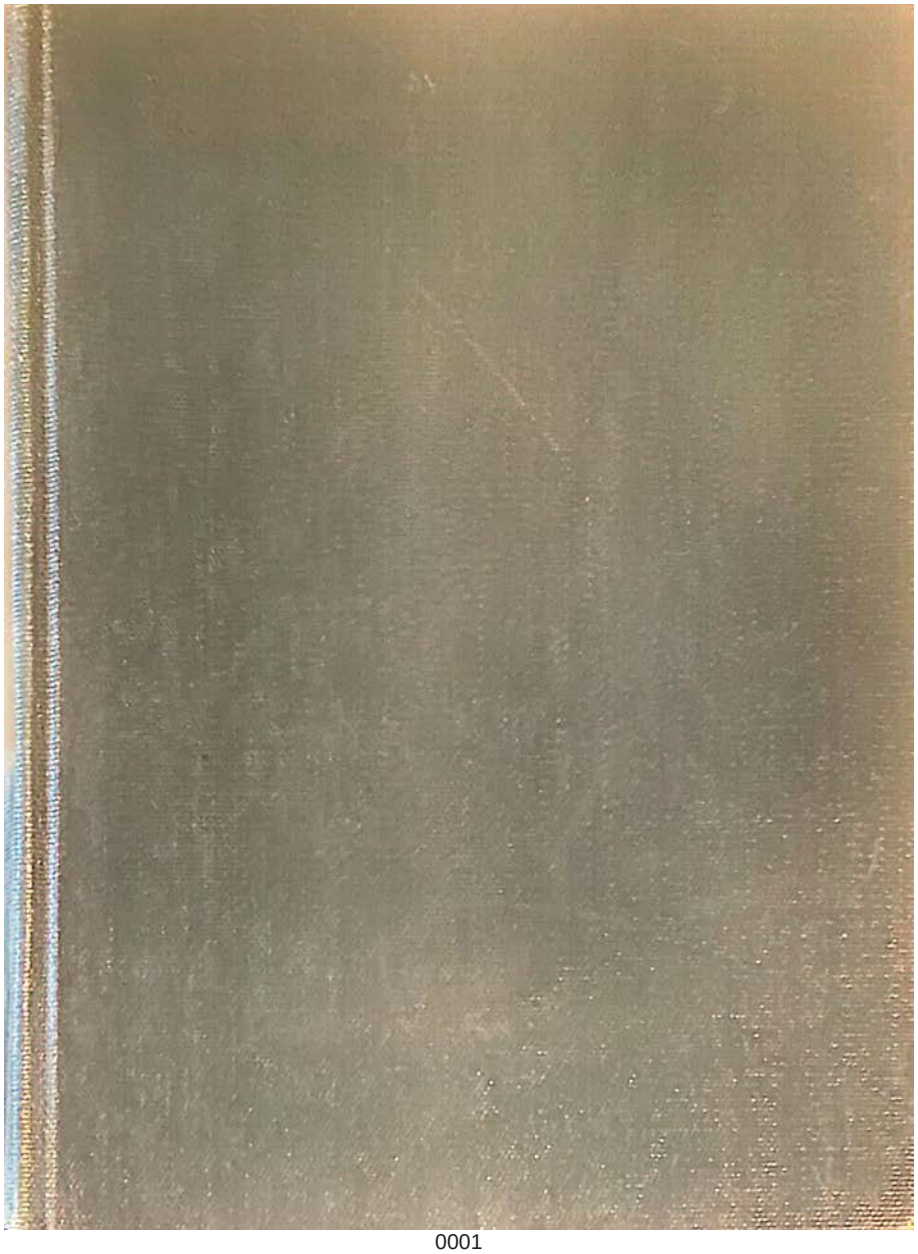


0001

DIGITAL CODING OF WAVEFORMS

PRENTICE-HALL SIGNAL PROCESSING SERIES

Alan V. Oppenheim, Editor

ANDREWS and HUNT *Digital Image Restoration*
BRIGHAM *The Fast Fourier Transform*
BURDIC *Underwater Acoustic System Analysis*
CASTLEMAN *Digital Image Processing*
CROCHIERE and RABINER *Multirate Digital Signal Processing*
DUDGEON and MERSEREAU *Multidimensional Digital Signal Processing*
HAMMING *Digital Filters, 2e*
HAYKIN, ED. *Array Signal Processing*
JAYANT and NOLL *Digital Coding of Waveforms: Principles and Applications to Speech and Video*
LEA, ED. *Trends in Speech Recognition*
LIM, ED. *Speech Enhancement*
MCCLELLAN and RADER *Number Theory in Digital Signal Processing*
OPPENHEIM, ED. *Applications of Digital Signal Processing*
OPPENHEIM, WILLSKY, with YOUNG *Signals and Systems*
OPPENHEIM and SCHAFER *Digital Signal Processing*
RABINER and GOLD *Theory and Applications of Digital Signal Processing*
RABINER and SCHAFER *Digital Processing of Speech Signals*
ROBINSON and TREITEL *Geophysical Signal Analysis*
TRIBOLET *Seismic Applications of Homomorphic Signal Processing*

DIGITAL CODING OF WAVEFORMS
Principles and Applications to Speech and Video

N. S. JAYANT
Bell Laboratories, Inc.
Murray Hill, NJ

PETER NOLL
Technical University of Berlin

PRENTICE-HALL, INC. Englewood Cliffs, New Jersey 07632

Library of Congress Cataloging in Publication Data

Jayant, Nuggehally S., 1946-
Digital coding of waveforms.

Includes index.
1. Signal processing—Digital techniques. 2. Coding theory. I. Noll, P. (Peter), 1936- II. Title.
TK5102.J39 1984 621.38'043 83-22170
ISBN 0-13-211913-7

Editorial/production supervision: *Shari Ingerman*
Cover design: *Edsal Enterprises*
Manufacturing buyer: *Tony Caruso*
Page layout: *Diane Koromhas*

© 1984 by Bell Telephone Laboratories, Incorporated

All rights reserved. No part of this book may be reproduced, in any form or by any means, without permission in writing from the publisher.

Printed in the United States of America

10 9 8 7 6 5 4 3 2 1

ISBN 0-13-211913-7 01

Prentice-Hall International, Inc., *London*
Prentice-Hall of Australia Pty. Limited, *Sydney*
Editora Prentice-Hall do Brasil, Ltda., *Rio de Janeiro*
Prentice-Hall Canada Inc., *Toronto*
Prentice-Hall of India Private Limited, *New Delhi*
Prentice-Hall of Japan, Inc., *Tokyo*
Prentice-Hall of Southeast Asia Pte. Ltd., *Singapore*
Whitehall Books Limited, *Wellington, New Zealand*

EMS
LIB
TK
5102.
J39
1984

To Amma, Anna
and Angelika

and to Julia

Contents

Preface xi

1. **Introduction 1**
   - 1.1 Digital Coding of Waveforms 1
   - 1.2 Subjective Quality, Bit Rate and Coder Complexity 9
   - 1.3 Information Theoretical Limits 20
2. **Waveform Characterization 26**
   - 2.1 Introduction 26
   - 2.2 Characteristics of Speech and Image Waveforms 27
   - 2.3 Mathematics of Random Waveforms 41
   - 2.4 Model Sources with Special Correlations and Spectra 59

* 3. **Sampling and Reconstruction of Bandlimited Waveforms 86**
   - 3.1 Introduction 86
   - 3.2 Deterministic Lowpass Signals 88
   - 3.3 Bandpass Signals 96
   - 3.4 Random Processes 98
   - 3.5 2-D and 3-D Signals 99
   - 3.6 Errors in Sampling — Aliasing Effects 102
   - 3.7 Discrete-Time Error and Continuous-Time Error 110

4. **Quantization 115**
   - 4.1 Introduction 115

Items with asterisks (*) are special topics that can be skipped without loss of continuity.

4.2 Calculation of Quantization Error Variance 119
4.3 Uniform Quantization 124
4.4 Non-uniform Quantization 129
4.5 Logarithmic Quantization 141
*4.6 Entropy-Coded Quantization 146
*4.7 Statistical Properties of Quantization Errors 158
*4.8 The Use of Dithering in Coarse Quantization 164
*4.9 Transmission Error Effects 175
4.10 Adaptive Quantization 188

5. **Pulse Code Modulation 221**
5.1 Introduction 221
5.2 PCM Coding of Speech, Video and Audio Waveforms 224
5.3 PCM Systems for High-Quality Speech Coding 228
*5.4 Transmission Error Effects 243

6. **Differential PCM 252**
6.1 Introduction 252
6.2 Linear Predictors of Order $N = 1$, 2 and 3 261
6.3 Linear Predictors of Order N 267
6.4 Low-Complexity DPCM Systems 280
6.5 Adaptive Prediction 290
6.6 Distant-Sample-Based Predictions for Periodic Signals 312
*6.7 Transmission Error Effects 324

* 7. **Noise Feedback Coding 351**
7.1 Introduction 351
7.2 D*PCM 356
7.3 Noise Feedback Coding 362

8. **Deltamodulation 372**
8.1 Introduction 372
8.2 Quantization Noise in Single Integration DM 379
*8.3 Double Integration, Second-Order Prediction and ΣDM 393
8.4 Adaptive Deltamodulation (ADM) 400
*8.5 Transmission Error Effects 417

* 9. **Delayed Decision Coding 428**
9.1 Introduction 428
9.2 Codebook, Tree and Trellis Coders 432
9.3 Search Algorithms 439
9.4 Delayed Decision Coding of Memoryless Sources; $R = 1$ 445
9.5 Delayed Decision Coding of Correlated Sources 451

* 10. **Run-Length Coding 465**
10.1 Introduction 465
10.2 Principles of Run-Length Coding 469
10.3 Run-Length Coding of Two-Level Graphics 474

0008

10.4 Extensions to Multi-Level Inputs 479

11. **Sub-Band Coding 486**

11.1 Introduction 486
11.2 Transmission Rate *I*, *SNR* and Gain Over PCM 490
11.3 The Integer-Band Filter Bank 494
*11.4 Quadrature-Mirror Filter-Banks 496
11.5 Sub-band Coding of Speech 500
*11.6 Transmission Error Effects 505

12. **Transform Coding 510**

12.1 Introduction 510
12.2 Linear Transforms of Order $N = 2$ 513
12.3 $N \times N$ Transforms 517
12.4 Optimum Bit Allocation and Zonal Sampling 524
*12.5 Optimum Transform (KLT) 535
12.6 Suboptimum Transforms (DHT, DWHT, DFT, DCT) 546
12.7 Adaptive Transform Coding of Speech and Images 563
*12.8 Transmission Error Effects 576

Computer Projects 591

* **Appendix A. Deterministic Signals and Linear Systems 598**

A.1 Fourier Representation 598
A.2 Two-Dimensional (2-D) Signals 600
A.3 Linear Systems 602
A.4 Continuous-Time Signals and Systems 606

* **Appendix B. Matrix Operations 609**

* **Appendix C. Information Theory of Sources and Channels 616**

C.1 Discrete-Amplitude Memoryless Sources 617
C.2 Discrete-Amplitude Sources with Memory 619
C.3 Noiseless Coding of Discrete-Amplitude Sources 621
C.4 Continuous-Amplitude Memoryless Sources 624
C.5 Continuous-Amplitude Sources with Memory 625
C.6 Channel Capacity 626

* **Appendix D. Distortion Rate Functions 632**

D.1 Introduction 632
D.2 Summary of Source Entropy Results 633
D.3 Discrete-Amplitude Sources 634
D.4 Continuous-Amplitude Sources 636

* **Appendix E. Objective Measurement of Coder Performance 650**

E.1 Signal-to-Noise Ratio, *SNR* 650
E.2 Segmental Signal-to-Noise Ratio, *SNRSEG* 651
E.3 Frequency-Weighted *SNR* and Reconstruction Error 652
E.4 *SNRMAX*, *SNRMIN* and Non-mmse Measures 655

E.5 Signal-to-Uncorrelated Noise Ratio, SNR_u 656

* **Appendix F. Subjective Measurement of Coder Performance 658**

F.1 Quality Measurements 659

F.2 Intelligibility Tests 662

* **Appendix G. A/D and D/A Conversion 666**

G.1 Analog-to-Digital Conversion 667

G.2 Digital-to-Analog Conversion 672

Index 676

Preface

Digital Coding is by no means a new subject. It has been practiced for the better part of fifty years, with the last twenty years witnessing an unprecedented amount of activity. The number of archival publications in this period is in the thousands. The momentum of this activity is expected to continue although there may be important shifts in emphasis. One such shift may be from concepts-type literature to implementation-type endeavors. As in many other subjects, fundamental innovations will probably, and hopefully, continue to be made in digital coding, leading to advances in bit rate reduction more impressive than those reported in this book. It may be fair to say, however, that our understanding of speech and image coding has now reached a very mature point, making the time ripe for a book of the type we are offering.

Emerging communications- and information networks require ubiquitous uses of digital speech, digital audio and digital video. Applications include transmission, storage, conferencing and many kinds of signal processing for information manipulation and display. Basic to all these systems are techniques for waveform digitization with acceptable levels of signal quality. This indeed is the topic of our book.

There have been outstanding books on specific aspects of waveform coding. These include pioneering works on Pulse Code Modulation and Deltamodulation. There have also been well-respected books on Speech and Image Processing with excellent special sections on Digital Coding. However, to our best knowledge, the present book is the first to offer a broad and in-depth coverage of waveform coding concepts and algorithms. This is also the first attempt to include both speech and image systems under one cover, not counting a reprint book on the subject edited by one of us about eight years ago.

The non-existence of a treatise on waveform coding was certainly a major motivation for our book, and its production has made possible the unification of a wealth of information that was hitherto scattered in diverse branches of literature such as Information Theory, Communications Technology and Signal Processing. The idea of a book that would treat

both speech and image signals was not obvious since the authors of this book were much closer to speech coding than to image coding. But it became apparent at a fairly early stage of the project that the fundamental principles of most coding algorithms, as well as the notion of using statistical input models for coder analyses, were equally valid for speech and image waveforms. In fact, these principles could best be illustrated by invoking examples from both fields, and this is indeed the approach used in this book.

In the interest of a unified exposition, issues that are very specific to one type of signal are usually avoided in this book. Exceptions are extremely important and standardized systems such as logarithmic PCM coders for speech and run-length coders for two-level graphics images. In general, optimization of coding algorithms to a given signal involve attention to specific signal properties as well as specific issues in perception, and these result in finely tuned coding structures that may not be covered in our basic treatment. Practicing professionals seeking such information will therefore be expected to supplement the chapters of this book with speech-specific or image-specific reading, as the case may be. There are fairly long, although not exhaustive, lists of references at the end of each chapter to help the readers in this regard.

An example of a topic that is entirely out of the scope of our book is analysis-synthesis vocoding for speech. But there are many fundamental signal processing algorithms covered in this book that have direct application in the above specialized area as well. Examples of such basic algorithms are those for quantization and prediction.

As a text for a university course on waveform coding, this book is perhaps unique. It makes possible, for the first time, a complete one-semester or one-quarter graduate course on coding, instead of teaching coding as a small aspect of a signal-processing or communications systems course. The problems and the computer projects in the text have been chosen to provide a useful mix of drill-type exercises and perspective-providing tools, and a solutions manual for the problems is being offered to faculty desiring to teach this course. Included in the table of contents are asterisks reflecting optional or specialized topics that could be eliminated for the purposes of teaching the course. Needless to mention, some lecturers intending to use this book for a course will tend to exercise their own specific choices with regard to scope and emphasis. There is also the possibility of teaching the book in two companion semesters or quarters.

Parts of this book have been used by its authors as the bases for regular courses in universities, intensive courses for practicing professionals and courses in continuing education programs in professional environments such as Bell Laboratories. In its present form, the book was first used to teach a 40-hour one-semester graduate course at the University of California at Santa Barbara in the beginning of 1983. The background that was expected from the students in this course was basic knowledge of linear systems, probability and statistics, and familiarity with programming techniques for carrying out computer projects.

Synopsis

The following paragraphs summarize the purpose and content of each of twelve chapters in this book. Most of these chapters include problems for solution. There is a separate section on computer projects. There are seven appendices.

Chapter 1 defines the waveform coding problem and summarizes the accomplishments in the field of bit rate reduction. It introduces objective as well as subjective criteria for the measurement of digitized signal quality. It also addresses the issues of coder complexity and coding delay by identifying three broad classes of coding algorithms, examples of which are discussed at length in later chapters of the book.

Chapter 2 describes properties of speech and image waveforms that are relevant to the coding problem. It discusses quantitative statistical models that are used in later chapters to motivate quantization and coding algorithms, and to calibrate their performances with respect to the promises of Rate Distortion theory. The statistical descriptors used in modelling waveforms include the probability density function, the autocorrelation function, power spectral density and a special property thereof called the spectral flatness measure. Also discussed in this chapter are properties that are specific to speech and image waveforms, such as nonstationarity and quasi-periodicity.

Chapter 3 discusses the process of waveform sampling. This process is a basic prerequisite for digital coding systems, and most of the chapter is an attempt to recapitulate well-known fundamental results and theorems on uniform time sampling. The chapter also includes discussions that are specific to speech and image sampling. For example, it discusses the rationale behind specific anti-aliasing or prefiltering operations that are tacitly assumed in most of speech and image work.

Chapter 4 deals with amplitude quantization. It is the longest chapter in this book, and it could very well have been a monograph by itself. After a fairly complete description of uniform quantization and a derivation of a fundamental 6 dB/bit signal-to-noise ratio result, the chapter goes on to the extremely important refinements of nonuniform quantization, logarithmic companding and adaptive quantization of nonstationary inputs. The chapter includes a section on the statistical properties of quantization errors which helps to put in perspective commonly assumed error properties such as uniformity of its probability density function and whiteness of its power spectral density. The chapter also shows the benefits of variable-length entropy coding in bringing quantizer performance closer to the Rate Distortion bound. Vector Quantization, which has a similar potential, is discussed in a later chapter.

Chapter 5 is on pulse code modulation (PCM). This is the earliest and best understood technique for digital coding and 8-bit PCM coding provides a standard against which other algorithms are compared, in both speech and image work. In deference to the ubiquitous use of 8-bit logarithmic PCM for high-quality speech communication, a good deal of this chapter is devoted to discussions specific to this standard.

Chapters 6, 7 and 8 are very closely related. They discuss, respectively, differential PCM, noise feedback coding and deltamodulation.

Chapter 6 begins the discussion of differential PCM (DPCM) with the theory of linear prediction and the notion of prediction gain as a measure of redundancy removal in a sequence of correlated samples. Several examples illustrate the dependence of prediction gain on the number of waveform samples used in the prediction of a sample to be coded. Later sections of this chapter discuss adaptive algorithms needed for the efficient prediction of a nonstationary process with time-varying spectrum; and algorithms based on distant samples, for the predictions of a quasi-periodic signal such as voiced speech or a moving image. Adaptive Prediction is the basis of high quality speech and image coding systems using 4 bits per sample. Combinations of adaptive and distant-sample-based prediction produce some of the best results in bit rate reduction, and make possible high-quality digitizations of speech and video at coding rates as low as 1 to 2 bits per sample.

Chapter 7 deals with the notions of open-loop DPCM and noise feedback coding, and provides a formal discussion of noise shaping. Linear prediction, as discussed in Chapter 6, can provide a very useful utilization of redundancies inherent in waveform production. Suitable designs of prediction error quantizers in DPCM can also exploit some of the properties of the human perceptual mechanisms, and provide coding configurations that enhance perceived quality at a given bit rate. Noise feedback coding provides an additional degree of freedom in coder design and makes possible a more direct control of reconstruction

noise spectrum. This kind of noise shaping can provide very powerful perceptual optimizations of DPCM-type coders.

Chapter 8 is on deltamodulation, the special case of DPCM with a two-level quantizer. Deltamodulation is the simplest form of digital coding, and like PCM, it has found wide applications in digital transmission as well as in signal processing. The coarseness of the one-bit quantizer usually necessitates the oversampling of the input waveform. The chapter therefore includes a discussion of prediction gain as a function of sampling frequency. For important classes of input spectra, the result is a 9 dB/octave result for the signal-to-noise ratio of deltamodulation with a single integrator or first-order predictor. After pointing out the advantages of double integration and delta sigma modulation, the chapter enters into a comprehensive discussion of adaptive step-size algorithms for nonstationary inputs. In particular, syllabically adaptive deltamodulation is well known for its transmission error resistance and this type of coder, usually known as continuously variable slope deltamodulation (CVSD), is discussed at length.

Chapter 9 discusses delayed decision coding, a class of algorithms that anticipate, or take into account by some means, the characteristics of the actual sequence being digitized. The mechanism for doing this involves the use of significant encoding delay by definition, and a search for a best output sequence, or a part thereof, as stored or generated in structures such as trees, trellises or codebooks. Coding algorithms based on codebooks are also known as vector quantizers. An interesting result of this chapter is that delayed decision coding is useful even when the input is a sequence of independent, identically distributed random variables. In fact, the technique provides an important approach to rate-distortion bounds with inputs such as gamma-distributed variables for which the performance of conventional scalar quantizers is very poor. Delayed decision coding is also a powerful technique for the encoding of correlated sequences. In this case, there are two alternative approaches. The first is based on the direct application of delayed decision coding to the correlated input sequence. The second method is based on redundancy removal as in differential coding, followed by delayed decision coding of the prediction error waveform. Examples from speech coding are used to illustrate the efficiency of these techniques at coding rates in the order of 1 bit per sample. Delayed Decision Coding also provides a natural framework for realizing fractional coding rates such as 0.5 bit per sample.

Chapter 10 deals with a very specific waveform, that of two-level graphics, or facsimile. Many important classes of two-level images are characterized by long clusters or runs of all-black or all-white pixels along a horizontal scan line. Run-length coding is a conceptually simple, and highly efficient technique for the error-free coding of such inputs. The techniques usually call for some kind of variable-length coding to account for the fact that the probability of a given run is a very nonuniform function of the run-length. Run-length coding techniques can also utilize vertical or inter-line correlations in the graphics input, for greater encoding efficiency in appropriate two-dimensional algorithms. Several run-length coding algorithms have been standardized internationally, and these standards are discussed at some length in this chapter. The more efficient examples of the run-length algorithms encode standard graphics documents using less than 0.1 bit per pixel on the average.

Chapters 11 and 12 return to the problem of encoding multi-level inputs with non-zero error. Unlike the predictive coding methods of earlier chapters, the last two chapters of the book use a frequency-domain or transform-domain approach.

In the sub-band coding approach of Chapter 11, an input such as speech is first split into contiguous sub-band signals which are encoded separately. The sub-band partition makes possible the use of variable bit allocation, so that input components that are deemed more

important can be quantized with greater numbers of levels. The bit allocation may reflect differences in speech power in the various sub-bands, and/or differences in the perception of noise occurring in those sub-bands. Emphasis in Chapter 11 is on relatively simple designs for speech coding, characterized by a fairly small number of sub-bands, and a time-invariant design of the bit allocation algorithm.

Chapter 12 is on transform coding. This chapter generalizes the sub-band approach to include the possibility of splitting the input into a much larger number of frequency-domain, or more generally, transform-domain components. These components are in fact the transform coefficients that result from a linear operation on an input sequence block whose length equals the number of transform coefficients desired. Transform Coding performance increases with block length. The maximum performance of transform coding is shown to be identical to the maximum performance of a differential or predictive coding system. Transform coding and predictive coding are however very different in the perceived properties of their reconstruction error sequences, and in the perceived effects of transmission errors. Practical versions of transform coding use suboptimal but easily implementable transforms. Several examples of these transforms and their properties are discussed at length. Later sections of this chapter concentrate on applications to speech and image coding. In the case of speech coding, emphasis is on algorithms for adaptive bit allocation to track the time-varying nature of the input spectrum. In the case of images, an additional point of focus is the need for efficient approaches in two-dimensional and three-dimensional coding. In both cases, transform coding is shown to lead to very high quality digitizations at a bit rate in the order of 2 bits/sample, a result that is comparable with the best results with adaptive predictive coding.

Acknowledgements

The following colleagues were kind enough to review aspects of our book: B. P. Agrawal, W. R. Daumer, H. Fehn, A. Gersho, J. Gibson, J. F. Kaiser, R. Lippmann, J. B. O'Neal, D. Preuss, A. Tomozawa, T. Thomas, R. Zelinski and Ms. C. Volmary. We would like to thank them for their time and effort.

Many of the image coding illustrations in this book were made using original material supplied by respective authors. For this kind co-operation, we thank N. Ahmed, R. Baker, R. M. Gray, R. Hunter, K. Iinuma, T. Kummerow, M. Kunt, H. Landau, R. Lei, R. Lippmann, H. Musmann, W. Pratt, D. Preuss and R. Steele.

The line drawings for this book were prepared by Comprehensive Designers Incorporated, Union, New Jersey. For co-ordinating this effort, we thank Mr. Roy Anderson of the art studio at Bell Laboratories, Murray Hill, and his ever-helpful colleagues.

Several members of the text processing department at Bell Laboratories have contributed to the realization of this document. For their efforts and expertise, we thank Mrs. Beverly Heravi, Ms. Babe Hill, Mrs. Donna Manganelli, Ms. Joan McCarthy, Ms. Carmela Patuto and Mrs. Susan Tarczynski. In particular, we would like to acknowledge the sustained contributions of Ms. Hill and Mrs. Tarczynski, and the invaluable role played by the latter in the final phases of a time-worn project.

The editorial and production departments at Prentice Hall have done an excellent job of bringing out this book in its present form. We thank them for their contributions. In particular, it is a pleasure to acknowledge the help of the copy editor, the production editor Ms. Shari Ingerman and the college text editor, Mr. Bernard Goodwin. And our thanks to Prof. Alan Oppenheim for having this book in his distinguished series on Signal Processing.

Students at the University of Bremen and the Technical University of Berlin have used early versions of chapters of this book to carry out projects and thesis work, and thereby have contributed to this book in many ways. We would like to thank them for their contributions.

Prof. A. Gersho invited one of us to try out the book draft as the basis of a new graduate course offered by the University of California at Santa Barbara. The teaching of this course was extremely beneficial to the book. Our thanks to Prof. Gersho.

The endorsement of this book by Bell Laboratories has been a major factor in the realization of this book. We are grateful to Bell for our extensive use of its excellent facilities, especially the photography department, the art studio and the text processing facilities at Murray Hill.

Our involvement with this book has been long and sometimes difficult. Two individuals at Bell Laboratories have made the process a little easier. For their encouragement and inspiration, we thank Drs. J. L. Flanagan and L. R. Rabiner. Dr. Flanagan was instrumental in bringing together the authors of this book about ten years ago. In doing so, he also set the stage for the production of this work.

N. S. Jayant Peter Noll

1

Introduction

1.1 Digital Coding of Waveforms

This book will discuss many techniques for representing analog waveforms by sequences of binary digits, or bits. *Digital coding* is the process, or sequence of processes, that leads to such digital representations. The benefits of digital representation are many and well known [Aaron, 1979]. Perhaps most significant is the fact that digital signals are less sensitive than analog signals to transmission noise; as a result, digital signals have offered the possibility of making better use of interference- and noise-limited communication media, although at the expense of greater bandwidth than that demanded in analog communications. To give a concise list of other advantages, digitized signals are easy to *regenerate* and *store*, to *error-protect* and *encrypt*, and to *multiplex, packetize* and *mix*. Digitization of waveforms also enables an efficient *unification of transmission and switching* functions in communications; it also permits an extensive application of *digital signal processing*. Time-shared digital signal processing, unlike analog processing, makes better use of new classes of technology, such as optical and millimeter wave devices, which excel in speed rather than in properties such as linearity. Digital processors and terminals also make better use of the rapid revolutions in microprocessor and integrated circuit technologies.

Digital representations of analog waveforms entail, however, the introduction of some kind of coding distortion; and a basic problem in waveform coding is to achieve the *minimum possible distortion for a given encoding rate* (bits per second, or bits per waveform sample) or, equivalently, to achieve a *given acceptable level of distortion with the least possible encoding rate*. The encoding rate is referred to as

0017

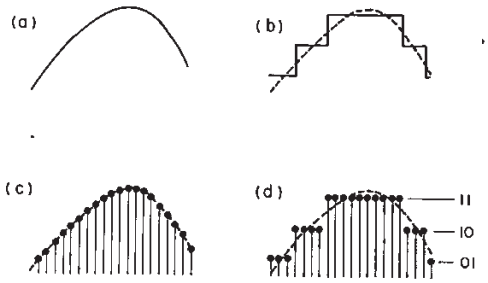
the *bit rate* in the rest of this book, and it is specified either as the number of bits per second, or as the number of bits per waveform sample. The distortion is specified either by an objective measure such as the *signal-to-noise ratio*, or by a subjective measure such as the *mean opinion score*. An additional parameter that enters the picture is coder complexity or cost, but the role of this parameter tends to diminish as advances in technology and signal processing bridge cost gulfs between the sophisticated and the simple.

Applications of digital representations are many and increasing. They include transmission and/or storage of speech, audio, video and graphics waveforms, satellite imagery, and geophysical and medical data, to mention important examples. Digital representations are also very helpful in the processing, copying, editing and displaying of the above-mentioned signals. This book focuses primarily on *telephone-bandwidth speech* and *monochrome video*. The digital coding of these signals will be referred to as *speech coding* and *image coding*, respectively. Our treatments of these topics can be generalized, sometimes in straightforward fashion, to apply to higher-bandwidth audio and color video. Chapter 10 provides a special discussion of two-level graphics.

The Basic Processes of Digital Coding. Figure 1.1 is a qualitative illustration of how analog waveforms, which are *time-continuous* and *amplitude-continuous*, enter the digital world by means of *time discretization (sampling)* and *amplitude discretization (quantizing)*. The waveform in (a) is *analog*, and that in (d) is its *digitized* version. Each of the time waveform samples in (d) has a finite number of allowed amplitude values, represented by appropriate groups of binary digits, such as the sequences 01, 10 and 11 in the illustration. The transformation from (a) to (d) can be via stage (b) or — as is more usual — via stage (c). Time discretization will be assumed for waveform sources in all of this book, unless otherwise mentioned. When performed with adequate resolution, time discretization will be seen to be an *information-preserving* operation. In contrast, amplitude quantization will be *information-lossy* by definition.

A *waveform encoder* typically accepts a time-discrete input such as (c) and provides a binary output sequence which is sufficient to derive the digitized

Figure 1.1 Four types of signals: (a) amplitude-continuous and time-continuous; (b) amplitude-discrete and time-continuous; (c) amplitude-continuous and time-discrete; and (d) amplitude-discrete and time-discrete.



0018

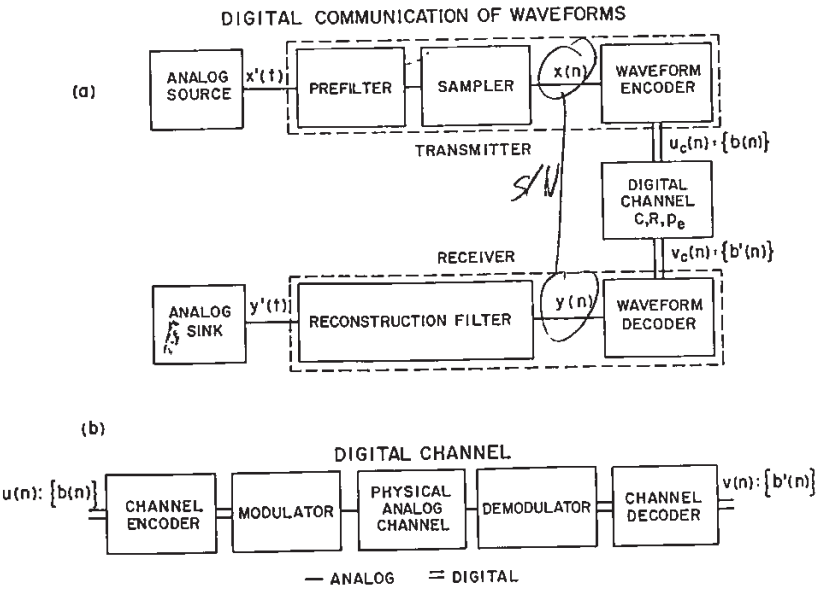
reconstruction (d). It is the function of a *waveform decoder* to accept that binary output sequence, or a version of it, at the output of a storage or transmission medium, and thence to deliver the reconstruction (d) — strictly, a lowpass filtered, or smooth version of it. The term *waveform coding* is sometimes used to suggest the encoding process alone. But throughout this book, as in most coding literature [Jayant, 1976] [Flanagan et al., 1979] [Netravali and Limb, 1980], the term will imply the combination of encoding and decoding processes. This is the notation in Figure 1.3, where the single box labeled *waveform coding system* signifies encoding as well as subsequent decoding.

1.1.1 The Digital Communication System

The overall digital communication scenario is shown in Figure 1.2, and most of this book focuses on the waveform encoder and waveform decoder boxes. The composite of these two boxes is sometimes called a *waveform coder* or *waveform codec* (*co*der + *dec*oder = *codec*). The latter terminology is especially used in the context where both coder and decoder are implemented within a single unit at transmitter and receiver terminals. A simple example of the waveform encoder box is a PCM encoder (Chapter 5), which involves an amplitude quantizer followed by a binary codeword generator.

Prefiltering and *sampling*, and the inverse operation of reconstruction filtering in the receiver, are procedures that are tacitly assumed in digital waveform communication. The concept of information-lossless sampling of a *bandlimited*

Figure 1.2 (a) Digital communication of waveforms; (b) digital channel.



analog source is very significant and is discussed at length in Chapter 3. It is also shown there that coder design based on the signal-to-noise ratio (*SNR*), obtained by comparing $x(n)$ and $y(n)$ as in Figure 1.1, is equivalent to a design that would result by comparing the analog versions $x'(t)$ and $y'(t)$. It should also be noted that coder inputs can sometimes be amplitude-discrete. Examples are two-level graphics images and computer-stored multi-level waveforms used in coder simulations and implementations. In the former example, the aim is to transmit two-level graphics without introducing further degradation. In the latter situation, one has a sequence of multi-level signal samples (say, 8-bit or 256-level image samples and 12-bit or 4096-level speech samples); and the object is to reduce the bit rate for ease of transmission or storage, with a non-zero but hopefully minimum possible amount of signal degradation.

