0)$ , and  $(1,0,0)$  into message 1 and the other sequences into message 2. Thus  $Y_1^c$  is the same as  $Y_2$  and is the set of sequences  $(0,1,1)$ ,  $(1,0,1)$ ,  $(1,1,0)$ , and  $(1,1,1)$ . For the BSC in Figure 5.2.1,  $P_3[(0,1,1) \mid (0,0,0)] = (1 - \epsilon)\epsilon^2$ , for example, and a similar calculation of  $P_3(\mathbf{y} \mid \mathbf{x}_1)$  for the other  $\mathbf{y}$  in  $Y_1^c$  yields  $P_{e,1} = 3(1 - \epsilon)\epsilon^2 + \epsilon^3$ .

Equations 5.2.5 and 5.2.7 are rather innocuous in appearance. However,

if the channel output alphabet contains  $J$  letters, then  $J^N$  terms appear in these sums. For moderate blocklengths such as  $N = 50$ , such a computation is far beyond the reach of modern-day computers. Even if such computations could be performed, they would give us little insight into choosing an appropriate block length for a code or an appropriate set of code words. Our approach here, rather than trying to calculate  $P_e$ , will be to find simple upper bounds to the probability of error that can be achieved. By so doing, we shall not only prove the coding theorem, but also gain considerable insight into the problems of choosing appropriate parameters for coding. We shall start out by considering the error probability for a set of 2 code words and then generalize the result to an arbitrarily large set of code words.

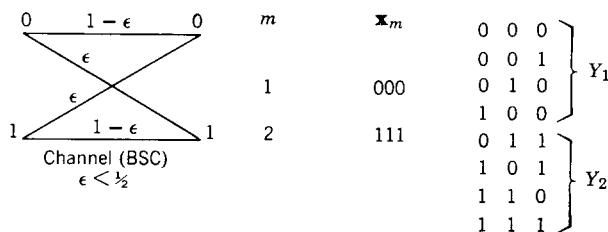


Figure 5.2.1. A code and decoding rule for  $N = 3, M = 2$ .

### 5.3 Error Probability for Two Code Words

Let  $\mathbf{x}_1$  and  $\mathbf{x}_2$  be two code words of length  $N$  and suppose that maximum likelihood decoding is to be used, decoding message 1 if  $P_N(\mathbf{y} \mid \mathbf{x}_1) > P_N(\mathbf{y} \mid \mathbf{x}_2)$  and decoding message 2 otherwise.

From (5.2.6), the error probability when message 1 is sent is

$$P_{e,1} = \sum_{\mathbf{y} \in Y_1^c} P_N(\mathbf{y} \mid \mathbf{x}_1)$$

For any  $\mathbf{y} \in Y_1^c$ , we can upper bound  $P_N(\mathbf{y} \mid \mathbf{x}_1)$  by

$$P_N(\mathbf{y} \mid \mathbf{x}_1) \leq P_N(\mathbf{y} \mid \mathbf{x}_1)^{1-s} P_N(\mathbf{y} \mid \mathbf{x}_2)^s; \quad \text{any } s, 0 < s < 1 \quad (5.3.1)$$

Equation 5.3.1 follows from the fact that, for  $\mathbf{y} \in Y_1^c$ ,  $P_N(\mathbf{y} \mid \mathbf{x}_2) \geq P_N(\mathbf{y} \mid \mathbf{x}_1)$  and thus  $P_N(\mathbf{y} \mid \mathbf{x}_2)^s \geq P_N(\mathbf{y} \mid \mathbf{x}_1)^s$ . Equation 5.3.1 is also valid for  $s \geq 1$ , but it will not be useful there. Substituting (5.3.1) into (5.2.6) and upper bounding by summing over all  $\mathbf{y}$ , we have

$$P_{e,1} \leq \sum_{\mathbf{y}} P_N(\mathbf{y} \mid \mathbf{x}_1)^{1-s} P_N(\mathbf{y} \mid \mathbf{x}_2)^s \quad \text{any } s, 0 < s < 1 \quad (5.3.2)$$

Bounding  $P_{e,2}$  in the same way, we have

$$P_{e,2} \leq \sum_{\mathbf{y}} P_N(\mathbf{y} \mid \mathbf{x}_2)^{1-r} P_N(\mathbf{y} \mid \mathbf{x}_1)^r \quad \text{any } r, 0 < r < 1 \quad (5.3.3)$$

If we now substitute  $1 - s$  for  $r$  in (5.3.3), we note that we have the same bound for  $P_{e,2}$  as for  $P_{e,1}$ . This entails no loss of generality since  $s$  is still arbitrary,  $0 < s < 1$ .

$$P_{e,m} \leq \sum_y P_N(y \mid \mathbf{x}_1)^{1-s} P_N(y \mid \mathbf{x}_2)^s \quad m = 1, 2 \quad 0 < s < 1 \quad (5.3.4)$$

We shall see later that, when  $s$  is appropriately chosen, the bound in (5.3.4) is surprisingly tight. If the channel is memoryless, (5.3.4) can be simplified as follows.

$$P_{e,m} \leq \sum_{y_1} \sum_{y_2} \cdots \sum_{y_N} \prod_{n=1}^N P(y_n \mid x_{1,n})^{1-s} P(y_n \mid x_{2,n})^s \quad (5.3.5)$$

Writing out the product, this becomes

$$\begin{aligned} P_{e,m} &\leq \sum_{y_1} P(y_1 \mid x_{1,1})^{1-s} P(y_1 \mid x_{2,1})^s \sum_{y_2} P(y_2 \mid x_{1,2})^{1-s} P(y_2 \mid x_{2,2})^s \cdots \\ &\quad \times \sum_{y_N} P(y_N \mid x_{1,N})^{1-s} P(y_N \mid x_{2,N})^s \\ P_{e,m} &\leq \prod_{n=1}^N \sum_{y_n} P(y_n \mid x_{1,n})^{1-s} P(y_n \mid x_{2,n})^s; \quad m = 1, 2 \end{aligned} \quad (5.3.6)$$

The sum over  $y_n$  in (5.3.6) is of fundamental importance in what follows and will be given additional interpretation in the next section. We denote it by

$$g_n(s) = \sum_{y_n} P(y_n \mid x_{1,n})^{1-s} P(y_n \mid x_{2,n})^s \quad (5.3.7)$$

Equation 5.3.6 then becomes

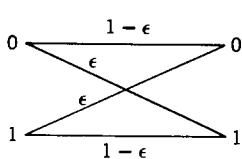
$$P_{e,m} \leq \prod_{n=1}^N g_n(s); \quad m = 1, 2 \quad 0 < s < 1 \quad (5.3.8)$$

The function  $g_n(s)$  is sketched for several channels in Figure 5.3.1. In each case, we take  $x_{1,n}$  as the channel input 0 and  $x_{2,n}$  as the channel input 1. For channels such as that in Figure 5.3.1b where some of the transition probabilities are zero,  $g_n(0)$  and  $g_n(1)$  can be indeterminate. For these cases, we define  $g_n(0)$  and  $g_n(1)$  by

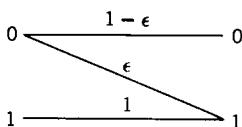
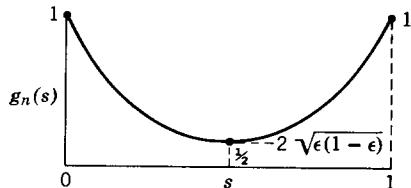
$$g_n(0) = \lim_{s \rightarrow 0^+} g_n(s) = \sum_{\substack{y_n \text{ for which} \\ P(y_n|x_{2,n}) \neq 0}} P(y_n \mid x_{1,n}) \quad (5.3.9)$$

$$g_n(1) = \lim_{s \rightarrow 1^-} g_n(s) = \sum_{\substack{y_n \text{ for which} \\ P(y_n|x_{1,n}) \neq 0}} P(y_n \mid x_{2,n}) \quad (5.3.10)$$

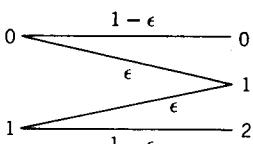
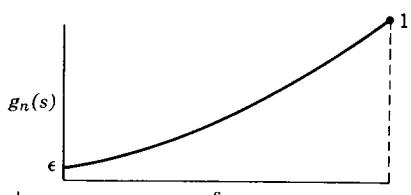
It can be seen that  $g_n(0)$  and  $g_n(1)$  are always less than or equal to 1. It also follows, from taking the second derivative, that  $g_n(s)$  is convex  $\cup$  for  $0 \leq s \leq 1$ . Consequently,  $g_n(s) \leq 1$  for  $0 \leq s \leq 1$ . It is shown in Problem



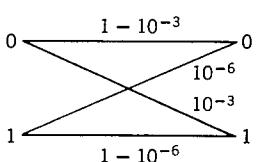
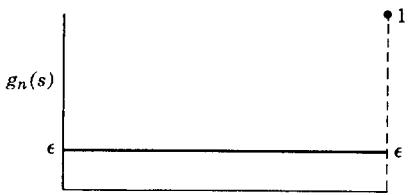
(a) BSC.



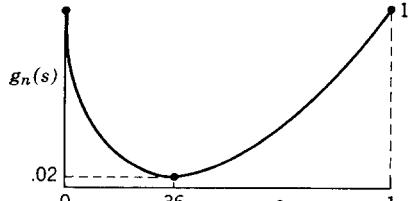
(b) Z channel.



(c) BEC.



(d) Asymmetric binary.



**Figure 5.3.1.** The function  $g_n(s) = \sum_{j=0}^{J-1} P(j \mid 0)^{1-s} P(j \mid 1)^s$ .

4.15a that  $g_n(s) < 1$  for  $0 < s < 1$  unless  $P(y_n \mid x_{1,n}) = P(y_n \mid x_{2,n})$  for all outputs  $y_n$ .

Using the definitions in (5.3.9) and (5.3.10), we see that (5.3.8) is a valid inequality for all  $s$ ,  $0 \leq s \leq 1$ . We clearly get the best bound by minimizing over  $s$ .

$$P_{e,m} \leq \min_{0 \leq s \leq 1} \prod_{n=1}^N g_n(s); \quad m = 1, 2 \quad (5.3.11)$$

**Example.** Consider the binary symmetric channel of Figure 5.3.1a. Let  $\mathbf{x}_1$  be a sequence of  $N$  zeros and  $\mathbf{x}_2$  be a sequence of  $N$  ones. Then

$$g_n(s) = \epsilon^{1-s}(1-\epsilon)^s + \epsilon^s(1-\epsilon)^{1-s}; \quad 1 \leq n \leq N$$

This is minimized at  $s = \frac{1}{2}$ , yielding

$$\min g_n(s) = g_n(\frac{1}{2}) = 2\sqrt{\epsilon(1-\epsilon)} \quad (5.3.12)$$

$$P_{e,m} \leq [2\sqrt{\epsilon(1-\epsilon)}]^N; \quad m = 1, 2 \quad (5.3.13)$$

For this simple example,  $P_{e,1}$  and  $P_{e,2}$  can be explicitly evaluated.  $P_N(\mathbf{y} | \mathbf{x}_2)$  will be greater than or equal to  $P_N(\mathbf{y} | \mathbf{x}_1)$  if the received sequence contains  $N/2$  or more ones. The probability of this when message 1 is transmitted (assuming  $N$  even) is

$$P_{e,1} = \sum_{i=N/2}^N \binom{N}{i} \epsilon^i (1-\epsilon)^{N-i} \quad (5.3.14)$$

The terms in this sum are terms in a binomial expansion centered around  $i = \epsilon N$ . Since  $\epsilon < \frac{1}{2}$ , the largest term in the sum is the first where  $i = N/2$ . Using the Stirling approximation to a factorial,  $N! \approx \sqrt{2\pi N} N^N e^{-N}$ , we have

$$\binom{N}{N/2} \approx \sqrt{\frac{2}{\pi N}} 2^N \quad (5.3.15)$$

$$P_{e,1} \approx \sqrt{\frac{2}{\pi N}} [2\sqrt{\epsilon(1-\epsilon)}]^N + \text{smaller terms} \quad (5.3.16)$$

In Problem 5.2c, the smaller terms in (5.3.16) are approximated, giving rise to an explicit approximation to  $P_{e,1}$ . The thing that interests us here, however, is the exponential dependence of  $P_{e,1}$  on  $N$  and the fact that this exponential dependence agrees with that in the bound of (5.3.13).

Since we have arbitrarily decided to decode message 2 if  $P_N(\mathbf{y} | \mathbf{x}_2) = P_N(\mathbf{y} | \mathbf{x}_1)$ , we have

$$P_{e,2} = \sum_{i=(N/2)+1}^N \binom{N}{i} \epsilon^i (1-\epsilon)^{N-i}$$

It is also shown, in Problem 5.2c, that the exponential dependence of  $P_{e,2}$  on  $N$  is the same as that of  $P_{e,1}$  and that the same exponential dependence results if  $N$  is odd.

For more complicated channels, it is much more difficult to obtain a good approximation to  $P_{e,1}$  and  $P_{e,2}$ . It turns out, however, that (5.3.11) always gives the correct exponential dependence of  $\frac{1}{2}(P_{e,1} + P_{e,2})$  on  $N$  [see Shannon, Gallager, and Berlekamp I (1967), section 3]. This will be discussed in more detail at the end of the next section.

## 5.4 The Generalized Chebyshev Inequality and the Chernoff Bound

In this section, the results of Section 5.3 will be rederived in a more general setting, thus providing additional insight into the technique used there. For maximum-likelihood decoding between two code words, when message 1 is transmitted, an error occurs if  $P_N(\mathbf{y} \mid \mathbf{x}_2) \geq P_N(\mathbf{y} \mid \mathbf{x}_1)$ . Equivalently, an error occurs if the log likelihood ratio,  $w(\mathbf{y})$ , satisfies

$$w(\mathbf{y}) \triangleq \ln \frac{P_N(\mathbf{y} \mid \mathbf{x}_2)}{P_N(\mathbf{y} \mid \mathbf{x}_1)} \geq 0 \quad (5.4.1)$$

For a memoryless channel, we can use (5.2.1) to represent  $w(\mathbf{y})$  as a sum of  $N$  terms,

$$w(\mathbf{y}) = \sum_{n=1}^N z_n(y_n) \quad (5.4.2)$$

where

$$z_n(y_n) = \ln \frac{P(y_n \mid x_{2,n})}{P(y_n \mid x_{1,n})} \quad (5.4.3)$$

Given that message 1 is transmitted, each  $z_n$ ,  $1 \leq n \leq N$ , is an independent random variable, taking on the values shown with the probability assignment  $P(y_n \mid x_{1,n})$ . Thus  $w(\mathbf{y})$  is a sum of independent random variables and  $P_{e,1}$  is given by

$$P_{e,1} = \Pr[w(\mathbf{y}) \geq 0 \mid \text{message 1 transmitted}] \quad (5.4.4)$$

Finding effective bounds on the probability that a sum of independent random variables will exceed a given number is a common problem both in information theory and in probability theory and therefore is worth considering in general.

First, suppose that  $t$  is a random variable taking on only nonnegative values. The simplest form of Chebyshev's inequality then states that, for arbitrary  $\delta > 0$ ,

$$\Pr(t \geq \delta) \leq \frac{\bar{t}}{\delta} \quad (5.4.5)$$

when  $\bar{t}$  is the mean value of  $t$ . In order to prove this, suppose that  $t$  is a discrete random variable with probability assignment  $P(t)$ . Then

$$\Pr(t \geq \delta) = \sum_{t \geq \delta} P(t) \leq \sum_{t \geq \delta} P(t) \frac{t}{\delta}$$

The inequality above results from the fact that  $t/\delta \geq 1$  over the region of summation. Since  $t/\delta$  is nonnegative for all  $t$ , we can further upper bound by summing over all  $t$ , obtaining (5.4.5). The same proof, with minor

notational changes, clearly applies to nondiscrete random variables. An equivalent form of (5.4.5) is obtained by letting  $\alpha$  be  $\delta/\bar{t}$ , yielding

$$\Pr(t \geq \alpha\bar{t}) \leq \frac{1}{\alpha} \quad (5.4.6)$$

As an example, to see how innocuous this inequality is, suppose that  $t$  is the height of a randomly chosen human being. If  $\bar{t} = 5$  feet, then (5.4.5) states that the probability of choosing a person over 10 ft tall is at most  $\frac{1}{2}$  and the probability of choosing a person over 50 ft tall is, at most,  $\frac{1}{10}$ .

Next, suppose that  $w$  is an arbitrary random variable with mean  $\bar{w}$  and variance  $\sigma^2$ . If we define  $t$  as  $t = (w - \bar{w})^2$ , then (5.4.5) becomes

$$\Pr[(w - \bar{w})^2 \geq \delta] \leq \frac{\sigma^2}{\delta} \quad (5.4.7)$$

Letting  $\epsilon = \sqrt{\delta}$ , we can rewrite this in the form in which Chebyshev's inequality is usually stated,

$$\Pr[|w - \bar{w}| \geq \epsilon] \leq \frac{\sigma^2}{\epsilon^2} \quad (5.4.8)$$

We can obtain a wide range of other inequalities, called generalized Chebyshev inequalities, by letting  $t$  be other functions of  $w$ . The inequality of particular interest here, usually called the Chernoff bound, results from taking  $t = e^{sw}$  for an arbitrary real number  $s$ . This yields

$$\Pr[e^{sw} \geq \delta] \leq \frac{e^{sw}}{\delta} \quad (5.4.9)$$

The expectation of  $e^{sw}$  is the moment-generating function of  $w$ ,

$$g_w(s) = \overline{e^{sw}} = \sum_w P(w)e^{sw} \quad (5.4.10)$$

Let  $\delta$  in (5.4.9) be  $e^{sA}$ , where  $A$  is an arbitrary real number. For  $s > 0$ ,  $e^{sw} \geq e^{sA}$  is equivalent to  $w \geq A$ , so that (5.4.9) becomes

$$\Pr[w \geq A] \leq e^{-sA} g_w(s); \quad \text{any } s > 0 \quad (5.4.11)$$

Likewise, if  $s < 0$ ,  $e^{sw} \geq e^{sA}$  is equivalent to  $w \leq A$ , yielding

$$\Pr[w \leq A] \leq e^{-sA} g_w(s); \quad s < 0 \quad (5.4.12)$$

The function  $e^{-sA} g_w(s)$  is sketched as a function of  $s$  in Figure 5.4.1. The function has the value 1 at  $s = 0$ , and its first derivative at  $s = 0$  is  $\bar{w} - A$ . The second derivative is always positive, as can be seen by the relation  $e^{-sA} g_w(s) = \overline{\exp [s(w - A)]}$ . It follows from this that, if  $A > \bar{w}$ , then the bound in (5.4.12) is greater than 1 for all  $s < 0$  and, therefore, useless.

Likewise, if  $A < \bar{w}$ , the bound in (5.4.11) is useless. In other words, (5.4.11) and (5.4.12) can be used only in bounding the “tails” of a distribution.

Since  $e^{-sA}g_w(s)$  is convex  $\cup$  in  $s$ , we may solve for the value of  $s$  that provides the tightest bound simply by finding a stationary point of the function. This yields

$$A = \frac{dg_w(s)}{ds} / g_w(s) \quad (5.4.13)$$

In most applications, it is more convenient to leave  $s$  as a free parameter rather than solving for  $s$  from (5.4.13).

The bounds in (5.4.11) and (5.4.12) are useful primarily where  $w$  is a sum of statistically independent random variables,

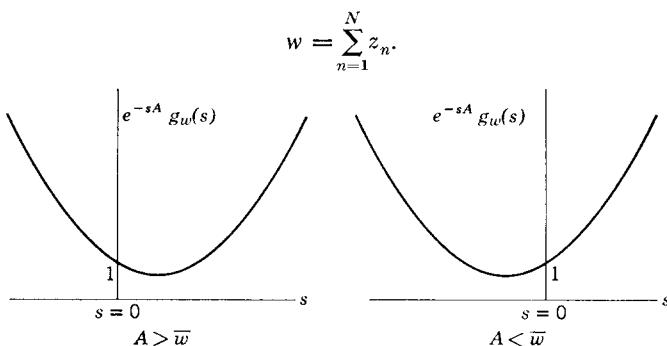


Figure 5.4.1. Sketch of Chernoff bound.

In this case, the moment-generating function of  $w$  can be found from the moment-generating functions of the  $z_n$  by the following steps.

$$g_w(s) = \overline{\exp\left(s \sum_{n=1}^N z_n\right)} = \overline{\prod_{n=1}^N \exp(sz_n)}$$

Since the  $z_n$  are statistically independent, the expectation of the product is the product of the expectations, yielding

$$g_w(s) = \prod_{n=1}^N \overline{\exp(sz_n)} = \prod_{n=1}^N g_n(s) \quad (5.4.14)$$

where  $g_n(s)$  is the moment-generating function of  $z_n$ . Substituting (5.4.14) into (5.4.11) and (5.4.12), we obtain

$$\Pr[w \geq A] \leq e^{-sA} \prod_{n=1}^N g_n(s); \quad s > 0 \quad (5.4.15)$$

$$\Pr[w \leq A] \leq e^{-sA} \prod_{n=1}^N g_n(s); \quad s < 0 \quad (5.4.16)$$

Equation 5.4.15 can now be applied to the probability of error  $P_{e,1}$  for a code with 2 code words (see 5.4.4). In this case,  $A = 0$ ,  $z_n$  is given by (5.4.3), and the probability measure is conditioned on message 1 being transmitted.

$$g_n(s) = \sum_{y_n} P(y_n | x_{1,n}) \exp \left[ s \ln \frac{P(y_n | x_{2,n})}{P(y_n | x_{1,n})} \right] = \sum_{y_n} P(y_n | x_{1,n})^{1-s} P(y_n | x_{2,n})^s \quad (5.4.17)$$

$$P_{e,1} \leq \prod_{n=1}^N g_n(s) = \prod_{n=1}^N \left[ \sum_{y_n} P(y_n | x_{1,n})^{1-s} P(y_n | x_{2,n})^s \right] \quad (5.4.18)$$

This is the same result as derived in (5.3.6).

The next problem is to determine how tight the bounds in (5.4.15) and (5.4.16) are. Roughly, the answer is that if  $N$  is large,  $A$  is far from  $\bar{w}$ , and  $s$  is chosen to minimize the bound, then the appropriate bound [(5.4.15) for  $A > \bar{w}$  and (5.4.16) for  $A < \bar{w}$ ] is also a good estimate of  $\Pr[w \geq A]$ . In order to state this more precisely, it is convenient to change the form of (5.4.15) and (5.4.16) somewhat. The semi-invariant moment-generating function of a random variable is by definition the natural logarithm of its generating function. Thus the semi-invariant moment-generating function of  $w$  is  $\mu_w(s) = \ln g_w(s)$  and that of  $z_n$  is  $\mu_n(s) = \ln g_n(s)$ . These functions are related through (5.4.14) by

$$\mu_w(s) = \ln \prod_{n=1}^N g_n(s) = \sum_{n=1}^N \mu_n(s) \quad (5.4.19)$$

Also, from (5.4.13), the  $s$  that optimizes the bound is given in terms of the derivative of  $\mu_w(s)$  as

$$A = \mu_w'(s) = \sum_{n=1}^N \mu_n'(s) \quad (5.4.20)$$

Substituting (5.4.19) and (5.4.20) into (5.4.15) and (5.4.16), we obtain the parametric bounds

$$\Pr \left[ w \geq \sum_{n=1}^N \mu_n'(s) \right] \leq \exp \left[ \sum_{n=1}^N \mu_n(s) - s\mu_n'(s) \right]; \quad s > 0 \quad (5.4.21)$$

$$\Pr \left[ w \leq \sum_{n=1}^N \mu_n'(s) \right] \leq \exp \left[ \sum_{n=1}^N \mu_n(s) - s\mu_n'(s) \right]; \quad s < 0 \quad (5.4.22)$$

In Appendix 5A, asymptotic expressions are evaluated for these probabilities for the special case where the  $z_n$  are identically distributed. In this case,  $\mu_n(s)$  is independent of  $n$  and we can drop the subscript  $n$ . The result depends upon whether or not the random variables  $z_n$  are lattice variables.\*

\* A lattice random variable is a variable for which all allowable values can be expressed in the form  $\alpha + hi$  where  $\alpha$  and  $h$  are fixed numbers and  $i$  is a variable integer (see Appendix 5A).

The asymptotic expressions are:

$$\Pr[w \geq N\mu'(s)] = \left[ \frac{1}{|s| \sqrt{2\pi N\mu''(s)}} + o\left(\frac{1}{\sqrt{N}}\right) \right] \exp\{N[\mu(s) - s\mu'(s)]\} \quad \text{nonlattice; } s > 0 \quad (5.4.23)$$

$$\Pr[w \geq N\mu'(s)] = \left[ \frac{he^{-|s|\Delta}}{\sqrt{2\pi N\mu''(s)(1 - e^{-|s|h})}} + o\left(\frac{1}{\sqrt{N}}\right) \right] \times \exp\{N[\mu(s) - s\mu'(s)]\} \quad \text{lattice; } s > 0 \quad (5.4.24)$$

| $\epsilon$ | $\alpha$ | $N$  | True Value             | Chernoff Bound        | Asymptotic Expression  | Gaussian Approximation |
|------------|----------|------|------------------------|-----------------------|------------------------|------------------------|
| 0.1        | 0.2      | 20   | 0.1327                 | 0.4114                | 0.1650                 | 0.1318                 |
|            |          | 100  | $1.95 \times 10^{-3}$  | $1.18 \times 10^{-2}$ | $2.12 \times 10^{-3}$  | $7.71 \times 10^{-3}$  |
|            | 0.3      | 20   | $1.12 \times 10^{-2}$  | $4.63 \times 10^{-2}$ | $1.22 \times 10^{-2}$  | $4.54 \times 10^{-3}$  |
|            |          | 100  | $2.50 \times 10^{-8}$  | $2.12 \times 10^{-7}$ | $2.54 \times 10^{-8}$  | $4.02 \times 10^{-11}$ |
| 0.5        | 0.6      | 20   | 0.2517                 | 0.6685                | 0.3649                 | 0.2512                 |
|            |          | 100  | $2.85 \times 10^{-2}$  | 0.1335                | $3.26 \times 10^{-2}$  | $2.87 \times 10^{-2}$  |
|            | 400      |      | $3.68 \times 10^{-5}$  | $3.18 \times 10^{-4}$ | $3.88 \times 10^{-5}$  | $3.91 \times 10^{-3}$  |
|            |          | 1000 | $1.36 \times 10^{-10}$ | $1.80 \times 10^{-9}$ | $1.39 \times 10^{-10}$ | $1.56 \times 10^{-10}$ |
| 0.8        | 20       |      | $5.91 \times 10^{-3}$  | $2.12 \times 10^{-2}$ | $6.29 \times 10^{-3}$  | $6.95 \times 10^{-3}$  |
|            |          | 100  | $5.60 \times 10^{-10}$ | $4.26 \times 10^{-9}$ | $5.66 \times 10^{-10}$ | $1.82 \times 10^{-9}$  |

Illustration of the behavior of different estimates of the tail of a binomial distribution,  $\Pr[w \geq N\alpha]$ , where  $w$  is the sum of  $N$  independent, identically distributed binary random variables taking on the value 1 with probability  $\epsilon$ . The Gaussian approximation is

$$1 - \Phi\left[\frac{N(\alpha - \epsilon) - \frac{1}{2}}{\sqrt{N\epsilon(1 - \epsilon)}}\right] \quad \text{where} \quad \Phi(x) = \int_{-\infty}^x (2\pi)^{-1/2} \exp(-z^2/2) dz$$

Figure 5.4.2.

In these expressions,  $o(1/\sqrt{N})$  is a quantity going to zero faster than  $1/\sqrt{N}$  with increasing  $N$ . For any given  $s$ ,  $o(1/\sqrt{N})$  can be ignored for large enough  $N$ , although as  $s$  approaches 0, the required  $N$  becomes larger and larger. In (5.4.24),  $h$  is the distance between adjacent sample points (see Appendix 5A) and  $\Delta$  is the distance from  $N\mu'(s)$  to the first subsequent sample point of  $w$ . For  $s < 0$ , the same expressions apply to  $\Pr[w \leq N\mu'(s)]$ . The expressions are also valid for continuous valued random variables  $z_n$  if  $\exp(sz_n) < \infty$  for  $s$  in some neighborhood of 0.

The asymptotic expression and the Chernoff bound are compared with the true value and with the Gaussian approximation in Figure 5.4.2 for various binomial distributions. It can be seen that the Chernoff bound and the asymptotic expressions in (5.4.24) are poor approximations for  $A$  close to  $\bar{w}$  (for small  $s$ ), but that for large  $A$ , the Gaussian approximation is poor and the Chernoff bound and asymptotic expression are good.

## 5.5 Randomly Chosen Code Words

We saw, in the last section, that for two code words the probability of decoding error goes to zero exponentially with increasing block length. On the other hand, only one binary source digit is being transmitted per block, so that the error probability is decreased only at the expense of transmission rate. Clearly, the only possibility for reducing error probability without decreasing the transmission rate is to consider larger sets of code words.

We wish to investigate the minimum achievable probability of decoding error as a function of the rate  $R$ , the block length  $N$ , and the channel. An upper bound to this error probability is found which decays exponentially with block length for all rates beneath capacity. This bound is derived by analyzing an ensemble of codes rather than just one good code. This peculiar approach is dictated by the fact that, for interesting values of  $N$  and  $R$ , no way is known to find codes that minimize the probability of decoding error, and even if such codes could be found, the number of possible received sequences would make a straightforward calculation of error probability prohibitive. In the next chapter, we shall discuss a number of explicit block encoding and decoding techniques which are of interest because of their relative ease of instrumentation. The error probability can be estimated for some of these techniques, but for large  $N$  the resulting error probability is far greater than the upper bound on minimum error probability derived here.

In order to define an ensemble of block codes, let  $Q_N(\mathbf{x})$  be an arbitrary probability assignment on the set of channel input sequences of length  $N$ , and let all the code words be chosen independently with these same probabilities. Thus, in this ensemble of codes, the probability of a particular code, say  $\mathbf{x}_1, \dots, \mathbf{x}_M$ , is

$$\prod_{m=1}^M Q_N(\mathbf{x}_m).$$

Each code in the ensemble has its own probability of decoding error, assuming maximum likelihood decoding for the code. We shall upper bound the expectation, over the ensemble, of this error probability. Since at least one code in the ensemble must have an error probability as small as the ensemble average, this will give us an upper bound on the probability of error for the best code (that is, the code with minimum  $P_e$ ).

In order to see why this approach is reasonable, consider the binary symmetric channel again. If we choose two code words of length  $N$  for this channel at random, selecting each digit of each word independently as 0 or 1 with equal probability, then for large  $N$  the two code words will, with high probability, differ in about half the positions in the block. From (5.3.12), if  $x_{1,n} \neq x_{2,n}$ , then  $\min g_n(s) = 2\sqrt{\epsilon(1-\epsilon)}$ . If  $x_{1,n} = x_{2,n}$ , then  $g_n(s) = 1$  for  $0 \leq s \leq 1$ . Thus, for 2 code words differing in  $N/2$  positions, the error probability is bounded by

$$P_{e,m} \leq [2\sqrt{\epsilon(1-\epsilon)}]^{N/2}; \quad m = 1, 2 \quad (5.5.1)$$

This is not the average error probability over the ensemble of codes; it is merely the error probability for a typical code within the ensemble. This distinction will be discussed in greater detail later.

The exponent in (5.5.1) is only half as large as when the two code words differ in every position, but still  $P_e$  approaches 0 exponentially with  $N$ . In return for this loss of exponent, we have a way to consider large sets of code words without fussing about the detailed choice of the words. For large sets of code words, however, each code word cannot differ from each other code word in every position, and it will turn out that the loss of exponent noted above disappears.

Let us now consider an arbitrary discrete channel where  $P_N(y | x)$  gives the probability of receiving a sequence  $y$  when  $x$  is transmitted. We have seen, in (5.3.4), that for a given two code words,  $\mathbf{x}_1$  and  $\mathbf{x}_2$ , the error probability when either word is transmitted is bounded by

$$P_{e,m}(\mathbf{x}_1, \mathbf{x}_2) \leq \sum_y P_N(y | \mathbf{x}_1)^{1-s} P_N(y | \mathbf{x}_2)^s; \quad m = 1, 2; \quad \text{any } s, \quad 0 < s < 1. \quad (5.5.2)$$

If we now consider the ensemble of codes where the code words are selected independently using the probability assignment  $Q_N(x)$ , then the probability of the code with the particular code words  $\mathbf{x}_1$  and  $\mathbf{x}_2$  is  $Q_N(\mathbf{x}_1) \times Q_N(\mathbf{x}_2)$ . Thus the average error probability over the ensemble is given by

$$\bar{P}_{e,m} = \sum_{\mathbf{x}_1} \sum_{\mathbf{x}_2} Q_N(\mathbf{x}_1) Q_N(\mathbf{x}_2) P_{e,y}(x_1, x_2) \quad (5.5.3)$$

$$\leq \sum_{\mathbf{x}_1} \sum_{\mathbf{x}_2} \sum_y Q_N(\mathbf{x}_1) Q_N(\mathbf{x}_2) P_N(y | \mathbf{x}_1)^{1-s} P_N(y | \mathbf{x}_2)^s \quad (5.5.4)$$

$$\bar{P}_{e,m} \leq \sum_y \left[ \sum_{\mathbf{x}_1} Q_N(\mathbf{x}_1) P_N(y | \mathbf{x}_1)^{1-s} \right] \left[ \sum_{\mathbf{x}_2} Q_N(\mathbf{x}_2) P_N(y | \mathbf{x}_2)^s \right] \quad (5.5.5)$$

$$m = 1, 2; \quad \text{any } s, \quad 0 < s < 1.$$

The minimum of (5.5.5) over  $s$  occurs at  $s = \frac{1}{2}$ . To see this,\* observe that  $\mathbf{x}_1$  and  $\mathbf{x}_2$  in (5.5.5) are simply dummy variables of summation. Thus, if we interchange  $s$  and  $1 - s$ , the function will not change and consequently is symmetric around  $s = \frac{1}{2}$ . Also, since we have already seen that the right-hand side of (5.5.2) is convex  $\cup$  in  $s$ , it follows that the right-hand side of (5.5.5) is also convex  $\cup$  in  $s$ . From the symmetry and convexity, the minimum must be at  $s = \frac{1}{2}$  and we have

$$\bar{P}_{e,m} \leq \sum_{\mathbf{y}} \left[ \sum_{\mathbf{x}} Q_N(\mathbf{x}) \sqrt{P_N(\mathbf{y} \mid \mathbf{x})} \right]^2; \quad m = 1, 2 \quad (5.5.6)$$

If the channel is memoryless,

$$P_N(\mathbf{y} \mid \mathbf{x}) = \prod_{n=1}^N P(y_n \mid x_n)$$

In this case, (5.5.6) can be simplified by choosing

$$Q_N(\mathbf{x}) = \prod_{n=1}^N Q(x_n)$$

where  $Q(k)$  is an arbitrary single-letter probability assignment. In other words, we are considering an ensemble in which each letter of each code word is chosen independently with the probability assignment  $Q(k)$ . Then (5.5.6) can be rewritten as

$$\bar{P}_{e,m} \leq \sum_{y_1} \cdots \sum_{y_N} \left[ \sum_{x_1} \cdots \sum_{x_N} \prod_{n=1}^N Q(x_n) \sqrt{P(y_n \mid x_n)} \right]^2 \quad (5.5.7)$$

Summing the  $x_n$  separately over each term in the product [as in (5.3.5)], this becomes

$$\bar{P}_{e,m} \leq \sum_{y_1} \cdots \sum_{y_N} \left[ \prod_{n=1}^N \sum_{x_n} Q(x_n) \sqrt{P(y_n \mid x_n)} \right]^2 \quad (5.5.8)$$

Interchanging the order of the squaring with the product and summing over the  $y_n$  in the same way as the  $x_n$ ,

$$\bar{P}_{e,m} \leq \prod_{n=1}^N \sum_{y_n} \left[ \sum_{x_n} Q(x_n) \sqrt{P(y_n \mid x_n)} \right]^2 \quad (5.5.9)$$

Since the sum over  $x_n$  in (5.5.9) is over the input alphabet  $(0, 1, \dots, K - 1)$ , and since the sum over  $y_n$  is over the output alphabet  $(0, \dots, J - 1)$ , this becomes

$$\bar{P}_{e,m} \leq \left\{ \sum_{j=0}^{J-1} \left( \sum_{k=0}^{K-1} Q(k) \sqrt{P(j \mid k)} \right)^2 \right\}^N; \quad m = 1, 2 \quad (5.5.10)$$

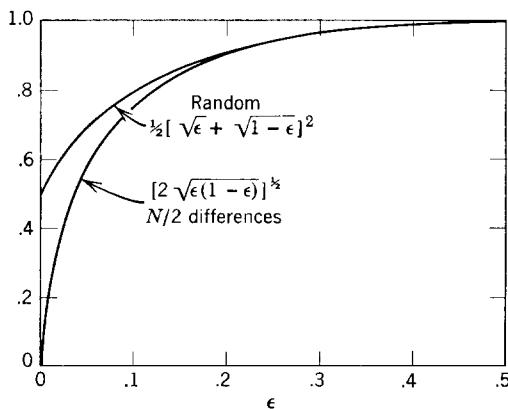
\* We can certainly choose  $s = \frac{1}{2}$  in (5.5.5) whether it minimizes the expression or not. Thus the student may safely ignore this and similar subsequent minimizations over free parameters without worrying about the validity of the result.

This is an upper bound on the average error probability over an ensemble of codes with two code words of length  $N$ . The letters of the code words are chosen independently with probabilities  $Q(k)$ , and the channel is a discrete memoryless channel with transition probabilities  $P(j|k)$ . For the binary symmetric channel, choosing  $Q(0) = Q(1) = \frac{1}{2}$ , (5.5.10) becomes

$$\bar{P}_{e,m} \leq \left\{ \frac{1}{2}(\sqrt{\epsilon} + \sqrt{1-\epsilon})^2 \right\}^N \quad (5.5.11)$$

It is seen that the bound in (5.5.11) is different from that in (5.5.1). Figure 5.5.1 sketches the difference as a function of  $\epsilon$ .

The reason for this difference can be seen most clearly in the limit as  $\epsilon$  approaches 0. The error probability for a typical code with the code words



*Figure 5.5.1. Error probability bound for two code words. Comparison between random selection and  $N/2$  differences.*

differing in half the positions is clearly approaching zero. On the other hand, over the ensemble of codes, the two code words will be chosen the same with probability  $2^{-N}$ ; this is just the limiting bound for  $\bar{P}_{e,m}$  in (5.5.11). In other words, for small  $\epsilon$ , the average error probability,  $\bar{P}_{e,m}$ , is determined not by the typical codes, but by the highly atypical codes for which  $P_{e,m}$  is large.

The above point is simple, but frequently recurring in information theory. If we want to achieve reliable transmission on noisy channels, we must focus our attention on the atypical events that cause errors, rather than on the typical events that do not cause errors. This is also an important but, unfortunately, often neglected principle in constructing models for physical communication channels.

We are now almost ready to find an upper bound on  $\bar{P}_{e,m}$  for more than two code words, but first we shall rederive (5.5.4) in a different way. Notice

that  $\bar{P}_{e,1}$  is an average over  $\mathbf{x}_1$ ,  $\mathbf{x}_2$ , and  $\mathbf{y}$ . When message 1 is encoded into  $\mathbf{x}_1$ ,  $\mathbf{y}$  occurs with probability  $P_N(\mathbf{y} \mid \mathbf{x}_1)$ , and we assume that an error occurs if  $P_N(\mathbf{y} \mid \mathbf{x}_2) \geq P_N(\mathbf{y} \mid \mathbf{x}_1)$ . Thus we can write  $\bar{P}_{e,1}$  as

$$\bar{P}_{e,1} = \sum_{\mathbf{x}_1} Q_N(\mathbf{x}_1) \sum_{\mathbf{y}} P_N(\mathbf{y} \mid \mathbf{x}_1) \Pr[\text{error} \mid m = 1, \mathbf{x}_1, \mathbf{y}] \quad (5.5.12)$$

where  $\Pr[\text{error} \mid m = 1, \mathbf{x}_1, \mathbf{y}]$  is the probability, over the ensemble of choices of  $\mathbf{x}_2$ , that an error will be made given that message 1 enters the encoder,  $\mathbf{x}_1$  is the first code word, and  $\mathbf{y}$  is received.

$$\Pr[\text{error} \mid m = 1, \mathbf{x}_1, \mathbf{y}] = \sum_{\mathbf{x}_2: P_N(\mathbf{y} \mid \mathbf{x}_2) > P_N(\mathbf{y} \mid \mathbf{x}_1)} Q_N(\mathbf{x}_2) \quad (5.5.13)$$

For  $P_N(\mathbf{y} \mid \mathbf{x}_1) > 0$ , we can upper bound (5.5.13) by multiplying each term by  $[P_N(\mathbf{y} \mid \mathbf{x}_2)/P_N(\mathbf{y} \mid \mathbf{x}_1)]^s$  for any  $s > 0$ . Further bounding by summing over all  $\mathbf{x}_2$ , we obtain

$$\Pr[\text{error} \mid m = 1, \mathbf{x}_1, \mathbf{y}] \leq \sum_{\mathbf{x}_2} Q_N(\mathbf{x}_2) \left[ \frac{P_N(\mathbf{y} \mid \mathbf{x}_2)}{P_N(\mathbf{y} \mid \mathbf{x}_1)} \right]^s \quad (5.5.14)$$

This result can alternately be interpreted as a Chernoff bound on

$$\Pr \left\{ \ln \frac{P_N(\mathbf{y} \mid \mathbf{x}_2)}{P_N(\mathbf{y} \mid \mathbf{x}_1)} \geq 0 \right\}$$

using  $Q_N(\mathbf{x}_2)$  as the probability measure.

Substituting (5.5.14) into (5.5.12), we get (5.5.4) again. The reason for going through this alternate derivation is that it can be generalized easily to an arbitrary number of code words.

## 5.6 Many Code Words-The Coding Theorem

**Theorem 5.6.1.** Let  $P_N(\mathbf{y} \mid \mathbf{x})$  be the transition probability assignment for sequences of length  $N \geq 1$  on a discrete channel. Let  $Q_N(\mathbf{x})$  be an arbitrary probability assignment on the input sequences and, for a given number  $M \geq 2$  of code words of block length  $N$ , consider the ensemble of codes in which each word is selected independently with the probability measure  $Q_N(\mathbf{x})$ . Suppose that an arbitrary message  $m$ ,  $1 \leq m \leq M$  enters the encoder and that maximum-likelihood decoding is employed. Then the average probability of decoding error over this ensemble of codes is bounded, for any choice of  $\rho$ ,  $0 \leq \rho \leq 1$ , by

$$\bar{P}_{e,m} \leq (M - 1)^\rho \sum_{\mathbf{y}} \left[ \sum_{\mathbf{x}} Q_N(\mathbf{x}) P_N(\mathbf{y} \mid \mathbf{x})^{1/(1+\rho)} \right]^{1+\rho} \quad (5.6.1)$$

---

We shall need the following simple lemma in the proof of the theorem.

LEMMA. Let  $P(A_1), \dots, P(A_M)$  be the probabilities of a set of events and

$$P\left(\bigcup_m A_m\right)$$

be the probability of their union. For any  $\rho$ ,  $0 < \rho \leq 1$ ,

$$P\left(\bigcup_m A_m\right) \leq \left[ \sum_{m=1}^M P(A_m) \right]^\rho \quad (5.6.2)$$

*Proof of Lemma.*

$$P\left(\bigcup_m A_m\right) \leq \begin{cases} \sum_{m=1}^M P(A_m) \\ 1 \end{cases} \quad (5.6.3)$$

$$(5.6.4)$$

Equation 5.6.3 is the usual union bound on probabilities (see Problem 2.16) and (5.6.4) is an obvious bound on a probability. If  $\sum P(A_m)$  is less than 1, then  $\sum P(A_m)$  is increased by raising it to the power  $\rho$  and (5.6.2) follows from (5.6.3). Conversely, if  $\sum P(A_m) \geq 1$ , then  $[\sum P(A_m)]^\rho \geq 1$ , so that (5.6.2) follows from (5.6.4). |

*Proof of Theorem.*

$$\bar{P}_{e,m} = \sum_{\mathbf{x}_m} \sum_{\mathbf{y}} Q_N(\mathbf{x}_m) P_N(\mathbf{y} \mid \mathbf{x}_m) \Pr[\text{error} \mid m, \mathbf{x}_m, \mathbf{y}] \quad (5.6.5)$$

where  $\Pr[\text{error} \mid m, \mathbf{x}_m, \mathbf{y}]$  is the probability of decoding error conditioned, first, on message  $m$  entering the encoder, second, on the selection of the particular sequence  $\mathbf{x}_m$  as the  $m$ th code word, and third, on the reception of sequence  $\mathbf{y}$ . The summations are respectively over all input sequences and all output sequences of length  $N$  for the channel.

For a given  $m, \mathbf{x}_m, \mathbf{y}$ , define the event  $A_{m'}$  for each  $m' \neq m$ , as the event that code word  $\mathbf{x}_{m'}$  is selected in such a way that  $P_N(\mathbf{y} \mid \mathbf{x}_{m'}) \geq P_N(\mathbf{y} \mid \mathbf{x}_m)$ .

We then have

$$\Pr[\text{error} \mid m, \mathbf{x}_m, \mathbf{y}] \leq P\left(\bigcup_{m' \neq m} A_{m'}\right) \quad (5.6.6)$$

$$\leq \left[ \sum_{m' \neq m} P(A_{m'}) \right]^\rho; \quad \text{any } \rho, \quad 0 < \rho \leq 1 \quad (5.6.7)$$

The inequality (rather than equality) in (5.6.6) follows from the fact that a maximum-likelihood decoder does not necessarily make an error if  $P_N(\mathbf{y} \mid \mathbf{x}_{m'}) = P_N(\mathbf{y} \mid \mathbf{x}_m)$  for some  $m'$ .

From the definition of  $A_{m'}$ , we have

$$\begin{aligned} P(A_{m'}) &= \sum_{\mathbf{x}_{m'}: P_N(\mathbf{y} \mid \mathbf{x}_{m'}) \geq P_N(\mathbf{y} \mid \mathbf{x}_m)} Q_N(\mathbf{x}_{m'}) \\ &\leq \sum_{\mathbf{x}_m} Q_N(\mathbf{x}_m) \frac{P_N(\mathbf{y} \mid \mathbf{x}_{m'})^s}{P_N(\mathbf{y} \mid \mathbf{x}_m)^s}; \quad \text{any } s > 0 \end{aligned} \quad (5.6.8)$$

Since  $\mathbf{x}_{m'}$  is a dummy variable of summation in (5.6.8), the subscript  $m'$  can be dropped and the bound is independent of  $m'$ . Since there are  $M - 1$  choices of  $m' \neq m$ , the substitution of (5.6.8) into (5.6.7) yields

$$\Pr[\text{error} \mid m, \mathbf{x}_m, \mathbf{y}] \leq \left[ (M - 1) \sum_{\mathbf{x}} Q_N(\mathbf{x}) \frac{P_N(\mathbf{y} \mid \mathbf{x})^s}{P_N(\mathbf{y} \mid \mathbf{x}_m)^s} \right]^\rho \quad (5.6.9)$$

Substituting (5.6.9) into (5.6.5), we have

$$\bar{P}_{e,m} \leq (M - 1)^\rho \sum_{\mathbf{y}} \left[ \sum_{\mathbf{x}_m} Q_N(\mathbf{x}_m) P_N(\mathbf{y} \mid \mathbf{x}_m)^{1-s\rho} \right] \left[ \sum_{\mathbf{x}} Q_N(\mathbf{x}) P_N(\mathbf{y} \mid \mathbf{x})^s \right]^\rho \quad (5.6.10)$$

Observe that, if  $P_N(\mathbf{y} \mid \mathbf{x}_m) = 0$ , that term can be omitted from the sum in (5.6.5). Thus we can take  $P_N(\mathbf{y} \mid \mathbf{x}_m)^{1-s\rho}$  to be zero in (5.6.10) if  $P_N(\mathbf{y} \mid \mathbf{x}_m) = 0$ . Finally, substituting\*  $s = 1/(1 + \rho)$  in (5.6.10), and recognizing that  $\mathbf{x}_m$  is a dummy variable of summation, we obtain (5.6.1), valid for  $0 < \rho \leq 1$ . The validity of (5.6.1) for the special case  $\rho = 0$  follows from observing that the right-hand side of (5.6.1) is 1 for  $\rho = 0$ . ¶

This theorem is surprisingly powerful and general. It applies both to memoryless channels and to channels with memory and we shall see, in Chapter 7, that it can be easily generalized to nondiscrete channels. Much of the remainder of this chapter will be devoted to the consequences and interpretations of this theorem. Observe that the mechanical details of the proof are quite simple and do not rely on any previous results. The motivation, however, depends quite heavily on the two code word result. The only new feature of the proof is contained in the lemma. The union bound of (5.6.3) is quite tight for independent events if the resulting bound is small relative to 1, but clearly very loose when the resulting bound is large relative to 1. The lemma provides a way to tighten the union bound in the latter cases at the expense of the former cases. The lemma never provides a tighter bound than the smaller of (5.6.3) and (5.6.4) but, as used in the theorem, it yields a bound on  $\bar{P}_{e,m}$  that is easy to work with.

We now specialize Theorem 5.6.1 to the case of discrete memoryless channels where

$$P_N(\mathbf{y} \mid \mathbf{x}) = \prod_n P(y_n \mid x_n)$$

Let  $Q(k)$ ,  $k = 0, 1, \dots, K - 1$ , be an arbitrary probability assignment on the channel input alphabet and let each letter of each code word be chosen

\* Although the fact is not needed in the proof, this choice of  $s$  minimizes (5.6.10) over  $s$  (see Problem 5.6).

independently with this probability so that

$$\begin{aligned}
 Q_N(\mathbf{x}) &= \prod_{n=1}^N Q(x_n) \\
 \bar{P}_{e,m} &\leq (M-1)^\rho \sum_{y_1} \cdots \sum_{y_N} \left\{ \sum_{x_1} \cdots \sum_{x_N} \prod_{n=1}^N Q(x_n) P(y_n | x_n)^{1/(1+\rho)} \right\}^{1+\rho} \\
 &= (M-1)^\rho \prod_{n=1}^N \sum_{y_n} \left[ \sum_{x_n} Q(x_n) P(y_n | x_n)^{1/(1+\rho)} \right]^{1+\rho} \\
 &= (M-1)^\rho \left\{ \sum_{j=0}^{J-1} \left[ \sum_{k=0}^{K-1} Q(k) P(j | k)^{1/(1+\rho)} \right]^{1+\rho} \right\}^N
 \end{aligned} \tag{5.6.11}$$

In these steps, we have used the same argument as in going from (5.5.6) to (5.5.10).

We now want to rewrite this bound in a way that will explicitly bring out the exponential dependence of the bound on  $N$  for a fixed rate  $R$ . Recall that  $R$  is defined as  $(\ln M)/N$ . Thus  $M = e^{NR}$  and, for fixed rate,  $M$  varies exponentially with  $N$ . Unfortunately, varying  $N$  for fixed  $R$  can lead to non-integer values of  $e^{NR}$ , and we circumvent this detail with the following definition: *For any positive integer  $N$  and any positive number  $R$ , an  $(N,R)$  block code is a code of block-length  $N$  with  $\lceil e^{NR} \rceil$  code words where, by the notation  $\lceil e^{NR} \rceil$ , we mean the smallest integer greater than or equal to  $e^{NR}$ .*

Considering the ensemble of codes above as an ensemble of  $(N,R)$  block codes with  $M-1 < e^{NR} \leq M$ , we have

$$\bar{P}_{e,m} \leq e^{NR\rho} \left\{ \sum_j \left[ \sum_k Q(k) P(j | k)^{1/(1+\rho)} \right]^{1+\rho} \right\}^N \tag{5.6.12}$$

We can summarize our results in the following theorem, which also rearranges (5.6.12).

**Theorem 5.6.2.** Let a discrete memoryless channel have transition probabilities  $P(j | k)$  and, for any positive integer  $N$  and positive number  $R$ , consider the ensemble of  $(N,R)$  block codes in which each letter of each code word is independently selected with the probability assignment  $Q(k)$ . Then, for each message  $m$ ,  $1 \leq m \leq \lceil e^{NR} \rceil$ , and all  $\rho$ ,  $0 \leq \rho \leq 1$ , the ensemble average probability of decoding error using maximum-likelihood decoding satisfies

$$\bar{P}_{e,m} \leq \exp \{ -N[E_o(\rho, \mathbf{Q}) - \rho R] \} \tag{5.6.13}$$

where

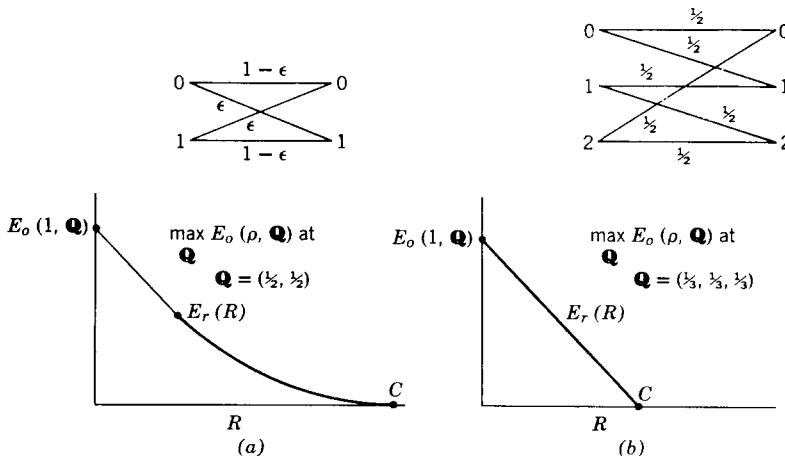
$$E_o(\rho, \mathbf{Q}) = -\ln \sum_{j=0}^{J-1} \left[ \sum_{k=0}^{K-1} Q(k) P(j | k)^{1/(1+\rho)} \right]^{1+\rho} \tag{5.6.14}$$

Since (5.6.13) is valid for each message in the code, we see that the average error probability over the messages, for an arbitrary set of message probabilities,  $\Pr(m)$ , satisfies

$$\bar{P}_e = \sum_{m=1}^M \Pr(m) \bar{P}_{e,m} \leq \exp \{-N[E_o(\rho, \mathbf{Q}) - \rho R]\} \quad (5.6.15)$$

Finally, since  $\rho$  and  $\mathbf{Q}$  are arbitrary in (5.6.13) and (5.6.14), we get the tightest bound by choosing  $\rho$  and  $\mathbf{Q}$  to maximize  $E_o(\rho, \mathbf{Q}) - \rho R$ . This leads us to define the *random coding exponent*  $E_r(R)$  by

$$E_r(R) = \max_{0 \leq \rho \leq 1} \max_{\mathbf{Q}} [E_o(\rho, \mathbf{Q}) - \rho R] \quad (5.6.16)$$



**Figure 5.6.1. Random-coding exponent,  $E_r(R)$ , for two channels. (a) Typical behavior. (b) Special case where  $\partial^2 E_o / \partial \rho^2 = 0$ .**

where the maximization over  $\mathbf{Q}$  is over all probability assignments  $\mathbf{Q} = [Q(0), \dots, Q(K-1)]$ . This yields the following corollary.

**COROLLARY 1.** For an ensemble of codes using the maximizing  $\mathbf{Q}$  in (5.6.16),

$$\bar{P}_{e,m} \leq \exp [-NE_r(R)]; \quad 1 \leq m \leq M \quad (5.6.17)$$

$$\bar{P}_e \leq \exp [-NE_r(R)] \quad (5.6.18)$$

---

The random coding exponent  $E_r(R)$  is sketched for several channels in Figure 5.6.1. We shall see, in the next section, that  $E_r(R) > 0$  for all  $R$ ,  $0 \leq R < C$ , where  $C$  is the channel capacity in natural units. Thus, by choosing codes appropriately, the error probability can be made to approach 0 exponentially with increasing block length for any rate less than capacity.

Since the average error probability over the ensemble of codes satisfies (5.6.18), it is clear that at least one code in the ensemble must have an error probability that small. Although the corollary provides no suggestion about how to find such a code, we would be rather surprised if a randomly selected code had an error probability far greater than the average. In particular, using the Chebyshev inequality, (5.4.6), we have

$$\Pr[P_e \geq \alpha \bar{P}_e] \leq \frac{1}{\alpha}; \quad \text{any } \alpha > 1 \quad (5.6.19)$$

The above result is quite important in practical uses of coding. It says that the difficult problem is not to find good codes of long block length but to find practical encoding and decoding techniques for such codes.

The above arguments give us considerable insight into the behavior of

$$P_e = \sum_m \Pr(m) P_{e,m}$$

for a code chosen at random. Unfortunately, it is entirely possible (and, in fact, highly probable) that, in such a randomly chosen code,  $P_{e,m}$  will be much larger than  $P_e$  for some values of  $m$  and much smaller for others. In many data-transmission systems, the probabilities with which messages are to be used are either unknown or unmeaningful. In such situations, it is usually desirable to have a code for which  $P_{e,m}$  is uniformly small for all  $m$ . In the following corollary, we establish the existence of such codes by starting with a good randomly chosen code and then removing all words for which  $P_{e,m}$  is too large.

**COROLLARY 2.** For any discrete memoryless channel, any positive integer  $N$ , and any positive  $R$ , there exists an  $(N,R)$  block code for which

$$P_{e,m} < 4 \exp[-NE_r(R)]; \quad \text{each } m, \quad 1 \leq m \leq M = \lceil e^{NR} \rceil \quad (5.6.20)$$

*Proof.* Choose a code with  $2M$  code words for which, with equally likely messages,

$$P_e = \frac{1}{2M} \sum_{m=1}^{2M} P_{e,m} \leq \exp \left[ -NE_r \left( \frac{\ln 2M}{N} \right) \right] \quad (5.6.21)$$

Remove  $M$  code words from the code, in particular removing all those for which

$$P_{e,m} \geq 2 \exp \left[ -NE_r \left( \frac{\ln 2M}{N} \right) \right] \quad (5.6.22)$$

There cannot be more than  $M$  words satisfying (5.6.22) because, if there were, (5.6.21) would be violated. Using maximum likelihood decoding, the decoding

subsets associated with the remaining words cannot lose any elements and, thus, for each remaining word

$$P_{e,m} < 2 \exp \left[ -NE_r \left( \frac{\ln 2M}{N} \right) \right]$$

Substituting (5.6.16) into this result,

$$\begin{aligned} P_{e,m} &< 2 \exp \left\{ -N \left[ \max_{0 \leq \rho \leq 1} \max_{\mathbf{Q}} E_o(\rho, \mathbf{Q}) - \rho \frac{\ln M}{N} - \rho \frac{\ln 2}{N} \right] \right\} \\ &\leq 2 \exp \left\{ -N \left[ -\frac{\ln 2}{N} + \max_{0 \leq \rho \leq 1} \max_{\mathbf{Q}} E_o(\rho, \mathbf{Q}) - \rho \frac{\ln M}{N} \right] \right\} \\ &= 4 \exp [-NE_r(R)] \end{aligned}$$

Thus this set of  $M$  code words satisfies (5.6.20). |

### Properties of the Random Coding Exponent, $E_r(R)$

In order to understand the behavior of  $E_r(R)$ , it is first necessary to

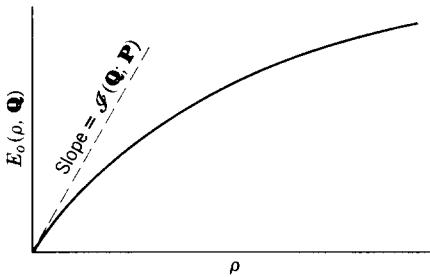


Figure 5.6.2. Sketch of  $E_o(P, Q)$ .

analyze  $E_o(\rho, \mathbf{Q})$  as a function of  $\rho$ . Figure 5.6.2 sketches  $E_o$  as a function of  $\rho$ , and the following theorem asserts that  $E_o$  always has this same general appearance. The average mutual information,

$$\mathcal{I}(\mathbf{Q}; \mathbf{P}) = \sum_{k=0}^{K-1} \sum_{j=0}^{J-1} Q(k) P(j \mid k) \ln \frac{P(j \mid k)}{\sum_i Q(i) P(j \mid i)} \quad (5.6.23)$$

plays a central role in the behavior of  $E_o(\rho, \mathbf{Q})$ . We write it as  $\mathcal{I}(\mathbf{Q}; \mathbf{P})$  to emphasize that we are using it as a mathematical function of  $\mathbf{Q}$  and the channel probability assignment.  $\mathcal{I}(\mathbf{Q}; \mathbf{P})$  has no simple interpretation as average mutual information per digit on the codes of the ensemble.

**Theorem 5.6.3.** Let the input probability assignment  $\mathbf{Q}$  and a discrete memoryless channel be such that  $\mathcal{I}(\mathbf{Q}; \mathbf{P}) > 0$ . Then  $E_o(\rho, \mathbf{Q})$  in (5.6.14) has

the following properties.

$$E_o(\rho, \mathbf{Q}) \geq 0; \quad \rho \geq 0 \quad (5.6.24)$$

$$\mathcal{I}(\mathbf{Q}; \mathbf{P}) \geq \frac{\partial E_o(\rho, \mathbf{Q})}{\partial \rho} > 0; \quad \rho \geq 0 \quad (5.6.25)$$

$$\frac{\partial^2 E_o(\rho, \mathbf{Q})}{\partial \rho^2} \leq 0; \quad \rho \geq 0 \quad (5.6.26)$$

Equality holds in (5.6.24) iff  $\rho = 0$ ; equality holds on the left-hand side of (5.6.25) if  $\rho = 0$ , and equality holds in (5.6.26) iff for all  $j, k$  such that  $Q(k)P(j | k) > 0$ , we have

$$\ln \frac{P(j | k)}{\sum_i Q(i)P(j | i)} = \mathcal{I}(\mathbf{Q}; \mathbf{P}), \quad (5.6.26a)$$

that is, if the mutual information random variable has zero variance.

---

It can be seen by inspection of (5.6.14) that  $E_o(0, \mathbf{Q}) = 0$ , and it can be easily verified by carrying out the differentiation that

$$\left. \frac{\partial E_o(\rho, \mathbf{Q})}{\partial \rho} \right|_{\rho=0} = \mathcal{I}(\mathbf{Q}; \mathbf{P}) \quad (5.6.27)$$

The proof of the rest of the theorem is contained in Appendix 5B.

Using this theorem, it is easy to maximize  $E_o(\rho, \mathbf{Q}) - \rho R$  over  $\rho$  for a given  $\mathbf{Q}$ . Define

$$E_r(R, \mathbf{Q}) = \max_{0 \leq \rho \leq 1} [E_o(\rho, \mathbf{Q}) - \rho R] \quad (5.6.28)$$

The equation for a stationary point of  $E_o(\rho, \mathbf{Q}) - \rho R$  with respect to  $\rho$  is

$$\left. \frac{\partial E_o(\rho, \mathbf{Q})}{\partial \rho} \right|_{\rho} - R = 0 \quad (5.6.29)$$

Since  $\partial^2 E_o(\rho, \mathbf{Q}) / \partial \rho^2 \leq 0$ , any solution of (5.6.29) in the range  $0 \leq \rho \leq 1$  maximizes (5.6.28). Furthermore, since  $\partial E_o / \partial \rho$  is continuous and decreasing with respect to  $\rho$ , a solution to (5.6.29) with  $0 \leq \rho \leq 1$  exists if

$$\left. \frac{\partial E_o(\rho, \mathbf{Q})}{\partial \rho} \right|_{\rho=1} \leq R \leq \left. \frac{\partial E_o(\rho, \mathbf{Q})}{\partial \rho} \right|_{\rho=0} = \mathcal{I}(\mathbf{Q}; \mathbf{P}) \quad (5.6.30)$$

The point  $\partial E_o / \partial \rho \Big|_{\rho=1}$  is called the critical rate,  $R_{cr}$ , for the given  $\mathbf{Q}$ .

For  $R$  in the above range, it is most convenient to use (5.6.29) to relate  $R$  and  $E_r(R, \mathbf{Q})$  parametrically in terms of  $\rho$ ,

$$R = \partial E_o(\rho, \mathbf{Q}) / \partial \rho; \quad 0 \leq \rho \leq 1 \quad (5.6.31)$$

$$E_r(R, \mathbf{Q}) = E_o(\rho, \mathbf{Q}) - \rho \partial E_o(\rho, \mathbf{Q}) / \partial \rho$$

Differentiating each equation in (5.6.31), we obtain  $\partial R / \partial \rho = \partial^2 E_o / \partial \rho^2$

and  $\partial E_r / \partial \rho = -\rho \partial^2 E_o / \partial \rho^2$ . Thus, as  $\rho$  goes from 0 to 1,  $R$  decreases monotonically from  $\mathcal{I}(\mathbf{Q}; \mathbf{P})$  to  $\partial E_o / \partial \rho |_{\rho=1}$  and  $E_r(R, \mathbf{Q})$  increases monotonically from 0 to  $E_o(1, \mathbf{Q}) - \partial E_o / \partial \rho |_{\rho=1}$ . Taking the ratio of the derivatives, we obtain

$$\frac{\partial E_r(R, \mathbf{Q})}{\partial R} = -\rho \quad (5.6.32)$$

Thus the parameter  $\rho$  is interpreted as the magnitude of the slope of the  $E_r(R, \mathbf{Q})$  versus  $R$  curve.

For  $R < \partial E_o / \partial \rho |_{\rho=1}$ ,  $E_o(\rho, \mathbf{Q}) - \rho R$  is maximized (over  $0 \leq \rho \leq 1$ ) by  $\rho = 1$ , yielding

$$E_r(R, \mathbf{Q}) = E_o(1, \mathbf{Q}) - R \quad (5.6.33)$$

Finally, in the uninteresting case where  $R > \mathcal{I}(\mathbf{Q}; \mathbf{P})$ ,  $E_o(\rho, \mathbf{Q}) - \rho R$  is maximized by  $\rho = 0$ , yielding  $E_r(R, \mathbf{Q}) = 0$ .

In summary, for  $R$  in the range given by (5.6.30),  $E_r(R, \mathbf{Q})$  and  $R$  are related by (5.6.31). For smaller values of  $R$ ,  $E_r(R, \mathbf{Q})$  and  $R$  are related by the linear relation (5.6.33), and for larger values,  $E_r(R, \mathbf{Q}) = 0$ . As a function of  $R$ ,  $E_r(R, \mathbf{Q})$  is strictly decreasing and positive for all  $R < \mathcal{I}(\mathbf{Q}; \mathbf{P})$ .

Now consider the special case where  $\partial^2 E_o / \partial \rho^2 = 0$ . From (5.6.26a) in Theorem 5.6.3, we see that this relation must be satisfied for all  $\rho \geq 0$  if it is satisfied for any  $\rho \geq 0$ . In this case,  $\partial E_o / \partial \rho$  is a constant and the range over which (5.6.30) is satisfied is of zero extent (see Figure 5.6.1b). This special case is rather pathological and occurs only for noiseless channels [for which  $H(X | Y) = 0$ ] and for some rather peculiar channels such as that in Figure 5.6.1b.

For the usual case, in which  $\partial^2 E_o / \partial \rho^2 < 0$ , the parametric equations of (5.6.31) apply over a nonzero range of rates. From (5.6.32) and (5.6.31), we have  $\partial^2 E_r / \partial R^2 = -[\partial^2 E_o / \partial \rho^2]^{-1} > 0$  and, thus,  $E_r(R, \mathbf{Q})$  is strictly convex  $\cup$  in  $R$  over this range of  $R$ . Since  $\partial^2 E_r / \partial R^2 = 0$  outside of this range, we see that  $E_r(R, \mathbf{Q})$  is convex  $\cup$  in  $R$  for all  $R \geq 0$ .

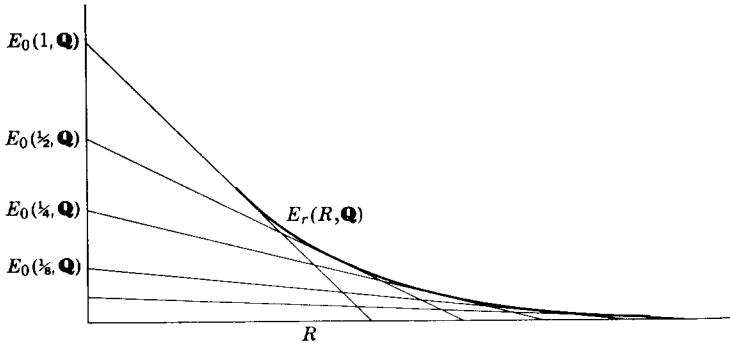
The random coding exponent,  $E_r(R)$  can now be related to  $E_r(R, \mathbf{Q})$  by

$$E_r(R) = \max_{\mathbf{Q}} E_r(R, \mathbf{Q}) \quad (5.6.34)$$

This is a maximum over a set of functions that are convex  $\cup$  and decreasing in  $R$ . It is easy to see (Problem 4.12) that the maximizing function is also convex  $\cup$  and decreasing in  $R$ . Also, for the  $\mathbf{Q}$  that yields capacity on the channel,  $E_r(R, \mathbf{Q})$  is positive for  $R < \mathcal{I}(\mathbf{Q}; \mathbf{P}) = C$  and, therefore,  $E_r(R)$  is positive for  $R < C$ . This proves the following fundamental theorem.

**Theorem 5.6.4 (Noisy-Channel Coding Theorem).** For any discrete memoryless channel, the random coding exponent  $E_r(R)$  [see (5.6.16) and (5.6.18)] is a convex  $\cup$ , decreasing, positive function of  $R$  for  $0 \leq R < C$ .

---



**Figure 5.6.3.**  $E_r(R, Q)$  as upper bound of linear functions  $E_o(\rho, Q) - \rho R$  for fixed values of  $\rho$ .

An interesting graphical interpretation of the maximization of  $E_o(\rho, Q) - \rho R$  over  $\rho$  and  $Q$  can be obtained by observing that, for fixed  $\rho$  and  $Q$ ,  $E_o(\rho, Q) - \rho R$  is a linear function of  $R$  with slope  $-\rho$ . Thus  $E_r(R, Q)$ , as shown in Figure 5.6.3, is the lowest upper bound to the family of straight lines generated by different values of  $\rho$ ,  $0 \leq \rho \leq 1$ . From this construction, we see that  $E_o(\rho, Q)$  is the zero rate intercept of the tangent to  $E_r(R, Q)$  at slope  $-\rho$ . The convexity  $\cup$  of  $E_r(R, Q)$  also follows immediately from this construction.

In order to maximize  $E_o(\rho, Q) - \rho R$  analytically over both  $\rho$  and  $Q$  it is more convenient to maximize over  $Q$  first.

$$E_r(R) = \max_{0 \leq \rho \leq 1} \left[ -\rho R + \max_Q E_o(\rho, Q) \right] \quad (5.6.35)$$

The function  $E_o(\rho, Q)$  is not a convex  $\cup$  function of  $Q$  but, fortunately, it turns out to be minus the logarithm of a convex  $\cap$  function. Define

$$F(\rho, Q) = \exp [-E_o(\rho, Q)] = \sum_{j=0}^{J-1} \left( \sum_{k=0}^{K-1} Q(k) P(j \mid k)^{1/(1+\rho)} \right)^{1+\rho} \quad (5.6.36)$$

The  $Q$  that minimizes  $F(\rho, Q)$  will maximize  $E_o(\rho, Q)$ .

**Theorem 5.6.5.** For any  $\rho \geq 0$ ,  $F(\rho, Q)$  as given by (5.6.36) is a convex  $\cup$  function of  $Q$  over the region where  $Q$  is a probability vector. Necessary and sufficient conditions on the probability vector  $Q$  that minimizes  $F(\rho, Q)$  [and maximizes  $E_o(\rho, Q)$ ] are

$$\sum_j P(j \mid k)^{1/(1+\rho)} \alpha_j(Q)^\rho \geq \sum_j \alpha_j(Q)^{1+\rho}; \quad \text{all } k \quad (5.6.37)$$

with equality for all  $k$  such that  $Q(k) > 0$ . The function  $\alpha_j(\mathbf{Q})$  is given by

$$\alpha_j(\mathbf{Q}) = \sum_k Q(k)P(j \mid k)^{1/(1+\rho)} \quad (5.6.38)$$

*Proof.* For  $\rho \geq 0$ ,  $\alpha_j(\mathbf{Q})^{1+\rho}$  is a convex  $\cup$  function of  $\alpha_j(\mathbf{Q}) \geq 0$  since its second derivative is nonnegative. Since  $\alpha_j(\mathbf{Q})$  is a linear function of  $\mathbf{Q}$ , it follows from the definition of convexity that  $\alpha_j(\mathbf{Q})^{1+\rho}$  is a convex  $\cup$  function of  $\mathbf{Q}$ . From this,

$$F(\rho, \mathbf{Q}) = \sum_j \alpha_j(\mathbf{Q})^{1+\rho}$$

is a convex  $\cup$  function of  $\mathbf{Q}$ .

Using Theorem 4.4.1, necessary and sufficient conditions on the probability vector  $\mathbf{Q}$  to minimize  $F(\rho, \mathbf{Q})$  are

$$\frac{\partial F(\rho, \mathbf{Q})}{\partial Q(k)} \geq A; \quad \text{all } k, \text{ equality if } Q(k) > 0 \quad (5.6.39)$$

Evaluating  $\partial F / \partial Q(k)$  and dividing by  $(1 + \rho)$ , we get (5.6.37). The constant on the right-hand side of (5.6.37) is evaluated by multiplying each equation by  $Q(k)$  and summing over  $k$ . |

The problem of actually solving (5.6.37) and (5.6.38) to find the maximum of  $E_o(\rho, \mathbf{Q})$  is almost identical to the problem of finding capacity. For some channels the maximizing  $\mathbf{Q}$  can be guessed and verified by (5.6.37). For any symmetric channel (see definition in Section 4.5), it is easily verified that  $E_o(\rho, \mathbf{Q})$  is maximized by making the  $Q(k)$  all the same. Next, if the number of inputs and outputs are equal, it is sometimes possible to solve (5.6.37) as a set of linear equations in  $\alpha_j(\mathbf{Q})^\rho$  and then to solve (5.6.38) for  $Q(k)$ . Finally, using the convexity of  $F(\rho, \mathbf{Q})$ , it is easy to maximize  $E_o(\rho, \mathbf{Q})$  with a computer.

As in finding capacity, the solution for  $\alpha_j(\mathbf{Q})$  in (5.6.37) and (5.6.38) is unique, but the solution for  $Q(k)$  need not be unique. If the input alphabet size  $K$  is larger than the output alphabet size  $J$ , it is always possible to maximize  $E_o(\rho, \mathbf{Q})$  with only  $J$  of the  $Q(k)$  nonzero. The only significant difference between maximizing  $I(X; Y)$  and  $E_o(\rho, \mathbf{Q})$  is that the output probabilities for capacity are always strictly positive whereas some of the  $\alpha_j(\mathbf{Q})$  can be zero.

Given the  $\mathbf{Q}$  that maximizes  $E_o(\rho, \mathbf{Q})$  for each  $\rho$ , we can use the graphical technique of Figure 5.6.3 to find the curve  $E_r(R)$ . Alternatively, we can use the equations (5.6.31) and (5.6.33), using for each  $\rho$  the  $\mathbf{Q}$  that maximizes  $E_o(\rho, \mathbf{Q})$ . To see that these equations generate all points on the  $E_r(R)$  curve, observe that for each  $R$ , there is some  $\rho$  and  $\mathbf{Q}$  such that  $E_r(R) = E_o(\rho, \mathbf{Q}) - \rho R$ . For that  $\mathbf{Q}$ ,  $E_r(R) = E_r(R, \mathbf{Q})$ . But since the parametric equations yield  $E_r(R, \mathbf{Q})$  for the given  $\rho$  and  $\mathbf{Q}$ , they also yield  $E_r(R)$ . We shall see, in Example 2, however, that it is possible for these equations to generate some additional points strictly below the  $E_r(R)$  curve.

**Example 1.** For the binary symmetric channel of Figure 5.3.1a,  $E_o(\rho, \mathbf{Q})$  is maximized over  $\mathbf{Q}$  by  $Q(0) = Q(1) = \frac{1}{2}$ . For this  $\mathbf{Q}$ , we have

$$E_o(\rho, \mathbf{Q}) = \rho \ln 2 - (1 + \rho) \ln [\epsilon^{1/(1+\rho)} + (1 - \epsilon)^{1/(1+\rho)}] \quad (5.6.40)$$

The parametric equations (5.6.31) can be manipulated into the form

$$\begin{aligned} R &= \ln 2 - \mathcal{H}(\delta) \\ E_r(R) &= T_\epsilon(\delta) - \mathcal{H}(\delta) \end{aligned} \quad (5.6.41)$$

where the parameter  $\delta$  is related to the parameter  $\rho$  in (5.6.31) by

$$\delta = \frac{\epsilon^{1/(1+\rho)}}{\epsilon^{1/(1+\rho)} + (1 - \epsilon)^{1/(1+\rho)}} \quad (5.6.42)$$

and  $\mathcal{H}(\delta)$  and  $T_\epsilon(\delta)$  are given by

$$\mathcal{H}(\delta) = -\delta \ln \delta - (1 - \delta) \ln (1 - \delta) \quad (5.6.43)$$

$$T_\epsilon(\delta) = -\delta \ln \epsilon - (1 - \delta) \ln (1 - \epsilon) \quad (5.6.44)$$

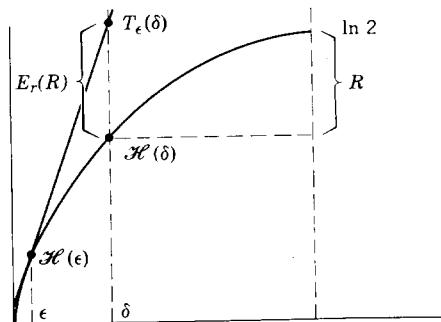


Figure 5.6.4. The random-coding exponent for a binary symmetric channel.

These equations are only valid for  $\delta$  in the range

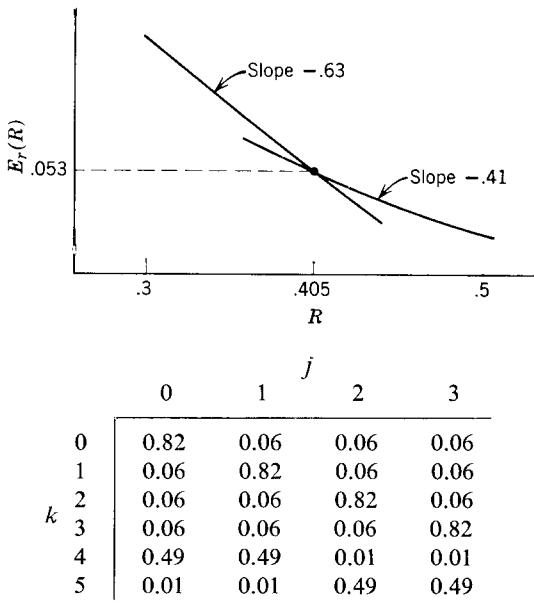
$$\epsilon \leq \delta \leq \sqrt{\epsilon}/(\sqrt{\epsilon} + \sqrt{1 - \epsilon})$$

For  $R < \ln 2 - \mathcal{H}[\sqrt{\epsilon}/(\sqrt{\epsilon} + \sqrt{1 - \epsilon})]$ , we can combine (5.6.33) with (5.6.40) to obtain

$$E_r(R) = \ln 2 - 2 \ln (\sqrt{\epsilon} + \sqrt{1 - \epsilon}) - R \quad (5.6.45)$$

Equation 5.6.41 can be interpreted graphically as in Figure 5.6.4. It can be seen that  $T_\epsilon(\delta)$ , as a function of  $\delta$ , is the equation of the tangent at  $\epsilon$  to the curve  $\mathcal{H}(\delta)$ .

The most significant point about this example is that, even for such a simple channel, there is no simple way to express  $E_r(R)$  except in parametric form.



*Figure 5.6.5. A slope discontinuity in  $E_r(R)$ .*

**Example 2.** Consider the channel with the transition probability matrix in Figure 5.6.5. It can be seen that if only the first four inputs are used, the channel is symmetric and, if only the last two inputs are used, the channel reduces essentially to a binary symmetric channel. We might guess that, at high rates, we should use only the first four inputs and that, at low rates, we should use only the last two somewhat less noisy inputs. Trying this hypothesis in (5.6.37), we find that for  $\rho \leq 0.51$ ,  $E_o(\rho, \mathbf{Q})$  is maximized by  $Q(0) = Q(1) = Q(2) = Q(3) = \frac{1}{4}$  and, for  $\rho > 0.51$ ,  $E_o(\rho, \mathbf{Q})$  is maximized by  $Q(4) = Q(5) = \frac{1}{2}$ .

Substituting these choices for  $\mathbf{Q}$  into (5.6.31), we get the  $E_r(R)$  curve partially shown in Figure 5.6.5. There is a discontinuity in slope from  $\rho = 0.41$  to  $\rho = 0.63$  and, for  $\rho$  in this range, (5.6.31) yields points as shown strictly beneath  $E_r(R)$ .

**Example 3 (Very Noisy Channels).** Let us consider a channel that is very noisy in the sense that the probability of receiving a given output is almost independent of the input. We shall derive an approximation to  $E_r(R)$  for such channels which depends only on the capacity. Let  $\omega_j, j = 0, \dots, J-1$  be a set of probabilities defined on the channel outputs and define  $\epsilon_{jk}$  by

$$P(j | k) = \omega_j(1 + \epsilon_{jk}) \quad (5.6.46)$$

We assume that  $|\epsilon_{jk}| \ll 1$  for all  $j, k$ , so that the channel is very noisy in the above sense. If (5.6.46) is summed over  $j$ , we obtain

$$\sum_j \omega_j \epsilon_{jk} = 0 \quad \text{for all } k \quad (5.6.47)$$

We now compute  $E_o(\rho, \mathbf{Q})$  for the channel by expanding  $E_o$  as a power series in the  $\epsilon_{jk}$  and dropping all terms of higher than second order.

$$E_o(\rho, \mathbf{Q}) = -\ln \sum_j \left[ \sum_k Q(k) \omega_j^{1/(1+\rho)} (1 + \epsilon_{jk})^{1/(1+\rho)} \right]^{1+\rho}$$

Removing  $\omega_j$  from the inner summation and expanding  $(1 + \epsilon_{jk})^{1/(1+\rho)}$ , we get

$$\begin{aligned} E_o(\rho, \mathbf{Q}) &\approx -\ln \sum_j \omega_j \left\{ \sum_k Q(k) \left[ 1 + \frac{\epsilon_{jk}}{1 + \rho} - \frac{\rho \epsilon_{jk}^2}{2(1 + \rho)^2} \right] \right\}^{1+\rho} \\ &\approx -\ln \sum_j \omega_j \left\{ 1 + \sum_k (1 + \rho) Q(k) \left[ \frac{\epsilon_{jk}}{1 + \rho} - \frac{\rho \epsilon_{jk}^2}{2(1 + \rho)^2} \right] \right. \\ &\quad \left. + \frac{\rho(1 + \rho)}{2} \left[ \sum_k Q(k) \frac{\epsilon_{jk}}{1 + \rho} \right]^2 \right\} \end{aligned}$$

Using (5.6.47), this becomes

$$\begin{aligned} E_o(\rho, \mathbf{Q}) &\approx -\ln \left\{ 1 - \frac{\rho}{2(1 + \rho)} \sum_j \omega_j \left[ \sum_k Q(k) \epsilon_{jk}^2 - \left( \sum_k Q(k) \epsilon_{jk} \right)^2 \right] \right\} \\ &\approx \frac{\rho}{2(1 + \rho)} \sum_j \omega_j \left[ \sum_k Q(k) \epsilon_{jk}^2 - \left( \sum_k Q(k) \epsilon_{jk} \right)^2 \right] \\ &= \frac{\rho}{1 + \rho} f(\mathbf{Q}) \end{aligned} \quad (5.6.48)$$

where

$$f(\mathbf{Q}) = \frac{1}{2} \sum_j \omega_j \left[ \sum_k Q(k) \epsilon_{jk}^2 - \left( \sum_k Q(k) \epsilon_{jk} \right)^2 \right] \quad (5.6.49)$$

The parametric equations (5.6.31) become

$$R \approx \frac{f(\mathbf{Q})}{(1 + \rho)^2} \quad (5.6.50a)$$

$$E_r(R, \mathbf{Q}) \approx \frac{\rho^2 f(\mathbf{Q})}{(1 + \rho)^2} \quad (5.6.50b)$$

The average mutual information using the input probabilities  $\mathbf{Q}$  is given by (5.6.50a) with  $\rho = 0$ . Thus channel capacity is given by

$$C \approx \max_{\mathbf{Q}} f(\mathbf{Q}) \quad (5.6.51)$$

Finally, solving (5.6.50) for  $\rho$  and using (5.6.51), we obtain

$$E_r(R) \approx (\sqrt{C} - \sqrt{R})^2; \quad \frac{C}{4} \leq R \leq C \quad (5.6.52)$$

For  $R < C/4$ , we can combine (5.6.33), (5.6.48), and (5.6.51) to obtain

$$E_r(R) \approx \frac{C}{2} - R; \quad 0 \leq R < \frac{C}{4} \quad (5.6.53)$$

This is sketched in Figure 5.6.6.

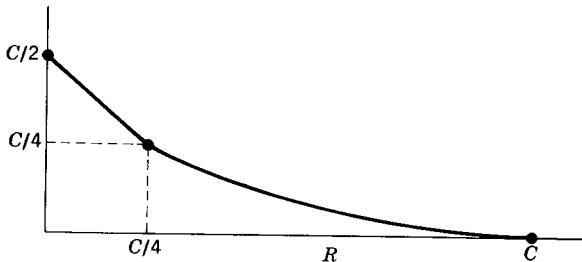


Figure 5.6.6.  $E_r(R)$  for very noisy channels.

**Example 4 (Parallel Channels).** Let  $P^*(j | k)$  and  $P^{**}(l | i)$  be the transition probabilities of two discrete memoryless channels. We shall consider using these channels in parallel; that is, each unit of time, the transmitter sends a symbol  $k$  over the first channel and a symbol  $i$  over the second channel. The channels will be assumed to be independent; that is, the probability of receiving symbol  $j$  on the first channel and  $l$  on the second channel, given that the pair  $(k, i)$  is sent, is  $P^*(j | k)P^{**}(l | i)$ .

These parallel channels can be considered as a single channel with inputs consisting of  $(k, i)$  pairs and outputs consisting of  $(j, l)$  pairs. We can apply the coding theorem to this combination channel, using sequences of input pairs as code words. Letting  $Q(k, i)$  be a probability assignment on the input pairs, we have

$$E_0(\rho, \mathbf{Q}) = -\ln \sum_{j, l} \left( \sum_{k, i} Q(k, i) [P^*(j | k)P^{**}(l | i)]^{1/(1+\rho)} \right)^{1+\rho} \quad (5.6.54)$$

If we restrict  $Q(k, i)$  to be

$$Q(k, i) = Q^*(k)Q^{**}(i) \quad (5.6.55)$$

where  $Q^*$  and  $Q^{**}$  are arbitrary input probability assignments on the separate channels, then  $E_0(\rho, \mathbf{Q})$  simplifies as follows.

$$\begin{aligned} E_0(\rho, \mathbf{Q}) &= -\ln \sum_{j, l} \left( \sum_k Q^*(k) P^*(j | k)^{1/(1+\rho)} \right)^{1+\rho} \left( \sum_i Q^{**}(i) P^{**}(l | i)^{1/(1+\rho)} \right)^{1+\rho} \\ &= E_0^*(\rho, \mathbf{Q}^*) + E_0^{**}(\rho, \mathbf{Q}^{**}) \end{aligned} \quad (5.6.56)$$

where

$$E_0^*(\rho, \mathbf{Q}^*) = -\ln \sum_j \left[ \sum_k Q^*(k) P^*(j | k)^{1/(1+\rho)} \right]^{1+\rho} \quad (5.6.57)$$

$$E_0^{**}(\rho, \mathbf{Q}^{**}) = -\ln \sum_l \left[ \sum_i Q^{**}(i) P^{**}(l | i)^{1/(1+\rho)} \right]^{1+\rho} \quad (5.6.58)$$

Thus  $E_o(\rho, \mathbf{Q})$  simplifies to the sum of the  $E_o$  functions for the individual channels.

If we choose  $Q^*(k)$  to maximize  $E_o^*(\rho, \mathbf{Q}^*)$  for a given  $\rho$  and  $Q^{**}(i)$  to maximize  $E_o^{**}(\rho, \mathbf{Q}^{**})$ , then it follows easily from (5.6.37) that  $E_o(\rho, \mathbf{Q})$  is maximized by  $Q(k, i) = Q^*(k)Q^{**}(i)$  and thus

$$\max_{\mathbf{Q}} E_o(\rho, \mathbf{Q}) = \max_{\mathbf{Q}^*} E_o^*(\rho, \mathbf{Q}^*) + \max_{\mathbf{Q}^{**}} E_o^{**}(\rho, \mathbf{Q}^{**}) \quad (5.6.59)$$

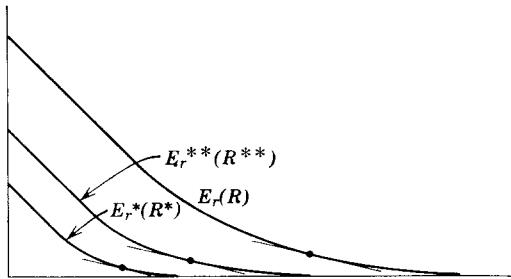


Figure 5.6.7.  $E_r(R)$  for parallel channels.

This result has an interesting geometric interpretation. Let  $E_r(\rho)$  and  $R(\rho)$  be the exponent and rate for the parallel combination as parametrically related by (5.6.31) using the optimizing  $\mathbf{Q}$ . Let  $E_r^*(\rho)$ ,  $R^*(\rho)$ ,  $E_r^{**}(\rho)$ , and  $R^{**}(\rho)$  be the analogous quantities for the individual channels, Then

$$E_r(\rho) = E_r^*(\rho) + E_r^{**}(\rho) \quad (5.6.60)$$

$$R(\rho) = R^*(\rho) + R^{**}(\rho) \quad (5.6.61)$$

Thus the parallel combination is formed by vector addition of points of the same slope from the individual  $E_r(\rho)$ ,  $R(\rho)$  curves (see Figure 5.6.7).

## 5.7 Error Probability for an Expurgated Ensemble of Codes

We saw, in Section 5.5 (Figure 5.5.1), that the average probability of error for an ensemble of two randomly chosen code words is quite different from the probability of error for a typical selection of two code words. The reason is that the poor codes in the ensemble, although quite improbable, have such

high error probability that they dominate the average error probability. This same problem turns out to have an adverse effect on the random-coding exponent at low rates. This is most clearly seen for the binary symmetric channel for which, from (5.6.45), as  $\epsilon \rightarrow 0$ , the upper bound on error probability approaches  $M(1/2)^N$ . This approximates the probability that, for a given transmitted code word, some other code word will be identically chosen.

In this section, we shall improve on the random-coding bound at low rates by expurgating poor code words from the codes in an ensemble. Our approach will be similar to that in the corollary establishing (5.6.20). We start with an ensemble of codes each with  $M' = 2M - 1$  code words. We then show that, for at least one code in the ensemble, there are at least  $M$  words for which  $P_{e,m}$  satisfies a given bound. This bound will be derived in terms of an arbitrary parameter,  $s > 0$ , which can be optimized later. Over the ensemble of codes,  $P_{e,m}^s$  is a random variable for each  $m$ ,  $1 \leq m \leq M'$ . Applying the Chebyshev inequality, (5.4.6), to this random variable, we obtain

$$\Pr[P_{e,m}^s \geq 2\overline{P_{e,m}^s}] \leq \frac{1}{2} \quad (5.7.1)$$

LEMMA. For any  $s > 0$ , there is at least one code in the ensemble of codes with  $M' = 2M - 1$  code words for which at least  $M$  of the code words satisfy

$$P_{e,m} < 2^{1/s} \overline{P_{e,m}^s}^{1/s} \quad (5.7.2)$$


---

*Proof.* For each  $m$ , let  $\varphi_m$  be a random variable over the ensemble of codes. Let  $\varphi_m = 1$  for the codes in which (5.7.2) is satisfied and let  $\varphi_m = 0$  otherwise. From 5.7.1, the probability of (5.7.2) being satisfied is at least  $\frac{1}{2}$  and thus  $\bar{\varphi}_m \geq \frac{1}{2}$ . The number of code words in a randomly selected code that satisfy (5.7.2) is the random variable

$$\sum_{m=1}^{M'} \varphi_m$$

The expectation of the number of words that satisfy (5.7.2) is thus

$$\sum_{m=1}^{M'} \bar{\varphi}_m \geq \frac{M'}{2}$$

It follows that there is at least one code for which  $\sum \varphi_m \geq M'/2$  and, for such a code,  $\sum \varphi_m \geq M$ . |

If all but  $M$  words satisfying (5.7.2) are expurgated from a code satisfying the lemma, the decoding regions of the remaining code words cannot be decreased, and thus we have demonstrated a code with  $M$  code words each

satisfying (5.7.2). We next must find a convenient upper bound on  $\overline{P_{e,m}^s}$ . For a particular code with code words  $\mathbf{x}_1, \dots, \mathbf{x}_{M'}$ , we have

$$P_{e,m} \leq \sum_{m' \neq m} \sum_{\mathbf{y}} \sqrt{P_N(\mathbf{y} \mid \mathbf{x}_m) P_N(\mathbf{y} \mid \mathbf{x}_{m'})} \quad (5.7.3)$$

To justify (5.7.3), observe from (5.3.4) that, for the given code

$$\sum_{\mathbf{y}} \sqrt{P_N(\mathbf{y} \mid \mathbf{x}_m) P_N(\mathbf{y} \mid \mathbf{x}_{m'})}$$

is an upper bound to the probability that  $P_N(\mathbf{y} \mid \mathbf{x}_{m'}) \geq P_N(\mathbf{y} \mid \mathbf{x}_m)$  given that  $\mathbf{x}_m$  is transmitted. Since  $P_{e,m}$  is the probability of the union over  $m' \neq m$  of these events,  $P_{e,m}$  can be bounded as in (5.7.3).

Now let  $s$  be restricted to  $0 < s \leq 1$ . Using the standard inequality,  $(\sum a_i)^s \leq \sum a_i^s$  (see Problem 4.15f), we have

$$P_{e,m}^s \leq \sum_{m' \neq m} \left[ \sum_{\mathbf{y}} \sqrt{P_N(\mathbf{y} \mid \mathbf{x}_m) P_N(\mathbf{y} \mid \mathbf{x}_{m'})} \right]^s; \quad 0 < s \leq 1 \quad (5.7.4)$$

Consider an ensemble of codes in which each code word is selected independently with the probability assignment  $Q_N(\mathbf{x})$ .

$$\overline{P_{e,m}^s} \leq \sum_{m' \neq m} \left\{ \sum_{\mathbf{x}_m} \sum_{\mathbf{x}_{m'}} Q_N(\mathbf{x}_m) Q_N(\mathbf{x}_{m'}) \left[ \sum_{\mathbf{y}} \sqrt{P_N(\mathbf{y} \mid \mathbf{x}_m) P_N(\mathbf{y} \mid \mathbf{x}_{m'})} \right]^s \right\} \quad (5.7.5)$$

Since  $\mathbf{x}_{m'}$  is a dummy variable of summation in (5.7.5), the term in braces is independent of  $m'$  and we have  $M' - 1 = 2(M - 1)$  identical terms. Substituting (5.7.5) with this modification into (5.7.2), we have

$$P_{e,m} < 2^{1/s} \left\{ 2(M - 1) \sum_{\mathbf{x}} \sum_{\mathbf{x}'} Q_N(\mathbf{x}) Q_N(\mathbf{x}') \left[ \sum_{\mathbf{y}} \sqrt{P_N(\mathbf{y} \mid \mathbf{x}) P_N(\mathbf{y} \mid \mathbf{x}')} \right]^s \right\}^{1/s} \quad (5.7.6)$$

This bound has a form quite similar to that of Theorem 5.6.1. This similarity can be brought out by defining  $\rho = 1/s$ . Since  $s$  is an arbitrary parameter in (5.7.6),  $0 < s \leq 1$ ,  $\rho$  is an arbitrary parameter,  $\rho \geq 1$ .

$$P_{e,m} < [4(M - 1)]^\rho \left\{ \sum_{\mathbf{x}} \sum_{\mathbf{x}'} Q_N(\mathbf{x}) Q_N(\mathbf{x}') \left[ \sum_{\mathbf{y}} \sqrt{P_N(\mathbf{y} \mid \mathbf{x}) P_N(\mathbf{y} \mid \mathbf{x}')} \right]^{1/\rho} \right\}^\rho \quad \rho \geq 1 \quad (5.7.7)$$

Equation 5.7.7 is valid for any discrete channel, memoryless or not, for which  $P_N(\mathbf{y} \mid \mathbf{x})$  can be defined. We now specialize this relation to discrete memoryless channels, using

$$P_N(\mathbf{y} \mid \mathbf{x}) = \prod_n P(y_n \mid x_n)$$

We also restrict  $Q_N(\mathbf{x})$  to be a product distribution,

$$Q_N(\mathbf{x}) = \prod_n Q(x_n).$$

These products can be expanded in (5.7.7). The details are the same as in (5.5.6) to (5.5.10) and the result is

$$\begin{aligned} P_{e,m} &< [4(M-1)]^\rho \prod_{n=1}^N \left\{ \sum_{x_n} \sum_{x'_n} Q(x_n)Q(x'_n) \left[ \sum_{y_n} \sqrt{P(y_n | x_n)P(y_n | x'_n)} \right]^{1/\rho} \right\}^\rho \\ &= [4(M-1)]^\rho \left\{ \sum_{k=0}^{K-1} \sum_{i=0}^{K-1} Q(k)Q(i) \left[ \sum_{j=0}^{J-1} \sqrt{P(j | k)P(j | i)} \right]^{1/\rho} \right\}^{\rho N} \end{aligned} \quad (5.7.8)$$

For an  $(N,R)$  code, we have  $(M-1) < e^{NR} \leq M$  code words, and (5.7.8) becomes

$$P_{e,m} < e^{N\rho[R + (\ln 4)/N]} \left\{ \sum_{k,i} Q(k)Q(i) \left[ \sum_j \sqrt{P(j | k)P(j | i)} \right]^{1/\rho} \right\}^{\rho N} \quad (5.7.9)$$

We can summarize our results in the following theorem.

**Theorem 5.7.1.** For an arbitrary discrete memoryless channel, let  $N$  be any positive integer and let  $R$  be any positive number. There exist  $(N,R)$  codes for which, for all  $m$ ,  $1 \leq m \leq [e^{NR}]$ ,

$$P_{e,m} \leq \exp \left[ -NE_{ex} \left( R + \frac{\ln 4}{N} \right) \right] \quad (5.7.10)$$

where the function  $E_{ex}$  is given by

$$E_{ex}(R') = \sup_{\rho \geq 1} \left[ -\rho R' + \max_{\mathbf{Q}} E_x(\rho, \mathbf{Q}) \right] \quad (5.7.11)$$

$$E_x(\rho, \mathbf{Q}) = -\rho \ln \sum_{k,i} Q(k)Q(i) \left[ \sum_j \sqrt{P(j | k)P(j | i)} \right]^{1/\rho} \quad (5.7.12)$$

and the max over  $\mathbf{Q}$  in (5.7.11) is over all probability assignments on the channel input letters.

The analysis of  $E_{ex}(R')$ , called the expurgated exponent, is almost the same as that of the random-coding exponent. Its behavior depends upon  $E_x(\rho, \mathbf{Q})$  in the same way that  $E_r(R)$  depends upon  $E_o(\rho, \mathbf{Q})$ .

**Theorem 5.7.2.** Let  $\mathbf{Q}$  be a probability assignment on the inputs of a discrete memoryless channel with transition probabilities  $P(j | k)$  and assume that  $\mathcal{I}(\mathbf{Q}; \mathbf{P}) > 0$  (see 5.6.23). Then, for all  $\rho > 0$ ,  $E_x(\rho, \mathbf{Q})$  is strictly increasing and convex  $\cap$  as a function of  $\rho$ . The convexity is strict unless the channel is noiseless in the sense that, for each pair  $(i,k)$  of inputs that are

used [that is,  $Q(i) > 0, Q(k) > 0$ ], we have either  $P(j | k)P(j | i) = 0$  for all  $j$  or  $P(j | k) = P(j | i)$  for all  $j$ .

This theorem is proved in Appendix 5B. We observe from (5.7.11) that  $E_{ex}(R)$  can be interpreted as the least upper bound of a set of linear functions of  $R$ ,

$$-\rho R + \max_Q E_x(\rho, \mathbf{Q})$$

a function of slope  $-\rho$  for each  $\rho \geq 1$ . (See Figure 5.7.1.) We observe that, for  $\rho = 1$ , the sum on  $j$  can be interchanged with that on  $i, k$  in the definition of  $E_x(\rho, \mathbf{Q})$ , and from this it can be seen that  $E_x(1, \mathbf{Q}) = E_o(1, \mathbf{Q})$ .

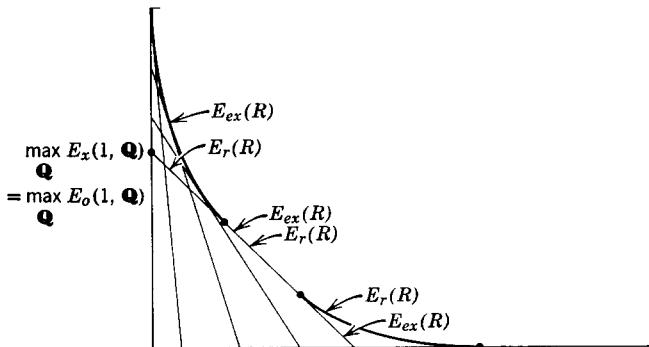


Figure 5.7.1. Comparison of  $E_{ex}(R)$  and  $E_r(R)$ .

This demonstrates the relation between  $E_{ex}(R)$  and  $E_r(R)$  as shown in Figure 5.7.1. Since  $E_x(\rho, \mathbf{Q})$  is strictly increasing with  $\rho$ , the above geometric interpretation proves that, in general,  $E_{ex}(R) > E_r(R)$  for sufficiently small  $R$ . In the limit of a very noisy channel, however, this difference becomes arbitrarily small, as shown in Problem 5.31.

In any region where  $E_{ex}(R) = E_r(R)$ , we note that  $\rho = 1$  and, thus,

$$P_{e,m} \leq \exp \left\{ -N \left[ E_{ex} \left( R + \frac{\ln 4}{N} \right) \right] \right\} = 4 \exp \left\{ -N \left[ -R + \max_Q E_x(1, \mathbf{Q}) \right] \right\} \quad (5.7.13)$$

which is precisely the same as the uniform bound on  $P_{e,m}$  given by (5.6.20). The expurgated bound of (5.7.10) is thus strictly tighter than the random-coding bound in the region where  $[R + (\ln 4)/N]$  is strictly less than the smallest  $R$  for which  $E_{ex}(R) = E_r(R)$ .

It can happen that  $E_{ex}(R)$  is infinite for sufficiently small  $R$ . To investigate this, we observe that the  $R$ -axis intercept of the linear function  $-\rho R + E_x(\rho, \mathbf{Q})$

is at  $E_x(\rho, \mathbf{Q})/\rho$ . As  $\rho \rightarrow \infty$ ,  $-\rho R + E_x(\rho, \mathbf{Q})$  approaches a vertical line at

$$R = \lim_{\rho \rightarrow \infty} E_x(\rho, \mathbf{Q})/\rho$$

and  $E_{ex}(R)$  is infinite for

$$R < \lim_{\rho \rightarrow \infty} E_x(\rho, \mathbf{Q})/\rho$$

(In other words, for such an  $R$ ,  $-\rho R + E_x(\rho, \mathbf{Q})$  approaches  $\infty$  as  $\rho \rightarrow \infty$ .) Evaluating this limit by L'Hospital's rule, we obtain

$$\lim_{\rho \rightarrow \infty} \frac{E_x(\rho, \mathbf{Q})}{\rho} = -\ln \left[ \sum_{k,i} Q(k)Q(i)\varphi_{k,i} \right] \quad (5.7.14)$$

$$\varphi_{k,i} = \begin{cases} 1; & \sum_j \sqrt{P(j|k)P(j|i)} \neq 0 \\ 0; & \text{otherwise} \end{cases} \quad (5.7.15)$$

Let  $R_{x,\infty}$  be the maximum value of (5.7.14) over  $\mathbf{Q}$ ,

$$R_{x,\infty} = \max_{\mathbf{Q}} -\ln \left[ \sum_{k,i} Q(k)Q(i)\varphi_{k,i} \right] \quad (5.7.16)$$

We have concluded that  $E_{ex}(R) = \infty$  for  $R < R_{x,\infty}$ .

It can be seen from (5.7.14) that  $R_{x,\infty} = 0$  if  $\varphi_{k,i} = 1$  for all  $k, i$  and, otherwise,  $R_{x,\infty} > 0$ . If  $\varphi_{k,i} = 0$  for some  $k$  and  $i$ , then those two inputs can never be confused at the receiver, and at least one bit per channel use can be transmitted over the channel, with no possibility of errors, by using only those two inputs. If we choose  $Q(k) = Q(i) = \frac{1}{2}$  for such a pair of inputs, the right-hand side of (5.7.14) is  $\ln 2$ . We see from this that, if  $R_{x,\infty} \neq 0$ , then  $R_{x,\infty}$  must be at least  $\ln 2$ . Shannon (1956) has defined the zero-error capacity of a channel as the largest rate at which data can be transmitted over the channel with zero-error probability (as opposed to arbitrarily small error probability). Since  $P_{e,m} = 0$  for  $R < R_{x,\infty}$ , we conclude that  $R_{x,\infty}$  is a lower bound to the zero-error capacity of the channel. Figure 5.7.2 sketches  $E_{ex}(R)$  for two channels with  $R_{x,\infty} > 0$ . In Problem 5.25, a simple expression is derived for  $R_{x,\infty}$ .

The actual maximization of  $E_{ex}(R)$  over  $\rho$  is quite similar to that of the random-coding exponent. If we define

$$E_{ex}(R, \mathbf{Q}) = \sup_{\rho \geq 1} -\rho R' + E_x(\rho, \mathbf{Q})$$

then we obtain the parametric relations

$$E_{ex}(R, \mathbf{Q}) = -\rho \frac{\partial E_x(\rho, \mathbf{Q})}{\partial \rho} + E_x(\rho, \mathbf{Q}) \quad (5.7.17)$$

$$R = \frac{\partial E_x(\rho, \mathbf{Q})}{\partial \rho} \quad (5.7.18)$$

valid for

$$R_{x,\infty} < R \leq \frac{\partial E_x(\rho, \mathbf{Q})}{\partial \rho} \Big|_{\rho=1} \quad (5.7.19)$$

For larger values of  $R$ , the bound is equivalent to the straight line portion of the random-coding bound, as discussed before. For the typical channel,

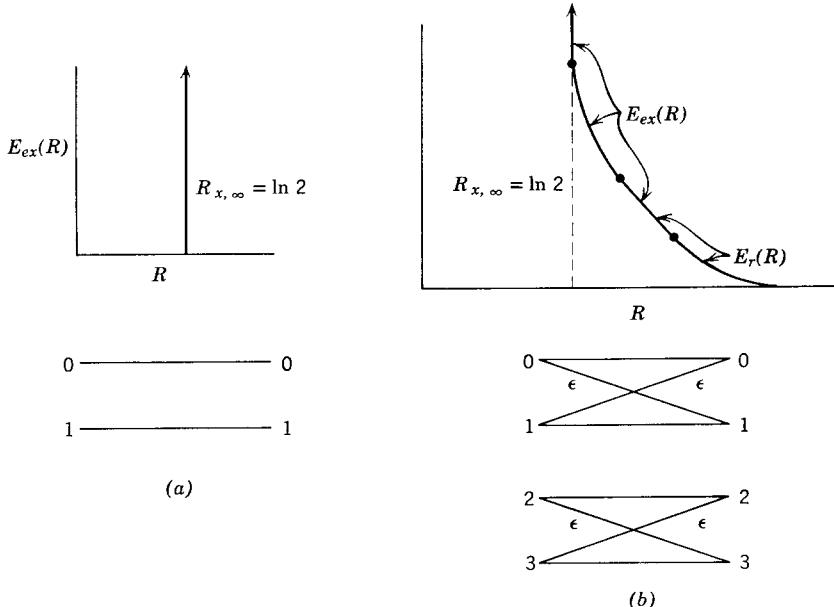


Figure 5.7.2.  $E_{ex}(R)$  for channels with  $R_{x,\infty} > 0$ .

$R_{x,\infty} = 0$ , and as shown in Problem 5.24, we then have

$$\lim_{R \rightarrow 0} E_{ex}(R, \mathbf{Q}) = - \sum_{k,i} Q(k)Q(i) \ln \left[ \sum_j \sqrt{P(j \mid k)P(j \mid i)} \right] \quad (5.7.20)$$

Relatively little is known about maximizing the bound over  $\mathbf{Q}$ . The function  $E_x(\rho, \mathbf{Q})$  is not convex  $\cap$  in  $\mathbf{Q}$  and can have several local maxima over  $\mathbf{Q}$ . Even more disturbing, if we try to optimize the expression for  $P_{e,m}$  in (5.7.7) over  $\mathcal{Q}_N(\mathbf{x})$ , not restricting ourselves to a product distribution, we sometimes find cases where a product distribution does not optimize the bound (see Problem 5.26). In the case of any discrete memoryless channel with a binary input, however, Jelinek (1968) has found that a product distribution always

optimizes the bound, and, in fact, the optimum  $\mathbf{Q}$  is  $Q(0) = Q(1) = \frac{1}{2}$  (see Problems 5.29 and 5.30).

### 5.8 Lower Bounds to Error Probability

In the preceding sections, we have established upper bounds on the probability of decoding error that can be achieved on a discrete memoryless channel in terms of the block length  $N$  and the rate  $R$  of the code. In this section, we shall be concerned with finding the minimum probability of error that can be achieved by any  $(N,R)$  code. We shall assume, throughout this section, that all code words are used with equal probability. Indeed, without some such assumption, no nonzero lower bound on error probability would be possible, for if one of the code words were used with probability 1, the decoder could always decode the message corresponding to that code word and no errors would ever be made.\*

For equally likely messages, the probability of decoding error for a code of  $M$  messages is

$$P_e = \frac{1}{M} \sum_{m=1}^M P_{e,m}$$

where  $P_{e,m}$  is the probability of error given that message  $m$  is transmitted. We saw in the last section that for any block length  $N$  and any rate  $R > 0$ , there exist  $(N,R)$  block codes for which both  $P_e \leq \exp[-NE_r(R)]$  and  $P_e \leq \exp[-NE_{ex}(R + \ln 4/N)]$ .

The known lower bounds on  $P_e$  for a given  $N$  and  $R$  are considerably more tedious and subtle to derive than the upper bounds. Thus, we shall merely state the relevant theorems here. Proofs may be found in Shannon, Gallager, and Berlekamp (1967). Proofs for the special case of the binary symmetric channel (BSC) will be presented here. Most of the ideas in finding lower bounds to  $P_e$  are present in this special case, but much of the detail is avoided.

**Theorem 5.8.1 (Sphere-Packing Bound).** For any  $(N,R)$  code on a discrete memoryless channel,

$$P_e \geq \exp(-N\{E_{sp}[R - o_1(N)] + o_2(N)\}) \quad (5.8.1)$$

where

$$E_{sp}(R) = \sup_{\rho > 0} \left[ \max_{\mathbf{Q}} E_0(\rho, \mathbf{Q}) - \rho R \right] \quad (5.8.2)$$

and  $E_o(\rho, \mathbf{Q})$  is given by (5.6.14). The quantities  $o_1(N)$  and  $o_2(N)$  go to zero

\* It is possible, however, to lower bound the error probability of the worst code word in a code, that is,  $\max P_{e,m}$ , without regard to the probabilities with which code words are used (see Problem 5.32).

with increasing  $N$  and can be taken as

$$o_1(N) = \frac{\ln 8}{N} + \frac{K \ln N}{N} \quad (5.8.3)$$

$$o_2(N) = \frac{\ln 8}{N} + \sqrt{\frac{2}{N}} \ln \frac{e^2}{P_{\min}} \quad (5.8.4)$$

where  $P_{\min}$  is the smallest nonzero transition probability for the channel and  $K$  is the size of the input alphabet.

It will be observed that the function  $E_{sp}(R)$ , called the sphere-packing exponent, is defined in almost the same way as the random-coding exponent  $E_r(R)$ , the only difference being in the range of  $\rho$  over which the maximization is performed. As a consequence, the results of Section 5.6 are immediately applicable to  $E_{sp}(R)$ . In particular,  $E_{sp}(R)$  is positive for  $0 < R < C$ , decreasing with increasing  $R$ , and convex  $\cup$ . In Figure 5.8.1,  $E_{sp}(R)$  is sketched for a number of channels and compared with  $E_r(R)$ . Figure 5.8.1a shows the typical behavior and the other figures are in a sense pathological,  $E_{sp}(R)$  being infinite for all rates less than a given constant called  $R_\infty$ . To find this constant, we interpret

$$[\max_Q E_o(\rho, \mathbf{Q}) - \rho R]$$

as a set of linear functions of  $R$  with  $\rho > 0$  as a parameter (see Figure 5.6.3). The  $R$  axis intercept of the above function for a given  $\rho$  is

$$\max_Q E_o(\rho, \mathbf{Q})/\rho$$

As  $\rho \rightarrow \infty$ , these straight lines approach infinite slopes and, since  $E_{sp}(R)$  is the convex hull of the above functions,  $R_\infty$  is given by the limiting  $R$  axis intercept as  $\rho \rightarrow \infty$ ,

$$R_\infty = \lim_{\rho \rightarrow \infty} \max_Q \frac{E_o(\rho, \mathbf{Q})}{\rho} \quad (5.8.5)$$

Finding the limit either by L'Hospital's rule or by expanding  $E_o(\rho, \mathbf{Q})$  in a power series in  $1/(1 + \rho)$ , we obtain

$$R_\infty = -\ln \left[ \min_Q \max_j \sum_k Q(k) \phi(j \mid k) \right] \quad (5.8.6)$$

where

$$\phi(j \mid k) = \begin{cases} 1; & P(j \mid k) \neq 0 \\ 0; & P(j \mid k) = 0 \end{cases} \quad (5.8.7)$$

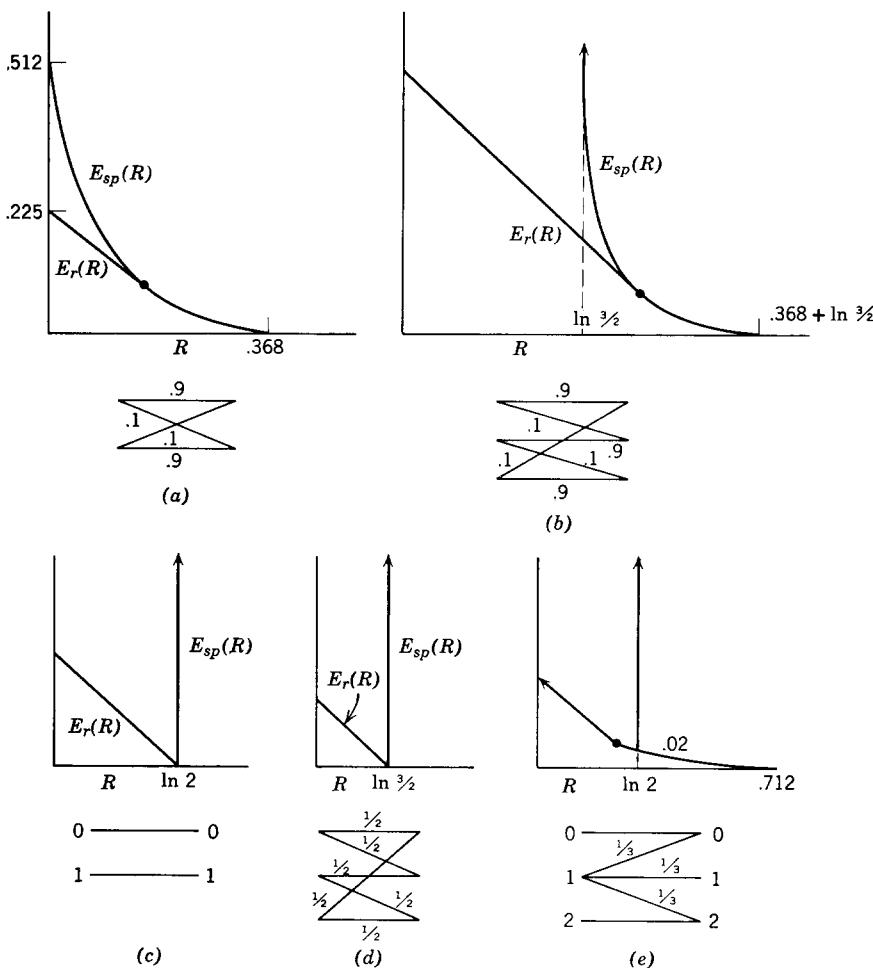


Figure 5.8.1. Comparison of sphere-packing and random-coding exponents.

That is, for each output, we sum the input probabilities from which that output can be reached. The input probabilities are adjusted to minimize the largest of these sums, and  $R_\infty$  is minus the logarithm of this min-max sum. It can be seen from this that  $R_\infty = 0$  unless each output is unreachable from at least one input.

We next observe that the value of  $\rho$  that maximizes (5.8.2) is decreasing with  $R$ . Furthermore, if the maximizing  $\rho$  lies between 0 and 1, then  $E_r(R) = E_{sp}(R)$ . Thus, if  $E_r(R) = E_{sp}(R)$  for one value of  $R$ , equality also holds for

all larger values of  $R$ . We define  $R_{cr}$  as the smallest such  $R$ , that is, the value such that  $E_{sp}(R) = E_r(R)$  iff  $R \geq R_{cr}$ . In other words, for any channel, there is a range of rates  $R_{cr} \leq R \leq C$  for which the upper and lower bounds to error probability agree in their exponential dependence on  $N$ .

The *reliability function* of a channel,  $E(R)$ , is defined as

$$E(R) = \lim_{N \rightarrow \infty} \sup \frac{-\ln P_e(N, R)}{N} \quad (5.8.8)$$

where  $P_e(N, R)$  is the minimum value of  $P_e$  over all  $(N, R)$  codes for a given  $N$  and  $R$ . Thus reliability is the exponent with which error probability may be made to vanish with increasing  $N$ . The exponents  $E_r(R)$  and  $E_{sp}(R)$  are lower and upper bounds respectively on  $E(R)$  and, as observed above,  $E(R)$  is known exactly for  $R_{cr} \leq R \leq C$ . It is, of course, surprising that the rather crude bounds that we used in Section 5.6 yield the true reliability function of the channel over a range of rates. On the other hand, those bounding techniques were selected in retrospect because they did yield the reliability function. There are many other ways to upper bound error probability, often appearing less crude, which give weaker results.

It can happen, as for example in parts *c* and *d* of Figure 5.8.1, that  $R_{cr} = C$ . This implies that, for values of  $R$  arbitrarily close to  $C$ ,

$$\left[ \max_{\mathbf{Q}} E_o(\rho, \mathbf{Q}) - \rho R \right]$$

is not maximized by  $\rho$  in the range  $0 \leq \rho \leq 1$ . This implies in turn either that the  $R$  axis intercept

$$\max_{\mathbf{Q}} E_o(\rho, \mathbf{Q})/\rho$$

is greater than or equal to  $C$  for some  $\rho \geq 1$  or that  $\max_{\mathbf{Q}} E_o(\rho, \mathbf{Q})/\rho$  approaches  $C$  arbitrarily closely as  $\rho \rightarrow \infty$ . In either case, since  $E_o(\rho, \mathbf{Q})$  is convex in  $\rho$  for fixed  $\mathbf{Q}$ , we must have  $E_o(\rho, \mathbf{Q}) = \rho C$  for some choice of  $\mathbf{Q}$ . From Theorem 5.6.3, this can happen iff (5.6.26a) is satisfied for that  $\mathbf{Q}$ . Using the same argument, these conditions are also necessary and sufficient for  $R_\infty = C$ . Summarizing, the three following statements are all equivalent: (1)  $R_{cr} = C$ ; (2)  $R_\infty = C$ ; and (3) Equation 5.6.26a is satisfied for some  $\mathbf{Q}$  yielding capacity.

**Theorem 5.8.2 (Straight-Line Bound).** For any discrete memoryless channel for which  $E_{ex}(0) < \infty$ , let  $E_{sl}(R)$  be the smallest linear function of  $R$  which touches the curve  $E_{sp}(R)$  and also satisfies  $E_{sl}(0) = E_{ex}(0)$ . Let  $R_1$  be the  $R$  for which  $E_{sl}(R) = E_{sp}(R)$ . Let  $o_3(N)$  and  $o_4(N)$  be functions

vanishing with increasing  $N$  which can be taken as

$$o_3(N) = \frac{\ln 2}{\sqrt{N}} + \frac{\ln 8 + K \ln N}{N} \quad (5.8.9)$$

$$\begin{aligned} o_4(N) = & \frac{2\sqrt{K} \max_{i,k} \left[ -2 \ln \sum_j \sqrt{P(j|i)P(j|k)} \right]}{\sqrt{\lceil \log_2 \sqrt{N} \rceil}} + \sqrt{\frac{8}{N}} \ln \frac{e}{P_{\min}} \\ & + \frac{\ln 2}{\sqrt{N}} + \frac{5 \ln 2}{N} + \frac{E_{\text{exp}}(0)}{N} \end{aligned} \quad (5.8.10)$$

Then, for any positive integer  $N$  and any  $R$ ,  $o_3(N) \leq R \leq R_1$ , every  $(N,R)$  code has an error probability satisfying

$$P_e > \exp(-N\{E_{\text{sl}}[R - o_3(N)] + o_4(N)\}) \quad (5.8.11)$$

---

This is Theorem 4 in Shannon, Gallager, and Berlekamp (1967) and a proof is given there. There is a slight error in the statement of that Theorem 4 in that the restriction  $R \geq o_3(N)$  is omitted. The proof is correct for the theorem as stated here, however. The straight-line exponent  $E_{\text{sl}}(R)$ , combined with the sphere-packing exponent, gives an upper bound to the reliability  $E(R)$  for all  $R$ ,  $0 < R < C$  in the limit of large  $N$ . These bounds are sketched in Figure 5.8.2 for the same channels as in Figure 5.8.1.

We now turn our attention to proving these theorems for the special case of the BSC. We shall start with an arbitrary  $(N,R)$  code and find a lower bound to  $P_e$  which is independent of the code and thus a lower bound to  $P_e$  for all codes of the given  $N$  and  $R$ . Let  $\mathbf{x}_1, \dots, \mathbf{x}_M$  be the code words in the code,  $M = \lceil e^{NR} \rceil$ , and let  $Y_1, \dots, Y_M$  be the decoding regions for the code (that is,  $Y_m$  is the set of channel-output sequences that are decoded into message  $m$ ). If message  $m$  enters the encoder,  $\mathbf{x}_m$  will be transmitted, and correct decoding will ensue if a  $\mathbf{y} \in Y_m$  is received. Since, conditional on  $m$ , this is an event of probability

$$\sum_{\mathbf{y} \in Y_m} P(\mathbf{y} | \mathbf{x}_m)$$

the overall probability of correct decoding is

$$P_c = \frac{1}{M} \sum_{m=1}^M \sum_{\mathbf{y} \in Y_m} P(\mathbf{y} | \mathbf{x}_m) \quad (5.8.12)$$

Now define the *Hamming distance*  $d(\mathbf{y}; \mathbf{x})$  between two binary sequences as the number of positions in which the two sequences differ. For example, the Hamming distance between  $(0,0,1,1,1)$  and  $(1,0,1,0,1)$  is 2. For a BSC with crossover probability  $\epsilon$ , if  $d(\mathbf{y}; \mathbf{x}_m) = n$ , then  $P(\mathbf{y} | \mathbf{x}_m) = \epsilon^n(1 - \epsilon)^{N-n}$ . If

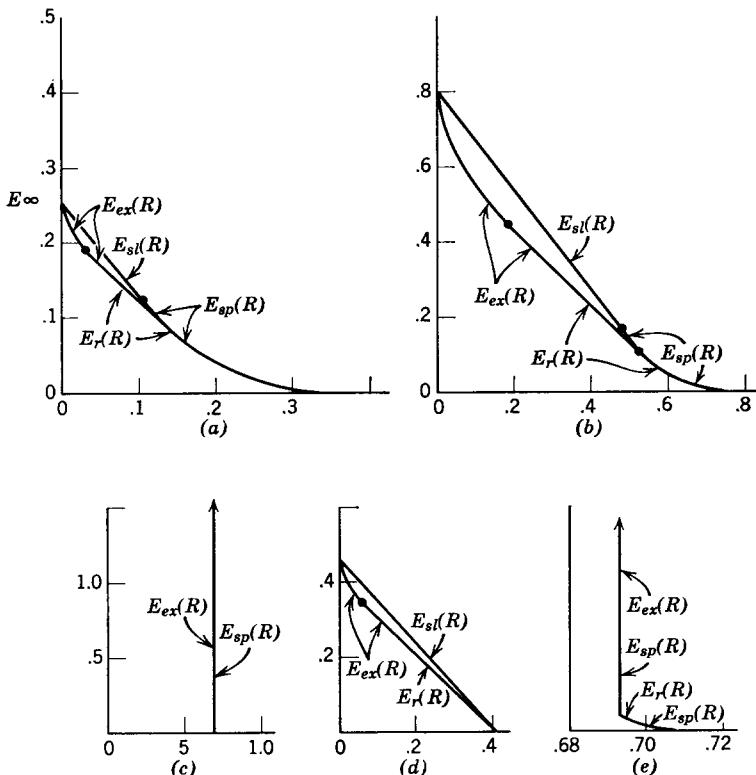


Figure 5.8.2. Bounds on reliability function (same channels as Figure 5.8.1).

we let  $A_{n,m}$  be the number of sequences  $\mathbf{y}$  that are decoded into message  $m$  and have distance  $n$  from  $\mathbf{x}_m$ , then we can rewrite (5.8.12) as

$$P_c = \frac{1}{M} \sum_{m=1}^M \sum_{n=0}^N A_{n,m} \epsilon^n (1 - \epsilon)^{N-n} \quad (5.8.13)$$

Using the binomial equality

$$1 = \sum_{n=0}^N \binom{N}{n} \epsilon^n (1 - \epsilon)^{N-n}$$

the probability of error,  $P_e = 1 - P_c$  is given by

$$P_e = \frac{1}{M} \sum_{m=1}^M \sum_{n=0}^N \left[ \binom{N}{n} - A_{n,m} \right] \epsilon^n (1 - \epsilon)^{N-n} \quad (5.8.14)$$

To interpret this, we notice that, if message  $m$  is transmitted,  $A_{n,m}$  is the

number of sequences at distance  $n$  from  $\mathbf{x}_m$  which, if received, will cause correct decoding. Since  $\binom{N}{n}$  is the total number of sequences at distance  $n$  from  $\mathbf{x}_m$ ,  $\left[ \binom{N}{n} - A_{n,m} \right]$  of these sequences will cause decoding errors if  $m$  is transmitted.

We shall now find some constraints on the set of numbers  $\{A_{n,m}\}$  which are valid for all codes with the given  $N$  and  $M$  and then minimize the right-hand side of (5.8.14) subject to these constraints. The constraints that we shall use are

$$A_{n,m} \leq \binom{N}{n}; \quad \text{all } n, m \quad (5.8.15)$$

$$\sum_{m=1}^M \sum_{n=0}^N A_{n,m} \leq 2^N \quad (5.8.16)$$

The constraint (5.8.16) arises from the fact that there are  $2^N$  output sequences; each is decoded into at most one message, and is at a unique distance from the associated code word.

The minimum of (5.8.14), subject to these constraints, is achieved for  $\epsilon < \frac{1}{2}$  by selecting, for all  $m$ ,

$$A_{n,m} = \begin{cases} \binom{N}{n}; & 0 \leq n \leq k-1 \\ 0; & k+1 \leq n \leq N \end{cases} \quad (5.8.17)$$

where  $k$  is chosen so that

$$M \sum_{n=0}^{k-1} \binom{N}{n} + \sum_{m=1}^M A_{k,m} = 2^N; \quad 0 < \sum_m A_{k,m} \leq M \binom{N}{k} \quad (5.8.18)$$

The individual values  $A_{k,m}$  are immaterial so long as their sum over  $m$  satisfies (5.8.18). To see that this choice minimizes (5.8.14), observe that for any other choice, we can find an  $n'$  and an  $n$ ,  $n' < n$  such that  $A_{n',m'} < \binom{N}{n'}$  for some  $m'$  and  $A_{n,m} > 0$  for some  $m$ . For such a choice, it can be seen that (5.8.14) is reduced by increasing  $A_{n',m'}$  by 1 and decreasing  $A_{n,m}$  by 1. Substituting (5.8.17) into (5.8.14) and observing that the result is a lower bound on  $P_e$  for all codes with the given values of  $N$  and  $M$ ,

$$P_e(N,M) > \left[ \binom{N}{k} - \frac{1}{M} \sum_{m=1}^M A_{k,m} \right] \epsilon^k (1-\epsilon)^{N-k} + \sum_{n=k+1}^N \binom{N}{n} \epsilon^n (1-\epsilon)^{N-n} \quad (5.8.19)$$

where  $P_e(N, M)$  is defined as the minimum probability of error over all codes of the given block length  $N$  and number of code words  $M$ .

This bound is known as the sphere-packing bound. We can interpret the set of sequences at distance  $k$  or less from a code word as a sphere of radius  $k$  around that code word. The bound in (5.8.17) is the error probability that would result if we could choose code words such that the set of spheres of radius  $k$  around the different code words exhausted the space of binary  $N$ -tuples and intersected each other only on the outer shells of radius  $k$ . Such codes are called sphere-packed codes, and the code in Figure 5.2.1 is an example (in that case, not even the outer shells at radius 1 intersect). This bound is often derived by first finding the error probability for a sphere-packed code and then showing that a sphere-packed code has a probability of error at least as small as that of any other code with the same  $N$  and  $M$ . There is a logical error in such a derivation in that, for most values of  $N$  and  $M$ , sphere-packed codes do not exist.

We now want to modify (5.8.19) to an analytically simpler but slightly weaker form. There are a number of ways of doing this, all resulting in the same exponential bound on error probability but with different coefficients. The coefficients here are quite weak, but eliminate some messy details in the proof of Theorem 5.8.4. For tight numerical results, especially for small  $N$ , we should work directly with (5.8.19).

**Theorem 5.8.3 (Sphere-Packing Bound for BSC).** For a binary symmetric channel with crossover probability  $\epsilon < \frac{1}{2}$ , let  $\delta$  be an arbitrary number  $\epsilon \leq \delta \leq \frac{1}{2}$  and let  $\mathcal{H}(\delta) = -\delta \ln \delta - (1 - \delta) \ln (1 - \delta)$ . If the number of code words  $M$  satisfies

$$M \geq \sqrt{8(N + 1)} \exp \{N[\ln 2 - \mathcal{H}(\delta)]\} \quad (5.8.20)$$

then

$$P_e(N, M) \geq \frac{\epsilon}{(1 - \epsilon)\sqrt{8(N + 1)}} \exp \{N[\mathcal{H}(\delta) + \delta \ln \epsilon + (1 - \delta) \ln (1 - \epsilon)]\} \quad (5.8.21)$$

*Proof.* Observe from (5.8.14) that any increase in any of the sums

$$\sum_{m=1}^M A_{n,m}$$

over the values in (5.8.17) and (5.8.18) will further lower bound  $P_e$ . Let  $n' = \lceil \delta N \rceil$ , and choose

$$\sum_{m=1}^M A_{n',m} = 2^N$$

For all  $n \neq n'$ , choose

$$\sum_{m=1}^M A_{n,m} = \binom{N}{n} M$$

These choices clearly overbound those in (5.8.17) and (5.8.18), and substituting them into (5.8.14) yields

$$P_e(N, M) \geq \left[ \binom{N}{n'} - \frac{2^N}{M} \right] \epsilon^{n'} (1 - \epsilon)^{N-n'} \quad (5.8.22)$$

Using the Stirling bound to a factorial, we can lower bound  $\binom{N}{n'}$  by (see Problem 5.8b):

$$\binom{N}{n'} \geq \frac{1}{\sqrt{2N}} \exp \left[ N \mathcal{H} \left( \frac{n'}{N} \right) \right] \quad (5.8.23)$$

If  $n' \leq N/2$ , we have  $\mathcal{H}(n'/N) \geq \mathcal{H}(\delta)$ . Also  $1/N \geq 1/(N+1)$  so that

$$\binom{N}{n'} \geq \frac{1}{\sqrt{2(N+1)}} \exp [N \mathcal{H}(\delta)] \quad (5.8.24)$$

Since  $\delta < \frac{1}{2}$  and  $n' = [\delta N]$ , the only possible value of  $n'$  greater than  $N/2$  is  $(N+1)/2$ . In this case, we use the special bound (see Problem 5.8b):

$$\binom{N}{(N+1)/2} \geq \frac{1}{\sqrt{2(N+1)}} 2^N \quad (5.8.25)$$

Since  $2^N \geq \exp N \mathcal{H}(\delta)$ , the bound in (5.8.24) is valid for all possible  $n'$ . Also, since  $n'$  exceeds  $\delta N$  by at most 1, we have

$$\left( \frac{\epsilon}{1-\epsilon} \right)^{n'} \geq \frac{\epsilon}{1-\epsilon} \left( \frac{\epsilon}{1-\epsilon} \right)^{\delta N}$$

Substituting this and (5.8.24) into (5.8.22) and using the bound on  $M$  in (5.8.20), we have (5.8.21), completing the proof. |

If we calculate  $E_{sp}(R)$  [as given by (5.8.2)], using the same argument as in Example 1 of Section 5.6, we obtain, for  $R < C$ ,

$$E_{sp}(R) = -\delta \ln \epsilon - (1 - \delta) \ln (1 - \epsilon) - \mathcal{H}(\delta) \quad (5.8.26)$$

$$R = \ln 2 - \mathcal{H}(\delta) \quad (5.8.27)$$

Now consider an arbitrary  $(N, R)$  code with  $(1/N) \ln [8(N+1)] < R < C$ . If we choose  $\delta$  to satisfy

$$R = \ln 2 - \mathcal{H}(\delta) + \frac{\ln [8(N+1)]}{N}$$

then  $M = \lceil \exp NR \rceil$  must satisfy (5.8.20), and from (5.8.21) we get a result equivalent to Theorem 5.8.1.

$$P_e(N, M) \geq \exp -N \left\{ E_{sp} \left( R - \frac{\ln \sqrt{8(N+1)}}{N} \right) + \frac{\ln [\sqrt{8(N+1)}(1-\epsilon)/\epsilon]}{N} \right\} \quad (5.8.28)$$

We shall subsequently derive a stronger bound than Theorem 5.8.1 for  $R > C$ .

In order to establish Theorem 5.8.2 for the BSC, we first need several lemmas which are rather interesting in their own right. The first involves a concept called list decoding. Suppose that, for a given set of  $M$  code words of length  $N$ , the decoder maps each received sequence into a list of, say,  $L$  messages. Such a scheme might be useful if we were planning to employ feedback in a communication system and in a subsequent transmission to resolve the uncertainty as to which of the  $L$  decoded messages were actually transmitted. If the transmitted message is not on the list of  $L$  decoded messages, then we say that a *list-decoding error* has occurred. Let  $P_e(N, M, L)$  be the minimum probability of list-decoding error over all codes with  $M$  code words, a block length of  $N$ , and a list of  $L$  decoded messages. We can repeat the sphere-packing bound for a list-decoding scheme, letting  $Y_m$  be the set of output sequences  $\mathbf{y}$  for which  $m$  is on the decoding list. Equation 5.8.14 is again valid, interpreting  $A_{n,m}$  as the number of output sequences  $\mathbf{y}$  for which  $m$  is on the decoding list and which are distance  $n$  from  $\mathbf{x}_m$ . Under these circumstances, the constraint (5.8.16) on the  $A_{n,m}$  is modified to

$$\sum_{m=1}^M \sum_{n=0}^N A_{n,m} = L2^N$$

since each  $\mathbf{y}$  is decoded into exactly  $L$  messages and, thus, contributes to exactly  $L$  of the terms  $A_{n,m}$ . With this modification, we obtain the following lemma.

**LEMMA 1.** For a BSC with transition probability  $\epsilon < \frac{1}{2}$ , let  $\delta$  be an arbitrary number,  $\epsilon < \delta < \frac{1}{2}$ . If

$$\frac{M}{L} \geq \sqrt{8(N+1)} \exp \{N[\ln 2 - \mathcal{H}(\delta)]\} \quad (5.8.29)$$

then

$$P_e(N, M, L) \geq \frac{\epsilon}{(1-\epsilon)\sqrt{8(N+1)}} \exp \{N[H(\delta) + \delta \ln \epsilon + (1-\delta) \ln (1-\epsilon)]\} \quad (5.8.30)$$


---

*Proof.* The proof is the same as that of Theorem 6.8.3, with the modification that we choose

$$\sum_{m=1}^M A_{m,n} = 2^N L$$

for all  $n \neq n'$ . |

We now shift our attention to obtaining a better lower bound on error probability than the sphere-packing bound for very small numbers of code words. The *minimum distance* of a binary code is defined as the distance between the two nearest code words.

**LEMMA 2 (PLOTKIN BOUND).** The minimum distance for any binary code of  $M$  code words and block length  $N$  satisfies

$$d_{\min} \leq \frac{NM}{2(M-1)} \quad (5.8.31)$$


---

*Proof.* Consider listing the code words of an  $(N, M)$  code in an  $N$  column,  $M$  row binary array, the  $m$ th code word being the  $m$ th row of the array. Now consider the sum of all distances in the code,

$$\sum_{m=1}^M \sum_{m'=1}^M d(\mathbf{x}_m; \mathbf{x}_{m'}) = \sum_{n=1}^N \sum_{m=1}^M \sum_{m'=1}^M d(x_{m,n}; x_{m',n}) \quad (5.8.32)$$

Let  $Z(n)$  be the number of zeros in the  $n$ th column of the array. Since there are  $Z(n)$  different values of  $m'$  for which  $x_{m',n} = 0$ , we notice that, if  $x_{m,n} = 1$ , then

$$\sum_{m=1}^M d(x_{m,n}; x_{m',n}) = Z(n).$$

Since there are  $M - Z(n)$  values of  $m$  for which  $x_{m,n} = 1$ ,

$$\sum_{m: x_{m,n}=1} \sum_{m'=1}^M d(x_{m,n}; x_{m',n}) = [M - Z(n)]Z(n) \quad (5.8.33)$$

Likewise, there are  $Z(n)$  values of  $m$  for which  $x_{m,n} = 0$ , and  $M - Z(n)$  values of  $m'$  for which  $x_{m',n} = 1$ , so that

$$\sum_{m=1}^M \sum_{m'=1}^M d(x_{m,n}; x_{m',n}) = 2[M - Z(n)]Z(n) \quad (5.8.34)$$

The right-hand side of (5.8.34) is upper bounded by  $M^2/2$ , which is the maximum value of  $2Z(M - Z)$  as a function of  $Z$ , achieved at  $Z = M/2$ . Thus

$$\sum_{m=1}^M \sum_{m'=1}^M d(\mathbf{x}_m; \mathbf{x}_{m'}) \leq \frac{NM^2}{2} \quad (5.8.35)$$

On the other hand, since  $d(\mathbf{x}_m; \mathbf{x}_{m'}) = 0$ , we can omit the terms above where  $m' = m$ , getting  $M(M - 1)$  nonzero terms,

$$\sum_{m=1}^M \sum_{m'=1}^M d(\mathbf{x}_m; \mathbf{x}_{m'}) = \sum_{m=1}^M \sum_{m' \neq m} d(\mathbf{x}_m; \mathbf{x}_{m'}) \geq M(M - 1)d_{\min} \quad (5.8.36)$$

Combining (5.8.35) and (5.8.36) yields (5.8.31). |

Now define  $P_{e,w}(N, M)$  as the minimum probability of error for the worst code word in a code, minimized over all codes of a given block length  $N$  and number of code words  $M$ .

$$P_{e,w}(N, M) = \min_{\text{codes}} \left[ \max_m P_{e,m} \right]$$

**LEMMA 3.** For a BSC with crossover probability  $\epsilon < \frac{1}{2}$  and for  $M > N + 2$ ,

$$P_{e,w}(N, M) \geq \frac{\sqrt{N} \epsilon}{(N + 4)(1 - \epsilon)} \exp[-NE_{ex}(0)] \quad (5.8.37)$$

where

$$E_{ex}(0) = -\frac{1}{4} \ln 4\epsilon(1 - \epsilon) \quad (5.8.38)$$

---

Notice that the exponent  $E_{ex}(0)$  is the value of the expurgated exponent at  $R = 0$  [see (5.7.20)].

*Proof.* Since  $d_{\min}$  must be an integer, it is easy to verify that (5.8.31) implies that for  $M \geq N + 2$ ,  $d_{\min} \leq N/2$ . Suppose that two code words in a given code,  $\mathbf{x}_m$  and  $\mathbf{x}_{m'}$ , are at distance  $d = d_{\min}$  from each other. We have

$$P_{e,m} + P_{e,m'} = \sum_{\mathbf{y} \in Y_m \setminus \mathbf{x}_m} P(\mathbf{y} \mid \mathbf{x}_m) + \sum_{\mathbf{y} \in Y_{m'} \setminus \mathbf{x}_{m'}} P(\mathbf{y} \mid \mathbf{x}_{m'}) \quad (5.8.38a)$$

We can lower bound  $P_{e,m} + P_{e,m'}$  by enlarging  $Y_m$  and  $Y_{m'}$  so that all  $\mathbf{y}$  are decoded into either  $m$  or  $m'$ , and further lower bound by modifying  $Y_m$  and  $Y_{m'}$  to be maximum-likelihood decoding regions for the two code words. With these modifications, we can ignore what is received in the positions for which  $\mathbf{x}_m$  and  $\mathbf{x}_{m'}$  agree, and thus consider only the probability of more than  $d/2$  channel crossovers among the  $d$  digits for which  $\mathbf{x}_m$  and  $\mathbf{x}_{m'}$  differ.

$$\frac{P_{e,m} + P_{e,m'}}{2} \geq \sum_{i > d/2} \binom{d}{i} \epsilon^i (1 - \epsilon)^{d-i} \dots \quad (5.8.39)$$

For  $d$  even, this is lower bounded by

$$\begin{aligned} \frac{P_{e,m} + P_{e,m'}}{2} &\geq \left( \frac{d}{\frac{d}{2} + 1} \right) \epsilon^{(d/2)+1} (1 - \epsilon)^{(d/2)-1} \\ &= \frac{d\epsilon}{(d+2)(1-\epsilon)} \left( \frac{d}{d/2} \right) \epsilon^{d/2} (1 - \epsilon)^{d/2} \\ &\geq \frac{d\epsilon}{(d+2)(1-\epsilon)\sqrt{2d}} \exp \left\{ \frac{d}{2} \ln [4\epsilon(1-\epsilon)] \right\} \end{aligned}$$

where we have used (5.8.23). This is a decreasing function of  $d$  for even  $d$ , and since  $d \leq N/2$ , we have

$$\frac{P_{e,m} + P_{e,m'}}{2} \geq \frac{\sqrt{N}\epsilon}{(N+4)(1-\epsilon)} \exp \left\{ \frac{N}{4} \ln [4\epsilon(1-\epsilon)] \right\} \quad (5.8.40)$$

For  $d$  odd, the argument is almost identical, using (5.8.25) to lower bound  $\binom{d}{(d+1)/2}$ . The result is

$$\frac{P_{e,m} + P_{e,m'}}{2} \geq \sqrt{\frac{\epsilon}{(1-\epsilon)(N+2)}} \exp \left\{ \frac{N}{4} \ln [4\epsilon(1-\epsilon)] \right\}$$

which is lower bounded by (5.8.40), completing the proof. |

To interpret the above lemma, we notice that the rate of a code with  $M = N + 2$  code words is approaching zero as  $N$  approaches  $\infty$ . Since the exponent in the lemma is the same as the zero-rate expurgated random-coding exponent,\* we know that  $E_{ex}(0)$  is the reliability of the channel in the limit as  $R$  approaches 0. We must be rather careful with the above statement, however. Lemma 3 does not apply for  $M < N + 2$ , and in fact for fixed  $M$ , it can be shown† for the BSC that

$$\lim_{N \rightarrow \infty} \frac{-\ln P_e(N,M)}{N} = \frac{M}{M-1} E_{ex}(0)$$

One interesting aspect of the lemma is that it brings out the importance of the minimum distance of a low-rate code in determining its error probability. This importance was also seen in the expurgated random-coding bound where we eliminated code words that were too close to each other. At high

\* See Problem 5.32 for the relation between  $P_{e,w}(N,M)$  and  $P_e(N,M)$ .

† Shannon, Gallager, and Berlekamp (1967-II), pp. 529–530.

rates, close to capacity, minimum distances become relatively unimportant, and it is not difficult to see that, in the random-coding ensemble, most codes have a very small minimum distance. It is possible to expurgate the ensemble, greatly increasing the minimum distance of the codes, but at high rates this cannot materially lower the ensemble average error probability since this average is close to the sphere-packing bound to start with.

LEMMA 4. For arbitrary positive integers  $N_1$ ,  $N_2$ ,  $M$ , and  $L$ ,

$$P_e(N_1 + N_2, M) \geq P_e(N_1, M, L)P_{e,w}(N_2, L + 1) \quad (5.8.41)$$


---

The intuitive idea behind this lemma is that, for a code of block length  $N_1 + N_2$ , a decoding error will be made if, on the basis of the first  $N_1$  received digits, there are  $L$  messages more likely than the transmitted message and if, on the basis of the final  $N_2$  received digits, one of these  $L$  messages is again more likely than the transmitted message. The probabilities on the right-hand side of (5.8.41) are related to the probabilities of these events. Although we are concerned here only with the BSC, the following proof applies to arbitrary discrete memoryless channels.

*Proof.* For a given code with  $M$  code words of length  $N_1 + N_2$ , let  $\mathbf{x}_m$  be the  $m$ th code word and let the prefix,  $\mathbf{x}_{m,1}$  be the first  $N_1$  digit of  $\mathbf{x}_m$  and let the suffix,  $\mathbf{x}_{m,2}$  be the final  $N_2$  digits. Similarly, the received sequence  $\mathbf{y}$  is separated into the prefix  $\mathbf{y}_1$  and the suffix  $\mathbf{y}_2$  of  $N_1$  and  $N_2$  letters respectively. For each  $m$ ,  $1 \leq m \leq M$ , let  $Y_m$  be the set of output sequences  $\mathbf{y}$  decoded into message  $m$  and let  $Y_m^c$  be the complement of  $Y_m$ . Then

$$P_e = \frac{1}{M} \sum_{m=1}^M \sum_{\mathbf{y} \in Y_m^c} P(\mathbf{y} \mid \mathbf{x}_m) \quad (5.8.42)$$

For each prefix  $\mathbf{y}_1$ , let  $Y_{m,2}(\mathbf{y}_1)$  be the set of suffixes  $\mathbf{y}_2$  for which  $(\mathbf{y}_1, \mathbf{y}_2) \in Y_m$ . For a memoryless channel,  $P(\mathbf{y}_1, \mathbf{y}_2 \mid \mathbf{x}_m) = P(\mathbf{y}_1 \mid \mathbf{x}_{m,1})P(\mathbf{y}_2 \mid \mathbf{x}_{m,2})$ , and we can rewrite (5.8.42) as

$$P_e = \frac{1}{M} \sum_{m=1}^M \sum_{\mathbf{y}_1} P(\mathbf{y}_1 \mid \mathbf{x}_{m,1}) \sum_{\mathbf{y}_2 \in Y_{m,2}^c(\mathbf{y}_1)} P(\mathbf{y}_2 \mid \mathbf{x}_{m,2}) \quad (5.8.43)$$

For any given  $\mathbf{y}_1$ , we can regard the set of suffixes  $\{\mathbf{x}_{m,2}\}$ ,  $1 \leq m \leq M$ , and the set of regions  $\{Y_{m,2}(\mathbf{y}_1)\}$   $1 \leq m \leq M$  as a code. The error probabilities for the words in this code, conditioned on  $\mathbf{y}_1$ , are given by

$$P_{e,m}(\mathbf{y}_1) = \sum_{\mathbf{y}_2 \in Y_{m,2}^c(\mathbf{y}_1)} P(\mathbf{y}_2 \mid \mathbf{x}_{m,2}) \quad (5.8.44)$$

Let  $m_1(\mathbf{y}_1)$  be the  $m$  for which  $P_{e,m}(\mathbf{y}_1)$  is smallest  $m_2(\mathbf{y}_1)$  the  $m$  for which  $P_{e,m}(\mathbf{y}_1)$  is next smallest, and so on. We maintain that for all  $m$  except

perhaps  $m_1(\mathbf{y}_1), \dots, m_L(\mathbf{y}_1)$ ,

$$P_{e,m}(\mathbf{y}_1) \geq P_{e,w}(N_2, L + 1)$$

If this were not so, then the set of  $L + 1$  code words  $\{\mathbf{x}_{m,2}\}$  for  $m = m_1(\mathbf{y}_1), \dots, m_{L+1}(\mathbf{y}_1)$  and the set of decoding regions  $\{Y_{m,2}(\mathbf{y}_1)\}$  for  $m = m_1(\mathbf{y}_1), \dots, m_{L+1}(\mathbf{y}_1)$  would all have error probabilities less than  $P_{e,w}(N_2, L + 1)$ , which is a contradiction. We then have the lower bound,

$$\sum_{\mathbf{y}_2 \in Y_{m,2}^c(\mathbf{y}_1)} P(\mathbf{y}_2 \mid \mathbf{x}_{m,2}) \geq \begin{cases} 0; & m = m_1(\mathbf{y}_1), \dots, m_L(\mathbf{y}_1) \\ P_{e,w}(N_2, L + 1) & \text{other } m \end{cases} \quad (5.8.45)$$

Interchanging the order of summation between  $m$  and  $\mathbf{y}_1$  in (5.8.43) and substituting in (5.8.45), we have

$$P_e \geq \frac{1}{M} \sum_{\mathbf{y}_1} \sum_{\substack{m_l(\mathbf{y}_1) \\ l > L}} P(\mathbf{y}_1 \mid \mathbf{x}_{m,1}) P_{e,w}(N_2, L + 1) \quad (5.8.46)$$

Finally, we can consider the set of prefixes  $\{\mathbf{x}_{m,1}\}$  as a set of  $M$  code words of block length  $N$  and we can consider  $m_l(\mathbf{y}_1)$ ,  $l = 1, \dots, L$  as a list decoding rule for this set of code words. Thus

$$\frac{1}{M} \sum_{\mathbf{y}_1} \sum_{\substack{m_l(\mathbf{y}_1) \\ l > L}} P(\mathbf{y}_1 \mid \mathbf{x}_{m,1}) > P(M, N_1, L) \quad (5.8.47)$$

Combining (5.8.47) and (5.8.46) completes the proof. |

We now combine the previous four lemmas to derive the straight-line exponent of Theorem 8.6.2. Let  $\delta, \epsilon < \delta < \frac{1}{2}$  be an arbitrary number for the present and define

$$R_1 = \ln 2 - \mathcal{H}(\delta) \quad (5.8.48)$$

$$E_{sp}(R_1) = -\mathcal{H}(\delta) - \delta \ln \epsilon - (1 - \delta) \ln (1 - \epsilon) \quad (5.8.49)$$

For an  $(N, R)$  code with given  $N$  and  $R$ , define the number  $\lambda$  by

$$R = \lambda R_1 + \frac{3 \ln [2(N + 1)]}{2N} \quad (5.8.50)$$

We restrict our attention to rates in the range

$$\frac{3}{2N} \ln [2(N + 1)] \leq R \leq R_1 \quad (5.8.51)$$

so that  $0 \leq \lambda < 1$ . Now define  $N_1 = \lfloor \lambda N \rfloor$  and  $N_2 = N - N_1$ . Observe that

$N_2 \geq 1$ . Using (5.8.50), the number of code words,  $M = \lceil \exp NR \rceil$  satisfies

$$\frac{M}{N+1} \geq \sqrt{8(N+1)} \exp(N\lambda R_1) \quad (5.8.52)$$

$$\geq \sqrt{8(N_1+1)} \exp(N_1 R_1) \quad (5.8.53)$$

It follows, from Lemma 1, then, that

$$\begin{aligned} P_e(N_1, M, N+1) &\geq \frac{\epsilon}{(1-\epsilon)\sqrt{8(N_1+1)}} \exp[-N_1 E_{sp}(R_1)] \\ &\geq \frac{\epsilon}{(1-\epsilon)\sqrt{8(N+1)}} \exp[-\lambda N E_{sp}(R_1)] \end{aligned} \quad (5.8.54)$$

Also, from Lemma 3, we have

$$\begin{aligned} P_{e,w}(N_2, N+2) &\geq \frac{\sqrt{N_2}\epsilon}{(N_2+4)(1-\epsilon)} \exp[-N_2 E_{ex}(0)] \\ &\geq \frac{\epsilon}{(N+4)(1-\epsilon)} \exp\{-[(1-\lambda)N+1]E_{ex}(0)\} \\ &= \frac{\sqrt{2}\epsilon^{\frac{5}{4}}}{(N+4)(1-\epsilon)^{\frac{3}{4}}} \exp\{-(1-\lambda)NE_{ex}(0)\} \end{aligned} \quad (5.8.55)$$

Combining (5.8.54) and (5.8.55) with Lemma 4 yields

$$\begin{aligned} P_e(N, M) &\geq \frac{\epsilon^{\frac{9}{4}}}{2(1-\epsilon)^{\frac{7}{4}}\sqrt{N+1}(N+4)} \\ &\times \exp\{-N[\lambda E_{sp}(R_1) + (1-\lambda)E_{ex}(0)]\} \end{aligned} \quad (5.8.56)$$

Finally, using (5.8.50) to express  $\lambda$  in terms of  $R$  and bringing the coefficient inside the exponent, we get

$$P_e(N, M) \geq \exp\left\{-N\left[E_{ex}(0) - R\left[\frac{E_{ex}(0) - E_{sp}(R_1)}{R_1}\right] + o(N)\right]\right\} \quad (5.8.57)$$

where

$$o(N) = \frac{1}{N}\left[\frac{E_{ex}(0) - E_{sp}(R_1)}{R_1}\right]\frac{3}{2}\ln[2(N+1)] - \ln\frac{\epsilon^{\frac{9}{4}}}{2(1-\epsilon)^{\frac{7}{4}}\sqrt{N+1}(N+4)} \quad (5.8.58)$$

The exponent in (5.8.57) is a linear function of  $R$ , going from  $E_{ex}(0)$  at  $R=0$  to  $E_{sp}(R_1)$  at  $R=R_1$ . We clearly get the tightest exponential bound by choosing  $R_1$  (that is,  $\delta$ ) to minimize the above linear function. The following theorem summarizes our results.

**Theorem 5.8.4.** For a BSC with crossover probability  $\epsilon$ , let  $R_1$  be the  $R$  at which a straight line through the point  $R = 0, E = E_{ex}(0)$  is tangent to the curve  $E_{sp}(R)$ . Then for any  $N \geq 1$  and any  $R$  satisfying (5.8.51),  $P_e$  for every  $(N, R)$  code satisfies (5.8.57).

---

### Block-Error Probability at Rates Above Capacity

In this section we shall show that, for any discrete memoryless channel, and for any fixed rate above capacity,  $P_e$  approaches one with increasing  $N$ .\* As we pointed out in Section 4.3, such a result does not necessarily preclude the possibility of reliable data transmission at rates above capacity since a large probability of decoding a block in error does not imply a large probability of error on individual source digits. Furthermore, such a result says nothing about error probability for nonblock codes. On the other hand, such a result is conceptually simpler than the converse to the coding theorem (Theorem 4.3.4) since it is a result solely about the channel rather than about both a source and a channel, and such a result certainly provides additional insight into the nature of channel capacity.

**Theorem 5.8.5 (Wolfowitz (1957)).** For an arbitrary discrete memoryless channel of capacity  $C$  nats and any  $(N, R)$  code with  $R > C$ ,

$$P_e \geq 1 - \frac{4A}{N(R - C)^2} - \exp\left[-\frac{N(R - C)}{2}\right] \quad (5.8.59)$$

where  $A$  is a finite positive constant depending on the channel but not on  $N$  or  $M$ .

---

*Discussion.* It can be seen from (5.8.59) that, for any fixed  $R > C$ ,  $P_e$  must approach 1 as  $N$  becomes large. It can also be seen that, if we choose  $R = C + \delta/\sqrt{N}$  for some fixed  $\delta > \sqrt{8A} + 2$ , say, then  $P_e$  is bounded away from 0 for all  $N$ .

*Proof.* Let  $P(j|k)$ ,  $0 \leq j \leq J - 1$ ,  $0 \leq k \leq K - 1$  be the transition probabilities for the channel where  $K$  and  $J$  are the input and output alphabet sizes respectively. Let  $Q(0), \dots, Q(K - 1)$  be the input probabilities that

\* Historically this result, due to Wolfowitz, has been called the strong converse to the coding theorem and Theorem 4.3.1, due to Fano, has been called the weak converse to the coding theorem. Since Wolfowitz' result neither implies nor is implied by Theorem 4.3.4, which we have called the converse to the coding theorem, we shall refer to the results here as Wolfowitz' converse or the block coding converse to the coding theorem.

achieve capacity and  $\omega(j) = \sum_k Q(k)P(j|k)$ ,  $0 \leq j \leq J-1$ , be the associated output probabilities. From Theorem 4.5.1, we know that

$$I(k;Y) \triangleq \sum_{j=0}^{J-1} P(j|k) \ln \frac{P(j|k)}{\omega(j)} \leq C; \quad 0 \leq k \leq K-1 \quad (5.8.60)$$

Let

$$P_N(\mathbf{y}|\mathbf{x}) = \prod_{n=1}^N P(y_n|x_n)$$

where  $\mathbf{y} = (y_1, \dots, y_N)$  and  $\mathbf{x} = (x_1, \dots, x_N)$ , let

$$\omega_N(\mathbf{y}) = \prod_{n=1}^N \omega(y_n)$$

and define

$$I(\mathbf{x};\mathbf{y}) = \ln \frac{P_N(\mathbf{y}|\mathbf{x})}{\omega_N(\mathbf{y})} = \sum_{n=1}^N I(x_n;y_n) \quad (5.8.61)$$

where  $I(x_n;y_n) = \ln [P(y_n|x_n)/\omega(y_n)]$ .

Now consider an  $(N,R)$  code with code words  $\mathbf{x}_1, \dots, \mathbf{x}_M$  and decoding regions  $Y_1, \dots, Y_M$ . The probability of correct decoding for that code is given by

$$P_c = \frac{1}{M} \sum_{m=1}^M \sum_{\mathbf{y} \in Y_m} P_N(\mathbf{y}|\mathbf{x}_m) \quad (5.8.62)$$

Let  $\epsilon > 0$  be an arbitrary number and define, for  $1 \leq m \leq M$ ,

$$B_m = [\mathbf{y}: I(\mathbf{x}_m;\mathbf{y}) > N(C + \epsilon)] \quad (5.8.63)$$

Letting  $B_m^c$  be the complement of  $B_m$ , we can rewrite (5.8.62) as

$$P_c = \frac{1}{M} \sum_{m=1}^M \sum_{\mathbf{y} \in Y_m \cap B_m} P_N(\mathbf{y}|\mathbf{x}_m) + \frac{1}{M} \sum_{m=1}^M \sum_{\mathbf{y} \in Y_m \cap B_m^c} P_N(\mathbf{y}|\mathbf{x}_m) \quad (5.8.64)$$

For  $\mathbf{y} \in B_m^c$ , we have  $P_N(\mathbf{y}|\mathbf{x}_m)/\omega_N(\mathbf{y}) \leq \exp[N(C + \epsilon)]$ , and thus the second term in (5.8.64) is bounded by

$$\begin{aligned} \frac{1}{M} \sum_{m=1}^M \sum_{\mathbf{y} \in Y_m \cap B_m^c} P_N(\mathbf{y}|\mathbf{x}_m) &\leq \frac{1}{M} \sum_{m=1}^M \sum_{\mathbf{y} \in Y_m \cap B_m^c} \omega_N(\mathbf{y}) \exp[N(C + \epsilon)] \\ &\leq \frac{\exp[N(C + \epsilon)]}{M} \sum_{m=1}^M \sum_{\mathbf{y} \in Y_m} \omega_N(\mathbf{y}) \\ &\leq \frac{\exp[N(C + \epsilon)]}{M} \end{aligned} \quad (5.8.65)$$

where we have used the facts that the decoding regions are disjoint, and that  $\omega_N(\mathbf{y})$  is a probability assignment.

We can overbound the first term in (5.8.64) by summing, for each  $m$ , over  $\mathbf{y} \in B_m$  rather than  $\mathbf{y} \in Y_m \cap B_m$ . We have

$$\sum_{\mathbf{y} \in B_m} P_N(\mathbf{y} \mid \mathbf{x}_m) = P[I(\mathbf{x}_m; \mathbf{y}) > N(C + \epsilon) \mid \mathbf{x}_m] \quad (5.8.66)$$

where for a given  $\mathbf{x}_m$  we are interpreting  $I(\mathbf{x}_m; \mathbf{y})$  as a random variable taking on any particular value  $I(\mathbf{x}_m; \mathbf{y})$  with probability  $P_N(\mathbf{y} \mid \mathbf{x}_m)$ . From (5.8.61), this random variable is a sum over  $n$  of independent random variables,  $\sum I(x_{m,n}; y_n)$ , and from (5.8.60) the average value of this sum is, at most,  $NC$ . Thus, from the Chebyshev inequality, we have

$$\sum_{\mathbf{y} \in B_m} P_N(\mathbf{y} \mid \mathbf{x}_m) \leq \frac{\sum_{n=1}^N \text{VAR}[I(x_{m,n}; y_n) \mid x_{mn}]}{N^2 \epsilon^2} \quad (5.8.67)$$

where

$$\begin{aligned} \text{VAR}[I(x_{m,n}; y_n) \mid x_{m,n}] &= \sum_{j=0}^{J-1} P(j \mid x_{m,n}) \left[ \ln \frac{P(j \mid x_{m,n})}{\omega(j)} \right]^2 \\ &\quad - \left[ \sum_{j=0}^{J-1} P(j \mid x_{m,n}) \ln \frac{P(j \mid x_{m,n})}{\omega(j)} \right]^2 \end{aligned} \quad (5.8.68)$$

Since  $x_{m,n}$  is one of the input letters  $0, \dots, K-1$ , this variance is always upper bounded by a finite number  $A$  defined as

$$A = \max_{0 \leq k \leq K-1} \text{VAR}[I(k; y_n) \mid k] \quad (5.8.69)$$

Thus, for all  $m$ ,

$$\sum_{\mathbf{y} \in B_m} P_N(\mathbf{y} \mid \mathbf{x}_m) \leq \frac{A}{N \epsilon^2} \quad (5.8.70)$$

Substituting (5.8.65) and (5.8.70) into (5.8.64) yields

$$P_c \leq \frac{A}{N \epsilon^2} + \frac{\exp[N(C + \epsilon)]}{M}$$

Since this is valid for all  $(N, M)$  codes,

$$P_e(N, M) \geq 1 - \frac{A}{N \epsilon^2} - \frac{\exp[N(C + \epsilon)]}{M}$$

Finally, for a given  $R > C$ , choosing  $\epsilon = (R - C)/2$  and using  $M = [e^{NR}]$ , we obtain (5.8.59), completing the proof. |

Some extensions of this theorem are treated in Problems 5.34 to 5.36. In particular, by using the Chernoff bound rather than the Chebyshev inequality

in (5.8.67), it can be shown that, for fixed  $R > C$ ,  $P_e$  approaches 1 exponentially with increasing  $N$ . Likewise, by replacing the Chebyshev inequality with the central limit theorem, we obtain stronger results for  $R$  close to  $C$ .

### 5.9 The Coding Theorem for Finite-State Channels

In Section 4.6 we described a finite-state channel by a conditional probability measure  $P(y_n, s_n | x_n, s_{n-1})$ . This probability measure determines the probability  $P_N(\mathbf{y} | \mathbf{x}, s_0)$  of any output sequence  $\mathbf{y} = (y_1, \dots, y_N)$  conditional on an input sequence  $\mathbf{x} = (x_1, \dots, x_N)$  and an initial state  $s_0$ . Theorem 5.6.1, on the other hand, provides us with a coding theorem which is valid for any channel probability assignment  $P_N(\mathbf{y} | \mathbf{x})$  (that is, Theorem 5.6.1 is not restricted to memoryless channels). The only difficulty in applying that result directly here lies in the problem of what to do about the initial state. Our objective here will be to prove a coding theorem that is applicable independent of the starting state. Our major result will be that, for any code rate  $R$  and any  $\epsilon > 0$ , there exist codes of sufficiently long block length such that, independent of the message and initial state,  $P_e < \exp\{-N[E_r(R) - \epsilon]\}$  where  $E_r(R)$  is positive for  $R < \underline{C}$ . We saw, in Section 4.6, that  $P_e$  cannot be made arbitrarily small, independent of the initial state, for  $R > \underline{C}$ .

For nonindecomposable channels, the error probability that can be achieved through coding, especially at rates between  $\underline{C}$  and  $\bar{C}$ , generally depends strongly both on the initial state and on the transmitter's knowledge of the initial state. We shall not treat this latter problem in any detail since it can usually be handled by a slight change in the model. For example, if the "panic-button" channel of Figure 4.6.5 has an initial state  $s_0 = 0$ , we can simply omit the panic button (input 2) from the channel and treat the channel as a noiseless binary channel. Likewise, if the initial state is known for the alternating phase channel of Figure 4.6.3, it can be remodelled as a pair of parallel memoryless channels.

In proving a coding theorem independent of the initial state, we have a problem that is formally the same as that of a compound channel. A compound channel is a channel described by a set of different transition probability assignments, say  $P_N^{(i)}(\mathbf{y} | \mathbf{x})$ , where the particular assignment,  $i$ , is unknown to transmitter and receiver. The object is to find a code that performs acceptably against all choices of  $i$ . Here we have a finite state channel with  $A$  states and thus there are  $A$  different transition probability assignments, one corresponding to each possible value of the starting state,  $s_0$ . Our approach (which only works well when  $A$  is finite) will be to simply *assume* for the time being that the initial states occur with equal probability. Under this assumption, we have

$$P_N(\mathbf{y} | \mathbf{x}) = \sum_{s_0} \frac{1}{A} P_N(\mathbf{y} | \mathbf{x}, s_0)$$

We can then apply Theorem 5.6.1, choosing  $M$  code words independently with a probability assignment  $Q_N(\mathbf{x})$ , obtaining

$$\bar{P}_{e,m} \leq (M-1)^\rho \sum_{\mathbf{y}} \left\{ \sum_{\mathbf{x}} Q_N(\mathbf{x}) \left[ \sum_{s_0} \frac{1}{A} P_N(\mathbf{y} \mid \mathbf{x}, s_0) \right]^{1/(1+\rho)} \right\}^{1+\rho} \quad (5.9.1)$$

for any  $\rho$ ,  $0 \leq \rho \leq 1$ . Clearly, we obtain the tightest bound in (5.9.1) by minimizing over the input probability assignment  $Q_N(\mathbf{x})$ .

By the same arguments as in the discussion following Theorem 5.6.2, the average error probability for at least one code in the ensemble satisfies the above bound and there is also a code with the given  $N$  and  $M$  for which  $P_{e,m}$ , for each  $m$ , is upper bounded by four times the right-hand side of (5.9.1). Finally, since the error probability for such a code is an average over the  $A$  equally likely states, the error probability, given any particular initial state, can be no more than  $A$  times the average. This then gives us a bound on error probability which is equally valid for every initial state and thus no longer depends on the assumption of equally likely states. The decoder assumed in the derivation of this bound decodes the message  $m$  which maximizes

$$\sum_{s_0} \frac{1}{A} P_N(\mathbf{y} \mid \mathbf{x}_m, s_0)$$

and is perfectly well defined whether or not a probability measure actually exists on the initial state. Combining the previous observations, we see that for any block length  $N$  and any number of code words  $M$ , a code exists such that the error probability for message  $m$  conditional on the initial state  $s_0$  is bounded, independent of  $m$  and  $s_0$ , by

$$P_{e,m}(s_0) \leq 4A(M-1)^\rho \min_{Q_N} \sum_{\mathbf{y}} \left\{ \sum_{\mathbf{x}} Q_N(\mathbf{x}) \left[ \sum_{s_0} \frac{1}{A} P_N(\mathbf{y} \mid \mathbf{x}, s_0) \right]^{1/(1+\rho)} \right\}^{1+\rho} \quad (5.9.2)$$

for all  $\rho$ ,  $0 \leq \rho \leq 1$ .

As a first step in simplifying this expression, it is convenient to remove the sum on  $s_0$  from the inside bracket in (5.9.2).