It must be emphasized that *digital* waveform communication still uses physical channels that transmit *analog* waveforms. Digital systems for waveform storage use analog media or "channels" as well. In Figure 1.2, double lines indicate digital signal paths, and the *modulator* and *demodulator* boxes provide the interface between analog and digital signals in the communication system.

In *binary* digital communication, which is assumed in this book, the output $\{b(n)\}$ of the waveform encoder, as well as the output of the channel encoder, are sequences whose elements are 1 or 0. Note that the binary code letters 1 and 0 are *logical* symbols and will be used as such in this book. Electrical signals that are used to convey the above binary information can be members of several binary or non-binary sets: for example, pulse, or sinusoidal signals with amplitude levels (S and 0) volts, or ($+S$ and $-S$) volts, or ($+S$, 0, and $-S$) volts. The choice of an electrical signal set is dictated by transmission requirements such as the need to avoid a dc or zero frequency component in the transmitted signal spectrum, ease of synchronization and automatic error detection [Bennett, 1978] [Shanmugam, 1979] [Schwartz, 1980]. An example of a practical electrical signal set is that in a *bipolar* system where a logical 0 is represented by 0 volts, and a logical 1 is represented alternately as $+S$ and $-S$, with the alternation taking place at every occurrence of a 1. With an equiprobable (1, 0) output from the digital waveform coder, the resulting ternary sequence does not have a dc component; further, the frequent zero crossings permit easy recovery, when needed, of the discrete-time information from the received channel signal; and finally, the $\pm S$ alternation property results in automatic error detection of any isolated error, which causes an immediate violation of the alternation rule.

Figure 1.2(a) has introduced the notion of a *digital channel*. The several components of such a channel are shown in Figure 1.2(b). The modulator in this figure maps the channel encoder outputs 1 and 0 into analog signals suitable for the physical transmission channel, while the demodulator uses an appropriate decision threshold to interpret received signals as corresponding to either 1 or 0. The role of the *channel encoder* is to add controlled redundancy into a bit stream so that its inverse, the *channel decoder*, can correctly interpret digital sequences even in the presence of some channel noise and consequent bit errors in the demodulator. The channel encoder, the modulator, the physical channel, the demodulator and the channel decoder are sometimes lumped into one ideal box, a *noiseless digital*

0020

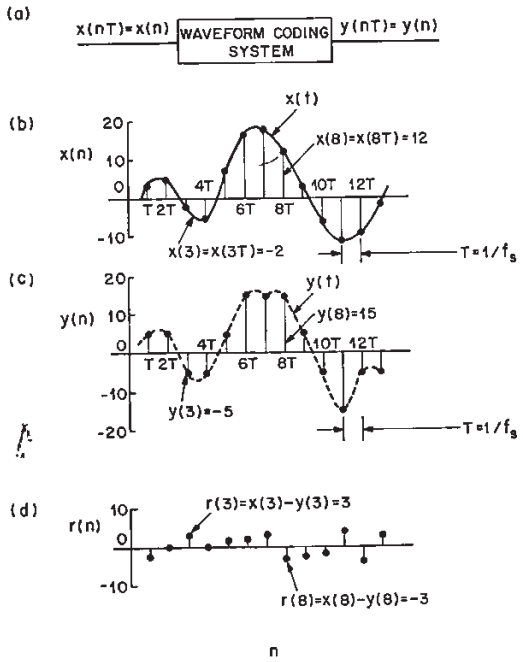
channel; in this ideal case, $b'(n) = b(n)$ in Figure 1.2. However, in this book we allow the possibility of a *noisy digital channel* with a non-zero *bit error probability* p_e. This error probability is sometimes called the *bit error rate*. When $p_e \neq 0$, the bit sequence $b'(n)$ available to the waveform decoder is no longer identical to the sequence $b(n)$ provided by the waveform encoder.

In Figure 1.2, codewords at digital channel inputs and outputs, which are really sequences of R consecutive values of $b(n)$ and $b'(n)$, respectively, have been denoted by symbols $u_c(n)$ and $v_c(n)$. In succeeding chapters, we will consider the channel as a mapping between an amplitude-quantized input $u(n)$ and an amplitude-quantized output $v(n)$. The mapping from $u(n)$ to $u_c(n)$ and the mapping from $v_c(n)$ to $v(n)$ will be implicit in those discussions.

1.1.2 Signal-to-Noise Ratio and Transmission Rate

Figure 1.3 can be used to quantify the encoding rate and distortion parameters mentioned earlier. Inputs and outputs to the waveform coding system are represented by illustrative samples $x(nT)$ and $y(nT)$, where T is the *sampling interval* — the reciprocal of *sampling rate* f_s — and n is a discrete-time index. For simplicity, these samples will henceforth be denoted by $x(n)$ and $y(n)$. In

Figure 1.3 Waveform coding example: (a) block diagram of coding system; (b) input waveform $x(n)$; (c) output waveform $y(n)$; and (d) reconstruction error waveform $r(n) = x(n) - y(n)$.



0021

waveform communication, information needed to reconstruct the *digitized* signal $y(n)$ is conveyed to the receiver by means of a code sequence, usually a binary code sequence $\{b(n)\}$. The *reconstruction error* $r(n)$ is defined as the difference between $x(n)$ and $y(n)$:

$$r(n) = x(n) - y(n) \tag{1.1}$$

Let the variances of $x(n)$, $y(n)$, and $r(n)$ be denoted by σ_x^2, σ_y^2 and σ_r^2. These variances are formally defined in Chapter 2, but we note here that in the special case of zero-mean signals, variances are simply equal to respective mean square values measured over an appropriate sequence length M:

$$\sigma_u^2 = \frac{1}{M}\sum_{n=1}^{M} u^2(n) \quad \text{where} \quad u = x, y, \text{ or } r \text{ is a zero-mean signal} \tag{1.2}$$

A standard objective measure of coded waveform quality is the *ratio of signal variance to reconstruction error variance*, referred to for historical reasons as the *signal-to-noise ratio (SNR)*, usually expressed in decibels (dB):

$$SNR \text{ (dB)} = 10 \log_{10} (\sigma_x^2/\sigma_r^2) \tag{1.3}$$

Unless otherwise specified, subsequent uses of the log $(\cdot)$ function in this book will imply $\log_{10}(\cdot)$, as in (1.3).

The use of a mean square function σ_x^2 to measure input signal level is convenient and mathematically tractable. In image coding work, input amplitudes are often considered to be all-positive quantities with a non-zero mean; and their maximum value x_{max} is sometimes used in the numerator of the *SNR* formula instead of the root-mean-square (rms) value σ_x used in (1.3). The values of *SNR* in such measurements will be much higher than the values obtained from (1.3), with a difference in the order of 10 dB.

The reconstruction error variance σ_r^2 is often referred to simply as the *mean square error* (mse) and coder designs that minimize this quantity are called *minimum mean square error* (mmse) designs.

The use of a mean square function for measuring waveform distortion implies that large values of reconstruction error are much more serious than smaller values thereof. The mean square formulation is also a time-honored one, and the signal-to-noise ratio (1.3) is the single most used criterion of coder performance. As we progress in this book, we shall make ubiquitous use of the above-defined *SNR*. But we shall also note, when appropriate, the inadequacy of *SNR* as a perceptually meaningful measure of digitized waveform quality. Very briefly, the inadequacy of *SNR* has to do with the fact that reconstruction error sequences in general do not have the character of signal-independent additive noise; and that therefore, the seriousness of the impairment cannot be measured by a simple power measurement. This can be illustrated by the simple example of a small geometric tilt of a picture signal. This kind of impairment can lead to a very large value of σ_r^2, and hence a very small value of *SNR*, although the slightly tilted image can be perceptually very acceptable. An example more relevant to digital coding is that of a thin but

bright horizontal streak in an image (Chapter 6). In this case, σ_r^2 can be quite small and the *SNR* can be very high, but the perceived image quality can be quite unacceptable. Similar examples can also be given in the field of speech digitization. In fact, in both image and speech coding work, the truly definitive measures of coded signal quality are perceptual ones, as measured by careful subjective experimentation (Appendix F). However, subjective testing of coding systems can be very time-consuming; and several refinements of the *SNR* will therefore be invoked in the course of this book in an attempt to make this time-honored objective quantity subjectively meaningful (Appendix E).

The *transmission rate* I of the system of Figure 1.3 is the product of the *sampling rate* f_s and the number of *bits per sample* R used by the system to represent input amplitudes:

$$I = f_s R \quad \textit{bits per second} \text{ (b/s)} \tag{1.4}$$

The rate I may also be expressed in *kilobits per second* (kb/s) or *megabits per second* (Mb/s). For lowpass signals, and in fact also for bandpass signals under a simple constraint (Chapter 3), the sampling rate f_s (in *samples per second*) should be no less than $2W$, where W is the input signal bandwidth (in *hertz*, or Hz); in fact, in practice, $f_s > 2W$. Very often, when W and f_s are known and fixed, the transmission rate is simply specified by the *rate*

$$R \quad \textit{bits per sample} \text{ (or } \textit{bits/sample}) \tag{1.5}$$

In coding literature, as well as in this book, the expression *bit rate* is used to represent both I and R. The units that are specified for the bit rate will indicate which of the two quantities is actually implied in a given context.

Several coding systems, including the simple example of PCM (Pulse Code Modulation) use a single 2^R-level quantizer for discretizing all amplitude samples. In such systems, the number of bits used to represent each sample is simply $\log_2 2^R = R$. In coding systems such as Sub-Band Coding and Transform Coding, the input signal is represented as a sum of several components each of which is separately digitized, with different values of sampling rate and/or bits per sample. The transmission rates I in these systems will be simple generalizations of the product in (1.4).

In the example of Figure 1.3, $R = 2$ bits/sample; each input amplitude is quantized to one of $2^2 = 4$ possible levels: +15, +5, −5 and −15. These levels could be digitally represented by using the set of four codewords 00, 01, 10 and 11. The coarseness of the quantization in Figure 1.3 is reflected by the occurrence of very large reconstruction error samples. For example, the magnitude of $r(3)$ in this example is actually greater than that of the corresponding input, $x(3)$.

Figure 1.4 is an example of much finer quantization; it is based on a speech input, $R = 8$ bits/sample and therefore $2^8 = 256$ levels. The speech waveform in the illustration is taken from the speech sound "Joe" which includes unvoiced speech (the left half of the waveform) as well as voiced speech (the right half of the waveform). It is one of the test waveforms used repeatedly in the course of this book. The output levels of the quantizer, unlike in Figure 1.3, are also non-

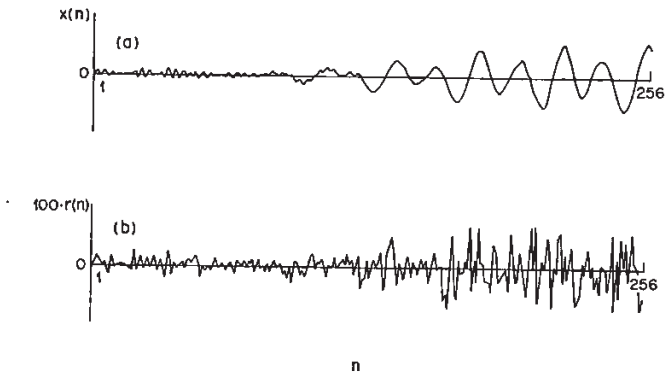


Figure 1.4 Waveform coding example with $R = 8$ bits/sample. Time-continuous versions of (a) speech waveform $x(n)$ and (b) reconstruction error waveform $r(n)$ in log-PCM coding (Chapter 5) magnified by a factor of 100 [Jayant, 1981].

uniformly spaced, a design that is characteristic of high-quality speech digitization (Chapter 4). The reconstruction error waveform has been magnified by a factor of 100 in order to bring out its characteristics. Notice also that this magnification makes the waveform in (b) roughly comparable in amplitude levels to the waveform in (a). According to (1.3), this would suggest an error variance that is smaller than the signal variance by a factor in the order of 10000, and hence an *SNR* in the order of 10 log 10000 = 40 dB. We will note in Chapter 4 that the *SNR* provided by the illustrated coding scheme is indeed very close to 40 dB — 38 dB, to be exact.

We will note later in this book that for PCM systems, the signal-to-noise ratio *SNR* increases at a rate very close to 6 dB per additional bit:

$$\sigma_r^2 = a\ 2^{-2R}\sigma_x^2\ ; \quad SNR\ (\text{dB}) = 6R - 10\log_{10} a$$

$$a \text{ is a constant in the order of 1 to 10 .} \tag{1.6}$$

Note finally that the waveforms in Figure 1.4 are really time-continuous versions of $x(n)$ and $r(n)$. In many future illustrations of waveforms in this book, the phrase "time-continuous" is often not stated, but implied.

1.1.3 Examples of High-Quality PCM Digitization

Table 1.1 puts the quantities f_s, R and I in perspective by invoking important signal examples. Many of the entries in Table 1.1 are order-of-magnitude values. For example, the bandwidth of telephone speech is usually 3.2 or 3.4 kHz, as in later chapters of this book, although the sampling rate is maintained at the standard value of 8 kHz. The 40 kHz rate for music is appropriate for studio-quality, 20 kHz bandwidth material. Lower bandwidths such as 15, or even 7 kHz are sometimes employed in audio systems. Corresponding sampling rates can be less than 40 kHz. Television signals may be sampled at 14.3 MHz or 13.5 MHz in

0024

Table 1.1 Examples of high-quality digitization with a PCM (Pulse Code Modulation) coder.

Waveform to Be Digitized	Signal Bandwidth W (kHz)	Sampling Rate f_s (kHz)	PCM Bit Rate	
			R (bits/sample)	I (kb/s)
Telephone speech	4	8	8	64
Music	20	40	12	480
Videophone	1000	2000	8	16000
Television	5000	10000	8	80000

systems for color television; *component coding* systems include two separate signals for color information, each sampled at a lower rate such as 6.75 MHz [CCIR, 1982]. The bit rates in the corresponding PCM systems will be higher than the value cited in the table. Even higher sampling rates and bit rates may be used in the processing of studio television images.

The 2000 kHz or 2 MHz rate for videophone refers to a sampling of 2×10^6 *pic*ture samples or *el*ements (*pixels*) per second. For a 30 Hz frame rate, this implies a resolution in the order of $2 \times 10^6/30 \sim 0.067$ million pixels/frame. Indeed, a 225-line picture, with a 4:3 aspect ratio, has a total of $300 \times 225 \sim 0.067$ million pixels/frame. An image of this kind is not necessarily a well-accepted standard, but it will serve as a useful example in many discussions of this book. The resolution in a 5 MHz bandwidth television frame is much greater, in the order of $700 \times 525 = 0.37$ million pixels/frame. Some of the illustrations in this book are indeed based on this degree of resolution, and these images will appear sharper than the lower resolution examples. High-quality graphics transmission offers frame (page) resolutions exceeding that in commercial TV. A high-resolution CCITT standard, for example, calls for $2128 \times 1728 \sim 3.7$ million pixels/page.

1.2 Subjective Quality, Bit Rate, and Coder Complexity

In Section 1.1, we noted that an important problem in waveform coding is one of maximizing *SNR* for a given transmission rate I. We shall now amplify on this theme. In doing so, we shall introduce several additional parameters. All of these parameters will need to be considered before a specific waveform coder is selected for a given communications task. The first additional parameter to be considered in this section is *subjective quality*. Before discussing this notion, it will be useful to attempt a more critical description of the term *waveform coding*.

1.2.1 Waveform-Preserving Coders

Waveform coders can be defined as that class of analog signal coders where an input signal is approximated by mimicking the amplitude-versus-time waveform.

A viable alternative for digitizing waveforms such as speech and image signals consists in a signal-analysis procedure for extracting perceptually significant parameters from the input signal and using these parameters to synthesize an output signal that is useful to the human receiver, although not a waveform replica of the input speech or picture. Such *analysis-synthesis* procedures (for example, speech *vocoders)* provide the greatest economies in terms of bit rate, but they also entail fundamental limitations on subjective quality, and great expenses in terms of coder complexity and cost. Furthermore, the design and performance of such coders tend by definition to be highly input-specific, even while dealing with signals within one class. Excellent discussions of input-specific devices appear elsewhere, in contexts such as vocoding of speech [Flanagan, 1972] [Rabiner and Schafer, 1978] [Flanagan et al., 1979] and contour coding of images [Pratt, 1978] [Gonzalez and Wintz, 1977] [Rosenfeld and Kak, 1976].

The waveform coding principles discussed in this book are by contrast largely non-signal-specific. Although refined versions of certain waveform coders are intentionally tailored to input classes, they differ from analysis-synthesis systems in that the coder outputs are not synthesized from features, but rather, made to follow and preserve the input waveforms per se. Thus, in the waveform coding of speech, a basic criterion will still be the input-output difference $r(n)$, although a sophisticated mapping from the $r(n)$ sequence to a distortion measure may include facts about the perception of speech signals, and a coder that seeks to minimize this distortion may have to use facts or statistics about speech production in order to refine the basic blocks (for example, quantizer, predictor) of the waveform coding system.

1.2.2 Subjective Quality Versus Bit Rate in Speech Coding

In the coding of signals such as speech and video, subjective measures of quality are more relevant than objective measures such as the *SNR*. However, reliable values of subjective measurements are in general more difficult to obtain.

The dependence of digitized signal quality upon coder bit rate is well exemplified by subjective results from speech coding, as shown in Figure 1.5. Output speech quality in Figure 1.5 is measured by a *Mean Opinion Score* (MOS) (Appendix F) as obtained from formal subjective testing [Daumer, 1982]. Unlike objective *SNR* measurements (Section 1.1.2) and bounds (Section 1.3.2), subjective quality of digitized speech tends to saturate as the bit rate is increased, with a "knee" of the MOS versus bit rate curve that is coder-dependent. A score of 5.0 implies perfect quality. But this is hardly ever attained, even by an undigitized speech stimulus, as explained in Appendix F. A score approaching 4.5, on the other hand, may be regarded as a *necessary* condition for *toll quality*, the quality of commercial telephony. This quality roughly implies that digitized speech will be indistinguishable from the bandlimited input in a subjective test involving pairwise comparisons of stimuli. An MOS score in the order of 4.5 also implies that the *intelligibility* of the coded (digitized) speech is practically equal to that of the original. Toll quality thus defined is higher than *communications quality* [Flanagan et al., 1979] which is a term used to connote detectable distortion, but

0026

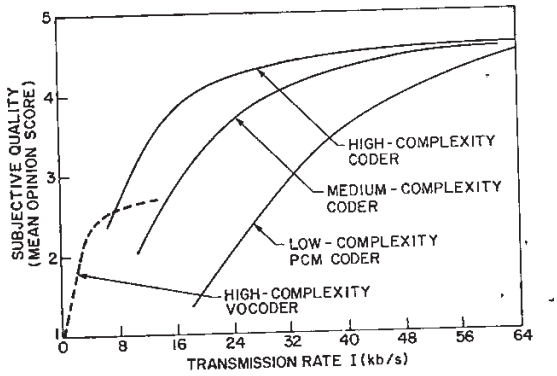


Figure 1.5 Subjective quality versus bit rate I in speech digitization [Daumer;©1982, IEEE] [Flanagan et al.;©1979, IEEE]. The three levels of waveform coder complexity are illustrated by several examples in later chapters.

still very little degradation of intelligibility. Toll quality will be more completely defined, using objective specifications by CCITT (International Telegraph and Telephone Consultative Committee), in Chapter 5. That discussion will include further *sufficient* conditions for a rigorous definition of toll quality. These conditions will reflect traditional communications network criteria such as coder performance with multiple stages of coding and decoding, and coder performance in the presence of impairments introduced by other parts of the network. Logarithmic PCM using $R = 8$ bits/sample ($I = 64$ kb/s) will be seen to be a toll quality coder in this traditional and rigorous sense (Chapter 5).

The upper solid curve in Figure 1.5 denotes the performance of the best waveform coder at a given bit rate, without regard to complexity or cost. In Figure 1.5, speech digitization with toll quality or almost toll quality is not realized at bit rates below $I = 16$ kb/s, or $R = 2$ bits/sample. In fact, even the high-complexity coders that achieve an MOS score approaching 4.5 at $I = 32$ kb/s, or $R = 4$ bits/sample, have to satisfy further criteria before they can be regarded as toll quality coders.

An important focus of continuing work in speech coding is to realize an upward shift of the topmost curve of Figure 1.5, and a leftward shift of the knee of that curve, implying improved performance in the range of $R = 2$ to 1 bits/sample, or less.

Lowest in the hierarchy of speech quality levels is *synthetic quality* that applies to vocoders. This quality is characterized by substantial loss of naturalness and robustness with respect to speakers and speaking environments. The broken curve in Figure 1.5 describes an optimistic estimate of the general performance of vocoders. These digitizers are appropriate for digitizations at very low bit rates, but their synthetic quality results in a ceiling of performance that is clearly well below the best performance of waveform coders. As indicated earlier, vocoders for synthetic speech are beyond the subject of waveform coding, and they will not be discussed henceforth in this book. For similar reasons, we will exclude formal

discussions of very specific techniques such as those using *time compression* or *harmonic scaling*. Instead, we will cite some specific examples of the use of such techniques, as appropriate.

Also beyond the scope of this book are bit rate reduction techniques such as TASI (*time assignment speech interpolation*) [Flanagan, 1972] and DSI (*digital speech interpolation*) [IEEE, 1982], which exploit silences in speech to reduce the effective bit rate, and allow efficient time-sharing of digital channels. These techniques can be used in conjunction with conventional digital coders such as PCM, as well as with all classes of bit rate reducing coders, to provide further reductions of overall bit rate. Examples of such combinations will be cited as appropriate.

Subjective results such as MOS scores constitute an indispensable supplement to objective measurements of *SNR* or noise variance. For example, based only on objective *SNR* measurements, 16 kb/s speech coders, however complex, are inferior to 64 kb/s PCM by an order of magnitude. On the other hand, according to the subjective results of Figure 1.5, speech coders at 16 kb/s can be designed to provide an MOS quality that falls short of the 64 kb/s score by no more than a half point. The fact that non-PCM coders perform better in subjective assessments is due to the fact that the perceived annoyance of the reconstruction errors in these coders is less than that of PCM errors, for the same error variance σ_r^2 (or equivalently, for the same measured *SNR*). This will be illustrated repeatedly in this book for both speech and video coding, and for both the medium-complexity and high-complexity coder classes in each case.

Subjective results are critical for image coding as well, and criteria have been developed for different levels of quality, the highest of which is *broadcast quality* [Netravali and Limb, 1980]. Broad conclusions for image coding will be similar to those from Figure 1.5 after an appropriate rescaling of the transmission rate axis. The rescaling roughly involves multiplication of the bit rates on the x-axis of Figure 1.5 by a factor of 1200, the ratio of image bandwidth (5 MHz) to telephone speech bandwidth (4 kHz).

1.2.3 Impressionistic Results for Four Classes of Signals

The results of Figure 1.6 refer to very high quality digitizations of the four classes of signals mentioned in Table 1.1, and they utilize both ways of expressing coder bit rate: bits per second I and bits per sample R. Figure 1.6(a) uses an exponential bit rate axis to accommodate a six-orders-of-magnitude range of I. It shows that television is the most demanding system in terms of overall bit rate I, a natural consequence of the very high sampling rates in commercial video (see Table 1.1). On the other hand, Figure 1.6(b) indicates that studio-grade music is the most demanding signal in terms of bits per sample R.

1.2.4 Coder Complexity

Figures 1.5 and 1.6 have introduced the notion of coder complexity. Complexity is a continuous quantity with strictly an infinite range, but the figures use only

0028

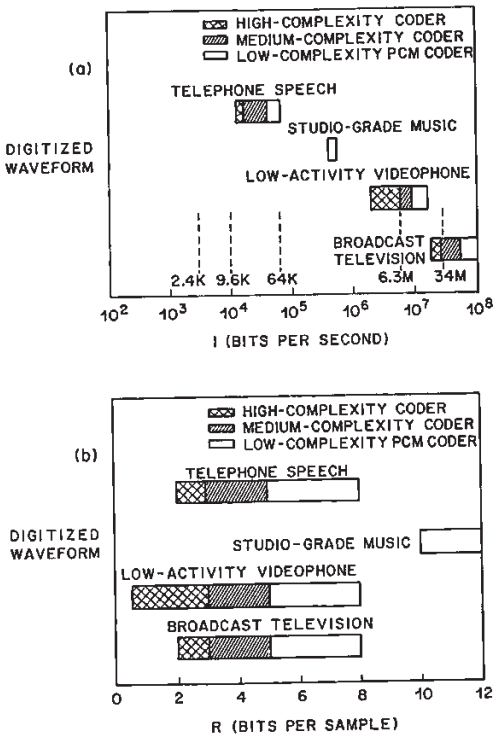


Figure 1.6 Exchange of coder complexity and transmission bit rate for very high quality digitizations of four signals [Jayant, 1981]. Bit rates are expressed as bits per second in (a), and as bits per sample in (b). An important focus of continuing research is the improvement of coder performance at bit rates lower than the limits indicated in this figure.

three levels of complexity for ease of illustration. The simplest of these levels would refer to *a low complexity coder* [such as PCM (Chapter 5) or DM (Chapter 8)]. The next level refers to a *medium complexity coder* such as DPCM with nonadaptive prediction (Chapter 6) or SBC with nonadaptive bit allocation (Chapter 11), which offers a significantly lower bit rate than PCM for a given quality. The last level denotes a *high complexity coder* [such as APC (Chapter 6) or ATC (Chapter 12)], which provides the very best results in bit rate reduction by utilizing highly adaptive algorithms and/or extensive use of long-term waveform memory.

The levels of coder complexity listed above can also be quantified by implementation criteria such as the number of multiply/add operations involved per waveform sample. Advances in technology tend to bridge the differences between algorithms of differing complexity, and they tend to make complex algorithms increasingly more practical. A factor that will continue to separate algorithms of different complexity, making some algorithms less useful than others, is the factor of encoding delay. The high-complexity coders of Figures 1.5 and 1.6 are also characterized by the highest levels of encoding delay.

1.2.5 Digital Hierarchies

The special values of I shown in Figure 1.6(a) by dashed lines are numbers that are standardized in practical digital systems. Examples of such systems are illustrated in the network hierarchy of Table 1.2.

Table 1.2 Digital Network Hierarchy.

	Level in Digital Hierarchy	1	2	3	4
United States	Number of 64 kb/s channels	24	96	672	4032
	Total bit rate (Mb/s)	1.544	6.312	44.736	274.176
Europe	Number of 64 kb/s channels	30	120	480	1920
	Total bit rate (Mb/s)	2.048	8.448	34.368	139.264
Japan	Number of 64 kb/s channels	24	96	480	1440
	Total bit rate (Mb/s)	1.544	6.312	32.064	97.728

Practical waveform coding systems assume an available bit rate I and seek to maximize coder performance with these constraints. Important numerical values of I depend on the digital system hierarchy in question (Table 1.2); for example, a rate of 1.544 Mb/s, which is the bit rate supported by the so-called T1 carrier system in the United States, can include 24 speech channels using 64 kb/s each. A television system in Europe may be required to have a bit rate equal to 34.368 Mb/s, while a television system in the United States may use a transmission rate of 44.736 Mb/s that fits into level 3 of Table 1.2. Coder rates slightly under $45/N$ and $35/N$, where N is integral, would also be attractive in respective situations; they would permit the multiplexing of N video channels. The sampling rate f_s depends on the input bandwidth, being at least twice that value. The value of f_s is typically 8 kHz for telephone speech, and 10 MHz or higher for TV images. In both speech and video, there will be systems that need to oversample the input (Chapter 8). In these systems, values of f_s will be significantly higher than the numbers cited above.

Submultiples of 64 kb/s, such as 32 kb/s and 16 kb/s, are of importance in so-called *low bit rate* systems for commercial telephony [IEEE, 1982]. Bit rates such as 9.6 kb/s and 2.4 kb/s [Figure 1.6(a)] have evolved from data communication practice. *Very low bit rate* speech coding, such as that with 2.4 kb/s, has been used for a long time in tactical speech communication systems. In these systems, the need for digital encryption has motivated the use of digitization at bit rates that are not high enough for high quality speech, but low enough for the output of the digital coder to be transmitted over conventional analog media such as a 4 kHz telephone line or a HF-band radio link.

Integrated Services Digital Networks (ISDN). We have seen from the numbers in Table 1.2 that 64 kb/s represents a standardized unit of bit rate. This unit, together with its multiples or sub-multiples, is also the basis of proposals for the

0030

designing of *Integrated Services Digital Networks* (ISDN) [Decina, 1982], which are networks providing end-to-end digital connectivity to support a wide range of services, including voice, data, audio, graphics and video. One important example of integration in such networks is the use of a 64 kb/s facility for the simultaneous transmission of graphics and voice. The bit rate allocated to voice coding in such a system would be in the range of 8 to 56 kb/s; for example, it may be one of the standard bit rates mentioned in the previous paragraph. The possibility of *variable rate coding* of the speech signal affords further flexibilities in the design of ISDN systems. Examples of such coding methods will be mentioned in this book when appropriate.

Circuit Switching and Packet Switching. The network architecture in digital systems may be based either on conventional *circuit switching* or *packet switching*. In circuit switching, a given pair of users (message sender and message receiver) have a dedicated connection for the length of their communication, with a guaranteed transmission rate. Time Division Multiplexing (TDM) is used to carry assemblies of lower-rate channels onto a single channel, such as a 64 kb/s facility, using appropriate padding/depadding techniques if necessary. In packet switching, messages from each user of the network are segmented into standard-sized smaller units, or *packets*, of appropriate length. The packets traverse the network as individual entities that are subsequently reassembled at the receiver in proper order. Packets from different parts of the message, as well as packets from several users, are statistically multiplexed before transmitting them over digital links that operate at a specific standard bit rate.

1.2.6 Waveform Properties and Encoding Delay

Coder complexity is a function of the signal processing involved. It is also related to *encoder delay* or *encoder memory*, which shows the extent to which a waveform has to be observed in order for the coder to exploit waveform structure for economical (digital) representation. Examples are listed in Table 1.3. This list attempts to grade quantizer and coder classes in order of increasing encoder memory or delay; and this hierarchy is closely matched to a similar hierarchy of waveform properties in Chapter 2. The utilization of these properties calls for increasing amounts of coder memory. The classifications in Table 1.3 are based on typical designs, but counter-examples are possible: for example, in speech coding, depending on the extent of memory utilized, sub-band coders, vector quantizers and tree coders may well be included in the class of coders with long-term memory.

The frame period in video is either 33.33 ms or 40 ms, depending on power line frequency, while the pitch period in speech can be in the range 3 to 15 ms, depending on speaker. The use of *long-term memory* in Table 1.3 indicates waveform observations over a duration in the order of the pitch period in speech, and the frame period in video. This implies transmission delays that can be several tens of milliseconds in both speech and image coding. Such delays can have important ramifications in aspects of waveform communications separate from coding per se, such as echo control in long-distance two-way transmissions.

Table 1.3 A listing of coding systems in terms of encoding delay.

Waveform Memory Used in Coding	Coding System Example
Zero Memory	• Instantaneous Quantizers: PCM
Short-term Memory	• Adaptive Quantizers • Differential PCM systems with near-sample-based predictions • Deltamodulators • Sub-Band Coders for Speech • Vector Quantizers, Tree- and Trellis Coders • Intraframe Transform Coders for Images
Long-term Memory	• Differential PCM systems with distant-sample-based predictions • Transform Coders for Speech • Interframe Transform Coders for Images

1.2.7 Irrelevancy and Redundancy

We have already noted that PCM coders are the simplest in principle; given a time-discrete input, they involve nothing more than quantization and an appropriate binary mapping. Quantizers are devices that remove the *irrelevancy* in this signal; they accomplish this through an irreversible, *information-lossy* procedure. For example, if a fractional resolution in image intensity that is smaller than 1/256 of the total amplitude range is visually unrecognizable or irrelevant, there is no need for digitized images to have more than 256 levels. If, indeed, the image has more levels (say, 1024 or infinity as with an analog image), this irrelevancy can be removed by passing the image signal into a quantizer with 256 output levels, each of which can be described by $R = \log_2 256 = 8$ bits/sample.

Quantization, as described above, is a zero-memory process, and is discussed as such in many parts of this book. However, there are important classes of quantizing systems which do depend on the utilization of waveform memory. Examples of such systems are adaptive quantizers and vector quantizers.

Non-trivial waveform coders (PCM will be the only trivial case) are designed to remove the *redundancy* in a signal for purposes of bit rate reduction. They do this through a reversible, *information-lossless* procedure that depends critically on the utilization of waveform memory. For example, consider an artificial $N \times N$ array of 8-bit picture elements (pixels) which is redundant in that every even pixel on a scan line (row of array) is identical in intensity to the odd pixel preceding it. A clever coding system exploits this redundancy by transmitting only odd pixels, and reduces the bit rate by a factor of 2. We will discover more realistic redundancies in video and speech, as we traverse this book, and learn about time- and frequency-domain systems which exploit these redundancies. Figure 1.7 formalizes the concept of reducing irrelevancy and redundancy. It indicates an interesting

0032

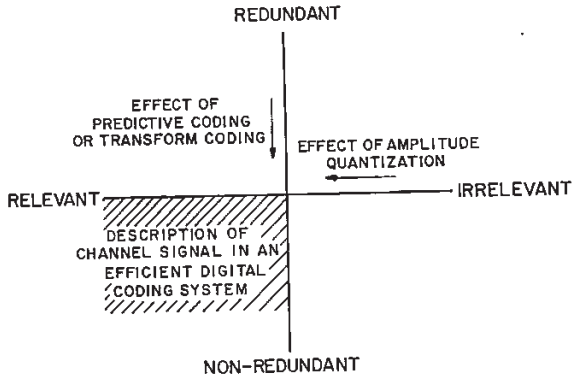


Figure 1.7 Irrelevancy and Redundancy [Schouten, 1957]. The distinction between quantization and coding is a convenient concept for most of the discussion in this book. There are examples of digitizing systems where the processes that reduce irrelevancy and redundancy cannot be clearly demarcated.

region that digital waveform coders seek to be in — a region where digitized signals are *perceptually relevant* and *statistically non-redundant*. In coding systems such as differential PCM and transform coding, there is significant and sometimes uncontrollable interaction between stages that reduce redundancy and irrelevancy; and in systems such as vector quantizers for correlated inputs, the stages that reduce redundancy and irrelevancy cannot be clearly demarcated.

1.2.8 Transmission Errors

The coders in Figures 1.5 and 1.6 have also been tested in the presence of binary transmission errors. Depending on the coder and the signal being digitized, *acceptable* values of random bit error probability p_e can be as demanding as 10^{-6} and less, or as high as 10^{-2} and more. Examples in this book will show that, in general, high quality image coding systems require lower values of p_e than high quality speech coding systems. *Actual* bit error rates can range from negligible values (say, less than 10^{-6}) in high-quality digital systems that may sometimes include explicit error control, to very high values (say, greater than 10^{-2}) in environments such as mobile radio. In the radio example, bit errors are also non-random and clustered in time. Most results in this book will refer to the simpler case of independent random errors, with the following order-of-magnitude descriptions of channel quality:

$$\begin{aligned} p_e &< 10^{-6} \quad \text{in } \textit{high quality} \text{ channels} \\ p_e &> 10^{-2} \quad \text{in } \textit{very noisy} \text{ channels} \end{aligned} \tag{1.7}$$

In this book, the quality of a digital channel is specified simply by the parameter p_e. In practice, digital channel performance may need additional characterization. An important performance parameter is the *percent of time*

intervals which have more than a specified error rate [Johannes, 1983]. For example, in high quality speech transmission at 64 kb/s, a standard requirement is for at least 90% of 1 minute intervals to have an error rate of 10^{-6} or less [CCITT, 1980].

For the sake of better output quality, practical waveform coding systems often include some kind of explicit error protection, incurring a small penalty in transmission rate in the process. The gains due to such error protection can be significant for one of two reasons. First, if the value of p_e is very small to start with, the effect of error protection on the *intelligibility* of the digitized signal is very small, but the protection can enhance *quality* by increasing the mean time between (the already infrequent) bit errors. On the other hand, if p_e is objectionably high to start with, the impairment in the received output may be so significant that error protection coding may make the crucial difference between an unacceptable system and an acceptable one. While we shall illustrate gains due to explicit error protection whenever appropriate, the actual design or description of error-protection procedures [Berlekamp, 1974] will be beyond the scope of this book.

More relevant to this book is the fact that there are ways of making waveform coding systems inherently more *robust* to transmission errors. In other words, the effects of a given p_e can be rendered perceptually more acceptable or tolerable by careful designs of coder components such as quantizers and predictors. We shall comment on such designs repeatedly in the course of this book.

Because of its special relevance to this book, we illustrate a binary digital channel more explicitly in Figure 1.8(a) [Berlekamp, 1974]. An important class is the *memoryless binary symmetric channel* defined in Figure 1.8(b), where the bit error probability p_e is the same whether the input was 1 or 0, and it is independent of whether the previous input was received correctly or not. The error rate p_e is a monotonically decreasing function of analog channel quality, and Figure 1.8(c) provides a quantitative illustration of this dependence for the example of binary phase modulation or binary *phase shift keying* (PSK) over an additive Gaussian noise channel [Viterbi and Omura, 1979]. Channels used in waveform communication may indeed be non-Gaussian, and the signal impairments may indeed be non-additive in some cases; but the additive Gaussian model is simple, often approximately valid, and adequate for the points we wish to make. The illustration also assumes coherent detection, which means that a perfect reference carrier is available at the receiver for demodulation. Without channel encoding for error reduction, this modulation system realizes a p_e value of 10^{-6} with a *signal-to-channel-noise ratio per binary digit* (ratio of signal energy per bit to one-sided power spectral density of channel noise), denoted by s_n, of about 10.8 dB. With $s_n = 7$ dB, on the other hand, $p_e = 10^{-3}$, a value that can be seriously high for certain classes of waveform digitization, as we shall see in later chapters. Also shown in Figure 1.8(c) is the *Shannon limit*, which suggests the possibility of a zero error rate whenever s_n exceeds a mere -1.6 dB (Appendix C). This figure is only a bound, however. With an analog channel corrupted by Gaussian noise, the above bound can strictly be realized only with Gaussian input signals and infinite delays in channel encoding and channel decoding. The gap between the two curves

0034

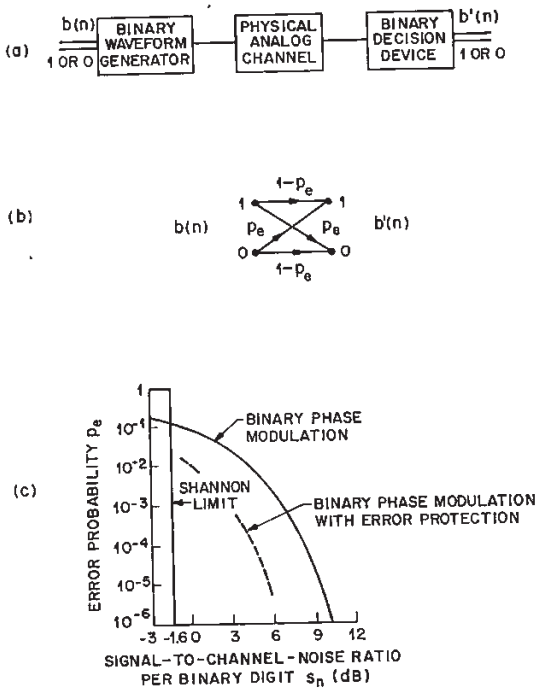


Figure 1.8 (a) Block diagram of a binary digital channel; (b) binary symmetric channel with bit error rate p_e; and (c) dependence of p_e on signal-to-channel-noise-ratio per binary digit [Forney;©1970, IEEE].

in Figure 1.8(c) can be significantly bridged by practical procedures for error protection. This is particularly appropriate for the better utilization of *power-limited* analog transmission media such as satellite communication channels. If power limitations are not serious, as in telephone lines, which are really *bandwidth-limited* media, better channel utilizations are also possible with more-than-two-level modulations (of phase and/or frequency and/or amplitude) [Lucky, Salz and Weldon, 1968] [Shanmugam, 1978] [Viterbi and Omura, 1979] [Schwartz, 1980]. Higher-level modulations, however, require greater values of s_n for a given p_e. For example, 16-level PSK requires an s_n of 18.7 dB for $p_e = 10^{-6}$, as against the 10.8 dB needed in 2-level PSK. The efficiency of a modulation scheme is also measured by the number of *bits per second per hertz (b/s/Hz)* of channel bandwidth that is characteristic of the scheme. This number is limited by considerations of intersymbol interference; upper bounds that are adequate for the present discussion are 1.0 *b/s/Hz* for two-level PSK and 4 *b/s/Hz* for 16-level PSK [Brodhage and Noack, 1979] [Oetting, 1979].

We shall now examine the relevance of the preceding discussions to digital waveform coding using the example of speech signals. Analog communication of

speech with commercial quality will require, as discussed earlier, a channel quality given by a signal-to-channel-noise ratio of at least 30 dB. Digital speech coders, on the other hand, are very robust to error rates in the order of 10^{-6}, and with the modulation systems of the previous paragraph, they require signal-to-channel-noise ratios of only 10.8 dB and 18.7 dB for operating at this error rate. The saving in terms of required channel quality is, of course, accompanied by greater demands on bandwidth. In place of the 4 kHz bandwidth required for analog speech communications, digital speech systems employing the above-mentioned modulation systems will need bandwidths of at least $I/1$ and $I/4$ Hz respectively, where I is the coder bit rate in bits/second. For a 32 kb/s speech coder, these lower bounds on bandwidth are 32 kHz and 8 kHz, respectively, signifying respective bandwidth expansions of at least 8 and 2 over the analog system. With appropriate changes of numbers — 50 dB instead of 30 dB for output signal quality, and 10^{-8} instead of 10^{-6} for tolerable p_e — the above arguments can be repeated for image coding as well. In both cases, digital communication is based on an *exchange of bandwidth for (channel) signal-to-noise ratio*; this exchange constitutes a fundamental advantage of digital coding and leads to better utilization of interference-limited media.

1.3 Information Theoretical Limits

For a given transmission rate of I bits/second, or equivalently for a given value of R bits/sample, and for a given encoding delay, waveform coding systems seek to attain the smallest value of reconstruction distortion (for example, error variance σ_r^2), complexity permitting. If complexity is indeed of no concern, an interesting parameter is the minimum value of σ_r^2 over all coding systems. The value of this minimum in turn is a monotonically decreasing function of allowed encoding delay; and the *source coding* results of information theory specify the asymptotic infinite-encoding-delay values of minimum distortion, denoted by D. These results are derived using stationary stochastic signal models, and hence their application to real signals such as speech and video waveforms should always be appropriately qualified.

1.3.1 Source and Channel Coding

The performance of a waveform communication link can be put in perspective (subject to qualifications warranted by the complex nature of real-life waveforms) by the *noisy channel coding* and *source coding* theorems of information theory [Shannon, 1948] [Slepian, 1973] [Viterbi and Omura, 1979] [Wyner, 1981].

According to the *noisy channel coding theorem*, information can be transmitted reliably (i.e., without error) over a noisy channel at any source rate, say R, below a so-called *capacity* C of the channel (Appendix C):

$$R < C \quad \text{for reliable transmission} \tag{1.8}$$

0036

According to the *source coding theorem*, there exists a mapping from the source waveform to codewords such that for a given *distortion* D, $R(D)$ bits (per source sample) are sufficient to enable waveform reconstruction with an average distortion that is arbitrarily close to D (Appendix D). Therefore, the actual *rate* R has to obey

$$R \geqslant R(D) \quad \text{for fidelity given by } D \tag{1.9}$$

The function $R(D)$ is called the *rate distortion function* [Berger, 1971]. Its inverse, $D(R)$, is the *distortion rate function.*

Both C and $R(D)$ can be related to the as-yet-undefined concept of *mutual information* (Appendix C). Refer to the waveform communication link of Figure 1.2. Let X and Y refer to sufficiently long sequences of coder input and decoder output; and let U and V refer to sufficiently long sequences of channel input and channel output. With stochastic modeling of these sequences, the average information transmitted to the destination *per sample* is given by the average mutual information $I(X;Y)$, which is a measure of the statistical dependence of X and Y. Distortion D and $I(X;Y)$ will depend on the type of source coding. There is a minimum of $I(X;Y)$, however, that is needed for a reconstruction at the destination if the average distortion must not exceed the specified upper limit D. This *minimum value of* $I(X;Y)$ is $R(D)$. The channel capacity C, on the other hand, is related to the average mutual information $I(U;V)$ *per sample* that characterizes channel input statistics and input-output mappings across the channel, as described by appropriate conditional probabilities. For example, if channel outputs do not depend on channel inputs, this information transfer is zero; while if the channel is perfect or error-free, the transfer is greatest, for any given statistics at the channel input. In fact, for a given channel, C is the *maximum of I(U;V) over all channel input statistics.* Later sections will include several discussions involving the function $R(D)$, and Appendix C will discuss the capacity C as a function of channel quality. The point to be made right now is that theorems (1.8) and (1.9) above, put together, lead to the *information transmission theorem*:

$$C \geqslant R(D) \quad \text{for reliable transmission and fidelity } D \tag{1.10}$$

This theorem states that waveform reconstruction with fidelity D is also possible after transmission over a noisy channel, provided that its capacity is greater than $R(D)$. The significance of this theorem is that it justifies the separation of *source coding* (source representation) and *channel coding* (information transmission) functions in waveform communication and provides a framework wherein the *channel controls the rate, but not necessarily the accuracy, of waveform reproduction.* Thus the only property of the channel that should concern the source coder is the single parameter C and the only requirement for the efficient utilization of the channel is that its input statistics (statistics at output of source coder) should be so as to maximize $I(U;V)$. *For the important case of a binary symmetric channel, this maximization merely requires that channel input symbols be an i.i.d. sequence (independent and identically distributed and hence*

equiprobable). This defines a *necessary* property of an optimal source coder. Additionally desired features of a digital coder output are that the i.i.d. sequence is in some sense matched to the waveform at its input; and that the number of the *i.i.d.* input symbols (say for one second of input waveform) is as small as possible.

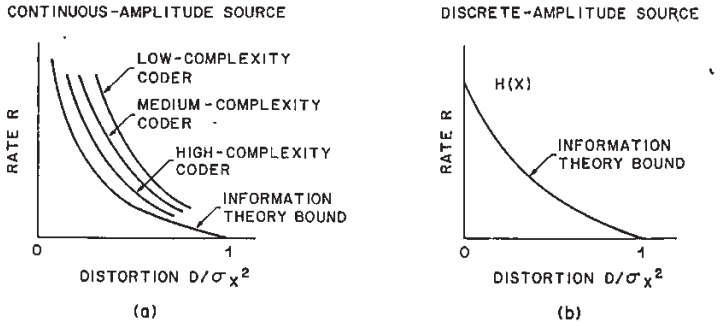
The above discussion also suggests the lumping of the channel coder, channel and channel decoder boxes of Figure 1.2 into one box, a "perfect digital channel," so that attention can be focused, if need be, on the source coding and source decoding boxes. This is indeed the approach used in most waveform coding literature. However, as mentioned earlier, because of an improper, or at best suboptimal use of an analog channel of capacity C, the binary digital channels in waveform communications will have non-zero bit error probability p_e.

1.3.2 Qualitative Distortion Rate Curves

The information transmission theorem is a very general statement. It applies to discrete-amplitude as well as continuous-amplitude sources, and to sources with or without memory (as defined in Chapter 2). The case to be discussed most in this book is that of continuous-amplitude sources with memory as indicated, for instance, by sample-to-sample correlations or, equivalently, non-flat power spectra. Discrete-amplitude sources will also be of interest in certain sections of this book. For example, facsimile systems involve two-level images; also, all quantizer outputs will be discrete-amplitude signals, by definition.

The plots in Figure 1.9 show that the rate distortion functions are monotonically non-increasing. It is reasonable that higher information-rate representations should lead to smaller (strictly, non-increasing) average distortions. It is also clear that the distortion at rate $R = 0$ should be finite if the input variance is finite. To appreciate this for the discrete-amplitude case, consider the binary signal generated by repeated tossings of an unbiased coin; let *heads and tails* signify signal amplitudes of 1 and -1, respectively (implying a variance of $\sigma_x^2 = 1$ for the zero-mean signal). Consider a zero-rate communication scheme where the receiver reproduces the binary signal by independent tossings of another unbiased coin; the

Figure 1.9 Qualitative sketches of rate distortion function for (a) a continuous-amplitude source, and (b) a discrete-amplitude source.



encoding error will be zero one-half of the time and ± 2 the other half of the time, and the average square distortion will be 2, and finite. The average square distortion will in fact be less, and equal to 1, as in Figure 1.9(b), if the zero-rate receiver has the option of constantly outputting the *neither-heads-nor-tails* amplitude of 0. The continuous-amplitude case is perhaps more interesting. Consider a zero-mean input sequence $\{x(n)\}$ with a variance (power) of σ_x^2. Regard, as the output of a zero-information-rate system, an all-zero sequence described by $y(n) = 0$; all n. The magnitude of the encoding error now equals that of the input for all its samples. The error variance therefore equals the input variance, which again is finite in the present example, and with all practical sources. In summary, $D(R)$, the inverse of the function $R(D)$, is monotonically non-increasing, and $D(0)$ is finite, as in Figure 1.9.

Discrete-Amplitude Sources. The value of $R(0)$ is by far the most interesting feature in the plots of Figure 1.9. This value is finite only for *discrete-amplitude* sources. $R(0)$ is the value of information rate necessary for exact reproduction ($D = 0$) of the source. This number is the entropy $H(X)$ (average information content) of the discrete-amplitude source. As detailed in Appendix C, all sources with a finite number K of possible amplitudes will have a finite informational entropy that is no greater than $\log_2 K$ bits/symbol. For the coin-toss source, the entropy is upper bounded by $\log_2 2 = 1$ bit/sample, and the bound is attained if the coin is unbiased; and a coding system that is willing to transmit 1 bit, or one binary item of information (say, the correct value, *heads* or *tails*, for each input toss), can reproduce coin-toss outcomes with zero distortion or errors.

Noiseless entropy coding procedures which realize or approach the bound $R(0) = H(X)$ will be described in Appendix C and applied for exact reproductions of quantizer outputs and facsimile images. Due to unequal symbol probabilities and inter-symbol dependencies, the entropy of these sources will in general be less than the maximum possible value of $\log_2 K$. The idea of entropy coding will be to exploit unequal symbol probabilities as well as source memory to realize average bit rates approaching $H(X)$ bits/symbol. Entropy coding techniques include many varieties of fixed-length-to-variable-length and variable-length-to-fixed-length mappings of codewords, and we shall illustrate different specific procedures as and when appropriate. Clearly, such mappings can also include coding with distortion; but it should be remembered that non-exact reproductions are not inevitable for discrete-amplitude sources in the sense that $R(0) = H(X)$ is strictly finite. On the other hand, non-exact mappings may be necessary purely for implementation reasons, for example, to mitigate buffering problems in variable-rate encoding.

As discussed earlier, all quantizer outputs are discrete-amplitude sources. Special examples are sequences of 2^{12}-level or 12-bit speech samples and 2^8-level or 8-bit image samples. Such sequences are often used as coder inputs prior to bit rate reduction. The entropies of these signals are typically much less than the respective values of 12 and 8 bits/sample for speech and video.

Continuous-Amplitude Sources. The average information content or entropy $H(X)$ of a *continuous-amplitude* source is in fact infinite, reflecting the fact that each

sample amplitude from such a source can have an infinite number of possible values. Another interpretation of the fact is that $R(0)$, the information rate needed for an exact reproduction of such a source, is infinite. More relevantly, no practical, i.e., finite-rate, coding procedure can reproduce a continuous waveform with perfect fidelity. Waveform coders therefore employ a value of R that is large enough so that the theoretical distortion D — or more relevantly, the actual reconstruction error variance σ_r^2 — is an adequately small fraction of input variance. The three practical coder curves in Figure 1.9(a) represent a complexity versus rate trade-off for any given value of σ_r^2. These coder classes have already been discussed in relation to Figures 1.5 and 1.6. The most complex of these coders is also the one that is closest to the rate distortion bound in Figure 1.9(a).

It will be noted in Appendix D that for a *memoryless Gaussian source*, the distortion rate function, for a mean square distortion measure, is

$$D(R)_G = 2^{-2R}\sigma_x^2 \quad \leftrightarrow \quad \max\{SNR\}\ (\text{dB}) = 6R \tag{1.11}$$

This can be shown to be an upper bound for the minimum distortion for *memoryless non-Gaussian* sources with representation rate R. In other words, smaller distortions than the above can be realized in digitizing non-Gaussian sources.

For *Gaussian sources with memory*, we shall note that

$$D(R)_G = 2^{-2R}\gamma_x^2\sigma_x^2\,; \quad 0 \leqslant \gamma_x^2 \leqslant 1 \tag{1.12}$$

where γ_x^2 is a *spectral flatness measure (sfm)* (Chapter 2) that measures redundancy as manifested in the structure and shape of the psd (power spectral density) function. For the special case of a white (flat) input spectrum, $\gamma_x^2 = 1$. Again, in the case of non-Gaussian sources, smaller distortions can be realized. Later chapters in the book will show how spectral redundancy can be removed by waveform coders with adequate encoding delay: specifically by recursive procedures called Differential or Predictive Coders, or by nonrecursive block-quantization schemes called Transform Coders.

References

M. R. Aaron, "The Digital (R)Evolution," IEEE Communications Magazine, pp. 21-22, January 1979.

J. C. Bellamy, *Digital Telephony*, John Wiley, New York, 1982.

G. H. Bennett, *PCM and Digital Transmission*, Marconi Instruments, 100 Stonehurst Court, Northvale, N.J., 1978.

T. Berger, *Rate Distortion Theory*, Prentice-Hall, Englewood Cliffs, N.J., 1971.

E. R. Berlekamp (ed.), *Key Papers in the Development of Coding Theory*, IEEE Press, New York, 1974.

H. Brodhage and W. Noack, "Planungsgrundlagen für den Einsatz von Richtfunksystemen," Siemens Telcom Report No. 2, pp. 123-127, 1979.

CCIR (International Radio Consultative Committee), "Encoding Parameters of Digital Television for Studios," Rec. 601, 1982.

CCITT (International Telephone and Telegraph Consultative Committee), "Error performance on an International Digital Connection forming part of an Integrated Services Digital Network," Rec. G.821, *Seventh Plenary Assembly Yellow Book*, Geneva, Switzerland, 1980.

W. R. Daumer, "Subjective Evaluation of Several Efficient Speech Coders," IEEE Trans. on Communications, pp. 655-662, April 1982.

M. Decina, "Managing ISDN through International Standards Activities," IEEE Communications Magazine, pp. 19-25, September 1982.

J. L. Flanagan, *Speech Analysis, Synthesis and Perception*, Springer-Verlag, New York, 1972.

J. L. Flanagan, M. R. Schroeder, B. S. Atal, R. E. Crochiere, N. S. Jayant and J. M. Tribolet, "Speech Coding," IEEE Trans. on Communications, pp. 710-737, April 1979.

G. D. Forney, "Coding and Its Application in Space Communications", IEEE Spectrum, pp. 47-58, June 1970.

R. G. Gallager, *Information Theory and Reliable Communication*, McGraw-Hill, New York, 1965.

R. C. Gonzalez and P. Wintz, *Digital Image Processing*, Addison-Wesley, Reading, Mass., 1977.

IEEE Transactions on Communications, Special Issue on Bit Rate Reduction and Interpolation, M. R. Aaron and N. S. Jayant (eds.), April 1982.

N. S. Jayant (ed.), *Waveform Quantization and Coding*, IEEE Press, New York, 1976.

N. S. Jayant, "Voice and Image Techniques," unpublished work, Bell Laboratories, 1981.

V. I. Johannes, "Performance Parameters for Digital Communications," Proc. IEEE, p. 539, April 1983.

R. W. Lucky, J. Salz and E. J. Weldon, Jr., *Principles of Data Communication*, McGraw-Hill, New York, 1968.

Members of Technical Staff, Bell Laboratories, *Transmission Systems for Communications*, Western Electric Company, Technical Publications, Winston-Salem, N.C., 1971.

A. N. Netravali and J. O. Limb, "Picture Coding: A Review," Proc. IEEE, pp. 366-406, March 1980.

J. D. Oetting, "A Comparison of Modulation Techniques for Digital Radio," IEEE Trans. on Communications, pp. 1752-1762, December 1979.

W. K. Pratt, *Digital Image Processing*, John Wiley, New York, 1978.

L. R. Rabiner and R. W. Schafer, *Digital Processing of Speech Signals*, Prentice-Hall, Englewood Cliffs, N.J., 1978.

A. Rosenfeld and A. C. Kak, *Digital Picture Processing*, Academic Press, New York, 1976.

J. F. Schouten, "Nachricht und Signal," Nachrichtentechnische Fachberichte, Bd. 6, pp. 1-2, 1957.

M. Schwartz, *Information Transmission, Modulation and Noise*, McGraw-Hill, New York, 1980.

K. S. Shanmugam, *Digital and Analog Communication Systems*, John Wiley, New York, 1979.

C. E. Shannon, "A Mathematical Theory of Communication," Bell System Tech. J., pp. 379-423 (Part I) and pp. 623-656 (Part II), 1948.

D. Slepian (ed.), *Key Papers in the Development of Information Theory*, IEEE Press, New York, 1973.

A. J. Viterbi and J. K. Omura, *Principles of Digital Communication and Coding*, McGraw-Hill, New York, 1979.

A. D. Wyner, "Fundamental Limits in Information Theory," Proc. IEEE, pp. 239-251, February 1981.

2

Waveform Characterization

2.1 Introduction

The topic of this book is the efficient transmission of waveforms over digital channels. These waveforms will typically be analog (time- and amplitude-continuous) and therefore time discretization and amplitude quantization are necessary prior to transmission.

Time discretization requires very little deliberation in that most waveform coding methods of interest use only one basic method for doing it, viz., uniform time-sampling. A very fundamental result in digital waveform coding is that a *bandlimited or bandpass signal* can be completely recovered from its samples provided that the sampling frequency is at least two times the highest frequency contained in the waveform (Chapter 3). In fact, the minimum sampling frequency will be two times the bandwidth if the lowest frequency is an integral multiple of the bandwidth. This includes the special case where the lowest frequency is zero, the case of a *lowpass* signal. The use of uniform time-sampling is so well-established in theory and practice that digital signal processing treatises tacitly assume the availability of time-discrete waveform samples. We shall do the same. In anticipation of sampling, *all waveforms in this book will be assumed to be bandlimited.* Also, in this book, the term *time discretization* will be used in a general sense that includes *space discretization* for two-dimensional image signals.

Efficient *amplitude quantization*, unlike time discretization, is more waveform-specific. For example, the theory of zero-memory quantizers, as discussed in Chapter 4, is related to the *probability density function* (pdf) for the amplitudes of input samples; while the design and performance of predictive coders (Chapters 6

0042

to 10) and transform coders (Chapters 11 and 12) depend directly on the correlations amongst the amplitudes of waveform samples. A good part of Chapter 2 will therefore be devoted to characterizing the so-called univariate and multivariate distributions of waveform amplitudes; the latter, in particular, will characterize the *predictability* of waveforms as described by the *autocorrelation function* (acf), *power spectral density* (psd), and also by the *spectral flatness measure* (*sfm*), a compact description of waveform predictability.

The study of distributions implies a modeling of waveforms or waveform samples by random processes [Dunn and White, 1940] [Kretzmer, 1952]. This kind of modeling will be discussed in Section 2.2. The *model sources* of Section 2.4 will be random processes with special acf and psd functions that resemble the real-signal properties described in Section 2.2. In coding literature, deterministic waveforms usually play the role of test signals that describe real waveforms in an average or partial sense; for example, speech waveforms are sometimes modeled by a 800 Hz sinusoid, and edge transitions in an image waveform are often modeled by step functions. Deterministic signals, however, have zero information content, and for communications signals such as speech and images, random process models, if affordable, are definitely more appropriate.

Chapter 2 will begin with a non-mathematical review of speech and video waveform properties in Section 2.2, with emphasis on pdf, acf and psd descriptions. Section 2.3 will provide a summary of random process mathematics, formalizing notions of pdf, acf and psd. Section 2.4 will describe the model sources with special acf and psd functions. These sources will be invoked repeatedly in later chapters.

The information content of random waveforms can be quantified in terms of the *rate distortion function* $R(D)$ or its inverse, the *distortion rate function* $D(R)$ (Appendix D). The function $R(D)$ specifies the amount of information R, in bits per sample, that one will have to expend for communicating the waveform with a specified fidelity, as measured by a distortion D. Numerical results exist only for a few tractable waveform descriptions, and one result, involving the spectral flatness measure, has already been mentioned in Chapter 1. Rate distortion bounds provide a useful calibration for the performance of specific quantizers and coders for real waveforms. However, much of the analytical development in this book, while depending heavily on pdf, acf and psd models, will be independent of $R(D)$ concepts per se. This stand recognizes the fact that perceptually adequate $R(D)$ models are untractable, if not impossible, for speech and video signals.

In developing the formalism of Chapter 2, we shall assume that the reader is familiar with the basic concepts of random processes and linear system theory. Our mathematical material should be regarded as an attempt to make our discussions self-contained, and not as an alternative to existing standard literature.

2.2 Characteristics of Speech and Image Waveforms

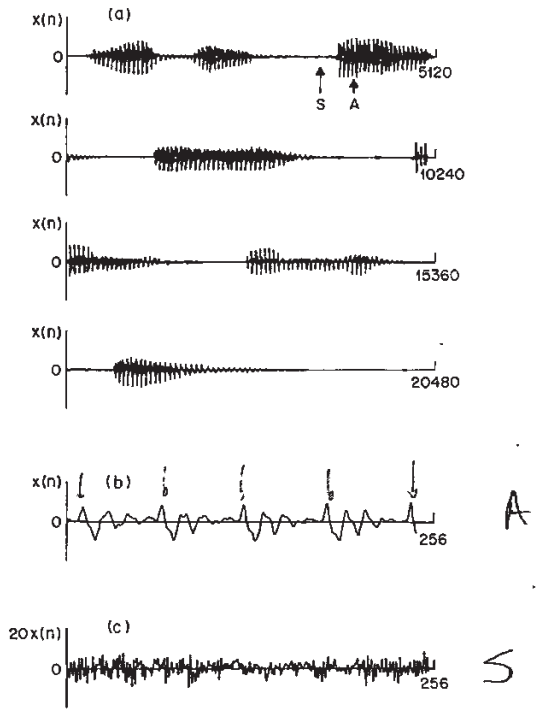
2.2.1 Amplitude versus Time Plots

The simplest description of any waveform is an amplitude-versus-time plot that typifies it. Examples appear in Figures 2.1 through 2.3. They refer respectively to

speech, two-level graphics and a multi-level, or gray-level image. These plots qualitatively indicate many significant waveform features, some of which are quantified in subsequent sections. These features include (a) the relative probabilities of different amplitudes, (b) the extent of sample-to-sample correlations and (c) the existence of macroscopic structure such as quasi-periodicity and non-stationarity. Some of the above descriptions have universal value and some, for example (c), may be signal-specific. Common to all the waveforms, however, is the important property [related to (b) above] that the waveform is appropriately bandlimited, as described by the values of W in Table 1.1 for speech and image examples.

Figure 2.1 shows a sentence-length speech waveform (a) (result of a male utterance "every salt breeze comes from the sea"), together with magnified waveforms (b) and (c) for the "A" and "S" segments in the word "salt". Notice that the low-frequency voiced waveform of "A" is more slowly varying than the high-frequency unvoiced fricative waveform of "S". Notice also that the voiced waveform is *quasi-periodic*, with slightly over four "periods" in 32 ms; this

Figure 2.1 Amplitude waveforms of speech for (a) a sentence-length (2.5 seconds) segment; (b) a short-time (32 ms) segment of voiced speech; and (c) a short-time (32 ms) segment of unvoiced speech [Flanagan et al.;©1979, IEEE].



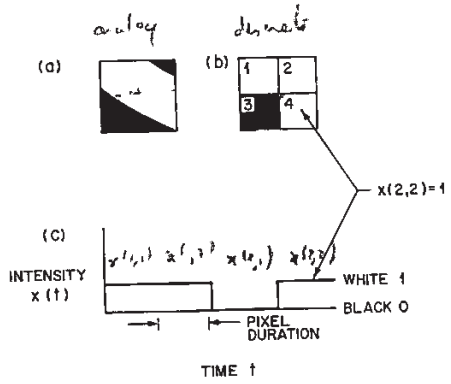


Figure 2.2 (a) Two-level image; (b) division of the image into pixels; and (c) intensity plot obtained by a systematic scan of the rows of (b). The amplitude levels 1 and 0, associated with white and black intensities, are sometimes rescaled to be +1 and −1, for analytical reasons.

corresponds to a *fundamental frequency* or *pitch frequency* that is somewhat less than $(8.0 \text{ ms})^{-1} = 125$ Hz in our example; the unvoiced waveform, on the other hand, is more noise-like. Also, its energy level is much lower than that of the voiced segment. Clearly, the speech waveform is extremely nonstationary, with significant inter-segment differences both in terms of amplitude level and spectral (frequency) content. Descriptions of locally stationary speech segments (of period in the order of 32 ms) will be formally termed *short-time* or *short-term* descriptions, while *long-time* or *long-term* descriptions will refer to speech waveform durations that are several orders of magnitude greater.

Figure 2.2(a) shows a tiny segment of a graphics image with two levels of intensity, *white* and *black*. The two black areas may be magnified versions, for example, of two nearly parallel black line segments on a weather-map. Figure 2.2(b) is a time-discrete, or space-discrete, representation of the sub-image in Figure 2.2(a) as a 2×2 array consisting of four picture elements, usually termed *pixels*, or *pels*. Note that only one of these pixels is black. The other three are white, suggesting predominantly white areas in corresponding portions (quarters) of the sub-image (a). For example, the pixel in row 2, column 2 is white. Using two-dimensional notation, and the convention that white = 1 and black = 0, $x(2,2) = 1$. The general intensity value is $x(m,n)$ for pixel location (m,n). Figure 2.2(c) shows an intensity-versus-time waveform obtained by scanning the four pixels — a left-to-right scan of the first row, followed by a left-to-right scan of the second row. The term *image waveform* will refer in this book to an *intensity* plot such as that in Figure 2.2(c), and pixel samples will have appropriate values of sample index and sample intensity. For example, in the present illustration, $x(4) = x(2,2) = 1$, while $x(3) = x(2,1) = 0$. In some analyses, it may be convenient to regard pixel amplitudes as zero-mean random variables, falling in the range -1 to $+1$. This can be realized by a simple rescaling and mean subtraction process. Thus, in the above example, we might have $x(4) = 2x(2,2) - 1 = 1$, and $x(3) = 2x(2,1) - 1 = -1$, instead of the values 1 and 0 cited earlier.

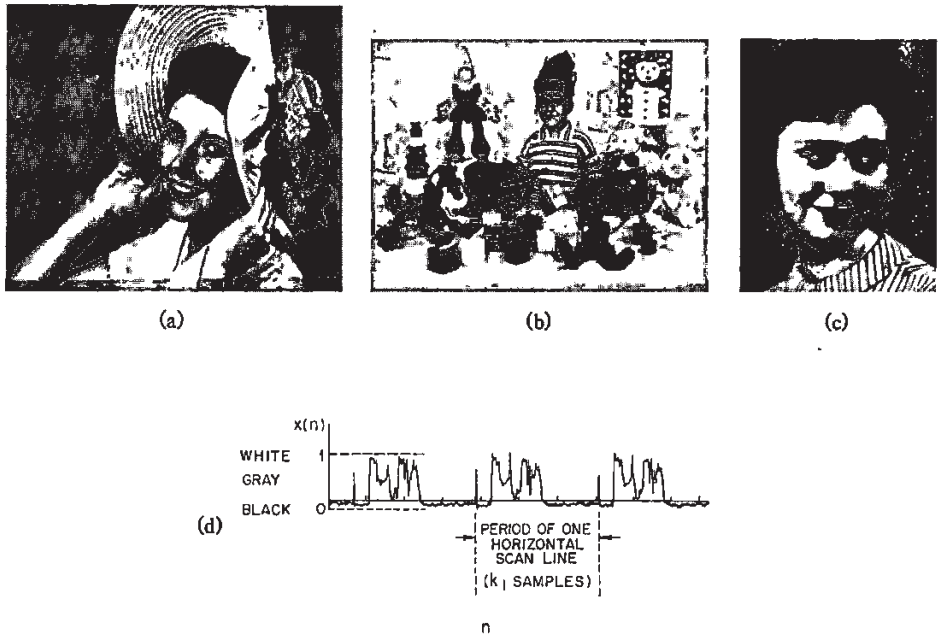
0045

The two-level plot in Figure 2.2(c) is a very simple example; it will be discussed again in Chapter 10. More generally, we will be interested in image waveforms with many levels of intensity; gray-level monochrome images, as well as color images, belong to this general class.