Using the inequality  $(\sum a_i)^r \leq \sum a_i^r$  for  $0 < r \leq 1$  (see Problem 4.15f), the right-hand side of (5.9.2) is upper bounded by

$$P_{e,m}(s_0) \leq 4A(M-1)^\rho \min_{Q_N} \sum_{\mathbf{y}} \left\{ \sum_{s_0} \sum_{\mathbf{x}} Q_N(\mathbf{x}) \left[ \frac{1}{A} P_N(\mathbf{y} \mid \mathbf{x}) \right]^{1/(1+\rho)} \right\}^{1+\rho} \quad (5.9.3)$$

Multiplying and dividing the sum on  $s_0$  by  $A$ , interpreting  $1/A$  as a probability

assignment on  $s_0$ , and using the inequality  $(\sum P_i a_i)^r \leq \sum P_i a_i^r$  for  $r \geq 1$  (see Problem 4.15d),

$$P_{e,m}(s_0) \leq 4A(M-1)^\rho A^\rho \min_{\mathbf{Q}_N} \sum_{s_0} \sum_{\mathbf{y}} \left\{ \sum_{\mathbf{x}} Q_N(\mathbf{x}) \left[ \frac{1}{A} P_N(\mathbf{y} \mid \mathbf{x}, s_0) \right]^{1/(1+\rho)} \right\}^{1+\rho} \quad (5.9.4)$$

$$\leq 4A(M-1)^\rho A^\rho \min_{\mathbf{Q}_N} \max_{s_0} \sum_{\mathbf{y}} \left\{ \sum_{\mathbf{x}} Q_N(\mathbf{x}) P_N(\mathbf{y} \mid \mathbf{x}, s_0)^{1/(1+\rho)} \right\}^{1+\rho} \quad (5.9.5)$$

where we have brought out the innermost  $1/A$  term and bounded the sum over  $s_0$  by  $A$  times the maximum term.

The order of the min-max in (5.9.5) is important. It is not hard to see (Problem 5.37) that if the transmitter knows the initial state and can use a different code for each initial state, then the min-max in (5.9.5) can be replaced by a max-min, typically decreasing the bound.

**Theorem 5.9.1.** For an arbitrary finite-state channel with  $A$  states, for any positive integer  $N$  and any positive  $R$ , there exists an  $(N,R)$  code for which, for all messages  $m$ ,  $1 \leq m \leq M = [e^{NR}]$ , all initial states, and all  $\rho$ ,  $0 \leq \rho \leq 1$ ,

$$P_{e,m}(s_0) \leq 4A \exp \{-N[-\rho R + F_N(\rho)]\} \quad (5.9.6)$$

where

$$F_N(\rho) = -\frac{\rho \ln A}{N} + \max_{\mathbf{Q}_N} \left[ \min_{s_0} E_{o,N}(\rho, \mathbf{Q}_N, s_0) \right] \quad (5.9.7)$$

$$E_{o,N}(\rho, \mathbf{Q}_N, s_0) = -\frac{1}{N} \ln \sum_{\mathbf{y}} \left\{ \sum_{\mathbf{x}} Q_N(\mathbf{x}) P_N(\mathbf{y} \mid \mathbf{x}, s_0)^{1/(1+\rho)} \right\}^{1+\rho} \quad (5.9.8)$$

*Proof.* Substituting (5.9.7) and (5.9.8) into (5.9.6), we see that (5.9.6) is the same as (5.9.5), with  $(M-1)$  upper bounded by  $e^{NR}$ . |

Although we do not need the result in what follows, it should be clear that this theorem is equally applicable to any compound channel with  $A$  states.

Equation 5.9.6 has the appearance of an exponential bound, and we now substantiate that  $F_N(\rho)$  approaches a constant as  $N \rightarrow \infty$ . In so doing, it will become clear why it was convenient to include the term  $-\rho(\ln A)/N$  in the definition of  $F_N(\rho)$ .

**LEMMA 5.9.1.** For any given finite-state channel,  $F_N(\rho)$ , as given by (5.9.7), satisfies

$$F_N(\rho) \geq \frac{n}{N} F_n(\rho) + \frac{l}{N} F_l(\rho) \quad (5.9.9)$$

for all positive integers  $n$  and  $l$  with  $N = n + l$ .

---

*Proof.* Let the input sequence  $\mathbf{x} = (x_1, \dots, x_N)$  be represented by the subsequences  $\mathbf{x}_1 = (x_1, \dots, x_n)$  and  $\mathbf{x}_2 = (x_{n+1}, \dots, x_N)$  and let  $\mathbf{y} = (y_1, \dots, y_N)$  be represented by  $\mathbf{y}_1 = (y_1, \dots, y_n)$  and  $\mathbf{y}_2 = (y_{n+1}, \dots, y_N)$ . For a given  $\rho$ , let  $\mathbf{Q}_n$  and  $\mathbf{Q}_l$  be the probability assignments that maximize  $F_n(\rho)$  and  $F_l(\rho)$  respectively and consider the probability assignment on  $\mathbf{x} = (x_1, \dots, x_N)$  given by

$$Q_N(\mathbf{x}) = Q_n(\mathbf{x}_1)Q_l(\mathbf{x}_2) \quad (5.9.10)$$

Finally, let  $s_0'$  be the initial state that minimizes  $E_{o,N}(\rho, \mathbf{Q}_N, s_0')$ . Then

$$F_N(\rho) > \frac{-\rho \ln A}{N} + E_{o,N}(\rho, \mathbf{Q}_N, s_0'),$$

and we have

$$\begin{aligned} \exp[-NF_N(\rho)] &\leq A^\rho \sum_{\mathbf{y}} \left\{ \sum_{\mathbf{x}} Q_N(\mathbf{x}) P_N(\mathbf{y} \mid \mathbf{x}, s_0')^{1/(1+\rho)} \right\}^{1+\rho} \\ &= A^\rho \sum_{\mathbf{y}_1} \sum_{\mathbf{y}_2} \left\{ \sum_{\mathbf{x}_1} \sum_{\mathbf{x}_2} Q_n(\mathbf{x}_1) Q_l(\mathbf{x}_2) \left[ \sum_{s_n} P_n(\mathbf{y}_1, s_n \mid \mathbf{x}_1, s_0') \right. \right. \\ &\quad \times \left. \left. P_l(\mathbf{y}_2 \mid \mathbf{x}_2, s_n) \right]^{1/(1+\rho)} \right\}^{1+\rho} \quad (5.9.11) \end{aligned}$$

$$\begin{aligned} &\leq A^{2\rho} \sum_{s_n} \sum_{\mathbf{y}_1} \sum_{\mathbf{y}_2} \left\{ \sum_{\mathbf{x}_1} \sum_{\mathbf{x}_2} Q_n(\mathbf{x}_1) Q_l(\mathbf{x}_2) P_n(\mathbf{y}_1, s_n \mid \mathbf{x}_1, s_0')^{1/(1+\rho)} \right. \\ &\quad \times \left. P_l(\mathbf{y}_2 \mid \mathbf{x}_2, s_n)^{1/(1+\rho)} \right\}^{1+\rho} \quad (5.9.12) \end{aligned}$$

In going from (5.9.11) to (5.9.12), we have used the same inequalities on the summation over  $s_n$  that we used in going from (5.9.2) to (5.9.4). Rearranging the sums [as in (5.5.7) to (5.5.9)], we obtain

$$\begin{aligned} \exp[-NF_N(\rho)] &\leq A^{2\rho} \sum_{s_n} \left\{ \sum_{\mathbf{y}_1} \left[ \sum_{\mathbf{x}_1} Q_n(\mathbf{x}_1) P_n(\mathbf{y}_1, s_n \mid \mathbf{x}_1, s_0')^{1/(1+\rho)} \right]^{1+\rho} \right\} \\ &\quad \times \left\{ \sum_{\mathbf{y}_2} \left[ \sum_{\mathbf{x}_2} Q_l(\mathbf{x}_2) P_l(\mathbf{y}_2 \mid \mathbf{x}_2, s_n)^{1/(1+\rho)} \right]^{1+\rho} \right\} \quad (5.9.13) \end{aligned}$$

$$\begin{aligned} &\leq A^\rho \sum_{s_n} \sum_{\mathbf{y}_1} \left[ \sum_{\mathbf{x}_1} Q_n(\mathbf{x}_1) P_n(\mathbf{y}_1, s_n \mid \mathbf{x}_1, s_0')^{1/(1+\rho)} \right]^{1+\rho} \\ &\quad \times \exp[-lF_l(\rho)] \quad (5.9.14) \end{aligned}$$

where we have upper bounded the final term in (5.9.13) by maximizing it

over  $s_n$ . Finally, using Minkowski's inequality (Problem 4.15h) to interchange the summation on  $s_n$  and  $\mathbf{x}_1$ , we have

$$\begin{aligned} \exp [-NF_N(\rho)] &\leq A^\rho \sum_{\mathbf{y}_1} \left\{ \sum_{\mathbf{x}_1} Q_n(\mathbf{x}_1) \left[ \sum_{s_n} P_n(\mathbf{y}_1, s_n \mid \mathbf{x}_1, s_0') \right]^{1/(1+\rho)} \right\}^{1+\rho} \\ &\quad \times \exp [-lF_l(\rho)] \\ &\leq \exp [-nF_n(\rho) - lF_l(\rho)] \end{aligned} \quad (5.9.15)$$

where we have summed over  $s_n$  and then maximized over the initial state  $s_0'$ . Rearranging (5.9.15), we have (5.9.9), completing the proof. |

LEMMA 5.9.2. Let

$$F_\infty(\rho) = \sup_N F_N(\rho)$$

Then

$$\lim_{N \rightarrow \infty} F_N(\rho) = F_\infty(\rho) \quad (5.9.16)$$

For  $0 \leq \rho \leq 1$ , the convergence is uniform in  $\rho$  and  $F_\infty(\rho)$  is uniformly continuous.

---

*Proof.* From Theorem 5.6.3, using  $Q_N(\mathbf{x})$  in place of  $Q(k)$  and  $P_N(\mathbf{y} \mid \mathbf{x}, s_0)$  in place of  $P(j \mid k)$ , it follows that

$$0 \leq \frac{\partial NE_o(\rho, \mathbf{Q}_N, s_0)}{\partial \rho} \leq \mathcal{J}(\mathbf{Q}_N; \mathbf{P}_N) \quad (5.9.17)$$

For an input alphabet of  $K$  letters, there are  $K^N$  input sequences of length  $N$ , and (5.9.17) can be further bounded by

$$0 \leq \frac{\partial E_o(\rho, \mathbf{Q}_N, s_0)}{\partial \rho} \leq \log K \quad (5.9.18)$$

From (5.9.7), it then follows that, for any  $0 \leq \rho_1 < \rho_2 \leq 1$ ,

$$\frac{-(\rho_2 - \rho_1) \ln A}{N} \leq F_N(\rho_2) - F_N(\rho_1) \leq (\rho_2 - \rho_1) \log K \quad (5.9.19)$$

One consequence of this is that, for each  $\rho$ ,  $0 \leq \rho \leq 1$ ,  $F_N(\rho)$  is bounded independent of  $N$ . Thus combining Lemma 5.9.1 with Lemma 4A.2, we have (5.9.16). The uniform convergence and uniform continuity follow from the bounded slope of  $F_N(\rho)$  for each  $N$  exhibited in (5.9.19). |

**Theorem 5.9.2.** For any given finite-state channel, let

$$E_r(R) = \max_{0 \leq \rho \leq 1} [F_\infty(\rho) - \rho R] \quad (5.9.20)$$

Then for any  $\epsilon > 0$ , there exists  $N(\epsilon)$  such that for each  $N \geq N(\epsilon)$  and each  $R \geq 0$  there exists an  $(N, R)$  code such that, for all  $m$ ,  $1 \leq m \leq M = \lceil e^{NR} \rceil$ , and all initial states

$$P_{e,m}(s_0) \leq \exp \{-N[E_r(R) - \epsilon]\} \quad (5.9.21)$$

Furthermore, for  $0 \leq R < \underline{C}$ ,  $E_r(R)$  is strictly positive, strictly decreasing in  $R$ , and convex  $\cup$ .

*Discussion.* This theorem establishes an exponential bound on error probability for all  $R < \underline{C}$ , when  $\underline{C}$  is defined in (4.6.3) and (4.6.4). It is believed that  $E_r(R)$  is the reliability of the channel for  $R$  close to  $\underline{C}$  but proof of this only exists in special cases. The theorem is somewhat weaker than the corresponding theorem for discrete memoryless channels since (5.9.21) is only valid for  $N \geq N(\epsilon)$  and little is known about the way that  $N(\epsilon)$  depends upon the channel. Finally, we know very little about how to calculate  $F_\infty(\rho)$  [and thus  $E_r(R)$ ] except in some special cases, the most important of which will be subsequently treated. The function  $F_\infty(\rho)$  can always be lower bounded, however, by

$$F_\infty(\rho) \geq \frac{-\rho \ln A}{N} + \min_{s_0} E_{o,N}(\rho, \mathbf{Q}_N, s_0) \quad (5.9.22)$$

for any  $N$  and  $\mathbf{Q}_N$ .

*Proof.* For any  $N, R$ , we can rewrite (5.9.6) as

$$P_{e,m}(s_0) \leq \exp \left\{ -N \left[ -\rho R + F_N(\rho) - \frac{\ln 4A}{N} \right] \right\} \quad (5.9.23)$$

For any  $\epsilon > 0$ , Lemma 5.9.2 asserts that we can choose  $N(\epsilon)$  such that, for  $N \geq N(\epsilon)$ ,

$$F_\infty(\rho) - F_N(\rho) + \frac{\ln 4A}{N} \leq \epsilon; \quad 0 \leq \rho \leq 1 \quad (5.9.24)$$

Substituting (5.9.24) into (5.9.23) yields

$$P_{e,m}(s_0) \leq \exp \{-N[-\rho R + F_\infty(\rho) - \epsilon]\}; \quad 0 \leq \rho \leq 1 \quad (5.9.25)$$

For the  $\rho$  that maximizes  $-\rho R + F_\infty(\rho)$ , (5.9.25) reduces to (5.9.21). Now we assume an  $R < \underline{C}$  and show that  $E_r(R) > 0$ . Define  $\delta$  by  $\underline{C} - R$ . From Theorem 4.6.1, we can choose an  $N$  large enough so that

$$R + \frac{\ln A}{N} < \underline{C}_N - \frac{\delta}{2} \quad (5.9.26)$$

For this  $N$ , let  $\mathbf{Q}_N$  be an input assignment that achieves  $\underline{C}_N$ . Then, from Theorem 5.6.3, we have

$$\left. \frac{\partial E_{o,N}(\rho, \mathbf{Q}_N, s_0)}{\partial \rho} \right|_{\rho=0} \geq \underline{C}_N; \quad \text{all } s_0 \quad (5.9.27)$$

Since  $\partial E_{o,N}(\rho, \mathbf{Q}_N, s_0)/\partial \rho$  is a continuous function of  $\rho$ , there is a range of  $\rho > 0$  for each  $s_0$  for which

$$E_{o,N}(\rho, \mathbf{Q}_N, s_0) - \rho \left( R + \frac{\ln A}{N} \right) > 0 \quad (5.9.28)$$

Since there are a finite number of initial states  $s_0$ , we can choose a sufficiently small  $\rho^* > 0$  so that

$$E_{o,N}(\rho^*, \mathbf{Q}_N, s_0) - \rho^* \left( R + \frac{\ln A}{N} \right) > 0; \quad \text{all } s_0 \quad (5.9.29)$$

But since

$$F_\infty(\rho^*) > F_N(\rho^*) > E_{o,N}(\rho^*, \mathbf{Q}_N, s_0) - \frac{\rho^* \ln A}{N}$$

for all  $s_0$ , this implies that

$$F_\infty(\rho^*) - \rho^* R > 0 \quad (5.9.30)$$

and, thus, that  $E_r(R) > 0$  for all  $R < C$ . The convexity of  $E_r(R)$  is established by observing that  $E_r(R)$  is the least upper bound of a set of straight lines,  $F_\infty(\rho) - \rho R$ . Finally, since  $F_\infty(\rho) - \rho R$  is zero for  $\rho = 0$ , we observe that for  $R < \underline{C}$ ,  $F_\infty(\rho) - \rho R$  is maximized over  $0 \leq \rho \leq 1$  by some  $\rho > 0$ . Any decrease in  $R$  for that fixed  $\rho$  will increase  $F_\infty(\rho) - \rho R$  and consequently  $E_r(R)$  is strictly increasing with decreasing  $R$  for  $R < \underline{C}$ . |

### **State Known at Receiver**

We now consider a special case of the preceding results in which the state at time  $n$  is a deterministic function of the output at time  $n$  and the state at time  $n - 1$ , that is,  $s_n = g(y_n, s_{n-1})$ . In such a situation, the receiver is able to track the channel state if it has ever known the state in the past. An example of such a channel is given in Figure 5.9.1, and in this example,  $s_n$

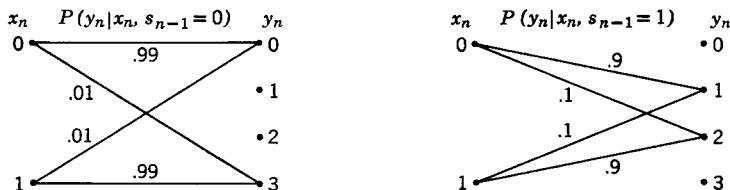
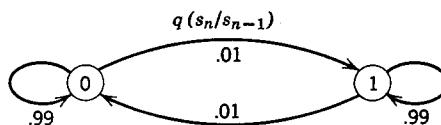


Figure 5.9.1. Simple model of fading channel.

is a function only of  $y_n$ . In each state, the channel is a BSC, the outputs 0 and 1 corresponding to the input 0 and the outputs 2 and 3 corresponding to the input 1. Outputs 0 and 3 correspond to state 0 and outputs 1 and 2 correspond to state 1. It can be seen that, aside from the numbers, this is the same type of channel model as in Figure 4.6.1 except that here the output alphabet has been expanded to incorporate the assumption of known state. The model is a simple (and rather crude) approximation to a fading channel with a binary signalling alphabet and a receiver that not only guesses at the input but also measures the level of the received signal.

To analyze this class of channels, we observe that an output sequence,  $\mathbf{y} = (y_1, \dots, y_N)$  and an initial state  $s_0$  uniquely determine a state sequence  $\mathbf{s} = (s_1, \dots, s_N)$ , which we denote as  $\mathbf{s}(\mathbf{y}, s_0)$ . We then have

$$P_N(\mathbf{y}, \mathbf{s} \mid \mathbf{x}, s_0) = \begin{cases} P_N(\mathbf{y} \mid \mathbf{x}, s_0) & \text{for } \mathbf{s} = \mathbf{s}(\mathbf{y}, s_0) \\ 0 & \text{otherwise} \end{cases} \quad (5.9.31)$$

$$E_{o,N}(\rho, \mathbf{Q}_N, s_0) = -\frac{1}{N} \ln \sum_{\mathbf{y}} \sum_{\mathbf{s}} \left\{ \sum_{\mathbf{x}} \mathbf{Q}_N(\mathbf{x}) P_N(\mathbf{y}, \mathbf{s} \mid \mathbf{x}, s_0)^{(1/(1+\rho))} \right\}^{1+\rho} \quad (5.9.32)$$

To verify (5.9.32), we observe that for each  $\mathbf{y}$  the sum over  $\mathbf{s}$  is nonzero only for  $\mathbf{s} = \mathbf{s}(\mathbf{y}, s_0)$ , and for that  $\mathbf{s}$ ,  $P_N(\mathbf{y}, \mathbf{s} \mid \mathbf{x}, s_0) = P_N(\mathbf{y} \mid \mathbf{x}, s_0)$ . Thus (5.9.32) is equivalent to the definition of  $E_{o,N}(\rho, \mathbf{Q}_N, s_0)$  in (5.9.8).

Now suppose  $\mathbf{Q}_N(\mathbf{x})$  is a product measure (that is, successive letters in the ensemble of code words are independently chosen) with

$$\mathbf{Q}_N(\mathbf{x}) = \prod_{n=1}^N Q(x_n) \quad (5.9.33)$$

Thus (5.9.32) becomes

$$E_{o,N}(\rho, \mathbf{Q}_N, s_0) = -\frac{1}{N} \ln \sum_{\mathbf{s}} \sum_{\mathbf{y}} \left\{ \sum_{\mathbf{x}} \prod_{n=1}^N Q(x_n) P(y_n, s_n \mid x_n, s_{n-1})^{1/(1+\rho)} \right\}^{1+\rho} \quad (5.9.34)$$

$$= -\frac{1}{N} \ln \sum_{\mathbf{s}} \prod_{n=1}^N \sum_{j=0}^{J-1} \left\{ \sum_{k=0}^{K-1} Q(k) P(j, s_n \mid k, s_{n-1})^{1/(1+\rho)} \right\}^{1+\rho} \quad (5.9.35)$$

where the interchange of product and sum is the same as in (5.5.6) to (5.5.10). If, for a given  $\rho$  and  $\mathbf{Q}$ , we define

$$\alpha(s_{n-1}, s_n) = \sum_{j=0}^{J-1} \left\{ \sum_{k=0}^{K-1} Q(k) P(j, s_n \mid k, s_{n-1})^{1/(1+\rho)} \right\}^{1+\rho} \quad (5.9.36)$$

we then have

$$E_{o,N}(\rho, \mathbf{Q}_N, s_0) = -\frac{1}{N} \ln \sum_{\mathbf{s}} \prod_{n=1}^N \alpha(s_{n-1}, s_n) \quad (5.9.37)$$

Now define the  $A$  by  $A$  matrix:

$$[\alpha] = \left\{ \begin{array}{cccc} \alpha(0,0) & \cdots & \alpha(0, A-1) \\ \cdot & & \cdot \\ \cdot & & \cdot \\ \cdot & & \cdot \\ \alpha(A-1, 0) & \cdots & \alpha(A-1, A-1) \end{array} \right\} \quad (5.9.38)$$

Also let  $1]$  be an  $A$ -dimensional column vector of all 1's and let  $e(s_0)$  be an  $A$ -dimensional unit row vector with a 1 in the position corresponding to  $s_0$  (that is, in the first position for  $s_0 = 0$ , and in the  $i$ th position for  $s_0 = i - 1$ ). With a little thought, it can be seen that (5.9.37) can be rewritten as

$$E_{o,N}(\rho, \mathbf{Q}_N, s_0) = -\frac{1}{N} \ln e(s_0)[\alpha]^N 1] \quad (5.9.39)$$

A square matrix  $[\alpha]$  is said to be *irreducible* if it is impossible, by a simultaneous permutation of corresponding rows and columns (that is, a re-numbering of the states in this case), to put it in the form

$$\left[ \begin{array}{c|c} \alpha_1 & 0 \\ \hline \cdots & \cdots \\ \alpha_2 & \alpha_3 \end{array} \right]$$

where  $\alpha_1$  and  $\alpha_3$  are square matrices. In this case it can be seen that  $[\alpha]$  is irreducible iff it is possible, with nonzero probability, to reach each state from each other state in a finite number of steps using the input measure  $Q(k)$ . If  $[\alpha]$  is irreducible, we can apply Frobenius' theorem,\* which states that an irreducible nonzero matrix with nonnegative components has a largest real positive eigenvalue  $\lambda$  with a right eigenvector  $v]$  and a left eigenvector  $u$  both with all components positive. That is, interpreting  $v]$  as a column vector,

$$[\alpha] v] = \lambda v] \quad (5.9.40)$$

LEMMA 5.9.3. Let  $\lambda$  be the largest eigenvalue of the irreducible matrix  $[\alpha]$  and let  $v_{\max}$  and  $v_{\min}$  be the largest and smallest component of the positive right eigenvector  $v]$  corresponding to  $\lambda$ .

Then for any  $s_0$

$$\frac{v_{\min}}{v_{\max}} \lambda^N \leq e(s_0)[\alpha]^N 1] < \frac{v_{\max}}{v_{\min}} \lambda^N \quad (5.9.41)$$

*Proof.* From (5.9.40), we have

$$\begin{aligned} [\alpha]^N v] &= [\alpha]^{N-1} [\alpha] v] = \lambda [\alpha]^{N-1} v] \\ &= \lambda^2 [\alpha]^{N-2} v] = \cdots = \lambda^N v] \end{aligned} \quad (5.9.42)$$

\* See, for example, Gantmacher (1959).

Now since  $[\alpha]$  has nonnegative components, the row vector  $\underline{e(s_0)}[\alpha]^N$  has nonnegative components and, thus,  $\underline{e(s_0)}[\alpha]^N \underline{v}$  is upper bounded by upper bounding the components of  $\underline{v}$ , in this case by  $(1/v_{\min})v$ . Thus

$$\begin{aligned} \underline{e(s_0)}[\alpha]^N \underline{v} &\leq \frac{1}{v_{\min}} \underline{e(s_0)}[\alpha]^N v = \frac{\lambda^N}{v_{\min}} \underline{e(s_0)} v \\ &\leq \frac{v_{\max}}{v_{\min}} \lambda^N \end{aligned} \quad (5.9.43)$$

Similarly, we complete the proof by

$$\underline{e(s_0)}[\alpha]^N \underline{v} \geq \frac{1}{v_{\max}} \underline{e(s_0)}[\alpha]^N v \geq \frac{v_{\min}}{v_{\max}} \lambda^N |$$

Substituting (5.9.41) into (5.9.39), and using  $\lambda(\rho, \mathbf{Q})$  to make explicit the dependence of  $\lambda$  on  $\rho$  and  $\mathbf{Q}$ , we have, independent of  $s_0$ ,

$$|E_{o,N}(\rho, \mathbf{Q}_N, s_0) + \ln \lambda(\rho, \mathbf{Q})| \leq \frac{1}{N} \ln \frac{v_{\max}}{v_{\min}} \quad (5.9.44)$$

We have thus proved the following theorem.

**Theorem 5.9.3.** For a finite state channel with  $s_n = g(y_n, s_{n-1})$ , assume that  $\mathbf{Q}$  is a probability assignment such that, when the inputs are independently selected with probability assignment  $\mathbf{Q}$ , each state can be reached with non-zero probability from each other state within a finite number of time units. Then for any  $R > 0$  and any positive integer  $N$ , there exist  $(N, R)$  codes such that, for each message,  $1 \leq m \leq [e^{NR}]$ , each  $s_0$  and all  $\rho$ ,  $0 \leq \rho \leq 1$ ,

$$P_{e,m}(s_0) \leq 4A \frac{v_{\max}}{v_{\min}} \exp \{-N[-\ln \lambda(\rho, \mathbf{Q}) - \rho R]\} \quad (5.9.45)$$

where  $\lambda(\rho, \mathbf{Q})$  is the largest eigenvalue of  $[\alpha]$  as given by (5.9.38) and (5.9.36), and  $v_{\max}$  and  $v_{\min}$  are the extreme components of the positive right eigenvector corresponding to  $\lambda(\rho, \mathbf{Q})$ .

Equation 5.9.45 of course can be optimized over  $\mathbf{Q}$  and  $\rho$  to give the tightest bound. Unfortunately, in general, the bound is weaker than that given by Theorem 5.9.2 since here we are restricted to random-coding ensembles with the successive letters of the code words independently chosen. There is an important class of channels, however, in which the independent letter choice optimizes the bound, and this is the case where the choice of  $\mathbf{Q}$  that minimizes  $\alpha(s_{n-1}, s_n)$  in (5.9.36) is independent of the values

of  $s_{n-1}$  and  $s_n$ . To see this, we can consider a fixed  $s_0$  and a fixed  $\mathbf{s} = (s_1, \dots, s_N)$  and consider minimizing

$$\sum_{\mathbf{y}} \left\{ \sum_{\mathbf{x}} Q_N(\mathbf{x}) \prod_{n=1}^N P(y_n, s_n | x_n, s_{n-1})^{1/(1+\rho)} \right\}^{1+\rho}$$

over  $Q_N(\mathbf{x})$ . By the same argument that we used for parallel channels in Example 4 of Section 4.6, this minimum is achieved by a product distribution

$$Q_N(\mathbf{x}) = \prod_{n=1}^N Q^{(n)}(x_n)$$

where for each  $n$ ,  $Q^{(n)}(x_n)$  is chosen to minimize

$$\sum_{y_n} \left\{ \sum_{x_n} Q^{(n)}(x_n) P(y_n, s_n | x_n, s_{n-1})^{1/(1+\rho)} \right\}^{1+\rho}$$

If the same  $\mathbf{Q}$  minimizes  $\alpha(s_{n-1}, s_n)$  for all  $s_{n-1}, s_n$ , then  $Q^{(n)}$  is independent of  $n$  and also independent of  $\mathbf{s}$  and  $s_0$ . Thus

$$Q_N(\mathbf{x}) = \prod_n Q(x_n)$$

minimizes the above expression for all  $\mathbf{s}, s_0$  and thus maximizes  $E_o(\rho, \mathbf{Q}_N, s_0)$  for all  $s_0$ . In this case,  $F_\infty(\rho)$ , as given by (5.9.16) is equal to  $-\ln \lambda(\rho, \mathbf{Q})$  for this minimizing  $\mathbf{Q}$ . Although it is a nontrivial problem to calculate  $\lambda(\rho, \mathbf{Q})$  for the minimizing  $\mathbf{Q}$ , it is at least a problem that is independent of block length.

The example of Figure 5.9.1 belongs to the above class, and it is almost obvious that the inputs should be used independently and equiprobably in the ensemble of codes. In Figure 5.9.2,  $E_r(R)$  is sketched for this channel.

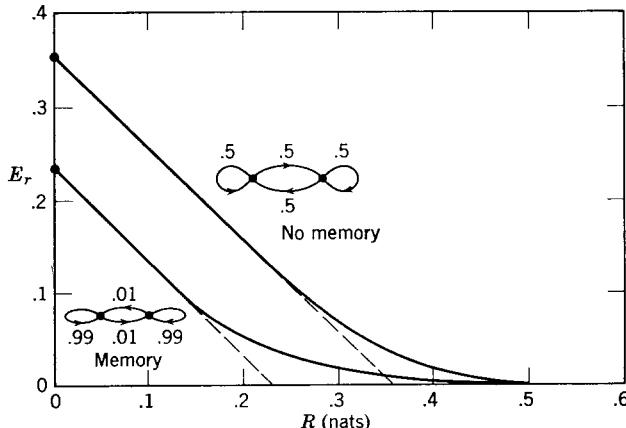


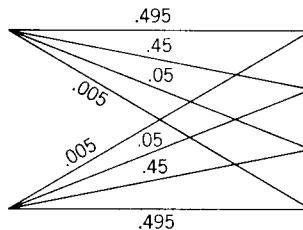
Figure 5.9.2. Two-state binary channel; known state at receiver.

As a comparison,  $E_r(R)$  is also sketched for the memoryless channel of Figure 5.9.3. This channel is equivalent to Figure 5.9.1 with  $q(s_n | s_{n-1})$  modified to be  $\frac{1}{2}$  for all  $s_{n-1}$  and  $s_n$ , or in more intuitive language, it is Figure 5.9.1 with the memory removed. Notice that the capacity  $C$  is unchanged by removing the memory (see Problem 5.39), but that the exponent is increased. Qualitatively, this can be explained by observing that the average time spent in each state is unchanged by removing the memory, but the probability of being in the bad state (state 1) a significantly larger-than-average amount of time is materially reduced by removing the memory. For example, in Figure 5.9.3, for  $N = 100$ , the probability of being in the bad state for the whole block is about  $1/(2e)$  with the memory and  $2^{-100}$  without the memory.

For channels in which the state is unknown at the receiver, there is another qualitative effect of long-persistence memory. The channel outputs plus a knowledge of the code allow the receiver to estimate the state. This increases the capacity of the channel over that without the memory (see Problem 5.38).

### Summary and Conclusions

The major result of this chapter was the noisy-channel coding theorem. We initially studied the simple hypothesis-testing problem where one of two code words is transmitted over a discrete memoryless channel and then proceeded to study the case of many code words. For many code words, we found that the major problem was in not knowing how to choose the code words. We handled this problem by finding an upper bound to the average error probability over an ensemble of codes and then pointing out that at least one code in the ensemble must have an error probability as small as the average. By investigating this upper bound, we found that, for any rate  $R$  less than the capacity of the channel, there exist codes of each block length  $N$  for which  $P_e \leq \exp[-NE_r(R)]$ . The rate  $R$  here was interpreted as  $\ln 2$  times the number of binary digits entering the encoder per transmitted channel digit. We also found a tighter bound on the error probability for small  $R$ . Next, we found that there was a lower bound on the error probability of the best codes for a given  $N$  and  $R$  for which the error probability decreases exponentially with  $N$  and we found that the exponents agree at rates close to capacity and in the limit as  $R \rightarrow 0$ . These lower bounds were derived only for the case of a binary symmetric channel. Finally, we found that the same type of exponential upper bound on error probability applies to finite-state



**Figure 5.9.3. Channel of Figure 5.9.1 with memory removed.**

channels. None of the results here give any direct indication of how to instrument encoders and decoders; that is the subject of the next chapter. In Chapters 7 and 8 we shall extend the results here to nondiscrete channels.

### **Historical Notes and References**

The noisy-channel coding theorem is due to Shannon (1948) and is undoubtedly the most significant result in information theory. The first rigorous proof of the theorem, for discrete memoryless channels, was given by Feinstein (1954) and was shortly followed by simpler proofs from Shannon (1957) and Wolfowitz (1957). The first demonstration that  $P_e$  approaches zero exponentially in  $N$  for fixed  $R < C$  was given by Feinstein (1955). The random-coding bound, the sphere-packing bound, and the observation that they are exponentially the same for rates close to capacity was first carried out by Elias (1955) for the special cases of the binary symmetric channel and the binary erasure channel. Fano (1961) used Shannon's random-coding and moment-generating function techniques to derive the random-coding exponent  $E_r(R)$  and to derive heuristically the sphere-packing bound for the general discrete memoryless channel. The expurgated random-coding bound and most of the properties of the random-coding exponent  $E_r(R)$  are due to Gallager (1965). The lower bounds on error probability in Section 5.8 for the discrete memoryless channel are due to Shannon, Gallager, and Berlekamp (1967). The coding theorem for finite state channels was first proved under somewhat more restrictive conditions by Gallager (1958), and, in stronger form, by Blackwell, Breiman, and Thomasian (1958). The random-coding exponent for finite-state channels and the development in Section 5.9 is due to Yudkin (1967). Theorem 5.9.4 is due to Gallager (1964). The only sphere-packing bound yet established for finite-state channels is for a class of binary channels treated by Kennedy (1963).

## **APPENDIX 5A**

Let

$$w = \sum_{n=1}^N z_n$$

be a sum of  $N$  discrete, independent, identically distributed random variables. The semi-invariant moment-generating function of each of the  $z_n$  is given in terms of the probability assignment  $P_z(z_n)$  by

$$\mu(s) = \ln g(s) = \ln \sum_z P_z(z) e^{sz} \quad (5A.1)$$

We assume that  $\mu(s)$  exists over an open interval of real values of  $s$  around  $s = 0$ . If the sample values for the  $z_n$  are bounded, this condition is clearly met. The first two derivatives of  $\mu(s)$  are given by

$$\mu'(s) = \frac{\sum_z z P_z(z) e^{sz}}{\sum_z P_z(z) e^{sz}} \quad (5A.2)$$

$$\mu''(s) = \frac{\sum_z z^2 P_z(z) e^{sz}}{\sum_z P_z(z) e^{sz}} - [\mu'(s)]^2 \quad (5A.3)$$

Notice that  $\mu'(0)$  and  $\mu''(0)$  are, respectively, the mean and variance of each of the  $z_n$ .

Let  $\mu_w(s)$  be the semi-invariant moment-generating function of the sum  $w$ .

$$\mu_w(s) = \ln g_w(s) = \ln \sum_w P_w(w) e^{sw} \quad (5A.4)$$

From (5.4.19), we have

$$\mu_w(s) = N\mu(s) \quad (5A.5)$$

In order to estimate  $\Pr(w \geq A)$ , where  $A \gg \bar{w}$ , we shall define a new sum of random variables, called tilted variables, whose probability assignments are related to  $P_z$ , but for which the mean of the sum is equal to  $A$ . We shall then apply the central-limit theorem to the sum of the tilted variables.

For any given  $s$  in the open interval where  $\mu(s)$  exists, define the tilted variables  $z_{n,s}$  as taking on the same values as the  $z_n$ , but with the probability assignment

$$Q_{z,s}(z) = \frac{P_z(z) e^{sz}}{\sum_z P_z(z) e^{sz}} = P_z(z) e^{sz - \mu(s)} \quad (5A.6)$$

Observe from (5A.2) and (5A.3) that  $\mu'(s)$  and  $\mu''(s)$  are, respectively, the mean and variance of each of the tilted variables  $z_{n,s}$ . It follows from this that  $\mu''(s)$  is positive (with the exception of the trivial random variables that take on a single value with probability 1). Thus  $\mu'(s)$  is strictly increasing. It can be seen from (5A.2) that

$$\lim_{s \rightarrow -\infty} \mu'(s)$$

is the smallest value taken on by  $z$  and

$$\lim_{s \rightarrow +\infty} \mu'(s)$$

is the largest value.

Now assume that the tilted variables  $z_{n,s}$  are statistically independent, and define the tilted sum  $w_s$  as

$$w_s = \sum_{n=1}^N z_{n,s} \quad (5A.7)$$

The mean and variance of  $w_s$  are given by

$$\bar{w}_s = N\mu'(s); \quad \text{VAR}(w_s) = N\mu''(s) \quad (5A.8)$$

We next relate the probability assignment on  $w_s$ , denoted by  $Q_{w,s}$ , to the probability assignment  $P_w$  in the original sum. The probability of any particular sequence of tilted variables is given by

$$\prod_{n=1}^N P_z(z_{n,s}) \exp [sz_{n,s} - \mu(s)]$$

Thus

$$Q_{w,s}(w_s) = \sum_{z_{1,s}} \cdots \sum_{z_{N,s}} \prod_{n=1}^N [P_z(z_{n,s}) e^{sz_{n,s} - \mu(s)}]$$

where the summation is restricted to those  $z_{1,s}, \dots, z_{N,s}$  satisfying  $\sum z_{n,s} = w_s$

$$Q_{w,s}(w_s) = \sum_{z_{1,s}} \cdots \sum_{z_{N,s}} \prod_n [P_z(z_{n,s})] e^{sw_s - N\mu(s)}$$

with the same restriction on  $z_{n,s}$ .

$$Q_{w,s}(w_s) = P_w(w_s) \exp [sw_s - N\mu(s)] \quad (5A.9)$$

Observe that  $Q_{w,s}$  is tilted from  $P_w$  in the same sense as  $Q_{z,s}$  is tilted from  $P_z$ .

If we wish to find  $\Pr(w \geq A)$ , for  $A > \bar{w}$ , we select that unique value of  $s$  for which

$$N\mu'(s) = A \quad (5A.10)$$

Since  $\mu'(s)$  is increasing, the  $s$  satisfying (5A.10) must be greater than 0. Using (5A.9), we now have

$$\begin{aligned} \Pr[w \geq N\mu'(s)] &= \sum_{w_s \geq N\mu'(s)} P_w(w_s) \\ &= \sum_{w_s \geq N\mu'(s)} Q_{w,s}(w_s) e^{-sw_s + N\mu(s)} \\ &= e^{N[\mu(s) - s\mu'(s)]} \sum_{w_s \geq N\mu'(s)} Q_{w,s}(w_s) e^{-s[w_s - N\mu'(s)]} \end{aligned} \quad (5A.11)$$

Observe that the sum in (5A.11) starts at the mean value of  $w_s$ , and that the exponential decay factor effectively chops off the sum for large  $w_s$ . In fact, since the standard deviation of  $w_s$  is proportional to  $\sqrt{N}$ , and since the rate of exponential decay is independent of  $N$ , we are interested only in  $Q_{w,s}$  within a small fraction of a standard deviation for large  $N$ . This suggests using a central-limit theorem to estimate  $Q_{w,s}$ , but also suggests using a form of the central-limit theorem that is sensitive to small changes in  $w_s$ . The appropriate theorem to use depends upon whether or not  $z_{n,s}$  is a lattice random variable. A lattice random variable is a variable whose sample values can be expressed in the form  $\alpha + ih$  where  $\alpha$  and  $h$  are fixed constants and  $i$  is an integer that varies with the sample value. For example, the sample values 0, 1, and 2 correspond to a lattice variable and the sample values 1,  $1 + \pi$ , and  $1 + 2\pi$  correspond to a lattice variable. The sample values 0, 1, and  $\pi$  do not correspond to a lattice variable. The span,  $h$ , of a lattice variable is the largest  $h$  that can be used in the above definition. If  $z_n$  is a lattice variable, then  $z_{n,s}$ ,  $w$ , and  $w_s$  are clearly also lattice variables with the same span. If  $z_n$  and hence  $z_{n,s}$  are nonlattice, then  $w_s$  behaves in a distinctly different way, the separation between adjacent sample points becoming less and less as  $N$  is increased.

For a lattice distribution with span  $h$ , the appropriate central-limit theorem\* states that, on the lattice points,

$$\left| Q_{w,s}(w_s) - \frac{h}{\sqrt{2\pi N\mu''(s)}} \exp \frac{-[w_s - N\mu'(s)]^2}{2N\mu''(s)} \right| \leq \frac{1}{\sqrt{N}} \epsilon(N) \quad (5A.12)$$

where  $\epsilon(N)$  is independent of the sample value  $w_s$  and

$$\lim_{N \rightarrow \infty} \epsilon(N) = 0$$

In other words,  $Q_{w,s}(w_s)$  is approximately equal to the spacing between lattice points,  $h$ , times the density of a Gaussian distribution with the same mean and variance as  $w_s$ .

Since we are interested only in values of  $w_s$  very close to the mean, we can use the relation  $1 \geq e^{-x} \geq 1 - x$  in (5A.12) to get

$$\left| Q_{w,s}(w_s) - \frac{h}{\sqrt{2\pi N\mu''(s)}} \right| \leq \frac{1}{\sqrt{N}} \epsilon(N) + \frac{h[w_s - N\mu'(s)]^2}{2\sqrt{2\pi} N^{\frac{3}{2}} [\mu''(s)]^{\frac{3}{2}}} \quad (5A.13)$$

In order to calculate the sum in (5A.11), we first use  $h/\sqrt{2\pi N\mu''(s)}$  in place of  $Q_{w,s}$ . Let  $\Delta$  be the distance from  $N\mu'(s)$  to the first sample point of  $w_s$  included in the sum. We then have

$$\begin{aligned} \sum_{w_s \geq N\mu'(s)} \frac{h}{\sqrt{2\pi N\mu''(s)}} \exp \{-s[w_s - N\mu'(s)]\} \\ = \frac{he^{-s\Delta}}{\sqrt{2\pi N\mu''(s)}} [1 + e^{-hs} + e^{-2hs} + \dots] \\ = \frac{he^{-s\Delta}}{\sqrt{2N\mu''(s)}(1 - e^{-sh})} \end{aligned} \quad (5A.14)$$

Similarly, we can multiply the two error terms in (5A.13) by  $\exp -s[w_s - N\mu'(s)]$  and sum over the sample values  $w_s \geq N\mu'(s)$ . The first sum goes to zero faster than  $1/\sqrt{N}$  as  $N \rightarrow \infty$  and the second term goes to zero as  $N^{-\frac{3}{2}}$ . Combining (5A.13) and (5A.14), we consequently have

$$\sum_{w_s \geq N\mu'(s)} Q_{w,s}(w_s) \exp \{-s[w_s - N\mu'(s)]\} = \frac{he^{-s\Delta}}{\sqrt{2N\mu''(s)}(1 - e^{-sh})} + o(1/\sqrt{N}) \quad (5A.15)$$

where  $o(1/\sqrt{N})$  goes to zero as  $N \rightarrow \infty$  faster than  $1/\sqrt{N}$ . Combining (5A.15) with (5A.11), we have the final result for lattice variables,

$$\Pr[w \geq N\mu'(s)] = \exp \{N[\mu(s) - s\mu'(s)]\} \left[ \frac{he^{-s\Delta}}{\sqrt{2N\mu''(s)}(1 - e^{-sh})} + o(1/\sqrt{N}) \right] \quad (5A.16)$$

\* Feller (1966) p. 490.

Equation 5A.16 is valid for any  $s > 0$ , but as can be seen from the second error term in (5A.13), the convergence in  $N$  becomes slower as  $s$  gets closer to zero. Notice that, for a given  $s$ ,  $\Delta$  fluctuates with  $N$ , but of course always satisfies  $0 \leq \Delta < h$ .

We next estimate (5A.11) for the nonlattice case. The sum in (5A.11) can be “integrated by parts” to yield

$$\begin{aligned} & \sum_{w_s \geq N\mu'(s)} Q_{w,s}(w_s) \exp \{-s[w_s - N\mu'(s)]\} \\ &= \int_{w_s=N\mu'(s)}^{\infty} s\{F(w_s) - F[N\mu'(s)]\} \exp \{-s[w_s - N\mu'(s)]\} dw_s \end{aligned} \quad (5A.17)$$

where

$$F(w_s) = \sum_{w \leq w_s} Q_{w,s}(w)$$

is the distribution function of  $w_s$ . Now let  $u = [w_s - N\mu'(s)]/\sqrt{N\mu''(s)}$  be the normalized version of the random variable  $w_s$  and let  $G(u)$  be the distribution function of  $u$ . The right-hand side of (5A.17) can be written as

$$s\sqrt{N\mu''(s)} \int_0^{\infty} [G(u) - G(0)] \exp [-s\sqrt{N\mu''(s)}u] du \quad (5A.18)$$

The appropriate version of the central-limit theorem is now given as\*

$$G(u) = \Phi(u) + \frac{\mu_3(1-u)^2}{6\mu_2^{3/2}\sqrt{2\pi N}} e^{-u^2/2} + o\left(\frac{1}{\sqrt{N}}\right) \quad (5A.19)$$

where

$$\begin{aligned} \Phi(u) &= \int_{-\infty}^u \frac{1}{\sqrt{2\pi}} e^{-x^2/2} dx, \\ \mu_2 &= \overline{(z_{n,s} - \bar{z}_{n,s})^2}, \quad \mu_3 = \overline{(z_{n,s} - \bar{z}_{n,s})^3}, \end{aligned} \quad (5A.20)$$

and  $o(1/\sqrt{N})$  goes to zero as  $N \rightarrow \infty$ , uniformly in  $u$ , and faster than  $1/\sqrt{N}$ .

Using the relation  $1 \geq e^{-x^2/2} \geq 1 - x^2/2$  in (5A.20) we have, for  $u > 0$ ,

$$\frac{u}{\sqrt{2\pi}} \geq \Phi(u) - \Phi(0) \geq \frac{u}{\sqrt{2\pi}} - \frac{u^3}{6\sqrt{2\pi}} \quad (5A.21)$$

Substituting (5A.21) into (5A.19) yields

$$\left| G(u) - G(0) - \frac{u}{\sqrt{2\pi}} \right| \leq \frac{u^3}{6\sqrt{2\pi}} + \frac{|\mu_3| [1 - (1 - \mu_2^2)e^{-u^2/2}]}{6\mu_2^{3/2}\sqrt{2\pi N}} + o(1/\sqrt{N}) \quad (5A.22)$$

Approximating  $G(u) - G(0)$  in (5A.18) by  $u/\sqrt{2\pi}$ , we get

$$s\sqrt{N\mu''(s)} \int_0^{\infty} \frac{u}{\sqrt{2\pi}} e^{-s\sqrt{N\mu''(s)}u} du = \frac{1}{s\sqrt{2\pi N\mu''(s)}} \quad (5A.23)$$

\* See Feller (1966) p. 512. It can be verified by direct calculation that  $\mu_3 = \mu'''(s)$  and is finite.

Multiplying each of the error terms in (5A.22) by  $s\sqrt{N\mu''(s)}e^{-s\sqrt{N\mu''(s)}u}$  and integrating, we see that each integral vanishes faster than  $1/\sqrt{N}$  as  $N \rightarrow \infty$ , yielding

$$s\sqrt{N\mu''(s)} \int_0^\infty [G(u) - G(0)] e^{[-s\sqrt{N\mu''(s)}u]} du = \frac{1}{s\sqrt{2\pi N\mu''(s)}} + o\left(\frac{1}{\sqrt{N}}\right) \quad (5A.24)$$

Recalling that these quantities equal the left side of (5A.17), we can substitute this result into (5A.11) to obtain

$$\Pr[w \geq N\mu'(s)] = e^{N[\mu(s) - s\mu'(s)]} \left[ \frac{1}{s\sqrt{2\pi N\mu''(s)}} + o\left(\frac{1}{\sqrt{N}}\right) \right] \quad (5A.25)$$

## APPENDIX 5B

In this appendix, we prove Theorems 5.6.3 and 5.7.2, establishing the behavior of  $E_o(\rho, \mathbf{Q})$  and  $E_x(\rho, \mathbf{Q})$  as functions of  $\rho$ . We begin with a lemma.

**LEMMA.** Let  $\mathbf{Q} = [Q(0), \dots, Q(K-1)]$  be a probability vector and let  $a_o, \dots, a_{K-1}$  be a set of nonnegative numbers. Then the function

$$f(s) = \ln \left[ \sum_{k=0}^{K-1} Q(k) a_k^{1/s} \right]^s \quad (5B.1)$$

is nonincreasing and convex  $\cup$  with  $s > 0$ . Moreover,  $f(s)$  is strictly decreasing unless all the  $a_k$  for which  $Q(k) > 0$  are equal. The convexity is strict unless all the nonzero  $a_k$  for which  $Q(k) > 0$  are equal.

*Proof.* The fact that  $f(s)$  is nonincreasing and the conditions for it to be strictly decreasing follow directly from the standard inequality

$$[\sum Q(k) a_k^{1/s}]^s \geq [\sum Q(k) a_k^{1/r}]^r$$

for  $0 < s < r$  given in Problem 4.15e. To establish the convexity, let  $s, r$ , and  $\theta$  be arbitrary numbers,  $0 < s < r$ ,  $0 < \theta < 1$ , and define

$$t = \theta s + (1 - \theta)r \quad (5B.2)$$

To show that  $f(s)$  is convex  $\cup$ , we must show that

$$f(t) \leq \theta f(s) + (1 - \theta)f(r) \quad (5B.3)$$

Define the number  $\lambda$  by

$$\lambda = \frac{s\theta}{t}; \quad 1 - \lambda = \frac{r(1 - \theta)}{t} \quad (5B.4)$$

These expressions can be seen to be consistent by adding them and using (5B.2). It also follows from (5B.4) that

$$\frac{1}{t} = \frac{\theta}{t} + \frac{(1 - \theta)}{t} = \frac{\lambda}{s} + \frac{1 - \lambda}{r} \quad (5B.5)$$

$$\sum_k Q(k)a_k^{1/t} = \sum_k Q(k)a_k^{\lambda/s}a_k^{(1-\lambda)/r} \leq \left[ \sum_k Q(k)a_k^{1/s} \right]^\lambda \left[ \sum_k Q(k)a_k^{1/r} \right]^{1-\lambda} \quad (5B.6)$$

where (5B.6) follows from Holder's inequality (see Problem 4.15c). Raising both sides of (5B.6) to the  $t$  power and using (5B.4),

$$\left[ \sum_k Q(k)a_k^{1/t} \right]^t \leq \left[ \sum_k Q(k)a_k^{1/s} \right]^{s\theta} \left[ \sum_k Q(k)a_k^{1/r} \right]^{r(1-\theta)} \quad (5B.7)$$

Taking the logarithm of (5B.7), we obtain (5B.3). The convexity is strict unless (5B.6) is satisfied with equality, which occurs iff there is a constant  $C$  such that  $Q(k)a_k^{1/s} = Q(k)a_k^{1/r} C$  for all  $k$  (see Problem 4.15c). This immediately implies the condition for strict convexity in the lemma. |

*Proof of Theorem 5.6.3*

$$E_o(\rho, \mathbf{Q}) = -\ln \sum_{j=0}^{J-1} \left[ \sum_{k=0}^{K-1} Q(k)P(j|k)^{1/(1+\rho)} \right]^{1+\rho} \quad (5B.8)$$

Letting  $P(j|k)$  correspond to  $a_k$  in the lemma and  $1 + \rho$  correspond to  $s$ , we see that

$$\left[ \sum_k Q(k)P(j|k)^{1/(1+\rho)} \right]^{1+\rho}$$

is nonincreasing with  $\rho$  for each  $j$ . By assumption,  $\mathcal{I}(\mathbf{Q}; \mathbf{P}) > 0$ , and thus  $P(j|k)$  is not independent of  $k$  over those  $k$  for which  $Q(k) > 0$ . Thus the expression above is strictly decreasing for at least one  $j$ ;  $E_o(\rho, \mathbf{Q})$  is strictly increasing with  $\rho$ , and  $\partial E_o(\rho, \mathbf{Q}) / \partial \rho > 0$  for  $\rho \geq 0$ . Since  $E_o(0, \mathbf{Q}) = 0$ , this implies also that  $E_o(\rho, \mathbf{Q}) > 0$  for  $\rho > 0$ . Next, we show that  $E_o(\rho, \mathbf{Q})$  is convex in  $\rho$ . Let  $\rho_1 > 0$  and  $\rho_2 > 0$  be arbitrary, and let  $\theta$  satisfy  $0 < \theta < 1$ . Define  $\rho_3$  as  $\rho_1\theta + \rho_2(1 - \theta)$ . From the lemma (Equation 5B.7), we have

$$\begin{aligned} & \sum_j \left[ \sum_k Q(k)P(j|k)^{1/(1+\rho_3)} \right]^{1+\rho_3} \\ & \leq \sum_j \left[ \sum_k Q(k)P(j|k)^{1/(1+\rho_1)} \right]^{(1+\rho_1)\theta} \left[ \sum_k Q(k)P(j|k)^{1/(1+\rho_2)} \right]^{(1+\rho_2)(1-\theta)} \end{aligned} \quad (5B.9)$$

We now apply Holder's inequality (see Problem 4.15b),

$$\sum_j a_j b_j \leq \left[ \sum_j a_j^{1/\theta} \right]^\theta \left[ \sum_j b_j^{1/(1-\theta)} \right]^{1-\theta} \quad (5B.10)$$

to the right-hand side of (5B.9), obtaining

$$\sum_j \left[ \sum_k Q(k)P(j|k) \right]^{1/\rho_3} \leq \left\{ \sum_j \left[ \sum_k Q(k)P(j|k) \right]^{1/\rho_1} \right\}^{\rho_3} \\ \times \left\{ \sum_j \left[ \sum_k Q(k)P(j|k) \right]^{1/\rho_2} \right\}^{1-\theta} \quad (5B.11)$$

Taking the logarithm of both sides of (5B.11),

$$-E_o(\rho_3, \mathbf{Q}) \leq -\theta E_o(\rho_1, \mathbf{Q}) - (1-\theta)E_o(\rho_2, \mathbf{Q}) \quad (5B.12)$$

This establishes that  $E_o(\rho, \mathbf{Q})$  is convex  $\cap$  in  $\rho$ . The convexity fails to be strict iff both (5B.9) and (5B.10) are satisfied with equality. From the lemma, (5B.9) is satisfied with equality iff  $P(j|k)$  is independent of  $k$  for all  $j, k$  satisfying  $Q(k)P(j|k) > 0$ . The condition for equality in (5B.10) (see Problem 4.15b) is that there is a constant  $C$  such that, for all  $j$ ,

$$\left[ \sum_k Q(k)P(j|k)^{1/(1+\rho_1)} \right]^{1+\rho_1} = C \left[ \sum_k Q(k)P(j|k)^{(1/1+\rho_2)} \right]^{1+\rho_2}$$

If (5B.9) is satisfied with equality, then the nonzero  $P(j|k)$  can be factored out of the above equation, leaving

$$\left[ \sum_{k:P(j,k)>0} Q(k) \right]^{1+\rho_1} = C \left[ \sum_{k:P(j,k)>0} Q(k) \right]^{1+\rho_2} \quad (5B.13)$$

for all  $j$ . This implies that the term in brackets above is some constant  $\alpha$ , independent of  $j$ , and thus for any  $j, k$  with  $Q(k)P(j|k) > 0$ , we have

$$\frac{\sum_i Q(i)P(j|i)}{P(j|k)} = \alpha \quad | \quad (5B.14)$$

*Proof of Theorem 5.7.2*

$$E_x(\rho, \mathbf{Q}) = -\ln \left\{ \sum_k \sum_i Q(k)Q(i) \left[ \sum_j \sqrt{P(i|k)P(j|i)} \right]^{1/\rho} \right\}^\rho \quad (5B.15)$$

We can apply the lemma directly to (5B.15) by associating the double summation in (5B.15) with the single summation in (5B.1), associating  $Q(k)Q(i)$  with  $Q(k)$  in (5B.1), and

$$\sum_j \sqrt{P(j|k)P(j|i)}$$

with  $a_k$  in (5B.1). Thus  $E_x(\rho, \mathbf{Q})$  is increasing and convex  $\cap$  in  $\rho$ . The convexity is strict unless the nonzero values of

$$\sum_j \sqrt{P(j|k)P(j|i)}$$

for which  $Q(k)Q(i) > 0$  are all the same. This sum is 1 for  $k = i$ , and (from Problem 4.15a) it is 1 for  $k \neq i$  iff  $P(j|k) = P(j|i)$  for all  $j$ . Likewise the above sum is 0 iff  $P(j|k)P(j|i) = 0$  for all  $j$ , establishing the conditions in the Theorem for strict convexity. |

## *Chapter 6*

### TECHNIQUES FOR CODING AND DECODING

#### 6.1 Parity-Check Codes

In the previous chapter, we considered the use of block coding as a means of transmitting data reliably over discrete memoryless channels. We showed that, for appropriately chosen codes of any given transmission rate  $R$  less than capacity, the probability of decoding error is bounded by  $P_e \leq \exp -NE_r(R)$  where  $N$  is the block length and  $E_r(R) > 0$  is given by (5.6.16). On the other hand, the number of code words and the number of possible received sequences are exponentially increasing functions of  $N$ ; thus, for large  $N$ , it is impractical to store the code words in the encoder and to store the mapping from received sequences to messages in the decoder.

In this chapter, we discuss techniques for encoding and decoding that avoid these storage problems by providing algorithms to generate code words from messages and messages from received sequences. Almost all of the known encoding and decoding techniques involve the ideas of parity-check codes and, thus, we start there. It will be helpful at first to visualize these codes as being designed for a binary symmetric channel (BSC), although we shall see later that this viewpoint is unnecessarily restrictive.

A parity-check code is a particular type of mapping from binary sequences of one length  $L$  into binary sequences of some longer length  $N$ . Before defining parity-check codes, we shall give a particularly simple and widely used example, that of a single parity check. Suppose that a sequence of binary digits is encoded simply by adding one binary digit at the end of the sequence, choosing that last digit so that the total number of ones in the encoded sequence is even. We call the original digits information digits and the final digit at the end a check digit. It is easy to see that if any single digit in the sequence is then changed, the total number of ones will be odd and the fact that an error occurred can be detected. If two errors occur, however, the parity will be even again and the errors cannot be detected. This type of coding

is widely used on magnetic tapes and in other situations where a small amount of error detecting capability is desired for a minimal expenditure in equipment.

It is convenient to consider these even-odd checks in terms of modulo-2 arithmetic. In this kind of arithmetic, there are only two numbers, **0** and **1**, and addition (denoted by  $\oplus$ ) is defined by  $\mathbf{0} \oplus \mathbf{0} = \mathbf{0}$ ;  $\mathbf{0} \oplus \mathbf{1} = \mathbf{1}$ ;  $\mathbf{1} \oplus \mathbf{0} = \mathbf{1}$ ;  $\mathbf{1} \oplus \mathbf{1} = \mathbf{0}$ . It can be seen that this is identical to ordinary addition except for  $\mathbf{1} \oplus \mathbf{1} = \mathbf{0}$ . Multiplication is the same as in ordinary arithmetic and will be denoted in the same way.

It is trivial to verify (by exhaustion, if one is a sceptic) that the usual associative, commutative, and distributive laws of arithmetic are satisfied by modulo-2 arithmetic. That is, if  $a$ ,  $b$ , and  $c$  are binary numbers

$$\begin{aligned} (a \oplus b) \oplus c &= a \oplus (b \oplus c) \\ (ab)c &= a(bc) \quad \left. \right\} \text{associative} \\ a \oplus b &= b \oplus a; ab = ba \quad \text{commutative} \\ (a \oplus b)c &= ac \oplus bc \quad \text{distributive} \end{aligned}$$

There are a number of interpretations of modulo-2 arithmetic. If we interpret **0** as even and **1** as odd, then these addition and multiplication rules are the rules for combining even and odd numbers. Alternatively, we can interpret the modulo-2 sum of a sequence of binary numbers as the remainder when the ordinary sum is divided by 2. We then see that the check digit in the example above is chosen so that the modulo-2 sum of all the digits is **0**. If the modulo-2 sum of the information digits is **0**, then the check digit is chosen to be **0**; if the modulo-2 sum of the information digits is **1**, the check digit is chosen to be **1**. Thus the check digit is the modulo-2 sum of the information digits.

As a simple (but extremely powerful) generalization of the use of a single parity check to check on a sequence of information digits, we can consider using a set of parity-check digits, each checking on some prespecified set of the information digits. More precisely, let  $\mathbf{u} = (u_1, \dots, u_L)$  denote a sequence of  $L$  binary information digits and consider forming a code word  $\mathbf{x}$  of block length  $N > L$  from  $\mathbf{u}$  by the rule

$$x_n = u_n; \quad 1 \leq n \leq L \quad (6.1.1)$$

$$x_n = \sum_{l=1}^L u_l g_{l,n}; \quad L + 1 \leq n \leq N \quad (6.1.2)$$

where  $\sum$  denotes modulo-2 addition.

The elements  $g_{l,n}$  for  $1 \leq l \leq L$  and  $L + 1 \leq n \leq N$  in (6.1.2) are fixed binary digits, independent of  $\mathbf{u}$ , and thus (6.1.1) and (6.1.2) provide a mapping from the set of  $2^L$  possible information sequences to a set of  $2^L$  code words of block length  $N$ . We call the first  $L$  digits in each code word *information digits* and the last  $N - L$  digits *check digits*.

A systematic parity-check code is defined as any binary block code of arbitrary block length  $N$  in which the set of messages is the set of  $2^L$  binary sequences of some fixed length  $L < N$  and for each message  $\mathbf{u} = (u_1, \dots, u_L)$ , the associated code word  $\mathbf{x} = (x_1, \dots, x_N)$  is given by (6.1.1) and (6.1.2) where the set of binary digits  $\{g_{l,n}\}$  for  $1 \leq l \leq L, L+1 \leq n \leq N$  is arbitrary but fixed independent of  $\mathbf{u}$ . Each selection of the set  $\{g_{l,n}\}$ , of course, yields a different systematic parity-check code.

As an example, for  $L = 3, N = 6$ , let  $\mathbf{u} = (u_1, u_2, u_3)$  denote an information sequence and  $\mathbf{x} = (x_1, x_2, x_3, x_4, x_5, x_6)$  denote the associated code word, formed as in Figure 6.1.1.

|                        | Information Sequences | Code Words |
|------------------------|-----------------------|------------|
| $x_1 = u_1$            | 000                   | 000000     |
| $x_2 = u_2$            | 001                   | 001110     |
| $x_3 = u_3$            | 010                   | 010101     |
| $x_4 = u_2 \oplus u_3$ | 011                   | 011011     |
| $x_5 = u_1 \oplus u_3$ | 100                   | 100011     |
| $x_6 = u_1 \oplus u_2$ | 101                   | 101101     |
|                        | 110                   | 110110     |
|                        | 111                   | 111000     |

Figure 6.1.1.

A parity-check code is defined in the same way as a systematic parity-check code with the exception that in place of (6.1.1) and (6.1.2), code words are formed from message sequences according to the rule

$$x_n = \sum_{l=1}^L u_l g_{l,n}; \quad 1 \leq n \leq N \quad (6.1.3)$$

where  $\{g_{l,n}\}$  is an arbitrary but fixed set of binary numbers,  $1 \leq l \leq L$ ,  $1 \leq n \leq N$ . By comparing (6.1.1) and (6.1.3) we see that a systematic parity-check code is a special case of a parity-check code in which

$$\begin{aligned} g_{l,n} &= 1; & l &= n \\ g_{l,n} &= 0; & 1 \leq n \leq L, l &\neq n \end{aligned} \quad (6.1.4)$$

When we want to specifically denote the block length  $N$  and message-sequence length  $L$  of a parity-check code, we shall refer to it as an  $(N,L)$  parity-check code [or  $(N,L)$  systematic parity-check code].

One of the reasons for considering parity-check codes, systematic or not, can be seen by considering the implementation of an encoder. A parity-check encoder will require a register to store the message sequence  $\mathbf{u}$ , a register to store the code word  $\mathbf{x}$ , and a number of modulo-2 adders proportional to

$NL$ . Thus parity-check encoders avoid the exponential growth of storage with  $L$  required in arbitrary, unstructured block codes with  $2^L$  code words.

### Generator Matrices

Equation 6.1.3 can be expressed in a more compact way by defining the *generator matrix*  $G$  of an  $(N,L)$  parity-check code. As shown in Figure 6.1.2, this is an  $L$  by  $N$  binary matrix with the components  $g_{l,n}$  given in (6.1.3). For a systematic parity-check code, we see from (6.1.4) that the submatrix corresponding to the first  $L$  columns is an identity matrix

Considering  $\mathbf{u}$  and  $\mathbf{x}$  as row vectors, we then have

$$\mathbf{x} = \mathbf{u}G \quad (6.1.5)$$

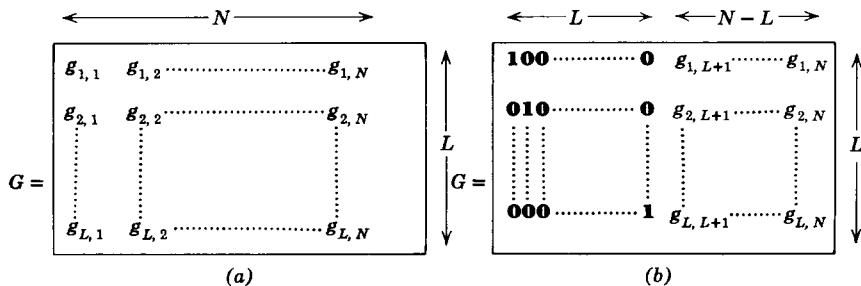


Figure 6.1.2. Generator matrix. (a) Arbitrary parity-check code. (b) Systematic parity-check code.

where  $\mathbf{u}G$  is matrix multiplication using modulo-2 addition [that is,  $\mathbf{u}G$  is defined to make (6.1.5) equivalent to (6.1.3)].

Now let  $\mathbf{u}'$  and  $\mathbf{u}''$  be two information sequences and define the modulo-2 sum of 2 binary vectors by  $\mathbf{u} = \mathbf{u}' \oplus \mathbf{u}'' = (u_1' \oplus u_1'', \dots, u_L' \oplus u_L'')$ . Then, if  $\mathbf{x}' = \mathbf{u}'G$  and  $\mathbf{x}'' = \mathbf{u}''G$ , we have

$$\mathbf{x}' \oplus \mathbf{x}'' = \mathbf{u}'G \oplus \mathbf{u}''G = (\mathbf{u}' \oplus \mathbf{u}'')G \quad (6.1.6)$$

$$= \mathbf{u}G \quad (6.1.7)$$

The last step in (6.1.6) follows from the associative, commutative, and distributive laws for modulo-2 addition and can be verified by going back to (6.1.3). We see from (6.1.7) that the modulo-2 sum of two code words is another code word, that corresponding to the information sequence  $\mathbf{u} = \mathbf{u}' \oplus \mathbf{u}''$ .

Next, observe that if the information sequence consists of a single **1**, say in the  $l$ th position, then the resulting code word is the  $l$ th row of the matrix  $G$ ,

denoted  $\mathbf{g}_l$ . Combining this with (6.1.6), we can represent an arbitrary code word by

$$\mathbf{x} = \sum_l u_l \mathbf{g}_l \quad (6.1.8)$$

In other words, the set of code words is the *row space* of  $G$ , that is, the set of linear modulo-2 combinations of the rows of  $G$  as given by (6.1.8).

### Parity-Check Matrices for Systematic Parity-Check Codes

We now restrict ourselves temporarily to  $(N, L)$  systematic parity-check codes and define a new matrix  $H$  called a *parity-check matrix*. It is an  $N$  by  $N - L$  matrix defined in terms of the  $g_{l,n}$  of (6.1.2) by Figure 6.1.3.

$$H = \left[ \begin{array}{cccc|c} g_{1,L+1} & \cdots & \cdots & \cdots & g_{1,N} \\ g_{2,L+1} & \cdots & \cdots & \cdots & g_{2,N} \\ \vdots & & & & \vdots \\ g_{L,L+1} & \cdots & \cdots & \cdots & g_{L,N} \\ \hline \mathbf{1} & \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{0} & \mathbf{1} & \mathbf{0} & \cdots & \mathbf{0} \\ \vdots & \vdots & \vdots & & \vdots \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \cdots & \mathbf{1} \end{array} \right] \quad N$$

Figure 6.1.3.

To see the significance of  $H$ , we rewrite (6.1.2), using (6.1.1) as

$$x_n = \sum_{l=1}^L x_l g_{l,n}; \quad L < n \leq N \quad (6.1.9)$$

Adding  $x_n$  to both sides (or subtracting  $x_n$ , which is equivalent in modulo-2 arithmetic), we get, for every code word  $\mathbf{x}$ ,

$$\mathbf{0} = \sum_{l=1}^L x_l g_{l,n} \oplus x_n; \quad L < n \leq N \quad (6.1.10)$$

Comparing this equation with Figure 6.1.3, we see that it can be rewritten in matrix terminology: for each code word  $\mathbf{x}$ ,

$$\mathbf{x}H = \mathbf{0} \quad (6.1.11)$$

Conversely, if an arbitrary sequence  $\mathbf{x}$  satisfies (6.1.11), it also satisfies (6.1.9). Such a sequence must be a code word, corresponding to the information sequence given by the first  $L$  digits of  $\mathbf{x}$ . Summarizing, the set of code words is the set of sequences  $\mathbf{x}$  for which  $\mathbf{x}H = \mathbf{0}$  (that is, the column null space of  $H$ ) and the set of code words is also the set of linear combinations of rows of  $G$  (that is, the row space of  $G$ ).

The matrix  $H$  is useful primarily in decoding. If sequence  $\mathbf{y}$  is received on a binary channel using a given systematic parity-check code, we define the syndrome  $\mathbf{S}$  associated with  $\mathbf{y}$  as

$$\mathbf{S} = \mathbf{y}H \quad (6.1.12)$$

The syndrome  $\mathbf{S}$  is a row vector  $(S_1, \dots, S_{N-L})$  with  $N - L$  components, one for each check digit. Since

$$S_i = \sum_{l=1}^L y_l g_{l,L+i} \oplus y_{L+i}$$

we see that  $S_i$  is 1 iff the received  $i$ th check digit  $y_{L+i}$  differs from the  $i$ th check digit as recomputed from the received information digits.

Observe that if  $\mathbf{x}_m$  is any code word, then

$$(\mathbf{y} \oplus \mathbf{x}_m)H = \mathbf{y}H \oplus \mathbf{x}_mH = \mathbf{S} \quad (6.1.13)$$

If  $\mathbf{x}_m$  is transmitted and  $\mathbf{y}$  received, then  $\mathbf{y} \oplus \mathbf{x}_m$  is a sequence that contains a 1 in each position in which  $\mathbf{y}$  and  $\mathbf{x}_m$  differ. This sequence is called the *error sequence*,  $\mathbf{z}_m$ , corresponding to  $\mathbf{x}_m$  and from (6.1.13), we have

$$\mathbf{z}_mH = \mathbf{S} \quad (6.1.14)$$

Since  $\mathbf{x}H = 0$  iff  $\mathbf{x}$  is a code word, we see from (6.1.13) that  $\mathbf{z}H = \mathbf{S}$  iff  $\mathbf{z}$  is one of the above error sequences.

It should be emphasized that (6.1.14) does not allow us to solve for the error sequence that actually occurred in transmission. It is an equation satisfied by each of the  $M = 2^L$  possible error sequences corresponding to the  $M$  code words. Which  $\mathbf{z}_m$  to choose (or which code word to choose) depends upon the channel.

Suppose now that a systematic parity-check code is used on a BSC (see Figure 5.3.1a) with transition probability  $\epsilon < \frac{1}{2}$ . Then, if a code word  $\mathbf{x}_m$  and a received sequence  $\mathbf{y}$  differ in  $e$  places, we have  $\Pr(\mathbf{y} \mid \mathbf{x}_m) = \epsilon^e(1 - \epsilon)^{N-e}$ . The number of places in which two binary sequences differ is called the *Hamming distance* between those sequences. We see that  $\Pr(\mathbf{y} \mid \mathbf{x}_m)$  is a decreasing function of the Hamming distance between  $\mathbf{y}$  and  $\mathbf{x}_m$ , and thus maximum-likelihood decoding is equivalent to choosing the code word at the smallest distance from  $\mathbf{y}$ . We also notice that the distance between  $\mathbf{y}$  and  $\mathbf{x}_m$  is equal to the number of 1's (called the *weight*) of the error sequence  $\mathbf{z}_m = \mathbf{y} \oplus \mathbf{x}_m$ . Thus maximum-likelihood decoding is accomplished by choosing the code word  $\mathbf{x}_m$  for which  $\mathbf{z}_m = \mathbf{y} \oplus \mathbf{x}_m$  has minimum weight. We can summarize these results in the following theorem.

**Theorem 6.1.1.** Let a systematic parity-check code have a parity-check matrix  $H$ , and let this code be used on a BSC with transition probability  $\epsilon < \frac{1}{2}$ . Then, given a received sequence  $\mathbf{y}$ , maximum-likelihood decoding can be accomplished by calculating  $\mathbf{S} = \mathbf{y}H$ , finding the minimum weight sequence  $\mathbf{z}$  that satisfies  $\mathbf{S} = \mathbf{z}H$ , and decoding to the code word  $\mathbf{x} = \underline{\mathbf{z} \oplus \mathbf{y}}$ .

### Decoding Tables

This theorem leaves unanswered the problem of how to find the minimum-weight solution to  $\mathbf{S} = \mathbf{zH}$ . One way to accomplish this is by means of a decoding table. We can simply list the  $2^{N-L}$  possible values of  $\mathbf{S}$  and associate with each the minimum weight  $\mathbf{z}$ . The easiest way to accomplish this is to start out with the zero weight  $\mathbf{z}$ , then list all  $\mathbf{z}$  of weight one, then weight 2, and so on. For each  $\mathbf{z}$ , we can calculate  $\mathbf{S} = \mathbf{zH}$ , and if that  $\mathbf{S}$  appears further up in the list, we simply omit the new  $\mathbf{z}$  from the list. When all  $2^{N-L}$   $\mathbf{S}$  vectors have appeared, the table is complete. Figure 6.1.4 shows such a decoding table for the code of Figure 6.1.1.

| Syndrome<br>$\mathbf{S}$ | Error Sequence<br>$\mathbf{z}$ | Parity-Check<br>Matrix<br>$H$ |
|--------------------------|--------------------------------|-------------------------------|
| 000                      | 000000                         | [ 011 ]                       |
| 011                      | 100000                         | [ 101 ]                       |
| 101                      | 010000                         | [ 110 ]                       |
| 110                      | 001000                         | [ 100 ]                       |
| 100                      | 000100                         | [ 010 ]                       |
| 010                      | 000010                         | [ 001 ]                       |
| 001                      | 000001                         |                               |
| 111                      | 100100                         |                               |

Figure 6.1.4. Decoding table for the code of Figure 6.1.1.

If, for this code, the received sequence is  $\mathbf{y} = 010011$ , then  $\mathbf{S} = \mathbf{yH} = 110$ . From the table,  $\mathbf{z} = 001000$ , and the most likely code word is  $\mathbf{x} = \mathbf{y} \oplus \mathbf{z} = 011011$ . Thus the decoded message is **011**.

We next observe that the set of error sequences appearing in the decoding table is precisely the set of errors that will be corrected (independent of the transmitted code word). When any of these error sequences occur, the associated syndrome is calculated at the decoder, and the decoder looks up that error sequence in the decoding table. Whenever an error sequence not in the table occurs, some error sequence in the table is selected by the decoder, and a decoding error results. For the code of Figure 6.1.1 and the decoding table of Figure 6.1.4, all single errors are corrected and a single pattern of two errors is corrected. There are several patterns of two errors that have the syndrome 111 in Figure 6.1.4, but only one of them can be entered in the table, and for a binary symmetric channel, it clearly makes no difference which one.

From the above argument, we see that the probability of decoding error for a systematic parity-check code on a binary symmetric channel using a decoding table is given simply by the probability that an error sequence will occur which is not in the table. For the table in Figure 6.1.4, since the no-error sequence, the six one-error sequences, and a single two-error sequence are in the table, this error probability is given by

$$P_e = 1 - (1 - \epsilon)^6 - 6(1 - \epsilon)^5\epsilon - (1 - \epsilon)^4\epsilon^2 \quad (6.1.15)$$

### Hamming Codes

We have seen that the syndrome corresponding to an error sequence  $\mathbf{z}$  is given by  $\mathbf{S} = \mathbf{z}\mathbf{H}$ . If  $\mathbf{z}$  consists of only a single error, say in the  $n$ th position, then  $\mathbf{z}\mathbf{H}$  will be the same as the  $n$ th row of the matrix  $\mathbf{H}$ . If all rows of  $\mathbf{H}$  are nonzero and distinct, then the zero-error sequence and all single-error sequences will have different syndromes and, thus, the code will be capable of correcting all single errors. For a code with  $N - L$  check digits, there are  $2^{N-L} - 1$  different nonzero sequences that can be chosen as rows for  $\mathbf{H}$ , and if  $N \leq 2^{N-L} - 1$ , the rows of  $\mathbf{H}$  can be chosen to be nonzero and distinct.

*Hamming codes* are codes for which the rows of  $\mathbf{H}$  are distinct and include all the nonzero sequences of length  $N - L$ . We see that, for such codes,  $N$  and  $L$  are related by

$$N = 2^{N-L} - 1 \quad (6.1.16)$$

Figure 6.1.5 gives a short table of values of  $N$  and  $L$  satisfying (6.1.16) and gives  $\mathbf{H}$  for the  $N = 7, L = 4$  case.

| N  | L  | H =  |
|----|----|--|
| 3  | 1  | $\begin{bmatrix} 011 \\ 101 \\ 110 \end{bmatrix}$        |
| 7  | 4  | $\begin{bmatrix} 111 \\ 100 \\ 010 \\ 001 \end{bmatrix}$ |
| 15 | 11 |  |
| 31 | 26 |  |

Choices of  $N, L$  for  
Hamming Codes

Parity-Check Matrix  
for  $N = 7, L = 4$

Figure 6.1.5.

Since the decoding table for Hamming codes contains the zero-error sequence, all single-error sequences, and nothing else, we see that every possible received sequence is either a code word or is distance 1 from a single code word.

These codes are examples of *sphere-packed* codes. By a sphere of radius  $e$  around a sequence, we mean the set of all sequences at distances  $e$  or less from that sequence. By an  $e$ -error-correcting sphere-packed code, we mean a code with the property that the set of spheres of radius  $e$  around the code words are nonoverlapping and that every sequence is distance at most  $e + 1$  from some code word. By decoding a received sequence into the closest code word, all error patterns of, at most,  $e$  errors are corrected, some of  $e + 1$  are corrected, and none of more errors are corrected. It is easy to see (Section 5.8) that, on a binary symmetric channel, a sphere-packed code with this decoding scheme minimizes the probability of error among all codes of the same block length and number of code words.

If an  $e$ -error-correcting sphere-packed code satisfies the stronger condition that every sequence is at distance at most  $e$  from some code word, it is called a *perfect* code. Then all patterns of  $e$  errors are corrected and none of more than  $e$  errors are corrected by the above decoder. The only binary perfect codes that have been found are the Hamming codes (which are single-error correcting), all codes of two code words and odd block length where the code words differ in all positions, and a three-error correcting parity-check code with  $N = 23$  and  $L = 12$  discovered by Golay (1949). It has been shown by Elias\* that, for each rate  $R$ ,  $0 < R < 1$ , in bits, there is a block length  $N$  beyond which binary sphere-packed codes cannot exist.

This problem of finding sphere-packed codes is unfortunately representative of finding optimum (minimum-error probability) parity-check codes and arbitrary codes. Peterson (1961) gives a table of known optimum parity-check codes for the binary symmetric channel, but generalizations to longer block lengths at arbitrary rates do not appear to be possible. On the other hand, from a practical point of view, this is not a pressing problem. We know that the error probability at a given rate can be made arbitrarily small by increasing the block length. A more important problem than that of finding the *best* code of a given block length is that of finding the most easily instrumented code of *any* block length that gives the desired error probability.

We next investigate the relationship between different parity-check codes and, in particular, the relationship between systematic and nonsystematic codes. If two generator matrices have the same row space, then they generate the same set of code words, although with a different mapping from information sequences to code words. We call such generator matrices *equivalent*.

\* See Shannon, Gallager, and Berlekamp (1967), p. 547, for a proof of Elias' result.

Notice that if two rows,  $\mathbf{g}_i$  and  $\mathbf{g}_j$ , of a generator matrix are interchanged, the resulting matrix is equivalent to the original matrix. Also if row  $\mathbf{g}_i$  is added to  $\mathbf{g}_j$  as in Figure 6.1.6, the resulting matrices are equivalent.

An arbitrary generator matrix  $G$  can now be reduced to an equivalent matrix in “reduced echelon” form by taking the first nonzero column, interchanging rows so that the first row has a “1” in that column and adding

$$\left[ \begin{array}{c} \mathbf{g}_1 \\ \vdots \\ \mathbf{g}_i \\ \vdots \\ \mathbf{g}_j \\ \vdots \\ \mathbf{g}_L \end{array} \right] \leftrightarrow \left[ \begin{array}{c} \mathbf{g}_1 \\ \vdots \\ \mathbf{g}_i \\ \vdots \\ \mathbf{g}_j \oplus \mathbf{g}_i \\ \vdots \\ \mathbf{g}_L \end{array} \right]$$

Figure 6.1.6.

that row to all other rows with a 1 in that column, leaving the column with only a single 1 in the first row. The same thing can be done with the next column that has any 1’s in the last  $L - 1$  rows, leaving that column with a single 1 in the second row, and so forth. The process terminates either with  $L$  columns each having a single 1 in a different row or with one or more all-zero rows at the bottom of the matrix. The latter case, which occurs when the rows of  $G$  are linearly dependent, is uninteresting, since it means that there are fewer than  $2^L$  different code words, and for each message, there is at least one other message with the same code word (see Problem 6.10). In the former case, we can interpret the  $L$  columns with a single 1 each as corresponding to information digits and the other  $N - L$  columns as corresponding to check digits.

Thus, for any generator matrix  $G$ , there is an equivalent matrix  $G'$  which, aside from the location of the information digits, corresponds to a systematic code. The parity-check matrix  $H$  for the equivalent systematic code is found as in Figure 6.1.3, with the difference that the identity part of the matrix occurs on the  $N - L$  rows corresponding to the check digit positions, and these need not be the last  $N - L$  rows. Since the original code has the same set of code words as the equivalent systematic code, the code words in each

satisfy  $\mathbf{x}H = 0$ . The syndrome of a received sequence  $\mathbf{y}$  is again defined as  $\mathbf{S} = \mathbf{y}H$ , and decoding can be accomplished by a syndrome-decoding table as before.

It can be seen that the useful property of a parity-check matrix is that a sequence  $\mathbf{x}$  is a code word iff  $\mathbf{x}H = 0$ . For any generator matrix with linearly independent rows, we have shown how to find one such parity-check matrix,  $H$ . It is not hard to see (Problem 6.11) that any matrix  $H'$  whose columns span the same space as the columns of  $H$  also has the property that  $\mathbf{x}$  is a code word iff  $\mathbf{x}H = 0$ . For this reason, we shall call any such matrix a parity-check matrix for the code.

## 6.2 The Coding Theorem for Parity-Check Codes

In proving the coding theorem for parity-check codes, it is convenient to define a slightly broader class of codes which, for reasons to be brought out in the next section, will be called coset codes. *An  $(N,L)$  coset code is defined as a code with  $2^L$  code words of block length  $N > L$  in which the messages are binary sequences of length  $L$  and the mapping from message  $\mathbf{u}$  to code word  $\mathbf{x}$  is given by*

$$\mathbf{x} = \mathbf{u}G \oplus \mathbf{v} \quad (6.2.1)$$

where  $G$  is a fixed but arbitrary  $L$  by  $N$  binary matrix and  $\mathbf{v}$  is a fixed but arbitrary sequence of  $N$  binary digits. The code words of a coset code are thus formed from the code words of a corresponding parity-check code,  $\mathbf{x}' = \mathbf{u}G$ , by adding the fixed sequence  $\mathbf{v}$  to each code word. For a BSC, this fixed sequence can be subtracted off from the received word before decoding, thus undoing its effect. More precisely, if  $\mathbf{y} = \mathbf{x} \oplus \mathbf{z}$  is received, then after subtracting  $\mathbf{v}$ , we have  $\mathbf{y}' = \mathbf{y} \oplus \mathbf{v}$ , which is just  $\mathbf{x}' \oplus \mathbf{z}$ . Since the noise sequence is independent of the transmitted sequence for a BSC, a maximum likelihood decoder will correctly decode the same set of noise sequences as for the associated parity-check code, and the error probability will be identical.

**Theorem 6.2.1.** Consider an ensemble of  $(N,L)$  coset codes where each digit in  $G$  and  $\mathbf{v}$  is selected to be **0** or **1**, independently and with equal probability. The average probability of error for each message for this ensemble of codes used on a BSC with maximum likelihood decoding satisfies

$$\bar{P}_{e,m} \leq \exp [-NE_r(R)] \quad (6.2.2)$$

where  $E_r(R)$  is given in terms of the channel crossover-probability  $\epsilon$  by (5.6.41) and (5.6.45) and  $R = (L \ln 2)/N$ .

*Proof.* Let  $\mathbf{u}_m$  be an arbitrary information sequence and  $\mathbf{x}_m$  be the associated code word. Over the ensemble of codes, the probability that  $\mathbf{u}_m$  will be

mapped into a given sequence  $\mathbf{x}_m$  is

$$Q_N(\mathbf{x}_m) = 2^{-N} \quad (6.2.3)$$

To see this, observe that there are  $2^{N(L+1)}$  ways of selecting  $G$  and  $\mathbf{v}$  and each has probability  $2^{-N(L+1)}$ . For each choice of  $G$ , there is one choice of  $\mathbf{v}$  that will give  $\mathbf{x}_m$  any fixed value. Since there are  $2^{NL}$  choices of  $G$ ,  $Q_N(\mathbf{x}_m) = 2^{NL} \cdot 2^{-N(L+1)} = 2^{-N}$ .

Next, let  $\mathbf{u}_{m'}$  be an information sequence other than  $\mathbf{u}_m$  and let  $\mathbf{x}_{m'}$  be the associated code word. We now show that  $\mathbf{x}_m$  and  $\mathbf{x}_{m'}$  are statistically independent over the ensemble of codes.

$$\mathbf{x}_m \oplus \mathbf{x}_{m'} = (\mathbf{u}_m \oplus \mathbf{u}_{m'})G \quad (6.2.4)$$

Suppose that  $\mathbf{u}_m$  and  $\mathbf{u}_{m'}$  differ in the  $j$ th position. Then, for any choice of  $\mathbf{g}_1, \dots, \mathbf{g}_{j-1}, \mathbf{g}_{j+1}, \dots, \mathbf{g}_L$ , there is one choice of  $\mathbf{g}_j$  that will give  $\mathbf{x}_m \oplus \mathbf{x}_{m'}$  any fixed value. Then, as before, there is one choice of  $\mathbf{v}$  that will give  $\mathbf{x}_{m'}$  any fixed value. Thus there are  $2^{N(L-1)}$  ways of selecting  $G$  and  $\mathbf{v}$  to achieve any desired value of the pair  $(\mathbf{x}_m, \mathbf{x}_{m'})$  and  $\Pr(\mathbf{x}_m, \mathbf{x}_{m'}) = 2^{-2N}$ . From this and (6.2.3), it follows that  $\mathbf{x}_m$  and  $\mathbf{x}_{m'}$  are independent.

Now we observe that, in Theorem 5.6.1, the code words were assumed to be chosen independently. On the other hand, if we inspect the proof, we find that only pairwise independence was used. Thus Theorem 5.6.1 applies to this ensemble. Consequently, Theorem 5.6.2 also applies and the calculation of  $E_r(R)$  in (5.6.41) and (5.6.45) for the BSC follows immediately. |

As in Section 5.6, this result asserts the existence of an  $(N, L)$  coset code with average error probability of, at most,  $\exp[-NE_r(R)]$  and, as we have observed, the corresponding parity-check code with the fixed sequence  $\mathbf{v}$  removed, has the same error probability. Also, since the set of correctly decoded error sequences is independent of the message,  $P_{e,m} \leq \exp[-NE_r(R)]$  for all messages in this code. Finally, as we argued in Section 6.1, the matrix  $G$  can be converted to a systematic generator matrix by elementary row operations and perhaps some column permutations.\* This proves the following corollary.

**COROLLARY.** For all positive integers  $L$  and  $N$ ,  $L < N$ , there exist  $(N, L)$  systematic parity-check codes for which, on a BSC,

$$P_{e,m} \leq \exp[-NE_r(R)]; \quad \text{all } m, 1 \leq m \leq 2^L \quad (6.2.5)$$

where  $R = (L \ln 2)/N$  and  $E_r(R)$  is given by (5.6.41) and (5.6.45).

---

\* This assumes that the rows of  $G$  are linearly independent. For a code with linearly dependent rows, there are at least two messages corresponding to each code word and a maximum likelihood decoder can never make an unambiguous decision. Since all such events were counted as errors in Theorem 5.6.1, there must exist codes for which the rows of  $G$  are linearly independent and which satisfy  $P_e \leq \exp[-NE_r(R)]$ .

Next, we consider the problem of using parity-check codes on arbitrary discrete memoryless channels. Suppose, for an initial example, that we have a channel with an input alphabet of three letters and that we wish to encode from binary input sequences of length  $L$  into code words of length  $N$ . The transmission rate here is  $R = (L \ln 2)/N$  and we suppose that the channel input probability assignment that maximizes  $E_r(R)$  in (5.6.16) for this channel and rate is given by  $Q(0) = \frac{3}{8}$ ,  $Q(1) = \frac{3}{8}$ ,  $Q(2) = \frac{1}{4}$ . First, our strategy will be to construct a coset code of block length  $3N$  using the encoding rule of (6.2.1) where  $G$  is an  $L$  by  $3N$  binary matrix and  $\mathbf{v}$  is a binary  $3N$ -tuple. We can consider the binary code words to be sequences of  $N$  binary triplets. We then encode these binary triplets into channel input letters by the rule given in Figure 6.2.1. This transformation will map each binary code word, which is a sequence of  $3N$  binary digits, into a channel code word which is a sequence of  $N$  channel digits.

Over the ensemble of codes in Theorem 6.2.1, each binary code word is a sequence of  $3N$  independent, equiprobable binary digits. Thus each channel code word is a sequence of  $N$  ternary independent digits with the probabilities  $Q(0) = \frac{3}{8}$ ,  $Q(1) = \frac{3}{8}$ ,  $Q(2) = \frac{1}{4}$ . Furthermore, the code words are pairwise statistically independent and, thus, Theorems 5.6.1 and 5.6.2 apply again. As a result, there exists a code of this type for which  $P_e \leq \exp[-NE_r(R)]$ .

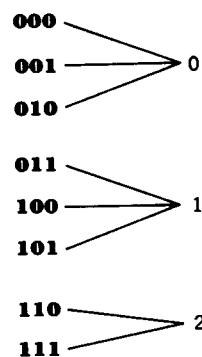
For a general discrete memoryless channel, we can apply the same technique except that the detailed transformation in Figure 6.2.1 will be different. We simply approximate the desired input probabilities  $\mathbf{Q}$  for the coding theorem by

$$Q(k) \approx \frac{i_k}{2^j}; \quad \sum_k i_k = 2^j \quad (6.2.6)$$

Then, in the transformation of Figure 6.2.1, for each  $k$ ,  $i_k$  binary  $j$ -tuples are mapped into the channel input  $k$ . How large  $j$  has to be in (6.2.6) to approximate  $Q(k)$  depends upon  $\mathbf{Q}$  and also upon how close we want to come to achieving the exponent  $E_r(R)$ . For any given  $j$ , any message length  $L$  and any channel code-word length  $N$ , we use the ensemble of binary codes in Theorem 6.2.1 with a binary block length  $jN$ . After mapping the binary code words into channel code words of length  $N$  by the above transformation, we see from Theorem 5.6.2 that a code exists for which

$$P_e \leq \exp \left\{ -N \left[ \max_{0 < \rho < 1} E_o(\rho, \mathbf{Q}) - \rho R \right] \right\} \quad (6.2.7)$$

where  $\mathbf{Q}$  is given by  $Q(k) = i_k/2^j$ .



*Figure 6.2.1.  
Mapping from binary sequences to channel input*

letters.

Summarizing, we have demonstrated a simple algorithm by which code words can be generated in such a way as to approximately achieve the results of the coding theorem. Unfortunately, the problem of finding *decoding* algorithms is not so simple.

### 6.3 Group Theory

In the previous two sections, we have made great use of modulo-2 addition. In doing this, we started out by defining a set of elements, **0** and **1**, and then we defined an operation  $\oplus$  on these elements. By an operation here, we mean a rule for combining two elements of the set into something else. In what follows, we shall be dealing with a number of other sets of elements and operations. Since all these will have the same general mathematical structure, that of a group, it will be efficient to pause here and develop some of the elementary results of group theory that we shall need.

*A group is defined as a set of elements,  $a, b, c, \dots$  and an operation, denoted by  $\cdot$  for which the following properties are satisfied.\**

- (1) For any elements  $a, b$ , in the set,  $a \cdot b$  is in the set.
- (2) The associative law is satisfied; that is, for any  $a, b, c$  in the set

$$a \cdot (b \cdot c) = (a \cdot b) \cdot c \quad (6.3.1)$$

- (3) There is an *identity element*,  $e$ , in the set such that

$$a \cdot e = e \cdot a = a; \quad \text{all } a \text{ in the set} \quad (6.3.2)$$

- (4) For each element  $a$ , there is an *inverse element*  $a^{-1}$  in the set satisfying

$$a \cdot a^{-1} = a^{-1} \cdot a = e \quad (6.3.3)$$

*An Abelian (or commutative) group is defined as a group for which the commutative law is also satisfied,*

$$a \cdot b = b \cdot a; \quad \text{all } a, b \text{ in the set} \quad (6.3.4)$$

In what follows, we are interested primarily in Abelian groups.

The operation  $\cdot$  above acts very much like ordinary multiplication, and it is readily verified that the nonzero real numbers satisfy all the axioms of a group using ordinary multiplication as the operation. The operation in a group can equally well be addition, however, or any other arbitrarily defined operation. For example, the integers are a group using addition as the operation. In this case, the identity element is 0 and the inverse of  $a$  is  $-a$ . Similarly, the elements **0** and **1** form a group using modulo-2 addition as the operation. We shall follow the convention that whenever the group operation is denoted by  $+$  or  $\oplus$ , the identity will be denoted by **0** and the inverse of an element  $a$  by  $-a$ .

\* This is not the smallest set of axioms that could be used (see, for example, Birkhoff and MacLane (1941)), but it is the most useful set here.

The following properties are quite useful in the manipulation of groups. Let  $e$  be the identity of (6.3.2),  $a$  be an arbitrary element of the group, and  $a^{-1}$  be the inverse of  $a$  in (6.3.3). The first two properties below assert the uniqueness of  $e$  and  $a^{-1}$ .

$$(1) \text{ The only element } x \text{ for which } a \cdot x = a \text{ is } e \quad (6.3.5)$$

$$(2) \text{ The only } x \text{ for which } a \cdot x = e \text{ is } a^{-1} \quad (6.3.6)$$

$$(3) \text{ If } a \cdot b = a \cdot c, \text{ then } b = c \quad (6.3.7)$$

$$(4) \text{ The equation } a \cdot x = b \text{ has the unique solution } x = a^{-1} \cdot b \quad (6.3.8)$$

To verify (1), we multiply both sides of  $a \cdot x = a$  by  $a^{-1}$ , obtaining  $a^{-1} \cdot a \cdot x = a^{-1} \cdot a$ , or  $e \cdot x = e$ , or  $x = e$ . Properties (2), (3), and (4) are verified in the same way.

### **Subgroups**

A subset  $S$  of the elements in a group  $G$  is called a *subgroup* of  $G$  if the subset satisfies the axioms of a group using the same operation as the group  $G$ . As an example, consider the set of all binary sequences of length  $N$ . This set forms a group under the operation of modulo-2 addition of sequences. The identity is the **0** sequence, and each sequence is its own inverse. The code words of any given parity-check code of block length  $N$  form a subgroup of this group. To verify this, we observe that **0** is always a code word, the inverse of a code word is itself and the modulo-2 sum of any two code words is another code word.

By the *order* of a group or subgroup is meant the number of elements in the group or subgroup. A fundamental result is now given by Lagrange's theorem.

**Theorem 6.3.1 (Lagrange).** The order of a group, if finite, is a multiple of the order of each subgroup.

---

Before proving this, we need to build up a few subsidiary results. A *right coset* (left coset) of a subgroup  $S$  of a group  $G$  is the subset of elements of  $G$  formed by taking any fixed element  $a$  in  $G$  and forming all multiples,  $s_1 \cdot a$ ,  $s_2 \cdot a, \dots$  ( $a \cdot s_1, a \cdot s_2, \dots$ ) where when  $s_1, s_2, \dots$  are all the elements of  $S$ . We see, immediately, that for a subgroup of finite order, the number of elements in a coset is the same as the number of elements in the subgroup because, if  $s_i \neq s_j$ , then  $s_i \cdot a \neq s_j \cdot a$ . It also turns out that, if two right cosets (left cosets) of the same subgroup have any elements in common, they must be identical subsets. To see this, suppose that one coset is generated by the element  $a$  and another by the element  $b$ . If  $s_i \cdot a = s_j \cdot b$ , then  $s_j^{-1} \cdot s_i \cdot a = b$  and  $b$  is in the coset generated by  $a$ . Thus, for any  $s_n$  in  $S$ ,  $s_n \cdot b = s_n \cdot s_j^{-1} \cdot s_i \cdot a$ , and every element in the coset generated by  $b$  is in the coset generated by  $a$ .

For a parity-check code, we have already seen that, if  $\mathbf{y}$  is a fixed sequence,  $\mathbf{y} \oplus \mathbf{x}$  has the same syndrome for each code word  $\mathbf{x}$ . Thus the set of sequences with a given syndrome is a coset of the subgroup of code words. We can reformulate the maximum-likelihood decoding rule for a binary symmetric channel then as: given  $\mathbf{y}$ , assume the error sequence  $\mathbf{z}$  to be the minimum weight sequence in the same coset as  $\mathbf{y}$ .

We can now prove Theorem 6.3.1. Suppose that a group  $G$  of order  $n$  contains a subgroup  $S$  of order  $m$ . If  $n > m$ , take an element of  $G$  not in  $S$  and form a right coset. The subgroup and coset together contain  $2m$  elements, and if  $n > 2m$ , we can take another element of  $G$  in neither the subgroup or coset and form another right coset, giving us altogether  $3m$  elements. Proceeding in this way, we eventually come to the point where all elements of  $G$  are in  $S$  or one of the cosets, and if there are  $u$  cosets aside from the subgroup we have  $n = m(u + 1)$ , completing the proof. |

### Cyclic Subgroups

Let  $a$  be an element of a finite group  $G$  and consider the sequence of elements

$$a, a^2, a^3, \dots \quad (6.3.9)$$

where by  $a^2$  we mean  $a \cdot a$ , by  $a^3$  we mean  $a \cdot a \cdot a$ , and so on. Since the group is finite, there must be two powers,  $i$  and  $j$ ,  $j > i$ , for which

$$a^i = a^j = a^i \cdot a^{j-i} \quad (6.3.10)$$

From (6.3.5), this implies that  $a^{j-i} = e$ . The *order* of an element  $a$  in a group is defined as the smallest positive integer  $m$  for which  $a^m = e$ . The above argument shows that every element of a finite group has a finite order. Moreover, the elements  $a, a^2, \dots, a^m = e$  must all be distinct, for  $a^i = a^j$  with  $j > i$  only for  $j - i \geq m$ . Thus the sequence of powers of  $a$  in (6.3.9) has the following cyclic behavior.

$$a, a^2, \dots, a^m = e, a, a^2, \dots, a^m = e, a, \dots \quad (6.3.11)$$

It follows from (6.3.11) that  $a^n = e$  iff  $n$  is a multiple of  $m$ .

**Theorem 6.3.2.** Let an element  $a$  of a finite group  $G$  have order  $m$ . Then the elements  $a, a^2, \dots, a^m$  form a subgroup of  $G$ , and  $m$  divides the order of  $G$ .

*Proof.* The subset  $a, a^2, \dots, a^m$  contains the identity,  $e = a^m$ . Also, for each  $a^i$  in the subset,  $a^i \cdot a^{m-i} = a^m = e$ , so that each element in the subset has an inverse in the subset. Finally, we must show that  $a^i \cdot a^j$  is in the subset if both  $a^i$  and  $a^j$  are. We have  $a^i \cdot a^j = a^{i+j}$  and, if  $i + j \leq m$ ,  $a^i \cdot a^j$  is in the

subset. If  $i + j > m$ , we have  $a^{i+j} = a^m \cdot a^{i+j-m} = a^{i+j-m}$ . Since  $i + j - m \leq m$ ,  $a_i \cdot a_j$  is in the subset. Thus the subset is a subgroup of order  $m$ . From Theorem 6.3.1,  $m$  divides the order of  $G$ . |

A group or subgroup is called *cyclic* if there is some element in the group or subgroup whose multiples constitute the whole group or subgroup. Thus the subgroups of Theorem 6.3.2 are cyclic subgroups.

The following results concerning the order of elements in a finite Abelian group will be useful in Section 6.6.

**LEMMA.** Let  $a$  be an element of order  $m$  and  $b$  be an element of order  $n$  in an Abelian group. Then, if  $m$  and  $n$  are relatively prime,\* the order of  $a \cdot b$  is given by  $mn$ .

*Proof.*

$$(a \cdot b)^{mn} = (a^m)^n \cdot (b^n)^m = e \quad (6.3.12)$$

Thus, letting  $l$  denote the order of  $a \cdot b$ , we have  $l \leq mn$ . Also,

$$e = (a \cdot b)^l = a^{ln} \cdot (b^n)^l = a^{ln} \quad (6.3.13)$$

From this, it follows that  $ln$  is a multiple of the order,  $m$ , of  $a$ . Since  $m$  and  $n$  are relatively prime,  $l$  is a multiple of  $m$ . Reversing the roles of  $m$  and  $n$  in the above argument,  $l$  is also a multiple of  $n$ , and since  $m$  and  $n$  are relatively prime,  $l \geq mn$ , completing the proof. |

**Theorem 6.3.3.** In a finite Abelian group, let  $m$  be the maximum of the orders of the elements. Then  $m$  is a multiple of the order of each element in the group.

*Proof.* Let  $a$  be an element of maximal order  $m$  and let  $n$  be the order of any other element  $b$ . Let  $p_1, p_2, \dots, p_r$  be all the prime numbers that divide either  $m$  or  $n$ , and represent  $m$  and  $n$  as

$$m = p_1^{m_1} p_2^{m_2} \cdots p_r^{m_r} \quad (6.3.14)$$

$$n = p_1^{n_1} p_2^{n_2} \cdots p_r^{n_r} \quad (6.3.15)$$

where  $m_1, \dots, m_r$  and  $n_1, \dots, n_r$  are nonnegative integers. If  $m$  is not a multiple of  $n$ , then for some  $i$ ,  $1 \leq i \leq r$ , we must have  $n_i > m_i$ . Now, for each  $j$ , let  $a_j$  be an element of order  $p_j^{m_j}$  (such an element is given by  $a$  to the power  $m/p_j^{m_j}$ ). Likewise, let  $b_i$  be an element of order  $p_i^{n_i}$ . Now consider the element  $c = a_1 \cdot a_2 \cdots a_{i-1} \cdot b_i \cdot a_{i+1} \cdots a_r$ . Applying the previous lemma successively as each new component is multiplied in, we see that  $c$  has order  $mp_i^{n_i}/p_i^{m_i} > m$ . This is a contradiction since  $m$  is the maximum of the element orders, and thus  $m$  is a multiple of  $n$ . |

\* Two positive integers are relatively prime if their greater common divisor is 1, that is, if no positive integer greater than 1 is a factor of both the given integers.

## 6.4 Fields and Polynomials

A considerable portion of algebraic coding theory is built around the theory of finite fields. Loosely, a field is a set of elements on which addition, subtraction, multiplication, and division can be performed using the ordinary rules of arithmetic. *More precisely, a field is a set of at least two elements, closed\* under two operations called addition (+) and multiplication (·), and satisfying the following axioms.*

- (1) The set of elements is an Abelian group under addition.
- (2) The set of nonzero elements (where  $\mathbf{0}$  is the identity of the group under addition) is an Abelian group under multiplication.
- (3) The distributive law is satisfied:

$$(a + b) \cdot c = a \cdot c + b \cdot c; \quad \text{all } a, b, c \quad (6.4.1)$$

It is immediately verified that the set of real numbers, using ordinary addition and multiplication, satisfies all these axioms. Also the set of binary elements,  $\mathbf{0}$  and  $\mathbf{1}$ , using modulo-2 addition and ordinary multiplication, satisfies the axioms. The set of integers, however, does not satisfy the axioms since any integer greater than 1 has no multiplicative inverse that is an integer.

We shall always denote the identity of the group under addition by  $\mathbf{0}$  and that under multiplication by  $\mathbf{1}$ . We shall denote the additive inverse and multiplicative inverse by  $-a$  and  $a^{-1}$ , respectively.

Some elementary properties of fields are given by

$$a \cdot \mathbf{0} = \mathbf{0} \cdot a = \mathbf{0}; \quad \text{all } a \quad (6.4.2)$$

$$a \cdot b \neq \mathbf{0}; \quad \text{all nonzero } a, b \quad (6.4.3)$$

$$-(a \cdot b) = (-a) \cdot b = a \cdot (-b); \quad \text{all } a, b \quad (6.4.4)$$

$$a \cdot b = a \cdot b' \Rightarrow b = b' \quad \text{for } a \neq \mathbf{0} \quad (6.4.5)$$

To verify (6.4.2), let  $a, b$  be arbitrary and observe that  $a \cdot b = a \cdot (b + \mathbf{0}) = a \cdot b + a \cdot \mathbf{0}$ . Thus  $a \cdot \mathbf{0}$  is the identity element of the additive group [see (6.3.5)], and  $a \cdot \mathbf{0} = \mathbf{0}$ . Equation 6.4.3 simply states that the nonzero elements are closed under multiplication; this follows from axiom 2. To verify (6.4.4), we have  $\mathbf{0} = \mathbf{0} \cdot b = [a + (-a)] \cdot b = a \cdot b + (-a) \cdot b$ . Thus  $(-a) \cdot b$  is the additive inverse of  $a \cdot b$ . The second half of (6.4.4) follows from a similar argument. Because of (6.4.4), we can use minus signs unambiguously without parentheses, and shall henceforth do so. To verify the cancellation law (6.4.5), observe that  $a \cdot b = a \cdot b'$  implies that  $\mathbf{0} = a \cdot b - a \cdot b' = a \cdot (b - b')$ . Since  $a \neq \mathbf{0}$ , this implies that  $b - b' = \mathbf{0}$  and  $b = b'$ .

\* A set of elements is closed under an operation  $\cdot$  if for every pair  $(a, b)$  of elements in the set,  $a \cdot b$  is also in the set.

We shall be interested here solely in *Galois fields*, which by definition are fields with a finite number of elements. The following theorem is often helpful in determining whether a finite set of elements forms a field.

**Theorem 6.4.1.** For a finite set of elements, axiom 2 in the above definition of a field can be replaced by the weaker condition (2'): the multiplication operation is commutative and associative and (6.4.3) is satisfied.

---

*Proof.* We observe that (6.4.2), (6.4.4), and (6.4.5) are still valid since their proofs did not rest on axiom 2. Let  $a$  be a nonzero element of the set, and consider the sequence  $a, a^2, a^3, \dots$  where  $a^2 = a \cdot a$ ,  $a^3 = a \cdot a \cdot a$ , etc. Since the set is finite there must be two integers,  $j > i$ , for which

$$a^i = a^j = a^i \cdot a^{j-i} \quad (6.4.6)$$

We now show that  $a^{j-i}$  is the multiplicative identity. Multiplying both sides of (6.4.6) by an arbitrary element  $b$ , and cancelling  $a^i$  (which is nonzero), we obtain

$$b = b \cdot a^{j-i}$$

Thus  $a^{j-i}$  is the multiplicative identity. Finally, the multiplicative inverse of  $a$  is given by  $a^{j-i-1}$  (or  $a$  if  $j = i + 1$ ). Since  $a$  is arbitrary, every nonzero element has an inverse and axiom 2 is satisfied. |

As an example of the use of this theorem, let  $p$  be a prime number, and consider the set of integers  $\mathbf{0}, \mathbf{1}, \dots, \mathbf{i}$  where  $i = p - 1$ . Define addition and multiplication to be modulo  $p$  [that is, the field element  $\mathbf{i} + \mathbf{j}$  is given by  $R_p(i + j)$ , by which we mean the remainder after dividing the usual arithmetic sum  $i + j$  by  $p$ ]. The additive inverse of an element  $\mathbf{i}$  is the element corresponding to  $p - i$ . The existence of the multiplicative inverse is not entirely obvious, but is implied by Theorem 6.4.1. Thus, for any prime  $p$ , the nonnegative integers less than  $p$  form a field using modulo- $p$  arithmetic. If  $p$  is not prime, these elements cannot form a field using modulo- $p$  arithmetic, since there are two nonzero elements whose ordinary product is  $p$  and, thus, whose modulo- $p$  product is zero. We shall see later that there are fields with  $p^n$  elements where  $p$  is prime and  $n$  is an integer greater than 1, but the addition and multiplication rules are not modulo- $p^n$  rules. As with groups, the *order* of a Galois field is the number of elements in the field, and we shall denote any particular field with  $q$  elements as  $GF(q)$ .

### Polynomials

An expression of the form  $f_n D^n + f_{n-1} D^{n-1} + \dots + f_0$ , denoted  $f(D)$ , is called a *polynomial* over  $GF(q)$  of degree  $n$  if the coefficients  $f_n, \dots, f_0$  are all elements of  $GF(q)$  and if the leading coefficient,  $f_n$ , is nonzero. It is

convenient to consider the coefficient  $f_i$  associated with an  $n$ -degree polynomial to be defined for all  $i \geq 0$ , but to satisfy  $f_i = \mathbf{0}$  for  $i > n$ . Thus we can consider a polynomial over  $GF(q)$  as being a way to represent an infinite sequence of field elements  $f_0, f_1, \dots$ , when only finitely many terms in the sequence are nonzero. The degree of the polynomial is then the largest  $n$  for which  $f_n \neq 0$ . In the special case of the  $\mathbf{0}$  polynomial,  $f_n = \mathbf{0}$  for all  $n \geq 0$ , but by convention we say that the  $\mathbf{0}$  polynomial has degree 0.

The symbol  $D$  in a polynomial  $f(D)$  is called an *indeterminate*. It is *not* to be interpreted as a variable or unknown element of the field, partly because we shall later occasionally substitute an element not in the original field for  $D$ , and partly because we are more concerned with the sequence of coefficients defined by a polynomial than with its functional aspects. At any rate, we shall define rules for manipulating polynomials, and once these are known, the question of what an indeterminate “is” becomes irrelevant.

Two polynomials are said to be equal if they each correspond to the same sequence of coefficients. For example, let  $f(D) = D^3 + D^2 + D$  and  $g(D) = D$  be polynomials over the modulo-2 field (by convention, in writing polynomials, we omit terms with  $\mathbf{0}$  coefficient and write  $\mathbf{1}D^i$  as  $D^i$ ). By our definition, these polynomials are unequal. On the other hand, if we substitute the field elements  $\mathbf{0}$  and  $\mathbf{1}$  for  $D$  above, we find that  $f(\mathbf{0}) = \mathbf{0}$ ,  $f(\mathbf{1}) = \mathbf{1}$  and  $g(\mathbf{0}) = \mathbf{0}$ ,  $g(\mathbf{1}) = \mathbf{1}$ . Thus, as functions of a variable in the modulo-2 field,  $f(D)$  and  $g(D)$  are equal, even though as polynomials they are unequal.

The sum of two polynomials over a given field is another polynomial over that field defined by the familiar rule

$$f(D) + g(D) = \sum_{i=0}^{\infty} (f_i + g_i) D^i \quad (6.4.7)$$

The degree of  $f(D) + g(D)$  is the largest  $n$  for which  $f_n + g_n \neq 0$ , and is, at most, the maximum of the degree of  $f(D)$  and that of  $g(D)$ . As an example, over the modulo-2 field,

$$(D^2 + D + \mathbf{1}) + (D^2 + \mathbf{1}) = (\mathbf{1} \oplus \mathbf{1}) D^2 + D + (\mathbf{1} \ominus \mathbf{1}) = D$$

The product of two polynomials over a given field is another polynomial over that field defined by

$$f(D)g(D) = \sum_i \left( \sum_{j=0}^i f_j g_{i-j} \right) D^i \quad (6.4.8)$$

It can be verified by inspection of (6.4.8) that, for  $g(D) = \mathbf{0}$ ,

$$f(D)\mathbf{0} = \mathbf{0}; \quad \text{all } f(D) \quad (6.4.9)$$

Also

$$f(D)g(D) \neq \mathbf{0} \quad \text{for } f(D) \neq \mathbf{0}, g(D) \neq \mathbf{0} \quad (6.4.10)$$

To see this, suppose that  $f(D)$  has degree  $n$ ,  $f_n \neq 0$  and  $g(D)$  has degree  $m$ ,  $f_m \neq 0$ . Then, from (6.4.8),  $f(D)g(D)$  has degree  $n + m$ , with the leading term being  $f_n g_m D^{n+m}$ .

The multiplication of a polynomial  $f(D)$  over a field by an element  $\alpha$  of the field is defined by

$$\alpha f(D) = \sum_i (\alpha f_i) D^i.$$

Similarly, the negative of a polynomial is defined by

$$-f(D) = \sum_i (-f_i) D^i.$$

It is easy to verify that, under addition, the set of polynomials over a field forms an Abelian group. It can also be seen that polynomial multiplication is both associative and commutative and that the distributive law,  $[f(D) + g(D)]h(D) = f(D)h(D) + g(D)h(D)$ , holds. The set of polynomials over a field is not a field, however, because of the lack of a multiplicative inverse. This gives an example in which Theorem 6.4.1 is false without the restriction to finite sets.

Some elementary properties of polynomials that will be useful later are given by:

$$-[f(D)g(D)] = [-f(D)]g(D) = f(D)[-g(D)] \quad (6.4.11)$$

$$f(D)g(D) = f(D)h(D) \Rightarrow g(D) = h(D) \quad \text{for } f(D) \neq 0 \quad (6.4.12)$$

The proofs of (6.4.11) and (6.4.12) are the same as (6.4.4) and (6.4.5).

Although, in general, we cannot divide one polynomial by another and get a polynomial quotient, we can perform division if we are willing to tolerate a remainder term.

**Theorem 6.4.2 (Euclidean Division Algorithm).** Let  $f(D)$  and  $g(D)$  be polynomials over  $GF(q)$  and let  $g(D)$  have degree at least 1. Then there exist unique polynomials  $h(D)$  and  $r(D)$  over  $GF(q)$  for which

$$f(D) = g(D)h(D) + r(D) \quad (6.4.13)$$

where the degree of  $r(D)$  is less than that of  $g(D)$ .

---

*Proof.* We first show how to find  $h(D)$  and  $r(D)$  to satisfy (6.4.13), and then show that the solution is unique. Let  $f(D)$  have degree  $n$  and let  $g(D)$  have degree  $m$ . If  $n < m$ , we set  $h(D) = 0$ ,  $r(D) = f(D)$ . If  $n \geq m$ , we divide  $g(D)$  into  $f(D)$  by the same procedure that we use for ordinary polynomials over the real-number field, letting  $h(D)$  be the quotient and  $r(D)$  the remainder. That is, the leading term of  $h(D)$  is  $f_n g_m^{-1} D^{n-m}$ . This leading term times  $g(D)$  is subtracted from  $f(D)$ , and the next term of  $h(D)$  is found by starting to

divide  $g(D)$  into this remainder. When the remainder has a smaller degree than  $g(D)$ , it is taken as  $r(D)$ . As an example, let  $g(D) = D^2 + D + 1$  and  $f(D) = D^3 + D + 1$  be polynomials over the modulo-2 field.

$$\begin{array}{r} D + 1 \\ D^2 + D + 1 \) D^3 + \end{array} \begin{array}{r} D + 1 \\ D^3 + D^2 + D \\ \hline D^2 \end{array} \begin{array}{r} + 1 \\ D^2 + D + 1 \\ \hline D \end{array}$$

Thus  $h(D) = D + 1$  and  $r(D) = D$  for this example.

We now suppose that there are two solutions to (6.4.13), given by  $h(D)$ ,  $r(D)$  and  $h'(D)$ ,  $r'(D)$ . Then

$$g(D)h(D) + r(D) = g(D)h'(D) + r'(D) \quad (6.4.14)$$

$$g(D)[h(D) - h'(D)] = r'(D) - r(D) \quad (6.4.15)$$

Now  $r'(D) - r(D)$  has a degree less than that of  $g(D)$ , and thus  $h(D) - h'(D) = 0$ . But this implies that  $r'(D) - r(D) = 0$ , and it follows that the solution is unique. |

In dealing with finite fields, we are often far more interested in the remainder  $r(D)$  than in the quotient  $h(D)$  in (6.4.13). *In analogy with arithmetic modulo a prime number, we define the remainder of a polynomial  $f(D)$  modulo a polynomial  $g(D)$ , denoted  $R_{g(D)}[f(D)]$ , as the remainder when  $f(D)$  is divided by  $g(D)$*

$$R_{g(D)}[f(D)] = r(D); \quad r(D) \text{ as in (6.4.13)} \quad (6.4.16)$$

We can use the Euclidean division algorithm to investigate the existence of factors and roots of a polynomial over  $GF(q)$ . *A polynomial  $g(D)$  is a factor of (or divides) another polynomial  $f(D)$  if there is a polynomial  $h(D)$  over the field satisfying*

$$f(D) = g(D)h(D) \quad (6.4.17)$$

In other words,  $g(D)$  divides  $f(D)$  if the Euclidean division algorithm, (6.4.13), yields  $r(D) = 0$ . *A polynomial  $f(D)$  is called reducible if there are two polynomials  $g(D)$  and  $h(D)$  over the field, each of degree at least 1, which satisfy (6.4.17). A polynomial is irreducible if it is not reducible.*

It is often useful to factor a polynomial into irreducible factors. In doing this, there is always a certain amount of ambiguity, since given one factorization, we can always multiply one factor by an arbitrary field element and multiply another factor by the inverse of that field element, not changing the

product. A *monic polynomial* is defined as a polynomial whose leading nonzero coefficient is  $\mathbf{1}$ , and we avoid the above ambiguity by factoring a polynomial into a field element times a product of monic irreducible factors.

**Theorem 6.4.3 (Unique Factorization).** A polynomial  $f(D)$  over a given field has a unique factorization into a field element times a product of monic irreducible polynomials over the field, each of degree at least 1. 

---

*Proof.* Clearly, the field element is unique and is just  $f_n$  where  $n$  is the degree of  $f(D)$ . Thus we can restrict our attention to monic polynomials. Assume that the theorem is not true and that there is some lowest degree monic polynomial  $f(D)$  that can be factored in two ways, as

$$a_1(D)a_2(D) \cdots a_k(D) = b_1(D) \cdots b_j(D) \quad (6.4.18)$$

where the  $a_k(D)$  and  $b_j(D)$  are monic and irreducible.

All of the  $a_k(D)$  must be different from all the  $b_j(D)$ , or a polynomial could be factored out, leaving two factorizations for a lower degree monic polynomial than  $f(D)$ . Suppose, without loss of generality, that the degree of  $b_1(D)$  is less than or equal to that of  $a_1(D)$ . We then have

$$a_1(D) = b_1(D)h(D) + r(D) \quad (6.4.19)$$

where  $r(D)$  has degree less than  $b_1(D)$  and less than  $a_1(D)$ . Substituting (6.4.19) into (6.4.18), we have

$$r(D)a_2(D) \cdots a_k(D) = b_1(D)[b_2(D) \cdots b_j(D) - h(D)a_2(D) \cdots a_k(D)]$$

Factoring  $r(D)$  and multiplying by a field element to make the factors monic, we have two factorizations of a lower degree monic polynomial than  $f(D)$ , and the irreducible polynomial  $b_1(D)$  does not appear as one of the irreducible factors on the left. Thus we have a contradiction and the theorem is true. |

An element  $\alpha$  of a field is defined to be a *root* of a polynomial  $f(D)$  over that field if  $f(\alpha) = \mathbf{0}$ .

**Theorem 6.4.4.** An element  $\alpha$  in a field is a root of a nonzero polynomial  $f(D)$  over that field iff  $(D - \alpha)$  is a factor of  $f(D)$ . Furthermore, if  $f(D)$  has degree  $n$ , at most  $n$  field elements are roots of  $f(D)$ . 

---

*Proof.* From the Euclidean division algorithm, we have

$$f(D) = (D - \alpha)h(D) + r(D) \quad (6.4.20)$$

Since  $D - \alpha$  has degree 1,  $r(D)$  has degree 0 and is just a field element  $r_0$ . Substituting  $\alpha$  for  $D$ , we have  $f(\alpha) = r_0$ . Thus, if  $f(\alpha) = \mathbf{0}$ ,  $r_0 = \mathbf{0}$ , and  $(D - \alpha)$  is a factor of  $f(D)$ . Conversely, if  $(D - \alpha)$  is a factor of  $f(D)$ , we have  $f(D) = (D - \alpha)h(D)$  and  $f(\alpha) = \mathbf{0}$ . Now, factor  $f(D)$  into a field element times a product of irreducible factors of degree at least 1. Since the

degree of  $f(D)$  is the sum of the degrees of the factors, there are at most  $n$  factors. These are unique, and hence  $f(D)$  has at most  $n$  roots in the field. |

These results concerning polynomials can now be used to construct another example of a finite field. This example is more important than it appears, since we shall see later that it gives us a concrete representation for *any* finite field.

Let  $f(D)$  be an irreducible polynomial of degree  $n$  over a finite field  $GF(q)$ , and consider the set of all polynomials of degree  $n - 1$  or less over  $GF(q)$ . Define the operation  $*$  among these polynomials as the remainder of the polynomial product modulo  $f(D)$ ,

$$g_1(D) * g_2(D) = R_{f(D)}[g_1(D)g_2(D)] \quad (6.4.21)$$

We now show that, under polynomial addition and  $*$  multiplication, the set of polynomials  $g(t)$  over  $GF(q)$  of degree  $n - 1$  or less form a field (the indeterminate will be denoted  $t$  here as a mnemonic aid to remember that we are dealing with a special set of polynomials, those of degree  $n - 1$  or less, and with  $*$  multiplication rather than polynomial multiplication). To verify that we have a field, observe that axioms 1 and 3 follow immediately from the properties of polynomial addition and multiplication. Next observe that, if  $g_1(t) * g_2(t) = \mathbf{0}$ , then

$$g_1(D)g_2(D) = f(D)h(D) \quad (6.4.22)$$

Since  $f(D)$  is irreducible, however, and since  $g_1(D)$  and  $g_2(D)$  each have lower degree than  $f(D)$ , the unique factorization theorem indicates that either  $g_1(D) = \mathbf{0}$  or  $g_2(D) = \mathbf{0}$ . Thus axiom 2' of Theorem 6.4.1 is satisfied and we have a field. Since each of the coefficients  $g_0, g_1, \dots, g_{n-1}$  can be any of the  $q$  original field elements, the new field contains  $q^n$  elements. *We shall call this new field the field of polynomials over  $GF(q)$  modulo  $f(D)$ .* It is necessary for  $f(D)$  to be irreducible here, since otherwise we would have  $f(D) = g_1(D)g_2(D)$  for some nonzero choice of  $g_1$  and  $g_2$ , and  $g_1(t)*g_2(t) = \mathbf{0}$ , violating (6.4.3).

As an example, let  $f(D) = D^2 + D + 1$  be a polynomial over  $GF(2)$ . Then the elements of the field modulo  $f(D)$  are given by  $\mathbf{0}, \mathbf{1}, t, t + 1$ . The addition and  $*$  multiplication tables are listed in Figure 6.4.1. To find  $t * t$ , for example, we use the Euclidean division algorithm to obtain  $D^2 = (D^2 + D + 1)\mathbf{1} + (D + 1)$ . Thus  $R_{f(D)}[D^2] = D + 1$  and  $t * t = t + 1$ .

## 6.5 Cyclic Codes

In this section, we shall consider a special class of parity-check codes known as cyclic codes. Such codes have two advantages over ordinary parity-check codes: first, the encoding operation is easier to implement;

| <b>0</b>                | <b>1</b>                | <b><math>t</math></b>   | <b><math>t+1</math></b> | <b>0</b>                | <b>1</b> | <b><math>t</math></b>   | <b><math>t+1</math></b> |
|-------------------------|-------------------------|-------------------------|-------------------------|-------------------------|----------|-------------------------|-------------------------|
| <b>0</b>                | <b>0</b>                | <b>1</b>                | <b><math>t</math></b>   | <b><math>t+1</math></b> | <b>0</b> | <b>0</b>                | <b>0</b>                |
| <b>1</b>                | <b>1</b>                | <b>0</b>                | <b><math>t+1</math></b> | <b><math>t</math></b>   | <b>0</b> | <b>1</b>                | <b><math>t</math></b>   |
| <b><math>t</math></b>   | <b><math>t</math></b>   | <b><math>t+1</math></b> | <b>0</b>                | <b>1</b>                | <b>0</b> | <b><math>t</math></b>   | <b><math>t+1</math></b> |
| <b><math>t+1</math></b> | <b><math>t+1</math></b> | <b><math>t</math></b>   | <b>1</b>                | <b>0</b>                | <b>0</b> | <b><math>t+1</math></b> | <b><math>t</math></b>   |

Addition

| <b>0</b>                | <b>1</b> | <b><math>t</math></b>   | <b><math>t+1</math></b> | <b>0</b>                | <b>1</b> | <b><math>t</math></b>   | <b><math>t+1</math></b> |
|-------------------------|----------|-------------------------|-------------------------|-------------------------|----------|-------------------------|-------------------------|
| <b>0</b>                | <b>0</b> | <b>0</b>                | <b>0</b>                | <b>0</b>                | <b>0</b> | <b>0</b>                | <b>0</b>                |
| <b>1</b>                | <b>0</b> | <b>1</b>                | <b><math>t</math></b>   | <b><math>t+1</math></b> | <b>0</b> | <b>1</b>                | <b><math>t</math></b>   |
| <b><math>t</math></b>   | <b>0</b> | <b><math>t</math></b>   | <b><math>t+1</math></b> | <b>1</b>                | <b>0</b> | <b><math>t</math></b>   | <b><math>t+1</math></b> |
| <b><math>t+1</math></b> | <b>0</b> | <b><math>t+1</math></b> | <b>1</b>                | <b>0</b>                | <b>0</b> | <b><math>t+1</math></b> | <b><math>t</math></b>   |

\*Multiplication

*Figure 6.4.1. Field of polynomials over GF(2) modulo  $D^2 + D + 1$ .*

and, second, the large amount of mathematical structure in the code makes it possible to find various simple decoding algorithms.

Before defining a cyclic code, we shall generalize parity-check codes to nonbinary alphabets. These generalizations will be called *linear* codes or *group* codes since the word parity is hardly appropriate for nonbinary alphabets. Let  $\mathbf{u} = (u_1, u_2, \dots, u_L)$  be an arbitrary sequence of information digits with each  $u_i$  an element of some finite field,  $GF(q)$ . An  $(N, L)$  linear code is a code in which the code word  $\mathbf{x} = (x_1, \dots, x_N)$  corresponding to each  $\mathbf{u}$  is a sequence of  $N > L$  letters from  $GF(q)$  generated by the rule

$$x_n = \sum_{l=1}^L u_l g_{l,n}; \quad 1 \leq n \leq N \quad (6.5.1)$$

where the elements  $g_{l,n}$  are arbitrarily chosen elements of  $GF(q)$  and the addition and multiplication are the operations in  $GF(q)$ . As before, we can represent the elements  $g_{l,n}$  by a generator matrix  $G$  as in Figure 6.1.2a and the code words are generated by

$$\mathbf{x} = \mathbf{u}G \quad (6.5.2)$$

If we multiply both sides of (6.5.2) by an arbitrary element,  $\alpha$  of  $GF(q)$ , we see that, if  $\mathbf{x}$  is the code word corresponding to  $\mathbf{u}$ , then  $\alpha\mathbf{x} = (\alpha x_1, \alpha x_2, \dots, \alpha x_N)$  is the code word corresponding to  $\alpha\mathbf{u}$ . Similarly, if  $\mathbf{x}_1$  and  $\mathbf{x}_2$  are the code words corresponding to  $\mathbf{u}_1$  and  $\mathbf{u}_2$ , respectively, then  $\mathbf{x}_1 + \mathbf{x}_2$  is the code word corresponding to  $\mathbf{u}_1 + \mathbf{u}_2$ . In mathematical terminology, the mapping from information sequences to code words is linear and the set of code words is a linear subspace of the space of  $N$ -sequences over  $GF(q)$ .

A systematic linear code is a linear code in which the first  $L$  components of each code word satisfy

$$x_l = u_l; \quad 1 \leq l \leq L \quad (6.5.3)$$

This is achieved by setting  $g_{l,n} = 1$  for  $l = n$  and  $g_{l,n} = 0$  for  $n \leq L$ ,  $n \neq l$ . The generator matrix for a systematic linear code is as shown in Figure 6.1.2b.

The check matrix  $H$  associated with a systematic linear code must be modified somewhat from Figure 6.1.3, since in going from (6.1.9) to (6.1.10), we get

$$0 = \sum_{l=1}^L x_l g_{l,n} - x_n \quad (6.5.4)$$

Thus  $H$  is given in Figure 6.5.1, and all the code words satisfy

$$\mathbf{x}H = 0 \quad (6.5.5)$$

Aside from this one change, the results of Sections 6.1 and 6.2 can be carried over to this generalization. For example, if these code words are transmitted over a channel whose input and output alphabets are the letters of  $GF(q)$ , then we can represent the received sequence by  $\mathbf{y}$  and the error sequence by  $\mathbf{z}$ , where  $\mathbf{y} = \mathbf{x} + \mathbf{z}$ . Defining the syndrome  $\mathbf{S}$  by  $\mathbf{S} = \mathbf{y}H$ , we have as before  $\mathbf{S} = \mathbf{y}H = \mathbf{z}H$ , and a decoding table can be constructed as before to find  $\mathbf{z}$  from  $\mathbf{S}$ .

*A cyclic code over  $GF(q)$  is a linear code with the special property that any cyclic shift of a code word is another code word.* That is, if  $(x_1, \dots, x_N)$  is any code word,  $(x_2, x_3, \dots, x_N, x_1)$  is another code word. It is more convenient, in discussing cyclic codes, to modify our notation somewhat, numbering successive elements backwards instead of forwards and going from  $N - 1$  down to 0. Thus we denote  $\mathbf{x}$  by  $(x_{N-1}, x_{N-2}, \dots, x_0)$ . Equivalently, we shall represent a sequence  $\mathbf{x}$  by the polynomial over  $GF(q)$ ,

$$x(D) = x_{N-1}D^{N-1} + x_{N-2}D^{N-2} + \dots + x_0 \quad (6.5.6)$$

If  $x(D)$  is a code word in a cyclic code (that is, if the coefficients form the letters of a code word), then it follows that the remainder of  $Dx(D)$  modulo  $D^N - 1$  is also a code word. To see this, we have

$$Dx(D) = x_{N-1}D^N + \dots + x_0D \quad (6.5.7)$$

$$Dx(D) = x_{N-1}(D^N - 1) + x_{N-2}D^{N-1} + \dots + x_0D + x_{N-1} \quad (6.5.8)$$

$$R_{D^N - 1}[Dx(D)] = x_{N-2}D^{N-1} + \dots + x_0D + x_{N-1} \quad (6.5.8)$$

Now let  $g(D)$  be the lowest degree monic polynomial which is a code word in a cyclic code and let  $m$  be the degree of  $g(D)$ . It follows immediately from the linearity that, if  $a_0$  is any element of  $GF(q)$ , then  $a_0g(D)$  is also a code word. Also, from the cyclic property,  $a_1Dg(D)$  is a code word for all  $a_1$  in

$$H = \left[ \begin{array}{ccccccccc} & & & & & & & & N-L \\ \hline g_{1,L+1} & \cdots & \cdots & \cdots & \cdots & \cdots & g_{1,N} & & \\ g_{2,L+1} & & & & & & g_{2,N} & & \\ \vdots & & & & & & \vdots & & \\ g_{L,L+1} & \cdots & \cdots & \cdots & \cdots & \cdots & g_{L,N} & & \\ \hline -1 & 0 & 0 & \cdots & 0 & & & & N \\ 0 & -1 & 0 & \cdots & 0 & & & & \\ \vdots & \vdots & \vdots & & \vdots & & \vdots & & \\ 0 & 0 & 0 & \cdots & -1 & & & & \end{array} \right]$$

Figure 6.5.1. Check matrix for systematic group code in  $GF(q)$ .

$GF(q)$ , and  $a_i D^i g(D)$  is a code word for  $i \leq N - 1 - m$ . Adding these code words together, we finally see that, for any polynomial  $a(D)$  in  $GF(q)$  with degree at most  $N - 1 - m$ ,  $a(D)g(D)$  is a code word.

We now show that *all* code words in a cyclic code have this form. From the Euclidean division algorithm, any  $x(D)$  of degree  $N - 1$  or less can be expressed as

$$x(D) = a(D)g(D) + r(D) \quad (6.5.9)$$

Since  $a(D)g(D)$  is a code word, it follows from the linearity that, if  $x(D)$  is a code word, then  $r(D)$  is also. But  $r(D)$  is of lower degree than  $g(D)$ , and if nonzero,  $r(D)$  can be multiplied by an element of  $GF(q)$  to get a monic polynomial which is a code word of lower degree than  $g(D)$ . This is impossible by the definition of  $g(D)$ , and hence  $r(D) = 0$ . Thus  $x(D)$  is a code word iff there is an  $a(D)$  with degree at most  $N - m - 1$  satisfying

$$x(D) = a(D)g(D) \quad (6.5.10)$$

Next, we show that  $g(D)$  must be a factor of  $D^N - 1$ . Since  $g(D)$  is monic and of degree  $m$ , we have

$$D^{N-m}g(D) = D^N - 1 + r(D) \quad (6.5.11)$$

Now  $r(D) = R_{D^N - 1}[D^{N-m}g(D)]$  and is thus a cyclic shift of a code word and must contain  $g(D)$  as a factor. Equation 6.5.11 then implies that  $D^N - 1$  contains  $g(D)$  as a factor.

The polynomial  $g(D)$  is called the generator polynomial of the cyclic code and has the significance that all code words contain  $g(D)$  as a factor. The set of code words is the set of linear combinations of  $g(D)$  and its first  $N - m - 1$  shifts. The generator matrix for a cyclic code can thus be represented in non-systematic form by Figure 6.5.2. In terms of the number of information digits in the code,  $L$ , we see that  $N - m = L$  and the degree of  $g(D)$  is  $N - L$ .

Now let us turn the problem around and let  $g(D)$  be any monic  $N - L$  degree polynomial in  $GF(q)$  that divides  $D^N - 1$ . We shall show that the code generated by  $g(D)$  [that is, the code whose code words are all linear combinations of  $g(D)$  and its first  $L - 1$  shifts] is a cyclic code. The code is clearly a linear code and has the generator matrix of Figure 6.5.2. Also, if  $x(D)$  is any code word, then

$$Dx(D) = x_{N-1}(D^N - 1) + x_{N-2}D^{N-1} + \cdots + x_0D + x_{N-1} \quad (6.5.12)$$

Since  $x(D)$  and  $D^N - 1$  are divisible by  $g(D)$ , it follows that  $x_{N-2}D^{N-1} + \cdots + x_0D + x_{N-1}$  is divisible by  $g(D)$  and any cyclic shift of a code word is another code word. These results are summarized by the following theorem.

**Theorem 6.5.1.** Any cyclic code over  $GF(q)$  with  $L$  information digits and a block length of  $N$  is generated by an  $N - L$  degree monic polynomial over

$$G = \begin{bmatrix} g_m & g_{m-1} & \cdots & g_0 \\ g_m & & & g_0 \\ g_m & & & g_0 \\ \vdots & \ddots & & \vdots \\ g_m & \cdots & \cdots & g_0 \end{bmatrix} \quad N-m$$

$g(t) = g_mt_m + \cdots + g_0$

Figure 6.5.2. Generator matrix for a cyclic code.

$GF(q)$ ,  $g(D)$ , which divides  $D^N - 1$ . Conversely, any  $N - L$  degree monic polynomial over  $GF(q)$  that divides  $D^N - 1$  generates a cyclic code with  $L$  information digits and a block length of  $N$ .

Just as the generator matrix of a cyclic code can be represented in terms of the  $N - L$  degree generator polynomial,  $g(D)$ , the check matrix can be represented in terms of an  $L$  degree polynomial,  $h(D)$ , known as the check polynomial, and given by

$$g(D)h(D) = D^N - 1 \quad (6.5.13)$$

If we multiply any code word,  $x(D) = a(D)g(D)$  by  $h(D)$ , we obtain

$$f(D) = x(D)h(D) = a(D)g(D)h(D) = D^N a(D) - a(D) \quad (6.5.14)$$

Since  $a(D)$  has degree at most  $L - 1$ , we see from (6.5.14) that  $f(D) = \sum f_n D^n$  must have zero terms for  $L \leq n \leq N - 1$ . Multiplying out  $x(D)h(D)$ , we then have

$$\sum_{i=0}^L h_i x_{n-i} = f_n = 0; \quad L \leq n \leq N - 1 \quad (6.5.15)$$

From (6.5.13), we see that  $h(D)$  is monic and  $h_L = 1$ . Thus we can rewrite (6.5.15):

$$x_{n-L} = - \sum_{i=0}^{L-1} h_i x_{n-i}; \quad L \leq n \leq N - 1 \quad (6.5.16)$$

Equation 6.5.16 gives us a recurrence relationship for calculating, in order, the check digits,  $x_{N-L-1}, x_{N-L-2}, \dots, x_0$  from the information digits,  $x_{N-1}, \dots, x_{N-L}$ . Thus, any  $x(D)$  that satisfies (6.5.15) is a code word and every code word satisfies (6.5.15). In matrix terminology, this says that  $\mathbf{x}H = 0$  if and only if  $\mathbf{x} = (x_{N-1}, \dots, x_0)$  is a code word where  $H$  is given in Figure 6.5.3.

Since  $h(D)$  divides  $D^N - 1$ , it also generates a cyclic code, called the *dual code* to that generated by  $g(D)$ . Thus all the cyclic shifts of  $h(D)$  are linear combinations of the  $N - L$  shifts shown in Figure 6.5.3, and it is often convenient to use these additional shifts as check relationships which the code words generated by  $g(D)$  must satisfy.

Since the discussion up to this point has been rather abstract, it is appropriate to discuss how the encoding for a cyclic code can actually be implemented. The first technique for encoding is suggested by (6.5.16) and is illustrated by Figure 6.5.4.

The information digits are initially stored in the stages of a shift register, in the positions shown in Figure 6.5.4. The contents of each stage are available at the output of the stage (that is, the line coming from the right side of the stage), and are multiplied by the appropriate  $h_i$ ; these products are summed and multiplied by  $-1$  to give  $x_{N-L-1}$ , as shown by (6.5.16). The shift register is then shifted one position to the right,  $x_{N-1}$  going out on the channel and  $x_{N-L-1}$  entering the left-hand stage. After the shift,  $x_{N-L-2}$  appears at the output of the  $-1$  multiplier. After shifting  $N$  times,  $x_0$  has been transmitted and the encoder is ready to have the next information sequence stored in it.

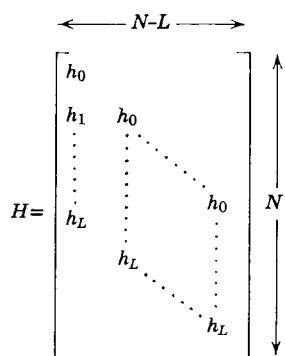


Figure 6.5.3. Check matrix for a cyclic code.

For binary cyclic codes, this circuitry is particularly simple, involving just an  $L$ -stage binary shift register and modulo-2 adders. The multiplication by  $h_i$  is done simply by an open circuit for  $h_i = 0$  and a wire for  $h_i = 1$ . For cyclic codes in an arbitrary field, the circuitry is somewhat more involved, requiring equipment to add and multiply elements in that field.

Next, we shall discuss another realization of a cyclic encoder requiring a shift register of  $N - L$  stages. Clearly, which circuit to use depends on whether the ratio of information digits to block length is large or small. The information digits in a code word can be represented by the polynomial  $x_{N-1}D^{N-1} + \cdots + x_{N-L}D^{N-L}$ . By the Euclidean division algorithm,

$$x_{N-1}D^{N-1} + \cdots + x_{N-L}D^{N-L} = a(D)g(D) + r(D) \quad (6.5.17)$$

where  $r(D)$  is of degree at most  $N - L - 1$ . From (6.5.17), we see that  $x_{N-1}D^{N-1} + \cdots + x_{N-L}D^{N-L} - r(D)$  is a code word and  $-r(D)$  is the polynomial corresponding to the check digits. We can find  $r(D)$  from any circuit that divides  $x_{N-1}D^{N-1} + \cdots + x_{N-L}D^{N-L}$  by  $g(D)$ , and such a circuit is given by Figure 6.5.5.

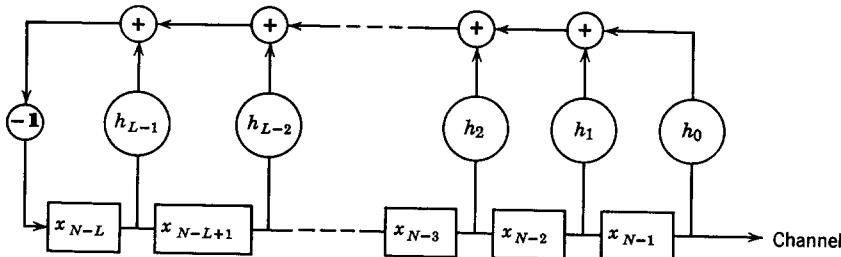


Figure 6.5.4. An  $L$  stage encoder for a cyclic code.

The shift register is initially filled with the first  $N - L$  information digits, right justified if there are fewer than  $N - L$  of them. The first operation is to subtract  $g_j x_{N-1}$  for  $0 \leq j \leq N - L - 1$  from  $x_{L-1+j}$  and then shift the register right one position, adding in a new information digit, if available. The digit  $x_{N-1}$  is no longer required in the calculation of  $r(D)$  and is dropped. It should be verified that the contents of the shift register at this point are precisely what we get after the first subtraction when performing polynomial division with pencil and paper. This cycle is repeated  $L$  times, and by continuing the previous argument, we see that  $r(D)$  is then in the shift register with highest order terms to the right. The switch at the right-hand corner of Figure 6.5.5 is then turned to the vertical position and  $-r(D)$  is read out onto the channel.

From the linearity of this circuit, it can be seen that the same result is achieved if the register is initially empty and the information digits are loaded in at point A by means of an adder which adds the incoming information digit to the output of the right-most stage of the register.

## 6.6 Galois Fields

In this section, we shall study the structure of Galois fields in somewhat more detail than in Section 6.4. These results will be needed in the next

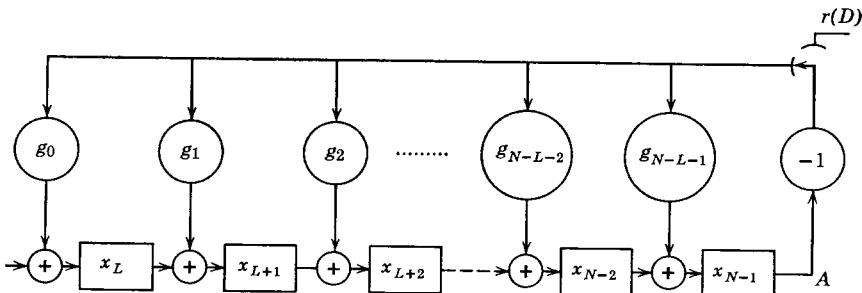


Figure 6.5.5. An  $N - L$  stage encoder for a cyclic code.

section on Bose-Chaudhuri-Hocquenghem (BCH) codes, and appear to play a fundamental role in much of the current research on algebraic coding techniques.

We begin by investigating the multiplicative order of the various nonzero elements in a Galois field with  $q$  elements,  $GF(q)$ . Since the nonzero elements of the field, say  $\alpha_1, \alpha_2, \dots, \alpha_{q-1}$ , form an Abelian group under the multiplication operation of the field, we know from Theorem 6.3.2 that each of these elements has a multiplicative order that divides  $q - 1$ . Thus  $\alpha_i^{q-1} - 1 = 0$  for all  $i$ ,  $0 \leq i \leq q - 1$ . This implies, however, that each  $\alpha_i$  is a root of the polynomial  $D^{q-1} - 1$  [which is here considered as a polynomial in  $GF(q)$ ]. Since this polynomial is of degree  $q - 1$  and since we have just found  $q - 1$  distinct roots, that is,  $\alpha_1, \dots, \alpha_{q-1}$ , we have

$$D^{q-1} - 1 = \prod_{i=1}^{q-1} (D - \alpha_i) \quad (6.6.1)$$

For example, in the field of integers modulo 3, this equation becomes

$$D^2 - 1 = (D - 1)(D - 2)$$

which is clearly satisfied using modulo-3 addition and multiplication of the integers.

We have seen above that all the nonzero elements of a field with  $q$  elements have a multiplicative order dividing  $q - 1$ . A primitive element of a field with  $q$  elements is defined as an element with a multiplicative order of exactly  $q - 1$ . If we can find a primitive element, say  $\alpha$ , of a field, then the sequence  $\alpha, \alpha^2, \dots, \alpha^{q-1}$  constitutes all the nonzero field elements and the multiplicative group of the nonzero elements of the field is cyclic. It then becomes an easy matter to find the multiplicative order of any power of  $\alpha$  (see Problem 6.26).

**Theorem 6.6.1.** Every Galois field contains a primitive element.

---

*Proof.* Let  $m$  be the maximum of the multiplicative orders of the nonzero elements of the field,  $GF(q)$ . From Theorem 6.3.3, each nonzero element has a multiplicative order dividing  $m$ , and thus each is a root of  $D^m - 1$ . Since this equation has  $q - 1$  roots, we must have  $m \geq q - 1$ . From (6.6.1), every nonzero element of the field has a multiplicative order dividing  $q - 1$ , so  $m \leq q - 1$ , completing the proof. |

The above theorem, in principle, completely characterizes the multiplicative group of the nonzero elements of a Galois field, but says nothing about the structure of the additive group or the relationship between multiplication and addition. In what follows, we shall tie down this relationship by showing that all Galois fields can be represented as fields of polynomials modulo an

irreducible polynomial. First, a subfield of a field is defined as a field whose elements form a subset of the elements of the original field and whose addition and multiplication operations are the same as those of the original field. If one field,  $F$ , has a subfield  $E$ , then we shall also define  $F$  to be an extension of  $E$ .

**Theorem 6.6.2.** Each Galois field contains a unique subfield with a prime number of elements.

*Proof.* Any subfield must contain the field elements  $\mathbf{0}$  and  $\mathbf{1}$ . It must also contain  $\mathbf{1} + \mathbf{1}$ ,  $\mathbf{1} + \mathbf{1} + \mathbf{1}$ , etc. We denote these additional elements as  $\mathbf{2}, \mathbf{3}, \mathbf{4}, \dots$ , and observe from Theorem 6.3.2 that these elements, called the integers of the field, form a cyclic subgroup under the operation addition. If this subgroup has  $p$  elements, then addition among these elements is simply addition modulo  $p$ . From the distributive law, multiplication among these elements is also multiplication modulo  $p$  [that is,  $\mathbf{2} \cdot \mathbf{3} = (\mathbf{1} + \mathbf{1}) \cdot \mathbf{3} = \mathbf{3} + \mathbf{3} = R_p(6)$ ]. It follows that  $p$  is prime, for if  $p$  were the product of two integers,  $i > 1$  and  $j > 1$ , then the corresponding field integers would satisfy  $\mathbf{i} \cdot \mathbf{j} = \mathbf{0}$ . This is impossible since  $\mathbf{i}$  and  $\mathbf{j}$  are nonzero elements of the original field. We have already seen that the integers modulo a prime form a field, so this is the subfield that we are looking for. Finally, any other subfield must contain these integers of the field and the additive group of any subfield must contain these integers as a subgroup. Thus the number of elements in any other subfield is divisible by  $p$  and not prime. |

One immediate consequence of this theorem is that any field or subfield with a prime number of elements is, with appropriate labelling of elements, just the field of integers modulo the prime number. The characteristic,  $p$ , of a Galois field is defined as the number of elements in the above prime subfield.

If  $P(D) = P_0 + P_1D + \cdots + P_nD^n$  is a polynomial over a field  $E$  and if a field  $F$  is an extension of  $E$ , then we say that an element  $\alpha$  in  $F$  is a root of  $P(D)$  if  $P(\alpha) = \mathbf{0}$ ; that is, if

$$\sum_i P_i \alpha^i = \mathbf{0}$$

As a common example, we often find complex roots of polynomials over the real-number field. If  $E$  is a subfield of a Galois field  $F$  then the *minimal polynomial*  $f_\alpha(D)$  over  $E$  of an element  $\alpha$  in  $F$  is defined to be the monic polynomial over  $E$  of lowest degree for which  $\alpha$  is a root. If the subfield is not explicitly specified, we shall always mean the subfield with a prime number of elements implied by Theorem 6.6.2.

**Theorem 6.6.3.** For any subfield  $E$  of a Galois field  $GF(q)$ , and each non-zero element  $\alpha$  in  $GF(q)$ ,  $\alpha$  has a unique minimal polynomial  $f_\alpha(D)$  over  $E$

and  $f_\alpha(D)$  is irreducible. Furthermore, for each polynomial  $P(D)$  over  $E$ ,  $f_\alpha(D)$  divides  $P(D)$  iff  $\alpha$  is a root of  $P(D)$ .

*Proof.* We have already seen that  $\alpha$  is a root of  $D^{q-1} - 1$ . Since  $D^{q-1} - 1$  can be considered as a polynomial over  $E$ , this shows the existence of polynomials over  $E$  for which  $\alpha$  is a root, and thus that there is some lowest degree monic polynomial, say  $f_\alpha(D)$ , for which  $\alpha$  is a root. If  $f_\alpha(D)$  is reducible, then it can be expressed as the product of two lower degree monic polynomials over  $E$ , say  $f_\alpha(D) = g(D)h(D)$ . This, however, implies that  $g(\alpha)h(\alpha) = 0$ , and since  $g(\alpha)$  and  $h(\alpha)$  are elements in  $GF(q)$ , one or the other must be zero, contradicting the lowest degree assumption for  $f_\alpha(D)$ . Thus  $f_\alpha(D)$  is irreducible. Now, for any  $P(D)$  over  $E$ ,

$$P(D) = f_\alpha(D)h(D) + r(D) \quad (6.6.2)$$

where  $r(D)$  is a polynomial over  $E$  of lower degree than  $f_\alpha(D)$ . Since  $f_\alpha(\alpha) = 0$ , we have  $P(\alpha) = r(\alpha)$ . Thus  $P(\alpha) = 0$  iff  $r(\alpha) = 0$ . But since  $r(D)$  is of lower degree than  $f_\alpha(D)$ , we have  $r(\alpha) = 0$  iff  $r(D)$  is the zero polynomial. Thus  $P(\alpha) = 0$  iff  $f_\alpha(D)$  divides  $P(D)$ . Finally, if  $P(D)$  is a monic polynomial of the same degree as  $f_\alpha(D)$ ,  $P(D) = f_\alpha(D)h(D)$  is only satisfied for  $h(D) = 1$ , showing that  $f_\alpha(D)$  is unique. |

In the above theorem, the statement that  $f_\alpha(D)$  is irreducible means that there are no lower degree polynomials over  $E$  whose product is  $f_\alpha(D)$ . If we interpret  $f_\alpha(D)$  as a polynomial over  $GF(q)$ , then  $D - \alpha$  is an obvious factor of  $f_\alpha(D)$ .

**COROLLARY.** For a subfield  $E$  of a Galois field  $GF(q)$ , let  $f_1(D), \dots, f_L(D)$  be the distinct minimal polynomials over  $E$  for the nonzero elements of  $GF(q)$  [that is,  $f_{\alpha_1}(D), \dots, f_{\alpha_{q-1}}(D)$  with repetitions removed]. Then

$$D^{q-1} - 1 = \prod_{i=1}^L f_i(D) \quad (6.6.3)$$

Observe that (6.6.1) factors  $D^{q-1} - 1$  into irreducible polynomials over  $GF(q)$ , and (6.6.3) factors  $D^{q-1} - 1$  into irreducible factors over  $E$ .

*Proof.* Since each nonzero element in  $GF(q)$  is a root of  $D^{q-1} - 1$ , each minimal polynomial divides  $D^{q-1} - 1$ . Since the minimal polynomials are irreducible (over the field  $E$ ),

$$\prod_{i=1}^L f_i(D)$$

also divides  $D^{q-1} - 1$ . Finally, all the nonzero elements of  $GF(q)$  are roots of

$$\prod_{i=1}^L f_i(D).$$

Hence the above product has degree at least  $q - 1$  and (6.6.3) is satisfied. |

**Theorem 6.6.4.** Let  $\alpha$  be a primitive element in a Galois field  $GF(q)$  of characteristic  $p$  and let the minimal polynomial of  $\alpha$ ,  $f(D)$  [over  $GF(p)$ ] have degree  $n$ . Then the number of elements in the Galois field is  $p^n$  and each element  $\beta$  in the field can be represented as

$$\beta = i_{n-1}\alpha^{n-1} + i_{n-2}\alpha^{n-2} + \cdots + i_1\alpha + i_0 \quad (6.6.4)$$

for some choice of  $i_0, i_1, \dots, i_{n-1}$  as integers of the field.

*Proof.* Since  $\alpha$  is a root of  $f(D) = D^n + f_{n-1}D^{n-1} + \cdots + f_0$ , we have  $\alpha^n + f_{n-1}\alpha^{n-1} + \cdots + f_0 = 0$ , or

$$\alpha^n = - \sum_{i=0}^{n-1} f_i \alpha^i \quad (6.6.5)$$

Since each  $f_i$  is a field integer, we see that  $\alpha^n$  can be represented as in (6.6.4). Multiplying (6.6.5) by  $\alpha$ , we get

$$\alpha^{n+1} = - \sum_{i=0}^{n-1} f_i \alpha^{i+1} \quad (6.6.6)$$

Thus, since  $\alpha^n$  can be represented as in (6.6.4),  $\alpha^{n+1}$  can also, and successively multiplying (6.6.6) by higher powers of  $\alpha$ , we see that all powers of  $\alpha$  can be represented as in (6.6.4). Thus each element of  $GF(q)$  can be represented as in (6.6.4). Now suppose that there are two different choices of the set of field integers in (6.6.4), say  $i_{n-1}, \dots, i_0$  and  $j_{n-1}, \dots, j_0$ , that correspond to the same element in  $GF(q)$ . Then

$$0 = (i_{n-1} - j_{n-1})\alpha^{n-1} + \cdots + (i_1 - j_1)\alpha + (i_0 - j_0) \quad (6.6.7)$$

This, however, asserts that  $\alpha$  is a root of the polynomial  $(i_{n-1} - j_{n-1})D^{n-1} + \cdots + (i_0 - j_0)$ , which is impossible since this polynomial has degree less than  $n$ . Thus each choice of  $i_{n-1}, \dots, i_0$  corresponds to a different field element, and since there are  $p^n$  choices for this set of field integers,  $q = p^n$ . |

This theorem has a number of interesting consequences. First, since every Galois field has some prime number  $p$  as characteristic, every Galois field must have  $q = p^n$  elements for some prime  $p$  and some integer  $n$ . Next, if in (6.6.4) we replace  $\alpha$  with an indeterminate  $t$ , we see that the set of elements of the field can be regarded as the set of polynomials over  $GF(p)$  of degree at most  $n - 1$  with multiplication modulo  $f(t)$ . In other words, the field here, after a relabelling of elements, is the same as the field of polynomials over  $GF(p)$  modulo  $f(t)$  (that is, it has the same set of elements and the same addition and multiplication table). Two such fields, differing only in the labelling of the elements, are said to be *isomorphic*, and we see from this that *every* field with  $p^n$  elements is isomorphic to some field of polynomials over

$GF(p)$  modulo an  $n$  degree irreducible polynomial. Finally, since (6.6.3) gives the unique factorization of a polynomial  $D^{p^n-1} - 1$  over  $GF(p)$  into irreducible factors, we see that every field with  $p^n$  elements has the same set of minimal polynomials. Thus, for any field with  $p^n$  elements, we can choose  $\alpha$  as a root of the fixed polynomial  $f(D)$  used in Theorem 6.6.4 and represent all elements in the field as in (6.6.4). We thus have proven:

**Theorem 6.6.5.** All Galois fields with  $p^n$  elements are isomorphic to a given field of polynomials over  $GF(p)$  modulo an  $n$ -degree irreducible polynomial.

### Maximal Length Codes and Hamming Codes

Suppose that  $h(D)$  is the minimal polynomial of a primitive element  $\alpha$  in  $GF(p^m)$ . From Theorem 6.6.5, we see that  $h(D)$  is the minimal polynomial of a primitive element in any representation of  $GF(p^m)$ , and such a polynomial is called a *primitive polynomial* of degree  $m$ . A cyclic code of block

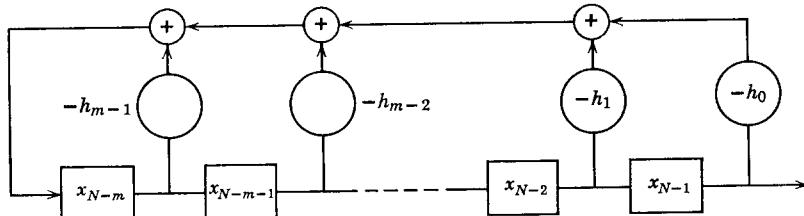


Figure 6.6.1. Encoder for maximal-length code.

length  $N = p^m - 1$  for which a primitive polynomial  $h(D)$  is the check polynomial is known as a maximal length code. The code words of such a code can be generated by the shift register circuit of Figure 6.5.4, redrawn in Figure 6.6.1.

One code word for this code is the code word corresponding to the generator polynomial  $g(D) = [D^{p^m-1} - 1]/h(D)$ . We now show that the set of code words consists simply of the all-zero sequence and the set of cyclic shifts of  $g(D)$ . Since there are  $m$  information digits in the code, there are  $p^m$  code words. Since there are  $p^m - 1$  cyclic shifts of  $g(D)$  (including  $g(D)$  itself), all we need show is that the cyclic shifts of  $g(D)$  are distinct. Suppose then that the  $i$ th and  $j$ th cyclic shifts are identical,  $0 \leq i < j < p^m - 1$ . We then have

$$R_{p^m-1}[D^i g(D)] = R_{p^m-1}[D^j g(D)]$$

This means that, for some  $a(D)$  and  $b(D)$ ,

$$D^i g(D) - a(D)[D^{p^m-1} - 1] = D^j g(D) - b(D)[D^{p^m-1} - 1]$$

Dividing both sides of the equation by  $g(D)$ , and rearranging terms, we have

$$[b(D) - a(D)]h(D) = D^j - D^i = D^i[D^{j-i} - 1]$$

Finally, since  $j - i < p^m - 1$ , the primitive element  $\alpha$  cannot be a root of  $D^{j-i} - 1$ , and thus (by Theorem 6.6.3),  $h(D)$  cannot divide  $D^{j-i} - 1$ . This is a contradiction and all cyclic shifts of  $g(D)$  are distinct.

Since each code word is determined by its  $m$  information digits, we see from the above argument that each cyclic shift of  $g(D)$  has a different initial set of  $m$  digits, and that these make up all the  $p^m - 1$  nonzero combinations of  $m$  digits from  $GF(p)$ . This can be interpreted pictorially by Figure 6.6.2 which shows the code word  $g(D)$  with its ends joined together in a ring so that the cyclic shifts of  $g(D)$  can be read off the same ring starting at different points.

Each consecutive sequence of  $m$  digits in the ring is the information sequence of a different cyclic shift of  $g(D)$ . Furthermore, it can be seen that the information sequence for the  $i$ th cyclic shift of  $g(D)$  is simply the contents of the

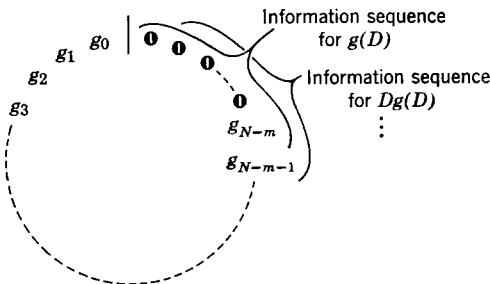


Figure 6.6.2. Digits of  $g(D)$  arranged in ring.

shift register in Figure 6.6.1 (with the highest clockwise term on the ring being on the left of the register) after  $i$  shifts of the register in the generation of  $g(D)$ .

We notice now that out of the  $p^m - 1$  nonzero  $m$ -sequences, exactly  $p^{m-1}$  begin in each of the nonzero digits of  $GF(p)$  and  $p^{m-1} - 1$  begin in 0. Since each digit in the code word corresponding to  $g(D)$  is the beginning of one of these  $m$  sequences, each nonzero code word contains exactly  $p^{m-1}$  appearances of each nonzero digit and  $p^{m-1} - 1$  appearances of 0. Since the difference of any two code words is another code word, it follows that each pair of different code words differs in all but  $p^{m-1} - 1$  positions. It can be shown (see Problem 6.24) that this is the largest number of positions in which all pairs of code words can differ for the given block length and thus these codes are very effective for error correction.

If the shift register in Figure 6.6.1 generates an unterminating sequence of digits instead of stopping after  $p^m - 1$  digits, then it can be seen that the

resulting sequence is periodic with period  $p^m - 1$  and successive digits in the sequence are found by reading clockwise around the ring of Figure 6.6.2. Since the future output of a feedback shift register depends only on its present contents, it can be seen that its output must repeat periodically after its contents are identical to its contents at some previous time. Ruling out the zero sequence, there are only  $p^m - 1$  possible contents for an  $m$  stage register and thus the output from every  $m$  stage feedback shift register with elements in  $GF(p)$  is periodic with a period of at most  $p^m - 1$ . Since the use of a primitive polynomial for the feedback connections achieves this maximum period, these registers are called *maximal-length* feedback shift registers and the codes are called maximal length codes.

The periodic sequences from a maximal-length feedback shift register are called *pseudo noise* (or  $p - n$ ) sequences. If we pick a sequence of  $i$  digits ( $i \leq m$ ) at a randomly chosen place in the sequence, then each nonzero sequence has probability  $p^{m-i}/(p^m - 1)$  and the zero sequence has probability  $(p^{m-i} - 1)/(p^m - 1)$ . Thus, for large  $m$  and for many purposes, the resulting sequence behaves like a random sequence of independent, equally likely digits from  $GF(p)$ . These sequences are frequently used for ranging and synchronization purposes in radar and communication. For example, in the binary case, we can distinguish between  $2^m - 1$  different range positions or synchronization positions with the use of an  $m$  stage shift register.

Next, consider the dual code to a maximal length code. The primitive polynomial  $h(D)$  is now the generator polynomial for the code and  $g(D)$  is the check polynomial. The columns of the check matrix can now be taken as  $g(D)$  and its first  $m - 1$  cyclic shifts. The set of rows of the check matrix is thus the set of sequences of  $m$  consecutive digits in the ring in Figure 6.6.2. Thus the rows are all distinct and we have a cyclic representation of a Hamming code.

**Example.** We shall now discuss a particular Galois field,  $GF(2^4)$ , in detail, partly to make the theory seem a little less abstract, and partly to discuss the implementation of operations in a Galois field. As a representation of  $GF(2^4)$ , we can consider the field of polynomials over  $GF(2)$  modulo the polynomial  $f(D) = D^4 + D + 1$ . It can be verified, by dividing  $f(D)$  by all 1 and 2 degree polynomials over  $GF(2)$ , that  $f(D)$  is irreducible. In Figure 6.6.3, we represent the elements of  $GF(2^4)$  in two ways, one as powers of the field element  $\alpha$  represented by the polynomial  $t$  and the other as polynomials  $g(t) = g_3t^3 + g_2t^2 + g_1t + g_0$ . As discussed in Section 6.4, addition of field elements is performed by polynomial addition and multiplication is performed by polynomial multiplication modulo  $f(t)$ . Multiplication by the field element  $\alpha = t$  is particularly easy and can be instrumented by the circuit in Figure 6.6.4. The register is loaded with a polynomial  $g(t)$ , the register is shifted right one place corresponding to multiplication by  $t$  and then reduced

|               | Elements of $GF(2^4)$ |       |                       |       | Minimal Polynomials       |
|---------------|-----------------------|-------|-----------------------|-------|---------------------------|
|               | As Powers of $\alpha$ |       | As Polynomials $g(t)$ |       |                           |
| <b>0</b>      | $g_3$                 | $g_2$ | $g_1$                 | $g_0$ |                           |
| <b>1</b>      | 0                     | 0     | 0                     | 1     | $D + 1$                   |
| $\alpha$      | 0                     | 0     | 1                     | 0     | $D^4 + D + 1$             |
| $\alpha^2$    | 0                     | 1     | 0                     | 0     | $D^4 + D + 1$             |
| $\alpha^3$    | 1                     | 0     | 0                     | 0     | $D^4 + D^3 + D^2 + D + 1$ |
| $\alpha^4$    | 0                     | 0     | 1                     | 1     | $D^4 + D + 1$             |
| $\alpha^5$    | 0                     | 1     | 1                     | 0     | $D^2 + D + 1$             |
| $\alpha^6$    | 1                     | 1     | 0                     | 0     | $D^4 + D^3 + D^2 + D + 1$ |
| $\alpha^7$    | 1                     | 0     | 1                     | 1     | $D^4 + D^3 + 1$           |
| $\alpha^8$    | 0                     | 1     | 0                     | 1     | $D^4 + D + 1$             |
| $\alpha^9$    | 1                     | 0     | 1                     | 0     | $D^4 + D^3 + D^2 + D + 1$ |
| $\alpha^{10}$ | 0                     | 1     | 1                     | 1     | $D^2 + D + 1$             |
| $\alpha^{11}$ | 1                     | 1     | 1                     | 0     | $D^4 + D^3 + 1$           |
| $\alpha^{12}$ | 1                     | 1     | 1                     | 1     | $D^4 + D^3 + D^2 + D + 1$ |
| $\alpha^{13}$ | 1                     | 1     | 0                     | 1     | $D^4 + D^3 + 1$           |
| $\alpha^{14}$ | 1                     | 0     | 0                     | 1     | $D^4 + D^3 + 1$           |

Figure 6.6.3. Elements of  $GF(2^4)$  as polynomials modulo  $D^4 + D + 1$ .

modulo  $f(t)$  by the feedback connections. The reader should verify that successive shifts of the register in Figure 6.6.4 generate the successive polynomials in Figure 6.6.3.

It will be observed that the field element  $\alpha = t$  in Figure 6.6.3 has multiplicative order 15 and is thus primitive. In any field of polynomials modulo a polynomial  $f(D)$ , the field element  $\alpha = t$  has the minimal polynomial  $f(D)$  [since the remainder of  $f(t)$  modulo  $f(t)$  is 0], and thus the fact that  $\alpha$  is primitive here simply means that we were fortunate enough to choose  $f(D)$  as a primitive polynomial.

The minimal polynomials for all the field elements in  $GF(2^4)$  are shown in Figure 6.6.3. Problem 6.29 develops a way to calculate these minimal

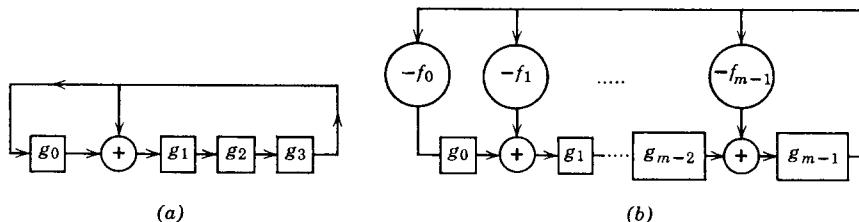


Figure 6.6.4. Circuit for multiplying arbitrary field element by  $\alpha = t$  in field of polynomials modulo  $f(D)$ . (a)  $f(D) = D^4 + D + 1$ . (b) Arbitrary  $f(D)$  of degree  $m$ .

polynomials based on subsequent results in this section. Peterson (1961) has tabulated these minimal polynomials for fields of characteristic 2 up to  $GF(2^{17})$ , so it is rarely necessary to go through the calculations. At this point, however, we can at least verify that the minimal polynomials in Figure 6.6.3 are correct. For example, the minimal polynomial for  $\alpha^{10}$  is listed as  $D^2 + D + 1$ . For  $\alpha^{10}$  to be a root of  $D^2 + D + 1$  we must have  $\alpha^{20} + \alpha^{10} + 1 = 0$ . Since  $\alpha^{15} = 1$ ,  $\alpha^{20} = \alpha^5$ , this reduces to  $\alpha^{10} + \alpha^5 + 1 = 0$ , and adding the polynomial representations for  $\alpha^{10}$ ,  $\alpha^5$ , and  $1$ , we find that this is indeed true.

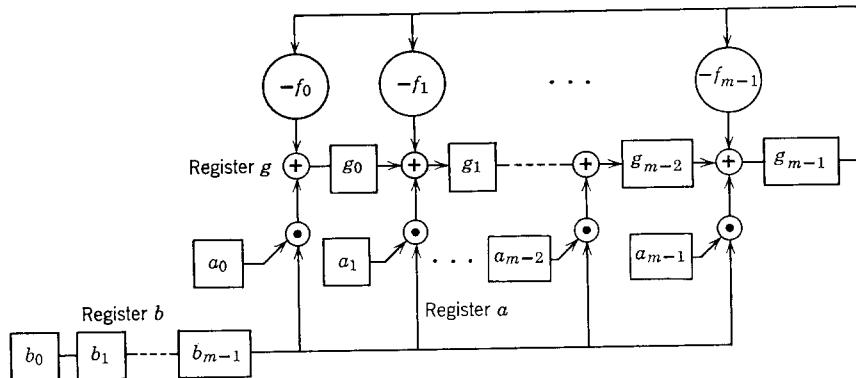


Figure 6.6.5. Multiplication of field elements  $a(t)$  and  $b(t)$  in field of polynomials modulo  $f(D)$ .

For implementing operations in an arbitrary Galois field  $GF(p^m)$ , it is usually convenient to represent the elements of the field as polynomials modulo an  $m$ -degree primitive polynomial over  $GF(p)$ , and operationally this means representing each element as a sequence of  $m$  elements in  $GF(p)$ . We have seen that addition in  $GF(p^m)$  then corresponds to adding the corresponding sequences in  $GF(p)$ . Also multiplication by the special element  $\alpha = t$  is accomplished by the circuit in Figure 6.6.4. Finally, multiplication of two arbitrary field elements can be performed by the circuit in Figure 6.6.5.

In the operation of this circuit, the register  $g$  is initially empty and  $a(t)$  and  $b(t)$  are loaded into registers  $a$  and  $b$ , respectively. Then the registers  $b$  and  $g$  are shifted right one position so that register  $g$  contains the polynomial  $b_{m-1}a(t)$ . Next, registers  $b$  and  $g$  are shifted right again and the resulting contents of  $g$  are  $R_{f(t)}[(b_{m-1}t + b_{m-2})a(t)]$ . After  $m$  shifts of registers  $g$  and  $b$ , register  $g$  contains  $R_{f(t)}[b(t)a(t)]$ , or the field element  $b(t) * a(t)$ . It should be noted that both the time required and the equipment required to perform multiplication in  $GF(p^m)$  by the circuit are directly proportional to  $m$ .

### Existence of Galois Fields

We have already shown that Galois fields only exist where the number of elements is a power of a prime and also that for each number of elements there is at most one field, except for a relabelling of the elements. In the next three theorems, we show that for each  $p^n$ , a Galois field  $GF(p^n)$  actually exists. All we must demonstrate, of course, is the existence of irreducible polynomials of all degrees over  $GF(p)$  for all prime numbers  $p > 1$ .

**Theorem 6.6.6.** Let  $\alpha$  and  $\beta$  be elements of  $GF(p^n)$  with characteristic  $p$ . Then for all integers  $m \geq 1$ ,

$$(\alpha + \beta)^{p^m} = \alpha^{p^m} + \beta^{p^m} \quad (6.6.8)$$


---

*Proof.* For  $m = 1$ , the binomial theorem asserts that

$$(\alpha + \beta)^p = \alpha^p + p\alpha^{p-1}\beta + \cdots + \binom{p}{i} \alpha^{p-i}\beta^i + \cdots + \beta^p \quad (6.6.9)$$

where  $\binom{p}{i} \alpha^{p-i}\beta^i$  is to be interpreted as the sum of  $\binom{p}{i}$  terms, each  $\alpha^{p-i}\beta^i$ .

On the other hand, for  $1 \leq i \leq p - 1$ ,

$$\binom{p}{i} = \frac{p(p-1)!}{i!(p-i)!} \quad (6.6.10)$$

Since  $\binom{p}{i}$  is an integer and  $p$  is prime, the denominator in (6.6.10) divides  $(p-1)!$  and thus  $\binom{p}{i}$  contains  $p$  as a factor. Thus, in  $GF(p^n)$ , 1 added to itself  $\binom{p}{i}$  times is 0, all the intermediate terms in (6.6.9) are 0, and

$$(\alpha + \beta)^p = \alpha^p + \beta^p \quad (6.6.11)$$

Raising (6.6.11) to the  $p$  power, we obtain

$$(\alpha + \beta)^{p^2} = (\alpha^p + \beta^p)^p = \alpha^{p^2} + \beta^{p^2}$$

and similarly, raising it to the  $p$  power  $m - 1$  times, we obtain (6.6.9). |

**COROLLARY.** If

$$f(D) = \sum_{i=1}^l f_i D^i$$

is a polynomial over  $GF(p)$  and  $\alpha$  is an element of  $GF(p^n)$ , then for each  $m \geq 1$

$$f(\alpha^{p^m}) = [f(\alpha)]^{p^m} \quad (6.6.12)$$

In particular, if  $\alpha$  is a root of  $f(D)$ , then for each  $m \geq 0$ ,

$$f(\alpha^{p^m}) = \mathbf{0} \quad (6.6.13)$$

*Proof.*

$$\begin{aligned} [f(\alpha)]^{p^m} &= \left[ f_i \alpha^l + \sum_{i=0}^{l-1} f_i \alpha^i \right]^{p^m} \\ &= (f_i \alpha^l)^{p^m} + \left( \sum_{i=0}^{l-1} f_i \alpha^i \right)^{p^m} \\ &= \sum_{i=0}^l (f_i \alpha^i)^{p^m} = \sum_{i=0}^l f_i^{p^m} \alpha^{ip^m} \end{aligned} \quad (6.6.14)$$

Since  $f_i$  is an element of  $GF(p)$ ,  $f_i^{p-1} = \mathbf{1}$ , and thus  $f_i^p = f_i$ . Successively raising both sides to the  $p$  power  $f_i^{p^m} = f_i$ . Thus (6.6.14) becomes

$$[f(\alpha)]^{p^m} = \sum_{i=0}^l f_i \alpha^{ip^m} = f(\alpha^{p^m}) \quad (6.6.15)$$

If  $f(\alpha) = 0$ , (6.6.13) follows. |

**Theorem 6.6.7.** Let  $f(D)$  be a monic irreducible  $n$ -degree polynomial over  $GF(p)$ . Then  $f(D)$  divides  $D^{p^m-1} - \mathbf{1}$  iff  $n$  divides  $m$ .

*Proof.* Consider the field  $GF(p^n)$  of polynomials over  $GF(p)$  modulo  $f(t)$ . The polynomial  $t$  is a field element in this field which we shall denote by  $\alpha$ . Since  $f(\alpha) = \mathbf{0}$ ,  $\alpha$  is a root of  $f(D)$ . Any other element in the field can be represented as

$$\beta = i_0 + i_1 \alpha + \cdots + i_{n-1} \alpha^{n-1} \quad (6.6.16)$$

where  $i_0, \dots, i_{n-1}$  are field integers. Let  $i_0, \dots, i_{n-1}$  be chosen so that  $\beta$  is a primitive element and let  $B(D)$  be the corresponding polynomial,  $B(D) = i_0 + i_1 D + \cdots + i_{n-1} D^{n-1}$ , so that  $\beta = B(\alpha)$ . Now suppose that  $f(D)$  divides  $D^{p^m-1} - \mathbf{1}$ . Then  $\alpha$  is a root of  $D^{p^m-1} - \mathbf{1}$  and

$$\alpha^{p^m} - \alpha = \mathbf{0} \quad (6.6.17)$$

Using (6.6.17) and (6.6.12), we then have

$$\beta = B(\alpha) = B(\alpha^{p^m}) = [B(\alpha)]^{p^m} = \beta^{p^m} \quad (6.6.18)$$

From (6.6.18), we see that the order of  $\beta$  divides  $p^m - 1$ . On the other hand, since  $\beta$  is primitive, its order is  $p^n - 1$ , and  $p^n - 1$  must divide  $p^m - 1$ . Carrying out the division,

$$p^m - 1 = (p^n - 1)(p^{m-n} + p^{m-2n} + \cdots), \quad (6.6.19)$$

we see that  $p^n - 1$  divides  $p^m - 1$  iff  $n$  divides  $m$ . Thus, if  $f(D)$  divides  $D^{p^m-1} - \mathbf{1}$ ,  $n$  must divide  $m$ . Conversely, if  $n$  divides  $m$ , the order of  $\alpha$  divides

$p^m - 1$  since it divides  $p^n - 1$ . Hence  $\alpha$  is a root of  $D^{p^m-1} - 1$  and its minimal polynomial  $f(D)$  divides  $D^{p^m-1} - 1$ . |

From this theorem, we see that for any  $m \geq 1$ , the irreducible factors of  $D^{p^m-1} - 1$  over  $GF(p)$  all have degrees given by  $m$  or divisors of  $m$ .

**Theorem 6.6.8.** For every positive integer  $m$  and prime number  $p$ , there exist irreducible polynomials in  $GF(p)$  of degree  $m$  and thus fields with  $p^m$  elements.

---

The proof of this theorem rests on the fact that there are not enough polynomials of degree  $m/2$  or less to constitute all the factors of  $D^{p^m-1} - 1$ . Before making use of this argument, we need the following lemma.

**LEMMA.**  $D^{p^m-1} - 1$ , as a polynomial over  $GF(p)$ , has no repeated monic irreducible factors of positive degree.

---

*Proof.* Suppose that  $f(D)$  is an  $n$ -degree monic irreducible factor of  $D^{p^m-1} - 1$ . Since  $n$  divides  $m$ ,  $p^n - 1$  divides  $p^m - 1$  [as in (6.6.19)]. Thus we have

$$D^{p^m-1} - 1 = (D^{p^n-1} - 1)A(D) \quad (6.6.20)$$

where

$$A(D) = D^{(p^m-1)-(p^n-1)} + D^{(p^m-1)-2(p^n-1)} + \cdots + 1 \quad (6.6.21)$$

The existence of  $f(D)$  asserts the existence of a Galois field  $GF(p^n)$  and thus  $f(D)$  cannot be contained as a repeated factor in  $(D^{p^n-1} - 1)$ . Also  $f(D)$  is the minimal polynomial of some element  $\alpha$  in  $GF(p^n)$ , and if  $f(D)$  divides  $A(D)$ , then  $\alpha$  is a root of  $A(D)$ . On the other hand, since  $\alpha$  has an order dividing  $p^m - 1$ ,  $\alpha^{(p^m-1)-i(p^n-1)} - 1 = 0$ . Hence

$$A(\alpha) = 1 + 1 + \cdots + 1 \quad (6.6.22)$$

where the number of terms on the right-hand side of (6.6.22) is  $(p^m - 1)/(p^n - 1) = p^{m-n} + p^{m-2n} + \cdots + 1$ . Since  $A(\alpha)$  is the remainder when this number of terms is divided by  $p$ ,  $A(\alpha) = 1$ . Thus  $\alpha$  is not a root of  $A(D)$ , and  $f(D)$  cannot divide  $A(D)$ , completing the proof. |

*Proof of Theorem.* Let  $a_i$  be the number of monic irreducible factors of  $D^{p^m-1} - 1$  of degree  $m/i$ . Since the sum of the degrees of the factors is  $p^m - 1$ , we have

$$p^m - 1 = a_1 m + a_2 \frac{m}{2} + a_3 \frac{m}{3} + \cdots + a_m \quad (6.6.23)$$

Since all irreducible factors of degree  $m/i$  divide  $D^{p^{m/i}-1} - 1$ ,

$$a_i \leq \frac{p^{m/i} - 1}{m/i} \quad (6.6.24)$$

$$p^m - 1 \leq a_1 m + \sum_{\substack{i=2 \\ i: \frac{m}{i} = \text{integer}}}^m [p^{m/i} - 1] \quad (6.6.25)$$

Replacing  $m/i$  by  $j$  in (6.6.25) and upper bounding by summing over all  $j$ ,  $1 \leq j \leq m/2$

$$p^m - 1 \leq a_1 m + \frac{p^{\lfloor m/2 \rfloor + 1} - p}{p - 1} - \left\lfloor \frac{m}{2} \right\rfloor$$

This inequality is clearly only satisfied for  $a_1 > 0$ , completing the proof. |

## 6.7 BCH Codes

The Bose, Chaudhuri, and Hocquenghem (BCH) codes, discovered by Hocquenghem (1959) and independently by Bose and Chaudhuri (1960), constitute a class of cyclic codes which both have powerful error-correcting properties and simple decoding algorithms. The most common examples of BCH codes are binary, but the alphabet can equally well be the elements from an arbitrary Galois field, say  $GF(q)$ . The generator polynomials for these codes are defined in terms of some extension field of  $GF(q)$ , say  $GF(q^m)$ . Let  $\alpha$  be an element of  $GF(q^m)$  of multiplicative order  $N$ , and for arbitrary integers  $r \geq 0$  and  $d$ ,  $2 \leq d \leq N$ , let  $f_r(D), f_{r+1}(D), \dots, f_{r+d-2}(D)$  be the minimal polynomials of  $\alpha^r, \alpha^{r+1}, \dots, \alpha^{r+d-2}$ . For each choice of the above parameters (that is,  $q, m, \alpha, r$ , and  $d$ ) there is a BCH code defined; its generator polynomial is defined to be

$$g(D) = \text{LCM } [f_r(D), f_{r+1}(D), \dots, f_{r+d-2}(D)] \quad (6.7.1)$$

The block length of the code is defined to be  $N$ , the multiplicative order of  $\alpha$ . Since each of the elements  $\alpha_r, \dots, \alpha_{r+d-2}$  are roots of  $D^N - 1$ , each of the polynomials  $f_r(D), \dots, f_{r+d-2}(D)$  divides  $D^N - 1$  (see Theorem 6.6.3), and thus  $g(D)$  divides  $D^N - 1$ , as required for a cyclic code. In the uninteresting case where  $g(D) = D^N - 1$ , we take the all-zero sequence as the only word by convention.

An alternate definition of a BCH code with the above parameters is that a sequence  $x_{N-1}, \dots, x_0$  is a code word iff  $\alpha^r, \alpha^{r+1}, \dots, \alpha^{r+d-2}$  are roots of  $x_{N-1}D^{N-1} + x_{N-2}D^{N-2} + \dots + x_0$ . To see this, we observe that, by the definition in (6.7.1), each code-word polynomial is divisible by  $g(D)$  and hence by each of the minimal polynomials  $f_r(D), \dots, f_{r+d-2}(D)$ . Thus

$\alpha^r, \alpha^{r+1}, \dots, \alpha^{r+d-2}$  are roots of the code-word polynomial. Conversely, if  $x_{N-1}D^{N-1} + \dots + x_0$  is not a code word, it is not divisible by  $g(D)$  and, hence, not divisible by at least one of the minimal polynomials, say  $f_i(D)$ . Thus  $\alpha^i$  is not a root of  $x_{N-1}D^{N-1} + \dots + x_0$ .

In most of the applications,  $\alpha$  is taken as a primitive element of  $GF(q^m)$  so that  $N = q^m - 1$ . Also  $r$  is usually taken as 1. We shall see, later, that the parameter  $d$  has the significance of being a lower bound to the minimum distance of the code.

**Example.** Since the above definitions undoubtedly seem somewhat abstract, we shall carry through a concrete example with  $q = 2, m = 4, r = 1, d = 5$ . As a representative of  $GF(2^4)$ , we shall use Figure 6.6.3. For the element  $\alpha$ , we select the element  $\alpha$  in Figure 6.6.3 represented by the polynomial  $t$ . As we found in Section 6.6, the minimal polynomial of  $\alpha$  over  $GF(2)$  is  $f_1(D) = D^4 + D + 1$ . Also, from (6.6.13),  $\alpha^2, \alpha^4$ , and  $\alpha^8$  all have the same minimal polynomial as  $\alpha$ . As shown in Section 6.6, the minimal polynomial of  $\alpha^3$  over  $GF(2)$  is  $f_3(D) = D^4 + D^3 + D^2 + D + 1$ . Thus

$$g(D) = f_1(D)f_3(D) = D^8 + D^7 + D^6 + D^4 + 1 \quad (6.7.2)$$

One interesting feature that this example has brought out is that, for  $q = 2$  and for any  $i, f_{2i}(D) = f_i(D)$ . Thus for  $r = 1, q = 2$ , and  $d$  odd, we can rewrite (6.7.1) as

$$g(D) = \text{LCM } [f_1(D), f_3(D), \dots, f_{d-2}(D)] \quad (6.7.3)$$

Since the degree of  $g(D)$  is the number of check digits in the cyclic code generated by  $g(D)$ , this asserts that for odd  $d$ , the BCH code with  $q = 2, r = 1$  and arbitrary  $\alpha$  and  $m$  has at most  $m[(d - 1)/2]$  check digits. Assuming, for the moment, that  $d$  is a lower bound to the minimum distance of the code, this asserts that by using  $em$  check digits, we can correct all combinations of  $e$  errors. By making  $\alpha$  primitive, the block length of the code will be  $2^m - 1$ . For  $r \neq 1$  or  $q \neq 2$ , the corresponding observation from (6.7.1) is that all combinations of  $e$  errors can be corrected by using, at most,  $2me$  check digits.

Since the BCH codes are cyclic codes, we can generate a check matrix for any BCH code by using the polynomial  $h(D) = [D^N - 1]/g(D)$  as in Section 6.5. A more convenient form of the check matrix for our purposes here comes from recalling that  $x(D)$  is a code word polynomial iff  $x(\alpha^i) = 0$  for  $r \leq i \leq r + d - 2$ . This equation can be rewritten as

$$\sum_{n=0}^{N-1} x_n \alpha^{in} = 0; \quad r \leq i \leq r + d - 2 \quad (6.7.4)$$

Defining the check matrix  $H$  as

$$H = \begin{bmatrix} \alpha^{(N-1)r} & \alpha^{(N-1)(r+1)} & \dots & \alpha^{(N-1)(r+d-2)} \\ \alpha^{(N-2)r} & & & \\ \cdot & \cdot & & \\ \cdot & \cdot & & \\ \cdot & \cdot & & \\ \alpha^r & \alpha^{r+1} & & \alpha^{r+d-2} \\ \mathbf{1} & \mathbf{1} & & \mathbf{1} \end{bmatrix} \quad (6.7.5)$$

we can rewrite (6.7.4) as

$$\mathbf{x}H = 0 \quad (6.7.6)$$

iff  $\mathbf{x} = (x_{N-1}, \dots, x_0)$  is a code word. If, for some  $i$  and  $j$ ,  $\alpha^i$  and  $\alpha^j$  have the same minimal polynomial, then  $x(\alpha^i) = 0$  iff  $x(\alpha^j) = 0$  and the column corresponding to  $\alpha^j$  can be omitted from (6.7.5). Thus, for the example just considered,

$$H^T = \begin{bmatrix} \alpha^{N-1} & \alpha^{N-2} & \dots & \alpha & 1 \\ \alpha^{3(N-1)} & \alpha^{3(N-2)} & \dots & \alpha^3 & 1 \end{bmatrix} \quad (6.7.7a)$$

We recall, from Section 6.6, that elements in an extension field,  $GF(q^m)$ , of  $GF(q)$  can be regarded as  $m$  vectors over  $GF(q)$ . Multiplication of an element  $\alpha$  in  $GF(q^m)$  by an element  $x$  in  $GF(q)$  corresponds to scalar multiplication of vector  $\alpha$  by scalar  $x$  in  $GF(q)$ . Thus, if we regard the elements of  $H$  in (6.7.5) as row vectors in  $GF(q)$ , the matrix multiplication  $\mathbf{x}H$  involves only operations in  $GF(q)$ . For the example just considered,  $H$  in (6.7.7a) can be rewritten in this way as follows (see Figure 6.6.3):

$$H^T = \begin{bmatrix} 1 & 1 & 1 & 1 & 0 & 1 & 0 & 1 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 1 & 0 & 1 & 0 & 1 & 1 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 1 & 0 & 1 & 0 & 1 & 1 & 0 & 0 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 & 1 & 0 & 1 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 1 \\ \hline 1 & 1 & 1 & 1 & 0 & 1 & 1 & 1 & 1 & 0 & 1 & 1 & 1 & 1 & 0 \\ 1 & 0 & 1 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 1 & 0 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 1 \end{bmatrix} \quad (6.7.7b)$$

**Theorem 6.7.1.** A BCH code with the parameter  $d$  as defined above has a minimum distance of at least  $d$ .

*Proof.* Let  $\mathbf{x} = (x_{N-1}, \dots, x_0)$  be a sequence of symbols from  $GF(q)$  and suppose that all but  $d - 1$  of these symbols are constrained to be  $\mathbf{0}$  and the remaining symbols, say  $x_{n_1}, x_{n_2}, \dots, x_{n_{d-1}}$ , can take on arbitrary values. We shall show that, for any choice of the integers  $n_j$  (satisfying  $N - 1 \geq n_1 > n_2 > \dots > n_{d-1} \geq 0$ ),  $\mathbf{x}$  can be a code word iff  $x_{n_j} = \mathbf{0}$  for  $1 \leq j \leq d - 1$ . This will show that no nonzero code word differs from the all-zero sequence in fewer than  $d$  positions and, hence, that the code has minimum distance at least  $d$ .

For a given choice of the integers  $\{n_j\}$ ,  $\mathbf{x}$  will be a code word iff [see (6.7.4)]:

$$\sum_{j=1}^{d-1} x_{n_j} \alpha^{in_j} = \mathbf{0}; \quad r \leq i \leq r + d - 2 \quad (6.7.8)$$

The notation can be simplified here by defining

$$\begin{aligned} V_j &= x_{n_j} & 1 \leq j \leq d - 1 \\ U_j &= \alpha^{n_j} \end{aligned} \quad (6.7.9)$$

Equation 6.7.8 then becomes

$$\begin{aligned} V_1 U_1^r &+ V_2 U_2^r + \dots + V_{d-1} U_{d-1}^r = \mathbf{0} \\ V_1 U_1^{r+1} &+ \dots + V_{d-1} U_{d-1}^{r+1} = \mathbf{0} \\ &\vdots && \vdots && \vdots \\ &\vdots && \vdots && \vdots \\ &\vdots && \vdots && \vdots \\ V_1 U_1^{r+d-2} &+ \dots + V_{d-1} U_{d-1}^{r+d-2} = \mathbf{0} \end{aligned} \quad (6.7.10)$$

This is a set of  $d - 1$  equations [over  $GF(q^m)$ ] in the  $d - 1$  unknowns  $V_1, \dots, V_{d-1}$ . One solution of the equations is the trivial one  $V_1 = \dots = V_{d-1} = 0$ . To complete the proof, we must show that this solution is unique [the demonstration that there is no other solution in  $GF(q^m)$  certainly establishes that there is none in  $GF(q)$ ]. On the other hand, this solution is unique unless the equations are linearly dependent (the proof of this is the same as in the field of real numbers). By the equations being linearly dependent, we mean that there is a set of field elements,  $\beta_0, \beta_1, \dots, \beta_{d-2}$ , not all zero, for which

$$\sum_{j=0}^{d-2} \beta_j U_i^{r+j} = \mathbf{0} \quad \text{for } 1 \leq i \leq d - 1 \quad (6.7.11)$$

Suppose that there is such a set of field elements and let  $\beta(D) = \beta_0 + \beta_1 D + \dots + \beta_{d-2} D^{d-2}$ . In terms of  $\beta(D)$ , (6.7.11) is

$$U_i^r \beta(U_i) = \mathbf{0}; \quad 1 \leq i \leq d - 1 \quad (6.7.12)$$

On the other hand,  $\beta(D)$  is a nonzero polynomial of degree at most  $d - 2$  and (6.7.12) asserts that  $\beta(D)$  has at least  $d - 1$  roots. This is a contradiction and, hence, the equations are linearly independent, completing the proof. |

Since the minimum distance of a BCH code is at least  $d$ , we know that any combination of  $\lfloor(d - 1)/2\rfloor$  or fewer errors can be corrected, and we now develop an algorithm for correcting all of these error patterns. Usually, there are also many patterns of more than  $\lfloor(d - 1)/2\rfloor$  errors that the code is capable of correcting but no simple algorithm is known for correcting them.

Let  $x(D) = x_{N-1}D^{N-1} + \cdots + x_0$  denote the transmitted code-word polynomial,  $y(D) = y_{N-1}D^{N-1} + \cdots + y_0$  the received sequence polynomial, and  $z(D) = y(D) - x(D)$  the noise sequence polynomial. Define the syndrome  $\mathbf{S}$  as a vector with components

$$S_i = y(\alpha^{r+i}) \quad 0 \leq i \leq d - 2 \quad (6.7.13)$$

Notice that each  $S_i$  is an element of  $GF(q^m)$ , and in terms of the matrix  $H$  in (6.7.5),  $\mathbf{S} = \mathbf{y}H$ . Since  $\alpha^{r+i}$  is a root of  $x(D)$  for  $0 \leq i \leq d - 2$ , this becomes

$$S_i = x(\alpha^{r+i}) + z(\alpha^{r+i}) = z(\alpha^{r+i}); \quad 0 \leq i \leq d - 2 \quad (6.7.14)$$

Now, suppose that some given number  $e \leq \lfloor(d - 1)/2\rfloor$  of errors has occurred in transmission, say in positions  $n_1, n_2, \dots, n_e$ . Then  $z_n = 0$  except for  $n = n_1, n_2, \dots, n_e$ , and (6.7.14) becomes

$$S_i = \sum_{j=1}^e z_{n_j} \alpha^{n_j(r+i)}; \quad 0 \leq i \leq d - 2 \quad (6.7.15)$$

To simplify notation, define the *error values*  $V_j$  and the *error locators*  $U_j$  by

$$\begin{aligned} V_j &= z_{n_j} & 1 \leq j \leq e \\ U_j &= \alpha^{n_j} \end{aligned} \quad (6.7.16)$$

Equation 6.7.15 then becomes

$$S_i = \sum_{j=1}^e V_j U_j^{r+i}; \quad 0 \leq i \leq d - 2 \quad (6.7.17)$$

To review, the decoder can calculate the syndrome  $\mathbf{S}$  from the received sequence  $\mathbf{y}$ . If the decoder can solve (6.7.17) to find the error values and error locators, the decoder can then find the error sequence  $\mathbf{z}$  from (6.7.16) (recall that  $\alpha^0, \alpha^1, \alpha^2, \dots, \alpha^{N-1}$  are all distinct since  $N$  is the multiplicative order of  $\alpha$ ). We now turn to the major problem, which, from a conceptual standpoint, is the solution of (6.7.17).

First, we define the infinite degree polynomial\*  $S_\infty(D)$  as

$$S_\infty(D) = S_0 + S_1 D + S_2 D^2 + \cdots \quad (6.7.18)$$

\* It is not conventional to call such expressions polynomials, but it is convenient here.

where  $S_i$ , for all  $i \geq 0$ , is given by

$$S_i = \sum_{j=1}^e V_j U_j^{r+i}; \quad i \geq 0 \quad (6.7.19)$$

Recall that, for  $0 \leq i \leq d-2$ ,  $S_i$  can be directly found from  $\mathbf{y}$ ; for  $i > d-2$ ,  $S_i$  is unknown but at least defined for the given error sequence  $\mathbf{z}$ . We can rewrite  $S_\infty(D)$  in the form

$$\begin{aligned} S_\infty(D) &= \sum_{i=0}^{\infty} \sum_{j=1}^e V_j U_j^{r+i} D^i = \sum_{j=1}^e V_j U_j^r \sum_{i=0}^{\infty} U_j^i D^i \\ &= \sum_{j=1}^e V_j U_j^r \frac{1}{1 - U_j D} \end{aligned} \quad (6.7.20)$$

where by  $1/(1 - U_j D)$  we mean the “polynomial,”  $1 + U_j D + U_j^2 D^2 + \dots$ . Next define the polynomial  $\sigma(D) = \sigma_0 + \sigma_1 D + \dots + \sigma_e D^e$  by

$$\sigma(D) = \prod_{j=1}^e (1 - U_j D) \quad (6.7.21)$$

Taking the product of (6.7.20) and (6.7.21), we then obtain\*

$$\sigma(D) S_\infty(D) = \sum_{j=1}^e V_j U_j^r \prod_{\substack{l=1 \\ l \neq j}}^e (1 - U_l D) \triangleq A(D) \quad (6.7.22)$$

To interpret this equation, it is helpful to develop some more notation. For an arbitrary polynomial  $B(D)$  (finite or infinite), let  $[B(D)]_i^j$  be defined by

$$[B(D)]_i^j = \begin{cases} \sum_{l=i}^j B_l D^l; & j \geq i \\ 0; & j < i \end{cases} \quad (6.7.23)$$

By convention, if  $j$  exceeds the degree, say  $L$ , of  $B(D)$ , we take  $B_{L+1} = B_{L+2} = \dots = B_j = 0$ . Next, define  $S(D) = S_0 + S_1 D + \dots + S_{d-2} D^{d-2}$ . In the notation above,

$$S(D) = [S_\infty(D)]_0^{d-2} \quad (6.7.24)$$

Since the terms in  $S_\infty(D)$  of degree greater than  $d-2$  effects  $S_\infty(D)\sigma(D)$  only in terms of degree greater than  $d-2$ , we have from (6.7.22)

$$[\sigma(D)S(D)]_0^{d-2} = [A(D)]_0^{d-2} \quad (6.7.25)$$

\* The reader not used to such formal manipulations should verify to himself that

$$\left[ \prod_{l=1}^e (1 - U_l D) \right] [1 + U_j D + U_j^2 D^2 + \dots] = \prod_{l \neq j} (1 - U_l D)$$

with equality for all powers of  $D$ .

Finally, we observe from the definition of  $A(D)$  in (6.7.22) that  $A(D)$  has degree at most  $e - 1$ . Thus the brackets around  $A(D)$  in (6.7.25) are unnecessary and, furthermore,

$$[\sigma(D)S(D)]_e^{d-2} = 0 \quad (6.7.26)$$

Equation 6.7.26 states that the coefficient of  $D^l$  in the product  $\sigma(D)S(D)$  is  $\mathbf{0}$  for  $e \leq l \leq d - 2$ . Even more explicitly, (6.7.26) is equivalent to the following set of equations.

$$\begin{aligned} \sigma_0 S_e + \sigma_1 S_{e-1} + \cdots + \sigma_e S_0 &= \mathbf{0} \\ \sigma_0 S_{e+1} + \sigma_1 S_e + \cdots + \sigma_e S_1 &= \mathbf{0} \\ \vdots &\quad \vdots \\ \vdots &\quad \vdots \\ \vdots &\quad \vdots \\ \sigma_0 S_{d-2} + \sigma_1 S_{d-3} + \cdots + \sigma_e S_{d-2-e} &= \mathbf{0} \end{aligned} \quad (6.7.27)$$

Equation 6.7.27 provides the decoder with a set of  $d - 1 - e$  linear equations to solve for the  $e$  unknowns  $\sigma_1, \dots, \sigma_e$  [from (6.7.21),  $\sigma_0 = 1$ ]. If these equations can be solved, then the error locators  $U_1, \dots, U_e$  can be found from  $\sigma(D)$  by observing that  $U_1^{-1}, \dots, U_e^{-1}$  are the roots of  $\sigma(D)$ . We can now see the outline of the decoding procedure and, for future reference, we summarize it in four steps.

*Step 1.* Calculate  $S_0, \dots, S_{d-2}$  from the received sequence  $\mathbf{y}$ .

*Step 2.* Find  $\sigma(D)$  from (6.7.26) or (6.7.27).

*Step 3.* Find the roots of  $\sigma(D)$  and hence the locations of the errors.

*Step 4.* Find the values of the errors,  $V_1, \dots, V_e$ . Notice that, for binary BCH codes, the error values must all be  $1$  (since, by definition, they are nonzero) and hence step 4 can be avoided.

We now discuss step 2 in detail and return later to discuss the implementation of the other steps. Observe that, in solving (6.7.26) for  $\sigma(D)$ , the decoder does not know the number of errors  $e$ . The following theorem asserts that  $\sigma(D)$  can be uniquely found without knowing  $e$  beforehand.

**Theorem 6.7.2.** Assume that  $e \leq \lfloor (d - 1)/2 \rfloor$  errors have occurred and that  $\sigma(D)$  is given by (6.7.21). Let  $\hat{e}$  be the smallest integer for which a polynomial  $\hat{\sigma}(D)$  exists of degree at most  $\hat{e}$  with  $\hat{\sigma}_0 = 1$  and satisfying  $[\hat{\sigma}(D)S(D)]_{\hat{e}}^{d-2} = 0$ . Then  $e = \hat{e}$  and  $\sigma(D) = \hat{\sigma}(D)$ .

*Proof.* The equation  $[\hat{\sigma}(D)S(D)]_{\hat{e}}^{d-2} = 0$  can be rewritten as

$$\sum_{i=0}^{\hat{e}} \hat{\sigma}_i S_{i-\hat{e}} = \mathbf{0}; \quad \hat{e} \leq i \leq d - 2 \quad (6.7.28)$$

From the definition of  $S_i$ , this becomes

$$\begin{aligned} \sum_{l=0}^{\hat{e}} \hat{\sigma}_l S_{i-l} &= \sum_{l=0}^{\hat{e}} \hat{\sigma}_l \sum_{j=1}^e V_j U_j^{r+i-l} \\ &= \sum_{j=1}^e V_j U_j^{r+i} \sum_{l=0}^{\hat{e}} \hat{\sigma}_l U_j^{-l} \\ &= \sum_{j=1}^e V_j \hat{\sigma}(U_j^{-1}) U_j^{r+i} = \mathbf{0}; \quad \hat{e} \leq i \leq d-2 \end{aligned} \quad (6.7.29)$$

Equation 6.7.29 can be regarded as a set of  $d-1-\hat{e}$  linear equations in the  $e$  unknowns  $V_j \hat{\sigma}(U_j^{-1})$  for  $1 \leq j \leq e$ . Since  $\hat{e}$  is the smallest integer for which (6.7.28) can be satisfied and since  $[\sigma(D)S(D)]_e^{d-2} = \mathbf{0}$ , we must have  $\hat{e} \leq e$ . Also, since  $e \leq \lfloor(d-1)/2\rfloor$ , it follows that  $e \leq d-1-\hat{e}$ . Now consider only the first  $e$  of the equations in (6.7.29), for  $\hat{e} \leq i \leq \hat{e}+e-1$ . As  $e$  equations in the  $e$  unknowns  $V_j \hat{\sigma}(U_j^{-1})$ , these equations are linearly independent by the same argument used to establish the linear independence of (6.7.10) in Theorem 6.7.1. Thus the only solution to these equations is given by

$$V_j \hat{\sigma}(U_j^{-1}) = \mathbf{0}; \quad 1 \leq j \leq e \quad (6.7.30)$$

Since  $V_j \neq 0$ ,  $1 \leq j \leq e$ , it follows that  $U_j^{-1}$  is a root of  $\hat{\sigma}(D)$  for  $1 \leq j \leq e$ . Since the degree of  $\hat{\sigma}(D)$  is at most  $e$ , since  $\hat{\sigma}(D)$  has the same  $e$  roots as  $\sigma(D)$ , and since  $\hat{\sigma}_0 = \sigma_0$ , it follows that  $\hat{\sigma}(D) = \sigma(D)$ . Since  $\hat{\sigma}(D)$  has degree  $e$ , we also have  $\hat{e} = e$ , completing the proof. |

We next describe an iterative algorithm for finding  $\sigma(D)$  in a particularly simple way.

#### *Iterative Algorithm\* for Finding $\sigma(D)$*

We saw in the previous theorem that if at most  $\lfloor(d-1)/2\rfloor$  errors occur, then  $\sigma(D)$  is given in terms of the syndrome polynomial  $S(D)$  by that solution to  $[\sigma(D)S(D)]_e^{d-2} = \mathbf{0}$  for which  $e$  is smallest and  $\sigma(D)$  has degree at most  $e$  with  $\sigma_0 = \mathbf{1}$ . This problem is most easily visualized in terms of the linear-feedback shift register (LFSR) shown in Figure 6.7.1.

The register is initially loaded with a sequence of elements  $S_0, S_1, \dots, S_{t-1}$  from a given field. The register then computes a new element  $S_t$  given in terms of the feedback connections  $-C_1, \dots, -C_t$ , by

$$S_t = -S_{t-1}C_1 - S_{t-2}C_2 - \cdots - S_0C_t \quad (6.7.31)$$

The elements  $C_1, \dots, C_t$  are elements of the same field as the  $S_i$ . The register is then shifted to the right one position,  $S_t$  entering the register from

\* This algorithm is due to Berlekamp (1967). We present here a modification of Berlekamp's algorithm due to Massey (1968) and follow Massey's approach closely.

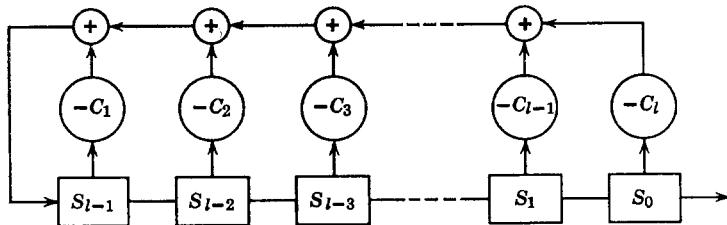


Figure 6.7.1. Linear feedback shift register (LFSR).

the left. On each successive shift to the right, a new element is computed, given by

$$S_i = -S_{i-1}C_1 - S_{i-2}C_2 - \cdots - S_{i-l}C_l; \quad i \geq l \quad (6.7.32)$$

We define the *register length* of an LFSR as the number of stages in the shift register ( $l$  in Figure 6.7.1). We also define the *connection polynomial*  $C(D)$  of an LFSR as  $C(D) = 1 + C_1D + C_2D^2 + \cdots + C_lD^l$  where  $C_1, C_2, \dots, C_l$  are the negatives of the feedback connections as shown in Figure 6.7.1. Since an LFSR is completely described (apart from the initial loading) by its register length and connection polynomial, we shall use the notation *register*  $[C(D), l]$  to denote an LFSR with register length  $l$  and connection polynomial  $C(D)$ . Any or all of the feedback connections  $-C_1, \dots, -C_l$  may be zero so that  $C(D)$  may be an arbitrary polynomial with  $C_0 = 1$  and degree at most  $l$ . Finally, if  $S(D)$  is a polynomial (either finite or infinite), we shall say that the register  $[C(D), l]$  generates  $[S(D)]_0^L$  iff, when the register is initially loaded with  $S_0, \dots, S_{l-1}$ , the remaining elements (if any)  $S_l, \dots, S_L$  are those generated by the register, as given by (6.7.32). Observing that (6.7.32) is equivalent to the statement that the coefficient of  $D^i$  in  $C(D)S(D)$  is zero, we see that register  $[C(D), l]$  generates  $[S(D)]_0^L$  iff

$$[C(D)S(D)]_l^L = 0 \quad (6.7.33)$$

The relationship between linear-feedback shift registers and finding  $\sigma(D)$  should now be clear. We want to find the register  $[\sigma(D), e]$  of smallest register length  $e$  that generates  $[S(D)]_0^{d-2}$ . Notice that the restrictions that  $\sigma_0 = 1$  and that the degree of  $\sigma(D)$  is at most  $e$  are built into the definition of register  $[\sigma(D), e]$ . The algorithm to be given below finds this shortest register. We shall show that the algorithm works for an arbitrary polynomial  $S(D)$  over an arbitrary field. The algorithm operates by finding a sequence of registers, first finding the shortest register that generates  $S_0$ , then the shortest register that generates  $S_0 + S_1D$ , and so forth. The register produced by the algorithm to generate  $[S(D)]_0^{n-1}$  is denoted  $[C_n(D), l_n]$ .

Roughly, the algorithm works as follows: given a register  $[C_n(D), l_n]$  that generates  $[S(D)]_0^{n-1}$ , the algorithm tests to see whether  $[C_n(D), l_n]$  also generates  $[S(D)]_0^n$ , that is, whether  $[C_n(D)S(D)]_{l_n}^n = 0$ . Since by assumption  $[C_n(D)S(D)]_{l_n}^{n-1} = 0$ , the question is whether the coefficient of  $D^n$  in  $C_n(D)S(D)$  is equal to zero. This sum is called the *next discrepancy*,  $d_n$ , for the algorithm and expressing  $C_n(D)$  by  $\mathbf{1} + C_{n,1}D + C_{n,2}D^2 + \dots$ , we have

$$d_n = S_n + \sum_{i=1}^n C_{n,i} S_{n-i} \quad (6.7.34)$$

Letting  $S'_n$  be the coefficient of  $D^n$  generated by the register, as given by (6.7.32), we have  $d_n = S_n - S'_n$  so that  $d_n$  is the difference between the desired next output  $S_n$  and the actual register output  $S'_n$ . If  $d_n = 0$ , the algorithm increases  $n$  by one, but keeps the same register. If  $d_n \neq 0$ , a correction term is added to the connection polynomial to make it generate  $S_n$  correctly.

The detailed rules for the algorithm are as follows: for each  $n$ , the register  $[C_{n+1}(D), l_{n+1}]$  is defined in terms of the register  $[C_n(D), l_n]$  and a prior register in the sequence  $[C_{k_n}(D), l_{k_n}]$  where  $k_n < n$ . The integer  $k_n$  is recursively defined for each  $n > 0$  by

$$k_n = \begin{cases} k_{n-1} & \text{if } l_n = l_{n-1} \\ n-1 & \text{if } l_n > l_{n-1} \end{cases} \quad (6.7.35)$$

$C_{n+1}(D)$  and  $l_{n+1}$  are given by

$$C_{n+1}(D) = C_n(D) - \frac{d_n}{d_{k_n}} D^{n-k_n} C_{k_n}(D) \quad (6.7.36)$$

$$l_{n+1} = \begin{cases} l_n & ; \quad d_n = 0 \\ \max [l_n, n - (k_n - l_{k_n})] & ; \quad d_n \neq 0 \end{cases} \quad (6.7.37)$$

where  $d_n$  and  $d_{k_n}$  are given by (6.7.34), or more explicitly,

$$d_{k_n} = S_{k_n} + \sum_{i=1}^{k_n} C_{k_n,i} S_{k_n-i}$$

The algorithm starts at  $n = 0$  with the initial conditions  $C_0(D) = C_{-1}(D) = \mathbf{1}$ ,  $l_0 = l_{-1} = 0$ ,  $k_0 = -1$ ,  $d_{-1} = \mathbf{1}$ .

The following theorem asserts that  $C_n(D)$  and  $l_n$  specify a register  $[C_n(D), l_n]$  and that that register generates  $[S(D)]_0^{n-1}$ . We show, in a later theorem, that  $[C_n(D), l_n]$  is the shortest register that generates  $[S(D)]_0^{n-1}$ .

**Theorem 6.7.3.** For each  $n \geq 0$ ,

$$(a) k_n < n \quad (6.7.38)$$

$$(b) C_{n,0} = 1 \text{ [where } C_n(D) = C_{n,0} + C_{n,1}D + \dots] \quad (6.7.39)$$

$$(c) \deg [C_n(D)] \leq l_n \quad (6.7.40)$$

$$(d) [C_n(D)S(D)]_{l_n}^{n-1} = 0 \quad (6.7.41)$$


---

*Proof.*

*Part a.* For  $n = 0$ ,  $k_0 < 0$  from the initial conditions. For  $n > 0$ , the proof is immediate from (6.7.35) using induction on  $n$ .

*Part b.* From the initial conditions  $C_{0,0} = 1$ . Now assume that  $C_{n,0} = 1$  for any given  $n$ . Since  $n - k_n > 0$ , it follows from (6.7.36) that  $C_{n+1,0} = 1$ . Thus, by induction,  $C_{n,0} = 1$  for all  $n \geq 0$ .

*Parts c and d.* We again use induction on  $n$ . From the initial conditions, (6.7.40) and (6.7.41) are satisfied for  $n = -1, 0$ . For any given  $n$ , assume that for  $-1 \leq i \leq n$ ,

$$\deg [C_i(D)] \leq l_i \quad (6.7.42)$$

$$[C_i(D)S(D)]_{l_i}^{i-1} = 0 \quad (6.7.43)$$

The proof will be complete if we show that this implies that (6.7.42) and (6.7.43) are also satisfied for  $i = n + 1$ . We consider separately the case in which  $d_n = 0$  and that in which  $d_n \neq 0$ . For  $d_n = 0$ ,  $C_{n+1}(D) = C_n(D)$  and  $l_{n+1} = l_n$ . Thus (6.7.42) for  $i = n$  implies (6.7.42) for  $i = n + 1$ . Also (6.7.43) for  $i = n$  implies that

$$[C_{n+1}(D)S(D)]_{l_{n+1}}^{n-1} = 0$$

From (6.7.34), we have  $[C_{n+1}(D)S(D)]_n^n = d_n = 0$ . Thus  $[C_{n+1}(D)S(D)]_{l_{n+1}}^n = 0$ , establishing (6.7.43) for  $i = n + 1$ . Now, assume that  $d_n \neq 0$ . From (6.7.36), we have

$$\begin{aligned} \deg [C_{n+1}(D)] &\leq \max \{\deg [C_n(D)], n - k_n + \deg [C_{k_n}(D)]\} \\ &\leq \max \{l_n, n - k_n + l_{k_n}\} = l_{n+1} \end{aligned}$$

where we have used (6.7.42) for  $i = n$  and  $i = k_n$  and then used (6.7.37). Finally, from (6.7.36)

$$[C_{n+1}(D)S(D)]_{l_{n+1}}^n = [C_n(D)S(D)]_{l_{n+1}}^n - \left[ \frac{d_n}{d_{k_n}} D^{n-k_n} C_{k_n}(D) S(D) \right]_{l_{n+1}}^n \quad (6.7.44)$$

Since  $l_{n+1} \geq l_n$ , we have

$$[C_n(D)S(D)]_{l_{n+1}}^n = d_n D^n \quad (6.7.45)$$

| $n$ | $S_n$    | $l_n$ | $C_n(D)$       | LFSR | $d_n$    | $k_n$ | $l_{k_n}$ | $C_{k_n}(D)$ | $d_{k_n}$ |
|-----|----------|-------|----------------|------|----------|-------|-----------|--------------|-----------|
| 0   | <b>1</b> | 0     | <b>1</b>       |      | <b>1</b> | -1    | 0         | <b>1</b>     | <b>1</b>  |
| 1   | <b>1</b> | 1     | <b>1+D</b>     |      | <b>0</b> | 0     | 0         | <b>1</b>     | <b>1</b>  |
| 2   | <b>1</b> | 1     | <b>1+D</b>     |      | <b>0</b> | 0     | 0         | <b>1</b>     | <b>1</b>  |
| 3   | <b>0</b> | 1     | <b>1+D</b>     |      | <b>1</b> | 0     | 0         | <b>1</b>     | <b>1</b>  |
| 4   | <b>1</b> | 3     | <b>1+D+D^3</b> |      | <b>0</b> | 3     | 1         | <b>1+D</b>   | <b>1</b>  |
| 5   | <b>1</b> | 3     | <b>1+D+D^3</b> |      | <b>1</b> | 3     | 1         | <b>1+D</b>   | <b>1</b>  |
| 6   | <b>0</b> | 3     | <b>1+D+D^2</b> |      | <b>0</b> | 3     | 1         | <b>1+D</b>   | <b>1</b>  |
| 7   | <b>1</b> | 3     | <b>1+D+D^2</b> |      | <b>0</b> | 3     | 1         | <b>1+D</b>   | <b>1</b>  |
| 8   |          | 3     | <b>1+D+D^2</b> |      |          |       |           |              |           |

Figure 6.7.2. Operation of algorithm in  $GF(2)$  for  $S(D) = 1 + D + D^2 + D^4 + D^5 + D^7$  up to  $n = 8$ .

For the final term in (6.7.44), we observe that  $D^{n-k_n}$  can be moved outside the brackets if the limits are simultaneously reduced by  $n - k_n$ . Thus

$$\begin{aligned} \left[ \frac{d_n}{d_{k_n}} D^{n-k_n} C_{k_n}(D) S(D) \right]_{l_{n+1}}^n &= \frac{d_n}{d_{k_n}} D^{n-k_n} [C_{k_n}(D) S(D)]_{(l_{n+1}-n+k_n)}^{k_n} \\ &= \frac{d_n}{d_{k_n}} D^{n-k_n} d_{k_n} D^{k_n} \end{aligned} \quad (6.7.46)$$

where we have used (6.7.37) to see that  $l_{n+1} - n + k_n \geq l_{k_n}$ . Substituting (6.7.45) and (6.7.46) into (6.7.44), we have  $[C_{n+1}(D) S(D)]_{l_{n+1}}^n = 0$ , completing the proof. |

In Figure 6.7.2 we give an example of how the algorithm works, using polynomials in  $GF(2)$  for simplicity. In Figure 6.7.3,  $l_n$  and  $n - l_n$  are

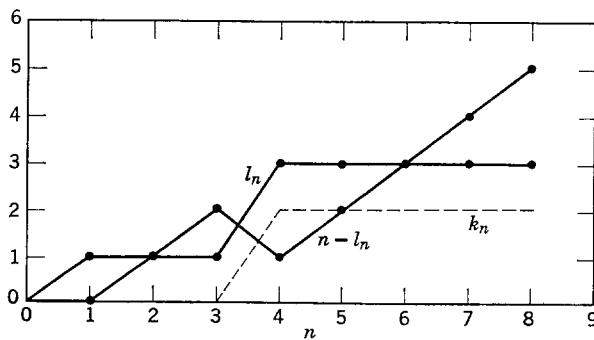


Figure 6.7.3. Sketch of  $l_n$  and  $n - l_n$  as functions of  $n$ .

sketched as functions of  $n$  for this example. There are some important aspects of this relationship between  $l_n$  and  $n - l_n$  that are valid in general. First, we observe that  $l_n$  is nondecreasing with  $n$  and it is obvious from (6.7.37) that this is valid in general. The next three results are more subtle.

LEMMA.1 For each  $n \geq 0$ ,

$$k_n - l_{k_n} > i - l_i; \quad -1 \leq i < k_n \quad (6.7.47)$$

$$l_n = k_n - l_{k_n} + 1 \quad (6.7.48)$$

$$l_{n+1} > l_n \text{ iff } l_n \leq \frac{n}{2} \text{ and } d_n \neq \mathbf{0} \quad (6.7.49)$$

*Proof.* We first show that (6.7.48) implies (6.7.49). From (6.7.37),  $l_{n+1} > l_n$  iff both  $d_n \neq \mathbf{0}$  and

$$n - (k_n - l_{k_n}) > l_n \quad (6.7.50)$$

From (6.7.48), (6.7.50) is equivalent to  $n > 2l_n - 1$  or  $l_n \leq n/2$ , establishing (6.7.49). Equations 6.7.47 and 6.7.48 are clearly valid for  $n = 0$  and we assume them to be valid for any given  $n$ . Using induction, the proof will be complete if we can show that the equations are valid for  $n + 1$ . If  $l_{n+1} = l_n$ , then  $k_{n+1} = k_n$ , and (6.7.47) and (6.7.48) are valid for  $n + 1$ . If  $l_{n+1} > l_n$ , then from (6.7.35),  $k_{n+1} = n$  and

$$k_{n+1} - l_{k_{n+1}} = n - l_n \quad (6.7.51)$$

Since  $k_n$  is where the most recent register length change occurred prior to  $n$ ,  $l_i = l_n$  for  $k_n < i < n$ , and thus

$$k_{n+1} - l_{k_{n+1}} > i - l_i; \quad k_n < i < n \quad (6.7.52)$$

Also, from (6.7.37),  $n - l_n > k_n - l_{k_n}$ , so that combining (6.7.51) with (6.7.47),

$$k_{n+1} - l_{k_{n+1}} > i - l_i; \quad -1 \leq i \leq k_n \quad (6.7.53)$$

establishing (6.7.47) for  $n + 1$ . Still assuming  $l_{n+1} > l_n$ , we can combine (6.7.50), which is valid for  $n$ , and (6.7.51) to obtain

$$k_{n+1} - l_{k_{n+1}} = l_{n+1} + 1 \quad (6.7.54)$$

This establishes (6.7.48) for  $n + 1$ , completing the proof. |

From this lemma, we can see that  $n - l_n$  as a function of  $n$  will have the appearance of an ascending sequence of peaks and for each  $n$ ,  $k_n$  gives the location of the peak prior to  $n$ , which is higher than any of the preceding peaks (we consider a peak to occur at  $n$  if  $n - l_n \geq (n + 1) - l_{n+1}$ ).

Before proving that the algorithm produces the shortest possible register for each  $n$ , we need two lemmas.

LEMMA 2. Suppose that  $[A(D), l]$  and  $[B(D), l]$  are two registers satisfying

$$[A(D)S(D)]_l^n = aD^n; \quad a \neq 0 \quad (6.7.55)$$

$$[B(D)S(D)]_l^n = 0 \quad (6.7.56)$$

then for some  $j$ ,  $0 \leq j \leq l$ , there is a register  $[F(D), l-j]$  satisfying

$$[F(D)S(D)]_{l-j}^{n-j} = fD^{n-j}; \quad f \neq 0 \quad (6.7.57)$$

*Proof.*

$$\{[A(D) - B(D)]S(D)\}_l^n = aD^n \quad (6.7.58)$$

Let  $j$  be the smallest integer for which  $A_j \neq B_j$  and let  $\gamma = A_j - B_j$ . Let  $F(D)$  be defined by

$$A(D) - B(D) = \gamma D^j F(D) \quad (6.7.59)$$

Now  $F_0 = 1$  and

$$\deg F(D) < \min [\deg A(D), \deg B(D)] - j \leq l - j$$

Thus  $[F(D), l-j]$  is a register. Substituting (6.7.59) into (6.7.58) and observing that we can remove  $D^j$  from the brackets if we reduce the limits by  $j$ , we have

$$\gamma [F(D)S(D)]_{l-j}^{n-j} = aD^{n-j}$$

Since  $a/\gamma \neq 0$ , this completes the proof. |

LEMMA 3. Assume that for a given  $S(D)$  and a given  $n$ , the register  $[C_i(D), l_i]$  is the shortest register that generates  $[S(D)]_0^{i-1}$  for each  $i \leq n$ . Then there exists no register  $[A(D), l_A]$  such that, for some  $n_A < n$ , both

$$n_A - l_A > k_n - l_{k_n} \quad (6.7.60)$$

and

$$[A(D)S(D)]_{l_A}^{n_A} = aD^{n_A}; \quad a \neq 0 \quad (6.7.61)$$


---

*Proof.* We shall assume that the lemma is false and exhibit a contradiction. Let  $[A(D), l_A]$  be the shortest register for which (6.7.60) and (6.7.61) are satisfied with  $n_A < n$ .

*Case a.* Assume that  $n_A > n_k$ . We have seen that  $l_i = l_n$  for  $n_k < i < n$  and thus  $[C_n(D), l_n]$  is the shortest register that generates  $[S(D)]_0^{i-1}$  for  $n_k < i < n$ . Taking  $i = n_A$ , this shows that  $l_n \leq l_A$ . Thus, since  $n_A < n$ , the register  $[C_n(D), l_n]$  satisfies

$$[C_n(D)S(D)]_{l_n}^{n_A} = 0 \quad (6.7.62)$$

From the previous lemma, (6.7.61) and (6.7.62) assert the existence of a register  $[F(D), l_A - j]$  for some  $j > 0$  satisfying

$$[F(D)S(D)]_{l_A-j}^{n_A-j} = fD^{n_A-j}; \quad f \neq 0$$

This register is shorter than  $[A(D), l_A]$  and satisfies (6.7.60) and (6.7.61), establishing a contradiction.

*Case b.* Assume  $n_A \leq n_k$ . The register  $[C_{n_A}(D), l_{n_A}]$  is by hypothesis the shortest register generating  $[S(D)]_0^{n_A-1}$ , and thus  $l_{n_A} \leq l_A$ . Thus, using (6.7.47),

$$k_n - l_{k_n} \geq n_A - l_{n_A} \geq n_A - l_A$$

contradicting (6.7.60). |

**Theorem 6.7.4.** For any  $S(D)$  and each  $n \geq 0$ , no register that generates  $[S(D)]_0^{n-1}$  has a smaller register length than the register  $[C_n(D), l_n]$  produced by the algorithm.

*Proof.* We use induction on  $n$ . The theorem is clearly valid for  $n = 0$ . Assume that, for any given  $S(D)$ , it is valid for a given  $n$ . If  $l_{n+1} = l_n$ , then clearly  $[C_{n+1}(D), l_{n+1}]$  is the shortest register generating  $[S(D)]_0^n$  since it is the shortest register generating  $[S(D)]_0^{n-1}$  and any register generating  $[S(D)]_0^n$  also generates  $[S(D)]_0^{n-1}$ . Now assume  $l_{n+1} > l_n$ , so that

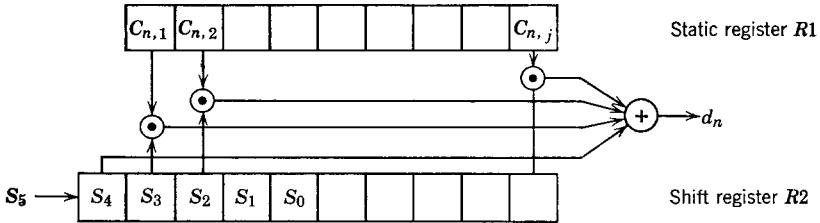
$$[C_n(D)S(D)]_{l_n}^n = d_n D^n; \quad d_n \neq 0$$

Let  $[B(D), l_B]$  be any register generating  $[S(D)]_0^n$ . We must have  $l_B \geq l_n$ , and, from Lemma 2, the registers  $[C_n(D), l_B]$  and  $[B(D), l_B]$  imply the existence of a register  $[F(D), l_B - j]$  for some  $j > 0$  so that

$$[F(D)S(D)]_{l_B-j}^{n-j} = f D^{n-j}; \quad f \neq 0$$

From Lemma 3,  $(n - j) - (l_B - j) \leq k_n - l_{k_n}$ . Thus  $l_B \geq n - (k_n - l_{k_n}) = l_{n+1}$ . Thus register  $[B(D), l_B]$  is no shorter than  $[C_{n+1}(D), l_{n+1}]$ , completing the proof. |

The block diagram in Figure 6.7.4, from Massey (1968), suggests a way of implementing the algorithm. Notice that it uses (6.7.49) as a test for when  $l_n$  and  $k_n$  change. The length of the registers,  $j$ , in Figure 6.7.4, must be long enough to store the connection polynomial of the longest register expected. For decoding BCH codes, we choose  $L = d - 2$  and  $\sigma(D)$ , except for the  $\sigma_0 = 1$  term, is left in  $R1$ . We can choose  $j = \lfloor (d - 1)/2 \rfloor$  and be guaranteed of correcting all combinations of at most  $\lfloor (d - 1)/2 \rfloor$  errors. For binary BCH codes, the elements  $\{S_i\}$  and  $\{C_i\}$  are elements of  $GF(2^m)$ , and each of the registers in Figure 6.7.4 can be implemented by  $m$  binary registers. The  $GF(2^m)$  multipliers can be implemented as in Figure 6.6.5. It can be seen that the equipment required for the registers and multipliers is proportional to  $m d$ . It can also be seen that the time required to find  $\sigma(D)$  is proportional to  $m d$  [or slightly more, depending on how  $(d^*)^{-1}$  is calculated]. There are, of course, a number of detailed design questions to be answered in building



Notes: For each  $n$ ,  $R1$  contains  $C_n(D)$ , except for  $C_{n,0} = \mathbf{1}$

$R2$  contains  $[S(D)]_0^n$

$R3$  contains  $F(D) = D^{n-k_n} C_{k_n}(D)$  (note that  $F_0 = \mathbf{0}$ )

$d^*$  is a memory cell containing  $d_{k_n}$

Shift register  $R3$

#### Control Functions

|  |
|--|
| <b>00.</b> . . . 0 $\rightarrow R1$          |
| $S_0 0.$ . . . 0 $\rightarrow R2$            |
| <b>10.</b> . . . 0 $\rightarrow R3$          |
| <b>0</b> $\rightarrow n; 0 \rightarrow l_n;$ |
| <b>1</b> $\rightarrow d^*$                   |

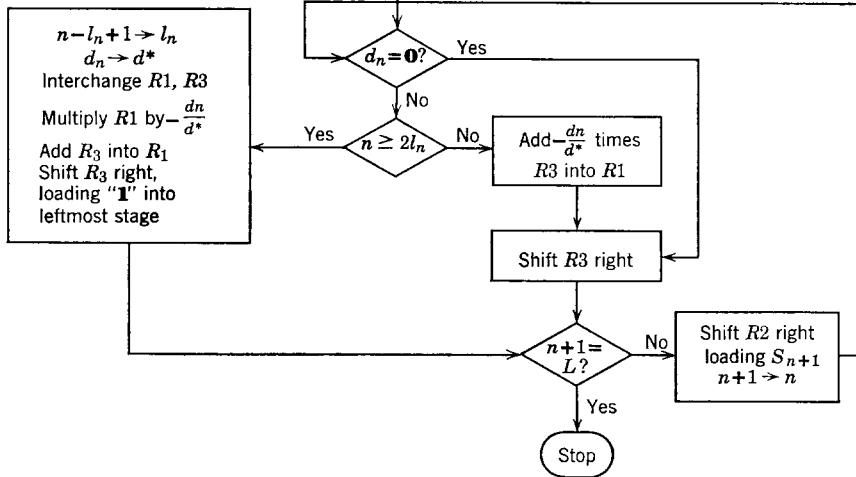


Figure 6.7.4. Implementation of LFSR algorithm to generate  $[S(D)]_0^L$ .

such a device and the only point to be made here is that such a device requires surprisingly little hardware and surprisingly little computing time. Berlekamp (1967) has also shown that, for binary codes with  $r = 1$ ,  $d_n$  is always zero for  $n$  odd. Making use of this fact essentially cuts the computation time to find  $\sigma(D)$  in half. This completes our discussion of step 2 in the BCH decoding procedure.

We now briefly discuss the implementation of steps 1, 3, and 4 in a BCH decoder. For step 1, the elements of the syndrome can be calculated by

$$S_i = \sum_{n=0}^{N-1} y_n \alpha^{(r+i)n} = (\cdots (y_{N-1} \alpha^{r+i} + y_{N-2}) \alpha^{r+i} + y_{N-3}) \alpha^{r+i} \cdots + y_0$$

Thus  $S_i$  can be calculated by adding each successive received digit into an initially empty register, the sum to be multiplied by  $\alpha^{r+i}$  and returned to the register awaiting the next received digit.

Step 3 is most easily implemented by a procedure due to Chien (1964). If at most  $\lfloor (d-1)/2 \rfloor$  errors have occurred, then  $\sigma(D)$ , as calculated in step 2, will be given by (6.7.21) and an error will have occurred in position  $n$  (that is,  $z_n \neq 0$ ) iff  $\sigma(\alpha^{-n}) = 0$ , or equivalently iff

$$\sum_{i=0}^r \sigma_i \alpha^{-ni} = 0 \quad (6.7.63)$$

If we define

$$\sigma_{i,n} = \sigma_i \alpha^{-ni} \quad (6.7.64)$$

then

$$\sigma_{i,N-1} = \sigma_i \alpha^{-(N-1)i} = \sigma_i \alpha^i \quad (6.7.65)$$

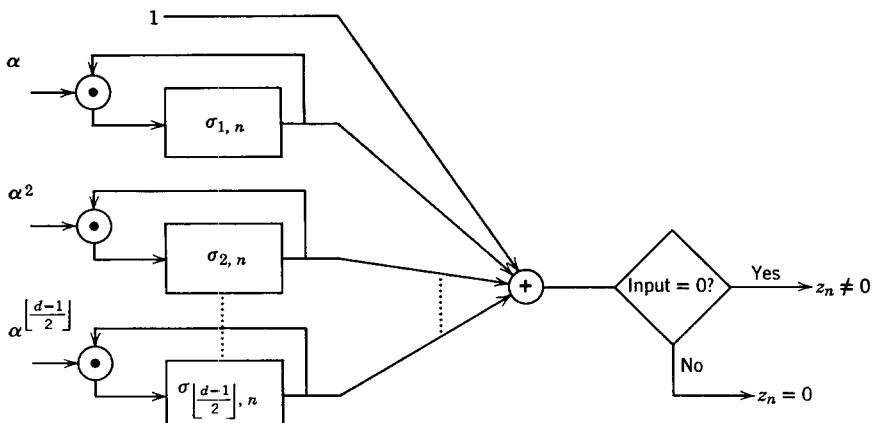
and for each  $n$

$$\sigma_{i,n-1} = \sigma_{i,n} \alpha^i \quad (6.7.66)$$

This suggests using the implementation in Figure 6.7.5 to perform the test in (6.7.63), first for  $n = N - 1$ , then for  $n = N - 2$ , and so forth.

We have already pointed out that, for a binary BCH code, step 4 in the decoding procedure is unnecessary. Thus the received digits can be fed out of the decoder synchronously with the operation of the circuit in Figure 6.7.5, simply adding  $y_n$  to  $z_n$ , modulo 2, as  $n$  goes from  $N - 1$  to 0, yielding the transmitted code word if at most  $\lfloor (d-1)/2 \rfloor$  errors have occurred.

If more than  $\lfloor (d-1)/2 \rfloor$  errors have occurred for a binary code, any one of the following three events might occur. First, the register  $[\sigma(D), l]$  generated in step 2 might have  $l > \lfloor (d-1)/2 \rfloor$ . If  $j = \lfloor (d-1)/2 \rfloor$  in the block diagram of Figure 6.7.4,  $\sigma(D)$  will not be found in this case, but it is trivial to detect the event. It is also possible for  $l$ , as found in step 2, to not exceed  $\lfloor (d-1)/2 \rfloor$ , but for  $\sigma(D)$  not to have  $l$  roots in  $GF(2^m)$ . In this case, fewer than  $l$  corrections will be made in step 3, but the decoded sequence will not be a code word. This



**Figure 6.7.5.** Step 3 of BCH decoding: finding error locations. Register initially loaded with  $\sigma_1, \dots, \sigma_{\lfloor(d-1)/2\rfloor}$ . Then multiplication takes place, and then test for  $z_{N-1} = 0$ ; then multiplication and test for  $z_{N-2} = 0$ ; and so on to  $z_0 = 0$ .

again can be easily detected, either by counting the number of corrections or by checking whether the decoded sequence is a code word. Finally, it is possible that the register  $[\sigma(D), l]$  has  $l \leq \lfloor(d-1)/2\rfloor$  and that  $l$  errors are found in decoding. In this case, the decoded sequence will be a code word and differ from the received sequence in at most  $\lfloor(d-1)/2\rfloor$  positions. The decoding error cannot be detected in this case, but at least we know that the decoder has made the maximum-likelihood decision for a binary symmetric channel.

We next turn to finding the error values (step 4) in the decoding process for nonbinary BCH codes. We defined the polynomial  $A(D)$  in (6.7.22) as

$$A(D) = \sum_{j=1}^e V_j U_j^r \prod_{i \neq j} (1 - U_i D) \quad (6.7.67)$$

Also  $A(D)$  is determined in terms of  $\sigma(D)$  by (6.7.25),

$$A(D) = [\sigma(D)S(D)]_0^{d-2} \quad (6.7.68)$$

$A(D)$  can be calculated directly from (6.7.68) or, more elegantly, it can be incorporated as part of the iterative algorithm for finding  $\sigma(D)$ . In particular, we use the initial conditions  $A_{-1}(D) = -D^{-1}$  and  $A_0(D) = 0$  and for each  $n \geq 0$ , calculate  $A_{n+1}(D)$  from

$$A_{n+1}(D) = A_n(D) - \frac{d_n}{d_{k_n}} D^{n-k_n} A_{k_n}(D) \quad (6.7.69)$$

where  $d_n$  and  $k_n$  are given by (6.7.34) and (6.7.35). This requires two extra registers in the block diagram of Figure 6.7.4 and virtually no extra control logic since the operations on the registers for  $A_n(D)$  and  $D^{n-k_n}A_{k_n}(D)$  are the same as those on the registers for  $C_n(D)$  and  $D^{n-k_n}C_{k_n}(D)$ . The proof that  $[C_n(D)S(D)]_0^{n-1} = A_n(D)$  for each  $n \geq 0$  is almost the same as the proof of Theorem 6.7.3 and is treated in Problem 6.35.

After  $A(D)$  has been found, we see from (6.7.67) that

$$A(U_j^{-1}) = V_j U_j^r \prod_{l \neq j} (1 - U_l U_j^{-1}) \quad (6.7.70)$$

The term on the right can be simplified if we define the derivative of  $\sigma(D) = \sigma_0 + \sigma_1 D + \cdots + \sigma_e D^e$  as

$$\sigma'(D) = \sigma_1 + 2\sigma_2 D + \cdots + e\sigma_e D^{e-1} \quad (6.7.71)$$

This calculation of  $\sigma'(D)$  can be easily instrumented from  $\sigma(D)$ , and if  $q$  is a power of 2, it is simply the odd power terms of  $\sigma(D)$  divided by  $D$ . Since

$$\sigma(D) = \prod_j (1 - U_j D)$$

we also have (see Problem 6.36):

$$\begin{aligned} \sigma'(D) &= - \sum_{j=1}^e U_j \prod_{l \neq j} [1 - U_l D] \\ \sigma'(U_j^{-1}) &= - U_j \prod_{l \neq j} (1 - U_l U_j^{-1}) \end{aligned} \quad (6.7.72)$$

Substituting (6.7.72) into (6.7.70),

$$V_j = - U_j^{1-r} \frac{A(U_j^{-1})}{\sigma'(U_j^{-1})} \quad (6.7.73)$$

Recalling the definitions of  $U_j$  and  $V_j$  in (6.7.16), each nonzero noise digit  $z_n$  is given by

$$z_n = - \frac{\alpha^{n(1-r)} A(\alpha^{-n})}{\sigma'(\alpha^{-n})} \quad (6.7.74)$$

Each of the three terms on the right-hand side of (6.7.74) can be calculated successively for  $n$  going from  $N - 1$  to 0 by the same type of circuit as in Figure 6.7.5.

This concludes our discussion of decoding for BCH codes. The major point to be remembered is that, although the decoding is conceptually complicated, it is very simple in terms of decoding time and required circuitry. Apart from storage of the received word and a circuit to take the inverse of elements in  $GF(q^m)$ , the amount of hardware required is proportional to

$m d$ . The decoding time in step 2 is proportional to  $m d$  and that in steps 3 and 4 is proportional to  $mN$ .

Let us see what can be said about the behavior of binary BCH codes in the limit as  $N \rightarrow \infty$ . Assume for the moment that  $m(d - 1)/2$  is a good estimate of the number of check digits in the code so that

$$\frac{m(d - 1)}{2N} \approx 1 - R$$

where  $R$  is the rate in binary digits. Since  $m \geq \log_2(N + 1)$ , we see that for fixed  $R$ ,  $(d - 1)/2N$  must approach 0 as  $N$  approaches infinity. Thus the number of errors that can be corrected by the decoding algorithm eventually drops below the expected number of errors on the channel. Peterson (1961) has calculated the exact number of check digits required in a variety of binary BCH codes, and his results indicate that, for fixed  $R$ ,  $(d - 1)/2N$  does decrease toward zero with increasing  $N$ . On the other hand, this decrease occurs at such large values of  $N$  that this limitation is of little practical importance.

The Reed Solomon (1960) codes are a particularly interesting class of BCH codes in which the parameter  $m$  is 1; that is, in which the extension field in which  $\alpha$  is defined is the same as the symbol field for the code letters. In this case, the minimal polynomial of  $\alpha^i$  is simply  $D - \alpha^i$ , so we have

$$g(D) = \prod_{i=r}^{r+d-2} (D - \alpha^i)$$

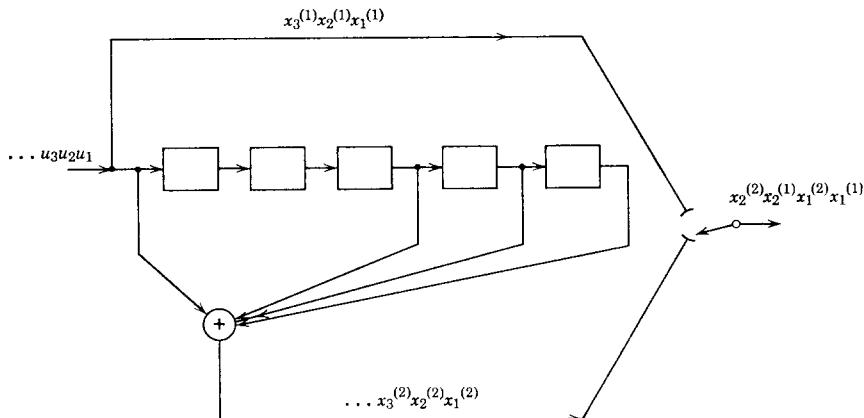
Thus this code has  $d - 1$  check digits and a minimum distance of  $d$ . It is easy to see that no group code with this alphabet size, block length and number of check digits can have a larger minimum distance, since if all the information digits but 1 are chosen to be zero, the resulting code word has at most  $d$  nonzero digits.

Since the block length  $N$  of a Reed-Solomon code is  $q - 1$  or a divisor of  $q - 1$ , it can be seen that these codes are useful only for larger alphabet sizes. They can be used effectively on continuous time channels where the input alphabet is chosen as a large set of waveforms. They have also been used effectively by Forney (1965) in a concatenation scheme where the symbols in the Reed-Solomon code are code words in a smaller, imbedded code. Forney has shown that such codes can be used at transmission rates arbitrarily close to capacity. The error probability is an exponentially decreasing function of the block length, and the decoding complexity is proportional to a small power of the block length. Reed-Solomon codes can also be used directly on a channel with a small input alphabet by representing each letter in a code word by a sequence of channel letters. Such a technique

is useful on channels where the errors are clustered, since the decoder operation depends only on the number of sequences of channel outputs that contain errors.

### 6.8 Convolutional Codes and Threshold Decoding

Convolutional codes, to be defined in this section, differ from all the codes previously discussed in that they are not block codes. Before defining these codes, we shall consider the simple example of a convolutional code illustrated in Figure 6.8.1. Each unit of time, a new binary source digit  $u_n$  enters



*Figure 6.8.1. Example of convolutional code.*

the encoder and each preceding source digit moves one place to the right in the shift register. The new source digit is transmitted directly on the channel as  $x_n^{(1)} = u_n$ . Following each such information digit is a check digit, as

$$x_n^{(2)} = u_n \oplus u_{n-3} \oplus u_{n-4} \oplus u_{n-5} \quad (6.8.1)$$

$$= x_n^{(1)} \oplus x_{n-3}^{(1)} \oplus x_{n-4}^{(1)} \oplus x_{n-5}^{(1)} \quad (6.8.2)$$

Assuming the shift register to contain zeros initially and assuming transmission to start with  $u_1$ , we have  $u_n = 0$ ,  $x_n^{(1)} = 0$ , and  $x_n^{(2)} = 0$  for  $n \leq 0$ .

We now describe a very simple decoding technique, threshold decoding, for decoding this code after transmission on a binary channel. Let

$\mathbf{x} = x_1^{(1)}, x_1^{(2)}, x_2^{(1)}, x_2^{(2)}, \dots$  be the transmitted sequence,

$\mathbf{y} = y_1^{(1)}, y_1^{(2)}, y_2^{(1)}, y_2^{(2)}, \dots$  be the received sequence and

$\mathbf{z} = z_1^{(1)}, z_1^{(2)}, z_2^{(1)}, z_2^{(2)}, \dots$  be the error sequence, where

$\mathbf{z} = \mathbf{y} \oplus \mathbf{x}$ . As with parity check codes, we define the syndrome,  $\mathbf{S} = S_1, S_2, \dots$ , by

$$S_n = y_n^{(2)} \oplus y_n^{(1)} \oplus y_{n-3}^{(1)} \oplus y_{n-4}^{(1)} \oplus y_{n-5}^{(1)} \quad (6.8.3)$$

Thus  $S_n = \mathbf{0}$  if the  $n$ th parity check, as recomputed at the receiver, is satisfied and  $S_n = \mathbf{1}$  otherwise. From (6.8.2), we also have

$$S_n = z_n^{(2)} + z_n^{(1)} + z_{n-3}^{(1)} + z_{n-4}^{(1)} + z_{n-5}^{(1)} \quad (6.8.4)$$

Spelling out these equations for  $1 \leq n \leq 6$ , we have

$$\begin{aligned} S_1 &= z_1^{(2)} \oplus z_1^{(1)} \\ S_2 &= z_2^{(2)} \oplus z_2^{(1)} \\ S_3 &= z_3^{(2)} \oplus z_3^{(1)} \\ S_4 &= z_4^{(2)} \oplus z_4^{(1)} \oplus z_1^{(1)} \\ S_5 &= z_5^{(2)} \oplus z_5^{(1)} \oplus z_2^{(1)} \oplus z_1^{(1)} \\ S_6 &= z_6^{(2)} \oplus z_6^{(1)} \oplus z_3^{(1)} \oplus z_2^{(1)} \oplus z_1^{(1)} \end{aligned} \quad (6.8.5)$$

We now turn our attention to decoding the first information digit,  $u_1$ ; this is equivalent to determining whether  $z_1^{(1)}$  is one or zero. Notice that  $S_1, S_4, S_5$ , and  $S_6$  all involve  $z_1^{(1)}$  directly. For example, if  $z_1^{(1)} = \mathbf{1}$  and no other errors occurred,  $S_1, S_4, S_5$ , and  $S_6$  all have the value  $\mathbf{1}$ , whereas if no errors occurred they all have the value  $\mathbf{0}$ . This suggested the following decoding strategy: decide  $z_1^{(1)} = \mathbf{1}$  if the majority of the elements  $S_1, S_4, S_5$ , and  $S_6$  have the value  $\mathbf{1}$ ; otherwise decide  $z_1^{(1)} = \mathbf{0}$ .

There is a simple improvement that can be made in the above strategy. We observe that  $z_2^{(1)}$  enters into the calculation of both  $S_5$  and  $S_6$ , and thus, if both  $z_1^{(1)} = \mathbf{1}$  and  $z_2^{(1)} = \mathbf{1}$ ,  $S_5$  and  $S_6$  will be  $\mathbf{0}$ ,  $S_1$  and  $S_4$  will be  $\mathbf{1}$ , and an incorrect decoding will result. We can avoid this difficulty by combining  $S_2$  and  $S_5$ , giving us

$$\begin{aligned} S_1 &= z_1^{(2)} \oplus z_1^{(1)} \\ S_4 &= z_4^{(2)} \oplus z_4^{(1)} \oplus z_1^{(1)} \\ S_5 \oplus S_2 &= z_5^{(2)} \oplus z_5^{(1)} \oplus z_2^{(2)} \oplus z_1^{(1)} \\ S_6 &= z_6^{(2)} \oplus z_6^{(1)} \oplus z_3^{(1)} \oplus z_2^{(1)} \oplus z_1^{(1)} \end{aligned} \quad (6.8.6)$$

A set of linear combinations of the noise digits is said to be *orthogonal* on one of the noise digits if that digit appears (with a nonzero coefficient) in each of the set of linear combinations and no other digit appears (with a nonzero coefficient) in more than one of the linear combinations. Thus the set of four linear combinations on the right-hand side of (6.8.6) is orthogonal on  $z_1^{(1)}$ .

Observe that, if  $z_1^{(1)} \neq 0$  and all other  $z_n^{(i)}$  appearing in the set are 0, then all four linear combinations have a nonzero value. If one additional error occurs (that is,  $z_n^{(i)} \neq 0$  for one other variable), the orthogonality assures us that at least three of the linear combinations are nonzero. On the other hand, if  $z_1^{(1)} = 0$  and at most two other noise digits in the set are nonzero, then at most two of the linear combinations are nonzero. Thus, if a decoder calculates the values of terms on the left-hand side of (6.8.6) and takes  $z_1^{(1)}$  as 1 when a majority of the terms are 1 and takes  $z_1^{(1)} = 0$  otherwise, correct decoding of  $z_1^{(1)}$  will result whenever at most two of the noise digits in (6.8.6) are nonzero. This principle immediately generalizes to the following theorem, which is stated so as to be applicable in an arbitrary field, even though we are primarily concerned with  $GF(2)$  here.

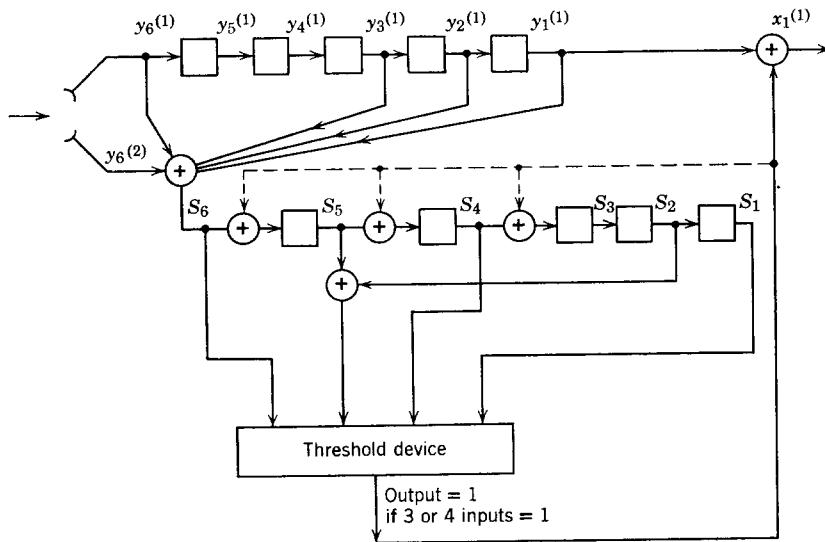


Figure 6.8.2. Threshold decoder.

**Theorem 6.8.1.** For an arbitrary positive integer,  $e$ , assume that a decoder can calculate the values of a set of  $2e$  linear combinations of a set of noise digits, orthogonal on  $z_1^{(1)}$ . Then, if at most  $e$  of the noise digits appearing in the linear combinations are nonzero, the following rule decodes  $z_1^{(1)}$  correctly: If over half of the linear combinations have the same value,  $\alpha$ , decode  $z_1^{(1)} = \alpha$ ; otherwise decode  $z_1^{(1)} = 0$ .

The circuit shown in Figure 6.8.2, called a *threshold decoder*, performs the decoding of the first information digit for the code of Figure 6.8.1 using this technique.

The dotted lines in Figure 6.8.2 are concerned with decoding successive information digits, to which we now turn. If all the subscripts on the left-hand side of (6.8.6) are increased by 1, we obtain

$$\begin{aligned} S_2 &= z_2^{(2)} \oplus z_2^{(1)} \\ S_5 &= z_5^{(2)} \oplus z_5^{(1)} \oplus z_2^{(1)} \oplus z_1^{(1)} \\ S_6 \oplus S_3 &= z_6^{(2)} \oplus z_6^{(1)} \oplus z_3^{(2)} \oplus z_2^{(1)} \oplus z_1^{(1)} \\ S_7 &= z_7^{(2)} \oplus z_7^{(1)} \oplus z_4^{(1)} \oplus z_3^{(1)} \oplus z_2^{(1)} \end{aligned} \quad (6.8.7)$$

It should be observed that, except for  $z_1^{(1)}$ , these equations are orthogonal on  $z_2^{(1)}$ , and the correction can be performed by Figure 6.8.2, shifting the syndrome digits to the right after correcting  $z_1^{(1)}$ . We can now see the purpose of the dotted lines in Figure 6.8.2. They represent feedback links that remove the effect of an error from the syndrome after it has been corrected. Thus, if  $z_1^{(1)} = 1$  and is decoded correctly, the decoder sets  $z_1^{(1)} = 0$  and changes  $S_4$ ,  $S_5$ , and  $S_6$  accordingly. Thus each information digit in the sequence will be decoded correctly providing that no more than two errors appear in the orthogonal equations and no previous decoding errors occur.

What happens to the decoder in Figure 8.6.2 *after* a decoding error has been made is a considerably more difficult question. It has been found experimentally that, after a decoding error, the decoder will typically make about one more error in decoding the next five or so information digits, and then correct decoding will begin again. It has also been proven theoretically (Massey, 1964) for this particular code that a long enough string of error-free digits into the decoder after a decoding error will restore the decoder to correct decoding. This tendency of decoding errors to propagate is characteristic of decoding schemes for convolutional codes. For very simple codes and decoders, such as the one discussed, this propagation effect is not very serious, usually involving only a short burst of decoding errors. As the constraint length of the code becomes longer, however, and as the decoding scheme becomes more sophisticated, this propagation becomes more serious. On the other hand, the more serious this propagation problem becomes, the easier it becomes for the decoder to recognize that it is making decoding errors. If the decoder can communicate back to the transmitter, the transmitter can then repeat the data that has been lost.\* Alternatively, the encoder can periodically insert a known sequence of 0's into the encoder, after which point the decoder can start decoding afresh.

The example of a convolutional code in Figure 6.8.1 can now be generalized in three directions. First, we can make the length of the shift register arbitrary

\* For particular strategies to accomplish this when the feedback link is noisy, see Wozen-craft and Horstein (1961), and Metzner and Morgan (1960).

and tap the register at arbitrary places to form check digits. Second, we can make the source digits and channel digits elements of an arbitrary Galois field. Finally, we can generalize the codes to rates other than one source digit per two channel digits. To accomplish this last generalization, we divide the source digits into blocks of a given length  $\lambda$ . For each  $\lambda$  source digits, the encoder produces a given number  $v$  of channel digits as shown in Figure

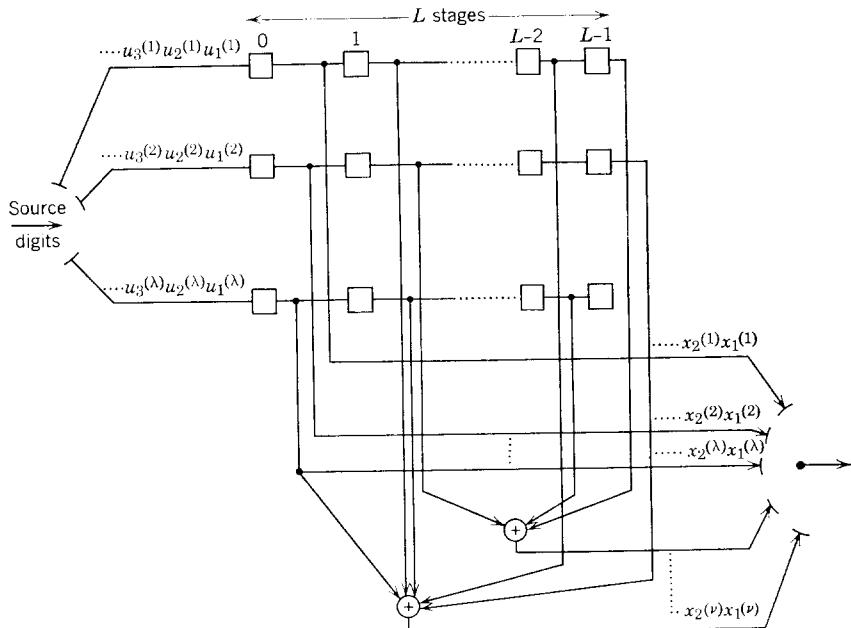


Figure 6.8.3. General systematic convolutional encoder.

6.8.3. If we denote the source sequence by  $u_1^{(1)}, u_1^{(2)}, \dots, u_1^{(\lambda)}, u_2^{(1)}, u_2^{(2)}, \dots, u_2^{(\lambda)}, \dots$  and the channel sequence by  $x_1^{(1)}, x_1^{(2)}, \dots, x_1^{(v)}, x_2^{(1)}, \dots$  then the rule for generating channel digits from source digits can be written as

$$x_n^{(i)} = \sum_{l=0}^{L-1} \sum_{j=1}^{\lambda} g_{j,i}(l) u_{n-l}^{(j)}; \quad 1 \leq i \leq v \quad (6.8.8)$$

where  $L$  is the length of the shift registers as shown in Figure 6.8.3 and the elements  $g_{j,i}(l)$  determine the connections between the shift registers and the adders. In the modulo-2 case,  $g_{j,i}(l) = 1$  if stage  $l$  of the  $j$ th shift register is connected to the adder forming the  $i$ th stream of channel digits. For the general case, all the elements  $u_l^{(j)}$ ,  $g_{j,i}(l)$ , and  $x_n^{(i)}$  are from a given Galois field, and the inputs to the adders (see Figure 6.8.3) must be weighted by the

appropriate elements  $g_{j,i}(l)$ . In the example of Figure 6.8.1,  $\lambda = 1$ ,  $v = 2$ ,  $L = 6$ , and  $g_{1,1}(0) = g_{1,2}(0) = g_{1,2}(3) = g_{1,2}(4) = g_{1,2}(5) = 1$ .

A convolutional code is called *systematic* if the first  $\lambda$  channel digits in each block of  $v$  are the same as the corresponding  $\lambda$  source digits. In that case, for  $i \leq \lambda$ ,  $g_{j,i}(l)$  is **1** for  $l = 0$ ,  $i = j$ , and is **0** otherwise. Both the encoder of Figure 6.8.1 and that of Figure 6.8.3 are systematic.

The *constraint length* of a convolutional code is defined to be  $N = vL$ ; this is the number of channel digits that come out of the encoder between when a given source digit enters the encoder and when it passes out of the encoder. The constraint length of a convolutional code plays a similar role to that of the block length of a block code.

Threshold decoding can be applied to any convolutional code (and, in fact, to block parity check codes as well). The only question is how many orthogonal linear combinations can be found on each digit. This is mostly a matter of trial and error. A number of convolutional codes of varying rate and blocklength and their orthogonalization rules are tabulated by Massey (1963). Threshold decoding is remarkably efficient for correcting errors at short constraint lengths, but it appears reasonably certain that arbitrarily small error probabilities cannot be achieved by increasing the constraint length. In Section 6.9, we discuss another decoding technique, sequential decoding, for convolutional codes by which the error probability can be made arbitrarily small by increasing the constraint length.

## 6.9 Sequential Decoding

The idea of sequential decoding can best be presented in terms of a simple example. Consider the binary convolutional encoder shown in Figure 6.9.1.

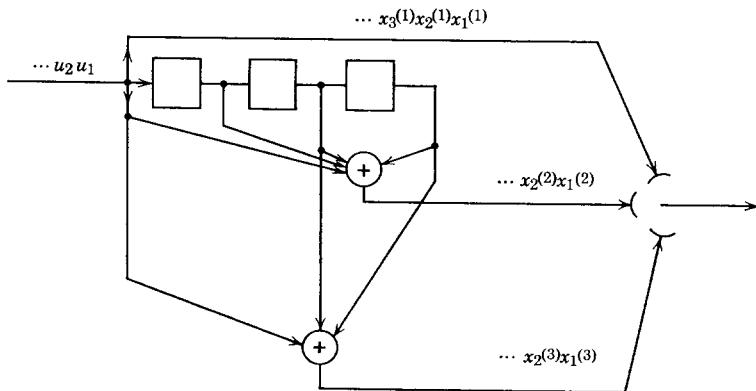


Figure 6.9.1. Example of convolutional code.

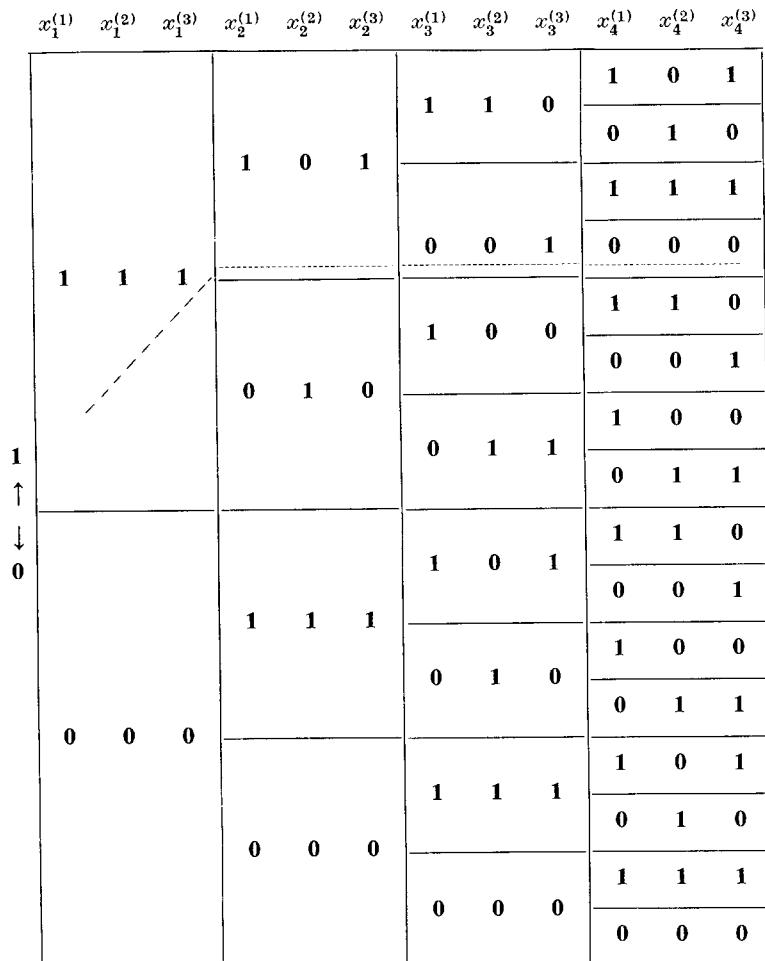


Figure 6.9.2. Tree structure of convolutional code.

As usual, we assume the shift register to contain all zeros before the arrival of the first source digit  $u_1$ . The first three channel digits,  $x_1^{(1)}, x_1^{(2)}, x_1^{(3)}$  are then determined solely by  $u_1$ . The next three channel digits are functions of both  $u_1$  and  $u_2$ , the following three are functions of  $u_1, u_2$ , and  $u_3$ , and so forth. This dependence imposes a treelike structure on the set of encoded channel sequences which is brought out in Figure 6.9.2.

The left-most binary triplets in Figure 6.9.2 correspond to the responses of the encoder to an initial 1 or 0 into the encoder. Branching off to the right

from the response **(111)** to an initial **1** are the responses to a **1** or **0** following an initial **1**, and so forth. For example, the dotted line in Figure 6.9.2 denotes the response to  $u_1 = 1, u_2 = 1, u_3 = 0, u_4 = 0$ .

We now investigate, in a qualitative sense, how this tree structure can be used in decoding. We subsequently shall define sequential decoding as a formalization of the rather common-sense approach that we now take.

Suppose that the encoder of Figure 6.9.1 is used on a binary symmetric channel and that the first twelve digits of the received sequence are **101 101 001 000**. We know from the tree in Figure 6.9.2 that the first three transmitted digits are either **111** or **000**. Thus, on the basis of the first three received digits, we can tentatively hypothesize that **111** was transmitted, corresponding to the upper fork at the first branch of the tree. Given this hypothesis, the next three digits must be one of the top two entries under  $x_2^{(1)}x_2^{(2)}x_2^{(3)}$  in Figure 6.9.2; **101**, corresponding to the upper fork, is clearly the more reasonable hypothesis. Continuing in the same way, we hypothesize the third triplet of transmitted digits to be **001** and the fourth to be **000**. These hypotheses are shown by dotted line in Figure 6.9.2. What we have done, in effect is to decode digit by digit, using our hypotheses on each triplet of digits to reduce the number of choices available on the next triplet. For the particular example shown, the agreement between the hypothesized transmitted sequence and the received sequences on all digits past the second tend to verify that the earlier hypotheses were correct.

Let us consider a second example now that illustrates what happens if an incorrect hypothesis is made. Suppose the same initial sequence, **111 101 001 000**, is transmitted, but **010 101 001 000 ···** is received. The decoder, not knowing what was transmitted, will hypothesize that the first three transmitted digits are **000**, corresponding to the lower branch at the first branch point. The next two hypotheses will then be **111** and **101**. In these first nine digits, the hypothesized transmitted sequence and the received sequence differ in three digits. What is happening is that after the decoder once makes an incorrect hypothesis, it is forced to make subsequent choices between sequences that have no relation to the received sequence. Thus, eventually, the decoder will usually be able to recognize that it has probably made an incorrect hypothesis. It can then go back and try alternative hypotheses, and presumably decode correctly. Thus each hypothesis simplifies future hypotheses by reducing the number of choices available and at the same time provides additional data on the correctness of earlier hypotheses.

A sequential decoder, in general, is a decoder that decodes a code with a tree structure by making tentative hypotheses on successive branches of the tree and by changing these hypotheses when subsequent choices indicate an earlier incorrect hypothesis. Unfortunately, the extreme simplicity of this

concept disappears somewhat when we discuss explicit strategies to determine when the decoder should continue to make new hypotheses and when the decoder should return to change old hypotheses. For our purposes, there are three requirements on a search strategy for sequential decoding. First, the strategy eventually must correctly decode the source sequence with high probability. Second, the amount of computation required by the decoder must not be excessive. Finally (and this requirement disappears for an operational system), the strategy must be amenable to analysis. The first such strategy (or algorithm) was devised by Wozencraft (1957), who is also responsible for the concept of sequential decoding. We shall restrict our attention here to a later improved algorithm due to Fano (1963).

We shall assume the channel to be a discrete memoryless channel with transition probabilities denoted by  $P(j | k)$ . The encoder will be described in greater detail later, but briefly it consists of three parts. First there is a binary convolutional encoder as in Figure 6.8.3 and Equation 6.8.8. Each unit of time the binary encoder accepts  $\lambda$  binary digits from the source and generates  $a\nu$  binary output digits where  $a$  and  $\nu$  are integers. Second, a fixed binary sequence is added to this binary output. Third, the sum is segmented into subblocks of  $a$  digits each and each subblock is mapped into a channel input letter by a mapping such as that in Figure 6.2.1.\* We see that with this encoding,  $\nu$  channel digits are generated for each  $\lambda$  source digits and the rate of the code, in natural units per channel digit is

$$R = \frac{\lambda}{\nu} \ln 2 \quad (6.9.1)$$

This encoding can be represented by a tree structure as in Figure 6.9.2, but there are  $2^\lambda$  choices at each branch point rather than the 2 choices of the figure, and each branch consists of  $\nu$  channel digits.

Now let  $\mathbf{x}_l = (x_1^{(1)}, \dots, x_1^{(\nu)}, x_2^{(1)}, \dots, x_l^{(\nu)})$  be the first  $\nu l$  digits of an encoded sequence and let

$$\mathbf{y}_l = (y_1^{(1)}, \dots, y_l^{(\nu)})$$

be the first  $\nu l$  digits of the received sequence. Define the function  $\Gamma(\mathbf{x}_l; \mathbf{y}_l)$  by

$$\Gamma(\mathbf{x}_l; \mathbf{y}_l) = \sum_{n=1}^l \sum_{i=1}^{\nu} \left[ \ln \frac{P(y_n^{(i)} | x_n^{(i)})}{\omega(y_n^{(i)})} - B \right] \quad (6.9.2)$$

\* This mapping destroys some of the information in the encoded binary sequence, but will not destroy any of the source information if the code is properly chosen. The coding theorem in Section 6.2 should convince one of this for block codes, and a similar argument will be given subsequently for convolutional codes.

In this expression  $B$  is an arbitrary bias term to be selected later and  $\omega(j)$  is the nominal probability of the  $j$ th letter of the channel output alphabet

$$\omega(j) = \sum_{k=0}^{K-1} Q(k) P(j | k) \quad (6.9.3)$$

where  $Q(k)$  is the relative frequency of letter  $k$  in the mapping from binary digits to channel inputs (see (6.2.6)). We shall call  $\Gamma(\mathbf{x}_l; \mathbf{y}_l)$  the *value* of the hypothesis  $\mathbf{x}_l$  and use it as a measure of the closeness of fit of  $\mathbf{x}_l$  to the received  $\mathbf{y}_l$ . This is a reasonable measure, since for a given  $\mathbf{y}_l$ ,  $\Gamma$  is an increasing function of  $P_l(\mathbf{y}_l | \mathbf{x}_l)$ . It is somewhat difficult at this point to see the purpose of  $\omega(y_n)$  in (6.9.2), but roughly the purpose is to reduce the dependence of the source on the unconditional probability of the output sequence.

For a binary symmetric channel with crossover probability  $\epsilon$ , the last two parts of the encoder described above can be omitted and (6.8.8) completely describes the encoding. In this case  $\omega(y_n)$  is a constant and  $\Gamma(\mathbf{x}_l; \mathbf{y}_l)$  simplifies to

$$\Gamma(\mathbf{x}_l; \mathbf{y}_l) = -d(\mathbf{x}_l; \mathbf{y}_l) \ln \frac{1 - \epsilon}{\epsilon} + \nu l \{\ln [2(1 - \epsilon)] - B\}$$

where  $d(\mathbf{x}_l; \mathbf{y}_l)$  is the Hamming distance between  $\mathbf{x}_l$  and  $\mathbf{y}_l$ .

In terms of the value  $\Gamma$ , the function of the decoder is to hypothesize in such a way that  $\Gamma(\mathbf{x}_l; \mathbf{y}_l)$  increases with  $l$ . When  $\Gamma$  starts to decrease with increasing  $l$ , presumably the decoder is on the wrong path and some searching is in order. This idea is brought out more clearly by Figure 6.9.3 where  $\Gamma(\mathbf{x}_l; \mathbf{y}_l)$  is sketched, using the tree structure of the possible choices of  $\mathbf{x}_l$ . The figure, which is called a received value tree, corresponds to the encoder of Figure 6.9.1 with the received sequence **010 101 001 000**.

The nodes in Figure 6.9.3 are labeled in terms of the information sequences leading to them. That is, each node at a depth  $l$  into the tree corresponds to a different information sequence  $\mathbf{u}_l$  of  $l/2$  binary digits. For example, the node in the upper right-hand corner of the figure corresponds to the information sequence  $\mathbf{u}_4 = 1100$  which leads to the transmitted sequence  $\mathbf{x}_4 = 111 101 001 000$ .

For an arbitrary convolutional code and an arbitrary received sequence, the decoder can *in principle* construct a received value tree as in Figure 6.9.3. The decoder cannot construct such a tree *in practice*, however, because of the exponential growth with  $l$  of the number of nodes in the tree. What the decoder can do is to maintain a replica of the convolutional encoder.

For any given hypothesized sequence of information digits,  $\mathbf{u}_l$ , the decoder can run  $\mathbf{u}_l$  through the replica encoder, find the associated  $\mathbf{x}_l$ , and calculate  $\Gamma(\mathbf{x}_l; \mathbf{y}_l)$ , which is the value of the node associated with  $\mathbf{u}_l$ .

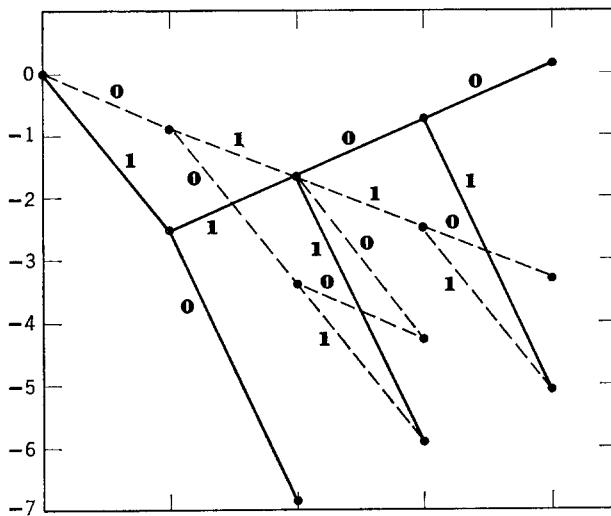


Figure 6.9.3. Received value tree; BSC;  $\varepsilon = 0.15$ ; encoder of Figure 6.9.1;  $y_4 = 010\ 101\ 001\ 000$ ;  $B = \frac{1}{3} \ln 2$ .

The decoding algorithm that we shall describe is a set of rules for moving from one node to another. We shall allow only three kinds of moves: forward, lateral, and backward. On a *forward move*, the decoder goes one branch to the right in the received value tree from the previously hypothesized node. Instrumentally, this corresponds to shifting the replica encoder shift registers to the right, inserting  $\lambda$  new hypothesized information digits at the left. Since the new information sequence  $\mathbf{u}_{i+1}$  differs from the old only in having  $\lambda$  digits added to it, the new value,  $\Gamma(\mathbf{x}_{i+1}; \mathbf{y}_{i+1})$  can be found from  $\Gamma(\mathbf{x}_i; \mathbf{y}_i)$  by adding one more term to the summation in (6.9.2):

$$\Gamma(\mathbf{x}_{i+1}; \mathbf{y}_{i+1}) = \Gamma(\mathbf{x}_i; \mathbf{y}_i) + \sum_{i=1}^v \left[ \ln \frac{P(y_{i+1}^{(i)} | x_{i+1}^{(i)})}{\omega(y_{i+1}^{(i)})} - B \right] \quad (6.9.4)$$

The digits involved in this calculation are simply the  $v$  channel input digits coming out of the replica encoder. A *lateral move* is a move from one node to another node differing only in the last branch of the tree. Instrumentally, this corresponds to changing the left-most  $\lambda$  digits in the replica encoder shift register. Again the change in value from one node to the other is determined by the change in the last  $v$  channel input digits. A *backward move* is a move one branch to the left in the received value tree. Instrumentally, this corresponds to shifting the encoder shift registers to the left, reinserting the last  $\lambda$  digits to pass out of the shift register on the right. The new value is

calculated from the old value by subtracting off the last term in the summation over  $n$  in (6.9.2). Thus, for each possible move, the change in value from the old node to the new is a function only of the last  $r$  hypothesized channel inputs.

The algorithm to be used in moving from one node to another is a modification of an algorithm due to Fano and is given in Figure 6.9.4. The rules involve the value  $\Gamma_l$  of the current node being hypothesized, the value  $\Gamma_{l-1}$  of the node one branch to the left of the current node, and a *threshold*  $T$ . The value of the threshold  $T$  is constrained to change in increments of some fixed number  $\Delta$ , the changes being determined by the algorithm.

| Rule | Conditions on Node |  | Action to be Taken |                   |
|------|--------------------|--|--------------------|-------------------|
|      | Previous Move      | Comparison of $\Gamma_{l-1}$ and $\Gamma_l$ With Initial Threshold | Final Threshold    | Move              |
| 1    | $F$ or $L$         | $\Gamma_{l-1} < T + \Delta$ $\Gamma_l \geq T$                      | Raise*             | $F\dagger$        |
| 2    | $F$ or $L$         | $\Gamma_{l-1} \geq T + \Delta$ $\Gamma_l \geq T$                   | No change          | $F\dagger$        |
| 3    | $F$ or $L$         | any $\Gamma_{l-1}$ $\Gamma_l < T$                                  | No change          | $L$ or $B\dagger$ |
| 4    | $B$                | $\Gamma_{l-1} < T$ any $\Gamma_l$                                  | Lower by $\Delta$  | $F\dagger$        |
| 5    | $B$                | $\Gamma_{l-1} \geq T$ any $\Gamma_l$                               | No change          | $L$ or $B\dagger$ |

\* Add  $j\Delta$  to threshold, where  $j$  is chosen to satisfy  $\Gamma_l - \Delta < T + j\Delta \leq \Gamma_l$ .

† Move *forward* to first of  $2^\lambda$  nodes stemming from current node (assuming pre-determined ordering of the  $2^\lambda$  nodes).

‡ Move *laterally* to next node differing from current node only in the final branch (assuming the same ordering above); if the current node is the last of the  $2^\lambda$  nodes, move backward.

Figure 6.9.4. Rules of motion for decoder

The initial conditions at the beginning of decoding are adjusted as follows: the initial node hypothesized is the origin  $\mathbf{u}_0$ , corresponding to all zeros in the shift register with no information digits yet hypothesized. We take  $\Gamma_0$  to be 0, and by convention  $\Gamma_{-1} = -\infty$ . The initial threshold is undefined, but the decoder by convention follows rule 1 on this initial hypothesis with the final threshold set at 0.

It can be seen that a forward move could take place to any of the  $2^\lambda$  nodes one branch to the right of the current node. We assume a predetermined ordering among the nodes, a forward move always occurring to the first node in order, and a lateral move to the next node in order after the current one. The ordering is immaterial to the analysis, but here we shall assume the ordinary numerical ordering according to the hypothesized information

subsequences of length  $\lambda$ , trying **00** ··· **0** first, **0** ··· **01** next, and **11** ··· **1** last.\*

We shall see later that rule 1 of the decoding algorithm is the rule that normally applies when the decoder is hypothesizing a node corresponding to the actual transmitted sequence and when the noise is not severe. In these circumstances the threshold will be raised so that it is less than  $\Delta$  below the value of the node. After a forward move, the value of this previous node will

| $u_l$     | Final Threshold | Move | $u_l$       | Final Threshold | Move |
|-----------|-----------------|------|-------------|-----------------|------|
| —         | 0               | F    | <b>010</b>  | -3              | L    |
| <b>0</b>  | 0               | L    | <b>011</b>  | -3              | F    |
| <b>1</b>  | 0               | B    | <b>0110</b> | -3              | L    |
| —         | -1.5            | F    | <b>0111</b> | -3              | B    |
| <b>0</b>  | -1.5            | F    | <b>011</b>  | -3              | B    |
| <b>00</b> | -1.5            | L    | <b>01</b>   | -3              | B    |
| <b>01</b> | -1.5            | B    | <b>0</b>    | -3              | L    |
| <b>0</b>  | -1.5            | L    | <b>1</b>    | -3              | F    |
| <b>1</b>  | -1.5            | B    | <b>10</b>   | -3              | L    |
| —         | -3              | F    | <b>11</b>   | -3              | F    |
| <b>0</b>  | -3              | F    | <b>110</b>  | -1.5            | F    |
| <b>00</b> | -3              | L    | <b>1100</b> | 0               | F    |
| <b>01</b> | -3              | F    |             |                 |      |

Figure 6.9.5. Record of decoder search for Figure 6.9.3 ( $\Delta = 1.5$ ).

appear in the algorithm as  $\Gamma_{l-1}$ , and the fact that  $\Gamma_{l-1} < T + \Delta$  assures us that if the new value  $\Gamma_l$  is above the old value  $\Gamma_{l-1}$ , then rule 1 can apply again and the threshold can be raised again. If the wrong node is initially hypothesized, then presumably the value of the node will be below the threshold and lateral moves will occur until the correct node is hypothesized and the threshold will again be raised.

When the noise is more severe, the behavior of the algorithm is considerably more complicated, as is illustrated by Figure 6.9.5. Roughly what happens is that the decoder tries to find a path in the received value tree that stays above the current threshold. As we shall see later, if no such paths exist, the decoder is forced backward to the node where the threshold was set to its current value. On that node, the threshold is lowered (by rule 4), and the decoder again moves forward, attempting to find a path remaining above this reduced threshold. The purpose of the restriction  $\Gamma_{l-1} < T + \Delta$

\* The original Fano (1963) algorithm orders the nodes in order of decreasing  $I'$  value. This generally decreases the number of nodes that must be hypothesized but increases the amount of computation required per hypothesis due to the computation of the ordering.

in rule 1 is to prevent the threshold from being raised again on one of these nodes that have been previously hypothesized. In fact, if this restriction were not imposed, the decoder would quickly find itself in a loop, hypothesizing the same node again and again with the same threshold.

Before making these ideas more precise, some additional terminology is necessary. An *F hypothesis* is a hypothesis to which rules 1, 2, or 4 apply (that is, for which the subsequent move is forward). The *antecedents* of a node  $\mathbf{u}_l$  are the nodes connecting  $\mathbf{u}_l$  with the origin; that is, the nodes corresponding to prefixes of the information sequence  $\mathbf{u}_l$ . The *path values*  $\Gamma_0, \dots, \Gamma_l$  associated with a node  $\mathbf{u}_l$  are the values of the antecedents of  $\mathbf{u}_l$  and of  $\mathbf{u}_l$  itself. The *path thresholds*  $T_0, \dots, T_l$  associated with a hypothesis of  $\mathbf{u}_l$  are the final thresholds on the most recent hypotheses of the antecedents of  $\mathbf{u}_l$  ( $T_l$  being the final threshold on the current hypothesis). The *descendents* of a node  $\mathbf{u}_l$  are the nodes for which  $\mathbf{u}_l$  is an antecedent; that is, the nodes branching to the right from  $\mathbf{u}_l$ . The *immediate descendants* of  $\mathbf{u}_l$  are the  $2^k$  descendants one branch removed from  $\mathbf{u}_l$ . The *path* from a node  $\mathbf{u}_i$  to a descendant  $\mathbf{u}_l$  is *above the threshold*  $T$  if the path values satisfy  $\Gamma_j \geq T$  for  $i \leq j \leq l$ .

The following theorem, which is proved in Appendix 6A, describes the operation of the algorithm. The important parts of the theorem, from the standpoint of the subsequent analysis, are parts b and c, and these parts can best be understood by reading the theorem in conjunction with the subsequent discussion.

### Theorem 6.9.1.

(a) The path thresholds  $T_0, \dots, T_l$  and path values  $\Gamma_0, \dots, \Gamma_l$  associated with a hypothesis of a node  $\mathbf{u}_l$  satisfy the following equations for  $0 \leq i \leq l - 1$ .

$$T_i \leq \Gamma_i \quad (6.9.5)$$

$$T_{i+1} \geq T_i \quad (6.9.6)$$

$$T_{i+1} \geq T_i + \Delta \Rightarrow T_i + \Delta > \Gamma_i \quad (6.9.7)$$

$$T_{i+1} \geq T_i + \Delta \Rightarrow T_{i+1} > \Gamma_i \quad (6.9.8)$$

(b) For each node that is ever *F* hypothesized, the final threshold  $T$  on the first *F* hypothesis of the node is related to the value  $\Gamma$  of the node by

$$T \leq \Gamma < T + \Delta \quad (6.9.9)$$

The final threshold on each subsequent *F* hypothesis of the node is  $\Delta$  below what it was on the previous *F* hypothesis of the node.

(c) Let node  $\mathbf{u}$  be *F* hypothesized with final threshold  $T$ . Then, before  $\mathbf{u}$  can be rehypothesized, every descendant of  $\mathbf{u}$  for which the path from  $\mathbf{u}$  is above

$T$  must be  $F$  hypothesized with the final threshold  $T$ . Furthermore, between the hypothesis of  $\mathbf{u}$  and its first rehypothesis, the threshold cannot be lowered below  $T$ .

*Discussion.* For a given infinitely long\* transmitted and received sequence, define the *correct path* as the sequence of node values  $\Gamma_0, \Gamma_1, \dots$  corresponding to the transmitted message sequence. Assume that this correct path fluctuates with an upward drift and every other path in the received value tree fluctuates with a negative drift. For any node  $\mathbf{u}_l$  on the correct path, let  $T_l$  be the final threshold on the first  $F$  hypothesis of  $\mathbf{u}_l$ . From part (b) of the

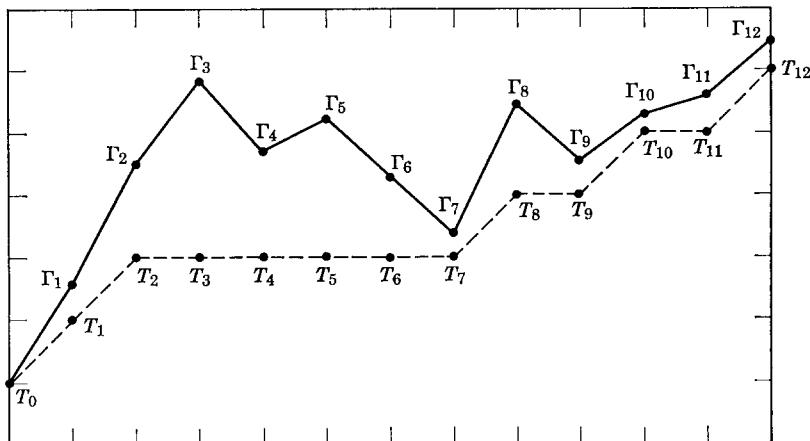


Figure 6.9.6. Path thresholds associated with the hypothesis of a sequence  $u_{12}$ .

theorem,  $T_l \leq \Gamma_l < T_l + \Delta$ . A node  $\mathbf{u}_l$  on the correct path is defined to be a *breakout* if  $\Gamma_i \geq T_l$  for all  $i > l$ . From part (c) of the theorem, we see that if  $\mathbf{u}_l$  is a breakout, then it will not be rehypothesized until after *every* node on the correct path is hypothesized, or in other words, it will never be rehypothesized. Since all incorrect paths fluctuate downward, the decoder eventually progresses forward from one breakout to the next and the algorithm eventually decodes each digit on the correct path. Notice, however, that the decoder is never quite sure when it has “decoded” a node since there is always the possibility that subsequent nodes will drop below any given threshold. This point will be discussed further later.

Figure 6.9.6 illustrates a typical sequence of nodes along a correct path. Assuming that  $\Gamma_l \geq \Gamma_{12}$  for all  $l > 12$ , the nodes at which breakouts occur are  $\mathbf{u}_0, \mathbf{u}_1, \mathbf{u}_7, \mathbf{u}_9, \mathbf{u}_{10}, \mathbf{u}_{11}, \mathbf{u}_{12}$ . The path thresholds for the breakout on  $\mathbf{u}_{12}$

\* If the sequences have finite length, but decoding ceases when the final digit in the sequence is  $F$  hypothesized, then the same discussion is applicable.

are also given in Figure 6.9.6. These thresholds are uniquely specified by part (a) of Theorem 6.9.1 in terms of the path values and the final threshold in the sequence. To see this, start from the final threshold ( $T_{12}$  in Figure 6.9.6) and move horizontally to the left. From (6.9.8), the thresholds drop only when necessary to avoid crossing the path, and from (6.9.7), they drop no more than necessary to avoid crossing the path.

### **Computation for Sequential Decoding**

The number of hypotheses that a sequential decoder will have to make in decoding a sequence of received digits is a random variable, depending upon the received sequence, the source sequence, and the convolutional encoder. If received digits arrive at the decoder at a fixed rate in time and hypotheses are performed at a fixed rate, there will be a waiting line of received digits at the decoder. Clearly the behavior of this waiting line is of central importance in determining how well sequential decoding works.

In order to gain some insight into how this waiting line behaves, it is convenient to visualize the received value tree corresponding to a given source sequence, encoder, and received sequence as consisting of a correct path, corresponding to the actual source sequence, and a set of trees of incorrect paths, a different tree growing from each node of the correct path. We define  $W_n$  as the number of  $F$  hypotheses made on the  $n$ th node of the correct path and on all nodes in the tree of incorrect paths stemming from this  $n$ th node. The sum over  $n$  of  $W_n$  is the total number of  $F$  hypotheses required to decode the sequence and we can interpret  $W_n$  as the number of  $F$  hypotheses required to decode the  $n$ th subblock. If the  $n$ th node on the correct path is a breakout, then all the  $F$  hypotheses entering into  $W_0, \dots, W_{n-1}$  take place before any of the  $F$  hypotheses entering into  $W_n, W_{n+1}, \dots$ .

In what follows, we shall be concerned with analyzing the distributions of the random variables  $W_n$ . This will give us a great deal of insight into the statistical behavior of the waiting line; if the  $W_n$  have a significant probability of being very large, then the waiting line will tend to be long. The justification for considering only  $F$  hypotheses and ignoring lateral and backward moves is as follows. Each node has  $2^\lambda$  immediate descendants and only one lateral or backward move from each may occur per  $F$  hypothesis of the given node. Thus for any length  $l$  into the tree, the total number of lateral and backward moves at length  $l$  is no more than  $2^\lambda$  times the number of forward moves at length  $l - 1$ . Summing over  $l$ , the total number of hypotheses at any time is at most  $2^\lambda + 1$  times the number of  $F$  hypotheses.

In light of the proof of the coding theorem, it should not come as any surprise that the random variable  $W_n$  is most easily treated for an ensemble of codes rather than for some particular code. For each code in the ensemble that we wish to consider, the encoded sequences are generated by passing

the message sequence through a binary, nonsystematic convolutional encoder, then adding an arbitrary binary sequence, and then transforming into channel input letters. Specifically, the binary message sequence is split up into subblocks of  $\lambda$  digits each,  $u_1^{(1)}, \dots, u_1^{(\lambda)}, u_2^{(1)}, \dots, u_2^{(\lambda)}, \dots$ . Corresponding to each  $\lambda$  input digits, the convolutional encoder has  $av$  output digits (for suitably chosen integers  $a$  and  $v$ ), forming a binary sequence  $s_1^{(1)}, \dots, s_1^{(av)}, s_2^{(1)}, \dots$ . As in (6.8.8) (and Figure 6.8.3) the outputs are generated from the inputs by the rule

$$s_n^{(i)} = \sum_{l=0}^{L-1} \sum_{j=1}^{\lambda} g_{j,i}(l) u_{n-l}^{(j)}; \quad 1 \leq i \leq av \quad (6.9.10)$$

where  $g_{j,i}(l) = 0$  for  $l < 0$ .

For analytic simplicity, we take  $L$  (the code constraint length in subblocks) to be infinite. Later, when discussing error probability, we shall of course consider finite constraint lengths. The binary convolutional output  $s$  is next added, modulo 2, to an arbitrary binary sequence  $v$ , forming the sequence  $s_1^{(1)} \oplus v_1^{(1)}, \dots, s_1^{(av)} \oplus v_1^{(av)}, s_2^{(1)} \oplus v_2^{(1)}, \dots$ . This sequence is then broken into subblocks of  $a$  digits each, and each  $a$ -digit binary sequence is mapped into a channel input letter by a mapping as in Figure 6.2.1, using the  $k$ th letter of the channel input alphabet with relative frequency  $Q(k)$ , as in (6.2.6). As in (6.2.6), the value of  $a$  to be used is determined by the desired values of  $Q(k)$ . After these encoding steps, it can be seen that  $v$  channel digits are generated for each  $\lambda$  binary source digits, and thus the rate of the code is  $R = (\lambda/v) \ln 2$  nats per channel symbol.

For any given choice of  $\lambda$ ,  $v$ ,  $a$ , and mapping from the  $a$ -digit binary sequences into channel symbols, we consider the ensemble of codes generated by choosing all the elements  $g_{j,i}(l)$  and all the elements of  $v$  as independent equally probable binary digits.

**LEMMA 6.9.1.** In the ensemble of codes considered above, the code sequence  $x_1^{(1)}, \dots, x_1^{(v)}, x_2^{(1)}, \dots$ , corresponding to any given message sequence, is a random sequence with the digits statistically independent of each other and each chosen with the probability assignment  $Q(k)$ . Furthermore, if two message sequences  $u$  and  $u'$  are the same in the first  $b - 1$  subblocks and differ in the  $b$ th subblock, then the associated code sequences are the same in the first  $b - 1$  subblocks and statistically independent in all subsequent subblocks.

*Proof.* For any given output  $s$  from the binary convolutional encoder, the fact that  $v$  is comprised of independent equally probable binary digits insures that  $s \oplus v$  is composed of independent equiprobable binary digits, and thus that any code sequence contains independent letters with the probability assignment  $Q(k)$ . Now let  $u$  and  $u'$  differ for the first time in

subblock  $b$  and let  $\mathbf{s}$  and  $\mathbf{s}'$  be the associated outputs from the binary convolutional encoder. Then  $\mathbf{s}'' = \mathbf{s} \oplus \mathbf{s}'$  is the output from the convolutional encoder for an input  $\mathbf{u}'' = \mathbf{u} \oplus \mathbf{u}'$ , and since  $\mathbf{u}''$  contains all zeros in the first  $b - 1$  blocks,  $\mathbf{s}''$  does also. Now, for at least one value of  $j$ , say  $j'$ ,  $u_b''^{(j')} = 1$ , and for any  $n \geq b$ ,

$$s_n''^{(i)} = g_{j',i}(n-b) \oplus \sum_{j \neq j'} g_{j,i}(n-b) u_b''^{(j)} \oplus \sum_{l=b+1}^n \sum_{j=1}^{\lambda} g_{j,i}(n-l) u_l''^{(j)}$$

Since  $g_{j',i}(n-b)$  is an equiprobable binary random variable, independent of all the other  $g_{j,i}(l)$ ,  $s_n''^{(i)}$  is also an equiprobable binary random variable. Also  $s_n''^{(i)}$  is statistically independent of all  $s_m''^{(i)}$  for  $m < n$  and of  $s_n''^{(i')}$  for  $i' \neq i$  since these are independent of  $g_{j',i}(n-b)$ . Thus  $\mathbf{s}''$  contains independent equiprobable binary digits in all subblocks except the first  $b - 1$ . It follows that  $\mathbf{s} \oplus \mathbf{v}$  and  $\mathbf{s}' \oplus \mathbf{v}$  are statistically independent beyond the first  $b - 1$  subblocks, and thus that  $\mathbf{x}$  and  $\mathbf{x}'$  are statistically independent beyond the first  $b - 1$  subblocks. |

We now find an upper bound to  $\bar{W}_0$ , the average number of  $F$  hypotheses performed by a sequential decoder on the origin node and the tree of incorrect paths stemming from the origin node. The average will be over the ensemble of codes, the channel noise, and the message sequence. Then, for each  $\rho \geq 1$ , we find an upper bound to  $\bar{W}_0^\rho$ , the  $\rho$ th moment of  $W_0$ . We shall see that the same bounds apply to  $W_n$  for each node on the correct path.

The random variable  $W_0$  depends both on the values of the nodes on the correct path and on the values of the nodes in the incorrect tree stemming from the origin node. There are  $2^\lambda - 1$  immediate descendants of the origin node in the incorrect tree (the other immediate descendant being on the correct path). There are  $2^\lambda(2^\lambda - 1)$  nodes at depth two in the incorrect tree, and in general  $2^{\lambda(l-1)}(2^\lambda - 1)$  nodes at depth  $l$ . Let  $m(l)$  be the  $m$ th of these nodes of depth  $l$  according to some arbitrary ordering, and let  $\Gamma'_{m(l)}$  be the value of that node. Let  $\Gamma_n$  be the value of the  $n$ th node on the correct path and let  $\Gamma_{\min}$  be the infimum of  $\Gamma_n$  over  $0 \leq n < \infty$ .

**LEMMA 6.9.2.** The node  $m(l)$  can be hypothesized for the  $i$ th time  $i \geq 1$ , only if  $\Gamma'_{m(l)} \geq \Gamma_{\min} - 2\Delta + i\Delta$ .

*Proof.* From Theorem 6.9.1, part (b), the final threshold when node  $m(l)$  is  $F$  hypothesized for the first time is less than or equal to  $\Gamma'_{m(l)}$ . Likewise, the final threshold on the  $i$ th  $F$  hypothesis of  $m(l)$  is at most  $\Gamma'_{m(l)} - i\Delta + \Delta$ . The lemma will follow if we can show that the threshold can never be lowered below  $\Gamma_{\min} - \Delta$ . From part (b) of Theorem 6.9.1, the threshold is lowered from 0 in increments of  $\Delta$  on successive  $F$  hypotheses of the origin node. From part (c) of the theorem, the threshold cannot be lowered on any other

node below the final threshold on the most recent  $F$  hypothesis of the origin node. Thus the threshold is first set at  $\Gamma_{\min}$  or below on an  $F$  hypothesis of the origin node and this threshold is greater than  $\Gamma_{\min} - \Delta$ . Since the correct path stays on or above this threshold, the origin node is never subsequently rehypothesized and the threshold is never lowered below  $\Gamma_{\min} - \Delta$ . |

Now define the binary random variable  $w[m(l),i]$  by

$$w[m(l),i] = \begin{cases} 1 & \text{if } \Gamma'_{m(l)} \geq \Gamma_{\min} - 2\Delta + i\Delta \\ 0 & \text{otherwise} \end{cases} \quad (6.9.11)$$

Since node  $m(l)$  can be  $F$  hypothesized for the  $i$ th time only if  $w[m(l),i] = 1$ , we see that the number of  $F$  hypotheses on node  $m(l)$  is upper bounded by

$$\sum_{i=1}^{\infty} w[m(l),i]$$

Thus the total number of  $F$  hypotheses on the origin node and on all nodes in the incorrect tree stemming from the origin node is bounded by

$$W_0 \leq \sum_{l=0}^{\infty} \sum_{m(l)}^{\infty} \sum_{i=1}^{\infty} w[m(l),i] \quad (6.9.12)$$

The sum over  $m(l)$  for  $l = 0$  is by convention over just one node, the origin node. For  $l \geq 1$ , the sum over  $m(l)$  is over the  $(2^l - 1)2^{\lambda(l-1)}$  nodes in the incorrect tree at depth  $l$ . We now wish to upper bound the expected value of  $W_0$ . Since the expectation of a sum is equal to the sum of the expectations, we have

$$\overline{W}_0 \leq \sum_{l=0}^{\infty} \sum_{m(l)}^{\infty} \sum_{i=1}^{\infty} \overline{w[m(l),i]} \quad (6.9.13)$$

On the other hand, from (6.9.11), we have

$$\overline{w[m(l),i]} = \Pr[\Gamma'_{m(l)} \geq \Gamma_{\min} - 2\Delta + i\Delta] \quad (6.9.14)$$

Because of the statistical independence between the transmitted code sequence and the encoded sequence corresponding to  $m(l)$ , the problem of finding the probability on the right-hand side of (6.9.14) is very similar to that of finding the error probability for two randomly chosen code words. The only difference is that different lengths of sequences on the correct path must be considered. In view of this similarity to the two code-word problems, it is not surprising that the function  $E_o(\rho, Q)$ , evaluated at  $\rho = 1$ , appears in the answer,

$$E_o(1, Q) = -\ln \sum_{j=0}^{J-1} \left[ \sum_{k=0}^{K-1} Q(k) \sqrt{P(j \mid k)} \right]^2 \quad (6.9.15)$$

The following lemma is proved in Appendix 6B.

LEMMA 6.9.3. Consider an ensemble of codes in which the code sequence corresponding to each message sequence is a sequence of statistically independent digits occurring with the probability assignment  $Q(k)$ . Let  $\Gamma_{\min}$  be the infimum of the values along the correct path in the received value tree and let  $\Gamma'_l$  be the value of a node  $\mathbf{u}'_l$  where  $\mathbf{u}'_l$  is a message sequence of  $l$  subblocks and where the first  $l$  subblocks of the code sequence corresponding to  $\mathbf{u}'_l$  are statistically independent of the transmitted code sequence. Then, if the bias  $B$  satisfies  $B \leq E_o(1, \mathbf{Q})$ ,

$$\Pr[\Gamma'_l \geq \Gamma_{\min} + (i - 2)\Delta] \leq (l + 1) \exp \left[ -\frac{(i - 2)\Delta}{2} - \nu l \frac{E_o(1, \mathbf{Q}) + B}{2} \right] \quad (6.9.16)$$


---

Combining (6.9.13), (6.9.14), and (6.9.16), we obtain

$$\bar{W}_0 \leq \sum_{l=0}^{\infty} \sum_{m(l)}^{\infty} (l + 1) \exp \left[ -\frac{(i - 2)\Delta}{2} - \nu l \frac{E_o(1, \mathbf{Q}) + B}{2} \right] \quad (6.9.17)$$

From (6.9.1), we observe that  $2^{\lambda} = e^{\nu R}$ , and thus the sum over  $m(l)$  above involves fewer than  $e^{\nu l R}$  identical terms, yielding

$$\bar{W}_0 \leq \sum_{l=0}^{\infty} \sum_{i=1}^{\infty} (l + 1) \exp \left\{ -\frac{(i - 2)\Delta}{2} - \nu l \left[ \frac{E_o(1, \mathbf{Q}) + B}{2} - R \right] \right\} \quad (6.9.18)$$

These sums are easily carried out using the expansion

$$\frac{z}{1 - z} = \sum_{i=1}^{\infty} z^i; \quad z < 1 \quad (6.9.19)$$

and its derivative

$$\frac{1}{(1 - z)^2} = \sum_{l=0}^{\infty} (l + 1) z^l; \quad z < 1 \quad (6.9.20)$$

The sums converge if

$$R < \frac{E_o(1, \mathbf{Q}) + B}{2} \quad (6.9.21)$$

and we obtain

$$\bar{W}_0 \leq \frac{e^{\Delta/2}}{1 - e^{-\Delta/2}} \left\{ 1 - \exp \left[ -\nu \frac{E_o(1, \mathbf{Q}) + B}{2} + \nu R \right] \right\}^{-2} \quad (6.9.22)$$

The preceding bound gives us some insight into what values should be used for the bias  $B$  and the threshold spacing  $\Delta$ . The bound decreases with increasing  $B$ , but is only valid for  $B \leq E_o(1, \mathbf{Q})$ . Thus the bound is minimized by picking  $B = E_o(1, \mathbf{Q})$ . Also, minimizing over  $\Delta$ , we find a minimum at  $e^{\Delta/2} = 2$  or  $\Delta = 2 \ln 2$ . Using these values, we see that for  $R < E_o(1, \mathbf{Q})$ ,

$$\bar{W}_0 \leq 4 \{1 - \exp[-\nu E_o(1, \mathbf{Q}) + \nu R]\}^{-2} \quad (6.9.23)$$

We should be somewhat skeptical of these values of  $B$  and  $\Delta$  in minimizing  $\overline{W}_0$  since they only minimize a bound on  $\overline{W}_0$ . For the small values of  $i$  and  $l$  that dominate the sum in (6.9.13), the exponential-type bounds that we have used are quite crude. Experimentally, it has been found by Bluestein and Jordan (1963) that  $\overline{W}_0$  is typically a couple of orders of magnitude smaller than the bound here and is relatively insensitive to the bias and the threshold spacing. The important thing shown by (6.9.23) is not the explicit value of the bound but the fact that it is finite for  $R < E_o(1, \mathbf{Q})$ .

Now consider an arbitrary node on the correct path, say the  $n$ th node. We observe that the statistical description of the incorrect tree stemming from this node and the correct path beyond this node are precisely the same as the behavior from the origin node. The only difference is that the values of the nodes have  $\Gamma_n$  added into them. The lemma applies in the same way here as before and (6.9.23) yields an upper bound on  $\overline{W}_n$ . Thus (6.9.23) is an upper bound on the average number of  $F$  hypotheses per decoded subblock.

We can also see quite clearly from the bound why sequential decoding works. The number of nodes in an incorrect tree is growing exponentially with depth into the tree, but the probability of hypothesizing a node is an exponentially decreasing function of the depth into the tree. So long as the rate is less than  $E_o(1, \mathbf{Q})$ , the decrease in probability more than compensates for the increasing number of nodes. It is of course the treelike structure of convolutional codes that makes this trade-off possible.

We now proceed to find an upper bound on  $\overline{W}_0^\rho$  for  $\rho \geq 1$ . From (6.9.12), we have

$$\overline{W}_0^\rho \leq \left\{ \sum_{l=0}^{\infty} \sum_{m(l)}^{\infty} \sum_{i=1}^{\infty} w[m(l), i] \right\}^\rho \quad (6.9.24)$$

Using Minkowski's inequality (see Problem 4.15h) on the right-hand side of (6.9.24), we obtain

$$\left[ \overline{W}_0^\rho \right]^{1/\rho} \leq \sum_{l=0}^{\infty} \sum_{m(l)}^{\infty} \sum_{i=1}^{\infty} \left[ \overline{w[m(l), i]}^\rho \right]^{1/\rho} \quad (6.9.25)$$

Since  $w[m(l), i]$  takes on only the values 0 and 1,  $w[m(l), i]^\rho = w[m(l), i]$  and from (6.9.14) and (6.9.16), we have

$$\begin{aligned} \overline{w[m(l), i]}^\rho &\leq (l+1) \exp \left[ -\frac{(i-2)\Delta}{2} - \nu l \frac{E_o(1, \mathbf{Q}) + B}{2} \right] \\ \left[ \overline{W}_0^\rho \right]^{1/\rho} &\leq \sum_{l=0}^{\infty} \sum_{m(l)}^{\infty} \sum_{i=1}^{\infty} (l+1)^{1/\rho} \exp \left[ -\frac{(i-2)\Delta}{2\rho} - \nu l \frac{E_o(1, \mathbf{Q}) + B}{2\rho} \right] \end{aligned} \quad (6.9.26)$$

Upper bounding  $(I + 1)^{1/\rho}$  by  $I + 1$  and summing in the same way that we summed (6.9.17), we see that the sums converge if

$$R < \frac{E_o(1, \mathbf{Q}) + B}{2\rho} \quad (6.9.27)$$

and

$$\left[ \overline{W}_n^\rho \right]^{1/\rho} \leq \frac{\exp [\Delta/(2\rho)]}{1 - \exp [-\Delta/(2\rho)]} \left\{ 1 - \exp \left[ -\nu \frac{E_o(1, \mathbf{Q}) + B}{2\rho} + \nu R \right] \right\}^{-2} \quad (6.9.28)$$

Again the same bound is valid for  $\left[ \overline{W}_n^\rho \right]^{1/\rho}$ . If the bias  $B$  is equal to  $E_o(1, \mathbf{Q})$ , we see that  $\overline{W}_n^\rho$  is finite for  $R$  less than  $E_o(1, \mathbf{Q})/\rho$ . We can use (6.9.28) to bound the distribution function of  $W_n$ . Using the generalized Chebyshev inequality, we have

$$\Pr[W_n \geq i] \leq i^{-\rho} \overline{W}_n^\rho \quad (6.9.29)$$

We can summarize our results in the following theorem.

**Theorem 6.9.2.** Using sequential decoding on a discrete memoryless channel the average number of  $F$  hypothesis required per decoded subblock satisfies

$$\overline{W}_n \leq 4 \{ 1 - \exp [-\nu E_o(1, \mathbf{Q}) + \nu R] \}^{-2} \quad (6.9.30)$$

for  $R < E_o(1, \mathbf{Q})$

Here  $\nu$  is the number of channel digits per subblock,  $R = (\lambda/\nu) \ln 2$  is the code rate in nats per channel symbol,  $E_o(1, \mathbf{Q})$  is given by (6.9.15), the bias  $B$  has been chosen as  $E_o(1, \mathbf{Q})$  and the threshold spacing as  $2 \ln 2$ . The average is over the ensemble of infinite length codes defined after (6.9.10). Furthermore, for any  $\rho \geq 1$  for which  $R < E_o(1, \mathbf{Q})/\rho$ ,  $\overline{W}_n^\rho$  is also finite.

---

Savage (1966) has obtained a stronger bound on  $\overline{W}_n^\rho$  by considering a larger ensemble of codes. He considered an ensemble of randomly selected tree codes, where the encoded sequences have the tree structure of Figure 6.9.2 but where each entry in the tree is chosen independently with the probabilities  $Q(k)$ ,  $0 \leq k \leq K-1$ . For this ensemble, Savage shows that, for integer values of  $\rho$ ,  $\overline{W}_n^\rho$  is finite if  $R < E_o(\rho, \mathbf{Q})/\rho$ . It does not necessarily follow that convolutional codes of such rates exist for which  $\overline{W}_n^\rho$  is finite. It is conjectured, however, that for any positive  $\rho$  and any  $R < E_o(\rho, \mathbf{Q})/\rho$ ,  $\overline{W}_n^\rho$  is finite where the average is over the ensemble of infinite constraint-length convolutional codes of rate  $R$ . This conjecture has been proved by Falconer (1966) for the case  $0 < \rho \leq 1$ .

The question of what happens to  $\overline{W_n}^\rho$  for  $R > E_o(\rho, \mathbf{Q})/\rho$  has been considered in detail by Jacobs and Berlekamp (1967). Their results apply to arbitrary tree codes (that is, codes with a structure such as in Figure 6.9.2) and to a class of sequential decoding algorithms including the Fano algorithm described here. Let  $E_o(\rho) = \max_{\mathbf{Q}} E_o(\rho, \mathbf{Q})$  and let  $\hat{E}_o(\rho)$  be the convex hull of  $E_o(\rho)$ . That is,  $\hat{E}_o(\rho)$  is the smallest convex function greater than or equal to  $E_o(\rho)$ ; it is formed by replacing all the nonconvex portions of  $E_o(\rho)$  with straight-line segments. In our notation, Jacobs and Berlekamp show that, for any tree code of rate  $R > \hat{E}_o(\rho)/\rho$ ,

$$\lim_{N \rightarrow \infty} \overline{\left( \frac{1}{N} \sum_{n=0}^{N-1} W_n \right)}^\rho = \infty \quad (6.9.31)$$

The fact that higher-order moments of  $W_n$  are infinite indicate that long waiting lines will be a severe problem in sequential decoding. Jacobs and Berlekamp show that the probability per digit that either the waiting line will exceed a given length  $i$  or that an error will be made can decrease no faster with  $i$  than as  $i^{-\rho}$  for a code of rate  $\hat{E}_o(\rho)/\rho$ .

### Error Probability for Sequential Decoding

In order to obtain results on error probability for sequential decoding, we must consider codes of finite constraint length. For a code of infinite constraint length, if  $\overline{W_n}$  is finite for each  $n$ , then from (6.9.18) the probability of hypothesizing a node of depth  $l$  in a tree of incorrect paths goes to zero with increasing  $l$  and the error probability is clearly zero.

Assume that the encoder has a constraint length of  $L$  subblocks. Assume also that the encoder, instead of accepting an infinite sequence of subblocks from the source, accepts a finite number of subblocks  $L_T$ , where  $L_T$  is typically much larger than  $L$ . After the  $L_T$  subblocks of data have entered the encoder, a sequence of  $L - 1$  subblocks of known zeros enter the encoder. In the received value tree for such a situation, each node of depth  $L_T - 1$  or less will have  $2^L$  immediate descendants and each node of depth  $L_T$  or greater will have only one immediate descendant. Decoding proceeds according to the rules of Figure 6.9.4, except that the decoder recognizes that nodes of order  $L_T$  or greater have only one immediate descendant and thus rules 3 and 5 always yield backward moves on nodes of order greater than  $L_T$ . The decoding terminates with the first  $F$  hypothesis of a node of order  $L_T + L - 1$  and the hypothesized source sequence at that point is taken as the decoded sequence. For finite  $L$  and  $L_T$  the decoding must terminate eventually.

If we consider the ensemble of codes defined after (6.9.10), with  $L$  finite, it can be shown that the pair of code sequences corresponding to a given pair of source sequences are statistically independent over the first  $L$  subblocks starting with the first subblock in which the message sequences differ, but are not independent thereafter. The probability of decoding error for this ensemble can be upper bounded by a quantity essentially equivalent to the random-coding bound for block codes (see Problem 6.42). Here, however, it will be more interesting to modify the ensemble slightly and then derive a much smaller upper bound on error probability. In the modification, we shall make the binary convolutional encoder time varying, with the binary outputs generated by the rule [compare with (6.9.10)].

$$s_n^{(i)} = \sum_{l=n-L+1}^n \sum_{j=1}^{\lambda} g_{j,i}(n,l) u_l^{(j)} ; \quad 1 \leq i \leq ar \quad (6.9.32)$$

In the ensemble, each element  $g_{j,i}(n,l)$  is chosen as a statistically independent equiprobable binary random variable. The rest of the ensemble is as before, with a random binary sequence  $v$  added to  $s$  and then a transformation to channel input letters. In terms of Figure 6.9.3, this ensemble corresponds to randomly reshuffling the connections between the  $\lambda$  shift registers and the  $r$  modulo-2 adders generating the binary outputs, reshuffling after each subblock of message digits enters the encoder.

**LEMMA 6.8.4.** For the ensemble of codes above, the code sequence corresponding to any given message sequence is a sequence of statistically independent letters each chosen with the probability assignment  $Q(k)$ , where  $Q(k)$  is the relative frequency of input letter  $k$  in the mapping from binary  $a$  length sequences into channel input letters. Furthermore, if  $x$  and  $x'$  are the code sequences corresponding to the message sequences  $u$  and  $u'$  respectively, then  $x$  and  $x'$  are identical over each subblock for which  $u$  and  $u'$  are identical in that subblock and the previous  $L - 1$  subblocks. Over the set of all other subblocks,  $x$  and  $x'$  are statistically independent.

---

The proof is almost identical to that of Lemma 6.9.1 and thus will be omitted.

In terms of Figure 6.8.3, the lemma says that  $x$  and  $x'$  are identical over those subblocks for which the contents of the shift registers are identical and that  $x$  and  $x'$  are independent elsewhere.

For a given code in the above ensemble, let  $u$  be the transmitted message sequence and let  $u'$  be the decoded message sequence using sequential decoding. Typically, when decoding errors occur, they occur in bursts. More specifically, *a burst of decoding errors is defined as a consecutive sequence of one or more subblocks with the following properties; the first and last subblock*

of the sequence contain errors (that is,  $\mathbf{u}$  and  $\mathbf{u}'$  differ in those subblocks); there are no consecutive  $L - 1$  error-free subblocks within the sequence; and there are no errors for  $L - 1$  subblocks on either side of the sequence. This definition uniquely specifies a set of one or more bursts of decoding errors for any  $\mathbf{u}' \neq \mathbf{u}$ .

Define  $P_e(b,c)$  as the probability (over the ensemble of codes, messages, and channel noises) that a burst of decoding errors occurs beginning on the  $b$ th subblock and ending on the  $c$ th subblock. Define  $P_{e,n}$  as the probability that the  $n$ th subblock is decoded in error. By our definition of burst, any subblock decoded in error must be in some burst of decoding errors, and we have

$$P_{e,n} \leq \sum_{b=1}^n \sum_{c=n}^{L_T} P_e(b,c) \quad (6.9.33)$$

In order to obtain an upper bound on  $P_e(b,c)$ , we start with the simplest case—that case where  $b = 1$ . Let  $\mathbf{u}$  be the transmitted message sequence and let  $\mathbf{u}'_{c+L-1}$  be the first  $c + L - 1$  subblocks of the decoded sequence. For a burst of decoding errors to occur from subblock 1 to  $c$ , we must have  $u'_1 \neq u_1$ ,  $u'_c \neq u_c$ , and  $u'_{c+i} = u_{c+i}$  for  $1 \leq i \leq L - 1$ . Since each intervening subblock can be chosen in at most  $2^{\lambda c}$  ways, there are fewer than  $2^{\lambda c}$  choices for  $\mathbf{u}'_{c+L-1}$  which will yield a burst of decoding errors from subblock 1 to  $c$ .

Let  $\Gamma_l$ ,  $0 \leq l \leq L_T + L - 1$ , be the path values in the received value tree for the correct path, let

$$\Gamma_{\min} = \min_{0 \leq l \leq L_T + L - 1} \Gamma_l$$

and let  $\Gamma'_{c+L-1}$  be the value of the node  $\mathbf{u}'_{c+L-1}$  for one of the choices of  $\mathbf{u}'_{c+L-1}$  that, if decoded corresponds to a burst of decoding errors from subblock 1 to  $c$ . It is certainly necessary for an  $F$  hypothesis of  $\mathbf{u}'_{c+L-1}$  to occur in order for  $\mathbf{u}'_{c+L-1}$  to be decoded. On the other hand, from Lemma 6.9.2, an  $F$  hypothesis of  $\mathbf{u}'_{c+L-1}$  can occur only if  $\Gamma'_{c+L-1} \geq \Gamma_{\min} - \Delta$ . Thus

$$\Pr[\text{decoding } \mathbf{u}'_{c+L-1}] \leq \Pr[\Gamma'_{c+L-1} \geq \Gamma_{\min} - \Delta] \quad (6.9.34)$$

From Lemma 6.9.4, the  $c + L - 1$  subblocks of the code sequence corresponding to  $\mathbf{u}'_{c+L-1}$  are statistically independent of the transmitted sequence, and thus from Lemma 6.9.3, we have

$$\Pr[\text{decoding } \mathbf{u}'_{c+L-1}] \leq (c + L) \exp \left[ \frac{\Delta}{2} - v(c + L - 1) \frac{E_o(1, \mathbf{Q}) + B}{2} \right] \quad (6.9.35)$$

for  $B \leq E_o(1, \mathbf{Q})$ . Since there are fewer than  $2^{\lambda c}$  choices of  $\mathbf{u}'_{c+L-1}$  that correspond to a burst from subblock 1 to  $c$ , we have

$$P_e(1, c) \leq 2^{\lambda c}(c + L) \exp \left[ \frac{\Delta}{2} - r(c + L - 1) \frac{E_o(1, \mathbf{Q}) + B}{2} \right] \quad (6.9.36)$$

Next we find an equivalent bound for  $P_e(b, c)$  for arbitrary  $b \geq 1$ . Consider a particular code in the ensemble, letting  $\mathbf{u}$  be the transmitted message, and  $\mathbf{x}$  be the transmitted code sequence. Let  $\mathbf{u}'_{b-1}$  be the first  $b - 1$  subblocks of an arbitrary message sequence satisfying  $u'_{b-i} = u_{b-i}$  for  $1 \leq i \leq \min(b - 1, L - 1)$ . Consider the descendants of  $\mathbf{u}'_{b-1}$  in the received value tree. One path of descendants is that corresponding to the transmitted message subblocks  $u_b, u_{b+1}, \dots$ . The code sequence in subblocks  $b, b + 1, \dots$ , corresponding to  $\mathbf{u}'_{b-1}, u_b, u_{b+1}, \dots$ , is the same as that corresponding to the transmitted message  $\mathbf{u}$  since the contents of the shift registers generating the code sequences agree over these subblocks. Thus the change in path value from node  $\mathbf{u}'_{b-1}$  to the descendant  $\mathbf{u}'_{b-1}, u_b, \dots, u_i$  is the same as the change in path value from  $\mathbf{u}_{b-1}$  to  $\mathbf{u}_i$ , denoted  $\Gamma_i - \Gamma_{b-1}$ .

Let  $\mathbf{u}'_{c+L-1}$  be any descendant of  $\mathbf{u}'_{b-1}$  with a length of  $c + L - 1$  subblocks and let  $\Gamma'_{c+L-1} - \Gamma'_{b-1}$  be the change in value in the received value tree from node  $\mathbf{u}'_{b-1}$  to  $\mathbf{u}'_{c+L-1}$ . In order for  $\mathbf{u}'_{c+L-1}$  to be decoded, it must at least be  $F$  hypothesized, and in order for it to be  $F$  hypothesized, it is necessary that

$$\Gamma'_{c+L-1} - \Gamma'_{b-1} \geq \min_{b-1 \leq i \leq L_T + L - 1} (\Gamma_i - \Gamma_{b-1}) - \Delta \quad (6.9.37)$$

This follows since the sequence  $\mathbf{u}'_{b-1}, u_b, u_{b+1}, \dots$  otherwise prevents the threshold from being lowered far enough to  $F$  hypothesize  $\mathbf{u}'_{c+L-1}$ . If the decoding of  $\mathbf{u}'_{c+L-1}$  leads to a burst of decoding errors from  $b$  to  $c$ , then it is necessary that  $u'_c \neq u_c$  and  $u'_{c+i} = u_{c+i}$  for  $1 \leq i \leq L - 1$ . Thus there are fewer than  $2^{\lambda(c-b+1)}$  choices of the subsequences  $u'_b, \dots, u'_{c+L-1}$  which, if decoded, lead to a burst of decoding errors from  $b$  to  $c$ . Finally, whether or not (6.9.37) is satisfied for any of these choices depends only on the code sequences from subblock  $b$  on and these are independent of the choice of  $\mathbf{u}'_{b-1}$  [aside from the restriction that  $u'_{b-i} = u_{b-i}$  for  $1 \leq i \leq \min(b - 1, L - 1)$ ]. Thus the probability of a burst of decoding errors from  $b$  to  $c$  is upper bounded by the probability that (6.9.37) is unsatisfied for all of the above fewer than  $2^{\lambda(c-b+1)}$  choices of  $u'_b, \dots, u'_{c+L-1}$ . Since this probability is independent of  $\mathbf{u}'_{b-1}$ , we can choose  $\mathbf{u}'_{b-1} = \mathbf{u}_{b-1}$  for convenience.

Over the ensemble of codes, the code sequence in subblocks  $b$  to  $c + L - 1$  corresponding to any of the above  $\mathbf{u}'_{c+L-1}$  is statistically independent of the code sequence in subblocks  $b$  to  $L_T + L - 1$  corresponding to  $\mathbf{u}$  (see Lemma

6.9.4). Thus, from Lemma 6.9.3, we have, for  $B \leq E_o(1, \mathbf{Q})$ ,

$$\begin{aligned} \Pr\{[\Gamma'_{c+L-1} - \Gamma'_{b-1}] &\leq \min_{b-1 \leq l \leq L_T-L-1} [\Gamma_l - \Gamma_{l-1}] - \Delta\} \\ &\leq (c + L - b + 1) \exp\left[\frac{\Delta}{2} - \nu(c + L - b) \frac{E_o(1, \mathbf{Q}) + B}{2}\right] \end{aligned} \quad (6.9.38)$$

$$P_e(b, c) \leq 2^{\lambda(c-b+1)} (c + L - b + 1) \exp\left[\frac{\Delta}{2} - \nu(c + L - b) \frac{E_o(1, \mathbf{Q}) + B}{2}\right] \quad (6.9.39)$$

Substituting (6.9.39) into (6.9.33) and summing over  $b$  and  $c$ , we get a bound on the probability of error per subblock. We can further upper bound by summing  $b$  from  $-\infty$  to  $n$  and  $c$  from  $n$  to  $\infty$ , and these sums can be evaluated from the geometric series and its derivative in (6.9.19) and (6.9.20). The result, for  $z < 1$  [see (6.9.42)], is

$$P_{e,n} \leq A \exp\left[-\nu L \frac{E_o(1, \mathbf{Q}) + B}{2}\right] \quad (6.9.40)$$

$$A = 2^\lambda e^{\Delta/2} \left[ \frac{L}{(1-z)^2} + \frac{1+z}{(1-z)^3} \right] \quad (6.9.41)$$

$$z = 2^\lambda \exp\left[-\nu \frac{E_o(1, \mathbf{Q}) + B}{2}\right] \quad (6.9.42)$$

We observe that the bound in (6.9.40) is minimized over choices of  $B \leq E_o(1, \mathbf{Q})$  by choosing  $B = E_o(1, \mathbf{Q})$ . We also define  $R = (\lambda/\nu) \ln 2$  as the source rate in natural units per channel letter for the first  $L_T$  subblocks. The actual rate is slightly lower by a factor of  $L_T/(L_T + L - 1)$  due to the terminating sequence for the overall block. Substituting the definition of  $R$  and the choice of bias  $B = E_o(1, \mathbf{Q})$  into (6.9.40) to (6.9.42), we obtain

$$P_{e,n} \leq A \exp[-\nu L E_o(1, \mathbf{Q})] \quad (6.9.43)$$

$$A = \left[ \frac{L}{(1-z)^2} + \frac{1+z}{(1-z)^3} \right] \exp\left[\nu R + \frac{\Delta}{2}\right] \quad (6.9.44)$$

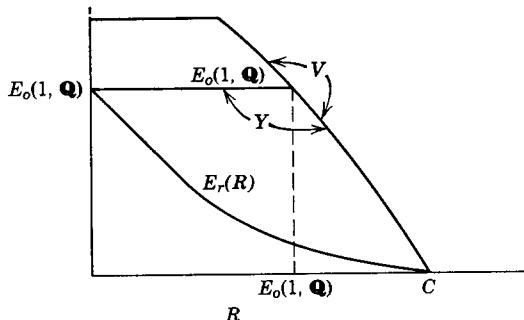
$$z = \exp\{-\nu [E_o(1, \mathbf{Q}) - R]\} \quad (6.9.45)$$

The bound is valid for  $R < E_o(1, \mathbf{Q})$ , which is the same restriction that we had for  $\overline{W}_n$  to be finite.

It is observed that the probability of decoding error per subblock in (6.9.43) decays exponentially with the encoding constraint length  $\nu L$  with an exponent  $E_o(1, \mathbf{Q})$ . The fact that this exponent is independent of the rate for  $R < E_o(1, \mathbf{Q})$  is not necessarily caused by the weakness of the bound.

From (6.9.39) this is the exponent for the probability of a burst of decoding errors of length 1. Since only  $2^{\lambda} - 1$  sequences differ from  $\mathbf{u}$  in only a given subblock, this exponent is independent of  $R$ .

In Figure 6.9.7 the exponent  $E_o(1, \mathbf{Q})$  is compared with the random-coding exponent  $E_r(R)$  for block codes. It is surprising that the exponent for sequential decoding is larger than  $E_r(R)$ . This is explained partly by observing that the constraints on a block of data in a convolutional code extend beyond the given block and partly by observing that the decoder is allowed to search the whole block of  $(L + L_T - 1)v$  received digits before making any decisions.



*Figure 6.9.7. Comparison of sequential decoding exponent,  $E_o(1, \mathbf{Q})$  with block random-coding exponent; curve  $Y$  is exponent in Yudkin's upper bound to  $P_e$  for sequential decoding; curve  $V$  is exponent in Viterbi's lower bound to  $P_e$  for convolutional codes.*

Yudkin (1964), in a more elaborate analysis than that leading to (6.9.43), has obtained an upper bound to error probability for sequential decoding for all rates up to capacity and his exponent is also sketched in Figure 6.9.7. Also Viterbi (1967) has recently found a lower bound to the minimum probability of error that can be achieved with convolutional codes and his exponent is also sketched in Figure 6.9.7. It is observed that sequential decoding has an error probability with the best possible exponent for  $R \geq E_o(1, \mathbf{Q})$ , and essentially the best possible exponent for  $R$  slightly less than  $E_o(1, \mathbf{Q})$ , where  $\overline{W}_n$  is also finite.

The bound on error probability in (6.9.40) is independent of the overall length  $L_T$ , and thus it is not necessary to periodically insert a terminating sequence with sequential decoding. On the other hand, in practical applications, particularly if  $L$  is more than 20 or so, it is desirable to keep  $L_T$  finite so that when one of the infrequent very long searches arises, it is possible for

the decoder to count a received block as erased and pass on to the next block. The probability of such erasures, of course, is intimately related to the distribution function of  $\overline{W}_n$ .

### **6.10 Coding for Burst Noise Channels**

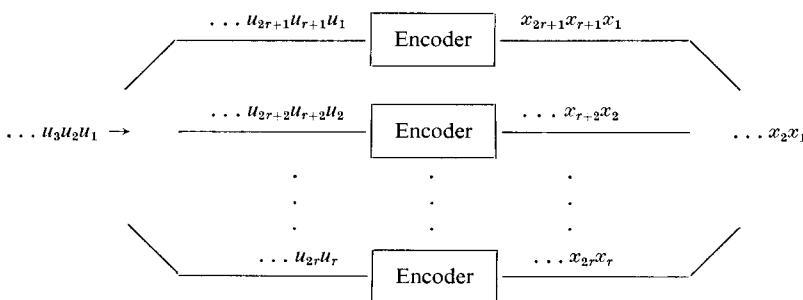
In the previous sections we have been concerned primarily with coding techniques for memoryless channels. In this section we shall be concerned with channels having binary input and output alphabets where the transmission errors tend to cluster together into bursts. Most binary communication systems (with the exception of space channels) exhibit this behavior to some extent. It is difficult to find probabilistic models for these channels that are appropriate for the study of coding. It is not enough to find models that describe the typical behavior of the channel, since it is the atypical behavior that causes decoding errors for any reasonable coding technique. The atypical behavior is caused by a variety of rare events which by their nature resist probabilistic modeling. For this reason we shall not be concerned with the error probability for various coding techniques on these channels, but will find other measures of performance.

The most obvious coding technique to use on channels with burst noise is that of error detection and retransmission. In this technique the source data are encoded by a parity check code (and preferably a cyclic code, for ease of implementation). The syndrome sequence  $S$  is calculated at the receiver and if  $S = 0$ , the information digits are accepted as correct. If  $S \neq 0$ , the receiver instructs the transmitter to retransmit the given code word. A decoding error occurs here only if transmission errors occur and the error sequence is the same as one of the code words. For an  $(N,L)$  parity check code only  $2^L$  of the  $2^N$  sequences of length  $N$  are code words, or only 1 of each  $2^{N-L}$  sequences. It can be seen from this that decoding errors can be made negligible with modest values of  $N - L$  and that the number of decoding errors is quite insensitive to the detailed statistics of the noise. For a cyclic code both the encoder and decoder in this scheme can be instrumented with  $N - L$  stage shift registers (see Figure 6.5.5).

Despite the simplicity of this technique, it has several drawbacks. The first is the need for a reliable feedback link from receiver to transmitter to transmit the retransmission requests. The second is due to a variety of buffering problems introduced at transmitter and receiver by the occasional retransmissions. The third is that if the channel is normally quite noisy, very few code words will be accepted. The seriousness of the first two drawbacks depends upon the existence of a feedback link and the nature of the source (that is, whether it produces data on call from the transmitter or whether it produces data at a constant rate in time). If the third drawback is serious, then the retransmission scheme above can be combined with some of the error

correction techniques to be discussed subsequently. It should be pointed out here that if the digital data modulator and demodulator have been efficiently designed to transmit a large number of binary digits per second, then frequent transmission errors will occur and error correction will certainly be necessary whether or not retransmission is used.

Another simple technique for achieving reliable transmission on a burst noise channel is that of interlacing or scrambling. From a conceptual standpoint, the incoming stream of binary data is separated into a fixed number, say  $r$ , of data streams as shown in Figure 6.10.1. Each of the  $r$  data streams is



*Figure 6.10.1. Interlaced encoding.*

then separately encoded and the encoded sequences are commutated together for transmission on the channel. At the channel output, the received data stream is again separated into  $r$  streams, each stream is separately decoded, and the decoded data is finally commutated together again.

The idea behind this technique is that successive letters within any code word will be separated on the channel by  $r$  channel time units. Thus if the channel memory dies away with increasing time separation, the channel noise affecting successive letters in a code word will be essentially independent for sufficiently large  $r$ . Consequently, any of the coding techniques previously discussed for memoryless channels can be used on a burst noise channel in conjunction with interlacing.

It can be seen from the above argument that, in a sense, having memory in a channel does not decrease its capacity. To make this statement more explicit, suppose that a single letter transition probability assignment  $P(y | x)$  can be defined for a discrete channel with memory.\* Then if the channel memory dies out quickly enough in time, any coding technique that yields a given error probability for the discrete memoryless channel with transition

\* As discussed in Section 4.6, this is not always possible, and in particular it is not possible for channels with intersymbol interference.

probabilities  $P(y | x)$  will yield essentially that same error probability for the given channel with memory using interlacing with large enough  $r$ . Thus the channel with memory has a capacity at least as large as that of the associated memoryless channel. We shall not state the above result as a theorem because of the difficulty in being precise about how quickly the channel memory must die away with time.

For the implementation of interlacing, it is not always necessary to construct  $r$  separate encoders and decoders. For example, if the encoders in Figure 6.10.1 are  $(N, L)$  cyclic encoders, each with the generator polynomial  $g(D)$ , then the interlacing and encoding in Figure 6.10.1 can be replaced by an  $(Nr, Lr)$  cyclic encoder with the generator polynomial  $g(D^r)$ . Likewise if  $r$  is odd and the encoders are identical convolutional encoders, each of rate  $\frac{1}{2}$  (in bits) as in Figure 6.8.1, then the interlacing and encoding can be replaced with a single convolutional encoder placing  $r - 1$  shift register stages between each shift register stage in the diagram of Figure 6.8.1, and by passing the check bits through an  $(r - 1)/2$  stage shift register.

The major advantage of interlacing, from a practical standpoint, is that it is quite insensitive to the detailed statistics of the channel memory, relying only on  $r$  being large enough to eliminate most of the effects of the memory. The disadvantage of interlacing (or at least of an interleaved decoder) is that the memory is ignored in making decoding decisions.

Before proceeding further we must set up a criterion for evaluating coding techniques for burst noise channels. For simplicity we shall assume throughout that the channel input  $x$  and output  $y$  are binary sequences. The error sequence  $z = \dots, z_{-1}, z_0, z_1, \dots$  is given by  $x \oplus y$ . In attempting to define what a burst is, we observe that two errors (i.e., 1's) in the sequence  $z$  separated by a number of 0's could be interpreted either as two isolated errors or as a burst containing two errors. To resolve this type of ambiguity we define a set of consecutive noise digits  $z_n, z_{n+1}, \dots, z_{n+b-1}$  to be a burst of errors relative to a guard space  $g$  if, first,  $z_n = z_{n+b-1} = 1$ , and, second, if the  $g$  consecutive digits on each side of the set  $z_n, \dots, z_{n+b-1}$  are all 0, and third, if there is no consecutive sequence of  $g$  0's within the set  $z_n, \dots, z_{n+b-1}$ . The length of the burst,  $b$ , is taken as the size of the set. Observe that all the noise digits in a burst need not be errors. In the special case  $b = 1$ , the burst is just an isolated error with at least  $g$  error free digits on each side. The above definition uniquely breaks up any  $z = \dots, z_{-1}, z_0, z_1, \dots$  into a set of bursts, each separated by at least  $g$  error free digits.

An encoder and decoder is said to have burst correcting capability  $b$  relative to a guard space  $g$  if  $b$  is the largest integer for which every noise sequence  $z$  containing only bursts of length  $b$  or less relative to the guard space  $g$  is correctly decoded. The burst correcting capability of an encoder (or code) relative to a guard space  $g$  is similarly defined as the largest integer

$b$  such that for some decoder, the encoder and decoder have burst correcting capability  $b$  relative to the guard space  $g$ . We shall use the burst capability of a code relative to a given guard space as a criterion of the effectiveness of the code against burst noise. It should be clear that this is only a crude criterion. For example, on a channel where long bursts containing relatively few errors are far more likely than shorter bursts containing many errors, one would prefer a code capable of correcting the likely longer bursts at the expense of the less likely shorter bursts.

We now develop an upper bound on the burst correcting capability of a code in terms of its rate  $R$  and its guard space  $g$ . The bound is valid for block codes, convolutional codes, and any other class of codes. We do assume,

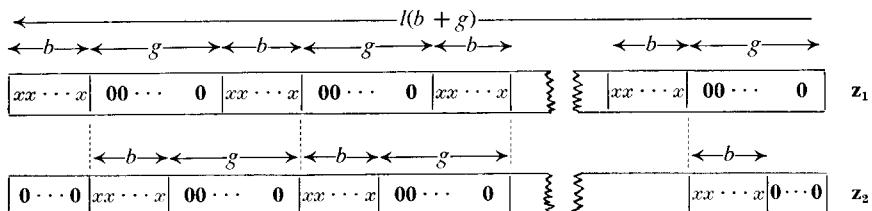


Figure 6.10.2. Two types of noise sequences;  $x$ 's represent arbitrary binary digits.

however, that there is an arbitrary but finite decoding delay of, say,  $N$  source digits. That is, by the time the  $n$ th source digit enters the encoder, ( $n > N$ ), at least  $n - N$  source digits must be decoded. Recalling that the rate  $R$  (in binary units) is defined as the number of source digits per channel digit, this condition can be translated into requiring that by the time  $L$  channel digits have been received, at least  $RL - N$  source digits must be decoded. Now suppose we are using a code that has burst correcting capability  $b$  relative to a guard space  $g$ , and consider a number of received digits  $L$  that is a multiple of  $b + g$ ,  $L = l(b + g)$ .

Next consider the two types of error sequences shown in Figure 6.10.2. In each type, the error sequence is constrained to have zero values in the positions shown and may have arbitrary values in the positions marked  $x$  (we assume here that  $b \leq g$ ). Let  $\mathbf{x}_1$  and  $\mathbf{x}_2$  be encoded sequences corresponding to different choices of the first  $[RL - N]$  source digits and  $\mathbf{z}_1$  and  $\mathbf{z}_2$  be error sequences of the first and second type respectively. Since by assumption these error sequences cannot cause decoding errors, we have

$$\mathbf{x}_1 \oplus \mathbf{z}_1 \neq \mathbf{x}_2 \oplus \mathbf{z}_2 \quad (6.10.1)$$

More generally, if  $\mathbf{z}_1$  and  $\mathbf{z}_1'$  are error sequences of the first type and  $\mathbf{z}_2$  and  $\mathbf{z}_2'$  are error sequences of the second type, we must have

$$\mathbf{x}_1 \oplus \mathbf{z}_1 \oplus \mathbf{z}_2 \neq \mathbf{x}_2 \oplus \mathbf{z}_1' \oplus \mathbf{z}_2' \quad (6.10.2)$$

To establish (6.10.2), we assume that (6.10.2) is false for some choice of sequences and establish a contradiction. If equality holds true in (6.10.2), then we can add  $\mathbf{z}_1'$  and  $\mathbf{z}_2$  to both sides of the equation, resulting in

$$\mathbf{x}_1 \oplus \mathbf{z}_1 \oplus \mathbf{z}_1' = \mathbf{x}_2 \oplus \mathbf{z}_2 \oplus \mathbf{z}_2' \quad (6.10.3)$$

Since  $\mathbf{z}_1 \oplus \mathbf{z}_1'$  is an error sequence of the first type and  $\mathbf{z}_2 \oplus \mathbf{z}_2'$  is an error sequence of the second type, (6.10.3) contradicts (6.10.1) and thus (6.10.2) is valid. Finally we observe that if  $\mathbf{x}_1$  and  $\mathbf{x}_2$  are equal and correspond to the same first  $[RL - N]$  source digits but if either  $\mathbf{z}_1 \neq \mathbf{z}_1'$  or  $\mathbf{z}_2 \neq \mathbf{z}_2'$ , then again inequality holds true in (6.10.2).

It can be seen now that we can consider at least  $2^{RL-N} = 2^{Rl(b+g)-N}$  different ways of choosing  $\mathbf{x}_1$ , each corresponding to a different choice of the first  $[RL - N]$  source digits. Similarly we can consider  $2^{lb}$  different ways of choosing  $\mathbf{z}_1$  and  $2^{lb}$  different ways of choosing  $\mathbf{z}_2$ . From (6.10.2), each different choice of the triple  $(\mathbf{x}_1, \mathbf{z}_1, \mathbf{z}_2)$  leads to a different sequence  $\mathbf{x}_1 \oplus \mathbf{z}_1 \oplus \mathbf{z}_2$ . Since there are  $2^{l(b+g)}$  different binary sequences of length  $l(b + g)$ , we have the inequality

$$2^{Rl(b+g)-N} \cdot 2^{lb} \cdot 2^{lb} \leq 2^{l(b+g)} \quad (6.10.4)$$

$$R(b + g) - \frac{N}{l} + 2b \leq b + g \quad (6.10.5)$$

Since  $N$  is fixed but (6.10.5) must be satisfied for all  $l \geq 1$ , we can pass to the limit  $l \rightarrow \infty$ , obtaining

$$R(b + g) + 2b \leq b + g \quad (6.10.6)$$

For  $b \geq g$ , the same analysis can be applied, letting the second type of noise sequence in Figure 6.10.2 have bursts of length  $g$ . This leads to the result  $R = 0$  (see Problem 6.46).