Figure 2.3 shows three examples (a) (b) and (c) of multi-level monochrome images that include shades of *gray* between the white and black extremes. Such images are also called *gray-level* images. The image in (a) will be used for several future illustrations in this book.

In gray-level images, the function $x(m,n)$ takes on values in the continuous range (0,1). Figure 2.3(d) is the intensity plot corresponding to a three-line row-by-row scan of part of the scene "Karen" in (c). Notice the bursts of high-intensity pixels for which $x(m,n) \rightarrow 1$, or $x(n) \rightarrow 1$ in the one-dimensional plot. This is the result of scanning an image segment with a predominance of white areas (the face of Karen). Notice also the presence of sharp intensity transitions. Such rapid transitions are typical of many classes of image waveforms. Notice, moreover, that as in the voiced waveform of Figure 2.1, the waveform of Figure 2.3 is quasi-periodic, with the "period" corresponding to the duration of one (horizontal) scan line. A second source of quasi-periodicity exists in moving pictures. This is due to pixel patterns that tend to repeat in successive *frames* in a moving image. Unlike the fluctuating pitch period in speech, the scan-line and frame periods in image

Figure 2.3 Gray-level images (a), (b) and (c); and (d) intensity waveform corresponding to three consecutive horizontal scans of the face in (c). The image (a) is used for several future illustrations in this book.



systems are constant numbers. Correlations and periodicities in image inputs are therefore more structured than in speech.

The speech waveform in Figure 2.1 is *one-dimensional* (1-D), or *scalar*; images may be regarded as *two-dimensional* (2-D) signals [Figure 2.2(b)], although the post-scanning intensity waveforms in Figures 2.2(c) and 2.3(b) are shown as being one-dimensional. Good examples of *multi-dimensional* waveforms are *vector* waveforms that occur in Adaptive Prediction, Transform Coding and Sub-band Coding systems. In Adaptive Prediction (Chapter 6) and Transform Coding (Chapter 12), the components of the vector signal are, respectively, predictor coefficients and transform coefficients of different order; in sub-band coding of speech (Chapter 11), the components correspond to the outputs from contiguous speech frequency bands.

2.2.2 Probability Density Functions

Consider a finite-length sequence $\{x(n)\}$; $n = 0,1,...,N-1$ obtained by observing a finite segment of a sampled continuous waveform. Very simple, but not complete, descriptions of the waveform are provided by the quantities

$$\mu_x = \frac{1}{N}\sum_{n=0}^{N-1} x(n); \qquad \chi_x^2 = \frac{1}{N}\sum_{n=0}^{N-1} x^2(n); \qquad \sigma_x^2 = \frac{1}{N}\sum_{n=0}^{N-1} [x(n)-\mu_x]^2 \quad (2.1)$$

called respectively the *mean value*, *mean square value (or average power)* and *variance* of that segment. The quantity σ_x, the square root of variance, is called the *standard deviation*. If the mean μ_x is known a priori, the third term in (2.1) is a so-called *unbiased* estimate of variance. When the mean is estimated by the first term in (2.1), an unbiased estimate of variance is obtained by replacing $1/N$ in the third term of (2.1) by $1/(N-1)$.

An interesting special case, one that is approximated in speech waveforms for large N, is that of a *zero-mean* sequence:

$$\mu_x = 0; \qquad \chi_x^2 = \sigma_x^2 \quad (2.2)$$

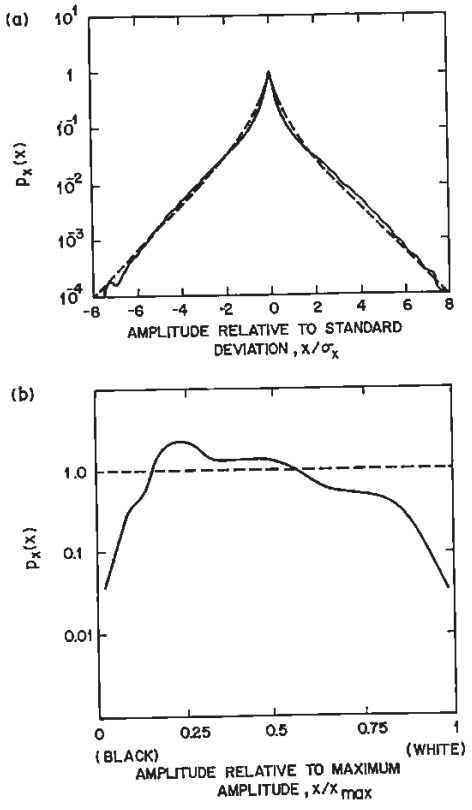
A more informative description is provided by a *probability density function* (pdf). Consider an amplitude interval $\mathscr{I}_k$ defined by an amplitude range $\mathscr{I}_k: (x_k \leqslant x < x_{k+1})$. Out of the N samples in sequence $\{x(n)\}$, let n_k amplitudes fall in interval $\mathscr{I}_k$. A plot of the normalized count n_k/N shows the shape of the (non-negative) pdf $p_x(x)$, which is really defined as a continuous function whose area in interval $\mathscr{I}_k$ equals n_k/N. The pdf $p_x(x)$ specifies not only the average values (2.1) and (2.2), but also indicates other waveform features such as boundedness and peakiness.

The quantities μ_x, σ_x^2 and $p_x(x)$ above are *short-term* or *segment-specific* descriptions obtainable by indicated measurements with an observed waveform of length N samples. They are also useful as *estimates* of corresponding random process characteristics if the input waveform is *stationary* and *ergodic* and N is large; ergodicity implies that ensemble statistics can be equated to those of one sample sequence.

Experimental pdf Examples. The pdf of *speech* amplitudes depends on input bandwidth and recording conditions. With a high-quality microphone and telephone bandwidth (200 to 3200 Hz), a first approximation to the *long-term* pdf or *long-time-averaged* pdf ($N \rightarrow \infty$) is provided by a *two-sided exponential* or *Laplacian* model; a better fit is provided by the so-called *gamma* pdf as shown by dashed lines in Figure 2.4(a). The experimental speech pdf in the figure is based on measurements over 55 seconds of speech, spoken by five speakers [Zelinski and Noll, 1974]. Note the distinct peak at zero in the long-term pdf; this is a result of pauses and low-level speech segments. Note also the significant probability of high amplitudes such as $\pm 4\sigma_x$. *Short-time* pdf's of speech segments are also unimodal or single-peaked; but they are better described by a bell-shaped *Gaussian* pdf, irrespective of whether the speech segments are voiced or unvoiced.

An experimental long-time-averaged pdf of *video* amplitudes is illustrated in Figure 2.4(b) [Kummerow, 1972]. It is based on four image frames that include the low- and high-detail scenes "lady in a straw hat" and "playboy" in Figures

Figure 2.4 Typical long-time-averaged probability density functions for (a) speech amplitudes [Zelinski and Noll, 1974] and (b) image amplitudes [Kummerow, 1972].



0048

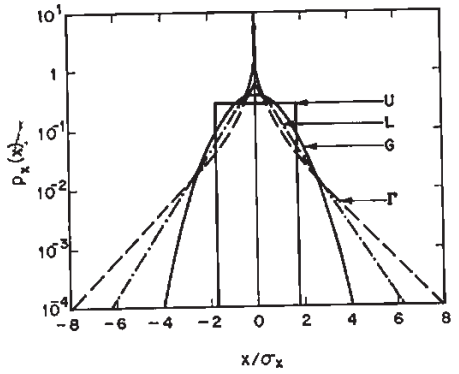
2.3(a) and (b); these examples will also be used in Section 2.2.3. Unlike the speech pdf, the video pdf is not unimodal. The average pdf of image material is crudely modeled (as shown by dashed lines) by a *uniform* pdf, where all luminances from black to white are equiprobable.

Waveforms are sometimes represented in terms of *adjacent-sample differences*. The pdf of the difference signal $x(n)-x(n-1)$ for an image waveform follows the *two-sided exponential* or *Laplacian* function. Neighboring pixels are very similar in amplitude since the signal is slowly changing. The difference pdf therefore has a significant peak at difference values near zero, corresponding to the situation $x(n) \sim x(n-1)$ (Problem 2.2). The pdf of the adjacent-sample difference in speech is qualitatively similar to the pdf of individual amplitudes themselves, and is again modeled best by the gamma distribution. The Laplacian and gamma models for adjacent-sample differences in image and speech waveforms are also applicable, respectively, to so-called linear prediction error signals in image and speech coding (Chapter 6).

Also significant for image coding is the pdf of the frame-to-frame differences of a given pixel. This pdf is also symmetrical and peaked at zero, but the actual shape is motion-dependent; increased motion results in increased probability content in the pdf tails, while smaller tails indicate greater interframe redundancy.

Model pdf's. Figure 2.5 and Table 2.1 are descriptions of the four model pdf's mentioned so far. For simplicity, we have set $\mu_x = 0$ in all cases, including the uniform pdf, although Figure 2.4(b) shows a positive mean; further, in Figure 2.5, all the pdf's have a variance $\sigma_x^2 = 1$. The four models will be invoked frequently in this book with the notations **U, G, L** and **Γ** used in Table 2.1. The *finite-support* uniform pdf **U** has been mentioned already as a model for image amplitudes. It will also be used as a model for quantization noise distribution in Chapter 4. The Gaussian pdf **G** in Table 2.1 will also be referred to sometimes as the zero-mean normal pdf $N(0,\sigma_x^2)$. The pdf **Γ** used in this book is a special case of a wider class of gamma pdf's [Richards, 1973]. A property that is common to the *infinite-support* models **G, L** and **Γ** is the fact that $p_x(x)$ monotonically *decreases* with $|x|$;

Figure 2.5 Four model pdf's with a mean value of zero [Zelinski and Noll, 1974].



0049

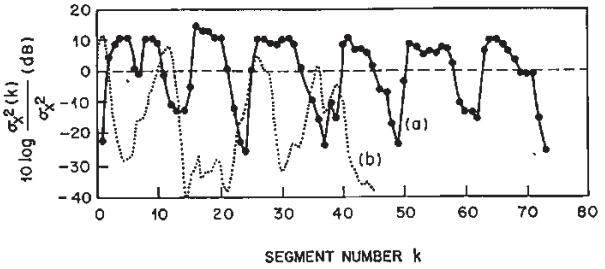
Table 2.1 Four model pdf's with a mean value of zero.

Name	Notation	$p_x(x)$
Uniform or Rectangular	U $(\sigma_x^2 = \Delta^2/12)$	$\frac{1}{\Delta}$; $x \in (-\frac{\Delta}{2}, \frac{\Delta}{2})$ 0; *otherwise*
Gaussian or Normal	G $N(0,\sigma_x^2)$	$\frac{1}{\sqrt{2\pi\sigma_x^2}}\exp\left[-x^2/2\sigma_x^2\right]$
Laplacian or Two-sided Exponential	L	$\frac{1}{\sqrt{2}\sigma_x}\exp\left[-\sqrt{2}\lvert x\rvert/\sigma_x\right]$
Gamma	Γ	$\frac{\sqrt[4]{3}}{\sqrt{8\pi\sigma_x\lvert x\rvert}}\exp\left[-\sqrt{3}\lvert x\rvert/2\sigma_x\right]$

significant differences in pdf shape exist, of course, both at $|x| = 0$ and at the tails, $|x| \to \infty$. A model where $p_x(x)$ *increases* monotonically with x is the *arcsine* pdf (Problem 2.1), which describes the distribution of sine-wave amplitudes. This pdf is of interest because a sine-wave input serves as a standard test signal for coding algorithms (Section 5.3).

Nonstationarity of Speech Signals. The distinction between short-time and long-time pdf in speech is related to the fact that speech waveforms are nonstationary. Figure 2.6 quantifies one aspect of this nonstationarity. It shows the time-dependence of segment variance $\sigma_x^2(k)$ (2.1) measured over contiguous 32 ms segments of the speech of a single speaker. The number of speech segments used is

Figure 2.6 Time dependence of short-term speech variance for (a) the 2.3 second waveform of Figure 2.1(a) [Jayant, 1982], and for (b) a different speech waveform, of duration 1.4 seconds [Noll, 1974]. The segment length is 32 ms.



0050

73 in example (a) and 44 in example (b). The 0 dB line corresponds to the average σ_x^2 of the 73 values of $\sigma_x^2(k)$ in example (a). Note that the dynamic range of short-time variance can exceed 40 dB over a time duration of no more than 1 second. Further, if one considers the variation of the average of short-term variance as a function of speaker, the resulting across-speaker distribution has an additional dynamic range of about 20 dB [Richards, 1973]. The above level variations are sometimes compounded by further level uncertainties in communication, resulting from ambient conditions and telephone connection efficiency. On the other hand, the 40 dB and 20 dB dynamic ranges mentioned above are not always additive (which would imply a dynamic range of 60 dB!). Also, when high-frequency pre-emphasis is present in telephone systems, the effect is to decrease the dynamic range of the speech signal significantly. Sustained performance over an input dynamic range in the order of 30 to 40 dB is therefore a well-accepted design criterion for coders and quantizers in high-quality speech communication.

2.2.3 Autocorrelation Functions

Successive sample amplitudes $x(n)$ and $x(n-1)$ of a slowly varying waveform are said to be highly correlated. As a result, the adjacent-sample difference (in general, a prediction error signal) has a much smaller variance than the original signal $x(n)$. This fact is exploited in redundancy-removing differential PCM (DPCM) coders for both speech and video.

A fundamental waveform property, therefore, is an *autocorrelation function* (acf) which quantifies the closeness of the amplitudes of two samples as a function of their time- or space-separation; it is a function that tells us how close, or similar, samples $x(n)$ and $x(n+k)$ are, on the average:

$$R_{xx}(k) = \frac{1}{N} \sum_{n=0}^{N-|k|-1} x(n)x(n+|k|) \tag{2.3}$$

The acf $R_{xx}(k)$, as a function of k, is a deterministic descriptor of a waveform which itself may best be modeled by a random sequence. The use of $|k|$ in (2.3) makes $R_{xx}(\cdot)$ symmetric about $k = 0$; while the summation to $N - |k| - 1$ recognizes the fact that only $N - |k|$ pairs are available for the product operation if $\{x(n)\}$ is an N-sample sequence defined for $n = 0,1,...,N-1$. If $R_{xx}(k)$ is used as an estimate of correlation, it can be shown [Jenkins and Watts, 1968] that (2.3) provides a better result than a formula that uses $N - |k|$ instead of N in the denominator, especially if k is close to N. An *autocovariance function* results if the mean value μ_x is subtracted from the terms in the product of (2.3), but the distinction disappears for $\mu_x = 0$. For *zero-mean* sequences, it is clear from (2.1) that

$$R_{xx}(0) = \sigma_x^2 \tag{2.4}$$

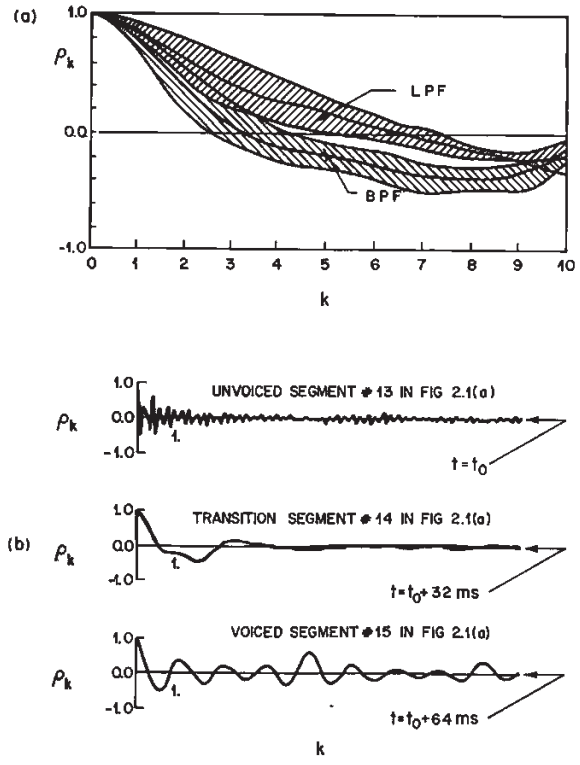
A *variance-normalized acf* is defined in the form

$$\rho_k = \rho_{xx}(k) = R_{xx}(k)/R_{xx}(0); \qquad \rho_0 = \rho_{xx}(0) = 1 \tag{2.5}$$

The quantity ρ_k has values in the range $(-1, +1)$. A value of $\rho_k = 1$ implies perfect positive correlation such as that between any two samples of a constant waveform. A value of $\rho_k = -1$ suggests perfect negative correlation such as that between any pair of adjacent samples of a sinusoid sampled only at its (positive and negative) peaks. A value of $\rho_k = 0$ implies total decorrelation as between samples of a totally random white noise process. The *adjacent sample correlation* $\rho_1 = \rho_{xx}(1)$ is a very useful parameter. It corresponds to a sample separation of $k = 1$. This signifies a separation of one Nyquist interval, except in oversampled systems.

Figure 2.7 illustrates a *long-time-averaged* acf based on measurements [Noll, 1972] with a 55-second speech sample that is either lowpass-filtered (LPF: 0 to 3400 Hz) or bandpass-filtered (BPF: 300 to 3400 Hz). The sampling frequency is 8 kHz. In each set of curves, the upper and lower limits represent acf maxima and minima, respectively, over four speakers (two male and two female), and the middle curve gives the mean acf value (average over four speakers). Notice that in

Figure 2.7 (a) Long-time-averaged acf plots [Noll, 1972] for LPF and BPF speech waveforms. (b) Evolution of the short-term speech acf with time for the example of three consecutive 32 ms segments in Figure 2.1 [Jayant, 1982].



general ρ_1 has a value close to +0.9, and that the ρ_k crosses zero at a higher value of k for LPF speech than for BPF speech. The increased evidence of lower frequencies in LPF speech causes higher correlations for a given separation k.

Table 2.2 lists four sets of ρ_k values for 8 kHz-sampled speech, for $k = 0$ to 5. These results are taken from different sources in speech literature, including the source used in Figure 2.7(a).

Table 2.2 Values of acf for 8 kHz-sampled speech.

Speech Source	LPF Source Average from Figure 2.7(a) [Noll, 1972]	BPF Source Average from Figure 2.7(a) [Noll, 1972]	LPF Source [McDonald, 1966]	BPF Source with High-Frequency Emphasis [Grizmala, 1972]
ρ_0	1.00	1.00	1.0000	1.000
ρ_1	0.86	0.79	0.8644	0.825
ρ_2	0.64	0.38	0.5570	0.562
ρ_3	0.40	0.05	0.2274	0.308
ρ_4	0.26	−0.08	−0.0297	0.004
ρ_5	0.20	−0.16	−0.1939	−0.243

The acf plots in Figure 2.7(b) are measured over *short segments* (32 ms; N=256) of a single speech waveform, and successive acf plots in Figure 2.7(b) are results of acf recomputation once every 32 ms. The evolution in Figure 2.7(b) is from (i) an aperiodic acf (for an aperiodic fricative segment), to (iii) a periodic acf (for a periodic voiced segment); the intermediate case (ii) is a fast-decaying transitional acf (indicating a non-steady-state segment during unvoiced-to-voiced transition). The acf plots of Figure 2.7(b) follow (2.3), (2.4) and (2.5). In periodic speech, the envelope of $|\rho_{xx}(k)|$ decreases monotonically with k until k equals the fundamental period, at which point a second peak occurs in $\rho_{xx}(k)$, similar to that at $\rho_{xx}(0) = 1$.

Intraframe acf plots for 256×256 monochrome images are shown in Figure 2.8(a). The plots are those of $\rho_{xx}(k_h, k_v)$ as a function of horizontal separation k_h and vertical separation k_v. The acf is seen to decay at a similar rate in both horizontal and vertical directions for the images considered. There is, however, a significant difference in the acf values of the *low-detail* and *high-detail* examples, for a given (k_h, k_v). As expected, acf values are generally higher for the low-detail image.

Table 2.3 compiles experimental intraframe acf values for two 512×512 images, differing once again in image detail.

The presence of significant intraframe correlations, or redundancies, carries over to the case of two-level graphics images, and this can be demonstrated by acf functions that are nearly exponential in both vertical and horizontal directions. We shall note in Chapter 10 that an even more useful description of redundancy in two-level graphics is in terms of so-called *run-length* or cluster-length distributions.

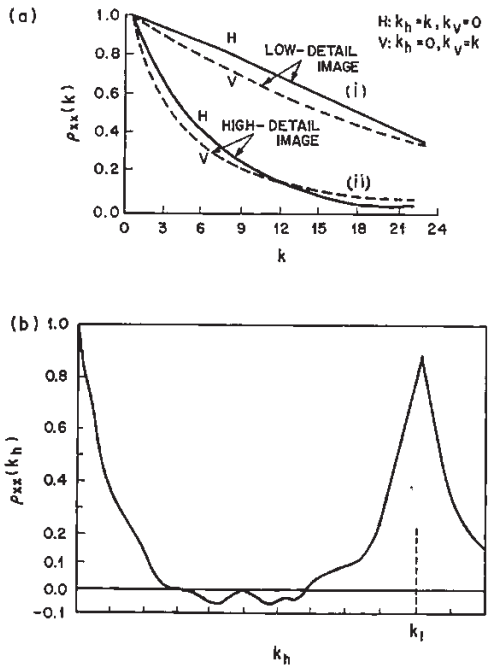


Figure 2.8 (a) Intraframe acf plots for gray-level images with (i) low detail (a face) and (ii) high detail (a crowd) [Habibi and Wintz;©1971, IEEE]. (b) An acf plot that includes values of horizontal spacing greater than the number of pixels/scan line. The input used in (b) is the image "Karen" in Figure 2.3 [O'Neal,1966,II; ©AT&T].

Table 2.3 Experimental values of intraframe acf for a low-detail image (Lady in straw hat) and for a high-detail image (Playboy). Values for the high-detail image are in parentheses. The parameters k_h and k_v refer to horizontal and vertical spacing (in pixels).

k_v \ k_h	0	1	2
0	1.000(1.000)	0.967(0.956)	0.926(0.892)
1	0.970(0.904)	0.954(0.884)	0.922(0.844)
2	0.934(0.816)	0.928(0.805)	0.907(0.783)

The Case of Large k. We have already mentioned that in voiced speech, the envelope of $|\rho_{xx}(k)|$ decreases monotonically with k at first; but when k is large enough to be equal to the pitch period k_p, a second peak appears in the acf function, after which the short-term acf pattern tends to repeat the behavior that characterized its first k_p values. We shall discuss this again in Chapter 6 in the

context of acf-based pitch extraction. What is worth noting here is that the pitch period k_p is a *time-varying* quantity.

Figure 2.8(b) is a one-dimensional intraframe acf $\rho_{xx}(k_h)$ display for an image that includes values of horizontal spacing k_h up to and greater than k_l, the number of pixels per scan *line*. A horizontal separation of $k_h = k_l$ corresponds to a vertical separation of $k_v = 1$. The acf plot therefore has a secondary peak at $k_h = k_l$. Unlike in speech, this secondary peak always occurs at a *time-invariant* lag k_l that corresponds to the scan-line length.

Interframe acf's in video are motion- and activity-dependent. An average characteristic would have approximately an exponential shape for small values of *frame* separation k_f. The interframe acf decays from a typical value in the order of 0.85 to 0.90 for $k_f = 1$ (temporal spacing of 1/30 s) to a value in the order of 0.30 for $k_f = 10$ (temporal spacing of 1/3 s).

2.2.4 Power Spectral Density Functions

The concept of a *power spectral density* function (psd), which shows the power in a waveform as a function of frequency, is closely related to the acf concept discussed in Section 2.2.3. In Section 2.3 we shall note that for stationary signals, the psd and acf functions indeed form a *Fourier Transform* pair.

The psd is a non-negative quantity that can be measured by using an appropriate filter bank and measuring the average power at the output of each filter. If the waveform is available as a finite-length sequence $\{x(n)\}$; $n = 0,1,...,N-1$, various other techniques exist to estimate its psd. The psd $S_{xx}(e^{j\omega})$ is defined such that if (ω_1,ω_2) is an interval on the radian frequency axis, the average power $\chi_x^2(\omega_1,\omega_2)$ in that interval is

$$\chi_x^2(\omega_1,\omega_2) = \frac{1}{\pi}\int_{\omega_1}^{\omega_2} S_{xx}(e^{j\omega})\,d\omega\,; \quad \omega_1, \omega_2 \geq 0\,; \quad \chi_x^2(-\pi, \pi) = \chi_x^2 \tag{2.6}$$

The definition implies a non-negative psd.

For a time-discrete waveform, the *periodogram*

$$P_{xx}(e^{j\omega}) = \frac{1}{N}\,|X(e^{j\omega})|^2 \tag{2.7}$$

may also serve as an estimate of $S_{xx}(e^{j\omega})$, with $X(e^{j\omega})$ signifying the Discrete Fourier Transform (DFT) of $\{x(n)\}$. A better estimate results if an N-sequence is divided into M non-overlapping subsequences, and corresponding periodograms are averaged. A totally different procedure for estimating $S_{xx}(e^{j\omega})$ is the so-called *autoregressive estimation* technique. This will be discussed in Chapter 6.

Figure 2.9(a) shows a *long-time-averaged* psd for two ensembles of speakers, together with the psd of the speech waveform of Figure 2.1(a). All the three plots show a *lowpass* characteristic in the frequency range above 500 Hz, signifying a preponderance in speech of low frequency segments such as sustained vowels. It is important to note, however, that high frequencies in speech waveforms, although

0055

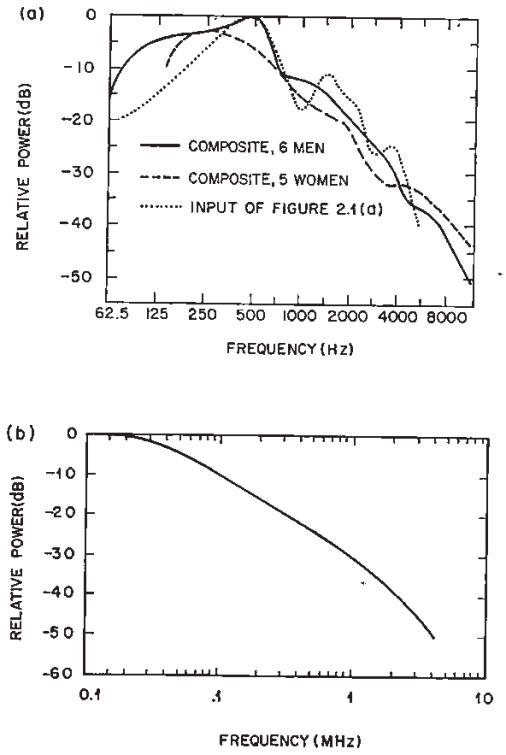


Figure 2.9 Long-time-averaged psd for (a) speech [Dunn and White, 1940] [Flanagan, 1972. Reprinted with permission] [Jayant, 1982] and (b) image waveforms [Connor, Brainard and Limb;©1972, IEEE].

having relatively less power, provide important information in the form of fricatives, stop sounds and unvoiced consonants.

A *long-time-averaged* psd for images is shown in Figure 2.9(b). The *lowpass* function is relatively flat for frequencies up to at least twice the line rate of about 15 kHz (corresponding to 525 lines in 1/30 s); it begins to fall at about 6 dB/octave in the middle region of the spectrum, and at about 12 dB/octave at very high frequencies. The prepondering low frequencies refer to the large area of constant brightness levels in an image frame. The power at high frequencies is comparatively small, but it provides the information about sharp changes in luminances.

Figure 2.10 shows *short-time-averaged* psd plots corresponding to the voiced and unvoiced segments "A" and "S" in Figure 2.1. In each case is shown a psd with low frequency resolution [Figures (a)(i) and (b)(i), obtained from a 64-point DFT analysis]; and a psd with high frequency resolution [Figures (a)(ii) and (b)(ii), obtained from a 256-point DFT analysis]. Note that while the short-time psd of "A" is *lowpass* as in the average plot in Figure 2.9(a), the psd for the fricative "S" is nearly flat, and in fact somewhat *highpass*, as expected. High-pass psd's also

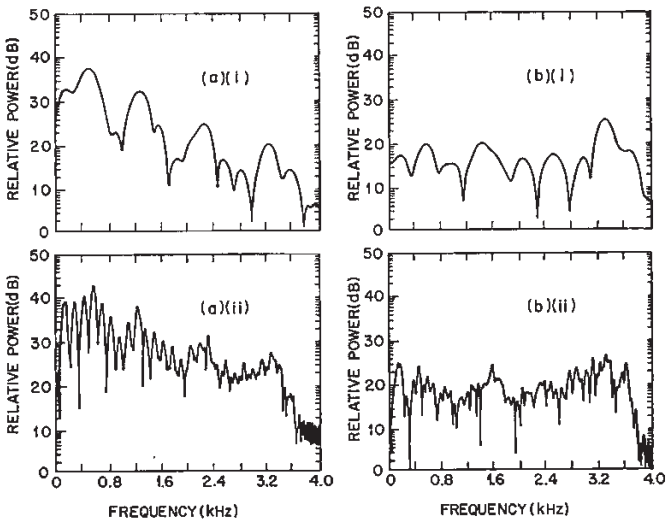


Figure 2.10 Short-time psd for the (a) voiced and (b) unvoiced speech examples in parts (b) and (c) of Figure 2.1 [Jayant, 1982]. In each case, the two psd's shown are results of analysis with (i) a low resolution in frequency [64-point DFT (Chapter 12 and Appendix A)] and (ii) a high resolution in frequency [256-point DFT].

occur in certain kinds of music material, for example, popular guitar. Moreover, in music, several lowpass and highpass sources (say, different instruments) can occur simultaneously at a given time, unlike in the speech of a single talker. As indicated earlier, lowpass and highpass segments in images are usually referred to as *low-detail* and *high-detail* segments.

Lowpass and highpass psd's are manifested, in the time domain, by different acf's — specifically by ρ_1 values that are positive and negative, respectively. The lowpass voiced speech psd (a) also exhibits four distinct resonances (*formants*) in the 0 to 4000 Hz band, corresponding to vocal tract resonances characteristic of the sound "A". The rapid oscillations in the voiced psd of part [(a)(ii)] of Figure 2.10 correspond to the fundamental frequency or pitch frequency, and they signify the quasi-periodic nature of voiced speech.

2.3 Mathematics of Random Waveforms

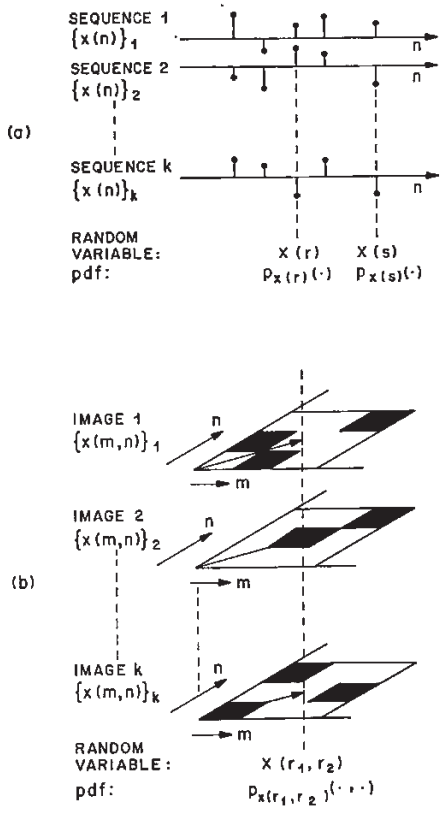
2.3.1 Random Processes

Although communication waveforms are not deterministic, their average properties (especially the long-time averages) are often regarded as deterministic and known a priori. Examples of such deterministic characterizations are those for long-time mean value and long-time acf. Further, the linear systems that communication waveforms are passed through during encoding and decoding

processes are also deterministic. We have therefore summarized in Appendix A some basic mathematics that applies to deterministic signals and systems. The mathematics of the present section will apply specifically to the characterization of discrete-time *random* or *stochastic* signals, specifically those that are *stationary* in a way that will be defined presently. This is a natural framework in which to describe communication waveforms for which the signal-generating rule is either not known or is so complex as to make precise signal description impossible.

Each sequence (or waveform) from a random process is considered to be a member of an ensemble of discrete-time signals characterized by a set of *probability distributions* or *probability density functions*. Let us suppose that $\{x(n)\}$, $n = 0, \pm 1, \pm 2, \ldots$ is a sequence of sampled values of a waveform. Its value at a specific time instant n is $x(n)$. The sequence can be interpreted as a *realization* or *sample sequence of a random process*. It is one of the strictly infinite number of possible sequences which form the ensemble and it is regarded as being chosen randomly by nature. At a given time instant n, the value $x(n)$ of the

Figure 2.11 Random processes in (a) one and (b) two dimensions [Rosenfield and Kak, 1976. Reprinted with permission].



0058

sequence $\{x(n)\}$ is a particular value of a *random variable (r.v.)* $X(n)$. The ordered set of r.v.'s $\{X(n)\}$ is the *random process* or *random sequence* [see Figure 2.11(a)]. At any time instant, the behavior of the random process can be described (at least conceptually) by a set of probability distributions or probability density functions. The above probability distributions are descriptors of multivariate statistics in general. We will therefore be able to characterize all the important properties of communication waveforms in the random process framework.