Rearranging (6.10.6) into (6.10.7) below, we have proved the following theorem.

**Theorem 6.10.1.** In order for a binary encoder and decoder of bounded decoding delay and of rate  $R > 0$  (in binary digits per channel digit) to have a burst correcting capability  $b$  relative to a guard space  $g$ , it is necessary to have

$$\frac{g}{b} \geq \frac{1 + R}{1 - R} \quad (6.10.7)$$

In what follows, we consider first cyclic burst error correcting codes and then convolutional burst error correcting codes. We shall see that (at least for the convolutional codes) the bound in (6.10.7) can be approached arbitrarily closely in the limit of large  $g$ . We shall also see that most bursts of lengths much greater than the bound on  $b$  in (6.10.7) can be corrected for large  $g$ .

### Cyclic Codes

Consider using a binary  $(N, L)$  cyclic code with the  $N - L$  degree generator polynomial  $g(D)$  for correcting bursts of errors. Let  $x(D)$  be the transmitted code word polynomial,  $z(D)$  be the error polynomial, and  $y(D) = x(D) + z(D)$  be the polynomial corresponding to the received sequence.

In terms of a cyclic code, it is convenient to define a burst of errors in a slightly different way than before. For an error sequence  $\mathbf{z} = (z_{N-1}, \dots, z_0)$ , we find the longest cyclically consecutive run of 0's and consider the rest of the sequence to be a burst. If a burst does not include both digits  $z_{N-1}$  and  $z_0$ , it is called an ordinary burst. For example, if  $\mathbf{z} = (\mathbf{1}, \mathbf{0}, \mathbf{0}, \dots, \mathbf{0}, \mathbf{0}, \mathbf{1})$  then by this definition,  $\mathbf{z}$  contains a burst of length 2, consisting of  $z_{N-1}$  and  $z_0$ , but it is not an ordinary burst. In order to see part of the reason for this rather strange definition, suppose a cyclic code is capable of correcting all bursts of length  $b$  or less. Then in terms of our old definitions, the code has at least a burst correcting capability  $b$  relative to a guard space  $N - b$ . If the non-ordinary bursts were not corrected, however, two bursts on the channel separated by  $N - b$  error free digits in the middle of a block would not be corrected.

An optimum burst decoder for a cyclic code is defined as a decoder which, given  $y(D)$ , selects the code word  $x(D)$  for which  $y(D) - x(D)$  contains the shortest burst. Such a decoder would minimize the probability of decoding error on a channel for which each burst of any given length is less likely than each burst of any shorter length. As we now show, it turns out to be surprisingly easy to construct optimum burst decoders.

For each  $i$ ,  $0 \leq i \leq N - 1$ , define the  $i$ th cyclic shift of the received sequence polynomial  $y(D)$  as

$$y^{(i)}(D) = R_{D^{N-i}}[D^{N-i}y(D)] \quad (6.10.9)$$

Also define  $B_i(D)$  as

$$B_i(D) = R_{g(D)}[y^{(i)}(D)] \quad (6.10.10)$$

Since  $y^{(i)}(D) - B_i(D)$  is divisible by  $g(D)$ , it is a code word. Since each cyclic shift of a code word is also a code word, we can shift  $y^{(i)}(D)$  back to  $y(D)$  and shift  $B_i(D)$   $i$  places in the same direction to see that  $y(D) - R_{D^{N-i}}[D^i B_i(D)]$  is a code word. In other words, given  $y(D)$ ,  $R_{D^{N-i}}[D^i B_i(D)]$ , for each  $i$ , is a possible error sequence that might have occurred to yield  $y(D)$ . From (6.10.10), each  $B_i(D)$  has a degree of at most  $N - L - 1$  and thus  $R_{D^{N-i}}[D^i B_i(D)]$  corresponds to a sequence with its nonzero digits localized to a cyclically consecutive string of at most  $N - L$  digits.

Next suppose that  $x(D)$  is transmitted and a burst of  $b \leq N - L$  errors occurs from  $z_i$  to  $z_{i+b-1}$  (or to  $z_{i+b-1-N}$  for a nonordinary burst). We can represent the burst by the polynomial  $R_{D^{N-i}}[D^i \beta(D)]$  where  $\beta(D)$  has degree

$b - 1$ , and  $y(D) = x(D) + R_{D^{N-1}}[D^i \beta(D)]$ . We then have  $y^{(i)}(D) = x^{(i)}(D) + \beta(D)$  where  $x^{(i)}(D)$  is a cyclic shift of  $x(D)$ . It follows that  $\beta(D) = B_i(D)$  as defined in (6.10.10).

It follows from the above argument that an optimum decoder can operate by simply finding  $B_i(D)$  for each  $i$ , then choosing the  $i$  for which  $B_i(D)$  corresponds to the shortest burst, and then adding  $R_{D^{N-1}}[D^i B_i(D)]$  to  $y(D)$  for that  $i$ . A circuit that calculates  $B_i(D)$  for each  $i$  is shown in Figure 6.10.3. It can be seen that this is simply a circuit for dividing  $y(D)$  by  $g(D)$  and the contents of the register after  $y_0$  enters the left hand side of the shift register is just  $B_0(D)$  (with high order terms at the right). However, if the register is shifted again after this point, with no more input from the received sequence, the contents of the register will be  $R_{g(D)}[Dy(D)]$ , which is easily seen to be

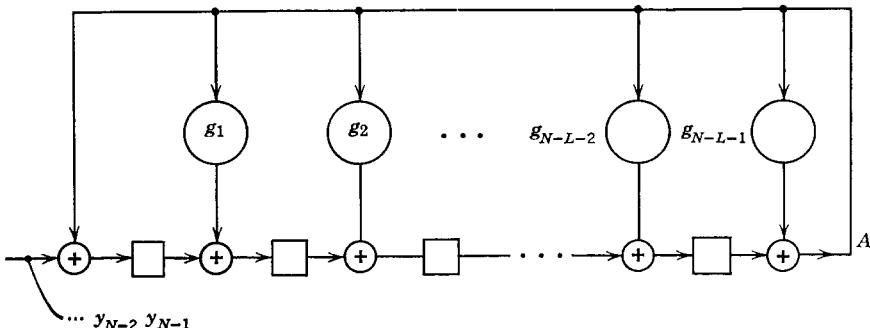


Figure 6.10.3. Circuit for calculating  $B_i(D)$  for each  $i$ .

just  $B_{N-1}(D)$ . Likewise successive shifts will generate  $B_{N-2}(D), B_{N-3}(D), \dots, B_1(D)$ . It can also be seen that the easiest way to find the  $B_i(D)$  corresponding to the shortest burst is to find the longest consecutive string of 0's at point  $A$  after  $y_0$  has entered the left-hand side of the shift register. Finally, to perform the decoding, it is simply necessary to cycle the shift register  $N$  more times and after the longest run of zeros has reappeared at  $A$ , the burst of errors is contained in the register.

A number of modifications are sometimes desirable in such a decoder. For example, the nonordinary bursts can be ignored since they are usually much less likely than the ordinary bursts. Also, any burst longer than some given length can be counted as a detected error. Finally, in determining which  $B_i(D)$  to use, the decoder can look at both the number of errors in the burst and the length of the burst.

We next investigate the burst correcting capability,  $b$ , of cyclic codes where  $b$  is the largest integer such that the optimum decoder can correct all bursts of length  $b$  or less. If a cyclic code has burst correcting capability  $b$ , then there is some burst of length  $b + 1$  which is decoded as some other burst of length  $b + 1$  or less, and consequently the sum of these bursts is a code

word. Cycling this code word so that the uncorrectable burst starts at position 0, we can represent the cycled code word as

$$x(D) = B(D) + D^m B'(D) \quad (6.10.11)$$

where  $B(D)$  has degree  $b$  and  $B'(D)$  has degree at most  $b$ .

Since  $x(D)$  is a code word, it can be expressed as  $A(D)g(D)$  where the degree of  $A(D)$ , say  $a$ , is the degree of the highest order term in  $D^m B'(D)$  minus  $(N - L)$ . For  $B'(D)$  to have degree at most  $b + 1$ , it is thus necessary for the coefficients of  $D^i$  in  $A(D)g(D)$  to be zero for

$$b + 1 \leq i \leq N - L + a - b - 1 \quad (6.10.12)$$

In other words, for an uncorrectable burst of length  $b + 1$  to exist, there must be some integer  $a$ ,  $0 \leq a \leq L$ , and some non-zero polynomial  $A(D)$  of degree  $a$ , such that  $A(D)g(D)$  has zero coefficients for  $i$  in the range given by (6.10.12). The correctable burst length of the cyclic code is thus the smallest  $b$  for which such a solution exists. For any given  $a$  and  $b$ , finding such a solution means to find a set of  $a - 1$  coefficients,  $A_1, \dots, A_{a-1}$ , to satisfy the  $N - L + a - 2b - 1$  linear equations indicated in (6.10.12). It is easy to see that  $b \leq (N - L)/2$ , for the burst  $B(D)$  generated by truncating  $g(D)$  to its first  $\lfloor (N - L + 1)/2 \rfloor$  terms is always confused with  $g(D) - B(D)$ . It is also easy to see that the bound  $b \leq \lfloor (N - L)/2 \rfloor$  for cyclic codes is equivalent to the more general bound in Theorem 6.10.1.

Unfortunately, relatively little is known about how to choose  $g(D)$  for a given  $N$  and  $L$  to maximize  $b$ . Fire (1959) has developed a large class of cyclic codes with reasonably large values of  $b$  and Elspas and Short (1962) have published a short table of cyclic codes with optimum values of  $b$ . Also Kasami (1963; 1964) has indicated a simplified procedure for solving the equations above and has also given a table of shortened cyclic codes with optimum values of  $b$ . For any  $(N, L)$  cyclic code, one generates a shortened cyclic code of any block length  $N'$ ,  $N - L < N' < N$  by omitting the first  $N - N'$  information digits of the cyclic code and regarding these omitted digits as being 0's in the generation of the check digits.

We next investigate the fraction of correctable bursts at lengths greater than the burst correcting capability.

**Theorem 6.10.2.** Let an  $(N, L)$  cyclic code have a burst correcting capability  $b$ . Then for  $b < b' \leq N - L$ , the fraction  $f(b')$  of bursts of length  $b'$  not correctly decoded by the optimum burst decoder is given by

$$f(b') \geq \begin{cases} 2^{-e(b')} & b' = b + 1 \\ \frac{1}{2}2^{-e(b')} & b + 1 < b' \leq N - L \end{cases} \quad (6.10.13)$$

$$f(b') \leq \begin{cases} (N - 1)2^{-e(b')} & b' = b + 1 \text{ and } N - L - b + 1 \leq b' \\ (N - 1)2^{-e(b')} & b + 1 < b' < N - L - b + 1 \end{cases} \quad (6.10.14)$$

where

$$e(b') = \begin{cases} b; & b + 1 \leq b' \leq N - L - b \\ N - L - b' + 1; & N - L - b + 1 < b' < N - L \end{cases}$$

*Discussion.* The theorem is of major interest when  $b$  and  $N - L$  are large, in which case the coefficients in (6.10.14) are relatively unimportant. The interesting point is that most bursts are corrected for  $b' < N - L - \log_2 N$ . For fixed rate  $L/N$  and increasing  $N$ , the ratio of this limit to the upper bound on burst correcting capability approaches 2. The function  $e(b')$  is sketched in Figure 6.10.4. It can be seen that increasing the burst correcting capability  $b$  raises the flat part of the  $e(b')$  function, decreasing the fraction of uncorrectable bursts in the vicinity of  $b' = (N - L)/2$ .

*Proof.* Define the syndrome  $S(D)$  associated with a noise sequence  $z(D)$  as

$$S(D) = R_{D^{N-1}}[z(D)h(D)] \quad (6.10.15)$$

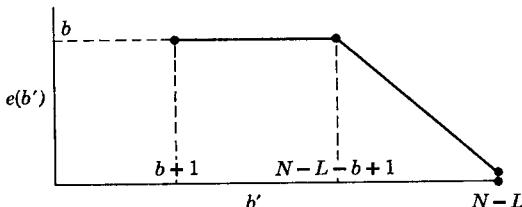


Figure 6.10.4. Exponent for fraction of uncorrectable bursts of length  $b'$ .

where  $h(D) = (D^N - 1)/g(D)$ . Note that two noise sequences have the same syndrome if their sum is a code word. Note that the set of all syndromes corresponding to all different  $z(D)$  is simply the set of code words in the dual code generated by  $h(D)$ . A burst of errors is called confusable if it has the same syndrome as another burst of shorter or equal length. Note that every uncorrectable burst is confusable and that at least half the confusable bursts of any length are incorrectly decoded by the optimum decoder. Finally the fraction of confusable bursts of length  $b'$  going from position 0 to  $b' - 1$  is the same as the fraction of confusable bursts of length  $b'$  starting at any other position.

For any integer  $m$ ,  $1 \leq m \leq N - 1$ , and any  $b'$ ,  $0 < b' \leq N - L$ , define  $A_{m,b'}(u,v)$  (see Figure 6.10.5) as the set of syndrome polynomials  $S(D)$  for which  $S_{m-1-(N-L-b')} = u$ ,  $S_{L+b'-1} = v$ , and  $S_i = 0$  for  $m - (N - L - b') \leq i \leq m - 1$  and  $L + b' \leq i \leq N - 1$ . The coefficients here are to be taken as remainders modulo  $N$ . It can be seen from the positions of the runs of 0's in Figure 6.10.5 that each syndrome in  $A_{m,b'}(1,1)$  can be represented both as  $S(D) = B_1(D)h(D)$  and as  $S(D) = R_{D^{N-1}}[D^m B_2(D)h(D)]$  where  $B_1(D)$  and

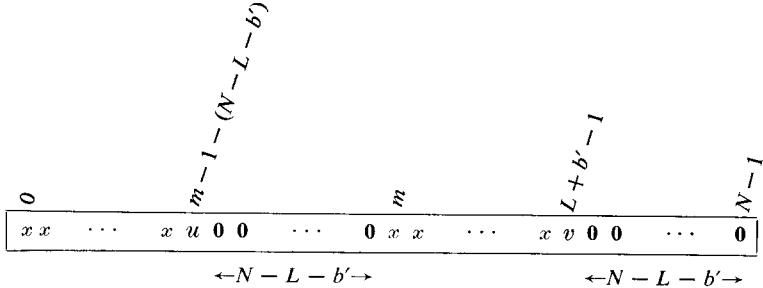


Figure 6.10.5 Set of syndromes in  $A_{m,b'}(u,v)$ ;  $x$ 's arbitrary binary digits.

$B_2(D)$  each have degree  $b'$ . We first bound the numbers of elements in these sets of syndromes and then determine their relationship to the number of uncorrectable bursts of length  $b'$ . Note that  $A_{m,b'}(\mathbf{0},\mathbf{0})$  always contains the all zero syndrome and is a subgroup of the set of all syndromes under modulo 2 polynomial addition. Note also that for each  $u, v$ ,  $A_{m,b'}(u,v)$  is either empty or a coset of  $A_{m,b'}(\mathbf{0},\mathbf{0})$ . Since the total number of syndromes is a power of 2, Lagrange's theorem (Theorem 6.3.1), implies that each of these subgroups and cosets have a number of elements given by a power of 2. Now define the set  $A_{m,b'}$  as

$$A_{m,b'} = \bigcup_{u=0,1} \bigcup_{v=0,1} A_{m,b'}(u,v) \quad (6.10.16)$$

From Figure 6.10.3, it is directly verified that

$$A_{m,b'-1} = A_{m,b'}(\mathbf{0},\mathbf{0}) \quad (6.10.17)$$

It follows that

$$\|A_{m,b'}\| = \|A_{m,b'-1}\| \alpha(m,b') \quad (6.10.18)$$

when  $\alpha(m,b')$  is the number of combinations of  $u, v$  such that  $A_{m,b'}(u,v)$  is nonempty. Since  $\|A_{m,b'}\|$  and  $\|A_{m,b'-1}\|$  are each powers of 2,  $\alpha(m,b')$  is 1, 2, or 4.

Next observe, from Figure 6.10.5, that if  $S(D) \in A_{m,b'-1}(u,v)$ , then  $DS(D) \in A_{m,b'}(u,v)$ . Thus if  $A_{m,b'}(u,v)$  is nonempty for one value of  $b'$  it is nonempty for all larger values of  $b'$ . It follows that  $\alpha(m,b')$  is nondecreasing with  $b'$ .

From the definition of  $b$  (the burst correcting capability of the code), there is some burst of length  $b+1$  going from position 0 to  $b$  which has the same syndrome as some burst of at most length  $b+1$  starting at, say, position  $m'$ . This syndrome is in  $A_{b+1,m'}(\mathbf{1},\mathbf{1})$ . Furthermore,

$$\|A_{b+1,m'}(\mathbf{1},\mathbf{1})\| = 1 \quad (6.10.19)$$

To see this, we observe that a larger number of syndromes would imply a nonzero syndrome in  $A_{b+1,m'}(\mathbf{0},\mathbf{0})$ . But such a syndrome would contain two runs of at least  $N - L - b$  zeros each, either connected or not. If connected, the entire run is of length at least  $N - L$ , which requires the syndrome to

be **0**. If unconnected, there are two bursts of length at most  $b$  with this syndrome, contradicting the definition of  $b$ .

Using the coset property of these sets and using (6.10.16) and (6.10.17), we have

$$\|A_{m',b+1}\| \geq 2; \quad \|A_{m',b}\| = 1 \quad (6.10.20)$$

Finally we observe that for  $b' = N - L$ , all syndromes are in  $A_{m',b'}$ ,

$$\|A_{m',N-L}\| = 2^{N-L} \quad (6.10.21)$$

Since  $\alpha(m', b')$  in (6.10.18) is nondecreasing with  $b'$  and takes on only the values 2 or 4 for  $b + 1 \leq b' \leq N - L$ , (6.10.20) and (6.10.21) completely specify  $\|A_{m',b'}\|$  for all intermediate values of  $b'$  as follows:

$$\|A_{m',b'}\| = \begin{cases} 2^{b'-b}; & b \leq b' \leq N - L - b \\ 2^{2b'-(N-L)}; & N - L - b \leq b' \leq N - L \end{cases} \quad (6.10.22)$$

Using (6.10.17) and the fact that  $\|A_{m',b'}(0,0)\| = \|A_{m',b'}(1,1)\|$ ,

$$\|A_{m',b'}(1,1)\| = \begin{cases} 2^{b'-b-1}; & b + 1 \leq b' \leq N - L - b + 1 \\ 2^{2b'-(N-L)}; & N - L - b + 1 \leq b' \leq N - L \end{cases} \quad (6.10.23)$$

Each syndrome  $S(D) \in A_{m',b'}(\mathbf{1},\mathbf{1})$  for which  $S_0 = \mathbf{1}$  is the syndrome for a confusable burst of length  $b'$  from 0 to  $b' - 1$ . For  $b' = b + 1$ , the single syndrome in  $A_{m',b+1}(\mathbf{1},\mathbf{1})$  has  $S_0 = \mathbf{1}$ , and for larger  $b'$ , exactly half the syndromes in  $A_{m',b'}(\mathbf{1},\mathbf{1})$  have  $S_0 = \mathbf{1}$ . To see this we observe that if  $S(D) \in A_{m',b+1}(\mathbf{1},\mathbf{1})$ , then  $S'(D) = D^{b'-b-1}S(D)$  is in  $A_{m',b'}(\mathbf{1},\mathbf{1})$  with  $S'_0 = \mathbf{0}$  and  $S''(D) = (\mathbf{1} + D^{b'-b-1})S(D)$  is in  $A_{m',b'}(\mathbf{1},\mathbf{1})$  with  $S''_0 = \mathbf{1}$ . Since the set of  $S(D) \in A_{m',b'}(\mathbf{1},\mathbf{1})$  with  $S_0 = \mathbf{1}$  and that with  $S_0 = \mathbf{0}$  are both cosets of the subgroup of  $S(D) \in A_{m',b'}(\mathbf{0},\mathbf{0})$  with  $S_0 = \mathbf{0}$ , the sets have the same size. Finally there are  $2^{b'-2}$  different bursts from position 0 to  $b' - 1$ , so that the fraction of bursts from position 0 to  $b' - 1$  that are confusable is lower bounded by

$$\begin{aligned} \|A_{m',b'}(1,1)\| 2^{-b'+2}; & \quad b' = b + 1 \\ \|A_{m',b'}(1,1)\| 2^{-b'+1}; & \quad b' > b + 1 \end{aligned}$$

Since at least one half of the confusable bursts are not correctly decoded, this, combined with (6.10.23), gives the lower bound on  $f(b')$  in (6.10.13).

To establish the upper bound on  $f(b')$ , we must first bound  $\|A_{m',b'}(\mathbf{1},\mathbf{1})\|$  for each  $m$ . For any given  $m$ , let  $b_m$  be the largest integer for which  $\|A_{m,b_m}\| = 1$ . Then  $\|A_{m,b_m+1}(u,v)\| = 1$  for some nonzero choice of  $u, v$ . If

$\|A_{m,b_m+1}(\mathbf{1},\mathbf{1})\| = 1$ , then all the previous arguments can be applied and  
 $\|A_{m,b'}(\mathbf{1},\mathbf{1})\| \leq \|A_{m',b'}(\mathbf{1},\mathbf{1})\|; \quad b + 1 \leq b' \leq N - L, \quad (6.10.24)$

On the other hand, if  $\|A_{m,b_m+1}(\mathbf{1},\mathbf{1})\| = 0$ , then  $\|A_{m,b'}(\mathbf{1},\mathbf{1})\| = 0$  for all  $b'$  such that  $\alpha(m,b') = 2$ , and  $\alpha(m,b') = 4$  for  $\|A_{m,b'}(\mathbf{1},\mathbf{1})\| > 0$ . It follows that

$$\|A_{m,b'}(\mathbf{1},\mathbf{1})\| = \begin{cases} 2^{2b'-2-(N-L)} \\ 0 \end{cases} \quad \text{or} \quad (6.10.25)$$

and thus (6.10.24) is valid for all  $m$ ,  $1 \leq m \leq N - L$ .

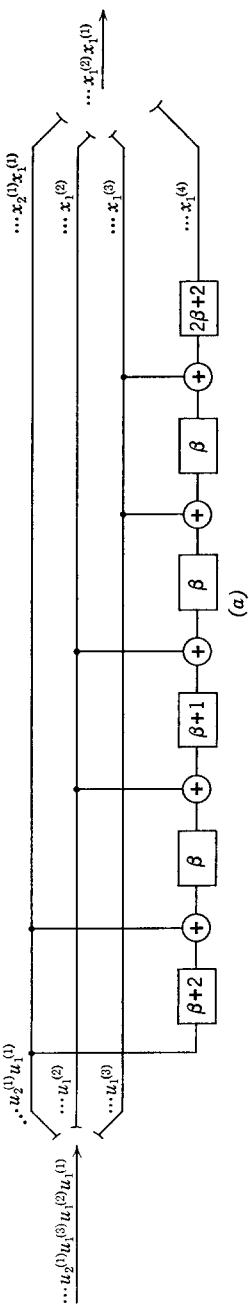
Every uncorrectable burst of length  $b'$  in position 0 to  $b' - 1$  corresponds to a syndrome  $S(D)$  in  $A_{m,b'}(\mathbf{1},\mathbf{1})$  for some  $m$ ,  $1 \leq m \leq N - L$ , with  $S_0 = \mathbf{1}$ . By the same argument as before, if  $\|A_{m,b_m+1}(\mathbf{1},\mathbf{1})\| = 1$  then half the syndromes in  $\|A_{m,b'}(\mathbf{1},\mathbf{1})\|$  for each  $b' > b_m + 1$  have  $S_0 = \mathbf{1}$ . Thus the total number of uncorrectable bursts of length  $b'$  in position 0 to  $b' - 1$  is upper bounded by  $(\frac{1}{2})(N - 1) \|A_{m',b'}(\mathbf{1},\mathbf{1})\|$  for  $b + 1 < b' \leq N - L - b + 1$  and by  $(N - 1) \|A_{m',b'}(\mathbf{1},\mathbf{1})\|$  elsewhere. Using (6.10.23), this yields the upper bound on  $f(b')$  in (6.10.13). |

Another block coding technique for burst noise channels involves the use of Reed-Solomon (1960) codes. Using symbols from  $GF(2^m)$  for some  $m$ , the block length is  $N = 2^m - 1$ . For an arbitrarily chosen odd minimum distance  $d$ , the number of information symbols is  $L = N - d + 1$  and any combination of  $(d - 1)/2 = (N - L)/2$  errors can be corrected. If we represent each letter in a code word by  $m$  binary digits, then we obtain a binary code with  $Lm$  information symbols and block length  $Nm$ . Any noise sequence which alters at most  $(N - L)/2$  of these  $m$  length sequences can be corrected, and thus the code has a burst correcting capability of  $m[(N - L)/2 - 1] + 1$ , along with the capability of correcting many combinations of multiple shorter bursts. It is seen that by increasing  $m$  for fixed  $L/N$ , we approach the theoretical limit on burst correcting capability given by Theorem 6.10.1.

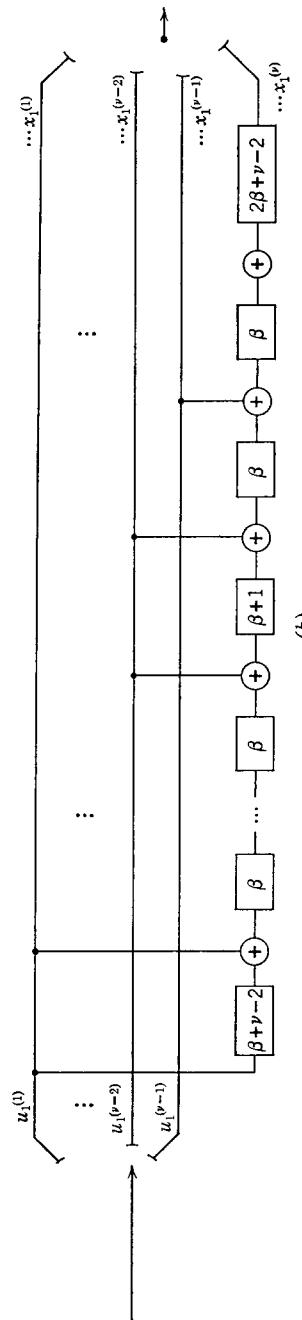
### Convolutional Codes

In this section we shall discuss a number of specific techniques for correcting bursts of errors using convolutional codes. All of the techniques may be modified and generalized in various ways, but at this stage of development the design of convolutional burst correcting codes is primarily an art best taught by example. The first technique to be discussed was developed independently by Iwadare (1967) and Massey.\* It applies to convolutional codes with a rate (in binary units) of the form  $(v - 1)/v$  where  $v$  is an arbitrary positive integer. For any positive integer  $\beta$ , this technique gives a burst correcting capability of  $\beta v$  relative to a guard space  $\beta v(2v - 1) + \frac{1}{2} v^2(v - 1) - 1$ . In the limit of large  $\beta$ , the ratio of guard space to burst

\* Unpublished memorandum, July 1967.



(a)



(b)

Figure 6.10.6. (a) Convolutional encoder for  $R = \frac{1}{2}$  (bit)  $\beta$  denotes shift register of  $\beta$  stages. (b) Convolutional encoder for arbitrary  $R = (v - 1)/v$ .

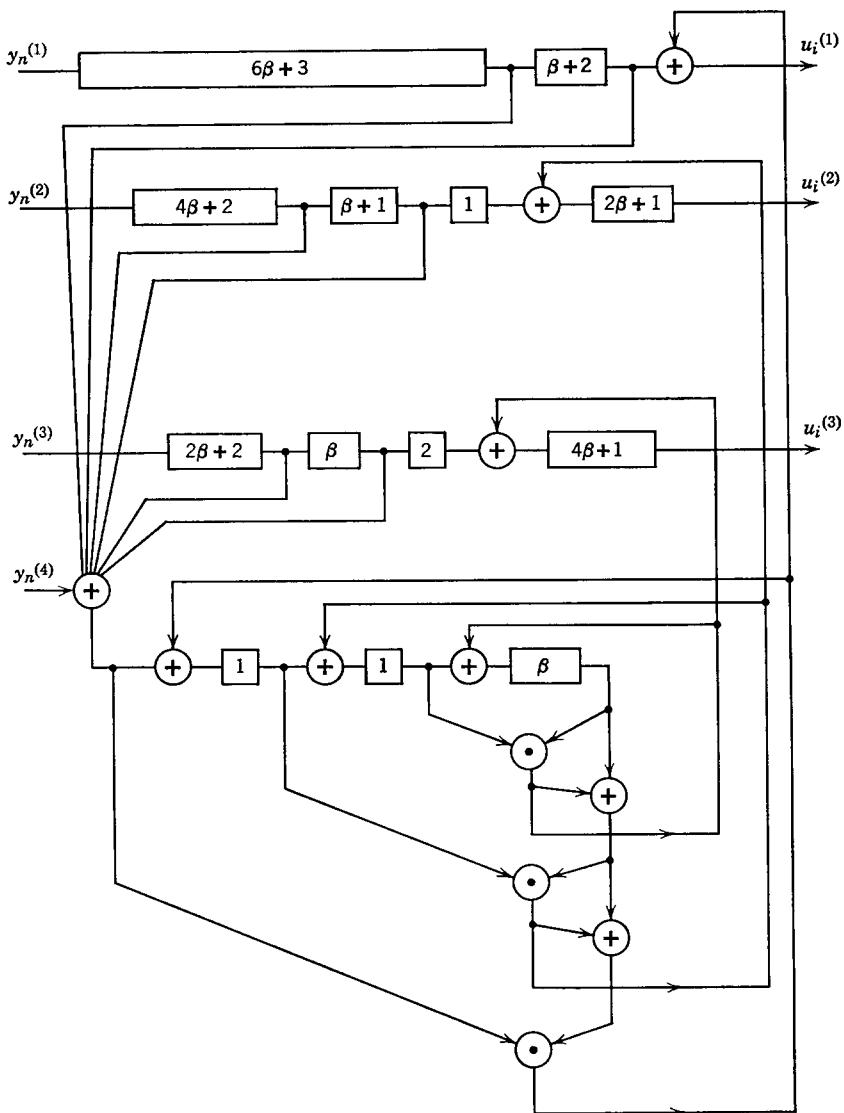


Figure 6.10.7. Burst-correcting decoder (Iwadare-Massey technique)  $R = \frac{3}{4}$ .

correcting capability tends to  $2v - 1$  which agrees with the upper bound on guard space to burst correcting capability given by Theorem 6.10.1.

A block diagram of the encoder for the Iwadare-Massey technique is given in Figure 6.10.6, first for the specific case  $v = 4$  ( $R = \frac{3}{4}$  bits) and then in general. The decoder for  $v = 4$  is given in Figure 6.10.7, and it should be

obvious how to modify it for arbitrary  $v$  after the following discussion. The encoding for  $v = 4$  follows the rule, for each  $n$ ,

$$\begin{aligned}x_n^{(i)} &= u_n^{(i)}; \quad i = 1, 2, 3 \\x_n^{(4)} &= u_{n-2\beta-2}^{(3)} \oplus u_{n-3\beta-2}^{(3)} \oplus u_{n-4\beta-2}^{(2)} \oplus u_{n-5\beta-3}^{(2)} \oplus u_{n-6\beta-3}^{(1)} \oplus u_{n-7\beta-5}^{(1)} \quad (6.10.26)\end{aligned}$$

The syndrome digits  $S_n$  are given in terms of the received digits from the channel by

$$S_n = y_n^{(4)} \oplus y_{n-2\beta-2}^{(3)} \oplus y_{n-3\beta-2}^{(3)} \oplus y_{n-4\beta-2}^{(2)} \oplus y_{n-5\beta-3}^{(2)} \oplus y_{n-6\beta-3}^{(1)} \oplus y_{n-7\beta-5}^{(1)} \quad (6.10.27)$$

It can be seen from (6.10.26) that the received digit values in (6.10.27) can be replaced by the noise digits. Also if the noise sequence contains only bursts

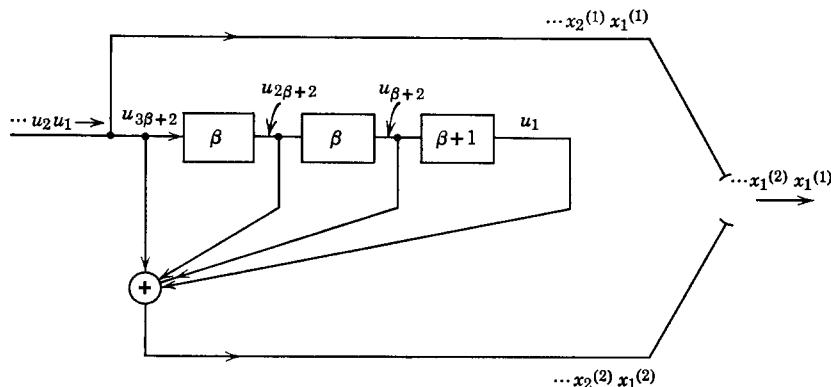


Figure 6.10.8. Encoder for diffuse threshold decoding.

of at most  $v\beta$  digits relative to a guard space  $\beta v(2v - 1) + \frac{1}{2}v^2(v - 1) - 1$ , then it is easily seen that each  $S_n$  can involve at most one incorrect noise digit. In fact the guard space above is exactly what is required to prevent  $z_n^{(4)}$  from being the first error in one burst when  $z_{n-7\beta-5}^{(1)}$  is the last error in the previous burst.

In order to understand the operation of the decoder, it is helpful to consider a burst of  $v\beta$  errors as it passes through the decoder. Such a burst will generate four successive bursts of 1's in the syndrome register. The first of these bursts will be generated by the errors in the fourth stream of incoming digits, the next by the errors in the third stream of digits, the next by the second stream and the last by the first stream. The first burst of syndrome 1's has a length of at most  $\beta$  and thus cannot trigger any of the "and" gates

at the bottom of Figure 6.10.7 that cause corrections. There must be a sequence of at least  $(\beta + 2)$  0's in the syndrome sequence between the first and second burst of 1's and thus these bursts cannot overlap in the syndrome register. Each error in the third stream of received digits gives rise to two syndrome 1's spaced  $\beta$  digits apart and thus each of these causes a correction at the appropriate time in the third stream. The modulo 2 adders in the lower part of the figure inhibit any corrections in the first and second streams during this period. Each correction also causes the appropriate syndrome digit to be reset so that after the last error in the third stream of received digits is corrected the syndrome register contains zeros except perhaps in the left most two positions. Correction of the second stream and then first stream of received digits proceeds in a similar way.

A very similar class of codes has been developed through the joint work of Wyner and Ash (1963), Berlekamp (1964), and Massey (1965). That technique yields a somewhat smaller guard space for a given burst correcting capability than the Iwadare-Massey technique, but the added complexity of implementation appears to make it less useful in practical applications.

Another technique, with a somewhat different approach, is diffuse threshold decoding, developed by Massey and Kohlenberg (1964). This technique requires a little more guard space and a little more instrumentation for a given burst correction capability than the previous techniques, but has more flexibility to deal with a combination of bursts and independent errors. The technique is best described in terms of the example given in Figures 6.10.8 and 6.10.9. The parameter  $\beta$  in these figures is arbitrary and determines the burst correcting capability of the code. As we shall soon see, such a code has a burst correcting capability of  $2\beta$  digits relative to a guard space of  $6\beta + 2$  digits.

The analysis of the decoder in Figure 6.10.9 is almost identical to that of the threshold decoder in Figure 6.8.2. In particular the inputs to the threshold device, assuming that  $y_1^{(1)}$  is in the right most stage of the shift register and that no previous decoding errors have been made, are

$$\begin{aligned} S_1 &= z_1^{(1)} \oplus z_1^{(2)} \\ S_{\beta+1} &= z_1^{(1)} \oplus z_{\beta+1}^{(1)} \oplus z_{\beta+1}^{(2)} \\ S_{2\beta+1} + S_{3\beta+1} &= z_1^{(1)} \oplus z_{2\beta+1}^{(2)} \oplus z_{3\beta+1}^{(1)} \oplus z_{3\beta+1}^{(2)} \\ S_{3\beta+2} &= z_1^{(1)} \oplus z_{\beta+2}^{(1)} \oplus z_{\beta+2}^{(1)} \oplus z_{3\beta+2}^{(1)} \oplus z_{3\beta+2}^{(2)} \end{aligned} \quad (6.10.28)$$

It can be seen that the linear combinations of noise digits on the right hand side of (6.10.28) are orthogonal on  $z_1^{(1)}$  and thus  $y_1^{(1)}$  will be correctly decoded if at most two errors appear in those equations. From the standpoint of bursts, it is easy to check from (6.10.28) that any burst of at most  $2\beta$  digits

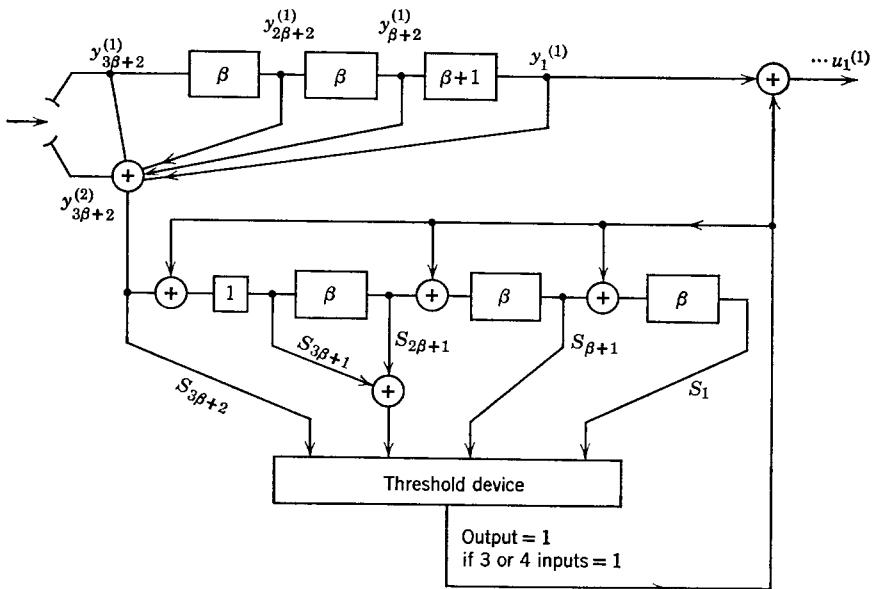


Figure 6.10.9. Diffuse threshold decoder.

can effect at most two of the digits in (6.10.28). The guard space is determined by observing that if  $z_1^{(2)}$  is the last digit in one burst, then  $z_{3\beta+2}^{(2)}$  must precede the beginning of the next burst, but if  $z_1^{(1)}$  is the last digit in one burst, then  $z_{3\beta+2}^{(2)}$  may be the first digit of the next burst.

It can be seen that there is a close conceptual relationship between this technique and the interlacing technique. In both techniques, the digits which are constrained together by the encoder are separated on the channel so as to essentially convert the bursts into independent errors. It can be seen however, that the guard to burst length ratio in Figure 6.10.9 is much smaller than that which would be obtained by interlacing the code in Figure 6.8.1.

There is also a close relationship between the Iwadare-Massey technique and diffuse threshold decoding. In terms of independent errors, the decoder in Figure 6.10.7 can be regarded as a single error correcting threshold decoder and that in Figure 6.10.9 as a double error correcting threshold decoder. Thus it is not surprising that the double error correcting diffuse threshold decoder has greater complexity but also greater flexibility in the types of errors it can decode than the Iwadare-Massey technique.

As a final example of burst error correction with convolutional codes, we consider the following technique due to Gallager (1965b). We first discuss the

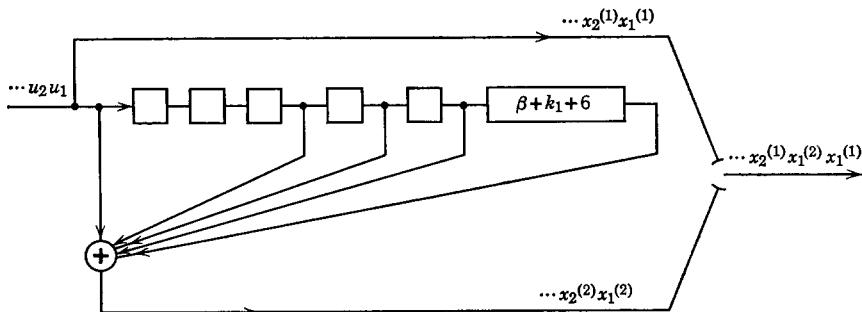


Figure 6.10.10. Time-diversity burst encoder.

particular example given in Figures 6.10.10 and 6.10.11. The coder and decoder shown there are designed to correct most bursts of length up to  $2\beta$  relative to a guard space  $2(\beta + 10 + k_1 + k_2) - 1$ . The parameters  $\beta$ ,  $k_1$ , and  $k_2$  are arbitrary but typically we can think of  $\beta$  as having the order of magnitude of 1000 and  $k_1$  and  $k_2$  as being less than 10. It is seen that we are trying to correct bursts with lengths almost three times the upper bound on burst correcting capability given by Theorem 6.10.1, so it should be clear that not *all* bursts of length  $2\beta$  can be corrected.

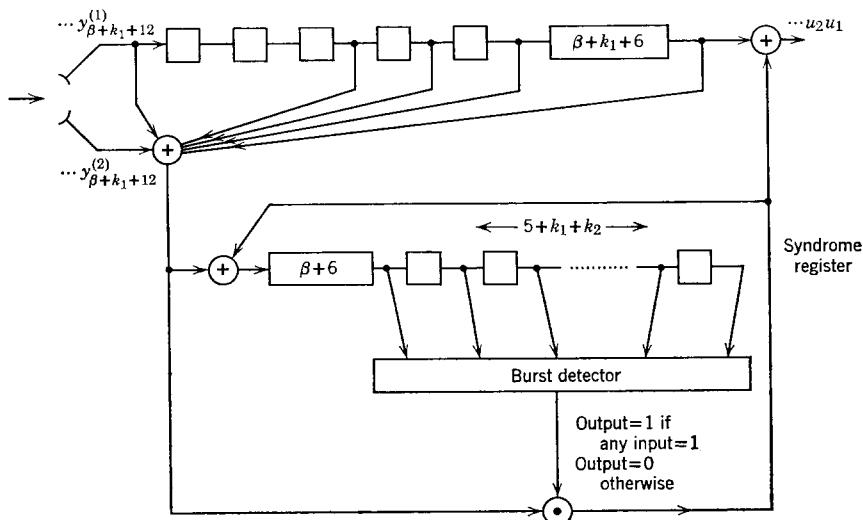


Figure 6.10.11. Time-diversity burst decoder.

The qualitative operation of the decoder in Figure 6.10.11 can be best understood by considering what happens when a burst of length  $2\beta$  enters the decoder. As the burst enters the decoder, the errors will cause many of the syndrome digits to be 1 and in fact the errors, as they pass through the first five stages of the upper shift register will generate a burst of 1's in the syndrome sequence of length at most  $\beta + 6$ . By the time the leading edge of this burst of 1's in the syndrome sequence hits the first stage of the syndrome register that leads to the burst detector, however, all the errors in the information stream  $y_{(1)}^i$  will be contained in the untapped upper shift register of length  $\beta + k_1 + 6$ . Thus each subsequent generation of a 1 in the syndrome sequence will be caused by an error in the information stream at the right hand side of the upper shift register. This error will be corrected if there are any 1's in the part of the syndrome register connected to the burst detector.

We can get a rough idea of the effectiveness of the technique by assuming that within the burst, the noise digits are independent equiprobable binary digits. If the first error in the burst is in the information stream, then at the time it is to be corrected, the left most  $6 + k_1$  syndrome digits connected to the burst detector are independent equiprobable binary digits and the probability of not correcting the error is  $2^{-6-k_1}$ . Similarly the probability of not correcting an error in the middle of the burst is  $2^{-6-k_1-k_2}$  and the probability of not correcting the final error in the burst is roughly  $2^{-6-k_2}$ . We see from this that the effect of increasing  $k_1$  and  $k_2$  is to decrease the probability of failing to correct a burst but to increase the equipment complexity and the required guard space.

In effect, this technique is a form of time-diversity transmission. If a burst of errors corrupts the information stream over a given interval, then the later digits in the parity check stream can be used to reconstitute the stream. The major difference is that in the usual form of time diversity, one uses channel measurements to determine which stream of digits is most reliable, whereas here the coding constraints are used to make that determination. On many physical channels, there are so many effects which can cause error bursts that it is difficult to make reliable determinations of where the bursts are from channel measurements, and the use of the coding constraints is both more reliable and more simple.

There are a number of modifications that can be made in this technique. Clearly the particular set of register stages at the left side of the register in Figure 6.10.10 which are added together to form the check digits can be changed. A more important modification is to use the burst detector in Figure 6.10.11 both to correct relatively isolated errors and also to detect more serious bursts. Finally the same technique can be used for arbitrary rates of the form  $R = (v - 1)/v$ , correcting most bursts of length  $v\beta$  with a guard space of slightly over  $v(v - 1)\beta$ .

## Summary and Conclusions

In the previous chapters we have been concerned with the theoretical limitations on the transmission of data through noisy channels and here we have been concerned with techniques for achieving performance close to those limitations. From a practical standpoint, one would like to know the tradeoff between data rate, system cost, and error probability for any given coding technique and channel. Unfortunately, system cost is a very nebulous parameter in a rapidly changing technology and real channels are usually subject to a variety of mathematically untractable effects. Thus, all we have been able to achieve here is to describe a number of coding and decoding techniques, to show what is involved in their instrumentation, and to give some insight into their performance on various simple mathematical models of channels. We have not attempted to show what technique should be used in a given situation, but have attempted to give the reader enough background to treat specific engineering problems in this area meaningfully.

Perhaps the most important point about coding techniques is that they are practical in a variety of communication systems. This point can be easily missed because of a strong human tendency to confuse the familiar with the practical. This point is also sometimes missed because of the large number of digital operations performed in a coder and decoder. Such operations tend to be highly reliable and easy to implement, even though difficult to understand.

The first six sections of the chapter were concerned with building up the background necessary to do research and to read the literature in algebraic coding theory. Section 6.7 dealt with the BCH codes, which are the most interesting and practically promising variety of the algebraic techniques. Section 6.8 introduced convolutional coding and threshold decoding. Section 6.9 provided a fairly complete introduction to sequential decoding. For channels not subject to long fades, sequential decoding appears to be the most promising practical coding technique in existence at the present. Finally in Section 6.10, a brief introduction was given to a variety of coding techniques for channels subject to burst noise.

## Historical Notes and References

Peterson (1961) is the classic reference work in the area of algebraic coding techniques and discusses a variety of topics not covered here. Berlekamp (1968) is a more advanced and up to date reference. For those who can obtain a copy, the lecture notes of Massey (1967) provide an excellent treatment of the subject. Wozencraft and Jacobs (1965) provide an excellent introduction to sequential decoding and other references on sequential decoding are given in Section 6.9. Massey (1963) is the standard reference on threshold decoding.

It is not feasible here to list all the significant papers on coding techniques. Quite extensive bibliographies in the area are given, however, by Peterson (1961), Peterson and Massey (1963), and Kotz (1967). For historical purposes, however, we should note the following. Hamming (1960) was the precursor to most of the work in algebraic coding theory. Slepian (1956) first put parity check codes on a firm mathematical basis. Elias (1955) established the results of Section 6.2, and also first discussed convolutional codes. Prange (1957) first treated cyclic codes. Decoding techniques for the Bose-Chaudhuri (1960) and Hocquenghem (1959) codes was first developed by Peterson (1960) and Zierler (1960). Sequential decoding was invented by Wozencraft (1957) and the algorithm discussed here by Fano (1963). The first effective burst error correcting codes were developed by Hagelbarger (1959) and Fire (1959).

## APPENDIX 6A

Before proving Theorem 6.9.1, it is helpful to rewrite the decoding algorithm rules as in Figure 6A.1. It will be observed that the conditions in Figure 6A.1 include all the conditions of Figure 6.9.4 and some others as well. Thus, if *any* rule in Figure 6A.1 applies, it must be the same rule as in Figure 6.9.4. To see that a rule applies to every hypothesis in Figure 6A.1, we can use induction on successive hypothesis. That is, for each rule in Figure 6A.1, we assume that the rule is applied and investigate the conditions that can arise on the next hypothesis. For example, if rule 1 is applied, then on the next hypothesis  $\Gamma_{l-1} \geq T$ , and this is the new condition that has been added to rules 1, 2, and 3.

| Rule | Conditions on Node |  |  | Action to be Taken |               |
|------|--------------------|--|--|--------------------|---------------|
|      | Previous Move      | Comparison of $\Gamma_{l-1}$ and $\Gamma_l$ with Initial Threshold $T$ |  | Final Threshold    | Move          |
| 1    | <i>F or L</i>      | $T \leq \Gamma_{l-1} < T + \Delta$ ; $\Gamma_l \geq T$                 |  | Raise              | <i>F</i>      |
| 2    | <i>F or L</i>      | $\Gamma_{l-1} \geq T + \Delta$ ; $\Gamma_l \geq T$                     |  | No change          | <i>F</i>      |
| 3    | <i>F or L</i>      | $\Gamma_{l-1} \geq T$ $\Gamma_l < T$                                   |  | No change          | <i>L or B</i> |
| 4    | <i>B</i>           | $\Gamma_{l-1} < T$ $\Gamma_l \geq T$                                   |  | Lower by $\Delta$  | <i>F</i>      |
| 5    | <i>B</i>           | $\Gamma_{l-1} \geq T$ $\Gamma_l \geq T$                                |  | No change          | <i>L or B</i> |

Figure 6A.1. Equivalent set of rules to those in Figure 6.9.4.

*Proof of Theorem 6.9.1 (Part a).* We wish to show that the path thresholds and values associated with a hypothesis of a node  $\mathbf{u}_l$  satisfy (for  $0 \leq i \leq l - 1$ ):

$$T_i \leq \Gamma_i \quad (6A.1)$$

$$T_{i+1} \geq T_i \quad (6A.2)$$

$$T_{i+1} \geq T_i + \Delta \Rightarrow T_i + \Delta > \Gamma_i \quad (6A.3)$$

$$T_{i+1} \geq T_i + \Delta \Rightarrow T_{i+1} > \Gamma_i \quad (6A.4)$$

The most recent hypothesis of any antecedent  $\mathbf{u}_i$  of the current node  $\mathbf{u}_l$  must have been an  $F$  hypothesis since  $\mathbf{u}_l$  can be reached only by moving forward from  $\mathbf{u}_i$ . Thus one of the rules 1, 2, or 4 must have applied to  $\mathbf{u}_i$  and (6A.1) is satisfied in each case. We next observe that (6A.4) is a direct consequence of (6A.3), the implication of (6A.4) being a combination of the inequalities in (6A.3). We now establish (6A.2) and (6A.3) by using induction on the successive nodes visited by the decoder. For the initial hypothesis of the origin node, (6A.2) and (6A.3) are vacuously valid since the range of  $i$  is empty for  $l = 0$ . Now assume that (6A.2) and (6A.3) are valid for a hypothesis of an arbitrary node  $\mathbf{u}_l$  with the threshold sequence  $T_0, \dots, T_l$ . By considering first a forward, then a lateral, and then a backward move from  $\mathbf{u}_l$ , we show by straightforward but tedious means that (6A.2) and (6A.3) are always satisfied on the next node hypothesized.

*Forward move.* After a move from  $\mathbf{u}_l$  forward to a node  $\mathbf{u}_{l+1}$ , the thresholds  $T_0, \dots, T_l$  are unchanged but a new threshold  $T_{l+1}$  is added to the sequence. Since the move to  $\mathbf{u}_{l+1}$  was forward, rules 1, 2, or 3 apply to  $l + 1$  and the final threshold  $T_{l+1}$  is greater than or equal to the initial threshold  $T_l$ , establishing (6A.2) for  $\mathbf{u}_{l+1}$ . If  $T_{l+1} \geq T_l + \Delta$ , the threshold was raised and rule 1 must have applied. Since the initial threshold is  $T_l$  and the prior value is  $\Gamma_l$ , the left-most condition on rule 1 is  $\Gamma_l < T_l + \Delta$ , establishing (6A.3).

*Lateral move.* Let  $T_0, \dots, T_l$  be the path thresholds for a hypothesis of  $\mathbf{u}_l$  and assume that a lateral or backward move occurs from  $\mathbf{u}_l$ . Either rule 3 or 5 must apply to  $\mathbf{u}_l$ , and in either case, from the rules,  $T_l \leq \Gamma_{l-1}$ . We see that lateral or backward moves can occur only for  $l > 0$ . By the inductive assumption, (6A.3) and thus (6A.4) apply to  $\mathbf{u}_l$ . From (6A.4) with  $i = l - 1$ ,  $T_l \geq T_{l-1} + \Delta \Rightarrow T_l > \Gamma_{l-1}$ . Since we have already seen that  $T_l \leq \Gamma_{l-1}$ , it must be that  $T_l < T_{l-1} + \Delta$ . Since  $T_l \geq T_{l-1}$  from (6A.2), and since  $T$  can change only in increments of  $\Delta$ , we conclude that:

$$\text{For an } L \text{ or } B \text{ move from } \mathbf{u}_l, T_l = T_{l-1} \quad (6A.5)$$

On a lateral move from  $\mathbf{u}_l$  to a node  $\mathbf{u}'_l$ , the path thresholds  $T_0, \dots, T_{l-1}$  are unchanged. The initial threshold on the hypothesis of  $\mathbf{u}'_l$  is  $T_l$  and from (6A.5)  $T_l = T_{l-1}$ . Let  $T'_l$  be the final threshold on the hypothesis of  $\mathbf{u}'_l$ . Since rules 1, 2, or 3 must apply to  $\mathbf{u}'_l$ , the threshold cannot be lowered and thus  $T'_l \geq T_{l-1}$ , establishing (6A.2). If  $T'_l \geq T_{l-1} + \Delta$ , then the threshold must have been raised on  $\mathbf{u}'_l$ , rule 1 applied, and  $\Gamma_{l-1} < T_{l-1} + \Delta$ , establishing (6A.3).

*Backward move.* Again let  $T_0, \dots, T_l$  be the path thresholds for the hypothesis of  $\mathbf{u}_l$  and assume a backward move to  $\mathbf{u}_{l-1}$ . From (6A.5), the initial threshold on

the hypothesis of  $\mathbf{u}_{l-1}$  is  $T_l = T_{l-1}$ . Since the move from  $\mathbf{u}_l$  is backward, either rule 5 or rule 4 applies to  $\mathbf{u}_{l-1}$ . If rule 5 applies, then the final threshold,  $T'_{l-1}$ , on the new hypothesis of  $\mathbf{u}_{l-1}$ , is the same as the initial threshold which, from above, is  $T_{l-1}$ . Thus, since (6A.2) and (6A.3) are valid for  $\mathbf{u}_l$ , they are also valid for the new hypothesis of  $\mathbf{u}_{l-1}$ . If rule 4 applies, then, from Figure 6A.1 (remembering that  $T_{l-1}$  is the initial threshold),  $\Gamma_{l-2} < T_{l-1}$ . From (6A.1),  $T_{l-2} \leq \Gamma_{l-2}$ , and since the thresholds can only change in increments of  $\Delta$ ,  $T_{l-2} \leq T_{l-1} - \Delta$ . The final threshold,  $T'_{l-1}$  on the new hypothesis of  $\mathbf{u}_{l-1}$ , is  $T_{l-1} - \Delta$ , so that  $T_{l-2} \leq T'_{l-1}$ , establishing (6A.2) for  $\mathbf{u}_{l-1}$ . Finally, if  $T'_{l-1} \geq T_{l-2} + \Delta$ , then  $T_{l-1} \geq T_{l-2} + \Delta$  and the validity of (6A.3) for  $\mathbf{u}_l$  implies its validity for  $\mathbf{u}_{l-1}$ , completing the proof of part (a).

The following corollary to part (a) of the theorem will be needed in the proofs of parts (b) and (c).

**COROLLARY.** If a node  $\mathbf{u}$  is  $F$  hypothesized with a final threshold  $T$ , then  $T$  is the initial threshold on the first subsequent hypothesis of each of the immediate descendants of  $\mathbf{u}$  and on the first subsequent hypothesis of  $\mathbf{u}$ .

*Proof.* The statement is obvious for the first immediate descendant of  $\mathbf{u}$  to be hypothesized. The first hypothesis of each of the other immediate descendants must occur on a lateral move from the previous immediate descendant. But from (6A.5), such a lateral move can occur only when the final threshold prior to the move is  $T$ . Likewise, the first backward move to  $\mathbf{u}$  is from the last immediate descendant and the threshold is again  $T$ . |

*Proof of Theorem 6.9.1 (Part b).* We wish to show that the final threshold  $T$  on the first  $F$  hypothesis of each node is related to the value  $\Gamma'$  of the node by

$$T \leq \Gamma' < T + \Delta \quad (6A.6)$$

and also that each subsequent  $F$  hypothesis of the node is with a final threshold  $\Delta$  below the previous one.

We use induction on the path length of the nodes, first verifying the theorem for the origin node and then showing that, if the theorem is true for any given node, it is true for each immediate descendant of that node. The initial hypothesis of  $\mathbf{u}_0$  is an  $F$  hypothesis and satisfies (6A.6) by the initial conditions of the decoder. From the corollary, the initial threshold on each subsequent hypothesis of the origin is the same as the final threshold on the previous hypothesis. Since  $\Gamma_{-1} = -\infty$ , rule 4 applies to each such hypothesis, making it an  $F$  hypothesis with the final threshold reduced by  $\Delta$ .

Now assume that part (b) of the theorem is valid for a node  $\mathbf{u}_{l-1}$  of value  $\Gamma_{l-1}$  and let  $\mathbf{u}_l$  be an immediate descendant with value  $\Gamma_l$ . By assumption, the final threshold  $T$  on the first  $F$  hypothesis of  $\mathbf{u}_{l-1}$  satisfies

$$T \leq \Gamma_{l-1} < T + \Delta \quad (6A.7)$$

From the corollary,  $T$  is the initial threshold on the first hypothesis of  $\mathbf{u}_l$ . We consider separately now the cases where  $\Gamma_l \geq T$  and  $\Gamma_l < T$ .

If  $\Gamma_l \geq T$ , the first hypothesis of  $\mathbf{u}_l$  satisfies the conditions of rule 1, and the final threshold  $T_l$  is adjusted to satisfy (6A.6) for  $\mathbf{u}_l$ . From the corollary, the initial

threshold on the next return to  $\mathbf{u}_l$  is  $T_l$ . If  $T_l \geq T + \Delta$  (that is, if the threshold was raised on the original hypothesis of  $\mathbf{u}_l$ ), then from (6A.7),  $T_l > \Gamma_{l-1}$ , rule 4 applies, and the return is an  $F$  hypothesis with final threshold  $T_l - \Delta$ . Using the same argument, the final threshold is reduced by  $\Delta$  on each subsequent return to  $\mathbf{u}_l$  until the final threshold is  $T$ . On the next return to  $\mathbf{u}_l$ , the initial threshold is less than or equal to  $\Gamma_{l-1}$ , rule 5 applies, and a lateral or backward move occurs. Before the next  $F$  hypothesis of  $\mathbf{u}_l$  can occur then, another  $F$  hypothesis of  $\mathbf{u}_{l-1}$  must occur, by assumption at a final threshold  $T - \Delta$ . Thus the initial and final threshold on the next  $F$  hypothesis of  $\mathbf{u}_l$  is also  $T - \Delta$ . Similarly, subsequent  $F$  hypotheses of  $\mathbf{u}_l$  alternate with  $F$  hypotheses of  $\mathbf{u}_{l-1}$ , and the threshold is  $\Delta$  lower on each.

Finally, consider the case where  $\Gamma_l < T$ . Rule 3 then applies to the first hypothesis of  $\mathbf{u}_l$ , a lateral or backward move occurs, and  $\mathbf{u}_l$  is rehypothesized only after the next  $F$  hypothesis of  $\mathbf{u}_{l-1}$ , this time at a final threshold  $T - \Delta$ . The initial threshold is  $\Delta$  lower on each successive hypothesis of  $\mathbf{u}_l$  until the threshold is less than or equal to  $\Gamma_l$ , at which time the hypothesis is an  $F$  hypothesis and (6A.6) is satisfied for  $\mathbf{u}_l$ . The final thresholds on successive hypotheses of  $\mathbf{u}_l$  are reduced in increments of  $\Delta$  as before.

*Proof of Theorem 6.9.1 (Part c).* Let node  $\mathbf{u}_l$  be  $F$  hypothesized with final threshold  $T_l$ . We wish to show that, before  $\mathbf{u}_l$  can be rehypothesized, every descendant of  $\mathbf{u}_l$  for which the path from  $\mathbf{u}_l$  is above  $T_l$  must be  $F$  hypothesized with final threshold  $T_l$ . In the proof of part (b), we showed that for each immediate descendant of  $\mathbf{u}_l$ , say  $\mathbf{u}_{l+1}$ , if the path from  $\mathbf{u}_l$  was above  $T_l$  (that is, if  $\Gamma_{l+1} \geq T_l$ ), then  $\mathbf{u}_{l-1}$  was  $F$  hypothesized with a final threshold  $T$  before the first rehypothesis of  $\mathbf{u}_l$ . The theorem follows by induction on the path length of descendants from  $\mathbf{u}_l$ . That is, if a descendant  $\mathbf{u}_{l+j}$ , at path length  $j$  from  $\mathbf{u}_l$ , is  $F$  hypothesized at a final threshold  $T_l$ , then each immediate descendant of  $\mathbf{u}_{l+j}$  for which  $\Gamma_{l+j+1} \geq T_l$  is  $F$  hypothesized at a final threshold  $T_l$  before a rehypothesis of  $\mathbf{u}_{l+j}$  and thus before a rehypothesis of  $\mathbf{u}_l$ . That the threshold cannot be lowered below  $T_l$  until  $\mathbf{u}_l$  is rehypothesized follows from (6.9.6). This completes the proof. |

## APPENDIX 6B

In this appendix, we want to find an upper bound to  $\Pr[\Gamma'_{m(l)} > \Gamma_{\min} + \alpha]$  where  $\alpha$  is an arbitrary constant. The random variable  $\Gamma_{\min}$  is  $\inf_{n \geq 0} \Gamma_n$  where  $\Gamma_0, \Gamma_1, \Gamma_2, \dots$  are the values on the correct path in the received value tree. In terms of the

transmitted and received sequences on the channel,  $\Gamma_0 = 0$ , and for  $n > 0$ ,

$$\Gamma_n = \sum_{i=1}^n \gamma_i \quad (6B.1)$$

$$\gamma_i = \sum_{a=1}^v \left[ \ln \frac{P(y_i^{(a)} | x_i^{(a)})}{\omega(y_i^{(a)})} - B \right] \quad (6B.2)$$

In the ensemble of codes, the  $x_n^{(a)}$  are independent selections from the input alphabet with the probability assignment  $Q(k)$ . The  $y_n^{(a)}$  are statistically related to the  $x_n^{(a)}$  by the channel transition probabilities  $P(j|k)$ , and it can be seen that the  $\gamma_i$  in (6B.2) are statistically independent random variables. The random variable  $\Gamma'_{m(l)}$  is the value of some given node at depth  $l$  in the received value tree, given by

$$\Gamma'_{m(l)} = \sum_{i=1}^l \gamma'_i \quad (6B.3)$$

where

$$\gamma'_i = \sum_{a=1}^v \left[ \ln \frac{P(y_i^{(a)} | x_i'^{(a)})}{\omega(y_i^{(a)})} - B \right] \quad (6B.4)$$

and  $\mathbf{x}' = x_1'^{(1)}, \dots, x_1'^{(v)}, x_2'^{(1)}, \dots$  is the code sequence corresponding to node  $m(l)$ . By assumption, the path  $m(l)$  corresponds to a source sequence differing from the transmitted noise sequence in the first subblock. Over the ensemble of codes, the  $x_n'^{(a)}$  are statistically independent of the  $x_n^{(a)}$  in (6B.2) and are also statistically independent of each other. Thus the  $\gamma'_i$  in (6B.4) are also statistically independent random variables. Since  $y_n^{(a)}$  in (6B.4) is the same as  $y_n^{(a)}$  in (6B.2), it should be observed that  $\gamma_i$  and  $\gamma'_i$  are not in general statistically independent (see Problem 6.43).

For  $n > l$ , define

$$\Gamma_{n,l} = \sum_{i=l+1}^n \gamma_i$$

and for  $n = l$ , define  $\Gamma_{n,l} = 0$ . We then have  $\Gamma_n = \Gamma_l + \Gamma_{n,l}$  for  $n \geq l$ . Let

$$\min \Gamma_{n,l} = \inf_{n \geq l} \Gamma_{n,l}$$

The event that  $\Gamma'_{m(l)} \geq \Gamma_{\min} + \alpha$  is the union of the events for  $0 \leq n \leq l-1$  that  $\Gamma'_{m(l)} \geq \Gamma_n + \alpha$  and the event that  $\Gamma'_{m(l)} \geq \Gamma_l + \min \Gamma_{n,l} + \alpha$ . We thus have

$$\begin{aligned} \Pr[\Gamma'_{m(l)} \geq \Gamma_{\min} + \alpha] &\leq \sum_{n=0}^{l-1} \Pr[\Gamma'_{m(l)} \geq \Gamma_n + \alpha] \\ &\quad + \Pr[\Gamma'_{m(l)} \geq \Gamma_l + \min \Gamma_{n,l} + \alpha] \end{aligned} \quad (6B.5)$$

We now upper bound each term in the sum on the right-hand side of (6B.5) by the Chernoff bound, using (6B.1) and (6B.3) for  $\Gamma'_{m(l)}$  and  $\Gamma_n$ .

$$\Pr[\Gamma'_{m(l)} \geq \Gamma_n + \alpha] \leq \overline{\exp \left\{ s \left[ \sum_{i=1}^l \gamma'_i - \sum_{i=1}^n \gamma_i - \alpha \right] \right\}} \quad (6B.6)$$

for any  $s \geq 0$ . Using the statistical independence of the  $\gamma_i'$  and  $\gamma_i$  between different values of  $i$ , this can be rewritten as

$$\Pr[\Gamma'_{m(l)} \geq \Gamma_n + \alpha] \leq e^{-s\alpha} \prod_{i=1}^n \overline{\exp [\gamma_i' - \gamma_i]} \prod_{i=n+1}^l \exp(s\gamma_i') \quad (6B.7)$$

It is convenient to choose  $s = \frac{1}{2}$ , and using (6B.2) and (6B.4) we have

$$\begin{aligned} \overline{\exp [\gamma_i' - \gamma_i]} &= \sum \prod_{a=1}^v Q(x_i^{(a)}) P(y_i^{(a)} | x_i^{(a)}) Q(x_i'^{(a)}) \\ &\quad \times \left[ \frac{P(y_i^{(a)} | x_i'^{(a)})}{\omega(y_i^{(a)})} \right]^{\frac{1}{2}} \left[ \frac{P(y_i^{(a)} | x_i^{(a)})}{\omega(y_i^{(a)})} \right]^{-\frac{1}{2}} \end{aligned} \quad (6B.8)$$

where the sum is over all choices for each  $x_i^{(a)}$ ,  $x_i'^{(a)}$ ,  $y_i^{(a)}$ . Summing separately for each  $a$ ,  $1 \leq a \leq v$  [as in (5.5.7) to (5.5.10)], this becomes

$$\begin{aligned} \overline{\exp [\gamma_i' - \gamma_i]} &= \left\{ \sum_{j=0}^{J-1} \left[ \sum_{k=0}^{K-1} Q(k) \sqrt{P(j|k)} \right] \left[ \sum_{k'=0}^{K-1} Q(k') \sqrt{P(j|k')} \right] \right\}^v \\ &= \left\{ \sum_{j=0}^{J-1} \left[ \sum_{k=0}^{K-1} Q(k) \sqrt{P(j|k)} \right]^2 \right\}^v = \exp[-vE_0(1, \mathbf{Q})] \end{aligned} \quad (6B.9)$$

Likewise, we have

$$\overline{\exp(\gamma_i')} = \sum \prod_{a=1}^v \left\{ Q(x_i^{(a)}) P(y_i^{(a)} | x_i^{(a)}) Q(x_i'^{(a)}) \left[ \frac{P(y_i^{(a)} | x_i'^{(a)})}{\omega(y_i^{(a)})} \right]^{\frac{1}{2}} e^{-B/2} \right\} \quad (6B.10)$$

Recalling that  $\omega(y_i^{(a)}) = \sum Q(x_i^{(a)}) P(y_i^{(a)} | x_i^{(a)})$ , this becomes

$$\overline{\exp(\gamma_i')} = \left[ \sum_{j=0}^{J-1} \sum_{k=0}^{K-1} Q(k) \omega(j)^{\frac{1}{2}} P(j|k)^{\frac{1}{2}} e^{-B/2} \right]^v \quad (6B.11)$$

Using the Cauchy inequality on the sum over  $j$ , we obtain

$$\begin{aligned} \overline{\exp(\gamma_i')} &\leq \left[ \sqrt{\sum_j \omega(j)} \sqrt{\sum_j \left[ \sum_k Q(k) \sqrt{P(j|k)} \right]^2} e^{-B/2} \right]^v \\ &= \exp \left\{ -\frac{v}{2} [E_0(1, \mathbf{Q}) + B] \right\} \end{aligned} \quad (6B.12)$$

With the restriction  $B \leq E_0(1, \mathbf{Q})$ , we can also upper bound  $\overline{\exp[\gamma_i' - \gamma_i]}$  in (6B.9) by  $\exp\{-(v/2)[E_0(1, \mathbf{Q}) + B]\}$ . Substituting this and (6B.12) into (6B.7) with  $s = \frac{1}{2}$ , we have, for  $n < l$ ,

$$\Pr[\Gamma'_{m(l)} \geq \Gamma_n + \alpha] \leq \exp \left\{ -\frac{\alpha}{2} - \frac{vl}{2} [E_0(1, \mathbf{Q}) + B] \right\} \quad (6B.13)$$

We next must find an upper bound to the final term in (6B.5),  $\Pr[\Gamma'_{m(l)} \geq \Gamma_l + \min \Gamma_{n,l} + \alpha]$ . Recall that

$$\Gamma_{n,l} = \sum_{i=l+1}^n \gamma_i$$

for  $n > l$  and the  $\gamma_i$  are independent identically distributed random variables. The sequence  $\Gamma_{n,l}$  for  $n = l, l+1, \dots$  is called a *random walk*. The following lemma is a well-known result in the theory of random walks\* and is frequently useful in information theory. We shall state and use the result here and then prove it in the next section of this appendix.

LEMMA 6B.1. Let  $z_1, z_2, \dots$  be a sequence of statistically independent identically distributed discrete random variables. Let  $S_0 = 0$  and for each  $n > 0$ , let

$$S_n = \sum_{i=1}^n z_i$$

Let

$$S_{\min} = \inf_{n \geq 0} S_n$$

Then for any  $r \leq 0$  such that  $\overline{\exp(rz_i)} \leq 1$ , and for any  $u$ ,

$$\Pr[S_{\min} \leq u] \leq e^{-ru} \quad (6B.14)$$

For the application here, we take  $z_i = \gamma_{i+l}$ ,  $i \geq 1$ , so that  $S_{\min} = \min \Gamma_{n,l}$ . We now show that  $\overline{\exp(-\frac{1}{2}\gamma_i)} \leq 1$ , from which it will follow from (6B.14) that

$$\Pr[\min \Gamma_{n,l} \leq u] \leq e^{u/2} \quad (6B.15)$$

As in (6B.10) to (6B.12), we have

$$\begin{aligned} \overline{\exp(-\frac{1}{2}\gamma_i)} &= \sum \prod_{a=1}^v \left\{ Q(x_i^{(a)}) P(y_i^{(a)} | x_i^{(a)}) \left[ \frac{P(y_i^{(a)} | x_i^{(a)})}{\omega(y_i^{(a)})} \right]^{-\frac{1}{2}} e^{B/2} \right\} \\ &= \left[ \sum_{k=0}^{K-1} \sum_{j=0}^{J-1} Q(k) \sqrt{\omega(j) P(j | k)} e^{B/2} \right]^v \\ &\leq \exp \left\{ -\frac{v}{2} [E_o(1, \mathbf{Q}) - B] \right\} \end{aligned} \quad (6B.16)$$

In the last step, we have used the Cauchy inequality in the same way as in (6B.11) to (6B.12). Since by assumption  $B \leq E_o(1, \mathbf{Q})$ ,  $\overline{\exp(-\frac{1}{2}\gamma_i)} \leq 1$ , and (6B.15) is valid.

The random variable  $\Gamma_l' - \Gamma_l - \alpha$  is statistically independent of the random variable  $\min \Gamma_{n,l}$ , and we thus have

$$\Pr[\Gamma_l' - \Gamma_l - \alpha - \min \Gamma_{n,l} \geq 0] = \sum_u \Pr[\Gamma_l' - \Gamma_l - \alpha = u] \Pr[\min \Gamma_{n,l} \leq u] \quad (6B.17)$$

where the sum over  $u$  is over all discrete values taken by the random variables

\* See, for example, Cox and Miller (1966) p. 60.

$\Gamma_l' - \Gamma_l - \alpha$ . Upper bounding (6B.17) by (6B.15), we have

$$\Pr[\Gamma_l' - \Gamma_l - \alpha - \min \Gamma_{n,l} \geq 0] \leq \sum_u \Pr[\Gamma_l' - \Gamma_l - \alpha = u] e^{u/2}$$

$$= \overline{\exp [\frac{1}{2}(\Gamma_l' - \Gamma_l - \alpha)]} \quad (6B.18)$$

$$= \overline{\exp \left\{ \frac{1}{2} \left[ \sum_{i=1}^l (\gamma_i' - \gamma_i) - \alpha \right] \right\}} \quad (6B.19)$$

$$= \exp \left( -\frac{\alpha}{2} \right) \overline{\left[ \exp \left( \frac{\gamma_i' - \gamma_i}{2} \right) \right]^l} \quad (6B.20)$$

$$= \exp \left[ -\frac{\alpha}{2} - \frac{\nu l}{2} E_o(1, \mathbf{Q}) \right] \quad (6B.21)$$

$$\leq \exp \left\{ -\frac{\alpha}{2} - \frac{\nu l}{2} [E_o(1, \mathbf{Q}) + B] \right\} \quad (6B.22)$$

In (6B.20) we have used the statistical independence of  $\gamma_i' - \gamma_i$  between different values of  $i$ , in (6B.21) we have used (6B.9), and in (6B.22) we have used the assumption that  $B \leq E_o(1, \mathbf{Q})$ . Finally, substituting (6B.22) and (6B.13) into (6B.5), we have

$$\Pr[\Gamma_{m(l)}' \geq \Gamma_{\min} + \alpha] \leq (l+1) \exp \left\{ -\frac{\alpha}{2} - \frac{\nu l}{2} [E_o(1, \mathbf{Q}) + B] \right\} \quad (6B.23)$$

completing the proof of Lemma 6.9.3. |

### Random Walks and the Proof of Lemma 6B.1

A sequence of random variables  $S_0 = 0, S_1, S_2, \dots$  is called a *random walk* if, for each integer  $n > 0$ , we can express  $S_n$  by

$$\sum_{i=1}^n z_i$$

where  $z_1, z_2, \dots$  is a sequence of independent identically distributed random variables. Let  $g(r) = \overline{\exp(rz_i)}$  be the moment-generating function of each of the  $z_i$ , and let  $P(z)$  denote the probability assignment on the random variables. For simplicity of notation, we assume that the  $z_i$  are discrete random variables, but the results easily extend to arbitrary random variables for which  $g(r)$  exists in an interval around  $r = 0$ .

For any given  $r$ , define the tilted probability assignment  $Q_r(z)$  by

$$Q_r(z) = \frac{P(z)e^{rz}}{g(r)} \quad (6B.24)$$

Let  $z_{i,r}; i = 1, 2, \dots$  be a sequence of statistically independent random variables, each with the probability assignment  $Q_r(z)$ , and consider the “tilted” random walk with  $S_{0,r} = 0$  and

$$S_{n,r} = \sum_{i=1}^n z_{i,r}$$

for  $n > 0$ .

We are interested in the probability that this tilted random walk drops below some point  $u < 0$  for the first time at a given integer  $n$ , and define the probability  $f_{r,n}(u,v)$ , for  $v \leq u$ , by

$$f_{r,n}(u,v) = \Pr[S_{l,r} > u; 1 \leq l \leq n-1; S_{n,r} = v] \quad (6B.25)$$

Observe that

$$\sum_{v \leq u}^{\infty} f_{r,n}(u,v)$$

is the probability that the tilted walk achieves a value less than or equal to  $u$  for the first time at the point  $n$ . Also,

$$\sum_{n=1}^{\infty} \sum_{v \leq u} f_{r,n}(u,v)$$

is the probability that the tilted random walk ever achieves a value less than or equal to  $u$ . Finally, for  $r = 0$ , these probabilities apply to the original random walk.

Now, suppose that  $a_1, \dots, a_n$  are a sequence of values that can be taken on by the random variables  $z_1, \dots, z_n$ . From (6B.24), we have

$$\Pr[z_{1,r} = a_1, \dots, z_{n,r} = a_n] = \frac{\Pr[z_1 = a_1, \dots, z_n = a_n] \exp\left(r \sum_{i=1}^n a_i\right)}{[g(r)]^n} \quad (6B.26)$$

Consequently

$$\Pr\left[S_{1,r} = a_1, \dots, S_{n,r} = \sum_{i=1}^n a_i\right] = \frac{\Pr\left[S_1 = a_1, \dots, S_n = \sum_{i=1}^n a_i\right] \exp\left(r \sum_{i=1}^n a_i\right)}{[g(r)]^n} \quad (6B.27)$$

If we sum (6B.27) over all sequences  $a_1, \dots, a_n$  such that

$$\sum_{i=1}^l a_i > u$$

for  $1 < l < n-1$  and

$$\sum_{i=1}^n a_i = v$$

we find that

$$f_{r,n}(u,v) = \frac{f_{0,n}(u,v)e^{rv}}{[g(r)]^n} \quad (6B.28)$$

Now, suppose that  $r$  is chosen so that  $\overline{z_{i,r}} < 0$ . Then using the law of large numbers,

$$\lim_{n \rightarrow \infty} \Pr[S_{n,r} \leq u] = 1$$

and the tilted random walk eventually achieves a value less than or equal to any  $u < 0$  with probability 1. Thus summing both sides of (6B.28) over  $v \leq u$  and over  $n$ , we obtain

$$1 = \sum_{n=1}^{\infty} \sum_{v \leq u} f_{0,n}(u,v)e^{rv}[g(r)]^{-n} \quad (6B.29)$$

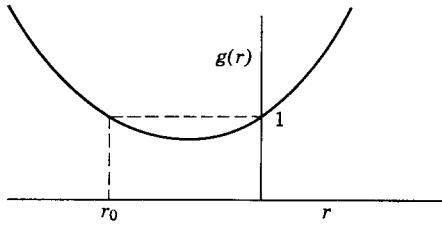
This result is known as Wald's equality for a single barrier at  $u$  (it is called a barrier since, in random walk theory, one often terminates the walk after it first crosses a given point).

We want to use this result here to upper bound

$$\Pr[S_{\min} \leq u] = \sum_{n=1}^{\infty} \sum_{v \leq u} f_{0,n}(u,v)$$

We assume that  $\bar{z}_i > 0$ , for otherwise  $\Pr[S_{\min} \leq u] = 1$ . We also assume that  $z_i$  takes on at least one negative value with nonzero probability, for otherwise  $\Pr[S_{\min} \leq u] = 0$ . Then  $g(r)$  is as sketched in Figure 6B.1. In particular,  $g(r)$  is convex  $\cup$  and approaches  $\infty$  both as  $r \rightarrow \infty$  and as  $r \rightarrow -\infty$ . Thus the equation  $g(r) = 1$  has two solutions, one for  $r = 0$ , and one for  $r = r_0$  where  $r_0 < 0$ . Since

$$\bar{z}_{i,r} = \frac{1}{g(r)} \frac{dg(r)}{dr}$$



**Figure 6B.1.** Sketch of  $g(r) = \exp(rz_i)$  for  $\bar{z}_i > 0$  where  $z_i$  takes on both positive and negative values.

we see that  $\bar{z}_{i,r} < 0$  for  $r = r_0$  and (6B.29) applies, yielding

$$1 = \sum_{n=1}^{\infty} \sum_{v \leq u} f_{0,n}(u,v) e^{r_0 v} \quad (6B.30)$$

$$\geq e^{r_0 u} \sum_{n=1}^{\infty} \sum_{v \leq u} f_{0,n}(u,v) \quad (6B.31)$$

$$\Pr[S_{\min} \leq u] \leq e^{-r_0 u} \quad (6B.32)$$

Since  $g(r) \leq 1$  only for  $r_0 \leq r \leq 0$ , we can further upper bound (6B.30) by  $e^{-ru}$  for any  $r \leq 0$  such that  $g(r) \leq 1$ . Finally, observing that  $\Pr[S_{\min} \leq u] \leq \exp(-ru)$  for  $u \geq 0$ , also, we have completed the proof of Lemma 6B.1. The bound in (6B.32) is quite tight. To see this, suppose that there is a minimum value, say  $z_{\min}$ , that can be taken on by the random variable  $z$ . Then  $v$  in (6B.30) must satisfy  $u - z_{\min} < v \leq u$ , and thus we can also lower bound  $\Pr[S_{\min} \leq u]$  by

$$\Pr[S_{\min} \leq u] \geq \exp[-r_0(u - z_{\min})] \quad (6B.33)$$

It can also be shown [see Feller (1966) Vol. II, p. 393] that asymptotically for large  $u$ ,  $\Pr[S_{\min} \leq u] \sim Ce^{-r_0 u}$  where  $C$  is independent of  $u$ .

## *Chapter 7*

### MEMORYLESS CHANNELS WITH DISCRETE TIME

#### 7.1 Introduction

In Chapters 4 and 5 we proved the coding theorem and its converse for discrete memoryless channels. Those channels were discrete in the sense that the channel input and output were time sequences of letters selected from finite alphabets. In this chapter, we shall generalize the results of Chapters 4 and 5 to situations where the input and output alphabets are infinite. The simplest and most important example of this generalization is the situation where the input and output alphabets each consist of the set of real numbers and the channel is described statistically by a conditional probability density  $p_{Y|X}(y | x)$ . We still assume that the channel is memoryless in the sense that if  $\mathbf{x} = (x_1, \dots, x_N)$  is a sequence of  $N$  inputs, then the corresponding output sequence  $\mathbf{y} = (y_1, \dots, y_N)$  has a joint conditional probability density given by

$$p_N(\mathbf{y} | \mathbf{x}) = \prod_{n=1}^N p_{Y|X}(y_n | x_n) \quad (7.1.1)$$

In other words, each output letter depends probabilistically only on the corresponding input and this probabilistic dependence is independent of time (that is, position in the sequence).

In general, a memoryless channel with discrete time is specified by an arbitrary input space  $X$ , an arbitrary output space  $Y$ , and for each element  $x$  in the input space, a conditional probability measure\* on the output  $P_{Y|X}$ . The channel input is a sequence of letters from the input space, the channel output is a sequence of letters from the output space, and each output letter

\* To be strictly correct, we must also define a class of output events, closed under complementation and under finite and countable unions and intersections. For each  $x$  in the input space,  $P_{Y|X}$  must be a probability measure on this class of output events.

depends probabilistically only on the corresponding input letter with the given probability measure  $P_{Y|X}$  (that is, as in Chapter 4, given  $x_n$ ,  $y_n$  is conditionally independent of all other inputs and outputs).

Our general approach to the treatment of these channels will be to restrict ourselves to the use of a finite set of letters in the input space, say  $a_1, a_2, \dots, a_K$ , and to partition the output space into a finite set of disjoint events, say  $B_1, \dots, B_J$ , whose union is the entire output space. Then, in principle, we can construct a “quantizer” for which the input each unit of time is the channel output  $y$  and for which the output is that event  $B_j$  that contains  $y$ . The combination of the channel and quantizer is then a discrete memoryless channel with transition probabilities  $P_{Y|X}(B_j | a_k)$ . We shall treat the original channel by considering the behavior of all such derived discrete memoryless channels. This approach has the advantages of being closely related to the way a channel is used physically and of being easy to treat analytically.

There is one new type of problem that will arise in the study of these channels and that is the problem of input constraints. Consider the channel of Example 4 in Chapter 2 in which the channel output is the sum of the input and an independent Gaussian random variable,

$$p_{Y|X}(y | x) = \frac{1}{\sqrt{2\pi}\sigma} \exp \left[ -\frac{(y-x)^2}{2\sigma^2} \right] \quad (7.1.2)$$

It was shown in (2.4.36) that, if the input is a Gaussian random variable of variance  $\mathcal{E}$ , then

$$I(X; Y) = \frac{1}{2} \log \left[ 1 + \frac{\mathcal{E}}{\sigma^2} \right] \quad (7.1.3)$$

By letting  $\mathcal{E}$  be arbitrarily large,  $I(X; Y)$  becomes arbitrarily large, and by choosing an arbitrarily large set of inputs spaced arbitrarily far apart in amplitude, we see that arbitrarily high transmission rates can be achieved with arbitrarily low error probability with essentially no coding. If we consider these inputs as samples of a transmitted waveform, however, we see that we are achieving this result through the expenditure of an arbitrarily large amount of power. For this, and a large class of related examples, we can obtain physically meaningful and mathematically interesting results only by constraining the channel input.

From a conceptual standpoint, the simplest type of constraint to impose on a channel input is an amplitude constraint; the input alphabet is simply limited to values of  $x$  less than or equal to some fixed value  $A$ . If the input space is defined as the interval from  $-A$  to  $+A$ , then the constraint can be otherwise ignored. A more common and important type of constraint is an energy constraint. This type of constraint will be formulated precisely later, but roughly it constrains the channel input to have a mean square value of,

at most, some fixed value  $\mathcal{E}$ . This type of constraint does not affect the input space but rather the relative frequency with which different inputs can be used. We shall see, in the next chapter, that an energy constraint is the natural constraint to use in representing a continuous time channel as a parallel combination of discrete time channels.

In the following sections, we first analyze channels with unconstrained (or amplitude constrained) inputs, then channels with constrained inputs, and then develop several examples, including the important example of additive Gaussian noise.

## 7.2 Unconstrained Inputs

We have seen that a general, discrete-time, memoryless channel can be used as a discrete memoryless channel by restricting the input to a finite set of letters and by partitioning the output. Thus any error probability that can be achieved through coding on any such discrete memoryless channel can be achieved on the general channel, using it as the appropriate discrete channel.

For a given discrete-time memoryless channel, let  $X_d$  represent a finite set of channel input letters ( $a_1, \dots, a_K$ ) and a probability assignment  $Q(a_1), \dots, Q(a_K)$ . Let  $Y_p$  represent a partition of the channel output into events  $B_1, \dots, B_J$ . The joint ensemble  $X_d Y_p$  has the joint probability assignment  $Q(a_k)P_{Y|X}(B_j | a_k)$  and the average mutual information (in nats):

$$I(X_d; Y_p) = \sum_{k=1}^K \sum_{j=1}^J Q(a_k) P_{Y|X}(B_j | a_k) \ln \frac{P_{Y|X}(B_j | a_k)}{\sum_{i=1}^K Q(a_i) P_{Y|X}(B_j | a_i)} \quad (7.2.1)$$

Define the function  $E_o(\rho, X_d, Y_p)$  by

$$E_o(\rho, X_d, Y_p) = -\ln \sum_{j=1}^J \left[ \sum_{k=1}^K Q(a_k) P_{Y|X}(B_j | a_k)^{1/(1+\rho)} \right]^{1+\rho} \quad (7.2.2)$$

From Theorem 5.6.2 we know that a code of block length  $N$  with  $M = e^{NR}$  code words exists for which the error probability on the channel with transition probabilities  $P_{Y|X}(B_j | a_k)$  satisfies  $P_e \leq \exp\{-N[E_o(\rho, X_d, Y_p) - \rho R]\}$  for all  $\rho$ ,  $0 \leq \rho \leq 1$ . This leads us to define the random-coding exponent for the given discrete-time memoryless channel as

$$E_r(R) = \sup [E_o(\rho, X_d, Y_p) - \rho R] \quad (7.2.3)$$

The supremum is over all finite selections of input letters, all probability assignments on the input letters, all partitions of the output space, and all  $\rho$ ,  $0 \leq \rho \leq 1$ . We similarly define the capacity (in nats) of the channel as

$$C = \sup I(X_d; Y_p) \quad (7.2.4)$$

where the supremum is defined as above.

**Theorem 7.2.1 (Coding Theorem).** Let  $E_r(R)$  and  $C$  be defined by (7.2.3) and (7.2.4) for a discrete-time memoryless channel. For any  $R \geq 0$ ,  $N \geq 1$ , and  $E < E_r(R)$ , there exists a code of block length  $N$  with  $M = [e^{NR}]$  code words for which

$$P_e \leq \exp(-NE) \quad (7.2.5)$$

Here

$$P_e = \sum_{m=1}^M \Pr(m) P_{e,m}$$

is the average error probability, where  $\Pr(m)$  is the probability of transmitting the  $m$ th code word and  $P_{e,m}$  is the error probability for the  $m$ th code word. Furthermore

$$E_r(R) > 0; \quad \text{all } R, 0 \leq R < C \quad (7.2.6)$$


---

(Comments. Equation 7.2.5 states that, for given  $N, R$ , we can either find codes for which  $P_e \leq \exp[-NE_r(R)]$ , or at least find codes for which  $P_e$  is as close as we wish to  $\exp[-NE_r(R)]$ . An alternate statement is that for any given  $N, R$ , we have  $\inf P_e \leq \exp[-NE_r(R)]$  where the infimum is over all codes of the given  $N, R$ .)

*Proof.* For a given  $N, R$ , and  $E < E_r(R)$ , choose  $\rho, X_d$ , and  $Y_p$  so that

$$E_o(\rho, X_d, Y_p) - \rho R \geq E \quad (7.2.7)$$

This is always possible since  $E$  is strictly less than the supremum of the left-hand side of (7.2.7) over  $\rho, X_d, Y_p$ . From Theorem 5.6.2 there exists a code of the given  $N$  and  $R$  for the discrete channel corresponding to  $X_d, Y_p$  for which

$$P_e \leq \exp\{-N[E_o(\rho, X_d, Y_p) - \rho R]\} \quad (7.2.8)$$

Since this error probability can also be achieved on the general discrete-time channel, we get (7.2.5) by combining (7.2.7) and (7.2.8). To establish (7.2.6), assume  $R < C$ , pick a number  $R_1$ ,  $R < R_1 < C$ , and choose  $X_d, Y_p$  to satisfy

$$I(X_d; Y_p) \geq R_1 \quad (7.2.9)$$

From Theorem 5.6.4, the random-coding exponent for the channel corresponding to  $X_d, Y_p$  is positive for  $R < R_1$ , and thus  $E_r(R)$  is also positive for the given  $R$ . |

The use of  $E < E_r(R)$  in (7.2.5) rather than  $E_r(R)$  is a confusing irritant, but the following example (Figure 7.2.1) shows that it cannot be avoided. It is easy to see that any code, when used on this channel, will have a nonzero probability of decoding error. On the other hand, by using only the inputs  $L < k \leq L + K$  for sufficiently large  $L$  and  $K$ , the error probability can be

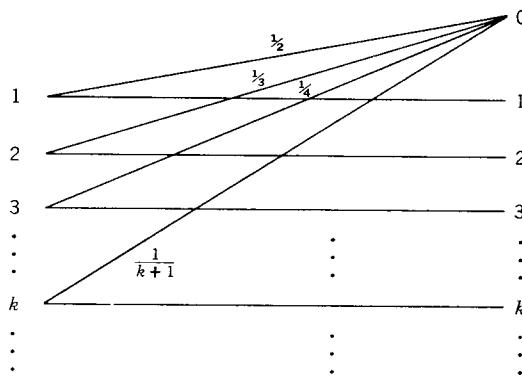


Figure 7.2.1. Infinite alphabet erasure channel.

reduced to any desired level greater than 0 for any  $N$  and  $R$ . Finally, by partitioning the output space, using one set for each  $j$ ,  $L < j \leq L + K$ , and one set for all other outputs, it can be seen, by letting  $L$  and  $K$  approach  $\infty$ , that  $C = \infty$ ,  $E_r(R) = \infty$ . Thus we cannot achieve  $P_e \leq \exp[-NE_r(R)]$  for this example, although we can achieve (7.2.5) for any finite  $E$ .

**Theorem 7.2.2 (Converse to Coding Theorem).** Let a discrete stationary source with an alphabet size of  $M$  have entropy  $H_\infty(U)$  and generate letters one each  $\tau_s$  seconds. Let a discrete-time memoryless channel have capacity  $C$  and be used once each  $\tau_c$  seconds. Let a source sequence of length  $L$  be connected to a destination through a sequence of  $N$  channel uses where  $N$  is  $[L\tau_s/\tau_c]$ . Then in the limit as  $L \rightarrow \infty$ , the error probability per source digit,  $\langle P_e \rangle$ , satisfies

$$\langle P_e \rangle \log(M-1) + \mathcal{H}(\langle P_e \rangle) \geq H_\infty(U) - \frac{\tau_s}{\tau_c} C \quad (7.2.10)$$

*Proof.* This is the same theorem as Theorem 4.3.4, but it applies to a more general class of channels. The proof of Theorem 4.3.4 involved the channel only insofar as establishing the following two relations:

$$I(\mathbf{U}^L; \mathbf{V}^L) \leq I(\mathbf{X}^N; \mathbf{Y}^N) \quad (7.2.11)$$

$$I(\mathbf{X}^N; \mathbf{Y}^N) \leq NC \quad (7.2.12)$$

Thus it suffices here to establish these two relations for the discrete-time memoryless channel.

*Proof of (7.2.11).* For any given value of  $L$  and any given source and encoder, the number of code words will be finite. Thus only a finite set of channel input letters can be used in the corresponding ensemble  $\mathbf{X}^N$ , and

$\mathbf{X}^N$  is thus a discrete ensemble. Likewise, any given decoder will partition the space  $\mathbf{Y}^N$  into at most  $M^L$  decoding regions, corresponding to the elements on the space  $\mathbf{V}^L$ . Denoting the partitioned output space as  $\mathbf{Y}_p^N$ , Theorem 4.3.3 establishes that

$$I(\mathbf{U}^L; \mathbf{V}^L) \leq I(\mathbf{X}^N; \mathbf{Y}_p^N) \quad (7.2.13)$$

Next, by definition (2.5.1),

$$I(\mathbf{X}^N; \mathbf{Y}^N) = \sup I(\mathbf{X}^N; \mathbf{Y}_p^N) \quad (7.2.14)$$

where the supremum is over all partitions of  $\mathbf{Y}^N$ . Combining (7.2.13) and (7.2.14), we have (7.2.11).

*Proof of (7.2.12).* We can rewrite  $I(\mathbf{X}^N; \mathbf{Y}^N)$  as\*

$$I(\mathbf{X}^N; \mathbf{Y}^N) = I(X_1 \cdots X_N; Y_1 \cdots Y_N) \quad (7.2.15)$$

Applying (2.2.29) repeatedly to the right-hand side of (7.2.15), and noting that all terms are finite, we have

$$I(\mathbf{X}^N; \mathbf{Y}^N) = \sum_{n=1}^N I(\mathbf{X}^N; Y_n | Y_1 \cdots Y_{n-1}) \quad (7.2.16)$$

Using (2.5.4) and (2.5.5)

$$\begin{aligned} I(\mathbf{X}^N; Y_n | Y_1 \cdots Y_{n-1}) &= I(Y_n; \mathbf{X}^N Y_1 \cdots Y_{n-1}) - I(Y_n; Y_1 \cdots Y_{n-1}) \\ &\leq I(Y_n; \mathbf{X}^N Y_1 \cdots Y_{n-1}) \end{aligned} \quad (7.2.17)$$

$$I(Y_n; \mathbf{X}^N Y_1 \cdots Y_{n-1}) = I(Y_n; X_n) + I(Y_n; Y_1 \cdots Y_{n-1} X_1 \cdots X_{n-1} X_{n+1} \cdots X_N | X_n) \quad (7.2.18)$$

The last term of (7.2.18) is zero by Theorem 2.3.3, and combining equations, we have

$$I(\mathbf{X}^N; \mathbf{Y}^N) \leq \sum_{n=1}^N I(X_n; Y_n) \quad (7.2.19)$$

Since  $X_n$  is discrete,  $I(X_n; Y_n)$  is defined as a supremum over all partitions of  $Y_n$ , and by the definition of  $C$  in (7.2.4), we have  $I(X_n; Y_n) \leq C$ , and (7.2.12) follows immediately. |

The previous results, while establishing a coding theorem and its converse in great generality, give little indication as to how to calculate either  $E_r(R)$  or  $C$ . We have already seen, in Section 2.5, that  $I(X_d; Y_p)$  is nondecreasing as the partitioning of the  $Y$  space is made increasingly fine. We now establish the same result for  $E_o(\rho, X_d, Y_p)$ . Let  $Y_p$  be a partition with the events  $B_1, \dots, B_J$  and let  $Y_{p'}$  be a subpartition of  $Y_p$  with the events  $B_{ij}$  where

$$\bigcup_i B_{ij} = B_j$$

\* This is not a trivial substitution; see the discussion following (2.5.3).

From the Minkowski inequality (Problem 4.15h), for  $\rho \geq 0$ , we have

$$\begin{aligned} \sum_i \left[ \sum_k Q(a_k) P_{Y|X}(B_{ij} | a_k)^{1/(1+\rho)} \right]^{1+\rho} &\leq \left\{ \sum_k Q(a_k) \left[ \sum_i P_{Y|X}(B_{ij} | a_k) \right]^{1/(1+\rho)} \right\}^{1+\rho} \\ &= \left[ \sum_k Q(a_k) P_{Y|X}(B_j | a_k)^{1/(1+\rho)} \right]^{1+\rho} \end{aligned} \quad (7.2.20)$$

Summing both sides of (7.2.20) over  $j$  and taking minus the logarithm of the result, we obtain

$$E_o(\rho, X_d, Y_p) \geq E_o(\rho, X_d, Y_p) \quad (7.2.21)$$

Physically, this result is not surprising. The more finely the channel output is partitioned, the more information about the received sequence is made available to the decoder, and the smaller the probability of decoding error will be.

If the channel output space is the real line and the channel is described by a probability density  $p_{Y|X}(y | x)$ , then we can partition the output space into intervals, subdivide the intervals more and more finely, and in the limit go to

$$E_o(\rho, X_d, Y) \triangleq -\ln \int_{-\infty}^{\infty} dy \left[ \sum_k Q(a_k) p_{Y|X}(y | a_k)^{1/(1+\rho)} \right]^{1+\rho} \quad (7.2.22)$$

That the right-hand side of (7.2.22) is indeed the supremum of  $E_o(\rho, X_d, Y_p)$  over all partitions of  $Y$  follows from the same argument as in (7.2.20), using the integral form of Minkowski's inequality,\*

$$\begin{aligned} \int_{y \in B_j} dy \left[ \sum_k Q(a_k) p_{Y|X}(y | a_k)^{1/(1+\rho)} \right]^{1+\rho} &\leq \left\{ \sum_k Q(a_k) \left[ \int_{y \in B_j} p_{Y|X}(y | a_k) \right]^{1/(1+\rho)} \right\}^{1+\rho} \\ &= \left\{ \sum_k Q(a_k) P_{Y|X}(B_j | a_k)^{1/(1+\rho)} \right\}^{1+\rho} \end{aligned} \quad (7.2.23)$$

Summing both sides over  $j$ , we obtain the desired result.

This reduces the problem of finding  $E_r(R)$  for a channel with a transition probability density to the problem of calculating

$$E_r(R) = \sup_{0 \leq \rho \leq 1} \sup_{X_d} [E_o(\rho, X_d, Y) - \rho R] \quad (7.2.24)$$

where  $E_o(\rho, X_d, Y)$  is given by (7.2.22). Very little can be said in general about finding the supremum of  $E_o(\rho, X_d, Y)$  over all discrete input assignments. In some cases (see Problem 7.5)  $E_o(\rho, X_d, Y)$  is maximized by a discrete input assignment, and in that case the conditions (5.6.37) and (5.6.38), generalized to a continuous output, verify that a maximum has been achieved. In other cases (see Problem 7.4), the supremum over  $E_o(\rho, X_d, Y)$  is achieved in the

\* See Hardy, Littlewood, and Polya, Theorem 201.

limit as  $X_d$  approaches a probability density on the input space; and this optimum probability density can sometimes be found\* through the use of (5.6.37) and (5.6.38). One final peculiarity of the function  $E_o(\rho, X_d, Y)$  is brought out in problem 7.3, where it is shown that

$$\sup_{X_d} E_o(\rho, X_d, Y)$$

can be discontinuous in  $\rho$  at  $\rho = 0$ . This leads to a channel with an infinite capacity but a finite random-coding exponent.

The expurgated random-coding bound of Section 5.7 extends to discrete-time memoryless channels in the same way as the random-coding bound.

Make the following definitions:

$$E_x(\rho, X_d, Y_p)$$

$$= -\rho \ln \sum_{k,i} Q(a_k)Q(a_i) \left[ \sum_j \sqrt{P_{Y|X}(B_j | a_k)P_{Y|X}(B_j | a_i)} \right]^{1/\rho} \quad (7.2.25)$$

$$E_{ex}(R') = \sup [-\rho R' + E_x(\rho, X_d, Y_p)] \quad (7.2.26)$$

where the supremum is over  $\rho \geq 1$ ,  $X_d$ , and  $Y_p$ . Then, for any  $R' \geq 0$ ,  $N \geq 1$ , and  $E < E_{ex}(R')$ , there is a code of block length  $N$  with  $M = \lfloor \frac{1}{4}e^{NR'} \rfloor$  code words for which

$$P_{e,m} \leq \exp(-NE); \quad \text{all } m, 1 \leq m \leq M \quad (7.2.27)$$

The proof is the same as the proof for Theorem 7.2.1.

### 7.3 Constrained Inputs

In Section 7.1 we described an energy constraint as a constraint which limited the mean square value of a channel input. Here we shall consider a somewhat more general problem. Let  $X$  denote the input space to a discrete-time memoryless channel and let  $f(x)$  be a real-valued function on the input letters. We shall constrain the channel to be used in such a way that the expected value of  $f(x)$  is less than or equal to some fixed value  $\mathcal{E}$ . If  $X$  is the set of real numbers and  $f(x) = x^2$ , then this is an energy constraint as described above. More generally, for example,  $X$  could be a class of functions,  $x(t)$ , and  $f(x)$ , could be  $\int x^2(t) dt$ , or for that matter, any other functional of  $x(t)$ .

From a coding point of view, we must be precise about what is meant by constraining the expected value of  $f(x)$ . One reasonable interpretation is to insist that each code word satisfy the constraint; that is, for each code word  $\mathbf{x}_m = (x_{m,1}, \dots, x_{m,N})$  we insist that

$$\sum_{n=1}^N f(x_{n,m}) \leq N\mathcal{E}$$

\* In this case, the sufficiency of (5.6.37) and (5.6.38) still holds, although the proof in Chapters 4 and 5 is no longer valid.

Another reasonable interpretation is to assign a probability measure to the messages,  $\Pr(m)$ , and insist that

$$\sum_{m=1}^M \Pr(m) \sum_{n=1}^N f(x_{n,m}) \leq N\mathcal{E}$$

Observe that the class of codes for which each code word satisfies the constraint is included in the class of codes for which the average over the code words satisfies the constraint. Thus any probability of decoding error that can be achieved for some code in the former class can also be achieved for a code (that is, the same code) in the latter class. Conversely, any lower bound on error probability for the latter class also applies to the former class. For this reason, we shall prove the coding theorem constraining *each* code word and we shall prove the converse constraining only the average over the set of code words. In this way, each theorem will apply to both cases, and we shall have shown that it makes no fundamental difference which of the two cases is considered. We shall begin with the converse to the coding theorem since it is almost the same as in the unconstrained case.

Using the notation of the last section, the capacity of a discrete-time, memoryless channel with an input constraint  $\overline{f(x)} \leq \mathcal{E}$  is defined to be

$$C = \sup I(X_a; Y_p) \quad (7.3.1)$$

where the supremum is over all partitions of the output space, over all discrete sets of inputs  $(a_1, \dots, a_K)$ , and all probability assignments  $Q(a_1), \dots, Q(a_K)$  for which the constraint

$$\sum_{k=1}^K Q(a_k) f(a_k) \leq \mathcal{E} \quad (7.3.2)$$

is satisfied. Notice that in this definition, the function  $f(x)$  and the constraining value  $\mathcal{E}$  are considered to be an integral part of the description of the channel.

**Theorem 7.3.1 (Converse to Coding Theorem).** Let a discrete stationary source with an alphabet size  $M$  have entropy  $H_\infty(U)$  and generate one letter each  $\tau_s$  seconds. Let a discrete-time, memoryless channel with an input constraint  $\overline{f(x)} \leq \mathcal{E}$  have capacity  $C$  as defined by (7.3.1). Let a source sequence of length  $L$  be connected to a destination through a sequence of  $N$  channel uses,  $x_1, \dots, x_N$ , where  $N$  is  $\lfloor L\tau_s/\tau_c \rfloor$ . Let the resulting channel input ensemble  $\mathbf{X}^N$  satisfy the constraint in the sense that

$$\sum_{n=1}^N \overline{f(x_n)} \leq N\mathcal{E} \quad (7.3.3)$$

Then in the limit as  $L \rightarrow \infty$ , the error probability per source digit,  $\langle P_e \rangle$ , satisfies

$$\langle P_e \rangle \log(M-1) + \mathcal{H}(\langle P_e \rangle) \geq H_\infty(U) - \frac{\tau_s}{\tau_c} C \quad (7.3.4)$$


---

(Comments. Notice that the restriction (7.3.3) asserts that the constraint is satisfied by the average over the code words. We are allowing individual code words to violate the constraint and also allowing individual letters in the sequence of  $N$  uses to violate the constraint. In other words, as applied to an energy constraint, the encoder is allowed to apportion the available energy for the block in any desired way between the  $N$  channel uses.)

*Proof.* The proof of Theorem 7.2.2 applies here insofar as establishing that  $I(\mathbf{U}^L; \mathbf{V}^L) \leq I(\mathbf{X}^N; \mathbf{Y}^N)$  and that

$$I(\mathbf{X}^N; \mathbf{Y}^N) \leq \sum_{n=1}^N I(X_n; Y_n)$$

We completed the proof of Theorem 7.2.2 by showing that  $I(X_n; Y_n) \leq C$  for each  $n$ . That result is not valid here since individual uses of the channel need not satisfy the constraint. To complete the proof here, we must show that

$$\sum_{n=1}^N I(X_n; Y_n) \leq NC \quad (7.3.5)$$

For any particular code, let  $a_1, \dots, a_K$  be the set of channel inputs that are used and let  $Q_n(a_k)$  be the probability of using input letter  $a_k$  on the  $n$ th channel use. From (7.3.3), we have

$$\sum_{n=1}^N \sum_{k=1}^K Q_n(a_k) f(a_k) \leq N\mathcal{E} \quad (7.3.6)$$

Define the probability assignment  $Q(a_k)$  by

$$Q(a_k) = \frac{1}{N} \sum_{n=1}^N Q_n(a_k) \quad (7.3.7)$$

Substituting (7.3.7) into (7.3.6), we have

$$\sum_{k=1}^K Q(a_k) f(a_k) \leq \mathcal{E} \quad (7.3.8)$$

Let  $I(X; Y)$  be the average mutual information on the channel using letters  $a_1, \dots, a_K$  with probabilities  $Q(a_1), \dots, Q(a_K)$ . Since (7.3.8) is equivalent to (7.3.2), we have

$$I(X; Y) \leq C \quad (7.3.9)$$

In Theorem 4.4.2, it was shown that the average mutual information on a discrete input channel is a convex  $\cap$  function of the input probability assignment.\* From (4.4.5) (taking  $\theta_n$  as  $1/N$ ), it follows that

$$\sum_n \frac{1}{N} I(X_n; Y_n) \leq I(X; Y) \quad (7.3.10)$$

Combining (7.3.9) and (7.3.10), we have (7.3.5), completing the proof. |

\* The statement of Theorem 4.4.2 was restricted to channels with a discrete output, but it will be observed that the proof applies equally to an arbitrary output.

Before stating and proving a coding theorem for discrete-time memoryless channels with an input constraint, we must consider the effect of an input constraint on a discrete memoryless channel. As in Chapter 5, let  $P(j|k)$  denote the transition probabilities for a discrete memoryless channel with input alphabet  $0, \dots, K - 1$  and output alphabet  $0, \dots, J - 1$ . Let  $f(k)$  be a function of the input and consider the class of codes in which each code word  $\mathbf{x} = (x_1, \dots, x_N)$  is constrained to satisfy

$$\sum_{n=1}^N f(x_n) \leq N\mathcal{E} \quad (7.3.11)$$

for a given constant  $\mathcal{E}$ .

We shall now construct an ensemble of codes in which each code word satisfies (7.3.11). Let  $Q(k)$  be a probability assignment on the channel input letters satisfying

$$\sum_k Q(k)f(k) \leq \mathcal{E} \quad (7.3.12)$$

Let  $Q_N(\mathbf{x})$  be a probability assignment on sequences of  $N$  channel inputs given by

$$Q_N(\mathbf{x}) = \mu^{-1}\varphi(\mathbf{x}) \prod_{n=1}^N Q(x_n) \quad (7.3.13)$$

where

$$\varphi(\mathbf{x}) = \begin{cases} 1; & \text{for } N\mathcal{E} - \delta < \sum_n f(x_n) \leq N\mathcal{E} \\ 0; & \text{otherwise} \end{cases} \quad (7.3.14)$$

$$\mu = \sum_{\mathbf{x}} \varphi(\mathbf{x}) \prod_{n=1}^N Q(X_n) \quad (7.3.15)$$

and  $\delta$  is an arbitrary positive number to be specified later. It can be seen that  $Q_N(\mathbf{x})$  is the conditional probability of sequence  $\mathbf{x}$ , conditional on  $N\mathcal{E} - \delta < \sum f(x_n) \leq N\mathcal{E}$ , if the letters of  $\mathbf{x}$  are chosen independently with the probability assignment  $Q(k)$ . The quantity  $\mu$  is the probability that such an independently chosen sequence satisfies  $N\mathcal{E} - \delta < \sum f(x_n) \leq N\mathcal{E}$ .

Consider an ensemble of codes with  $M$  code words of block length  $N$  in which the code words are independently chosen with the probability assignment  $Q_N(\mathbf{x})$ . From Theorem 5.6.1, the average error probability for each message,  $1 \leq m \leq M$ , over the ensemble of codes, is upper bounded for all  $\rho$ ,  $0 \leq \rho \leq 1$ , by

$$P_{e,m} \leq (M-1)^\rho \sum_{\mathbf{y}} \left[ \sum_{\mathbf{x}} Q_N(\mathbf{x}) P_N(\mathbf{y} | \mathbf{x})^{1/(1-\rho)} \right]^{1+\rho} \quad (7.3.16)$$

Equation 7.3.16 is not in a very useful form since, for large  $N$ , the sums are unmanageable. We can obtain a more useful result by upper bounding

$Q_N(\mathbf{x})$  appropriately and simplifying the resultant expression. For any  $r \geq 0$ , we can upper bound  $\varphi(\mathbf{x})$  in (7.3.14) by

$$\varphi(\mathbf{x}) \leq \exp \{r[\sum f(x_n) - N\mathcal{E} + \delta]\} \quad (7.3.17)$$

Equation 7.3.17 is clearly valid when  $\varphi(\mathbf{x}) = 0$ . When  $\varphi(\mathbf{x}) = 1$ , the term in brackets in (7.3.17) is nonnegative, and the right-hand side of (7.3.17) is greater than or equal to 1. Combining (7.3.13) and (7.3.17), we have, for any  $r \geq 0$ ,

$$Q_N(\mathbf{x}) \leq \mu^{-1} e^{r\delta} \prod_{n=1}^N Q(x_n) e^{r[f(x_n) - \mathcal{E}]} \quad (7.3.18)$$

Bounding  $Q_N(\mathbf{x})$  in (7.3.16) by (7.3.18), and repeating the steps from (5.6.11) to (5.6.13), we obtain

$$P_{e,m} \leq \left[ \frac{e^{r\delta}}{\mu} \right]^{1+\rho} \exp \{-N[E_o(\rho, \mathbf{Q}, r) - \rho R]\} \quad (7.3.19)$$

$$E_o(\rho, \mathbf{Q}, r) = -\ln \sum_j \left( \sum_k Q(k) e^{r[f(k) - \mathcal{E}]} P(j | k)^{1/(1+\rho)} \right)^{1+\rho} \quad (7.3.20)$$

where  $M$  and  $R$  are related by  $M = [e^{Nr}]$ . Applying the argument in Corollary 2 of Theorem 5.6.2, we see that there also exists a code for which, for each  $m$ ,  $1 \leq m \leq M$ , and each  $\rho$ ,  $0 \leq \rho \leq 1$ ,

$$P_{e,m} \leq \left[ \frac{2e^{r\delta}}{\mu} \right]^2 \exp \{-N[E_o(\rho, \mathbf{Q}, r) - \rho R]\} \quad (7.3.21)$$

The above bound is given in terms of a number of arbitrary parameters, namely  $0 \leq \rho \leq 1$ ,  $r \geq 0$ ,  $\mathbf{Q}$ , and  $\delta \geq 0$ . Before attempting any optimization over these parameters, it is helpful to investigate the behavior of the bound when we choose  $r = 0$  and  $\delta$  arbitrarily large. In this case, (7.3.19) simplifies to

$$P_{e,m} \leq \left( \frac{1}{\mu} \right)^{1+\rho} \exp \{-N[E_o(\rho, \mathbf{Q}) - \rho R]\} \quad (7.3.22)$$

where  $E_o(\rho, \mathbf{Q}) - \rho R$  is the familiar exponent in Theorem 5.6.2, and aside from the factor  $(1/\mu)^{1+\rho}$ , (7.3.22) is equivalent to Theorem 5.6.2. Since  $\mu$  is now the probability that an  $\mathbf{x}$  sequence, chosen independently with the probability assignment  $Q(k)$ , satisfies

$$\sum_{n=1}^N f(x_n) \leq N\mathcal{E}$$

we see from the central limit theorem that  $\mu$  tends to  $1/2$  with increasing  $N$  if

$$\sum_k Q(k)f(k) = \mathcal{E}$$

and  $\mu$  tends to 1 if  $\sum Q(k)f(k) < \mathcal{E}$ . Thus the factor  $(1/\mu)^{1+\rho}$  does not affect the exponential dependence on  $N$  of the bound.

The capacity of this channel in nats is given, as a special case of (7.3.1), by

$$C = \max \sum_k \sum_j Q(k)P(j \mid k) \ln \frac{P(j \mid k)}{\sum_i Q(i)P(j \mid i)} \quad (7.3.23)$$

where the maximum is over all probability assignments  $Q(k)$  satisfying (7.3.12). For the  $\mathbf{Q}$  that maximizes (7.3.23) subject to (7.3.12),

$$\max_{0 \leq \rho \leq 1} E_o(\rho, \mathbf{Q}) - \rho R$$

is positive for  $R < C$ , by the same argument as used in Theorem 5.6.4. In summary, we have seen that for  $r = 0$ ,  $\delta$  arbitrarily large, and  $\rho$  and  $\mathbf{Q}$  appropriately chosen,  $P_{e,m}$ , as given by (7.3.21), is exponentially decreasing in  $N$  for any  $R < C$ .

As a byproduct of the above considerations, notice that if  $E_o(\rho, \mathbf{Q})$  is maximized over all probability vectors  $\mathbf{Q}$  by a  $\mathbf{Q}$  satisfying the constraint  $\sum Q(k)F(k) \leq \mathcal{E}$ , then we can achieve the same exponent for the constrained channel as for the unconstrained channel.

We now turn to the more interesting situation in which for a given  $\rho$ ,  $E_o(\rho, \mathbf{Q})$  is maximized by a  $\mathbf{Q}$  that violates the constraint. In this case, it turns out that  $E_o(\rho, \mathbf{Q}, r)$  is maximized, subject to the constraints, by a  $\mathbf{Q}$  satisfying  $\sum f(k)Q(k) = \mathcal{E}$  and by  $r > 0$ . Roughly, the reason for this is as follows. Suppose that we used an ensemble of codes in which all code-word letters were chosen independently with  $\sum Q(k)f(k) = \mathcal{E}$ . The sum

$$\sum_{n=1}^N f(x_n)$$

for most code words would be close to  $N\mathcal{E}$ . However, the few code words for which  $\sum f(x_n)$  is substantially less than  $N\mathcal{E}$  would be representative of an ensemble with a smaller value of  $E_o(\rho, \mathbf{Q})$  and thus a much higher error probability. The error probability due to these few words dominates the bound on error probability with  $r = 0$  [actually, these words are not in the ensemble, but they appear in the bound because of the bound on  $Q_N(\mathbf{x})$  in (7.3.18)]. The purpose of choosing  $r > 0$  is to reduce the effect of these few poor words on the bound.

The easiest way to maximize  $E_o(\rho, \mathbf{Q}, r)$  over  $r$  and  $\mathbf{Q}$  is to look for a stationary point with respect to  $r$  and  $\mathbf{Q}$  subject to the constraints

$$\sum_k Q(k) = 1 \quad \text{and} \quad \sum_k Q(k)[f(k) - \mathcal{E}] = 0.$$

Using  $\lambda$  and  $\gamma$  as Lagrange multipliers, we want to find a stationary point of the function

$$\sum_j \left\{ \sum_k Q(k) e^{r[f(k)-\mathcal{E}]} P(j|k)^{1/(1+\rho)} \right\}^{1+\rho} + \lambda \sum_k Q(k) + \gamma \sum_k Q(k)[f(k) - \mathcal{E}] \quad (7.3.24)$$

Taking partial derivatives with respect to each  $Q(k)$ , we obtain the conditions

$$(1 + \rho) \sum_j \alpha_j^\rho e^{r[f(k)-\mathcal{E}]} P(j|k)^{1/(1+\rho)} + \lambda + \gamma[f(k) - \mathcal{E}] \geq 0 \quad (7.3.25)$$

with equality for  $Q(k) > 0$  where  $\alpha_j$  is defined by

$$\alpha_j = \sum_k Q(k) e^{r[f(k)-\mathcal{E}]} P(j|k)^{1/(1+\rho)} \quad (7.3.26)$$

The inequality in (7.3.25) accounts for maxima of  $E_o(\rho, \mathbf{Q}, r)$  on the boundary where some of the  $Q(k) = 0$  in the same way as in Theorem 4.4.1. Taking the partial derivative of (7.3.24) with respect to  $r$ , we obtain the condition

$$(1 + \rho) \sum_j \alpha_j^\rho \sum_k Q(k)[f(k) - \mathcal{E}] e^{r[f(k)-\mathcal{E}]} P(j|k)^{1/(1+\rho)} = 0 \quad (7.3.27)$$

Multiplying (7.3.25) by  $Q(k)$  and summing over  $k$ , we find that

$$\lambda = -(1 + \rho) \sum_j \alpha_j^{1+\rho}$$

Likewise, multiplying (7.3.25) by  $Q(k)[f(k) - \mathcal{E}]$ , summing over  $k$ , and comparing with (7.3.27), we find that  $\gamma = 0$ . Thus (7.3.25) becomes

$$\sum_j \alpha_j^\rho e^{r[f(k)-\mathcal{E}]} P(j|k)^{1/(1+\rho)} \geq \sum_j \alpha_j^{1+\rho} \quad (7.3.28)$$

for all  $k$  with equality if  $Q(k) > 0$ . Although the above derivation does not prove it, it can be shown that (7.3.28) is a necessary and sufficient set of conditions on  $r$  and on the constrained  $\mathbf{Q}$  that maximizes  $E_o(\rho, \mathbf{Q}, r)$ . More important, it can be shown that the resulting exponent gives the true reliability function of the constrained channel for rates between  $R_{cr}$  and  $C$  where  $R_{cr}$  is defined as in Section 5.8.\*

The quantity  $\mu$  in (7.3.19) is difficult to bound neatly but it can be estimated closely for large  $N$ . Assume that  $\mathbf{Q}$  satisfies  $\sum Q(k)f(k) = \mathcal{E}$  and interpret

$$\sum_{n=1}^N f(x_n)$$

\* To prove this, start with the lower bound to error probability for fixed composition codes given in Shannon, Gallager, and Berlekamp (1967), Equation 4.16. After optimizing over the compositions satisfying the constraint, proceed as in their Theorem 6.

as a sum of independent random variables chosen according to the probability measure  $\mathbf{Q}$ . Then  $\mu$  is the probability that this sum is between its mean and  $\delta$  less than the mean. From the central limit theorem, for fixed  $\delta$ ,\*

$$\lim_{N \rightarrow \infty} \sqrt{N} \mu = \frac{\delta}{\sqrt{2\pi} \sigma_f} \quad (7.3.29)$$

$$\sigma_f^2 = \sum Q(k)[f(k) - \mathcal{E}]^2 \quad (7.3.30)$$

It can be seen from (7.3.29) that  $[e^{r\delta}/\mu]^{1+\rho}$  grows with  $N$  as  $N^{(1+\rho)/2}$  for fixed  $r$  and  $\delta$ , and thus this coefficient does not affect the exponential dependence of the bound on  $N$ .

The expurgated bound on error probability of Section 5.7 can be modified for an input constraint in the same way as the random-coding bound. We simply take (5.7.7), which is valid for any ensemble of codes, and upper bound  $Q_N(\mathbf{x})$  and  $Q_N(\mathbf{x}')$  by (7.3.18). Expanding the products, we find that for all  $M \geq 2$ ,  $N \geq 1$ , there exists a code of block length  $N$  with  $M$  code words, each code word  $\mathbf{x}_m$  satisfying the constraint

$$\sum_{n=1}^N f(x_{n,m}) \leq N\mathcal{E}$$

and also satisfying the bound

$$P_{e,m} \leq -N[E_x(\rho, \mathbf{Q}, r) - \rho R'] \quad (7.3.31)$$

where

$$E_x(\rho, \mathbf{Q}, r) = -\rho \ln \left\{ \sum_{k=0}^{K-1} \sum_{i=0}^{K-1} Q(k)Q(i) e^{r[f(k)+f(i)-2\mathcal{E}]} \right. \\ \times \left. \left( \sum_{j=0}^{J-1} \sqrt{P(j \mid k)P(j \mid i)} \right)^{1/\rho} \right\} \quad (7.3.32)$$

$$R' = \frac{\ln M}{N} + \frac{2}{N} \ln \frac{2e^{r\delta}}{\mu} \quad (7.3.33)$$

In the above  $\rho \geq 1$ ,  $r \geq 0$ , and  $\delta > 0$  are arbitrary and  $\mu$  is given by (7.3.15) and estimated by (7.3.29).

We can now apply these results to an arbitrary discrete-time, memoryless channel with an input constraint  $\overline{f(x)} \leq \mathcal{E}$ . As in Section 7.2, let  $X_d$  represent a finite set of channel input letters,  $a_1, \dots, a_K$ , with a probability assignment  $Q(a_1), \dots, Q(a_K)$  satisfying the constraint  $\sum Q(a_k)f(a_k) \leq \mathcal{E}$ . Let  $Y_p$

\* For a nonlattice distribution, (7.3.29) follows from Theorem 1, p. 512 of Feller (1966), Vol. 2. For a lattice distribution, (7.3.29) is only valid when  $\delta$  is a multiple of the span; it follows from Theorem 3, p. 490 of Feller (a lattice random variable  $f$  is a random variable which can be scaled to an integer-valued random variable  $z$  by the transformation  $f = zh + a$ ; the span is the largest  $h$  for which this can be done).

represent a partition of the output space into events  $B_1, \dots, B_J$ . Let  $E_o(\rho, X_d, Y_p, r)$  and  $E_x(\rho, X_d, Y_p, r)$  be given by

$$E_o(\rho, X_d, Y_p, r) = -\ln \sum_{j=1}^J \left[ \sum_{k=1}^K Q(a_k) e^{r[f(a_k) - \delta]} P_{Y|X}(B_j | a_k)^{1/(1+\rho)} \right]^{1+\rho} \quad (7.3.34)$$

$$\begin{aligned} E_x(\rho, X_d, Y_p, r) = & -\rho \ln \left\{ \sum_{k=1}^K \sum_{i=1}^K Q(a_k) Q(a_i) e^{r[f(a_k) + f(a_i) - 2\delta]} \right. \\ & \times \left. \left( \sum_{j=1}^J \sqrt{P_{Y|X}(B_j | a_k)} P_{Y|X}(B_j | a_i) \right)^{1/\rho} \right\} \end{aligned} \quad (7.3.35)$$

Define the random-coding exponent and expurgated exponent for the channel as

$$E_r(R) = \sup [E_o(\rho, X_d, Y_p, r) - \rho R] \quad (7.3.36)$$

$$E_{ex}(R) = \sup [E_x(\rho, X_d, Y_p, r) - \rho R] \quad (7.3.37)$$

The supremum in both of the above equations is over all finite selections of input letters, all probability assignments satisfying the constraint, all output partitions, all  $r \geq 0$ , and all  $\rho$  satisfying  $0 \leq \rho \leq 1$  for (7.3.36) and  $\rho \geq 1$  for (7.3.37).

**Theorem 7.3.2 (Coding Theorem).** Let  $E_r(R)$ ,  $E_{ex}(R)$ , and  $C$  be defined by (7.3.36), (7.3.37), and (7.3.1) for an arbitrary discrete-time memoryless channel with an input constraint  $f(x) \leq \mathcal{E}$ . Let  $R \geq 0$  be arbitrary and let  $E < \max [E_r(R), E_{ex}(R)]$  be arbitrary. Then, for all sufficiently large  $N$ , there exists a code of block length  $N$  with  $M = \lceil e^{NR} \rceil$  code words  $\mathbf{x}_1, \dots, \mathbf{x}_M$ , each satisfying the constraint

$$\sum_n f(x_{m,n}) \leq N\mathcal{E},$$

and each satisfying

$$P_{e,m} \leq \exp(-NE) \quad (7.3.38)$$

Furthermore,  $E_r(R) > 0$  for all  $R$ ,  $0 \leq R < C$ .

---

*Proof.* Choose  $E_1$  to satisfy  $E < E_1 < \max [E_r(R), E_{ex}(R)]$ . If  $E_r(R) \geq E_{ex}(R)$ ; choose  $X_d, Y_p$ ,  $r \geq 0$ , and  $0 \leq \rho \leq 1$  so that

$$E_1 \leq E_o(\rho, X_d, Y_p, r) - \rho R$$

From (7.3.21), for any  $\delta > 0$ , and each  $N \geq 1$ , there exists a code for this  $X_d, Y_p$  with  $M = \lceil e^{NR} \rceil$  code words each satisfying the constraint and satisfying

$$P_{e,m} \leq \left[ \frac{e^{r\delta}}{\mu} \right]^{1+\rho} \exp(-NE_1); \quad 1 \leq m \leq M \quad (7.3.39)$$

Since  $[e^{r\delta}/\mu]^{1+\rho}$  grows as  $N^{(1+\rho)/2}$  for sufficiently large  $N$ , and since  $E < E_1$ , we have, for sufficiently large  $N$ ,

$$P_{e,m} \leq [e^{r\delta}/\mu]^{1+\rho} \exp(-NE_1) \leq \exp(-NE) \quad (7.3.40)$$

The argument is the same if  $E_{ex}(R) > E_r(R)$ , except that  $E_x$  is used in place of  $E_o$ . Next assume  $R < C$ , and choose  $X_d, Y_p$  such that  $R < I(X_d; Y_p) < C$ . We saw in the argument following (7.3.23) that, for  $r = 0$ ,

$$\max_{0 \leq \rho \leq 1} [E_o(\rho, X_d, Y_p, r) - \rho R] > 0$$

and thus  $E_r(R) > 0$ . |

Theorem 7.3.2 has great generality, but is often difficult to apply because of the difficulty in calculating the suprema used in defining  $E_r(R)$  and  $E_{ex}(R)$ . For channels defined by a transition probability density, both  $E_o(\rho, X_d, Y_p, r)$  and  $E_x(\rho, X_d, Y_p, r)$  are nondecreasing with finer partitioning of the  $Y$  space. This result follows by exactly the same argument as we used for channels with an unconstrained input. The supremum over  $Y_p$  is achieved by

$$E_o(\rho, X_d Y, r) = -\ln \left[ \int_y \left[ \sum_k Q(a_k) e^{r[f(a_k) - \delta]} p(y | a_k)^{1/(1+\rho)} \right]^{1+\rho} dy \right] \quad (7.3.41)$$

$$E_x(\rho, X_d Y, r) = -\rho \ln \left\{ \sum_k \sum_i Q(a_k) Q(a_i) e^{r[f(a_k) + f(a_i) - 2\delta]} \times \left[ \int_y \sqrt{p(y | a_k) p(y | a_i)} dy \right]^{1/\rho} \right\} \quad (7.3.42)$$

If the supremum of  $E_o$  and  $E_x$  over  $X_d$  is achieved in the limit by a distribution approaching a probability density  $q(x)$ , we can make the definition

$$E_o(\rho, X, Y, r) = -\ln \left[ \int_y \left[ \int_x q(x) e^{r[f(x) - \delta]} p(y | x)^{1/(1+\rho)} dx \right]^{1+\rho} dy \right] \quad (7.3.43)$$

$$E_x(\rho, X, Y, r) = -\rho \ln \int_x \int_{x'} q(x) q(x') e^{r[f(x) + f(x') - 2\delta]} \times \left[ \int_y \sqrt{p(y | x) p(y | x')} dy \right]^{1/\rho} dx dx' \quad (7.3.44)$$

The exponents  $E_r(R)$  and  $E_{ex}(R)$  can then be found directly from (7.3.43) and (7.3.44), maximizing only over  $\rho$  and  $r$ . In this case, we can get somewhat more precise results by rederiving Theorems 5.6.1 and 5.7.1 using densities in place of probabilities and integrals in place of sums. Simplifying the result as in (7.3.16) to (7.3.21), we find that a code exists for which each code

word both satisfies the constraint and the following bounds on error probability

$$P_{e,m} \leq [2e^{r\delta}/\mu]^2 \exp \{-N[E_o(\rho, X, Y, r) - \rho R]\}; \quad 0 \leq \rho \leq 1 \quad (7.3.45)$$

$$P_{e,m} \leq \exp \{-N[E_x(\rho, X, Y, r) - \rho R']\}; \quad \rho \geq 1 \quad (7.3.46)$$

where  $R'$  is given by (7.3.33),  $\delta > 0$  is arbitrary, and  $\mu$  is given approximately by (7.3.29), where  $\sigma_f^2 = \int q(x)[f(x) - \mathcal{E}]^2 dx$ . The restrictions on this derivation are that the densities and integrals exist, that  $\int q(x)f(x) dx = \mathcal{E}$ , and that  $\int_{-\infty}^{\infty} q(x)[f(x)]^3 dx$  is finite. It is clear that (7.3.45) and (7.3.46) are valid whether the input density  $q(x)$  maximizes  $E_0$  and  $E_x$  or not.

## 7.4 Additive Noise and Additive Gaussian Noise

In this section, we shall apply the general results derived in the last two sections to the important and simple special case of additive noise channels. An additive noise channel is a channel for which the input space is the set of real numbers (or real vectors) and the output is the sum of the input and a statistically independent random variable (or vector) called the noise.\* For simplicity, assume that the noise  $z$  has a probability density  $p_Z(z)$ . For a given input  $x$ , the output  $y$  occurs iff  $z = y - x$ , and since  $z$  is independent of  $x$ , the transition probability density for the channel is given by

$$p_{Y|X}(y | x) = p_Z(y - x) \quad (7.4.1)$$

Calculating average mutual informations and capacity for an additive noise channel is greatly simplified by the fact that the conditional entropy of output given input,  $H(Y | X)$ , is equal to the noise entropy  $H(Z)$  and thus independent of the input distribution. To see this, let  $p_X(x)$  be a probability density on the input. Using (7.4.1) in the definition of conditional entropy given by (2.4.25), we have

$$\begin{aligned} H(Y | X) &= - \iint_{-\infty}^{\infty} p_X(x)p_Z(y - x) \log p_Z(y - x) dy dx \\ &= - \int_{-\infty}^{\infty} p_X(x) \int_{z=-\infty}^{\infty} p_Z(z) \log p_Z(z) dz dx \\ &= \int p_X(x)H(Z) dx = H(Z) \end{aligned} \quad (7.4.2)$$

\* For the purist who is concerned with defining statistical independence in the absence of any probability measure on the input space, an additive noise channel can be defined as a channel that satisfies (7.4.1), that is, for which the transition probability measure is a function only of the difference  $y - x$ .

The same argument clearly applies for a discrete distribution on the input. The average mutual information between channel input and output thus is given by

$$I(X; Y) = H(Y) - H(Y | X) = H(Y) - H(Z) \quad (7.4.3)$$

In this expression,  $H(Y)$  depends upon the input distribution but  $H(Z)$  does not. Thus the problem of finding capacity for an additive noise channel revolves around maximizing  $H(Y)$  subject to the input constraints. The following two examples show how this can sometimes be done.

**Example.** First consider a noise with a flat probability density,  $p_Z(z) = \frac{1}{2}$  for  $-1 \leq z \leq 1$  and  $p_Z(z) = 0$  elsewhere. Suppose that the input is amplitude constrained to lie in the interval  $-1 \leq x \leq 1$ . Since the output  $y$  is  $x + z$ , the output is constrained to the interval  $-2 \leq y \leq 2$ . Our strategy will be to find  $p_Y(y)$  in this interval to maximize  $H(Y)$  and then try to find an input distribution that yields this maximizing  $p_Y(y)$ . One might reasonably guess that  $H(Y)$  is maximized by a uniform probability density, and we shall substantiate this by the use of variational calculus. Let

$$F[p(y)] = \int_{-2}^2 -p(y) \log p(y) dy + \lambda \int_{-2}^2 p(y) dy \quad (7.4.3a)$$

where  $\lambda$  is a Lagrange multiplier for the constraint  $\int p(y) dy = 1$ . The function  $p(y)$  will yield a stationary point if

$$\frac{\partial F[p(y) + \epsilon h(y)]}{\partial \epsilon} \Big|_{\epsilon=0}$$

is zero for all choices of  $h(y)$ ,

$$\frac{\partial F[p(y) + \epsilon h(y)]}{\partial \epsilon} \Big|_{\epsilon=0} = - \int_{-2}^2 h(y) [\log p(y) + \log e - \lambda] dy \quad (7.4.4)$$

Thus a stationary point is achieved if

$$\log p(y) + \log e - \lambda = 0; \quad -2 \leq y \leq 2 \quad (7.4.5)$$

This implies that  $p(y)$  is constant over the interval, or  $p(y) = \frac{1}{4}$ .

It can be seen that a discrete input distribution, choosing  $P_X(-1) = P_X(+1) = \frac{1}{2}$ , yields  $P_Y(y) = \frac{1}{4}$  for  $-2 < y \leq 2$ , and thus presumably achieves capacity. To rigorously prove that this distribution achieves capacity, choose any finite set of input letters,  $a_1, \dots, a_K$  between  $-1$  and  $+1$ , including  $-1$  and  $+1$ . With the above input distribution

$$\begin{aligned} I(x = a_k; Y) &= \int_{y=-2}^2 p_Z(y - a_k) \log \frac{p_Z(y - a_k)}{\frac{1}{4}} dy \\ &= 1 \text{ bit} \end{aligned}$$

Thus this distribution satisfies the necessary and sufficient conditions on capacity of Theorem 4.5.1. It can be similarly verified that this distribution satisfies the necessary and sufficient conditions on maximizing  $E_r(R)$ . This should not be surprising since using only the inputs  $x = -1$  and  $x = +1$  converts the channel into a noiseless binary channel,  $y > 0$  implying  $x = +1$  and  $y < 0$  implying  $x = -1$ . In Problem 7.5 this example is continued, considering an arbitrary amplitude constraint on  $x$ . It turns out that the uniform probability density on  $y$  cannot be achieved in general, although capacity is always achieved by a discrete input distribution.

### Additive Gaussian Noise with an Energy Constrained Input

Suppose, for a second example, that an additive noise channel has an energy constraint on the input,

$$\bar{x^2} \leq \mathcal{E} \quad (7.4.6)$$

Let the noise have a density  $p_Z(z)$  with a mean of zero and a variance  $\sigma^2$ . The assumption of  $\bar{z} = 0$  entails no loss of generality since it can always be achieved by shifting the zero reference on the  $z$  and  $y$  scales. We shall assume that  $p_Z(z)$  is Gaussian later, but it will be arbitrary for the time being. The mean square value of the channel output is limited to

$$\bar{y^2} = \overline{(x+z)^2} = \bar{x^2} + \bar{z^2} \leq \mathcal{E} + \sigma^2 \quad (7.4.7)$$

We now maximize  $H(Y)$  subject to a constraint on  $\bar{y^2}$  and then return to try to find an appropriate input distribution to yield the maximizing  $p_Y(y)$ . We can again use a calculus of variations approach, extending the limits to  $\infty$  in (7.4.3) and adding a second constraint,  $\gamma \int y^2 p(y) dy$ . This leads to the condition, analogous to (7.4.5),

$$\log p(y) + \log e - \lambda - \gamma y^2 = 0; \quad \text{all } y$$

A solution to this, satisfying the constraint of (7.4.7), is

$$p(y) = \frac{1}{\sqrt{2\pi(\mathcal{E} + \sigma^2)}} \exp\left[-\frac{y^2}{2(\mathcal{E} + \sigma^2)}\right] \quad (7.4.8)$$

**Theorem 7.4.1.** The maximum value of the entropy

$$H(Y) = - \int_{-\infty}^{\infty} p(y) \log p(y) dy$$

over all choices of probability density  $p(y)$  satisfying

$$\int_{-\infty}^{\infty} y^2 p(y) dy = A \quad (7.4.9)$$

E

is uniquely achieved by the Gaussian density

$$\varphi_A(y) = \frac{1}{\sqrt{2\pi A}} \exp\left(-\frac{y^2}{2A}\right) \quad (7.4.10)$$

and has the value

$$H(Y) = \frac{1}{2} \log(2\pi e A) \quad (7.4.11)$$


---

*Proof.* We could prove the theorem by extending the convexity arguments in Chapter 4 to show that the calculus of variations solution in (7.4.8) forms a unique maximum. The following proof, however, is somewhat simpler. Let  $p(y)$  be an arbitrary density satisfying (7.4.9), and let  $\varphi_A(y)$  be the Gaussian density of (7.4.10).

$$\begin{aligned} \int p(y) \log \frac{1}{\varphi_A(y)} dy &= \int p(y) \left[ \log \sqrt{2\pi A} + \frac{y^2}{2A} \log e \right] dy \\ &= \log \sqrt{2\pi A} + \frac{A}{2A} \log e = \frac{1}{2} \log(2\pi e A) \end{aligned} \quad (7.4.12)$$

Thus, using (7.4.12),

$$\begin{aligned} H(Y) - \frac{1}{2} \log(2\pi e A) &= \int p(y) \log \frac{\varphi_A(y)}{p(y)} dy \\ &\leq \log e \int p(y) \left[ \frac{\varphi_A(y)}{p(y)} - 1 \right] dy = 0 \end{aligned}$$

where we have used the inequality  $\log z \leq (z - 1) \log e$ . Equality is achieved here if and only if  $\varphi_A(y)/p(y) = 1$  for all  $y$ . |

Since  $H(Y)$  increases with  $A$ , it is clear that changing the constraint in the theorem to  $\int y^2 p(y) dy \leq A$  would not change the result. The remaining problem in finding capacity is to find an input probability density which yields a Gaussian output probability density. It is an unfortunate fact of life that, if the sum of two independent random variables is Gaussian, then each of the variables alone must be Gaussian.\* Fortunately, the situation of greatest interest is that in which the additive noise is Gaussian. In this case,  $H(Y)$  is maximized, and capacity is achieved by choosing  $x$  to be Gaussian. We have already evaluated  $I(X; Y)$  for this case in 2.4.36 and this proves the following theorem.

\* Cramer, H. *Random Variables and Probability Distributions*, Cambridge Tracts in Mathematics No. 36, Cambridge, England, 1937.

**Theorem 7.4.2.** Let a discrete-time, memoryless, additive noise channel have Gaussian noise with variance  $\sigma^2$  and let the input be constrained by  $\bar{x}^2 \leq \mathcal{E}$ . Then the capacity is

$$C = \frac{1}{2} \log \left( 1 + \frac{\mathcal{E}}{\sigma^2} \right) \quad (7.4.13)$$


---

For non-Gaussian additive noise, the calculation of capacity is a tedious and unrewarding task. We shall content ourselves with the bounds on capacity given in the following theorem; this theorem says, in effect, that for a given noise variance, Gaussian noise is the worst kind of additive noise from a capacity standpoint.

**Theorem 7.4.3.** Let a discrete memoryless channel have additive noise of variance  $\sigma^2$  and have the input constraint  $\bar{x}^2 \leq \mathcal{E}$ . Then

$$\frac{1}{2} \log [2\pi e(\mathcal{E} + \sigma^2)] - H(Z) \geq C \geq \frac{1}{2} \log \left( 1 + \frac{\mathcal{E}}{\sigma^2} \right) \quad (7.4.14)$$


---

*Proof.* The left-hand inequality follows from  $I(X;Y) = H(Y) - H(Z)$ , using  $\frac{1}{2} \log [2\pi e(\mathcal{E} + \sigma^2)]$  as an upper bound on  $H(Y)$ . To establish the right-hand inequality, we shall let

$$p_X(x) = \frac{1}{\sqrt{2\pi\mathcal{E}}} \exp \left( -\frac{x^2}{2\mathcal{E}} \right),$$

and show that the resulting average mutual information satisfies

$$I(X;Y) \geq \frac{1}{2} \log \left( 1 + \frac{\mathcal{E}}{\sigma^2} \right) \quad (7.4.15)$$

Since  $C$  is the supremum of  $I(X;Y)$  over all allowable input distributions, this will complete the proof. Let  $p_Z(z)$  be the density of the noise, let  $\varphi_{\sigma^2}(z)$  be the Gaussian density of variance  $\sigma^2$ , let  $p_Y(y)$  be the output density, and let  $\varphi_A(y)$  be the Gaussian density with variance  $A = \mathcal{E} + \sigma^2$ . We then have, as in (7.4.12):

$$\begin{aligned} & \iint p_X(x)p_Z(y-x) \log \frac{\varphi_{\sigma^2}(y-x)}{\varphi_A(y)} dy dx \\ &= \int p_Z(z) \log \varphi_{\sigma^2}(z) - \int p_Y(y) \log \varphi_A(y) \\ &= -\frac{1}{2} \log (2\pi e \sigma^2) + \frac{1}{2} \log (2\pi e A) = \frac{1}{2} \log \left( 1 + \frac{\mathcal{E}}{\sigma^2} \right) \quad (7.4.16) \end{aligned}$$

Then, using (7.4.16),

$$\begin{aligned} -I(X; Y) + \frac{1}{2} \log \left( 1 + \frac{\mathcal{E}}{\sigma^2} \right) \\ = \iint p_X(x) p_Z(y-x) \log \frac{p_Y(y) \varphi_{\sigma^2}(y-x)}{p_Z(y-x) \varphi_A(y)} dy dx \\ \leq \log e \left\{ \iint \frac{p_X(x) p_Y(y) \varphi_{\sigma^2}(y-x)}{\varphi_A(y)} dy dx - 1 \right\} \quad (7.4.17) \end{aligned}$$

But since  $p_X(x)$  is Gaussian,  $\int p_X(x) \varphi_{\sigma^2}(y-x) dx = \varphi_A(y)$ . Thus the double integral in (7.4.17) reduces to  $\int p_Y(y) dy = 1$ , and the right-hand side of (7.4.17) is zero, completing the proof. |

We next apply the error probability bounds (7.3.45) and (7.3.46) to the additive Gaussian noise channel described by the density

$$p(y|x) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp \left[ -\frac{(y-x)^2}{2\sigma^2} \right] \quad (7.4.18)$$

Each codeword,  $\mathbf{x}_m = (x_{m,1}, \dots, x_{m,N})$ , is constrained to satisfy

$$\sum_{n=1}^N x_{m,n}^2 \leq N\mathcal{E} \quad (7.4.19)$$

As an input density for our ensemble of codes, we choose

$$q(x) = \frac{1}{\sqrt{2\pi}} \exp \left( -\frac{x^2}{2\mathcal{E}} \right) \quad (7.4.20)$$

There are a number of reasons for choosing a Gaussian density here. First, it is easy to integrate; second, the resulting joint density on sequences has spherical symmetry; and, third, it turns out to yield a random-coding exponent which agrees over the nonstraight line portion with the exponent in a lower bound to error probability derived by Shannon (1959).

Substituting (7.4.18) and (7.4.20) into the expression for  $E_o(\rho, X, Y, r)$  in (7.3.43), and using  $x^2$  for  $f(x)$ , we can integrate by completing the square in the exponent. The result, replacing  $r$  with the variable  $s$ , is

$$E_o(\rho, X, Y, s) = s(1+\rho)\mathcal{E} + \frac{1}{2} \ln(1-2s\mathcal{E}) + \frac{\rho}{2} \ln \left[ 1 - 2s\mathcal{E} + \frac{\mathcal{E}}{(1+\rho)\sigma^2} \right] \quad (7.4.21)$$

This result is valid for  $0 \leq s < 1/(2\mathcal{E})$ . For larger  $s$ ,  $E_o = -\infty$ , which is useless. Instead of maximizing this expression over  $s$ , it is convenient to make the following substitutions:

$$A = \mathcal{E}/\sigma^2 \quad (7.4.22)$$

$$\beta = 1 - 2s\mathcal{E} + A/(1+\rho) \quad (7.4.23)$$

The quantity  $A$  is the signal to noise ratio on the channel, and by scaling we would expect our final bound to depend only on  $A$  rather than  $\mathcal{E}$  and  $\sigma^2$  separately. Eliminating  $\mathcal{E}$ ,  $\sigma^2$ , and  $s$  in (7.4.21) by the use of (7.4.22) and (7.4.23), we obtain an expression for  $E_o$  as a function of  $A$ ,  $\beta$ , and  $\rho$ .

$$\tilde{E}_o(A, \beta, \rho) = \frac{1}{2} \left[ (1 - \beta)(1 + \rho) + A + \ln \left( \beta - \frac{A}{1 + \rho} \right) + \rho \ln \beta \right] \quad (7.4.24)$$

The constraint  $0 \leq s < 1/(2\mathcal{E})$  appears here as

$$\frac{A}{1 + \rho} < \beta \leq 1 + \frac{A}{1 + \rho} \quad (7.4.25)$$

The function  $\tilde{E}_o$  has a stationary point with respect to  $\beta$  where

$$\frac{\partial \tilde{E}_o}{\partial \beta} = \frac{1}{2} \left[ -(1 + \rho) + \frac{1 + \rho}{\beta(1 + \rho) - A} + \frac{\rho}{\beta} \right] = 0 \quad (7.4.26)$$

The left-hand side of (7.4.26) is decreasing with  $\beta$  in the range given by (7.4.25), going from  $+\infty$  to a negative value. Thus  $\tilde{E}_o$  is maximized by the unique  $\beta$  in the given range that satisfies (7.4.26). This value is found by rearranging (7.4.26) and solving the resulting quadratic equation.

$$\beta^2 - \beta \left( 1 + \frac{A}{1 + \rho} \right) + \frac{A\rho}{(1 + \rho)^2} = 0 \quad (7.4.27)$$

$$\beta = \frac{1}{2} \left( 1 + \frac{A}{1 + \rho} \right) \left[ 1 + \sqrt{1 - \frac{4A\rho}{(1 + \rho + A)^2}} \right] \quad (7.4.28)$$

Next  $\tilde{E}_o - \rho R$  has a stationary point with respect to  $\rho$  where

$$\frac{\partial [\tilde{E}_o - \rho R]}{\partial \rho} = \frac{1}{2} \left[ 1 - \beta + \frac{\beta}{(1 + \rho)\beta - A} - \frac{1}{(1 + \rho)} + \ln \beta \right] - R = 0 \quad (7.4.29)$$

For the  $\beta$  satisfying (7.4.26), this reduces to

$$R = \frac{1}{2} \ln \beta \quad (7.4.30)$$

For the  $\beta$  and  $\rho$  satisfying (7.4.26) and (7.4.29), we then have

$$E_r(R) = \tilde{E}_o - \rho R = \frac{1}{2} \left[ (1 - \beta)(1 + \rho) + A + \ln \left( \beta - \frac{A}{1 + \rho} \right) \right] \quad (7.4.31)$$

We can now obtain an explicit expression for  $E_r(R)$  by solving (7.4.26) for  $1 + \rho$  in terms of  $\beta$  and  $A$ . This gives us

$$1 + \rho = \frac{A}{2\beta} \left[ 1 + \sqrt{1 + \frac{4\beta}{A(\beta - 1)}} \right] \quad (7.4.32)$$

Substituting (7.4.32) into (7.4.31) and simplifying the resulting expression somewhat, we obtain

$$E_r(R) = \frac{A}{4\beta} \left[ (\beta + 1) - (\beta - 1) \sqrt{1 + \frac{4\beta}{A(\beta - 1)}} \right] \\ + \frac{1}{2} \ln \left\{ \beta - \frac{A(\beta - 1)}{2} \left[ \sqrt{1 + \frac{4\beta}{A(\beta - 1)}} - 1 \right] \right\} \quad (7.4.33)$$

where from (7.4.30),  $\beta = e^{2R}$

Equation 7.4.33 is valid for  $0 \leq \rho \leq 1$ . Using the values of  $\beta$  in (7.4.28)

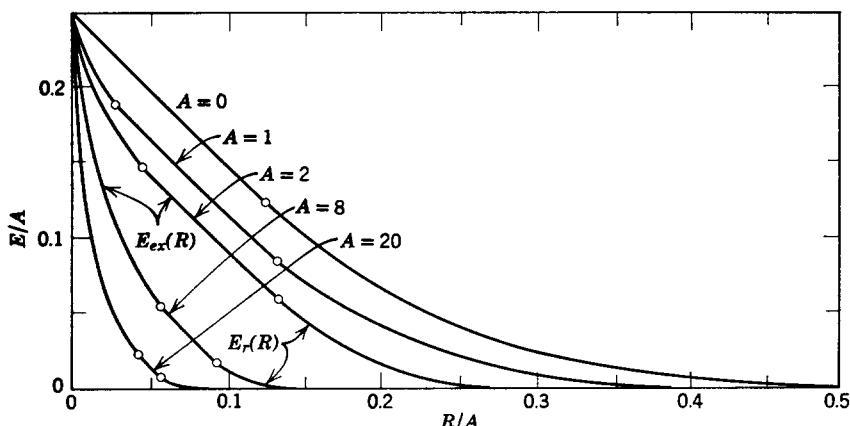


Figure 7.4.1.  $E_r(R)$  and  $E_{ex}(R)$  for discrete-time additive Gaussian noise channel for various values of signal to noise ratio  $A$ .

for  $\rho = 0$  and  $\rho = 1$  and substituting them in (7.4.30), we find that (7.4.33) is valid for

$$\frac{1}{2} \ln \left[ \frac{1}{2} + \frac{A}{4} + \frac{1}{2} \sqrt{1 + \frac{A^2}{4}} \right] \leq R \leq \frac{1}{2} \ln (1 + A) \quad (7.4.34)$$

For  $R$  less than the left-hand side of (7.4.34), we must choose  $\rho = 1$ , yielding

$$E_r(R) = 1 - \beta + \frac{A}{2} + \frac{1}{2} \ln \left( \beta - \frac{A}{2} \right) + \frac{1}{2} \ln \beta - R \quad (7.4.35)$$

where

$$\beta = \frac{1}{2} \left[ 1 + \frac{A}{2} + \sqrt{1 + \frac{A^2}{4}} \right] \quad (7.4.36)$$

The random-coding exponent,  $E_r(R)$  is sketched for several values of  $A$  in Figure 7.4.1.

We still must discuss the coefficient  $[2e^{s\delta}/\mu]^2$  in (7.3.45). We can solve for  $s$  from (7.4.23), obtaining

$$2s\mathcal{E} = 1 - \beta + A/(1 + \rho) \quad (7.4.37)$$

Multiplying numerator and denominator of the right-hand side by  $\beta$  and comparing with (7.4.27), we obtain an expression for  $s$  which will be useful later.

$$2s\mathcal{E} = \frac{\rho A}{(1 + \rho)^2 \beta} \quad (7.4.38)$$

From (7.3.29), we have, for large  $N$ ,

$$\mu \approx \frac{\delta}{\sqrt{2\pi N} \sigma_f}$$

$$\sigma_f^2 = \int q(x)[x^2 - \mathcal{E}]^2 dx = 2\mathcal{E}^2 \quad (7.4.39)$$

Using this approximation for  $\mu$ ,  $e^{s\delta}/\mu$  is minimized by choosing  $\delta = 1/s$ , yielding

$$\left[ \frac{2e^{s\delta}}{\mu} \right] \approx 2s\mathcal{E} e \sqrt{4\pi N} = \frac{\rho A e \sqrt{4\pi N}}{(1 + \rho)^2 \beta} \quad (7.4.40)$$

To obtain an exact expression for  $\mu$ , we observe that  $\mu$  is the probability that a chi-square random variable of  $N$  degrees of freedom lies between its mean and  $\delta/\mathcal{E}$  below its mean; thus,  $\mu$  can be found from tables of the chi-square distribution.

For the expurgated random-coding bound, we substitute (7.4.18) and (7.4.20) into (7.3.44).

Integrating, we obtain

$$E_x(\rho, X, Y, s) = 2ps\mathcal{E} + \frac{\rho}{2} \ln \left( 1 - 2s\mathcal{E} + \frac{\mathcal{E}}{2\rho\sigma^2} \right) + \frac{\rho}{2} \ln (1 - 2s\mathcal{E}) \quad (7.4.41)$$

Making the substitutions  $A = \mathcal{E}/\sigma^2$  and

$$\beta = 1 - 2s\mathcal{E} + \frac{A}{2\rho} \quad (7.4.42)$$

(7.4.41) becomes

$$\tilde{E}_x(A, \beta, \rho) = (1 - \beta)\rho + \frac{A}{2} + \frac{\rho}{2} \ln \left[ \beta \left( \beta - \frac{A}{2\rho} \right) \right] \quad (7.4.43)$$

where  $\beta$  is constrained by  $A/(2\rho) < \beta < 1 + A/(2\rho)$ . Taking  $\partial \tilde{E}_x / \partial \beta$ , we find that a maximum with respect to  $\beta$  exists where

$$\beta^2 - \beta \left( 1 + \frac{A}{2\rho} \right) + \frac{A}{4\rho} = 0 \quad (7.4.44)$$

or

$$\beta = \frac{1}{2} + \frac{A}{4\rho} + \frac{1}{2} \sqrt{1 + \frac{A^2}{4\rho^2}} \quad (7.4.45)$$

Equation (7.4.44) can be written equivalently as

$$\rho = \frac{A(2\beta - 1)}{4\beta(\beta - 1)} \quad (7.4.46)$$

Next,  $\tilde{E}_x(A, \beta, \rho) - \rho R'$  has a maximum with respect to  $\rho$  where

$$R' = 1 - \beta + \frac{A}{2(2\rho\beta - A)} + \frac{1}{2} \ln \left[ \beta \left( \beta - \frac{A}{2\rho} \right) \right] \quad (7.4.47)$$

For the  $\beta$  and  $\rho$  that satisfy both (7.4.44) and (7.4.47), we can substitute (7.4.46) for the left-hand  $\rho$  in (7.4.47), obtaining

$$R' = \frac{1}{2} \ln \left[ \beta \left( \beta - \frac{A}{2\rho} \right) \right] \quad (7.4.48)$$

$$E_{ex}(R') = (1 - \beta)\rho + \frac{A}{2} \quad (7.4.49)$$

Using (7.4.46) for  $\rho$ , these equations simplify to

$$R' = \frac{1}{2} \ln \frac{\beta^2}{2\beta - 1} \quad (7.4.50)$$

$$E_{ex}(R') = \frac{A}{4\beta} \quad (7.4.51)$$

Solving (7.4.50) for  $\beta$ , we obtain the explicit solution

$$E_{ex}(R') = \frac{A}{4} (1 - \sqrt{1 - e^{-2R'}}) \quad (7.4.52)$$

This is valid for  $\rho \geq 1$ , or combining (7.4.45) and (7.4.50), for

$$R' \leq \frac{1}{2} \ln \left( \frac{1}{2} + \frac{1}{2} \sqrt{1 + \frac{A^2}{4}} \right) \quad (7.4.53)$$

From (7.3.33),  $R'$  is related to the rate  $R$  by,

$$R' = R + \frac{2}{N} \ln \frac{2e^{\delta}}{\mu} \quad (7.4.54)$$

The parameter  $s$  is given by (7.4.42) as

$$\begin{aligned} 2s\mathcal{E}\rho &= (1 - \beta)\rho + \frac{A}{2} \\ &= \frac{A}{4\beta} \end{aligned} \quad (7.4.55)$$

when we have used (7.4.46). Using this value of  $s$  in (7.4.40), we obtain

$$\frac{2e^{s\delta}}{\mu} \approx 2s\mathcal{E}e^{\sqrt{4\pi N}} = \frac{Ae^{\sqrt{4\pi N}}}{4\rho\beta} \quad (7.4.56)$$

These results are summarized in the following theorem.

**Theorem 7.4.4.** Let a discrete-time additive Gaussian noise channel have the transition probability density

$$p(y | x) = \frac{1}{\sqrt{2\pi}\sigma} \exp [-(y - x)^2/(2\sigma^2)]$$

and the constraint  $\bar{x^2} \leq \mathcal{E} = \sigma^2 A$ . Then for any block length  $N$  and any rate  $0 \leq R < C = \frac{1}{2} \ln(1 + A)$ , there exists a code with  $M = [e^{NR}]$  code words, each satisfying the constraint (7.4.19), and each satisfying the bound on error probability

$$P_{e,m} \leq \left[ \frac{2e^{s\delta}}{\mu} \right]^2 e^{-NE_r(R)} \quad (7.4.57)$$

where  $E_r(R)$  is given by (7.4.33) to (7.4.36) and  $2e^{s\delta}/\mu$  is given approximately by (7.4.40). Furthermore, if  $R'$ , as given by (7.4.54) and (7.4.56), satisfies (7.4.53), we also have, for all code words

$$P_{e,m} \leq \exp \left\{ -\frac{NA}{4} [1 - \sqrt{1 - e^{-2R'}}] \right\} \quad (7.4.58)$$

Although it is not immediately obvious from the expression for  $E_r(R)$ , it follows from Section 7.3 that  $\max [E_r(R), E_{ex}(R)]$  is convex  $\cup$ , non-increasing, and positive for  $0 \leq R < \frac{1}{2} \ln(1 + A)$ .

## 7.5 Parallel Additive Gaussian Noise Channels

In the next chapter, we shall reduce a continuous time channel with an additive Gaussian noise process to a set of parallel, discrete-time, additive Gaussian noise channels. The following theorem finds the capacity of such a parallel combination.

**Theorem 7.5.1.** Consider a set of  $N$  parallel, discrete-time, memoryless additive Gaussian noise channels with noise variances  $\sigma_1^2, \dots, \sigma_N^2$ . Let the channel inputs be constrained to satisfy

$$\sum_{n=1}^N \overline{x_n^2} \leq \mathcal{E} \quad (7.5.1)$$

Then capacity is achieved by choosing the inputs to be statistically independent, zero mean, Gaussian random variables of variance

$$\overline{x_n^2} = \mathcal{E}_n \quad (7.5.2)$$

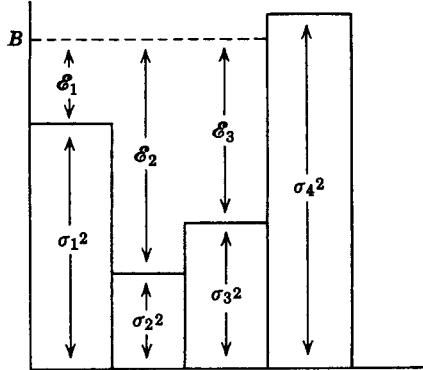


Figure 7.5.1. Water-filling interpretation of parallel, discrete-time, additive Gaussian noise channels.

where the  $\mathcal{E}_n$  satisfy

$$\sigma_n^2 + \mathcal{E}_n = B \quad \text{for } \sigma_n^2 < B \quad (7.5.3)$$

$$\mathcal{E}_n = 0 \quad \text{for } \sigma_n^2 \geq B \quad (7.5.4)$$

and where  $B$  is chosen so that  $\sum \mathcal{E}_n = \mathcal{E}$ . The capacity of the parallel combination is

$$C = \sum_{n=1}^N \frac{1}{2} \ln \left( 1 + \frac{\mathcal{E}_n}{\sigma_n^2} \right) \text{ nats} \quad (7.5.5)$$

$$= \sum_{n: \sigma_n^2 \leq B} \frac{1}{2} \ln (B/\sigma_n^2) \quad (7.5.6)$$

*Discussion.* The solution for the input energies to be used on the various channels has the graphical interpretation of Figure 7.5.1. We can think of the total energy  $E$  as being a quantity of water to be placed in a reservoir with an irregularly shaped bottom determined by the noise variances. The

level to which the water rises is  $B$  and the  $\mathcal{E}_n$  give the depths of water in the various parts of the reservoir.

*Proof.* Let  $\mathbf{X}^N = (X_1 X_2 \cdots X_N)$  and  $\mathbf{Y}^N = (Y_1 \cdots Y_N)$  be the joint input and output ensembles. By the same proof as in (7.2.19),

$$I(\mathbf{X}^N; \mathbf{Y}^N) \leq \sum_{n=1}^N I(X_n; Y_n)$$

with equality if the inputs are independent. Let  $\mathcal{E}_n$  be the mean square value of the  $n$ th input for any particular joint ensemble. Then, using Theorem 7.4.2, we have

$$\sum I(X_n; Y_n) \leq \sum \frac{1}{2} \ln \left( 1 + \frac{\mathcal{E}_n}{\sigma_n^2} \right) \quad (7.5.7)$$

with equality if the inputs are zero mean Gaussian. The right-hand side of (7.5.7) is a convex  $\cap$  function of the vector  $(\mathcal{E}_1, \dots, \mathcal{E}_N)$ . The remaining problem is to maximize this function over the convex region where  $\mathcal{E}_n \geq 0$ ,  $1 \leq n \leq N$ , and  $\sum \mathcal{E}_n \leq \mathcal{E}$ . The maximum clearly occurs when  $\sum \mathcal{E}_n / \mathcal{E} = 1$ , and thus the problem is identical to maximizing a convex function of a probability vector. From Theorem 4.4.1, the necessary and sufficient conditions for the maximum are

$$\frac{\partial \sum \frac{1}{2} \ln (1 + \mathcal{E}_n / \sigma_n^2)}{\partial \mathcal{E}_n} \leq \alpha; \quad \text{all } n$$

with equality if  $\mathcal{E}_n > 0$  and with  $\alpha$  chosen to satisfy  $\sum \mathcal{E}_n = \mathcal{E}$ . Differentiating, we obtain

$$\frac{1}{2(\sigma_n^2 + \mathcal{E}_n)} \leq \alpha$$

Choosing  $B = 1/(2\alpha)$ , we obtain (7.5.3) and (7.5.4). The resulting maximum is given in (7.5.5) and (7.5.6) follows from combining (7.5.7), (7.5.4), and (7.5.5). |

We next extend the random-coding bound and expurgated bound to parallel additive Gaussian noise channels. For simplicity of notation, we shall assume a block length of 1. The results can be applied to an arbitrary block length  $N'$  by considering a set of parallel channels consisting of  $N'$  repetitions of each of the original channels.

For  $N$  parallel channels with individual noise variances  $\sigma_1^2, \dots, \sigma_N^2$ , the joint transition probability density for the set of channels is

$$p_N(\mathbf{y} \mid \mathbf{x}) = \prod_{n=1}^N \frac{1}{\sqrt{2\pi\sigma_n^2}} \exp \left[ -\frac{(y_n - x_n)^2}{2\sigma_n^2} \right] \quad (7.5.8)$$

We assume that each code word  $\mathbf{x}_m$  must satisfy

$$\sum_n (x_{m,n})^2 \leq \mathcal{E}$$

As an input density for the ensemble of codes, we choose

$$q_N(\mathbf{x}) = \frac{\varphi(\mathbf{x})}{\mu} \prod_{n=1}^N \frac{1}{\sqrt{2\pi\mathcal{E}_n}} \exp\left(-\frac{x_n^2}{2\mathcal{E}_n}\right) \quad (7.5.9)$$

where the  $\mathcal{E}_n$  will be chosen later but satisfy

$$\sum_n \mathcal{E}_n = \mathcal{E}. \quad (7.5.10)$$

Also, as in Section 7.3,

$$\varphi(\mathbf{x}) = \begin{cases} 1; & \mathcal{E} - \delta < \sum_n x_n^2 \leq \mathcal{E} \\ 0; & \text{elsewhere} \end{cases} \quad (7.5.11)$$

and  $\mu$  is chosen so that  $q_N(\mathbf{x})$  integrates to 1. Using Theorem 5.6.1 in integral form, the ensemble average probability of error for a code with  $M$  code words satisfies

$$\overline{P_{e,m}} \leq (M-1)^\rho \int_{y_1} \cdots \int_{y_N} \left[ \int_{x_1} \cdots \int_{x_N} q_N(\mathbf{x}) p_N(\mathbf{y} \mid \mathbf{x})^{1/(1+\rho)} d\mathbf{x} \right]^{1+\rho} d\mathbf{y} \quad (7.5.12)$$

for any  $\rho$ ,  $0 \leq \rho \leq 1$ . Upper bounding  $\varphi(\mathbf{x})$  for any  $s \geq 0$  by

$$\varphi(\mathbf{x}) \leq \exp\left[s\delta + s \sum_n (x_n^2 - \mathcal{E}_n)\right] \quad (7.5.13)$$

we can simplify (7.5.12) to

$$\overline{P_{e,m}} \leq \left[ \frac{e^{s\delta}}{\mu} \right]^{1+\rho} (M-1)^\rho \exp\left[-\sum_{n=1}^N E_0(\rho, X_n, Y_n, s)\right] \quad (7.5.14)$$

$$\begin{aligned} E_0(\rho, X_n, Y_n, s) &= -\ln \int_y \left\{ \int_x \frac{1}{\sqrt{2\pi\mathcal{E}_n}} \exp\left[-\frac{x^2}{2\mathcal{E}_n} + s(x^2 - \mathcal{E}_n)\right] \right. \\ &\quad \times \left. \left[ \frac{1}{\sqrt{2\pi\sigma_n^2}} \exp\left[-\frac{(y-x)^2}{2\sigma_n^2}\right] \right]^{1/(1+\rho)} dx \right\}^{1+\rho} dy \quad (7.5.15) \end{aligned}$$

The expression in (7.5.15) is the same integral that we evaluated in (7.4.2), so that

$$\begin{aligned} E_0(\rho, X_n, Y_n, s) &= s(1 + \rho)\mathcal{E}_n + \frac{1}{2} \ln(1 - 2s\mathcal{E}_n) \\ &\quad + \frac{\rho}{2} \ln \left( 1 - 2s\mathcal{E}_n + \frac{\mathcal{E}_n}{(1 + \rho)\sigma_n^2} \right) \quad (7.5.16) \end{aligned}$$

As in (7.3.21), for any  $R \geq 0$ , we can now assert the existence of a code with  $M = [e^R]$  code words, each satisfying

$$P_{e,m} \leq \left[ \frac{2e^{s\delta}}{\mu} \right]^2 \exp \left[ \rho R - \sum_{n=1}^N E_o(\rho, X_n, Y_n, s) \right] \quad (7.5.17)$$

We now maximize the exponent in (7.5.17) over  $0 \leq \rho \leq 1$ ,  $s \geq 0$  by the same procedure that we used for the single additive Gaussian noise channel. We have the additional problem of maximizing over the individual input energies  $\mathcal{E}_n$  subject to the constraints

$$\mathcal{E}_n \geq 0, \sum_n \mathcal{E}_n = \mathcal{E}.$$

Define

$$A_n = \mathcal{E}_n / \sigma_n^2 \quad (7.5.18)$$

$$\beta_n = 1 - 2s\mathcal{E}_n + A_n/(1 + \rho) \quad (7.5.19)$$

We can then replace  $E_o(\rho, X_n, Y_n, s)$  in (7.5.16) by

$$\tilde{E}_o(A_n, \beta_n, \rho) = \frac{1}{2} \left[ (1 - \beta_n)(1 + \rho) + A_n + \ln \left( \beta_n - \frac{A_n}{1 + \rho} \right) + \rho \ln \beta_n \right] \quad (7.5.20)$$

We want to maximize

$$E = -\rho R + \sum_{n=1}^N \tilde{E}_o(A_n, \beta_n, \rho) \quad (7.5.21)$$

over  $A_n$ ,  $\beta_n$ , and  $\rho$ . The  $A_n$  are constrained by  $A_n \geq 0$ ,  $\sum A_n \sigma_n^2 = \mathcal{E}$ . Since  $E$  is a convex  $\cap$  function of the  $A_n$ , a necessary and sufficient condition on the maximum of  $E$  over the  $A_n$  is, for some  $\lambda$  and all  $n$ ,

$$\frac{\partial E}{\partial A_n} = \frac{1}{2} \left[ 1 - \frac{1}{\beta_n(1 + \rho) - A_n} \right] \leq \lambda \sigma_n^2 \quad (7.5.22)$$

with equality if  $A_n > 0$ . We shall temporarily ignore the relationship between the  $\beta_n$  imposed by (7.5.19) and return later to show that our solution satisfies (7.5.19) for some  $s \geq 0$ . Thus, as in (7.4.26) and (7.4.28), we have

$$\frac{\partial E}{\partial \beta_n} = \frac{1}{2} \left[ -(1 + \rho) + \frac{(1 + \rho)}{\beta_n(1 + \rho) - A_n} + \frac{\rho}{\beta_n} \right] = 0 \quad (7.5.23)$$

$$\beta_n = \frac{1}{2} \left( 1 + \frac{A_n}{1 + \rho} \right) \left[ 1 + \sqrt{1 - \frac{4A_n \rho}{(1 + \rho + A_n)^2}} \right] \quad (7.5.24)$$

We can combine (7.5.22) and (7.5.23) to obtain

$$\frac{\rho}{2(1 + \rho)\beta_n} \leq \lambda \sigma_n^2 \quad (7.5.25)$$

If we define  $B$  as  $\rho/[2(1 + \rho)\lambda]$ , (7.5.25) becomes

$$\beta_n \sigma_n^2 \geq B; \quad \text{equality if } A_n > 0 \quad (7.5.26)$$

We next observe from (7.5.24) that, if  $A_n = 0$ ,  $\beta_n = 1$ . Thus, if  $A_n = 0$ , (7.5.26) implies that  $\sigma_n^2 \geq B$ . Furthermore, from (7.5.23),  $\partial E/\partial \beta_n$  is decreasing in  $\beta_n$  and increasing in  $A_n$ . Thus the solution for  $\beta_n$  in (7.5.24) is an increasing function of  $A_n$ , and  $\beta_n > 1$  for  $A_n > 0$ . Since (7.5.26) is satisfied with equality for  $A_n > 0$ , this implies that  $\sigma_n^2 < B$  if  $A_n > 0$ . In other words, all channels with noise variances less than  $B$  are used with positive energy and all channels with noise variances exceeding  $B$  are not used (that is, their input is always 0). For those channels satisfying  $\sigma_n^2 < B$ , we can solve for  $A_n$  in terms of  $\beta_n$  from (7.5.23), obtaining

$$A_n = \frac{(1 + \rho)^2(\beta_n - 1)\beta_n}{(1 + \rho)\beta_n - \rho}$$

Since  $\beta_n = B/\sigma_n^2$  and  $A_n = \mathcal{E}_n/\sigma_n^2$ , this becomes

$$\mathcal{E}_n = \frac{(1 + \rho)^2(B - \sigma_n^2)B}{(1 + \rho)B - \rho\sigma_n^2}; \quad \sigma_n^2 < B \quad (7.5.27)$$

Summing over  $n$ , this gives us an implicit expression for the parameter  $B$ :

$$\mathcal{E} = \sum_{n: \sigma_n^2 < B} \frac{(1 + \rho)^2(B - \sigma_n^2)B}{(1 + \rho)B - \rho\sigma_n^2} \quad (7.5.28)$$

The right-hand side of (7.5.28) is a continuous increasing function of  $B$ . This follows since each term is continuous and whenever a new term enters into the sum, it increases from an initial zero value. Thus (7.5.28) has a unique solution for  $B$ , which in turn specifies  $\beta_n$  from (7.5.26) and  $\mathcal{E}_n$  from (7.5.27). If  $\mathcal{E}_n = 0$ , then any  $s$  is consistent with (7.5.19). If  $\mathcal{E}_n > 0$ , and  $\beta_n$  satisfies (7.5.23), then the value of  $s$  satisfying (7.5.19) has been calculated in (7.4.38) to be

$$s = \frac{\rho A_n}{2(1 + \rho)^2 \beta_n \mathcal{E}_n} = \frac{\rho}{2(1 + \rho)^2 B} \quad (7.5.29)$$

Thus the same value of  $s$  satisfies (7.5.19) for all  $n$  and our solution is consistent.

Next, we maximize over  $\rho$ .

$$\frac{\partial E}{\partial \rho} = -R + \frac{1}{2} \sum_n \left[ 1 - \beta_n + \frac{\beta_n}{(1 + \rho)\beta_n - A_n} - \frac{1}{1 + \rho} + \ln \beta_n \right] = 0 \quad (7.5.30)$$

As in (7.4.29), this simplifies to

$$R = \sum_n \frac{1}{2} \ln \beta_n \quad (7.5.31)$$

$$R = \sum_{n; \sigma_n^2 < B} \frac{1}{2} \ln \frac{B}{\sigma_n^2} \quad (7.5.32)$$

Finally,  $E_r(R)$  is simply  $E$  evaluated for the given  $\rho$ ,  $\beta_n$ , and  $A_n$ ,

$$E_r(R) = \frac{1}{2} \sum_n \left[ (1 - \beta_n)(1 + \rho) + A_n + \ln \left( \beta_n - \frac{A_n}{1 + \rho} \right) \right] \quad (7.5.33)$$

Only the terms with  $\sigma_n^2 < B$  in (7.5.33) contribute to the sum. Using (7.5.19) on the non-logarithmic terms and (7.5.23) on the logarithmic term, we obtain

$$E_r(R) = \frac{1}{2} \sum_{n; \sigma_n^2 < B} 2s\mathcal{E}_n(1 + \rho) - \ln \left( 1 + \rho - \frac{\rho}{\beta_n} \right)$$

Summing over  $n$  and using (7.5.29) for  $s$ ,

$$E_r(R) = \frac{\rho\mathcal{E}}{2B(1 + \rho)} - \sum_{n; \sigma_n^2 < B} \frac{1}{2} \ln \left( 1 + \rho - \frac{\rho\sigma_n^2}{B} \right) \quad (7.5.34)$$

Equations 7.5.28, 7.5.32, and 7.5.34 are parametric relations, with the parameters  $B$  and  $\rho$  ( $0 \leq \rho \leq 1$ ), relating the energy  $\mathcal{E}(B, \rho)$ , the rate  $R(B)$ , and the exponent  $E_r(B, \rho)$ . It is easy to see that  $\mathcal{E}(B, \rho)$  is strictly increasing and continuous in  $B$  and  $\rho$  (for  $B > \min \sigma_n^2$ ). Thus the equation  $\mathcal{E} = \mathcal{E}(B, \rho)$  determines  $B$  as a function of  $\rho$  for a fixed energy  $\mathcal{E}$ , and this implicitly determines  $R(B)$  as a function of  $\rho$  (or  $\rho$  as a function of  $R$  and  $\mathcal{E}$ ). For  $\rho = 0$ , the resulting rate is simply the channel capacity for the given,  $\mathcal{E}$ , as can be seen by comparing (7.5.28) and (7.5.32) with Theorem 7.5.1. The exponent,  $E(B, \rho)$ , is zero for  $\rho = 0$ . As  $\rho$  increases, for fixed energy  $\mathcal{E}$ ,  $B$  decreases,  $R(B)$  decreases, and  $E(B, \rho)$  increases. As before, the slope of  $E$  as a function of  $R$  is  $-\rho$ , as seen most easily from the graphical interpretation of Figure 5.6.4. When  $\rho$  increases to 1, for fixed  $\mathcal{E}$ ,  $B$  has decreased to a critical value  $B_{cr}$  given by

$$\mathcal{E} = \mathcal{E}(B_{cr}, 1) = \sum_{n; \sigma_n^2 \leq B_{cr}} \frac{4(B_{cr} - \sigma_n^2)B_{cr}}{2B_{cr} - \sigma_n^2} \quad (7.5.35)$$

Corresponding to  $B_{cr}$  is a critical value of rate determined by

$$R_{cr} = \sum_{n; \sigma_n^2 \leq B_{cr}} \frac{1}{2} \ln \frac{B_{cr}}{\sigma_n^2} \quad (7.5.36)$$

The parametric equations (7.5.28), (7.5.32), and (7.5.34) are only applicable, for fixed  $\mathcal{E}$ , for  $R$  in the range  $R_{cr} \leq R \leq C$ . For  $R < R_{cr}$ , the exponent  $E$

is maximized by  $\rho = 1$  with the solution for  $A_n$  and  $\beta_n$  as before. This leads to an exponent  $E_r(R)$  given by

$$E_r(R) = E(B_{cr}, 1) + R_{cr} - R \quad (7.5.37)$$

It should be observed that over the entire range of rates, the appropriate distribution of energy over the set of channels is given by (7.5.27) for the appropriate  $B$  and  $\rho$ . As  $R$  changes for a fixed  $\mathcal{E}$ , both  $B$  and  $\rho$  change and the distribution of energies changes. In other words, using Theorem 7.5.1 as an engineering guide in how to distribute energy among parallel Gaussian noise channels is not always a wise thing, even in a system using sophisticated coding techniques.

The coefficient  $[2e^{s\delta}/\mu]^2$  in the probability of error expression can be estimated by the central limit theorem as before. We have\*

$$\mu \approx \frac{\delta}{\sqrt{4\pi \sum_n \mathcal{E}_n^2}} \quad (7.5.38)$$

Choosing  $\delta = 1/s$ , and using (7.5.29), we obtain

$$\frac{2e^{s\delta}}{\mu} \approx \frac{e\rho\sqrt{4\pi \sum \mathcal{E}_n^2}}{(1 + \rho)^2 B} \quad (7.5.39)$$

The expurgated random-coding bound for parallel additive Gaussian noise channels is derived as an extension of the result for a single Gaussian noise channel in the same way as the random-coding bound. Using the ensemble of codes described by (7.5.9), we find that for any  $R \geq 0$ , there is a code of  $M = [e^R]$  code words, each satisfying the energy constraint and each satisfying

$$P_{e,m} \leq \exp -\rho R' + \sum_n E_x(\rho, X_n, Y_n, s) \quad (7.5.40)$$

where

$$R' = R + 2 \ln \frac{2e^{s\delta}}{\mu} \quad (7.5.41)$$

$$E_x(\rho, X_n, Y_n, s) = 2\rho s \mathcal{E}_n + \frac{\rho}{2} \ln \left( 1 - 2s \mathcal{E}_n + \frac{\mathcal{E}_n}{2\rho \sigma_n^2} \right) + \frac{\rho}{2} \ln (1 - 2s \mathcal{E}_n)$$

\* This is only a good approximation if each  $\mathcal{E}_n$  is small compared to  $\mathcal{E}$  [see Feller (1966), Vol. II, Problem 19, p. 502]. We have also assumed that  $\delta$  is small relative to the standard deviation  $\sqrt{2\sum \mathcal{E}_n^2}$ , thus assuming that the density of

$$\sum_n x_n^2$$

is constant between  $\mathcal{E} - \delta$  and  $\mathcal{E}$ . In other words, (5.7.38) is close only if both  $\mu \ll 1$  and if  $\mathcal{E}_n \ll \mathcal{E}$  for all  $n$ .

Making the substitutions  $A_n = \mathcal{E}_n/\sigma_n^2$  and

$$\beta_n = 1 - 2s\mathcal{E}_n + A_n/(2\rho) \quad (7.5.42)$$

we can rewrite  $E_x(\rho, X_n, Y_n, s)$  as

$$\tilde{E}_x(A_n, \beta_n, \rho) = (1 - \beta_n)\rho + \frac{A_n}{2} + \frac{\rho}{2} \ln \left[ \beta_n \left( \beta_n - \frac{A_n}{2\rho} \right) \right] \quad (7.5.43)$$

We now maximize

$$E = -\rho R + \sum_{n=1}^N \tilde{E}_x(A_n, \beta_n, \rho) \quad (7.5.44)$$

over  $A_n$ ,  $\beta_n$ , and  $\rho$ . The  $A_n$  are again constrained by  $A_n \geq 0$ ,  $\sum \sigma_n^2 A_n = \mathcal{E}$ . Since  $E$  is a convex  $\cap$  function of the  $A_n$ , a necessary and sufficient condition on the maximum of  $E$  over the  $A_n$  is, for some  $\lambda$  and all  $n$ ,

$$\frac{\partial E}{\partial A_n} = \frac{1}{2} \left[ 1 - \frac{\rho}{2\rho\beta_n - A_n} \right] \leq \lambda\sigma_n^2 \quad (7.5.45)$$

with equality if  $A_n > 0$ . We shall again temporarily ignore the relationship between the  $\beta_n$  imposed by (7.5.42) and maximize over each  $\beta_n$  separately. This yields

$$\frac{\partial E}{\partial \beta_n} = -\rho + \frac{\rho}{2\beta_n} + \frac{\rho^2}{2\rho\beta_n - A_n} = 0 \quad (7.5.46)$$

$$\beta_n = \frac{1}{2} + \frac{A_n}{4\rho} + \frac{1}{2} \sqrt{1 + \frac{A_n^2}{4\rho^2}} \quad (7.5.47)$$

For  $A_n$ ,  $\beta_n$  satisfying both (7.5.46) and (7.5.45), (7.5.45) simplifies to

$$\frac{1}{4\beta_n} \leq \lambda\sigma_n^2 \quad (7.5.48)$$

Thus, defining  $B$  as  $1/(4\lambda)$ , (7.4.48) becomes

$$\beta_n\sigma_n^2 \geq B \quad (7.5.49)$$

with equality if  $A_n > 0$ .

From (7.5.47),  $\beta_n = 1$  for  $A_n = 0$  and thus from (7.5.49)  $\sigma_n^2 \geq B$ . Also,  $\beta_n > 1$  for  $A_n > 0$ , and thus, since  $\beta_n\sigma_n^2 = B$ ,  $\sigma_n^2 < B$ . Again, only those channels with  $\sigma_n^2 < B$  are used with positive energy.

We next solve for the  $\mathcal{E}_n$ . From (7.5.46),

$$A_n = \frac{4\rho\beta_n(\beta_n - 1)}{2\beta_n - 1} \quad (7.5.50)$$

For  $\sigma_n^2 < B$ , we have  $\beta_n = B/\sigma_n^2$ , and (7.5.50) becomes

$$\mathcal{E}_n = \frac{4\rho B(B - \sigma_n^2)}{2B - \sigma_n^2} \quad (7.5.51)$$

$$\mathcal{E} = \sum_{n: \sigma_n^2 < B} \frac{4\rho B(B - \sigma_n^2)}{2B - \sigma_n^2} \quad (7.5.52)$$

Equation 7.5.52 gives an implicit solution for  $B$  in terms of  $\mathcal{E}$ , and this in turn determines  $\mathcal{E}_n$  from (7.5.51) and  $\beta_n$  from (7.5.49).

For  $A_n > 0$  and  $\beta_n$  satisfying (7.5.46), we have calculated the value of  $s$  satisfying (7.5.42) in (7.4.55).

$$s = \frac{A_n}{8\beta_n \mathcal{E} \rho} = \frac{1}{8B\rho} \quad (7.5.53)$$

Thus the same value of  $s$  satisfies (7.5.42) for all  $n$  and the solution for  $\beta_n$ ,  $\mathcal{E}_n$ , is consistent.

Next  $E$  is maximized with respect to  $\rho$  where  $\partial E / \partial \rho = 0$ , or where

$$R' = \sum_n 1 - \beta_n + \frac{A_n}{2(2\rho\beta_n - A_n)} + \frac{1}{2} \ln \left[ \beta_n \left( \beta_n - \frac{A_n}{2\rho} \right) \right] \quad (7.5.54)$$

As in (7.4.47) to (7.4.51), this leads to

$$R' = \sum_n \frac{1}{2} \ln \frac{\beta_n^2}{2\beta_n - 1} \quad (7.5.55)$$

$$E_{ex}(R') = \sum_n \frac{A_n}{4\beta_n} \quad (7.5.56)$$

Using (7.5.49), these equations reduce to

$$R' = \sum_{n: \sigma_n^2 < B} \frac{1}{2} \ln \frac{B^2}{\sigma_n^2(2B - \sigma_n^2)} \quad (7.5.57)$$

$$E_{ex}(R') = \frac{\mathcal{E}}{4B} \quad (7.5.58)$$

Equations 7.5.52, 7.5.57, and 7.5.58 relate  $\mathcal{E}$ ,  $R'$ , and  $E_{ex}(R')$  parametrically in terms of  $\rho$  and  $B$ . They are valid only for  $\rho \geq 1$ , which for a given  $\mathcal{E}$ , puts an upper bound on  $B$  from (7.5.52). Comparing (7.5.52) for  $\rho = 1$  with (7.5.36), we see that this limiting  $B$  is just  $B_{cr}$ . Thus, for a given  $\mathcal{E}$ , the expurgated bound is valid for

$$0 \leq R' \leq \sum_{n: \sigma_n^2 < B_{cr}} \frac{1}{2} \ln \frac{B_{cr}^2}{\sigma_n^2(2B_{cr} - \sigma_n^2)} \quad (7.5.59)$$

Finally, using the approximation (7.5.38) for  $\mu$ , choosing  $\delta = 1/s$ , and using (7.5.53) for  $s$ , we can relate  $R'$  to the rate  $R$  in (7.5.41) by

$$R' \approx R + 2 \ln \left[ \frac{e}{4B\rho} \sqrt{4\pi \sum_n \mathcal{E}_n^2} \right] \quad (7.5.60)$$

The results of this section are summarized in the following theorem from Ebert (1965).

**Theorem 7.5.2.** Let a set of  $N$  parallel, discrete-time, additive Gaussian noise channels have noise variances  $\sigma_1^2, \dots, \sigma_N^2$ . For any\*  $B > 0$  and any  $\rho$ ,  $0 \leq \rho \leq 1$ , there exists a code with  $M = [e^{R(B)}]$  code words, each code word  $\mathbf{x}_m$  satisfying the constraint

$$\sum_{n=1}^N x_{m,n}^2 \leq \mathcal{E}(B, \rho)$$

and each having an error probability bounded by

$$P_{e,m} \leq \left[ \frac{2e^{s\delta}}{\mu} \right]^2 \exp [-E(B, \rho)] \quad (7.5.61)$$

where  $R(B)$ ,  $\mathcal{E}(B, \rho)$ , and  $E(B, \rho)$  are given by (7.5.32), (7.5.28), and (7.5.34). For fixed  $\mathcal{E} = \mathcal{E}(B, \rho)$ ,  $R(B)$  is strictly and continuously decreasing from  $C$  to  $R_{cr}$  and  $E(B, \rho)$  is strictly and continuously increasing from 0 to  $E(B_{cr}, 1)$  as  $\rho$  goes from 0 to 1. For  $R < R_{cr}$ , there exist codes with  $M = [e^R]$  code words, each of energy at most  $\mathcal{E}$ , and each with

$$P_{e,m} \leq \left[ \frac{2e^{s\delta}}{\mu} \right]^2 \exp \{-[E(B_{cr}, 1) + R_{cr} - R]\} \quad (7.5.62)$$

Furthermore, for any

$$B \geq \min \sigma_n^2$$

and any  $\rho \geq 1$ , there exist codes with  $M = [\exp [R'_x(B) - 2 \ln (2e^{s\mu}/\delta)]]$  words, each of energy at most  $\mathcal{E}_x(B, \rho)$ , and each with

$$P_{e,m} \leq \exp [-E_{ex}(B, \rho)] \quad (7.5.63)$$

where  $R'_x$ ,  $\mathcal{E}_x$ , and  $E_{ex}$  are given by (7.5.57), (7.5.52), and (7.5.58) respectively. For fixed  $\mathcal{E} = \mathcal{E}_x(B, \rho)$ ,  $R'_x(B)$  is strictly and continuously decreasing from  $R_{xcr}$  to 0 and  $E_x(B, \rho)$  strictly increasing as  $\rho$  increases from 1 toward  $\infty$ .

---

For the application of these parallel channels to continuous time channels in the next chapter, we shall see that the coefficient in (7.5.61) and the difference between  $R$  and  $R'$  in (7.5.60) are negligible, leading to our emphasis

\* In the trivial case where  $B \leq \min \sigma_n^2$ ,  $\mathcal{E}$ ,  $R$ , and  $E$  are all zero.

on the exponents  $E_r(R)$  and  $E_{ex}(R')$ . Ebert (1965) has also derived lower bounds on error probability for parallel additive Gaussian noise channels and the exponent in his lower bound agrees with  $E_r(R)$  for  $R_{cr} \leq R \leq C$ .

### **Summary and Conclusions**

In this chapter, we have extended the results of Chapters 4 and 5 to discrete-time, memoryless channels with arbitrary input and output spaces. The basic approach here has been to use only a finite set of letters in the input alphabet and to partition the output alphabet in order to reduce the channel to a discrete memoryless channel. The major difference between this chapter and Chapters 4 and 5 is the inclusion of constraints on the input alphabet. In Sections 7.4 and 7.5 we have applied the general results of the first three sections to the important cases of discrete-time additive Gaussian noise channels and parallel discrete-time additive Gaussian noise channels. The importance of these channels will become apparent in Chapter 8, when we reduce a continuous-time channel with additive Gaussian noise to a set of parallel, discrete-time, additive Gaussian noise channels.

### **Historical Notes and References**

Most of the results in Sections 7.1 to 7.3 are new except for an earlier appearance of the analysis of input constraints (Gallager (1965)). A number of the results have been discovered independently, however, by Wagner (1968). The random-coding bound on error probability for the discrete-time additive Gaussian noise channel is due to Shannon (1959), who also derived a sphere-packing lower bound to error probability. Slepian (1963) has numerically evaluated these upper and lower bounds for a range of block lengths and number of code words. The results of Section 7.5 on parallel, discrete-time, additive noise channels are due to Ebert (1965), who also derived lower bounds to error probability for these channels.

## *Chapter 8*

### WAVEFORM CHANNELS

#### **8.1 Orthonormal Expansions of Signals and White Gaussian Noise**

In this chapter, we shall discuss channels for which the input and output are functions of time where time will now be taken as defined on the continuum rather than at discrete instants. This immediately faces us with the conceptual problem of the “probability” of a waveform. The probability of one of a discrete set of events is a rather simple concept and the probability distribution function of a continuous-valued random variable is not much more involved conceptually. Of course, we could describe a random waveform (or random process, as it is usually called) at a particular instant of time with a probability distribution but, in general, not even a joint probability distribution over a large number of instants of time would suffice as a complete statistical description of the waveform. In principle, we consider a random process to be completely specified if we have a rule from which the joint probability distribution over *any* finite set of times could be calculated. We shall not pursue that approach in this chapter, however; instead we shall take the approach of representing any given real or complex waveform as a series expansion of a set of orthonormal waveforms. We shall then characterize random waveforms in terms of joint probability distributions over the coefficients in such a series expansion. While this approach may seem rather abstract and cumbersome at first, it will turn out to be more useful, both in developing intuition and in proving theorems, than an approach based on the behavior of the waveforms at various instants of time.

Throughout this chapter, we shall be faced with a number of mathematical details such as the convergence of series, the interchange of orders of summation and integration, and the existence of limits. These problems cannot be

argued away on physical grounds since we are dealing here only with mathematical models of physical problems. In fact, we can often obtain a great deal of physical insight about a problem by investigating where mathematical limits fail to exist. On the other hand, if we fuss incessantly about convergence and limits, we shall obscure the more important issues and lose those readers who have neither the background nor the inclination to follow intricate limiting arguments. Many of these questions concerning convergence can be circumvented or, at least, simplified if we restrict our attention to finite energy functions, that is, functions  $x(t)$  for which\*  $\int |x(t)|^2 dt < \infty$ . These functions are often called  $L_2$  functions. While they constitute quite a general class, it should be observed that neither impulse functions nor infinite duration sine wave have finite energy.

Two functions,  $\varphi_1(t)$  and  $\varphi_2(t)$ , are called orthogonal if  $\int \varphi_1(t)\varphi_2^*(t) dt = 0$  where  $\varphi_2^*(t)$  is the complex conjugate of  $\varphi_2(t)$ . A function is said to be normalized if its energy  $\int |\varphi_1(t)|^2 dt$  is equal to 1. An orthonormal set is defined as a set of functions  $\varphi_1(t)$ ,  $\varphi_2(t)$ , ... each of which is normalized and each pair of which is orthogonal; thus, for all  $\varphi_i$ ,  $\varphi_j$  in the set, we have

$$\int \varphi_i(t)\varphi_j^*(t) dt = \delta_{ij} \quad (8.1.1)$$

where  $\delta_{ij} = 1$  for  $i = j$  and  $\delta_{ij} = 0$  otherwise.

Suppose now that a function  $x(t)$  can be represented in terms of an orthonormal set by

$$x(t) = \sum_{i=1}^k x_i \varphi_i(t) \quad (8.1.2)$$

Under these circumstances, the coefficients  $x_i$  will satisfy

$$x_i = \int x(t)\varphi_i^*(t) dt \quad (8.1.3)$$

To see this, substitute (8.1.2) into (8.1.3) and integrate with the help of (8.1.1).

Now let  $x(t)$  be an arbitrary  $L_2$  function and let  $x_i$  be given by (8.1.3). We want to investigate whether the expansion

$$\sum_{i=1}^{\infty} x_i \varphi_i(t)$$

can still be used to represent  $x(t)$ . Let  $x_{r,k}(t)$  be the remainder when  $k$  terms of the expansion are used to represent  $x(t)$ ,

$$x_{r,k}(t) = x(t) - \sum_{i=1}^k x_i \varphi_i(t) \quad (8.1.4)$$

\* Throughout this chapter, any integral without specified limits is to be interpreted as being from  $-\infty$  to  $+\infty$ .

If we multiply both sides of (8.1.4) by  $\varphi_i^*(t)$  and integrate, we see immediately that  $x_{r,k}(t)$  is orthogonal to  $\varphi_i(t)$  for all  $i \leq k$ .

The energy in  $x_{r,k}(t)$  is given by

$$\begin{aligned} \int |x_{r,k}(t)|^2 dt &= \int \left[ x(t) - \sum_{i=1}^k x_i \varphi_i(t) \right] \left[ x^*(t) - \sum_{i=1}^k x_i^* \varphi_i^*(t) \right] dt \\ &= \int |x(t)|^2 dt - \sum_i x_i^* \int x(t) \varphi_i^*(t) dt \\ &\quad - \sum_i x_i \int x^*(t) \varphi_i(t) dt + \sum_{i,j} x_i x_j^* \int \varphi_i(t) \varphi_j^*(t) dt \\ &= \int |x(t)|^2 dt - \sum_{i=1}^k |x_i|^2 \end{aligned} \quad (8.1.5)$$

To get (8.1.5), we have used (8.1.3) and its complex conjugate,  $x_i^* = \int x^*(t) \varphi_i(t) dt$ .

Since the energy of  $x_{r,k}$  is nonnegative for all  $k$ , we have

$$\sum_{i=1}^k |x_i|^2 \leq \int |x(t)|^2 dt \quad (8.1.6)$$

$$\sum_{i=1}^{\infty} |x_i|^2 \leq \int |x(t)|^2 dt \quad (8.1.7)$$

Equation 8.1.7 is known as Bessel's inequality and will be used frequently in what follows.

We can now investigate the limiting behavior of  $x_{r,k}(t)$  by considering the difference between  $x_{r,k}(t)$  and  $x_{r,l}(t)$  for  $l > k$ . From (8.1.4), this difference is

$$\sum_{i=k+1}^l x_i \varphi_i(t)$$

and has an energy

$$\sum_{i=k+1}^l |x_i|^2$$

Bessel's inequality shows that

$$\sum_{i=1}^{\infty} |x_i|^2$$

is bounded and therefore

$$\sum_{i=k+1}^l |x_i|^2$$

must approach 0 as  $k$  and  $l$  increase without limit. Thus  $x_{r,k}(t)$  approaches a limit,\*  $x_r(t)$ , which is orthogonal to all the  $\varphi_i(t)$ .

$$x_r(t) = \lim_{k \rightarrow \infty} x_{r,k}(t) \quad (8.1.8)$$

\* The existence of this limit function is implied by the Riesz-Fischer Theorem. See, for example, Riesz and Nagy (1955).

The notation l.i.m. in (8.1.8) stands for limit in the mean. It means that

$$\lim_{k \rightarrow \infty} \int |x_r(t) - x_{r,k}(t)|^2 dt = 0$$

and does not necessarily imply that  $x_{r,k}(t)$  approaches a limit for each value of  $t$ . We can now expand  $x(t)$  as

$$x(t) = \sum_{i=1}^{\infty} x_i \varphi_i(t) + x_r(t) \quad (8.1.9)$$

$$x_i = \int x(t) \varphi_i^*(t) dt; \quad \int x_r(t) \varphi_i^*(t) dt = 0; \quad \text{all } i.$$

By the infinite sum, both in (8.1.9) and throughout this chapter, we mean

$$\text{l.i.m.}_{k \rightarrow \infty} \sum_{i=1}^k x_i \varphi_i(t)$$

In geometric terms, (8.1.9) states that a function  $x(t)$  can be decomposed into two terms, one in the subspace generated by the  $\varphi_i(t)$  and the other orthogonal to the subspace of the  $\varphi_i(t)$ .

In dealing with (8.1.9), we can usually treat the infinite series just as if it were a finite series. We can usually justify this use by means of the Schwarz inequality, which states that, for two  $L_2$  functions,  $x(t)$  and  $y(t)$ ,

$$\left| \int x(t) y^*(t) dt \right|^2 \leq \int |x(t)|^2 dt \int |y(t)|^2 dt \quad (8.1.10)$$

To verify (8.1.10), let

$$\varphi_1(t) = y(t) / \sqrt{\int |y(t)|^2 dt} \quad (8.1.11)$$

be an orthonormal set consisting of just one member. Then Bessel's inequality, applied to  $x(t)$  is

$$\left| \int x(t) \varphi_1^*(t) dt \right|^2 \leq \int |x(t)|^2 dt \quad (8.1.12)$$

Substituting (8.1.11) into (8.1.12), we have (8.1.10).

To give an example of how the Schwarz inequality can be used to deal with the infinite series expansion of a function  $x(t)$  in (8.1.9), consider the integral

$$\begin{aligned} \int \sum_{i=1}^{\infty} x_i \varphi_i(t) y^*(t) dt &= \sum_{i=1}^k x_i \int \varphi_i(t) y^*(t) dt \\ &\quad + \int \left[ \sum_{i=k+1}^{\infty} x_i \varphi_i(t) \right] y^*(t) dt \end{aligned} \quad (8.1.13)$$

By applying the Schwarz inequality to the last integral (8.1.13), we get

$$\begin{aligned} \left| \int \sum_{i=k+1}^{\infty} x_i \varphi_i(t) y^*(t) dt \right|^2 &\leq \int \left| \sum_{i=k+1}^{\infty} x_i \varphi_i(t) \right|^2 dt \int |y(t)|^2 dt \\ &= \sum_{i=k+1}^{\infty} |x_i|^2 \int |y(t)|^2 dt \end{aligned} \quad (8.1.14)$$

Remembering that we are dealing only with square integrable functions, the limit of the right-hand side of (8.1.14) as  $k \rightarrow \infty$  is 0. We can then pass to the limit in (8.1.13), thus interchanging the order of integration and summation

$$\int \sum_{i=1}^{\infty} x_i \varphi_i(t) y^*(t) dt = \sum_{i=1}^{\infty} x_i \int \varphi_i(t) y^*(t) dt \quad (8.1.15)$$

We say that a set of orthonormal functions is complete over a class of functions if all functions in that class are in the subspace generated by the  $\varphi_i(t)$ , or alternatively, if  $x_r(t)$  is 0 for all functions in the class. In this case, we see that the term on the left in (8.1.15) goes to 0 with increasing  $k$  and thus

$$\int |x(t)|^2 dt = \sum_{i=1}^{\infty} |x_i|^2 \quad (8.1.16)$$

if  $x(t)$  is in the subspace generated by the set  $\varphi_i(t)$ . Equation 8.1.16 is called the energy equation and will be used frequently in what follows.

If  $x(t)$  is in the subspace generated by the orthonormal set  $[\varphi_i(t)]$ , and if  $y(t)$  is any other  $L_2$  function then we also have Parseval's equation,

$$\int x(t) y^*(t) dt = \sum x_i y_i^* \quad (8.1.17)$$

where  $y_i = \int y(t) \varphi_i^*(t) dt$ . To derive (8.1.17), notice that  $x(t) = \sum x_i \varphi_i(t)$ , and thus (8.1.17) is equivalent to (8.1.15).

As a very common example of a set of orthonormal functions, consider

$$\varphi_i(t) = \begin{cases} \sqrt{\frac{1}{T}} \exp\left(\frac{j2\pi it}{T}\right); & |t| \leq T/2 \\ 0 & ; \text{ elsewhere} \end{cases} \quad (8.1.18)$$

where  $j = \sqrt{-1}$  and  $i$  is any integer. Equation 8.1.9 then becomes

$$x(t) = \begin{cases} \sqrt{\frac{1}{T}} \sum_{i=-\infty}^{\infty} x_i \exp\left(\frac{j2\pi it}{T}\right); & |t| \leq T/2 \\ 0 & ; \text{ elsewhere} \end{cases} \quad (8.1.19)$$

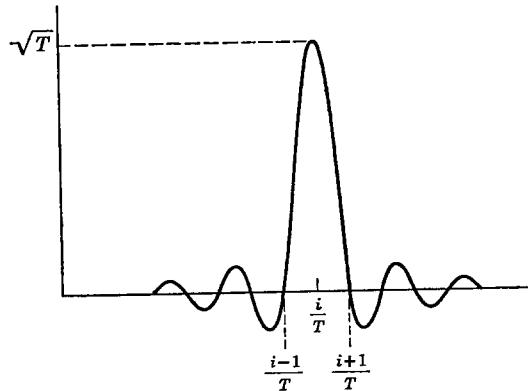
where

$$x_i = \frac{1}{\sqrt{T}} \int_{-T/2}^{T/2} x(t) \exp\left(\frac{-j2\pi it}{T}\right) dt$$

This is simply the Fourier series expansion of a function within an interval  $(-T/2, T/2)$ . This set of functions is complete over the class of finite energy functions that are limited to the interval  $(-T/2, T/2)$ ,\* and thus the remainder term  $x_r(t)$  accounts for  $x(t)$  outside of the interval.

Suppose that we wish to generate a set of signals that are time limited to  $(-T/2, T/2)$  and also approximately frequency limited to frequencies less than some maximum value  $W$ . We can get some insight into this problem by considering  $\varphi_i(t)$  in (8.1.18) as a complex sinusoid of frequency  $i/T$ . Since  $\varphi_i(t)$  is truncated, however, it spreads out over a range of frequencies and has a Fourier transform given by

$$\Phi_i(f) = \int \varphi_i(t) e^{-j2\pi f t} dt = \sqrt{T} \frac{\sin \pi T(f - i/T)}{\pi T(f - i/T)} \quad (8.1.20)$$



**Figure 8.1.1.** Sketch of  $\Phi_i(f) = \sqrt{T} \frac{\sin \pi T[f - (i/T)]}{\pi T[f - (i/T)]}$ .

The function  $\Phi_i(f)$  is sketched in Figure 8.1.1 and it is clear from the figure that  $\varphi_i(t)$  does have most of its energy at frequencies in the vicinity of  $i/T$ . If we consider functions  $x(t)$  which are linear combinations of the  $\varphi_i(t)$  for  $-WT \leq i \leq WT$  where  $WT$  is an integer, we have

$$x(t) = \sum_{i=-WT}^{WT} x_i \varphi_i(t) \quad (8.1.21)$$

This class of functions is strictly time limited to  $(-T/2, T/2)$  and, in some sense, frequency limited to  $(-W, W)$ . Thus an arbitrary complex function in this class is specified by specifying  $2WT + 1$  complex numbers. If we require  $x(t)$  to be real, then  $x_{-i} = x_i^*$ , and  $x(t)$  is specified by  $2WT + 1$  real numbers,

\* See, for example, Akheizer and Glazman, p. 24 (1961).

namely  $x_0$ , which is real, and the real and imaginary parts of  $x_i$  for  $i > 0$ . A class of functions in which any particular function can be specified by  $n$  real numbers is said to have  $n$  degrees of freedom, and thus the class of real  $x(t)$  satisfying (8.1.21) has  $2WT + 1$  degrees of freedom. Notice that it does not make any sense to talk about the number of degrees of freedom in a waveform without first specifying the class of waveforms from which it comes.

If we try to be more precise about the sense in which the functions satisfying (8.1.21) are band limited in frequency, we run into a number of problems, and in fact it is possible by a sufficiently malicious choice of the  $x_i$  to place most of the energy in  $x(t)$  outside the frequency band  $(-W, W)$ . In Section 8.4, this problem of time- and frequency-limited waveforms will be treated in a mathematically more satisfying way by using a different set of orthonormal functions. The approach here, however, using time-limited sinusoids, is very useful in gaining engineering insight into a variety of problems; it should not be shunned because of the lack of precision in the notion of band limiting.

A second set of orthonormal functions which are very useful in acquiring insight into many problems are the sampling functions

$$\theta_i(t) = \sqrt{2W} \frac{\sin 2\pi W[t - i/(2W)]}{2\pi W[t - i/(2W)]} \quad (8.1.22)$$

To see that these functions are orthonormal, we must first state the Parseval relation for Fourier transforms. Let  $x(t)$  and  $y(t)$  have transforms  $X(f)$  and  $Y(f)$ . Then\*

$$\int x(t)y^*(t) dt = \int X(f)Y^*(f) df \quad (8.1.23)$$

Thus, if  $\{\varphi_i(t)\}$  is any orthonormal set and  $\Phi_i(f)$  is the Fourier transform of  $\varphi_i(t)$  for each  $i$ , then  $\{\Phi_i(f)\}$  is also an orthonormal set, satisfying

$$\delta_{i,i} = \int \varphi_i(t)\varphi_i^*(t) dt = \int \Phi_i(f)\Phi_i^*(f) df \quad (8.1.24)$$

Letting  $\Phi_i(f)$  be given by (8.1.20) and substituting  $t$  for  $f$  and  $2W$  for  $T$ , we see that the  $\theta_i(t)$  in (8.1.22) are orthonormal.

Using (8.1.20) as a guide, we can find the Fourier transform of  $\theta_i(t)$ ,

$$\Theta_i(f) = \begin{cases} \sqrt{\frac{1}{2W}} \exp\left(-\frac{j2\pi if}{2W}\right); & |f| \leq W \\ 0 & ; |f| > W \end{cases} \quad (8.1.25)$$

\* For general finite energy functions, these Fourier transforms exist in the sense that  $X(f) = \text{l.i.m.}_{T \rightarrow \infty} \int_{-T}^T x(t)e^{-j2\pi ft} dt$ ;  $x(t) = \text{l.i.m.}_{F \rightarrow \infty} \int_{-F}^F X(f)e^{j2\pi ft} df$  and (8.1.23) is always valid.

See Titchmarsh (1948). Theorems 48 and 49.

Thus we see that the  $\theta_i(t)$  are all bandlimited to  $W$  in the sense that their Fourier transforms are 0 for  $|f| > W$ .

The functions  $\theta_i(t)$  are called sampling functions because an arbitrary function  $x(t)$ , bandlimited to  $|f| \leq W$ , can be represented in terms of these functions and the values of  $x(t)$  at intervals of  $1/2W$ .

$$x(t) = \sum_{i=-\infty}^{\infty} \sqrt{\frac{1}{2W}} x\left(\frac{i}{2W}\right) \theta_i(t) \quad (8.1.26)$$

To derive (8.1.26), let  $X(f)$  be the Fourier transform of  $x(t)$ . Since  $X(f) = 0$  for  $|f| > W$ , it can be expanded in a Fourier series,

$$X(f) = \sum_{i=-\infty}^{\infty} x_i \Theta_i(f) \quad (8.1.27)$$

$$x_i = \int X(f) \Theta_i^*(f) df$$

Using (8.1.25) and the fact that  $X(f) = 0$  for  $|f| > W$ , we have

$$\begin{aligned} x_i &= \int_{-W}^W X(f) \sqrt{\frac{1}{2W}} \exp\left(\frac{j2\pi if}{2W}\right) df \\ &= \sqrt{\frac{1}{2W}} \int_{-\infty}^{\infty} X(f) \exp\left(\frac{j2\pi if}{2W}\right) df = \sqrt{\frac{1}{2W}} x\left(\frac{i}{2W}\right) \end{aligned}$$

Substituting this expression for  $x_i$  into (8.1.27) and Fourier transforming, we have (8.1.26). We have assumed in the above derivation that  $x(t)$  is sufficiently well behaved that the inverse transform of  $X(f)$  converges to  $x(t)$  everywhere.

Just as we used the Fourier series expansion to generate signals time limited to  $(-T/2, T/2)$  and approximately bandlimited to  $|f| \leq W$ , we can use the sampling expansion to generate signals that are precisely bandlimited and approximately time limited. In this case, we use the  $\theta_i(t)$  for which  $|i| \leq WT$ . Again, we get  $2WT + 1$  degrees of freedom and the  $x(t)$  in the class are zero for all sample points  $|i/(2W)| > T/2$ . This representation is subject to the same limitations as the Fourier series representation, and is, in reality, the same representation with the roles of time and frequency reversed.

### Gaussian Random Processes

In this section, we shall give a brief description of Gaussian random processes and show why they are so often reasonable models for physical noise waveforms. A random process\*  $z(t)$  can be thought of as a set of waveforms

\* More specifically, we are discussing continuous parameter random processes, in distinction to the discrete parameter random processes of Section 3.5.

with a probability measure on the set. More specifically, it can be defined as a collection  $z(t)$  of random variables, one random variable for each choice of the real number parameter  $t$ . One way to specify a random process is by a rule which, for each finite set of instants of time,  $t_1, \dots, t_n$ , specifies the joint distribution function  $F_{t_1, t_2, \dots, t_n}(z_1, z_2, \dots, z_n)$  of the random variables  $z(t_1), \dots, z(t_n)$ . For each choice of the real numbers  $z_1, \dots, z_n$ , this distribution is the probability that  $z(t_1) \leq z_1, z(t_2) \leq z_2, \dots, z(t_n) \leq z_n$ . A random process is stationary if the probability measure is invariant to a shift of the time scale, or more specifically if for each finite set of times  $t_1, \dots, t_n$ , for each time interval  $T$ , and for each choice of real numbers  $z_1, \dots, z_n$ , we have

$$F_{t_1, \dots, t_n}(z_1, \dots, z_n) = F_{t_1+T, \dots, t_n+T}(z_1, \dots, z_n) \quad (8.1.28)$$

A random process is said to be zero mean if, for each  $t$ , the expected value of  $z(t)$  is zero. In what follows, we shall deal only with zero mean random processes. There is really no loss of generality here since an arbitrary random process can be split into two parts,  $\bar{z}(t)$  and  $z(t) - \bar{z}(t)$  where  $\bar{z}(t)$  is a deterministic function (although not necessarily  $L_2$ ) and  $z(t) - \bar{z}(t)$  is a zero mean random process.

The *autocorrelation function* of a random process  $z(t)$  is a function of two real variables defined by

$$\mathcal{R}(t_1, t_2) = \overline{z(t_1)z(t_2)} \quad (8.1.29)$$

The autocorrelation function of a random process obviously does not provide a complete characterization of the process, but as we shall now see, it provides a sufficient characterization to answer a number of questions concerning the linear filtering of a random process.

Suppose that a linear time-invariant filter has an impulse response  $h(t)$ . By this, we mean that the output waveform from the filter is the convolution of the input waveform with  $h(t)$ , and thus if the input is a random process  $z(t)$ , the output is another random process  $y(t)$  given by\*

$$y(t) = \int h(t - \tau)z(\tau) d\tau \quad (8.1.30)$$

The autocorrelation function of  $y(t)$  is then given by

$$\mathcal{R}_y(t_1, t_2) = \overline{y(t_1)y(t_2)} = \overline{\int \int h(t_1 - \tau_1)h(t_2 - \tau_2)z(\tau_1)z(\tau_2) d\tau_1 d\tau_2} \quad (8.1.31)$$

\* The argument from here to (8.1.34) is needed only for motivation and insight and hence we omit a careful definition of what is meant by the integral in (8.1.30) and also we omit any justification for the interchange of the order of integration and expectation in (8.1.32). For a more careful treatment of this argument, see, for example, Davenport and Root (1958), Chapter 4 or Yaglom (1962), Chapters 1 and 2.

Interchanging the order of integration and expectation, this becomes

$$\mathcal{R}_y(t_1, t_2) = \int \int h(t_1 - \tau_1) h(t_2 - \tau_2) \mathcal{R}_z(\tau_1, \tau_2) d\tau_1 d\tau_2 \quad (8.1.32)$$

Thus the correlation function of the output random process from a linear time-invariant filter is determined from the correlation function of the input random process. If the random process  $z(t)$  is stationary, then  $\mathcal{R}_z(t_1, t_2)$  is a function only of the difference  $t = t_1 - t_2$  and we express  $\mathcal{R}_z$  as a function of only one variable  $\mathcal{R}_z(t)$ . More generally, a random process is called *wide-sense stationary* if its autocorrelation function  $\mathcal{R}(t_1, t_2)$  is a function only of  $t = t_1 - t_2$ . It is easy to see from (8.1.32) that, if  $z(t)$  is wide-sense stationary, then  $y(t)$  is also. If  $z(t)$  is wide-sense stationary, we can interpret (8.1.32) as  $\mathcal{R}_z(t)$  convolved with  $h(t)$  and the result convolved with  $h(-t)$ . Consequently, if we define the *power spectral density* of  $S_z(f)$  of a wide-sense stationary random process  $z(t)$  as the Fourier transform of  $\mathcal{R}_z(t)$ ,  $S_z(f) = \int \mathcal{R}_z(t) e^{-j2\pi ft} dt$  and let  $H(f) = \int h(t) e^{-j2\pi ft} dt$  be the frequency response of the filter, we obtain

$$S_y(f) = S_z(f) |H(f)|^2 \quad (8.1.33)$$

To interpret the meaning of power spectral density, we define the *power* in a wide-sense stationary random process  $y(t)$  as  $\overline{y^2(t)} = \mathcal{R}_y(0)$ . Since  $S_y(f)$  is the Fourier transform of  $\mathcal{R}_y(t)$ , we have

$$\overline{y^2(t)} = \int_{-\infty}^{\infty} S_y(f) = \int_{-\infty}^{\infty} S_z(f) |H(f)|^2 \quad (8.1.34)$$

If  $|H(f)|^2$  is one over a narrow band of frequencies and zero elsewhere, then the power in the filter output is the integral of  $S_z(f)$  over that narrow band and we see that  $S_z(f)$  has the physical interpretation of being the power density per unit bandwidth at frequency  $f$ .

We shall now discuss physical noise waveforms briefly and indicate why they are often appropriately modeled by a special class of random processes called Gaussian random processes. For many types of noise, the physical coupling is essentially zero between the values of the noise at any two instants of time separated by more than some very small interval  $\Delta$  which we shall call the *coherence time* of the noise. In a random process model of the noise, it is reasonable to assume that the noise is approximately statistically independent between two times separated by more than  $\Delta$ . Thus, if such a noise waveform is the input to a filter with an impulse response  $h(t)$  and if  $h(t)$  is appreciably nonzero over an interval very much larger than  $\Delta$ , then the central limit theorem leads us to expect that the response of the filter at any given instant can be appropriately modeled as a Gaussian random variable. If  $\Delta$  is so small that this assumption is reasonable for all filters that we are

interested in considering, then we can simplify our random process model of the noise by assuming that the output from any linear filter at any given time is a Gaussian random variable. Such random processes are known as Gaussian. More specifically, a zero mean Gaussian random process\*  $z(t)$  is defined as a process with the property that for any  $L_2$  function  $x(t)$ ,  $\int x(t)z(t) dt$  is a zero mean, finite variance Gaussian random variable. For the random processes previously discussed, this integral can be interpreted in a straightforward way. Here, however, we want to discuss a somewhat broader class of random processes including the white Gaussian random process (or white Gaussian noise) which is defined by the property that for any  $L_2$  function  $x(t)$ ,  $x = \int x(t)z(t) dt$  is a zero mean Gaussian random variable with the variance

$$\overline{x^2} = \frac{N_o}{2} \int x^2(t) dt \quad (8.1.35)$$

where  $N_o/2$  is a constant independent of  $x(t)$ . As we shall see later, this process is so “wild” that it cannot be defined as a collection of random variables, one for each value of the parameter  $t$ . On the other hand, all we shall need to work with in what follows are the random variables  $\int x(t)z(t) dt$ , and thus we shall take a random process as being defined if there is a rule defining the random variable  $\int x(t)z(t) dt$  for all  $L_2$  functions  $x(t)$ . With this approach, we need worry neither about what the above integral means nor about how  $z(t)$  is specified as a collection of random variables in  $t$ . The above approach is similar to that used in discussing generalized functions where we do not define the generalized function in terms of its value for each value of the argument but, instead, define it in terms of its integral against each of a suitably defined class of functions. For this reason, white Gaussian noise is usually called a generalized random process. The following linearity condition will have to be imposed on the random variables  $\int \phi(t)z(t) dt$ : for any set of constants  $a_1, \dots, a_k$  and any set of  $L_2$  functions  $\phi_1(t), \dots, \phi_k(t)$ , we require that

$$\int \left[ \sum_{i=1}^k a_i \phi_i(t) \right] z(t) dt = \sum_{i=1}^k a_i \int \phi_i(t) z(t) dt \quad (8.1.36)$$

For a zero mean Gaussian random process, both the random variables  $z_i = \int \phi_i(t)z(t) dt$  and the random variable  $y = \int [\sum a_i \phi_i(T)]z(t) dt$  are zero mean and Gaussian. Thus each linear combination of the  $z_i$  is also Gaussian. A set of random variables for which each finite linear combination is Gaussian is called a *jointly Gaussian* set of random variables, so that the set of  $z_i$  above are jointly Gaussian.

\* This is not the broadest definition that could be given. For stationary processes, it can be shown that the definition here restricts the spectral density of the process to be bounded (see Problem 8.2).

We can easily find the joint characteristic function and joint probability density of a set  $z_1, \dots, z_k$  of jointly Gaussian, zero mean random variables as follows. The joint characteristic function of  $z_1, \dots, z_k$  is by definition

$$M_{z_1, \dots, z_k}(v_1, \dots, v_k) = \exp j \sum_{i=1}^k v_i z_i \quad (8.1.37)$$

Let the random variable  $y$  be

$$\sum_{i=1}^k v_i z_i$$

Since  $y$  is zero mean Gaussian, its characteristic function is

$$\begin{aligned} M_y(u) &= \overline{e^{juy}} = \int \frac{1}{\sqrt{2\pi\sigma_y^2}} \exp\left(-\frac{y^2}{2\sigma_y^2} + juy\right) dy \\ &= \exp(-\sigma_y^2 u^2) \end{aligned} \quad (8.1.38)$$

where  $\sigma_y^2$  is the variance of  $y$ ,

$$\sigma_y^2 = \sum_{i=1}^k \sum_{l=1}^k v_i v_l \bar{z}_i z_l.$$

Noting that  $\overline{e^{juy}}$  is just the right-hand side of (8.1.37), we can substitute  $u = 1$  in (8.1.38), and obtain

$$M_{z_1, \dots, z_k}(v_1, \dots, v_k) = \exp\left(-\sum_{i=1}^k \sum_{l=1}^k v_i v_l \bar{z}_i z_l\right) \quad (8.1.39)$$

Since the joint characteristic function of  $z_1, \dots, z_k$  is the multidimensional Fourier transform of the joint probability density of  $z_1, \dots, z_k$ , we can find the joint probability density by taking the inverse Fourier transform of (8.1.39). This gives us:

$$p(z_1, \dots, z_k) = \frac{\exp\left(-\frac{1}{2|\Lambda|} \sum_{i=1}^k \sum_{l=1}^k \Lambda_{i,l} \bar{z}_i z_l\right)}{(2\pi)^{n/2} |\Lambda|^{1/2}} \quad (8.1.40)$$

where  $\Lambda$  is the  $k$  by  $k$  matrix with components  $\bar{z}_i z_l$ ,  $|\Lambda|$  is the determinant of  $\Lambda$  and  $\Lambda_{i,l}$  is the cofactor of the  $i, l$  term of  $\Lambda$ . If  $|\Lambda| = 0$ , then some linear combination of the  $z_i$  has zero variance and the joint probability density will only exist in the sense of containing impulses. It is important to observe that this joint density is determined solely by the set of correlation coefficients  $\bar{z}_i z_l$ . If  $\bar{z}_i z_l = 0$  for all  $i \neq l$ , then it can be seen that  $p(z_1, \dots, z_k)$  will factor into a product  $p_{z_1}(z_1) \cdots p_{z_k}(z_k)$ . Thus we have the important result that, if a set of jointly Gaussian, zero mean random variables are uncorrelated (that is,  $\bar{z}_i z_l = 0$  for  $i \neq l$ ), then they are statistically independent.

For white Gaussian noise, this result has the interesting corollary that if  $\phi_1(t), \dots, \phi_k(t)$  are orthogonal, then the set of  $z_i = \int \phi_i(t)z(t) dt, 1 \leq i \leq k$ , are statistically independent. To see this, we observe that

$$\overline{(z_i + z_l)^2} = \frac{N_o}{2} \int [\phi_i(t) + \phi_l(t)]^2 dt.$$

Expanding and cancelling out the square terms,

$$\overline{z_i z_l} = \frac{N_o}{2} \int \phi_i(t)\phi_l(t) dt = 0 \quad (8.1.41)$$

Now let  $\{\phi_i(t)\}$  be a complete set of orthonormal functions and for a zero mean Gaussian random process let the set of jointly Gaussian random variables  $\{z_i\}$  be given by  $z_i = \int \phi_i(t)z(t) dt$ . For an arbitrary  $L_2$  function,

$$x(t) = \sum_{i=1}^{\infty} x_i \phi_i(t),$$

we then have\*

$$\int x(t)z(t) dt = \text{l.i.m.}_{k \rightarrow \infty} \sum_{i=1}^k x_i z_i \quad (8.1.42)$$

by which we mean that

$$\lim_{k \rightarrow \infty} \left[ \int x(t)z(t) dt - \sum_{i=1}^k x_i z_i \right]^2 = 0.$$

It follows from this that a zero mean Gaussian random process is completely determined by the correlation coefficients  $\overline{z_i z_l}$  in the above expansion. For white Gaussian noise,  $\overline{z_i z_l} = (N_o/2)\delta_{il}$  where  $\delta_{il}$  is 1 for  $i = l$  and 0 for  $i \neq l$ .

Now suppose that  $z(t)$  is a zero mean Gaussian random process for which  $z(t)$  is also defined as a family of random variables in the parameter  $t$ . Then for a complete orthonormal set  $\{\phi_i(t)\}$  with  $z_i = \int \phi_i(t)z(t) dt$ , the correlation coefficient  $\overline{z_i z_l}$  is given by

$$\overline{z_i z_l} = \iint \phi_i(t)\phi_l(\tau) \overline{z(t)z(\tau)} dt d\tau \quad (8.1.43)$$

Thus we see that the process is completely determined by its autocorrelation function  $\mathcal{R}(t, \tau) = \overline{z(t)z(\tau)}$ .

Next we show that if  $\mathcal{R}(t, \tau)$  is continuous, then for each  $t$ ,  $z(t)$  is a zero

\* See Problem 8.1 for a proof that equality holds in (8.1.42) and that the limit exists in general.

mean Gaussian random variable. For any given  $t$ , define  $u_n(\tau)$  by

$$u_n(\tau) = \begin{cases} n; & t - \frac{1}{2n} \leq \tau \leq t + \frac{1}{2n} \\ 0; & \text{elsewhere} \end{cases} \quad (8.1.44)$$

and let  $y_n = \int u_n(\tau)z(\tau) d\tau$ . For each  $n$ ,  $y_n$  is zero mean Gaussian. Also  $y_n - z(t) = \int u_n(\tau)[z(\tau) - z(t)] d\tau$ , and thus

$$\overline{[y_n - z(t)]^2} = \iint u_n(\tau_1)u_n(\tau_2) \overline{[z(\tau_1) - z(t)][z(\tau_2) - z(t)]} d\tau_1 d\tau_2 \quad (8.1.45)$$

Since  $\mathcal{R}(t, \tau)$  is continuous,  $\overline{[z(\tau_1) - z(t)][z(\tau_2) - z(t)]}$  approaches zero as  $\tau_1$  and  $\tau_2$  approach  $t$  and consequently

$$\lim_{n \rightarrow \infty} \overline{[y_n - z(t)]^2} = 0 \quad (8.1.46)$$

It follows that  $z(t)$  is a zero mean Gaussian random variable. By a slight extension of this argument, considering linear combinations of  $z(t_1), \dots, z(t_k)$ , it can be seen that  $z(t_1), \dots, z(t_k)$  is a jointly Gaussian set of random variables. Conversely, it can be shown [Davenport and Root (1958), p. 155] that if a random process has a continuous autocorrelation function and if for each set of time instants  $t_1, \dots, t_k$ , the random variables  $z(t_1), \dots, z(t_k)$  are zero mean and jointly Gaussian, then  $z(t)$  is a zero mean Gaussian random process (according to our original definition).

Now consider passing a zero mean Gaussian random process through a linear time-invariant filter with an  $L_2$  impulse response  $h(t)$ . The resulting random process  $y(t) = \int h(t - \tau)z(\tau) d\tau$  is then also a zero mean Gaussian random process since, for each set of time instants  $t_1, \dots, t_k$ , the set of random variables  $y(t_1), \dots, y(t_k)$  is zero mean and jointly Gaussian [see Problem 8.5 for a proof that  $y(t)$  has a continuous autocorrelation function].

For the special case in which  $z(t)$  is white Gaussian noise the process  $y(t) = \int h(t - \tau)z(\tau) d\tau$  has a very simple characterization. By the same argument as in (8.1.41), we have

$$\overline{y(\tau_1)y(\tau_2)} = \frac{N_o}{2} \int h(t - \tau_1)h(t - \tau_2) dt$$

Thus  $y(t)$  is a stationary process with the autocorrelation function

$$\mathcal{R}_y(\tau) = \frac{N_o}{2} \int h(t)h(t + \tau) dt \quad (8.1.47)$$

Setting  $\tau = 0$ , letting  $H(f)$  be the Fourier transform of  $h(t)$ , and recalling from (8.1.23) that  $\int |H(f)|^2 df = \int h^2(t) dt$ , we have

$$\overline{y^2(t)} = \frac{N_0}{2} \int |H(f)|^2 df \quad (8.1.48)$$

Comparing this with (8.1.34), we see that  $N_0/2$  can be interpreted as the power spectral density of white Gaussian noise. We see from this that white Gaussian noise is a reasonable random process model to use for noise which has an essentially flat spectral density over the frequency region of interest. The assumption that the spectral density is flat over all frequencies then greatly simplifies calculations with the model. Formally, the autocorrelation function of white Gaussian noise  $z(t)$  is the inverse Fourier transform of  $N_0/2$ , which is an impulse of magnitude  $N_0/2$  at  $\tau = 0$ . This means that if we wish to interpret  $z(t)$  as a collection of random variables in the parameter  $t$ , we must conclude that the variance of  $z(t)$  for each  $t$  is infinite. This same conclusion can be reached by applying the argument in (8.1.44) and (8.1.45) to  $z(t)$ . This phenomenon helps to illustrate why we have been emphasizing the linear operations  $\int x(t)z(t) dt$  on a random process rather than the time variables. The random variables  $z(t)$  in the parameter  $t$  are often strongly affected by the power spectral density of the process in frequency regions that are of no physical interest, and thus there is often little relation between a noise waveform and the sample functions (in time) of a random process model for the noise.

### **Mutual Information for Continuous-Time Channels**

Let  $x(t)$  be an  $L_2$  input and  $y(t)$  the output from a continuous time channel. Let  $\varphi_1(t), \varphi_2(t), \dots$  be a complete set of real orthonormal functions defined over the interval  $(0, T)$ . Then we can represent  $x(t)$  over the interval  $(0, T)$  by

$$\begin{aligned} x(t) &= \sum x_n \varphi_n(t) \quad 0 \leq t \leq T \\ x_n &= \int x(t) \varphi_n(t) dt \end{aligned} \quad (8.1.49)$$

Similarly, if  $\theta_1(t), \theta_2(t), \dots$  is another complete set of orthonormal functions over  $(0, T)$ , we can represent  $y(t)$  by the variables

$$y_n = \int y(t) \theta_n(t) dt \quad (8.1.50)$$

The set  $\{\theta_n(t)\}$  can be the same as the set  $\{\varphi_n(t)\}$ , and such a choice is often convenient.

Let  $\mathbf{x}^N$  and  $\mathbf{y}^N$  be the sequences  $\mathbf{x}^N = (x_1, \dots, x_N)$ ,  $\mathbf{y}^N = (y_1, \dots, y_N)$ . The channel can be described statistically in terms of the joint conditional

probability densities  $p_N(\mathbf{y}^N | \mathbf{x}^N)$  given for all  $N$ . To avoid the influence of  $x(t)$  for  $t < 0$ , we assume that  $x(t) = 0$  for  $t < 0$ . For simplicity, we also assume that the input ensemble can be described by a joint probability density  $q_N(\mathbf{x}^N)$  for all  $N$ . The mutual information between  $x(t)$  and  $y(t)$  for  $0 \leq t \leq T$ , if it exists, is then given by\*

$$I_T[x(t);y(t)] = \lim_{N \rightarrow \infty} I(\mathbf{x}^N; \mathbf{y}^N) \quad (8.1.51)$$

where†

$$I(\mathbf{x}^N; \mathbf{y}^N) = \log \frac{p_N(\mathbf{y}^N | \mathbf{x}^N)}{\int_{\mathbf{x}_1^N} q_N(\mathbf{x}_1^N) p_N(\mathbf{y}^N | \mathbf{x}_1^N) d\mathbf{x}_1^N} \quad (8.1.52)$$

The average mutual information between input and output over the interval  $(0, T)$  is given, if it exists, by

$$I_T[X(t);Y(t)] = \lim_{N \rightarrow \infty} I(\mathbf{X}^N; \mathbf{Y}^N) \quad (8.1.53)$$

$$I(\mathbf{X}^N; \mathbf{Y}^N) = \int_{\mathbf{x}^N, \mathbf{y}^N} q_N(\mathbf{x}^N) p_N(\mathbf{y}^N | \mathbf{x}^N) I(\mathbf{x}^N; \mathbf{y}^N) d\mathbf{x}^N d\mathbf{y}^N \quad (8.1.54)$$

Notice that  $I(\mathbf{X}^N; \mathbf{Y}^N)$  is implicitly a function both of  $T$  and of  $q_N(\mathbf{x}^N)$ . The capacity of the channel, per unit time, is defined by

$$C = \lim_{T \rightarrow \infty} C_T \quad (8.1.55)$$

where

$$C_T = \frac{1}{T} [\sup I(\mathbf{X}^N; \mathbf{Y}^N)] \quad (8.1.56)$$

where the supremum is over all input probability distributions consistent with the input constraints on the channel. The quantity in brackets above is the maximum mutual information that can be transmitted in time  $T$ . For an arbitrary continuous channel, the limit as  $T \rightarrow \infty$  above need not exist and capacity is only defined if the limit does exist. If  $C$  does exist, then the converse to the coding theorem clearly applies, but the coding theorem need not apply; that is, we can construct examples of channels with a capacity according to (8.1.55) but such that, at rates below capacity, data cannot be transmitted with arbitrarily low error probability.

\* It should not be surprising that this definition is equivalent to the general definition of mutual information in Chapter 2. There are some mathematical subtleties involved in the proof, however, and the interested reader is referred to Gel'fand and Yaglom (1957). That the definition is equivalent to the general definition in Chapter 2 also implies that the definition is independent of the particular set of orthonormal functions used.

†  $I(\mathbf{x}^N; \mathbf{y}^N)$  often exists even if the probability densities do not, as discussed in Chapter 2.

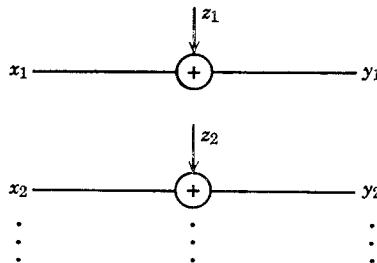
## 8.2 White Gaussian Noise and Orthogonal Signals

In this section, we shall apply the orthonormal expansions of the last section to finding the capacity of a channel with additive white Gaussian noise with an input power constraint. We then investigate the error probability achievable with orthogonal signals on such a channel.

Let  $\varphi_1(t), \varphi_2(t), \dots$  be a complete set of real orthonormal functions over  $(0, T)$ . The input  $x(t)$ , for  $0 \leq t \leq T$ , can then be represented by (8.1.49). Similarly, the noise  $z(t)$  can be represented by

$$z_n = \int z(t) \varphi_n(t) dt \quad (8.2.1)$$

For white Gaussian noise of spectral density  $N_o/2$ , the noise components  $z_1, z_2, \dots$  are by definition statistically independent Gaussian random



*Figure 8.2.1. Parallel discrete-time channels corresponding to a continuous-time channel.*

variables with mean 0 and variance  $N_o/2$ . It is also assumed that they are statistically independent of  $x(t)$ . The received waveform  $y(t)$  is the sum of  $x(t)$  and  $z(t)$  and has the representation

$$y_n = \int y(t) \varphi_n(t) dt = x_n + z_n \quad (8.2.2)$$

Equation 8.2.2 reduces the channel from a continuous time channel to an infinite set of parallel, discrete-time, additive Gaussian noise channels, as shown in Figure 8.2.1.

Suppose that the channel input is power constrained to a power  $S$  so that

$$\overline{\int_0^T x^2(t) dt} \leq ST$$

From the energy relation of (8.1.16), this is equivalent to

$$\sum_n \overline{x_n^2} \leq ST \quad (8.2.3)$$

From Theorem 7.4.2, the average mutual information on the  $n$ th channel is upper bounded by

$$I(X_n; Y_n) \leq \frac{1}{2} \log \left( 1 + \frac{2\overline{x_n^2}}{N_o} \right)$$

with equality if  $x_n$  is Gaussian with zero mean. Using the inequality  $\log(1+z) \leq z \log e$ , we then get

$$I(X_n; Y_n) \leq \frac{\overline{x_n^2}}{N_o} \log e \quad (8.2.4)$$

Also, by the same proof as in (7.2.19), the average mutual information on the first  $N$  channels is upper bounded by

$$\begin{aligned} I(\mathbf{X}^N; \mathbf{Y}^N) &\leq \sum_{n=1}^N I(X_n; Y_n) \\ &\leq \sum_{n=1}^N \frac{1}{2} \log \left( 1 + \frac{2\overline{x_n^2}}{N_o} \right) \end{aligned} \quad (8.2.5)$$

with equality if  $x_1, \dots, x_N$  are zero mean, Gaussian, and statistically independent. From (8.2.3) and (8.2.4), we see that we can pass to the limit as  $N \rightarrow \infty$ , obtaining

$$I_T[X(t); Y(t)] \leq \sum_{n=1}^{\infty} \frac{1}{2} \log \left( 1 + \frac{2\overline{x_n^2}}{N_o} \right) \quad (8.2.6)$$

$$\leq \frac{ST}{N_o} \log e \quad (8.2.7)$$

where in (8.2.7) we have used (8.2.3) and (8.2.4). Equality is obtained in (8.2.7) if the  $x_n$  are zero mean, Gaussian, and statistically independent.

Suppose now that we constrain  $x(t)$  to be a linear combination of only the first  $N$  orthonormal functions. From Theorem 7.5.1,  $I(\mathbf{X}^N; \mathbf{Y}^N)$  is maximized, subject to the constraint

$$\sum_{n=1}^N \overline{x_n^2} \leq ST,$$

by choosing  $x_1, \dots, x_N$  as independent zero mean Gaussian random variables of variance  $ST/N$  each, yielding

$$\max I(\mathbf{X}^N; \mathbf{Y}^N) = \frac{N}{2} \log \left( 1 + \frac{2ST}{N_o N} \right)$$

With this constraint,  $\overline{x_n^2} = 0$  for  $n > N$ , and thus  $I(\mathbf{X}^N; \mathbf{Y}^N) = I_T[X(t); Y(t)]$ . The channel capacity per second under this constraint is then given by

$$C = \frac{N}{2T} \log \left( 1 + \frac{2ST}{N_o N} \right) \quad (8.2.8)$$

We state this result as a theorem, using  $2WT$  for  $N$ .

**Theorem 8.2.1.** Let the output of a continuous time channel be given by the sum of the input and white Gaussian noise of spectral density  $N_o/2$ . Let the input be power constrained to a power  $S$  and let the input be constrained over some time interval  $T$  to be a linear combination of  $2WT$  orthonormal functions. Then the capacity of the channel per unit time is given by

$$C = W \log \left( 1 + \frac{S}{WN_o} \right) \quad (8.2.9)$$


---

Equation 8.2.9 is Shannon's famous formula for the capacity of a white Gaussian noise channel with a band-limited, power-limited input. We have already seen that, for  $WT$  large, there are about  $2WT$  degrees of freedom in

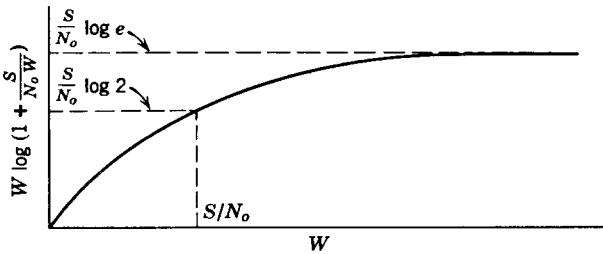


Figure 8.2.2. Capacity of a white Gaussian noise channel with  $2WT$  degrees of freedom.

the set of inputs that are time limited to  $T$  and approximately band limited to  $W$ . In section 8.5, this connection between number of degrees of freedom and bandwidth will be made precise and (8.2.9) will be established for band-limited channels rather than channels with a constrained number of degrees of freedom.

Figure 8.2.2 sketches (8.2.9) as a function of  $W$ . It is seen that  $C$  increases rapidly with  $W$  until  $W \approx S/N_o$ . Beyond this,  $C$  increases more slowly, approaching a limit of  $S/N_o \log e$  as  $W \rightarrow \infty$ . This, in conjunction with (8.2.7), yields the following corollary to Theorem 8.2.1.

**COROLLARY.** The capacity per unit time of a white Gaussian noise channel with the input power constrained to  $S$  and with the number of degrees of freedom unconstrained is given by

$$C_\infty = (S/N_o) \log e \quad (8.2.10)$$


---

To achieve capacity with a constraint of  $2WT$  degrees of freedom, the

signal energy per degree of freedom is given by  $\overline{x_n^2} = S/2W$ . Thus the point  $W = S/N_o$  in Figure 8.2.2 corresponds to a unit energy to noise ratio per degree of freedom. For  $W > S/N_o$  the signal energy per degree of freedom is less than the noise energy per degree of freedom, and as  $W \rightarrow \infty$ , the signal energy per degree of freedom approaches zero. This result seems strange and unintuitive at first, since as  $W$  increases, the signal power is being spread more thinly over more degrees of freedom and thus appears to be getting buried in the noise. We shall see later, however, that the distinguishability of any given code word from noise has nothing to do with the number of orthonormal functions used in specifying the code word. The only effect of many degrees of freedom is to allow a large separation between the different code words.

### Error Probability for Two Code Words

For a continuous time channel, the code words in a code will be functions of time (or, more generally, vector functions of time). Given an orthonormal set of functions,  $\varphi_1(t), \varphi_2(t), \dots$ , these code words can be represented as vectors. Thus the  $m$ th code word,  $x_m(t)$ , can be represented as

$$\begin{aligned} x_m(t) &= \sum_n x_{m,n} \varphi_n(t) \\ x_{m,n} &= \int x_m(t) \varphi_m(t) dt \end{aligned} \quad (8.2.11)$$

If the code words are constrained to  $N$  degrees of freedom, then the code words can be considered to have block length  $N$  and the results of Chapter 7 can be applied immediately. Here, however, it will be more instructive to start at the beginning and rederive those results for the special case where the available number of degrees of freedom is unlimited. We start with the case of two code words,  $x_1(t)$  and  $x_2(t)$ , and assume that  $x_1(t)$  and  $x_2(t)$  are linear combinations of the first  $N$  of a set of orthonormal functions

$$\begin{aligned} x_1(t) &= \sum_{n=1}^N x_{1,n} \varphi_n(t) \\ x_2(t) &= \sum_{n=1}^N x_{2,n} \varphi_n(t) \end{aligned} \quad (8.2.12)$$

White Gaussian noise of spectral density  $N_o/2$  is added to whichever signal is transmitted and the received waveform,  $y(t)$ , has components

$$y_n = \begin{cases} x_{1,n} + z_n; & n \leq N, x_1(t) \text{ sent} \\ x_{2,n} + z_n; & n \leq N, x_2(t) \text{ sent} \\ z_n & n > N \end{cases} \quad (8.2.13)$$

where the  $z_n$  are independent Gaussian random variables with mean zero and variance  $N_o/2$ . Let the amplitude scale for  $x(t)$  and  $y(t)$  be chosen so that  $N_o/2 = 1$ . Then, letting  $\mathbf{x}_1 = (x_{1,1}, x_{1,2}, \dots, x_{1,N})$ ,  $\mathbf{x}_2 = (x_{2,1}, \dots, x_{2,N})$ , and  $\mathbf{y} = (y_1, \dots, y_N)$ , the joint conditional probability density of  $\mathbf{y}$  given  $\mathbf{x}_1$  or  $\mathbf{x}_2$  is

$$\begin{aligned} p_N(\mathbf{y} \mid \mathbf{x}_1) &= \frac{1}{(2\pi)^{N/2}} \exp \left[ -\frac{1}{2} \sum_{n=1}^N (y_n - x_{1,n})^2 \right] \\ p_N(\mathbf{y} \mid \mathbf{x}_2) &= \frac{1}{(2\pi)^{N/2}} \exp \left[ -\frac{1}{2} \sum_{n=1}^N (y_n - x_{2,n})^2 \right] \end{aligned} \quad (8.2.14)$$

Define the log likelihood ratio  $r_{1,2}(\mathbf{y})$  as

$$r_{1,2}(\mathbf{y}) = \ln \frac{p_N(\mathbf{y} \mid \mathbf{x}_1)}{p_N(\mathbf{y} \mid \mathbf{x}_2)} \quad (8.2.15)$$

$$= -\frac{1}{2} \sum_{n=1}^N (y_n - x_{1,n})^2 + \frac{1}{2} \sum_{n=1}^N (y_n - x_{2,n})^2 \quad (8.2.16)$$

$$= \sum_{n=1}^N y_n x_{1,n} - \sum_{n=1}^N y_n x_{2,n} - \frac{1}{2} \sum_{n=1}^N x_{1,n}^2 + \frac{1}{2} \sum_{n=1}^N x_{2,n}^2 \quad (8.2.17)$$

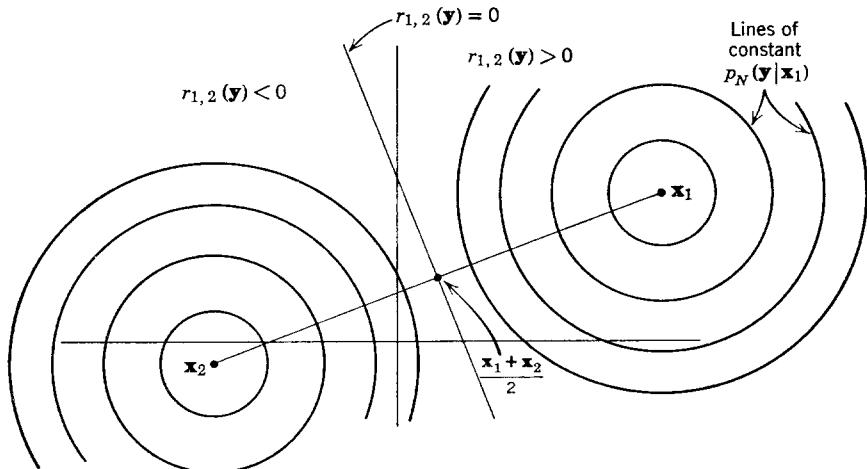
The log likelihood ratio has much the same significance here as in Chapter 5. For maximum-likelihood decoding, we decode message 1 when  $r_{1,2}(\mathbf{y}) > 0$  and message 2 otherwise. For minimum-error probability decoding with a priori probabilities  $q_1$  and  $q_2$ , we decode message 1 when  $r_{1,2}(\mathbf{y}) > \ln(q_2/q_1)$ . Observe that we have omitted  $y_n$  for  $n > N$  from consideration since these quantities are independent of the transmitted message and cancel out of the log likelihood ratio. Even if  $N$  is infinite in (8.2.12),  $r_{1,2}(\mathbf{y})$  is well defined although the limit of the conditional probabilities in (8.2.14) does not exist.

The log likelihood ratio can be calculated quite easily from the received waveform by noting that (8.2.17) can be rewritten with Parseval's equation as

$$r_{1,2}(\mathbf{y}) = \int y(t)x_1(t) dt - \int y(t)x_2(t) dt - \frac{1}{2} \int x_1^2(t) dt + \frac{1}{2} \int x_2^2(t) dt \quad (8.2.18)$$

Thus the only operations that need to be performed on  $y(t)$  are to multiply it by  $x_1(t)$  and  $x_2(t)$  and integrate; this is called correlation decoding. An equivalent way to perform the same operation is to construct filters with impulse responses  $h_1(t) = x_1(T-t)$  and  $h_2(t) = x_2(T-t)$ . These are called matched filters and when  $y(t)$  is passed through them, the output at  $t = T$  is seen to be the same as above; this is called matched-filter decoding.

From (8.2.17), it is seen that  $r_{1,2}(\mathbf{y})$  is linearly related to  $\mathbf{y}$  and is simply a constant plus the projection of  $\mathbf{y}$  on  $\mathbf{x}_1 - \mathbf{x}_2$ . Thus  $r_{1,2}(\mathbf{y})$  is constant along



**Figure 8.2.3.** Geometric interpretation of two code words on white Gaussian noise channel.

any hyperplane perpendicular to the line joining  $\mathbf{x}_1$  and  $\mathbf{x}_2$ , and  $r_{1,2}(\mathbf{y}) = 0$  for  $\mathbf{y} = \frac{1}{2}[\mathbf{x}_1 + \mathbf{x}_2]$ , as can be verified easily from (8.2.16) (see Figure 8.2.3).

The probability of error using maximum-likelihood decoding can now be calculated. If message 1 is sent, we can use  $x_{1,n} + z_n$  for  $y_n$ , getting

$$r_{1,2}(\mathbf{y}) = \sum_n z_n (x_{1,n} - x_{2,n}) + \frac{1}{2} \sum_n (x_{1,n} - x_{2,n})^2 \quad (8.2.19)$$

Thus  $r_{1,2}(\mathbf{y})$  is a Gaussian random variable with mean

$$\frac{1}{2} \sum_n (x_{1,n} - x_{2,n})^2$$

and variance

$$\sum_n (x_{1,n} - x_{2,n})^2.$$

The probability of error is the probability that  $r_{1,2}(\mathbf{y}) < 0$ , or the probability that it be more than  $\frac{1}{2}\sqrt{\sum_n (x_{1,n} - x_{2,n})^2}$  standard deviations below the mean,

$$P_{e,1} = \Phi\left[-\frac{1}{2}\sqrt{\sum_n (x_{1,n} - x_{2,n})^2}\right] \quad (8.2.20)$$

where  $\Phi$  is the distribution function of a normalized Gaussian random variable,

$$\Phi(u) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^u \exp(-v^2/2) dv \quad (8.2.21)$$

Since the error probability when message 2 is sent is clearly the same, the overall error probability is given by

$$P_e = \Phi \left\{ -\frac{1}{2} \sqrt{\int [x_1(t) - x_2(t)]^2 dt} \right\} \quad (8.2.22)$$

where we have used the energy equation (8.1.16). It should be emphasized that  $P_e$  depends only on the energy of the difference  $x_1(t) - x_2(t)$  and not on the detailed choice of waveforms.

Next, consider the important situation where both code words have the same energy

$$\int x_1^2(t) dt = \int x_2^2(t) dt = \frac{2E}{N_o} \quad (8.2.23)$$

We are using an amplitude scale here where  $N_o/2 = 1$ , but we can interpret  $E$  in (8.2.23) as the code word energy with an arbitrary amplitude scale and  $N_o/2$  as the noise spectral density with the same scale. Letting

$$\lambda = \frac{N_o}{2E} \int x_1(t)x_2(t) dt$$

be the normalized correlation between the code words, we can rewrite (8.2.22) as

$$P_e = \Phi \left[ -\sqrt{\frac{(1-\lambda)E}{N_o}} \right] \quad (8.2.24)$$

The error probability is minimized with respect to  $\lambda$  by picking  $x_2(t) = -x_1(t)$  in which case  $\lambda = -1$ . Any decrease in error probability beyond this point requires increasing  $E$  which, for a fixed signal power, requires increasing the time required to transmit one bit. The other alternative, which we now investigate, is to increase the number of code words,  $M$ . This allows us to increase the duration of the code words (and thus  $E$ ) without decreasing the transmission rate  $R$ .

When  $M$  is large, there is a problem in choosing the code words. From the two code-word analysis, it is clear that the difference energy,  $\int [x_m(t) - x_{m'}(t)]^2 dt$ , should be large for all  $m \neq m'$ . We can get an idea about the possible magnitudes of these difference energies by bounding the average difference energy over  $m$  and  $m'$  with the constraint that

$$\frac{I}{M} \sum_m \int x_m^2(t) dt \leq \frac{2E}{N_o}$$

The average difference energy then satisfies

$$\begin{aligned} \frac{1}{M(M-1)} \sum_m \sum_{m' \neq m} \int [x_m(t) - x_{m'}(t)]^2 dt \\ = \frac{1}{M(M-1)} \sum_m \sum_{m'} \int [x_m(t) - x_{m'}(t)]^2 dt \quad (8.2.25) \end{aligned}$$

$$\begin{aligned} &= \frac{1}{M(M-1)} \sum_{m,m'} \int [x_m^2(t) + x_{m'}^2(t)] dt \\ &\quad - \frac{2}{M(M-1)} \int \sum_m x_m(t) \sum_{m'} x_{m'}(t) dt \quad (8.2.26) \end{aligned}$$

$$\leq \frac{M}{M-1} \frac{4E}{N_o} \quad (8.2.27)$$

Equation 8.2.27 follows from (8.2.26) by neglecting the second term of (8.2.26), which is always negative, and using the energy constraint on  $x_m(t)$ . When  $M = 2$ , this bound is  $8E/N_o$  and is equal to the difference energy for  $x_{m'}(t) = -x_m(t)$ . As  $M \rightarrow \infty$ , the bound approaches  $4E/N_o$ , which is easily seen to be the difference energy for orthogonal code words of energy  $2E/N_o$ . Thus the implication of (8.2.27) is that, for large  $M$ , there must be a large number of pairs of code words for which the difference energy is, at most, little more than that for orthogonal code words.

In the remainder of this section, we shall compute upper and lower bounds on the error probability for a set of equal energy orthogonal code words. Before doing this, however, we relate orthogonal codes to another well-known class of codes, the simplex codes. Let  $x_1(t), \dots, x_M(t)$  be a set of equal energy orthogonal waveforms, and define the code words of the associated simplex code by

$$\xi_m(t) = x_m(t) - \frac{1}{M} \sum_{m'} x_{m'}(t); \quad 1 \leq m \leq M \quad (8.2.28)$$

Geometrically, the  $\xi_m(t)$  can be interpreted as the vertices of an  $M-1$  dimensional equilateral simplex centered on the origin. Since the code words of a simplex code are simply translations of the code words of the associated orthogonal code, we see that they have the same probability of decoding error. However, the energy of the simplex code words is smaller than that of the orthogonal words by a factor of  $(M-1)/M$ . It is certainly plausible intuitively that the simplex codes yield the minimum possible error probability on a white Gaussian noise channel for a given  $M$  and  $N$ . A rigorous proof of this, however, is yet to be found.

### Error Probability for Orthogonal Code Words

Let  $x_1(t), \dots, x_M(t)$  be orthogonal signals each of energy  $A^2 = 2E/N_o$ , where, as in the two code-word case, we are scaling the amplitude to normalize the noise. The noise is assumed to be additive, white, and Gaussian. Define the orthonormal set  $\varphi_1(t), \dots, \varphi_M(t)$  by

$$\varphi_m(t) = \frac{x_m(t)}{A} \quad (8.2.29)$$

Then  $x_m(t)$  has the representation  $\mathbf{x}_m = (0, \dots, 0, A, 0, \dots, 0)$  where the  $A$  is in position  $m$ . Let  $y(t)$  be the received waveform, and let  $y_m = \int y(t)\varphi_m(t) dt$ . If message  $m$  is transmitted,  $y_m = A + z_m$ , and for  $m' \neq m$   $y_{m'} = z_{m'}$ , where the  $z_m$  are independent, normalized, Gaussian random variables. Defining  $\mathbf{y} = (y_1, \dots, y_M)$ , we then have the conditional joint probability density

$$p(\mathbf{y} | \mathbf{x}_m) = \left(\frac{1}{2\pi}\right)^{M/2} \exp\left[-\frac{(y_m - A)^2}{2} - \sum_{m' \neq m} \frac{y_{m'}^2}{2}\right] \quad (8.2.30)$$

$$p(\mathbf{y} | \mathbf{x}_m) = \left(\frac{1}{2\pi}\right)^{M/2} \exp\left[y_m A - \frac{A^2}{2} - \sum_{m'} \frac{y_{m'}^2}{2}\right] \quad (8.2.31)$$

Assume that maximum-likelihood decoding is used. From (8.2.31), the decoding rule is to pick the  $m$  for which  $y_m$  is largest. Then, when message  $m$  is sent, the error probability is the probability that  $y_{m'} \geq y_m$  for some  $m' \neq m$ ,  $1 \leq m' \leq M$ . We can write this as

$$P_{e,m} = \int_{-\infty}^{\infty} \frac{1}{\sqrt{2\pi}} \exp\left[\frac{-(y_m - A)^2}{2}\right] Q(y_m) dy_m \quad (8.2.32)$$

where, for a given value of  $y_m$ ,  $Q(y_m)$  is the probability, that for some  $m' \neq m$ ,  $y_{m'} \geq y_m$ . This is 1 minus the probability that  $y_{m'} < y_m$  for all  $m'$ , and since all the  $y_{m'}$  are independent normalized Gaussian random variables, we have

$$\begin{aligned} Q(y_m) &= 1 - [\Phi(y_m)]^{M-1} \\ &= 1 - [1 - \Phi(-Y_m)]^{M-1} \end{aligned} \quad (8.2.33)$$

Equations 8.2.32 and 8.2.33 yield an exact expression for  $P_{e,m}$  which has been tabulated by Viterbi (1961) for a variety of values of  $A$  and  $M$ . Our major concern here, however, is to find simple, interpretable bounds on  $P_{e,m}$ .

As a very simple initial bound, observe that for a given  $m$  and  $m'$ , the probability that  $y_{m'} \geq y_m$  when  $m$  is transmitted is simply the probability of error for two code words given in (8.2.24) with  $\lambda = 0$ . Thus, using the union bound on the  $M - 1$  choices for  $m'$ ,

$$P_{e,m} \leq (M - 1)\Phi(-A/\sqrt{2}) \quad (8.2.34)$$

As an alternate technique, which will turn out to be better when  $M$  is large, we can use the union bound on  $Q(y_m)$  to get

$$Q(y_m) \leq (M - 1)\Phi(-y_m) \quad (8.2.35)$$

For  $y_m$  small, the right side of (8.2.35) is larger than 1, although  $Q(y_m)$  is always upper bounded by 1. Define  $y_o$  by

$$M \exp\left(\frac{-y_o^2}{2}\right) = 1 \quad (8.2.36)$$

For large  $M$ ,  $y_o$  is a simple approximation to the value of  $y_m$  for which  $(M - 1)\Phi(-y_m) = 1$ . Therefore, we use (8.2.35) for  $y_m > y_o$ , and  $Q(y_m) \leq 1$  for  $y_m \leq y_o$ . Then (8.2.32) becomes

$$\begin{aligned} P_{e,m} \leq & \int_{-\infty}^{y_o} \frac{1}{\sqrt{2\pi}} \exp\left[\frac{-(y_m - A)^2}{2}\right] dy_m + \\ & + (M - 1) \int_{y_o}^{\infty} \frac{1}{\sqrt{2\pi}} \exp\left[\frac{-(y_m - A)^2}{2}\right] \Phi(-y_m) dy_m \end{aligned} \quad (8.2.37)$$

We now bound  $\Phi(-y_m)$ , using the following standard inequality\* on the Gaussian distribution function for  $y_m > 0$ :

$$\left(\frac{1}{y_m} - \frac{1}{y_m^3}\right) \frac{\exp(-y_m^2/2)}{\sqrt{2\pi}} < \Phi(-y_m) < \frac{1}{y_m \sqrt{2\pi}} \exp(-y_m^2/2) \quad (8.2.38)$$

Substituting the right side of (8.2.38) into (8.2.37) and bounding  $1/y_m$  in the integral by  $1/y_o$ , we have

$$P_{e,m} \leq \Phi(y_o - A) + \frac{(M - 1)}{y_o \sqrt{2\pi}} \int_y^{\infty} \frac{1}{\sqrt{2\pi}} \exp\left[-\frac{(y_m - A)^2}{2} - \frac{y_m^2}{2}\right] dy_m$$

Completing the square in the integral, and upper bounding  $M - 1$  by  $\exp y_o^2/2$ , we obtain

$$P_{e,m} \leq \Phi(y_o - A) + \frac{\exp[y_o^2/2 - A^2/4]}{\sqrt{4\pi y_o}} \Phi\left[\sqrt{2}\left(\frac{A}{2} - y_o\right)\right] \quad (8.2.39)$$

If  $A/2 < y_o < A$ , we can use (8.2.38) on both terms in (8.2.39). The exponential part of each term is the same and we get

$$P_{e,m} \leq \frac{\exp[-(A - y_o)^2/2]}{\sqrt{2\pi}} \left[ \frac{1}{A - y_o} + \frac{1}{\sqrt{2\pi} y_o (2y_o - A)} \right] \quad (8.2.40)$$

\* See Feller (1950), Chap VII section 1.

Now let the signals have duration  $T$  and power  $S = E/T$ . Then

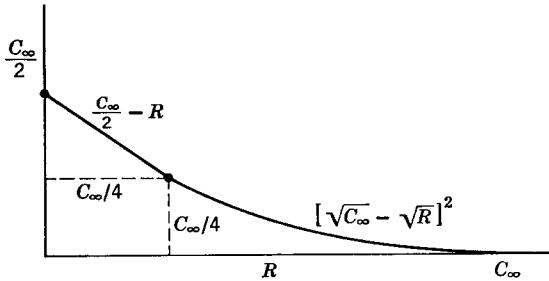
$$A = \sqrt{2TS/N_o} = \sqrt{2TC_\infty} \quad (8.2.41)$$

where  $C_\infty$  is the channel capacity in natural units per second. Also from (8.2.36)

$$y_o = \sqrt{2 \ln M} = \sqrt{2RT} \quad (8.2.42)$$

where  $R$  is the rate in natural units per second. Thus (8.2.40) is valid for  $C_\infty/4 < R < C_\infty$ , and with the above substitutions (8.2.40) becomes

$$P_{e,m} \leq \frac{\exp [-T(\sqrt{C_\infty} - \sqrt{R})^2]}{\sqrt{4\pi T}} \left[ \frac{1}{\sqrt{C_\infty} - \sqrt{R}} + \frac{1}{\sqrt{4\pi TR(2\sqrt{R} - \sqrt{C_\infty})}} \right] \quad (8.2.43)$$



*Figure 8.2.4. Exponent, rate curve for a white Gaussian noise channel.*

For  $R \leq C_\infty/4$ , we use (8.2.34). After bounding  $M - 1$  by  $e^{RT}$  and using (8.2.38), (8.2.34) becomes

$$P_{e,m} \leq \frac{\exp \left[ -T \left( \frac{C_\infty}{2} - R \right) \right]}{\sqrt{2\pi TC_\infty}} \quad (8.2.44)$$

Equations 8.2.43 and 8.2.44 show that for any given  $R < C_\infty$ , the error probability goes to zero at least exponentially with  $T$ . The exponent [that is,  $(\sqrt{C_\infty} - \sqrt{R})^2$  for  $C_\infty/4 < R < C_\infty$ , and  $C_\infty/2 - R$  for  $R \leq C_\infty/4$ ] is sketched in Figure 8.2.4. It is seen that it is the same as the exponent-rate curve for very noisy channels. This is not surprising in view of the fact that, as  $T$  becomes large, the number of discrete-time channels required to represent the code words is growing exponentially and the average energy to noise ratio per degree of freedom is approaching 0.

We can compare these results with those of Chapter 7, which correspond

to the same channel with a constraint on the number of degrees of freedom per second available. We see that the loss in exponent incurred by the constraint is small so long as the number of degrees of freedom is sufficient to make the energy to noise ratio per degree of freedom less than 1.

We now lower bound  $P_{e,m}$ . Let  $y$  be an arbitrary number to be chosen later. Using (8.2.32) and recognizing that  $Q(y_m)$  is decreasing with  $y_m$ , we have

$$P_{e,m} \geq \int_{-\infty}^y \frac{1}{\sqrt{2\pi}} \exp \left[ -\frac{(y_m - A)^2}{2} \right] Q(y_m) dy_m \quad (8.2.45)$$

$$\geq Q(y)\Phi(y - A) \quad (8.2.46)$$

In other words, we are lower bounding  $P_{e,m}$  by counting only errors for which  $y_m < y$  and  $y_{m'} \geq y$  for some  $m'$ . Using the binomial expansion for  $Q(y)$  as given in (8.2.33), we obtain

$$Q(y) = (M - 1)\Phi(-y) - \binom{M - 1}{2}\Phi^2(-y) + \dots$$

This is an alternating series and the first two terms constitute a lower bound to  $Q(y)$ . This follows because the terms are either decreasing in magnitude or else the bound is negative. For  $y \geq y_o$ , we can further bound  $Q(y)$  as follows:

$$\begin{aligned} Q(y) &\geq (M - 1)\Phi(-y) \left[ 1 - \frac{M - 2}{2}\Phi(-y) \right] \\ &\geq (M - 1)\Phi(-y) \left[ 1 - \frac{M - 2}{2\sqrt{2\pi}y} e^{-y^2/2} \right] \\ &\geq (M - 1)\Phi(-y) \left[ 1 - \frac{1}{2\sqrt{2\pi}y} \right]; \quad y \geq y_o \end{aligned} \quad (8.2.47)$$

We have used (8.2.38) to lower bound  $-\Phi(-y)$  and then used  $M = \exp(y_o^2/2)$  and  $y \geq y_o$ . Substituting (8.2.47) into (8.2.46), using  $M = \exp(y_o^2/2)$  again, and lower bounding  $\Phi$  by (8.2.38), we obtain

$$\begin{aligned} P_{e,m} &\geq \left(1 - \frac{1}{M}\right) \left(1 - \frac{1}{2\sqrt{2\pi}y}\right) \left(\frac{1}{y} - \frac{1}{y^3}\right) \left[\frac{1}{A - y} - \frac{1}{(A - y)^3}\right] \frac{1}{2\pi} \\ &\quad \times \exp \left[ \frac{y_o^2}{2} - \frac{y^2}{2} - \frac{(A - y)^2}{2} \right] \end{aligned} \quad (8.2.48)$$

Equation 8.2.48 is valid for any  $y$  between  $y_o$  and  $A$ . The exponential term is maximized for  $A/2 < y_o < A$  by  $y = y_o$  and for  $y_o \leq A/2$  by  $y = A/2$ .

Using  $A = \sqrt{2TC_\infty}$  and  $y_o = \sqrt{2TR}$ , (8.2.48) becomes

$$\begin{aligned} P_{e,m} &\geq \left(1 - \frac{1}{M}\right) \left(1 - \frac{1}{4\sqrt{\pi RT}}\right) \left(1 - \frac{1}{2RT}\right) \left[1 - \frac{1}{2T(\sqrt{C_\infty} - \sqrt{R})^2}\right] \\ &\quad \times \frac{\exp[-T(\sqrt{C_\infty} - \sqrt{R})^2]}{4\pi T[\sqrt{RC_\infty} - R]} \quad \text{for } \frac{C_\infty}{4} < R < C_\infty \end{aligned} \quad (8.2.49)$$

$$\begin{aligned} P_{e,m} &\geq \left(1 - \frac{1}{M}\right) \left(1 - \frac{1}{2\sqrt{\pi TC_\infty}}\right) \left(1 - \frac{2}{TC_\infty}\right)^2 \frac{\exp\left[-T\left(\frac{C_\infty}{2} - R\right)\right]}{\pi C_\infty T} \\ &\quad \text{for } R \leq \frac{C_\infty}{4} \end{aligned} \quad (8.2.50)$$

We see that the exponents in this lower bound agree with the upper bound exponents for every  $R < C_\infty$ .

Since simplex codes have the same error probability as orthogonal codes with an energy only  $(M - 1)/M$  as great, we can apply the bounds in (8.2.43), (8.2.44), (8.2.49), and (8.2.50) to the simplex codes simply by replacing  $C_\infty$  in each equation with  $C_\infty[M/(M - 1)]^{1/2}$ .

### 8.3 Heuristic Treatment of Capacity for Channels with Additive Gaussian Noise and Bandwidth Constraints

In the previous sections, we have shown how to represent signal and noise in terms of orthonormal expansions and have used this to find the capacity of a power-limited channel with additive white Gaussian noise. We then considered the resulting probability of decoding error when the set of code words was a set of orthogonal waveforms.

There are two very disturbing things about that analysis. First, to make the error probability small at rates close to capacity, we must use an enormous number of orthogonal waveforms and this requires an enormous bandwidth. Second, we justified the use of white Gaussian noise, as a model for physical noise, on the basis of its being a reasonable and simple approximation over the range of frequencies of interest. On the other hand, as we use more and more orthogonal waveforms in coding, eventually we must exceed the range of frequencies over which the assumption of white Gaussian noise makes any sense. In fact, if we take the viewpoint that the total power in the received noise is finite, then the spectral density of the noise must go to zero with increasing frequency, and the average mutual information on the channel can be made arbitrarily large by putting the input at arbitrarily large frequencies.

Physically, the above argument is not really valid, since some of the additive noise comes from the receiver, and as we raise the frequency of the input waveforms, we must modify the receiver to receive these higher frequencies; this, in turn, generates additional noise at these higher frequencies.

Physical arguments like this do not provide an entirely satisfactory way out of the dilemma, however. The real difficulty is that the model of white Gaussian noise and of a signal unconstrained in frequency is a very unstable model. The results that we get are very dependent on what happens at frequencies approaching infinity.

One commonly attempted device to avoid these difficulties is to arbitrarily restrict the signal to contain no frequencies greater than some maximum frequency  $W$ . The sampling theorem can then be used to represent the input, and since there are  $2W$  samples per second, we can reason from (8.2.9) that the capacity is  $W \log [1 + S/(N_0 W)]$ . There are some subtle mathematical difficulties with this approach, however, one of them being that the definition of capacity given in Chapter 4 does not apply here since a bandlimited signal cannot also be strictly time limited. We shall return later to resolve these difficulties and make the above result precise.

Another commonly attempted way to avoid worrying about arbitrarily large frequencies is to assume that the noise spectral density is increasing with frequency as  $f \rightarrow \infty$ . This approach is unsatisfying both on physical grounds and on mathematical grounds.

The approach that we shall take here is to assume that the signal is first constrained in power, and then constrained in frequency by being sent through a linear time-invariant filter before transmission. Thus, if we attempt to use very high frequency signals, the filter will attenuate those frequencies far below the spectral density of the noise.

Mathematically, this approach has the advantage of being perfectly well defined and capable of precise analysis. Physically, it has the advantage of being much closer to the kind of frequency constraint imposed on real communication systems than the other approaches. One final advantage of this approach is that, by varying the filter response and the noise spectral density, we can find out a great deal about the stability of capacity and of the error exponents with respect to small changes in the model. In fact, by taking this approach, we shall be able to rigorously derive capacity for a strictly band-limited input and to see how much stability that result has.

In this section, we shall give an heuristic derivation of channel capacity for a filtered input and additive nonwhite Gaussian noise. The derivation is quite simple and quite convincing. On the other hand, it is not rigorous and it contains some small holes that cannot be satisfactorily plugged up. The next two sections will be devoted to deriving the same result rigorously by a different method.

The situation that we wish to analyze is depicted in Figure 8.3.1. The channel input,  $x(t)$ , is zero outside the interval  $(-T/2, T/2)$  and is power limited to  $S$  in the sense that  $x(t)$  must be chosen from an ensemble for which

$$\overline{\int_{-T/2}^{T/2} x^2(t) dt} \leq ST \quad (8.3.1)$$

The input is passed through a filter with frequency characteristic  $H_1(f)$ , and the filter output is denoted by  $u(t)$ . The noise consists of a sample function,  $z(t)$ , of stationary Gaussian noise of spectral density  $N(f)$ , which is added to  $u(t)$ . The channel output,  $y(t)$ , is  $u(t) + z(t)$ , truncated to the interval  $(-T/2, T/2)$ . We wish to see how large the average mutual information per second between  $x(t)$  and  $y(t)$  can be made.

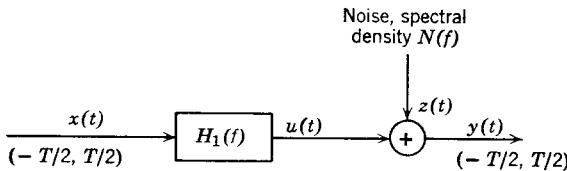


Figure 8.3.1. Filtered signal plus noise.

Let us represent  $x(t)$  in a Fourier series:

$$x(t) = \sum_{i=-\infty}^{\infty} x_i \phi_i(t)$$

$$\phi_i(t) = \begin{cases} \sqrt{2/T} \cos \frac{2\pi i t}{T}; & i > 0 \\ \sqrt{1/T} ; & i = 0 \\ \sqrt{2/T} \sin \frac{2\pi i t}{T}; & i < 0 \end{cases} \quad |t| \leq T/2 \quad (8.3.2)$$

$$\phi_i(t) = 0; \quad |t| > T/2$$

The response of the filter to  $\phi_i(t)$  is given by  $\int \phi_i(\tau) h_1(t - \tau) d\tau$  where  $h_1(t)$  is the impulse response of the filter and the inverse Fourier transform of  $H_1(f)$ . If  $T$  is very much greater than the effective duration of the impulse response of the filter, we expect the response of the filter to  $\phi_i(\tau)$  to be approximately a sinusoid of frequency  $|i|/T$  and duration  $(-T/2, T/2)$ . We also expect this response to be attenuated from  $\phi_i(\tau)$ , roughly by a factor  $|H_1(i/T)|$ . This motivates us to define a set of functions  $\theta_i(t)$  by

$$\theta_i(t) = \frac{1}{|H_1(i/T)|} \int \phi_i(\tau) h_1(t - \tau) d\tau \quad (8.3.3)$$

From the above discussion, we expect  $\theta_i(t)$  to be roughly equal to  $\phi_i(t)$  except for a phase shift. That is,

$$\theta_i(t) \approx \alpha\phi_i(t) + \beta\phi_{-i}(t); \quad \alpha^2 + \beta^2 = 1 \quad (8.3.4)$$

What is more, it can be seen from (8.3.3) that the phase shift between  $\theta_i(t)$  and  $\phi_i(t)$  is roughly the same as the phase shift between  $\theta_{-i}(t)$  and  $\phi_{-i}(t)$ , and therefore  $\theta_i(t)$  and  $\theta_{-i}(t)$  are approximately orthogonal. Also, from (8.3.4),  $\theta_i(t)$  is approximately orthogonal to  $\theta_j(t)$  for each  $j \neq i$ . Thus, to an approximation that should improve with increasing  $T$ , the  $\theta_i(t)$  can be roughly considered as an orthonormal set.

Next we calculate  $u(t)$  in terms of the  $\theta_i(t)$ :

$$\begin{aligned} u(t) &= \int x(\tau)h_1(t - \tau) d\tau \\ &= \int \sum_i x_i \phi_i(\tau) h_1(t - \tau) d\tau \end{aligned}$$

Using (8.3.3), this becomes

$$\begin{aligned} u(t) &= \sum u_i \theta_i(t) \\ u_i &= x_i |H_1(i/T)| \end{aligned} \quad (8.3.5)$$

The noise process,  $z(t)$ , can also be expanded in terms of the  $\theta_i(t)$  to give

$$z_i = \int z(t) \theta_i(t) dt \quad (8.3.6)$$

For the time being, we can regard the noise as resulting from passing white Gaussian noise of unit spectral density through a nonrealizable filter with a frequency characteristic given by  $\sqrt{N(f)}$ .

Thus, if we let  $n(\tau)$  be the white Gaussian noise,

$$z_i = \int z(t) \theta_i(t) dt = n(\tau) g(t) \int -\tau \theta_i(t) dt d\tau$$

where

$$g(t) = \int \sqrt{N(f)} e^{j2\pi ft} df$$

Defining

$$\psi_i(\tau) = \int g(t - \tau) \theta_i(t) dt \quad (8.3.7)$$

this becomes

$$z_i = \int n(\tau) \psi_i(\tau) d\tau \quad (8.3.8)$$

From the same argument as before, each  $\psi_i(\tau)$  is approximately sinusoidal, the set  $\{\psi_i(\tau)\}$  is approximately orthogonal, and

$$\int \psi_i^2(\tau) d\tau \approx N(i/T) \quad (8.3.9)$$

Using (8.1.41), we then have

$$\bar{z}_i z_j \approx N(i/T) \delta_{ij} \quad (8.3.10)$$

Adding  $u(t)$  and  $z(t)$  together, the received signal over the interval  $(-T/2, T/2)$ , is given approximately by

$$y(t) \approx \sum_i y_i \theta_i(t) \quad (8.3.11)$$

$$y_i = x_i |H_1(i/T)| + z_i \quad (8.3.12)$$

The  $y_i$  can be thought of as outputs from a set of parallel, discrete-time, additive Gaussian noise channels. Essentially, we have divided the channel into frequency slots, the width of each slot being  $1/T$ . Each slot has two degrees of freedom, corresponding to sine and cosine.

All of the above statements could be made quite a bit more carefully, but there is a basic flaw in the argument that seems quite difficult to overcome. As  $T$  gets large, the number of parallel channels per unit frequency increases. Thus, even though the noise on any two channels becomes statistically independent as  $T \rightarrow \infty$ , it is not clear that any one channel becomes statistically independent of the *set* of all other channels. In making the argument precise, it is easier to give up the Fourier series approach and use a different set of orthonormal functions; that will be done in the next two sections.

In the remainder of this section, however, we shall pretend that (8.3.10) is satisfied with strict equality and that (8.3.12) represents a set of parallel (independent) channels. We can use the results of Chapter 7 now to find the capacity of this parallel combination.

Using Bessel's inequality, the power constraint (8.3.1) becomes

$$\sum_i \bar{x}_i^2 \leq ST \quad (8.3.13)$$

If we regard the output of the  $i$ th channel as  $y_i/|H_1(i/T)|$ , then this output is  $x_i$  plus an independent Gaussian random variable of variance  $N(i/T)/|H_1(i/T)|^2$ . From Theorem 7.5.1, the capacity of the parallel combination (normalized to capacity per unit time) is

$$C_T = \sum_{i \in I_B} \frac{1}{2T} \log \frac{|H_1(i/T)|^2 B}{N(i/T)} \quad (8.3.14)$$

where  $I_B$  is the set of  $i$  for which  $N(i/T)/|H_1(i/T)|^2 \leq B$  and  $B$  is the solution to

$$S = \frac{1}{T} \sum_{i \in I_B} \left[ B - \frac{N(i/T)}{|H_1(i/T)|^2} \right] \quad (8.3.15)$$

To achieve capacity, the amount of energy that must be used in each channel is given by

$$\overline{x_i^2} = \begin{cases} B - \frac{N(i/T)}{|H_1(i/T)|^2}; & i \in I_B \\ 0; & i \notin I_B \end{cases} \quad (8.3.16)$$

The interpretation of this result is the same as that of Theorem 7.5.1 (see Fig. 7.5.1).

Now, we can let  $T \rightarrow \infty$ , and in the limit, (8.3.14) and (8.3.15) become Riemann integrals

$$C = \lim_{T \rightarrow \infty} C_T = \int_{f \in F_B} \frac{1}{2} \log \left[ \frac{|H_1(f)|^2 B}{N(f)} \right] df \quad (8.3.17)$$

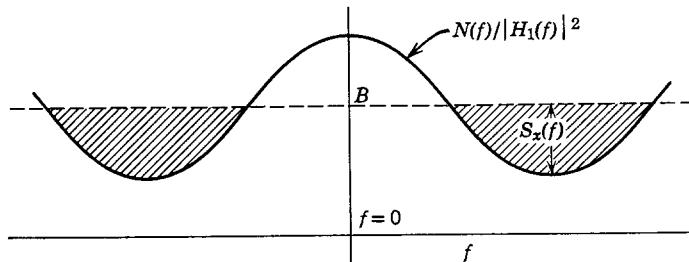


Figure 8.3.2. Interpretation of input power for capacity.

where  $F_B$  is the range of  $f$  for which  $N(f)/|H_1(f)|^2 \leq B$ , and  $B$  is the solution to

$$S = \int_{f \in F_B} \left[ B - \frac{N(f)}{|H_1(f)|^2} \right] df \quad (8.3.18)$$

The power spectral density for the input ensemble that achieves capacity is given from (8.3.16) as

$$S_x(f) = \begin{cases} B - \frac{N(f)}{|H_1(f)|^2}; & f \in F_B \\ 0; & f \notin F_B \end{cases} \quad (8.3.19)$$

We shall prove, in Section 8.5, that under certain minor constraints on  $N(f)$  and  $H_1(f)$ , (8.3.17) to (8.3.19) are indeed correct.

The interpretation of these equations is almost identical to the interpretation of Theorem 7.5.1, and is given in Figure 8.3.2.

Comparing Figure 8.3.2 with (8.3.17) to (8.3.19) we see that the power  $S$  is given by the total area of the shaded regions in Figure 8.3.2, and that the

appropriate power spectral density is given by the height of the shaded region at any given  $f$ . This is generally called a water-filling interpretation, since we can think of  $N(f)/|H_1(f)|^2$  as being the bottom of a container of unit depth and of pouring in an amount of water  $S$ . Assuming the region to be connected, we see that the water (power) will distribute itself in such a way as to achieve capacity. The capacity is proportional to the integral (over the shaded portion of  $f$ ) of the log of the ratio of the water level  $B$  to the container bottom  $N(f)/|H_1(f)|^2$ .

One of the interesting features of this result is that channel capacity is independent of the value of  $N(f)/|H_1(f)|^2$  for frequencies outside the shaded region so long as  $N(f)/|H_1(f)|^2$  stays larger than  $B$ . In other words,  $C$  is independent of the detailed behavior of  $N(f)$  and  $|H_1(f)|^2$  as  $f \rightarrow \infty$ . An exception to this occurs if  $N(f)$  approaches 0 with increasing  $f$  more quickly than  $|H_1(f)|^2$  does. In this case, the capacity is infinite and any desired mutual information can be achieved by transmitting at sufficiently high frequencies. Such a situation of course indicates that the mathematical model does not represent the fundamental limitations of the physical situation.

We can now apply these results to the case of a bandlimited signal in white Gaussian noise of spectral density  $N(f) = N_o/2$ . If the input is limited to a bandwidth  $W$  around some center frequency  $f_c$ , we can represent the limitation by

$$H_1(f) = \begin{cases} 1 & 0 \leq f_c - \frac{W}{2} \leq |f| \leq f_c + \frac{W}{2} \\ 0 & \text{elsewhere} \end{cases} \quad (8.3.20)$$

In this case,  $N(f)/|H_1(f)|^2$  is either  $N_o/2$  or  $\infty$  depending upon whether  $f$  is within the band or not. The integrands in (8.3.17) and (8.3.18) are independent of  $f$  for  $f$  within the band, and  $F_B$  must be the whole range of frequencies within the band. Thus, integrating, we get

$$C = W \log \frac{2B}{N_o} \quad (8.3.21)$$

$$S = 2W \left[ B - \frac{N_o}{2} \right] \quad (8.3.22)$$

Solving (8.3.22) for  $B$  and substituting the solution into (8.3.21), we have

$$C = W \log \left( 1 + \frac{S}{N_o W} \right) \quad (8.3.23)$$

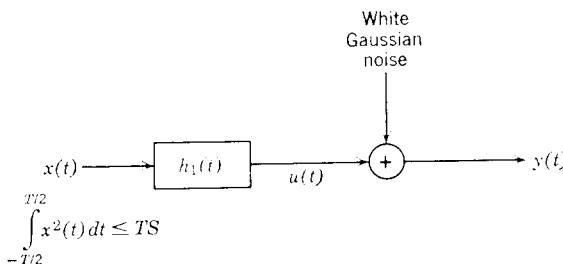
This is Shannon's famous theorem on the capacity of a bandlimited channel. It has been widely misused, mostly through a failure to recognize that it applies only to additive Gaussian noise. On the other hand, for frequencies

outside the band where  $H_1(f) = 1$ , it makes no difference what the noise spectral density is, since  $N(f)/|H_1(f)|^2 = \infty$  for any nonzero  $N(f)$ .

There have been a number of mathematical paradoxes published in the literature, however, concerning (8.3.23), and in these it is assumed that  $N(f) = 0$  outside the band. In this case,  $N(f)/|H_1(f)|^2$  is indeterminate outside the band and the capacity can be made arbitrarily large by assigning an arbitrarily small value to  $N(f)/|H_1(f)|^2$  outside the band. Physically, of course, the problem can be easily resolved by observing that  $N(f)$  cannot be strictly zero. To put it another way, when an analysis of a mathematical model of a physical problem is indeterminate, it means that the model has been over-idealized and the model must be changed.

#### 8.4 Representation of Linear Filters and Nonwhite Noise

In developing a precise treatment of signals that are both power and frequency limited, we shall start with analyzing the situation depicted in Figure 8.4.1.



**Figure 8.4.1.** Simple power- and frequency-limited channel.

The channel input,  $x(t)$ , in Figure 8.4.1 has an arbitrary duration,  $T$ , is power limited to a power  $S$ , and is passed through a linear time-invariant filter with impulse response  $h_1(t)$ . The filter output,  $u(t)$ , is given by

$$u(t) = \int x(\tau)h_1(t - \tau) d\tau$$

The received signal,  $y(t)$ , is the sum of  $u(t)$  and the white Gaussian noise. We shall consider both the situation where the channel output is the infinite duration waveform  $y(t)$  and where the output is considered to be only a finite duration portion of  $y(t)$ . As explained before, we can think of the filter\*  $h_1(t)$  as being part of the channel or as being an arbitrary constraint imposed to regulate the spectral density of  $u(t)$ .

\* We do not assume that  $h_1(t)$  is realizable; that is, that  $h_1(t) = 0$  for  $t < 0$ . If the filter plays the role of a mathematical constraint, there is no reason for such a restriction.

The first problem in analyzing the situation in Figure 8.4.1 is that of finding convenient representations for  $x(t)$ ,  $u(t)$ , and  $y(t)$ . Naturally, it would be convenient if we could represent each of these waveforms by orthonormal expansions and the question is which orthonormal expansion to choose. It would be particularly convenient if we could find two sets of orthonormal functions, say  $\{\varphi_i(t)\}$  and  $\{\theta_i(t)\}$ , where the set  $\{\varphi_i(t)\}$  could be used to represent the filter input and the set  $\{\theta_i(t)\}$  could be used to represent the filter output, and where for each  $i$  the normalized response to  $\varphi_i(t)$  is  $\theta_i(t)$ . The purpose of this section is to show that such sets of functions do exist and that they have a number of other properties that make them eminently natural sets of functions to use in representing the filter and its input and output. There is one rather disturbing feature about these functions that is often quite annoying at first contact. At best, they are rather messy to calculate, and at worst they are virtually impossible. This will not concern us here, since in what follows we shall never have to actually calculate these functions. We shall use them solely as conceptual tools, recognizing that they can be calculated but recognizing also that the insight to be gained from this would not repay the effort.

Fortunately for later generalizations, the development of these sets of functions does not depend upon the time invariance of the filter, and thus our development will be in terms of arbitrary linear time-varying filters. Let  $h(t, \tau)$  be the output at time  $t$  due to an impulse in the input at time  $\tau$ . That is, for a given input  $x(\tau)$ , the output is given by

$$u(t) = \int_{-\infty}^{\infty} h(t, \tau) x(\tau) d\tau \quad (8.4.1)$$

We shall assume, in what follows, that the filter input is constrained to be zero outside the interval  $(-T/2, T/2)$  and also that we are interested in representing the output only over some interval  $(-T_o/2, T_o/2)$ . In order to build this into our model and avoid interminable fussing about limits of integration, we simply define  $h(t, \tau)$  to be zero outside of the range of interest. That is,

$$h(t, \tau) = 0; \quad |\tau| \geq T/2 \quad \text{or } |t| \geq T_o/2 \quad (8.4.2)$$

Thus, if a linear time-invariant filter has an impulse response  $h_1(t)$ , we shall represent it here by

$$h(t, \tau) = \begin{cases} h_1(t - \tau); & |\tau| \leq T/2, |t| \leq T_o/2 \\ 0; & \text{elsewhere} \end{cases} \quad (8.4.3)$$

If  $x(\tau)$  is restricted to be nonzero only over  $(-T/2, T/2)$ , then the output  $u(t)$  given by (8.4.1) will be the output of the linear time-invariant filter  $h_1(t)$  over the interval  $(-T_o/2, T_o/2)$ . On the other hand,  $u(t)$  as given by

(8.4.1) will be zero outside of that interval, while the actual filter output might well be nonzero there.

Either the input duration  $T$  or the output duration  $T_o$  or both may be infinite, but we shall always assume that

$$\iint h^2(t, \tau) dt d\tau < \infty \quad (8.4.4)$$

Along with avoiding pathological situations, the assumption in (8.4.4) rules out two situations that are quite common in engineering usage. One is the case where  $h(t, \tau)$  contains impulses and the other is the case where  $T$  and  $T_o$  are both infinite and the filter is time invariant. Thus, in this development, both of these situations must be treated as limiting cases and there is not always a guarantee that the limit will exist.

We now want to find functions  $\varphi_i(\tau)$  and  $\theta_i(t)$  which are related by

$$\theta_i(t) = \alpha_i \int h(t, \tau) \varphi_i(\tau) d\tau \quad (8.4.5)$$

The requirement that the  $\theta_i(t)$  be orthonormal is then

$$\delta_{ij} = \int \theta_i(t) \theta_j(t) dt = \alpha_i \alpha_j \int \int \int h(t, \tau_1) h(t, \tau_2) \varphi_i(\tau_1) \varphi_j(\tau_2) d\tau_2 d\tau_1 dt \quad (8.4.6)$$

$$= \alpha_i \alpha_j \int \int \mathcal{R}(\tau_1, \tau_2) \varphi_i(\tau_1) \varphi_j(\tau_2) d\tau_1 d\tau_2 \quad (8.4.7)$$

where

$$\mathcal{R}(\tau_1, \tau_2) = \int h(t, \tau_1) h(t, \tau_2) dt \quad (8.4.8)$$

Finding a set of orthonormal functions to satisfy (8.4.7) is a very common problem in mathematics and physics. As we shall see later, it is equivalent to finding a set of numbers  $\lambda_1, \lambda_2, \dots$  and a set of functions  $\varphi_1(\tau), \varphi_2(\tau), \dots$  that satisfy the integral equation

$$\int \mathcal{R}(\tau_1, \tau_2) \varphi_i(\tau_2) d\tau_2 = \lambda_i \varphi_i(\tau_1) \quad (8.4.9)$$

In the following theorem, we shall summarize the properties that the  $\lambda_i$ ,  $\varphi_i(\tau)$ , and  $\theta_i(t)$  have. These properties will make it easy to manipulate the input and output from the filter  $h(t, \tau)$  but, as stated before, they give no indication as to how to actually calculate solutions to (8.4.9); fortunately, we shall not need explicit solutions for most of our results.

**Theorem 8.4.1.** Let  $h(t, \tau)$  be nonzero and square integrable [that is satisfy (8.4.4)]. Then there exists a sequence (infinite or finite) of decreasing

positive numbers  $\lambda_1 \geq \dots \geq \lambda_i \geq \dots > 0$ , and, in one to one correspondence with these numbers,\* there exist two sets of orthonormal functions  $\varphi_i(\tau)$  and  $\theta_i(t)$ ; these numbers and functions have the following properties.

(a) The  $\varphi_i(\tau)$  and  $\lambda_i$  satisfy the integral equation

$$\int \mathcal{R}(\tau_1, \tau_2) \varphi_i(\tau_2) d\tau_2 = \lambda_i \varphi_i(\tau_1) \quad (8.4.10)$$

where  $\mathcal{R}(\tau_1, \tau_2)$  is given by (8.4.8).

(b) The  $\varphi_i(\tau)$  and  $\theta_i(t)$  are related by

$$\sqrt{\lambda_i} \theta_i(t) = \int h(t, \tau) \varphi_i(\tau) d\tau \quad (8.4.11)$$

$$\sqrt{\lambda_i} \varphi_i(\tau) = \int h(t, \tau) \theta_i(t) dt \quad (8.4.12)$$

(c) The  $\theta_i(t)$  and  $\lambda_i$  satisfy the integral equation

$$\int \mathcal{R}_o(t_1, t_2) \theta_i(t_2) dt_2 = \lambda_i \theta_i(t_1) \quad (8.4.13)$$

where

$$\mathcal{R}_o(t_1, t_2) = \int h(t_1, \tau) h(t_2, \tau) d\tau \quad (8.4.14)$$

(d) Let  $x(\tau)$  be an arbitrary  $L_2$  function and let  $(x, \varphi_i) = \int x(\tau) \varphi_i(\tau) d\tau$ . Then the following three statements imply each other:

$$\begin{aligned} (x, \varphi_i) = 0 \quad \text{for all } i &\Leftrightarrow \int \mathcal{R}(\tau_1, \tau_2) x(\tau_2) d\tau_2 = 0 \\ &\Leftrightarrow \int h(t, \tau) x(\tau) d\tau = 0 \end{aligned} \quad (8.4.15)$$

Also, if  $x(\tau)$  is expanded as

$$x(\tau) = \sum_i x_i \varphi_i(\tau) + x_r(\tau); \quad x_i = (x, \varphi_i) \quad (8.4.16)$$

Then

$$\int h(t, \tau) x(\tau) d\tau = \sum x_i \sqrt{\lambda_i} \theta_i(t) \quad (8.4.17)$$

$$\int \mathcal{R}(\tau_1, \tau_2) x(\tau_2) d\tau_2 = \sum x_i \lambda_i \varphi_i(\tau_1) \quad (8.4.18)$$

\* Note that any solutions  $\varphi(\tau)$  to (8.4.10) with  $\lambda = 0$  are not considered to be in the orthonormal set  $\{\varphi_i(\tau)\}$ .

(e) Let  $u(t)$  be an arbitrary  $L_2$  function. Then the following statements imply each other:

$$\begin{aligned} (u, \theta_i) = 0 \quad \text{for all } i &\Leftrightarrow \int \mathcal{R}_o(t_1, t_2) u(t_2) dt_2 = 0 \\ &\Leftrightarrow \int h(t, \tau) u(t) d\tau = 0 \end{aligned} \quad (8.4.19)$$

If  $u(t)$  is expanded as

$$u(t) = \sum u_i \theta_i(t) + u_r(t); \quad u_i = (u, \theta_i) \quad (8.4.20)$$

then

$$\int h(t, \tau) u(t) d\tau = \sum u_i \sqrt{\lambda_i} \varphi_i(\tau) \quad (8.4.21)$$

$$\int \mathcal{R}_o(t_1, t_2) u(t_2) dt_2 = \sum u_i \lambda_i \theta_i(t) \quad (8.4.22)$$

$$(f) \quad h(t, \tau) = \sum \sqrt{\lambda_i} \varphi_i(\tau) \theta_i(t) \quad (8.4.23)$$

$$\iint h^2(t, \tau) d\tau = \sum \lambda_i \quad (8.4.24)$$

$$\mathcal{R}(\tau_1, \tau_2) = \sum \lambda_i \varphi_i(\tau_1) \varphi_i(\tau_2) \quad (8.4.25)$$

$$\mathcal{R}_o(t_1, t_2) = \sum \lambda_i \theta_i(t_1) \theta_i(t_2) \quad (8.4.26)$$

$$\iint \mathcal{R}^2(\tau_1, \tau_2) d\tau_1 d\tau_2 = \iint \mathcal{R}_o^2(t_1, t_2) dt_1 dt_2 = \sum \lambda_i^2 \quad (8.4.27)$$

(g) The  $\lambda_i$  and  $\varphi_i(\tau)$  are solutions of the following maximization problems.

$$\lambda_i = \max \left\| \int h(t, \tau) x(\tau) d\tau \right\|^2 \quad (8.4.28)$$

$$\lambda_i = \max \left\| \int \mathcal{R}(\tau_1, \tau_2) x(\tau_2) d\tau_2 \right\| \quad (8.4.29)$$

The maximizations above are subject to the constraints  $\|x\| = 1$  and  $(x, \varphi_j) = 0$  for  $1 \leq j < i$  where  $\|x\|$  is defined as  $\sqrt{\int x^2(\tau) d\tau}$ . In each case,  $\varphi_i(\tau)$  can be taken as the  $x(\tau)$  that maximizes the above quantities.

*Proof.* Let  $\lambda_i \neq 0$ ,  $\varphi_i(\tau)$  and  $\lambda_j \neq 0$ ,  $\varphi_j(\tau)$  be any two normalized solutions to the integral equation 8.4.10. Let  $\theta_i(t)$  and  $\theta_j(t)$  be given by

$$\sqrt{|\lambda_i|} \theta_i(t) = \int h(t, \tau) \varphi_i(\tau) d\tau$$

Then we have, as in (8.4.7):

$$\begin{aligned} \int \sqrt{|\lambda_i \lambda_j|} \theta_i(t) \theta_j(t) dt &= \iint \mathcal{R}(\tau_1, \tau_2) \varphi_i(\tau_1) \varphi_j(\tau_2) d\tau_1 d\tau_2 \\ &= \lambda_j \int \varphi_i(\tau_1) \varphi_j(\tau_1) d\tau_1 \end{aligned} \quad (8.4.30)$$

$$= \lambda_i \int \varphi_i(\tau_2) \varphi_j(\tau_2) d\tau_2 \quad (8.4.31)$$

In (8.4.30) we have used (8.4.10) to integrate over  $\tau_2$ . In (8.4.31) we have used the fact that  $\mathcal{R}(\tau_1, \tau_2) = \mathcal{R}(\tau_2, \tau_1)$  and integrated first over  $\tau_1$ . From the equality of (8.4.30) and (8.4.31), we see that if  $\lambda_i \neq \lambda_j$ , then  $\varphi_i$  and  $\varphi_j$  must be orthogonal and it follows that  $\theta_i$ ,  $\theta_j$  are orthogonal. If  $i = j$ , it follows from (8.4.30) and the normalization of  $\varphi_i(\tau)$  that  $\theta_i(t)$  is also normalized; also, since both integrals in (8.4.30) are now positive, it follows that  $\lambda_i > 0$ . Finally, if  $\lambda_i = \lambda_j$  but  $\varphi_i(\tau)$  and  $\varphi_j(\tau)$  are linearly independent, any linear combination of  $\varphi_i(\tau)$  and  $\varphi_j(\tau)$  will also satisfy (8.4.10). In what follows, if more than one linearly independent  $\varphi(\tau)$  satisfies (8.4.10) for the same value of  $\lambda$ , we shall include in the set  $\varphi_i(\tau)$  only an orthonormal basis for the set of solutions of (8.4.10) with that value of  $\lambda$  and repeat that value an appropriate number of times in the sequence  $\lambda_i$ . With this convention, we have shown that the nonzero  $\lambda$  satisfying (8.4.10) are positive, that the associated  $\varphi_i(\tau)$  are orthonormal, and that the  $\theta_i(t)$  given by (8.4.11) are orthonormal. We shall return later to show that the nonzero  $\lambda_i$  satisfying (8.4.10) can be arranged in a decreasing sequence, and assume for the moment only that they are arranged in an arbitrary fashion.

We next verify (8.4.12) by multiplying both sides of (8.4.11) by  $h(t, \tau_1)$  and integrating over  $t$ .

$$\begin{aligned} \sqrt{\lambda_i} \int \theta_i(t) h(t, \tau_1) dt &= \iint h(t, \tau) h(t, \tau_1) \varphi_i(\tau) d\tau dt \\ &= \int \mathcal{R}(\tau_1, \tau_2) \varphi_i(\tau_2) d\tau_2 \\ &= \lambda_i \varphi_i(\tau_1) \end{aligned}$$

This is equivalent to (8.4.12). Equation 8.4.13 follows in the same way by multiplying both sides of (8.4.12) by  $h(t_1, \tau)$ , integrating over  $\tau$ , and using (8.4.11).

Up to this point, we have demonstrated some properties that must be possessed by solutions of the integral equation (8.4.10), but we have not yet demonstrated that (8.4.10) has any solutions.

On the one hand, it requires a lengthy proof to demonstrate the existence of solutions to (8.4.10), and on the other hand this existence is a central point in the theory of linear integral equations and functional analysis; thus we shall simply state the result.\* If a kernel  $\mathcal{R}(\tau_1, \tau_2)$  is nonzero and square integrable, and if  $\mathcal{R}(\tau_1, \tau_2) = \mathcal{R}(\tau_2, \tau_1)$ , then  $\mathcal{R}(\tau_1, \tau_2)$  has at least one non-zero eigenvalue  $\lambda_i$  and eigenfunction  $\varphi_i(\tau)$  [that is, a solution to (8.4.10)], and in fact enough eigenfunctions and eigenvalues that if a function  $x(\tau)$  is orthogonal to all the  $\varphi_i(\tau)$ , it must satisfy  $\int \mathcal{R}(\tau_1, \tau_2)x(\tau_2) d\tau_2 = 0$ . To use this result here, we must show that  $\mathcal{R}(\tau_1, \tau_2)$  is square integrable. Applying the Schwarz inequality to  $\mathcal{R}(\tau_1, \tau_2) = \int h(t, \tau_1)h(t, \tau_2) dt$ , we get

$$\begin{aligned}\mathcal{R}^2(\tau_1, \tau_2) &\leq \int h^2(t, \tau_1) dt \int h^2(t, \tau_2) dt \\ \iint \mathcal{R}^2(\tau_1, \tau_2) d\tau_1 d\tau_2 &\leq \left[ \int h^2(t, \tau) dt \right]^2 < \infty\end{aligned}$$

Thus the above result applies, showing that the first statement of (8.4.15) implies the second. The second statement also implies the third, because multiplying the second statement by  $x(\tau_1)$  and integrating, we get

$$0 = \iint \mathcal{R}(\tau_1, \tau_2)x(\tau_2)x(\tau_1) d\tau_2 d\tau_1$$

Using (8.4.8) for  $\mathcal{R}(\tau_1, \tau_2)$ , this becomes

$$\begin{aligned}0 &= \iiint h(t, \tau_2)x(\tau_2)h(t, \tau_1)x(\tau_1) d\tau_1 d\tau_2 dt \\ 0 &= \int \left[ \int h(t, \tau)x(\tau) d\tau \right]^2 dt\end{aligned}$$

Thus

$$\int h(t, \tau)x(\tau) d\tau = 0.$$

We shall return later to show that the third statement of (8.4.15) implies the first and proceed to establish (8.4.17).

Using the expansion of  $x(\tau)$  in (8.4.16), we have

$$\int h(t, \tau)x(\tau) d\tau = \int h(t, \tau) \sum_{i=1}^{\infty} x_i \varphi_i(\tau) d\tau + \int h(t, \tau)x_r(\tau) d\tau \quad (8.4.32)$$

Using (8.4.15), the last term in (8.4.32) is zero since  $x_r(\tau)$  is orthogonal to all

\* See, for example, Riesz and Nagy (1955), p. 242, Akheizer and Glazman (1961) p. 127. or Courant and Hilbert (1953) p. 122–136. Courant and Hilbert do not consider the case where the interval  $T$  is infinite, however.

the  $\varphi_i(\tau)$ . From (8.1.15), we can interchange the order of summation and integration in the first term of (8.4.32), and thus we have

$$\int h(t, \tau) x(\tau) d\tau = \sum_{i=1}^{\infty} \int h(t, \tau) x_i \varphi_i(\tau) d\tau \quad (8.4.33)$$

Using (8.4.11) to integrate the right-hand side, we get (8.4.17). Equation 8.4.18 follows in the same way except that (8.4.10) is used to integrate each term. We now return to show that if  $u(t) = \int h(t, \tau) x(\tau) d\tau$  and if  $u(t)$  is 0, then  $x(\tau)$  is orthogonal to all the  $\varphi_i$ . From (8.4.17) and Bessel's inequality,

$$0 = \int u^2(t) dt \geq \sum x_i^2 \lambda_i$$

Since all the  $\lambda_i$  are positive, all the  $x_i$  must be zero, and the third statement of (8.4.15) implies the first.

Part (e) of the theorem is proved in the same way as part (d), and is in fact a dual statement to part (d). We proceed to part (f) and the expansion of  $h(t, \tau)$ . Since  $h(t, \tau)$  is a function of two variables, we can expand it as a series in terms of the  $\varphi_i(\tau)$  and  $\theta_i(t)$ . The coefficients are

$$\begin{aligned} h_{ij} &= \iint h(t, \tau) \varphi_i(\tau) \theta_j(t) d\tau dt = \int_t \sqrt{\lambda_i} \theta_i(t) \theta_j(t) dt \\ &= \sqrt{\lambda_i} \delta_{ij} \\ h(t, \tau) &= \sum_i \sqrt{\lambda_i} \varphi_i(\tau) \theta_i(t) + h_r(t, \tau) \end{aligned}$$

where  $h_r(t, \tau)$  is orthogonal to all  $\varphi_i(\tau) \theta_j(t)$  terms. We now show that, for any  $x(\tau)$ ,  $\int h_r(t, \tau) x(\tau) d\tau = 0$ .

$$\int h_r(t, \tau) x(\tau) d\tau = \int h(t, \tau) x(\tau) d\tau - \int \sum_i \sqrt{\lambda_i} \theta_i(t) \varphi_i(\tau) x(\tau) d\tau$$

Using (8.4.17) on the first term above and using (8.1.15) to interchange the order of summation and integration in the second term, we get

$$\int h_r(t, \tau) x(\tau) d\tau = \sum x_i \sqrt{\lambda_i} \theta_i(t) - \sum x_i \sqrt{\lambda_i} \theta_i(t) = 0$$

Since  $\int h_r(t, \tau) x(\tau) d\tau = 0$  for all  $x(\tau)$ ,  $h_r(t, \tau)$  must be zero, and (8.4.23) is established. To see that this implies that  $h_r(t, \tau) = 0$ , we can expand  $h_r$  in terms of complete sets of orthonormal functions and get an immediate contradiction if any term is assumed nonzero. Equation 8.4.24 follows from (8.4.23) by applying the argument in (8.1.5), using  $h(t, \tau)$  for  $x(t)$  and integrating over  $t$  and  $\tau$ . Since

$$\sum_i \lambda_i$$

is finite, the  $\lambda_i$  can have no limit point other than zero and can be arranged in decreasing order.

Equations 8.4.25 and 8.4.26 are proved in the same way as (8.4.23), and (8.4.27) is proved in the same way as (8.4.24).

Finally, we turn to the maximization problem in (8.4.28). Using (8.4.17), we have

$$\begin{aligned} \left\| \int h(t, \tau) x(\tau) d\tau \right\|^2 &= \left\| \sum x_j \sqrt{\lambda_j} \theta_j(t) \right\|^2 \\ &= \sum_j x_j^2 \lambda_j \end{aligned}$$

Since, in the maximization,  $x_1, \dots, x_{i-1}$  are restricted to be 0, and  $\lambda_j$  is decreasing with  $j$ , we have

$$\left\| \int h(t, \tau) x(\tau) d\tau \right\|^2 \leq \lambda_i \sum_{j=i}^{\infty} x_j^2$$

Since  $\|x\|$  is constrained to be 1,  $\lambda_i$  is an upper bound to the right side of (8.4.28). If  $x(\tau) = \varphi_i(\tau)$ , however,  $\left\| \int h(t, \tau) x(\tau) d\tau \right\|^2 = \lambda_i$ . Thus (8.4.28) is valid and  $\varphi_i(\tau)$  is the maximizing function. Equation 8.4.29 follows in the same way. |

### **Filtered Noise and the Karhunen-Loeve Expansion**

As an example of the use of the preceding theorem, consider passing white Gaussian noise of unit spectral density through a filter with impulse response  $g(t, \tau)$ . We shall be primarily interested in the case where  $g(t, \tau)$  is a time-invariant filter with a time-limited output.

$$g(t, \tau) = \begin{cases} g_1(t - \tau); & |t| \leq T_o/2 \\ 0; & |t| > T_o/2 \end{cases} \quad (8.4.34)$$

Whether (8.4.34) is satisfied or not, we shall assume that  $g(t, \tau)$  is square integrable. Let  $\varphi_i(\tau)$ ,  $\theta_i(t)$ , and  $\lambda_i$  be the input eigenfunctions, output eigenfunctions, and eigenvalues respectively of  $g(t, \tau)$  in the sense of Theorem 8.4.1. Let  $n(\tau)$  represent a sample function of white Gaussian noise of unit spectral density. Recall, from our previous discussion of white noise, that the sample functions  $n(\tau)$  are not well-defined functions of time, but by assumption the effect of  $n(\tau)$  on a linear filter is well defined. Thus, let  $z(t)$  be the output from  $g(t, \tau)$  corresponding to the input  $n(\tau)$ .

$$z(t) = \int g(t, \tau) n(\tau) d\tau \quad (8.4.35)$$

We also define  $n_i$  and  $z_i$  by

$$n_i = \int n(\tau) \varphi_i(\tau) d\tau \quad (8.4.36)$$

$$z_i = \int z(t) \theta_i(t) dt \quad (8.4.37)$$

From (8.1.41), we see that the  $n_i$  are statistically independent, zero mean, unit-variance, Gaussian random variables. The  $z_i$  and  $n_i$  can now be related by substituting the expansion for  $g(t, \tau)$  given by (8.4.23) into (8.4.35).

$$\begin{aligned} z(t) &= \sum_i \sqrt{\lambda_i} \varphi_i(\tau) \theta_i(t) n(\tau) d\tau \\ &= \sum_{i=1}^L \sqrt{\lambda_i} n_i \theta_i(t) + \int \sum_{i=L+1}^{\infty} \sqrt{\lambda_i} \varphi_i(\tau) \theta_i(t) n(\tau) d\tau \end{aligned} \quad (8.4.38)$$

Now denote the remainder term in (8.4.38) by  $z_L(t)$  and let

$$\begin{aligned} g_L(t, \tau) &= \sum_{i=L+1}^{\infty} \sqrt{\lambda_i} \varphi_i(\tau) \theta_i(t) \\ z_L(t) &= \int g_L(t, \tau) n(\tau) d\tau \end{aligned} \quad (8.4.39)$$

From (8.1.47), using  $g_L(t, \tau)$  for  $h(t - \tau)$ , we have

$$\overline{z_L^2(t)} = \int g_L^2(t, \tau) d\tau$$

Since  $g_L(t, \tau)$  approaches 0 with increasing  $L$  as a limit in the mean, we have

$$\lim_{L \rightarrow \infty} \int \overline{z_L^2(t)} dt = 0 \quad (8.4.40)$$

Since  $z_L^2(t)$  is nonnegative, it follows from (8.4.40) that  $\lim z_L^2(t) = 0$  almost everywhere with probability 1. Thus we can represent  $z(t)$  by

$$z(t) = \sum_{i=1}^{\infty} \sqrt{\lambda_i} n_i \theta_i(t) = \sum_{i=1}^{\infty} z_i \theta_i(t) \quad (8.4.41)$$

$$z_i = \sqrt{\lambda_i} n_i \quad (8.4.42)$$

Since the  $n_i$  are independent with variance 1, the  $z_i$  are independent, zero-mean, Gaussian random variables, satisfying

$$\overline{z_i z_j} = \delta_{ij} \lambda_i \quad (8.4.43)$$

The expansion (8.4.41) is known as a Karhunen-Loeve expansion of  $z(t)$ . We see that the functions  $\theta_i(t)$  are doubly useful for representing  $z(t)$ . First,

they are orthonormal and, second, the random variables  $z_i$  are statistically independent.

Notice that we have not shown that the set of functions  $\theta_i(t)$  are complete, but only that they are sufficiently complete to represent the output from passing white Gaussian noise through  $g(t,\tau)$ . We shall now show that, if  $g(t,\tau)$  is time limited by (8.4.34), then the  $\theta_i(t)$  are complete over the interval  $(-T_o/2, T_o/2)$ .

Suppose that  $v(t)$  is square integrable, nonzero only in the interval  $(-T_o/2, T_o/2)$ , and orthonormal to all the  $\theta_i(t)$ . Then from (8.4.19),

$$\begin{aligned} \int g(t,\tau)v(t) dt &= 0 \\ \int g_1(t-\tau)v(t) dt &= 0 \end{aligned} \quad (8.4.44)$$

Letting  $G_1(f)$  and  $V(f)$  be the Fourier transforms of  $g_1$  and  $v$ , and using the convolution theorem, (8.4.44) becomes

$$\begin{aligned} G_1^*(f)V(f) &= 0 \\ V(f) = 0; \quad f: G_1(f) &\neq 0 \end{aligned}$$

Next, we show that  $V(f)$  is an analytic function of  $f$  everywhere, and, thus, the fact that  $V(f) = 0$  on any interval where  $G_1(f) \neq 0$  implies that all terms are zero in the Taylor series expansion of  $V(f)$  around a point in that interval; this in turn implies that  $V(f) = 0$  everywhere.

$$\begin{aligned} V(f) &= \int_{-T_o/2}^{T_o/2} v(t)e^{-j2\pi ft} dt \\ \frac{dV(f)}{df} &= \int_{-T_o/2}^{T_o/2} -j2\pi t v(t)e^{-j2\pi ft} dt \end{aligned}$$

Since  $v(t)$  is square integrable,  $dV(f)/df$  exists and is finite for all complex  $f$ . Thus  $V(f)$  is analytic and  $v(t) = 0$  almost everywhere.

The autocorrelation of the output process,  $\overline{z(t_1)z(t_2)}$  can now be found with the help of (8.4.41):

$$\overline{z(t_1)z(t_2)} = \sum_{i,j} z_i z_j \theta_i(t_1) \theta_j(t_2) \quad (8.4.45)$$

Interchanging the summation and expectation and using (8.4.43):

$$\overline{z(t_1)z(t_2)} = \sum_i \lambda_i \theta_i(t_1) \theta_i(t_2) \quad (8.4.46)$$

$$= \mathcal{R}_o(t_1, t_2) \quad (8.4.47)$$

where  $\mathcal{R}_o$  is given by

$$\mathcal{R}_o(t_1, t_2) = \int g(t_1, \tau)g(t_2, \tau) d\tau \quad (8.4.48)$$

These equations are valid in our usual mean square sense.

At this point, we can go through the previous development again, starting with the assumption that  $z(t)$  is a sample function or truncated sample function of an arbitrary, zero mean, Gaussian random process with the autocorrelation function  $\mathcal{R}_o(t_1, t_2)$ . Assuming that  $\mathcal{R}_o(t_1, t_2)$  is square integrable, we can let  $\theta_i(t)$  and  $\lambda_i$  be the eigenfunctions and eigenvalues of (8.4.13). We can again represent  $z(t)$  by (8.4.37) and the coefficients  $z_i$  are jointly Gaussian and have zero mean. We now show that the coefficients are uncorrelated and hence statistically independent.

$$\overline{z_i z_j} = \overline{\int \int z(t_1)z(t_2)\theta_i(t_1)\theta_j(t_2) dt_1 dt_2} \quad (8.4.49)$$

$$= \int \int \mathcal{R}_o(t_1, t_2)\theta_i(t_1)\theta_j(t_2) dt_1 dt_2 \quad (8.4.50)$$

$$= \lambda_i \delta_{ij} \quad (8.4.51)$$

Next, we must investigate whether the remainder term

$$z_r(t) = z(t) - \sum_i z_i \theta_i(t)$$

is zero. Using Bessel's inequality in the form of (8.1.5), we have

$$\begin{aligned} \int z_r^2(t) dt &= \int z^2(t) dt - \sum_{i=1}^{\infty} z_i^2 \\ \overline{\int z_r^2(t) dt} &= \int \mathcal{R}_o(t, t) dt - \sum_i \lambda_i \end{aligned} \quad (8.4.52)$$

If there is some square integrable function  $g(t, \tau)$  for which  $\mathcal{R}_o(t_1, t_2)$  is the output correlation given by (8.4.48), then

$$\int \mathcal{R}_o(t, t) dt = \int \int g^2(t, \tau) d\tau dt$$

From (8.4.24), this is equal to  $\sum \lambda_i$ , and  $z_r(t)$  is zero almost everywhere with probability 1.

Also, if  $T_o$  is finite and  $\mathcal{R}_o(t_1, t_2)$  is continuous, then Mercer's theorem\* states that (8.4.26) converges uniformly for all  $t_1, t_2$ . Substituting (8.4.26) into (8.4.52) and integrating, the right-hand side is again zero. Mathematically,  $z_r(t)$  is not zero for completely arbitrary autocorrelations. For example, if  $\mathcal{R}_o$  is the sum of a continuous function and another function which

\* See, for example, Riesz and Nagy (1955), p. 245.

is 1 for  $t_1 = t_2$  and 0 elsewhere, then the additional function will not affect the  $\theta_i(t)$  or the  $\lambda_i$ , but it will change  $\int \mathcal{R}_o(t,t) dt$ . We shall tend to ignore such pathologies in what follows since they do not correspond to cases of any physical interest. Thus, for all cases of interest, we again have the Karhunen-Loeve\* representation, (8.4.41).

Summarizing the previous results, we can represent filtered white Gaussian noise by (8.4.41) and we can represent a Gaussian random process with autocorrelation  $\mathcal{R}_o(t_1, t_2)$  by (8.4.41). Thus, if for a given autocorrelation we can find a function  $g(t, \tau)$  that satisfies (8.4.48), then we can consider that random process to be white noise passed through the filter  $g(t, \tau)$ . Regarding nonwhite Gaussian noise as filtered white noise is a very useful conceptual tool.

In the case of a stationary Gaussian process with an integrable spectral density,  $N(f)$ , finding a  $g(t, \tau)$  to satisfy (8.4.48) is quite easy. We simply define

$$g_1(t) = \int \sqrt{N(f)} e^{j2\pi f t} df \quad (8.4.53)$$

Letting  $g(t, \tau)$  be the truncated version of  $g_1(t)$  given by (8.4.34), we have

$$\int g(t_1, \tau) g(t_2, \tau) d\tau = \int g_1(t_1 - \tau) g_1(t_2 - \tau) d\tau \quad (8.4.54)$$

$$= \int N(f) e^{j2\pi f(t_1 - t_2)} df \quad (8.4.55)$$

$$= \mathcal{R}(t_1 - t_2); \quad |t_1| \leq T_o/2, |t_2| \leq T_o/2 \quad (8.4.56)$$

Here  $\mathcal{R}(\tau)$  is the autocorrelation of the process, given by the inverse Fourier transform of  $N(f)$ . Letting  $\mathcal{R}_o(t_1, t_2)$  be  $\mathcal{R}(t_1 - t_2)$  for  $|t_1|$  and  $|t_2|$  less than or equal to  $T_o/2$ , we have (8.4.48). As a check, observe that white noise, if filtered with a frequency response  $\sqrt{N(f)}$ , yields an output spectral density  $N(f)$ .

### Low-Pass Ideal Filters

An important special case of Theorem 8.4.1 is that of an ideal low-pass filter with a cutoff at some given frequency  $W$ . The filter is described by the frequency response

$$H_1(f) = \begin{cases} 1; & |f| \leq W \\ 0; & |f| > W \end{cases} \quad (8.4.57)$$

\* The Karhunen-Loeve (see Loeve (1955)) theorem actually applies to the case where  $R_o$  is continuous and  $T_o$  is finite. It also says that (8.4.41) converges in mean square over the ensemble for each value of  $t$ , whereas we have merely asserted mean-square convergence over both the ensemble and  $t$ .

The impulse response  $h_1(t)$  is the inverse Fourier transform of  $H_1(f)$ ,

$$h_1(t) = \frac{\sin 2\pi Wt}{\pi t} \quad (8.4.58)$$

We now limit the input to some interval  $(-T/2, T/2)$  and define

$$h(t, \tau) = \begin{cases} h_1(t - \tau); & |\tau| \leq T/2 \\ 0; & |\tau| > T/2 \end{cases} \quad (8.4.59)$$

The set of input eigenfunctions,  $\{\varphi_i(\tau)\}$ , for the filter  $h(t, \tau)$  are given by (8.4.10) as the solutions to the integral equation

$$\int \mathcal{R}(\tau_1, \tau_2) \varphi_i(\tau_2) d\tau_2 = \lambda_i \varphi_i(\tau_1) \quad (8.4.60)$$

where

$$\mathcal{R}(\tau_1, \tau_2) = \begin{cases} \mathcal{R}_1(\tau_2 - \tau_1); & |\tau_1| \leq T/2, |\tau_2| \leq T/2 \\ 0; & \text{elsewhere} \end{cases}$$

$$\mathcal{R}_1(\tau_2 - \tau_1) = \int h_1(t - \tau_1) h_1(t - \tau_2) dt \quad (8.4.61)$$

Taking the Fourier transform of both sides of (8.4.61), we find that the Fourier transform of  $\mathcal{R}_1$  is simply  $|H_1(f)|^2$ , which is equal to  $H_1(f)$ . Thus

$$\mathcal{R}(\tau_1, \tau_2) = \begin{cases} \frac{\sin 2\pi(\tau_1 - \tau_2)W}{\pi(\tau_1 - \tau_2)}; & |\tau_1| \leq T/2, |\tau_2| \leq T/2 \\ 0; & \text{otherwise} \end{cases} \quad (8.4.62)$$

The input eigenfunctions  $\varphi_i(\tau)$  [which are limited to  $(-T/2, T/2)$ ] are related to the output eigenfunctions  $\theta_i(t)$  by

$$\theta_i(t) = \frac{1}{\sqrt{\lambda_i}} \int h_1(t - \tau) \varphi_i(\tau) d\tau \quad (8.4.63)$$

$$\varphi_i(\tau) = \begin{cases} \frac{1}{\sqrt{\lambda_i}} \int h_1(t - \tau) \theta_i(t) dt; & |\tau| \leq T/2 \\ 0; & |\tau| > T/2 \end{cases} \quad (8.4.64)$$

From (8.4.63) we see that  $\theta_i(t)$  is  $1/\sqrt{\lambda_i}$  times  $\varphi_i(t)$  bandlimited to  $|f| \leq W$ . In other words, the Fourier transforms of  $\theta_i(t)$  and  $\varphi_i(t)$  are related by

$$\Theta_i(f) = \begin{cases} \frac{1}{\sqrt{\lambda_i}} \Phi_i(f); & |f| < W \\ 0; & |f| > W \end{cases} \quad (8.4.65)$$

Furthermore, since  $\theta_i(t)$  is bandlimited, it can pass through the filter  $h_1(t)$  unchanged and  $\int h_1(\tau - t)\theta_i(t) dt$  is just  $\theta_i(\tau)$ . Since  $h_1(t)$  is an even function, this combines with (8.4.64) to yield

$$\varphi_i(\tau) = \begin{cases} \frac{1}{\sqrt{\lambda_i}} \theta_i(\tau); & |\tau| \leq T/2 \\ 0; & |\tau| > T/2 \end{cases} \quad (8.4.66)$$

Thus the  $\theta_i(t)$  have the peculiar property that they are orthonormal over the infinite interval and also orthogonal over the interval  $(-T/2, T/2)$ . The functions  $\theta_i(t)$  are known as prolate spheroidal wave functions and frequently appear in diverse problems in physics and mathematics. There is a large literature on these functions and the reader is particularly referred to the papers of Slepian, Pollak, and Landau (1961), (1962), and (1964). Some useful properties of these functions are that the eigenvalues  $\lambda_i$  are all distinct, yielding unique normalized eigenfunctions (except for sign). Also  $\theta_i(t)$  has precisely  $i - 1$  zeros inside the interval  $(-T/2, T/2)$  and is an even function for  $i$  odd and vice versa.

From our discussion of Theorem 8.4.1, we see that  $\varphi_1(\tau)$  is that normalized function limited to  $(-T/2, T/2)$  which contains the largest energy ( $\lambda_1$ ) in the band  $-W \leq f \leq W$ . Likewise,  $\varphi_i(\tau)$  is the normalized function in  $(-T/2, T/2)$  which has the largest energy ( $\lambda_i$ ) in  $-W \leq f \leq W$  subject to being orthogonal to  $\varphi_1(\tau), \dots, \varphi_{i-1}(\tau)$ . From the same argument,  $\theta_i(t)$  is the normalized function bandlimited to  $(-W, W)$  which has the largest energy ( $\lambda_i$ ) in the time interval  $(-T/2, T/2)$  subject to being orthogonal to  $\theta_1(t), \dots, \theta_{i-1}(t)$ .

We can now return to give a more precise interpretation to the idea that the class of signals that are approximately time and frequency limited have about  $2WT$  degrees of freedom. Consider the set of functions that are linear combinations of the first  $n$  eigenfunctions of (8.4.60),  $\varphi_1(\tau), \dots, \varphi_n(\tau)$ . Let

$$x_n(\tau) = \sum_{i=1}^n x_i \varphi_i(\tau)$$

be an arbitrary function in this class and let

$$u_n(t) = \sum_{i=1}^n x_i \sqrt{\lambda_i} \theta_i(t)$$

be the portion of  $x_n(\tau)$  that is bandlimited to  $-W \leq f \leq W$ . The fraction of energy in  $x_n(\tau)$  that is contained in the band  $-W \leq f \leq W$  is then given by

$$1 \geq \frac{\int u_n^2(t) dt}{\int x_n^2(\tau) d\tau} = \frac{\sum_{i=1}^n \lambda_i x_i^2}{\sum_{i=1}^n x_i^2} \geq \lambda_n \quad (8.4.67)$$

The final inequality above follows from the fact that all the  $\lambda_i$  in the above sum are lower bounded by  $\lambda_n$ . Thus we are considering a class of time-limited functions with  $n$  degrees of freedom all of which have a fractional energy between  $\lambda_n$  and 1 in the frequency band  $-W \leq f \leq W$ . Furthermore, we notice that one of the functions,  $\varphi_n(\tau)$ , has exactly  $\lambda_n$  of its energy in  $-W \leq f \leq W$ .

Before proceeding to discuss the behavior of  $\lambda_n$  with  $n$ , we want to establish that any other set of functions in  $(-T/2, T/2)$  with  $n$  degrees of freedom also contains a function with  $\lambda_n$  or less of its energy in  $-W \leq f \leq W$ . By a set of functions with  $n$  degrees of freedom, we mean the set of linear combinations of  $n$  linearly independent functions. Either these  $n$  functions span the same space as  $\varphi_1(\tau), \dots, \varphi_n(\tau)$  or else they have a linear combination orthogonal to  $\varphi_1(\tau), \dots, \varphi_n(\tau)$ . In the first case,  $\varphi_n(\tau)$  is included in the set

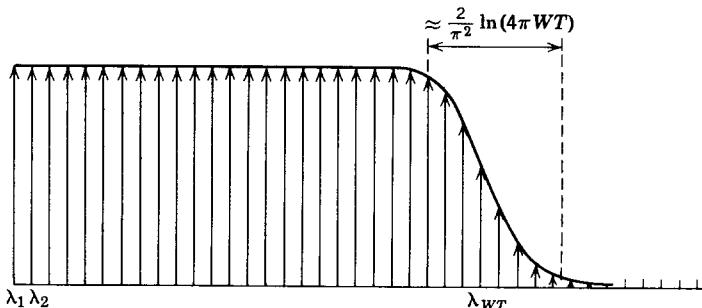


Figure 8.4.2. Sketch of eigenvalues of (8.4.60) for large  $WT$ .

and has  $\lambda_n$  of its energy in  $-W \leq f \leq W$ . In the second case, the function orthogonal to  $\varphi_1(\tau), \dots, \varphi_n(\tau)$  is a linear combination of the  $\varphi_i(\tau)$  with  $i > n$ . Thus, since  $\lambda_i$  is upper bounded by  $\lambda_n$  for  $i > n$ , this function has, at most,  $\lambda_n$  of its energy in  $-W \leq f \leq W$ .

The final question that we must answer now is how  $\lambda_n$  varies with  $n$ ,  $W$ , and  $T$ . By changing the time scale in (8.4.10) and (8.4.62), we can see that  $\lambda_n$  varies only with  $n$  and the product  $WT$ , and we shall use  $\lambda_n(WT)$  to bring out the dependence. Slepian (1965) has shown recently that, if we define a number  $\alpha$  for each  $n$  and  $WT$  by

$$n = 2WT + 1 + \frac{\alpha}{\pi^2} \ln(4\pi WT) \quad (8.4.68)$$

then

$$\lim_{WT \rightarrow \infty} \left| \lambda_n(WT) - \frac{1}{1 + e^\alpha} \right| = 0 \quad (8.4.69)$$

where  $n$  in (8.4.69) is restricted to a range for each  $WT$  such that  $\alpha$  is within a bounded range independent of  $WT$ .

A sketch of this relationship is given in Figure 8.4.2. The major points to

observe are that, for  $n \ll 2WT + 1$ ,  $\lambda_n \approx 1$  and for  $n \gg 2WT + 1$ ,  $\lambda_n \approx 0$ . The transition region between these extremes has a width proportional to  $\ln(4\pi WT)$ . In particular, for any fixed  $\epsilon > 0$ , these equations imply that

$$\lim_{WT \rightarrow \infty} \lambda_{2WT(1+\epsilon)}(WT) = 0 \quad (8.4.70)$$

$$\lim_{WT \rightarrow \infty} \lambda_{2WT(1-\epsilon)}(WT) = 1 \quad (8.4.71)$$

Thus, if we use a set of functions in  $(-T/2, T/2)$  with  $2WT(1 + \epsilon)$  degrees of freedom, some of these functions will have a vanishingly small fraction of their energy in  $-W \leq f \leq W$  for  $WT$  arbitrarily large. Conversely, with  $2WT(1 - \epsilon)$  degrees of freedom, the minimum fraction of energy in the band  $-W \leq f \leq W$ , over all functions in the class, will approach 1 as  $WT$  gets arbitrarily large.

One final peculiar feature of the prolate spheroidal functions that we shall now derive is that the  $\varphi_i(\tau)$  and the  $\theta_i(t)$  are scaled Fourier transforms of each other. Using the definition of  $H_1(f)$  from (8.4.57) to provide a time limitation, we can rewrite (8.4.66) as

$$\varphi_i(\tau) = \frac{1}{\sqrt{\lambda_i}} \theta_i(\tau) H_1\left(\frac{2W\tau}{T}\right) \quad (8.4.72)$$

Taking the Fourier transform of both sides of (8.4.72), we have

$$\Phi_i(f_2) = \frac{T}{2W\sqrt{\lambda_i}} \int \Theta_i(f_1) h_1\left[\frac{T(f_2 - f_1)}{2W}\right] df_1 \quad (8.4.73)$$

Substituting (8.4.65) into (8.4.73) and using the fact that  $\mathcal{R}(\tau_1, \tau_2) = h_1(\tau_2 - \tau_1)$  for  $|\tau_1|, |\tau_2| \leq T/2$ , this becomes

$$\Theta_i(f_2) = \frac{T}{2W\lambda_i} \int_{-W}^W \Theta_i(f_1) \mathcal{R}\left(\frac{Tf_2}{2W}, \frac{Tf_1}{2W}\right) df_1 \quad (8.4.74)$$

Finally, scaling  $f_1$  and  $f_2$  by  $2W/T$ , we see that  $\Theta_i(2Wf/T)$  satisfies the same integral equation, (8.4.10), as  $\varphi_i(\tau)$ . Since these solutions are unique except for a multiplicative factor, we can use the normalization and parity of  $\theta_i(t)$  to obtain

$$\Theta_i\left(\frac{2Wf}{T}\right) = \pm \sqrt{\frac{T}{2W}} \varphi_i(f) (\sqrt{-1})^{i-1} \quad (8.4.75)$$

We can see from (8.4.75) that the  $\theta_i(t)$  do not tend toward either sinusoids or sampling functions in the interval  $(-T/2, T/2)$  as  $T$  becomes large; this, in turn, provides some idea of why the heuristic arguments in Section 8.3 cannot easily be made precise.

## 8.5 Additive Gaussian Noise Channels with an Input Constrained in Power and Frequency

In this section, we shall use the expressions developed in the last section to provide a rigorous solution of the problems considered in Section 8.3. The channel is shown in Figure 8.5.1. The input  $x(t)$  is time constrained to the interval  $(-T/2, T/2)$  and then passed through a linear time-invariant filter with impulse response  $h_1(\tau)$  and frequency response  $H_1(f) = \int h_1(\tau)e^{-j2\pi f\tau} d\tau$ . Stationary, zero mean, Gaussian noise of spectral density  $N(f)$  is added to the filter output and the result is observed over an interval  $(-T_o/2, T_o/2)$ . The input  $x(t)$  is power limited to a power  $S$ . In the converse to the coding theorem, this will be taken to mean that the expected value of

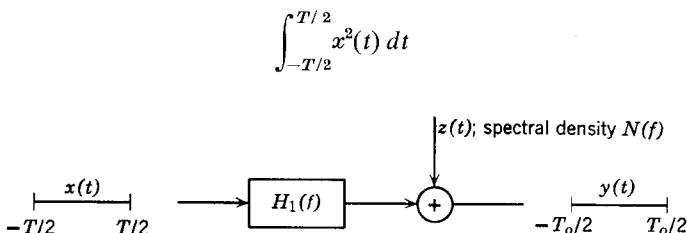


Figure 8.5.1.

is at most  $ST$ . For the coding theorem, we shall use the more restricted condition that

$$\int_{-T/2}^{T/2} x_m^2(t) dt \leq ST$$

for each code word.

First, we shall show how to reduce the continuous time channel in Figure 8.5.1 to a set of parallel, discrete-time, additive Gaussian noise channels. The results of Sections 7.5 then can be directly applied. We then go on to consider the more difficult problem of passing to the limit as  $T \rightarrow \infty$ , holding  $T = T_o$ . The final results will be the same as those in Section 8.3. We shall assume, throughout, that

$$\int_{-\infty}^{\infty} \frac{|H_1(f)|^2}{N(f)} df < \infty \quad (8.5.1)$$

and also that either  $\int N(f) df < \infty$  or that the noise is white [that is,  $N(f)$  is independent of  $f$ ].

One weakness of the approach taken here is that we assume that the input is zero outside of the interval  $(-T/2, T/2)$ . In other words, when we use a coding constraint length of  $T$ , we shall be ignoring the intersymbol interference between successive code words. This is no problem for the converse

to the coding theorem since it is easy to show that the intersymbol interference cannot reduce error probability. It is also no problem in showing that arbitrarily small error probabilities can be achieved at any rate less than capacity, since we can transmit, in principle, only one code word of arbitrarily long duration. It also appears that this intersymbol interference will not reduce the exponent to error probability in the limit of large  $T$ , but thus far it has not been possible to prove this rigorously. The intuitive argument is as follows. If we separate code words of duration  $T$  on the channel by a guard space of  $T^{1-\epsilon}$  for some small  $\epsilon > 0$ , then in the limit as  $T \rightarrow \infty$ , the ratio of guard space to code-word length approaches 0. On the other hand, the contribution in any observation interval due to code words in other intervals is approaching zero in energy as  $T \rightarrow \infty$ , and thus this contribution should not affect the error probability.

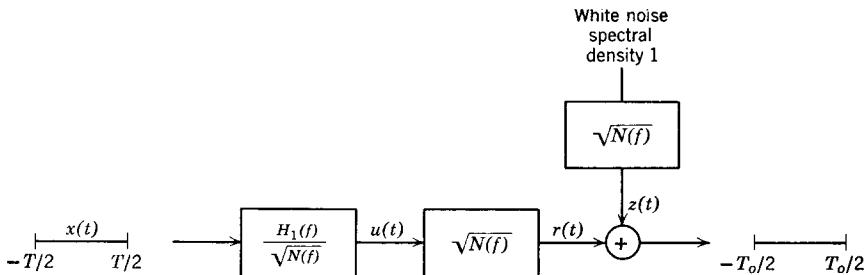


Figure 8.5.2. Equivalent representation of Figure 8.5.1.

In Section 8.4 we showed that Gaussian noise of integrable and finite spectral density  $N(f)$  could be treated as white Gaussian noise of unit spectral density passed through a (nonrealizable) filter with frequency response  $\sqrt{N(f)}$  and impulse response

$$g_1(t) = \int \sqrt{N(f)} e^{j2\pi ft} df \quad (8.5.2)$$

Since  $N(f)$  is an even function of  $f$ ,  $g_1(t)$  is real, and since  $\sqrt{N(f)}$  is square integrable,  $g_1(t)$  is also. If the noise is white with spectral density  $N_o/2$ , we can consider the filter  $g_1(t)$  as a multiplier, multiplying the input by  $\sqrt{N_o/2}$ .

We shall also, for conceptual purposes, split up the filter  $H_1(f)$  into two parts, one with frequency response  $H_1(f)/\sqrt{N(f)}$  and the other with frequency response  $\sqrt{N(f)}$  (see Figure 8.5.2). Again, if  $N(f) = N_o/2$ , the second filter is to be considered as a multiplier. If  $H_1(f)$  and  $N(f)$  are both zero for any  $f$ , we define  $H_1(f)/\sqrt{N(f)}$  to be zero for that  $f$ . The reader with a limited background in electrical engineering will find it instructive to verify that the

frequency response of two linear time-invariant filters in cascade is indeed the product of the individual frequency responses. We shall denote the impulse response of the filter  $H_1(f)/\sqrt{N(f)}$  by

$$K_1(t) = \int \frac{H_1(f)}{\sqrt{N(f)}} e^{j2\pi ft} df \quad (8.5.3)$$

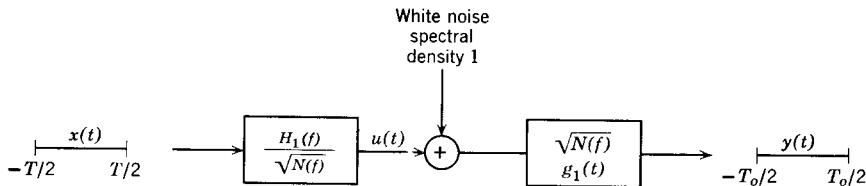
The function  $K_1(t)$ , like  $g_1(t)$ , is real and square integrable.

In terms of the new filters  $K_1(t)$  and  $g_1(t)$ , the input  $x(\tau)$  and output  $r(t)$  from the original filter are related by

$$r(t_1) = \int g_1(t_1 - t) u(t) dt \quad (8.5.4)$$

$$u(t) = \int K_1(t - \tau) x(\tau) d\tau \quad (8.5.5)$$

Since  $u(t)$  and the white noise in Figure 8.5.2 are filtered by the same filter and then added, we can redraw the system as in Figure 8.5.3 where the white



**Figure 8.5.3. Equivalent representation of Figure 8.5.1.**

noise is added directly to  $u(t)$  and then the result is filtered by  $g_1(t)$ . While the channel in Figure 8.5.3 looks quite different from that in Figure 8.5.1, they are identical in the sense that the output waveform  $y(t)$  is, in both cases, the sum of  $r(t)$  and Gaussian noise of spectral density  $N(f)$ ; as long as we accept the ground rule that the receiver observes only  $y(t)$ , we can use the figures interchangeably.

In order to use the expansions of the last section, it is convenient to replace the time-invariant filters in Figure 8.5.3 with time-varying filters that limit the input to  $(-T/2, T/2)$  and the output to  $(-T_o/2, T_o/2)$  automatically. We thus define

$$K(t, \tau) = \begin{cases} K_1(t - \tau); & |\tau| \leq T/2 \\ 0; & |\tau| > T/2 \end{cases} \quad (8.5.6)$$

$$g(t, \tau) = \begin{cases} g_1(t - \tau); & |t| \leq T_o/2 \\ 0; & |t| > T_o/2 \end{cases} \quad (8.5.7)$$

This channel is redrawn in Figure 8.5.4, but the restrictions on input duration and observation interval are now dropped since these operations are performed internally by the channel.

Let  $\xi_i(\tau)$ ,  $\eta_i(t)$ , and  $\mu_i$ ,  $1 \leq i \leq \infty$  be respectively the input eigenfunctions output eigenfunctions, and eigenvalues of the filter  $K(t, \tau)$  in the sense of Theorem 8.4.1. We can then represent  $x(\tau)$  and  $u(t)$  by

$$x(\tau) = \sum_i x_i \xi_i(\tau) \quad (8.5.8)$$

$$u(t) = \sum_i x_i \sqrt{\mu_i} \eta_i(t) \quad (8.5.9)$$

If we could forget about the filter  $g(t, \tau)$  in Figure 8.5.4, then we could simply represent the white noise in terms of the orthonormal functions  $\eta_i(t)$ , and

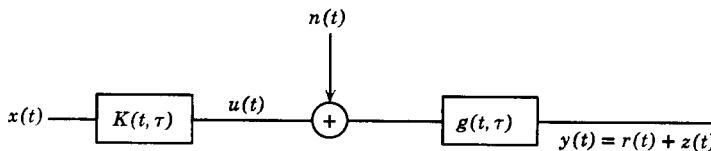


Figure 8.5.4. Equivalent representation of Figure 8.5.1.

for each  $i$ , the receiver could calculate  $x_i \sqrt{\mu_i} + n_i$ , where the  $n_i$  are independent normalized Gaussian random variables.