We can extend Figure 2.11(a) to the two-dimensional case, as in the simple example of Figure 2.11(b), which shows sample images with two levels of intensity, white and black. In characterizing two-dimensional images, we use the symbol $x(m,n)$ to denote an image element, or pixel intensity, at coordinates (m,n). The notation $\{x(m,n)\}$; $m,n=0,1,\ldots,N-1$ is used to define an evolving 2-D sequence. This sequence $\{x(m,n)\}$ is a realization of an ensemble $\{X(m,n)\}$ of possible images. In the 1-D case, $\{X(n)\}$ was a random process, a r.v. $X(n)$ evolving in time. In the 2-D case, $\{X(m,n)\}$ is a *random field*, a r.v. $X(m,n)$ evolving in two spatial directions. Similarly, in the case of moving image sequences, we could describe an image element $x\ (m,n,o)$, where o is the time index, as a sample of a 3-D random sequence or field $\{X(m,n,o)\}$; this is alternatively describable as a r.v. $X(m,n,o)$ evolving in three directions.

Most of the material in the rest of Section 2.3 will refer to one-dimensional (scalar) random processes. Extensions to higher-dimensional (vector) random processes (see Section 2.2.1) will be invoked in the book as and when necessary.

2.3.2 Univariate Distributions

The value $x(n)$ of a sample sequence $\{x(n)\}$ of a random process can be interpreted as a particular outcome of a random variable $X(n)$. The r.v. $X(n)$ is a real-valued quantity; we assume that it is sufficiently well behaved so that the probability of the event $\{X(n)\leqslant x\}$ is defined for all real values of the dummy variable x. The probability of this event is termed the *probability distribution function* of the r.v. $X(n)$ and will be denoted by

$$P_{x(n)}(x) = P\{X(n) \leqslant x\} \tag{2.8}$$

We now assume that the distribution function is continuous and define a *probability density function* (pdf) as the derivative of the distribution function:

$$p_{x(n)}(x) = d/dx\ P_{x(n)}(x) \tag{2.9}$$

In general, with a nonstationary process, the pdf is a function of absolute time n as in Figure 2.11. Let us, however, consider the case where the statistical properties of the random process are independent of time. The minimum requirement for this to hold is that the univariate distribution and density functions are independent of time *(stationarity of first order)*. We may then simplify the notation by omitting the time index n:

$$p_x(x) = d/dx\ P_x(x) \tag{2.10}$$

From (2.8) it follows that the distribution function is monotonically nondecreasing, and that $P_x(-\infty) = 0$ and $P_x(\infty) = 1$. The probability that the r.v. X is in the range (a,b) is

$$P\{a<X\leqslant b\} = P_x(b) - P_x(a) = \int_a^b p_x(x)dx \ , \tag{2.11}$$

the area of the shaded element in Figure 2.12(a). Also, by definition,

$$p_x(x) \geqslant 0 \ , \text{all } x \ ; \qquad \int_{-\infty}^{\infty} p_x(x)dx = 1 \tag{2.12}$$

Expectations of a r.v. Simple descriptions of a r.v. X are provided by *expectations*. Very generally, if $g(X)$ is a function of the r.v., the expectation of the new r.v. $g(X)$ is

$$E[g(X)] = \int_{-\infty}^{\infty} g(x)p_x(x)dx \tag{2.13a}$$

In particular, the *mean value* is

$$\mu_x = E[X] = \int_{-\infty}^{\infty} xp_x(x)dx \ , \tag{2.13b}$$

the center of gravity of the pdf of X. Also of interest are the expected values of the nth moment $g(X) = X^n$ and the nth central moment $g(X) = (X-\mu_x)^n$. With $n=2$, we obtain the *mean square value*

$$\chi_x^2 = E[X^2] = \int_{-\infty}^{\infty} x^2p_x(x)dx \tag{2.14a}$$

and the *variance*

$$\sigma_x^2 = E[(X-\mu_x)^2] = \int_{-\infty}^{\infty} (x-\mu_x)^2p_x(x)dx \tag{2.14b}$$

The variance is a measure of the spread of the distribution about the mean. The positive square root σ_x of the variance is called the *standard deviation* or *rms value*. From the last two equations, it follows that

$$\sigma_x^2 = \chi_x^2 - \mu_x^2 \tag{2.14c}$$

Note that μ_x, σ_x^2 and χ_x^2 can be estimated using N-sample sequences as in (2.1).

Example 2.1. Distribution function of the normal pdf

The *normal* probability distribution function is defined by

0060

$$P\{X \leqslant x\} = P_x(x) = \frac{1}{2}\left[1+\text{erf}\,\frac{x-\mu_x}{\sqrt{2}\sigma_x}\right] \; ; \quad \text{erf}\, x = \frac{2}{\sqrt{\pi}}\int_0^x e^{-t^2/2}dt \tag{2.15}$$

The *error function* erf x is widely tabulated [Thomas, 1969] [Abramowitz and Stegun, 1964], and used to compute the probability that a r.v. X falls into a given range. For example,

$$P\{(-4\sigma_x + \mu_x) \leqslant X \leqslant (4\sigma_x + \mu_x)\} = 2\,P_x(4\sigma_x + \mu_x) - 1 = \text{erf}(2\sqrt{2}) = 0.999936$$

is the probability that a normally distributed r.v. has a magnitude not exceeding $4\sigma_x$. The closeness of this probability to 1.0 is the rationale for the so-called *four-sigma loading* in quantizer design. •

In the case of *nonstationary* processes, all expectations will be functions of time index n. In most sections of this book, we shall assume *wide-sense stationarity* which requires only that the first two moments of the distribution are independent of time:

$$\mu_{x(n)} = \mu_{x(n+k)} \; ; \qquad \sigma^2_{x(n)} = \sigma^2_{x(n+k)} \tag{2.16}$$

Table 2.1 and Figure 2.5 have already provided a comparison of four model pdf's for amplitude-continuous sources. Another significant comparison of these models is in terms of two related pdf-specific properties, *differential entropy* and *entropy power*. These are discussed at length in Appendix C and Chapter 4.

The *characteristic function* $\Phi_x(j\,\Omega)$ of a random variable X is defined as

$$\Phi_x(j\,\Omega) = \int_{-\infty}^{\infty} p_x(x)e^{j\,\Omega x}dx \tag{2.17a}$$

the complex conjugate of the Fourier transform of the pdf $p_x(x)$. The inverse Fourier transform expresses $p_x(x)$ in terms of the characteristic function

$$p_x(x) = \frac{1}{2\pi}\int_{-\infty}^{\infty} \Phi_x(j\,\Omega)e^{-j\,\Omega x}d\,\Omega \tag{2.17b}$$

$\Phi_x(j\,\Omega)$ always exists because $p_x(x)$ is absolutely integrable in view of (2.12). Also, from (2.13), $\Phi_x(j\,\Omega)$ is identical with the expectation $E[e^{j\,\Omega X}]$.

Amplitude-discrete r.v.'s. An amplitude-discrete r.v. X can only assume one out of N possible values (outcomes) x_k; associated with each of these possible outcomes is a probability

$$P(x_k) = P\{X=x_k\} \tag{2.18a}$$

and a pdf can be defined by using the Dirac delta-function $\delta(x)$:

$$p_x(x) = \sum_{k=1}^{N} P(x_k)\;\delta(x-x_k) \tag{2.18b}$$

Note that the expectations defined earlier still apply. The mean and variance, for example, are found to be

$$\mu_x = \sum_{k=1}^{N} x_k P(x_k) \quad \text{and} \quad \sigma_x^2 = \sum_{k=1}^{N} (x_k - \mu_x)^2 P(x_k)$$

by using (2.18) in (2.13b) and (2.14b).

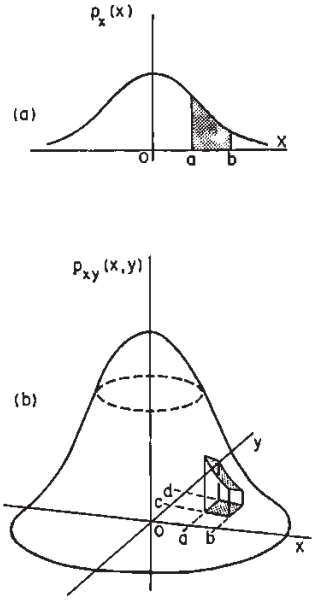
2.3.3 Bivariate Distributions

As a first step toward formalizing correlation functions, consider two r.v.'s X and Y whose mean values and variances are denoted by μ_x, σ_x^2, μ_y, σ_y^2, respectively. They have a *joint probability distribution* $P_{xy}(x,y)$ and a *joint pdf* [Figure 2.12(b)] defined, as in (2.8) and (2.10), by

$$P_{xy}(x,y) = P\{X \leqslant x,\ Y \leqslant y\}$$

$$p_{xy}(x,y) = \frac{\partial^2}{\partial x\, \partial y} P_{xy}(x,y); \qquad \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} p_{xy}(x,y)\, dx\, dy = 1 \tag{2.19a}$$

Figure 2.12 (a) pdf of the amplitude of r.v. X; total area under $p_x(x)$ is unity; (b) joint pdf of the amplitudes of two r.v.'s X and Y; total volume under $p_{xy}(x,y)$ is unity.



As in (2.11), the probability that the r.v.'s X and Y are jointly in ranges (a,b) and (c,d) is

$$P\{a<X\leqslant b,\ c<Y\leqslant d\} = \int_a^b \int_c^d p_{xy}(x,y)\,dx\,dy\ , \tag{2.19b}$$

the volume of the shaded element in Figure 2.12(b).

Two zero-mean r.v.'s X and Y are *jointly normal* if

$$p_{xy}(x,y) = \frac{1}{2\pi\sqrt{1-\rho_{xy}^2}\sigma_x\sigma_y}\exp\left[-\frac{1}{2(1-\rho_{xy}^2)}\left\{\left[\frac{x}{\sigma_x}\right]^2 - \frac{2xy\rho_{xy}}{\sigma_x\sigma_y} + \left[\frac{y}{\sigma_y}\right]^2\right\}\right] \tag{2.20}$$

where ρ_{xy} is the correlation coefficient to be defined in (2.26b).

Univariate pdf's can be derived from the joint pdf, in the form of *marginal pdf's*

$$p_x(x) = \int_{-\infty}^{\infty} p_{xy}(x,y)dy\ ; \quad p_y(y) = \int_{-\infty}^{\infty} p_{xy}(x,y)dx \tag{2.21}$$

That is, the value $p_x(x_1)$ of the pdf $p_x(\cdot)$ is the area under the profile of $p_{xy}(x_1,y)$ obtained by intersecting the surface $p_{xy}(\cdot,\cdot)$ by the plane $x = x_1$. The r.v.'s X and Y are said to be *statistically independent* if

$$p_{xy}(x,y) = p_x(x)p_y(y) \tag{2.22}$$

The expectation of a function $g(X,Y)$ of the two r.v.'s X and Y is

$$E[g(X,Y)] = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} g(x,y)p_{xy}(x,y)dxdy \tag{2.23}$$

We are mainly interested in the joint moments $E[X^mY^n]$. It is easy to see that $E[X^mY^n] = E[X^m]E[Y^n]$ if X and Y are statistically independent. In particular, X and Y are said to be *uncorrelated* if $E[XY] = E[X]E[Y]$; and *orthogonal* if $E[XY] = 0$. Clearly, if either of $E[X]$ or $E[Y] = 0$, a pair of uncorrelated r.v.'s is also orthogonal.

The mean of a sum of two r.v.'s is the sum of the means, regardless of statistical dependence:

$$\mu_{x+y} = E[X+Y] = E[X] + E[Y] = \mu_x + \mu_y \tag{2.24}$$

The power of the sum is

$$\chi_{x+y}^2 = E[(X+Y)^2] = E[X^2] + 2E[XY] + E[Y^2]$$

Using (2.14) in the last equation, we see that

$$\sigma_{x+y}^2 = \sigma_x^2 + \sigma_y^2 \quad \text{for uncorrelated r.v.'s}\ , \tag{2.25}$$

suggesting a simple *addition* of variances in this special case. Another value of interest is the *joint central moment*

$$\sigma_{xy} = E[(X-\mu_x)(Y-\mu_y)] , \tag{2.26a}$$

the *covariance of the r.v.'s X and Y*. The normalized covariance of X and Y is

$$\rho_{xy} = \sigma_{xy} / (\sigma_x \sigma_y) \tag{2.26b}$$

This quantity is also known as the *correlation coefficient*. Covariance and correlation coefficients are zero if X and Y are uncorrelated.

Conditional Distributions. The *conditional probability distribution function* of a r.v. X, *knowing* a single point event $\{Y = y\}$ [Papoulis, 1965], is

$$p_x(x|Y = y) = p_{xy}(x,y) / p_y(y) \tag{2.27}$$

provided that $p_y(y) \neq 0$. The r.v.'s are *independent* when the occurrence of any value $X = x$ is in no way influenced by the occurrence of any value $Y = y$. The conditional pdf is then $p_x(x|Y=y) = p_x(x)$ and the joint pdf is simply that of (2.22). It turns out that independent r.v.'s X and Y are also uncorrelated, as seen by using (2.26), (2.22) and (2.23); the converse of this statement is not true in general. But the converse does apply in the case of Gaussian statistics. In the special case of jointly normal (2.20) r.v.'s, if X and Y are uncorrelated ($\rho_{xy}=0$), they are also statistically independent. (See Problem 2.4).

As a generalization of (2.13a), we can define a *conditional expectation*

$$E[g(X)|Y = y] = \int_{-\infty}^{\infty} g(x) p_x(x|Y = y) dx , \tag{2.28}$$

the expectation of $g(X)$ knowing that event $Y = y$ has occurred.

In addition, we note that the *conditional* pdf of a r.v. X, given that event $\{X \in I\}$ has occurred, is

$$p_x(x|X \in I) = \begin{cases} p_x(x)/P\{X \in I\} & \text{if } X \in I \\ 0 & \text{otherwise} \end{cases} \tag{2.29}$$

Linear and Nonlinear Estimation. Bivariate distributions can be used directly in the solution of fundamental problems in estimation or prediction. For example, given the value y of a r.v. Y, *the best linear estimate* $\hat{x}_{opt}$ of the value x of another r.v. X, and the associated minimum mean square error in estimation, are

$$\hat{x}_{opt} = c_{opt}\, y = (\rho_{xy}\sigma_x/\sigma_y)\, y \; ; \qquad \min\{E[X-\hat{X}]^2\} = \sigma_x^2(1-\rho_{xy}^2) \tag{2.30}$$

For the same problem, the *best nonlinear estimate* is defined by

0064

$$\hat{x}_{opt} = E[X|Y=y]; \qquad \min\{E[X-\hat{X}]^2\} \leqslant \sigma_x^2(1-\rho_{xy}^2) \tag{2.31}$$

If X and Y are jointly Gaussian (2.20), the results of linear and nonlinear estimation are identical (Problem 2.8).

PDF of the Sum of Two Random Variables. We have already derived the mean value, power and variance of the sum of two r.v.'s, $Z = X + Y$. Events $\{Y = y\}$ and $\{X = z-y\}$ imply the event $\{Z = z\}$. Therefore the pdf of r.v. Z is

$$p_z(z) = \int_{-\infty}^{\infty} p_x(z-y|Y=y)p_y(y)dy = \int_{-\infty}^{\infty} p_{xy}(z-y, y)dy \tag{2.32}$$

If X and Y are *statistically independent*, we can use (2.22) to obtain

$$p_z(z) = \int_{-\infty}^{\infty} p_x(z-y)p_y(y)dy = p_x(z)*p_y(z) \tag{2.33}$$

where * denotes convolution.

The result of (2.33) can easily be extended to a sum of n variables. The *central limit theorem* states that, if these r.v.'s are independent of each other, the pdf of the sum of these r.v.'s approaches a Gaussian distribution very quickly as a function of n, regardless of the pdf's of the individual r.v.'s (Problem 2.5). In the case of a set of Gaussian r.v.'s, any linear transform (Chapter 12) yields another set of Gaussian r.v.'s, irrespective of whether they are independent or not.

2.3.4 Correlation Functions of Random Processes

We now come back to the description of properties of a random process $\{X(n)\}$; $n = 0, \pm 1, \pm 2$ by a simultaneous description of the behavior of two r.v.'s $X(m)$ and $X(n)$ of this process. These r.v.'s are illustrated by $X(r)$ and $X(s)$ in Figure 2.11(a). Equations (2.19a) and (2.28) have already defined joint and conditional pdf's. Partial descriptions of joint statistics are given by the *autocovariance function*

$$C_{xx}(m,n) = E[(X(m)-\mu_{x(m)})\ (X(n)-\mu_{x(n)})] \tag{2.34}$$

and the *autocorrelation function* (acf)

$$R_{xx}(m,n) = E[X(m)\ X(n)] = C_{xx}(m,n) + \mu_{x(m)}\mu_{x(n)} \tag{2.35}$$

For zero-mean sequences, the expectations above are identical:

$$R_{xx}(m, n) = C_{xx}(m, n) \quad \text{if} \quad \mu_{x(m)} = \mu_{x(n)} = 0 \tag{2.36}$$

Unless otherwise stated, we assume throughout the book that mean values are zero, and use the term acf *to signify* (2.36). Clearly, the acf and the autocovariance function depend on two time variables m and n or, alternatively, on

absolute time instant m and time displacement or time lag $k = n - m$. A *wide-sense stationary process*, however, is defined by the fact that its first-order pdf $p_{x(n)}(x)$ is independent of time (implying a constant mean) and that its joint pdf $p_{x(m)x(n)}(x,y)$ depends only on time shift (implying that the acf depends only on $n-m=k$). Indeed, wide-sense stationarity refers only to first- and second-order statistics. In the *strict-sense stationary process*, on the other hand, none of the statistics is affected by a time shift. This book will mainly assume wide-sense stationarity. It can be shown, however, that Gaussian processes are stationary in the strict sense if they are stationary in the wide sense [Papoulis, 1965].

For wide-sense stationary processes, the acf and autocovariance function do not depend on the absolute instant of time; they may be rewritten as

$$C_{xx}(k) = E[(X(n)-\mu_x)(X(n+k)-\mu_x)]; \qquad R_{xx}(k) = E[X(n)X(n+k)] \tag{2.37}$$

It can be shown that $R_{xx}(k) = R_{xx}(-k) = R_{xx}(|k|)$ and $|R_{xx}(k)| \leqslant R_{xx}(0)$ for all k; note also that $R_{xx}(0) = \chi_x^2 = \sigma_x^2 + \mu_x^2$ [see (2.14c)]. We will also make use of the *variance-normalized acf* defined in (2.5):

$$\rho_k = \rho_{xx}(k) = R_{xx}(k)/R_{xx}(0) ; \qquad \rho_{xx}(0) = 1 \tag{2.38}$$

The acf and autocovariance function are functions of time lag k; they are also *deterministic* since they describe average quantities of the process. Further, as we shall see presently, values of these functions cannot be chosen arbitrarily.

At this point it is convenient to introduce matrix terminology, which will be used extensively in Chapters 6 and 12. We denote an observation N-sequence and the associated r.v. sequence by column vectors

$$\mathbf{x} = \{x(k)\}; \qquad \mathbf{X} = \{X(k)\}; \qquad k = 0,1,\ldots,N-1 \tag{2.39a}$$

and the *autocorrelation* (or *mean square value) matrix* by

$$\begin{aligned} \mathbf{R}_{xx} &= \{R_{xx}(|k-l|)\} ; \\ &= R_{xx}(0)\ \{\rho_{xx}(|k-l|)\}; \qquad k,l = 0,1,\ldots,N-1 ; \end{aligned} \tag{2.39b}$$

$$\mathbf{R}_{xx} = \begin{bmatrix} R_0 & R_1 & R_2 & \cdots & R_{N-1} \\ R_1 & R_0 & R_1 & \cdots & R_{N-2} \\ R_2 & R_1 & R_0 & \cdots & R_{N-3} \\ \cdot & \cdot & \cdot & \cdots & \cdot \\ R_{N-1} & \cdot & \cdot & \cdots & R_0 \end{bmatrix} = R_0 \begin{bmatrix} 1 & \rho_1 & \rho_2 & \cdots & \rho_{N-1} \\ \rho_1 & 1 & \rho_1 & \cdots & \rho_{N-2} \\ \rho_2 & \rho_1 & 1 & \cdots & \rho_{N-3} \\ \cdot & \cdot & \cdot & \cdots & \cdot \\ \rho_{N-1} & \cdot & \cdot & \cdots & 1 \end{bmatrix} \tag{2.40}$$

where we have used the notation

$$R_{|k-l|} = R_0\, \rho_{|k-l|} = R_{xx}(|k-l|) = E[X(n+l)X(n+k)] \tag{2.41}$$

for the element in the kth row and lth column. The autocorrelation matrix can also be described by

$$\mathbf{R}_{xx} = E[\mathbf{X}\,\mathbf{X}^T] \tag{2.42}$$

where T denotes the *transpose* of a vector. Similarly, the *covariance matrix* (or *variance matrix*) is obtained by taking into account mean values μ_x:

$$\mathbf{C}_{xx} = \mathbf{R}_{xx} - \boldsymbol{\mu}_x\boldsymbol{\mu}_x^T = \{C_{xx}(|k-l|)\}\,; \qquad k,l = 0,1,\ldots,N-1 \tag{2.43}$$

where $\boldsymbol{\mu}_x$ is the mean-value vector with identical elements μ_x:

$$\boldsymbol{\mu}_x = E[\mathbf{X}] = \{E[X(k)]\}\,; \qquad k = 0,1,\ldots,N-1 \tag{2.44}$$

Clearly, $\mathbf{R}_{xx} = \mathbf{C}_{xx}$ for a zero-mean vector r.v. $\mathbf{X}$. It is easy to see that $\mathbf{R}_{xx}$ and $\mathbf{C}_{xx}$ are *nonnegative definite*; i.e., the scalar quantities $\boldsymbol{\alpha}^T\mathbf{R}_{xx}\boldsymbol{\alpha}$ and $\boldsymbol{\alpha}^T\mathbf{C}_{xx}\boldsymbol{\alpha}$ are nonnegative for any arbitrary N-dimensional vector $\boldsymbol{\alpha}$. For example, in the case of $\mathbf{R}_{xx}$,

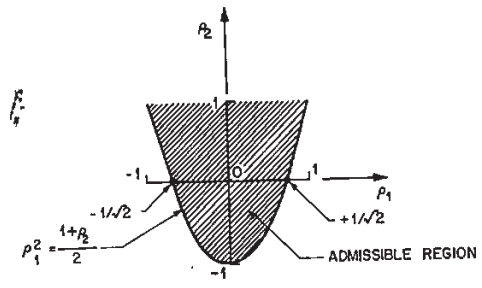
$$\boldsymbol{\alpha}^T\mathbf{R}_{xx}\boldsymbol{\alpha} = E[\boldsymbol{\alpha}^T\mathbf{X}\,\mathbf{X}^T\boldsymbol{\alpha}] = E[(\boldsymbol{\alpha}^T\mathbf{X})^2] \geq 0 \tag{2.45}$$

where the inequality holds because $\boldsymbol{\alpha}^T\,\mathbf{X}$ is a dot product. In the following we will assume that $\mathbf{R}_{xx}$ is *positive definite*, which implies that the equality in (2.45) is only achieved if $\boldsymbol{\alpha}$ is a zero-element vector. A positive-definite correlation matrix is also *nonsingular*, i.e., it has an *inverse*. This also implies that the psd and all *eigenvalues* (Section 2.3.5), which are measures of variances, are non-negative. (See the references in Appendix B). Finally, we note that

$$|\mathbf{R}_{xx}| > 0 \tag{2.46}$$

is a necessary and sufficient condition for $\mathbf{R}_{xx}$ to be a correlation matrix. For $N = 2$, (2.46) implies that $|\rho_1| < 1$. The case of $N = 3$ is illustrated in Problem 2.7 and Figure 2.13.

Figure 2.13 Constraints on ρ_1 and ρ_2 for $\mathbf{R}_{xx}$ to be a 3×3 correlation matrix.



0067

The correlation matrix is symmetric; also, elements along any diagonal in it are equal. Such a matrix is called a *symmetric Toeplitz* matrix. The elements $R_{xx}(k,l)$ of $\mathbf{R}_{xx}$ can refer either to N samples from a stationary scalar source (as in our context), or to N components of a vector source. In the latter case, the autocorrelation matrix can be non-Toeplitz. Elements along the main diagonal will now be mean square values of N different signal components, and these will differ in general.

A useful partial description of the dependencies between two random processes $\{X(n)\}$ and $\{Y(n)\}$ is the *crosscorrelation function*

$$R_{xy}(m,n) = E[X(m)Y(n)] \tag{2.47}$$

Again, if the processes are *jointly stationary* [i.e., $\{X(n)\}$ and $\{Y(n)\}$ are wide-sense stationary and the cross-correlation depends only on time shift], this function is a function of only one variable, the time lag k:

$$R_{xy}(k) = E[X(n)Y(n+k)] \tag{2.48}$$

It can be shown that $R_{xy}^2(k) \leqslant R_x(0)\, R_y(0)$ and $R_{xy}(k) = R_{yx}(-k)$ for all k. A *crosscovariance function* can be defined by including mean values in (2.48).

Effect of a Linear System on Mean and acf. Consider a linear, time-invariant and stable system with impulse response $h(n)$ and transfer function $H(e^{j\omega})$ (Appendix A). Consider a wide-sense stationary random process $\{Z(n)\}$ applied to its input. The mean value and acf of the output process $\{X(n)\}$ (Problem 2.10) are then given by

$$\mu_x = \mu_z\, H(0) \tag{2.49}$$

$$\begin{aligned} R_{xx}(k) &= R_{zz}(k) * h(k) * h(-k) \\ &= \sum_{i=-\infty}^{\infty} \sum_{j=-\infty}^{\infty} h(i)h(j)R_{zz}(i-j-|k|) \end{aligned} \tag{2.50}$$

Therefore, the mean of $\{Y(n)\}$ is constant and its acf depends solely on time displacement k. Hence $\{Y(n)\}$ is also wide-sense stationary.

The Multivariate Gaussian Distribution. Descriptions of N r.v.'s by joint pdf's are too complex to be of practical use, and these pdf's are usually not known, nor are they easy to measure. One relatively simple model, however, is the multivariate Gaussian distribution; this is of great importance in applications, because, as we have already mentioned, many physical events can be modeled by Gaussian distributions.

The joint Gaussian pdf of a vector random variable $\mathbf{X} = (X_0, X_1, \ldots, X_{N-1})^T$ is

$$p_{\mathbf{x}}(\mathbf{x}) = (2\pi)^{-N/2}|\mathbf{C}_{xx}|^{-1/2}\exp\left[\frac{1}{2}(\mathbf{x}-\mu_x)^T\mathbf{C}_{xx}^{-1}(\mathbf{x}-\mu_x)\right] \tag{2.51}$$

0068

in which $\mathbf{x} = [x_0, x_1, \ldots, x_{N-1}]^T$ is a vector of dummy variables, $|\mathbf{C}_{xx}|$ is the determinant of the covariance matrix $\mathbf{C}_{xx}$, and $\mathbf{C}_{xx}^{-1}$ is its inverse. Notice the important fact that the joint pdf in the Gaussian case is totally determined by the mean values of the elements of the vector r.v. $\mathbf{X}$ and by the covariances between its elements.

Second-order statistics are therefore sufficient for describing Gaussian processes. If the elements of $\mathbf{X}$ are uncorrelated, they are also statistically independent. In this special case,

$$p_{\mathbf{x}}(\mathbf{x}) = \prod_{k=0}^{N-1} p_{x_k}(x_k) \quad ; \qquad p_{x_k}(x_k) = \frac{1}{\sqrt{2\pi}\sigma_k} \exp\left[-\frac{1}{2}(x_k-\mu_k)^2/2\sigma_k^2\right] \tag{2.52}$$

Finally, consider any *linear transform* $\mathbf{Y} = \mathbf{AX}$ (Appendix B and Chapter 12); if $\mathbf{X}$ is Gaussian, the vector r.v. $\mathbf{Y}$ also has a joint Gaussian pdf.

2.3.5 Eigenvalues and Eigenvectors

Associated with any correlation matrix $\mathbf{R}_{xx}$ is a set of *eigenvalues* λ_i and *eigenvectors* $\mathbf{l}_i$ defined by

$$\mathbf{R}_{xx}\mathbf{l}_i = \lambda_i \mathbf{l}_i \; ; \qquad i = 0,1,\ldots,N-1 \tag{2.53}$$

Eigenvectors and eigenvalues will play an important role in the analyses of Transform Coding (Chapter 12).

Let $\mathbf{l}$ be an eigenvector of $\mathbf{R}_{xx}$. The corresponding eigenvalue λ is then defined by

$$\mathbf{R}_{xx}\,\mathbf{l} = \lambda \mathbf{l} \quad \text{or} \quad (\mathbf{R}_{xx}-\lambda \mathbf{I})\,\mathbf{l} = 0 \tag{2.54}$$

This is a linear homogeneous system of equations for the unknown λ. In order to have a nonzero eigenvector, it is both necessary and sufficient that

$$|\mathbf{R}_{xx}-\lambda \mathbf{I}| = 0 \tag{2.55}$$

The above equation is called the *characteristic equation* of $\mathbf{R}_{xx}$. This determinant, when expanded, is a polynomial in λ of degree N, and it will have N zeros λ_i with corresponding eigenvectors $\mathbf{l}_i$. The eigenvectors are of course determined only to within a scalar multiple. Recall that $\mathbf{R}_{xx}$ is a real symmetric matrix. It can be shown therefore that the characteristic equation has N distinct and real zeros (with multiplicities, in general); and consequently, it has exactly N distinct eigenvectors. In general, it is very difficult to find the eigenvalues of a matrix, and numerical techniques are needed.

Example 2.2. Example of $N = 2$

Consider a stationary source with variance $\sigma_x^2 = 1$ and normalized autocorrelation ρ between neighboring samples

$$\mathbf{R}_{xx} = \begin{bmatrix} 1 & \rho \\ \rho & 1 \end{bmatrix} \tag{2.56}$$

The eigenvalues λ are obtained from the characteristic equation (2.55):

$$\begin{vmatrix} 1-\lambda & \rho \\ \rho & 1-\lambda \end{vmatrix} = 0 ; \qquad \lambda_0 = 1+\rho ; \quad \lambda_1 = 1-\rho \tag{2.57}$$

To find the eigenvector $\mathbf{l}_0$ we use (2.53):

$$\begin{bmatrix} 1 & \rho \\ \rho & 1 \end{bmatrix} \begin{bmatrix} l_0(0) \\ l_0(1) \end{bmatrix} = (1+\rho) \begin{bmatrix} l_0(0) \\ l_0(1) \end{bmatrix} ; \qquad l_0(0) = l_0(1) \tag{2.58a}$$

Similarly, for $\mathbf{l}_1$, we find that $l_1(0) = -l_1(1)$. If we scale the vectors so that

$$l_0(0)^2 + l_0(1)^2 = l_1^2(0) + l_1^2(1) = 1 , \tag{2.58b}$$

$$\mathbf{l}_0 = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 \\ 1 \end{bmatrix} ; \quad \mathbf{l}_1 = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 \\ -1 \end{bmatrix} \tag{2.59}$$

The above equations show that the product of the eigenvalues λ_0 and λ_1 equals $1-\rho^2 = |\mathbf{R}_{xx}|$. In addition, the eigenvectors are *orthonormal*, i.e.,

$$\mathbf{l}_0^T \mathbf{l}_1 = \mathbf{l}_1^T \mathbf{l}_0 = 0 ; \qquad \mathbf{l}_0^T \mathbf{l}_0 = \mathbf{l}_1^T \mathbf{l}_1 = 1 \tag{2.60}$$ •

For $N > 2$, computations are tedious. However, so-called MA(1) and AR(1) processes (Section 2.4) will have explicit solutions.

Some Properties of Eigenvalues and Eigenvectors

(a) Eigenvectors are *orthogonal* ($\mathbf{l}_i^T \mathbf{l}_j = 0$ for $i \neq j$), and by choosing a proper scaling in $\mathbf{R}_{xx}\mathbf{l} = \lambda \mathbf{l}$, they can be made to be *orthonormal*:

$$\mathbf{l}_i^T \mathbf{l}_j = \delta_{ij} \tag{2.61}$$

(b) Eigenvectors define $\mathbf{R}_{xx}$ (Problem 2.20) in the form

$$\mathbf{R}_{xx} = \sum_{i=0}^{N-1} \lambda_i \mathbf{l}_i \mathbf{l}_i^T \tag{2.62}$$

(c) Eigenvalues have the properties

$$|\mathbf{R}_{xx}| = \prod_{j=0}^{N-1} \lambda_j ; \qquad \text{trace } (\mathbf{R}_{xx}) = \sum_{j=0}^{N-1} \lambda_j \tag{2.63a}$$

Note that trace $(\mathbf{R}_{xx}) = N\sigma_x^2$ for zero-mean processes. Hence for any N, the average value of eigenvalues equals the variance of the process:

0070

$$\sigma_x^2 = \frac{1}{N}\sum_{j=0}^{N-1}\lambda_j \tag{2.63b}$$

Since the eigenvalues are measures of variance, they are non-negative. In addition, if λ_j's are the eigenvalues of $\mathbf{R}_{xx}$, then the eigenvalues of $\mathbf{R}_{xx}^{-1}$ are given as λ_j^{-1}.

(d) Let $S_{xx}(e^{j\omega})$ be the psd (see Section 2.3.6) of the process with correlation matrix $\mathbf{R}_{xx}$. Then

$$\lambda_j = S_{xx}(e^{j\Omega_j}) \tag{2.64}$$

i.e., eigenvalues are samples, although not equidistant ones, of the psd. Specific Ω_j-values, the *eigenfrequencies*, are usually not simply obtainable from the psd.

(e) For any function $f(\cdot)$, the *Toeplitz distribution theorem (Szego-theorem)* [Grenander and Szego, 1958] states that for any function $f(\cdot)$,

$$\lim_{N\to\infty}\frac{1}{N}\sum_{j=0}^{N-1}f(\lambda_j) = \frac{1}{2\pi}\int_{-\pi}^{\pi} f[S_{xx}(e^{j\omega})]d\omega \tag{2.65}$$

For example, with $f(\cdot) = \log_e(\cdot)$, we have

$$\lim_{N\to\infty}\log_e\left[\prod_{j=0}^{N-1}\lambda_j\right]^{1/N} = \frac{1}{2\pi}\int_{-\pi}^{\pi}\log_e S_{xx}(e^{j\omega})d\omega \tag{2.66}$$

Hence, with property (c),

$$\lim_{N\to\infty}|\mathbf{R}_{xx}|^{1/N} = \exp\left[\frac{1}{2\pi}\int_{-\pi}^{\pi}\log_e S_{xx}(e^{j\omega})d\omega\right] \tag{2.67}$$

2.3.6 Power Spectral Density of Random Processes

Many results on stationary random processes, involving only second moments, can also be discussed in the frequency domain. Suppose that $\{X(n)\}$ is a discrete-time wide-sense stationary process with acf $R_{xx}(k)$ (2.37). The *power spectral density function* (psd) is defined to be the Fourier transform of the acf of the process; we thus obtain a Fourier transform pair

$$R_{xx}(k) = \frac{1}{2\pi}\int_{-\pi}^{\pi} S_{xx}(e^{j\omega})e^{jk\omega}d\omega\,; \tag{2.68a}$$

$$S_{xx}(e^{j\omega}) = \sum_{k=-\infty}^{\infty} R_{xx}(k)e^{-jk\omega} \tag{2.68b}$$

which is a statement of the *Wiener-Khinchine theorem*. The function $S_{xx}(e^{j\omega})$ exists if $R_{xx}(k)$ is absolutely summable. Also, using the fact that $R_{xx}(k)$ is a real-valued even function,

$$S_{xx}(e^{j\omega}) = R_{xx}(0) + 2\sum_{k=1}^{\infty} R_{xx}(k)\cos k\omega \tag{2.69a}$$

Hence, the psd is a real even function of frequency ω, i.e., $S_{xx}(e^{j\omega}) = S_{xx}(e^{-j\omega})$. As a result, the acf can also be rewritten in the form

$$R_{xx}(k) = \frac{1}{2\pi}\int_{-\pi}^{\pi} S_{xx}(e^{j\omega})\cos k\omega\, d\omega \tag{2.69b}$$

Further, the area under the power spectral density function in the range $(-\pi,\pi)$ is 2π times the average power in the process [see (2.14a)]:

$$\chi_x^2 = E[X^2(n)] = R_{xx}(0) = \frac{1}{2\pi}\int_{-\pi}^{\pi} S_{xx}(e^{j\omega})d\omega \tag{2.70}$$

Specifically, the power of the process in the frequency range $(\omega_k - \Delta\omega/2, \omega_k + \Delta\omega/2)$ is approximately $S_{xx}(e^{j\omega_k})\Delta\omega/\pi$. Since the power cannot assume negative values, it is clear that we must have $S_{xx}(e^{j\omega}) \geqslant 0$ for all ω. This is the spectral equivalent to the requirement of a non-negative-definite $\mathbf{R}_{xx}$.

The *cross spectral density* of two jointly stationary processes $\{X(n)\}$ and $\{Y(n)\}$ is

$$S_{xy}(e^{j\omega}) = \sum_{k=-\infty}^{\infty} R_{xy}(k)e^{-jk\omega} \tag{2.71}$$

This function exists if $R_{xy}(\cdot)$ is absolutely summable. Since $R_{xy}(k) = R_{yx}(-k)$, it can be shown that $S_{yx}(e^{j\omega}) = S_{xy}^*(e^{j\omega})$, where * denotes a conjugate. Note that the cross spectral density function is not necessarily even or real, or positive.

Effect of a Linear System on psd. Consider a process $\{Z(n)\}$ applied to a linear, time-invariant and stable filter with transfer function $H(e^{j\omega})$. The psd of output $\{X(n)\}$ is given by (Problem 2.10)

$$S_{xx}(e^{j\omega}) = S_{zz}(e^{j\omega})\,|H(e^{j\omega})|^2 \tag{2.72}$$

2.3.7 Spectral Flatness Measure

Consider a zero-mean process $\{X(n)\}$ with psd $S_{xx}(e^{j\omega})$. Its variance is

$$\sigma_x^2 = \frac{1}{2\pi}\int_{-\pi}^{\pi} S_{xx}(e^{j\omega})d\omega \tag{2.73}$$

0072

We define the *spectral flatness measure (sfm)* [Makhoul and Wolf, 1972] as

$$\gamma_x^2 = \frac{\exp\left[\frac{1}{2\pi}\int_{-\pi}^{\pi} \log_e S_{xx}(e^{j\omega})\,d\omega\right]}{\frac{1}{2\pi}\int_{-\pi}^{\pi} S_{xx}(e^{j\omega})\,d\omega} = \frac{\eta_x^2}{\sigma_x^2} \tag{2.74}$$

Note that $\gamma_x^2 = 1$ if $S_{xx}(e^{j\omega}) = \sigma_x^2$, which is the psd that describes a *white noise* process. We will see presently that

$$0 \leqslant \gamma_x^2 \leqslant 1 \tag{2.75}$$

We will use *sfm* as an important tool to describe the shape of a psd by a single value. Indeed, the *sfm* bounds the performance of two classes of coding schemes, classes that exploit waveform redundancy (as shown by a non-flat psd) by *predictive coding* and by *transform coding*. In fact, we shall see later that η_x^2, the numerator in (2.74), can be interpreted as a *minimum prediction error variance*, or as the variance of a white innovations process that is modified by a linear filter to yield a process $\{X(n)\}$ that is colored and hence predictable.

The inverse of the *sfm*, γ_x^{-2}, is a measure of *waveform predictability*. This number is typically 3 and 16 for the long-term psd's of speech and video, respectively. Short-term psd's are sometimes characterized by much higher values of γ_x^{-2}, implying higher predictability. From Wiener theory, it is known further that processes with $S_{xx}(e^{j\omega}) = 0$ over finite intervals are predictable with zero error. For these cases, $\gamma_x^2 = 0$, $\gamma_x^{-2} = \infty$, and the predictability is infinite.

We now show that $0 \leqslant \gamma_x^2 \leqslant 1$, as claimed in (2.75). Let the psd $S_{xx}(\cdot)$ be divided into N sections of width $\Delta\omega = \pi/N$ within the interval $(0,\pi)$. The samples of the psd are

$$S_k = S_{xx}(e^{j\omega_k}) \quad \text{where} \quad \omega_k = \frac{\pi}{N}(k - \tfrac{1}{2}); \qquad k = 1,2,\ldots,N \tag{2.76}$$

The denominator of (2.74) then reads

$$\sigma_x^2 = \lim_{N\to\infty} \frac{1}{\pi}\sum_{k=1}^{N} S_k \Delta\omega = \lim_{N\to\infty} \frac{1}{N}\sum_{k=1}^{N} S_k$$

and the numerator is

$$\eta_x^2 = \lim_{N\to\infty} 2^{\frac{1}{N}\sum_{k=1}^{N}\log_2 S_k} = \lim_{N\to\infty}\left[\prod_{k=1}^{N} S_k\right]^{1/N} \tag{2.77}$$

Therefore, γ_x^2 is the ratio of the geometric mean and the arithmetic mean of the samples of the psd. This ratio, by definition, is not greater than unity. The value of unity is only obtained if all samples are equal. Later sections will give numerical examples to illustrate (2.75).

The spectral flatness measure is reflected directly in the *entropy power* Q (Appendix C). For a Gaussian source, $Q = \gamma_x^2 \sigma_x^2$. Also, the distortion rate function for this source (Appendix D) is $D(R)_G = 2^{-2R}\gamma_x^2 \sigma_x^2$. Thus, smaller γ_x^2 values signify lesser entropy, greater predictability, and a smaller theoretically possible mean square distortion D for given rate of R bits/sample.

2.3.8 Optimum Linear Predictions and Transforms

The significance of the acf and psd descriptions developed so far is in the fact that such second-order designs are adequate for optimal designs of *linear predictive* and *linear transform* coders. Prediction and transform operations serve to remove the redundancy in an input waveform, and thus permit bit rate-efficient digitizations thereof. Nonadaptive and adaptive versions of these coders remove input redundancy by utilizing long-time-averaged and short-time statistics, respectively. Further, predictive and transform coding techniques respectively use *time-domain* and *frequency-domain* approaches for redundancy removal, as we shall see in later chapters. A common feature is that because of the linear operations involved, both of these approaches are regarded as means for utilizing *linear redundancy*; and the performances of both coder classes are upper-bounded by a measure of this linear redundancy that we have already developed, the *spectral flatness measure* γ_x^2.

While discussing predictive and transform coders, we shall come across several parameters that depend directly on waveform properties. These parameters could therefore be discussed appropriately in this chapter on waveform characterization. However, purely for convenience, we shall defer detailed discussions of these to respective coding chapters. The properties we have in mind are *linear prediction coefficients*, *parcor coefficients* (Chapter 6) and *transform coefficients*, especially those resulting from a *Karhunen-Loeve Transform* (Chapter 12). The following is merely a brief introduction to these concepts, using the simple example of a 1-D input.

Linear prediction coefficients h_j are used in *one-step-ahead* predictions

$$\hat{x}(n) = \sum_{j=1}^{N} h_j \, x(n-j) \tag{2.78}$$

For any given order of prediction N, the mean square value ${}^N\sigma_d^2$ of prediction error $d(n) = x(n) - \hat{x}(n)$ will be minimized by

$$\mathbf{h}_{opt} = \mathbf{R}_{xx}^{-1} \, \mathbf{r}_{xx} \tag{2.79}$$

where $\mathbf{R}_{xx}$ is the correlation matrix and $\mathbf{r}_{xx}$ is a vector which has the first N acf values as its elements. If waveform sources are modeled as the output of a linear filter with a white noise or *innovations* process $\{Z(n)\}$ as its input, the asymptotic value of minimum mean square prediction error $d(n)$, as $N \to \infty$, merely given by the variance σ_z^2 of the innovations

$$\eta_x^2 = \min \{{}^\infty\sigma_d^2\} = \sigma_z^2 \tag{2.80}$$

0074

The optimum prediction error filter is in fact a *whitening filter* for the waveform at the input to a coder, and its transfer function is the inverse of the input psd. The decoder in a linear predictive system operates on a quantized version of $d(n)$ to recover an approximation to $x(n)$. The transfer function of this decoder in fact matches the input psd.

Autoregressive or Markov source models (Section 2.4) are very useful in the analysis of predictive coders. In this case, the reconstruction at the decoder is often performed with an infinite impulse response filter which is the inverse of the prediction error filter. An interesting special property is that for an AR(N) process, or a Markov process of order N, the optimum linear predictor has a finite order, N.

Linear transforms of an input random vector $\mathbf{X}$ of length N are defined by

$$\mathbf{Y} = \mathbf{AX} \tag{2.81}$$

where $\mathbf{X}$ is the input column vector of length N, $\mathbf{A}$ is an $N \times N$ transform matrix, and $\mathbf{Y}$ is the random vector of transform coefficients.

Second-order statistics of $\mathbf{Y}$ are described by

$$\mathbf{R}_{yy} = E[\mathbf{Y}\mathbf{Y}^T] = \mathbf{A}\,\mathbf{R}_{xx}\,\mathbf{A}^T \tag{2.82}$$

If $\mathbf{A}$ is the so-called *Karhunen-Loeve Transform* (KLT) for the input process, $\mathbf{R}_{yy}$ will be diagonal. This special *covariance-diagonalizing* transform is input-process-specific by definition and therefore complex to implement. Suboptimum linear transforms such as the Discrete Fourier Transform (DFT) approach the above ideal of diagonalization for special input models or classes. We shall see later why the diagonalization of $\mathbf{R}_{yy}$ is an important ideal in transform coding. We shall merely note at this point that a side result of such diagonalization will be that the transform coefficient variances $R_{yy}(i,i) = \sigma_i^2$; $i = 0,1,...,N-1$ (the main diagonal elements of $\mathbf{R}_{yy}$) will be *unequal*, unlike the main diagonal elements in the Toeplitz matrix $\mathbf{R}_{xx}$, which are equal by definition. The success of Transform Coding depends in fact on this ordering of coefficient variances and the related strategy of *variable bit allocation*. In the KLT, the main diagonal elements of $\mathbf{R}_{yy}$ are called the *eigenvalues* λ_i that are characteristic of the input process. The *arithmetic mean* of λ_i; $i = 0,1,...,N-1$ is merely the input variance σ_x^2 [see (2.63b)]; and for $N \to \infty$, we can use (2.63a) and (2.67) to show that the *geometric mean* is the input-specific parameter η_x^2 defined earlier. This is the numerator of (2.74), a measure of input redundancy.

For a white input spectrum, $\eta_x^2 = \sigma_x^2$; in this special case, the linear redundancy is zero, the gains due to predictive coding and transform coding are both 0 dB, and efficient digitization can be realized by a simple PCM coder (Chapter 5).

2.4 Model Sources with Special Correlations and Spectra

In the preceding sections we have defined various quantities that give a partial description of a random process. The quantities of main interest are the pdf, the

acf and the psd. Measurements of these properties are simplified if the process can be assumed to be ergodic. With such a process, statistics can be computed simply by averaging over a sample sequence. In the case of a stationary process, the psd is an equivalent description of the acf, and the spectral flatness measure is a compact description of waveform redundancy. Finally, in the case of a Gaussian process, probability distributions are completely defined by its mean and autocorrelation functions.

Waveforms such as speech are not Gaussian, ergodic, or stationary. However, *local stationarity* or *quasi-stationarity* makes random process models very meaningful for short-time descriptions. In fact, even long-time-averaged random process descriptions have demonstrated their value in terms of improving coder design and performance.

The purpose of this section is to describe model sources characterized mainly by special acf and psd functions; we shall subsequently use these as simple probabilistic models for the speech and image statistics described in Section 2.2. These simple models will be used to derive physical insight as well as to provide bases for analytical design. Some of the models are appropriate for globally averaged or *long-time-averaged* descriptions, while some models (for example, the autoregressive model for voiced speech) are excellent descriptions of *short-time* behavior.

2.4.1 Discrete-Time Models

Equations (2.49), (2.50) and (2.72) have specified the effect of linear filters on the mean, acf and psd of an input process. Interesting classes of discrete-time linear processes can be generated by passing *constant-psd* or *white-noise* sequences through a linear, time-invariant filter with impulse response $h(n)$ (Figure 2.14). A wide-sense-stationary zero-mean white-noise process $\{Z(n)\}$ is defined by

$$S_{zz}(e^{j\omega}) = \sigma_z^2 \tag{2.83}$$

$$R_{zz}(k) = E[Z(n)Z(n+k)] = \sigma_z^2\,\delta(k)\,; \quad \delta(k) = \begin{cases} 1 & \text{for } k = 0 \\ 0 & \textit{otherwise} \end{cases} \tag{2.84}$$

In (2.84), $\delta(k)$ is the *Kronecker delta* function. For any nonzero time lag k there is zero correlation between the samples of the process. Suppose that this white-noise or *innovations* signal is passed through a linear time-invariant filter whose transfer function $H(z)$ is a rational function of z. Such a function can be represented as a ratio of two polynomials in z:

$$H(z) = \frac{A(z)}{B(z)} = \frac{\sum_{j=0}^{M} a_j z^{-j}}{1 - \sum_{j=1}^{N} b_j z^{-j}} \tag{2.85}$$

0076

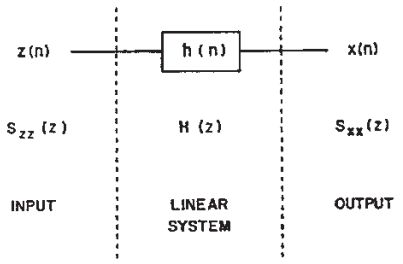


Figure 2.14 Linear system with wide-sense-stationary input and output.

with N poles (roots of the denominator polynomial) and M zeros (roots of the numerator polynomial). The output is given by the difference equation

$$X(n) = \sum_{j=0}^{M} a_j Z(n-j) + \sum_{j=1}^{N} b_j X(n-j) \tag{2.86}$$

If the filter impulse response is absolutely summable, the filter is stable and its output is a wide-sense stationary process $\{X(n)\}$ with an acf (Problem 2.10)

$$R_{xx}(k) = R_{zz}(k) * h(k) * h(-k) = \sigma_z^2 \sum_{j=0}^{\infty} h(j)h(j+|k|) \tag{2.87a}$$

with zero mean

$$E[X(n)] = 0 \tag{2.87b}$$

and a z-transform

$$S_{xx}(z) = \sigma_z^2 H(z) \tilde{H}(z^{-1}) \tag{2.88}$$

The average power in the output (Problem 2.10) is

$$E[X^2(n)] = R_{xx}(0) = \sigma_x^2 = \sigma_z^2 \sum_{j=0}^{\infty} h^2(j) \tag{2.89}$$

The ratio of the variances of the source and the innovations will be called the *power transfer factor*

$$\alpha = \sigma_x^2/\sigma_z^2 = \sum_{j=0}^{\infty} h^2(j) = \frac{1}{2\pi} \int_{-\pi}^{\pi} |H(e^{j\omega})|^2 d\omega \tag{2.90}$$

The factor α is the energy in the impulse response, and the right-hand equality in (2.90) is due to *Parseval's theorem*. Since $\{Z(n)\}$ is a white process, the coloring or non-flatness of the psd of $\{X(n)\}$ is due entirely to $H(z)$. In fact, for any rational $S_{xx}(e^{j\omega})$, *the sfm is just the inverse of the power transfer factor* (Problem 2.11) if $h(0)$ is normalized to unity:

$$\gamma_x^2 = \sigma_z^2/\sigma_x^2 = \alpha^{-1} \tag{2.91}$$

In the specific models used in the rest of this section, the value of $h(0)$ is indeed unity. If this normalization is not ensured, the power transfer factor will include a gain or scaling factor that will be irrelevant to spectral shaping. In the degenerate case of $h(j) = 0$ for all $j \neq 0$, and $h(0) \neq 1$, the filter will not color the white input, but it will simply scale it by a factor $h(0) \neq 1$. The value of α will be $h^2(0) \neq 1$, but the value of *sfm* will continue to be unity, the value characteristic of a white psd according to (2.74). In other words, the *sfm* will *not* be the reciprocal of α in this un-normalized case.

From (2.86), each r.v. of the process can be generated from the input and linear combinations of previous inputs and outputs. It is useful to consider in some detail two special cases that have proven to be of importance in modeling waveform sequences: processes that result from filtering white noise with either an *infinite impulse response* filter or a *finite impulse response* filter. In the first case the process is called an *all-pole process*, an *autoregressive* (AR) *process* or a *Markov process*. In the second case, the process is called an *all-zero process*, or a *moving average* (MA) *process*. The process described by (2.85) and (2.86) includes AR and MA terms and is sometimes called an *autoregressive moving average* (ARMA) *process*.

2.4.2 Autoregressive or All-Pole Processes

An autoregressive (AR) process is generated by passing the *innovations* or *white-noise* process $\{Z(n)\}$ of psd $S_{zz}(z) = \sigma_z^2$ through a filter

$$B(z) = \frac{1}{1 - \sum_{j=1}^{N} b_j z^{-j}} = \frac{z^N}{z^N - \sum_{j=1}^{N} b_j z^{N-j}} \tag{2.92}$$

This transfer function is termed *all-pole* since the N multiple zeros at $z = 0$ are trivial. The following is a difference equation which generates the AR process $\{X(n)\}$:

$$X(n) = Z(n) + \sum_{j=1}^{N} b_j X(n-j) \tag{2.93}$$

The process $\{X(n)\}$ is called an AR(N) *process* or an *Nth order Markov process*. The sample sequences are also called *Markov-N* sequences or *autoregressive* sequences, and the b_j are called *autoregressive constants*. Figure 2.15 shows the filter that realizes the AR-process, with z^{-1} denoting unit delay.

The impulse response of an all-pole filter is not finite in general. However, the acf of the output process $\{X(n)\}$ can be calculated recursively for any given set of coefficients b_j; $j = 1,2,...,N$. This is seen by multiplying $X(n)$ in (2.93) with $X(n-k)$ and taking expectations on both sides of the equation; and by noting that $E[Z(n)X(n-k)] = 0$ for $k > 0$ because the innovations signal $Z(n)$ is by definition uncorrelated with past outputs. As a result of the above steps,

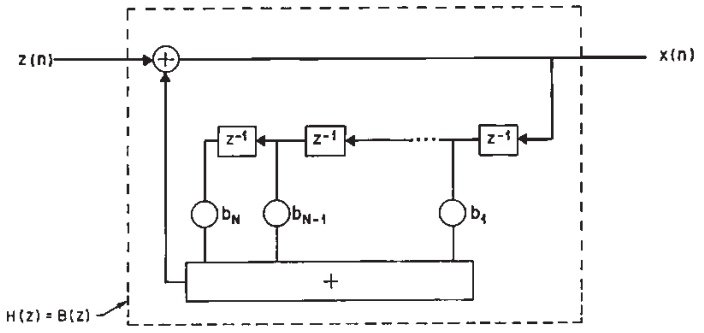


Figure 2.15 Generation of an AR(N) process.

$$\sigma_x^2 = R_{xx}(0) = \sum_{j=1}^{N} b_j R_{xx}(j) + \sigma_z^2 \tag{2.94}$$

$$R_{xx}(k) = \sum_{j=1}^{N} b_j R_{xx}(k-j)\ ; \qquad k > 0 \tag{2.95}$$

For a given set of AR constants b_j; j=1,2,...,N (a given all-pole filter), the values $R_{xx}(k)$ for $k = 0,1,...,N$ can be solved recursively via fast algorithms [Atal and Hanauer, 1971]. For $k > N$, (2.95) can be used to calculate all needed acf-values $R_{xx}(k)$ successively as the linear combination of acf predecessors weighted by the set of coefficients b_j; $j = 1,2,...,N$. A very related problem is that of finding the set b_j; $j = 1,2,...,N$ of coefficients of a recursive filter whose first N output acf-values match a set of N given or measured acf-values. For $k = 1,2,...,N$, we obtain from (2.95) a set of N linear equations for those coefficients b_j, $j = 1,2,...,N$ in terms of given acf values. Chapter 6 will discuss the problem of solving this set of linear equations. The coefficients b_j will be shown to be numerically identical to a set of *optimum predictor coefficients.*

Example 2.3. First-order Markov or AR(1) process:

The AR(1) process with zero mean, $N = 1$ and $b_1 = \rho$ is generated by

$$X(n) = Z(n) + \rho X(n-1) \tag{2.96}$$

$$= Z(n) + \sum_{j=1}^{\infty} \rho^j Z(n-j) \tag{2.97}$$

i.e., by passing white noise of variance σ_z^2 through a filter with impulse response

$$h(n) = \begin{cases} \rho^n & \text{for } n \geqslant 0 \\ 0 & \text{for } n < 0 \end{cases} \tag{2.98}$$

and frequency response

$$H(e^{j\omega}) = (1 - \rho e^{-j\omega})^{-1} \tag{2.99}$$

which is that of a first-order all-pole filter. We shall now calculate the acf $R_{xx}(k)$. From (2.97),

$$X(n+k) = Z(n+k) + \sum_{l=1}^{\infty} \rho^l Z(n+k-l) \tag{2.100}$$

Hence

$$R_{xx}(k) = E[X(n)X(n+k)]$$

$$= R_{zz}(k) + \sum_{j=1}^{\infty} \rho^j R_{zz}(k+j) + \sum_{l=n}^{\infty} \rho^l R_{zz}(k-l) + \sum_{j=1}^{\infty} \sum_{l=1}^{\infty} \rho^j \rho^l R_{zz}(k+j-l) \tag{2.101}$$

With $R_{zz}(k) = \sigma_z^2 \delta(k)$, and the identity $1 + \sum_{j=1}^{\infty} \rho^{2j} = (1-\rho^2)^{-1}$,

$$\sigma_x^2 = R_{xx}(0) = \frac{\sigma_z^2}{1-\rho^2}; \qquad R_{xx}(k) = \sigma_x^2 \rho^{|k|} \tag{2.102}$$

Note that acf values could also have been calculated recursively using (2.95):

$$\rho_k = \rho_{xx}(k) = \rho\, \rho_{xx}(k-1) = \rho^k \tag{2.103}$$

The first part of (2.102) also implies that the *sfm* is

$$\gamma_x^2 = 1 - \rho^2 = \alpha^{-1} \tag{2.104}$$

The spectrum is flat for $\rho = 0$ and $\gamma_x^2 \to 0$ for $\rho \to 1$. The special case of $\rho = 1$ is the *Wiener process* to be discussed in Example 2.4.

The psd of the AR(1) process is

$$S_{xx}(e^{j\omega}) = \sigma_z^2 |H(e^{j\omega})|^2 = \frac{1-\rho^2}{1+\rho^2-2\rho\cos\omega}\,\sigma_x^2 \tag{2.105}$$

and the reader may wish to verify (Problem 2.12) that this psd indeed has the *sfm* value of $\gamma_x^2 = (1-\rho^2)$. Note that the AR constant b_1 is the first normalized acf value of the AR(1) process, i.e., $b_1 = \rho = \rho_1$.

The process is stationary if the filter is stable; therefore we require that $|\rho| < 1$, since

$$\sum_{n=0}^{\infty} |h(n)| = (1 - |\rho|)^{-1} \tag{2.106}$$

is then finite. Figures 2.16(a) and 2.16(b) discuss examples of $\rho = 0.9$ and $\rho = -0.9$, respectively. In each of the figures (a) and (b), (i) is a segment of a sample sequence, (ii) is the normalized acf, and (iii) is the psd. Note that the acf takes a long time to decay in the case of a large positive correlation $\rho = 0.9$. The smooth nature of $x(n)$ in this case is also reflected in the psd of Figure 2.16(a), which has a *lowpass* characteristic. In Figure 2.16(b), where $\rho = -0.9$, adjacent samples are highly negatively correlated and the acf oscillates; and the psd has a *highpass* characteristic.

0080

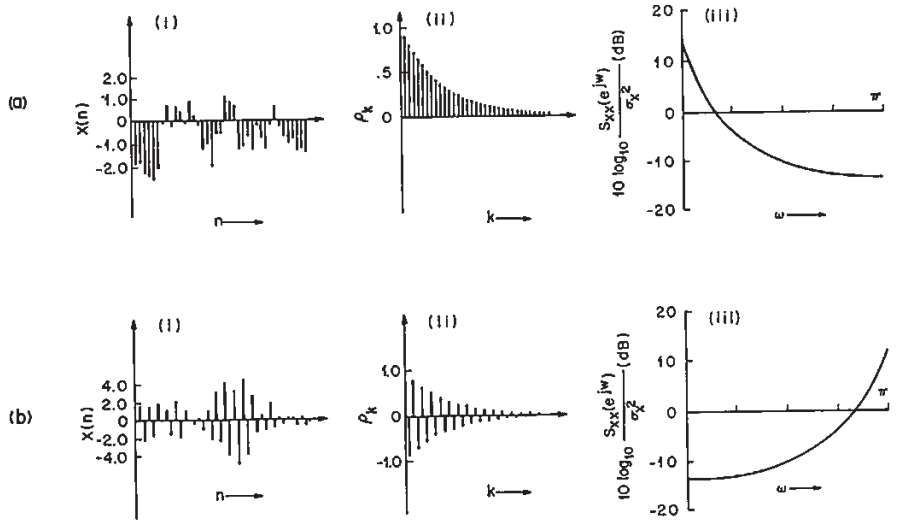


Figure 2.16 First-order Markov or AR(1) process: (i) typical sequence $x(n)$; (ii) acf ρ_k [Jenkins and Watts, 1968. Reprinted with permission]; and (iii) psd $S_{xx}(e^{j\omega})$; for (a) $\rho = 0.9$ and (b) $\rho = -0.9$.

Finally, the first-order Markov process has an Lth *order* correlation matrix $\mathbf{R}_{xx} = E[\mathbf{X}\,\mathbf{X}^T]$ with elements $R_{xx}(k-l) = \sigma_x^2\,\rho^{|k-l|}$:

$$\mathbf{R}_{xx} = \sigma_x^2 \begin{bmatrix} 1 & \rho & \rho^2 & \cdot & \cdot & \rho^{L-1} \\ \rho & 1 & \rho & & \cdot & \cdot \\ \rho^2 & \rho & 1 & \rho & \cdot & \cdot \\ \cdot & & \rho & 1 & \cdot & \cdot \\ \cdot & & & \rho & \cdot & \rho \\ \rho^{L-1} & \cdot & \cdot & & \cdot & 1 \end{bmatrix} \tag{2.107}$$

which is of course symmetric Toeplitz. Its eigenvalues are (Section 2.3) samples of the psd (2.105). They can be obtained as roots of the characteristic polynomial [Berger, 1971]. Also, see Example 12.3. For highly correlated processes ($\rho \to 1$), the eigenfrequencies Ω_k are uniformly distributed in $(0,\pi)$; and the eigenvectors are sampled sine waves with phases depending on Ω_k. •

Example 2.4. Wiener process

We have stated that the process generated by (2.96) is stationary for $|\rho| < 1$. We will now show that a nonstationary process results if $\rho = 1$. Let the process start at $k = 1$. At time n,

$$X(n) = \sum_{k=1}^{n} Z(k) \tag{2.108}$$

If $\{Z(n)\}$ is assumed to be stationary with mean μ_z and variance σ_z^2, the output process then has a *linearly increasing* mean value and variance:

0081

$$E[X(n)] = n\mu_z\ ; \qquad E[(X(n)-n\mu_z)^2] = n\sigma_z^2 \tag{2.109}$$

The process $\{X(n)\}$ is called a *random walk* since the increment $X(n) - X(n-1) = Z(n)$ at time instant n is not correlated with increments at any other time instant. This process is also called a *Wiener process*. •

Example 2.5. Second-order Markov or AR(2) process

The AR(2) process of zero mean is generated by

$$X(n) = Z(n) + b_1 X(n-1) + b_2 X(n-2) \tag{2.110}$$

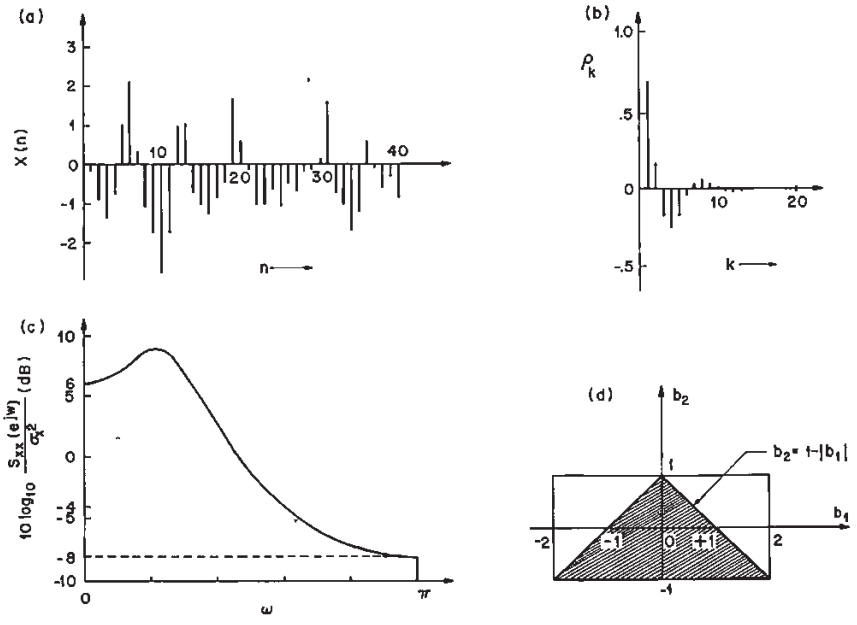
i.e., by passing white noise through a second-order all-pole filter with transfer function

$$H(z) = (1 - b_1 z^{-1} - b_2 z^{-2})^{-1} \tag{2.111}$$

It can be shown that the impulse response and acf can both be described by oscillatory terms with exponentially damped amplitudes. Figure 2.17 is an example with $b_1 = 1.0$, $b_2 = -0.5$. It shows (a) an illustrative realization of $\{X(n)\}$, (b) the acf and (c) the psd. Note that unlike with an AR(1) process, it is now possible to produce spectra with a peak or a trough at an intermediate angular frequency. The psd is given (Problem 2.15) by

$$S_{xx}(e^{j\omega}) = \sigma_z^2 |H(e^{j\omega})|^2 = \frac{1}{1+b_1^2+b_2^2-2b_1(1-b_2)\cos\omega-2b_2\cos 2\omega}\sigma_x^2 \tag{2.112}$$

Figure 2.17 Second-order Markov or AR(2) process: (a) typical sequence $x(n)$; (b) acf ρ_k [Jenkins and Watts, 1968. Reprinted with permission]; (c) psd $S_{xx}(e^{j\omega})$ for $b_1 = 1.0$ and $b_2 = -0.5$; and (d) admissible regions for b_1 and b_2.



It has an extremum at a frequency $\omega = \arccos [b_1(b_2-1)/(4b_2)]$; the extremum is a maximum if $b_2 < 0$ (Problem 2.15). The roots of the denominator polynomial [the poles $z_{\infty 1,2}$ of $H(z)$] may be real or conjugate complex. The filter is stable if the poles are inside the unit circle ($|z_{\infty 1,2}| < 1$). It is not difficult to show that b_1 and b_2 are bounded by $b_2 + b_1 < 1$; $b_2 - b_1 < 1$, and $|b_2| < 1$. Figure 2.17(d) shows this stability region.