Unfortunately, the receiver cannot, even in principle, calculate these quantities. The difficulty is that the output from the filter  $g(t, \tau)$  does not uniquely specify the filter input. In other words, there are generally nonzero inputs to  $g(t, \tau)$  for which the output is zero. Analytically, let  $\varphi_{i,g}(\tau)$ ,  $\theta_{i,g}(t)$ , and  $\lambda_{i,g}$ , for  $1 \leq i < \infty$ , be respectively the input eigenfunctions, output eigenfunctions, and eigenvalues of  $g(t, \tau)$ . Then, from (8.4.15), an input to  $g(t, \tau)$  yields no output if and only if that input is orthogonal to  $\varphi_{i,g}(\tau)$  for each  $i$ ,  $1 \leq i < \infty$ .

Now, suppose that we separate  $u(t)$  [the signal input to the filter  $g(t, \tau)$ ] into two components, one a linear combination of the  $\varphi_{i,g}(\tau)$ , and the other orthogonal to all the  $\varphi_{i,g}(\tau)$ . If the filter  $K(t, \tau)$  is modified so as to suppress the component orthogonal to all the  $\varphi_{i,g}(\tau)$ , then the signal output from  $g(t, \tau)$  will be unchanged. On the other hand, we shall soon see that when  $K(t, \tau)$  is modified in this way, the filter  $g(t, \tau)$  destroys no information about  $x(\tau)$ .

In order to see precisely how to modify the filter  $K(t, \tau)$ , observe that the output from  $K(t, \tau)$  is given by

$$u(t) = \int x(\tau) K(t, \tau) d\tau \quad (8.5.10)$$

We want to modify  $K(t, \tau)$  to a new filter  $K_o(t, \tau)$  whose output is

$$u_0(t) = \sum_i \varphi_{i,g}(t) \int_{-\infty}^{\infty} u(t_1) \varphi_{i,g}(t_1) dt_1 \quad (8.5.11)$$

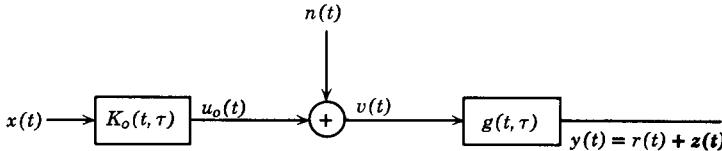
Substituting (8.5.10) into (8.5.11) and interchanging the order of integration, we have

$$u_0(t) = \int x(\tau) \left[ \sum_i \varphi_{i,g}(t) \int K(t_1, \tau) \varphi_{i,g}(t_1) dt_1 \right] d\tau \quad (8.5.12)$$

Thus the modified filter must have the response

$$K_o(t, \tau) = \sum_i \varphi_{i,g}(t) \int K(t_1, \tau) \varphi_{i,g}(t_1) dt_1 \quad (8.5.13)$$

We can now replace  $K(t, \tau)$  in Figure 8.5.4 with  $K_o(t, \tau)$  as shown in Figure 8.5.5. To recapitulate,  $u_0(t)$  in Figure 8.5.5 differs from  $u(t)$  in Figure



**Figure 8.5.5. Equivalent representation of Figure 8.5.1.**

8.5.4 only by a term orthogonal to all the eigenfunctions  $\varphi_{i,g}$ , and thus the response  $r(t)$  in Figure 8.5.5 is exactly the same as in Figure 8.5.4. For the special case when  $N(f) = N_o/2$ ,  $g(t, \tau)$  simply multiplies the input by  $N_o/2$  for  $|t| \leq T_o/2$ , and we define  $K_o(t, \tau)$  analogously by

$$K_o(t, \tau) = \begin{cases} K(t, \tau); & |t| \leq T_o/2 \\ 0; & |t| \geq T_o/2 \end{cases} \quad (8.5.14)$$

Next we show that  $K_o(t, \tau)$  is square integrable, so that the expansions of Section 8.4 can be used. For  $K_o$  given by (8.5.13), we see that for any given  $\tau$ ,  $K_o(t, \tau)$  is the expansion of  $K(t, \tau)$  in terms of the functions  $\varphi_{i,g}(t)$ . Thus, from Bessel's inequality

$$\int_{t=-\infty}^{\infty} K_o^2(t, \tau) dt \leq \int_{-\infty}^{\infty} K^2(t, \tau) dt; \quad \text{all } \tau \quad (8.5.15)$$

$$\int \int K_o^2(t, \tau) dt d\tau \leq \int_{-T/2}^{T/2} \left[ \int K_1^2(t - \tau) dt \right] d\tau = T \int_{-\infty}^{\infty} K_1^2(t) dt < \infty \quad (8.5.16)$$

For  $K_o(t, \tau)$  given by (8.5.14), the square integrability of  $K_o$  follows immediately from that of  $K(t, \tau)$ .

Now let  $\varphi_i(\tau)$ ,  $\theta_i(t)$ , and  $\lambda_i$ , for  $1 \leq i < \infty$ , be respectively the input eigenfunctions, output eigenfunctions, and eigenvalues of  $K_o(t, \tau)$  in the sense of Theorem 8.4.1. As pointed out before, we have no intention of actually calculating these eigenfunctions, but only of investigating the limiting behavior as  $T \rightarrow \infty$ . Thus the complicated expression for  $K_o$  in (8.5.13) really presents no special problem. We can expand  $x(\tau)$ ,  $u_o(t)$ , and  $v(t) = u_o(t) + n(t)$  as

$$x(\tau) = \sum_i x_i \varphi_i(\tau) + x_r(\tau) \quad (8.5.17)$$

$$u_o(t) = \sum_i x_i \sqrt{\lambda_i} \theta_i(t) \quad (8.5.18)$$

$$v(t) = \sum_i [x_i \sqrt{\lambda_i} + n_i] \theta_i(t) + n_r(t) \quad (8.5.19)$$

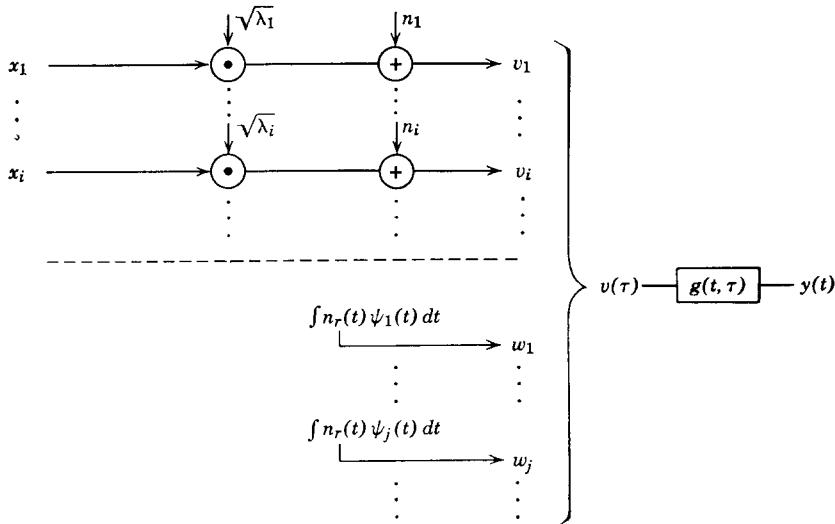


Figure 8.5.6. Equivalent representation of Figure 8.5.1.

As observed before,  $v(t)$  is not well defined since it involves white Gaussian noise, but by the definition of white Gaussian noise, the coefficients  $v_i = x_i \sqrt{\lambda_i} + n_i$  are well defined, and the random variables  $n_i$  are independent normalized Gaussian random variables. The remainder term  $n_r(t)$  in (8.5.19) is the component of the noise orthogonal to all the functions  $\theta_i(t)$ . More precisely,  $\int n_r(t) \theta_i(t) dt = 0$  for each. Also, for every unit energy function  $\psi(t)$  orthogonal to  $\theta_i(t)$  for all  $i$ ,  $\int n_r(t) \psi(t) dt$  is a normalized Gaussian random variable independent of all  $x_i$  and  $n_i$ . If we generate an orthonormal set of functions,  $\{\psi_i(t)\}$ , each orthonormal to the set  $\{\theta_i(t)\}$ , then the channel

can be represented as an infinite set of parallel channels in cascade with the filter  $g(t, \tau)$  (see Figure 8.5.6). We shall show, in what follows, that the coefficients  $v_1, v_2, \dots$  can be determined from the channel output  $y(t)$ .

Now suppose that a receiver had the waveform  $v(t)$  available to it (that is, the sequence  $\{v_i\}$  and the sequence  $\{w_j\}$ ). For any probability measure on  $x(\tau)$ , the average mutual information between  $x(\tau)$  and  $v(t)$  is clearly the average mutual information between the sequence  $\{x_i\}$  and the sequence  $\{v_i\}$  (since the sequence  $\{w_j\}$  in Figure 8.5.6 is independent of both  $\{x_j\}$  and  $\{v_j\}$ ). Likewise, for any set of code words, the maximum likelihood decoding rule and maximum a posteriori probability decoding rule depend only on the sequence  $\{v_i\}$ . Thus only the sequence  $\{v_i\}$  is relevant in the waveform  $v(t)$ . We shall show now that the sequence  $\{v_i\}$  can be determined from the final channel output  $y(t)$ . Since  $y(t)$  is determined by  $v(t)$ , it will follow from this that the extraneous noise terms  $\{w_j\}$  can be omitted from consideration (as is

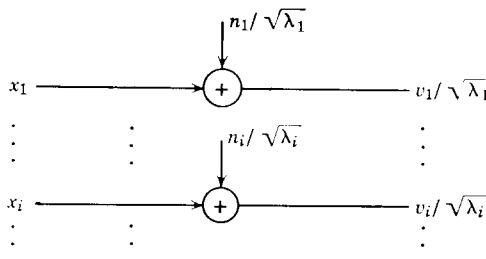


Figure 8.5.7. Final representation of Figure 8.5.1.

almost obvious anyway), and the channel can be represented as in Figure 8.5.7. In Figure 8.5.7, we have moved the additive noise to the left of the multiplier in order to make the final representation equivalent to the parallel channel model in Section 7.5. The noise on the  $i$ th channel is now a zero mean Gaussian random variable of variance  $1/\lambda_i$ .

**LEMMA 8.5.1.** The sequence of variables  $\{v_i\}$  defined above can be uniquely determined from the channel output  $y(t)$ .

*Proof.* If the noise is white,  $y(t) = \sqrt{N_o/2} v(t)$  and the proof is trivial. For nonwhite noise, the variables  $v_i$  are defined in terms of the output eigenfunctions of the filter  $K_o(t, \tau)$  by

$$v_i = \int v(t) \theta_i(t) dt \quad (8.5.20)$$

From (8.5.11), the output from  $K_o(t, \tau)$ , for any input, is a linear combination

of the input eigenfunctions  $\varphi_{i,g}(t)$  for the filter  $g(t,\tau)$ . Thus, for each  $i$ ,  $\theta_i(t)$  is a linear combination of the set  $\{\varphi_{i,g}(t)\}$ , and can be represented by

$$\theta_i(t) = \sum_{j=1}^{\infty} \alpha_{ij} \varphi_{j,g}(t); \quad \alpha_{ij} = \int \theta_i(t) \varphi_{j,g}(t) dt \quad (8.5.21)$$

[It should be observed that this expansion, in general, cannot be made for the output eigenfunctions of the filter  $K(t,\tau)$ , and this was, in fact, the reason for defining the filter  $K_o(t,\tau)$ ]. Substituting (8.5.21) into (8.5.20) and interchanging the sum and the integral for an arbitrary number  $m$ , of the terms, we get

$$v_i = \sum_{j=1}^m \alpha_{ij} \int v(t) \varphi_{j,g}(t) dt + \int v(t) \sum_{j=m+1}^{\infty} \alpha_{ij} \varphi_{j,g}(t) dt \quad (8.5.22)$$

In terms of the output eigenfunction,  $\theta_{j,g}(t)$  of the filter  $g(t,\tau)$ , we have

$$\int y(t) \theta_{j,g}(t) dt = \int r(t) \theta_{j,g}(t) dt + \int z(t) \theta_{j,g}(t) dt \quad (8.5.23)$$

But

$$\int r(t) \theta_{j,g}(t) dt = \iint u_0(\tau) g(t,\tau) \theta_{j,g}(t) d\tau dt = \sqrt{\lambda_{j,g}} \int u_0(\tau) \varphi_{j,g}(\tau) d\tau \quad (8.5.24)$$

Also, from (8.4.42):

$$\int z(t) \theta_{j,g}(t) dt = z_{j,g} = \sqrt{\lambda_{j,g}} \int n(\tau) \varphi_{j,g}(\tau) d\tau \quad (8.5.25)$$

Combining (8.5.24) and (8.5.25):

$$\int y(t) \theta_{j,g}(t) dt = \sqrt{\lambda_{j,g}} \int v(\tau) \varphi_{j,g}(\tau) d\tau \quad (8.5.26)$$

Substituting this in (8.5.22):

$$v_i = \sum_{j=1}^m \frac{\alpha_{ij}}{\sqrt{\lambda_{j,g}}} \int y(t) \theta_{j,g}(t) dt + \int v(t) \sum_{j=m+1}^{\infty} \alpha_{ij} \varphi_{j,g}(t) dt \quad (8.5.27)$$

We now show that the remainder term in (8.5.27) goes to zero as  $m \rightarrow \infty$ .

$$\int v(t) \sum_{j=m+1}^{\infty} \alpha_{ij} \varphi_{j,g}(t) dt = \int u_o(t) \sum_{j=m+1}^{\infty} \alpha_{ij} \varphi_{j,g}(t) dt + \int n(t) \sum_{j=m+1}^{\infty} \alpha_{ij} \varphi_{j,g}(t) dt \quad (8.5.28)$$

The first term in (8.5.28) can be bounded by the Schwartz inequality

$$\begin{aligned} \left[ \int u_o(t) \sum_{j=m+1}^{\infty} \alpha_{ij} \varphi_{j,g}(t) dt \right]^2 &\leq \left[ \int u_o^2(t) dt \right] \int \left[ \sum_{j=m+1}^{\infty} \alpha_{ij} \varphi_{j,g}(t) \right]^2 dt \\ &= \left[ \int u_o^2(t) dt \right] \sum_{j=m+1}^{\infty} \alpha_{ij}^2 \end{aligned} \quad (8.5.29)$$

From the orthonormality of the set  $\theta_i(t)$ , we see from (8.5.21) that

$$\sum_{j=1}^{\infty} \alpha_{ij}^2 = 1,$$

and thus

$$\lim_{m \rightarrow \infty} \sum_{j=m+1}^{\infty} \alpha_{ij}^2 = 0.$$

Since  $u_o(t)$  has finite energy, it follows from (8.5.29) that

$$\lim_{m \rightarrow \infty} \left[ \int u_o(t) \sum_{j=m+1}^{\infty} \alpha_{ij} \varphi_{j,g}(t) dt \right]^2 = 0 \quad (8.5.30)$$

Similarly, the final term in (8.5.28) is a zero mean, Gaussian random variable of variance

$$\sum_{j=m+1}^{\infty} \alpha_{ij}^2.$$

Thus this term converges in mean square to zero as  $m \rightarrow \infty$ , and

$$v_i = \lim_{m \rightarrow \infty} \sum_{j=1}^m \frac{\alpha_{ij}}{\sqrt{\lambda_{j,g}}} \int y(t) \theta_{j,g}(t) dt \quad (8.5.31)$$

completing the proof. |

Notice in (8.5.30) that we have not asserted that

$$\lim_{m \rightarrow \infty} \sum_{j=1}^m \frac{\alpha_{ij}}{\sqrt{\lambda_{j,g}}} \theta_{j,g}(t)$$

exists, and in fact this function does not exist in general. This means that  $v_i$  in general cannot be calculated exactly from  $y(t)$  by a single correlation operation, although it can be approximated as closely as desired by such an operation.

We have shown that the parallel channel model of Figure 8.5.7 is equivalent to the channel of Figure 8.5.1. Theorem 7.5.1 now determines the maximum average mutual information and Theorem 7.5.2 provides upper bounds to the minimum achievable error probability for any given input duration  $T$  and observation interval  $T_o$ . The energy constraint  $\mathcal{E}$  in those theorems is here replaced by  $ST$  and each noise variance  $\sigma_i^2$  is replaced by  $1/\lambda_i$ .

We next investigate the behavior of the eigenvalues  $\lambda_i$  in the limit as  $T$  and  $T_o$  approach  $\infty$ . We shall do this in a series of lemmas, first investigating the limiting behavior of the eigenvalues  $\mu_i$  of the filter  $K(t,\tau)$  and then relating the set  $\{\mu_i\}$  to the eigenvalues  $\lambda_i$  of the filter  $K_o(t,\tau)$ . The reader is advised to omit the proofs on the first reading. For a given interval  $T$ , the

eigenvalues  $\mu_i$  [hereafter denoted as  $\mu_i(T)$ ] are the solutions to the integral equation [see (8.4.10) and (8.4.56)].

$$\int_{-T/2}^{T/2} \mathcal{R}(\tau_1 - \tau_2) \phi_i(\tau_2) d\tau_2 = \mu_i(T) \phi_i(\tau_1); \quad -T/2 \leq \tau_1 \leq T/2 \quad (8.5.32)$$

where

$$\mathcal{R}(\tau) = \int \frac{|H_1(f)|^2}{N(f)} e^{j2\pi f\tau} df \quad (8.5.33)$$

The behavior of the set of eigenvalues  $\{\mu_i(T)\}$  in the limit as  $T \rightarrow \infty$  has been found by Kac, Murdock, and Szego (1953). Their result is given in the following lemma and the reader is referred to page 139 of Grenander and Szego (1958) for a proof of the result as stated here.

**LEMMA 8.5.2.** Let  $\mathcal{R}(\tau)$  be a real function with a real, absolutely integrable, bounded, Fourier transform  $F(f)$ . For each  $T > 0$ , let  $N_T(a,b)$  be the number of eigenvalues of (8.5.32) satisfying  $a \leq \mu_i(T) < b$ . Then if  $a$  and  $b$  are both positive (or both negative) and if the set of  $f$  for which  $F(f) = a$  or  $F(f) = b$  has zero measure [that is, if  $F(f)$  does not equal  $a$  or  $b$  over any nonzero interval], then

$$\lim_{T \rightarrow \infty} \frac{1}{T} N_T(a,b) = \int_{f:a \leq F(f) < b} df \quad (8.5.34)$$

Notice that (8.5.34) is just the result that would arise if the eigenfunctions of (8.5.32) approached the set of sines and cosines separated in frequency by  $1/T$  which we used in Section 8.3.

In our applications, we are interested not in the number of eigenvalues in a given range but in the asymptotic behavior of a sum of a function of the eigenvalues. The next lemma treats this problem.

**LEMMA 8.5.3.** Let  $\mathcal{R}(\tau)$  be a correlation function with eigenvalues  $\mu_i(T)$  given by (8.5.32), and let  $F(f) \geq 0$  be the Fourier transform of  $\mathcal{R}(\tau)$ . Assume that  $F(f)$  is integrable and bounded. Let  $g(x)$  be a nondecreasing function of  $x$  defined for  $x \geq 0$  with  $g(0) = 0$ . Let  $g(x)$  be of bounded slope, satisfying  $|g(x_1) - g(x_2)| \leq B |x_1 - x_2|$  for some fixed number  $B$  and all  $x_1 > 0, x_2 > 0$ . Then

$$\lim_{T \rightarrow \infty} \frac{1}{T} \sum_i g[\mu_i(T)] = \int g[F(f)] df \quad (8.5.35)$$

---

*Proof.* Since  $F(f)$  is bounded we can find a number  $A$  large enough so that

$$F(f) < A; \text{ all } f \quad (8.5.36)$$

Multiplying (8.5.32) by  $\varphi_i(\tau_1)$  and integrating both sides over  $\tau_1'$ , we get

$$\begin{aligned}\mu_i(T) &= \iint \varphi_i(\tau_1)\varphi_i(\tau_2)\mathcal{R}(\tau_1 - \tau_2) d\tau_1 d\tau_2 \\ &= \int |\Phi_i(f)|^2 F(f) df \\ < A \int |\Phi_i(f)|^2 df &= A\end{aligned}$$

Thus, for all  $T$ , all the eigenvalues are less than  $A$ . Now let  $\epsilon$  be an arbitrarily small positive number. Since  $F(f)$  is integrable we can pick  $f_1$  large enough that

$$\int_{|f| \geq f_1} F(f) df \leq \epsilon$$

Let  $a_0$  be a positive number,  $a_0 \leq \epsilon/f_1$ . Then

$$\int_{f: F(f) < a_0} F(f) df \leq \int_{\substack{f: F(f) < a_0 \\ |f| \leq f_1}} F(f) df + \int_{f: |f| \geq f_1} F(f) df \leq 2f_1 a_0 + \epsilon \leq 3\epsilon \quad (8.5.37)$$

Now let  $a_1 < \dots < a_n$  be a set of numbers with  $a_1 > a_0$ ,  $a_n = A$  and with

$$a_j - a_{j-1} \leq \epsilon a_0/B; \quad 1 \leq j \leq n$$

Furthermore, let these numbers be such that the set of  $f$  for which  $F(f) = a_j$  is of zero measure for  $0 \leq j \leq n$ . For any  $T > 0$ , we have

$$\sum_{j=1}^n g(a_{j-1}) N_T(a_{j-1}, a_j) \leq \sum_{j=1}^n \sum_{i: a_{j-1} \leq \mu_i(T) < a_j} g[\mu_i(T)] \leq \sum_{j=1}^n g(a_j) N_T(a_{j-1}, a_j) \quad (8.5.38)$$

where  $N_T(a_{j-1}, a_j)$  is the number of integers  $i$  for which  $a_{j-1} \leq \mu_i(T) < a_j$ . Observe that the center expression in (8.5.38) is just  $\sum g[\mu_i(T)]$  summed over those  $i$  for which  $a_0 \leq \mu_i(T)$ . Dividing all terms in (8.5.38) by  $T$ , passing to the limit  $T \rightarrow \infty$ , and using (8.5.34), we obtain

$$\begin{aligned}\sum_{j=1}^n \left[ g(a_{j-1}) \int_{f: a_{j-1} \leq F(f) < a_j} df \right] &\leq \lim_{T \rightarrow \infty} \frac{1}{T} \sum_{i: a_0 \leq \mu_i(T)} g[\mu_i(T)] \\ &\leq \sum_{j=1}^n \left[ g(a_j) \int_{f: a_{j-1} \leq F(f) < a_j} df \right] \quad (8.5.39)\end{aligned}$$

Observe that the outside terms in (8.5.39) also bound  $\int g[F(f)] df$ , where the integral is over the set of  $f$  for which  $a_0 \leq F(f)$ . The difference of the

outside terms thus bound the difference between the center term in (8.5.39) and this integral. Thus

$$\begin{aligned}
 & \left| \lim_{T \rightarrow \infty} \frac{1}{T} \sum_{a_0 \leq \mu_i(T)} g[\mu_i(T)] - \int_{f: a \leq F(f)} g[F(f)] df \right| \\
 & \leq \sum_{j=1}^n [g(a_j) - g(a_{j-1})] \int_{f: a_{j-1} \leq F(f) < a_j} df \\
 & \leq a_0 \epsilon \sum_{j=1}^n \int_{f: a_{j-1} \leq F(f) < a_j} df \\
 & = a_0 \epsilon \int_{f: F(f) \geq a_0} df \\
 & \leq a \epsilon \int_f \frac{F(f)}{a_0} df = \epsilon \int F(f) df
 \end{aligned} \tag{8.5.40}$$

Next, we must treat the values of  $i$  for which  $\mu_i(T) < a_0$ . We observe that, since the quantities  $a_0$  and  $A$  in (8.5.40) were selected independently of the function  $g$ , (8.5.40) is also valid for the special case  $g(x) = x$ , yielding

$$\left| \lim_{T \rightarrow \infty} \frac{1}{T} \sum_{a_0 \leq \mu_i(T)} \mu_i(T) - \int_{f: a_0 \leq F(f)} F(f) df \right| \leq \epsilon \int F(f) df \tag{8.5.41}$$

Now let  $K_1(t) = \int \sqrt{F(f)} e^{j2\pi f t} df$ . We have seen in (8.4.24) that, for any  $T$ ,

$$\begin{aligned}
 \sum_i \mu_i(T) &= \int_{\tau=-T/2}^{T/2} \int_{t=-\infty}^{\infty} K_1^2(t-\tau) dt d\tau \\
 &= T \int_{-\infty}^{\infty} K_1^2(t) dt = T \int F(f) df
 \end{aligned} \tag{8.5.42}$$

where in (8.5.42) we have used the Parseval relation for Fourier transforms (8.1.23). Substituting (8.5.42) into (8.5.41), we obtain

$$\left| \lim_{T \rightarrow \infty} \frac{1}{T} \sum_{\mu_i(T) < a_0} \mu_i(T) - \int_{f: F(f) < a_0} F(f) df \right| \leq \epsilon \int F(f) df \tag{8.5.43}$$

From (8.5.37), we then have

$$\lim_{T \rightarrow \infty} \frac{1}{T} \sum_{\mu_i(T) < a_0} \mu_i(T) \leq \epsilon \int F(f) df + 3\epsilon \tag{8.5.44}$$

Since  $g[\mu_i(T)] \leq B\mu_i(T)$ ,

$$\lim_{T \rightarrow \infty} \frac{1}{T} \sum_{\mu_i(T) < a_0} g[\mu_i(T)] \leq B\epsilon \left[ \int F(f) df + 3 \right] \quad (8.5.45)$$

Likewise

$$\int_{f: F(f) < a_0} g[F(f)] df \leq 3\epsilon B \quad (8.5.46)$$

Combining (8.5.45) and (8.5.46) with (8.5.40), we obtain

$$\left| \lim_{T \rightarrow \infty} \frac{1}{T} \sum_i g[\mu_i(T)] - \int_{-\infty}^{\infty} g[F(f)] df \right| \leq \epsilon \left[ (B+1) \int F(f) df + 6B \right] \quad (8.5.47)$$

Since  $\epsilon > 0$  is arbitrary, this completes the proof. |

The purpose of the next three lemmas is to show that, as  $T$  and  $T_o$  approach  $\infty$ , (8.5.34) also applies to the eigenvalues  $\lambda_i$  of  $K_o(t, \tau)$ .

**LEMMA 8.5.4.** For any given  $T$  and  $T_o$ , let  $\{\mu_i\}$  be the set of eigenvalues of  $K(t, \tau)$  and  $\{\lambda_i\}$  be the set of eigenvalues of  $K_o(t, \tau)$ , ordered as in Theorem 8.4.1. Then, for all  $i \geq 1$ ,

$$\lambda_i \leq \mu_i \quad (8.5.48)$$


---

*Proof.* The first  $i$  input eigenfunctions  $\phi_1(\tau), \dots, \phi_i(\tau)$  of  $K_o(t, \tau)$  are linearly independent and thus all cannot be linear combinations of the first  $i-1$  eigenfunctions  $\xi_1(\tau), \dots, \xi_{i-1}(\tau)$  of  $K(t, \tau)$ . Thus we can choose a function

$$x(\tau) = \sum_{j=1}^i x_j \phi_j(\tau) \quad (8.5.49)$$

which is a linear combination of  $\phi_1(\tau), \dots, \phi_i(\tau)$  but is orthogonal to  $\xi_1(\tau), \dots, \xi_{i-1}(\tau)$ . Then  $x(\tau)$  can also be represented in terms of the set  $\{\xi_i(\tau)\}$  by

$$x(\tau) = \sum_{j=i}^{\infty} y_j \xi_j(\tau); \quad y_j = \int \xi_j(\tau) x(\tau) d\tau \quad (8.5.50)$$

The function  $x(\tau)$  can be normalized to unit energy so that

$$\sum_{j=1}^i x_j^2 = 1 = \sum_{j=i}^{\infty} y_j^2.$$

The response of  $K_o(t, \tau)$  and of  $K(t, \tau)$  to  $x(\tau)$  is given by

$$u_o(t) = \int x(\tau) K_o(t, \tau) d\tau = \sum_{j=1}^i x_j \sqrt{\lambda_j} \theta_j(t) \quad (8.5.51)$$

$$u(t) = \int x(\tau) K(t, \tau) d\tau = \sum_{j=i}^{\infty} y_j \sqrt{\mu_j} \eta_j(t) \quad (8.5.52)$$

$$\int u_o^2(t) dt = \sum_{j=1}^i x_j^2 \lambda_j \geq \lambda_i \sum_{j=1}^i x_j^2 = \lambda_i \quad (8.5.53)$$

$$\int u^2(t) dt = \sum_{j=i}^{\infty} y_j^2 \mu_j \leq \mu_i \sum_{j=i}^{\infty} y_j^2 = \mu_i \quad (8.5.54)$$

For  $K_0$  given by (8.5.13),  $u_o(t)$  is the projection of  $u(t)$  on the space generated by  $\{\phi_{i,g}(t)\}$  and thus  $\int u_o^2(t) dt \leq \int u^2(t) dt$ . For  $K_0$  given by (8.5.14),  $u_o(t)$  is  $u(t)$  truncated to  $(-T_o/2, T_o/2)$  and the same result is valid. Combining this result with (8.5.53) and (8.5.54), we have  $\lambda_i \leq \mu_i$ . |

For simplicity, we now restrict the output interval  $T_o$  to be equal to the input interval  $T$ . Letting  $\lambda_i(T)$  and  $\mu_i(T)$  be the eigenvalues of  $K(t, \tau)$  and  $K_o(t, \tau)$ , we want to show that  $(1/T) \sum \lambda_i(T)$  approaches  $(1/T) \sum \mu_i(T)$  in the limit  $T \rightarrow \infty$ . The key to this is the following lemma, which is a slight generalization of a result by Kelly, Reed, and Root.\*

**LEMMA 8.5.5.** For an  $L_2$  function  $g_1(t)$ , let

$$g_T(t, \tau) = \begin{cases} g_1(t - \tau); & t \leq T/2 \\ 0 & ; t > T/2 \end{cases}$$

That is,  $g_T(t, \tau)$  represents the filter  $g_1(t)$  with the output truncated to  $(-T/2, T/2)$ . Let  $\{\varphi_{i,T}(\tau)\}$  be the input eigenfunctions of  $g_T(t, \tau)$  in the sense of Theorem 8.4.1. Let  $T_1 > 0$  and  $\tau_1$  satisfy

$$T \geq T_1 + 2|\tau_1| \quad (8.5.55)$$

Then, if a function  $v(\tau)$  is orthogonal to  $\phi_{i,T}(\tau + \tau_1)$  for all  $i$ ,  $v(\tau)$  is also orthogonal to  $\phi_{i,T_1}(\tau)$  for all  $i$ . Also, if  $v(\tau)$  is expanded as

$$v(\tau) = \sum_i v_{i,T} \phi_{i,T}(\tau) + v_{r,T}(\tau); \quad v_{i,T} = (v, \phi_{i,T}) \quad (8.5.56)$$

and if the Fourier transform of  $v(\tau)$  is zero wherever the Fourier transform of  $g_1(t)$  is 0, then

$$\lim_{T \rightarrow \infty} v_{r,T}(\tau) = 0 \quad (8.5.57)$$

---

\* Kelly, Reed, and Root, "The Detection of Radar Echoes in Noise, I," *Jour. Siam*, **8** (2), June 1960 (see Appendix A).

Roughly, the lemma states that as  $T$  gets large, we can represent more and more of an arbitrary function in terms of the set  $\{\phi_{i,T}\}$  or of time translates of the  $\phi_{i,T}$ .

*Proof.* If  $v(\tau)$  is orthogonal to  $\phi_{i,T}(\tau + \tau_1)$  for all  $i$ , then

$$\int v(\tau - \tau_1) \phi_{i,T}(\tau) d\tau = 0; \quad \text{all } i$$

From (8.4.15), this implies that

$$\int g_1(t - \tau) v(\tau - \tau_1) d\tau = 0; \quad |t| \leq T/2$$

Substituting  $\tau_2 = \tau - \tau_1$  and  $t_2 = t + \tau_1$ ,

$$\int g_1(t_2 - \tau_2) v(\tau_2) d\tau_2 = 0; \quad |t_2 - \tau_1| \leq T/2 \quad (8.5.58)$$

From (8.5.55), (8.5.58) must be satisfied for  $|t_2| \leq T_1/2$ , and using the implication of (8.4.15) in the reverse direction,  $v(\tau)$  is orthogonal to  $\phi_{i,T_1}(\tau)$  for all  $i$ .

Next, we show that  $v_{r,T}(\tau) \rightarrow 0$  as  $T \rightarrow \infty$ . Since, by definition,  $v_{r,T}(\tau)$  is orthogonal to  $\phi_{i,T}(\tau)$  for all  $i$ , we can use (8.5.56) to obtain

$$\int v_{r,T}(\tau) v(\tau) d\tau = \int v_{r,T}^2(\tau) d\tau = \|v_{r,T}\|^2 \quad (8.5.59)$$

Also, for  $T > T_1$ ,  $v_{r,T}(\tau)$  is orthogonal to  $\phi_{i,T_1}(\tau)$  for all  $i$ . Using  $T_1$  for  $T$  in (8.5.56) and substituting this in (8.5.59), we obtain

$$\int v_{r,T}(\tau) v_{r,T_1}(\tau) d\tau = \|v_{r,T}\|^2 \quad (8.5.60)$$

Applying the Schwarz inequality to the left side of (8.5.60), we see that

$$\|v_{r,T}\| \leq \|v_{r,T_1}\|; \quad T_1 \leq T$$

Thus  $\|v_{r,T}\|$  is decreasing with  $T$  and must approach a limit. Finally, using (8.5.59) and (8.5.60), we have

$$\int [v_{r,T_1}(\tau) - v_{r,T}(\tau)]^2 d\tau = \|v_{r,T_1}\|^2 - \|v_{r,T}\|^2 \quad (8.5.61)$$

Since  $\|v_{r,T}\|$  approaches a limit, the limit of the right side of (8.5.60) as  $T$  and  $T_1$  approach  $\infty$  is 0. Thus  $v_{r,T}(\tau)$  approaches a limit function,\*  $v_{r,\infty}(\tau)$ , and this limit function is orthogonal to  $\phi_{i,T}(\tau)$  for all  $i$  and all  $T$ . From (8.4.15), however, this implies that

$$\int g_1(t - \tau) v_{r,\infty}(\tau) d\tau = 0; \quad \text{all } t \quad (8.5.62)$$

\* This is guaranteed by the Riesz-Fischer Theorem, Riesz and Nagy (1955) p. 59.

Taking the Fourier transform of (8.5.62), we have

$$\sqrt{N(f)} V_{r,\infty}(f) = 0; \quad \text{all } f$$

By hypothesis,  $V_{r,\infty}(f) = 0$  where  $\sqrt{N(f)}$  is 0; thus  $V_{r,\infty}(f) = 0$  and (8.5.57) is established. |

**LEMMA 8.5.6.** Let  $\mu_i(T)$  and  $\lambda_i(T)$  be the eigenvalues of the filters  $K_T(t,\tau)$  and  $K_{o,T}(t,\tau)$  where  $K_T(t,\tau)$  is  $K_1(t - \tau)$  truncated to  $|\tau| \leq T/2$  and  $K_{o,T}(t,\tau)$  is given by (8.5.13) or (8.5.14) with  $T_o = T$ . Then

$$\lim_{T \rightarrow \infty} \frac{1}{T} \sum_i \lambda_i(T) = \lim_{T \rightarrow \infty} \frac{1}{T} \sum_i \mu_i(T) \quad (8.5.63)$$


---

*Proof.* Using (8.4.24), we have

$$\frac{1}{T} \sum_i \mu_i(T) = \frac{1}{T} \iint K_T^2(t,\tau) dt d\tau = \int K_1^2(t) dt \quad (8.5.64)$$

Since  $\lambda_i(T) \leq \mu_i(T)$ , we can prove the lemma by showing that, for every  $\epsilon > 0$ ,

$$\lim_{T \rightarrow \infty} \frac{1}{T} \iint K_{o,T}^2(t,\tau) dt d\tau \geq \int K_1^2(t) dt - \epsilon \quad (8.5.65)$$

For  $K_{o,T}$  given by (8.5.13), we take  $K_1(\tau)$  as the  $v(\tau)$  in the previous lemma, and from (8.5.13),  $K_1(t) - K_{o,T}(t,0)$  plays the role of  $v_{r,T}(\tau)$ . Since  $H_1(f)/\sqrt{N(f)} = 0$  whenever  $\sqrt{N(f)} = 0$ , the previous lemma asserts that  $K_{o,T}(t,0)$  approaches  $K_1(t)$  as  $T$  approaches infinity, and for any  $\epsilon > 0$ , we can pick  $T_\epsilon$  such that

$$\int K_{o,T}^2(t,0) dt \geq \int K_1^2(t) dt - \epsilon; \quad T \geq T_\epsilon \quad (8.5.66)$$

Next, we can write out

$$K_{o,T}(t + \tau, \tau) = \sum_i \phi_{i,T}(t + \tau) \int_{t_1} \phi_{i,T}(t_1 + \tau) K_1(t_1) dt_1 \quad (8.5.67)$$

Thus  $K_{o,T}(t + \tau, \tau)$  is the projection of  $K_1(t)$  on the set of functions  $\phi_{i,T}(t + \tau)$ . The remainder [that is, the portion of  $K_1(t)$  orthogonal to all  $\phi_{i,T}(t + \tau)$ ] is, by the previous lemma, also orthogonal to all  $\phi_{i,T_\epsilon}(t)$  if  $T \geq T_\epsilon + 2|\tau|$ . Thus the energy in the remainder is upper bounded by the energy in the remainder of the expansion of  $K_1(t)$  in terms of the set  $\{\phi_{i,T_\epsilon}(t)\}$ . From (8.5.66), this energy is upper bounded by  $\epsilon$ , and

$$\int K_{o,T}^2(t + \tau, \tau) dt \geq \int K_1^2(t) dt - \epsilon; \quad T \geq T_\epsilon + 2|\tau| \quad (8.5.68)$$

$$\frac{1}{T} \int_{\tau=-T/2}^{T/2} \int_{t=-\infty}^{\infty} K_{o,T}^2(t, \tau) dt d\tau \geq \frac{1}{T} \int_{|t| \leq \frac{T-T_\epsilon}{2}} \int_t K_{o,T}^2(t, \tau) dt d\tau \quad (8.5.69)$$

Equation 8.5.68 is valid over the range of  $\tau$  on the right side of (8.5.69).

$$\frac{1}{T} \int \int K_{o,T}^2(t,\tau) dt d\tau \geq \left( \frac{T - T_\epsilon}{T} \right) \left[ \int K_1^2(t) dt - \epsilon \right] \quad (8.5.70)$$

Taking the limit of (8.5.70) as  $T \rightarrow \infty$ , we have (8.5.65).

If the channel noise is white, then  $K_{o,T}(t,\tau)$  is given by (8.5.14), and we see immediately that (8.5.68) is again satisfied for any  $\epsilon > 0$  and sufficiently large  $T_\epsilon$ . Thus (8.5.63) follows as before. |

The following lemma now ties our results together.

**LEMMA 8.5.7.** For the channel of Figure 8.5.1, let  $T_o = T$ , and assume that  $F(f) = |H_1(f)|^2/N(f)$  is bounded and integrable. Let  $g(x)$  be a non-decreasing function of bounded slope defined for  $x > 0$  with  $g(0) = 0$ . Then the eigenvalues  $\lambda_i(T)$  of  $K_o(t,\tau)$  satisfy

$$\lim_{T \rightarrow \infty} \frac{1}{T} \sum_i g[\lambda_i(T)] = \int g \left[ \frac{H_1(f)^2}{N(f)} \right] df \quad (8.5.71)$$

---

*Proof.* Let  $B$  be an upper bound on the slope of  $g$ . Then, since  $\lambda_i(T) \leq \mu_i(T)$ , we have

$$\begin{aligned} 0 &\leq g[\mu_i(T)] - g[\lambda_i(T)] \leq B[\mu_i(T) - \lambda_i(T)] \\ 0 &\leq \frac{1}{T} \sum_i g[\mu_i(T)] - \frac{1}{T} \sum_i g[\lambda_i(T)] \leq \frac{B}{T} \left[ \sum_i \mu_i(T) - \sum_i \lambda_i(T) \right] \end{aligned}$$

From Lemma 8.5.6, it follows that

$$\lim_{T \rightarrow \infty} \left\{ \frac{1}{T} \sum_i g[\mu_i(T)] - \frac{1}{T} \sum_i g[\lambda_i(T)] \right\} = 0$$

Combining this with Lemma 8.5.3 completes the proof. |

We can now use this lemma to find the capacity of the channel in Figure 8.5.1 subject to a power constraint and to find exponential bounds on error probability. To define the capacity for a power limitation  $S$  on the input, we first define  $C_T$  as  $1/T$  times the maximum average mutual information between  $x(t)$  and  $y(t)$  when  $x(t)$  and  $y(t)$  are limited to the interval  $(-T/2, T/2)$  and the maximization is over all input probability distributions subject to the expected value of  $\int x^2(t) dt$  being at most  $ST$ . In principle, we have found  $C_T$  from the parallel channel representation of Figure 8.5.7 combined with Theorem 7.5.1. The capacity  $C$  is then defined as

$$C = \lim_{T \rightarrow \infty} C_T$$

The existence of this limit will be demonstrated in the following theorem.

**Theorem 8.5.1.** For the channel of Figure 8.5.1 with a power constraint  $S$  and with  $T_o = T$ , assume that  $H_1(f)^2/N(f)$  is bounded and integrable, and that either  $\int N(f) df < \infty$  or that  $N(f)$  is white. Then the capacity  $C$  of the channel is given parametrically by

$$C = \int_{f: \frac{N(f)}{|H_1(f)|^2} \leq B} \frac{1}{2} \log \left[ \frac{|H_1(f)|^2 B}{N(f)} \right] df \quad (8.5.72)$$

$$S = \int_{f: \frac{N(f)}{|H_1(f)|^2} \leq B} \left[ B - \frac{N(f)}{|H_1(f)|^2} \right] df \quad (8.5.73)$$


---

*Proof.* From Theorem 7.5.1, the maximum average mutual information per unit time for the set of parallel channels in Figure 8.5.7 is related to the power constraint by the parametric equations

$$\tilde{C}_T(B) = \frac{1}{T} \sum_{i: \lambda_i(T) \geq (1/B)} \frac{1}{2} \log [\lambda_i(T) B] \quad (8.5.74)$$

$$\tilde{S}_T(B) = \frac{1}{T} \sum_{i: \lambda_i(T) \geq (1/B)} \left[ B - \frac{1}{\lambda_i(T)} \right] \quad (8.5.75)$$

That is, for a power constraint  $S$ , we find the value of  $B$  that satisfies  $S = \tilde{S}_T(B)$ , and using that value of  $B$ ,  $C_T = \tilde{C}_T(B)$ . If, for a given  $B$ , we define the function

$$g(x) = \begin{cases} 0; & x \leq 1/B \\ \frac{1}{2} \log (xB); & x \geq 1/B \end{cases} \quad (8.5.76)$$

then (8.5.74) can be rewritten as

$$\tilde{C}_T(B) = \frac{1}{T} \sum_i g[\lambda_i(T)] \quad (8.5.77)$$

Using Lemma 8.5.7, we then have

$$\begin{aligned} \tilde{C}_\infty(B) &\triangleq \lim_{T \rightarrow \infty} \tilde{C}_T(B) = \int g \left[ \frac{|H_1(f)|^2}{N(f)} \right] df \\ &= \int_{f: \frac{|H_1(f)|^2}{N(f)} \geq \frac{1}{B}} \frac{1}{2} \log \left[ \frac{|H_1(f)|^2}{N(f)} B \right] df \end{aligned} \quad (8.5.78)$$

We can apply the same lemma to (8.5.73) by redefining  $g(x)$  to be 0 for  $x \leq 1/B$  and  $B - 1/x$  for  $x \geq 1/B$ . Thus

$$\tilde{S}_\infty(B) \triangleq \lim_{T \rightarrow \infty} \tilde{S}_T(B) = \int_{f: \frac{|H_1(f)|^2}{N(f)} \geq \frac{1}{B}} \left[ B - \frac{N(f)}{|H_1(f)|^2} \right] df \quad (8.5.79)$$

For the given power constraint  $S$ , let  $B$  satisfy  $S = \tilde{S}_\infty(B)$ . Let  $\epsilon > 0$  be arbitrarily small. Since  $\tilde{C}_\infty(B)$  is a continuous function, there exists a  $\delta > 0$  such that

$$\tilde{C}_\infty(B + \delta) \leq \tilde{C}_\infty(B) + \epsilon$$

For this  $\delta$ , there exists some  $T_1$  so that, for  $T \geq T_1$ ,

$$\tilde{C}_T(B + \delta) \leq \tilde{C}_\infty(B + \delta) + \epsilon \leq \tilde{C}_\infty(B) + 2\epsilon \quad (8.5.80)$$

On the other hand, since  $\tilde{S}_\infty(B)$  is strictly monotone increasing in  $B$  [for  $B > \inf N(f)/|H_1(f)|^2$ ] we have  $\tilde{S}_\infty(B + \delta) > \tilde{S}_\infty(B)$ , and there exists a  $T_2$  so that for  $T \geq T_2$ ,

$$\tilde{S}_T(B + \delta) \geq \tilde{S}_\infty(B) = S \quad (8.5.81)$$

Finally, for any  $T$ ,  $C_T$  is a monotonic increasing function of the power constraint, so that using (8.5.81) for  $T \geq T_2$ , we have  $C_T \leq \tilde{C}_T(B + \delta)$ . Consequently, for  $T \geq T_1$ ,  $T \geq T_2$ , we have  $C_T \leq C_\infty(B) + 2\epsilon$ . Reversing the inequalities in the above argument, we also have  $C_T \geq C_\infty(B) - 2\epsilon$  for all sufficiently large  $T$ . Since  $\epsilon$  is arbitrary, it follows that

$$\lim_{T \rightarrow \infty} C_T = \tilde{C}_\infty(B)$$

completing the proof. |

The converse to the coding theorem applies here in the same way as for discrete channels. To be specific, we have the following theorem. The proof will be omitted since it is virtually identical to those in Chapters 4, 6 and 7.

**Theorem 8.5.2.** Let a discrete stationary source with an alphabet size  $M$  have an entropy  $H_\infty(U)$  and generate one letter each  $\tau_s$  seconds. Let a sequence of source letters of arbitrary length  $L$  be connected to a destination through the use of a continuous time channel for  $T = L\tau_s$  seconds. Let  $C_T$  be the supremum, over all input probability assignments, of  $1/T$  times the average mutual information between channel input and output for this interval. Assume that  $\lim_{T \rightarrow \infty} C_T$  exists and define

$$C = \lim_{T \rightarrow \infty} C_T$$

Then, for any  $\epsilon > 0$ , and for all sufficiently large  $L$ , the error probability per digit  $\langle P_e \rangle$  over the sequence of  $L$  source letters satisfies

$$\langle P_e \rangle \log (M - 1) + \mathcal{H}(\langle P_e \rangle) \geq H_\infty(U) - \tau_s C - \epsilon$$


---

We can apply the same arguments to the random coding bound and expurgated random-coding bound of Theorem 7.5.2. Define

$$\tilde{S}_T(B, \rho) = \frac{1}{T} \sum_{i: \lambda_i(T) \geq \frac{1}{B}} \frac{(1 + \rho)^2 [B\lambda_i(T) - 1]B}{(1 + \rho)B\lambda_i(T) - \rho} \quad (8.5.82)$$

$$\tilde{R}_T(B) = \frac{1}{T} \sum_{i: \lambda_i(T) \geq \frac{1}{B}} \frac{1}{2} \ln [B\lambda_i(T)] \quad (8.5.83)$$

$$\tilde{E}_T(B, \rho) = \frac{\rho}{2B(1 + \rho)} - \frac{1}{T} \sum_{i: \lambda_i(T) \geq \frac{1}{B}} \frac{1}{2} \ln \left[ 1 + \rho - \frac{\rho}{B\lambda_i(T)} \right] \quad (8.5.84)$$

From Theorem 7.5.2, for any choice of  $\rho$ ,  $0 < \rho \leq 1$ , and any  $B > 0$ , there exists a code with  $M = [\exp[T\tilde{R}_T(B)]]$  code words, each constrained to the interval  $(-T/2, T/2)$ , each with energy at most  $T\tilde{S}_T(B, \rho)$ , and each with an error probability bounded by

$$P_{e,m} \leq \left[ \frac{2e^{s\delta}}{\mu} \right]^2 \exp [-T\tilde{E}_T(B, \rho)] \quad (8.5.85)$$

For fixed  $B$  and  $\rho$ , we can apply Lemma 8.5.7 to (8.5.82), (8.5.83), and (8.5.84) in the same way as in Theorem 8.5.1, obtaining

$$\tilde{S}_\infty(B, \rho) = \int_{f: \frac{|H_1(f)|^2}{N(f)} \geq \frac{1}{B}} \frac{(1 + \rho)^2 [B|H_1(f)|^2 - N(f)]B}{(1 + \rho)B|H_1(f)|^2 - \rho N(f)} df \quad (8.5.86)$$

$$\tilde{R}_\infty(B) = \int_{f: \frac{|H_1(f)|^2}{N(f)} \geq \frac{1}{B}} \frac{1}{2} \ln \left[ B \frac{|H_1(f)|^2}{N(f)} \right] df \quad (8.5.87)$$

$$\tilde{E}_\infty(B, \rho) = \frac{\rho \tilde{S}_\infty(B, \rho)}{2B(1 + \rho)} - \int_{f: \frac{|H_1(f)|^2}{N(f)} \geq \frac{1}{B}} \frac{1}{2} \ln \left[ 1 + \rho - \frac{\rho N(f)}{B|H_1(f)|^2} \right] df \quad (8.5.88)$$

By going through the same kind of  $\epsilon, \delta$  argument as in the preceding theorem, we find that, for any  $\epsilon > 0$ , there is a  $T_1$  so that for all  $T \geq T_1$ , there are codes with  $M = [\exp[T\tilde{R}_\infty(B)]]$  code words, each constrained to

$(-T/2, T/2)$ , each with energy at most  $T\tilde{S}_\infty(B, \rho)$ , and each with error probability bounded by

$$P_{e,m} \leq \left[ \frac{2e^{s\delta}}{\mu} \right]^2 \exp \{-T[\tilde{E}_\infty(B, \rho) - \epsilon]\} \quad (8.5.89)$$

As shown in (7.5.39), the coefficient is given approximately by

$$\frac{2e^{s\delta}}{\mu} \approx \frac{e\rho\sqrt{4\pi \sum_i \mathcal{E}_i^2}}{(1 + \rho)^2 B} \quad (8.5.90)$$

where

$$\mathcal{E}_i = \begin{cases} \frac{(1 + \rho)^2[B\lambda_i(T) - 1]B}{(1 + \rho)B\lambda_i(T) - \rho}; & \lambda_i(T) \geq \frac{1}{B} \\ 0; & \lambda_i(T) < \frac{1}{B} \end{cases} \quad (8.5.91)$$

It is easy to see from (8.5.91) that  $\mathcal{E}_i \leq (1 + \rho)B$ , and thus

$$\sum_i \mathcal{E}_i^2 \leq (1 + \rho)B \sum_i \mathcal{E}_i = (1 + \rho)B\tilde{S}_T(B, \rho)T \quad (8.5.92)$$

From (8.5.92) we see that, for fixed  $B$  and  $\rho$ , the approximation in (8.5.90) is proportional to  $\sqrt{T}$  for large  $T$  and also the approximation becomes better with increasing  $T$ . It follows that, for large enough  $T$ , the coefficient in (8.5.89) can be incorporated into the  $\epsilon$ . Thus, for any  $\epsilon$  there exists a  $T_2$  (depending on  $\epsilon$ ,  $B$ , and  $\rho$ ) so that for  $T \geq T_2$

$$P_{e,m} \leq \exp \{-T[\tilde{E}_\infty(B, \rho) - \epsilon]\} \quad (8.5.93)$$

As in Section 7.5, this bound on error probability is only valid over a range of rates [ $R = \tilde{R}_\infty(B)$ ] and powers [ $S = \tilde{S}_\infty(B, \rho)$ ] since  $\rho$  is constrained to the range  $0 < \rho \leq 1$ . Since  $\tilde{S}_\infty(B, \rho)$  is strictly increasing and continuous in  $B$  and  $\rho$  [for  $B \geq \inf N(f)/|H_1(f)|^2$ ], the equation  $S = \tilde{S}_\infty(B, \rho)$  determines  $B$  as a function of  $\rho$  for a fixed  $S$ , and this implicitly determines  $\tilde{R}_\infty(B)$  as a function of  $\rho$ . For  $\rho = 0$ ,  $\tilde{R}_\infty(B)$  is just the channel capacity for the given  $S$ , as can be seen by comparing (8.5.86) and (8.5.87) with (8.5.72) and (8.5.73). Also, for  $\rho = 0$ ,  $\tilde{E}_\infty(B, \rho) = 0$ . As  $\rho$  increases, holding  $S$  fixed (and thus varying  $B$ ),  $\tilde{R}_\infty(B)$  decreases and  $\tilde{E}_\infty(B, \rho)$  increases. As before, the slope of  $E$  as a function of  $R$  for fixed  $S$  is  $-\rho$ . When  $\rho$  increases to 1,  $B$  has decreased to a critical value  $B_{cr}$  given by

$$S = \tilde{S}_\infty(B_{cr}, 1) = \int_{f: \frac{|H_1(f)|^2}{N(f)} \geq \frac{1}{B_{cr}}} \frac{4B_{cr}[B_{cr}|H_1(f)|^2 - N(f)]}{2B_{cr}|H_1(f)|^2 - N(f)} df \quad (8.5.94)$$

Corresponding to  $B_{cr}$  is a critical value of  $R$  given by

$$R_{cr} = \tilde{R}_\infty(B_{cr}) = \int_{f: \frac{|H_1(f)|^2}{N(f)} \geq \frac{1}{B_{cr}}} \frac{1}{2} \ln \left[ B_{cr} \frac{|H_1(f)|^2}{N(f)} \right] df \quad (8.5.95)$$

Thus, for a fixed  $S$ , the bound on error probability in (8.5.93) is valid for rates in the range  $R_{cr} \leq R < C$ .

For rates  $R < R_{cr}$ , and any given  $T$ , we have the bound on error probability given by (7.5.60). By passing to the limit as above, we find that for any  $\epsilon > 0$ , there is a  $T_1$  sufficiently large that for all  $T \geq T_1$ , there is a code with  $M = [\exp(RT)]$  code words, each constrained to  $(-T/2, T/2)$ , and with energy at most  $ST$ , and each with

$$P_{e,m} \leq \exp\{-T[\tilde{E}_\infty(B_{cr}, 1) + R_{cr} - R - \epsilon]\} \quad (8.5.96)$$

Finally, for the expurgated bound, we define

$$\tilde{S}_{x,T}(B, \rho) = \frac{1}{T} \sum_{i: \lambda_i(T) \geq \frac{1}{B}} \frac{4\rho B[B\lambda_i(T) - 1]}{2B\lambda_i(T) - 1} \quad (8.5.97)$$

$$\tilde{R}'_{x,T}(B) = \frac{1}{T} \sum_{i: \lambda_i(T) \geq \frac{1}{B}} \frac{1}{2} \ln \frac{B^2 \lambda_i^2(T)}{2B\lambda_i(T) - 1} \quad (8.5.98)$$

$$\tilde{E}_{x,T}(B, \rho) = \frac{\tilde{S}_{x,T}(B, \rho)}{4B} \quad (8.5.99)$$

From Theorem 7.5.2, for any  $\rho > 1$  and any  $B > 0$ , there exists a code with  $M = [\exp[T\tilde{R}'_{x,T}(B) - 2 \ln(2e^{s\delta}/\mu)]]$  code words, each constrained to  $(-T/2, T/2)$ , each with energy at most  $T\tilde{S}_x(B, \rho)$ , and each with

$$P_{e,m} \leq \exp[-T\tilde{E}_x(B, \rho)] \quad (8.5.100)$$

For fixed  $B$  and  $\rho$ , we can apply Lemma 8.5.7 to (8.5.97) and then to (8.5.99), obtaining

$$\tilde{S}_{x,\infty}(B, \rho) = \lim_{T \rightarrow \infty} \tilde{S}_{x,T}(B, \rho) = \int_{f: \frac{|H_1(f)|^2}{N(f)} \geq \frac{1}{B}} \frac{4\rho B[B|H_1(f)|^2 - N(f)]}{2B|H_1(f)|^2 - N(f)} df \quad (8.5.101)$$

$$\tilde{R}'_{x,\infty}(B) = \lim_{T \rightarrow \infty} \tilde{R}'_{x,T}(B) = \int_{f: \frac{|H_1(f)|^2}{N(f)} \geq \frac{1}{B}} \frac{1}{2} \ln \frac{B^2 |H_1(f)|^4 / N(f)}{2B|H_1(f)|^2 - N(f)} df \quad (8.5.102)$$

$$\tilde{E}_{x,\infty}(B, \rho) = \lim_{T \rightarrow \infty} \tilde{E}_{x,T}(B, \rho) = \frac{\tilde{S}_{x,\infty}(B, \rho)}{4B} \quad (8.5.103)$$

If we define

$$\tilde{R}_{x,T}(B, \rho) = \tilde{R}'_{x,T}(B) - \frac{2}{T} \ln \frac{2e^{s\delta}}{\mu} \quad (8.5.104)$$

then, from the same argument used in (8.5.90) to (8.5.92), it is clear that

$$\tilde{R}_{x,\infty}(B,\rho) = \lim_{T \rightarrow \infty} \tilde{R}_{x,T}(B,\rho) = \tilde{R}'_{x,\infty}(B) \quad (8.5.105)$$

Finally, by using the type of  $\epsilon, \delta$  argument used in Theorem 8.5.1, we have the result that, for any  $B > 0$ , any  $\rho > 1$ , and any arbitrarily small  $\epsilon > 0$ , there is a  $T_1$  so that for  $T \geq T_1$  there is a code with  $M = [\exp [T\tilde{R}_{x,\infty}(B,\rho)]]$  code words, each constrained to  $(-T/2, T/2)$ , each of energy at most  $\tilde{S}_{x,\infty}(B,\rho)$ , and each with

$$P_{e,m} \leq \exp\{-T[\tilde{E}_{x,\infty}(B,\rho) - \epsilon]\} \quad (8.5.106)$$

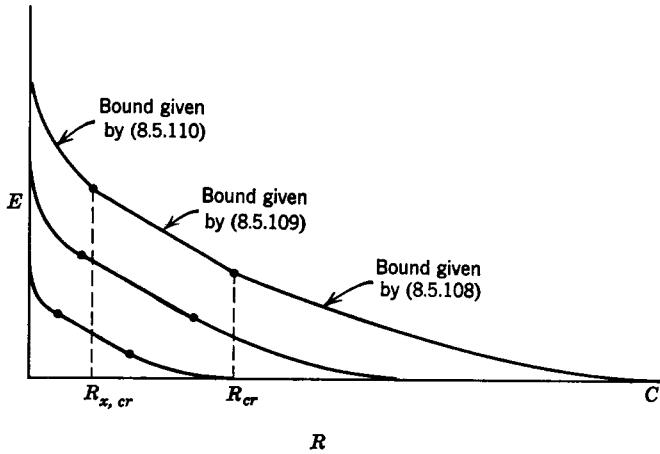


Figure 8.5.8. Exponent-rate curves for fixed power  $S$  (different curves corresponding to different  $S$ ).

For a fixed power  $S = \tilde{S}_{x,\infty}(B,\rho)$ , as  $\rho$  increases from 1,  $B$  decreases from  $B_{cr}$  [given by (8.5.94)] and approaches

$$\min_f \frac{N(f)}{|H_1(f)|^2}$$

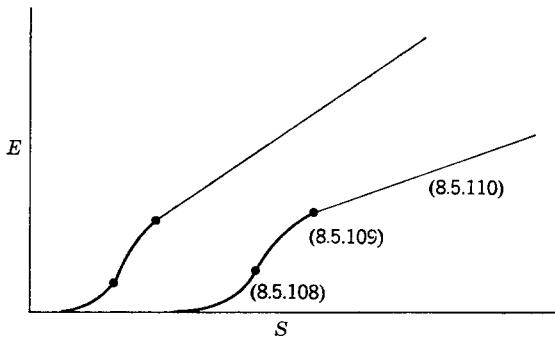
as  $\rho \rightarrow \infty$ . Correspondingly, as  $\rho$  increases from 1,  $\tilde{R}_{x,\infty}(B,\rho)$  decreases from  $R_{x,cr}$  to 0 where

$$R_{x,cr} = \int_{f: \frac{|H_1(f)|^2}{N(f)} > \frac{1}{B_{cr}}} \frac{1}{2} \ln \frac{B_{cr}^2 |H_1(f)|^4 / N(f)}{2B_{cr} |H_1(f)|^2 - N(f)} df \quad (8.5.107)$$

Likewise,  $\tilde{E}_{x,\infty}(B,\rho)$  increases, approaching a value

$$\frac{S}{4} \max_f \frac{|H_1(f)|^2}{N(f)}$$

as  $\rho \rightarrow \infty$  and  $R \rightarrow 0$ . Figures 8.5.8 and 8.5.9 sketch the behavior of these



**Figure 8.5.9.** Exponent-power curves for fixed rate  $R$  (higher curve corresponding to lower rate).

exponents as a function of rate  $R$  and power  $S$ . Our results are summarized in the following theorem.

**Theorem 8.5.2.** For the channel of Figure 8.5.1 with  $T_o = T$ , assume the same conditions as in Theorem 8.5.1. Then for any  $B > 0$ , any  $\rho$ ,  $0 \leq \rho \leq 1$ , and any arbitrarily small  $\epsilon > 0$ , there exists a  $T_1(\epsilon, B, \rho)$  such that for any  $T \geq T_1(\epsilon, B, \rho)$  there exists a code with  $M = [\exp [T\tilde{R}_\infty(B)]]$  code words, each limited to  $(-T/2, T/2)$ , each with energy at most  $TS_\infty(B, \rho)$ , and each satisfying

$$P_{e,m} \leq \exp \{-T[\tilde{E}_\infty(B, \rho) - \epsilon]\} \quad (8.5.108)$$

For fixed  $S = \tilde{S}_\infty(B, \rho)$ ,  $\tilde{R}_\infty(B)$  is strictly and continuously decreasing from  $C$  to  $R_{cr}$  as  $\rho$  goes from 0 to 1 and  $\tilde{E}_\infty(B, \rho)$  is strictly and continuously increasing from 0 as  $\rho$  goes from 0 to 1 [that is,  $\tilde{E}_\infty(B, \rho) > 0$  for every  $\tilde{R}_\infty(B) < C$ ]. For the same fixed  $S$  and  $T \geq T_1(\epsilon, B_{cr}, 1)$ , and any  $R \leq R_{cr}$ , there exists a code with  $M = [\exp (TR)]$  code words, each limited to  $(-T/2, T/2)$ , each with energy at most  $TS$ , and each satisfying

$$P_{e,m} \leq \exp \{-T[\tilde{E}_\infty(B_{cr}, 1) + R_{cr} - R - \epsilon]\} \quad (8.5.109)$$

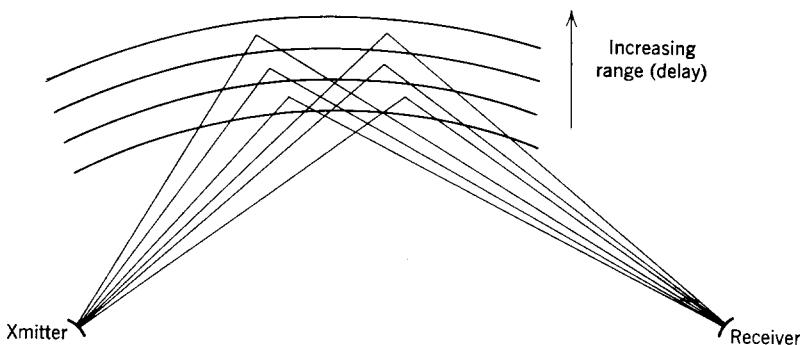
Finally, for any  $B > 0$ , any  $\rho > 1$ , and any  $\epsilon > 0$ , there exists a  $T_1(\epsilon, B, \rho)$  such that for  $T \geq T_1(\epsilon, B, \rho)$ , there exists a code of  $M = [\exp [T\tilde{R}_{x,\infty}(B, \rho)]]$  code words, each limited to  $(-T/2, T/2)$ , each with energy at most  $T\tilde{S}_{x,\infty}(B, \rho)$ , and each satisfying

$$P_{e,m} \leq \exp \{-T[\tilde{E}_{x,\infty}(B, \rho) - \epsilon]\} \quad (8.5.110)$$

For fixed  $S = \tilde{S}_{x,\infty}(B, \rho)$ ,  $\tilde{R}_{x,\infty}(B, \rho)$  is decreasing with increasing  $\rho$  from  $R_{x,cr}$  to 0 and  $\tilde{E}_{x,\infty}(B, \rho)$  is increasing with increasing  $\rho$ . For a fixed  $S$ , the exponent, rate curve defined by these three bounds (using the second for  $R_{x,cr} < R < R_{cr}$ ) is continuous with a continuous derivative.  $\rule{1cm}{0.4pt}$

## 8.6 Fading Dispersive Channels

In the previous sections, we treated channel models where the received signal was the sum of the transmitted signal (perhaps filtered and attenuated in a known way) and Gaussian noise. Such a model is usually appropriate for space-communication channels and, somewhat more tenuously, for wire and cable communication. In many communication systems, however, the transmission path is varying with time in a way that is only statistically predictable. The variations in the path give rise to time variations in the received signal energy, called fading, and also to a time-varying dispersion of the received signal. Communication systems employing tropospheric scatter, orbital dipole scatter, and high-frequency radio transmission are all particularly susceptible to these effects. In what follows, we shall first motivate



**Figure 8.6.1. Fading dispersive channel.**

a mathematical model for communication over such channels and then prove a coding theorem for the model. Our development of the model will be quite sketchy and the reader is referred to Kennedy (1969) for a careful discussion of such models.

We can most easily visualize the behavior of such systems in terms of an electromagnetic wave being scattered by a cloud of scattering particles as shown in Figure 8.6.1. The received signal (apart from any additive noise) can be viewed as a weighted sum of delayed replicas of the transmitted waveform, each replica corresponding to the scattering from one of the range intervals shown. Within any range interval, the scattering particles will typically be both moving and rotating, so that each scattering particle will introduce a certain amount of doppler shift into the received waveform. Thus, if a sinusoid,  $\cos 2\pi f_c t$ , is transmitted, the return from any particular range interval will be spread out over a range of frequencies around  $f_c$ . The *scattering function*  $\sigma(\tau, f)$  for such a channel is defined, apart from a normalization

factor, as the time average received power in a frequency interval at frequency  $f_c + f$  due to a range interval at delay  $\tau$ . The scattering function is generally normalized so that

$$\iint_{\tau,f} \sigma(\tau,f) d\tau df = 1$$

There is a tacit assumption here that  $\sigma(\tau,f)$  is independent of  $f_c$ , but this assumption is usually good for a wide range of  $f_c$ .

If there are a large number of scatters, all in more or less random motion with respect to each other, and if multiple scattering from one particle to another can be ignored, then we can interpret the received waveform caused by a fixed input as a sum of a very large number of more or less independent incremental waveforms. It is then reasonable to assume that, for any fixed input, the output is a Gaussian random process. Under this assumption, if  $A \cos 2\pi f_c t$  is transmitted, the received waveform (in the absence of additive noise) can be represented as

$$r(t) = v_1(t) \cos 2\pi f_c t + v_2(t) \sin 2\pi f_c t \quad (8.6.1)$$

where  $v_1(t)$  and  $v_2(t)$  are Gaussian random processes. If we further assume that the statistics of the medium are stationary and that the phase of the return from each scattering particle is uniformly distributed between 0 and  $2\pi$ , then it can be shown that  $v_1(t)$  and  $v_2(t)$  have the same spectral density. Furthermore if the scattering function satisfies  $\sigma(\tau,f) = \sigma(\tau,-f)$ , then  $v_1(t)$  and  $v_2(t)$  will be statistically independent. Finally (assuming  $f_c$  much greater than any of the doppler shifts),  $v_1(t)$  and  $v_2(t)$  can be observed separately at the receiver.

Let  $S\sigma(f)$  be the spectral density of  $v_1(t)$  [and of  $v_2(t)$ ] when  $\sigma(f)$  is normalized,  $\int \sigma(f) dt = 1$ . It can be seen that  $S$  is the average power in the received waveform  $r(t)$ , and that  $\sigma(f) = \int \sigma(\tau,f) d\tau$  where  $\sigma(\tau,f)$  is the scattering function defined previously. The average power  $S$  is a function of the channel but is also, of course, directly proportional to the transmitter power  $A^2/2$ .

We now want to investigate the error probability that can be achieved by coding for the above type of channel. To make the analysis as simple as possible, we consider the case where the code words are a set of sinusoids over a fixed interval  $(-T/2, T/2)$  separated in frequency; that is,

$$x_m(t) = \begin{cases} A \cos 2\pi(f_c + \Delta m)t; & -T/2 \leq t \leq T/2 \\ 0; & |t| > T/2 \end{cases} \quad (8.6.2)$$

We assume that  $\Delta$  is chosen large enough so that  $\sigma(f) = 0$  for  $|f| \geq \Delta/2$ . We can then imagine a bank of unit-gain filters at the receiver, each of bandwidth  $\Delta$  with the  $m$ th filter centered at frequency  $f_c + \Delta m$ . If we consider the received waveform now to be the sum of the received signal plus white

Gaussian noise of spectral density  $N_o/2$ , we can express the output from each filter, where message  $m$  is sent, as

$$\begin{aligned} y_m(t) &= [v_{1,m}(t) + n_{1,m}(t)] \cos 2\pi(f_c + \Delta m)t \\ &\quad + [v_{2,m}(t) + n_{2,m}(t)] \sin 2\pi(f_c + \Delta m)t \end{aligned} \quad (8.6.3)$$

$$y_{m'}(t) = n_{1,m'}(t) \cos 2\pi(f_c + \Delta m')t + n_{2,m'}(t) \sin 2\pi(f_c + \Delta m')t \quad \text{all } m' \neq m \quad (8.6.4)$$

In these expressions,  $v_{1,m}(t)$ ,  $n_{1,m}(t)$ ,  $n_{1,m'}(t)$ , and  $n_{2,m'}(t)$  are independent zero mean stationary Gaussian random processes, each with a spectral density of  $N_o$  for  $|f| \leq \Delta/2$  and 0 for  $|f| \geq \Delta/2$ . Since we are primarily interested in finding an upper bound to the error probability for the system, and since it turns out that the noise is unimportant for  $|f| > \Delta/2$  anyway, we can simplify our model by assuming that all the above noise processes are white with spectral density  $N_o$ .

The processes  $v_{1,m}(t)$  and  $v_{2,m}(t)$  in (8.6.3) are not stationary since the input  $s_m(t)$  is time limited to  $(-T/2, T/2)$ . On the other hand, if we define  $L$  to be the spread in delay introduced both by the medium and the receiving filter, and if we shift the time axis appropriately at the receiver, then we can assert that the output over the interval  $[-(T-L)/2, (T-L)/2]$  is the same as if the input were not truncated to  $(-T, T)$ . In other words, over the interval  $[-(T-L)/2, (T-L)/2]$ , we can take  $v_{1,m}(t)$  and  $v_{2,m}(t)$  to be independent sample functions of a stationary Gaussian random process of spectral density  $S\sigma(f)$ .

Up to this point, our development has been heuristic and approximate, which is appropriate since we have been dealing with a class of imprecisely specified physical channels. Now however, we have a mathematical model that we can work with and the development from this point on will be precise. To review the model, we have a set of  $M$  code words, each of duration  $T$ , given by (8.6.2). The receiver observes the waveforms  $y_{1,m}(t)$  and  $y_{2,m}(t)$  for  $1 \leq m \leq M$ , where, if message  $m$  is transmitted,

$$y_{i,m}(t) = v_{i,m}(t) + n_{i,m}(t); \quad i = 1, 2 \quad (8.6.5)$$

and for all  $m' \neq m$ ,

$$y_{i,m'}(t) = n_{i,m'}(t); \quad i = 1, 2 \quad (8.6.6)$$

We assume that all the waveforms  $y_{i,m}(t)$ , for  $i = 1, 2$  and  $1 \leq m \leq M$ , are observed only in the interval  $[-(T-L/2), (T-L/2)]$ . In this interval, all the waveforms  $v_{i,m}(t)$ ,  $n_{i,m}(t)$ , and  $n_{i,m'}(t)$  for  $m' \neq m$  are sample functions of independent, stationary, zero mean, Gaussian random processes;  $v_{i,m}(t)$  (for  $i = 1, 2$ ) having a spectral density  $S\sigma(f)$ , and  $n_{i,m}$  and  $n_{i,m'}$ , (for all  $m' \neq m$  and  $i = 1, 2$ ) having a spectral density  $N_o$ . Finally, we assume that the model is valid for all choices of  $M$  and  $T$ .

Let  $\mathcal{R}(\tau) = \int S\sigma(f)e^{j2\pi f\tau} df$  be the autocorrelation function of the untruncated process  $v_{i,m}(t)$ , and let  $\{\phi_j(\tau)\}$  and  $\{\lambda_j\}$  be the eigenfunctions and eigenvalues of  $\mathcal{R}(\tau)$  in the interval  $(-T_1/2, T_1/2)$  where  $T_1 = T - L$ ,

$$\int_{-T_1/2}^{T_1/2} \mathcal{R}(\tau_2 - \tau_1) \phi_j(\tau_2) d\tau_2 = \lambda_j \phi_j(\tau_1); \quad |\tau_1| < T_1/2 \quad (8.6.7)$$

Define the random variables  $v_{i,m,j}$ ,  $n_{i,m,j}$ , and  $y_{i,m',j}$  by

$$v_{i,m,j} = \int_{-T_1/2}^{T_1/2} v_{i,m}(t) \phi_j(t) dt \quad (8.6.8)$$

$$n_{i,m,j} = \int_{-T_1/2}^{T_1/2} n_{i,m}(t) \phi_j(t) dt \quad (8.6.9)$$

$$y_{i,m',j} = \int_{-T_1/2}^{T_1/2} y_{i,m}(t) \phi_j(t) dt = \begin{cases} v_{i,m',j} + n_{i,m',j}; & m' = m \\ n_{i,m',j}; & m' \neq m \end{cases} \quad (8.6.10)$$

By our assumptions,  $y_{i,m',j}$  can be calculated at the receiver for all  $i, m', j$ . It is a sample value of a zero mean Gaussian random variable with variance given by

$$\overline{y_{i,m',j}^2} = \begin{cases} \lambda_j + N_o; & m' = m \\ N_o; & m' \neq m \end{cases} \quad (8.6.11)$$

where  $m$  is the transmitted message.

Let  $\mathbf{y}_{m'}$  be the sequence of output variables  $(y_{1,m',1}, \dots, y_{1,m',J}, y_{2,m',1}, \dots, y_{2,m',J})$ . For the moment  $J$  is arbitrary but fixed and, later, we shall consider the limit as  $J \rightarrow \infty$ . For  $m' \neq m$ , the joint probability density of  $\mathbf{y}_{m'}$  is given by

$$p_o(\mathbf{y}_{m'}) = \prod_{i=1}^2 \prod_{j=1}^J \frac{1}{\sqrt{2\pi N_o}} \exp\left(-\frac{y_{i,m',j}^2}{2N_o}\right) \quad (8.6.12)$$

For  $m' = m$ , the transmitted message, the joint probability density of  $\mathbf{y}_{m'}$  is given by

$$p_1(\mathbf{y}_{m'}) = \prod_{i=1}^2 \prod_{j=1}^J \frac{1}{\sqrt{2\pi(N_o + \lambda_j)}} \exp\left[-\frac{y_{i,m',j}^2}{2(N_o + \lambda_j)}\right] \quad (8.6.13)$$

Given that message  $m$  is transmitted, the joint probability density of the entire set of received variables  $y_{i,m',j}$  for  $1 \leq j \leq J$  is thus given by

$$p(\mathbf{y}_1, \dots, \mathbf{y}_M | \mathbf{x}_m) = p_1(\mathbf{y}_m) \prod_{m' \neq m} p_o(\mathbf{y}_{m'}) \quad (8.6.14)$$

$$= \frac{p_1(\mathbf{y}_m)}{p_o(\mathbf{y}_m)} \prod_{m'=1}^M p_o(\mathbf{y}_{m'}) \quad (8.6.15)$$

A maximum-likelihood decoder operating on this set of variables (that is,  $1 \leq j \leq J$ ) will decode that  $m$  that maximizes  $p(\mathbf{y}_1, \dots, \mathbf{y}_m | \mathbf{x}_m)$  or, equivalently, that  $m$  that maximizes  $p_1(\mathbf{y}_m)/p_o(\mathbf{y}_m)$ . We can upper bound the probability of error for such a maximum-likelihood decoder by the same set of steps used in proving Theorem 5.6.1. In particular,

$$P_{e,m} = \int_{\mathbf{y}_m} p_1(\mathbf{y}_m) \Pr(\text{error} | m, \mathbf{y}_m) d\mathbf{y}_m \quad (8.6.16)$$

where  $\Pr(\text{error} | m, \mathbf{y}_m)$  is the probability of error given that message  $m$  is transmitted and given a particular sequence  $\mathbf{y}_m$ . For a given  $\mathbf{y}_m$ , let  $A_{m'}$  be the event that

$$\frac{p_1(\mathbf{y}_{m'})}{p_o(\mathbf{y}_{m'})} \geq \frac{p_1(\mathbf{y}_m)}{p_o(\mathbf{y}_m)} \quad (8.6.17)$$

An error occurs if and only if the event  $A_{m'}$  occurs for some  $m'$ , and thus

$$\begin{aligned} \Pr(\text{error} | m, \mathbf{y}_m) &= P\left(\bigcup_{m' \neq m} A_{m'}\right) \\ &\leq \left[ \sum_{m' \neq m} P(A_{m'}) \right]^\rho; \quad \text{any } \rho, \quad 0 \leq \rho \leq 1 \end{aligned} \quad (8.6.18)$$

where we have used (5.6.2) and the probabilities on the right are all conditional on  $m$  and  $\mathbf{y}_m$ . Using the same bounding technique as in (5.6.8), we have

$$P(A_{m'}) = \int_{\substack{\mathbf{y}_{m'}: \frac{p_1(\mathbf{y}_{m'})}{p_o(\mathbf{y}_{m'})} \geq \frac{p_1(\mathbf{y}_m)}{p_o(\mathbf{y}_m)}}} p_o(\mathbf{y}_{m'}) d\mathbf{y}_{m'} \leq \int p_o(\mathbf{y}_{m'}) \left[ \frac{p_1(\mathbf{y}_{m'}) p_o(\mathbf{y}_m)}{p_o(\mathbf{y}_{m'}) p_1(\mathbf{y}_m)} \right]^{1/(1+\rho)} d\mathbf{y}_{m'} \quad (8.6.19)$$

Substituting (8.6.19) into (8.6.18) and (8.6.18) into (8.6.16) and observing that  $\mathbf{y}_{m'}$  is at this point merely a dummy variable of integration, we have

$$\begin{aligned} P_{e,m} &\leq \int_{\mathbf{y}_m} p_1(\mathbf{y}_m) (M-1)^\rho \left\{ \int_{\mathbf{y}_{m'}} p_o(\mathbf{y}_{m'}) \left[ \frac{p_1(\mathbf{y}_{m'}) p_o(\mathbf{y}_m)}{p_o(\mathbf{y}_{m'}) p_1(\mathbf{y}_m)} \right]^{1/(1+\rho)} d\mathbf{y}_{m'} \right\}^\rho d\mathbf{y}_m \\ &= (M-1)^\rho \int_{\mathbf{y}_m} p_1(\mathbf{y}_m)^{1/(1+\rho)} p_o(\mathbf{y}_m)^{\rho/(1+\rho)} d\mathbf{y}_m \\ &\quad \times \left[ \int_{\mathbf{y}_{m'}} p_1(\mathbf{y}_{m'})^{1/(1+\rho)} p_o(\mathbf{y}_{m'})^{\rho/(1+\rho)} d\mathbf{y}_{m'} \right]^\rho \\ &= (M-1)^\rho \left[ \int_{\mathbf{y}_m} p_1(\mathbf{y}_m)^{1/(1+\rho)} p_o(\mathbf{y}_m)^{\rho/(1+\rho)} d\mathbf{y}_m \right]^{1+\rho} \end{aligned} \quad (8.6.20)$$

Substituting (8.6.12) and (8.6.13) into (8.6.20), we observe that the integral over  $\mathbf{y}_m$  breaks up into a product of integrals over the components  $y_{i,m,j}$ .

These are simple Gaussian integrals, dependent on  $j$  but not  $i$ . We obtain

$$\left[ \int_{\mathbf{y}_m} p_1(\mathbf{y}_m)^{1/(1+\rho)} p_o(\mathbf{y}_m)^{\rho/(1+\rho)} d\mathbf{y}_m \right]^{1+\rho} = \prod_{j=1}^J \frac{[1 + (\lambda_j/N_o)]^\rho}{\{1 + \rho\lambda_j/[(1 + \rho)N_o]\}^{1+\rho}} \quad (8.6.21)$$

Since this result is valid for every  $J$ , we can pass to the limit  $J \rightarrow \infty$  and rewrite our result as

$$P_{e,m} \leq (M - 1)^\rho \exp [-TE_o(\rho, T)] \quad (8.6.22)$$

where

$$E_o(\rho, T) = \frac{1}{T} \sum_{j=1}^{\infty} \left[ (1 + \rho) \ln \left( 1 + \frac{\rho\lambda_j}{(1 + \rho)N_o} \right) - \rho \ln \left( 1 + \frac{\lambda_j}{N_o} \right) \right] \quad (8.6.23)$$

As usual with problems involving eigenvalues, we can simplify the results by letting the time interval become large. To make the dependence of the eigenvalues on the receiver time interval  $T_1 = T - L$  explicit, we now use  $\lambda_j(T_1)$  in place of  $\lambda_j$ . By taking the derivative of  $E_o(\rho, T)$  with respect to  $\lambda_j(T_1)$ , we see that each term is increasing in  $\lambda_j$  with a bounded slope. Thus if  $S\sigma(f)$  is bounded and integrable, we can apply Lemma 8.5.3, obtaining\*

$$\begin{aligned} & \lim_{T_1 \rightarrow \infty} \frac{1}{T_1} \sum_{j=1}^{\infty} (1 + \rho) \ln \left[ 1 + \frac{\rho\lambda_j(T_1)}{(1 + \rho)N_o} \right] - \frac{1}{T_1} \sum_{j=1}^{\infty} \rho \ln \left[ 1 + \frac{\lambda_j(T_1)}{N_o} \right] \\ &= \int_{-\infty}^{\infty} (1 + \rho) \ln \left[ 1 + \frac{\rho S\sigma(f)}{(1 + \rho)N_o} \right] - \rho \ln \left[ 1 + \frac{S\sigma(f)}{N_o} \right] df \end{aligned} \quad (8.6.24)$$

Since  $L$  is fixed,  $T_1/T$  approaches 1 as  $T_1 \rightarrow \infty$ , and we have

$$\begin{aligned} E_o(\rho) &= \lim_{T \rightarrow \infty} E_o(\rho, T) \\ &= \int (1 + \rho) \ln \left[ 1 + \frac{\rho S\sigma(f)}{(1 + \rho)N_o} \right] - \rho \ln \left[ 1 + \frac{S\sigma(f)}{N_o} \right] df \end{aligned} \quad (8.6.25)$$

It follows that, for any  $\rho$ ,  $0 \leq \rho \leq 1$ , and any  $\epsilon > 0$ , we can pick  $T$  large enough so that, for any  $M$ , and all  $m$ ,  $1 \leq m \leq M$ ,

$$P_{e,m} \leq (M - 1)^\rho \exp \{-T[E_o(\rho) - \epsilon]\} \quad (8.6.26)$$

Before interpreting this result, it is helpful to introduce one more degree of freedom into the situation. The quantity  $S$  is the total signal power available at the receiver, and since  $S$  is proportional to the transmitter power, we can interpret it as transmitter power normalized to the receiver. If the transmitter has no peak power constraint on it but has an average power constraint of  $S$  (as normalized at the receiver), then one option open to the transmitter is to

\* It can be seen from the proof of Lemma 8.5.3 that the convergence of (8.6.24) is uniform in  $\rho$  for  $0 \leq \rho \leq 1$ , but we shall not need that result here.

spend only a fraction  $\theta$  of the time transmitting code words, using a power  $S/\theta$  during transmission. We call  $\theta$  the *duty factor* of the transmission. It is also possible to make  $\theta$  larger than 1 by overlapping the transmission of successive code words, using different frequency bands for successive words. Let  $T' = T/\theta$  be the time (in seconds) separating the beginning of successive code words. Then we can achieve an arbitrary rate  $R$  of transmission in natural units per second by using a number of code words  $M$  given by

$$M = [e^{T'R}] \quad (8.6.27)$$

Upper bounding (8.6.26) by  $M - 1 < \exp(T'R)$ , using  $S/\theta$  as the average received power during transmission, and using  $T'\theta$  for  $T$ , we find that for any  $\epsilon > 0$ , there is a  $T'$  large enough so that

$$P_{e,m} \leq \exp \{-T'[-\rho R + \tilde{E}_o(\rho, S, \theta) - \epsilon]\} \quad (8.6.28)$$

$$\tilde{E}_o(\rho, S, \theta) = \theta \int (1 + \rho) \ln \left[ 1 + \frac{\rho A(f)}{1 + \rho} \right] - \rho \ln [1 + A(f)] df \quad (8.6.29)$$

where

$$A(f) = \frac{S\sigma(f)}{\theta N_o} \quad (8.6.30)$$

In this situation, the required magnitude of  $T'$  for a given  $\epsilon > 0$  depends upon  $\theta$  and becomes unbounded as  $\theta \rightarrow 0$ .

The analysis of the exponent in (8.6.28) for fixed  $\theta$  and  $S$  is exactly the same as that of the exponent  $-\rho R + E_o(\rho, Q)$  for fixed  $Q$  in Section 5.6. In particular,  $E_o(0, S, \theta) = 0$  and

$$\frac{\partial E_o(\rho, S, \theta)}{\partial \rho} = \theta \int \left\{ \frac{A(f)}{1 + \rho + \rho A(f)} - \ln \left[ 1 + \frac{A(f)}{1 + \rho + \rho A(f)} \right] \right\} df > 0 \quad (8.6.31)$$

$$\frac{\partial^2 E_o(\rho, S, \theta)}{\partial \rho^2} = \theta \int \frac{-A^2(f)}{[1 + \rho + \rho A(f)]^2 (1 + \rho)} df < 0 \quad (8.6.32)$$

Defining  $E(R, S, \theta)$  as

$$\max_{0 \leq \rho \leq 1} [-\rho R + \tilde{E}_o(\rho, S, \theta)]$$

we have the usual parametric equations relating  $R$  and  $E(R, S, \theta)$ .

$$R = \theta \int \left\{ \frac{A(f)}{1 + \rho + \rho A(f)} - \ln \left[ 1 + \frac{A(f)}{1 + \rho + \rho A(f)} \right] \right\} df \quad (8.6.33)$$

$$E(R, S, \theta) = \theta \int \left\{ \ln \left[ 1 + \frac{\rho A(f)}{1 + \rho} \right] - \frac{\rho A(f)}{1 + \rho + \rho A(f)} \right\} df \quad (8.6.34)$$

These equations are valid for  $0 \leq \rho \leq 1$ , and for  $R$  less than the value given in (8.6.33) with  $\rho = 1$ , we have

$$E(R,S,\theta) = \theta \int \left\{ 2 \ln \left[ 1 + \frac{A(f)}{2} \right] - \ln [1 + A(f)] \right\} df - R \quad (8.6.35)$$

We define the capacity  $C(S,\theta)$  for a given duty factor and power, as the  $R$  given by (8.6.33) with  $\rho = 0$ ,

$$\begin{aligned} C(S,\theta) &= \theta \int \{ A(f) - \ln [1 + A(f)] \} df \\ &= \theta \int \left\{ \frac{S\sigma(f)}{\theta N_o} - \ln \left[ 1 + \frac{S\sigma(f)}{\theta N_o} \right] \right\} df \end{aligned} \quad (8.6.36)$$

By the same argument as in Section 5.6 [or by direct observation of (8.6.34) and (8.6.35)], we see that  $E(R,S,\theta) > 0$  for all  $R < C(S,\theta)$ .

Finally, we define the capacity  $C(S)$  of the channel for a given power  $S$ , as the supremum of  $C(S,\theta)$  over  $\theta > 0$ . It is easy to see that this supremum is given by

$$\begin{aligned} C(S) &= \lim_{\theta \rightarrow 0} C(S,\theta) = \int \frac{S\sigma(f)}{N_o} df \\ &= S/N_o \end{aligned} \quad (8.6.37)$$

where we have recalled that  $\sigma(f)$  was normalized to  $\int \sigma(f) df = 1$ .

This result is interesting and unexpected since it asserts that the capacity of this channel is the same as that of an infinite bandwidth additive Gaussian noise channel with power constraint  $S$  and noise spectral density  $N_o/2$ . For any  $R < C(S)$ , we can choose a  $\theta$  small enough that  $R < C(S,\theta)$ . For that duty factor, the error probability decays exponentially with  $T'$  for large enough  $T'$ .

In order to establish the converse result—that reliable communication is impossible if  $R > C(S)$ —we view the channel as a cascade of two channels, one the fading dispersive part and the second the part in which white noise is added.\* Since the average power at the input to the additive noise part is constrained to  $S$ , the average mutual information per second is, at most,  $S/N_o$ . From (2.3.19b), the average mutual information per second on the overall channel is at most  $S/N_o$ . Thus Theorem 8.5.2 applies to this channel also. It can be seen that the above argument does not depend on the details of the model that we have analyzed, but only on the white noise, the power limitation on the received signal, and the independence between the white noise and the rest of the channel and signal.

We can summarize our results in the following theorem.

\* Actually, with the model as given, we should regard the second channel as  $2M$  parallel white Gaussian noise channels each of spectral density  $N_o$  and with a total power constraint of  $2S$  [that is,  $S$  for  $v_{1,m}(t)$  and  $S$  for  $v_{2,m}(t)$ ]. The same conclusion as above follows.

**Theorem 8.6.1.** For the channel model defined in the paragraph surrounding (8.6.5) and (8.6.6), with the additional restriction that  $\sigma(f)$  be bounded and integrable, the channel capacity with a power constraint  $S$  is given by  $C(S) = S/N_o$ . For any  $R < C(S)$ , we can achieve arbitrarily small error probability by making the constraint time  $T'$  large enough and the duty factor  $\theta$  appropriately small. For  $R > C(S)$ , Theorem 8.5.2 applies. For any given duty factor  $\theta$ , for any  $\epsilon > 0$ , for all sufficiently large  $T'$ , and for any  $R < C(S, \theta)$ , the code of  $M = [e^{T' R}]$  code words has an error probability for each  $m$ ,  $1 \leq m \leq M$ , satisfying

$$P_{e,m} \leq \exp \{-T'[E(R,S,\theta) - \epsilon]\} \quad (8.6.38)$$

where  $E(R,S,\theta)$  is given by (8.6.33) to (8.6.35).

---

It should not be inferred from the above results that a code made up of frequency displaced sinusoids in any sense minimizes the probability of error that can be achieved for the given channel. By not restricting ourselves to sinusoids, we obtain a measure of control over the eigenvalues  $\{\lambda_j\}$  in (8.6.23). If we assume complete control over the  $\lambda_j$ , subject to

$$\sum_j \lambda_j = S,$$

then Kennedy (1964) has shown that there is an optimal value of  $\lambda_j$ , say  $\lambda_{opt}$ , depending only on  $\rho$ , and  $E_o(\rho, T)$  is maximized by choosing  $S/\lambda_{opt}$  of the  $\lambda_j$  to be  $\lambda_{opt}$  and choosing the rest to be zero.

## Summary and Conclusions

In this chapter, we have considered two specific classes of continuous time channels. The first was the class in which the transmitted signal is first filtered and then added to stationary Gaussian noise. The filter can be considered either as a frequency constraint on the input or as part of the channel. The second class was that in which the transmission medium is dispersive and time varying.

In the first section, we showed how to represent time functions and Gaussian random processes in terms of orthonormal expansions. Our treatment of Gaussian random processes was rather unconventional in that we defined the process in terms of linear operations on the process rather than in terms of the joint statistics of the process at all finite sets of points in time. Such an approach has the advantages of allowing white noise to be treated in a physically satisfying way and of avoiding all the subtle and intricate mathematical arguments to go from the point description to the linear operation description. In Section 8.2, we analyzed the error probability and optimal receiver for a set of orthogonal signals in white Gaussian noise. We also showed that the results could be translated directly into results for a simplex set of signals.

In Section 8.3 we gave an heuristic derivation of the capacity of a filtered channel with additive stationary Gaussian noise. In Sections 8.4 and 8.5 we followed this with a rigorous analysis of capacity and with upper bounds on the minimum achievable error probability. The analysis was a "one-shot" analysis however, omitting the effects of interference between successive code words. This latter problem is still an open research problem.

In Section 8.6, we first developed a mathematical model for the transmission of a code of frequency-displaced sinusoids on a fading dispersive channel with additive stationary white Gaussian noise. We then derived exponential bounds on error probability for the channel model using this class of codes and showed that the channel capacity is  $S/N_o$  where  $S$  is the constraint on received signal power and  $N_o/2$  is the noise spectral density.

### **Historical Notes and References**

There are many suitable references for orthonormal expansions and integral equations of the type considered here, among them Courant and Hilbert (1953) and Riesz and Sz. Nagy (1955). For Gaussian random processes, Wozencraft and Jacobs (1965) and Davenport and Root (1958) are excellent engineering references and Doob (1953) and Loève (1963) are excellent mathematical references. For an alternate approach to the detection of signals in nonwhite Gaussian noise that avoids orthonormal expansions, see Kailath (1967). The upper and lower bounds on error probability for orthogonal signals in white Gaussian noise are due to Fano (1961) and Zetterberg (1961), respectively. The capacity of channels with additive nonwhite Gaussian noise was given by Shannon (1948), with a rigorous derivation by Pinsker (1957). The capacity and a coding theorem for the filtered channels considered here was given by Holsinger (1964). Sections 8.4 and 8.5 follow Holsinger's approach quite closely except that proofs have been supplied here for a number of steps that were formal manipulations before. Wyner (1966) has analyzed a number of different mathematical models for the special case of a strictly bandlimited signal in white Gaussian noise and found coding theorems for each. His results give one added confidence in the insensitivity of the results to small changes in the model. Root and Varaiya (1968) have considered a generalization of the model here when the filter and noise are imperfectly known.

Kennedy (1969) has given a readable and much more complete treatment of reliable communication over fading dispersive channels than given here, deriving both upper and lower bounds on error probability for a larger class of communication systems. The results here are primarily due to Kennedy. The upper bound on error probability given by (8.6.22) and (8.6.23) was derived independently by Yudkin (1964) and Kennedy (1964), and Pierce (1961) earlier found an expression for  $P_{e,m}$  for  $M = 2$  for an equivalent

diversity problem where the  $\lambda_j$  were all equal. The result that  $C(S) = S/N_0$  was first discovered (without any accompanying coding theorem) by Jacobs (1963). Viterbi (1967) has also analyzed the case of frequency offset sinusoids analyzed here, giving upper and lower bounds on  $P_{e,m}$  that agree exponentially in the range where (8.6.33) and (8.6.34) apply. Richters (1967) has extended Kennedy's results to the situation where the channel input is band-limited. He shows that the bandwidth required for reliable communication increases rapidly as the rate approaches  $C(S)$ , but that at small rates and modest bandwidths, the exponent is close to the infinite bandwidth result. Such a result is clearly important, since with the frequency displaced sinusoids analyzed here the required bandwidth grows exponentially with  $T_1$ , a situation which quickly becomes physically unreasonable.

## *Chapter 9*

### SOURCE CODING WITH A FIDELITY CRITERION

#### 9.1 Introduction

In Chapter 3, we considered encoding the output of a source in such a way as to minimize the average number of code letters per source letter subject to the requirement that the source letters be retrievable from the encoded sequence. If the source output is either a sequence of continuous valued random variables or a random process, it is clearly impossible to encode the source output into a sequence of discrete code letters from which the source output can be exactly reconstructed. In such cases, we can only ask for the reconstruction to approximate the source within a given fidelity criterion. For example, if the source output produces a sequence of real numbers  $u_1 u_2, \dots$ , and this is reconstructed as a sequence  $v_1 v_2, \dots$ , we might demand that the expected value of  $(u_t - v_t)^2$ , averaged over the source and over time, be less than some prescribed value. More generally, we can define a distortion measure  $[(u_t - v_t)^2]$  in the above example] as an arbitrary real-valued function of source and reconstruction sequences. The fidelity criterion is then specified as a maximum tolerable value for the average distortion.

The major purpose of this chapter is to find the minimum number of binary digits required per source letter or per unit time to encode a source so that it can be reconstructed to satisfy a given fidelity criterion. Naturally, this limit will depend upon the source statistics, the distortion measure, and the fidelity criterion.

The purpose of representing the source output by a sequence of binary digits is to isolate the problem of source representation from that of information transmission. We already know from the coding theorem that, if the rate of this binary sequence (in bits per second) is less than the capacity (in bits per second) for the channel over which the sequence must be transmitted, then the sequence can be reproduced at the channel output with arbitrarily small error probability. Since the effect of these errors on the overall

distortion normally becomes negligible as the error probability approaches 0, we can indeed isolate the problem of channel coding from that of source coding. We shall see later that, in a sense, nothing is lost by this intermediate representation by binary digits. More precisely, we shall see that if a channel has a capacity too small to reliably transmit the minimum rate binary sequence required to represent the source with a given fidelity criterion, then that criterion cannot be met no matter what kind of processing is used between source and channel.

In principle, the theory developed here is applicable to the problems usually classified as quantization, analog to digital conversion, bandwidth compression, and data reduction. In practice, the theory has not had much impact yet. Part of the reason for this is the lack of an effective set of techniques for implementing source coding subject to a fidelity criterion. A more important reason is the difficulty of finding reasonable mathematical models for important sources. For example, a statistical characterization of speech waveforms is very difficult, and finding a meaningful measure of distortion seems even more difficult. Even in a problem of this complexity, however, the insights and relationships arising from this theoretical approach could be very valuable.

## 9.2 Discrete Memoryless Sources and Single-Letter Distortion Measures

In this section and the following three sections, we shall restrict our analysis of source coding with a fidelity criterion to the following case. We shall consider a discrete memoryless source  $U$  with the alphabet  $(0, 1, \dots, K-1)$  and the letter probabilities  $Q(0), \dots, Q(K-1)$ . We assume throughout that  $K$  is finite and that  $Q(k) > 0$  for each  $k$ ,  $0 \leq k \leq K-1$  [if  $Q(k)$  were 0, we could simply omit that letter from consideration]. The source output is a sequence  $u_1, u_2, \dots$  of independent selections from the given alphabet with the given letter probabilities. The source sequence is to be represented at the destination by a sequence of letters  $v_1, v_2, \dots$ , each selected from an alphabet  $(0, 1, \dots, J-1)$ ,  $J < \infty$ . Finally, there is a distortion measure  $d(k;j)$  defined for  $0 \leq k \leq K-1$ ,  $0 \leq j \leq J-1$ , assigning a numerical value to the distortion if source letter  $k$  is represented at the destination by letter  $j$ . We shall assume throughout that  $d(k;j) \geq 0$  and that for each  $k$ , there is at least one  $j$  for which  $d(k;j) = 0$ . It can be seen that there is really no loss of generality in this assumption, for if  $d'(k;j)$  is an arbitrary function of  $k, j$ , and we define

$$m(k) = \min_j d'(k;j),$$

then  $d(k;j) = d'(k;j) - m(k)$  will have the desired form. Since the difference between  $d(k;j)$  and  $d'(k;j)$  is a function only of the source letter  $k$ , it is independent of the processing done between source and destination. We shall

sometimes restrict our attention to *finite* distortion measures, where  $d(k;j)$  is finite for all  $k, j$ , but for the most part  $d(k;j)$  will be allowed to take on infinite values. An infinite distortion between a given  $k$  and  $j$  has the effect of absolutely forbidding that source letter  $k$  to be represented by destination letter  $j$ . The total distortion between a sequence  $u_1, \dots, u_L$  of source letters and a sequence  $v_1, \dots, v_L$  of destination letters is defined to be

$$\sum_{i=1}^L d(u_i;v_i).$$

As an example of a distortion measure, we might have  $J = K$  and  $d(k;j) = 0$  for  $j = k$  and  $d(k;j) = 1$  for  $j \neq k$ . Such a distortion measure would be appropriate if we were interested in reproducing the source letters exactly and counted all errors as being equally serious. As a second example, let  $J = K + 1$ , and let  $d(k;j) = 0$  for  $j = k$ , let  $d(k;j) = 1$  for  $j \neq k, j < J - 1$ , and let  $d(k;J-1) = \frac{1}{2}$  for all  $k$ . In this example, the output  $J - 1$  can be interpreted as an erasure symbol. Such a distortion measure is reasonable if all errors are equally serious, but an erasure is only half as serious as an error. As a third example, let  $J = K$ , and let  $d(k;j) = (j - k)^2$ , all  $j, k$ . Such a distortion measure is appropriate if the source letters are amplitudes and large errors in amplitude are more serious than small errors. As a final example, let  $J = 2$  and let  $d(k;j) = 0$  for  $K + j$  even and  $d(k;j) = 1$  for  $k + j$  odd. Such a distortion is appropriate if we are only concerned with whether the source letters are even or odd. This last example is rather artificial, but brings out the point that the distortion measure can be used to assign greater or less importance to the reconstruction of particular aspects of the source output.

When a source is connected to the destination via a channel and some processing, the statistics of the source, the statistics of the channel, and the operation of the processors together define a joint probability measure on the input sequence  $\mathbf{u}$  and the output sequence  $\mathbf{v}$ . The probability measure in turn defines an average distortion per source letter. We are interested in finding the minimum average distortion that can be achieved with a given channel. We shall find that this minimum depends only on the capacity of the channel and can be approached arbitrarily closely by a processor that first converts the source into a binary data stream with a rate arbitrarily close to the capacity of the given channel, and then encodes the binary data stream for transmission over the channel.

We begin with a rather arbitrary appearing, but fundamental, definition. The *rate-distortion function* of a given discrete memoryless source and distortion measure is defined as follows. We consider an arbitrary set of transition probabilities  $P(j|k)$ , where  $P(j|k)$  is interpreted as the probability

of destination letter  $j$  conditional on source letter  $k$ . These probabilities, together with the source probabilities, determine both an average mutual information

$$\mathcal{I}(\mathbf{Q}; \mathbf{P}) = \sum_{k=0}^{K-1} \sum_{j=0}^{J-1} Q(k) P(j | k) \ln \frac{P(j | k)}{\sum_i Q(i) P(j | i)} \quad (9.2.1)$$

and an average distortion

$$\bar{d} = \sum_k \sum_j Q(k) P(j | k) d(k; j) \quad (9.2.2)$$

The rate-distortion function of the source relative to the given distortion measure, is then defined as

$$R(d^*) = \min_{\mathbf{P}, \bar{d} \leq d^*} \mathcal{I}(\mathbf{Q}; \mathbf{P}) \quad (9.2.3)$$

The minimization in (9.2.3) is over all assignments of transition probabilities subject to the constraint that the average distortion is less than or equal to  $d^*$ . We shall soon see that, for a given  $d^*$ , if a channel of capacity less than  $R(d^*)$  nats per source symbol connects the source to the destination, then regardless of what processing is done, the average distortion must exceed  $d^*$ . Conversely, we shall see that the source can be encoded into  $R(d^*)/\ln 2$  binary digits per source letter and that the binary digits can be decoded into destination letters in such a way that the average distortion per letter does not exceed  $d^*$  by more than an arbitrarily small amount. Given these results, it is reasonable to interpret  $R(d^*)$  as the rate of the source, in nats per symbol, relative to the fidelity criterion  $d^*$ .

We shall postpone the question of how to calculate  $R(d^*)$  until after developing the above results and some general properties of this function. We first show that  $R(d^*)$  is nonnegative, nonincreasing in  $d^*$ , and convex  $\cup$  in  $d^*$ . The nonnegativity is obvious since  $\mathcal{I}(\mathbf{Q}; \mathbf{P}) \geq 0$ . Observe next that the minimization in (9.2.3) is over a constraint set which is enlarged as  $d^*$  is increased. Thus the resulting minimum  $R(d^*)$  is nonincreasing with  $d^*$ . To see that  $R(d^*)$  is convex  $\cup$ , let  $P_1(j | k)$  achieve the minimum in (9.2.3) for  $d_1^*$  and let  $P_2(j | k)$  achieve the minimum for  $d_2^*$ . Then since  $d_1^*$  and  $d_2^*$  are respectively greater than or equal to the distortions corresponding to  $\mathbf{P}_1$  and  $\mathbf{P}_2$ , we have for any  $\theta$ ,  $0 < \theta < 1$ ,

$$\theta d_1^* + (1 - \theta) d_2^* \geq \sum_{k,j} Q(k) [\theta P_1(j | k) + (1 - \theta) P_2(j | k)] d(k; j) \quad (9.2.4)$$

This implies that  $\theta \mathbf{P}_1 + (1 - \theta) \mathbf{P}_2$  is in the constraint set over which  $\mathcal{I}(\mathbf{Q}; \mathbf{P})$  is minimized to find  $R[\theta d_1^* + (1 - \theta) d_2^*]$ . Thus

$$R[\theta d_1^* + (1 - \theta) d_2^*] \leq \mathcal{I}[\mathbf{Q}; \theta \mathbf{P}_1 + (1 - \theta) \mathbf{P}_2] \quad (9.2.5)$$

Since Theorem 4.4.3 asserts that  $\mathcal{I}(\mathbf{Q}; \mathbf{P})$  is convex  $\cup$  in  $\mathbf{P}$ , we can further upper bound (9.2.5) by

$$\begin{aligned} R[\theta d_1^* + (1 - \theta) d_2^*] &\leq \theta \mathcal{I}(\mathbf{Q}; \mathbf{P}_1) + (1 - \theta) \mathcal{I}(\mathbf{Q}; \mathbf{P}_2) \\ &= \theta R(d_1^*) + (1 - \theta) R(d_2^*) \end{aligned} \quad (9.2.6)$$

which is the desired result.

Figure 9.2.1 sketches the form of  $R(d^*)$ . It can be seen that the smallest possible value for the average distortion is zero and is achieved by mapping each letter  $k$  of the source alphabet into a destination letter  $j$  for which  $d(k; j) = 0$ . For  $d^* < 0$ ,  $R(d^*)$  is undefined, since by definition  $d(k; j) \geq 0$ .

Next define  $d_{\max}^*$  as the smallest  $d^*$  for which  $R(d^*) = 0$  (see Figure 9.2.1). We can calculate  $d_{\max}^*$  from

$$d_{\max}^* = \min_j \sum_k Q(k) d(k; j) \quad (9.2.7)$$

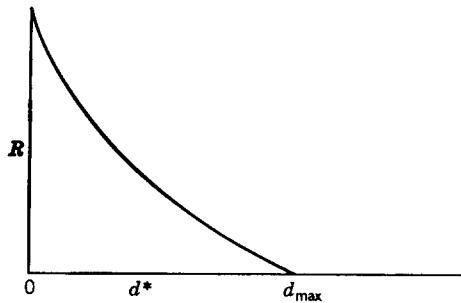


Figure 9.2.1. Sketch of typical function  $R(d^*)$ .

To see this, we observe that, if  $\mathcal{I}(\mathbf{Q}; \mathbf{P}) = 0$ , then the destination letter must be statistically independent of the source letter and, thus, the average distortion conditional on the selection of destination letter  $j$  is

$$\sum_k Q(k) d(k; j).$$

Thus  $\bar{d}$  is minimized subject to  $\mathcal{I}(\mathbf{Q}; \mathbf{P}) = 0$ , by always selecting the  $j$  specified by (9.2.7). If, for each  $j$ , there is some  $k$  for which  $d(k; j) = \infty$ , then clearly  $d_{\max}^* = \infty$ .

We now begin to develop the result that if the output from a source of given  $R(d^*)$  is transmitted over a channel of capacity  $C$  nats per source symbol then the average distortion per letter,  $\bar{d}$ , must satisfy  $C \geq R(\bar{d})$ .

**Theorem 9.2.1.** Let  $R(d^*)$  be the rate distortion function for a given discrete memoryless source and distortion measure. Let  $\mathbf{U}^L \mathbf{V}^L$  be a joint ensemble,  $\mathbf{U}^L$  the ensemble of sequences  $\mathbf{u} = (u_1, \dots, u_L)$  of length  $L$  from

the source and  $\mathbf{V}^L$  an ensemble of sequences  $\mathbf{v} = (v_1, \dots, v_L)$  from the destination alphabet. Let

$$\bar{d}_L = \overline{\frac{1}{L} \sum_{l=1}^L d(u_l; v_l)} \quad (9.2.8)$$

be the average distortion per letter for this joint ensemble. Then

$$\frac{1}{L} I(\mathbf{U}^L; \mathbf{V}^L) \geq R(\bar{d}_L) \quad (9.2.9)$$


---

*Proof.*

$$\frac{1}{L} I(\mathbf{U}^L; \mathbf{V}^L) = \frac{1}{L} [H(\mathbf{U}^L) - H(\mathbf{U}^L | \mathbf{V}^L)] \quad (9.2.10)$$

$$H(\mathbf{U}^L) = \sum_{l=1}^L H(U_l) \quad (9.2.11)$$

$$H(\mathbf{U}^L | \mathbf{V}^L) = \sum_{l=1}^L H(U_l | \mathbf{V}^L U_1 \dots U_{l-1}) \leq \sum_{l=1}^L H(U_l | V_l) \quad (9.2.12)$$

In (9.2.11) we have used the fact that the source is memoryless, and in (9.2.12) we have used (2.2.30) and (2.3.13). Substituting (9.2.11) and (9.2.12) into (9.2.10), we obtain

$$\frac{1}{L} I(\mathbf{U}^L; \mathbf{V}^L) \geq \frac{1}{L} \sum_{l=1}^L I(U_l; V_l) \quad (9.2.13)$$

Letting  $\bar{d}(l)$  be the average distortion on the  $l$ th letter, the definition of  $R(d^*)$  in (9.2.3) yields  $I(U_l; V_l) \geq R[\bar{d}(l)]$ . Using the convexity  $\cup$  of  $R(d^*)$ , we then have

$$\frac{1}{L} I(\mathbf{U}^L; \mathbf{V}^L) \geq \sum_{l=1}^L \frac{1}{L} R[\bar{d}(l)] \geq R\left[\sum_{l=1}^L \frac{1}{L} \bar{d}(l)\right] = R(\bar{d}_L) \quad |$$

As an interesting interpretation of the theorem, let  $P(j | k)$  be the transition probabilities that achieve  $R(d^*)$  for a given  $d^*$  in (9.2.3) and consider transmitting an  $L$  letter sequence from the source over a discrete memoryless channel with these transition probabilities. Then

$$\frac{1}{L} I(\mathbf{U}^L; \mathbf{V}^L) = R(d^*)$$

and the average distortion per letter between  $\mathbf{U}^L$  and  $\mathbf{V}^L$  is at most  $d^*$ . On the other hand, for any other  $\mathbf{U}^L \mathbf{V}^L$  with an average distortion per letter of at most  $d^*$ , the theorem implies that

$$\frac{1}{L} I(\mathbf{U}^L; \mathbf{V}^L) \geq R(d^*)$$

Thus  $R(d^*)$  could be equivalently defined, for any  $L \geq 1$ , as

$$R(d^*) = \min_{d_L \leq d^*} \frac{1}{L} I(\mathbf{U}^L; \mathbf{V}^L) \quad (9.2.14)$$

when the minimization is over  $P_L(\mathbf{v} | \mathbf{u})$  subject to  $d_L \leq d^*$ .

Suppose now that, for a given  $L$ , we choose a set of, say,  $M$  destination sequences,  $\mathbf{v}_1, \dots, \mathbf{v}_M$ , each of length  $L$ , and map the set of source sequences of length  $L$  into this set of destination sequences. We shall refer to any such set of sequences and such a mapping as an  $(L, M)$  source code, and refer to the set of sequences  $\mathbf{v}_1, \dots, \mathbf{v}_M$  as the code words. We can represent each code word in such an encoding by a sequence of  $\lceil \log_2 M \rceil$  binary digits, and if these binary digits are transmitted reliably over a noisy channel, then the distortion between the destination sequence (one of the code words) and the source sequence is simply the distortion introduced by this original encoding.

The source probabilities on sequences of  $L$  source letters and the encoding specified by a given  $(L, M)$  source code together define a joint ensemble  $\mathbf{U}^L \mathbf{V}^L$  of source sequences and destination sequences. More specifically, the probability of a source sequence  $\mathbf{u} = (u_1, \dots, u_L)$  is given by

$$Q_L(\mathbf{u}) = \prod_{i=1}^L Q(u_i)$$

and the conditional probability of a destination sequence  $\mathbf{v}$  given a source sequence  $\mathbf{u}$  is given by

$$P(\mathbf{v} | \mathbf{u}) = \begin{cases} 1; & \text{if } \mathbf{v} \text{ is the code word into which} \\ & \mathbf{u} \text{ is mapped} \\ 0; & \text{otherwise} \end{cases}$$

From Theorem 9.2.1, if  $d_L$  is the average distortion per letter between source sequences and destination sequences for this code, then

$$\frac{1}{L} I(\mathbf{U}^L; \mathbf{V}^L) \geq R(d_L)$$

On the other hand, since there are at most  $M$  destination sequences (that is, the code words) with nonzero probability,

$$\log M \geq H(\mathbf{V}^L) \geq I(\mathbf{U}^L; \mathbf{V}^L) \quad (9.2.15)$$

Combining these equations, we have the following simple corollary to Theorem 9.2.1.

**COROLLARY.** For an  $(L, M)$  source code to have average distortion per letter  $d_L$ , it is necessary for the entropy of the set of code words to satisfy

$$\frac{1}{L} H(\mathbf{V}^L) \geq R(d_L) \quad (9.2.16)$$

and for  $M$  to satisfy

$$\frac{\log M}{L} \geq R(\bar{d}_L) \quad (9.2.17)$$


---

We now go on to our main result, which in reality contains this corollary as a special case.

**Theorem 9.2.2.** Let a discrete memoryless source  $U$  and distortion measure  $d(k;j)$  have a rate distortion function  $R(d^*)$  and produce a letter each  $\tau_s$  seconds. Let a sequence of  $L$  source letters be connected to the destination by a sequence of  $N = \lfloor L\tau_s/\tau_c \rfloor$  uses of a discrete time channel, and let  $\bar{d}_L$  be the resulting distortion per source letter.

(a) If the channel is memoryless, with either a constrained or unconstrained input, and has a capacity  $C$  nats per channel use, then

$$R(\bar{d}_L) \leq \frac{\tau_s}{\tau_c} C \quad (9.2.18)$$

(b) If the channel is finite state, with upper capacity  $\bar{C}$  in nats per channel use, then in the limit as  $L \rightarrow \infty$ ,  $N = L\lfloor \tau_s/\tau_c \rfloor$ ,

$$R(\bar{d}_\infty) \leq \frac{\tau_s}{\tau_c} \bar{C} \quad (9.2.19)$$

(c) If the channel is finite state with  $A$  states and lower capacity  $\underline{C}$  in nats per channel use, and if the channel input ensemble  $\mathbf{X}^N$  is independent of the previous state  $s_0$ , then for at least one value of  $s_0$ ,

$$R(\bar{d}_L) \leq \frac{\tau_s}{\tau_c} \left[ \underline{C} + \frac{\log A}{N} \right]. \quad (9.2.20)$$


---

*Discussion.* For discrete-time memoryless channels and for indecomposable finite-state channels in the limit of large  $L$ , the theorem states that  $R(\bar{d}_L)$  is less than or equal to the capacity of the channel in nats per source symbol. Thus, if we mark the capacity (nats/source symbol) on the ordinate of the  $R(d^*)$  curve, then the corresponding abscissa is a lower bound to the average distortion per letter, no matter how the source and channel are processed. Alternatively, if we mark a desired fidelity criterion on the abscissa, the corresponding ordinate is a lower bound to the channel capacity required to achieve that fidelity criterion. This theorem is usually known as the converse to the coding theorem for sources relative to a distortion measure.

*Proof (Part a).* From (7.2.11), taking  $\mathbf{X}^N, \mathbf{Y}^N$  as the channel input and output ensembles respectively, we have

$$I(\mathbf{U}^L; \mathbf{V}^L) \leq I(\mathbf{X}^N; \mathbf{Y}^N) \quad (9.2.21)$$

From (7.2.12) for unconstrained channel inputs and (7.3.5) for constrained channel inputs, we have

$$I(\mathbf{X}^N; \mathbf{Y}^N) \leq NC \quad (9.2.22)$$

Combining (9.2.9), (9.2.21), and (9.2.22), we then have

$$R(\bar{d}_L) \leq \frac{N}{L} C \quad (9.2.23)$$

Since  $N \leq L\tau_s/\tau_c$ , we have (9.2.18).

*Part b.* For a given initial state  $s_0$  and given  $L$ , (9.2.9) asserts that

$$R(\bar{d}_L) \leq \frac{1}{L} I(\mathbf{U}^L; \mathbf{V}^L | s_0) \quad (9.2.24)$$

From (4.6.15) and (4.6.21), we have

$$I(\mathbf{U}^L; \mathbf{V}^L | s_0) \leq I(\mathbf{X}^N; \mathbf{Y}^N | s_0) \leq N\bar{C}_N \quad (9.2.25)$$

Thus, regardless of the initial state,

$$R(\bar{d}_L) \leq \frac{\tau_s}{\tau_c} \bar{C}_N \quad (9.2.26)$$

Passing to the limit as  $L \rightarrow \infty$ , we have  $\bar{C}_N \rightarrow \bar{C}$ , and (9.2.19) follows.

*Part c.* From (4.6.15) and (4.6.23) there is some initial state  $s_0$  for which

$$I(\mathbf{U}^L; \mathbf{V}^L | s_0) \leq I(\mathbf{X}^N; \mathbf{Y}^N | s_0) \leq N\underline{C}_N \quad (9.2.27)$$

From Theorem 4.6.1,

$$\underline{C}_N \leq C + \frac{\log A}{N} \quad (9.2.28)$$

Combining (9.2.24), (9.2.27), and (9.2.28), we have (9.2.20). |

The previous theorem is somewhat artificial in that it deals with only two classes of channels. The only difficulty in establishing an equivalent theorem for other channels lies in finding an appropriate definition for capacity. As we saw in Section 4.6, the maximum average mutual information per letter, even on a finite-state channel, can be radically dependent on the starting state and whether or not the starting state is known at the transmitter. For most interesting specific channels not in these classes, such as the additive Gaussian noise channel treated in the last chapter, it is relatively easy to define capacity as a maximum average mutual information per unit time in the limit of a large time interval. As soon as capacity is so defined, and the problem of past history somehow circumvented, it follows, as in the theorem that  $R(\bar{d}_\infty) \leq \tau_s C$ , where  $C$  is the channel capacity in nats per second.

### 9.3 The Coding Theorem for Sources with a Fidelity Criterion

We now turn to the problem of showing that  $R(d^*)/\ln 2$  is the number of binary digits per source digit required to achieve an average distortion arbitrarily close to  $d^*$ . For a given source  $U$  and destination alphabet  $V$ , a *source code* of  $M$  code words with block length  $L$  is defined as a mapping from the set of source sequences of length  $L$  into a set of  $M$  code words, where each code word,  $v_m = (v_{m1}, \dots, v_{mL})$  is a sequence of  $L$  letters from the destination alphabet. For example, Figure 9.3.1 illustrates a source code with two code words of block length 3 for binary source and destination alphabets.

For a given distortion measure  $d(k; j)$ , the average distortion per letter of a source code is given by

$$\bar{d}_L = \frac{1}{L} \sum_{\mathbf{u}} Q_L(\mathbf{u}) D[\mathbf{u}; v(\mathbf{u})] \quad (9.3.1)$$

when  $v(\mathbf{u})$  is the code word that  $\mathbf{u}$  is mapped into and

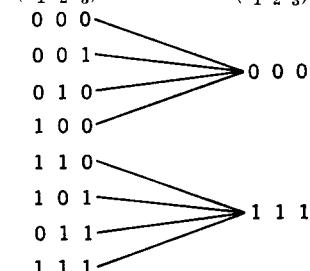
$$D(\mathbf{u}; \mathbf{v}) = \sum_{i=1}^L d(u_i; v_i) \quad (9.3.2)$$

For a source code with  $M$  code words, each code word used can be represented by a distinct binary sequence of length  $\lceil \log_2 M \rceil$ , or  $(1/L)\lceil \log_2 M \rceil$  binary digits per source digit. From the noisy-channel coding theorem, these sequences can be transmitted with arbitrary reliability over a channel whose capacity is greater than  $(1/L)\lceil \log_2 M \rceil$  bits per source symbol. Thus our major problem is to find how small  $\bar{d}_L$  can be made for a given  $L$  and  $M$ .

At this point, it should not be surprising that we shall attempt to solve this problem by analyzing the behavior of a randomly chosen set of code words. Let  $P(j | k)$  be a given set of transition probabilities between source and destination letters. For convenience, we shall refer to a discrete memoryless channel with these transition probabilities as a *test channel*, and if  $P(j | k)$  achieves  $R(d^*)$  for a given  $d^*$ , the associated test channel will be said to achieve  $R(d^*)$  for that  $d^*$ . The output probabilities,  $\omega(j)$  for a given test channel, are

$$\omega(j) = \sum_k Q(k)P(j | k).$$

Figure 9.3.1. Source code with two code words of block length 3 (lines indicate mapping from  $\mathbf{u}$  into  $\mathbf{v}$ ).



For any given test channel, we consider an ensemble of source codes in which each letter of each code word is chosen independently with the probability

assignment  $\omega(j)$ . For a given set of code words  $\mathbf{v}_1, \dots, \mathbf{v}_M$  in the ensemble, each source sequence  $\mathbf{u}$  will be mapped into that code word  $\mathbf{v}_m$  for which  $D(\mathbf{u}; \mathbf{v}_m)$  is minimized over  $m$ .

In the following argument, we shall have to consider simultaneously two different probability measures on the input and output sequences, one the test-channel ensemble and the other the random-coding ensemble. For the test-channel ensemble, the probability measure on input sequences  $\mathbf{u} = (u_1, \dots, u_L)$  and output sequences  $\mathbf{v} = (v_1, \dots, v_L)$  is given by  $Q_L(\mathbf{u})P_L(\mathbf{v} | \mathbf{u})$  where

$$Q_L(\mathbf{u}) = \prod_{i=1}^L Q(u_i) \quad (9.3.3)$$

$$P_L(\mathbf{v} | \mathbf{u}) = \prod_{i=1}^L P(v_i | u_i) \quad (9.3.4)$$

In these expressions  $Q(u_i)$  and  $P(v_i | u_i)$  are the source and test-channel probabilities respectively. The mutual information is

$$I(\mathbf{u}; \mathbf{v}) = \ln \frac{P_L(\mathbf{v} | \mathbf{u})}{\omega_L(\mathbf{v})} = \sum_{i=1}^L I(u_i; v_i) \quad (9.3.5)$$

where

$$\omega_L(\mathbf{v}) = \sum_{\mathbf{u}} Q_L(\mathbf{u})P_L(\mathbf{v} | \mathbf{u}) = \prod_{i=1}^L \omega(v_i) \quad (9.3.6)$$

The other ensemble is the ensemble of codes in which  $M$  code words are independently chosen with the probability assignment  $\omega_L(\mathbf{v})$ , the source sequence is chosen with the same assignment  $Q_L(\mathbf{u})$  as above, and for each code in the ensemble,  $\mathbf{u}$  is mapped into that  $\mathbf{v}_m$ , denoted  $\mathbf{v}(\mathbf{u})$ , that minimizes  $D(\mathbf{u}; \mathbf{v}_m)$  over  $1 \leq m \leq M$ .

**LEMMA 9.3.1.** For a given source, distortion measure, and test channel, let  $P_c[D > L \hat{d}]$  be the probability, over the above ensemble of codes with  $M$  code words of block length  $L$ , that  $D[\mathbf{u}; \mathbf{v}(\mathbf{u})]$  (that is, the distortion between  $\mathbf{u}$  and the code word into which it is mapped) exceeds a given number,  $L\hat{d}$ . Then

$$P_c[D > L \hat{d}] \leq P_t(A) + \exp[-Me^{-L\hat{R}}] \quad (9.3.7)$$

where  $\hat{R}$  is an arbitrary positive number,  $A$  is the set of  $\mathbf{u}, \mathbf{v}$  sequences given by

$$A = \{\mathbf{u}, \mathbf{v}: \text{either } I(\mathbf{u}; \mathbf{v}) > L\hat{R} \text{ or } D(\mathbf{u}; \mathbf{v}) > L\hat{d}\} \quad (9.3.8)$$

and  $P_t(A)$  is the probability of  $A$  in the test-channel ensemble.

---

Before proving this lemma, we shall use it in proving the following source coding theorem.

**Theorem 9.3.1.** Let  $R(d^*)$  be the rate distortion function of a discrete memoryless source with a *finite* distortion measure. For any  $d^* \geq 0$ , any  $\delta > 0$ , and any sufficiently large block length  $L$ , there exists a source code with  $M \leq \exp\{L[R(d^*) + \delta]\}$  code words for which the average distortion per letter satisfies

$$\bar{d}_L \leq d^* + \delta \quad (9.3.9)$$


---

*Proof.* We apply the lemma to the test channel that achieves  $R(d^*)$  for the given  $d^*$ , choosing  $\bar{d}$  as  $d^* + \delta/2$  and  $\bar{R}$  as  $R(d^*) + \delta/2$ . The average distortion per letter,  $\bar{d}_L$ , over the ensemble of codes in the lemma satisfies

$$\bar{d}_L \leq d^* + \delta/2 + P_c[D > L(d^* + \delta/2)] \max_{k,j} d(k;j) \quad (9.3.10)$$

This arises from upper bounding the distortion per letter by  $d^* + \delta/2$  when  $D[\mathbf{u}, \mathbf{v}(\mathbf{u})] \leq L(d^* + \delta/2)$  and by

$$\max_{k,j} d(k;j)$$

otherwise. For  $M = \lfloor \exp[LR(d^*) + L\delta] \rfloor$ , we have

$$\begin{aligned} Me^{-L\bar{R}} &\geq \{\exp[LR(d^*) + L\delta] - 1\} \exp\{-L[R(d^*) + \delta/2]\} \\ &\geq e^{L\delta/2} - 1 \end{aligned}$$

Thus (9.3.7) becomes

$$P_c[D > L(d^* + \delta/2)] \leq P_t(A) + \exp(-e^{L\delta/2} + 1) \quad (9.3.11)$$

Also, by the definition of  $A$ ,

$$P_t(A) \leq \Pr\{I(\mathbf{u}; \mathbf{v}) > L[R(d^*) + \delta/2]\} + \Pr[D(\mathbf{u}; \mathbf{v}) > L(d^* + \delta/2)] \quad (9.3.12)$$

where the probabilities on the right are over the test-channel ensemble. Since  $I(\mathbf{u}; \mathbf{v})$  and  $D(\mathbf{u}; \mathbf{v})$  are each the sum of  $L$  independent identically distributed random variables with means  $R(d^*)$  and  $d^*$  respectively, the Chebyshev inequality yields

$$P_t(A) \leq \frac{4\sigma_I^2}{L\delta^2} + \frac{4\sigma_d^2}{L\delta^2} \quad (9.3.13)$$

where  $\sigma_I^2$  is the variance of  $I(u;v)$  and  $\sigma_d^2$  is the variance of  $d(u;v)$  using the single-letter source and test-channel probabilities.

$$P_c[D > L(d^* + \delta/2)] \leq \frac{4\sigma_I^2}{L\delta^2} + \frac{4\sigma_d^2}{L\delta^2} + \exp(-e^{L\delta/2} + 1) \quad (9.3.14)$$

We see from this that the last term in (9.3.10) approaches 0 with increasing  $L$  for any  $\delta > 0$ . Thus, for large enough  $L$ ,

$$\bar{\bar{d}}_L \leq d^* + \delta \quad (9.3.15)$$

Since at least one code in the ensemble must have a distortion as small as the average, the theorem is proved. |

We notice that the code words of the given code can be represented by  $[L[R(d^*) + \delta]/\ln 2]$  binary digits, or at most  $[R(d^*) + \delta]/\ln 2 + 1/L$  binary digits per source digit. Thus by making  $L$  large enough, we can come as close as we please to an average distortion  $d^*$  with arbitrarily close to  $R(d^*)/\ln 2$  binary digits per source digit.

*Proof of Lemma.* We can express  $P_c(D > L \hat{d})$  as

$$P_c(D > L \hat{d}) = \sum_{\mathbf{u}} Q_L(\mathbf{u}) P_c(D > L \hat{d} \mid \mathbf{u}) \quad (9.3.16)$$

For a given  $\mathbf{u}$ , define  $A_{\mathbf{u}}$  as the set of  $\mathbf{v}$  for which  $\mathbf{u}, \mathbf{v}$  is in  $A$ ,

$$A_{\mathbf{u}} = \{\mathbf{v}: \text{either } I(\mathbf{u}; \mathbf{v}) > L \hat{R} \text{ or } D(\mathbf{u}; \mathbf{v}) > L \hat{d}\} \quad (9.3.17)$$

We observe that, for a given  $\mathbf{u}$ ,  $D[\mathbf{u}; \mathbf{v}(\mathbf{u})] > L \hat{d}$  only if  $D(\mathbf{u}; \mathbf{v}_m) > L \hat{d}$  for all  $m$ ,  $1 \leq m \leq M$ , and thus only if  $\mathbf{v}_m \in A_{\mathbf{u}}$  for all  $m$ . Since the  $\mathbf{v}_m$  are independently chosen,

$$P_c(D > L \hat{d} \mid \mathbf{u}) \leq \left[ 1 - \sum_{\mathbf{v} \in A_{\mathbf{u}}^c} \omega_L(\mathbf{v}) \right]^M \quad (9.3.18)$$

where the sum is over all  $\mathbf{v}$  in the complement of  $A_{\mathbf{u}}$ . For  $\mathbf{v} \in A_{\mathbf{u}}^c$ , we have

$$I(\mathbf{u}; \mathbf{v}) = \ln \frac{P_L(\mathbf{v} \mid \mathbf{u})}{\omega_L(\mathbf{v})} \leq L \hat{R} \quad (9.3.19)$$

$$\omega_L(\mathbf{v}) \geq P_L(\mathbf{v} \mid \mathbf{u}) e^{-L \hat{R}} \quad (9.3.20)$$

$$P_c(D > L \hat{d} \mid \mathbf{u}) \leq \left[ 1 - e^{-L \hat{R}} \sum_{\mathbf{v} \in A_{\mathbf{u}}^c} P_L(\mathbf{v} \mid \mathbf{u}) \right]^M \quad (9.3.21)$$

We now need the following inequalities.

$$[1 - \alpha x]^M = \exp [M \ln (1 - \alpha x)] \leq \exp (-M \alpha x) \quad (9.3.22)$$

$$\exp (-M \alpha x) \leq 1 - x + e^{-Mx}; \quad 0 \leq x \leq 1 \quad (9.3.23)$$

Equation 9.3.23 is clearly satisfied for  $x = 0$  and  $x = 1$ , and since the left-hand side is convex  $\cup$  in  $x$  and the right-hand side is linear in  $x$ , it is satisfied for  $0 \leq x \leq 1$ . Applying these inequalities to (9.3.21), letting  $x$  be the summation over  $\mathbf{v}$ ,

$$P_c(D > L \hat{d} \mid \mathbf{u}) \leq 1 - \sum_{\mathbf{v} \in A_{\mathbf{u}}^c} P_L(\mathbf{v} \mid \mathbf{u}) + \exp (-M e^{-L \hat{R}}) \quad (9.3.24)$$

Substituting this into (9.3.16), we obtain

$$P_c(D > L \hat{d}) \leq \sum_{\mathbf{u}} Q_L(\mathbf{u}) \left[ \sum_{\mathbf{v} \in A_{\mathbf{u}}} P_L(\mathbf{v} \mid \mathbf{u}) + \exp(-M e^{-L \hat{R}}) \right] \quad (9.3.25)$$

$$= P_t(A) + \exp(-M e^{-L \hat{R}}) \quad | \quad (9.3.26)$$

It is not difficult to see that the source coding theorem just proved and various generalizations of it play the same role in rate distortion theory as the noisy-channel coding theorem plays for noisy channels. It is a result that, even in retrospect, is rather surprising.

Theorem 9.3.1 was restricted to finite distortion measures, and that restriction was used in (9.3.10) to bound the distortion for those improbable source sequences for which the distortion to each code word was greater than  $d^* + \delta/2$ . The following theorem applies to infinite distortion measures.

**Theorem 9.3.2.** Let  $R(d^*)$  be the rate distortion function of a discrete memoryless source with an arbitrary single-letter distortion measure. For any  $d^* \geq 0$ ,  $\delta > 0$ , and sufficiently large  $L$ , there exists a source code where the entropy of the set of code words,  $H(\mathbf{V}^L)$ , satisfies  $H(\mathbf{V}^L) \leq L[R(d^*) + \delta]$  and the average distortion per letter satisfies  $\bar{d}_L \leq d^* + \delta$ .

*Proof.* We apply the lemma to the test channel which achieves  $R(d^*)$  choosing  $\hat{d}$  as  $d^* + \delta$ ,  $\hat{R}$  as  $R(d^*) + \delta/3$ . Over an ensemble with  $M = [\exp [LR(d^*) + 2L\delta/3]]$  code words, we have, as in (9.3.14):

$$P_c[D > L(d^* + \delta)] \leq \frac{9\sigma_d^2}{L\delta^2} + \frac{\sigma_d^2}{L\delta^2} + \exp(-e^{L\delta/3} + 1) \quad (9.3.27)$$

We have  $\sigma_d^2 < \infty$  since the test channel assigns zero probability to all infinite distortion transitions. In any particular code of this ensemble, there is a set  $B$  of source sequences which cannot be encoded with a distortion of  $d^* + \delta$  or less per letter, and there must be at least one code for which  $P(B)$  is upper bounded by the right-hand side of (9.3.27). Take this code and add to it one code word,  $\mathbf{v}$ , for each  $\mathbf{u}$  in  $B$ , choosing  $\mathbf{v}$  so that  $D(\mathbf{u}; \mathbf{v}) = 0$ . Mapping each  $\mathbf{u}$  not in  $B$  into one of the original  $M$  words with distortion  $d^* + \delta$  per letter or less and mapping each  $\mathbf{u}$  in  $B$  into one of the new words with distortion 0, we have  $\bar{d}_L \leq d^* + \delta$ . The original  $M$  words of the code have a total probability  $1 - P(B)$ . The entropy of the set of code words is upper bounded by assuming all the original words to be equiprobable and all the new words to be equiprobable. Since there are at most  $K^L$  new words,

$$\begin{aligned} H(\mathbf{V}^L) &\leq [1 - P(B)] \ln M + P(B) \ln K^L + \mathcal{H}[P(B)] \\ &\leq L \left\{ R(d^*) + 2\delta/3 + P(B) \ln K + \frac{\mathcal{H}[P(B)]}{L} \right\} \end{aligned} \quad (9.3.28)$$

Since  $P(B)$  approaches 0 with increasing  $L$ , we can pick  $L$  large enough so that  $H(\mathbf{V}^L) \leq L[R(d^*) + \delta]$ , completing the proof. |

From the variable-length source coding theorem of Chapter 3, this set of code words can be encoded into a variable-length binary code with an average length at most  $[H(\mathbf{V}^L) + \ln 2]/(L \ln 2)$  binary digits per source digit.

In summary, we have seen that a source can be encoded into arbitrarily little more than  $R(d^*)/\ln 2$  binary digits per source digit in such a way that the reconstructed sequences at the destination will have arbitrarily little more than  $d^*$  average distortion per letter.

The major difference between infinite and finite distortion measures appears when we attempt to transmit the encoded source output over a noisy channel. For finite distortion measures, there is a maximum distortion that can occur when a channel decoding error is made, and thus the contribution of channel errors to average distortion goes to zero as the probability of channel error goes to zero. For infinite distortion measures, if there is *any* code word in use that has an infinite distortion to any source sequence, then for a channel with all nonzero transition probabilities, there is a nonzero probability of that distortion occurring, and thus infinite average distortion.

We now return to Theorem 9.3.1 to investigate how quickly the rate of a code (that is,  $(\ln M)/L$ ) can be made to approach  $R(d^*)$  with increasing  $L$ . For this purpose, it is convenient to use the central limit theorem rather than just the Chebyshev inequality. We then have

$$\Pr\{I(\mathbf{u};\mathbf{v}) \geq L[R(d^*) + \delta/2]\} \approx \Phi\left(-\frac{\sqrt{L}\delta}{2\sigma_I}\right) \quad (9.3.29)$$

$$\Pr[D(\mathbf{u};\mathbf{v}) \geq L(d^* + \delta/2)] \approx \Phi\left(-\frac{\sqrt{L}\delta}{2\sigma_d}\right) \quad (9.3.30)$$

where

$$\Phi(-x) = \int_x^\infty \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{y^2}{2}\right) dy \leq \exp\left(-\frac{x^2}{2}\right); \quad x \geq 0$$

Thus  $P_t(A)$  in (9.3.12) is approximately upper bounded by

$$P_t(A) \lesssim \exp\left(-\frac{L\delta^2}{8\sigma_I^2}\right) + \exp\left(-\frac{L\delta^2}{8\sigma_d^2}\right) \quad (9.3.31)$$

Substituting this into (9.3.11) and (9.3.10), we obtain

$$\begin{aligned} \bar{d}_L &\lesssim d^* + \delta/2 \\ &+ \left[ \exp\left(-\frac{L\delta^2}{8\sigma_I^2}\right) + \exp\left(-\frac{L\delta^2}{8\sigma_d^2}\right) + \exp(-e^{L\delta/2} + 1) \right] \max_{k,j} d(k;j) \end{aligned}$$

Choosing†  $\delta = 2(\sigma_I + \sigma_d)\sqrt{(\ln L)/L}$ , we find that  $\bar{\bar{d}}_L$  approaches  $d^*$  with increasing  $L$  as  $\sqrt{(\ln L)/L}$  and  $(\ln M)/L$  approaches  $R(d^*)$  in the same way.

This says that for any given point on the  $R(d^*)$  curve, the bound on rate and distortion for a source code of length  $L$  lies on a line of slope  $45^\circ$  through the given point and approaches it as  $\sqrt{(\ln L)/L}$ . Since the  $d^*$  under consideration can be varied with  $L$  to hold either  $\bar{d}_L > 0$  fixed or  $(\ln M)/L$  fixed, the  $\delta$  can be omitted from either the bound on  $M$  or the bound on  $\bar{d}_L$  in Theorem 9.3.1 for  $\bar{d}_L > 0$ . It is easy to see that the same rate of convergence applies to Theorem 9.3.2.

By using more elaborate techniques, Pilc (1967) has demonstrated the existence of codes of rate  $R(d^*)$  where the distortion approaches  $d^*$  with increasing block length as  $(\ln L)/L$ .

#### 9.4 Calculation of $R(d^*)$

We turn our attention now to finding the set of transition probabilities (that is, the test channel) that achieves  $R(d^*)$ . The usual approach to minimizing a function  $\mathcal{I}(\mathbf{Q}; \mathbf{P})$  subject to a constraint,  $\bar{d} \leq d^*$  is to use a Lagrange multiplier, say  $\rho$ , and to minimize

$$R_0(\rho, \mathbf{P}) = \mathcal{I} + \rho \bar{d} \triangleq \sum_{k,j} Q(k) P(j | k) \left[ \ln \frac{P(j | k)}{\omega(j)} + \rho d(k; j) \right] \quad (9.4.1)$$

over all choices of the set of transition probabilities  $\mathbf{P}$ , where

$$\omega(j) = \sum_k Q(k) P(j | k).$$

We shall deal with the other constraints,

$$P(j | k) \geq 0,$$

$$\sum_j P(j | k) = 1$$

later. By varying  $\rho \geq 0$ , we shall see that it is possible to find  $R(d^*)$  for all values of  $d^*$ . In this problem, the multiplier  $\rho$  has the geometric significance of being the magnitude of the slope of the  $R(d^*)$  curve at the point generated by that value of  $\rho$ .

To see this, consider plotting  $\mathcal{I}(\mathbf{Q}; \mathbf{P})$  and  $\bar{d}$  for any particular  $\mathbf{P}$  on a graph with ordinate  $\mathcal{I}$  and abscissa  $\bar{d}$ . A line of slope  $-\rho$  through this point will intercept the  $\mathcal{I}$  axis at  $\mathcal{I}(\mathbf{Q}; \mathbf{P}) + \rho \bar{d}$ . The  $\mathbf{P}$  that minimizes  $R_0(\rho, \mathbf{P})$  will minimize that  $\mathcal{I}$  axis intercept, and all points on the  $R(d^*)$  curve will lie on or above the line of slope  $-\rho$  through this intercept. Thus this line is a

† To verify that the error in the central limit theorem approximation approaches zero faster than  $\delta$  for this choice, see Feller (1966), Theorem 1, p. 517.

tangent to the  $R(d^*)$  curve. We shall see later that, for finite distortion measures, the slope of  $R(d^*)$  cannot change discontinuously except perhaps at  $d_{\max}^*$ .

In actually carrying out the minimization of  $R_0(\rho, \mathbf{P})$  over  $\mathbf{P}$ , it is reasonable to add in Lagrange multipliers for the constraints

$$\sum_j P(j | k) = 1$$

for each  $k$ . It is convenient to denote these multipliers by  $-Q(k) \ln \frac{f_k}{Q(k)}$ , which suggests minimizing

$$F(\rho, \mathbf{P}, \mathbf{f}) = \sum_{k,j} Q(k)P(j | k) \left[ \ln \frac{P(j | k)}{\sum_i Q(i)P(j | i)} + \rho d(k; j) - \ln \frac{f_k}{Q(k)} \right] \quad (9.4.2)$$

and then choosing  $\mathbf{f} = (f_0, \dots, f_{K-1})$  so that

$$\sum_j P(j | k) = 1,$$

for each  $k$ . Differentiating  $F$ , we obtain

$$\frac{\partial F(\rho, \mathbf{P}, \mathbf{f})}{\partial P(j | k)} = Q(k) \left[ \ln \frac{P(j | k)}{\omega(j)} + \rho d(k; j) - \ln \frac{f_k}{Q(k)} \right] \quad (9.4.3)$$

where

$$\omega(j) = \sum_k Q(k)P(j | k).$$

Thus the conditions for  $\mathbf{P}$  to yield a stationary point to  $F$  are, for all  $k, j$ ,

$$\ln \frac{P(j | k)}{\omega(j)} + \rho d(k; j) - \ln \frac{f_k}{Q(k)} = 0 \quad (9.4.4)$$

$$P(j | k) = \frac{\omega(j)f_k}{Q(k)} e^{-\rho d(k; j)}; \quad \text{all } k, j \quad (9.4.5)$$

If we multiply both sides of (9.4.5) by  $Q(k)$ , sum over  $k$ , and cancel out

$$\omega(j) = \sum_k Q(k)P(j | k),$$

we obtain

$$1 = \sum_k f_k e^{-\rho d(k; j)}; \quad \text{all } j \quad (9.4.6)$$

Also, summing (9.4.5) over  $j$  and using the constraint

$$\sum_j P(j | k) = 1,$$

we obtain

$$1 = \frac{f_k}{Q(k)} \sum_j \omega(j)e^{-\rho d(k; j)}; \quad \text{all } k \quad (9.4.7)$$

Equation 9.4.6 provides us with  $J$  linear equations in the variables  $f_k$ , and (9.4.7) then gives us  $K$  linear equations in  $\omega(j)$ . If  $J = K$ , we can usually solve these equations and then find  $P(j | k)$  from (9.4.5). Since  $\mathcal{J}(\mathbf{Q}; \mathbf{P})$  is convex  $\cup$  in  $\mathbf{P}$ ,  $F(\rho, \mathbf{P}, \mathbf{f})$  is also convex  $\cup$  and the solution is a minimum.

The difficulty with the above approach is that the resulting  $P(j | k)$  need not be nonnegative. The following theorem makes the above approach rigorous and includes the constraint  $P(j | k) \geq 0$ . It also provides necessary and sufficient conditions on the set of transition probabilities  $\mathbf{P}$  that minimize  $R_0(\rho, \mathbf{P})$ , and provides us with a convenient lower bound to  $R_0(\rho, \mathbf{P})$ , and thus to  $R(d^*)$ .

**Theorem 9.4.1.** For a given source of entropy  $H(U)$  and a given distortion measure, let

$$R_0(\rho, \mathbf{P}) = \sum_{k,j} Q(k) P(j | k) \left[ \ln \frac{P(j | k)}{\sum_i Q(i) P(j | i)} + \rho d(k; j) \right]$$

where the sum is interpreted, here and throughout, as being over only those  $k, j$  for which  $P(j | k) > 0$ . Then, for any  $\rho > 0$ ,

$$\min_{\mathbf{P}} R_0(\rho, \mathbf{P}) = H(U) + \max_{\mathbf{f}} \sum_k Q(k) \ln f_k \quad (9.4.8)$$

where the maximization is over all  $\mathbf{f} = (f_0, \dots, f_{K-1})$  with nonnegative components satisfying the constraints

$$\sum_k f_k e^{-\rho d(k; j)} \leq 1; \quad 0 \leq j \leq J - 1 \quad (9.4.9)$$

Necessary and sufficient conditions on  $\mathbf{f}$  to achieve the maximum in (9.4.8) are that a set of nonnegative numbers  $\omega(0), \dots, \omega(J - 1)$  exist satisfying (9.4.7), and that (9.4.9) is satisfied with equality for each  $j$  with  $\omega(j) > 0$ . In terms of this  $\mathbf{f}$  and  $\omega$ ,  $\mathbf{P}$  as given by (9.4.5) minimizes  $R_0(\rho, \mathbf{P})$ . —————

*Discussion.* One consequence of (9.4.8) is that, for any set of  $f_k \geq 0$  that satisfy (9.4.9),

$$\min_{\mathbf{P}} R_0(\rho, \mathbf{P}) \geq H(U) + \sum_k Q(k) \ln f_k \quad (9.4.10)$$

Since we have already seen that

$$\min_{\mathbf{P}} R_0(\rho, \mathbf{P}) - \rho d^* \leq R(d^*),$$

this means that, for any  $\rho \geq 0$  and any  $\mathbf{f}$  satisfying (9.4.9) for that  $\rho$ ,

$$R(d^*) \geq H(U) + \sum_k Q(k) \ln f_k - \rho d^* \quad (9.4.11)$$

This provides a lower bound to  $R(d^*)$ , and in fact gives  $R(d^*)$  if  $\rho$  and  $\mathbf{f}$  are optimally chosen.

In attempting to actually find

$$\min_{\mathbf{P}} R_0(\rho, \mathbf{P}),$$

the situation is very much like that of finding channel capacity. We can attempt to solve (9.4.9) for  $\mathbf{f}$ , first guessing the set of  $j$  for which equality should hold; this  $\mathbf{f}$  can then be used to solve (9.4.7) for  $\omega(j)$ , and  $P(j | k)$  can be found from (9.4.5). Summing (9.4.5) over  $j$  and using (9.4.7), we see that  $P(j | k)$  is indeed a transition probability assignment. Multiplying (9.4.5) by  $Q(k)$  summing over  $k$ , and using (9.4.9), we see that  $\omega(j)$  is indeed the output probability

$$\sum_k Q(k)P(j | k).$$

Although (9.4.8) gives  $\min R_0(\rho, \mathbf{P})$  solely in terms of  $\mathbf{f}$ , we must still solve (9.4.7) to ensure that  $\omega(j) \geq 0$  and to ensure that (9.4.9) is satisfied with equality for each  $j$  with  $\omega(j) > 0$ . In the following section, this procedure will be applied to an important example. Another approach to finding  $\min R_0(\rho, \mathbf{P})$ , if the source and distortion measure have enough symmetry, is to guess the solution and verify that it satisfies the conditions of the theorem. As a final approach, since

$$\sum_k Q(k) \ln f_k$$

is convex  $\cap$  in  $\mathbf{f}$ ,  $R_0(\rho, \mathbf{P})$  is convex  $\cup$  in  $\mathbf{P}$ , and the constraint sets are convex, it is relatively easy to perform either the maximization or the minimization by numerical techniques on a computer.

It should be observed from the strict convexity of  $\sum Q(k) \ln f_k$  in  $\mathbf{f}$  that the maximizing  $\mathbf{f}$  in (9.4.8) is unique. On the other hand, neither  $\mathbf{P}$  nor  $\omega$  are necessarily unique.

One easy way to interpret the maximizing  $\mathbf{f}$  in the theorem is to observe that for the  $\mathbf{P}$  and  $\mathbf{f}$  satisfying (9.4.5), the probability of an input given an output (sometimes called a backward transition probability) is

$$P_b(k | j) = f_k e^{-\rho d(k; j)}$$

Equation 9.4.9, with equality, is then the condition that these actually be conditional probabilities, and (9.4.7) is the condition that these backward probabilities be consistent with the input probabilities.

*Proof of Theorem.* First, we shall show that, for all  $\mathbf{P}$  and  $\mathbf{f}$  satisfying the given constraints,

$$R_0(\rho, \mathbf{P}) \geq H(U) + \sum_k Q(k) \ln f_k \quad (9.4.12)$$

Consider the function

$$F(\rho, \mathbf{P}, \mathbf{f}) = \sum_{k,j} Q(k)P(j | k) \left[ \ln \frac{P(j | k)}{\sum_i Q(i)P(j | i)} + \rho d(k; j) - \ln \frac{f_k}{Q(k)} \right] \quad (9.4.13)$$

Separating the third term in (9.4.13) from the first two, and summing it over  $j$ , we obtain

$$\begin{aligned} F(\rho, \mathbf{P}, \mathbf{f}) &= R_0(\rho, \mathbf{P}) - \sum_k Q(k) \ln \frac{f_k}{Q(k)} \\ &= R_0(\rho, \mathbf{P}) - H(U) - \sum_k Q(k) \ln f_k \end{aligned} \quad (9.4.14)$$

We now show that  $F(\rho, \mathbf{P}, \mathbf{f})$  is nonnegative by using the usual inequality  $\ln x \leq x - 1$ . From (9.4.13):

$$-F(\rho, \mathbf{P}, \mathbf{f}) = \sum_{k,j} Q(k) P(j | k) \ln \left[ \frac{\omega(j) f_k}{P(j | k) Q(k)} e^{-\rho d(k;j)} \right] \quad (9.4.15)$$

$$\leq \sum_{k,j} \omega(j) f_k e^{-\rho d(k;j)} - \sum_{k,j} Q(k) P(j | k) \quad (9.4.16)$$

Summing first over  $k$  and using the constraint (9.4.9):

$$-F(\rho, \mathbf{P}, \mathbf{f}) \leq \sum_j \omega(j) - \sum_j \omega(j) = 0 \quad (9.4.17)$$

Combining (9.4.17) with (9.4.14) yields (9.4.12).

Equation 9.4.12 is satisfied with equality iff both the inequalities going from (9.4.15) to (9.4.17) are satisfied with equality, or iff

$$\frac{\omega(j) f_k}{P(j | k) Q(k)} e^{-\rho d(k;j)} = 1 \quad \text{for all } P(j | k) > 0 \quad (9.4.18)$$

$$\sum_k f_k e^{-\rho d(k;j)} = 1 \quad \text{for all } \omega(j) > 0 \quad (9.4.19)$$

If we multiply both sides of (9.4.18) by  $P(j | k)$  and sum over  $j$ , we obtain (9.4.7), so that the conditions in the theorem are necessary for equality in (9.4.12). Likewise, if nonnegative numbers  $\omega(0), \dots, \omega(J-1)$  satisfy (9.4.7) and if (9.4.19) is satisfied, then we have already seen that  $P(j | k)$  as given by (9.4.5) is a transition probability assignment with output probabilities  $\omega(j)$ . By (9.4.5) the choice satisfies (9.4.18), so that the conditions of the theorem are sufficient for equality in (9.4.12).

To complete the proof, we must demonstrate the existence of a  $\mathbf{P}$  and  $\mathbf{f}$  satisfying the given constraints and satisfying (9.4.12) with equality. Let  $\mathbf{P}$  minimize\*  $R_0(\rho, \mathbf{P})$  over the allowed region where  $\mathbf{P}$  is a set of transition probabilities. For each  $k$ , pick some  $j$ , say  $j(k)$  for which  $P(j | k) > 0$  and define  $f_k$  by

$$\ln \frac{P[j(k) | k]}{\omega[j(k)]} + \rho d(k; j(k)) - \ln \frac{f_k}{Q(k)} = 0 \quad (9.4.20)$$

In the remainder of the argument, this  $\mathbf{f} = (f_0, \dots, f_{K-1})$  will be considered as fixed. Consider the function  $F(\rho, \mathbf{P}, \mathbf{f})$  in (9.4.13) and observe from (9.4.14)

\* There must be such a  $\mathbf{P}$  since the region over which the minimization is performed is closed and bounded and  $R_0$  is continuous in  $\mathbf{P}$  over this region.

that since the  $\mathbf{P}$  under consideration minimizes  $R_0(\rho, \mathbf{P})$ , it also minimizes  $F(\rho, \mathbf{P}, \mathbf{f})$ . For any  $k, j$  with  $\omega(j) > 0$  and  $d(k; j) < \infty$ , we have

$$\frac{\partial F(\rho, \mathbf{P}, \mathbf{f})}{\partial P(j \mid k)} = \left[ \ln \frac{P(j \mid k)}{\omega(j)} + \rho d(k; j) - \ln \frac{f_k}{Q(k)} \right] Q(k) \quad (9.4.21)$$

If this is negative (positive) for any  $k, j$  [with  $\omega(j) > 0, d(k; j) < \infty$ ], then an incremental increase (decrease) in that  $P(j \mid k)$  and an incremental decrease (increase) in  $P[j(k) \mid k]$  [for which  $\partial F / \partial P[j(k) \mid k] = 0$ ] will cause an incremental decrease in  $F(\rho, \mathbf{P}, \mathbf{f})$ , contradicting the assumption that  $\mathbf{P}$  minimizes  $F(\rho, \mathbf{P}, \mathbf{f})$ . It follows that, for the minimizing  $\mathbf{P}$ ,

$$\ln \frac{P(j \mid k)}{\omega(j)} + \rho d(k; j) - \ln \frac{f_k}{Q(k)} = 0 \quad (9.4.22)$$

for all  $k, j$  with  $\omega(j) > 0, d(k; j) < \infty$ . Also,  $P(j/k) = 0$  if either  $\omega(j) = 0$  or  $d(k; j) = \infty$ . Consequently, for the  $\mathbf{P}, \mathbf{f}$  under consideration, (9.4.5) is satisfied for all  $k, j$ . Multiplying (9.4.5) by  $Q(k)$  and summing over  $k$ , we see that (9.4.9) is satisfied with equality for  $j$  such that  $\omega(j) > 0$ . Thus  $\mathbf{P}$  and  $\mathbf{f}$  satisfy the sufficient conditions for equality in (9.4.12). We are still not finished since we must show that  $\mathbf{f}$  is in the appropriate constraint region, that is, that (9.4.9) is also satisfied for  $j$  such that  $\omega(j) = 0$ . Suppose that  $\omega(j') = 0$  for a given  $j'$  and define  $\mathbf{P}'$  in terms of the minimizing  $\mathbf{P}$  as follows:

$$P'(j' \mid k) = \epsilon \frac{f_k}{Q(k)} e^{-\rho d(k; j')} \quad \text{all } k \quad (9.4.23)$$

$$P'(j(k) \mid k) = P(j(k) \mid k) - P'(j' \mid k) \quad \text{all } k \quad (9.4.24)$$

$$P'(j \mid k) = P(j \mid k) \quad \text{all other } j, k \quad (9.4.25)$$

For small enough  $\epsilon > 0$ ,  $\mathbf{P}'$  is a set of transition probabilities. Let  $F_j(\rho, \mathbf{P}, \mathbf{f})$  be given by

$$F_j(\rho, \mathbf{P}, \mathbf{f}) = \sum_r Q(k)P(j \mid k) \left[ \ln \frac{P(j \mid k)}{\omega(j)} + \rho d(k; j) - \ln \frac{f_k}{Q(k)} \right] \quad (9.4.26)$$

Then

$$F(\rho, \mathbf{P}', \mathbf{f}) - F(\rho, \mathbf{P}, \mathbf{f}) = F_{j'}(\rho, \mathbf{P}', \mathbf{f}) + \sum_{j' \neq j} [F_j(\rho, \mathbf{P}', \mathbf{f}) - F_j(\rho, \mathbf{P}, \mathbf{f})] \quad (9.4.27)$$

Since  $\partial F_j(\rho, \mathbf{P}, \mathbf{f}) / \partial P(j(k) \mid k) = 0$ , we observe that the sum over  $j \neq j'$  in (9.4.27) has no first-order variation in  $\epsilon$ , so that to first order in  $\epsilon$ ,

$$\begin{aligned} F(\rho, \mathbf{P}', \mathbf{f}) - F(\rho, \mathbf{P}, \mathbf{f}) &= \sum_k Q(k)P'(j' \mid k) \ln \left[ \frac{\epsilon}{\sum_k Q(k)P'(j' \mid k)} \right] \\ &= \sum_k Q(k)P'(j' \mid k) \ln \frac{1}{\sum_k f_k e^{-\rho d(k; j')}} \end{aligned} \quad (9.4.28)$$

Since  $\mathbf{P}$  minimizes  $F(\rho, \mathbf{P}, \mathbf{f})$ , the right-hand side of (9.4.28) is nonnegative which implies

$$\sum_k f_k e^{-\rho d(k; j')} \leq 1,$$

completing the proof. |

**COROLLARY 9.4.1.** For a finite distortion measure with  $d_{\max}^* > 0$ , the slope of  $R(d^*)$  is continuous for  $0 < d^* < d_{\max}^*$  and approaches  $-\infty$  as  $d^*$  approaches 0. |

*Proof.* We have seen that  $R(d^*)$  is convex  $\cup$  and that

$$\min_{\mathbf{P}} R_0(\rho, \mathbf{P})$$

is the  $R$  axis intercept of a tangent of slope  $-\rho$  to  $R(d^*)$ . If the slope is discontinuous, or if it approaches a finite limit as  $d^* \rightarrow 0$ , then there must be some point on the  $R(d^*)$  curve which is the point of tangency for a range of different tangent slopes. Any  $\mathbf{P}$  that achieves that point on the  $R(d^*)$  curve must then minimize  $R_0(\rho, \mathbf{P})$  for that range of slopes. We complete the proof by showing that if  $\mathbf{P}$  minimizes  $R_0(\rho, \mathbf{P})$  for two different values of  $\rho$ , say  $\rho$  and  $\rho'$ , then  $\mathcal{I}(\mathbf{Q}; \mathbf{P}) = 0$ . Letting  $\mathbf{f}'$  be the maximizing  $\mathbf{f}$  for  $\rho'$ , we have

$$P(j \mid k) = \frac{\omega(j) f_k}{Q(k)} e^{-\rho d(k; j)} = \frac{\omega(j)}{Q(k)} f'_k e^{-\rho' d(k; j)} \quad (9.4.29)$$

Thus if  $\omega(j) > 0$ ,

$$\frac{f_k}{f'_k} = e^{(\rho - \rho') d(k; j)} \quad (9.4.30)$$

This implies that  $d(k; j)$  is independent of  $j$  for all  $j$  with  $\omega(j) > 0$ , and thus, from (9.4.29),  $Q(k)P(j \mid k) = \omega(j) \alpha(k)$  where  $\alpha(k) = f'_k \exp -\rho' d(k; j)$  is independent of  $j$ . Thus  $P(j \mid k) = \omega(j)$  and  $\mathcal{I}(\mathbf{Q}; \mathbf{P}) = 0$ . |

Problem 9.1 gives an example of an  $R(d^*)$  curve with a slope discontinuity at  $d_{\max}^*$ , and Problem 9.4 shows that the restriction to finite distortion measures in the corollary is necessary.

**COROLLARY 9.4.2.**  $R(d^*)$  is a continuous function of  $d^*$  for  $d^* \geq 0$ . |

*Proof.* Since  $R(d^*)$  is convex  $\cup$  and nonincreasing in  $d^*$  for  $d^* \geq 0$ , we know that it must be continuous except perhaps at  $d^* = 0$ . Also, since  $R(d^*) \leq H(U)$  for  $d^* \geq 0$  and since  $R(d^*)$  cannot decrease as  $d^*$  decreases, we know that

$$\lim_{d^* \rightarrow 0^+} R(d^*)$$

exists. Thus our only problem is to show that

$$R(0) = \lim_{d^* \rightarrow 0^+} R(d^*).$$

To do this, consider a new distortion measure  $d'(k;j)$  given by

$$d'(k;j) = \begin{cases} 0 & \text{if } d(k;j) = 0 \\ \infty & \text{otherwise} \end{cases} \quad (9.4.31)$$

The rate distortion function for  $d'(k;j)$  is simply

$$R'(d^*) = R(0); \quad d^* \geq 0$$

since the average distortion for any  $\mathbf{P}$  is either 0 or  $\infty$  and  $R(0)$  is the minimum  $\mathcal{I}(\mathbf{Q};\mathbf{P})$  for which the average distortion is 0. The function

$$\min_{\mathbf{P}} R'_0(\rho, \mathbf{P})$$

for  $d'(k;j)$  is thus also equal to  $R(0)$  for all  $\rho > 0$ , and using (9.4.8) for

$$\min_{\mathbf{P}} R'_0(\rho, \mathbf{P})$$

at any  $\rho > 0$ , we obtain

$$R(0) = H(U) + \max_{f'} \sum_k Q(k) \ln f'_k \quad (9.4.32)$$

subject to the constraint

$$\sum_{k:d(k;j)=0} f'_k \leq 1; \quad \text{all } j \quad (9.4.33)$$

By making  $\rho$  sufficiently large, we can approximate the  $\mathbf{f}'$  that maximizes (9.4.32) subject to (9.4.33) arbitrarily closely by an  $\mathbf{f}(\rho)$  that satisfies

$$\sum_k f_k(\rho) e^{-\rho d(k;j)} \leq 1; \quad \text{all } j \quad (9.4.34)$$

Thus, for any  $\epsilon > 0$ , we can choose  $\rho$  large enough so that

$$\begin{aligned} \min_{\mathbf{P}} R_0(\rho, \mathbf{P}) &\geq H(U) + \sum_k Q(k) \ln f_k(\rho) \\ &\geq R(0) - \epsilon \end{aligned} \quad (9.4.35)$$

Since

$$R(d^*) \geq \min_{\mathbf{P}} R_0(\rho, \mathbf{P}) - \rho d^*,$$

we have

$$\lim_{d^* \rightarrow 0} R(d^*) \geq R(0) - \epsilon; \quad \text{any } \epsilon > 0.$$

Since  $R(d^*) \leq R(0)$  for  $d^* \geq 0$ , this completes the proof. |

**COROLLARY 9.4.3.** At most  $K + 1$  letters of the destination alphabet need be used to achieve  $R(d^*)$  for any given  $d^*$ . On any part of the  $R(d^*)$  curve where the slope is nonconstant, at most  $K$  destination letters need be used.

*Proof.* For any given  $\rho$ , let  $\mathbf{f}$  maximize the right-hand side of (9.4.8) subject to (9.4.9). From the theorem, a  $\mathbf{P}$  that minimizes  $R_0(\rho, \mathbf{P})$  can be found from (9.4.5), where the output probabilities satisfy

$$\frac{f_k}{Q(k)} \sum_j \omega(j) e^{-\rho d(k;j)} = 1; \quad \text{all } k \quad (9.4.36)$$

This is a set of  $K$  linear equations in  $J$  unknowns, and there is at least one solution with  $\omega(j) \geq 0$ . Thus, by the same proof as in Corollary 3 in Section 4.5, there is a solution with only  $K$  of the  $\omega(j) > 0$ . If the line

$$\min_{\mathbf{P}} R_0(\rho, \mathbf{P}) - \rho d^*$$

is tangent to the curve  $R(d^*)$  in only one point, then this solution must yield that point. On the other hand, if  $\min_{\mathbf{P}} R_0(\rho, \mathbf{P}) - \rho d^*$  is tangent to  $R(d^*)$  over an interval, then to find a  $\mathbf{P}$  that yields a given  $d^*$ , we need the additional constraint

$$\sum_{k,j} Q(k) P(j | k) d(k;j) = d^*$$

Using (9.4.5) and (9.4.9) with equality for  $\omega(j) > 0$ , this becomes

$$\sum_j \omega(j) \sum_k f_k e^{-\rho d(k;j)} d(k;j) = d^* \quad (9.4.37)$$

Combining (9.4.37) with (9.4.36), we have  $K + 1$  linear equations in  $J$  unknowns, and as before there must be a solution with only  $K + 1$  of the  $\omega(j)$  positive. |

## 9.5 The Converse to the Noisy-Channel Coding Theorem Revisited

We have already observed that the distortion function

$$d(k;j) = \begin{cases} 0; & k = j \\ 1; & \text{otherwise} \end{cases} \quad (9.5.1)$$

where  $K = J$ , is an appropriate distortion measure for studying errors in the reconstruction of a source output. In fact, the average error probability per source letter,  $\langle P_e \rangle$ , which we studied in Chapter 4, is simply the average distortion per letter for the above distortion measure. In this section, we shall calculate  $R(d^*)$  for an arbitrary discrete memoryless source and this distortion measure, thus finding the actual minimum error probability per source letter than can be achieved for source rates above capacity. We shall find that the

lower bound on  $\langle P_e \rangle$  derived in Chapter 4 is, in fact, this minimum error probability for a range of channel capacities just below the entropy of the source. This calculation will have the subsidiary effect of showing how the results of the last section can be applied. The constraint equations

$$\sum_k f_k e^{-\rho d(k;j)} \leq 1$$

simplify for this distortion measure to

$$f_j + \left[ \left( \sum_{k=0}^{K-1} f_k \right) - f_j \right] e^{-\rho} \leq 1; \quad 0 \leq j \leq J-1 \quad (9.5.2)$$

From the symmetry, we can satisfy all these equations with equality by making all the  $f_k$  the same, say  $f_o$ . Then

$$f_k = f_o = [1 + (K-1)e^{-\rho}]^{-1} \quad (9.5.3)$$

Using this  $f$  in the lower bound (9.4.10), we obtain

$$\min_{\mathbf{P}} R_0(\rho, \mathbf{P}) \geq H(U) - \ln [1 + (K-1)e^{-\rho}] \quad (9.5.4)$$

For all  $\rho > 0$ , we then have

$$R(d^*) \geq -\rho d^* + H(U) - \ln [1 + (K-1)e^{-\rho}] \quad (9.5.5)$$

The right-hand side is maximized over  $\rho$ , thus yielding the tightest bound, for  $\rho$  satisfying

$$d^* = \frac{(K-1)e^{-\rho}}{1 + (K-1)e^{-\rho}} \quad (9.5.6)$$

$$\rho = \ln(K-1) + \ln \frac{1-d^*}{d^*} \quad (9.5.7)$$

Substituting (9.5.7) into (9.5.5) and rearranging terms,

$$R(d^*) \geq H(U) - \mathcal{H}(d^*) - d^* \ln(K-1) \quad (9.5.8)$$

where

$$\mathcal{H}(d^*) = -d^* \ln d^* - (1-d^*) \ln(1-d^*) \quad (9.5.9)$$

In conjunction with Theorem 9.2.2, (9.5.8) is equivalent for discrete memoryless sources, to the converse to the noisy-channel coding theorem, Theorem 4.3.4. To achieve an error probability per source digit of  $d^*$ , a channel must have a capacity at least equal to the right-hand side of (9.5.8), and equivalently, for a channel of that capacity,  $d^*$  is a lower bound on the error probability per digit.

We now find the conditions under which (9.5.8) is satisfied with equality. From Theorem 9.4.1, (9.5.4) is satisfied with equality if a solution with  $\omega(j) \geq 0$  exists to (9.4.7) which, for this distortion measure, is

$$\omega(k) + [1 - \omega(k)]e^{-\rho} = Q(k)/f_k; \quad 0 \leq k \leq K - 1 \quad (9.5.10)$$

For  $f_k = f_0 = [1 + (K - 1)e^{-\rho}]^{-1}$ ,

$$\omega(k) = \frac{Q(k)[1 + (K - 1)e^{-\rho}] - e^{-\rho}}{1 - e^{-\rho}} \quad (9.5.11)$$

All  $\omega(k)$  will be nonnegative if

$$Q(k) \geq \frac{1}{e^\rho + (K - 1)} \quad \text{all } k \quad (9.5.12)$$

Thus (9.5.4) is satisfied with equality for all sufficiently large  $\rho$ . Since

$$\min_{\mathbf{P}} R_0(\rho, \mathbf{P}) - \rho d^*$$

touches the  $R(d^*)$  curve in at least one point for each  $\rho > 0$ , and since this point of contact is given by (9.5.6) for all  $\rho$  satisfying (9.5.12), we have

$$R(d^*) = H(U) - \mathcal{H}(d^*) - d^* \ln(K - 1) \quad (9.5.13)$$

for

$$d^* \leq (K - 1)Q_{\min} \quad (9.5.14)$$

where (9.5.14) comes from combining (9.5.12) and (9.5.7), and where  $Q_{\min}$  is the smallest of the  $Q(k)$ . Combining this result with Theorems 9.2.2 and 9.3.1, we have the following theorem.