The acf values, obtained from (2.94) and (2.95), satisfy

$$\rho_k = b_1\rho_{k-1} + b_2\rho_{k-2}\,; \qquad k > 0 \tag{2.113}$$

Therefore, since $\rho_0 = 1$ and $\rho_{-1} = \rho_1$, we have

$$\rho_1 = \frac{b_1}{1-b_2}\,; \qquad \rho_2 = b_2 + \frac{b_1^2}{1-b_2} \tag{2.114}$$

Note that all acf-values ρ_k; $k > 2$ can be determined recursively. Also, (2.114) can be solved for the AR coefficients:

$$b_1 = \frac{\rho_1(1-\rho_2)}{1-\rho_1^2}\,; \qquad b_2 = \frac{\rho_2 - \rho_1^2}{1-\rho_1^2} \tag{2.115}$$

Figure 2.17(d) has already shown the admissible region for these filter coefficients for stability. We may also calculate the admissible region for ρ_1 and ρ_2 from it. This will be the same as the admissible region for *positive semi-definiteness* of $\mathbf{R}_{xx}$, in Figure 2.13. For the example of Figure 2.17 ($b_1 = 1.0$, $b_2 = -0.5$), (2.114) shows that $\rho_1 = 2/3$ and $\rho_2 = 1/6$. Also, using (2.90), the spectral flatness measure is $\gamma_x^2 = 0.417$ (Problem 2.15). An AR(1) process with the same $\rho_1 = 2/3$, on the other hand, would have $\gamma_x^2 = 0.55$. •

The importance of AR processes lies in the fact that a wide variety of acf's (and psd's) can be produced from them, for modeling stationary processes. The AR(1) model is a good analytical tool for first-order prediction analysis; the AR(2) provides a good fit to the long-time-averaged second-order statistics of speech (damped sinusoidal acf and peaked psd); and AR(N) models with $N \geqslant 10$ are appropriate for adequate short-term descriptions of voiced speech [Markel and Gray, 1976] [Flanagan et al., 1979]. This is seen in the AR(N) simulations of a voiced speech spectrum in Figure 2.18, with $N = 6$, 8, 10 and 12. The five curves in the figure have been vertically separated out for clarity. The separation does *not* indicate differences in average power level across the different curves. The actual spectrum, the uppermost characteristic in the figure, includes the fine structure due to pitch periodicity. The characteristic also exhibits four peaks in spectral envelope, indicated by dark triangular spikes, and these peaks are well reflected in the AR(N) model spectra with $N = 10$ and 12. It will be interesting to compare the spectra of Figure 2.18 with that of Figure 2.10(a). In that spectrum, the zeros are more pronounced than the poles. This has to do with the fact that the Fourier transform operation used to obtain that spectrum is a *finite impulse response (all-zero)* operation, as implied in the transfer function (2.116) to be mentioned presently.

The importance of AR processes will become quite evident in Chapter 6 where DPCM coders and decoders will be matched to the psd of the waveform to be transmitted, by using an AR model of its psd. We will also see there that optimum

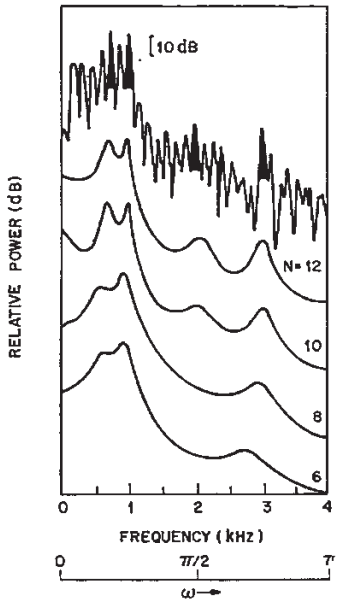


Figure 2.18 AR(N) spectrum models for short-term voiced speech example; N = 6, 8, 10 and 12 [Flanagan et al.;©1979, IEEE].

predictor coefficients $h_{j,opt}$ used for this matching are indeed equal to the autoregressive coefficients b_j. The redundancy of AR(N) waveforms can be completely removed by linear prediction of order N.

2.4.3 Moving Average or All-Zero Processes

A moving average (MA) process is generated by passing a white noise sequence $\{Z(n)\}$ with psd $S_{zz}(z) = \sigma_z^2$ through a filter with an *all-zero* transfer function

$$A(z) = \sum_{j=0}^{M} a_j z^{-j} \tag{2.116}$$

The psd of the generated process $\{X(n)\}$ is

$$S_{xx}(z) = \sigma_z^2 A(z) A(z^{-1}) \tag{2.117}$$

The MA(M) process is generated by the difference equation

$$X(n) = \sum_{j=0}^{M} a_j Z(n-j) \tag{2.118}$$

and the structure of the filter is shown in Figure 2.19. The impulse response of the all-zero filter is

0084

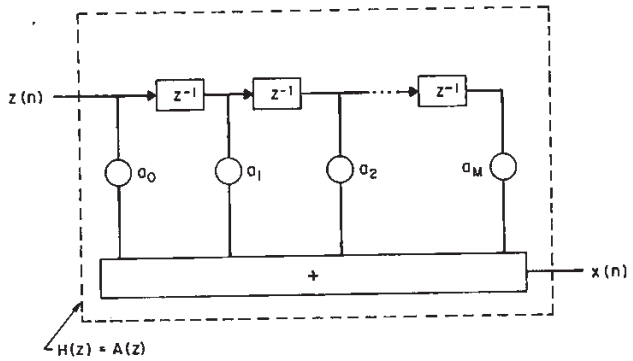


Figure 2.19 Generation of a MA(M) process.

$$h(n) = \begin{cases} a_n & \text{for } n = 0,1,\ldots,M \\ 0 & \text{otherwise} \end{cases} \tag{2.119}$$

The acf of the moving average process (Problem 2.18) will therefore be

$$R_{xx}(k) = \begin{cases} \sigma_z^2 \sum_{j=0}^{M-|k|} a_j a_{j+k} & \text{for } |k| \leqslant M \\ 0 & \text{otherwise} \end{cases} \tag{2.120}$$

$\{X(n)\}$ is a stationary process if the filter is stable, i.e., if its impulse response is absolutely summable:

$$\sum_{j=0}^{M} |a_j| < \infty \tag{2.121}$$

Note the important fact that for finite M, the impulse response is finite and therefore the acf is also finite.

Example 2.6. MA(1) process

The MA(1) process with zero mean and $a_0 = 1$ is generated by

$$X(n) = Z(n) + a_1 Z(n-1) \tag{2.122}$$

i.e., each output of the two-tap filter is a weighted sum of the actual input sample and its predecessor. It is easy to verify that

$$\sigma_x^2 = R_{xx}(0) = \sigma_z^2(1 + a_1^2) \tag{2.123}$$

$$\rho_{xx}(1) = \frac{a_1}{1 + a_1^2} \quad \text{and} \quad \rho_{xx}(k) = 0 \;\; ; \;\; |k| > 1 \tag{2.124}$$

0085

Note that (2.122) causes correlations between neighboring samples, and that unlike in the AR process we have to solve a *nonlinear* equation to obtain a_1 from a specified $\rho_{xx}(1)$. Also unlike in an AR(1) process, the requirement of zero $\rho_{xx}(k)$ for $|k|>1$ in (2.124) is reflected in constraints on allowable $\rho_{xx}(1)$. In fact, by setting $\rho_{xx}(2) = \rho_2 = 0$ in Figure 2.13, we see that $\rho_{xx}(1) \leqslant 1/\sqrt{2}$ (also see Problem 2.16). Note that (2.124) signifies a matrix

$$\mathbf{R}_{xx} = \sigma_x^2 \begin{bmatrix} 1 & \rho_1 & 0 & \cdot & \cdot & 0 \\ \rho_1 & 1 & \rho_1 & \cdot & \cdot & \cdot \\ 0 & \rho_1 & 1 & \cdot & \cdot & \cdot \\ \cdot & & & & & \cdot \\ \cdot & & & & 1 & \rho_1 \\ 0 & & & & \rho_1 & 1 \end{bmatrix} \tag{2.125}$$

From (2.68), the psd is

$$S_{xx}(e^{j\omega}) = \sigma_x^2(1 + a_1^2 + 2a_1\cos\omega) \tag{2.126}$$

If $a_1 = 1$ *(sum filter)*, the psd is a simple *raised cosine, lowpass* characteristic. If $a_1 = -1$ *(difference filter)*, the spectrum is *highpass*.

The correlation matrix $\mathbf{R}_{xx}$ of order L can be used to determine its eigenvalues:

$$\lambda_k = S_{xx}(e^{j\Omega_k}); \qquad k = 0,1,\ldots,L-1 \tag{2.127}$$

$$\Omega_k = \pi\,\frac{k+1}{L+1}; \qquad k = 0,1,\ldots,L-1 \tag{2.128}$$

i.e., the eigenvalues are equally spaced samples of the psd in this specific example. The eigenvectors have sampled sine-wave values as its elements:

$$\mathbf{l}_k = \{\sin(n+1)\Omega_k\}_{n=0,1,\ldots,L-1}; \qquad k=0,1,\ldots,L-1 \tag{2.129a}$$

From (2.91) and (2.126) the *sfm* of the MA(1) process is

$$\gamma_x^2 = (1 + a_1^2)^{-1} \tag{2.129b}$$ •

Constraints on acf values exist for higher-order MA processes as well. For example, the admissible region for (ρ_1, ρ_2) in a MA(2) process with $\rho_{xx}(k) = 0$; $|k|>2$, is a subset of the (ρ_1, ρ_2) admissible region of Figure 2.13, and its coefficients a_1 and a_2 have the same admissible region as b_1 and b_2 in Figure 2.17(d). The *sfm* of the MA(2) process is given in Problem 2.17.

2.4.4 First-Order Models for Two-Dimensional Sources

Consider a random field in which the mean does not depend on spatial coordinates, i.e.,

$$\mu_{x(m,n)} = \mu_x = \textit{constant} \quad \text{for all } m,n \tag{2.130}$$

and in which the acf is translation-invariant. The acf can then be written as a function of two variables

$$R_{xx}(k_h,k_v) = E[X(m,n)X(m+k_h,n+k_v)] \tag{2.131}$$

where h and v refer to *horizontal* and *vertical* directions. Random fields which obey the above two equations are called *homogeneous*. They are *wide-sense-stationary* random fields with a psd-acf description

$$S_{xx}(e^{j\omega_1},e^{j\omega_2}) = \sum_{k_h}\sum_{k_v} R_{xx}(k_h,k_v)e^{-j\omega_1 k_h}e^{-j\omega_2 k_v} \tag{2.132a}$$

$$R_{xx}(k_h,k_v) = \left(\frac{1}{2\pi}\right)^2 \int_{-\pi}^{\pi}\int_{-\pi}^{\pi} S_{xx}(e^{j\omega_1},e^{j\omega_2})e^{j\omega_1 k_h}e^{j\omega_2 k_v}d\omega_1 d\omega_2 \tag{2.132b}$$

The interpretation of psd is very similar to that for 1-D signals. For example, if all values of $R_{xx}(k_h,k_v)$ are zero except $R_{xx}(0,0)$, the psd is white in both spatial directions:

$$R_{xx}(k_h,k_v) = \sigma_x^2\delta(k_h,k_v)\ ; \qquad S_{xx}(e^{jw_1},e^{jw_2}) = \sigma_x^2 \tag{2.133}$$

According to experimental data, the acf of a large variety image data follow one of two models — the *separable* model appropriate especially to many artificial or man-made images; and the *isotropic* model appropriate especially to images of natural objects [Pratt, 1978]. Using earlier nomenclature, we shall discuss corresponding discrete-space models as *discrete-time* models.

The Separable acf Model. In continuous coordinates, an important acf model is the stationary first-order Markov model

$$R_{xx}(k'_h,k'_v) = \sigma_x^2 \exp(-\alpha|k'_h|-\beta|k'_v|) \tag{2.134}$$

where k'_h and k'_v are the spatial shifts in horizontal and vertical image directions and α and β describe respective amounts of correlation. The model (2.134) is called *separable* because of the absence of terms like $k'^p_h k'^q_v$ $(p,q \neq 0)$ in the acf expression (2.134). Let us use a spatially discretized description, with

$$\rho_h = e^{-\alpha}\ ; \qquad \rho_v = e^{-\beta} \tag{2.135}$$

Setting (2.135) in (2.134), and using integer-valued variables k_h and k_v,

$$R_{xx}(k_h,k_v) = \sigma_x^2\, \rho_h^{|k_h|}\, \rho_v^{|k_v|} \tag{2.136}$$

Adjacent rows and columns in a square grid of pixels (Figure 2.20) are separated by unit distances $k_h = 1$ and $k_v = 1$, respectively. Note the special case

$$R_{xx}(k_h,k_v) = \sigma_x^2\rho^{|k_h|+|k_v|} \quad \text{if} \quad \rho_v = \rho_h = \rho\ , \tag{2.137a}$$

which is a generalization of the 1-D expression in (2.102).

From (2.137a), the normalized correlation between *diagonal neighbors* [case of $k_h = 1$, $k_v = 1$ in (2.137a) as well as in Table 2.3] is

0087

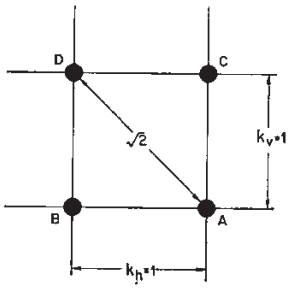


Figure 2.20 A (2 × 2) grid of pixels.

$$\rho_{AD} = \rho_{BC} = \rho_{xx}(1,1) = \rho^{1+1} = \rho^2 \tag{2.137b}$$

whereas *horizontal and vertical neighbors* have the correlations

$$\rho_{AB} = \rho_{AC} = \rho_{CD} = \rho_{BD} = \rho^{0+1} = \rho^{1+0} = \rho \tag{2.137c}$$

The separable model is matched to images wherein contours of equal autocorrelation are rectangles or squares with horizontal and vertical edges, as in Figure 2.21(a).

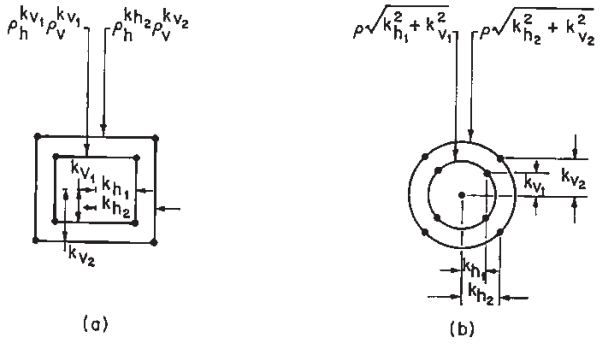
A discrete random field with a separable acf and with equal correlations in the two spatial directions can be represented by the difference equation

$$X(m,n) = b\,X(m,n-1) + c\,X(m-1,n) + d\,X(m-1,n-1) + Z(m,n) \tag{2.138}$$

$$b = c = \rho\,; \quad d = -\rho^2\,; \quad |\rho| \leqslant 1 \tag{2.139}$$

where $\{(Z(m,n)\}$ is a 2-D zero-mean random field of uncorrelated r.v.'s with common variance σ_z^2. This yields a source variance (Problem 2.19)

Figure 2.21 Constant-acf contours in (a) separable and (b) circularly symmetric (isotropic) 2-D Markov processes.



0088

$$\sigma_x^2 = \sigma_z^2\,(1-\rho^2)^{-2} \tag{2.140a}$$

The recursive formula (2.138) can also be interpreted as the output of a 2-D recursive filter with $\{Z(m,n)\}$ as input. Its transfer function $H(e^{j\omega_1}, e^{j\omega_2})$ is given in (A.42), and conditions for its stability are given in Example 6.9. Using earlier nomenclature, $X(m,n)$ in (2.138) is an *autoregressive* or *Markov* process, a generalization of the 1-D process (2.96). From (2.140a), the *sfm* of the process (Problem 2.19) is

$$\gamma_x^2 = (1-\rho^2)^2 \tag{2.140b}$$

The Isotropic acf Model. The acf model

$$R_{xx}(k'_h,k'_v) = \sigma_x^2\exp(-\alpha k') = \sigma_x^2\exp[-\alpha(k'^2_h + k'^2_v)^{1/2}] \tag{2.141}$$

describes, with continuous co-ordinates, a 2-D acf model whose spatial behavior is describable by a single Euclidean distance term k'. The rotational symmetry of the model suggests that it is appropriate [O'Neal and Natarajan, 1977] for natural objects with no preferred orientation. The function in (2.141) is called *isotropic*; unlike (2.134), it is *non-separable*. Using discrete-space co-ordinates k_h and k_v for horizontal and vertical separation, and with

$$\exp(-\alpha) = \rho, \tag{2.142}$$

$$R_{xx}(k_h,k_v) = \sigma_x^2\,\rho^{(k_h^2+k_v^2)^{1/2}} \tag{2.143}$$

Contours of equal autocorrelation are circles, as in Figure 2.21(b). Normalized values of adjacent pixel correlations (Figure 2.20) are

$$\rho_{AB} = \rho_{CD} = \rho_{AC} = \rho_{BD} = \rho_{xx}(1,0) = \rho_{xx}(0,1) = \rho \tag{2.144}$$

and the diagonal neighbor correlation (Figure 2.20) is

$$\rho_{BC} = \rho_{AD} = \rho_{xx}(1,1) = \rho^{\sqrt{2}} \tag{2.145}$$

2.4.5 Models in the Continuous-Time Domain

Most communication signals are originally continuous-time signals; sampling leads to discrete-time sequences which are almost always assumed in this book. However, we will sometimes need a description of the original continuous-time process as well. One example is the traditional modeling of a correlated source by the output of a resistor-capacitor network driven by white noise (Chapter 8). A second application of continuous-time descriptions is in the discussion of the effects of bandpass filtering and sampling on analog signal reconstruction (Chapter 3).

The autocorrelation function (acf) of a stationary, continuous-time random process $\{X(t)\}$ is

$$R_{xx}(\tau) = E[X(t)X(t+\tau)] = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} xy\, p_{x(t)x(t+\tau)}(x,y)\,dxdy \qquad (2.146)$$

where τ is the time lag. The normalized acf is $\rho_{xx}(\tau) = R_{xx}(\tau)/R_{xx}(0)$, and the properties for acf's of discrete-time processes apply also to the continuous-time case. The psd is

$$S_{xx}(\Omega) = \int_{-\infty}^{\infty} R_{xx}(\tau)e^{-j\Omega\tau}d\tau \qquad (2.147)$$

provided that $R_{xx}(\tau)$ is absolutely integrable. The inverse operation yields

$$R_{xx}(\tau) = \frac{1}{2\pi}\int_{-\infty}^{\infty} S_{xx}(\Omega)e^{j\Omega\tau}d\,\Omega \qquad (2.148)$$

The mean power in the process is

$$\sigma_x^2 = R_{xx}(0) = E[X^2(t)] = \frac{1}{2\pi}\int_{-\infty}^{\infty} S_{xx}(\Omega)d\,\Omega \qquad (2.149)$$

Many techniques given in the literature on Digital Signal Processing can be applied if a discrete-time analogue of a continuous-time system or signal has to be designed. For example, the continuous time filter that generates an analog spectrum can be transformed into an equivalent digital filter. One approach is to replace the differentials in the generating equation for $X(t)$ with finite differences, in order to obtain a difference equation that approximates the differential equation. Another possibility is to make use of the *impulse-invariant transformation*: here we choose the impulse response of the digital filter as equally spaced samples of the impulse response of the analog filter. In this case, the values of the acf $\rho_{xx}(n)$ [resulting from passing white noise sequences through a digital filter with impulse response $h(n)$] will also be samples of the continuous-time acf $\rho_{xx}(\tau)$. This will be illustrated by the equivalence of the acf's in integrated power spectrum and AR(1) processes, in Example 2.7. It must be emphasized, however, that the psd in the digital domain is never identical with the psd in the analog domain. This is due to aliasing effects (Chapter 3) and the use of samples of the continuous-time acf whose Fourier transform (the psd) may not be bandlimited.

Example 2.7. Integrated power spectra

A widely used approximation to many spectra with *lowpass* character (i.e., with a main power contribution at low frequencies) is the *integrated power spectrum* obtained at the output of a resistor-capacitor integrator network with a transfer function of the form

$$H(j\,\Omega) = \frac{1}{1 + j\,\Omega/\Omega_C} \qquad (2.150)$$

and a white-noise input $\{Z(n)\}$ of spectral density

$$S_{zz}(\Omega) = \sigma_z^2(2/\Omega_C) \qquad (2.151)$$

0090

$H(j\Omega)$ is also named a *first-order Butterworth* lowpass filter. The output psd is

$$S_{xx}(\Omega) = S_{zz}(\Omega)|H(j\Omega)|^2 = \frac{2}{\Omega_C}\,\frac{\sigma_x^2}{1+(\Omega/\Omega_C)^2} \tag{2.152}$$

Note that the above represents a psd which tends to decrease very slowly with frequency until the *corner frequency* Ω_C, after which the psd falls rapidly with frequency, with an asymptotic decay of 6 dB/octave [Figure 2.22(b)(i)].

Using (2.148) and the standard integral

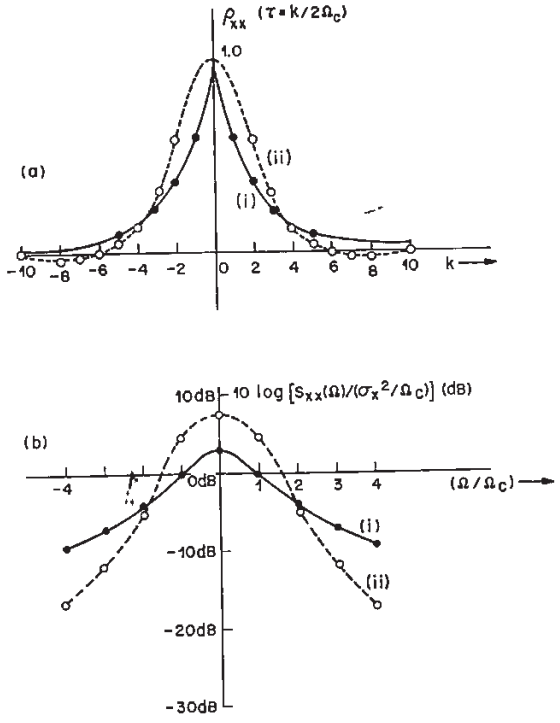
$$\int_0^\infty \frac{\cos ax}{1+x^2}\,dx = \frac{\pi}{2}\,e^{-|a|}, \tag{2.153}$$

we obtain an exponential acf

$$R_{xx}(\tau) = \sigma_x^2 \exp(-\Omega_C|\tau|)\,; \qquad \rho_{xx}(\tau) = \exp(-\Omega_C|\tau|) \tag{2.154}$$

That is, the spectral density is that of a process $\{X(n)\}$ whose acf has a Laplacian (two-sided) exponential characteristic [Figure 2.22(a)(i)]. The frequency Ω_C is the -3 dB point

Figure 2.22 Plots of (a) acf and (b) psd for (i) the integrated power spectrum with a 6 dB per octave roll-off, and (ii) a spectrum with 12 dB per octave roll-off.



0091

in the psd: $S_{xx}(\Omega_C) = S_{xx}(0)/2$. The acf $R_{xx}(\tau)$ could alternatively be obtained from a convolution of white noise acf and filter-acf, i.e.,

$$R_{xx}(\tau) = R_{zz}(\tau) * h(\tau) * h(-\tau); \qquad h(\tau) = \Omega_C \exp(-\Omega_C \tau); \qquad \tau \geqslant 0 \tag{2.155}$$

We note in passing that random processes which are generated by passing a white noise process through a linear system are called *linear processes*. It can be shown that the generated process is stationary with finite variance if the impulse response is absolutely integrable. In our example, the linear process $\{X(t)\}$ can also be defined by a differential equation of the form

$$\frac{1}{\Omega_C}\frac{d}{dt}X(t) + X(t) = Z(t) \tag{2.156}$$

which is a *continuous-time autoregressive* or *first-order Markov process.*

We will use the impulse-invariant transformation mentioned earlier to show that the integrated power spectrum process is indeed equivalent to the discrete-time Markov AR(1) process. The variance-normalized samples $\rho_{xx}(kT)$ from (2.154) define a normalized acf sequence

$$\rho_{xx}(kT) = \exp(-\Omega_C |k| T) \tag{2.157}$$

Let ρ denote the first value of this sequence as in the discrete-time case, i.e., $\rho_{xx}(T) = \rho$. It follows immediately that

$$\rho_{xx}(kT) = \rho^{|k|} \tag{2.158}$$

as in the acf of the AR(1) process (2.103).

Note, however, that the psd $S_{xx}(\Omega)$ cannot be reconstructed from samples $\rho_{xx}(kT)$ since it is not bandlimited; the discrete-time model can only approximate the psd $S_{xx}(\Omega)$. •

Example 2.8. Bandlimited integrated power spectrum

The integrated power spectrum (2.152) is not bandlimited, although its power density decreases at 6 dB/octave at frequencies well above the −3 dB point. If a signal with this spectrum is to be the input to a discrete-time system, the signal has to be bandlimited to an upper frequency Ω_W which is no higher than the Nyquist frequency of the discrete-time system. Therefore the psd has to be multiplied by rect $(\Omega/2\Omega_W)$. The resultant acf is a convolution of the acf of (2.154) with $\Omega_W \operatorname{si}(\Omega_W \tau)/\pi$. An approximate result [O'Neal, 1966, I] for the normalized acf of the bandlimited signal is

$$\rho_{xx}(\tau) = \frac{\exp(-\Omega_C\tau) - \dfrac{2\Omega_C\tau}{\pi}\left[\dfrac{\cos\Omega_W\tau}{\Omega_W\tau} + \operatorname{Si}(\Omega_W\tau) - \dfrac{\pi}{2}\right]}{1 - \dfrac{2}{\pi}\dfrac{\Omega_C}{\Omega_W}} \quad \text{for} \quad \begin{matrix}\tau \to 0 \text{ and} \\ \Omega_C \ll \Omega_W\end{matrix}; \tag{2.159}$$

$$\operatorname{Si}(x) = \frac{\pi}{2} - \int_x^\infty \frac{\sin t}{t}\,dt = \frac{\pi}{2} + \operatorname{si}(x) \tag{2.160}$$

$\operatorname{Si}(x)$ is the *sine integral function.* The non-exponential term in (2.159) has a strong influence on the acf, especially at small values of the lag τ. Such values are particularly

0092

important in discussing the effect of oversampling a signal; the correlations between samples are then given by values $\rho_{xx}(\tau)$ close to $\tau = 0$ (Chapter 8). Table 2.4 illustrates the modeling [O'Neal, 1966, I] of three communication signals with a bandlimited integrated power spectrum. The numerical values of Ω_W and Ω_C in rows 3 and 1 of the table correspond closely with the values in the lowpass spectra of Figure 2.9.

Table 2.4 Integrated power spectrum modeling of three sources (Ω_W and Ω_C denote cutoff and corner frequencies).

Signal	$\psi = \Omega_W/\Omega_C$	$\Omega_W/2\pi$ (kHz)	$\Omega_C/2\pi$ (kHz)
Black- and-White TV	$\frac{2\pi}{.068}$	4500	49
Picturephone	$\frac{2\pi}{0.25} \sim 25$	1000	40
Telephone Speech	4	3.2	0.8

The parameter ψ is the ratio of cutoff frequency Ω_W to the corner frequency Ω_C. The numerical values of ψ in Table 2.4 will be utilized in Chapter 8 to characterize the performance of deltamodulation coders for speech and images.

The bandlimited signals above can be Nyquist-sampled at $T = f_s^{-1} = \pi/\Omega_W$. Let us compare acf values from (2.154) and (2.159) at $\tau = T$. For the picturephone example of Table 2.4, $\rho_{xx}(T) = \exp(-0.125) = 0.883$ using (2.154), and $\rho_{xx}(T) = 0.907$ using (2.159). The increase in the acf value reflects the higher relative contributions of low frequency components in the bandlimited case. One of the most significant consequences of the bandlimited model is that

$$\lim_{\tau \to 0} \rho_{xx}(\tau) \to 1 - \beta\tau^2; \qquad \beta = \Omega_W \Omega_C/\pi \tag{2.161}$$

This formulation will be very useful in prediction error analysis in differential coders, especially those involving input *oversampling*, so that $T << \pi/\Omega_W$ (Chapter 8). •

Example 2.9. 12 dB spectrum (second-order Butterworth characteristic)

Another spectrum of interest is one that falls off with a slope of 12 dB/octave at high frequencies. Its psd

$$S_{xx}(\Omega) = \frac{4\sqrt{2}\sigma_x^2}{\Omega_C} \frac{1}{1 + (\Omega/\Omega_C)^4} \tag{2.162}$$

is obtained by filtering white noise of variance $4\sqrt{2}\sigma_x^2/\Omega_C$ with a second-order Butterworth filter whose amplitude-squared frequency response is $[1 + (\Omega/\Omega_C)^4]^{-1}$.

As in the case of the integrated power spectrum, Ω_C is the 3 dB corner frequency. For speech, $\Omega_C/2\pi = 800$ Hz is a reasonable value (Figure 2.9). The acf of (2.162) is

$$\rho_{xx}(\tau) = \sqrt{2}\exp(-\Omega_C|\tau|/\sqrt{2})\sin\left[\frac{\pi}{4} + \Omega_C|\tau|/\sqrt{2}\right] \tag{2.163}$$

The psd and acf functions in (2.162) and (2.163) are shown along with those of the integrated power spectrum [equations (2.152) and (2.154)] in Figure 2.22. The relatively higher contributions of low frequency components in the 12 dB spectrum are reflected by higher values of acf for small values of delay k. Comparing these with statistics measured on speech (Figure 2.9a), one notes that neither of the two models predicts the non-monotonic nature of the speech psd. But in the frequency range of Ω_C to Ω_W (800 to 3200 Hz), the 6 dB per octave model is a better descriptor of average speech psd. On the other hand, a comparison with the LPF speech acf (Figure 2.7a) shows that the damped-cosinusoidal property (implying the presence of negative acf values) is better matched by the acf of the 12 dB model.

Bandlimitation does not affect the general shape of the acf as much as in the case of the 6 dB/octave psd—except, however, for extremely small values of time lag τ which will be of interest in oversampling (Chapter 8). •

Example 2.10. Bandlimited white spectrum

Consider a spectrum that is uniformly distributed over a specified range (Ω_l, Ω_u) and zero outside. It is obtained by passing a white signal through a bandpass filter with center frequency $\Omega' = (\Omega_l + \Omega_u)/2$, and bandwidth $2\Delta\Omega = (\Omega_u - \Omega_l)$. The psd and acf are

$$S_{xx}(\Omega) = \begin{cases} \pi\sigma_x^2/\Delta\Omega & \text{for } \Omega_l \leqslant |\Omega| \leqslant \Omega_u \\ 0 & \text{otherwise} \end{cases} \tag{2.164}$$

$$\rho_{xx}(\tau) = \cos\left[(\Omega_l + \Omega_u)\tau/2\right] \text{ si } (\Delta\Omega\tau) = \cos\,\Omega'\tau \text{ si}(\Delta\Omega\tau)$$

Non-bandlimited *white noise* results if $\Delta\Omega \to \infty$:

$$S_{xx}(\Omega) = \text{constant for all } \Omega\,; \qquad \rho_{xx}(\tau) = \delta(\tau) \tag{2.165}$$

No two samples of this process have non-zero correlation, but the non-bandlimited process is unrealizable since its power is infinite. •

Example 2.11. Raised cosine spectrum

A good model for voiceband data signals or modem waveforms is that described by a *raised cosine* psd:

$$S_{xx}(\Omega) = \begin{cases} \dfrac{\pi}{\Delta\Omega}\,\sigma_x^2\left[1 + \cos 2\pi\,\dfrac{|\Omega| - \Omega'}{2\Delta\Omega}\right] & \text{for } \Omega' - \Delta\Omega \leqslant |\Omega| \leqslant \Omega' + \Delta\Omega \\ 0 & \text{otherwise} \end{cases} \tag{2.166}$$

$$\rho_{xx}(\tau) = \frac{\cos\dfrac{\Delta\Omega\tau}{2}\cos\,\Omega'\tau}{1 - \left(\dfrac{\Delta\Omega\tau}{\pi}\right)^2} \cdot \text{si}\left(\frac{\Delta\Omega\tau}{2}\right) \tag{2.167}$$ •

The spectrum models discussed in Examples 2.10 and 2.11 are sketched in Figure 2.23, together with a Gaussian psd (Problem 2.21) whose center frequency is Ω', and whose four-sigma point equals the half-bandwidth $\Delta\Omega$. The psd's are

0094

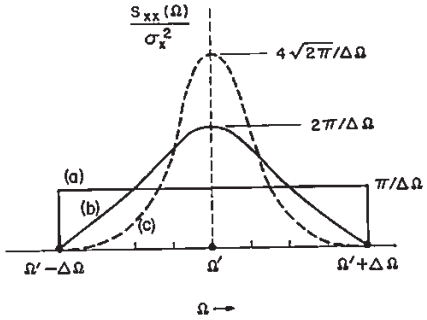


Figure 2.23 Positive-frequency halves of three psd's of equal variance σ_x^2: (a) bandlimited white- and (b) raised-cosine psd's with half-bandwidth $\Delta\Omega$; and (c) Gaussian psd with a four-sigma point of $\Delta\Omega$.

scaled such that all processes have identical variance σ_x^2. Note that these spectra are *non-lowpass* and *non-speech-like*. Digital coders designed for speech-like spectra consequently perform poorly when actual input spectra resemble those of Figure 2.23. This is the situation, for example, if voice-band data signals (analog modem waveforms, typically with a raised cosine psd) are encountered as inputs in switched telephone networks, and they are to be digitized using waveform coders designed for speech transmission.

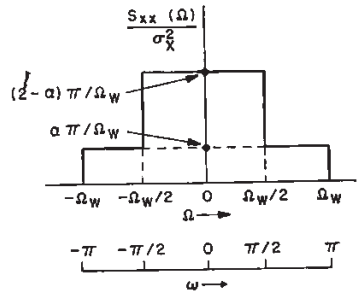
Example 2.12. A highly structured, two-level psd

Figure 2.24 shows a *two-level* psd

$$S_{xx}(\Omega) = \begin{cases} (\pi/\Omega_W)(2-\alpha)\sigma_x^2 & \text{for } 0 \leqslant |\Omega| \leqslant \Omega_W/2 \\ (\pi/\Omega_W)\alpha\sigma_x^2 & \text{for } \Omega_W/2 \leqslant |\Omega| \leqslant \Omega_W \\ 0 & \text{for } |\Omega| > \Omega_W \end{cases} \tag{2.168}$$

with $\alpha \leqslant 1$. The psd has a variance σ_x^2 and an *sfm* given by $\gamma_x^2 = [\alpha(2-\alpha)]^{1/2} \sim \sqrt{2\alpha}$ if $\alpha \to 0$ (Problem 2.22). The special case of $\alpha = 1$ is simply the bandlimited white psd (with $\Omega_W = 2\Delta\Omega$). We shall see in Chapter 11 that sub-band coding with a two-filter bank is adequate to remove all the redundancy in this input process. •

Figure 2.24 A highly structured two-level psd; bandlimited white spectra result when $\alpha = 0$ or 1.



0095

Problems

(2.1) Let $X = A \sin \phi$, where the phase ϕ has a uniform pdf $p(\phi) = \frac{1}{2\pi}$ in the interval I: $(0, 2\pi)$.