**Theorem 9.5.1.** Given a discrete memoryless source of entropy  $H(U)$  nats, alphabet size  $K$ , and minimum letter probability  $Q_{\min}$ , and given that the source is connected to a destination by a discrete memoryless channel of capacity  $C$  (in nats per source symbol), then for any  $d^* \leq (K - 1)Q_{\min}$ , an error probability per source digit of  $d^*$  can always be achieved through appropriate coding if

$$C > H(U) - \mathcal{H}(d^*) - d^* \ln(K - 1) \quad (9.5.15)$$

and can never be achieved by any means if

$$C < H(U) - \mathcal{H}(d^*) - d^* \ln(K - 1) \quad (9.5.16)$$

---

The theorem was restricted to discrete memoryless channels for simplicity. It is clearly valid whenever  $C$  can be interpreted both in terms of a maximum average mutual information and a coding theorem. It is remarkable that this minimum error probability can be uniquely specified in terms of so few parameters.

We shall now calculate  $R(d^*)$  for larger values of  $d^*$ . To simplify the notation, we assume that the source letter probabilities are ordered

$$Q(0) \geq Q(1) \geq \cdots \geq Q(K-1) \quad (9.5.17)$$

Both from physical reasoning and from observation of (9.5.11), we would guess that if any of the output probabilities are 0, it should be those corresponding to the least likely inputs. Thus we assume that, for some  $m$  to be selected later,  $\omega(k) = 0$  for  $k \geq m$  and  $\omega(k) > 0$  for  $k \leq m-1$ . For  $k \geq m$ , (9.5.10) then becomes

$$Q(k) = f_k e^{-\rho}; \quad k \geq m \quad (9.5.18)$$

The constraint equation  $f_j + (\sum f_k - f_j)e^{-\rho} \leq 1$  must be satisfied with equality for  $j \leq m-1$ . Thus all the  $f_j$  must be the same, say  $f_o$  for  $j \leq m-1$ , and  $f_j \leq f_o$  for  $j \geq m$ . Evaluating the constraint equation for  $j = 0$ , and using (9.5.18), we get

$$f_o[1 + (m-1)e^{-\rho}] + \sum_{k=m}^{K-1} Q(k) = 1 \quad (9.5.19)$$

$$f_k = f_o = \frac{S_m}{1 + (m-1)e^{-\rho}}; \quad k \leq m-1 \quad (9.5.20)$$

where

$$S_m = \sum_{k=0}^{m-1} Q(k).$$

Observing from (9.5.18) that  $f_m \geq f_{m+1} \geq \cdots \geq f_{K-1}$ , the constraint  $f_j \leq f_o$  will be met for  $j \geq m$  if

$$Q(m) \leq \frac{S_m e^{-\rho}}{1 + (m-1)e^{-\rho}} \quad (9.5.21)$$

The  $\mathbf{f}$  given by (9.5.18) and (9.5.20) will maximize

$$\sum_k Q(k) \ln f_k$$

if all the  $\omega(k)$  given by (9.5.10) are nonnegative. Since  $\omega(m-1) \leq \omega(m-2) \leq \cdots \leq \omega(0)$ , this requires only

$$Q(m-1) \geq \frac{S_m e^{-\rho}}{1 + (m-1)e^{-\rho}} \quad (9.5.22)$$

Thus for the range of  $\rho$  where (9.5.21) and (9.5.22) are satisfied, the given  $\mathbf{f}$  yields

$$\begin{aligned} \min_{\mathbf{P}} R_0(\rho, \mathbf{P}) &= H(U) + \sum_{k=0}^{m-1} Q(k) \ln \frac{S_m}{1 + (m-1)e^{-\rho}} \\ &\quad + \sum_{k=m}^{K-1} Q(k) \ln [Q(k)e^\rho] \end{aligned} \quad (9.5.23)$$

Knowing

$$\min_{\mathbf{P}} R_0(\rho, \mathbf{P})$$

over a range of  $\rho$  now specifies  $R(d^*)$  over the corresponding range of slopes. The parameter  $\rho$  is related to  $d^*$  by

$$d^* = \frac{\partial \min R_0(\rho, \mathbf{P})}{\partial \rho} = S_m \frac{(m-1)e^{-\rho}}{1 + (m-1)e^{-\rho}} + 1 - S_m \quad (9.5.24)$$

For  $\rho$  and  $d^*$  related by (9.5.24),

$$R(d^*) = \min_{\mathbf{P}} R_0(\rho, \mathbf{P}) - \rho d^*.$$

After some manipulation, this becomes

$$R(d^*) = S_m [H(U_m) - \mathcal{H}(\hat{d}) - \hat{d} \ln(m-1)] \quad (9.5.25)$$

when  $H(U_m)$  is the entropy of a reduced ensemble with probabilities  $Q(0)/S_m, \dots, Q(m-1)/S_m$  and  $\hat{d}$  is defined as

$$\hat{d} = \frac{d^* - (1 - S_m)}{S_m} \quad (9.5.26)$$

Substituting (9.5.24) into (9.5.21) and (9.5.22), we find that this solution is valid for a given  $m$  if

$$mQ(m) + \sum_{k=m+1}^{K-1} Q(k) \leq d^* \leq (m-1)Q(m-1) + \sum_{k=m}^{K-1} Q(k) \quad (9.5.27)$$

We observe that, for  $m = K - 1$ , the lower limit on  $d^*$  is the same as the upper limit on  $d^*$  for which (9.5.13) is valid. Also the upper limit on  $d^*$  for one value of  $m$  is the same as the lower limit for the next smaller  $m$ .

Finally, the upper limit on  $d^*$  for  $m = 2$  is  $d_{\max}$ , so that for any  $d^*$  greater than that for which (9.5.13) gives  $R(d^*)$ , (9.5.27) specifies an  $m$  for which (9.5.25) gives  $R(d^*)$ .

There is a simple physical interpretation of (9.5.25). For a given  $m$ , we are trying to represent the source using only the destination letters 0 to  $m - 1$ . With probability  $1 - S_m$ , the source emits one of the letters  $m$  to  $K - 1$ , unit distortion must result, and there is no point in wasting any mutual information on these letters. Thus the minimum average mutual information should be  $S_m$  times the minimum conditional on occurrences of one of the letters 0 to  $m - 1$ . From (9.5.26), we can interpret  $\hat{d}$  as the minimum average distortion conditional on the occurrence of one of the letters 0 to  $m - 1$ . For this conditional distortion, the term in brackets is just the minimum conditional average mutual information that we would expect from (9.5.13).

## 9.6 Discrete-Time Sources with Continuous Amplitudes

Let us now consider sources for which the output is a sequence of statistically independent, identically distributed, continuous-valued, real random variables,  $\dots u_{-1}, u_0, u_1, \dots$ . Assume that the individual letters from the source are described by a probability density  $q(u)$ . The source output is to be represented at the destination by a sequence of real numbers  $\dots v_{-1}, v_0, v_1, \dots$ , and there is a distortion measure  $d(u;v)$  assigning a numerical value, for all real numbers  $u$  and  $v$ , to the distortion if source output  $u$  is represented at the destination by  $v$ . We shall assume as before that  $d(u;v) \geq 0$  and that for each  $u$ , there is a  $v$  (usually  $v = u$ ) for which  $d(u;v) = 0$ . The most common distortion measures are *difference* distortion measures where  $d(u;v)$  is a function only of the difference  $v - u$ , and the most common difference distortion measure is  $d(u;v) = (v - u)^2$ .

The  $R(d^*)$  function for such a source is defined as

$$R(d^*) = \inf I(U;V) \quad (9.6.1)$$

where the infimum is over all joint probability measures on the  $UV$  space subject to the restrictions that  $q(u)$  is the probability density on  $U$  and that the average distortion is, at most,  $d^*$ . We shall use the same definition for  $R(d^*)$  if  $U$  is an arbitrary space rather than the real line and if the probability measure on  $U$  (and a given  $\sigma$  algebra of subsets) is arbitrary rather than being a probability density.

The curve  $R(d^*)$  is nonincreasing and convex  $\cup$  in  $d^*$  for these sources and distortion measures. This can be seen from the same argument as in the discrete case, using integrals here in place of sums. The major difference between  $R(d^*)$  here and  $R(d^*)$  for the discrete case is that here, typically,

$$\lim_{d^* \rightarrow 0} R(d^*) = \infty$$

(compare Figures 9.2.1 and 9.7.1).

Theorem 9.2.1 [which asserts that if  $\bar{d}_L$  is the average distortion per letter on a sequence of  $L$  source letters, then  $(1/L)I(\mathbf{U}^L; \mathbf{V}^L) \geq R(\bar{d}_L)$ ] also holds where  $U$  is an arbitrary space with an arbitrary probability measure. The proof must be modified as follows, however, when the entropy of  $\mathbf{U}^L$  is undefined.

$$\begin{aligned} I(\mathbf{U}^L; \mathbf{V}^L) &= \sum_{i=1}^L I(U_i; \mathbf{V}^L \mid U_1 \cdots U_{i-1}) \\ I(U_i; \mathbf{V}^L \mid U_1 \cdots U_{i-1}) &= I(U_i; \mathbf{V}^L U_1 \cdots U_{i-1}) - I(U_i; U_1 \cdots U_{i-1}) \\ &= I(U_i; \mathbf{V}^L U_1 \cdots U_{i-1}) \geq I(U_i; V_i) \\ I(\mathbf{U}^L; \mathbf{V}^L) &\geq \sum_{i=1}^L I(U_i; V_i) \end{aligned} \quad (9.6.2)$$

The rest of the proof follows as before.

**Theorem 9.6.1.** Theorem 9.2.2 (the converse to the coding theorem for sources relative to a distortion measure) applies in general to all discrete-time memoryless sources with a single letter distortion measure. 

---

The proof is the same as that of Theorem 9.2.2, and thus will be omitted.

The coding theorem for continuous amplitude sources is also quite similar to that for discrete sources. For a given source and test channel (that is, a given joint probability measure on the  $UV$  space), we again consider an ensemble of independently selected code words with each letter selected according to the probability measure on  $V$ . Lemma 9.3.1 applies without change here. If the source and test channel are described by a joint probability density, then the proof of the lemma can be modified simply by replacing all sums with integrals and all sequence probabilities with densities. For an arbitrary joint probability measure [assuming  $I(u;v)$  and  $D(u;v)$  measurable and  $I(U;V) < \infty$ ], the same proof, doctored up by standard results of measure theory, applies.

**Theorem 9.6.2.** Let  $R(d^*)$  be the rate distortion function of a discrete-time, memoryless source with a distortion measure  $d(u;v)$  and assume that a finite set of destination letters  $a_1, \dots, a_J$  exist such that

$$\min_j d(u;a_j)$$

is finite, where the expectation is over the source ensemble. For any  $d^* > 0$ ,  $\delta > 0$ , and sufficiently large  $L$  there exists a source code where the entropy of the set of code words satisfies  $H(V^L) \leq L[R(d^*) + \delta]$  and the average distortion per letter satisfies  $\bar{d}_L \leq d^* + \delta$ . Furthermore, if  $\ln J < R(d^*)$ , there exists a code with  $M' \leq \exp [LR(d^*) + L\delta]$  code words with average distortion  $\bar{d}_L \leq d^* + \delta$ . 

---

*Discussion.* The interpretation of  $H(V^L)$  and  $M$  in terms of the number of binary digits per source digit required to represent the source is the same as for discrete sources. The condition  $\min d(u;a_j) < \infty$  means that there is a way to partition (or quantize) the source output into a finite number of regions with finite average distortion. If this condition is not met, then any block code with a finite number of code words must have an infinite average distortion, since it uses only a finite set of destination letters.

*Proof.* Select a test channel (that is, joint probability measure) for which  $\hat{d} \leq d^*$  and  $I(U;V) \leq R(d^*) + \delta/4$ . Apply Lemma 9.3.1 to this test channel, choosing  $\hat{d} = d^* + \delta/2$ ,  $\hat{R} = R(d^*) + \delta/2$ , and

$$M = \left\lceil \exp \left[ LR(d^*) + \frac{3\delta}{4} L \right] \right\rceil.$$

As in (9.3.11), we then have

$$P_c[D > L(d^* + \delta/2)] \leq P_t(A) + \exp[-e^{L\delta/4} + 1] \quad (9.6.3)$$

where

$$P_t(A) \leq \Pr\{I(\mathbf{u}; \mathbf{v}) > L[R(d^*) + \delta/2]\} + \Pr[D(\mathbf{u}, \mathbf{v}) > L(d^* + \delta/2)] \quad (9.6.4)$$

$$\leq \Pr\{I(\mathbf{u}; \mathbf{v}) > L[I(U; V) + \delta/4]\} + \Pr[D(\mathbf{u}, \mathbf{v}) > L(d^* + \delta/2)] \quad (9.6.5)$$

Since  $I(U; V)$  and  $d^*$  are finite, the law of large numbers asserts that for fixed  $\delta$ ,  $P_t(A)$  approaches 0 with increasing  $L$ , and thus,

$$\lim_{L \rightarrow \infty} P_c[D > L(d^* + \delta/2)] = 0 \quad (9.6.6)$$

For any given code, let  $B$  be the set of source sequences for which  $D[\mathbf{u}; \mathbf{v}(\mathbf{u})] > L(d^* + \delta/2)$ . For any  $L$ , some code in the ensemble with the given  $M$  satisfies  $P(B) \leq P_c[D > L(d^* + \delta/2)]$ . To this code, we add a set of  $J^L$  additional code words, one for each sequence of  $L$  letters from the alphabet  $a_1, \dots, a_J$ . We map the source sequences not in  $B$  into the original  $M$  code words, with distortion per letter at most  $d^* + \delta/2$ . We map the source sequences in  $B$  into the closest of the additional  $J^L$  code words. The entropy of the entire set of code words is bounded, as in (9.3.28), by

$$H(\mathbf{V}^L) \leq L \left\{ R(d^*) + \delta/2 + P(B) \ln J + \frac{\mathcal{H}[P(B)]}{L} \right\} \quad (9.6.7)$$

If  $\ln J \leq R(d^*)$ , the total number of code words is

$$M' = M + J^L \leq 2 \exp L[R(d^*) + \delta/2] \quad (9.6.8)$$

Thus, for sufficiently large  $L$ , the conditions on  $H(\mathbf{V}^L)$  and  $M'$  are met.

Let  $x_B$  be the random variable,

$$x_B = \begin{cases} 1; & \mathbf{u} \in B \\ 0; & \mathbf{u} \notin B \end{cases} \quad (9.6.9)$$

and let  $z_i$ ,  $1 \leq i \leq L$  be the identically distributed variables,

$$z_i = \min_j d(u_i; a_j)$$

The distortion per letter for a sequence  $\mathbf{u} \in B$  is thus

$$\frac{1}{L} \sum_{i=1}^L z_i$$

The average distortion per letter for the given code is thus bounded by

$$\bar{d}_L \leq d^* + \delta/2 + \overline{\frac{x_B}{L} \sum_{i=1}^L z_i} \quad (9.6.10)$$

where  $d^* + \delta/2$  upper bounds the distortion due to  $\mathbf{u} \notin B$ , and the final term is the distortion due to  $\mathbf{u} \in B$ . Let  $z'$  be a number to be selected later, and for each  $l$ ,  $1 \leq l \leq L$ , let

$$x_l = \begin{cases} 1; & z_l \leq z' \\ 0; & z_l > z' \end{cases} \quad (9.6.11)$$

We then have

$$\begin{aligned} \overline{x_B z_l} &= \overline{x_B z_l x_l} + \overline{x_B z_l (1 - x_l)} \\ &\leq \overline{x_B z'} + z_l (1 - x_l) \end{aligned} \quad (9.6.12)$$

The first term above was upperbounded by observing that since  $x_l = 0$  for  $z_l > z'$ ,  $x_l z_l \leq z'$ . The second term was upperbounded by using  $x_B \leq 1$ . Using the definition of  $x_B$  and  $x_l$ , we have

$$P(B) = \overline{x_B}; \overline{z_l (1 - x_l)} = \int_{z'}^{\infty} z_l dF(z_l) \quad (9.6.13)$$

where  $F(z_l)$  is the distribution function of  $z_l$ . Substituting (9.6.12) and (9.6.13) into (9.6.10), we get

$$\bar{d}_L \leq d^* + \delta/2 + z' P(B) + \int_{z'}^{\infty} z dF(z) \quad (9.6.14)$$

Since, by assumption,  $\bar{z}_l < \infty$ , the final integral in (9.6.14) approaches 0 as  $z'$  approaches  $\infty$ ; thus the final integral is less than  $\delta/4$  for sufficiently large  $z'$ . For such a  $z'$ ,  $z' P(B) \leq \delta/4$  for large enough  $L$ . Thus for large enough  $L$ ,  $\bar{d}_L \leq d^* + \delta$ . |

Although this theorem shows that source codes can have rates arbitrarily close to  $R(d^*)$  with average distortions arbitrarily close to  $d^*$ , the conditions are not strong enough to assure that distortions close to  $d^*$  can be achieved after transmission through a noisy channel. The added condition needed is given in the following theorem.

**Theorem 9.6.3.** If a discrete-time memoryless source with distortion measure  $d(u;v)$  satisfies  $\overline{d(u;v)} < \infty$  for every  $v$ , where the expectation is over the source ensemble, and if this source is connected to the destination by a channel of capacity  $C$  nats per source letter for which arbitrarily small error probability can be achieved at any data rate below  $C$ , then an average distortion per letter arbitrarily close to  $d^*$  can be achieved where  $C = R(d^*)$ . More generally, the condition  $\overline{d(u;v)} < \infty$  can be replaced with the condition that  $R(d^*)$  can be approached arbitrarily closely by joint probability measures for which the probability of the set of  $v$  for which  $\overline{d(u;v)} = \infty$  is zero.

*Proof.* For any  $\delta > 0$ , we show that an average distortion per letter  $d < d^* + \delta$  can be achieved. If  $C = 0$ , the theorem is trivial, so we assume that  $C = R(d^*) > 0$ . From the conditions of the theorem,  $d_{\max}$  is finite, where  $d_{\max}$  is the smallest value for which  $R(d_{\max}) = 0$ . Thus  $d^* < d_{\max}$  and  $R(d^*)$  is strictly decreasing with  $d^*$  where  $C = R(d^*)$ . Let  $d_1 = d^* + \delta/2$  and let  $\delta_1$  be the smaller of  $\frac{1}{2}[C - R(d_1)]$  and  $\delta/4$ . From Theorem 9.6.2, we can choose a code of sufficiently long block length  $L$  where the number of code words satisfies

$$M \leq \exp [LR(d_1) + L\delta_1] \leq \exp [L(C - \delta_1)] \quad (9.6.15)$$

and the distortion per letter satisfies

$$\bar{d}_L \leq d_1 + \delta_1 \leq d^* + \frac{3}{4}\delta \quad (9.6.16)$$

and for every  $v$  in the code,  $\overline{d(u;v)} < \infty$ .

Since there are a finite number of letters in this given code, we can choose  $\hat{d} < \infty$  as the maximum  $\overline{d(u;v)}$  over the letters  $v$  in the code. Now the average distortion per sequence given that a source code word  $\mathbf{v}_m$  is erroneously reproduced by the channel as  $\mathbf{v}_{m'}$  is simply the average distortion between  $\mathbf{v}_{m'}$  and those  $\mathbf{u}$  mapped into  $\mathbf{v}_{m'}$ . This is upper bounded by  $L\hat{d}/\Pr(\mathbf{v}_m)$ . Since the probability of channel decoding error can be made arbitrarily small (perhaps by coding many source blocks at a time), the contribution to average distortion from channel errors can be made arbitrarily small for the given source code, and the average distortion per letter including channel errors satisfies  $\bar{d}_L \leq d^* + \delta$ . |

In order to gain insight about why we required  $\overline{d(u;v)} < \infty$  for all  $v$  in the theorem, it is helpful to consider a code for which  $\overline{d(u;v_{l,m})} = \infty$  for, say, the  $l$ th letter of the  $m$ th code word in a code. Suppose that this source code is to be transmitted over a discrete memoryless channel with a minimum transition probability  $P_{\min} > 0$ . For a channel code of block length  $N$ , if there is any channel output that causes  $\mathbf{v}_m$  to be decoded, then for every choice of  $\mathbf{u}_l$ ,  $\mathbf{v}_m$  (and hence  $v_{l,m}$ ) will be produced with probability at least  $P_{\min}^N$ . The average distortion for the  $l$ th letter of the code then satisfies

$$\begin{aligned} \overline{d(l)} &\geq \int q(u_l) P_l(v_{l,m} \mid u_l) d(u_l; v_{l,m}) du_l \\ &\geq P_{\min}^N \int q(u_l) d(u_l; v_{l,m}) du_l = \infty \end{aligned}$$

In attempting to calculate the  $R(d^*)$  function for any given continuous amplitude source and distortion measure, it is convenient (as in Section 9.4) to deal with the function

$$R_0(\rho, \mathbf{P}) = I(U;V) + \rho d \quad (9.6.17)$$

where  $\mathbf{P}$  denotes a given test channel, and  $I(U;V)$  and  $\bar{d}$  denote the average mutual information and average distortion for the source, test-channel combination. As before, the infimum of  $R_0(\rho, \mathbf{P})$  over all test channels (that is, all joint probability measures with the appropriate input probability measure) is the  $R$  axis intercept of a tangent of slope  $-\rho$  to the  $R(d^*)$  curve. If  $q(u)$  is the probability density on the source letters, the lower bound to  $R_0(\rho, \mathbf{P})$  of Theorem 9.4.1 becomes

$$R_0(\rho, \mathbf{P}) \geq \int_u q(u) \ln \frac{f(u)}{q(u)} du \quad (9.6.18)$$

where  $f(u)$  satisfies the constraint

$$\int_u f(u) e^{-\rho d(u,v)} du \leq 1; \quad \text{all } v \quad (9.6.19)$$

Necessary and sufficient conditions on  $f(u)$  and on a transition probability density  $p(v | u)$  for equality in (9.6.18) are that a function  $\omega(v) \geq 0$  satisfy

$$\int \omega(v) e^{-\rho d(u,v)} dv = \frac{q(u)}{f(u)}; \quad \text{all } u \quad (9.6.20)$$

and that (9.6.19) is satisfied with equality for all  $v$  with  $\omega(v) > 0$ . Then

$$p(v | u) = \frac{\omega(v) f(u)}{q(u)} e^{-\rho d(u,v)} \quad (9.6.21)$$

The proof of these statements is the same as the proof of the first half of Theorem 9.4.1, replacing sums with integrals. We cannot show, however, that functions  $p(v | u)$  and  $f(u)$  always exist for which (9.6.18) is satisfied with equality. We cannot even prove that equality can be approached by choosing  $p(v | u)$  and  $f(u)$  more and more carefully, although this latter statement appears to be true. Fortunately, for the important example in the next section, equality can be achieved and the lower bound enables us to easily calculate  $R(d^*)$ .

## 9.7 Gaussian Sources with Square-Difference Distortion

Consider a source whose output is a sequence of statistically independent, identically distributed Gaussian random variables,  $\dots, u_{-1}, u_0, u_1, \dots$ , each with the probability density

$$q(u) = \frac{1}{\sqrt{2\pi A}} \exp\left(-\frac{u^2}{2A}\right) \quad (9.7.1)$$

We want to find the rate distortion function,  $R(d^*)$ , for this source with the distortion measure  $d(u;v) = (u - v)^2$ . We first find a lower bound to  $\min R_0(\rho, \mathbf{P})$  [the  $R$  axis intercept of the tangent of slope  $-\rho$  to the  $R(d^*)$  curve]

and use the bound to lower bound  $R(d^*)$ . We then show that the lower bound is equal to  $R(d^*)$  and discuss the resulting test channel. From (9.6.18),

$$\min R_0(\rho, \mathbf{P}) \geq \int_{-\infty}^{\infty} q(u) \ln \frac{f(u)}{q(u)} du \quad (9.7.2)$$

where  $f(u)$  is any function satisfying the constraint

$$\int_{-\infty}^{\infty} f(u) \exp [-\rho(u - v)^2] du \leq 1; \quad \text{all } v \quad (9.7.3)$$

By a change of variables,  $y = u - v$ , this integral becomes

$$\int f(y + v) \exp [-\rho y^2] dy.$$

Therefore (9.7.3) can be satisfied with equality for all  $v$  if  $f(u)$  is a constant,\* given by

$$f(u) = \sqrt{\rho/\pi} \quad (9.7.4)$$

Substituting this in (9.7.2) and integrating, we obtain

$$\min R_0(\rho, \mathbf{P}) \geq \frac{1}{2} \ln (2\rho eA) \quad (9.7.5)$$

For any  $\rho$ ,  $R(d^*) \geq \min R_0(\rho, \mathbf{P}) - \rho d^*$ , and thus

$$R(d^*) \geq \frac{1}{2} \ln (2\rho eA) - \rho d^* \quad (9.7.6)$$

Maximizing the right-hand side of (9.7.6) over  $\rho$  yields

$$\rho = \frac{1}{2 d^*} \quad (9.7.7)$$

$$R(d^*) \geq \frac{1}{2} \ln \frac{eA}{d^*} - \frac{1}{2} = \frac{1}{2} \ln \frac{A}{d^*} \quad (9.7.8)$$

For  $d^* > A$  the bound in (9.7.8) is negative and can be replaced with  $R(d^*) \geq 0$ . On the other hand, by mapping all  $u$  into  $v = 0$ , we achieve an average distortion  $\bar{d} = A$  with zero-average mutual information, so that

$$R(d^*) = 0; \quad d^* \geq A \quad (9.7.9)$$

We next show that (9.7.8) is satisfied with equality for  $d^* < A$ . In order to do this, we first show that (9.7.5) is satisfied with equality for  $\rho > 1/(2A)$  and find the associated test channel. The necessary condition for equality

\* Note that a constant value for  $f(u)$  will satisfy (9.7.3) with equality for any difference distortion measure and that this constant does not depend on  $q(u)$ .

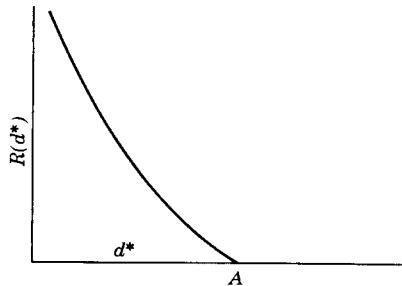
in (9.7.5) is that there is a solution, with  $\omega(v) \geq 0$ , to the equation

$$\int_{-\infty}^{\infty} \omega(v) \exp [-\rho(u - v)^2] dv = \frac{q(u)}{f(u)} ; \quad \text{all } u \quad (9.7.10)$$

$$\int_{-\infty}^{\infty} \omega(v) \sqrt{\rho/\pi} \exp [-\rho(u - v)^2] dv = q(u) \quad (9.7.11)$$

The left side of (9.7.11) is the convolution of a Gaussian probability density of variance  $1/(2\rho)$  with  $\omega(v)$  and the right-hand side is a Gaussian probability density of variance  $A$ . Thus (9.7.11) is satisfied for all  $u$  by

$$\omega(v) = \frac{1}{\sqrt{2\pi A - \pi/\rho}} \exp -\frac{v^2}{2A - 1/\rho} \quad (9.7.12)$$



**Figure 9.7.1.** Rate-distortion function for discrete-time Gaussian source of variance  $A$  with square-difference distortion.

The function  $\omega(v)$  is a probability density for  $\rho > 1/(2A)$ , approaches a unit impulse as  $\rho \rightarrow 1/(2A)$ , and fails to exist as a real solution of (9.7.11) for  $\rho < 1/(2A)$ . It follows that (9.7.5) is satisfied with equality for  $\rho > 1/(2A)$  and consequently that

$$R(d^*) = \begin{cases} \frac{1}{2} \ln \frac{A}{d^*} ; & d^* < A \\ 0 ; & d^* \geq A \end{cases} \quad (9.7.13)$$

This function is sketched in Figure 9.7.1. The  $R$  axis intercept of a tangent of slope  $-\rho$  to this curve is given, for all  $\rho > 0$ , by

$$\min_{\mathbf{P}} R_0(\rho, \mathbf{P}) = \begin{cases} \frac{1}{2} \ln (2\rho e A) ; & \rho > \frac{1}{2A} \\ \rho A ; & \rho \leq \frac{1}{2A} \end{cases} \quad (9.7.14)$$

Since this argument for equality rests on the rather abstract ideas of Sections 9.4 and 9.6, we shall find the test channel implied by (9.7.12) and show that it actually yields distortion  $d^* = 1/(2\rho)$  and average mutual information  $\frac{1}{2} \ln (A/d^*)$ . From (9.6.21), the test-channel transition probability density is

$$p(v | u) = \frac{\omega(v)}{q(u)} p_b(u | v)$$

where the backward transition probability  $p_b(u | v)$  is given by

$$p_b(u | v) = f(u) \exp [-\rho(u - v)^2] \quad (9.7.15)$$

$$= \sqrt{\rho/\pi} \exp [-\rho(u - v)^2] \quad (9.7.16)$$

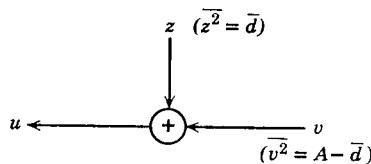


Figure 9.7.2. Test channel (backward form).

If we temporarily think of  $v$  as the channel input, we can think of  $p_b(u | v)$  as arising from an additive Gaussian random variable  $z$  of zero mean and variance  $\frac{1}{2}\rho$ . In this interpretation,  $u = v + z$  where  $v$  and  $z$  are independent Gaussian random variables (see Figure 9.7.2), and the test channel is a “backwards” additive Gaussian noise channel. The average distortion is then just the mean square value of  $z$ ,

$$\bar{d} = \overline{(u - v)^2} = \overline{z^2} = \frac{1}{2\rho} \quad (9.7.17)$$

The average mutual information is

$$\begin{aligned} I(U;V) &= \frac{1}{2} \ln \left[ 1 + \frac{v^2}{z^2} \right] = \frac{1}{2} \ln \frac{\overline{u^2}}{\bar{d}} \\ &= \frac{1}{2} \ln \frac{A}{\bar{d}} \end{aligned} \quad (9.7.18)$$

A somewhat more intuitively satisfying form for the test channel is given

in Figure 9.7.3 where the output  $v$  is generated by first adding  $u$  to an independent, zero mean Gaussian random variable  $w$  of variance  $\bar{d}A/(A - \bar{d})$  and then multiplying the result by  $(A - \bar{d})/A$ . Since  $u$  and  $v$  are jointly Gaussian, the equivalence of these figures is established simply by showing that  $\bar{v}^2$  and  $\bar{wv}$  are the same for each figure. It can also be shown that the multiplier is exactly the operation required to form the minimum variance estimate of  $u$  from the sum  $u + w$ . Both of the source coding theorems, 9.6.2 and 9.6.3, apply to this source since, for any given  $v$ , the average value of  $d(u; v)$  averaged with respect to  $q(u)$ , is  $A + v^2$ .

There is one particularly important channel, namely the discrete-time additive Gaussian noise channel with an energy constraint, over which we can achieve the minimum mean square error  $d^*$ , given by  $C = R(d^*)$ ,

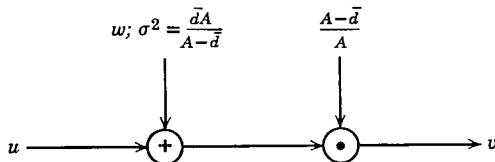


Figure 9.7.3. Test channel (forward form).

without any coding. We simply amplify the source output to meet the channel energy constraint and then appropriately attenuate the channel output. A more interesting variation of this problem arises when we use the channel, say,  $N$  times for each source output. This is, of course, a mathematical abstraction of a situation where a band-limited, Gaussian random-process source of one bandwidth is to be transmitted over a continuous-time, additive Gaussian noise channel using a bandwidth  $N$  times the source bandwidth. In this situation, source and channel coding or processing must be used to approach the limiting distortion of the source coding theorem. Frequency modulation, phase modulation, and pulse code modulation are familiar and simple processing techniques for achieving a moderately small distortion in such situations. As we shall soon see, however, if there is noiseless feedback available from the channel output to the source output, it is possible to achieve the minimum possible distortion with virtually no coding.

Let  $\sigma^2$  be the variance of the noise  $w$  on the channel and let  $A$  be the input energy constraint. Define the quantity  $\bar{d}$  by  $\bar{d} = \sigma^2 A / (A + \sigma^2)$  and notice that  $\sigma^2 = \bar{d}A / (A - \bar{d})$ . It is convenient to multiply the output of the channel by  $(A - \bar{d})/A$  so that the channel takes on the form of the test channel in Figure 9.7.3. Let  $x_1, \dots, x_N$  be the  $N$  channel inputs corresponding to a single source letter and  $y_1, \dots, y_N$  be the corresponding outputs (to the right of the multiplier). Let  $z_n = x_n - y_n$  ( $1 \leq n \leq N$ ) and observe that, if each

input is zero mean and Gaussian with variance  $A$ , then the backward channel of Figure 9.7.2 is equivalent to Figure 9.7.3 and each  $z_n$  is a zero mean Gaussian random variable, independent of  $y_n$ , and with variance  $\bar{d}$ .

Assume, initially, that the source output  $u$  has variance  $A$ , the same as the channel energy constraint. Consider the following choice of channel inputs.

$$\begin{aligned}x_1 &= u \\x_n &= z_{n-1} \sqrt{A/\bar{d}}; \quad 2 \leq n \leq N\end{aligned}\tag{9.7.19}$$

Since  $x_1$  is zero mean and Gaussian with variance  $A$ , it follows that  $z_1$  is zero mean, Gaussian, independent of  $y_1$ , and has variance  $\bar{d}$ . Since  $x_2 = z_1 \sqrt{A/\bar{d}}$ , it follows that  $x_2$  is zero mean and Gaussian with variance  $A$ . Continuing this argument, each  $x_n$  is zero mean and Gaussian with variance  $A$  and each  $z_n$  is zero mean, Gaussian, independent of  $y_n$ , and has variance  $\bar{d}$ . Observe that this transmission scheme requires the transmitter to know the  $n - 1$ st received digit before transmitting the  $n$ th digit.

Suppose that the receiver, for each  $n$ , calculates an estimate,  $v_n$ , of the source digit  $u$  from the equation

$$\begin{aligned}v_1 &= y_1 \\v_n &= v_{n-1} + y_n \left( \frac{\bar{d}}{A} \right)^{(n-1)/2}; \quad 2 \leq n \leq N\end{aligned}\tag{9.7.20}$$

We now show that  $v_n$  can also be expressed as

$$v_n = u - z_n \left( \frac{\bar{d}}{A} \right)^{(n-1)/2}\tag{9.7.21}$$

This is established by induction. For  $n = 1$ , we have  $v_1 = y_1 = x_1 - z_1 = u - z_1$ , which agrees with (9.7.21). Now assume that (9.7.21) is valid for  $n - 1$ , and substitute it for  $v_{n-1}$  in (9.7.20). Using  $y_n = x_n - z_n$ , and using (9.7.19) for  $x_n$  we find that (9.7.21) is valid for  $n$ .

We see from (9.7.21) that  $z_n$  is proportional to the error of the estimate on the  $n$ th transmission. Thus on each successive transmission we are transmitting an amplified replica of the previous error (see (9.7.19)). From (9.7.20), the receiver uses each received signal to correct the error on the previous transmission.

The final average distortion, after  $N$  transmissions, from (9.7.21), is

$$\bar{d}_N = \overline{(v_N - u)^2} = \bar{d} \left( \frac{\bar{d}}{A} \right)^{N-1}\tag{9.7.22}$$

It follows that  $\frac{1}{2} \ln (A/\bar{d}_N) = N/2 \ln (A/\bar{d})$ . This is  $N$  times the capacity of the channel, and thus  $\bar{d}_N$  is the minimum possible distortion for this source and channel as determined from the source coding theorem. If the source

variance is different from the channel energy constraint, we simply scale  $u$  before the first transmission and rescale the final estimate, leaving the ratio of source variance to final distortion at the same optimum value. The above result is due to Elias (1961).

The above technique for using feedback to transmit Gaussian random variables with minimum square-difference distortion is closely related to a technique for transmitting digital data over a Gaussian noise channel with feedback, discovered by Schalkwijk and Kailath (1966). In order to develop this result, we first need the result that, for each  $n$ ,  $1 \leq n \leq N$ ,  $v_n$  and  $z_n$  in (9.7.21) are statistically independent. To see this, we recall that  $z_n$  and  $y_n$  are independent. Now, if we use induction and assume that  $z_{n-1}$  and  $v_{n-1}$  are independent, it follows from (9.7.19) that  $x_n$  and  $v_{n-1}$  are independent. Since  $z_n$  is a linear combination of  $x_n$  and  $w_n$ , it follows that  $z_n$  and  $v_{n-1}$

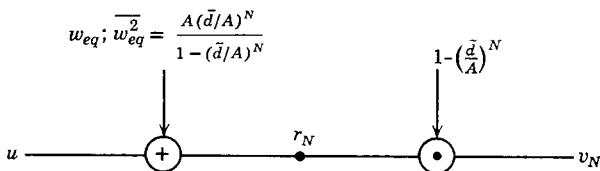


Figure 9.7.4. Model for feedback transmission of a Gaussian random variable.

are independent. Thus, from (9.7.20),  $z_n$  and  $v_n$  are independent if  $z_{n-1}$  and  $v_{n-1}$  are independent. Since  $z_1$  and  $v_1$  are clearly independent, the result follows. From this result, we can represent the entire transmission, from  $u$  to  $v_N$ , by Figure 9.7.2, using  $z_N(\bar{d}/A)^{(N-1)/2}$  as the noise. Equivalently, we can represent the transmission as in Figure 9.7.4.

Since the equivalent noise  $w_{eq}$  in Figure 9.7.4 is Gaussian and independent of  $u$ , we can use Figure 9.7.4 for the transmission scheme in (9.7.19) whether or not  $u$  is Gaussian.\* Now, suppose that we wish to transmit one of a set of  $M$  messages. We can encode these messages into a set of numbers from  $-\sqrt{A}$  to  $\sqrt{A}$  by

$$m \rightarrow u_m = -\sqrt{A} + \frac{2m\sqrt{A}}{M-1}; \quad 0 \leq m \leq M-1 \quad (9.7.23)$$

We can then decode by mapping the received number,  $r_N = v_N/[1 - (\bar{d}/A)^N]$  into the closest message point  $u_m$ . If the noise  $|w_{eq}|$  is less than  $\sqrt{A}/(M-1)$ ,

\* The student who is wary of the abstract nature of this argument can express  $z_n$  in terms of the actual channel noise variables  $w_n$  by  $z_n = x_n \bar{d}/A - w_n(A - \bar{d})/A$ . Using this and (9.7.19) for each value of  $n$ ,  $z_N$  can be expressed as a linear combination of  $u$  and  $w_n$ ,  $1 \leq n \leq N$ . Combining this with (9.7.21), the model in Figure 9.7.4 can be directly verified.

no error can occur. Thus, for each message  $m$ ,

$$P_{e,m} \leq 2\Phi\left[-\frac{\sqrt{A}}{(M-1)\sqrt{w_{ea}^2}}\right]; \quad \Phi(x) = \int_{-\infty}^x \frac{1}{\sqrt{2\pi}} e^{-t^2/2} dt \quad (9.7.24)$$

Taking  $M = e^{NR}$ , where  $R$  is the rate in nats per channel symbol, and recalling that the channel capacity in nats per channel symbol is  $C = \frac{1}{2}\ln(A/d)$ , this becomes

$$P_{e,m} \leq 2\Phi\left[-e^{NC-NR}\left(\frac{\sqrt{1-e^{-2NC}}}{1-e^{-NR}}\right)\right] \quad (9.7.24)$$

For  $R < C$ , the term in parentheses is lower bounded by 1, and using the bound on  $\Phi(-x)$  in (8.2.38), we have

$$P_{e,m} \leq \sqrt{\frac{2}{\pi}} e^{-N(C-R)} \exp\left[-\frac{e^{2N(C-R)}}{2}\right] \quad (9.7.25)$$

If the messages are equally likely, then the separation between the messages can be increased to  $2\sqrt{3A/\sqrt{M^2-1}}$  without violating the power constraint on the first transmission. The resulting error probability is then

$$P_{e,m} \leq \sqrt{\frac{2}{3\pi}} e^{-N(C-R)} \exp\left[-\frac{3e^{2N(C-R)}}{2}\right]$$

The significant and unusual thing about this result is that  $P_{e,m}$  decreases as a double exponential in  $N$  rather than as a single exponential like all of our other results on error probability.

### Gaussian Random-Process Sources

The results on discrete-time Gaussian sources will now be extended to a source whose output is a stationary, zero mean Gaussian random process with a correlation function  $\mathcal{R}_1(\tau)$ . We shall consider the source output over an interval  $(-T/2, T/2)$  and expand it in a Karhunen-Loeve expansion,

$$u(t) = \sum_i u_i \theta_i(t); \quad -T/2 \leq t \leq T/2 \quad (9.7.26)$$

where the orthonormal functions  $\theta_i(t)$  are the solutions, over  $-T/2 \leq t \leq T/2$ , to

$$\int_{-T/2}^{T/2} \mathcal{R}_1(t_1 - t_2) \theta_i(t_2) dt_2 = \lambda_i \theta_i(t_1) \quad (9.7.27)$$

As we saw in Chapter 8, the  $u_i$  are statistically independent, zero mean, Gaussian random variables with variances

$$\overline{u_i^2} = \lambda_i \quad (9.7.28)$$

Suppose that we wish to represent  $u(t)$  at a destination by a waveform  $v(t)$ , and define the distortion per unit time between  $u(t)$  and  $v(t)$  over the interval  $(-T/2, T/2)$  to be

$$d_T[u(t); v(t)] = 1/T \int_{-T/2}^{T/2} [u(t) - v(t)]^2 dt \quad (9.7.29)$$

If  $v(t)$  is expanded in the same orthonormal expansion as  $u(t)$ ,  $v(t) = \sum v_i \theta_i(t)$ , then

$$d_T[u(t); v(t)] = 1/T \sum_i (u_i - v_i)^2 \quad (9.7.30)$$

For any given probability measure over the sequence  $u_i$  and the sequence  $v_i$  for which the  $u_i$  are independent, zero mean, Gaussian random variables with  $\overline{u_i^2} = \lambda_i$  for each  $i$ , we can consider the mutual information in nats per unit time,  $(1/T) I(\mathbf{U}; \mathbf{V})$ , between the  $\mathbf{u}$  and  $\mathbf{v}$  sequences, and the average distortion,  $\bar{d}_T$ , per unit time, given as the average of (9.7.30) over  $\mathbf{u}$  and  $\mathbf{v}$ . The rate distortion function for the given source and given interval  $T$  is then defined as

$$R_T(d^*) = \inf \frac{1}{T} I(\mathbf{U}; \mathbf{V}) \quad (9.7.31)$$

where the infimum is over probability measures for which the  $u_i$  are independent, zero mean, Gaussian random variables with  $\overline{u_i^2} = \lambda_i$  and for which  $\bar{d}_T \leq d^*$ .

We first find  $R_T(d^*)$ , then take the limit

$$R(d^*) = \lim_{T \rightarrow \infty} R_T(d^*), \quad (9.7.32)$$

and finally show that  $R(d^*)$  has the same interpretations as  $R(d^*)$  for discrete-time memoryless sources.

In order to calculate  $R_T(d^*)$ , we first define

$$R_{0,T}(\rho, \mathbf{P}) = \frac{1}{T} I(\mathbf{U}; \mathbf{V}) + \rho \bar{d}_T \quad (9.7.33)$$

where  $\mathbf{P}$  in (9.7.33) denotes a joint probability measure with the given source statistics and  $I(\mathbf{U}; \mathbf{V})$  and  $\bar{d}_T$  are calculated with that measure. Since the  $u_i$  are statistically independent, we can use the same argument as in (9.6.2) to obtain

$$\frac{1}{T} I(\mathbf{U}; \mathbf{V}) \geq \frac{1}{T} \sum_{i=1}^{\infty} I(U_i; V_i) \quad (9.7.34)$$

with equality if each  $U_i V_i$  pair is statistically independent from the other pairs. Also, from (9.7.30):

$$\bar{d}_T = \frac{1}{T} \sum_{i=1}^{\infty} \overline{(u_i - v_i)^2} \quad (9.7.35)$$

Thus

$$R_{0,T}(\rho, \mathbf{P}) \geq \frac{1}{T} \sum_{i=1}^{\infty} [I(U_i; V_i) + \rho \overline{(u_i - v_i)^2}] \quad (9.7.36)$$

with equality if each pair is statistically independent from the other pairs. We have seen, in (9.7.14), that each term of this type is individually minimized by

$$\min [I(U_i; V_i) + \rho \overline{(u_i - v_i)^2}] = g(\lambda_i, \rho) \triangleq \begin{cases} \frac{1}{2} \ln(2\rho e \lambda_i); & \rho > \frac{1}{2\lambda_i} \\ \rho \lambda_i; & \rho \leq \frac{1}{2\lambda_i} \end{cases} \quad (9.7.37)$$

The left-hand side of (9.7.36) is minimized by choosing each  $U_i V_i$  joint ensemble to satisfy (9.7.37) and by making each pair independent of all other pairs.

$$\inf_{\mathbf{P}} R_{0,T}(\rho, \mathbf{P}) = \frac{1}{T} \sum_i g(\lambda_i, \rho) \quad (9.7.38)$$

$$R_T(d_T^*) = \min_{\rho \geq 0} \left[ \frac{1}{T} \sum_i g(\lambda_i, \rho) - \rho d_T^* \right] \quad (9.7.39)$$

Observing that  $g(\lambda_i, \rho)$  is differentiable with respect to  $\rho$ , even at the boundary  $\rho = 1/(2\lambda_i)$ , we can set the derivative with respect to  $\rho$  of the right-hand side of (9.7.39) equal to 0, obtaining for the minimizing  $\rho$ ,

$$d_T^* = \frac{1}{T} \left[ \sum_{i: \lambda_i > \frac{1}{2\rho}} \frac{1}{2\rho} + \sum_{i: \lambda_i \leq \frac{1}{2\rho}} \lambda_i \right] \quad (9.7.40)$$

Substituting (9.7.40) into (9.7.39) gives us

$$R_T(d_T^*) = \frac{1}{T} \sum_{i: \lambda_i > \frac{1}{2\rho}} \frac{1}{2} \ln(2\rho \lambda_i) \quad (9.7.41)$$

It can be seen that (9.7.40) and (9.7.41) are parametric equations determining  $d_T^*$  and  $R_T(d_T^*)$  in terms of  $\rho$  where  $-\rho$  is the slope of  $R_T(d_T^*)$ . For  $\rho$  small enough to satisfy  $\lambda_i \leq 1/(2\rho)$  for all  $i$ ,  $R_T(d_T^*) = 0$  and  $d_T^* = (1/T) \sum \lambda_i$  which is the average power of the source output, henceforth denoted as  $d_{\max}^*$ .

We now observe that (9.7.40) and (9.7.41) are each in the form of  $1/T$  times the sum over  $i$  of a function of the eigenvalues. In (9.7.40) the function is  $1/(2\rho)$  for  $\lambda_i > 1/(2\rho)$  and  $\lambda_i$  for  $\lambda_i \leq 1/(2\rho)$ . In (9.7.41) the function is  $\frac{1}{2} \ln(2\rho \lambda_i)$  for  $\lambda_i > 1/(2\rho)$  and 0 for  $\lambda_i \leq 1/(2\rho)$ . Lemma 8.5.3 applies to

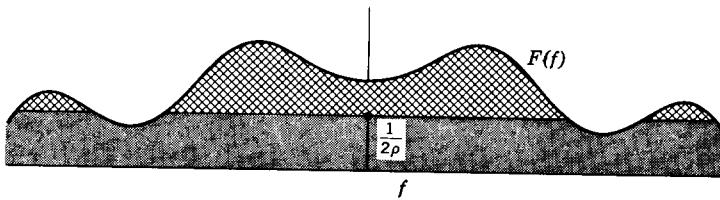


Figure 9.7.5. Interpretation of integrals for  $d^*$  and  $R(d^*)$  for stationary Gaussian random process source.

both of these functions, and for any fixed  $\rho > 0$ , we can pass to the limit  $T \rightarrow \infty$ , obtaining the parametric equations:

$$d^* = \lim_{T \rightarrow \infty} d_T^* = \frac{1}{2\rho} \int_{f: F(f) > \frac{1}{2\rho}} df + \int_{f: F(f) \leq \frac{1}{2\rho}} F(f) df \quad (9.7.42)$$

$$R(d^*) = \lim_{T \rightarrow \infty} R_T(d^*) = \int_{f: F(f) > \frac{1}{2\rho}} \frac{1}{2} \ln [2\rho F(f)] df \quad (9.7.43)$$

where  $F(f) = \int \mathcal{R}_1(\tau) e^{j2\pi f\tau} d\tau$  is the spectral density of the source. Figure 9.7.5 illustrates the meaning of these integrals. We notice that  $d^*$  is the area of the shaded region and that the range of integration for  $R(d^*)$  is the range of frequencies where there is a cross-hatched region.

The test channel that achieves this  $R(d^*)$  function can be represented either by Figure 9.7.6, which is the continuous analog of Figure 9.7.2 or by Figure 9.7.7, which is the continuous analog of a slight variation of Figure 9.7.3. The noise process  $z(t)$  in each figure has the spectral density given by the shaded portion of Figure 9.7.5. The filters in the forward test channel of Figure 9.7.7 are unrealizable, but by adding sufficient delay to the representation  $v(t)$ , they can be approximated arbitrarily closely by realizable filters. It should be remembered, however, that these test channels are artificial constraints in rate-distortion theory. In distinction to the discrete-time Gaussian source, the test channel for a Gaussian random-process source is not generally used at capacity (assuming an appropriate power limitation

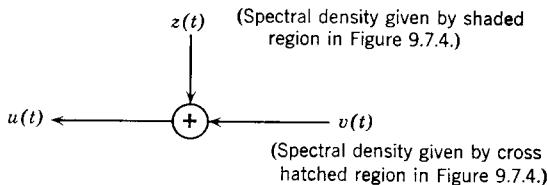


Figure 9.7.6. Backward test channel for stationary Gaussian random process source.

on the channel input) by the given source. Thus, even if we had the appropriate test channel available to achieve distortion  $d^*$  with rate  $R(d^*)$ , we could achieve an average distortion less than  $d^*$  by using additional data processing and increasing the average mutual information on the channel above  $R(d^*)$ . The major purpose of these test channels is to provide us with an easily remembered picture of what an efficient representation of a Gaussian random process should accomplish. In particular, the representation should completely ignore those frequency components of the source where the spectral density is small.

We have defined  $R(d^*)$  as the limit over a very large time interval of the minimum average mutual information per unit time between  $u(t)$  and  $v(t)$  for a given average distortion  $d^*$  per unit time. It is clear, from the same

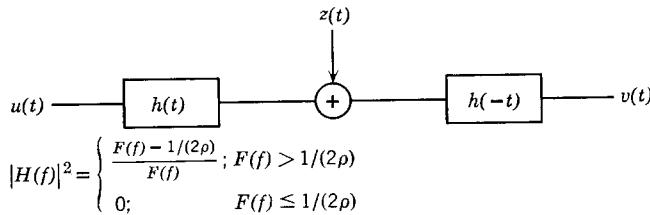


Figure 9.7.7. Forward test channel.

argument as in Theorem 9.2.2, that if the source is connected to the destination by means of a channel of capacity  $C$  nats per unit time, then the average distortion that can be achieved, by any type of processing, is lower bounded by the  $d^*$  for which  $C = R(d^*)$ . The source coding theorem also applies here but, unfortunately, the proof differs from that of Theorem 9.6.2 in enough details to require a separate proof.

**Theorem 9.7.1.** Let the output of a source be a stationary Gaussian random process  $u(t)$  with a correlation function  $\mathcal{R}_1(\tau)$  and an integrable and bounded spectral density  $F(f)$ . Let the distortion measure between  $u(t)$  and  $v(t)$  over an interval  $(-T/2, T/2)$  be given by (9.7.29). Then, for any  $d^* > 0$ , any  $\delta > 0$ , and any  $T$  sufficiently large there exists a code with  $M \leq \exp [TR(d^*) + T\delta]$  code words [where  $R(d^*)$  is given by (9.7.42) and (9.7.43)] such that the average distortion per unit time between  $u(t)$ ,  $-T/2 \leq t \leq T/2$ , and the code word into which it is mapped, is less than or equal to  $d^* + \delta$ .

---

*Proof.* If  $d^* \geq d_{\max}^*$ , the theorem is trivial since the code can consist of the single code word  $v(t) = 0$ . If  $d^* < d_{\max}^*$ , choose  $\rho$  to satisfy (9.7.42) for

the given  $d^*$ . For any given  $\delta > 0$ , let the time interval  $T$  be arbitrary but large enough to satisfy

$$R_T(d_T^*) \leq R(d^*) + \delta/4 \quad (9.7.44)$$

$$d_T^* \leq d^* + \delta/4 \quad (9.7.45)$$

where  $d_T^*$  and  $R_T(d_T^*)$  are given by (9.7.40) and (9.7.41). Now consider an ensemble of codes with

$$M' = \lfloor \exp [TR(d^*) + T\delta] \rfloor - 1 \quad (9.7.46)$$

code words. The code words are chosen independently according to the output probabilities of the test channel that yields  $R_T(d_T^*)$ ; that is, the code words are given by

$$v_m(t) = \sum_l v_{m,l} \theta_l(t).$$

$1 \leq m \leq M'$ , where the coefficients,  $v_{l,m}$ , are independent zero mean Gaussian random variables with

$$\overline{v_{l,m}^2} = \begin{cases} \lambda_l - \frac{1}{2\rho} & \lambda_l > \frac{1}{2\rho}; \\ 0; & \lambda_l \leq \frac{1}{2\rho} \end{cases} \quad (9.7.47)$$

Let  $L$  be the number of eigenvalues,  $\lambda_l$ , that are greater than  $1/(2\rho)$ . Then, for every code word,  $v_{l,m} = 0$  for all  $l > L$ , and the distortion between any code word  $v_m(t)$  and  $u(t)$  is given by

$$D[u(t); v_m(t)] = \sum_{l=1}^L (u_l - v_{m,l})^2 + \sum_{l=L+1}^{\infty} u_l^2 \quad (9.7.48)$$

Observe from this that, for any given set of code words, the minimum-distortion code word for a given  $u(t)$  is determined solely from the first  $L$  components of  $\sum u_l \theta_l(t)$ . Observe also that the contribution to average distortion per unit time from all components except the first  $L$  is simply

$$\frac{1}{T} \sum_{l=L+1}^{\infty} \lambda_l.$$

Thus, if we let  $\mathbf{u} = (u_1, \dots, u_L)$ ,  $\mathbf{v}_m = (v_{m,1}, \dots, v_{m,L})$ , and

$$D_1(\mathbf{u}; \mathbf{v}_m) = \sum_{l=1}^L (u_l - v_{m,l})^2,$$

we can express the average distortion per unit time for any code in the ensemble as

$$\bar{d}_T = \frac{1}{T} \left[ \overline{D_1(\mathbf{u}; \mathbf{v}(\mathbf{u}))} + \sum_{l=L+1}^{\infty} \lambda_l \right] \quad (9.7.49)$$

where  $\mathbf{v}(\mathbf{u})$  is the code word into which  $\mathbf{u}$  is mapped. Next define  $\hat{R}$  and  $\hat{d}$  by

$$\hat{R} = R_T(d_T^*) + \frac{\delta}{2} \quad (9.7.50)$$

$$\hat{d} = \frac{1}{T} \sum_{t=1}^L \frac{1}{2\rho} + \frac{\delta}{2} \quad (9.7.51)$$

Let  $P_c(D_1 > T \hat{d})$  be the probability, over the ensemble of codes and source waveforms, that  $D_1[\mathbf{u};\mathbf{v}(\mathbf{u})] > T \hat{d}$ . As in Lemma 9.3.1,

$$P_c(D_1 > T \hat{d}) \leq P_t(A) + \exp(-M'e^{-T\hat{R}}) \quad (9.7.52)$$

where

$$A = \{\mathbf{u}, \mathbf{v} : I(\mathbf{u}; \mathbf{v}) > T\hat{R} \text{ or } D_1(\mathbf{u}; \mathbf{v}) > T \hat{d}\} \quad (9.7.53)$$

and  $P_t(A)$  is the probability of event  $A$  in the test-channel ensemble. In what follows, we shall first upper bound the right-hand side of (9.7.52), and use that to obtain an upper bound on  $\overline{D_1[\mathbf{u};\mathbf{v}(\mathbf{u})]}$ .

Since, on the test channel,  $I(\mathbf{U}; \mathbf{V}) = TR_T(d_T^*)$  and

$$\overline{D_1(\mathbf{u}; \mathbf{v})} = \sum_{l=1}^L \frac{1}{2\rho}$$

we can upper bound  $P_t(A)$  by

$$\begin{aligned} P_t(A) &\leq \text{Prob}[I(\mathbf{u}; \mathbf{v}) > I(\mathbf{U}; \mathbf{V}) + (\delta/2)T] \\ &\quad + \text{Prob}[D_1(\mathbf{u}; \mathbf{v}) > \overline{D_1(\mathbf{u}; \mathbf{v})} + (\delta/2)T] \end{aligned} \quad (9.7.54)$$

From the Chebyshev inequality, it follows that

$$P_t(A) \leq \frac{4 \text{VAR}[I(\mathbf{u}; \mathbf{v})]}{\delta^2 T^2} + \frac{4 \text{VAR}[D_1(\mathbf{u}; \mathbf{v})]}{\delta^2 T^2} \quad (9.7.55)$$

Since the  $u_l, v_l$  pairs are independent over the test-channel ensemble

$$\text{VAR}[I(\mathbf{u}; \mathbf{v})] = \sum_{l=1}^L \text{VAR}[I(u_l; v_l)] \quad (9.7.56)$$

Recalling that  $v_l$  and  $u_l - v_l$  are independent Gaussian random variables of variance  $\lambda_l - 1/(2\rho)$  and  $1/(2\rho)$ , respectively, we obtain

$$I(u_l; v_l) = \ln \frac{p_l(u_l | v_l)}{q_l(u_l)} = \frac{1}{2} \ln(2\rho\lambda_l) - \rho(u_l - v_l)^2 + \frac{u_l^2}{2\lambda_l} \quad (9.7.57)$$

$$\text{VAR}[I(u_l; v_l)] = 1 - \frac{1}{2\lambda_l\rho} \leq 1 \quad (9.7.58)$$

$$\text{VAR}[I(\mathbf{u}; \mathbf{v})] \leq L \quad (9.7.59)$$

Likewise

$$\begin{aligned}\text{VAR}[D_1(\mathbf{u}; \mathbf{v})] &= \sum_{l=1}^L \text{VAR}[(u_l - v_l)^2] \\ &= \sum_{l=1}^L \frac{2}{(2\rho)^2} = \frac{L}{2\rho^2}\end{aligned}\quad (9.7.60)$$

where we have used the fact that  $u_l - v_l$  is Gaussian with variance  $1/(2\rho)$ . Substituting these results into (9.7.55),

$$P_t(A) \leq \frac{4L[1 + 1/(2\rho^2)]}{\delta^2 T^2} \quad (9.7.61)$$

Combining (9.7.46), (9.7.44), and (9.7.50), we obtain

$$M' \geq \exp E[\hat{R} + \delta/4] - 2 \quad (9.7.62)$$

Substituting (9.7.61) and (9.7.62) into (9.7.52) yields

$$P_c(D_1 > T \hat{d}) \leq \frac{4L[1 + 1/(2\rho^2)]}{\delta^2 T^2} + \exp(-e^{T\delta/4} + 2) \quad (9.7.63)$$

Now consider a particular code in the ensemble for which (9.7.63) is satisfied and let

$$B = \{\mathbf{u}: D_1[\mathbf{u}; \mathbf{v}(\mathbf{u})] > T \hat{d}\} \quad (9.7.64)$$

We now add one additional code word  $v_0(t) = 0$  to the  $M'$  code words already selected and observe that  $M = M' + 1$  code words satisfies the conditions of the theorem. If  $\mathbf{u} \in B$ , we shall map  $\mathbf{u}$  into  $v_0(t)$  and otherwise we map  $\mathbf{u}$  into the closest code word. For this new code, we have

$$\overline{D_1[\mathbf{u}; \mathbf{v}(\mathbf{u})]} \leq T \hat{d} + P(B) \sum_{l=1}^L \bar{d}_{l,B} \quad (9.7.65)$$

where  $\bar{d}_{l,B}$  is the average distortion, or average value of  $u_l^2$ , given that  $\mathbf{u} \in B$ . As in Theorem 9.6.2, we can upper bound  $\bar{d}_{l,B}$  by assuming that all large values of  $u_l^2$  belong to sequences  $\mathbf{u}$  in  $B$ . Since  $q_l(u_l) = q_l(-u_l)$ , this gives us

$$P(B) \bar{d}_{l,B} \leq 2 \int_{u_l'}^{\infty} u_l^2 q_l(u_l) du_l \quad (9.7.66)$$

where  $u_l'$  is determined by

$$P(B) = 2 \int_{u_l'}^{\infty} q_l(u_l) du_l \quad (9.7.67)$$

Using the Schwartz inequality on (9.7.66), we obtain

$$\begin{aligned}P(B) \bar{d}_{l,B} &\leq \sqrt{\left[ 2 \int_{u_l'}^{\infty} u_l^4 q_l(u_l) du_l \right] \left[ 2 \int_{u_l'}^{\infty} q_l(u_l) du_l \right]} \\ &\leq \sqrt{\left[ \int_{-\infty}^{\infty} u_l^4 q_l(u_l) du_l \right] P(B)} = \lambda_l \sqrt{3P(B)}\end{aligned}\quad (9.7.68)$$

Substituting (9.7.68) into (9.7.65) and upper bounding by summing over all  $l$ ,

$$\overline{D_1[\mathbf{u}; \mathbf{v}(\mathbf{u})]} \leq T[\hat{d} + \sqrt{3P(B)} d_{\max}^*] \quad (9.7.69)$$

where  $d_{\max}^* = (1/T)\sum \lambda_l$  is the average source power. Substituting (9.7.69), (9.7.51), (9.7.40), and (9.7.45) into (9.7.49), we get

$$\overline{d_T} \leq d^* + \frac{3\delta}{4} + \sqrt{3P(B)} d_{\max}^* \quad (9.7.70)$$

Finally,  $P(B)$  is upper bounded by the right-hand side of (9.7.63). The proof will be completed by showing that this goes to 0 as  $T \rightarrow \infty$ . We observe that  $L$  is a function of  $T$ , but from the Kac, Murdock, and Szego theorem (see Lemma 8.5.2),

$$\lim_{T \rightarrow \infty} \frac{L}{T} = \int_{f: F(f) > \frac{1}{2\rho}} df \quad (9.7.71)$$

If  $F(f) = 1/(2\rho)$  over a nonzero range of frequencies, (9.7.71) is not valid, but  $L/T$  can be upper bounded in the limit of large  $T$  by using a smaller value of  $\rho$  in (9.7.71). Thus  $P(B)$  approaches 0 as  $T \rightarrow \infty$ , and for large enough  $T$ ,  $\overline{d_T} \leq d^* + \delta$ , completing the proof. |

## 9.8 Discrete Ergodic Sources

In this section, we consider a discrete stationary ergodic source with the alphabet  $(0, 1, \dots, K - 1)$ . The source output,  $\dots, u_{-1}, u_0, u_1, \dots$ , is to be represented at a destination by a sequence of letters  $\dots v_{-1}, v_0, v_1, \dots$  each selected from an alphabet  $(0, 1, \dots, J - 1)$ . We shall assume that there is a distortion measure  $d(\mathbf{u}; v_0)$  between sequences of source letters and individual destination letters. For example, if the destination were interested only in the occurrence of pairs of consecutive ones in the output of a binary source, a reasonable distortion measure would be

$$d(u_{-1}, u_0; v_0) = \begin{cases} 0 & u_{-1} \cdot u_0 = v_0 \\ 1 & \text{otherwise} \end{cases}$$

For this example, the distortion for the  $l$ th letter of the output sequence would be  $d(u_{l-1}, u_l; v_l)$ , where  $d$  is the same function as above. For a general distortion measure  $d(\mathbf{u}; v_0)$ , we can define the distortion for the  $l$ th destination letter by first defining the  $l$ th shift of an input sequence  $\mathbf{u}$  by  $T^l \mathbf{u} = \mathbf{u}'$  where  $u'_n = u_{n+l}$ . In terms of this shift operator, the distortion between  $\mathbf{u}$  and  $v_l$  is given by  $d(T^l \mathbf{u}; v_l)$ . Finally, in terms of a distortion measure  $d(\mathbf{u}; v_0)$ , we define the total distortion between  $\mathbf{u}$  and a sequence  $v_0, v_1, \dots, v_L$  as

$$D(\mathbf{u}; v_0, \dots, v_L) = \sum_{l=0}^L d(T^l \mathbf{u}; v_l) \quad (9.8.1)$$

For simplicity, we assume throughout that  $d(\mathbf{u}; v_0)$  is nonnegative and bounded. We do not assume, however, that for every  $\mathbf{u}$  there is a choice of  $v_0$  for which  $d(\mathbf{u}; v_0) = 0$ . We shall refer to the class of distortion measures just defined as *additive, time-invariant* distortion measures.

In what follows, we first define a rate distortion function for such sources and distortion measures. We then prove a source coding theorem showing that this rate distortion function has the same interpretation as the rate distortion function for discrete memoryless sources with a single-letter distortion measure.

Let  $\mathbf{u} = (\dots, u_{-1}, u_0, u_1, \dots)$  denote an infinite sequence from the source and let  $\mathbf{u}_L = (u_1, u_2, \dots, u_L)$  denote a sequence of  $L$  letters from  $\mathbf{u}$ . Similarly, let  $\mathbf{v}_L = (v_1, \dots, v_L)$  denote the corresponding sequence of destination letters. Consider transmitting the sequence  $\mathbf{u}_L$  over a noisy channel and receiving  $\mathbf{v}_L$ . We can describe the channel and any accompanying processing by an  $L$ th order transition probability assignment  $P_L(\mathbf{v}_L | \mathbf{u}_L)$ . Assume that, given  $\mathbf{u}_L$ , the received sequence is statistically independent of the other letters in the infinite length sequence  $\mathbf{u}$ , that is, that  $P(\mathbf{v}_L | \mathbf{u}) = P_L(\mathbf{v}_L | \mathbf{u}_L)$ . In conjunction with the source probability measure,  $P_L(\mathbf{v}_L | \mathbf{u}_L)$  determines an average mutual information  $I(U_1, \dots, U_L; V_1, \dots, V_L)$  between the sequences  $\mathbf{u}_L$  and  $\mathbf{v}_L$ . Similarly,  $P_L(\mathbf{v}_L | \mathbf{u}_L)$  determines an average value of total distortion\* between  $\mathbf{u}$  and  $\mathbf{v}_L$ ,  $D(\mathbf{u}; \mathbf{v}_L)$ .

The  $L$ th order rate-distortion function for the source and distortion measure is defined as

$$R_L(d^*) = \min_{P_L: \frac{1}{L} \bar{D} \leq d^*} \frac{1}{L} I(U_1, \dots, U_L; V_1, \dots, V_L) \quad (9.8.2)$$

The minimization is over all transition probability assignments,  $P_L(\mathbf{v}_L | \mathbf{u}_L)$  such that the average distortion per letter,  $(1/L)\bar{D}(\mathbf{u}; \mathbf{v}_L)$  does not exceed  $d^*$ . For those situations where

$$\min_{v_0} d(\mathbf{u}; v_0)$$

is not 0 for all  $\mathbf{u}$ , the constraint set in the above minimization might be empty for small  $d^*$ , and we take  $R_L(d^*)$  as  $\infty$  in those cases. By the same argument as in Section 9.2,  $R_L(d^*)$  is nonnegative, nonincreasing, and convex  $\cup$  in  $d^*$ . The rate-distortion function for the source is defined as

$$R(d^*) = \lim_{L \rightarrow \infty} R_L(d^*) \quad (9.8.3)$$

The following simple theorem asserts that this limit exists and also that, for any  $L$ ,  $R_L(d^*)$  is an upper bound to  $R(d^*)$ .

\* This average will exist if, for each choice of  $v_0$ ,  $d(\mathbf{u}; v_0)$  is a measurable function on the Borel field of sets of input sequences over which the source probability measure is defined. For all practical purposes, this includes any function of any conceivable interest.

**Theorem 9.8.1.**

$$\inf_L R_L(d^*) = \lim_{L \rightarrow \infty} R_L(d^*) \quad (9.8.4)$$


---

*Proof.* Let  $L, n$  be arbitrary positive integers and, for a given  $d^*$ , let  $P_L(v_L | u_L)$  and  $P_n(v_n | u_n)$  be the transition probability assignments that achieve  $R_L(d^*)$  and  $R_n(d^*)$ , respectively. Consider a test channel over which  $L + n$  digits are transmitted, using  $P_L$  for the first  $L$  digits and, independently,  $P_n$  for the last  $n$ .

$$P_{L+n}(v_{L+n} | u_{L+n}) = P_L(v_L | u_L)P_n(v_{L+1}, \dots, v_{L+n} | u_{L+1}, \dots, u_{L+n}) \quad (9.8.5)$$

Let  $\mathbf{U}_1$  denote the ensemble of the first  $L$  source letters  $u_1, \dots, u_L$  and  $\mathbf{U}_2$  the ensemble of the succeeding  $n$  letters,  $u_{L+1}, \dots, u_{L+n}$ . Similarly let  $\mathbf{V}_1$  and  $\mathbf{V}_2$  denote the ensembles of the first  $L$  and succeeding  $n$  destination letters respectively. By the same argument as in Theorem 4.2.1, we have

$$\begin{aligned} I(\mathbf{U}_1\mathbf{U}_2; \mathbf{V}_1\mathbf{V}_2) &= H(\mathbf{V}_1\mathbf{V}_2) - H(\mathbf{V}_1\mathbf{V}_2 | \mathbf{U}_1\mathbf{U}_2) \\ H(\mathbf{V}_1\mathbf{V}_2) &\leq H(\mathbf{V}_1) + H(\mathbf{V}_2) \end{aligned}$$

Also, from the channel independence (see (9.8.5)),

$$H(\mathbf{V}_1\mathbf{V}_2 | \mathbf{U}_1\mathbf{U}_2) = H(\mathbf{V}_1 | \mathbf{U}_1) + H(\mathbf{V}_2 | \mathbf{U}_2)$$

It follows that

$$I(\mathbf{U}_1\mathbf{U}_2; \mathbf{V}_1\mathbf{V}_2) \leq I(\mathbf{U}_1; \mathbf{V}_1) + I(\mathbf{U}_2; \mathbf{V}_2) \quad (9.8.6)$$

By the definition of  $P_L$  and  $P_n$  and by the stationarity of the source, the right-hand side of (9.8.6) is equal to  $LR_L(d^*) + nR_n(d^*)$ . Also, from (9.8.1), the total average distortion on the  $L + n$  digits is equal to the total average distortion on the first  $L$  plus the total on the last  $n$ . Thus the average distortion per letter on the  $L + n$  digits is at most  $d^*$ , and  $(L + n)R_{L+n}(d^*)$  is a lower bound to the left-hand side of (9.8.6). Thus

$$(L + n)R_{L+n}(d^*) \leq LR_L(d^*) + nR_n(d^*)$$

The theorem follows from Lemma 4A.2. It is easy to see now that  $R(d^*)$ , as well as  $R_L(d^*)$ , is nonnegative, nonincreasing with  $d^*$ , and convex  $\cup$ .

The converse to the source coding theorem, given by Theorem 9.2.2, applies without change to the sources and distortion measures discussed here. The source coding theorem itself, however, will take a little more work. We first establish a subsidiary result which is useful in its own right.

**Theorem 9.8.2.** Let  $R_1(d^*)$  be the first-order rate-distortion function for a discrete ergodic source with an additive time-invariant distortion measure. For any  $d^*$  such that  $R_1(d^*) < \infty$ , any  $\delta > 0$ , and any  $L$  sufficiently large,

there exists a source code of block length  $L$  with  $M \leq \exp [LR_1(d^*) + L\delta]$  code words for which the average distortion per letter satisfies

$$\bar{d}_L \leq d^* + \delta \quad (9.8.7)$$

Before proving the theorem, we need the following lemma.

LEMMA 9.8.1. Let the output  $\dots u_{-1}, u_0, u_1, \dots$  of a discrete ergodic source be passed through a discrete memoryless channel with output  $\dots v_{-1}, v_0, v_1, \dots$ . Then the joint process  $\dots, u_{-1}v_{-1}, u_0v_0, u_1v_1, \dots$  is ergodic.

We shall omit a formal proof of this lemma, a generalization of which is proved in Wolfowitz (1961), Section 10.3, or Feinstein (1958), p. 99. The idea of the proof is as follows. Let  $\mathbf{u}_l'$  be any particular sequence of  $l$  source letters and  $\mathbf{v}_l'$  any sequence of  $l$  channel outputs. Since the source is ergodic, the relative frequency of occurrence of  $\mathbf{u}_l'$ , over all starting positions, in a source sequence of length  $L > l$  will tend to  $Q_l(\mathbf{u}_l')$ , the probability of  $\mathbf{u}_l'$ , with probability 1 as  $L \rightarrow \infty$ . Since the channel is memoryless, the relative frequency of  $\mathbf{v}_l'$ , over those starting positions where  $\mathbf{u}_l'$  occurs, will tend to  $P_l(\mathbf{v}_l' | \mathbf{u}_l')$  with probability 1. Thus the relative frequency of  $\mathbf{u}_l', \mathbf{v}_l'$  will tend to  $Q_l(\mathbf{u}_l')P_l(\mathbf{v}_l' | \mathbf{u}_l')$ , which is sufficient to establish ergodicity.

*Proof of Theorem.* Let  $P(j | k)$  be the transition probability measure that achieves  $R_1(d^*)$  for the given  $d^*$  and let  $Q(k)$  denote the single-letter source probability assignment. Let

$$\omega(j) = \sum_k Q(k)P(j | k).$$

For any  $L$ , consider an ensemble of codes with  $M = [\exp [LR_1(d^*) + L\delta]]$  code words in which each letter of each code word is chosen independently with the probability assignment  $\omega(j)$ . Within any code of the ensemble, each source sequence  $\mathbf{u}$  is mapped into the code word which minimizes the distortion with  $\mathbf{u}$ . Let

$$P_L(\mathbf{v}_L | \mathbf{u}_L) = \prod_{l=1}^L P(v_l | u_l),$$

let

$$\omega_L(\mathbf{v}_L) = \prod_{l=1}^L \omega(v_l),$$

and define

$$I_1(\mathbf{u}_L; \mathbf{v}_L) = \ln \frac{P_L(\mathbf{v}_L | \mathbf{u}_L)}{\omega_L(\mathbf{v}_L)} \quad (9.8.8)$$

Notice that  $I_1(\mathbf{u}_L; \mathbf{v}_L)$  is not the mutual information between  $\mathbf{u}_L$  and  $\mathbf{v}_L$  using the source on the memoryless test channel with transition probabilities given by  $P(j|k)$ , since  $\omega_L(\mathbf{v}_L)$  is not the probability of  $\mathbf{v}_L$  at the output of the test channel.

Let  $\hat{d}$  and  $\hat{R}$  be arbitrary, and define

$$A = \{\mathbf{u}, \mathbf{v}_L : \text{either } I_1(\mathbf{u}_L; \mathbf{v}_L) > L\hat{R} \text{ or } D(\mathbf{u}; \mathbf{v}_L) > L\hat{d}\} \quad (9.8.9)$$

Let  $P_t(A)$  be the probability of  $A$  on the test-channel ensemble. Notice that this ensemble involves the infinite-sequence source output, but only a finite-length destination sequence  $\mathbf{v}_L$ . Let  $P_c(D > L\hat{d})$  be the probability, over the ensemble of codes and the ensemble of source outputs, that the distortion (between  $\mathbf{u}$  and the code word into which it is mapped) exceeds  $L\hat{d}$ . Then, from the proof of Lemma 9.3.1, it follows that

$$P_c(D > L\hat{d}) \leq P_t(A) + \exp(-Me^{-L\hat{R}}) \quad (9.8.10)$$

The reader should review that proof at this time. The significant difference is that  $\omega_L(\mathbf{v})$  in the lemma was both the output probability measure from the test channel and the probability measure with which code words were chosen. When we observe that only the latter property was used in the proof, it becomes apparent that (9.8.10) applies here.

Now let  $\hat{R} = R_1(d^*) + \delta/2$  and  $\hat{d} = d^* + \delta/2$ . As in Theorem 9.3.1, the average distortion per letter,  $d_L$ , over the ensemble of codes satisfies

$$\bar{d}_L \leq d^* + \frac{\delta}{2} + P_c\left[D > L\left(d^* + \frac{\delta}{2}\right)\right] \sup_{\mathbf{u}, v_0} d(\mathbf{u}, v_0) \quad (9.8.11)$$

Since  $d(\mathbf{u}, v_0)$  is by assumption bounded, the theorem will be proved if we can show that  $P_c[D > L(d^* + \delta/2)]$  approaches 0 as  $L \rightarrow \infty$ . The final term in (9.8.10) goes to 0 with increasing  $L$  as in Theorem 9.3.1. The first term is upper bounded by

$$\begin{aligned} P_t(A) &\leq \Pr\{I_1(\mathbf{u}_L; \mathbf{v}_L) > L[R_1(d^*) + \delta/2]\} \\ &\quad + \Pr\{D(\mathbf{u}; \mathbf{v}_L) > L[d^* + \delta/2]\} \end{aligned} \quad (9.8.12)$$

$$\Pr\{I_1(\mathbf{u}_L; \mathbf{v}_L) > L[R_1(d^*) + \delta/2]\} = \Pr\left[\frac{1}{L} \sum_{l=1}^L \ln \frac{P(v_l | u_l)}{\omega(v_l)} > R_1(d^*) + \delta/2\right] \quad (9.8.13)$$

On the other hand, since the joint  $u, v$  process is ergodic, (3.5.13) yields, with probability 1,

$$\begin{aligned} \lim_{L \rightarrow \infty} \frac{1}{L} \sum_{l=1}^L \ln \frac{P(v_l | u_l)}{\omega(v_l)} &= \overline{\ln \frac{P(v_l | u_l)}{\omega(v_l)}} \\ &= R_1(d^*) \end{aligned} \quad (9.8.14)$$

Thus the probability in (9.8.13) approaches 0 with increasing  $L$ . The same argument applies to the final term in (9.8.12), thus completing the proof. |

One of the most interesting aspects of the preceding theorem is that both  $R_1(d^*)$  and the ensemble of codes used in the proof depend only on the distortion measure and the single-letter probabilities of the source. This gives a strong indication that, if we construct a good source code using only a knowledge of the single-letter source probabilities, we may have a fair degree of confidence that any memory in the source will only reduce the average distortion from that expected in the memoryless case. Unfortunately, this statement can not be made precise, and it is not hard to find examples of codes for which the average distortion on a source with memory exceeds that on the memoryless source with the same single-letter probabilities.

We have shown that, for a discrete ergodic source, we can find source codes with rates arbitrarily close to  $R_1(d^*)$  with average distortions per letter arbitrarily close to  $d^*$ . We now want to establish a stronger result where  $R_1(d^*)$  can be replaced by  $R(d^*)$ . Our strategy first will be to pick an  $n$  large enough so that  $R_n(d^*)$  is close to  $R(d^*)$ . We shall then consider sequences of  $n$  source letters as single letters from a “super source” which produces one letter each  $n$  time units. We shall then apply the previous theorem to this super source, and after appropriate normalizing by  $n$ , we shall obtain the desired result. There is one minor difficulty in this procedure: the super source need not be ergodic. To handle this difficulty, we shall first show that the super source can be separated into, at most,  $n$  “ergodic modes,” and then apply the previous theorem to each mode separately.

Suppose now that a discrete ergodic source has alphabet size  $K$  and we consider the  $n$ th order super source of alphabet size  $K^n$  where each letter of the super source is a sequence of  $n$  letters from the original source. Corresponding to each sequence  $\mathbf{u} = (\dots, u_{-1}, u_0, u_1, \dots)$  from the original source there will be a sequence  $\mathbf{u}' = (\dots, u'_{-1}, u'_0, u'_1, \dots)$  from the super source where  $u'_1 = (u_1, u_2, \dots, u_n)$ ,  $u'_2 = (u_{n+1}, \dots, u_{2n})$ , and so on. We shall denote this correspondence between sequences from the original source and the super source by  $\mathbf{u} \leftrightarrow \mathbf{u}'$ . Since a shift of  $n$  time units on the original source corresponds to a shift of one time unit for the super source, we have  $T^{ln}\mathbf{u} \leftrightarrow T^l\mathbf{u}'$  for  $\mathbf{u} \leftrightarrow \mathbf{u}'$ . Likewise, any set  $S$  of sequences from the original source will correspond to a set  $S'$  from the super source, and  $T^{ln}S \leftrightarrow T^lS'$ . We recall, from Section 3.5, that an invariant set  $S$  is a set for which  $TS = S$  and that a source is ergodic iff every measurable invariant set has probability either 1 or 0.

Now suppose that  $S'_0$  is an invariant set of sequences from the super source and that its probability,  $Q'(S'_0)$  is greater than 0. The corresponding set  $S_0 \leftrightarrow S'_0$  for the original source has probability  $Q(S_0) = Q'(S'_0)$ , and

satisfies  $T^n S_0 = S_0$ . Define the sets  $S_i$ ,  $1 \leq i \leq n - 1$ , by

$$S_i = T^i S_0 \quad (9.8.15)$$

Since the source is stationary,  $Q(S_i) = Q(S_0)$  for  $1 \leq i \leq n - 1$ . Now, let the set  $A$  be the union of  $S_0, S_1, \dots, S_{n-1}$ . We then have

$$TA = T \left( \bigcup_{i=0}^{n-1} S_i \right) = \bigcup_{i=0}^{n-1} TS_i = S_1 \cup S_2 \cup \dots \cup TS_{n-1} \quad (9.8.16)$$

Since  $TS_{n-1} = TT^{n-1}S_0 = S_0$ , this reduces to  $TA = A$ . Since the source is ergodic and  $Q(A) > 0$ , we must have  $Q(A) = 1$ , and thus  $Q(S_0) = Q'(S'_0) \geq 1/n$ . Thus we have shown that every measurable invariant set for the super source either has probability 0 or probability at least  $1/n$ .

Next, suppose that  $S'_0$  is an invariant set of sequences from the super source and that  $B'$  is an invariant subset of  $S'_0$  with  $0 < Q'(B') < Q'(S'_0)$ . Let  $S'_0 - B'$  be the sequences in  $S'_0$  but not  $B'$ , and observe that  $T(S'_0 - B') = TS'_0 - TB' = S'_0 - B'$ , so that  $S'_0 - B'$  is also an invariant subset of  $S'_0$ . It follows from the previous result, that  $1/n \leq Q'(B') \leq Q'(S'_0) - (1/n)$ . If we now let  $B'$  play the role of  $S'_0$  and repeat the above argument, we see that eventually we must arrive at an  $S'_0$  which has no invariant subset  $B'$  with  $0 < Q'(B') < Q'(S'_0)$ . We call such an  $S'_0$  an *ergodic mode* of the super source. A source that generates super letters according to the conditional probability measure of  $Q'$  conditional on  $S'_0$  is clearly an ergodic source since, conditional on  $S'_0$ , each invariant subset of  $S'_0$  has probability either 0 or 1.

Now let  $S_0 \leftrightarrow S'_0$ , let  $S_i = T^i S_0$ , and let  $S_i \leftrightarrow S'_i$ . Each  $S'_i$  must also constitute an ergodic mode of the super source since, if  $B'_i$  is a subset of  $S'_i$  and  $TB'_i = B'_i$ , then for the corresponding set  $B_i$  on the original source, we have  $T^n B_i = B_i$ . Letting  $B_0 = T^{-i} B_i$ ,  $B_0$  is a subset of  $S_0$  and  $T^n B_0 = B_0$ . Thus  $B_0$  is a subset of  $S'_0$  and  $TB'_0 = B'_0$ . Since  $Q'(B'_i) = Q'(B'_0)$ , we must have  $Q'(B'_i) = 0$  or  $Q'(B'_i) = Q'(S'_i)$ . Finally, let us consider the intersection  $S'_i \cap S'_j$ . This intersection is an invariant set and is a subset both of  $S'_i$  and of  $S'_j$ . Thus either  $Q'(S'_i \cap S'_j) = 0$  or  $Q'(S'_i \cap S'_j) = Q'(S'_i)$ . In the latter case,  $S'_i$  and  $S'_j$  are the same sets (except perhaps for a difference of zero probability). It is easy to see that if  $S'_i$  and  $S'_j$  are the same, then also  $S'_{i+k}$  and  $S'_{j+k}$  are the same, where  $i + k$  and  $j + k$  are taken modulo  $n$ . It follows easily from this that the number of different ergodic modes, say  $n'$ , is a factor of  $n$ , and that the ergodic modes can be taken as  $S'_0, S'_1, \dots, S'_{n'-1}$ . In this case, each ergodic mode has probability  $1/n'$ . In the case of greatest physical interest,  $n' = 1$ , and the super source is ergodic to start with.

The following lemma summarizes our results.

**LEMMA 9.8.2.** Consider the  $n$ th-order super source whose letters are sequences of  $n$  letters from a discrete ergodic source. The set of sequences from the super source can be separated into  $n'$  ergodic modes, each of

probability  $1/n'$ , where  $n'$  divides  $n$ . The modes are disjoint except perhaps for sets of zero probability. The sets  $S_0, S_1, \dots, S_{n'-1}$  of original source sequences corresponding to these ergodic modes can be related by  $T(S_i) = S_{i+1}$ ,  $0 \leq i \leq n' - 2$ , and  $T(S_{n'-1}) = S_0$ .

---

As an example, consider a binary source for which the output consists of pairs of identical digits. With probability  $\frac{1}{2}$ , each digit in an even position is an independent equiprobable selection from the alphabet and each odd-numbered digit is the same as the preceding even-numbered digit. Likewise, with probability  $\frac{1}{2}$ , each odd-numbered digit is the independent and equiprobable selection, and each even-numbered digit is the same as the preceding odd-numbered digit. This is an ergodic source but the second-order super source has two modes. In one mode, the super source is memoryless and produces 00 with probability  $\frac{1}{2}$  and 11 with probability  $\frac{1}{2}$ . In the second mode, all four pairs of digits are equally likely, but the last digit of one pair is the same as the first digit of the next pair. It is obvious, in this example, that the output from one mode (in original source digits) is statistically identical to that of the other shifted in time by one digit.

We have seen that, in general, the  $n$ th-order super source corresponding to an ergodic source can be divided into  $n'$  ergodic modes, where  $n'$  divides  $n$ . It will be more convenient in what follows to regard this as  $n$  ergodic modes where only  $n'$  of them are different. We define the  $i$ th-phase super source,  $0 \leq i \leq n - 1$ , as the source that produces the sequences in  $S'_i$  according to the original probability measure on sequences of super letters conditional on the occurrence of  $S'_i$ . For  $n' < n$ , then, the  $i$ th phase supersource, the  $(i + n')$ th phase supersource, the  $(i + 2n')$ th phase supersource, and so on, are all identical. The original supersource is then modelled as this collection of  $n$  ergodic supersources, each used, over the entire infinite sequence, with probability  $1/n$ .

We now use this model in the construction of source codes for an ergodic source. For a given  $n$ , let  $R_n(d^*)$  be the  $n$ th order rate-distortion function, and for a given  $d^*$  with  $R_n(d^*) < \infty$ , let  $P_n(\mathbf{v}_n | \mathbf{u}_n)$  denote the set of transition probabilities which achieves  $R_n(d^*)$ . In terms of the  $n$ th-order super source,  $\mathbf{u}_n$  is a single letter  $u'_1$ , and we can similarly regard  $\mathbf{v}_n$  as a single letter,  $v'_1$ , in a super alphabet of sequences of  $n$  destination letters. Letting  $U'$  be the ensemble of single super letters and  $V'$  be the ensemble of single super destination letters determined by  $P_n$ , we have

$$R_n(d^*) = \frac{1}{n} I(U'; V') \quad (9.8.17)$$

Similarly

$$d^* \geq \frac{1}{n} \overline{D(\mathbf{u}; \mathbf{v}_n)} = \frac{1}{n} \overline{d'(\mathbf{u}'; v'_1)} \quad (9.8.18)$$

where  $d'$  is defined by  $D(\mathbf{u}; \mathbf{v}_n) = d'(\mathbf{u}'; v_1')$  and the inequality is to account for the possibility  $R_n(d^*) = 0$ .

Now let  $I(U'; V' | i)$  be the average mutual information between a super letter of the  $i$ th-phase super source,  $0 \leq i \leq n - 1$ , and a letter of the super destination alphabet using the transition probabilities  $P_n$ . Since the average mutual information on a channel is a convex  $\cap$  function of the input probabilities, we have

$$I(U'; V') \geq \frac{1}{n} \sum_{i=0}^{n-1} I(U'; V' | i) \quad (9.8.19)$$

Similarly, the average value of  $d'$  is a linear function of the probability measure on  $u'$ , so that

$$\overline{d'(\mathbf{u}', v_1')} = \frac{1}{n} \sum_{i=0}^{n-1} \overline{d'_i} \quad (9.8.20)$$

where  $\overline{d'_i}$  is the average value of  $d'(\mathbf{u}'; v_1')$  for the  $i$ th-phase super source.

We now observe that  $I(U'; V' | i)$  is an upper bound to the first-order rate-distortion function of the  $i$ th-phase super source evaluated at an average distortion  $\overline{d'_i}$ . Since the  $i$ th-phase super source is ergodic, Theorem 9.8.2 applies, and for any  $\delta > 0$ , and any sufficiently large  $L$ , there is a set of  $M_i \leq \exp [LI(U'; V' | i) + L\delta]$  code words of length  $L$  (in super letters) for which the average distortion per super letter for that source is, at most,  $\overline{d'_i} + \delta$ . For a given  $\delta > 0$ , let  $L$  be large enough so that such a code can be selected for each of the  $n$  phases and consider such a set of  $n$  codes.

It is convenient to think of the code words in these codes not as sequences of  $L$  super letters but as sequences of  $Ln$  letters from the original destination alphabet. We refer to the  $i$ th such code, as chosen above for the  $i$ th-phase super source, as the  $i$ th little code. We now use these  $n$  little codes to construct a new set of  $n$  big codes of block length  $Ln^2 + n$ , as shown in Figure 9.8.1. The  $i$ th big code,  $0 \leq i \leq n - 1$ , is designed to encode the output of the  $i$ th phase supersource. As shown in Figure 9.8.1, for each  $l$ ,  $0 \leq l \leq n - 1$ , the set of letters in positions  $l[nL + 1] + 1$  to  $l[nL + 1] + Ln$  is an arbitrary code word from the  $(i + l)$ th little code, where  $i + l$  is taken modulo  $n$ .

Fixed letters

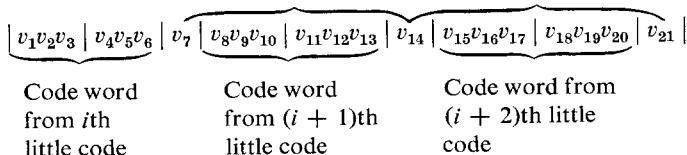


Figure 9.8.1. Construction of big code from little code;  $n = 3$ ;  
 $L = 2$ .

The letters in positions  $Ln + 1, 2[Ln + 1], \dots, n[Ln + 1]$  are fixed letters from the destination alphabet. The total number of code words in the  $i$ th big code is thus

$$\prod_{i=0}^{n-1} M_i,$$

where  $M_i$  is the number of code words in the  $i$ th little code. Now recall from Lemma 9.8.2 that, if  $S_i$  is the set of sequences of letters from the original source alphabet produced by the  $i$ th-phase super source, then  $TS_i = S_{i+1}$ . It follows from this that the probability measure for the  $i$ th phase super source on letters in positions  $l(Ln + 1) + 1$  to  $l(Ln + 1) + Ln$  is the same as that for the  $(i + l)$ th-phase super source in positions 1 to  $Ln$ . Thus the average distortion per letter between the  $i$ th-phase super source and the  $i$ th big code, for positions  $(Ln + 1) + 1$  to  $(Ln + 1) + Ln$ , is  $1/n$  times the average distortion per super letter between the  $(i + l)$ th-phase super source and the  $(i + l)$ th little code (where  $i + l$  is taken modulo  $n$ ). We have chosen the little codes so that this average distortion per letter is upper bounded by  $(1/n)[\overline{d'_{i+l}} + \delta]$ . Observing that the distortion between each of the  $n$  fixed positions in the code and the source is upper bounded by  $\sup d(\mathbf{u}; v_0)$ , the average distortion per letter between the  $i$ th super source and the  $i$ th big code is upper bounded by

$$\frac{n \sup d(\mathbf{u}; v_0) + \sum_{i=0}^{n-1} L(\overline{d'_{i+l}} + \delta)}{Ln^2 + n} \quad (9.8.21)$$

This is further upper bounded by reducing the denominator to  $Ln^2$  and by using (9.8.18) and (9.8.20), resulting in

$$\frac{\sup d(\mathbf{u}; v_0)}{Ln} + d^* + \frac{\delta}{n} \quad (9.8.22)$$

For large enough  $L$ , the average distortion per letter between the  $i$ th-phase super source and the  $i$ th big code is thus upper bounded by  $d^* + \delta$ , where  $\delta > 0$  is arbitrary.

If all the code words of these  $n$  big codes are combined into one code, then no matter which ergodic super source is in effect, the average distortion per letter will be at most  $d^* + \delta$ . The total number of code words is

$$\begin{aligned} M &= n \prod_{i=0}^{n-1} M_i \leq n \exp \left\{ \sum_{i=0}^{n-1} [LI(U'; V' | i) + \delta] \right\} \\ &\leq n \exp [LnI(U'; V') + Ln \delta] \\ &= n \exp [Ln^2 R_n(d^*) + Ln \delta] \\ &\leq \exp \left\{ (Ln^2 + n) \left[ R_n(d^*) + \frac{\delta}{n} + \frac{\ln n}{Ln^2 + n} \right] \right\} \end{aligned} \quad (9.8.23)$$

We have thus shown that, for any  $n > 1$ , any  $\delta > 0$ , and large enough  $L$ , we can find codes of length  $Ln^2 + n$  with  $M \leq \exp \{(Ln^2 + n)[R_n(d^*) + \delta]\}$  and average distortion per letter less than or equal to  $d^* + \delta$ . Furthermore, the restriction that the block length must be of the form  $Ln^2 + n$  is clearly unnecessary, since for any long enough block length  $L'$ , we can find the largest  $L$  such that  $Ln^2 + n \leq L'$ , find a code for that  $L$  satisfying (9.8.22) and (9.8.23), and then add a sequence of at most  $n^2 - 1$  fixed digits to the end of each code word. The average distortion per letter is thus increased by at most  $\sup d(\mathbf{u}; v_0)/L$ , which for large  $L$  is negligible. Finally, since  $n$  can be chosen large enough so that  $R_n(d^*)$  is arbitrarily close to  $R(d^*)$ , we have proved the following fundamental theorem.

**Theorem 9.8.3.** Let  $R(d^*)$  be the rate-distortion function of a discrete ergodic source with an additive time-invariant distortion measure. For any  $d^*$  with  $R(d^*) < \infty$ , any  $\delta > 0$ , and any sufficiently large  $L$ , there exists a source code of block length  $L$  with  $M \leq \exp [LR(d^*) + L\delta]$  code words with an average distortion per letter of at most  $d^* + \delta$ .

---

If he wishes, the ambitious and patient reader can combine the techniques of proving this theorem with the results in Sections 9.3 and 9.6 to avoid both the restriction that  $d(\mathbf{u}; v_0)$  be bounded and the restriction that the source be discrete.

### Summary and Conclusions

In this chapter, we considered the problem of reconstructing the output of a source at a destination subject to a fidelity criterion. The source was described in terms of the set of possible outputs from the source and a probability measure on those outputs. As a criterion of fidelity, we took the average distortion per letter or per unit time between source and destination. From the standpoint of the theory, the distortion measure is a nonnegative function of the source output and the reconstruction. From the standpoint of any application, of course, the distortion measure must be chosen to reflect, in some sense, the cost to the user of having any particular source output reconstructed in any particular way.

We started with discrete memoryless sources with single-letter distortion measures and proceeded to generalize our results to more and more general cases. In each case, we first defined a rate-distortion function,  $R(d^*)$ , and then gave this function significance by proving a source coding theorem and its converse. What the source coding theorem says, in essence, is that for a given  $d^*$ , the source output can be encoded into  $R(d^*)/\ln 2$  binary digits per source digit [or per unit time, depending on the units of  $R(d^*)$ ] and that these binary digits can be converted into destination letters in such a way that the average

distortion per letter (or per unit time) is arbitrarily close to  $d^*$ . The same result holds true if the binary digits are encoded and transmitted over a noisy channel of capacity greater than  $R(d^*)$  where the capacity is in nats per source letter (or nats per unit time). The converse states that, if the source output is transmitted over a channel of capacity less than  $R(d^*)$ , then independent of the processing at source and destination, the average distortion per letter (or per unit time) must exceed  $d^*$ .

The reader must have noticed the many analogies between the source coding results developed here and the theory of coding for noisy channels, but it may be helpful to stress some of the differences here. The noisy-channel coding theorem related the achievable probability of decoding error  $P_e$  to the code block length and the code rate  $R$ . We found that, for fixed  $R$  less than capacity,  $P_e$  decreased exponentially with increasing block length. For source coding, the equivalent parameters of interest are the average distortion per letter  $\bar{d}$ , the code block length  $L$  and the code rate  $R$ . Here, for fixed  $R$ ,  $\bar{d}$  decreases toward the  $d^*$  given by  $R = R(d^*)$  as  $L$  increases. The convergence of  $\bar{d}$  to  $d^*$  is at least as fast as  $\sqrt{(\ln L)/L}$  and apparently no faster than as  $1/L$  [see Pilc (1966)]. Thus, in source coding, we must work very hard (in terms of increased block length) to achieve a very small fractional decrease in  $\bar{d}$ , whereas for noisy channels, a modest increase of block length can yield a striking decrease in error probability.

As an example (perhaps atypical) of just how little can be gained by sophisticated source coding, we can consider a discrete-time Gaussian source with a square-difference distortion measure. Goblick (1967) and Max (1960) have considered the average distortion that can be achieved by using a quantizer (that is, a source code of unit block length). They have found that, if the quantized letters are noiselessly encoded (by Huffman coding), then for small values of distortion the average distortion is only about  $\frac{1}{4}$  db above the minimum predicted by the  $R(d^*)$  curve. If the Huffman coding is also omitted, the loss is still below 1 db.

The other major difference between channel coding and source coding appears to lie in the difficulty of obtaining reasonable probabilistic models and meaningful distortion measures for sources of practical interest. Because of this difficulty, it is not at all clear whether the theoretical approach here will ever be a useful tool in problems such as speech digitization or television bandwidth compression.

### Historical Notes and References

Most of the results in this chapter are due to Shannon (1959). The proof of Lemma 9.3.1 and Theorem 9.3.1 employs both the techniques of Shannon's original proof and later techniques developed by Goblick and Stiglitz. It is simpler than the original proof and gives better results on the convergence

to  $R(d^*)$  for codes as the block length gets longer. The extension of the theory to the case where  $d(u;v)$  takes on infinite values and Theorem 9.3.2 are from Pinkston (1967). Theorem 9.4.1 is original here, although the lower bound to  $R_0(\rho, \mathbf{P})$  in (9.4.10) was derived by Shannon in the special case of a square-difference distortion measure. The results on the converse to the noisy-channel coding theorem in Section 9.5 are from Pinkston (1967). The calculation of the rate-distortion function for a discrete-time Gaussian source is due to Shannon (1948), and the rate-distortion function for a Gaussian random process is due to Kolmogoroff (1956). Evidently, the coding theorem for a Gaussian random process source has not previously appeared in the literature. The results of Section 9.8 are new although a similar theorem not requiring a discrete source but requiring a slightly stronger condition than ergodicity has been given by Goblick (1967).

## EXERCISES AND PROBLEMS\*

### CHAPTER 2

- 2.1.** Three events  $E_1$ ,  $E_2$ , and  $E_3$ , defined on the same space, have probabilities  $P(E_1) = P(E_2) = P(E_3) = \frac{1}{4}$ . Let  $E_0$  be the event that one or more of the events  $E_1$ ,  $E_2$ ,  $E_3$  occurs.

(a) Find  $P(E_0)$  when:

- (1) The events  $E_1$ ,  $E_2$ ,  $E_3$  are disjoint.
- (2) The events  $E_1$ ,  $E_2$ ,  $E_3$  are statistically independent.
- (3) The events  $E_1$ ,  $E_2$ ,  $E_3$  are in fact three names for the same event.

(b) Find the maximum values that  $P(E_0)$  can assume when:

- (1) Nothing is known about the independence or disjointness of  $E_1$ ,  $E_2$ ,  $E_3$ .
- (2) It is known that  $E_1$ ,  $E_2$ ,  $E_3$  are pairwise independent, i.e., that the probability of realizing both  $E_i$  and  $E_j$  is  $P(E_i)P(E_j)$ ,  $1 < i \neq j < 3$ . but nothing is known about the probability of realizing all three events together.

*Hint:* Use Venn diagrams.

- 2.2.** A dishonest gambler has a loaded die which turns up the number 1 with probability  $\frac{2}{3}$  and the numbers 2 to 6 with probability  $\frac{1}{15}$  each. Unfortunately, he left his loaded die in a box with two honest dice and could not tell them apart. He picks one die (at random) from the box, rolls it once, and the number 1 appears. Conditional on this result, what is the probability that he picked up the loaded die? He now rolls the die once more and it comes up 1 again. What is the possibility after this second rolling that he has picked the loaded die?

- 2.3.** Let  $x$  and  $y$  be discrete random variables.

- (a) Prove that the expectation of the sum of  $x$  and  $y$ ,  $\overline{x+y}$ , is equal to the sum of the expectations,  $\bar{x} + \bar{y}$ .
- (b) Prove that if  $x$  and  $y$  are statistically independent, then  $x$  and  $y$  are also uncorrelated (by definition,  $x$  and  $y$  are uncorrelated if  $\overline{xy} = \bar{x}\bar{y}$ ). Find

\* Mimeographed solutions, by chapter, are available to instructors by writing to the author.

an example in which  $x$  and  $y$  are statistically dependent and uncorrelated and another example where  $x$  and  $y$  are statistically dependent but correlated (i.e.,  $\bar{xy} \neq \bar{x}\bar{y}$ ).

- (c) Prove that if  $x$  and  $y$  are statistically independent, then the variance of the sum is equal to the sum of the variances. Is this relationship valid if  $x$  and  $y$  are uncorrelated but not statistically independent?
- 2.4.** (a) One form of the Chebyshev inequality states the following: for any random variable  $x$  that takes on only non-negative values, and for any  $\delta > 0$ ,

$$\Pr[x \geq \delta] \leq \frac{\bar{x}}{\delta}$$

Prove this inequality and for any given  $\delta > 0$  exhibit a random variable that satisfies the relationship with equality. *Hint:* Write out  $\bar{x}$  as a summation and then restrict the range of summation.

- (b) Let  $y$  be any random variable with mean  $\bar{y}$  and variance  $\sigma_y^2$ . By letting the random variable  $x$  above be  $(y - \bar{y})^2$ , show that for any  $\epsilon > 0$ ,

$$\Pr[|y - \bar{y}| \geq \epsilon] \leq \frac{\sigma_y^2}{\epsilon^2}$$

- (c) Let  $z_1, z_2, \dots, z_N$  be a sequence of statistically independent, identically distributed random variables with mean  $\bar{z}$  and variance  $\sigma_z^2$ . Let  $y_N$  be the sample mean,

$$y_N = \frac{1}{N} \sum_{n=1}^N z_n$$

Show that for any  $\epsilon > 0$ ,

$$\Pr[|y_N - \bar{y}_N| \geq \epsilon] \leq \frac{\sigma_z^2}{N\epsilon^2} \quad (\text{i})$$

*Hint:* First find the variance of  $\sum z_n$  using Problem 2.3c.

Find the limit as  $N \rightarrow \infty$  of  $\Pr[|y_N - \bar{y}_N| \geq \epsilon]$  for any fixed  $\epsilon > 0$  (this is known as the law of large numbers).

- (d) Let the random variables  $z_1, \dots, z_N$  of part (c) correspond to  $N$  independent trials of an experiment. Let  $E$  be a given event and let  $z_n = 1$  if  $E$  occurs on the  $n$ th trial and  $z_n = 0$  otherwise. State in words what (i) says in this context and evaluate  $\bar{y}_N$  and  $\sigma_z^2$ . Find an exact expression for  $\Pr[|y_N - \bar{y}_N| \geq \epsilon]$  in this case.
- 2.5.** A source produces statistically independent, equally probable letters from an alphabet  $(a_1, a_2)$  at a rate of 1 letter each 3 seconds. These letters are transmitted over a binary symmetric channel which is used once each second by encoding the source letter  $a_1$  into the code word **000** and encoding  $a_2$  into the code word **111**. If, in the corresponding 3-second interval at the channel output, any of the sequences **000**, **001**, **010**, **100** is received,  $a_1$  is decoded; otherwise  $a_2$  is decoded. Let  $\epsilon < \frac{1}{2}$  be the channel crossover probability (see Fig. 1.3.1).

- (a) For each possible received 3-digit sequence in the interval corresponding to a given source letter, find the probability that  $a_1$  came out of the source given that received sequence.
- (b) Using part (a), show that the above decoding rule minimizes the probability of an incorrect decision.

- (c) Find the probability of an incorrect decision [using part (a) is not the easy way here].
- (d) Suppose the source is slowed down to produce one letter each  $2n + 1$  seconds,  $a_1$  being encoded into  $2n + 1$  0's and  $a_2$  into  $2n + 1$  1's. What decision rule minimizes the probability of an incorrect decision at the decoder? Find this probability of incorrect decision in the limit as  $n \rightarrow \infty$ .

*Hint:* Use the law of large numbers.

- 2.6.** In a female population  $X$ , consisting of  $\frac{1}{4}$  blondes,  $\frac{1}{2}$  brunettes, and  $\frac{1}{4}$  redheads, blondes are always on time for engagements, redheads are always late, and each brunette always flips an unbiased coin for each engagement to decide whether to be prompt or tardy.
- How much information is given by the statement “ $x$ , a member of  $X$ , arrived on time” about each of the following propositions:
    - $x$  is a blonde,
    - $x$  is a brunette,
    - $x$  is a redhead.
  - How much information is given by the statement “ $x$ , a member of  $X$ , arrived on time for three engagements in a row” about the proposition “ $x$  is a brunette”?
- 2.7.** A set of eight equiprobable messages are encoded into the following set of eight code words for transmission over a binary symmetric channel with crossover probability  $\epsilon$  (see Figure 2.2.1):

$$\mathbf{x}_1 = \mathbf{0000} \quad \mathbf{x}_5 = \mathbf{1001}$$

$$\mathbf{x}_2 = \mathbf{0011} \quad \mathbf{x}_6 = \mathbf{1010}$$

$$\mathbf{x}_3 = \mathbf{0101} \quad \mathbf{x}_7 = \mathbf{1100}$$

$$\mathbf{x}_4 = \mathbf{0110} \quad \mathbf{x}_8 = \mathbf{1111}$$

If the sequence of digits  $\mathbf{y} = \mathbf{0000}$  is received at the channel output, determine:

- The amount of information provided about  $\mathbf{x}_1$  by the first digit received.
- The additional (conditional) amounts of information about  $\mathbf{x}_1$  provided by the second received digit, the third digit, and the fourth digit.

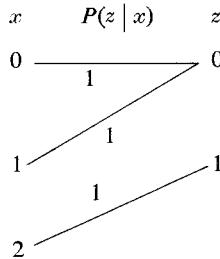
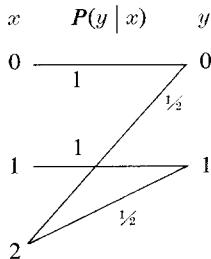
- 2.8.** Consider an ensemble of sequences of  $N$  binary digits,  $x_1, x_2, \dots, x_N$ . Each sequence containing an even number of 1's has probability  $2^{-N+1}$  and each sequence with an odd number of 1's has probability zero. Find the average mutual informations

$$I(X_2; X_1), I(X_3; X_2 | X_1), \dots, I(X_N; X_{N-1} | X_1 \cdots X_{N-2}).$$

Check your result for  $N = 3$ .

- 2.9.** A source  $X$  produces letters from a three-symbol alphabet with the probability assignment  $P_X(0) = \frac{1}{4}$ ,  $P_X(1) = \frac{1}{4}$ ,  $P_X(2) = \frac{1}{2}$ . Each source letter  $x$  is directly transmitted through two channels simultaneously with outputs

$y$  and  $z$  and the transition probabilities indicated below:

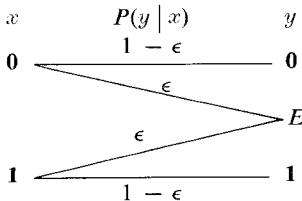


(Note that this could be considered as a single channel with output  $yz$ ). Calculate  $H(X)$ ,  $H(Y)$ ,  $H(Z)$ ,  $H(YZ)$ ,  $I(X;Y)$ ,  $I(X;Z)$ ,  $I(X;Y|Z)$ ,  $I(X;YZ)$ . Interpret the mutual information expressions.

- 2.10.** The binary erasure channel shown below is a “noisy” channel with a particularly simple type of noise in that transmitted digits may be “erased” but they can never be erroneously received.

- (a) Let  $P_X(0) = p = 1 - P_X(1)$ . Calculate  $I(X;Y)$  in terms of  $p$  and find the value of  $p$  that maximizes  $I(X;Y)$ . For this maximizing distribution, calculate  $I(x;y)$  at all points of the  $XY$  ensemble in addition to finding  $I(X;Y)$ .
- (b) Suppose it is desired to transmit a stream of statistically independent and equiprobable binary digits over the binary erasure channel. Suppose further that there is a noiseless feedback channel from the receiver so that the sender knows each digit as it is received. Consider the following stratagem for sending the stream with absolute reliability: Send the digits in order over the channel, repeating each digit, should an erasure occur, until it is received correctly. Calculate the average number of digits transmitted per use of the channel.

*Note:* It is intuitively clear that the communication scheme envisaged here is making optimum use of the channel. Note the fundamental role played by  $I(X;Y)$ .



- 2.11.** Let  $x = a_1$  denote the event that the ball in a roulette game comes to rest in a “red” compartment, and  $x = a_2$  similarly denote a ball coming to rest in “black.” Assume that the house has no take, i.e., that a one-dollar bet on red or black, when successful, returns one additional dollar. We suppose that  $P_X(a_1) = P_X(a_2) = \frac{1}{2}$ .

The croupier of the roulette table has developed a scheme to defraud the house. After years of patient study, he has learned to partially predict the color that will turn up by observing the path of the ball up to the last instant that bets may be placed. By communicating this knowledge to an accomplice, the croupier expects to use his inside knowledge to gain a tidy sum for his retirement.

Let  $y$  denote the croupier's signal; a cough,  $y = b_1$ , indicates a red prediction and a blink,  $y = b_2$ , indicates a black prediction. Assume  $P_{X/Y}(a_1/b_1) = P_{X/Y}(a_2/b_2) = \frac{3}{4}$  and assume that successive spins are independent.

- (a) Calculate  $I(X; Y)$ .
- (b) The accomplice has some initial capital  $C_0$ . He has decided to wager a fixed fraction  $1 - q$  of his total capital on the predicted color on each successive wager and a fraction  $q$  on the other color (note that this is equivalent to wagering a fraction  $1 - 2q$  on the predicted color and withholding  $2q$ ). After  $N$  trials, show that the accomplice's capital is the random variable

$$C_N = C_0 \prod_{n=1}^N [2(1 - q)]^{z_n} [2q]^{1-z_n}$$

where  $z_n = 1$  if the prediction is correct and 0 otherwise. Define the rate of growth as

$$E_N = \frac{1}{N} \log_2 \frac{C_N}{C_0}; \quad C_N = C_0 2^{N E_N}$$

Find the values of  $q$  that maximize the expectations  $\bar{C}_N$  and  $\bar{E}_N$ . Compare the maximum expected growth  $\bar{E}_N$  with  $I(X; Y)$ .

- (c) If you were the accomplice, which value of  $q$  would you use and why?  
*Hint:* Does the law of large numbers apply to either  $E_N$  or  $C_N$  as  $N \rightarrow \infty$ ?

See J. L. Kelly (1956) for a detailed treatment of this class of problems.

- 2.12.** An ensemble  $X$  has the non-negative integers as its sample space. Find the probability assignment  $P_X(n)$ ,  $n = 0, 1, \dots$ , that maximizes  $H(X)$  subject to the constraint that the mean value of  $X$ ,

$$\sum_{n=0}^{\infty} n P_X(n),$$

is a fixed value  $A$ . Evaluate the resulting  $H(X)$ .

- 2.13.** The weatherman's record in a given city is given in the table below, the numbers indicating the relative frequency of the indicated event.

| Prediction | Actual |         |
|------------|--------|---------|
|            | Rain   | No rain |
| Rain       | 1/8    | 3/16    |
| No rain    | 1/16   | 10/16   |

A clever student notices that the weatherman is right only 12/16 of the time but could be right 13/16 of the time by always predicting no rain. The student explains the situation and applies for the weatherman's job, but the weatherman's boss, who is an information theorist, turns him down. Why?

- 2.14.** Let  $X$  be an ensemble of  $M$  points  $a_1, \dots, a_M$  and let  $P_X(a_M) = \alpha$ . Show that

$$H(X) = \alpha \log \frac{1}{\alpha} + (1 - \alpha) \log \frac{1}{1 - \alpha} + (1 - \alpha)H(Y)$$

where  $Y$  is an ensemble of  $M - 1$  points  $a_1, \dots, a_{M-1}$  with probabilities  $P_Y(a_j) = P_X(a_j)/(1 - \alpha)$ ;  $1 \leq j \leq M - 1$ . Show that

$$H(X) \leq \alpha \log \frac{1}{\alpha} + (1 - \alpha) \log \frac{1}{1 - \alpha} + (1 - \alpha) \log (M - 1)$$

and determine the condition for equality.

- 2.15.** The entropy of a discrete ensemble is regarded as a measure of uncertainty. Substantiate this interpretation by proving that any transfer of probability from one member of the ensemble to another that makes their probabilities more nearly equal will increase the entropy of the ensemble.
- 2.16.** Give an example of a joint ensemble  $XY$  where  $X$  has the sample space  $(a_1, a_2)$ ,  $Y$  has the sample space  $(b_1, b_2)$ , and

$$H(X) + \sum_{k=1}^2 P_{X|Y}(a_k | b_j) \log P_{X|Y}(a_k | b_j)$$

is positive for  $j = 1$  and negative for  $j = 2$ .

- 2.17.** Let  $a_1, \dots, a_K$  be a set of disjoint events and let  $P(a_1), \dots, P(a_K)$ , and  $Q(a_1), \dots, Q(a_K)$  be two different probability assignments on the events

$$\left[ \sum_{k=1}^K (P(a_k)) = \sum_{k=1}^K (Q(a_k)) = 1 \right]$$

Prove the following statements

$$(a) \quad \sum_{k=1}^K P(a_k) \log \frac{P(a_k)}{Q(a_k)} \geq 0$$

$$(b) \quad \sum_{k=1}^K \frac{[P(a_k)]^2}{Q(a_k)} \geq 1$$

The quantity in (a) is often called the entropy of  $P$  relative to  $Q$ .

- 2.18.** Consider the cascaded channels in Fig. 2.3.2 and assume that  $I(X;Z) = I(X;Y)$ : i.e., that no information about the input is lost in the second channel. Define 2 letters in the  $Z$  ensemble, say  $c_i$  and  $c_l$ , as being equivalent if  $P_{X|Z}(a_k | c_i) = P_{X|Z}(a_k | c_l)$  for all  $k$ .
- (a) Show that  $I(X;Z) = I(X;Y)$  iff, for each letter  $b_j$  in the  $Y$  alphabet,  $P_{Y|Z}(b_j | c_i) > 0$  and  $P_{Y|Z}(b_j | c_l) > 0$  implies that  $c_i$  and  $c_l$  are equivalent. In other words, the second channel destroys no information

about  $X$  if no nonequivalent letters in the  $Z$  alphabet are ever confused in the second channel.

- (b) Show that if  $I(X;Z) = I(X;Y)$ , then the same result is true for all choices of input distribution  $P_X(a_k)$ . Assume  $P_X(a_k) > 0$  for all  $k$  (if the second channel is interpreted as a receiver one often calls such receivers sufficient receivers.)

- 2.19.** Let  $XYZ$  be a discrete joint ensemble. Prove the validity of the following inequalities and if true find conditions for equality:

- (a)  $I(XY;Z) \geq I(X;Z).$
- (b)  $H(XY|Z) \geq H(X|Z).$
- (c)  $I(X;Z|Y) \geq I(Z;Y|X) - I(Z;Y) + I(X;Z).$
- (d)  $H(XYZ) - H(XY) \leq H(XZ) - H(X)$

- 2.20.** Let  $X$ ,  $Y$ , and  $Z$  be ensembles with two elements in each ensemble so that the eight elements in the joint  $XYZ$  ensemble can be taken as the vertices of a unit cube.

- (a) Find a joint probability assignment  $P(x,y,z)$  such that  $I(X;Y) = 0$  and  $I(X;Y|Z) = 1$  bit.
- (b) Find a joint probability assignment  $P(x,y,z)$  such that  $I(X;Y) = 1$  bit and  $I(X;Y|Z) = 0$ .

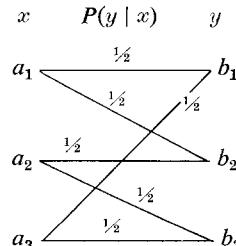
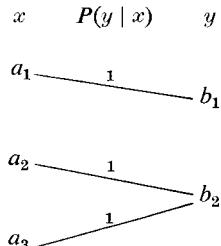
The point of the problem is that no general inequality exists between  $I(X;Y)$  and  $I(X;Y|Z)$ .

- 2.21.** In a joint ensemble  $XY$ , the mutual information  $I(x;y)$  is a random variable. In this problem we are concerned with the variance of that random variable,  $\text{VAR}[I(x;y)]$ .

- (a) Prove that  $\text{VAR}[I(x;y)] = 0$  iff there is a constant  $\alpha$  such that, for all  $x, y$  with  $P(x;y) > 0$ ,

$$P(x,y) = \alpha P_X(x)P_Y(y)$$

- (b) Express  $I(X;Y)$  in terms of  $\alpha$  and interpret the special case  $\alpha = 1$ .
- (c) For each of the channels below, find a probability assignment  $P_X(x)$  such that  $I(X;Y) > 0$  and  $\text{VAR}[I(x;y)] = 0$ . Calculate  $I(X;Y)$ .



- 2.22.** A zero mean, unit variance, Gaussian random variable (G.r.v.) by definition has the probability density  $p_X(x) = (1/\sqrt{2\pi}) \exp[-x^2/2]$ . That this is a

probability density with unit variance follows from the standard definite integrals [see Feller (1950), Section VII. 1 for a derivation]

$$\int_{-\infty}^{\infty} \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{x^2}{2}\right) dx = 1 \quad (\text{i})$$

$$\int_{-\infty}^{\infty} x^2 \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{x^2}{2}\right) dx = 1 \quad (\text{ii})$$

- (a) Show that if  $X$  is a zero mean, unit variance, G.r.v., then  $Y = a + bX$  has mean  $a$ , variance  $b^2$ , and density

$$p_Y(y) = \frac{1}{\sqrt{2\pi b^2}} \exp\left\{-\frac{(y-a)^2}{2b^2}\right\}$$

A random variable with this density is called a G.r.v. with mean  $a$ , variance  $b^2$ .

- (b) Let  $Y$  and  $Z$  be zero mean, statistically independent G.r.v. with variances  $b^2$  and  $c^2$ , respectively. By convolving  $p_Y(y)$  with  $p_Z(z)$ , show that  $W = X + Y$  is a zero mean G.r.v. with variance  $b^2 + c^2$ . Hint: Complete the square in the exponent and use (i).
- (c) Show that the characteristic function

$$\varphi_Y(\omega) = \overline{e^{j\omega y}} = \int_{-\infty}^{\infty} p_Y(y) e^{j\omega y} dy; \quad j = \sqrt{-1}$$

of a zero mean G.r.v. of variance  $b^2$  is given by  $\varphi_Y(\omega) = \exp[-\omega^2 b^2/2]$ .

- (d) If  $Y$  and  $Z$  are statistically independent, show that  $W = Y + Z$  has the characteristic function  $\varphi_W(\omega) = \varphi_Y(\omega)\varphi_Z(\omega)$ . Use this to obtain an independent derivation of your result in (b).

- 2.23.** A physical channel disturbed by thermal-agitation noise can often be represented by the following model. The channel input is a sequence of pulses, each pulse having a fixed duration  $T$  and an amplitude  $x$  of fixed magnitude,  $|x| = \sqrt{S}$ , but arbitrary sign. The receiver averages the channel output over each pulse interval and produces an output  $y$  given by  $y = x + z$ , where  $z$  is the average noise in the interval. Assume that  $z$  is a zero mean G.r.v. independent of  $x$ , with variance  $\sigma^2$  (for white noise of spectral density  $N_0/2$ ,  $\sigma^2 = N_0/2T$ ),

$$p_Z(z) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left\{-\frac{z^2}{2\sigma^2}\right\}$$

- (a) Find and sketch, as a function of  $S/\sigma^2$ , the probability that the sign of  $y$  is opposite to that of  $x$ .
- (b) Show that the above probability tends to  $\frac{1}{2} - \sqrt{S/(2\pi\sigma^2)}$  as  $S/\sigma^2 \rightarrow 0$  and tends to  $\sqrt{\sigma^2/(2\pi S)} \exp[-S/(2\sigma^2)]$  as  $S/\sigma^2 \rightarrow \infty$ .

*Hint:* See Feller (1950), Section VIII.1, for standard bounds on the “tail” of a Gaussian distribution.

- 2.24.** Let a continuous valued joint ensemble  $XY$  have a probability density  $p_{XY}(x,y)$  and individual densities  $p_X(x)$  and  $p_Y(y)$ . Show that  $I(X;Y) \geq 0$  and find an example where  $H(X)$ , as given in (2.4.24), is negative.
- 2.25.** A channel has a phase angle  $x$  as its input,  $0 \leq x \leq 2\pi$ , and a phase angle  $y$  as its output. The output is given by  $y = x + z$  where  $z$  is a noise variable, independent of the input  $x$ , and with probability density  $p_Z(z)$ . The sum  $x + z$  is to be interpreted as the remainder modulo  $2\pi$  (i.e., in the usual sense of adding phase angles). Let  $p_X(x) = 1/2\pi$ ;  $0 \leq x \leq 2\pi$ .
- Find  $I(X;Y)$  in terms of  $p_Z(z)$ .
  - Let  $p_Z(z) = 1/(b-a)$  for  $a < z \leq b$  and calculate  $I(X;Y)$ . Assume  $b-a \leq 2\pi$ . Interpret the fact that your answer depends only on  $b-a$  and not  $b$  or  $a$  separately.
- 2.26.** A channel has an input ensemble  $X$  consisting of the numbers  $+1$  and  $-1$  used with the probabilities  $P_X(+1) = P_X(-1) = \frac{1}{2}$ . The output  $y$  is the sum of the input  $x$  and an independent noise random variable  $Z$  with the probability density  $p_Z(z) = \frac{1}{4}$  for  $-2 < z \leq 2$  and  $p_Z(z) = 0$  elsewhere. In other words, the conditional probability density of  $y$  conditional on  $x$  is given by  $p_{Y|X}(y|x) = \frac{1}{4}$  for  $-2 < y-x \leq 2$  and  $p_{Y|X}(y|x) = 0$  elsewhere.
- Find and sketch the output probability density for the channel.
  - Find  $I(X;Y)$ .
  - Suppose the output is transformed into a discrete processed output  $u$  defined by  $u = 1$  for  $y > 1$ ;  $u = 0$  for  $-1 < y \leq 1$ ;  $u = -1$  for  $y \leq -1$ . Find  $I(X;U)$  and interpret your result in terms of the discussion at the end of Section 2.3.
- 2.27.** Let  $X$ ,  $Y$ , and  $Z$  be binary ensembles and consider the following probability assignment over the joint ensemble:
- $$P_{XYZ}(0,0,0) = P_{XYZ}(1,1,1) = \frac{1}{2}$$
- Show that  $I(X;Y|Z) = 0$ .
  - Show that  $\sup I(X_p; Y_p | Z_p) = 1$  bit, where the supremum is over all partitions of each ensemble. Hint: a partition can contain only one event, i.e., the entire sample space for that ensemble.
  - Show that for an arbitrary joint ensemble,  $I(X;Y|Z)$ , is unequal to  $\sup I(X_p; Y_p | Z_p)$  whenever  $I(X;Y) > I(X;Y|Z)$ .

## CHAPTER 3

- 3.1.** A source produces a sequence of statistically independent binary digits with the probabilities  $P(\mathbf{1}) = 0.005$ ;  $P(\mathbf{0}) = 0.995$ . These digits are taken 100 at a time and a binary code word is provided for every sequence of 100 digits containing 3 or fewer 1's.
- If the code words are all of the same length, find the minimum length required to provide the specified set of code words.
  - Find the probability of getting a source sequence for which no code word has been provided.

- (c) Use the Chebyshev inequality to bound the probability of getting a sequence for which no code word has been provided and compare with part (b).
- 3.2.** A source produces statistically independent binary digits with the probabilities  $P(0) = \frac{3}{4}$ ;  $P(1) = \frac{1}{4}$ . Consider sequences  $\mathbf{u}$  of  $L$  successive digits and the associated inequality
- $$\Pr\left[\left|\frac{I(\mathbf{u})}{L} - H(U)\right| \geq \delta\right] \leq \epsilon \quad (\text{i})$$
- where  $H(U)$  is the entropy of the source.
- (a) Find  $L_0$  such that (i) holds for  $L \geq L_0$  when  $\delta = 0.05$ ,  $\epsilon = \frac{1}{10}$ . Hint: Use the Chebyshev inequality.
- (b) Repeat for  $\delta = 10^{-3}$ ,  $\epsilon = 10^{-6}$ .
- (c) Let  $A$  be the set of sequences  $\mathbf{u}$  for which
- $$\left|\frac{I(\mathbf{u})}{L} - H(U)\right| < \delta$$
- Find upper and lower bounds for the number of sequences in  $A$  when  $L = L_0$  for the cases in (a) and (b).
- 3.3.** A source has an alphabet of 4 letters. The probabilities of the letters and two possible sets of binary code words for the source are given below:

| Letter | Prob. | Code I     | Code II     |
|--------|-------|------------|-------------|
| $a_1$  | 0.4   | <b>1</b>   | <b>1</b>    |
| $a_2$  | 0.3   | <b>01</b>  | <b>10</b>   |
| $a_3$  | 0.2   | <b>001</b> | <b>100</b>  |
| $a_4$  | 0.1   | <b>000</b> | <b>1000</b> |

For each code, answer the following questions (no proofs or numerical answers are required).

- (a) Does the code satisfy the prefix condition?
- (b) Is the code uniquely decodable?
- (c) What is the mutual information provided about the event that the source letter is  $a_1$  by the event that the first letter of the code word is **1**?
- (d) What is the average mutual information provided about the source letter by the specification of the first letter of the code word? Give an heuristic description of the purpose of the first letter in the code words of code II.
- 3.4.** A code is not uniquely decodable iff there exists a finite sequence of code letters that can be resolved in two different ways into sequences of code words. That is, a situation such as

|       |       |       |       |             |       |
|-------|-------|-------|-------|-------------|-------|
| $A_1$ |       | $A_2$ |       | $A_3 \dots$ | $A_m$ |
| $B_1$ | $B_2$ | $B_3$ | $B_4$ | $\dots$     | $B_n$ |

must occur where each  $A_i$  (and each  $B_j$ ) is a code word. Note that  $B_1$  must be a prefix of  $A_1$  with some resulting “dangling suffix.” Each dangling suffix in this sequence in turn must either be a prefix of a code word or have a code word as prefix, resulting in another dangling suffix. Finally, the last dangling suffix in the sequence must itself be a code word. Thus one can set up a test for unique decodability [which is essentially the Sardinas-Patterson (1953) test] in the following way. Construct a set  $S$  of all possible dangling suffixes. The code is uniquely decodable if and only if  $S$  contains no code word.

- State the precise rules for building the set  $S$ .
- Suppose the code word lengths are  $m_i$ ,  $i = 1, 2, \dots, M$ . Find a good upper bound on the number of elements in the set  $S$ .
- Determine which of the following codes are uniquely decodable:

|                 |                            |
|-----------------|----------------------------|
| $\{0, 10, 11\}$ | $\{00, 01, 10, 11\}$       |
| $\{0, 01, 11\}$ | $\{110, 11, 10\}$          |
| $\{0, 01, 10\}$ | $\{110, 11, 100, 00, 10\}$ |
| $\{0, 01\}$     |                            |

- For each uniquely decodable code in (c), construct if possible an infinitely long encoded sequence with known starting point such that it can be resolved into code words in two different ways. (This illustrates that unique decodability does not always imply finite decodability.) Prove that such a sequence cannot arise for a prefix code.
- 3.5.** Consider the following method of constructing binary code words for a message ensemble  $U$  with probability assignment  $P(u)$ . Let  $P(a_k) \leq P(a_j)$  for  $k > j \geq 1$ , and define

$$Q_i = \sum_{k=1}^{i-1} P(a_k) \quad \text{for } i > 1; Q_1 = 0$$

The code word assigned to message  $a_i$  is formed by finding the “decimal” expansion of  $Q_i < 1$  in the binary number system (i.e.,  $\frac{1}{2} \rightarrow 100 \dots$ ,  $\frac{1}{4} \rightarrow 0100 \dots$ ,  $\frac{5}{8} \rightarrow 10100 \dots$ ) and then truncating this expansion to the first  $n_i$  digits, where  $n_i$  is the integer equal to or just larger than  $I(a_i)$  bits.

- Construct binary code words for a set of eight messages occurring with probabilities  $\frac{1}{4}, \frac{1}{4}, \frac{1}{8}, \frac{1}{8}, \frac{1}{16}, \frac{1}{16}, \frac{1}{16}, \frac{1}{16}$ .
- Prove that the method described above yields in all cases a set of code words satisfying the prefix condition, and whose average length  $\bar{n}$  satisfies the inequality

$$H(U) \leq \bar{n} < H(U) + 1$$

- 3.6.** It is sometimes desired to encode a set of source messages  $a_1, \dots, a_K$  into an “alphabetic” binary code. A binary code is alphabetic if for each  $i, k$  with  $i < k$ , the code word for  $a_i$  (interpreted as a binary “decimal” expansion) is less than the code word for  $a_k$ . If  $P(a_k) \geq P(a_i)$  for all  $i$  and  $k$  with  $i < k$ , we could use the procedure of Problem 3.5. If this condition is not satisfied,

we can still proceed as follows. Let

$$Q_i = \sum_{k=1}^{i-1} P(a_k) \quad \text{for } i > 1; Q_1 = 0$$

The code word for message  $a_i$  is formed by truncating the binary “decimal” expansion of  $Q_i$  to the first  $n_i$  digits where  $n_i$  is the integer equal to or just greater than  $I(a_i) + 1$ . Prove that a code generated by the above rule is alphabetic, satisfies the prefix condition, and has an average length satisfying

$$\bar{n} \leq H(U) + 2$$

- 3.7.** (a) Show that Theorems 3.2.1 and 3.2.2 are valid for  $K = \infty$ . *Hint:* For Theorem 3.2.2, show that unique decodability implies that

$$\sum_{i=1}^k D^{-n_i} \leq 1$$

for every finite  $k$  and then pass to the limit.

- (b) Show that Theorem 3.3.1 is valid for a source ensemble with an infinite countable alphabet and finite entropy.  
**3.8.** The source coding theorem states that for a source of entropy  $H(U)$  it is possible to assign a binary code word to each letter in the alphabet in such a way as to generate a prefix condition code of average length

$$\bar{n} < H(U) + 1$$

Show by example that this theorem cannot be improved upon. That is, for any  $\epsilon > 0$ , find a source for which the minimum  $\bar{n}$  satisfies

$$\bar{n} \geq H(U) + 1 - \epsilon$$

- 3.9.** Consider two discrete memoryless sources. Source 1 has an alphabet of 6 letters with the probabilities 0.3, 0.2, 0.15, 0.15, 0.1, 0.1. Source 2 has an alphabet of 7 letters with the probabilities 0.3, 0.25, 0.15, 0.1, 0.1, 0.05, 0.05. Construct a binary Huffman code and a ternary Huffman code for each set of messages. Find the average number of code letters per source letter for each code.  
**3.10.** For the binary Huffman code for source 1 in Problem 3.9, let  $N_L$  be a random variable representing the total number of code letters generated by a sequence of  $L$  source letters passing into the encoder.  
 (a) Find  $\lim_{L \rightarrow \infty} (N_L/L)$  and state carefully the sense in which this limit exists.  
 (b) Suppose a binary code is designed using the set of all sequences of length  $K$  from the same source as the message set and let  $N_{LK}(K)$  be a random variable representing the total number of code letters generated by a sequence of  $LK$  source letters passing into the encoder. Find

$$\lim_{K \rightarrow \infty} \lim_{L \rightarrow \infty} \frac{N_{LK}(K)}{LK}$$

and state the sense in which the limit exists.

- 3.11.** For each set of messages in Problem 3.9 find a prefix condition code of minimum average length under the constraint that the first letter of each word must be **0** or **1** and each successive code letter can be **0**, **1**, or **2**. Find a general rule for constructing prefix condition codes of minimum average length with this constraint and outline why it works.
- 3.12.** For source 1 in Problem 3.9, find a binary code of minimum average length that satisfies the *suffix condition*. The suffix condition is that no code word be the same as any suffix of any other code word. Show that a suffix condition code is always uniquely decodable and show that the minimum average length over all codes satisfying the suffix condition for a given source is the same as the average length of the Huffman code for that source.
- 3.13.** A set of eight messages with probabilities of 0.2, 0.15, 0.15, 0.1, 0.1, 0.1, 0.1, 0.1 is to be encoded into a ternary prefix condition code. Construct two sets of code words whose lengths have the same minimum average value but different variances. Evaluate the common average length and the two variances. State reasons why one or the other code might be preferable in applications.
- 3.14.** A discrete memoryless source of alphabet size  $K$  has an entropy of  $H(U)$  bits. A set of ternary code words satisfying the prefix condition is found for the source letters, and the average length of the code words satisfies

$$\bar{n} = \frac{H(U)}{\log_2 3}$$

- (a) Prove that each source letter has a probability of the form  $3^{-i}$  where  $i$  is an integer depending on the letter.  
 (b) Prove that the number of messages in the source alphabet is odd.
- 3.15.** A source has an alphabet of  $K$  letters and each letter is used with the same probability. These letters are encoded into binary code words by the Huffman technique (i.e., so as to minimize the average code word length). Let  $j$  and  $x$  be chosen so that  $K = x 2^j$ , where  $j$  is an integer and  $1 \leq x < 2$ .
- (a) Do any code words have lengths not equal to either  $j$  or  $j + 1$ ? Why?  
 (b) In terms of  $j$  and  $x$ , how many code words have length  $j$ ?  
 (c) What is the average code word length?
- 3.16.** Define the mismatch between a binary code and a message ensemble  $U$  as  $\bar{n} - H(U)$ . Show that the mismatch between  $U$  and an optimum binary code for  $U$  is always greater than or equal to that between  $U'$  and an optimum binary code for  $U'$  where  $U'$  is the reduced ensemble for  $U$  discussed in Section 3.4.
- 3.17.** The following problem concerns a technique known as run length coding. Along with being a useful technique, it should make you look carefully into the sense in which the Huffman coding technique is optimum. A source  $U$  produces a sequence of independent binary digits with the probabilities

$P(0) = 0.9$ ,  $P(1) = 0.1$ . We shall encode this sequence in two stages, first counting the number of zeros between successive ones in the source output and then encoding their run lengths into binary code words. The first stage of encoding maps source sequences into intermediate digits by the following rule:

| Source Sequence | Intermediate Digits<br>(# zeros) |
|-----------------|----------------------------------|
| <b>1</b>        | 0                                |
| <b>01</b>       | 1                                |
| <b>001</b>      | 2                                |
| <b>0001</b>     | 3                                |
| .               | .                                |
| .               | .                                |
| <b>00000001</b> | 7                                |
| <b>00000000</b> | 8                                |

Thus the following sequence is encoded as follows:

$\begin{matrix} \mathbf{1} & \mathbf{0} & \mathbf{0} & \mathbf{1} & \mathbf{0} & \mathbf{1} & \mathbf{1} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{1} \\ 0, & 2, & & & & & & 8, & & & & & & 2, 0, & & & & 4 \end{matrix}$

The final stage of encoding assigns a code word of one binary digit to the intermediate integer 8 and code words of four binary digits to the other intermediate integers.

- (a) Justify, in whatever detail you find convincing to yourself, that the overall code is uniquely decodable.
- (b) Find the average number  $\bar{n}_1$  of source digits per intermediate digit.
- (c) Find the average number  $\bar{n}_2$  of encoded binary digits per intermediate digit.
- (d) Show, by appeal to the law of large numbers, that for a very long sequence of source digits, the ratio of the number of encoded binary digits to the number of source digits will with high probability be close to  $\bar{n}_2/\bar{n}_1$ . Compare this ratio to the average number of code letters per source letter for a Huffman code encoding four source digits at a time.

- 3.18.** (a) A source has five letters with the following probabilities:

$$P(a_1) = 0.3$$

$$P(a_2) = 0.2$$

$$P(a_3) = 0.2$$

$$P(a_4) = 0.15$$

$$P(a_5) = 0.15$$

These letters are to be coded into binary digits for use on a noiseless channel. It takes 1 second to transmit a 0 and 3 seconds to transmit a 1. Using cut and try techniques, find a code with the prefix condition that minimizes the average time required to transmit a source letter and calculate this minimum average time.

- (b) Any such code can be represented by a tree in which the length of a branch is proportional to the time required to transmit the associated digit. Show that for a code to minimize the average transmission time, the probabilities associated with intermediate and terminal nodes must be nonincreasing with length.

- 3.19.** A number  $M$  of pennies are given of which  $M - 1$  are known to have the same weight. The  $M$ th penny may have the same weight as the others, may be heavier, or may be lighter. A balance is available on which two groups of pennies may be compared. It is desired to find the odd penny, if there is one, and to determine whether it is heavier or lighter than the others. Find the maximum value of  $M$  for which the problem may be solved with  $n$  weighing operations, and describe in detail the procedure required if:
- A standard penny is available for comparison, in addition to the  $M$  pennies.
  - A standard penny is not available.

*Suggestion:* Consider the  $2M + 1$  possible answers as forming a set of equiprobable alternatives, and assign to each alternative a code word representing the results of successive weighing operations.

- 3.20.** Define the conditional entropy

$$H_{L|L}(U) = \frac{1}{L} H(U_{2L} \cdots U_{L+1} | U_L \cdots U_1)$$

for a discrete stationary source of alphabet size  $K$ .

- Prove that  $H_{L|L}(U)$  is nonincreasing with  $L$ .
- Prove that

$$\lim_{L \rightarrow \infty} H_{L|L}(U) = H_\infty(U)$$

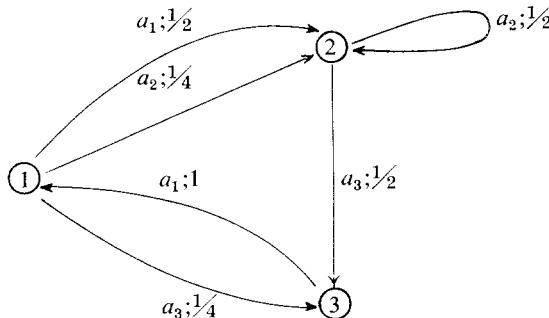
- Consider a source coding technique in which for each of the  $K^L$  choices of the previous  $L$  letters into the encoder, a Huffman code is generated for the set of choices for the subsequent  $L$  letters into the encoder. Give the analogue of Theorem 3.5.2 for this technique.

- 3.21.** Consider a stationary ergodic binary source with the source sequence denoted by  $\mathbf{u} = (\dots, u_{-1}, u_0, u_1, \dots)$ . Assume that the even-numbered digits,  $\dots, u_2, u_0, u_2, \dots$ , are statistically independent equiprobable binary random variables. With probability  $\frac{1}{2}$ ,  $u_{2n+1} = u_{2n}$  for all  $n$ , and with probability  $\frac{1}{2}$ ,  $u_{2n-1} = u_{2n}$  for all  $n$ . Suppose pairs of source digits  $(u_{2n}, u_{2n+1})$  are encoded by the rule

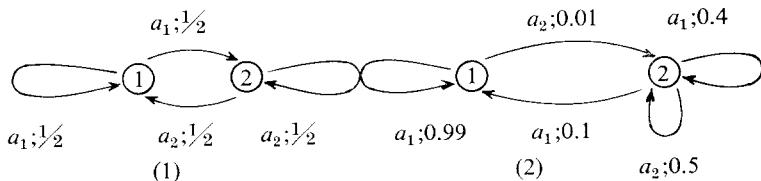
$$\begin{aligned} 00 &\rightarrow 0 \\ 11 &\rightarrow 10 \\ 01 &\rightarrow 110 \\ 10 &\rightarrow 111 \end{aligned}$$

Show that with probability  $\frac{1}{2}$ , the number of code letters per source letter in a long sequence tends to  $\frac{3}{4}$  and with probability  $\frac{1}{2}$  tends to  $\frac{9}{8}$ .

- 3.22. (a) Find the limiting state probabilities  $q(j)$  for the Markov source indicated below and find the limiting single letter probabilities.



- (b) Find the entropy of the source output letter conditional on the current source state  $H(U|s=j)$ , for  $1 \leq j \leq 3$ .  
 (c) Find the entropy per letter of the source sequence  $H_\infty(U)$ .  
 (d) For each source state  $s=j$ , find an optimal variable length binary code for the letters in the source alphabet for which  $P_j(a_k) > 0$ . Show that the entire code, choosing code words according to current state and output, is uniquely decodable, even though you choose the single code word for state 3 to have zero length.  
 (e) Calculate the average number of code letters per source letter  $\bar{n}$ . State general conditions under which  $\bar{n} = H_\infty(U)$  for such a coding strategy.
- 3.23. For the sources indicated below, there is an underlying Markov process with states indicated by circles. Each time unit, the source is in a given state, emits a letter ( $a_1$  or  $a_2$ ) and passes to a new state. The number on each branch is the probability of emitting the indicated letter and going to the indicated state at the head of the branch conditional on being in the state at the tail of the branch. Note that these are not Markov sources since the new state is not specified by the current state and output. Sources of this type are often called derived Markov sources.



- (a) Show that source (1) in the figure is a thinly camouflaged version of a binary source with independent equally probable letters and has an entropy of 1 bit per letter [a careless application of (3.6.21) would give an entropy of zero].

- (b) Show that the entropy of the second source (and of any such source) is bounded by

$$H(U_l \mid U_{l-1} \cdots U_1 S_1) \leq H_\infty(U) \leq H(U_l \mid U_{l-1} \cdots U_1)$$

Show that the lower bound is nondecreasing with  $l$  and the upper bound is nonincreasing with  $l$ . [See Gilbert (1960) for a series expansion technique to find  $H_\infty(U)$  for the particular form of source here.]

#### CHAPTER 4

- 4.1.** A set of  $N$  discrete memoryless channels have the capacities  $C_1, C_2, \dots, C_N$ . Let these channels be connected in parallel in the sense that each unit of time an arbitrary symbol is transmitted and received over each channel. Thus the input  $\mathbf{x} = (x_1, \dots, x_N)$  to the parallel channel is an  $N$ -tuple, the components of which are inputs to the individual channels, and the output  $\mathbf{y} = (y_1, \dots, y_N)$  is an  $N$ -tuple whose components are the individual channel outputs. Prove that the capacity of the parallel channel is

$$\sum_{n=1}^N C_n.$$

Assume that the output from each channel is statistically related *only* to the input to that channel: i.e.,

$$\Pr(\mathbf{y} \mid \mathbf{x}) = \prod_n P(y_n \mid x_n)$$

Do *not* assume that the inputs are independent. *Hint:* See the proof of Theorem 4.2.1.

- 4.2.** The output of a channel is passed through a data processor in such a way that no information is lost. That is,

$$I(X;Y) = I(X;Z)$$

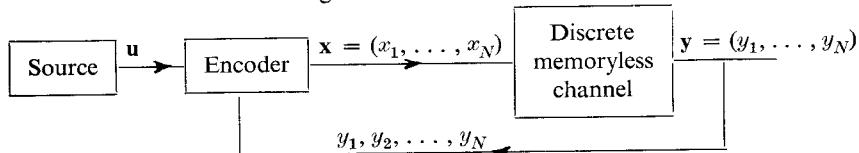
where  $X$  is the channel input ensemble,  $Y$  is the channel output ensemble, and  $Z$  is the processor output. Find an example where  $H(Y) > H(Z)$  and find an example where  $H(Y) < H(Z)$ . The data processor does not have to be deterministic.

- 4.3.** Another proof of Theorem 4.3.1 goes as follows: We can interpret this joint ensemble  $UV$  as a joint ensemble  $UVE$  where the sample points of  $E$  are the two events  $e$  and  $c$ ,  $e$  corresponding to an error ( $u \neq v$ ) and  $c$  corresponding to correct reception ( $u = v$ ). Then

$$\begin{aligned} H(U \mid V) &= H(UE \mid V) = H(E \mid V) + H(U \mid VE) \\ &= H(E \mid V) + P_e H(U \mid Ve) + (1 - P_e) H(U \mid Vc) \\ &\leq H(E) + P_e H(U \mid Ve) \\ &\leq H(E) + P_e \log(M - 1) \end{aligned}$$

Justify the steps above that are non-obvious.

- 4.4.** A source produces statistically independent equiprobable binary digits at a rate of 1 per second. The source output is encoded and transmitted over a binary symmetric channel with a crossover probability  $\epsilon$ ,  $0 < \epsilon < \frac{1}{2}$ . One channel digit is transmitted each second. Using the converse to the coding theorem, find upper and lower bounds to  $\langle P_e \rangle$ . Interpret why it is impossible to make the error probability larger than your upper bound. Compare your bounds with  $\langle P_e \rangle$  if no encoding or decoding is performed.
- 4.5.** A discrete memoryless source produces one digit per second and has an entropy of 2 bits. The output from the source is encoded and transmitted over a channel of capacity  $C_T$  bits per second.
- Assuming a source alphabet size  $M = 4$ , carefully sketch, as a function of  $C_T$ , the lower bound to error probability per source digit of Theorem 4.3.4.
  - For any given  $C_T$ , find a channel for which the preceding lower bound is achieved exactly without any coding.
  - Now assume that the source alphabet size  $M$  is unknown but that  $C_T = 1$  and  $\langle P_e \rangle = 10^{-6}$ . Find a lower bound to  $M$  using Theorem 4.3.4.
- 4.6.** In this problem, we wish to show that the capacity of a discrete memoryless channel is not increased by the presence of feedback from receiver to transmitter. Consider the diagram below:



Each channel output as received is fed back into the encoder and can affect the selection of subsequent channel inputs. We wish to show that

$$I(\mathbf{U}; \mathbf{Y}^N) \leq \sum_{n=1}^N I(X_n; Y_n) \leq NC$$

We can show this by the following steps; your problem is to show that each of the steps is valid.

$$\begin{aligned} I(\mathbf{U}; \mathbf{Y}^N) &= \sum_{n=1}^N I(\mathbf{U}; Y_n | Y_{n-1} \cdots Y_1) \\ I(\mathbf{U}; Y_n | Y_1 \cdots Y_{n-1}) &\leq I(\mathbf{U}X_n; Y_n | Y_1 \cdots Y_{n-1}) \\ &= I(X_n; Y_n | Y_1 \cdots Y_{n-1}) \\ &\leq I(X_n; Y_n) \end{aligned}$$

With the aid of this result, show that the converse to the coding theorem is valid for a discrete memoryless channel with feedback. Note: This result is not valid for channels with memory.

- 4.7.** In Theorem 4.3.1, we found a lower bound on  $P_e$  in terms of  $H(U | V)$ . Here we shall assume that the decoder uses minimum error probability

decoding and derive an upper bound to  $P_e$ . Letting  $U$  and  $V$  have the same sample space, say  $a_1, \dots, a_k$ , minimum error probability decoding has the property that  $P_{U|V}(a_i | a_i) \geq P_{U|V}(a_k | a_i)$  for all  $k \neq i$ . Using this property show that

$$P_e \leq H(U | V) \text{ nats} \quad (\text{i})$$

*Hint:* First let  $W$  be an arbitrary ensemble with probabilities  $q_i$ . Show that (in nats)

$$H(W) \geq \sum_i q_i(1 - q_i) \geq 1 - q_{\max} \quad (\text{ii})$$

Let

$$H(U | v) = - \sum_u P(u | v) \ln P(u | v)$$

and let  $P_e(v)$  be the probability of error given that  $v$  is the decoded message. Use (ii) to show that  $H(U | v) \geq P_e(v)$ , and use this result to establish (i).

*Bonus:* Show that (ii) can be replaced with the stronger inequality that  $H(W)$  in bits satisfies  $H(W) \geq 2(1 - q_{\max})$ , and thus  $P_e \leq \frac{1}{2} H(U | V)$  bits.

- 4.8. Use the inequality  $\log z \leq (z - 1) \log e$  to show that (4.3.16) is valid.
- 4.9. Prove that (4.4.4) and (4.4.5) are valid inequalities. *Hint* [for (4.4.4)]: Let  $\alpha, \beta$  be two points in the interval and let  $\delta = \lambda\alpha + (1 - \lambda)\beta$  be a point between  $\alpha$  and  $\beta$ . Then use the Taylor series expansion theorem that for any  $x$

$$f(x) = f(\delta) + (x - \delta)f'(\delta) + \frac{(x - \delta)^2}{2}f''(y)$$

for some  $y$  between  $x$  and  $\delta$ . Also draw a figure! *Hint* [for (4.4.5)]: Use induction on  $L$ .

- 4.10. Let  $Q_1(x)$  and  $Q_2(x)$  be any two probability assignments over the same discrete sample space and let  $H_1(X)$  and  $H_2(X)$  be the corresponding entropies.
  - (a) For  $0 \leq \lambda \leq 1$ , show that  $Q(x) = \lambda Q_1(x) + (1 - \lambda)Q_2(x)$  is a probability assignment over the sample space.
  - (b) Let  $H(X)$  be the entropy corresponding to the probability assignment  $Q(x)$  in (a). Prove that

$$H(X) \geq \lambda H_1(X) + (1 - \lambda)H_2(X)$$

and state the conditions for equality.

- (c) Give a geometric interpretation of the result in (b).
- 4.11. A function  $f(\alpha)$  is defined on a convex region  $R$  of vector space. Prove that  $f$  is convex  $\cup$  if and only if the function  $f(\lambda\alpha_1 + (1 - \lambda)\alpha_2)$  is a convex function of  $\lambda$ ,  $0 \leq \lambda \leq 1$ , for all  $\alpha_1, \alpha_2$  in  $R$ .
- 4.12. Let  $f_i(\alpha)$ ,  $i \in I$  ( $I$  is an arbitrary index set) be a set of functions each of which is convex  $\cup$  and decreasing with the real variable  $\alpha$ . Assume that  $f(\alpha) = \sup_{i \in I} f_i(\alpha)$  is finite everywhere and prove that  $f(\alpha)$  is convex  $\cup$  and decreasing in  $\alpha$ .

- 4.13.** Consider the function of 2 variables

$$f(\alpha) = \alpha_1(2 - \alpha_1) - (\alpha_2 + 1)^2$$

over the region  $\alpha_1 \geq 0, \alpha_2 \geq 0$ .

- (a) Show that  $f(\alpha)$  is convex  $\cap$  over the given region.
- (b) Find the maximum of  $f(\alpha)$  over the region and indicate clearly why it is a maximum.

- 4.14.** An additive Gaussian noise channel can often be modeled as a set of parallel, discrete time channels where the set of channels is known to have the capacity

$$C(S_1, \dots, S_L) = \sum_{l=1}^L \frac{1}{2} \log \left( 1 + \frac{S_l}{N_l} \right)$$

where  $N_l$  is the noise power on the  $l$ th channel and  $S_l$  is the signal power on the  $l$ th channel. Suppose a total amount of signal power  $S$  is available which can be apportioned among the channels in any desired way. Let  $\alpha_l$  be the fraction of power apportioned to the  $l$ th channel, so that  $S_l = \alpha_l S$ .

- (a) Find a set of necessary and sufficient conditions on the fractions  $\alpha_1, \dots, \alpha_L$  to maximize  $C(S_1, \dots, S_L)$ , subject to the constraints

$$\alpha_l \geq 0, \sum_l \alpha_l = 1.$$

- (b) Find the maximizing values for  $S_1, S_2$ , and  $S_3$  for the case  $L = 3, S = 2, N_1 = 1, N_2 = 2, N_3 = 3$ . Hint: Use Theorem 4.4.1.

- 4.15.** *Useful Inequalities in Information Theory.* In the following inequalities let  $a_i, b_i, P_i, Q_i$  be non-negative numbers defined over a finite set of  $i, 1 \leq i \leq A$ , say. Assume

$$\sum_i P_i = \sum_i Q_i = 1.$$

Let  $s$  and  $r$  be positive numbers and  $\lambda$  satisfy  $0 < \lambda < 1$ . Prove the validity of the following inequalities [a variety of proofs for each may be found in Hardy, Littlewood, and Polya (1934)].

- (a)

$$\sum_i Q_i^\lambda P_i^{1-\lambda} \leq 1,$$

with equality iff  $P_i = Q_i$ , all  $i$ .

Hint: Show that left side is convex  $\cup$  in  $\lambda$  and evaluate as  $\lambda \rightarrow 0, \lambda \rightarrow 1$ .

- (b) Holder's inequality:

$$\sum_i a_i b_i \leq \left( \sum_i a_i^{1/\lambda} \right)^\lambda \left[ \sum_i b_i^{1/(1-\lambda)} \right]^{1-\lambda}$$

with equality iff for some  $c$ ,  $a_i^{1-\lambda} = b_i^\lambda c$  for all  $i$ .

Hint: Define

$$Q_i = a_i^{1/\lambda} / \left( \sum_i a_i^{1/\lambda} \right)$$

and

$$P_i = b_i^{1/(1-\lambda)} / \left[ \sum_i b_i^{1/(1-\lambda)} \right]$$

and use (a). In the special case  $\lambda = 1/2$ , this is called the Cauchy inequality, and the integral analogue is the Schwarz inequality.

(c) Variant of Holder's inequality:

$$\sum_i P_i a_i b_i \leq \left( \sum_i P_i a_i^{1/\lambda} \right)^\lambda \left[ \sum_i P_i b_i^{1/(1-\lambda)} \right]^{1-\lambda}$$

with equality iff for some  $c$ ,  $P_i a_i^{1/\lambda} = P_i b_i^{1/(1-\lambda)} c$  for all  $i$ .

*Hint:* Use  $P_i^\lambda a_i$  for  $a_i$  in (b) and  $P_i^{1-\lambda} b_i$  for  $b_i$  in (b).

(d)

$$\begin{aligned} (\sum_i P_i a_i)^r &\leq \sum_i P_i a_i^r & r > 1 \\ &\geq \sum_i P_i a_i^r & r < 1 \end{aligned}$$

with equality iff the  $a_i$  such that  $P_i > 0$  are constant.

*Hint:* For  $r > 1$ , use (c) with  $b_i = 1$ ; for  $r < 1$  use  $a_i^r$  for  $a_i$  in (c).

$$(e) \quad \left( \sum_i P_i a_i^r \right)^{1/r} \leq \left( \sum_i P_i a_i^s \right)^{1/s}; \quad 0 < r < s$$

with equality iff the  $a_i$  such that  $P_i > 0$  are constant.

*Hint:* Use (c) with  $b_i = 1$ ,  $\lambda = r/s$ .

$$(f) \quad \begin{aligned} \left( \sum_i a_i \right)^r &\leq \sum_i a_i^r; & r \leq 1 \\ &\geq \sum_i a_i^r; & r \geq 1 \end{aligned}$$

with equality iff  $r = 1$  or if only one  $a_i$  is nonzero. Note difference between (d) and (f). *Hint:* Let  $P_i = a_i / \sum a_i$  and consider the ratio of the right terms to the left term.

(g) Minkowski's inequality: Let  $a_{jk}$  be a non-negative set of numbers,  $1 \leq j \leq J$ ,  $1 \leq k \leq K$ . Then

$$\begin{aligned} \left[ \sum_j \left( \sum_k a_{jk} \right)^{1/r} \right]^r &\leq \sum_k \left( \sum_j a_{jk}^{1/r} \right)^r; & r < 1 \\ &\geq \sum_k \left( \sum_j a_{jk}^{1/r} \right)^r; & r > 1 \end{aligned}$$

with equality iff  $a_{jk}$  independent of  $k$ . *Hint:* For  $r < 1$ , use

$$\left( \sum_k a_{jk} \right)^{1/r} = \left( \sum_k a_{jk} \right) \left( \sum_i a_{ji} \right)^{(1-r)/r}$$

to obtain

$$\sum_j \left( \sum_k a_{jk} \right)^{1/r} = \sum_k \left[ \sum_j a_{jk} \left( \sum_i a_{ji} \right)^{(1-r)/r} \right]$$

Apply Holder's inequality to the term in brackets for each  $k$ , getting

$$\sum_j \left( \sum_k a_{jk} \right)^{1/r} \leq \sum_k \left( \sum_j a_{jk}^{1/r} \right)^r \left[ \sum_j \left( \sum_i a_{ji} \right)^{1/r} \right]^{1-r}$$

Divide both sides by the term in brackets to obtain the result. For  $r > 1$ , let  $r' = 1/r$  and use (g) with  $r' < 1$ , using  $a_{jk}^{1/r}$  in place of  $a_{jk}$ .

(h) Variant of Minkowski's inequality:

$$\left[ \sum_j Q_j \left( \sum_k a_{jk} \right)^{1/r} \right]^r \leq \sum_k \left( \sum_j Q_j a_{jk}^{1/r} \right)^r; \quad r < 1$$

with the inequality reversed for  $r > 1$ . Hint: Use  $Q_j^r a_{jk}$  for  $a_{jk}$  in (g). Note that if  $a_1, \dots, a_K$  is a set of random variables taking on values  $a_{j1}, a_{j2}, \dots, a_{jK}$  on the  $j$ th element of a sample space which occurs with probability  $Q_j$ , then (h) can be expressed as

$$\left[ \left( \sum_k a_k \right)^{1/r} \right]^r \leq \sum_k \left[ a_k^{1/r} \right]^r; \quad r < 1$$

with the inequality reversed for  $r > 1$ .

- 4.16.** (a) Let  $X$  and  $Y$  be the input and output ensembles, respectively, for a discrete memoryless channel (DMC). Show that  $H(Y)$  is a convex  $\cap$  function of the *input* probability vector. Hint: Consider the output probability vectors resulting from the input probability vectors.  
 (b) Give an example of a DMC for which the convexity as in (a) is not strict.  
 (c) Again considering the DMC, show that  $-H(Y|X)$  is a linear function of the input probability vector.  
 (d) Combining (a) and (c), show that  $I(X;Y)$  is a convex  $\cap$  function of the input probability vector.

- 4.17.** For an arbitrary assignment  $\mathbf{Q}_0$  for a DMC, let

$$I_0(x = k; Y) = \sum_j P(j|k) \log \frac{P(j|k)}{\sum_i Q_0(i)P(j|i)}$$

and show that

$$\sum_k Q_0(k) I_0(x = k; Y) \leq C \leq \max_k I_0(x = k; Y) \tag{i}$$

Use this to show that

$$C = \min_{\mathbf{Q}_0} \max_k I_0(x = k; Y) \tag{ii}$$

*Hint:* Let  $I_1(X; Y)$  correspond to some  $\mathbf{Q}_1 \neq \mathbf{Q}_0$ , and by considering the partial derivative of  $I_0(X; Y)$  with respect to  $Q_0(k)$ , show that

$$I_1(X; Y) \leq \sum_k Q_1(k) I_0(x = k; Y).$$

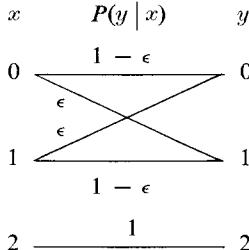
Note that in a computer search for  $C$ , (i) provides bounds on how close any given  $\mathbf{Q}_0$  is to achieving  $C$  and this provides a way to terminate the search when  $C$  has been approximated sufficiently closely.

- 4.18.** (a) Consider  $n$  (in general different) DMC's with capacities  $C_1, C_2, \dots, C_n$ . The "sum" channel associated therewith is that channel whose input and output alphabets are the unions of those of the original channels; i.e., the sum channel has all  $n$  channels available for use but only one channel may be used at any given time. Show that the capacity of the sum channel is given by

$$C = \log_2 \sum_{i=1}^n 2^{C_i}$$

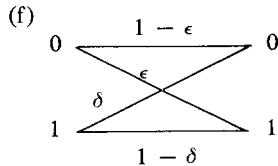
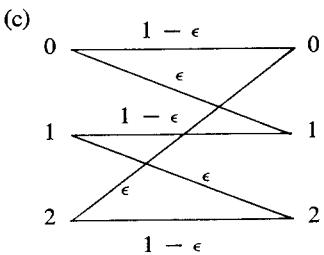
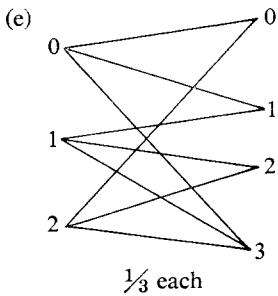
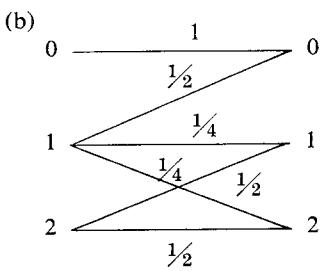
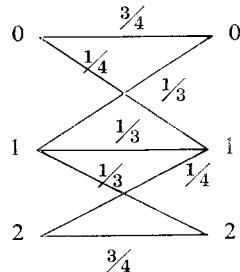
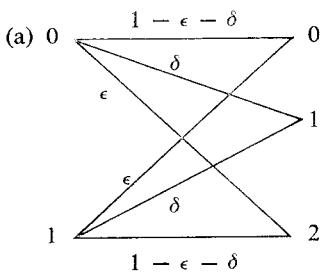
and find  $q(i)$ , the probability of using the  $i$ th channel. Interpret  $C$  as the average of the capacities of the individual channels plus the information conveyed by the selection of a channel. *Hint:* Write the input probability distribution as  $q(i)Q_i(k)$  where  $Q_i(k)$  is the probability distribution on the inputs of the  $i$ th channel given that the  $i$ th channel is used. Then use Theorem 4.5.1.

- (b) Use the above result to find the capacity of the channel below.



- 4.19.** A DMC is called additive modulo  $K$  if it has the input and output alphabet  $0, 1, \dots, K - 1$  and the output  $y$  is given in terms of the input  $x$  and a noise variable  $Z$  by  $y = x \oplus z$ . The noise variable takes on values  $0, 1, \dots, K - 1$  and is statistically independent of the input and the addition  $x \oplus z$  is taken modulo  $K$  (i.e., as  $x + z$  or  $x + z - K$ , whichever is between 0 and  $K - 1$ ).
- Show that  $I(X; Y) = H(Y) - H(Z)$ .
  - Find the capacity in terms of  $H(Z)$  and find the maximizing input probability assignment.
- 4.20.** Find the capacity and an optimizing input probability assignment for each

of the DMC's below.



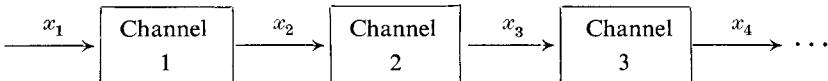
*Hint:* Only (f) requires appreciable algebraic manipulation.

- 4.21. Suppose a DMC has the property that for some input assignment  $Q(k) > 0$ ,  $0 \leq k \leq K - 1$ , the variance of  $I(x;y) = 0$ . Prove that that input assignment achieves channel capacity (see Problem 2.21).
- 4.22. Consider again the channel of Problem 2.23. Find an expression for the capacity of that channel, first taking the received number  $y$  as the channel output and then taking the sign of  $y$  as the channel output. Show that the first capacity exceeds the second. Show that in the limit as  $S \rightarrow 0$ , the first capacity approaches  $S/(2\sigma^2)$  and the second approaches  $S/(\pi\sigma^2)$  nats.
- 4.23. Verify the values of  $\underline{C}$  and  $\bar{C}$  given for the channels in Figures 4.6.3 to 4.6.5.
- 4.24. Find an example of a 2-state binary channel with only intersymbol interference memory for which the first 3 inequalities in (4.6.11) are satisfied with equality for all positive  $N$ .

- 4.25.** Consider a semi-infinite cascade of discrete channels as shown below where for each positive  $n$ ,  $x_n$  is the input to the  $n$ th channel and the output from the  $(n - 1)$ st. Assume that the noise on each channel is statistically independent.

- (a) Show that the sequence  $x_1, x_2, \dots$ , forms an inhomogeneous Markov chain.  
 (b) Assume that each channel has an input and output alphabet of size  $K$  and assume that there is some  $\delta > 0$  such that for all  $n$ , the  $n$ th channel in the cascade has at least one output letter, say  $x_{n+1}$ , such that  $P(x_{n+1} | x_n) \geq \delta$  for all choices of input  $x_n$ . Show that  $I(X_1; X_n)$  approaches zero with increasing  $n$  and is upper bounded by  $2K(1 - \delta)^n$ .

*Hint:* Use Lemma 4.6.2.



- 4.26.** Suppose a finite state channel with only intersymbol interference memory (i.e.,  $s_n$  is a function of  $s_{n-1}$  and  $x_n$ ) can be driven into a known state. Show that it can be driven into such a state with at most  $2^A - A$  inputs where  $A$  is the number of states. *Hint:* For a sequence of inputs that drives the channel into a known state, let  $B_n$  be the set of possible states for the channel after the  $n$ th input. If the sequence  $B_n$  has any repetitions before arriving at the known state, show that the driving sequence can be shortened.

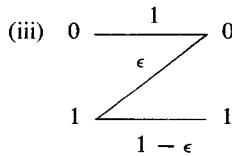
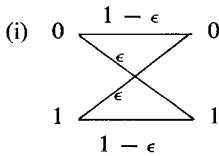
## CHAPTER 5

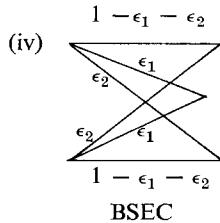
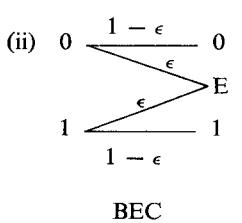
- 5.1.** Suppose that the integers from 1 to  $M$  are encoded into sequences of channel inputs  $x_1$  to  $x_M$ . Let  $y$  be a sequence of channel outputs and let  $P_N(y | x_m)$ ,  $1 \leq m \leq M$ , be given. If the cost of decoding a transmitted message  $m$  as  $m'$  is given by  $C(m, m')$ , and a probability assignment  $Q(m)$  is given on the (input) integers from 1 to  $M$ , derive the decoding rule that yields minimum average cost.

- 5.2.** (a) Compute

$$g_n(s) = \sum_{y_n} P(y_n | 0)^{1-s} P(y_n | 1)^s$$

for each of the following channels and for each minimize  $g_n(s)$  over  $0 \leq s \leq 1$ . Use the result to provide an upper bound to the probability of maximum likelihood decoding error  $P_{e,m}$  ( $m = 1, 2$ ) attained with a code of two code words  $x_1$  (a sequence of  $N$  zeros) and  $x_2$  (a sequence of  $N$  ones).





(b) For the first three channels above find exact expressions for the probability of error. Evaluate the expressions to 2 significant figures (using the Stirling approximation  $N! \approx \sqrt{2\pi N} (N/e)^N$  where necessary) for  $N = 25$  and  $\epsilon = 0.1$  and compare with the bound in (a).

(c) For the BSC, show that for large even  $N$ ,  $P_{e,1}$  and  $P_{e,2}$  [see (5.3.14) and (5.3.15)] are given approximately by

$$P_{e,1} \approx \sqrt{\frac{2}{\pi N}} \left( \frac{1-\epsilon}{1-2\epsilon} \right) [2\sqrt{\epsilon(1-\epsilon)}]^N; \quad P_{e,2} \approx P_{e,1} \left( \frac{\epsilon}{1-\epsilon} \right)$$

Show that for large odd  $N$ ,

$$P_{e,1} = P_{e,2} \approx \sqrt{\frac{2\epsilon}{\pi N(1-\epsilon)}} \left( \frac{1-\epsilon}{1-2\epsilon} \right) [2\sqrt{\epsilon(1-\epsilon)}]^N$$

*Hint:* Use the fact that in vicinity of  $i = N/2$ ,  $\binom{N}{i}$  is varying slowly relative to  $\epsilon^i(1-\epsilon)^{N-i}$ .

(d) For the Z-channel, both upper-bound and evaluate  $P_{e,m}$  ( $m = 1, 2$ ) for a two-word code in which  $\mathbf{x}_1$  is  $N/2$  zeros followed by  $N/2$  ones and  $\mathbf{x}_2$  is  $N/2$  ones followed by  $N/2$  zeros. Observe that for the other channels, this change of code will make no difference.

- 5.3. (a) Evaluate the bound on average error probability over an ensemble of codes with two code words given in (5.5.10) for the channels in Problem 5.2. Minimize the bounds over  $Q(k)$ .

(b) For the same ensemble and the same channels, find an upper bound on  $1/N \ln \overline{P_{e,m}}$ , i.e., the average exponent to error probability using  $Q(0) = Q(1) = \frac{1}{2}$  and interpret your result as a bound on error probability for a typical code in the ensemble.

- 5.4. Given  $M$  messages to be encoded into sequences of binary digits of length  $N$ , select the particular  $M$  sequences to be used from among the  $2^N$  possibilities with equal probability and statistical independence. Since any error that occurs in the maximum likelihood decoding of a received sequence occurs because one or more of the  $M - 1$  incorrect messages are more likely than the correct one, use your bound on the average probability of error in the two randomly selected code word case, and the fact that the probability of realizing one or more events is equal to or less than the sum of their probabilities, to bound the error probability for  $M$  messages for

each of the four channels in Problem 5.2. What is the maximum transmission rate  $R = (\ln M)/N$  for which your bound is exponentially decreasing with  $N$ ?

**5.5.** Let

$$z = \sum_{n=1}^N x_n$$

where the  $x_n$  are identically distributed zero mean random variables with unit variance.

(a) Let the  $x_n$  be Gaussian random variables and estimate  $\Pr(z \geq N)$  (use the estimate

$$\int_{y=W}^{\infty} \frac{1}{\sqrt{2\pi}\sigma} e^{-y^2/2\sigma^2} dy \approx \frac{\sigma}{\sqrt{2\pi}W} e^{-W^2/2\sigma^2}$$

for  $W$  large). Compare this with the Chernoff bound and the ordinary Chebyshev bound.

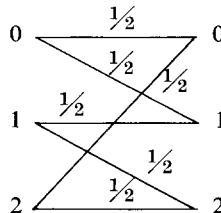
(b) Let  $x_n = 1$  with probability  $\frac{1}{2}$  and  $-1$  with probability  $\frac{1}{2}$ . Show that for  $N = 100$ ,  $\Pr(z \geq N) = 2^{-100}$ . Compare this with the Chernoff bound to  $\Pr(z \geq N)$ , the Chebyshev inequality, and the central limit theorem approximation. (This provides an extreme example of why the word *central* is in central limit theorem.)

**5.6.** Prove that (5.6.10) is minimized over  $s > 0$  by choosing  $s = 1/(1 + \rho)$

*Hint:* Use Holder's inequality (Problem 4.15) to show that

$$\left[ \sum_{\mathbf{x}} Q_N(\mathbf{x}) P_N(\mathbf{y} \mid \mathbf{x})^{\frac{1}{1+\rho}} \right]^{1+\rho} \leq \left[ \sum_{\mathbf{x}_m} Q_N(\mathbf{x}_m) P_N(\mathbf{y} \mid \mathbf{x}_m)^{1-s\rho} \right] \times \left[ \sum_{\mathbf{x}} Q_N(\mathbf{x}) P_N(\mathbf{y} \mid \mathbf{x})^s \right]^{\rho}$$

**5.7.** Suppose that two code words of length  $N$  are chosen for the channel below by selecting each letter of each code word independently with the probability assignment  $Q(0) = Q(1) = Q(2) = \frac{1}{3}$ .



(a) Find the ensemble average probability of error, assuming maximum likelihood decoding. In case of ambiguity, assume that the decoder makes an error. Do not use the coding theorem here; start from the beginning and make use of the special properties of the channel. *Hint:* Let  $\mathbf{x}_1$  be transmitted and  $\mathbf{y}$  received. Find how many channel input sequences can lead to  $\mathbf{y}$  and

find  $P_N(\mathbf{y} \mid \mathbf{x}_1)$  for each of them. Then find the probability that code word  $\mathbf{x}_2$  has been chosen in this set.

(b) Use your result in (a) to upper-bound the error probability for a code with  $M$  words chosen independently from the same ensemble. Use a union bound over the set of incorrect messages. Compare your answer with that given by the coding theorem.

(c) Suppose that a code consists of the two code words  $\mathbf{x}_1 = (0, 0, 0, \dots, 0)$  and  $\mathbf{x}_2 = (1, 1, 1, \dots, 1)$ . Find the probability of error for this code, using the same decoding procedure as in (a). Explain the discrepancy between your answers in (a) and (c).

- 5.8.** In dealing with binary channels it is often useful to have tight bounds on the binomial coefficient  $\binom{N}{j}$ .

(a) Show that for  $j \geq 1$ ,  $N - j \geq 1$

$$\sqrt{\frac{N}{8j(N-j)}} \leq \binom{N}{j} e^{-N\mathcal{H}(j/N)} < \sqrt{\frac{N}{2\pi j(N-j)}}$$

where

$$\mathcal{H}(j/N) = -\frac{j}{N} \ln \frac{j}{N} - \left(1 - \frac{j}{N}\right) \ln \left(1 - \frac{j}{N}\right)$$

*Hint:* Use the Stirling formula [Feller (1950), 3rd ed., p. 53]

$$N! = \sqrt{2\pi N} \left(\frac{N}{e}\right)^N \exp(\epsilon_N)$$

where  $\epsilon_N$  is decreasing with  $N$  and satisfies  $0 < \epsilon_N < \frac{1}{12N}$ . The lower bound must be verified by direct calculation when the larger of  $j$  and  $N - j$  is two or less. Note that in the limit of large  $j$  and  $N - j$ , the upper bound approaches equality.

(b) Using (a), establish the weaker bounds

$$\begin{aligned} \sqrt{\frac{1}{2N}} &\leq \binom{N}{j} e^{-N\mathcal{H}(j/N)} < 1 \\ \binom{2N-1}{N} &\geq \frac{1}{\sqrt{4N}} 2^{2N-1} \end{aligned}$$

*Hint:* On the latter inequality, show that  $\binom{2N}{N} = 2\binom{2N-1}{N}$ .

(c) Let  $\epsilon$  be arbitrary,  $0 < \epsilon < 1$ ,  $j < N$ , and  $j/N > \epsilon$ . Show that

$$\begin{aligned} \binom{N}{j} \epsilon^j (1-\epsilon)^{N-j} &\leq \sum_{n=j}^N \binom{N}{n} \epsilon^n (1-\epsilon)^{N-n} \\ &\leq \frac{j(1-\epsilon)}{j(1-\epsilon) - (N-j)\epsilon} \binom{N}{j} \epsilon^j (1-\epsilon)^{N-j} \end{aligned}$$

*Hint:* For the upper bound, show that

$$\binom{N}{n+1} < \binom{N}{n} \left(\frac{N-n}{n}\right)$$

and use this to show that

$$\binom{N}{j+m} < \binom{N}{j} \left(\frac{N-j}{j}\right)^m$$

Then sum over  $j$  by using a geometric series. Combine this result with that of (a) to obtain upper and lower bounds to

$$\sum_{n=j}^N \binom{N}{n} \epsilon^n (1-\epsilon)^{N-n}$$

(d) Let

$$w = \sum_{n=1}^N x_n$$

where the  $x_n$  are statistically independent random variables taking on the value 1 with probability  $\epsilon$  and 0 with probability  $1 - \epsilon$ . Show that

$$P_N[w \geq j] = \sum_{n=j}^N \binom{N}{j} \epsilon^j (1-\epsilon)^{N-j}$$

and use the Chernoff bound to show that

$$\Pr[w \geq j] \leq \exp \left\{ N \left[ \mathcal{H} \left( \frac{j}{N} \right) + \frac{j}{N} \ln \epsilon + \frac{N-j}{N} \ln (1-\epsilon) \right] \right\}$$

Compare with your result in (c).

- 5.9. (Coding theorem for BSC using binomial coefficients.) For a BSC with crossover probability  $\epsilon$ , consider an ensemble of codes of block length  $N$  with  $M = e^{NR}$  code words, with each letter of each code word independently chosen with  $Q(0) = Q(1) = \frac{1}{2}$ . For any code in the ensemble consider a decoder which, given  $y$ , chooses the message  $m$  for which  $d(x_m, y)$  is minimum where  $d(x_m, y)$  is the Hamming distance between  $x_m$  and  $y$  (i.e., the number of digits in which  $x_m$  and  $y$  differ).

(a) Show that this is a maximum likelihood decoder.

(b) Given that message  $m$  enters the encoder, show that

$$\Pr[d(x_m, y) = i] = \binom{N}{i} \epsilon^i (1-\epsilon)^{N-i}$$

$$\Pr[d(x_{m'}, y) = i] = \binom{N}{i} 2^{-N}; \quad \text{each } m' \neq m$$

where the second probability is over the ensemble of codes.

(c) Given message  $m$ , show that

$$\Pr[\text{error} \mid d(\mathbf{x}_m, \mathbf{y}) = i] \leq \begin{cases} (M - 1) \sum_{n=0}^i \binom{N}{n} 2^{-N} \\ 1 \end{cases}$$

$$\leq \begin{cases} \frac{N-i}{N-2i} \binom{N}{i} \exp[-N(\ln 2 - R)] & i < N/2 \\ 1 & i \geq N/2 \end{cases}$$

*Hint:* Use your result in Problem 5.7c, with  $\epsilon = \frac{1}{2}$  and  $i = N - j$ .

(d) Using (b) and (c), show that the ensemble average error probability is bounded for any  $j \leq N/2$  by

$$P_e \leq \sum_{i=0}^{j-1} \frac{N-i}{N-2i} \exp[-N(\ln 2 - R)] \binom{N}{i}^2 \epsilon^i (1-\epsilon)^{N-i}$$

$$+ \sum_{i=j}^N \binom{N}{i} \epsilon^i (1-\epsilon)^{N-i}$$

(e) Choose  $j$  to satisfy

$$\mathcal{H}\left(\frac{j-1}{N}\right) < \ln 2 - R \leq \mathcal{H}\left(\frac{j}{N}\right)$$

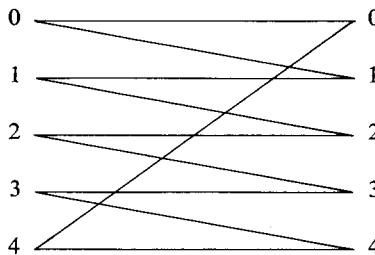
and upper-bound each of these sums either by the geometric series technique of Problem 5.7c or by upper bounding by the largest term times the number of terms in the sum. Be careful about whether

$$\left(\frac{j}{N-j}\right)^2 < \frac{\epsilon}{1-\epsilon}$$

or not. Show that you obtain the same exponent in  $N$  as in Example 1 of Section 5.6.

- 5.10.** Find the random coding exponent  $E_r(R)$  (in parametric form) for a binary erasure channel with erasure probability  $\epsilon$ . Sketch  $E_r(R)$  for  $\epsilon = \frac{1}{2}$ , labeling with the specific numerical values of  $E_r(0)$ ,  $R_{cr} = E_0(\rho)/\partial\rho|_{\rho=1}$ , and  $E_r(R_{cr})$ . Find a graphical interpretation of  $E_r(R)$  similar to that of Fig. 5.6.4 for the BSC.
- 5.11.** (a) Find the random coding exponent  $E_r(R)$  for the 5-input, 5-output channel below where all transition probabilities are  $\frac{1}{2}$ .
- (b) Find a code of block length 1 with  $R = \ln 2$  (i.e., one binary digit per channel use) such that  $P_e = 0$ . Find a code of block length 2 with  $R = (\ln 5)/2$  (i.e., 5 code words) such that  $P_e = 0$ . *Note:* The zero error capacity,  $C_0$ , of a channel is defined as the largest rate for which  $P_e = 0$  can be achieved with finite block length [Shannon (1956)]. Thus you have shown that

$C_0 \geq (\ln 5)/2$  for this channel. It is unknown whether or not  $C_0$  is strictly greater than  $(\ln 5)/2$ .



- 5.12.** Let  $\mathbf{Q}$  be the input probability assignment that achieves capacity in a discrete memoryless channel. Show that for  $R$  close to  $C$ ,  $E_r(R, \mathbf{Q})$  is approximated by  $\alpha(C - R)^2$  and evaluate the constant  $\alpha$ .
- 5.13.** Show that for an arbitrary DMC with a binary input, the function  $E_0(\rho, \mathbf{Q})$  is maximized over  $\mathbf{Q}$  at  $\rho = 1$  by  $Q(0) = Q(1) = \frac{1}{2}$ .
- 5.14.** (Coding with erasures and errors.) Consider a DMC with transition probabilities  $P(j | k)$ , let  $Q(k)$  denote the input probabilities that achieve capacity, and let

$$\omega(j) = \sum_k Q(k)P(j | k)$$

Consider an ensemble of codes of block length  $N$  with  $M = e^{NR}$  code words in which each letter of each code word is independently chosen with the probability assignment  $Q(k)$ .

Consider a decoder that operates as follows. Given the received sequence  $\mathbf{y}$ , the decoder computes

$$I_m = \ln \frac{P_N(\mathbf{y} | \mathbf{x}_m)}{\omega_N(\mathbf{y})} = \sum_{n=1}^N \ln \frac{P(y_n | x_{m,n})}{\omega(y_n)}$$

for each code word  $\mathbf{x}_m$ ,  $1 \leq m \leq M$ . The decoder compares each  $I_m$  with  $TN$ , where  $T$  is a fixed threshold. If there is one and only one value of  $m$  for which  $I_m \geq TN$ , the decoder decodes that message. Otherwise the decoder produces an erasure symbol and does not decode to any message.

(a) Let  $\bar{P}_1$  be the probability, over the ensemble of codes, that the transmitted code word does not satisfy the threshold and let  $\bar{P}_2$  be the probability that one or more of the other code words do satisfy the threshold.

(b) Use the Chernov bound to show that

$$\bar{P}_1 \leq \exp[-N\alpha]$$

$$\bar{P}_2 \leq \exp[-N(\alpha + T - R)]$$

$$\alpha = \max_{0 \leq s \leq 1} -\ln \left[ \sum_{j,k} Q(k)\omega(j)^s P(j | k)^{1-s} e^{sT} \right]$$

(c) Show that  $\alpha > 0$  for  $T < C$ , where  $C$  is the channel capacity in nats, and that  $\alpha \rightarrow 0$  as  $T \rightarrow C$ .

- (d) Show that  $\bar{P}_1 + \bar{P}_2$  upper bounds the probability of a decoding erasure and that  $\bar{P}_2$  is an upper bound on the probability of decoding error. Sketch  $\alpha + T - R$  as a function of  $R$  in the limit  $T \rightarrow C$  and compare with the random coding exponent. Point out the implications of this for a channel where a noiseless “feedback” link is available from receiver to transmitter. For a similar, but much stronger, bound on the probability of erasures and errors, see Forney (1968).
- 5.15.** In the previous problem we observe that the probability of not decoding correctly (i.e., making either an error or an erasure) is upper-bounded by  $\bar{P}_e \leq \bar{P}_1 + \bar{P}_2$ ; and if we set  $T = R$ , then  $\bar{P}_e \leq 2 \exp(-N\alpha)$  where

$$\alpha = \max_{0 \leq s \leq 1} \left\{ -sR - \ln \left[ \sum_{j,k} Q(k) \omega(j)^s P(j | k)^{1-s} \right] \right\} \quad (1)$$

Note that since  $\alpha > 0$  for  $R < C$ , this provides a very simple proof of the coding theorem (although not with the best error exponent).

(a) Replace  $s$  by  $\rho/(1+\rho)$  and show that for the binary symmetric channel, (1) reduces to

$$\alpha = \max_{\rho \geq 0} \frac{1}{1+\rho} [E_0(\rho) - \rho R]$$

Compare  $\alpha$  to  $E_r(R)$  by the graphical technique of Fig. 5.6.3.

(b) By using Holder's inequality (Problem 4.15c) on the sum over  $j$  in (1), show that for a general DMC,

$$\alpha \geq \max_{\rho \geq 0} \frac{1}{1+\rho} [E_0(\rho) - \rho R]$$

(c) Setting  $s = \rho$  in (1) and using Holder's inequality on

$$\sum_k Q(k) P(j | k)^{1/(1+\rho)}$$

show that

$$\alpha \leq E_r(R)$$

- 5.16. Joint source and channel coding theorem.**

(a) Let  $P_N(y | x)$  be the transition probability assignment for sequences of length  $N$  on a discrete channel and consider an ensemble of codes, in which  $M$  code words are independently chosen, each with a probability assignment  $Q_N(x)$ . Let the messages encoded into these code words have a probability assignment  $q_m$ ,  $1 \leq m \leq M$ , and consider a maximum a-posteriori probability decoder, which, given  $y$ , chooses the  $m$  that maximizes  $q_m P_N(y | x_m)$ . Let

$$\bar{P}_e = \sum_m q_m \bar{P}_{e,m}$$

be the average error probability over this ensemble of messages and codes, and by modifying the proof of Theorem 5.6.1 where necessary, show that

$$\bar{P}_e \leq \left[ \sum_{m=1}^M q_m^{1/(1+\rho)} \right]^{1+\rho} \sum_{\mathbf{y}} \left[ \sum_{\mathbf{x}} Q_N(\mathbf{x}) P_N(\mathbf{y} \mid \mathbf{x})^{1/(1+\rho)} \right]^{1+\rho} \quad (1)$$

(b) Let the channel be memoryless with transition probabilities  $P(j \mid k)$ , let the letters of the code words be independently chosen with probability assignment  $Q(k)$ , and let the messages be sequences of length  $L$  from a discrete memoryless source  $U$  with probability assignment  $\Pi(i)$ ,  $0 \leq i \leq A-1$ . Show that (1) is equivalent to

$$\bar{P}_e \leq \exp \{ -NE_0(\rho, \mathbf{Q}) + LE_s(\rho) \} \quad (2)$$

$$E_s(\rho) = (1 + \rho) \ln \left[ \sum_{i=0}^{A-1} \Pi(i)^{1/(1+\rho)} \right]$$

(c) Show that  $E_s(0) = 0$ ,

$$\frac{\partial E_s(\rho)}{\partial \rho} \Big|_{\rho=0} = H(U) \text{ (in nats)}$$

and that  $E_s(\rho)$  is strictly increasing in  $\rho$  (if no  $\Pi(i) = 1$ ).

(d) Let  $\lambda = L/N$ , and let  $N \rightarrow \infty$  with  $\lambda$  fixed. Show that  $\bar{P}_e \rightarrow 0$  if  $\lambda H(U) < C$  and if  $Q(k)$  is appropriately chosen.

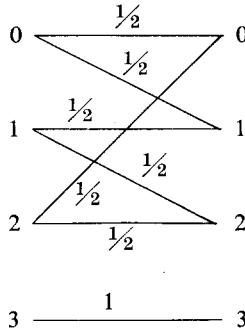
(e) Show that (2) is equivalent to (5.6.13) in the case where  $\Pi(i) = 1/A$  for  $0 \leq i \leq A-1$  and is equivalent to the positive part of the source coding Theorem 3.1.1 (except for the exponential convergence here) if the channel is noiseless. The above results are due to the author and were first used in a take-home quiz in 1964.

- 5.17.** Consider the sum channel (as in Problem 4.18) associated with a set of  $n$  DMC's. Let  $E_{0,i}(\rho) = \max_Q E_{0,i}(\rho, \mathbf{Q})$  for the  $i$ th of the set of DMC's and let  $E_0(\rho) = \max_Q E_0(\rho, \mathbf{Q})$  for the sum channel. Let  $q(i)$  be the maximizing probability of using the  $i$ th channel and let  $Q_i(k)$  be the probability of using input  $k$  on the  $i$ th channel, given that the  $i$ th channel is to be used. Show that

$$\exp \left[ \frac{E_0(\rho)}{\rho} \right] = \sum_{i=1}^n \exp \left[ \frac{E_{0,i}(\rho)}{\rho} \right]$$

$$q(i) = \frac{\exp \left[ \frac{E_{0,i}(\rho)}{\rho} \right]}{\sum_{i=1}^n \exp \left[ \frac{E_{0,i}(\rho)}{\rho} \right]}$$

Apply your result to find  $E_0(\rho)$  for the channel below



- 5.18.** The following proof of the coding theorem fails to achieve the tight bound on error probability found in Section 5.6, but brings out more clearly the significance of channel capacity. Let  $P(j | k)$  denote the transition probabilities for a DMC, let  $Q(k)$  be an input probability assignment that achieves capacity, and let

$$\omega(j) = \sum_k Q(k)p(j | k)$$

For any block length  $N$ , let

$$P_N(y | x) = \prod_{n=1}^N P(y_n | x_n), \quad Q_N(x) = \prod_n Q(x_n), \quad \omega_N(y) = \prod_n \omega(y_n)$$

Let  $R$  be an arbitrary rate,  $R < C$ , and for each  $N$  consider choosing a code of  $M = [e^{NR}]$  code words, choosing each word independently with the probability assignment  $Q_N(x)$ . Let  $\epsilon = (C - R)/2$ , and for each  $N$  define the “typical” set  $T_N$  as the set of  $x, y$  pairs for which

$$\left| \frac{1}{N} I(x,y) - C \right| \leq \epsilon$$

where

$$I(x,y) = \ln \frac{P_N(y | x)}{\omega_N(y)}$$

For each  $N$  and each code in the ensemble, consider a decoder which, given  $y$ , selects the  $m$  for which  $I(x_m, y)$  is in  $T_N$ . If there are no such code words or more than 1 such code word, we assume an error.

(a) Show that for given  $N$ , the probability of error, given that message  $m$  enters the encoder, satisfies

$$\bar{P}_{e,m} \leq \Pr[(x_m, y) \notin T_N | m] + \sum_{m' \neq m} \Pr[(x_{m'}, y) \in T_N | m]$$

(b) Show that over the ensemble of codes,  $\Pr[(x_m, y) \notin T_N | m]$  approaches zero as  $N$  approaches  $\infty$ . Hint: Regard  $I(x_m, y)$  as a sum of independent identically distributed variables and use the law of large numbers.

- (c) Show that over the ensemble of codes, for any  $m' \neq m$ ,

$$\Pr[(\mathbf{x}_{m'}, \mathbf{y}) \in T_N \mid m] \leq \exp[-N(C - \epsilon)]$$

*Hint:* Show that

$$\Pr[(\mathbf{x}_{m'}, \mathbf{y}) \in T_N \mid m] = \sum_{(\mathbf{x}_{m'}, \mathbf{y}) \in T_N} [Q_N(\mathbf{x}_{m'}) \omega_N(\mathbf{y})]$$

Then show that for  $(\mathbf{x}_{m'}, \mathbf{y}) \in T_N$

$$\omega_N(\mathbf{y}) \leq P_N(\mathbf{y} \mid \mathbf{x}_{m'}) \exp[-N(c - \epsilon)]$$

- (d) Combining (a), (b), and (c), show that  $\bar{P}_{e,m}$  approaches zero for each  $m$  as  $N$  approaches  $\infty$ .

- 5.19.** Let  $\mathbf{x}_1, \dots, \mathbf{x}_M$  be a set of code words of block length  $N$  for use on a BSC with crossover probability  $\epsilon$ . Let  $d(\mathbf{x}_m, \mathbf{x}_{m'})$  be the Hamming distance between  $\mathbf{x}_m$  and  $\mathbf{x}_{m'}$  (i.e., the number of positions in which  $\mathbf{x}_m$  differs from  $\mathbf{x}_{m'}$ ).

- (a) By using the Chernoff bound and the union bound, show that for a maximum likelihood decoder,

$$P_{e,m} \leq \sum_{m' \neq m} \exp[d(\mathbf{x}_m, \mathbf{x}_{m'}) \ln \sqrt{(2\epsilon(1-\epsilon))}]$$

- (b) The minimum distance of a code is defined as

$$d_{\min} = \min_{m \neq m'} d(\mathbf{x}_m, \mathbf{x}_{m'})$$

Show that, for all  $m$ ,

$$P_{e,m} \leq (M-1) \exp[d_{\min} \ln \sqrt{(2\epsilon(1-\epsilon))}]$$

- (c) Consider choosing a code with a desired minimum distance  $d_{\min} = d$  by the following procedure:

- (i) List all  $2^N$  distinct binary  $N$ -tuples.
- (ii) Choose an arbitrary code word from the list.
- (iii) Delete from the list the code word just selected and all  $N$ -tuples distance  $d-1$  or less from that code word. If the list is empty, stop, and otherwise go back to step (ii).

Show that the number of code words so selected satisfies

$$M \geq 2^N \left[ \sum_{i=0}^{d-1} \binom{N}{i} \right]^{-1}$$

This is known as the Gilbert (1952) bound.

- (d) By combining the above result with Problem 5.8, show that for  $d \leq N/2$ ,

$$M \geq \exp \left\{ N \left[ \ln 2 - \mathcal{H} \left( \frac{d_{\min} - 1}{N} \right) \right] \right\}$$

Show from this that for any rate  $R = (\ln M)/N < \ln 2$ , there exists a code of minimum distance  $d_{\min} \geq \delta N$  where  $\delta < \frac{1}{2}$  is defined by  $\mathcal{H}(\delta) = \ln 2 - R$ .

- (e) Combine (b) and (d) to obtain a bound on error probability of the form  $P_e \leq \exp[-NE(R)]$  and give parametric expressions defining  $E(R)$ . Show that at zero rate this bound agrees with (5.5.1). Sketch  $E(R)$  for  $\epsilon = 0.01$  and compare with  $E_r(R)$ . [Note that  $E(R) < E_{ex}(R)$  as developed in Section 5.7.]
- 5.20.** Suppose that the conditions of Theorem 5.6.1 are modified in that the decoder, given a received sequence  $\mathbf{y}$ , produces a list of the  $L$  messages  $m_1$  to  $m_L$  for which  $P_N(\mathbf{y} | \mathbf{x}_m)$  is largest, where  $L$  is a given integer. If the transmitted message is not on the list, we say that a list-decoding error has occurred. Let  $\bar{P}_{L,e}$  be the probability of a list-decoding error
- Show that for any  $\rho_0$ ,  $0 \leq \rho_0 \leq 1$ , and any  $s > 0$
- $$\Pr[\text{list error} | m, \mathbf{x}_m, \mathbf{y}] \leq \binom{M-1}{L}^{\rho_0} \left\{ \sum_x Q_N(x) \left[ \frac{P_N(\mathbf{y} | \mathbf{x})}{P_N(\mathbf{y} | \mathbf{x}_m)} \right]^s \right\}^{L\rho_0}$$
- Upper-bound  $\binom{M-1}{L}$  by  $(M-1)^L$ , let  $\rho = L\rho_0$ , and show that for  $0 \leq \rho \leq L$ ,
- $$\bar{P}_{L,e} \leq (M-1)^\rho \sum_y \left( \sum_x Q_N(x) P_N(y | x)^{\frac{1}{1+\rho}} \right)^{1+\rho}$$
- Show that for a DMC, this becomes
- $$\bar{P}_{L,e} \leq \exp \left[ -N \left\{ \max_{0 \leq p \leq L} [-\rho R + E_0(\rho, \mathbf{Q})] \right\} \right]$$
- Sketch the term in braces as a function of  $R$  for given  $L$  and compare with  $E_r(R, \mathbf{Q})$ .
- 5.21.** Consider the following alternatives for transmitting data from point  $A$  to point  $C$  in the diagram below.
- At point  $B$ , each output of the first channel is directly connected to the corresponding input of the second channel, and data are transmitted by using a block code of length  $N$  and rate  $R$ .
  - A block code of length  $N/2$  and rate  $R$  is used on the first channel, and the data are decoded at point  $B$  and re-encoded into a block code of length  $N/2$  and rate  $R$  for the second channel, using only the outer two inputs.
  - The binary source digits are directly transmitted from  $A$  to  $C$  with corresponding inputs and outputs connected at point  $B$ . By means of a noiseless return channel, the source digit is repeated whenever the middle output is received at  $C$ .
    - Evaluate the random coding exponents for (i) and (ii) and compare these alternatives on that basis.
    - Use the Chernoff bound to upper-bound the probability that fewer than  $RN$  source digits will be transmitted at the end of  $N$  channel uses using

alternative 3, and compare with the first two alternatives.



- 5.22. A discrete memoryless channel has the transition probabilities  $P(j|k)$ . Unfortunately, the decoder for the channel is a maximum likelihood decoder designed under the mistaken impression that the transition probabilities are  $P'(j|k)$ . That is, message  $m$  is decoded if  $P_N'(\mathbf{y}|\mathbf{x}_m) > P_N'(\mathbf{y}|\mathbf{x}_{m'})$  for all  $m' > m$ , where

$$P_N'(\mathbf{y}|\mathbf{x}_m) = \prod_n P'(y_n|x_{mn})$$

Find an upper bound to the average probability of decoding error, over an ensemble of codes where each letter of each code word is independently selected with a probability assignment  $Q(k)$ . Your bound should have the form

$$\bar{P}_e \leq \exp \{-N[-\rho R + f(\rho, \mathbf{Q}, \mathbf{P}, \mathbf{P}')]\} \quad \text{for } 0 \leq \rho \leq 1$$

Find an example of transition probabilities  $\mathbf{P}$  and  $\mathbf{P}'$  where  $f$  is not positive and explain why this is not surprising. See Stiglitz (1966).

- 5.23. Given a DMC with transition probabilities  $P(j|k)$ , expand  $E_0(\rho, \mathbf{Q})$  for the  $\mathbf{Q}$  that yields capacity in a power series to show that

$$\bar{P}_e \leq \exp - N \left[ \frac{(C - R)^2}{2\alpha} \right] \quad (\text{i})$$

for  $C - R \leq \alpha$  where  $\alpha$  is an upper bound to  $-E_0''(\rho, \mathbf{Q})$  for  $0 \leq \rho \leq 1$ . Show that

$$-E_0'(\rho, \mathbf{Q}) = \sum_j \omega_j \sum_k q_{kj} \ln \left[ \frac{Q(k)}{P(j|k)^{1/(1+\rho)}} \right] \quad (\text{ii})$$

where

$$\omega_j = \frac{\alpha_j^{1+\rho}}{\sum_j \alpha_j^{1+\rho}} ; \quad q_{kj} = Q(k) \frac{P(j|k)^{1/(1+\rho)}}{\alpha_j}$$

$$\alpha_j = \sum_k Q(k) P(j|k)^{1/(1+\rho)}$$

By taking the derivative of (ii), show that

$$-E_0''(\rho) \leq \sum_j \omega_j \sum_k q_{kj} \left( \ln \frac{Q(k)}{q_{kj}} \right)^2$$

From this, show that

$$\bar{P}_e \leq \exp \left\{ -N \left[ \frac{(C - R)^2}{8/e^2 + 4[\ln J]^2} \right] \right\}$$

where  $J$  is the size of the output alphabet.

- 5.24.** Consider a DMC for which the zero error capacity is zero [i.e.,  $R_{x,\infty}$ , as given by (5.7.16), is zero]. Show that

$$\lim_{R \rightarrow 0} E_{ex}(R, \mathbf{Q}) = - \sum_{k,i} Q(k)Q(i) \ln \left[ \sum_j \sqrt{P(j|k)P(j|i)} \right]$$

*Hint:* First show that  $\lim_{R \rightarrow 0} E_{ex}(R, \mathbf{Q}) = \lim_{\rho \rightarrow \infty} E_x(\rho, \mathbf{Q})$ . Then either use L'Hospital's rule or expand  $E_x(\rho, \mathbf{Q})$  as a power series in  $1/\rho$ .

- 5.25.** Show that  $R_{x,\infty}$  [see (5.7.16)] is given by  $R_{x,\infty} = \ln L$ , where  $L$  is the size of the largest set  $I$  of integers such that for all  $i \in I$ ,  $k \in I$ ,  $i \neq k$ , we have  $\varphi_{k,i} = 0$  [see (5.7.15)]. (In other words,  $I$  is the largest set of inputs such that no output can be reached from more than one input in the set; when only this set of inputs is used, each output letter uniquely specifies the input.)

*Hint:* Assume  $\varphi_{0,1} = 1$  and show that

$$\begin{aligned} \sum_{k,i} Q(k)Q(i)\varphi_{k,i} &= [Q(0) + Q(1)]^2 + 2 \sum_{i=2}^{K-1} Q(i)[Q(0)\varphi_{0,i} + Q(1)\varphi_{1,i}] \\ &\quad + \sum_{k=2}^{K-1} \sum_{i=2}^{K-1} Q(k)Q(i)\varphi_{k,i} \end{aligned}$$

Show that for any fixed set of values for  $Q(2), \dots, Q(K-1)$ , this is minimized over  $Q(0)$  and  $Q(1)$  by either  $Q(0) = 0$  or  $Q(1) = 0$ . Use this to show that

$$\sum_{k,i} Q(k)Q(i)\varphi_{k,i}$$

is minimized by choosing  $Q(k)$  nonzero only on some set  $I$  for which  $\varphi_{k,i} = 0$  for all  $k \in I$ ,  $i \in I$ ,  $k \neq i$ .

- 5.26.** For the channel of Problem 5.11, choose a block length of  $N = 2$  and choose  $R = (\ln 5)/2$ . Show that the expression for  $P_{e,m}$  in (5.7.7) is not minimized over  $\rho$  and  $Q_N(\mathbf{x})$  by a product distribution. *Hint:* Look carefully at your solution to part *b* of Problem 5.11.

- 5.27.** Show that  $E_x(\rho, \mathbf{Q})$  is positive for all  $\rho > 0$  and that

$$\partial E_x(\rho, \mathbf{Q}) / \partial \rho \leq -\ln \sum_k [Q(k)]^2$$

- 5.28.** Calculate and sketch  $E_x(\rho, \mathbf{Q})$  and  $E_0(\rho, \mathbf{Q})$  for a noiseless binary channel with  $Q(0) = 0.1$  and  $Q(1) = 0.9$ . Sketch  $E_{ex}(R, \mathbf{Q})$  and  $E_r(R, \mathbf{Q})$ .

- 5.29.** Show that for any two input DMC,  $E_x(\rho, \mathbf{Q})$  is maximized over  $\mathbf{Q}$  by  $Q(0) = Q(1) = \frac{1}{2}$ . *Hint:* Observe that the terms in the double sum over  $k$  and  $i$  in the definition of  $E_x$  have only two different values, one for  $k = i$  and one for  $k \neq i$ . This result and the one in the following problem are due to Jelinek (1968).

- 5.30.** Show that if the matrix  $A$  with elements

$$a_{ik} = \left[ \sum_j \sqrt{P(j|k)P(j|i)} \right]^{1/\rho}$$

is nonnegative definite, then

$$\sum_{k,i} Q(k)Q(i) \left[ \sum_j \sqrt{P(j|k)P(j|i)} \right]^{1/\rho}$$

is convex  $\cup$  in  $\mathbf{Q}$ . Show that it also follows that

$$\left[ \sum_y \sqrt{P_N(y|x)P_N(y|x')} \right]^{1/\rho}$$

as a matrix with  $\mathbf{x}, \mathbf{x}'$  elements is also nonnegative definite and thus that

$$\sum_{\mathbf{x}, \mathbf{x}'} Q_N(\mathbf{x})Q_N(\mathbf{x}') \left[ \sum_y \sqrt{P_N(y|\mathbf{x})P_N(y|\mathbf{x}')} \right]^{1/\rho}$$

is convex  $\cup$  in  $\mathbf{Q}_N$ . Hint: Show that the eigenvalues of the latter matrix are products of  $N$  eigenvalues of the former matrix. Show from this that the above sum over  $\mathbf{x}, \mathbf{x}'$  is minimized by a product distribution. Hint: See Example 4 of Section 5.6.

- 5.31.** Show that in the limit of a very noisy channel,

$$\lim_{R \rightarrow 0} E_{ex}(R, \mathbf{Q})$$

[see (5.7.20)] approaches  $E_r(R, \mathbf{Q})$ . Use this to show that for such a channel,  $E_{ex}(R, \mathbf{Q})$  approaches  $E_r(R, \mathbf{Q})$  for  $R \leq R_{cr}$ .

- 5.32.** For any given code and decoding scheme, let  $P_{e,m}$  be the error probability for the  $m$ th code word and

$$P_{\max} = \max_{1 \leq m \leq M} P_{e,m}$$

Let  $P_{\max}(N, M)$  be the minimum value of  $P_{\max}$  for a given channel over all codes of block length  $N$  with  $M$  code words. Show that

$$P_{\max}(N, M) \geq P_e(N, M)$$

and that

$$P_{\max}(N, M) \leq 2P_e(N, 2M)$$

where  $P_e(N, M)$  is as defined in Section 5.8. Hint: See Corollary 2 to Theorem 5.6.2.

- 5.33.** Show that the sphere-packing bound for the BSC, (5.8.19), is still valid if feedback between receiver and transmitter is allowed (i.e., if the transmitter observes what is received and each transmitted digit is a function both of the message and of previous received digits.) Hint: Again let  $Y_m$  be the set of received messages decoded into message  $m$  and show that these sets are mutually exclusive. Let  $\mathbf{z}(\mathbf{y})$  be the error sequence for  $\mathbf{y}$ . That is, if  $\mathbf{y} \in Y_m$ , and  $\mathbf{x}_m(\mathbf{y})$  is the transmitted sequence for message  $m$  and received sequence  $\mathbf{y}$ , let  $\mathbf{z}(\mathbf{y}) = \mathbf{x}_m(\mathbf{y}) + \mathbf{y}$ . Show that for a given  $m$ , if  $\mathbf{y} \in Y_m$  and  $\mathbf{y}' \in Y_m$ , with  $\mathbf{y} \neq \mathbf{y}'$ , then  $\mathbf{z}(\mathbf{y}) \neq \mathbf{z}(\mathbf{y}')$ .

- 5.34.** Show that the parameter  $A$  in Theorem 5.8.5 satisfies

$$A \leq \frac{4}{e^2} + \frac{2(\ln J)(\ln J + 2/e)}{\min_k \max_j P(j | k)}$$

*Hint:* Observe that

$$\begin{aligned} A &\leq \max_k \sum_j P(j | k) \left[ \ln \frac{P(j | k)}{\omega(j)} \right]^2 \\ \sum_j P(j | k) \left[ \ln \frac{P(j | k)}{\omega(j)} \right]^2 &\leq \sum_{j: P(j|k) < \omega(j)} P(j | k) \left[ \ln \frac{\omega(j)}{P(j | k)} \right]^2 \\ &\quad + \sum_{j: P(j|k) > \omega(j)} P(j | k) \ln \frac{P(j | k)}{\omega(j)} \ln \frac{1}{\omega(j)} \end{aligned}$$

For the first sum, use the inequality

$$[\ln x]^2 \leq \frac{4}{e^2} x \quad \text{for } x \geq 1$$

For the second sum, use (5.8.60) to show that

$$\ln \frac{1}{\omega(j)} \leq \frac{C + H(Y | x = k)}{P(j | k)} \quad \text{for all } k$$

- 5.35.** Use the Chernoff bound in place of the Chebyshev inequality in (5.8.67) to show that for fixed  $R > C$ ,

$$P_e(N, [e^{NR}]) \geq 1 - 2 \exp[-N\alpha(R)]$$

when  $\alpha(R) > 0$  for  $R > C$ .

- 5.36.** Apply the Berry-Esseen form of the central limit theorem [see Feller (1966), p. 521] in place of the Chebyshev inequality in (5.8.67). Assume that the channel is such that for all  $k$ ,  $\ln [P(j | k)/\omega(j)]$  is not independent of  $j$ . Let  $\delta$  be an arbitrary real number (positive or negative) and let  $R(\delta, N) = C + \delta/\sqrt{N}$ . Show that for any  $\epsilon > 0$  there exists an  $N(\delta, \epsilon)$  such that for all  $N \geq N(\delta, \epsilon)$ ,

$$P_e(N, [\exp[NR(\delta, N)]] \geq f(\delta) - \epsilon$$

where  $f(\delta)$  is positive for all  $\delta$ , increasing in  $\delta$ , and  $f(0) = \frac{1}{2}$ .

- 5.37.** (a) Given a finite state channel, assume that the transmitter knows the initial state and uses a separate code for each initial state. Assuming the same decoder rule as in (5.9.1) to (5.9.5), show that the order of the min max in (5.9.5) can be interchanged for this situation [see Yudkin (1968)].

(b) Assume that the receiver knows the initial state and decodes the message  $m$  which maximizes  $P_N(y | x, s_0)$ . Show that (5.9.5) is again valid for this case and that the factor  $A^{1+\rho}$  can be omitted from the bound.

(c) For the channel in Fig. 4.6.3 show that the min max in (5.9.5) is achieved by an input distribution of equally likely independent inputs. Show

that the max min in (a) is achieved by the same distribution that achieves  $\bar{C}$  (see Section 4.6).

- 5.38.** Consider a binary input, binary output channel where the output letters  $y_n$  are related to the input  $x_n$  by  $y_n = x_n \oplus z_n$ . Assume the noise sequence  $z_1, z_2, \dots$ , is independent of the input and is the output from a Markov source with an ergodic state sequence (see Section 3.6).

(a) Show that the capacity of the channel is  $\log 2 - H_\infty(Z)$  when  $H_\infty(Z)$  is the entropy of the noise sequence [see (3.6.21)]. Show that this capacity is larger than that of a BSC with crossover probability  $P_{z_n}(1)$ .

(b) Consider an ensemble of codes with  $Q_N(\mathbf{x}) = 2^{-N}$  for all  $\mathbf{x}$ . Show that

$$E_{0,N}(\rho, \mathbf{Q}_N, s_0) = \rho \ln 2 - \frac{1 + \rho}{N} \ln \sum_z P_{z^N}(\mathbf{z} \mid s_0)^{1/(1+\rho)}$$

(c) Let the  $A$  by  $A$  matrix  $[\alpha(\rho)]$  have the components

$$\alpha_{li} = \sum_{z_n} P(z_n, s_n = i \mid s_{n-1} = l)^{1/(1+\rho)}$$

Let  $\lambda(\rho)$  be the largest eigenvalue of  $[\alpha(\rho)]$  and show that

$$\lim_{N \rightarrow \infty} E_{0,N}(\rho, \mathbf{Q}_N, s_0) = \rho \ln 2 - (1 + \rho) \ln \lambda(\rho)$$

- 5.39.** Consider the class of indecomposable finite state channels for which there is no intersymbol interference memory (i.e., the state sequence is independent of the input) and for which  $s_n$  is a function of  $y_n$  for each  $n$ . For example, the channel of Fig. 5.9.1 belongs to this class. Show that for any channel in this class, capacity is achieved by independent identically distributed inputs and that the capacity is equal to that of a DMC with

$$P(y_n \mid x_n) = \sum_{s_{n-1}} P(y_n \mid x_n, s_{n-1}) q(s_{n-1})$$

## CHAPTER 6

- 6.1.** A code maps pairs of information digits into code words of length 5 as follows

| <i>Information Sequences</i> | <i>Code Words</i> |
|------------------------------|-------------------|
| <b>00</b>                    | <b>00000</b>      |
| <b>01</b>                    | <b>01101</b>      |
| <b>10</b>                    | <b>10111</b>      |
| <b>11</b>                    | <b>11010</b>      |

(a) Show that the code is a systematic parity check code and express each digit in the code word as a linear combination of the information digits.

(b) Find the generator matrix and a parity check matrix for the code.

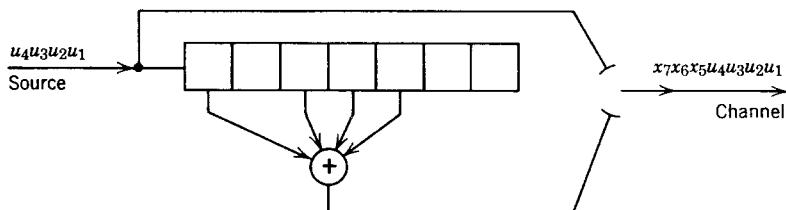
(c) Give a decoding table that yields maximum likelihood decoding for a BSC with crossover probability  $\epsilon < \frac{1}{2}$ .

(d) Using the decoding table, how many patterns of 1, 2, and 3 errors are

correctly decoded? How many patterns of 1, 2, and 3 errors are there altogether? What is the probability of incorrect decoding?

- 6.2.** The circuit below is used to encode binary digits for transmission over a binary symmetric channel with crossover probability  $\epsilon < \frac{1}{2}$ . The shift register is initially filled with 0's; then four information digits are shifted into the shift register and simultaneously transmitted. Next, three check digits are transmitted; the four information digits in the shift register shift one place to the right between each check digit calculation.

Find the parity check matrix, the generator matrix, a decoding table, and the probability of decoding error for the code.



- 6.3.** (a) Show that in a parity check code, either all the code words contain an even number of ones or half the code words contain an odd number of ones and half an even number.

(b) Let  $x_{m,n}$  be the  $n$ th digit in the  $m$ th code word of a parity check code. Show that for any given  $n$ , either exactly half or all of the  $x_{m,n}$  are zero. If all of the  $x_{m,n}$  are zero for a given  $n$ , explain how the code could be improved.

(c) Show that the average number of ones per code word, averaged over all code words in a parity check code of block length  $N$ , must be at most  $N/2$ .

- 6.4.** The weight of a binary  $N$ -tuple is the number of 1's in the  $N$ -tuple. The (Hamming) distance between two binary  $N$ -tuples is the weight of their modulo-2 sum.

(a) Let  $x_1$  be an arbitrary code word in a parity check code of block length  $N$  and let  $x_0$  be the all-zero code word corresponding to the all-zero information sequence. Show that for each  $n \leq N$ , the number of code words at distance  $n$  from  $x_1$  is the same as the number of code words at distance  $n$  from  $x_0$ .

(b) The minimum distance,  $d_{\min}$ , of a binary code is the minimum of the distances between each pair of code words. Show that in a parity check code,  $d_{\min}$  is the minimum of the weights of the code words, excluding  $x_0 = (0, 0, \dots, 0)$ .

(c) Show that for a binary code of minimum distance  $d_{\min}$  used on any channel with a binary output sequence, it is possible to correct all combinations of fewer than  $d_{\min}/2$  errors. Hint: Show that if fewer than  $d_{\min}/2$  errors occur, the received sequence is closer in Hamming distance to the transmitted code word than to any other code word.

- (d) Show that if a binary code of minimum distance  $d_{\min}$  is used on a binary erasure channel, all combinations of fewer than  $d_{\min}$  erasures can be corrected.
- 6.5.** Show that if a parity check code has odd minimum weight, adding a parity check digit that is a check on every digit in the code increases the minimum weight by 1.
- 6.6.** (Hamming bound.) The error-correcting capability of a binary block code is defined as the largest integer  $e$  such that all combinations of  $e$  and fewer errors in a block can be corrected

(a) How many different syndrome sequences are there in an  $(N, L)$  parity check code? How many different patterns of  $j$  errors can occur in a sequence of  $N$  digits? Show from this that the error-correcting capability  $e$  of an  $(N, L)$  parity check code must satisfy

$$\sum_{j=0}^e \binom{N}{j} \leq 2^{N-L}$$

(b) Show that for an arbitrary binary code of block length  $N$  with  $M$  code words,  $e$  must satisfy the corresponding inequality

$$\sum_{j=0}^e \binom{N}{j} \leq \frac{2^N}{M}$$

*Hint:* Consider the set of sequences at distance  $e$  or less from each code word and show that these sets must be disjoint

(c) By combining (b) with Problem 6.4c, show that

$$\sum_{j=0}^{\lfloor \frac{d_{\min}-1}{2} \rfloor} \binom{N}{j} \leq \frac{2^N}{M}$$

- 6.7.** (Plotkin bound.) (a) Using Problem 6.3c, show that the average weight of the nonzero code words in an  $(N, L)$  parity check code is at most

$$\frac{N}{2} \frac{2^L}{2^L - 1}$$

Show from this that  $d_{\min}$  satisfies

$$d_{\min} \leq \frac{N}{2} \left( \frac{2^L}{2^L - 1} \right) \quad (\text{i})$$

Observe that this bound agrees with that in (5.8.1) for arbitrary binary codes.

(b) The above bound is effective when  $L$  is small relative to  $N$ . For larger values of  $L$ , the following bound is tighter. Show that for all  $j$ ,  $1 \leq j \leq L$ ,

$$d_{\min} \leq \frac{N - L + j}{2} \left( \frac{2^j}{2^j - 1} \right) \quad (\text{ii})$$

*Hint:* Consider the  $2^j$  code words in the code with the first  $L - j$  information digits constrained to be 0. Consider that set of code words with the first  $L - j$  information digits omitted as an  $(N - L + j, j)$  parity check code and

apply the bound in (a). *Bonus:* Show that (ii) is valid for any binary code of block length  $N$  with  $2^L$  code words.

(c) Now consider  $N$  and  $d_{\min}$  as fixed,  $N \geq 2d_{\min} - 1$ , and show that the number of check digits must satisfy

$$N - L \geq 2d_{\min} - 2 - \lceil \log_2 d_{\min} \rceil$$

*Hint:* Choose  $j = 1 + \lceil \log_2 d_{\min} \rceil$  and remember that  $N - L$ ,  $d_{\min}$ , and  $j$  are integers.

(d) Show that in the limit as  $N \rightarrow \infty$ , with fixed  $d_{\min}/N < \frac{1}{2}$ , the rate in binary units,  $R = L/N$ , must satisfy

$$R \leq 1 - \frac{2d_{\min}}{N}$$

- 6.8.** [Varshamov (1957)-Gilbert bound.] Consider the following strategy for constructing the parity check matrix for a parity check code. For a given number  $r$  of check digits, we start with a diagonal  $r$  by  $r$  matrix and start adding additional rows of  $r$  binary digits each on the top of this set of  $r$  rows. For some given number  $d$ , we ensure that each new row chosen is not a linear combination of any set of  $d - 2$  previously chosen rows. The procedure is terminated when no more rows exist satisfying this condition and the block length  $N$  of the code is the total number of rows that have been put in the matrix.

(a) Show that the code so constructed has a minimum distance of at least  $d$ .

(b) Show that the total number of linear combinations of sets of  $d - 2$  rows is

$$\sum_{i=0}^{d-2} \binom{N}{i}$$

and thus show that  $N$  must satisfy

$$\sum_{i=0}^{d-2} \binom{N}{i} \geq 2^r = 2^{N-L}$$

where  $L$  is the number of information digits for the code. *Note:* This provides a lower bound on the minimum distance that can be achieved with parity check codes (and thus binary codes in general). Observe that the bound is slightly stronger than the Gilbert bound in Problem 5.19, increasing the achievable value of  $d_{\min}$  by 1.

- 6.9.** Consider two parity check codes. Code I is generated by the rule

$$\begin{aligned} x_1 &= u_1 & x_4 &= u_1 \oplus u_2 \\ x_2 &= u_2 & x_5 &= u_1 \oplus u_3 \\ x_3 &= u_3 & x_6 &= u_2 \oplus u_3 \\ && x_7 &= u_1 \oplus u_2 \oplus u_3 \end{aligned}$$

Code II is the same except that  $x_6 = u_2$ .

(a) Write down the generator matrix and parity check matrix for code I.

(b) Write out a decoding table for code I, assuming a BSC with crossover probability  $\epsilon < \frac{1}{2}$ .

- (c) Give an exact expression for the probability of decoding error for code I and for code II. Which is larger?
- (d) Find  $d_{\min}$  for code I and for code II.
- (e) Give a counterexample to the conjecture that if one  $(N,L)$  parity check code has a larger minimum distance than another  $(N,L)$  parity check code, it has a smaller error probability on a BSC.
- 6.10.** Consider a parity check code for which the rows of the generator matrix are not linearly independent. Show that some nonzero information sequence is mapped into an all-zero code word. Use this to show that for each information sequence, at least one other information sequence is mapped into the same code word. Clearly such codes have no practical interest.
- 6.11.** Consider two parity check matrices  $H_1$  and  $H_2$  with the same column space (i.e., the same set of linear combinations of columns).
- Show that a sequence  $\mathbf{x}$  satisfies  $\mathbf{x}H_2 = \mathbf{0}$  iff  $\mathbf{x}H_1 = \mathbf{0}$  and thus that  $H_1$  and  $H_2$  correspond to the same set of code words.
  - Show that two error sequences have different syndromes using  $H_2$  (i.e.,  $\mathbf{e}_1 H_2 \neq \mathbf{e}_2 H_2$ ) iff they have different syndromes using  $H_1$ . Show from this that the same set of error sequences can be corrected using a decoding table constructed from  $H_2$  or from  $H_1$ .
  - Assume that  $H_1$  has  $r$  columns and that  $r' < r$  is the size of the largest set of linearly independent columns. Show that the code has  $2^{N-r}$  code words and  $2^{r'}$  entries in the decoding table.
- 6.12.** Consider two parity check codes of the same block length. Interpret the set of code words in each code as a group with modulo 2 addition. Show that the set of sequences that are code words in both codes forms a subgroup of each of the above groups. Interpreting the sequences in this subgroup as a parity check code, describe a procedure for finding a parity check matrix for the new code in terms of the parity check matrices for the original codes.
- 6.13.** Show that any finite group in which each element is its own inverse has  $2^n$  elements for some  $n$  and is isomorphic to the group of binary  $n$  tuples with modulo 2 addition. (Two groups are isomorphic if there is a one-to-one correspondence between their elements that preserves the group operation.)  
*Hint:* First show, by assuming a contradiction, that the group is Abelian. Then demonstrate a subgroup with two elements. Then show that the elements of any subgroup and a single coset of that subgroup form a new subgroup with twice as many elements.
- 6.14.** Let  $1, a, a^2, \dots, a^5$  denote the elements of a cyclic group of order 6. Find the order of each element in the group and list all the subgroups of the group.
- 6.15.** (a) Write out addition and multiplication tables for the field of the integers  $0, 1, \dots, 4$ , using addition and multiplication modulo 5.  
(b) Prove Fermat's theorem: for any prime number  $p$  and any integer  $a$  not divisible by  $p$ , the remainder of  $a^{p-1}$  modulo  $p$ ,  $R_p(a^{p-1}) = 1$ .  
*Hint:* consider the field of elements modulo  $p$ , let the field element  $\mathbf{a}$  be  $R_p(a)$ , and consider the multiplicative order of  $\mathbf{a}$ .
- 6.16.** Let  $D^4 + D^2 + D + 1$  be a polynomial over  $GF(2)$ . Express it as a product

of irreducible monic factors. Indicate the reasoning by which you arrived at your answer.

- 6.17.** Consider the field whose elements are polynomials of degree 1 or less, with coefficients in  $GF(3)$  and with multiplication being defined as polynomial multiplication modulo  $D^2 + 1$ .

(a) Prove that  $D^2 + 1$  is irreducible over  $GF(3)$ .

(b) Write out the addition and multiplication table for this field.

- 6.18.** Consider a code with ternary symbols where each code word,  $\mathbf{x} = (x_1, x_2, x_3, x_4)$  is generated from a ternary message sequence,  $\mathbf{u} = (u_1, u_2)$  by the rules

$$x_1 = u_1$$

$$x_2 = u_2$$

$$x_3 = u_1 \oplus u_2$$

$$x_4 = u_1 \oplus 2u_2$$

where  $\oplus$  denotes addition modulo 3.

(a) Write out the generator matrix and a check matrix for this code.

(b) Write out a syndrome to error sequence decoding table in such a way as to minimize the probability of incorrect decoding. Assume a ternary symmetric channel with  $P(j|k) = \epsilon < \frac{1}{3}$  for  $j \neq k$  and  $P(j|k) = 1 - 2\epsilon$  for  $j = k$ .

(c) For any number  $m \geq 2$  of check digits, show how to construct a check matrix of a linear code with ternary symbols so that the decoding table contains the all-zero sequence, all single error sequences, and no sequences with more than a single error. Find the block length of these codes as a function of  $m$ .

(d) Repeat (c) for linear codes with symbols in an arbitrary field  $GF(q)$ .

- 6.19.** Let the  $5 \times 8$  matrix below be the *generator* matrix of a binary parity check code with 5 information digits and 3 check digits.

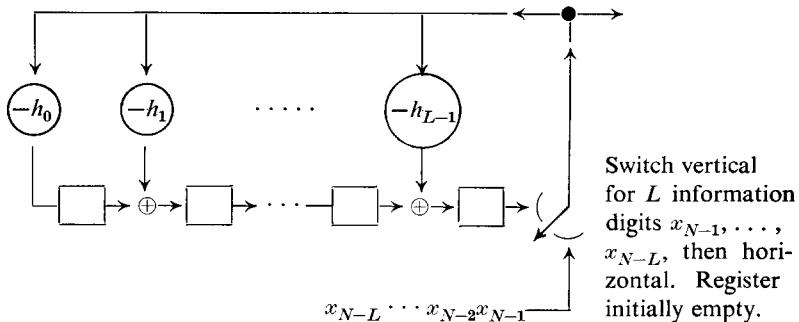
$$\begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 1 & 1 & 1 \\ 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \end{bmatrix}$$

Show that this code is cyclic and find the generator polynomial and check polynomial.

- 6.20.** Consider the cyclic code of length 7 with the generator polynomial  $D^3 + D + 1$ . Show two circuits for implementing the encoder, one employing a three-stage shift register and the other employing a four-stage shift register. Show that this code is capable of correcting all single errors and is thus equivalent to a Hamming single-error-correcting code of length 7.

- 6.21.** Let  $g(D) = g_m D^m + \cdots + g_0$  be the generator polynomial of a cyclic code. Show that  $g_0 \neq 0$ .

- 6.22. Show that the circuit below is an encoder for an  $(N, L)$  cyclic code with check polynomial  $h(D)$ .



*Hint:* Calculate what is in the rightmost stage of the register after the  $L$  information digits are shifted out on the line. *Note:* This circuit has two practical advantages over that in Fig. 6.5.4: first, the modulo 2 addition can be performed internally in the shift register stages, and, second, for high-speed applications there is no need of a serial modulo 2 addition.

- 6.23. Suppose that a binary  $(N, L)$  cyclic code is used on a burst erasure channel. That is, when a code word  $x_{N-1}, \dots, x_1, x_0$  is transmitted, the received sequence will have the form

$$\mathbf{y} = x_{N-1}, \dots, x_l, e, e, \dots, e, x_j, \dots, x_2, x_0$$

In other words, all received digits will be correct except for a sequence of consecutive digits that are erased.

(a) Show that if the number of erased digits is at most  $N - L$ , correct decoding can always be accomplished.

(b) Show that if more than  $N - L$  digits are erased, a maximum likelihood decoder can never make an unambiguous choice.

(c) Draw a block diagram of the simplest decoder you can design to correct all combinations of  $N - L$  or fewer consecutive erasures (assume  $L > N/2$ ).

- 6.24. The (Hamming) distances between two  $N$ -sequences of symbols from  $GF(q)$  is the number of positions in which the sequences differ, and the weight of a sequence is the number of nonzero symbols in the sequence. The minimum distance  $d_{\min}$  of a linear code with symbols from  $GF(q)$  is the minimum of the distances between each pair of code words.

(a) Show that  $d_{\min}$  is the minimum of the weights of the nonzero code words. Show that all combinations of fewer than  $d_{\min}/2$  errors can be corrected (see Problem 6.4).

(b) Show that the following extensions of the bounds on  $d_{\min}$  in Problems 6.6, 6.7, and 6.8 are valid for linear codes with symbols from  $GF(q)$ . *Hint:*

Use the same arguments as in Problems 6.6 to 6.8

(i) Hamming bound:

$$\sum_{j=0}^{\lfloor \frac{d_{\min}-1}{2} \rfloor} (q-1)^j \binom{N}{j} \leq q^{N-L}$$

(ii) Plotkin bound:

$$d_{\min} \leq \frac{N(q-1)}{q} \left( \frac{q^L}{q^L - 1} \right)$$

For any  $j$ ,  $1 \leq j \leq L$

$$d_{\min} \leq \frac{(N-L+j)(q-1)}{q} \frac{q^j}{q^j - 1}$$

For  $N \geq (qd_{\min} - 1)/(q-1)$ , let  $j = \lfloor \log_q d_{\min} \rfloor + 1$  to obtain

$$N - L \geq \frac{qd_{\min} - 1}{q - 1} - 1 - \log_q d_{\min}$$

(iii) Varshamov-Gilbert bound: An  $(N,L)$  linear code exists with

$$\sum_{i=0}^{d_{\min}-2} \binom{N}{i} (q-1)^i \geq q^{N-L}$$

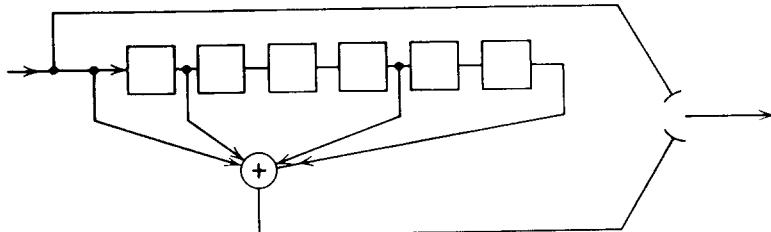
- 6.25.** Verify (6.6.1) for the field of polynomials over  $GF(2)$  modulo the polynomial  $D^2 + D + 1$ . *Hint:* To avoid confusion, represent the field elements as polynomials in  $t$ .
- 6.26.** Let  $\alpha$  be a primitive element of  $GF(q)$ . Show that the multiplicative order of  $\alpha^i$  is  $(q-1)/[G.C.D.(i, q-1)]$ , or equivalently,  $L.C.M.(i, q-1)/i$ . Show that if  $n$  divides  $q-1$ , there are  $n$  field elements with multiplicative orders of  $n$  or divisors of  $n$  and that these elements form a cyclic group under multiplication.
- 6.27.** Suppose that a given element  $\alpha$  in  $GF(p^n)$  has a minimal polynomial [over  $GF(p)$ ] of degree  $m < n$ .
- Show that the smallest subfield of  $GF(p^n)$  containing  $\alpha$  contains  $p^m$  elements. Show how to represent all the elements of this subfield in terms of  $\alpha$  and the integers of the field. *Hint:* Review the proof of Theorem 6.6.4.
  - Show that  $m$  must divide  $n$ .
  - Show that the subfield found in (a) is the only subfield of  $p^m$  elements in  $GF(p^n)$ . *Hint:* Observe that  $D^{p^m-1} - 1$  has only  $p^m - 1$  roots in  $GF(p^n)$ .
  - Suppose that an element  $\beta$  in  $GF(p^n)$  has multiplicative order  $j$ . Show that the degree,  $i$ , of the minimal polynomial [over  $GF(p)$ ] of  $\beta$  is such that  $j$  divides  $p^i - 1$ .
- 6.28.** Consider the field  $GF(2^4)$  of polynomials modulo  $D^4 + D + 1$ . Find the polynomials in this field which are in the subfield  $GF(2^2)$ .
- 6.29.** Consider the field  $GF(2^4)$  of polynomials modulo  $D^4 + D + 1$ .
- Find the minimal polynomial  $f_\alpha(D)$  of the element  $\alpha = t^3 + 1$  by observing that  $\alpha, \alpha^2, \alpha^4$ , and  $\alpha^8$  are all roots of  $f_\alpha(D)$  and thus  $f_\alpha(D) = (D - \alpha)(D - \alpha^2)(D - \alpha^4)(D - \alpha^8)$ .

(b) Repeat (a) by solving the equation  $f_\alpha(\alpha) = 0$ . More explicitly, letting  $f_2(D) = f_0 + f_1D + \dots + f_4D^4$ , solve the equation  $f_0 + f_1(t^3 + 1) + f_2(t^3 + 1)^2 + \dots + f_4(t^3 + 1)^4 = 0$ . This can be interpreted as a set of 4 equations in binary variables, one involving  $t^3$ , one  $t^2$ , one  $t^1$ , and the last  $t^0$ . You can take  $f_0 = 1$ , since  $f_\alpha(D)$  is irreducible, and solve for the binary variables  $f_1, \dots, f_4$ .

- 6.30.** (a) Find a generator polynomial for a two error correcting binary BCH code of length 15 and a three-error-correcting binary BCH code of length 15.  
 (b) Suppose that two binary BCH codes are defined in terms of having  $\alpha, \dots, \alpha^{d-1}$  as roots of all code words. However  $\alpha$  for one code is a primitive element of  $GF(2^n)$  with minimal polynomial  $f_1(D)$ , and for the other code  $\alpha$  is a primitive element of  $GF(2^m)$  with a different minimal polynomial  $f_2(D)$ . Show that the set of code words for one code can be obtained by a fixed permutation of the digits of the code words in the other code.
- 6.31.** (a) Show that a linear code with  $L$  information symbols and block length  $N$  must have a minimum distance  $d_{\min}$  at most  $N - L + 1$ .  
 (b) A Reed Solomon code is a BCH code with  $m = 1$ . Show that all Reed Solomon codes meet the above upper bound on  $d_{\min}$  with equality.  
 (c) Suppose that a Reed Solomon code is used on an erasure channel (i.e., each channel output is either the same as the corresponding input or else an erasure symbol). Show that if fewer than  $d_{\min}$  erasures occur in a block, correct decoding can always be accomplished, and if  $d_{\min} + i$ ,  $i \geq 0$ , erasures occur, there are exactly  $q^{i+1}$  code words that could have given rise to the decoded word. Here  $q$  is the size of the input alphabet.  
 (d) For a Reed Solomon code of block length  $N$  and input alphabet size  $q$ , find the total number of code words with exactly  $d_{\min}$  nonzero elements.
- 6.32.** Consider a binary BCH code with arbitrary  $m$ , primitive  $\alpha$ , and  $r = 1$  and  $d = 3$ . Show that this is a Hamming code, in cyclic form, with a block length  $2^m - 1$ . Solve (6.7.27) for this case and show that the solution is equivalent to the decoding table technique previously devised for Hamming codes.
- 6.33.** Use the iterative algorithm to find the minimal length binary shift register that generates the polynomial [over  $GF(2)$ ]  $1 + D^3 + D^4 + D^7$ ; find the minimal length binary shift register that generates  $D^2 + D^5 + D^6$  [over  $GF(2)$ ].
- 6.34.** The polynomial  $f(D) = D^4 + D + 1$  is primitive over  $GF(2)$ . Let  $\alpha$  be the polynomial  $t$  in the field of polynomials modulo  $f(D)$  (see Fig. 6.6.3). Consider the BCH code with  $q = 2$ ,  $m = 4$ ,  $\alpha$  as above,  $r = 1$ , and  $d = 5$ . Draw a block diagram of a circuit to calculate the syndrome polynomial  $S(D)$ , representing each  $S_i$  as an element of the field of polynomials modulo  $f(D)$ .
- 6.35.** Show that  $A_n(D) = [C_n(D)S(D)]_0^{n-1}$  for each  $n \geq 0$  [see (6.7.69)]. *Hint:* See the proof of (c) and (d) of Theorem 6.7.3.
- 6.36.** Show that (6.7.22) follows from (6.7.21). Note that it is sufficient to show that

if the derivative of a polynomial over a finite field is defined as in (6.7.71) and if  $f(D) = g(D)h(D)$ , then  $f'(D) = g'(D)h(D) + g(D)h'(D)$ .

- 6.37.** Draw a block diagram of a two-error-correcting threshold decoder for the convolutional encoder shown below.



- 6.38.** Draw a block diagram of a two-error-correcting, three-error-detecting threshold decoder for a systematic convolutional encoder with a rate (in bits) of  $\frac{1}{2}$  and with the check digits generated by the rule

$$x_n^{(2)} = u_n + u_{n-6} + u_{n-7} + u_{n-9} + u_{n-10}$$

*Hint:* Find a set of five linear combinations of the noise digits orthogonal on  $z_1^{(1)}$  and use these as inputs to the threshold device.

- 6.39.** The idea of threshold decoding can be applied to block codes as well as convolutional codes. Consider a maximal length code with  $L$  information digits and block length  $N = 2^L - 1$ .

(a) Show that the decoder can calculate  $(N - 1)/2$  linear combinations of the noise digits  $z_0, \dots, z_{N-1}$  that are orthogonal on  $z_{N-1}$ . *Hint:* Recall from Section 6.6 that the dual code is a Hamming code and that every code word in the dual code corresponds to a check equation for the maximal length code. Also recall that in a Hamming code every sequence (in particular those of weight 2) is at a distance of at most one from some code word.

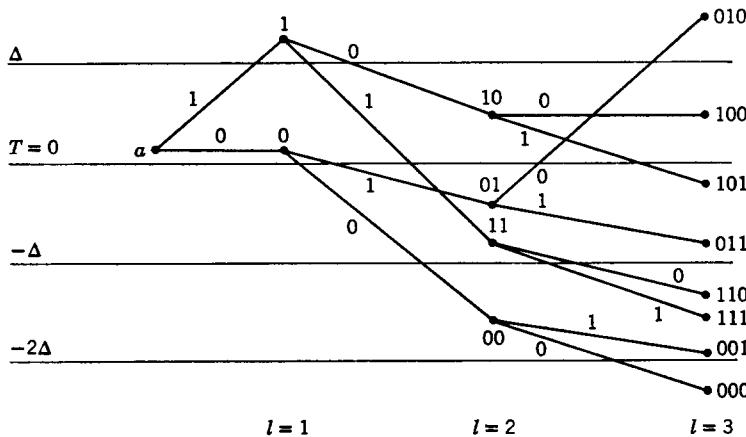
(b) Show that if a threshold decoder is built to correct  $z_{N-1}$ , the same threshold decoder can be used to correct all the noise digits. Show that all combinations of  $(N - 3)/4$  or fewer errors can be corrected in this way.

(c) Draw a detailed block diagram of such a decoder for  $L = 4$ .

- 6.40.** A sequential decoder starts at node  $a$  with a zero threshold and moves to node 0 in the received value tree below. List the sequence of nodes hypothesized up to the first hypothesis of a node at depth  $l = 3$  into the tree. Assume that on a forward move, the decoder always takes the branch labeled 0.

- 6.41.** (a) Let  $\Gamma_l$  be the value of the  $l$ th node along the correct path in the received value tree for a sequential decoder. Find the average value of  $\Gamma_l$  over the ensemble of codes and of channel noises. Use the Chernoff bound to upper-bound  $\Pr [\Gamma_l \leq i\Delta]$  and show that  $\Pr [\Gamma_l \leq i\Delta]$  approaches 0 exponentially with increasing  $l$  for any bias  $B < C$ .

(b) Let  $\Gamma'_l$  be the value of the  $l$ th node in the received value tree corresponding to some information sequence differing from the transmitted

**3Δ****2Δ**

sequence in the first digit. Use the Chernoff bound on  $\Pr [\Gamma'_l \geq i\Delta]$ . Show that this bound differs from the bound on  $\Pr [\Gamma_l < i\Delta]$  by a factor of  $\exp [-\nu LB - i\Delta]$ . Show that if  $R \leq B < C$ , the probability that any such  $\Gamma'_l$  exceed  $i\Delta$  is exponentially decreasing with increasing  $l$ .

- 6.42.** The bounds on error probability and average computation in Section 6.9 appear somewhat artificial because of the use of either a time-varying code or an infinite constraint length. In this problem, we exchange those artificialities for a new one. Specifically, we use the code ensemble defined prior to lemma 6.9.1, using a finite constraint length of  $L$  subblocks. We modify the decoding algorithm, however, as follows. At each depth  $l$  ( $l > L$ ) on the received value tree, when the decoder makes the first  $F$ -hypothesis of a node at depth  $l$ , it unconditionally accepts the hypothesized source digits in the  $l - L + 1$ th subblock and sets  $\Gamma_{l-L} = -\infty$ . In other words, the decoder is prevented from changing hypotheses more than a constraint length away from its maximum depth of penetration into the tree. We define an error at node  $n$  if the decoder ever makes an  $F$ -hypothesis at depth  $n + L$  on any incorrect path that branches off from the correct path for the first time at node  $n$ . We define  $W_n$  as the number of  $F$ -hypotheses made on the  $n$ th node on the correct path and the tree of incorrect paths stemming from that  $n$ th node before an error occurs.

(a) Show that  $W_n$  is upper-bounded by the right-hand side of (6.9.23).  
*Hint:* First show that the transmitted digits in subblocks  $n + 1$  to  $n + L$ , corresponding to two source sequences differing in subblock  $n + 1$ , are

statistically independent. Then follow the procedure used to obtain (6.9.23).

(b) Show that the probability of error at node  $n$ , for  $B \leq E_0(1, Q)$  satisfies

$$P_{e,n} \leq (L + 1)e^{\Delta/2} \exp \left\{ -\nu L \left[ \frac{E_0(1, Q) + B}{2} - R \right] \right\}$$

Compare this result with that in (6.9.40).

- 6.43.** (a) Show that  $\gamma_i$  and  $\gamma'_i$  in (6B.2) and (6B.4) are statistically independent if the matrix of transition probabilities has the property that each row is a permutation of each other row and each column is a permutation of each other column and if the inputs are equally likely (this class of channels is a subset of the symmetric channels of Section 4.5 and includes the B.S.C.).

(b) Show that for this class of channels, the result of lemma 6.9.3 can be strengthened to

$$\Pr[\Gamma'_l \geq \Gamma_{\min} + (i - 2)\Delta] \leq \exp \left[ -\frac{(i-2)\Delta}{2} - \nu l \frac{E_0(1, Q) + B}{2} \right]$$

*Hint:* Use your result in (a) to show that  $\Gamma'_l$  and  $\Gamma_{\min}$  are statistically independent. Then suitably modify the argument in (6B14) to (6B22) to apply to  $\Gamma_{\min}$  rather than  $\min \Gamma_{n,i}$ .

(c) Use the above result to tighten the bound on  $\bar{W}_n$  in (6.9.30) and the bound on  $P_{e,n}$  in (6.9.40) to (6.9.42).

- 6.44.** Again assume a binary symmetric channel with  $Q(0) = Q(1) = 1/2$  so that  $\Gamma'_l$  and  $\Gamma_{\min}$  are statistically independent. Let  $\bar{W}_0(u)$  be the average number of forward hypotheses performed by the decoder conditional on  $\Gamma_{\min} = u$ . Assume that the bias  $B$  is equal to the rate  $R$  and that  $R > E_0(1, Q)$ . Show that

$$\bar{W}_0(u) \leq A \exp \left( \frac{-u}{1 + \rho_r} \right)$$

where  $\rho_r$  is the solution to

$$\rho_r R = E_0(\rho_r)$$

and where  $A$  is at most linearly increasing with decreasing  $u$ .

*Hint:* Show that

$$\Pr[\Gamma'_l \geq u + (i - 2)\Delta] \leq \exp \left\{ -\frac{1}{1 + \rho} [u + (i - 2)\Delta + Bl\nu + E_0(\rho)l\nu] \right\}$$

for  $\rho \geq 0$ . In summing over  $l$ , use  $\rho = \rho_r$  for small  $l$  and a small value of  $\rho$  for large  $l$ .

- 6.45.** Consider a random walk

$$S_n = \sum_{i=1}^n z_i$$

where the  $z_i$  are identically distributed and  $\bar{z}_i < 0$ . For a fixed  $u < 0$ , let  $N$  (a random variable) be the smallest value of  $n$  for which  $S_n \leq u$  and let  $S_N$  be  $S_n$  evaluated at that  $n$ . By taking the derivative of Wald's equality (6B.29) with respect to  $r$  at  $r = 0$ , show that  $\bar{N} = \bar{S}_N/\bar{z}$ . If  $z_{\min}$  is the smallest value taken on by the random variables  $z_i$ , show that  $u + z_{\min} < \bar{N}\bar{z} \leq u$ .

- 6.46.** Consider a binary code of arbitrarily long block length with two code words. Show that it is impossible to correct all bursts of length  $g$  relative to a guard space  $g$ . Hint: Consider the two types of error sequences in Fig. 6.10.2 and show that it is possible to choose  $\mathbf{z}_1$  and  $\mathbf{z}_2$  so that  $\mathbf{x}_1 \oplus \mathbf{z}_1 = \mathbf{x}_2 \oplus \mathbf{z}_2$ .

## CHAPTER 7

- 7.1.** A channel has an input  $x$  consisting of the numbers  $+1$  and  $-1$  used with the probabilities  $P_X(+1) = P_X(-1) = \frac{1}{2}$ . The output  $y$  is the sum of  $x$  and an independent noise random variable  $z$  with the probability density  $P_Z(z) = \frac{1}{4}$  for  $-2 < z \leq 2$ , and  $P_Z(z) = 0$  elsewhere. In other words, the conditional probability density of  $y$  conditional on  $x$  is given by  $P_{Y|X}(y|x) = \frac{1}{4}$  for  $-2 < y - x \leq 2$  and  $P_{Y|X}(y|x) = 0$  elsewhere.
- (a) Find and sketch the output probability density for the channel.
  - (b) Find  $I(X; Y)$ .
  - (c) Suppose that the output is transformed into a new discrete random variable  $z$  defined by  $z = 1$  for  $y > 1$ ;  $z = 0$  for  $-1 < y \leq 1$ ;  $z = -1$  for  $y \leq -1$ . Find  $I(X; Z)$  and interpret your result.
  - (d) For a discrete  $X, Y$  ensemble, let  $Z$  be a new ensemble with elements determined by the elements in the  $Y$  ensemble,  $z = z(y)$ . Show that if  $P(x|z(y)) = P(x|y)$ , then  $I(X; Z) = I(X; Y)$ .
- 7.2.** A discrete time memoryless channel has inputs  $+1$  and  $-1$  and real number outputs determined by the transition probability density

$$p(y|1) = \begin{cases} \frac{1}{a+b} \exp(-y/a); & y \geq 0 \\ \frac{1}{a+b} \exp(y/b); & y \leq 0 \end{cases}$$

and  $p(-y|-1) = p(y|1)$ . The constants  $a$  and  $b$  are arbitrary,  $a \geq b$ . (It turns out that this probability density arises in a Rayleigh fading channel if the inputs are equal energy orthogonal waveforms and the output is the output log likelihood ratio. See Problem 8.21.)

- (a) Find an expression for the channel capacity and evaluate in the limit as  $b/a \rightarrow 1$  and  $b/a \rightarrow 0$ .
- (b) Evaluate  $E_0(1)$ .
- (c) Show that maximum likelihood decoding can be accomplished by choosing  $m$  to minimize

$$\sum_{n: x_m, n y_n < 0} |y_n|$$

(d) Evaluate the probability of error for no coding using maximum likelihood detection.

- 7.3.** A discrete time memoryless channel has the interval  $(0,1)$  as its input alphabet and the interval  $(0,1)$  plus the erasure symbol  $E$  as its output alphabet. For each input  $x$ ,  $0 \leq x \leq 1$ , the output  $y$  takes on the value  $x$  with probability  $\frac{1}{2}$  and takes on the symbol  $E$  with probability  $\frac{1}{2}$ .

- (a) Find the capacity of this channel, the function  $E_0(\rho)$ , and the random coding exponent  $E_r(R)$ . Note in particular the discontinuity of  $E_0(\rho)$  at  $\rho = 0$ . *Hint:* Consider using only a finite set of input letters and take the limit as the size of this set goes to infinity.
- (b) Show that for  $M$  equally likely messages, your bound is exactly  $M/(M - 1)$  times the actual minimum achievable probability of decoding error.
- 7.4.** A discrete time memoryless channel has the set of phase angles,  $0 \leq \varphi < 2\pi$ , as input alphabet and output alphabet. The channel is subject to additive phase noise  $z$  where  $z$  is independent of the input  $x$  and has a probability density  $p_Z(z)$  which is nonzero only for  $0 \leq z < 2\pi$ . The channel output  $y$  is the sum of  $x + z$  reduced modulo  $2\pi$ . For example, if  $x = 7\pi/4$  and  $z = 3\pi/2$ , then  $y = x + z - 2\pi = 5\pi/4$ .
- (a) Show that  $C$  and  $E_0(\rho)$  are achieved by an input probability density,  $p_X(x) = 1/2\pi$ ,  $0 \leq x < 2\pi$ .
- (b) Evaluate  $C$ ,  $E_0(\rho)$ , and  $E_r(R)$  for the following two cases:
- $p_Z(z) = 1/\alpha$ ;  $0 \leq z < \alpha$ , and  $p_Z(z) = 0$  elsewhere.
- (ii) 
$$p_Z(z) = \frac{\alpha e^{-\alpha z}}{1 - e^{-2\pi\alpha}}; \quad 0 \leq z < 2\pi.$$
- 7.5.** A discrete time memoryless channel has an input  $x$  constrained to the interval  $(-A, A)$  and has additive noise  $z$  with the probability density  $p_Z(z) = 1/2$  for  $-1 < z \leq 1$  and  $p_Z(z) = 0$  elsewhere.
- (a) For  $A = 1/2$ , find the capacity of the channel and the input distribution that leads to it. Show that for the set of inputs that yields capacity, the channel is equivalent to a binary erasure channel. Find the random coding exponent  $E_r(R)$  for the channel and verify that it is the same as that for the BEC. *Hint:* Guessing the input distribution and verifying that it yields capacity is the easy approach here.
- (b) For arbitrary noninteger  $A$ , show that both the average mutual information and  $E_0(\rho, Q)$  are maximized by a discrete distribution given by
- $$Q(A - 2i) = Q(-A + 2i) = \frac{n - i}{n(n + 1)}$$
- for each integer  $i$ ,  $0 \leq i < n$  where  $n = [A]$ . Sketch the output probability density for this input assignment and find  $C$  and  $E_r(R)$  (in parametric form).
- (c) Find the maximizing distribution for integer  $A$  and interpret as a limit of the noninteger case.
- 7.6.** A set of  $N$  independent, discrete time, additive Gaussian noise channels are connected in parallel. The noise variance on the  $n$ th channel is given by  $\sigma_n^2 = n^2$  for each  $n$ . The input energy is constrained by the condition
- $$\sum_{n=1}^N \mathcal{E}_n/n \leq 5.$$
- (a) Find the channel capacity of the parallel combination and find the values of  $\mathcal{E}_n$  that achieve capacity for the cases  $N = 2$ ,  $N = 4$ ,  $N = \infty$ .

*Hint:* First scale the signal and noise on each channel to reduce the problem to that solved in Section 7.5.

(b) For the case  $N = \infty$ , find the critical rate  $R_{cr}$  and find  $E_r(R_{cr})$  and  $E_r(0)$ .

(c) Change the constraint to

$$\sum_{n=1}^N \mathcal{E}_n/n \leq 50$$

and find the new values of  $\mathcal{E}_n$  that achieve capacity for  $N = \infty$ .

- 7.7. Consider a set of  $N$  parallel, discrete-time, Gaussian noise channels with noise variances  $\sigma_1^2, \sigma_2^2, \dots, \sigma_N^2$ . Suppose that we choose two words for a single use of the parallel set, letting

$$x_1 = (\sqrt{\mathcal{E}_1}, \sqrt{\mathcal{E}_2}, \dots, \sqrt{\mathcal{E}_N}) \quad \text{and} \quad x_2 = (-\sqrt{\mathcal{E}_1}, -\sqrt{\mathcal{E}_2}, \dots, -\sqrt{\mathcal{E}_N})$$

(a) Find an exact expression for the error probability using maximum likelihood decoding. Your answer will be the tail of an appropriate Gaussian distribution.

(b) Given the constraint

$$\sum_{n=1}^N \mathcal{E}_n \leq \mathcal{E},$$

find the choice of  $\mathcal{E}_n$ ,  $1 \leq n \leq N$  that minimizes the error probability.

(c) Compare your answer with the zero rate exponent in the expurgated random coding bound.

## CHAPTER 8

- 8.1. (a) Let  $z(t)$  be a zero mean Gaussian random process and assume that the variance of the random variable  $\int x(t)z(t) dt$  is finite for all  $L_2 x(t)$ . Show that this implies the existence of some number  $M$  such that the variance of  $\int x(t)z(t) dt$  is less than or equal to  $M$  for all normalized functions  $x(t)$ . *Hint:* Let  $\{\varphi_i(t)\}$  be a complete orthonormal set and  $z_i = \int \varphi_i(t)z(t) dt$ . Show that if the set  $\{z_i\}$  has bounded variance, then for any normalized function,  $x(t)$ , the variance of  $\int x(t)z(t) dt$  has the same bound. If the set  $\{z_i\}$  has unbounded variance, choose a subset  $z_{i_1}, z_{i_2}, \dots$  such that  $\overline{z_{i_n}}^2 \geq 2^{2n}$ . Let

$$x(t) = \sum_n 2^{-n/2} \varphi_{i_n}(t)$$

and show that  $x(t)$  is normalized but that  $\int x(t)z(t) dt$  has infinite variance.

(b) Use the above result, with  $z(t)$  as above,  $\{\varphi_i(t)\}$  a complete orthonormal set,

$$x(t) = \sum_i x_i \varphi_i(t)$$

an  $L_2$  function, and  $z_i = \int \varphi_i(t)z(t) dt$ , to show that

$$\lim_{k \rightarrow \infty} \left[ \int x(t)z(t) dt - \sum_{i=1}^k x_i z_i \right]^2 = 0$$

- 8.2.** Define a Gaussian random process  $z(t)$  to be stationary if for all  $L_2$  functions  $x(t)$  and for all  $\tau$ , the variance of  $\int x(t)z(t) dt$  is equal to the variance of  $\int x(t + \tau)z(t) dt$ . Define the spectral density of a stationary Gaussian random process, if and where it exists, to be given by

$$S(f) = \lim_{k \rightarrow \infty} \left[ \int_0^{k/f} \sqrt{2f/k} (\cos 2\pi ft) z(t) dt \right]^2$$

Show that  $S(f)$ , where it exists, is less than or equal to  $M$  as defined in Problem 8.1.

- 8.3.** Let  $z(t)$  be a zero mean random process with the correlation function  $\mathcal{R}(t, \tau)$ . Show that  $\mathcal{R}(t, \tau)$  is continuous everywhere iff

$$\lim_{\epsilon \rightarrow 0} [z(t) - z(t + \epsilon)]^2 = 0 \quad \text{for all } t$$

*Hint:* To establish the iff, apply the Schwarz inequality to  $\mathcal{R}(t, \tau) - \mathcal{R}(t + \epsilon, \tau)$ .

- 8.4.** Let  $x(t)$  and  $X(f)$  be a Fourier transform pair,  $X(f) = \int x(t)e^{-j2\pi ft} dt$ ;  $x(t) = \int X(f)e^{j2\pi ft} df$ . Show that if  $X(f)$  is absolutely integrable (i.e.,

$$\int_{-\infty}^{\infty} |X(f)| df < \infty \Big),$$

then  $x(t)$  is continuous. *Hint:* Observe that

$$x(t) - x(t - \epsilon) = \int X(f)[e^{j2\pi f t}] [1 - e^{-j2\pi f \epsilon}] df$$

Bound the integral separately for large  $f$  and small  $f$  and show that the magnitude goes to 0 as  $\epsilon$  goes to 0.

- 8.5.** Let  $z(t)$  be a zero mean Gaussian random process as defined in Section 8.1, let  $h(t)$  be an  $L_2$  function and let  $y(t)$  be a random process defined by  $y(t) = \int h(t - \tau)z(\tau) d\tau$ . Show that the correlation function of  $y(t)$ ,  $\mathcal{R}_y(t, \tau)$  is continuous. *Hint:* Let  $M$  be an upper bound on the variance of  $\int x(t)z(t) dt$  for all normalized  $x(t)$  (see Problem 8.1) and show that

$$\overline{[y(t) - y(t + \epsilon)]^2} \leq M \int [h(t) - h(t + \epsilon)]^2 dt$$

Next define  $w(\epsilon) = \int h(t)h(t + \epsilon) dt$  and use Problem 8.4 to show that  $w(\epsilon)$  is continuous. Use this to show that

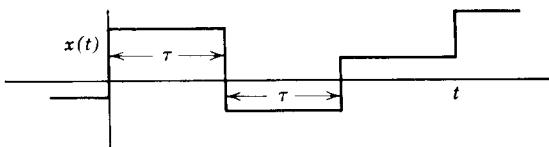
$$\lim_{\epsilon \rightarrow 0} \int [h(t) - h(t + \epsilon)]^2 dt = 0.$$

Finally, use Problem 8.3.

- 8.6.** Let  $z(t)$  be a stationary, zero mean, Gaussian random process with the correlation function  $\mathcal{R}(\tau)$ . Let  $y(t)$  be another random process, statistically independent of  $z(t)$ , for which, for each  $t$ ,  $y(t)$  is a random variable uniformly distributed between  $-1$  and  $+1$ . Assume that  $y(t)$  for each  $t$  is statistically independent of  $y(t')$  for all other values of  $t$ . Show that  $z(t) + y(t)$  is a Gaussian

random process by the definition in Section 8.1, but that for each  $t$ ,  $y(t) + z(t)$  is not a Gaussian random variable. (Note that from a physical standpoint, the random process  $y(t)$  can never be observed since any measuring device must perform some averaging. From a mathematical standpoint, however, this problem shows the kind of pathological effects that can arise from random processes without a continuous correlation function.)

- 8.7.** An additive white Gaussian noise channel has a noise spectral density  $N_0/2$  and a power limitation  $S$ . Suppose that for some given time interval  $\tau$ , the channel input is constrained to be constant within each interval of length  $\tau$  and to change only at instants separated by  $\tau$ .



(a) Find an appropriate orthonormal expansion for the input and represent the channel as a discrete time channel. Find the capacity of the channel (as constrained above) in nats per second and find the limiting capacity as  $\tau \rightarrow 0$ .

(b) Now impose the additional constraint that the input can take on only the values  $\pm \sqrt{S}$  and find the capacity as  $\tau \rightarrow 0$ .

(c) Now assume that for the input constraint of (b) the receiver is constrained to integrate the output over each  $\tau$  second interval and save only the sign of the integral. In other words, the channel is reduced to a BSC, transmitting one channel digit each  $\tau$  seconds. Find the capacity (in nats/sec) as  $\tau \rightarrow 0$ . (Note that all of these channels are “very noisy channels” in the limit  $\tau \rightarrow 0$ , and that the entire exponent-rate curve is determined by the capacity.)

- 8.8.** A white Gaussian noise channel has a noise spectral density of  $N_0/2$ . The transmitted signal is constrained to have an average power  $S$  and for each integer  $i$  is constrained to have the following form for  $t$  between  $i$  and  $i + 1$  seconds:

$$\sum_{m=M_1}^{M_2} x_m(i) \cos [2\pi \cdot 100mt + \varphi_m(i)]; \quad i \leq t < i + 1$$

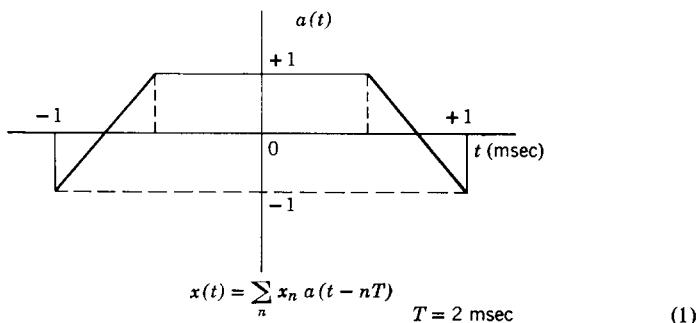
Both  $x_m(i)$  and  $\varphi_m(i)$  are arbitrary except for the power constraint, but must not change within the one second interval.

(a) Find the capacity of the channel in nats/sec for arbitrary integers  $M_2 > M_1 > 0$ .

(b) State clearly the relation between your answer in (a) and the capacity of a white Gaussian noise channel with just a power and bandwidth constraint on the input.

(c) Find the limit of your answer in (a) as  $M_2 \rightarrow \infty$  with  $M_1$  fixed.

- (d) What is the significance of the capacity found in (a)? State clearly what would be meant by a block code for this channel with this set of constraints.
- 8.9.** Consider a communication system in which the transmitted waveform,  $x(t)$ , is made up of time translates of the basic waveform below.



- The  $x_n$  can be arbitrary random variables, but the average transmitted power is constrained to be at most  $S$ . The received waveform is equal to  $x(t)$  plus additive white Gaussian noise of spectral density  $N_0/2$ .
- (a) Find the capacity of the channel in natural units per second subject to the power constraint and the constraint of (1).
- (b) Repeat (a), with the repetition time  $T$  in (1) reduced to 1 msec, thus making the basic waveforms overlap.
- 8.10.** A continuous time channel has its output corrupted by additive white Gaussian noise of spectral density  $N_0/2$ . The input is restricted to be a sequence of sinusoidal pulses, each of duration  $T$  of the form

$$\sqrt{2S} \sin\left(\frac{2\pi k}{T} t + \varphi\right); \quad 0 < t \leq T$$

where  $\varphi$  can assume the values  $45^\circ, 135^\circ, -135^\circ, -45^\circ$ . The received signal over an interval  $T$  is decoded into one of the four signals on the basis of maximum likelihood decoding.

- (a) Find an appropriate orthonormal expansion for these four waveforms and represent the four waveforms by coefficients in this expansion.
- (b) Represent the received waveforms by the same expansion and on a sketch using these coefficients as axes show which received signals should be decoded into each transmitted signal.
- (c) Show that the decoding operation can be separated into a pair of decisions and that this separation corresponds to transforming the 4-input, 4-output discrete channel into a pair of independent binary symmetric channels in parallel; find the transition probabilities for these binary symmetric channels.
- (d) Find an expression for the capacity in bits per second of the combination channel and find its limit as  $T \rightarrow 0$ .

- 8.11.** A set of  $L$  statistically independent white Gaussian noise channels all go from a transmitter at point  $A$  to a receiver at point  $B$ . The power spectral density of the noise on the  $l$ th channel is  $N_0(l)/2$ . The transmitter is power-limited to a power  $SL$ . Find the channel capacity of the set of channels under each of the following conditions:

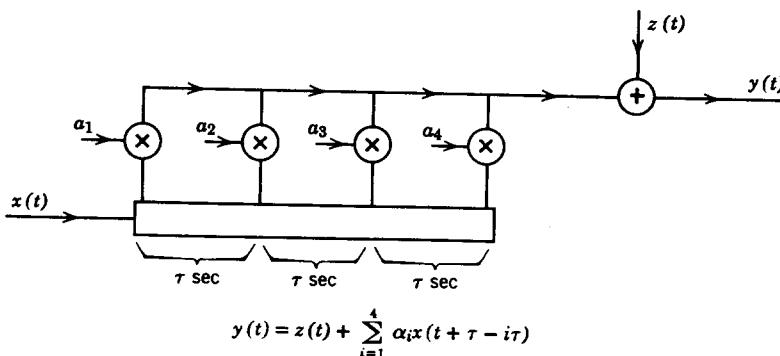
(a) The encoder can send arbitrary signals over each of the  $L$  channels and can divide the power between the channels in any way it desires. The receiver receives each of the  $L$  signals separately and can process the signals in any way it pleases.

(b) The transmitter is constrained to send the same signal over each channel; i.e.,  $s_1(t) = s_2(t) = \dots = s_L(t)$ , and  $s_l(t)$  has power  $S$  for  $1 \leq l \leq L$ . The receiver receives each signal separately and can process them in any way.

(c) The transmitter sends the same signal over each channel and the noisy signals are added together before being received.

*Hint:* In (b), expand the input and each output in an orthonormal expansion and show that no information is lost by taking an appropriate linear combination of the outputs.

- 8.12.** Suppose that a selection of one out of two equally likely messages, of duration  $T$  is to be sent over each of the three channels in the preceding problem. Find an optimum choice of the two signals in each case (subject to the power limitation) and find an expression for the resulting error probability.
- 8.13.** A model for a simple time-dispersive communication channel is shown below. The noise  $z(t)$  is additive, white, and Gaussian, with spectral density  $N_0/2$ . The tap gains are known positive constants,



(a) Assume a given set of code words  $x_1(t), \dots, x_M(t)$ . Determine and draw a block diagram of a maximum likelihood decoder. Assume that the values  $a_1, \dots, a_4$  are known.

(b) Assume that the *average* transmitted signal power is constrained to be  $S$  and that the allowable signal bandwidth is infinite. Find the channel capacity

in nats per second. Hint: Find the ratio of signal output power to input power for a sinusoid of long duration with a frequency divisible by  $1/\tau$ .

(c) Sketch and dimension the largest achievable  $E(R)$  such that  $P_e \leq \exp -TE(R)$  can be achieved in the limit of large  $T$  where  $R$  is the rate in nats per second.

- 8.14.** A doubly orthogonal code of  $M$  code words ( $M$  even) is a code for which the first  $M/2$  code words are equal energy orthogonal words, and the final  $M/2$  code words are the negatives of the first  $M/2$ . That is,

$$x_m(t) = \sqrt{\mathcal{E}} \varphi_m(t); \quad 1 \leq m \leq M/2$$

$$x_m(t) = -\sqrt{\mathcal{E}} \varphi_{m-M/2}(t); \quad M/2 + 1 \leq m \leq M$$

(a) Show that the upper bound on error probability for an orthogonal code on a white Gaussian noise channel of spectral density  $N_0/2$  in (8.2.43) and (8.2.44) also applies to a doubly orthogonal code.

(b) Consider next a decoder which, for each  $m$ ,  $1 \leq m \leq M/2$ , compares  $y_m = \int y(t) \varphi_m(t) dt$  with a fixed threshold  $A$ . If  $|y_m| \geq A$  for exactly one  $m$ , the decoder decodes  $m$  if  $y_m \geq A$  and  $m + M/2$  if  $y_m \leq -A$ . Otherwise the decoder refuses to decode. Let  $P_a$  be the probability that the decoder refuses to decode and let  $P_e$  be the probability of decoding in error. Show that  $P_a \leq P_1 + P_2$  and  $P_e \leq P_1 P_2$  where

$$P_1 = \int_{-\infty}^A \frac{1}{\sqrt{\pi N_0}} \exp \left[ -\frac{(y - \sqrt{\mathcal{E}})^2}{N_0} \right] dy$$

$$P_2 = M \int_A^{\infty} \frac{1}{\sqrt{\pi N_0}} \exp \left[ -\frac{y^2}{N_0} \right] dy$$

(c) Find the value of  $A$  which minimizes the above bound on  $P_a$  and express the resulting bounds on  $P_a$  and  $P_e$  in terms of an exponent, rate curve such as in (8.2.43) and (8.2.44).

(d) Choose  $A = \sqrt{\mathcal{E}} - \epsilon$  where  $\epsilon$  is small. Again express  $P_a$  and  $P_e$  in terms of exponent, rate curves. Sketch the exponent, rate curves found in (a), (c), and (d).

- 8.15.** (a) Define the distance between two waveforms  $x(t)$  and  $y(t)$  as

$$\sqrt{\int [x(t) - y(t)]^2 dt}$$

Show that the distance between any pair of code words in a simplex code is the same as the distance between any other pair of code words.

(b) Show that this distance is the square root of the upper bound on average squared distance given in (8.2.27). Show from this that the simplex codes are optimum in the sense of maximizing the distance between the two closest code words for a given number of code words and a given energy constraint.

(c) Let  $x_m = (x_{m,1}, \dots, x_{m,N})$  be the  $m$ th code word in a binary maximal

length code (see Section 6.6). Let  $\varphi_1(t), \dots, \varphi_N(t)$  be orthonormal functions. Show that the set of waveforms

$$x_m(t) = \sum_{n=1}^N (2x_{m,n} - 1)\varphi_n(t), \quad 1 \leq m \leq N + 1$$

forms a simplex code.

- 8.16.** Consider a discrete time channel consisting of an additive white Gaussian noise channel (spectral density  $N_0/2 = 1$ ) and a digital data modulator. In each interval of duration  $T_0$  the modulator transmits one of a set of  $K$  orthogonal waveforms, each of energy  $ST_0$  and limited to the given interval. Consider the channel output for a given interval as the outputs from a set of  $K$  matched filters, one matched to each modulation waveform. Show that the function  $E_0(1, Q)$  for the channel, using  $Q(k) = 1/K$ ,  $0 < k < K - 1$ , is given by

$$E_0(1, Q) = \frac{ST_0}{4} - \ln \left[ 1 + \frac{1}{K} (e^{ST_0/4} - 1) \right]$$

Show that for any rate  $R$  (in nats per second) and block length  $N$  there exist codes for this discrete time channel with

$$P_e < \exp \left[ TR - T \frac{S}{4} \left\{ 1 - \frac{4}{ST_0} \ln \left[ 1 + \frac{e^{ST_0/4} - 1}{K} \right] \right\} \right]$$

where  $T = NT_0$ .

Show that for  $R < S/8$  and for any fixed  $T_0$ , the exponent here approximates that for orthogonal code words as  $K$  becomes large and discuss qualitatively how large  $K$  must be for this approximation to be close. Hint: The output alphabet for this channel is the set of  $K$  dimensional vectors, hence

$$E_0(\rho, Q) = - \ln \int_{y_1} \cdots \int_{y_K} \left\{ \sum_{k=1}^K Q(k) p(y_1, \dots, y_K | x = k)^{\frac{1}{1+\rho}} \right\}^{1+\rho} dy_1 \cdots dy_K$$

For  $\rho = 1$ , you can expand the square and integrate in closed form. See Wozencraft and Kennedy (1966).

- 8.17.** The coefficients in the bounds on  $P_{e,m}$  for orthogonal code words in (8.2.43) and (8.2.44) can be avoided by replacing (8.2.35) by

$$Q(y_m) \leq [(M-1)\Phi(-y_m)]^\rho \leq (M-1)^\rho \exp \left[ -\frac{y_m^2 \rho}{2} \right]; \quad y_m \geq 0$$

valid for any  $\rho$ ,  $0 \leq \rho \leq 1$ . Substitute this in (8.2.32), integrate, and derive the exponential bound of (8.2.43) and (8.2.44) by optimizing over  $\rho$ . Note: your bound will not have the coefficient in (8.2.43) and (8.2.44).

- 8.18.** Let

$$x_m(t) = \sqrt{2S} \cos \left( \frac{2\pi mt}{T} \right);$$

$0 \leq t \leq T; 1 \leq m \leq M$  be a set of orthogonal code words, and suppose that when message  $m$  is sent, the received waveform is

$$y(t) = \sqrt{2S} \cos \left[ \frac{2\pi mt}{T} + \theta \right] + z(t); \quad 0 \leq t \leq T$$

where  $\theta$  is a random phase, uniformly distributed between 0 and  $2\pi$ , and  $z(t)$  is white Gaussian noise of spectral density  $N_0/2$ . Let

$$y_{m,1} = \int_0^T y(t) \sqrt{\frac{2}{T}} \cos \left( \frac{2\pi mt}{T} \right) dt$$

$$y_{m,2} = \int_0^T y(t) \sqrt{\frac{2}{T}} \sin \left( \frac{2\pi mt}{T} \right) dt$$

and assume that decoding is accomplished by choosing the  $m$  that maximizes  $r_m = y_{m,1}^2 + y_{m,2}^2$ .

(a) Show that the probability of error, given that message  $m$  is transmitted, is given by

$$\begin{aligned} P_{e,m} &= \int_{y_{m,1}} \int_{y_{m,2}} p(y_{m,1}, y_{m,2} | m) \Pr[r_{m'} \geq r_m, \text{any } m' \neq m | r_m, m] \\ &\leq (M-1)^\rho \int_{y_{m,1}} \int_{y_{m,2}} p(y_{m,1}, y_{m,2} | m) \Pr[r_{m'} \geq r_m | r_m, m]^\rho \end{aligned} \quad (\text{i})$$

for any  $\rho$ ,  $0 \leq \rho \leq 1$ .

(b) Show that for  $m$  transmitted,  $r_m$  has the probability density  $N_0^{-1} \exp[-r_m/N_0]$  and thus

$$\Pr[r_{m'} \geq r_m | r_m, m] = \exp[-r_m/N_0]$$

Substituting this in (i), show that the right-hand side of (i) is

$$\frac{(M-1)^\rho}{1+\rho} \exp \left[ -\frac{\rho ST}{N_0(1+\rho)} \right] \quad (\text{ii})$$

*Hint:* First perform the integration conditional on  $\theta = 0$  and then show that the result is the same for any value of  $\theta$ .

(c) Show that for  $M = 2$ ,  $\rho = 1$ , (ii) gives  $P_{e,m}$  exactly. For arbitrary  $M$ , upper-bound  $(M-1)^\rho/(1+\rho)$  by  $M^\rho$  and show that

$$P_{e,m} \leq \begin{cases} \exp[-T(\sqrt{C} - \sqrt{R})^2]; & \frac{1}{4}C \leq R \leq C \\ \exp \left[ -T \left( \frac{C}{2} - R \right) \right]; & R \leq \frac{1}{4}C \end{cases}$$

where  $C = S/N_0$ . [Grettenberg (1968).]

- 8.19.** Verify the validity of (8.5.89). Start with the continuity of  $\tilde{E}_\infty(\rho, B)$  and show that for any  $\epsilon > 0$  there exists a  $\delta_1$  and  $\delta_2$  such that

$$\tilde{E}_\infty(B + \delta_2, \rho - \delta_1) \geq \tilde{E}_\infty(B, \rho) - \epsilon$$

$$\tilde{S}_\infty(B + \delta_2, \rho - \delta_1) \leq \tilde{S}_\infty(B, \rho)$$

$$\tilde{R}_\infty(B + \delta_2) \geq \tilde{R}_\infty(B)$$

Then show that for large enough  $T$ , (8.5.85), used with the arguments  $B + \delta_2, \rho - \delta_1$ , implies (8.5.89).

- 8.20.** Show that for constant  $R, E$  as a function of  $S$  (as shown in Fig. 8.5.9) has a continuous derivative everywhere.
- 8.21.** The input to a channel over an interval  $(-T/2, T/2)$  is one of the two waveforms  $x_1(t) = A \cos(2\pi f_1 t)$  or  $x_2(t) = A \cos(2\pi f_2 t)$ . The output from the channel over the interval  $(-T/2, T/2)$  when the  $i$ th waveform is sent ( $i = 1, 2$ ) is

$$y(t) = v_{1,i} \cos 2\pi f_i t + v_{2,i} \sin 2\pi f_i t + z(t)$$

where  $z(t)$  is white Gaussian noise of spectral density  $N_0/2$  and where  $v_{1,i}$  and  $v_{2,i}$  are zero mean independent Gaussian random variables, each of variances  $\mathcal{E}/T$ , note that  $\mathcal{E}$  is the total average received signal energy in the interval.

(a) Let

$$y_{1,i} = \int_{-T/2}^{T/2} y(t) \frac{2}{\sqrt{T}} \cos(2\pi f_i t) dt$$

$$y_{2,i} = \int_{-T/2}^{T/2} y(t) \frac{2}{\sqrt{T}} \sin(2\pi f_i t) dt$$

Show that if message  $i$  is sent,  $y_{1,i}$  and  $y_{2,i}$  are independent Gaussian random variables with variance  $N_0 + \mathcal{E}$  each and that  $y_{1,j}$  and  $y_{2,j}, j \neq i$ , are independent Gaussian random variables of variance  $N_0$ . (Assume that  $f_1 T$  and  $f_2 T$  are integers.)

(b) Let  $\mathbf{y} = (y_{1,1}, y_{2,1}, y_{1,2}, y_{2,2})$  and calculate

$$r(\mathbf{y}) = \ln \frac{p(\mathbf{y} \mid x_1(t))}{p(\mathbf{y} \mid x_2(t))}$$

Show that the probability density of  $r$ , given  $x_1(t)$  transmitted, is

$$p(r \mid x_1(t)) = \begin{cases} \frac{N_0(N_0 + \mathcal{E})}{\mathcal{E}(2N_0 + \mathcal{E})} \exp \left[ -\frac{N_0}{\mathcal{E}} r \right]; & r \geq 0 \\ \frac{N_0(N_0 + \mathcal{E})}{\mathcal{E}(2N_0 + \mathcal{E})} \exp \left[ \frac{N_0 + \mathcal{E}}{\mathcal{E}} r \right]; & r \leq 0 \end{cases}$$

Show that  $p(r \mid x_2(t)) = p(-r \mid x_1(t))$ .

(c) Use your result in (b) to show that the probability of error in detecting which of these waveforms was transmitted using maximum likelihood decoding is given by

$$P_e = \frac{1}{2 + \mathcal{E}/N_0}$$

(d) Compare this result with the bound on error probability given by (8.6.22), taking  $\lambda_1 = \mathcal{E}$  and  $\lambda_j = 0$  for  $j > 1$ .

(e) In the integral equation (8.6.7), take  $T_1 = T$  and assume that  $\mathcal{R}(\tau) = \mathcal{E}/T$  for all  $\tau$  in the range  $(-T, T)$ . Show that this implies that  $\lambda_1 = \mathcal{E}$  and  $\lambda_j = 0$  for  $j > 1$ . State in qualitative terms what the above assumption implies about the magnitude of the Doppler spread on the channel relative to  $T$ .

- 8.22.** (a) Using the communication model and notation of Section 8.6, show that maximum likelihood decoding can be accomplished by choosing the message  $m$  for which

$$\sum_{i=1}^2 \sum_{j=1}^{\infty} y_{i,m,j}^2 \frac{\lambda_j}{N_0 + \lambda_i}$$

is maximum.

(b) Show that the above sum is equal to

$$\sum_{i=1}^2 \int_{-T_1/2}^{T_1/2} \left\{ \int_{-T_1/2}^{T_1/2} y_{i,m}(\tau) \left[ \sum_j \sqrt{\frac{\lambda_j}{N_0 + \lambda_j}} \varphi_j(t) \varphi_j(t) \right] d\tau \right\}^2 dt$$

Show that the term in brackets can be interpreted as the impulse response from an appropriately chosen time-varying linear filter.

- 8.23.** Suppose that the eigenvalues  $\lambda_j$  in (8.6.7) have the property that

$$\lambda_j = \begin{cases} \lambda; & 1 \leq j \leq n \\ 0; & j \end{cases}$$

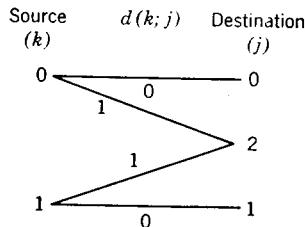
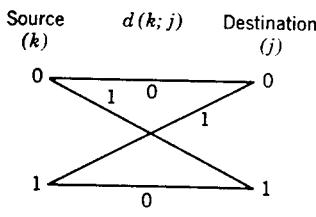
where  $n = ST_1/\lambda$ . For  $\rho = 1$ , sketch  $E_0(\rho, T)$  as a function of  $\lambda/N_0$ , holding  $ST$  fixed. Also sketch

$$\lim_{\rho \rightarrow 0} \frac{E_0(\rho, T)}{\rho}$$

as a function of  $\lambda/N_0$  for fixed  $ST$ . What does this say about desirable values of signal-to-noise ratio per degree of freedom in the received signal (i.e.,  $\lambda/N_0$ ) for low and high transmission rates?

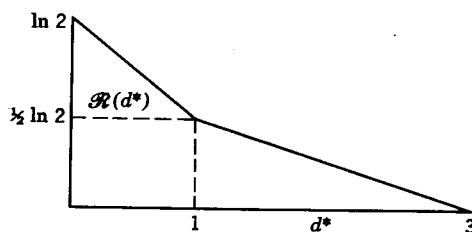
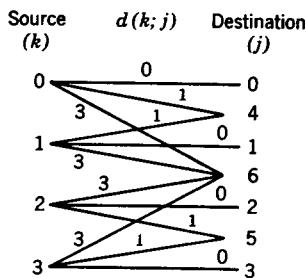
## CHAPTER 9

- 9.1** A source produces independent, equiprobable binary digits. Find and sketch the rate distortion function of the source for each of the distortion measures shown below. Omitted transitions in the diagrams correspond to infinite distortion.



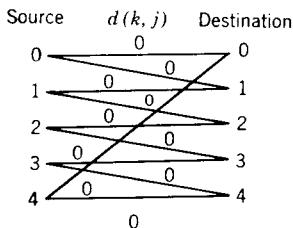
*Hint:* Use the symmetry to guess  $P(j|k)$  and check your result by the convexity of  $\mathcal{I}(\mathbf{Q}, \mathbf{P})$  in  $\mathbf{P}$ . Note that the second distortion measure provides an example where  $R(d^*)$  is not strictly convex  $\cup$ .

- 9.2. (a) Consider the following encoding scheme for the source and first distortion measure of the preceding problem. Break up the source output into sequences of 7 digits each. For a given (7,4) Hamming parity check code, encode each sequence of 7 digits into the 4 information digits of the code word closest to the given sequence. At the destination, represent the source sequence by the chosen code word. The rate of such a scheme is  $(4/7) \ln 2$  nats per source letter. Find the average distortion for such a scheme and compare with  $R(d^*)$ .  
(b) For arbitrary  $l$ , use the same scheme with a  $(2^l - 1, 2^l - l - 1)$  Hamming code and find the rate and average distortion.
- 9.3. For the source and second distortion measure of Problem 9.1 find a simple encoding scheme for which the rate for any desired average distortion is equal to  $R(d^*)$  evaluated at the desired average distortion.
- 9.4. A source produces independent equiprobable letters from an alphabet of size 4. Show that the rate distortion function for the source and the distortion measure given below is as shown.



- 9.5. A source produces independent equiprobable letters from an alphabet of 5 letters. The distortion measure is as shown below (omitted transitions correspond to infinite distortion and included transitions to zero distortion).  
(a) Find the rate-distortion function for this source and distortion measure.

(b) Show that for any rate  $R > \ln(5/2)$ , there exist codes of sufficiently long block length  $N$  with at most  $e^{RN}$  code words and zero distortion.



*Hint:* Use lemma 9.3.1 and observe that if, over the ensemble of codes,  $P_e(D > 0) < 5^{-N}$ , then at least one code must have zero distortion. [Pinkston (1967).]

In a sense this problem is the dual of that in Problem 5.11.b. It is curious, however, that  $C_0$  is unknown in that problem, whereas the corresponding result here is so simple.

- 9.6. A source produces independent letters from a ternary alphabet with the probabilities  $Q(0) = 0.4$ ,  $Q(1) = 0.4$ ,  $Q(2) = 0.2$ . The destination alphabet is ternary, and the distortion measure is  $d(k; j) = 0$  for  $k = j$ ;  $d(k; j) = 1$  for  $d \neq j$ . Find and sketch  $R(d^*)$ . As a function of  $C$ , find and sketch the minimum error probability per source digit that can be achieved transmitting this source over a channel of capacity  $C$ .
- 9.7. Consider a discrete-time memoryless source whose output is a sequence of real-valued random variables, each with the probability density  $q(u)$ , variance  $A$ , and entropy  $H(U) = -\int q(u) \ln[q(u)] du$ . Show that for  $d^* \leq A$ ,  $R(d^*)$  lies between the limits

$$H(U) - \frac{1}{2} \ln(2\pi e d^*) \leq R(d^*) \leq \frac{1}{2} \ln \frac{A}{d^*}$$

*Hint:* For the lower bound, review the argument from (9.7.2) to (9.7.8). For the upper bound, consider the test channel given in Fig. 9.7.3.

- 9.8. (a) Find the rate distortion function  $R(d^*)$  for a source whose output is a stationary Gaussian random process with spectral density

$$F(f) = \begin{cases} A; & |f| \leq W_1 \\ 0; & |f| > W_1 \end{cases}$$

The distortion measure is square difference.

(b) Find the capacity of an additive white Gaussian noise channel with power constraint  $S$ , noise spectral density  $N_0/2$  and bandwidth constraint  $W_2$ .

(c) Find an expression for the minimum mean square error that can be achieved transmitting the source in (a) over the channel in (b). Sketch and dimension this minimum mean square error as a function of  $W_2$ .

## REFERENCES AND SELECTED READINGS

- Abramson, N. (1963), *Information Theory and Coding*, McGraw-Hill, New York.
- Akheiser, N. I., and I. M. Glazman (1961), *Theory of Linear Operators in Hilbert Space*, Vol. 1, Ungar, New York.
- Artin, E. (1946), "Galois Theory," *Notre Dame Mathematical Lectures*, Notre Dame, Indiana.
- Ash, R. B. (1965), *Information Theory*, Interscience Publishers, New York.
- Berger, J. M., and B. M. Mandelbrot (1963), "A New Model for Error Clustering in Telephone Circuits," *IBM J. Res. and Dev.*, 7, 224-236.
- Berlekamp, E. R. (1964), "Note on Recurrent Codes," *IEEE Trans. Inform. Theory*, IT-10, 257-258.
- Berlekamp, E. R. (1967), "Nonbinary BCH Decoding," *IEEE Int. Symp. on Inform. Theory*, San Remo, Italy [also in Chaps. 7 and 10 of Berlekamp (1968)].
- Berlekamp, E. R. (1968), *Algebraic Coding Theory*, McGraw-Hill, New York.
- Billingsley, P. (1965), *Ergodic Theory and Information*, Wiley, New York.
- Birkhoff, G., and S. Mac Lane (1941), *A Survey of Modern Algebra*, Macmillan, New York.
- Blackwell, D. (1957), "The Entropy of Functions of Finite-state Markov Chains," *Transactions of the First Prague Conference on Information Theory, Statistical Decision Functions, and Random Processes*, Publishing House of the Czechoslovak Academy of Sciences, Prague, 13-20.
- Blackwell, D. (1961), "Exponential Error Bounds for Finite State Channels," *Proc. Fourth Berkeley Symp. on Math. Stat. and Prob.*, Univ. of California Press, Berkeley, Calif., 1, 57-63.
- Blackwell, D., L. Breiman, and A. J. Thomasian (1958), "Proof of Shannon's Transmission Theorem for Finite-state Indecomposable Channels," *Ann. Math. Stat.* 29, 1209-1220.
- Blackwell, D., L. Breiman, and A. J. Thomasian, (1959), "The Capacity of a Class of Channels," *Ann. Math. Stat.* 30, 1229-1241.
- Blackwell, D., and M. A. Girshick (1954), *Theory of Games and Statistical Decisions*, Wiley, New York.

- Bluestein G., and K. L. Jordan (1963), "An Investigation of the Fano Sequential Decoding Algorithm by Computer Simulation," *MIT-Lincoln Laboratory, Group Rept.* 62G-5.
- Bose, R. C., and D. K. Ray-Chaudhuri (1960), "On a Class of Error Correcting Binary Group Codes," *Inform. and Control*, **3**, 68-79.
- Bose, R. C., and D. K. Ray-Chaudhuri (1960), "Further Results on Error Correcting Binary Group Codes," *Inform. and Control*, **3**, 279-290.
- Breiman, L. (1957), "The Individual Ergodic Theorem of Information Theory," *Ann. Math. Stat.*, **28**, 809-811; correction to this paper, *Ann. Math. Stat.*, **31**, 809-810 (1960).
- Buck, R. C. (1956), *Advanced Calculus*, McGraw-Hill, New York.
- Carmichael, R. D. (1956), *Introduction to the Theory of Groups of Finite Order*, Dover, New York.
- Chernoff, H. (1952), "A Measure of Asymptotic Efficiency for Tests of a Hypothesis Based on a Sum of Observations," *Ann. Math. Stat.* **23**, 493-507.
- Chien, R. T. (1964), "Cyclic Decoding Procedures for the Bose-Chaudhuri-Hocquenghem Codes," *IEEE Trans. Inform. Theory*, **IT-10**, 357-363.
- Courant, R., and D. Hilbert (1959), *Methods of Mathematical Physics*, Vol. 1, Interscience, New York.
- Cox, D. R., and H. D. Miller (1965), *The Theory of Stochastic Processes*, Wiley, New York.
- Cramer, H. (1937), "Random Variables and Probability Distributions," *Cambridge Tracts in Mathematics No. 36*, Cambridge, England.
- Davenport, W. B., and W. L. Root (1958), *Random Signals and Noise*, McGraw-Hill, New York.
- Dobrushin, R. L. (1959), "General Formulation of Shannon's Main Theorem in Information Theory," *Usp. Math. Nauk*, **14**, No. 6 (90), 3-104, translated in *Am. Math. Soc. Translations*, **33**, Series 2, 323-438.
- Doob, J. L. (1953), *Stochastic Processes*, Wiley, New York.
- Eastman, W. L. (1965), "On the Construction of Comma-free Codes," *IEEE Trans. Inform. Theory*, **IT-11**, 263-267.
- Ebert, P. M. (1966), "Error Bounds for Parallel Communication Channels," *MIT Research Lab. of Electronics, Tech. Rept.* 448.
- Eisenberg, E. (1963), "On Channel Capacity," *Technical Memorandum M-35*, Electronics Research Laboratory, Univ. of California, Berkeley, Calif.
- Elias, P. (1954), "Error Free Coding," *IRE Trans. Inform. Theory*, **IT-4**, 29-37.
- Elias, P. (1955), "Coding for Noisy Channels," *IRE Convention Record*, Part. 4, 37-46.
- Elias, P. (1956), "Coding for Two Noisy Channels," in C. Cherry (Ed.), *Information Theory*, Butterworth, London.
- Elias, P. (1957), "List Decoding for Noisy Channels," *MIT Research Lab. of Electronics Tech. Rept.* 335.
- Elias, P. (1961), "Channel Capacity without Coding," in *Lectures on Communication System Theory*, E. J. Baghdady (Ed.), McGraw-Hill, New York; also *MIT Research Lab. of Elect. QPR*, Oct. 1956, 90-93.
- Elias, P. (1967), "Networks of Gaussian Channels with Applications to Feedback Systems," *IEEE Trans. Inform. Theory*, **IT-13**, 493-501.

- Elspas, B., and R. A. Short (1962) "A Note of Optimum Burst-Error-Correcting Codes," *IRE Trans. Inform. Theory*, **IT-8**, 39-42.
- Falconer, D. D. (1966), "A Hybrid Sequential and Algebraic Decoding Scheme," Ph.D. Thesis, Dept. of E.E., MIT, Cambridge, Mass.
- Fano, R. M. (1952), "Class Notes for Transmission of Information," Course 6.574, MIT, Cambridge, Mass.
- Fano, R. M. (1961), *Transmission of Information*, MIT Press, Cambridge, Mass. and Wiley, New York.
- Fano, R. M. (1963), "A Heuristic Discussion of Probabilistic Decoding," *IEEE Trans. Inform. Theory*, **IT-9**, 64-74.
- Feinstein, A. (1954), "A New Basic Theorem of Information Theory," *IRE Trans. Inform. Theory*, **PGIT-4**, 2-22.
- Feinstein, A. (1955), "Error Bounds in Noisy Channels without Memory," *IRE Trans. Inform. Theory*, **IT-1**, 13-14.
- Feinstein, A. (1958), *Foundations of Information Theory*, McGraw-Hill, New York.
- Feller, W. (1950), *An Introduction to Probability Theory and its Applications*, Vol. 1, Wiley, New York (3rd. ed., 1968).
- Feller, W. (1966), *An Introduction to Probability Theory and its Applications*, Vol. 2, Wiley, New York.
- Fire, P. (1959), "A Class of Multiple-Error-Correcting Binary Codes for Non-Independent Errors," *Sylvania Report RSL-E-2*, Sylvania Reconnaissance Systems Laboratory, Mountain View, Calif.
- Forney, G. D. (1965), "On Decoding BCH Codes," *IEEE Trans. Inform. Theory*, **IT-11**, 549-557.
- Forney, G. D. (1967), *Concatenated Codes*, MIT Press, Cambridge, Mass.
- Forney, G. D. (1968), "Exponential Error Bounds for Erasure, List, and Decision Feedback Schemes," *IEEE Trans. Inform. Theory*, **IT-14**, 206-220.
- Gallager, R. G. (1962), "Class Notes Distributed for Courses 6.575, 6.626," MIT, Cambridge, Mass.
- Gallager, R. G. (1964), "Information Theory," Chapter 4 of *The Mathematics of Physics and Chemistry*, H. Margenau and G. M. Murphy (Eds.), Van Nostrand, Princeton, N.J., Vol. 2.
- Gallager, R. G. (1965), "A Simple Derivation of the Coding Theorem and some Applications," *IEEE Trans. Inform. Theory*, **IT-11**, 3-18.
- Gallager, R. G. (1965b), "Lower Bounds on the Tails of Probability Distributions," *MIT Research Lab. of Electronics, QPR 77*, 277-291.
- Gantmacher, F. R. (1959), *Applications of the Theory of Matrices*, Interscience Publishers, New York.
- Gelfand, I. M., and A. M. Yaglom (1957), "Calculation of the Amount of Information about a Random Function contained in another such Function," *Usp. Mat. Nauk*, **12**, no. 1, 3-52.
- Gilbert, E. N. (1952), "A Comparison of Signalling Alphabets," *Bell System Tech. J.*, **31**, 504-522.
- Gilbert, E. N. (1960), "Capacity of Burst Noise Channel," *Bell System Tech. J.*, **39**, 1253-1256.
- Goblick, T. J., Jr. (1965), "Theoretical Limitations on the Transmission of

- Data from Analogue Sources," *IEEE Trans. Inform. Theory*, **IT-11**, 558-567.
- Goblick, T. J., and J. L. Holsinger (1967), "Analog Source Digitization: A Comparison of Theory and Practice," *IEEE Trans. Inform. Theory*, **IT-13**, 323-326.
- Golay, M. J. E. (1954), "Binary Coding," *IRE Trans. Inform. Theory*, **PGIT-4**, 23-28.
- Golumb, S. W., B. Gordon, and L. R. Welch (1958), "Comma-Free Codes," *Canad. J. Math.*, **10**, No. 2, 202-209.
- Grenander, U., and G. Szego (1958), *Toeplitz Forms and their Applications*, Univ. of Calif. Press, Berkeley, Calif.
- Grettenberg, T. L. (1968), "Exponential Error Bounds for Incoherent Orthogonal Signals," *IEEE Trans. Inform. Theory*, **IT-14**, 163-164.
- Hagelbarger, D. W. (1959). "Recurrent Codes: Easily Mechanized, Burst-Correcting, Binary Codes," *Bell System Tech. J.*, **38**, 969-984.
- Halmos, P. R. (1950), *Measure Theory*, Van Nostrand, Princeton, N.J.
- Hamming, R. W. (1950), "Error Detecting and Error Correcting Codes," *Bell System Tech. J.*, **29**, 147-160.
- Hardy, G. H., J. E. Littlewood, and G. Polya (1934), *Inequalities*, Cambridge Univ. Press, London (2nd ed., 1952, 1959).
- Hocquenghem, A. (1959), "Codes Correcteurs D'erreurs," *Chiffres*, **2**, 147-156.
- Holsinger, J. L. (1964), "Digital Communication over fixed Time-continuous Channels with Memory, with Special Application to Telephone Channels," *MIT Research Lab. of Electronics, Tech. Rept. 430* (also *MIT Lincoln Lab. T.R. 366*).
- Huffman, D. A. (1962), "A Method for the Construction of Minimum Redundancy Codes," *Proc. IRE*, **40**, 1098-1101.
- Iwadare, Y. (1967), "Simple and Efficient Procedures of Burst-error Correction," Ph.D. Thesis, Univ. of Tokyo.
- Jacobs, I. (1963), "The Asymptotic Behavior of Incoherent M-ary Communication Systems," *Proc. IRE*, **51**, 251-252.
- Jacobs, I. M., and E. R. Berlekamp (1967), "A Lower Bound to the Distribution of Computation for Sequential Decoding," *IEEE Trans. Inform. Theory*, **IT-13**, 167-174.
- Jelinek, F. (1968), "Evaluation of Expurgated Bound Exponents," *IEEE Trans. Inform. Theory*, **IT-14**, 501-505.
- Jelinek, F. (1968), *Probabilistic Information Theory*, McGraw-Hill, New York.
- Kac, M., W. L. Murdock, and G. Szego (1953), "On the Eigenvalues of Certain Hermitian Forms," *J. Rat. Mech. and Anal.*, **2**, 767-800.
- Kailath, T. (1967), "A Projection Method for Signal Detection in Colored Gaussian Noise," *IEEE Trans. Inform. Theory*, **IT-13**, 441-447.
- Karush, J. (1961), "A Simple Proof of an Inequality of McMillan," *IRE Trans. Inform. Theory*, **IT-7**, 118.
- Kasami, T. (1963), "Optimum Shortened Cyclic Codes for Burst-error Correction," *IEEE Trans. Inform. Theory*, **IT-9**, 105-109.
- Kelly, E. J., I. S. Reed, and W. L. Root (1960), "The Detection of Radar Echoes in Noise," *J. Soc. Ind. Appl. Math.*, **8**, 309-341.
- Kelly, J. L. (1956), "A New Interpretation of Information Rate," *Bell System Tech. J.*, **35**, 917-926.

- Kemperman, J. H. B. (1962), "Studies in Coding Theory," an unpublished memorandum, Rochester.
- Kendall, W. B., and I. S. Reed (1962), "Path Invariant Comma free Codes," *IRE Trans. Inform. Theory*, **IT-8**, 350-355.
- Kennedy, R. S. (1963), "Finite State Binary Symmetric Channels," Sc.D. Thesis, Dept. of E.E., MIT, Cambridge, Mass.
- Kennedy, R. S. (1964), "Performance Limitations of Dispersive Fading Channels," *International Conference on Microwaves, Circuit Theory, and Information Theory*, Tokyo; abstract in *IEEE Trans. Inform. Theory*, **IT-10**, 398.
- Kennedy, R. S. (1969), *Fading Dispersive Communication Channels*, Wiley, New York.
- Khinchin, A. (1957), *Mathematical Foundations of Information Theory*, Dover, New York.
- Kohlenberg, A. (1965), "Random and Burst Error Control," *First IEEE Annual Communications Convention*, Boulder, Colo., June 7-9.
- Kolmogorov, A. N. (1941), "Interpolation and Extrapolation of Stationary Sequences," *Izv. Akad. Nauk, SSSR., Ser. Mat.* **5**, 3-14.
- Kolmogorov, A. N. (1956), "On the Shannon Theory of Information in the Case of Continuous Signals," *IRE Trans. Inform. Theory*, **IT-2**, 102-108.
- Kotel'nikov, V. A. (1947), *The Theory of Optimum Noise Immunity*, doctoral dissertation presented before the academic council of the Molotov Energy Institute in Moscow, Translation into English by R. A. Silverman, McGraw-Hill, New York, 1959.
- Kotz, S. (1965), "Recent Developments in Information Theory," *J. Applied Probability*.
- Kraft, L. G. (1949), "A Device for Quantizing, Grouping and Coding Amplitude Modulated Pulses," M.S. Thesis, Dept. of E.E., MIT, Cambridge, Mass.
- Kuhn, H. W., and A. W. Tucker (1951), "Nonlinear Programming," *Proc. 2nd Berkeley Symposium on Math. Stat. and Prob.*, Univ. of Calif. Press, Berkeley, 481-492.
- Landau, H. J., and H. O. Pollak (1961), "Prolate Spheroidal Wave Functions, Fourier Analysis, and Uncertainty-II," *Bell System Tech. J.*, **40**, 65-84.
- Landau, H. J., and H. O. Pollak (1962), "Prolate Spheroidal Wave Functions, Fourier Analysis, and Uncertainty-III," *Bell System Tech. J.*, **41**, 1295-1336.
- Loeve, M. (1955) *Probability Theory*, Van Nostrand, Princeton, N.J.
- Massey, J. L. (1963), *Threshold Decoding*, MIT Press, Cambridge, Mass.
- Massey, J. L. (1965), "Implementation of Burst Correcting Convolutional Codes," *IEEE Trans. Inform. Theory*, **IT-11**, 416-422.
- Massey, J. L. (1968), "Shift-Register Synthesis and BCH Decoding," *IEEE Trans. Inform. Theory*, to appear.
- Massey, J. L., and R. W. Liu (1964), "Application of Lyapunov's direct method to the Error Propagation Effect in Convolutional Codes," *IEEE Trans. Inform. Theory*, **IT-10**, 248-250.
- Max, J. (1960), "Quantizing for Minimum Distortion," *IRE Trans. Inform. Theory*, **IT-6**, 7-12.
- McMillan, B. (1953), "The Basic Theorems of Information Theory" *Ann. Math. Stat.*, **24**, 196-219.

- McMillan, B. (1956), "Two Inequalities Implied by Unique Decipherability," *IRE Trans. Inform. Theory*, **IT-2**, 115-116.
- Metzner, J. J., and K. C. Morgan (1960), "Coded Feedback Communication Systems," *National Electronics Conference*, Chicago, Ill., October.
- Ott, G. (1967), "Compact Encoding of Stationary Markov Sources," *IEEE Trans. Inform. Theory*, **IT-13**, 82-86.
- Peterson, W. W. (1960), "Encoding and Error-Correction Procedures for the Bose-Chaudhuri Codes," *IRE Trans. Inform. Theory*, **IT-6**, 459-470.
- Peterson, W. W. (1961), *Error Correcting Codes*, MIT Press Cambridge, Mass., and Wiley, New York.
- Peterson, W. W., and J. L. Massey (1963), "Report on Progress in Information Theory in the U.S.A., 1960-1963, Coding Theory," *IEEE Trans. Inform. Theory*, **IT-9**, 223-229.
- Pierce, J. N. (1961), "Theoretical Limitations on Frequency and Time Diversity for Fading Binary Transmission," *IRE Trans. on Communication Systems*, **CS-9**, 186-187.
- Pilc, R. J. (1967), "Coding Theorems for Discrete Source-Channel Pairs," Ph.D. thesis, Dept. of E.E., MIT, Cambridge, Mass.
- Pinkston, J. T. (1967), "Encoding Independent Sample Information Sources," *MIT Research Lab. of Electronics, Tech. Rept.* 462.
- Pinsker, M. S. (1957), "Calculation and Estimation of the Quantity of Information, The Capacity of a Channel, and the Rate of Production of Information, in terms of the Second Moments of the Distributions," Doctoral dissertation, Moscow.
- Pinsker, M. S. (1964), *Information and Information Stability of Random Variables and Processes*, translation into English by A. Feinstein, Holden-Day, San Francisco.
- Plotkin, M. (1951), "Binary Codes with Specified Minimum Distance," *IRE Trans. Inform. Theory*, **IT-6**, 445-450 (1960). Also *Research Division Report 51-20*, University of Pennsylvania.
- Prange, E. (1957), "Cyclic Error-Correcting Codes in Two Symbols," AFCRC-TN-57-103, Air Force Cambridge Research Center, Cambridge, Mass.
- Prange, E. (1958), "Some Cyclic Error-Correcting Codes with Simple Decoding Algorithms," AFCRC-TN-58-156, Air Force Cambridge Research Center, Bedford, Mass.
- Reed, I. S. (1954), "A Class of Multiple-Error-Correcting Codes and the Decoding Scheme," *IRE Trans. Inform. Theory*, **IT-4**, 38-49.
- Reed, I. S., and G. Solomon (1960), "Polynomial Codes over Certain Finite Fields," *J. Soc. Indust. Appl. Math.*, **8**, 300-304.
- Reiffen, B. (1960), "Sequential Encoding and Decoding for the Discrete Memoryless Channel," *MIT Research Lab. of Electronics Tech. Rept.* 374.
- Reiffen, B. (1963), "A Note on 'Very Noisy' Channels," *Inform. and Control*, **6**, 126-130.
- Reiffen, B. (1966), "A Per Letter Converse to the Channel Coding Theorem," *IEEE Trans. Inform. Theory*, **IT-12**, 475-480.
- Richters, J. S. (1967), "Communication over Fading Dispersive Channels," *MIT Research Lab. of Electronics, Tech. Rept.* 464.

- Riesz, F., and B. Sz-Nagy (1955), *Functional Analysis*, Ungar, New York.
- Sakrison, D. (1968), *Communication Theory: Transmission of Waveforms and Digital Information*, Wiley, New York.
- Sardinas, A. A., and G. W. Patterson (1953), "A Necessary and Sufficient Condition for the Unique Decomposition of Coded Messages," *IRE Convention Record*, Part 8, 104-108.
- Savage, J. E. (1965), "The Computation Problem with Sequential Decoding," *MIT Research Lab. of Electronics, Tech. Rept.* 439 (also *MIT Lincoln Lab. T. R. 371*).
- Schalkwijk, J. P. M., and T. Kailath (1966), "A Coding Scheme for Additive Noise Channels with Feedback, Part 1," *IEEE Trans. Inform. Theory*, **IT-12**, 172-182.
- Schalkwijk, J. P. M. (1966), "A coding Scheme for Additive Noise Channels with Feedback, Part 2". *IEEE Trans. Inform. Theory*, **IT-12**, 183-189.
- Schalkwijk, J. P. M. (1968), "Center of Gravity Information Feedback," *IEEE Trans. Inform. Theory*, **IT-14**, 324-331.
- Scholtz, R. A. (1966), "Codes with Synchronization Capability," *IEEE Trans. Inform. Theory*, **IT-12**, 135-140.
- Shannon, C. E. (1948), "A Mathematical Theory of Communication," *Bell System Tech. J.*, **27**, 379-423 (Part I), 623-656 (Part II). Reprinted in book form with postscript by W. Weaver, Univ. of Illinois Press, Urbana, 1949.
- Shannon, C. E. (1949), "Communication in the Presence of Noise," *Proc. IRE*, **37**, 10-21.
- Shannon, C. E. (1956), "The Zero Error Capacity of a Noisy Channel," *IRE Trans. Inform. Theory*, **IT-2**, 8-19.
- Shannon, C. E. (1957), "Certain Results in Coding Theory for Noisy Channels," *Inform. and Control*, **1**, 6-25.
- Shannon, C. E. (1959), "Coding Theorems for a Discrete Source with a Fidelity Criterion," *IRE Nat. Conv. Record*, Part 4, 142-163. Also in *Information and Decision Processes*, R. E. Machol, Ed., McGraw-Hill, New York (1960).
- Shannon, C. E. (1959), "Probability of Error for Optimal Codes in a Gaussian Channel," *Bell System Tech. J.*, **38**, 611-656.
- Shannon, C. E. (1961), "Two-Way Communication Channels," in *Proc. Fourth Berkeley Symp. on Prob. and Stat.*, **1**, 611-644, University of California Press, Berkeley, Calif.
- Shannon, C. E., R. G. Gallager, and E. R. Berlekamp (1967), "Lower Bounds to Error Probability for Coding on Discrete Memoryless Channels," *Inform. and Control*, **10**, 65-103 (Part I), 522-552 (Part II).
- Slepian, D. (1956), "A Class of Binary Signaling Alphabets," *Bell System Tech. J.*, **35**, 203-234.
- Slepian, D. (1963), "Bounds on Communication," *Bell System Tech. J.* **42**, 681-707.
- Slepian, D. (1965), "Some Asymptotic Expansions for Prolate Spheroidal Wave Functions," *J. Math. and Phys.*, **44**, 99-140.
- Slepian, D., and H. O. Pollak (1961), "Prolate Spheroidal Wave Functions, Fourier Analysis, and Uncertainty-I," *Bell System Tech. J.*, **40**, 43-64 (see Landau and Pollak for Parts II and III).

- Stiglitz, I. G. (1966), "Coding for a class of Unknown Channels," *IEEE Trans. Inform. Theory*, **IT-12**, 189-195.
- Stiglitz, I. G. (1967), "A Coding Theorem for a Class of Unknown Channels," *IEEE Trans. Inform. Theory*, **IT-13**, 217-220.
- Thomasian, A. J. (1960), "An Elementary Proof of the AEP of Information Theory," *Ann. Math. Stat.*, **31**, 452-456.
- Thomasian, A. J. (1963), "A Finite Criterion for Indecomposable Channels," *Ann. Math. Stat.*, **34**, 337-338.
- Titchmarsh, E. C. (1937), *Introduction to the Theory of Fourier Integrals*, Oxford Univ. Press, London (2nd ed., 1948).
- Van Trees, H. L. (1965), "Comparison of Optimum Angle Modulation Systems and Rate-Distortion Bounds," *Proc. IEEE*, **53**, 2123-2124.
- Varshamov, R. R. (1957), "Estimate of the Number of Signals in Error Correcting Codes," *Dokl. Akad. Nauk SSSR*, **117**, No. 5, 739-741.
- Viterbi, A. J. (1961), "On Coded Phase-Coherent Communications," *IRE Trans. Space Elect. and Telemetry*, **SET-7**, 3-14.
- Viterbi, A. J. (1967), "Error Bounds for Convolutional Codes and an Asymptotically Optimum Decoding Algorithm," *IEEE Trans. Inform. Theory*, **IT-13**, 260-269.
- Viterbi, A. J. (1967b), "Performance of an M-ary Orthogonal Communication System Using Stationary Stochastic Signals," *IEEE Trans. Inform. Theory*, **IT-13**, 414-422.
- Wagner, T. J. (1968), "A Coding Theorem for Abstract Memoryless Channels," unpublished memorandum.
- Wald, A. (1947), *Sequential Analysis*, Wiley, New York.
- Wiener, N. (1949), *Extrapolation, Interpolation, and Smoothing of Stationary Time Series*, MIT Press, Cambridge, Mass., and Wiley, New York.
- Wolfowitz, J. (1957), "The Coding of Messages Subject to Chance Errors," *Illinois J. of Math.*, **1**, 591-606.
- Wolfowitz, J. (1961), *Coding Theorems of Information Theory*, Springer-Verlag and Prentice-Hall, Englewood Cliffs, N.J. (2nd ed., 1964).
- Wozencraft, J. M. (1957), "Sequential Decoding for Reliable Communication," *1957 National IRE Convention Record*, **5**, Part 2, 11-25; also *MIT Research Lab. of Electronics Tech. Rept.* 325, Cambridge, Mass.
- Wozencraft, J. M., and M. Horstein (1961), "Coding for Two-way Channels," *MIT Research Lab. of Electronics Tech. Rept.* 383, Cambridge, Mass.
- Wozencraft, J. M., and I. M. Jacobs (1965), *Principles of Communication Engineering*, Wiley, New York.
- Wozencraft, J. M., and B. Reiffen (1961), *Sequential Decoding*, MIT Press, Cambridge, Mass., and Wiley, New York.
- Wyner, A. D. (1966), "Capacity of the Band-limited Gaussian Channel," *Bell System Tech. J.*, **45**, 359-395.
- Wyner, A. D., and R. B. Ash (1963), "Analysis of Recurrent Codes," *IEEE Trans. Inform. Theory*, **IT-9**, 143-156.
- Yaglom, A. M. (1962), *An Introduction to the Theory of Stationary Random Functions*, translation into English by R. A. Silverman, Prentice-Hall, Englewood Cliffs, N.J.

- Yudkin, H. L. (1964), "Channel State Testing in Information Decoding," Sc.D. Thesis, Dept. of E.E., MIT, Cambridge, Mass.
- Yudkin, H. L. (1964), "An Error Bound for Gaussian Signals in Gaussian Noise," *MIT Research Lab. of Electronics QPR*, No. 73, 149-155, Cambridge, Mass.
- Yudkin, H. (1967), "On the Exponential Error Bound and Capacity for Finite State Channels," *International Symposium on Inform. Theory*, San Remo, Italy.
- Zetterberg, L. H. (1961), "Data Transmission over a Noisy Gaussian Channel," *Trans. Roy. Inst. Technol.*, Stockholm, No. 184.
- Zierler, N. (1960), "A Class of Cyclic Linear Error-Correcting Codes in  $p^m$  Symbols," MIT Lincoln Laboratory Group Report 55-19, Lexington, Mass.

## GLOSSARY OF SYMBOLS

|                     |   |
|---------------------|---|
| $\sum_{k=1}^n x_k$  | Sum; $x_1 + x_2 + \cdots + x_n$ .   |
| $\prod_{k=1}^n x_k$ | Product; $x_1 x_2 \cdots x_n$ .   |
| $\oplus$            | Modulo-2 sum.   |
| $\sum_{k=1}^n x_k$  | Modulo-2 sum; $x_1 \oplus x_2 \oplus \cdots \oplus x_n$ .   |
| $\lfloor x \rfloor$ | Largest integer less than or equal to $x$ (the integer part of $x$ ).   |
| $\lceil x \rceil$   | Smallest integer greater than or equal to $x$ .   |
| $ x $               | Magnitude of real number $x$ ( $ x  = x$ ; $x \geq 0$ , $ x  = -x$ ; $x < 0$ ).   |
| $x \approx y$       | $x$ is approximately equal to $y$ .   |
| $x \ll y$           | $x$ is negligible compared to $y$ .   |
| $n!$                | Factorial; $n! = 1 \cdot 2 \cdot 3 \cdots n$ .  |
| $\binom{n}{m}$      | Number of combinations of $n$ objects taken $m$ at a time;<br>$\binom{n}{m} = \frac{n!}{m!(n-m)!}$                              |
| $\bar{z}$           | Average value (expectation) of random variable $z$ .  |
| $f \triangleq g$    | $f$ is defined to be equal to $g$ .   |
| $\ f\ $             | Norm of function $f$ ;<br>$\ f\  = \sqrt{\int f^2(x) dx}$   |
| $(f \cdot g)$       | Inner product of functions $f$ and $g$ ;<br>$(f \cdot g) = \int f(x)g(x) dx$  |
| $[f(D)]_m^n$        | $f_m D^m + f_{m+1} D^{m+1} + \cdots + f_n D^n$ for $n \geq m$ and 0 for $m > n$<br>where $f(D) = \sum f_k D^k$ is a polynomial. |
| $A^c$               | Complement of set $A$ .   |
| $A \cup B$          | Union of sets $A$ and $B$ (that is, the set of elements in $A$ or $B$ or both).   |
| $\cup_m A_m$        | $A_1 \cup A_2 \cup A_3 \cup \cdots$   |
| $A \cap B$          | Intersection of sets $A$ and $B$ (that is, the set of elements commonly contained in sets $A$ and $B$ ).                        |
| $\ A\ $             | The number of elements in the set $A$ .   |
| $a \in A$           | The element $a$ is contained in the set $A$ .   |

|                            |  |
|----------------------------|--|
| $a : S$                    | The set of elements $a$ such that statement $S$ is satisfied. For example,   |
|                            | $\sum_{n:x_n > 1} x_n$   |
|                            | is the sum of $x_n$ over those $n$ such that $x_n > 1$ .   |
| $A = \{a : S\}$            | $A$ is defined as the set of elements $a$ for which statement $S$ is satisfied.  |
| $S_1 \Rightarrow S_2$      | Statement $S_1$ implies statement $S_2$ .  |
| $S_1 \Leftrightarrow S_2$  | Statement $S_1$ implies $S_2$ and $S_2$ implies $S_1$ .  |
| —                          | Denotes the end of a theorem.  |
|                            | Denotes the end of a proof.  |
| GCD                        | Greatest common divisor.   |
| $H(\ )$                    | Entropy; see Section 2.2.  |
| $\mathcal{H}(x)$           | $-x \log x - (1 - x) \log (1 - x); 0 \leq x \leq 1$ .  |
| $I(\ )$                    | Information; see Section 2.2.  |
| $\inf f(x)$                | Infimum; largest number never exceeding $f(x)$ over allowed range of $x$ . If the minimum exists, then $\inf f(x) = \min f(x)$ .     |
| LCM                        | Lowest common multiple.  |
| $\lim_{x \rightarrow y}$   | Limit as $x$ approaches $y$ .  |
| $\lim_{x \rightarrow y^+}$ | Limit as $x$ approaches $y$ from above.  |
| $\lim_{x \rightarrow y^-}$ | Limit as $x$ approaches $y$ from below.  |
| ln                         | Natural logarithm.   |
| log                        | Logarithm to arbitrary base ( $\log_n$ indicates logarithm to base $n$ ).  |
| $\max f(x)$                | Maximum value of $f(x)$ over allowed range of $x$ .  |
| $\min f(x)$                | Minimum value of $f(x)$ over allowed range of $x$ .  |
| $R_n(m)$                   | Remainder after dividing $m$ by $n$ ( $m$ and $n$ positive integers).  |
| $R_g(D)f(D)$               | Remainder after dividing polynomial $f(D)$ by $g(D)$ .   |
| $\sup f(x)$                | Supremum; smallest number never exceeded by $f(x)$ over allowed range of $x$ . If the maximum exists, then $\sup f(x) = \max f(x)$ . |



# INDEX

- Abramson, N., 37  
AEP (Asymptotic equipartition property), 70  
Akheizer and Glazman, 360, 396  
Alphabetic source codes, 513  
Analog to digital conversion, 443  
Ash, R., 70  
Associative law, 209  
Autocorrelation function, 363  
at output of linear filter in terms of input, 364  
  
Bandwidth compression, 443  
BCH codes, 238–258  
decoding by the iterative algorithm, 246  
effect of varying primitive element, 551  
minimum distance, 240  
asymptotic behavior, 257  
syndrome, 242  
Berlekamp, E., 245, 254, 301, 305  
Bessel's inequality, 357  
Binary symmetric channel, 17  
capacity, 92  
random coding exponent, 147  
sphere packing bound, 164  
straight line bound, 171  
Binomial coefficients, bounds on, 530  
Binomial distribution function, bounds on, 531  
Biological organisms as communication systems, 3  
Bits, 16  
Blackwell, D., 67  
Blackwell, Breiman, and Thomasian, 111, 188  
Blackwell and Girshick, 82  
Bluestein and Jordan, 278  
Bose and Chaudhuri, 238, 306  
Breiman, L., 70  
  
Buck, R. C., 74  
Burst correcting capability, 288  
Burst of errors, relative to guard space, 288  
for cyclic codes, 290  
  
Capacity, 8, 71  
binary symmetric channel, 92  
continuous time channel, 369  
discrete memoryless channel, 74  
calculation, 91–96  
upper bound and interpretation as min-max, 524  
discrete-time memoryless channel, 318  
additive noise, 335  
Gaussian additive noise, 335–343  
input constraints, 324  
fading dispersive channel, 439  
finite state channel, 97–112  
indecomposable, 109  
lower and upper capacities, 100  
no intersymbol interference, 543  
Gaussian additive noise channel, filtered input, 384  
heuristic derivation, 383  
white noise and constraint on degrees of freedom, 373  
white noise and no bandwidth constraint, 371–383  
parallel channels, discrete memoryless, 519  
Gaussian, 343, 522  
Cascaded channels, 25, 508, 527  
Central limit theorem, 191, 192  
Channels, discrete memoryless, 6, 73–97  
discrete with memory, 97–112  
discrete-time memoryless, 316–354  
additive noise, 333–343  
Gaussian additive noise, 335–354  
fading dispersive, 431–441  
mathematical model, 433

- see also* Fading dispersive channels
- finite state, 97–112
  - indecomposable, 105–112
  - state known at receiver, 182
- Gaussian additive noise, filtered input, 384, 407–430
  - white, 7, 371–383
- parallel, discrete time, Gaussian, 343–354
- Characteristic of Galois field, 225–238
- Chebyshev inequality, 126–127
- Check matrix, 221; *see also* Parity check, matrix
- Check polynomial of cyclic code, 223
- Chernoff bound, 127–131
- Chien, R. T., 254
- Codes, 116
  - BCH, 238–258
  - block, 116
    - ( $N, R$ ), 138
  - for burst noise channels, 286–306
  - concatenation of, 257
  - convolutional, 258–263
  - cyclic, 221
  - doubly orthogonal, 562
  - group, 220
  - Hamming, 201, 230–235
  - linear, 220
  - maximal length, 230–235, 562
  - orthogonal, 379
  - parity check, 196
  - Reed-Solomon, 257, 297
  - simplex, 378, 383
  - source, *see* Source codes
    - see also* above subtitles
- Coding theorem, 9–12, 116–119, 135
  - binary symmetric channel, 146, 531
  - continuous time channels, converse, 425
  - discrete channels, 135
  - discrete memoryless channels, 138, 143
    - alternate, capacity oriented, derivation, 533
    - alternate simplified derivation with weakened exponent, 534
  - block coding converse, 173
  - converse, 76
    - “strong converse,” 173
    - “weak converse,” 173
  - discrete-time, memoryless channels, 318
    - with constrained input, 331
      - converse, 324
    - converse, 320
- error detection and correction, 533
- fading dispersive channels, 439
- finite state channels, 176
  - converse, 102–108
  - noise independent of input, 543
  - state known at receiver, 185
- Gaussian additive noise channel with filtered input, 430
  - noisy channels; converse, 465
  - parity check codes, 206–209
  - source, *see* Source coding theorem
- Complete code tree, 54
- Compound channel, 176
- Concave function, 84
- Connected, 80
- Convex function, 82–91
- Convex region, 83
- Convolutional codes, 258–263
  - burst error correcting, 297
  - constraint length, 263
  - systematic, 263
  - tree structure, 264
- Correlation decoding, 375
- Correlation function, *see* Autocorrelation function
- Coset, 210
- Coset code, 206
- Courant and Hilbert, 396, 440
- Cox and Miller, 65, 312
- Cramer, H., 336
- Cyclic codes, 221
  - burst error correcting, 291
    - optimum decoder, 291
  - check polynomial, 223
  - encoder instrumentation, 224–225, 549
  - generator polynomial, 222
- Data processing theorem, 80
- Data reduction, 443
- Davenport and Root, 363, 440
- Decisions, effect on channel capacity, 526
- Decoding, 120; *see also* Codes; Sequential decoding; and Threshold decoding
- Decoding table, 202
- Degrees of freedom, 361
- Difference energy, bound on average value, 377
- Digital data demodulators, 8
- Digital data modulators, 7
- Distance, *see* Hamming distance
- Distortion measure, 443

- Distribution function**, 27  
 joint, 28  
**DMC**, *see* **Channels**  
**Dobrushin's theorem**, 36  
**Doob, J. L.**, 440  
**Doubly orthogonal codes**, 562  
**Dual codes**, 224  
**Duty factor of transmission**, 437

**Eastman, W. L.**, 70  
**Ebert, P.**, 123, 354  
**Eisenberg, E.**, 111  
**Elapas and Short**, 293  
**Elias, P.**, 188, 204, 306, 481  
**Energy equation**, 359  
**English text as Markov source**, 64  
**Ensemble**, 13  
   of block codes, 131, 150, 326  
   of convolutional codes, 274, 281  
   joint, 15  
**Entropy**, 5, 20, 23–27  
   conditional, 21  
   continuous ensemble, 31  
   convexity of, 85  
   discrete stationary source, 56  
   Markov source, 66  
   of one probability assignment relative to another, 508  
   thermodynamic, 20  
**Ergodic**, 59, 497  
   modes, 495–497  
   set of states in Markov chain, 64  
**Error detection and retransmission**, 286, 533  
**Error locators**, BCH codes, 242  
**Error sequence**, 201  
**Error values**, BCH codes, 242  
**Euclidian division algorithm**, 216  
**Exponent of error probability**,  $E_{ex}(R)$ , 153–157  
 $E_r(R)$ , 139–143  
 $E_{sl}(R)$ , 161  
 $E_{xp}(R)$ , 158  
*see also* **Expurgated exponent**; **Random coding exponent**  
**Expurgated exponent**,  $E_{ex}(R)$ , 153–157  
   discrete memoryless channel; calculation of  $R_{x_{\infty}}$ , 540  
   limit as  $\bar{R} \rightarrow 0$ , 540  
   maximization over  $\mathbf{Q}$ , 540  
   discrete-time memoryless channel, 323.

**331**  
**discrete-time Gaussian channel**, 340  
**Gaussian additive noise channel with filtered input**, 430  
**parallel**, discrete-time Gaussian channels, 352  
**Extension fields**, 227

**Fading dispersive channels**, 431–441  
   maximum likelihood receiver, 566  
   optimal choice of eigenvalues, 439  
**Falconer, D.**, 279  
**Fano, R. M.**, 37, 70, 173, 188, 266, 269, 270, 306, 440  
**Feedback**, binary erasure channel, 506  
   burst noise channels, 286–306  
   effect, on error exponent, 534  
   on sphere packing bound, 541  
   failure to change capacity of discrete memoryless channel, 520  
   use, in data transmission on additive Gaussian noise channel, 481  
   in transmitting Gaussian source, 479  
**Feinstein, A.**, 34, 188, 493  
**Feller, W.**, 37, 41, 65, 191, 192, 315, 330, 350, 380, 530, 542  
**Fermat's theorem**, 547  
**Fidelity criterion**, 445; *see also* **Source coding theorem**  
**Fields**, 213–214  
**Galois**, 214, 227  
   existence of, 237  
   implementation of operations, 234–235  
   integers of, 227  
   minimal polynomials, 227  
   orders of, 214  
   of polynomials modulo a polynomial, 219  
   primitive elements, 226  
**Filtered white noise expansions**, 398–406  
**Finite energy functions**, 356  
**Fire, P.**, 293, 306  
**Forney, G. D.**, 257, 534  
**Fourier series**, 360  
**Fourier transform of a truncated sinusoid**, 360  
**Frequency modulation**, 479  
**FSC**, *see* **Channels, finite state**  
**Full tree**, 47

**Gantmacher, F. R.**, 184

- Gaussian, channels, discrete-time source  
     with square difference distortion, 475–490; *see also* Channels, Gaussian  
 random process, definition, 365  
     represented as filtered white noise, 402  
     source with square difference distortion, 482  
 random variables, 509  
     bounds on distribution function, 380  
 Gelfand and Yaglom, 37, 370  
 Generalized random process, 365  
 Generator matrix, 199–201, 220  
     equivalent, 204  
 Generator polynomial of cyclic code, 222  
 Gilbert, E., 519  
 Gilbert bound on minimum distance, 537;  
     *see also* Varshamov-Gilbert bound  
 Goblick, T. J., 501, 502  
 Golay, M., 204  
 Golomb, Gordon, and Welch, 70  
 Grenander and Szego, 416  
 Groups, 209–212  
     Abelian, 209  
     cyclic, 212  
     order of, 210  
     order of element, 211  
 Growth of capital in gambling games, 507  
 Guard space, 288
- Hägelbarger, D. W., 306  
 Halmos, P. R., 36  
 Hamming bound, 545, 550  
 Hamming codes, 203–204, 230–238  
     in cyclic form, 551  
     symbols in arbitrary field, 548  
 Hamming distance, 161, 201  
 Hamming, R. W., 306  
 Hardy, Littlewood, and Polya, 322, 522  
 Hocquenghem, A., 238, 306  
 Holsinger, J., 440  
 Huffman, D., 70  
 Huffman codes, 52–55
- Identity element, 209  
 Indeterminant, 215  
 Infinite series of functions, convergence, 358  
 Information, 5  
     mutual, 16  
     average, 18, 23–27  
     for continuous-time-channel, 369
- convexity, 89–91, 524  
 conditional, 21  
     average, 21  
 continuous, 31  
 measure theoretic ensemble, 33–37  
 variance of, 509, 526  
 self, 5, 19  
     average, *see* Entropy  
 conditional, 20  
 nonexistence for continuous ensemble, 31  
 Input constraints for continuous channels, 317  
 Instantaneous source code, 46  
 Interlacing, 287  
 Intersymbol interference, 408  
 Invariant set of sequences, 59  
 Inverse element, 209  
 Irreducible set of states of Markov chain, 65  
 Irrelevance of independent noise, 413  
 Isomorphic Galois fields, 229
- Jacobs, I., 440  
 Jacobs, I. M., and Berlekamp, 280  
 Jelinek, F., 156, 540  
 Jointly Gaussian random variables, 365  
     characteristic function, 366  
     probability density, 366
- Kac, Murdock, and Szego theorem, 416  
 Kailath, T., 440  
 Karhunen-Loeve expansion, 399  
 Karush, J., 49  
 Kasami, T., 293  
 Kelly, J. L., 507  
 Kelly, Reed and Root, 419  
 Kendall and Reed, 70  
 Kennedy, R. S., 188, 431, 439  
 Khinchin, A. I., 59  
 Kohlenberg, A., 301  
 Kolmogorov, A. N., 12, 502  
 Kotelnikov, 12  
 Kotz, S., 306  
 Kraft inequality, 47, 49  
     countably infinite alphabets, 514
- L*<sub>2</sub> functions, 356  
 Lagrange's theorem on order of groups, 210  
 Lattice random variable, 190  
 Law of large numbers (weak), 41, 504  
 Limit in the mean, 358

- Linear codes, 220  
 Linear feedback shift register; maximal length, 232  
 synthesis algorithm, 246  
 Linear filters, 363  
 output specifying input, 410–414  
 time-varying, 391  
 List decoding, 166  
 upper bound on  $P_e$ , 538  
 Loeve, M., 402, 440  
 Log likelihood ratio, 375  
 Low pass ideal filters, 402–406
- Markov chains, finite homogeneous, 64  
 finite nonhomogeneous, 106  
 Markov source, 63–70  
 derived, 518  
 Massey, J. L., 245, 261, 263, 301, 305  
 Matched filters, 375  
 Max, J., 501  
 Maximal length codes, 230  
 Maximal length feedback shift register, 232  
 Maximum likelihood decoding, 120  
 BSC, 201  
 fading dispersive channels, 566  
 white Gaussian noise, 374  
 Maximum of convex functions, 85–89  
 McMillan, B., 49  
 McMillan's AEP theorem, 60  
 Metzner and Morgan, 261  
 Minimal polynomials, 227  
 calculation of, 550  
 Minimum cost decoding, 121, 527  
 Minimum distance, 167, 241  
 Minimum error probability decoding, 120  
 Mismatched decoder, 538  
 Models, 2  
 channels, 6  
 source, 4  
 Modulo-2 arithmetic, 197  
 Morse code, 39  
 Mutual information, *see* Information, mutual  
 Nats, 16  
 Normalized functions, 356  
 Normal random variable, *see* Gaussian, random variables  
 Null space of matrix, 200  
 Optimum decoders, *see* Codes, cyclic;
- Maximum likelihood decoding; Minimum cost decoding; and Minimum error probability decoding  
 Orthogonal codes, 379  
 Orthogonal functions, 356  
 Orthogonal set of linear combinations of discrete noise digits, 259  
 Orthonormal expansions, 355–370  
 asymptotic behavior of eigenvalues, 416  
 for the input and output of a linear filter, 391–394  
 Orthonormal sets, 356  
 complete, 359  
 Ott, G., 70
- Pairwise independence, 207, 503  
 Panic button channel, 102  
 Paradoxes about capacity of bandlimited Gaussian noise channel, 390  
 Parallel channels, 149–150, 343, 519  
 Parity check, 196  
 codes, 196–206  
 on arbitrary DMC, 208  
 matrix, 200–206  
 $(N,L)$ , 198  
 systematic, 198  
*see also* Linear codes  
 Parseval relation for Fourier transforms, 361  
 Parseval's equation, 359  
 Perfect codes, 204  
 Periodic set of states of Markov chain, 65  
 Peterson, W. W., 204, 234, 305, 306  
 Phase modulation, 479  
 Pierce, J., 440  
 Pilc, R., 457, 501  
 Pinkston, J., 502, 568  
 Pinsker, M. S., 34, 36, 440  
 Plotkin bound, 167, 545, 550; *see also* Difference energy  
 Polynomials, 214–219  
 degree of, 214  
 equality of, 215  
 factorization of, 218  
 irreducible, 217  
 monic, 218  
 remainder modulo g(D), 217  
 roots of, 218  
 sum and product of, 215  
 Power spectral density, 364  
 Prange, E., 306  
 Prefix-condition source codes, 45

- Primitive elements of Galois field, 226  
 Primitive polynomials, 230  
 Probability, 13–16  
     continuous, 27  
     density, 27  
         conditional, 28  
         joint, 28  
     discrete, 13  
         conditional, 14  
         joint, 14  
     measure, 34  
 Probability of decoding error, 121  
     bounds on minimum, expurgated ensemble bound, 150–157  
     in terms of  $(C-R)^2$ , 539  
     random code ensemble, 135  
     sphere packing, 157  
     straight line, 161  
     *see also* Coding theorem, Exponent of error probability; and Expurgated exponent  
 codes with two code words, 122  
     randomly chosen words, 131  
 Gaussian white noise channel; orthogonal code, 363, 379  
     orthogonal code, unknown phase, 564  
     two code words, 374  
 rates above capacity, error probability per block, 173  
     error probability per source digit, 77, 466  
 source codes, 42  
 Prolate spheroidal wave functions, 404  
     asymptotic behavior of eigenvalues, 405  
     Fourier transform property, 406  
 Propagation of decoding errors, 261  
 Pseudo-noise sequences, 232  
 Pulse code modulation, 479  
  
 Quantization, 443, 501  
  
 Radon-Nikodym derivative, 37  
 Random coding exponent,  $E_r(R)$ , 139–150  
     binary symmetric channel, 146  
     discrete memoryless channels, 139–150  
     discrete parallel channels, 149–150  
     discrete-time memoryless channels, 318, 331  
     Gaussian additive noise, 340  
     parallel, 349  
  
     finite state channels, 180  
     Gaussian additive noise channels with filtered input, 426  
     very noisy channels, 147–150  
 Random process, specification, 362  
     stationary, 363  
     wide sense stationary, 363  
     zero mean, 363  
 Random variable, 18  
 Random walks, 312–315  
 Rate, block code, 118  
     convolutional code, 266  
 Rate-distortion function, 444  
     calculation of, 457  
     convexity, 445  
     discrete ergodic source, 490–502  
     discrete-time memoryless Gaussian source, 477  
     discrete-time memoryless source, 470  
     Gaussian random process source, 485  
     lower bound, 459  
 Recurrent codes, *see* Convolutional codes  
 Reduced echelon form, 205  
 Reduced ensemble for source codes, 53  
 Reed-Solomon codes, 257, 297  
     minimum distance, 551  
 Reiffen, B., 111  
 Relatively prime integers, 212  
 Reliability function, 160  
 Richters, J., 441  
 Riesz and Nagy, 357, 396, 421, 440  
 Riesz-Fischer theorem, 357  
 Root and Varaiya, 440  
 Row space of matrix, 200  
 Run length source coding, 515  
  
 Sample space, 13  
 Sampling expansion, 362  
 Sampling functions, 361  
 Sardinas and Patterson, 45, 513  
 Savage, J., 279  
 Schalkwijk and Kailath, 481  
 Scholtz, R. A., 70  
 Schwarz inequality, 358  
 Scrambling, 287  
 Self-synchronizing codes, 70  
 Semi-invariant moment generating functions, 188  
 Sequential decoding, 263–286  
     bias term, 267  
     computation and error probability with

- bounded depth of search, 553
- correct path, 272
- descendants of a node, 271
  - immediate descendants, 271
- Fano algorithm, 269
- F*-hypothesis, 271
- forward, lateral, and backward moves, 268
- number of computations per decoded
  - subblock,  $W_n$ , 273
  - average computation,  $W_n$ , 278
  - moments of computation,  $W_n^p$ , 279
- path thresholds, 271
- path values, 271
- probability of decoding error, 280–286
- proof that  $W_n < \infty$  for  $R < R_{comp}$ 
  - where  $R_{comp} = E_O(1, \mathbf{Q})$ , 278
- statistical independence of correct and
  - incorrect paths, 554
- threshold,  $T$ , 269
- value of hypothesis,  $\Gamma(\mathbf{x}_I; \mathbf{y}_P)$ , 267
- Shannon, C. E., 1, 12, 64, 111, 155, 188,
 338, 354, 373, 440, 501, 533
- Shannon, Gallager, and Berlekamp, 157,
 161, 169, 188, 204, 329
- Shannon's theorem on the capacity of
  - bandlimited Gaussian noise channels, 389
- Simplex codes, 378, 562
  - probability of decoding error, 383
- Slepian, D., 306, 354
- Slepian, Pollak, and Landau, 404
- Source, 4
  - discrete memoryless, 38, 40
  - discrete periodic, 56
  - discrete stationary, 56
  - discrete-time, memoryless, continuous
    - amplitude, 470–475, 475–490
  - ergodic discrete, 490–502
  - Gaussian random process, 362
  - Markov, 63–70
    - derived Markov, 518
  - models, 4
  - nondiscrete, 6
  - physical, 4
  - subject to a fidelity criterion, 442–502
- Source and channel coding theorem, 534
- Source codes, 4, 38–70
  - with fidelity criterion, 448
  - fixed length, 39–43
  - prefix condition, 45
  - unique decodability, 45, 512
- variable length, 39, 43–49
  - optimum, 52–55
- Source coding theorem, 5
- discrete memoryless sources, fixed length
  - codes, 43
- variable length codes, 50–51, 513
  - countably infinite alphabet, 514
- discrete stationary sources, variable length
  - 4, 56
- ergodic discrete sources, fixed length codes,
 65
- with fidelity criterion, 451
  - discrete memoryless sources, 455
    - converse, 449
    - infinite distortions, 455
    - speed of convergence, 456
  - discrete-time memoryless sources, 470–475
    - converse, 471
    - effect of noisy channels, 473
- ergodic discrete source, 500
  - converse, 492
- Gaussian random process source, 485
- Spectral density, 364; *see also* Power spectral density
- Speech waveforms, 443
- Sphere-packed codes, 204
- Sphere-packing exponent,  $E_{Sp}/R$ , 158
- Statistical independence, 15
  - ensembles, 15
  - pairwise, 207, 503
- Stiglitz, I., 501, 539
- Straight line exponent of error probability,
  $E_{sl}/R$ , 161
- Subfields, 227
- Subgroups, 210
- Sufficient receiver, 509
- Suffix condition, 515
- Sum channel, capacity of, 525
  - random coding exponent, 535
- Super-sources, 495
- Symmetric discrete memoryless channel, 91
- Syndrome, 200
  - for BCH codes, 242
  - for convolutional codes, 259
- Systematic linear codes, 220
- Systematic parity-check codes, 198
- Telephone line, 1
- Threshold decoding, 260–263
  - diffuse, 301

- of maximal length codes, 552  
Tilted random variables, 189  
Time and frequency limited functions, 360  
Time-diversity burst decoder, 303  
Titchmarsh, E. C., 361  
Transient state of Markov chain, 65  
Tree for prefix-condition code, 46  
Typical sequences, 41, 536
- Uniquely decodable source codes, 45
- Variance of mutual information, 509  
relation to capacity, 526  
Variance of sum of random variables, 504  
Varshamov-Gilbert bound, 546, 550  
Very noisy channels, 147  
expurgated exponent, 541  
Viterbi, A., 379, 441
- Wald's equality, 315, 554  
Wagner, T., 354  
Water filling interpretation of capacity, 389
- Weight of binary sequence, 201  
White Gaussian noise, 365  
statistical independences of expansion coefficients, 367  
White Gaussian random process, *see* White Gaussian noise  
Wiener, N., 12  
Wolfowitz, J., 59, 173, 188, 493  
Wozencraft, J. M., 266, 305  
Wozencraft and Horstein, 261  
Wozencraft and Jacobs, 12, 305, 440  
Wozencraft and Kennedy, 563  
Wyner, A., 440  
Wyner and Ash, 301
- Yaglom, A. M., 363  
Yudkin, H., 111, 188, 440, 542
- Zero error capacity, 155, 533  
Zetterberg, L. H., 440  
Zierler, N., 306

GALLAGER

INFORMATION THEORY and  
RELIABLE COMMUNICATION

Q  
360  
.G3  
C.1

WILEY

ISBN 0-471-29048-3