(a) Show that the distribution of X is the arcsine pdf $\frac{1}{\pi\sqrt{A^2 - x^2}}$ for $|x| \leqslant A$.

(b) Show that $\mu_x = 0$ and $\sigma_x^2 = A^2/2$.

(2.2) The pdf of an image signal can be approximately described by a uniform pdf in the interval $(0,a)$. In some differential coding systems, one considers the difference between neighboring samples as the message to be transmitted. Show that the pdf of this difference signal would be triangular in the interval $(-2a, 2a)$ if the samples of the waveform were statistically independent.

(2.3) Assume two statistically independent zero-mean Gaussian r.v.'s X and Y. Show that $Z = X+Y$ has the pdf $N(0,\sigma_x^2 + \sigma_y^2)$.

(2.4) Refer to the bivariate normal pdf (2.20).

(a) Show that marginal pdf's $p_x(\cdot)$ and $p_y(\cdot)$ are those of $N(0,\sigma_x^2)$ and $N(0,\sigma_y^2)$ r.v.'s, respectively.

(b) Show by computing appropriate expectations that
(i) $E[X^2] = \sigma_x^2$
(ii) $E[Y^2] = \sigma_y^2$
(iii) $\rho_{xy} = E[XY]/\sigma_x\sigma_y$

(c) Verify that the r.v.'s X and Y are statistically independent $[p_{xy}(x,y) = p_x(x)p_y(y)]$ if they are uncorrelated $(\rho_{xy} = 0)$.

(d) Show that the conditional pdf of X, assuming that $Y = y$, is

$$p_x(x|Y=y) = \frac{1}{\sqrt{2\pi(1-\rho_{xy}^2)}\sigma_x} \exp\left[-\frac{1}{2\sigma_x^2}(1-\rho_{xy}^2)^{-1}\left(x - \frac{\sigma_x}{\sigma_y}\rho_{xy}y\right)^2\right]$$

[Papoulis, 1965], which is the pdf of a Gaussian r.v., of mean $(\sigma_x/\sigma_y)\rho_{xy}y$, and variance $(1 - \rho_{xy}^2)\sigma_x^2$. Note that its mean depends linearly on the given value of y. The conditional expectation of x is simply $(\sigma_x\rho_{xy}/\sigma_y)y$, which in turn is simply $\rho_{xy}y$ if $\sigma_x = \sigma_y$.

(2.5) Show that the pdf of the sum of n identical independent r.v.'s with a uniform pdf in the range $(-A,A)$ approaches a $N(0,nA^2/3)$ pdf in the limit of large n. For finite n, the approximation is, of course, only valid for values well within the range $(-nA,nA)$ because the sum cannot exceed this range [Bennett, 1956]. (Hint: For independent r.v.'s, the characteristic function of the sum is the product of individual characteristic functions.)

(2.6) Let the real and imaginary components u and v of a complex r.v. (e.g., transform coefficient of Chapter 12) be zero-mean Gaussian variables with identical pdf's $N(0,\sigma^2)$. Show that the magnitude $x = \sqrt{u^2 + v^2}$ has the *Rayleigh* pdf $p(x) = (x/\sigma^2) \exp(-x^2/2\sigma^2)$, with $\mu_x = \sigma\sqrt{\pi/2}$, $\sigma_x^2 = (2-\pi/2)\sigma^2$. This pdf is sometimes used as a model for image amplitudes, instead of the simpler uniform pdf.

0096

(2.7) Show that (2.46) implies the following conditions for $N = 2,3$:

(a) $N = 2$: $|\rho_1| < 1$

(b) $N = 3$: $|\rho_1| < 1$; $|\rho_2| < 1$; $\rho_1^2 < (1+\rho_2)/2$

(2.8) Verify the results of (2.30) and (2.31).

(2.9) Show that the acf of $x(n) = \sin(2\pi fn)$ is cosinusoidal, also with frequency f. Use (2.3), $N \to \infty$, and assume a sampling rate that is an integral multiple of f.

(2.10) Consider a linear, time-invariant, and stable system with impulse response $h(n)$ and a stationary random process $\{Z(n)\}$ of finite mean and variance applied to its input.

(a) Show that the mean value of the output random process is

$$\mu_x = \mu_z \sum_{n=-\infty}^{\infty} h(n) = \mu_z H(0)$$

(b) Use the simple case of $\mu_z = 0$ to show that

$$R_{xx}(k) = R_{zz}(k) * h(k) * h(-k) = \sum_{i=-\infty}^{\infty} \sum_{j=-\infty}^{\infty} h(i)h(j)R_{zz}(|k-i+j|)$$

$$S_{xx}(e^{j\omega}) = S_{zz}(e^{j\omega})\ |H(e^{j\omega})|^2$$

(c) Hence show that if the input $\{Z(n)\}$ is a white process,

$$R_{xx}(k) = \sigma_z^2 \sum_{j=-\infty}^{\infty} h(j)h(j+k) \ ; \quad \sigma_x^2 = \sigma_z^2 \sum_{j=0}^{\infty} h^2(j)$$

(2.11) Let $S_{xx}(e^{j\omega})$ be a rational spectrum, i.e.,

$$S_{xx}(e^{j\omega}) = \sigma_z^2 \frac{\prod_l |e^{j\omega} - z_{ol}|^2}{\prod_l |e^{j\omega} - z_{\infty l}|^2}$$

where σ_z^2 is the variance of the "innovations" white noise process that is fed into the filter with zeroes z_{ol} and poles $z_{\infty l}$. Use the result $\int \log|e^{j\omega} - z|^2 d\omega = 0$ to show that [Bunin, 1969]

$$\gamma_x^2 = \sigma_z^2/\sigma_x^2$$

This is a very important result since it shows that the model sources of Section 2.4 have a value of *sfm* that is simply the ratio of the variances of innovations and source waveforms.

(2.12) Verify that the psd (2.105) for an AR(1) process with $b_1 = \rho_1 = \rho$ has a spectral flatness measure $\gamma_x^2 = (1 - \rho^2)$.

0097

(2.13) Verify that for an AR(1) process [Grenander and Szego, 1958]

$$\mathbf{R}_{xx}^{-1} = \frac{1}{(1-\rho^2)\sigma_x^2}\begin{bmatrix} 1 & -\rho & 0 & . & . & 0 \\ -\rho & (1+\rho^2) & -\rho & 0 & . & 0 \\ . & & & & & . \\ . & & & & (1+\rho^2) & -\rho \\ 0 & . & . & . & -\rho & 1 \end{bmatrix}$$

is indeed the inverse of (2.107).

(2.14) Use the results of Problem (2.13) and (2.79) to show that the optimum predictor of an AR(1) process is simply a first-order predictor. This result will be mentioned again in Chapter 6.

(2.15) Consider the AR(2) process (2.110).

(a) Show that its psd is that in (2.112).

(b) Show that the psd has an extremum at a frequency given by $\omega = \arccos [b_1(b_2-1)/(4b_2)]$, assuming that $b_1^2(1-b_2)^2 < 16b_2^2$; and show that this extremum is a maximum if $b_2 < 0$.

(c) Using (2.90) and (2.91), show that

$$\gamma_x^2 = \frac{(1+b_2)(1-b_1-b_2)(1+b_1-b_2)}{(1-b_2)}$$

(2.16) Define a MA(1) process (2.122) by insisting that

$$R_{xx}(0) = \sigma_x^2 \;\; ; \;\; R_{xx}(\pm 1) = \alpha\sigma_x^2 \;\; ; \;\; R_{xx}(k) = 0 \;\; ; \;\; |k| > 1$$

Calculate its psd and show that we must have $|\alpha| \leq 1/\sqrt{2}$ for a nonnegative determinant $|\mathbf{R}_{xx}|$ of order 3. Note that the above constraint on α is satisfied for any value of a_1 in (2.122).

(2.17) (a) Calculate the acf and psd of the MA(2) process

$$X(n) = Z(n) + a_1 Z(n-1) + a_2 Z(n-2)$$

(b) Show that the spectral flatness measure is

$$\gamma_x^2 = (1 + a_1^2 + a_2^2)^{-1},$$

using the results of (a), *and* by evaluating the power transfer factor α.

(2.18) Show that the general MA process (2.118) has the acf described in (2.120).

(2.19) Show that (2.138) and (2.139) can also be written as

$$X(m,n) = \sum_{i=0}^{\infty}\sum_{j=0}^{\infty} \rho^{i+j} Z(m-i,n-j)$$

which is the 2-D equivalent of (2.97). Show that this random field indeed has the normalized acf $\rho_{xx}(k_h,k_v) = \rho^{|k_h|+|k_v|}$ and that $\sigma_z^2 = \sigma_x^2(1-\rho^2)^2$. [Hint: Use the

0098

simplifying concept of the 2-D z-transform $X(z_1,z_2)$, although this is strictly valid only for a deterministic signal.]

(2.20) The acf values $\rho_{xx}(k_h,k_v)$ in the following table have been measured for an interlaced picturephone image (128 lines and 120 pixels per line) [Habibi and Wintz, 1971]. With $\alpha = 0.124$ and $\beta = 0.110$ in (2.135), and with $\alpha = (0.124 + 0.110)/2 = 0.117$ in (2.142), compute the acf-values of a first-order source with separable and isotropic statistics. Verify that the latter one is a much better model of the above measured statistics, especially for the higher values of k_h and k_v in the table.

Table of measured acf values.

k_v \ k_h	0	1.000	2	3
0	1	0.896	0.808	0.725
1	0.883	0.857	0.793	0.725
2	0.776	0.778	0.748	0.702
3	0.691	0.700	0.692	0.661

(2.21) Consider the Gaussian psd

$$S_{xx}(\Omega) = \frac{\sqrt{2\pi}}{\Omega_\sigma}\,\sigma_x^2 \exp(-\Omega^2/2\Omega_\sigma^2)$$

(a) Show that $\pm\Omega_\sigma$ are the frequencies at which the value of the psd is -2.2 dB relative to its maximum value.

(b) Show that the acf of the process is also Gaussian: $\rho_{xx}(\tau) = \exp(-\Omega_\sigma^2\,\tau^2/2)$.

(2.22) Consider the two-level psd of Figure 2.24 and (2.168).

(a) Verify that the variance is σ_x^2.

(b) Show that the continuous-time version of the spectral flatness measure (2.74) for a psd bandlimited in $(-\Omega_W, \Omega_W)$ is

$$\gamma_x^2 = \frac{\eta_x^2}{\sigma_x^2} = \frac{\dfrac{\Omega_W}{\pi}\exp\dfrac{1}{2\Omega_W}\displaystyle\int_{-\infty}^{\infty}\log_e S_{xx}(\Omega)\,d\,\Omega}{\dfrac{1}{2\pi}\displaystyle\int_{-\infty}^{\infty}S_{xx}(\Omega)\,d\,\Omega}.$$

[Hint: Set $\omega = \pi\Omega/\Omega_W$, $d\omega = \pi d\,\Omega/\Omega_W$ and $S_{xx}(e^{j\pi\Omega/\Omega_W}) = S_{xx}(e^{j\omega}) = S_{xx}(\Omega)\,\Omega_W/\pi$.]

(c) Use the result in (b) to show that the *sfm* of the two-level psd is $[\alpha(2-\alpha)]^{1/2} \sim \sqrt{2\alpha}$ if $\alpha \to 0$.

(d) Derive the same result using (2.74), and the discrete-time version of the psd obtained upon replacing Ω_W by π in Figure 2.24.

(e) Discuss the special cases of $\alpha = 0$ and $\alpha = 1$.

0099

References

M. Abramowitz and I. A. Stegun, *Handbook of Mathematical Functions*, U.S. Dept. of Commerce, National Bureau of Standards, Applied Mathematics Series, June 1964.

B. S. Atal and S. L. Hanauer, "Speech Analysis and Synthesis by Linear Prediction of the Speech Wave," J. Acoust. Soc. Am., pp. 637-655, August 1971.

W. R. Bennett, "Methods of Solving Noise Problems," Proc. IRE, p. 609, May 1956.

T. Berger, *Rate Distortion Theory*, Prentice-Hall, Englewood Cliffs, N.J., 1971.

B. J. Bunin, "Rate Distortion Functions for Gaussian Markov Processes," Bell System Tech. J., pp. 2059-3074, November 1969.

D. J. Connor, R. C. Brainard and J. O. Limb, "Intraframe Coding for Picture Transmission," Proc. IEEE, pp. 779-791, July 1972.

H. K. Dunn and S. D. White, "Statistical Measurements on Conversational Speech," J. Acoust. Soc. Am., pp. 278-288, January 1940.

J. L. Flanagan, *Speech Analysis, Synthesis and Perception*, Springer-Verlag, New York, 1972.

J. L. Flanagan, M. R. Schroeder, B. S. Atal, R. E. Crochiere, N. S. Jayant and J. M. Tribolet, "Speech Coding," IEEE Trans. on Communications, pp. 710-737, May 1979.

U. Grenander and G. Szego, *Toeplitz Forms and Their Applications*, University of California Press, Berkeley, Calif., 1958.

F. R. Grizmala, unpublished work, Bell Laboratories, 1972.

A. Habibi and P. A. Wintz, "Image Coding by Linear Transformation and Block Quantization," IEEE Trans. on Communications, pp. 50-63, February 1971.

N. S. Jayant, unpublished work, Bell Laboratories, 1982.

G. M. Jenkins and D. G. Watts, *Spectral Analysis and Its Applications*, Holden-Day, San Francisco, 1968.

E. R. Kretzmer, "Statistics of Television Signals," Bell System Tech. J., pp. 751-763, July 1952.

T. Kummerow, "Statistics for Efficient Linear and Nonlinear Picture Coding," Int. Telemetering Conf., pp. 149-161, October 1972.

J. I. Makhoul and J. J. Wolf, "Linear Prediction and the Spectral Analysis of Speech," Bolt, Beranek and Newman, Inc. Technical Report, 1972.

J. D. Markel and A. H. Gray, Jr., *Linear Prediction of Speech*, Springer-Verlag, New York, 1976.

R. A. McDonald, "Signal-to-Noise and Idle Channel Performance of Differential Pulse Code Modulation Systems — Particular Applications to Voice Signals," Bell System Tech. J., pp. 1123-1151, September 1966.

P. Noll, "Non-adaptive and Adaptive DPCM of Speech Signals," Polytech. Tijdschr. Ed. Elektrotech./Electron (The Netherlands), No. 19, 1972.

P. Noll, "Adaptive Quantizing in Speech Coding Systems," Int. Zurich Seminar on Digital Commun., pp. B3.1-B3.6, March 1974.

J. B. O'Neal, Jr., "Delta Modulation Quantizing Noise — Analytical and Computer Simulation Results for Gaussian and Television Signals," Bell System Tech. J., pp. 117-141, January 1966.

J. B. O'Neal, Jr., "Predictive Quantizing Systems (DPCM) for the Transmission of Television Signals," Bell System Tech. J., pp. 689-719, May-June 1966.

J. B. O'Neal, Jr. and T. R. Natarajan, "Coding Isotropic Images," IEEE Trans. on Information Theory, pp. 697-707, November 1977.

A. Papoulis, *Probability, Random Variables and Stochastic Processes*, McGraw-Hill, New York, 1965.

W. K. Pratt, *Image Coding*, John Wiley, New York, 1978.

D. L. Richards, *Telecommunication by Speech*, Halsted Press, John Wiley, New York, 1973.

J. B. Thomas, *Introduction to Statistical Communication Theory*, John Wiley, New York, 1969.

R. Zelinski and P. Noll, "Optimale Quantisierung gedächtnisfreier Gammaquellen und Anwendungen für die PCM — Codierung von Sprache", Technischer Bericht Nr. 169, Heinrich Hertz Institut, Berlin, 1974.

4

Quantization

4.1 Introduction

Information-bearing waveforms are typically *continuous-amplitude* and *continuous-time* in nature. Hence *analog-to-digital (A/D) conversion* is needed to produce a discrete representation of the waveform. We have discussed *sampling* or *time discretization* in Chapter 3. Resulting samples are still continuous in amplitude, as in Figure 1.1(c), and the process of *amplitude quantization* is needed to limit the number of possible amplitudes to a finite value, as in Figure 1.1(d). The operations of sampling and quantization could be performed in either order; but in practice, sampling precedes quantization. Amplitude quantization is a crucial step in a digital communication link, since it determines to a great extent the overall distortion as well as the bit rate necessary to communicate the waveform to the receiver.

Amplitude quantization is the procedure of transforming a given signal amplitude $x(n)$ at time n into an amplitude $y(n)$ taken from a finite set of possible amplitudes. In many of the discussions in this chapter, we will assume the important simple situation of *memoryless* and *instantaneous quantization*, a procedure where the transformation at time n is not affected by earlier or later input samples. Such a scheme is simple, but it is not optimum, even if the input samples are statistically independent; this is discussed in Chapter 9. Memoryless quantization is particularly inefficient if the samples are statistically dependent; these dependencies are therefore typically eliminated prior to quantization in the coders to be discussed in later chapters.

Coding systems involving Entropy Coding or certain types of Adaptive

Quantization need the use of memory and encoding delay, but even in these systems, the process of quantization itself can be regarded as instantaneous. But this is not true of the Vector Quantization systems to be discussed in Chapter 9.

When dealing with memoryless quantizers, we will drop the time index, and use symbols such as x instead of $x(n)$. We shall see later that the quantizer input need not always be the amplitude of the primary waveform to be coded. It may as well be a function of this waveform, for example, a difference between neighboring samples or some type of spectral information about input samples; and the quantizing process may only be one component of the overall coding system.

A digital system consists of at least three mappings. At the transmitter, the real-valued amplitude x is mapped into an L-ary number k. The channel maps k into k, deterministically, if the transmission is error-free; if not, it maps k into another L-ary number k', probabilistically. At the receiver, the L-ary number k or k' is mapped into representation level y_k or $y_{k'}$. These mappings are shown in Figure 4.1. Technically, the mappings at the transmitter and receiver are performed by an *Analog-to-Digital (A/D) Converter* and a *Digital-to-Analog (D/A) Converter*, respectively (Appendix G). In the absence of transmission errors, the combination of A/D and D/A operations is exactly identical to the quantizer mapping $Q(\cdot)$.

As shown in Figure 4.2, the signal amplitude x is specified by index k if it falls into the interval

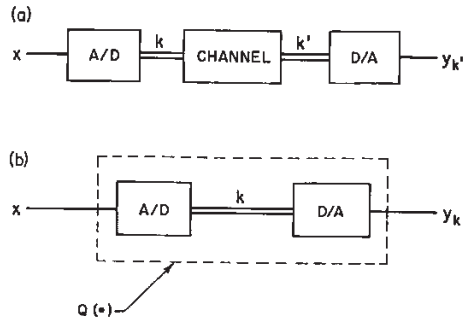
$$\mathcal{I}_k: \{x_k < x \leqslant x_{k+1}\}; \quad k = 1,2,...,L \tag{4.1}$$

The L-ary number k is transmitted to the receiver, typically in binary format. Let $L = 2^R$; then a *bit rate* of

$$R = \log_2 L \quad \textit{bits/sample} \tag{4.2}$$

is needed to inform the receiver about that index. At the receiver, the index k is transformed into an amplitude y_k that represents all amplitudes of the interval $\mathcal{I}_k$. The amplitudes y_k are called the *representation levels* or the *reconstruction values*, and the amplitudes x_k are called *decision levels* or *decision thresholds*. The

Figure 4.1 Digital communication system (a) with and (b) without transmission errors.



0103

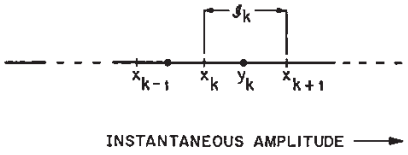


Figure 4.2 Quantization of amplitude x.

number of intervals equals the number of possible outputs y_k and each of these equals the finite number L. The set of possible outputs is described by

$$y \in \{y_1, y_2, \ldots, y_L\} \tag{4.3}$$

Unless otherwise mentioned, we assume error-free transmission. We thus have an output

$$y = y_k \quad \text{if} \quad x \in \mathcal{I}_k \tag{4.4}$$

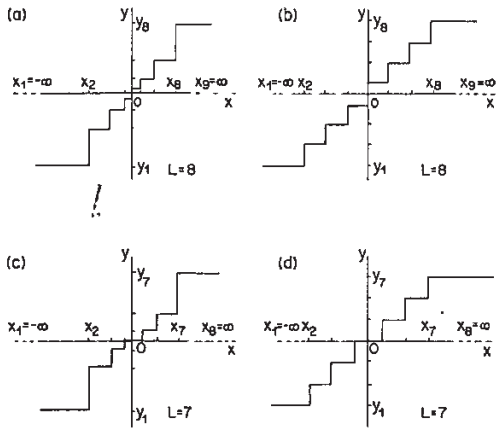
For a close approximation to the input, one requires that $y_k \in \mathcal{I}_k$. The actual mapping

$$y = Q(x) \tag{4.5}$$

is the *quantizer characteristic*, a staircase function by definition.

Several types of $Q(x)$ are shown in Figure 4.3 for $L = 8$ and $L = 7$. All of these characteristics are symmetric about zero. Unless otherwise stated, all future discussions will assume such symmetry. The quantizers in parts (a) and (c) of the figure are *nonuniform*, with decision level- and reconstruction level intervals being different in length and being functions of k. We will also be discussing at length

Figure 4.3 Quantizer characteristic types: (a) nonuniform midrise; (b) uniform midrise; (c) nonuniform midtread; and (d) uniform midtread.



the special but important case of a *uniform* quantizer, shown in parts (b) and (d) of the figure:

$$\text{Uniform Quantizer:} \quad y_{k+1} - y_k = \Delta; \quad k = 1,2,\ldots,L-1 \qquad (4.6)$$
$$x_{k+1} - x_k = \Delta; \quad \text{for finite } x_k, x_{k+1}$$

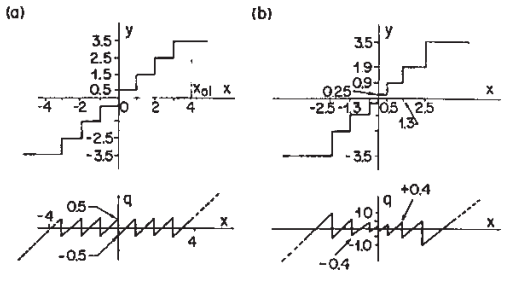
The characteristic $Q(x)$ can also be of *midtread* or *midrise* type, depending on whether zero is one of the output levels or not. For symmetry, L is chosen to be odd for midtread quantizers, as in parts (c) and (d) of Figure 4.3. Unless otherwise stated, we always assume midrise quantizers with even L.

Inherent in the quantization process is an error between input x and output y. This error is called *quantization error*, and we shall define it as the difference between x and y:

$$q = x - y = x - Q(x) \qquad (4.7)$$

where $y \in \{y_1, y_2, \ldots, y_L\}$. It is important to realize that the quantization error is totally deterministic for a given x because an input $x \in \mathcal{I}_k$ always leads to a single output y_k. Figure 4.4 shows this deterministic error as a function of x for an $L=8$ midrise example. However, if x is an outcome of a random variable X, it is reasonable to consider q as an outcome of another random variable Q; for this reason, we sometimes use the term *quantization noise* to describe the errors introduced by the quantizer. The use of the term "noise" is also motivated by an attempt to unify nomenclature, consistent with "channel noise" and "thermal noise" in analog systems. Indeed, quantization noise can often be modeled analytically as additive noise, especially if the quantization is very fine, as discussed in Section 4.7; in fact, with a technique known as dithering (Section 4.8), resulting errors in quantization can be modeled as input-independent additive noise even with coarse quantization. However, it must be remembered that unlike channel noise or thermal noise, quantization noise is deliberately introduced and in general it is signal-correlated.

Figure 4.4 Illustration of the deterministic nature of quantization error q in zero-memory quantization: (a) uniform quantization characteristic (step size $\Delta = 1$) and (b) nonuniform quantization characteristic (min $\{\Delta\} = 0.5$; max $\{\Delta\} = \max\{y_k - y_{k-1}\} = 1.6$).



4.2 Calculation of Quantization Error Variance

4.2.1 Basic Formulas

Let X be a zero-mean random variable at the quantizer input, with variance σ_x^2 and pdf $p_x(\cdot)$:

$$\mu_x = E[X] = \int_{-\infty}^{\infty} x p_x(x)dx = 0; \qquad \sigma_x^2 = E[X^2] = \int_{-\infty}^{\infty} x^2 p_x(x)dx \tag{4.8}$$

If the input X has a nonzero mean, we can remove it by subtracting $E[X]$ from the input, and add it back after quantization. Alternatively, we could interpret this procedure as a shift of the center of the quantizer to the mean value. A zero mean will therefore be assumed in our analytical studies without loss of generality.

A quantizer $Q(\cdot)$ maps X into the discrete-valued r.v. Y:

$$Y = Q(X) = y_k \quad \text{if } X \in \mathcal{I}_k \tag{4.9}$$

Thus, the quantization error $Q = X - Y$ is also a random variable with pdf $p_q(\cdot)$ and variance

$$\sigma_q^2 = E[Q^2] = \int_{-\infty}^{\infty} q^2 p_q(q)dq \tag{4.10}$$

Alternatively, as $Q = X - Q(X)$ is a function of X,

$$\sigma_q^2 = E[Q^2] = \int_{-\infty}^{\infty} [x - Q(x)]^2 p_x(x)dx \tag{4.11}$$

By breaking up the region of integration into L intervals $\mathcal{I}_k$ (4.1) and by making use of (4.9), we obtain the expression

$$\sigma_q^2 = \sum_{k=1}^{L} \int_{x_k}^{x_{k+1}} (x - y_k)^2 p_x(x)dx \tag{4.12}$$

Quantization error variance is the most important quantity for comparing the performances of quantizers, and we often use (4.12) to calculate this variance. Its calculation via (4.10) is possible only if the pdf $p_q(q)$ is known, which is typically not the case. However, we shall see in Example 4.1 and later in Section 4.7 that $p_q(q)$ may be approximated by a constant over a finite range in the case of uniform quantization.

Example 4.1. Uniform quantization of a bounded input

Consider the example of an input x with amplitudes in the range

$$x \in (-x_{max}, x_{max}) \tag{4.13}$$

0106

and a uniform quantizer with $-x_{max} = x_2 - \Delta$; $x_{max} = x_8 + \Delta$, as in the 3-bit example of Figure 4.3(b). In this case,

$$\Delta = 2x_{max}/2^R \tag{4.14}$$

Quantization errors [as seen in Figure 4.4(a)] will have values in the range

$$-\Delta/2 \leqslant q \leqslant \Delta/2 \tag{4.15}$$

If Δ is sufficiently small, it is reasonable to assume that they are uniform in the above range:

$$p_q(q) = \begin{cases} 1/\Delta & |q| \leqslant \Delta/2 \\ 0 & \textit{otherwise} \end{cases} \tag{4.16}$$

and (4.10) yields

$$\sigma_q^2 = \Delta^2/12 \tag{4.17}$$

Using (4.14),

$$\sigma_q^2 = \frac{1}{3} x_{max}^2 \, 2^{-2R} \tag{4.18}$$

Equation (4.17) shows that the standard deviation (rms value) of the noise increases linearly with step size, and (4.18) shows that the rms value decreases exponentially with increasing bit rate R. Since $L = 2^R$, an increase of R by 1 bit/sample implies a doubling of the number of steps L and hence, for a given x_{max}, a reduction in step size Δ by a factor of 2. Note finally that (4.15) and (4.16) are true only if the input is bounded. For example, if $|x| > 4$ in Figure 4.4(a), the error would increase linearly with $|x|$, as shown by the dashed extensions in the lower part of the figure.

Maximum Fractional Error. Consider the uniform quantization formula $\Delta = 2x_{max}/2^R$ (4.14). For a bounded input, we define the *maximum fractional error* as $(\Delta/2)/x_{max}$. For this number to be 1% and 0.1%, we respectively need $R = 7$ ($L = 128$) and $R = 10$ ($L = 1024$). •

The preceding example has shown that σ_q^2 can be obtained from (4.10). But (4.12) is also significant; it allows us to calculate σ_q^2 when the quantizer parameters (thresholds and representation levels) and the pdf of the input are given. Moreover, it also allows us to find quantizer parameters which result in the *minimum mean squared error variance (mmse)*. We shall discuss specific mmse designs in Sections 4.3 and 4.4.

Quantizer Performance Factor. It is useful to relate the error variance σ_q^2 to the input signal variance σ_x^2. The ratio

$$\epsilon_q^2 = \sigma_q^2/\sigma_x^2 \tag{4.19}$$

will be called the *quantizer performance factor*. It is the quantization error

variance for a unit-variance input signal. The inverse of ϵ_q^2 is the *signal-to-quantization noise ratio SNR*:

$$SNR \text{ (dB)} = 10 \log_{10} (\sigma_x^2/\sigma_q^2) = 10 \log_{10} (1/\epsilon_q^2) \tag{4.20}$$

It is sometimes useful to split the quantizer performance in the form

$$\sigma_q^2 = \epsilon_*^2 \, 2^{-2R} \, \sigma_x^2 \tag{4.21}$$

The parameter ϵ_*^2 is in fact the reciprocal of the constant a in (1.6). We shall note presently that ϵ_*^2 is greater than unity for inputs with the four pdf's in Figure 2.5, except in some instances where a refinement called entropy coding is applied (Section 4.6).

The foregoing notation allows us to compute a general expression for the necessary bit rate for a given *SNR*:

$$R = \frac{1}{2} \log_2 (\sigma_x^2/\sigma_q^2) + \frac{1}{2} \log_2 \epsilon_*^2 \quad \text{bits/sample} \tag{4.22a}$$

For a given value of σ_q^2, the minimum bit rate as given by rate distortion theory (Appendix D) is of the form

$$\min\{R\} = \frac{1}{2} \log_2 (\sigma_x^2/\sigma_q^2) - \alpha_R \quad \text{bits/sample} \tag{4.22b}$$

where $\alpha_R = 0$ for a Gaussian i.i.d. source (D.19) and $\alpha_R > 0$ for a non-Gaussian i.i.d. source. The $\log_2 \epsilon_*^2$ term, on the other hand, is positive; but it can be made negative by employing entropy coding. Indeed, we will show in Section 4.6 that for all these sources, the second term can be made at least arbitrarily close to $0.25 - \alpha_R$, leading to a required bit rate R that is only 0.25 bit greater than min $\{R\}$.

4.2.2 Granular Noise and Overload Distortion

Let us go back to Figure 4.4. Recall that step size in that figure was $\Delta = 1$. The dashed extensions in the error plots refer to a region of input amplitude where the error q no longer has the finite range $(-\Delta/2, \Delta/2)$. The value $|x| = 4\Delta$ in Figure 4.4(a), for example, is called the *overload amplitude* x_{ol}, the region $|x| > x_{ol}$ is called the *overload region*, and samples of $|q|$ that exceed $\Delta/2$ are called *overload errors*. Errors in the finite $(-\Delta/2, \Delta/2)$ range are called *granular errors*. Granularity refers to the rapid oscillatory or "fine-grained" structure of the solid line part of the $q(x)$ function of Figure 4.4. This structure is also characteristic of the $q(n)$-versus-n waveforms that result when $x(n)$ values increase or decrease rapidly with n in an input waveform segment. Overload error waveforms, on the other hand, are characterized by bursts of $q(n)$ samples of identical polarity (Figure 4.5) that occur if the overloading input $x(n)$ is slowly varying, as most Nyquist-sampled and over-sampled waveforms are. The above distinction in the structure of the two types of errors is emphasized by the terms *granular noise* and *overload distortion*. The difference in corresponding error

0108

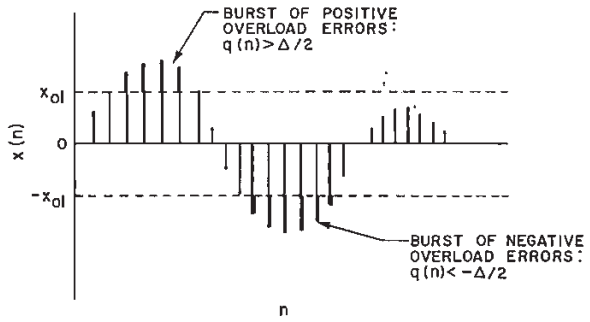
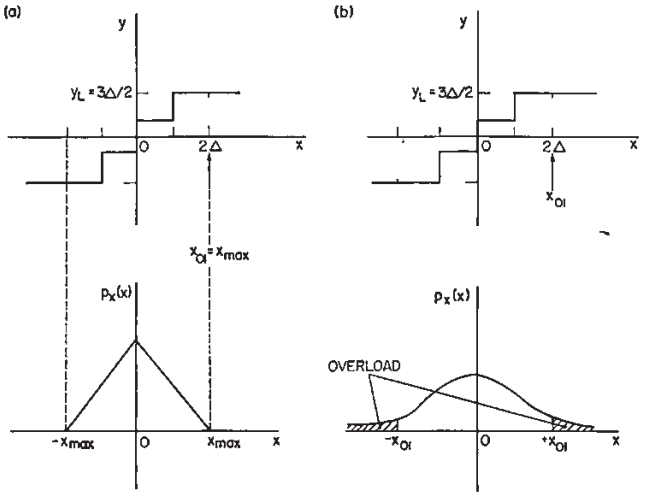


Figure 4.5 Overload error bursts. Heavy lines indicate overload error samples caused by bursts of $|x(n)|$ values that exceed $|x_{ol}|$.

spectra is reflected by significant differences in the perceptual annoyance of the two types of errors in the coding of physical signals, and we shall comment on this many times in this book. We note at this point that purely *from an error-variance viewpoint*, overload errors can be more serious, simply because these error samples are unbounded in the case of unbounded inputs, and these values are squared in computing σ_q^2.

Although overload errors can be unbounded, the *probability of overload* can be reduced by a proper design of the quantizer. In fact, this probability can be made zero if the input pdf is *bounded* to $x_{\max}$ and the quantizer overload amplitude x_{ol} is designed to equal or exceed $x_{\max}$. In Figure 4.6(a), $L=4$ and $x_{\max} = x_{ol} = 2\Delta$. Designs such as these will indeed be shown to be optimal for a uniform input pdf (Section 4.3). If the input pdf is *unbounded* ($x_{\max} = \infty$), there is always a non-

Figure 4.6 Four-level quantizers with (a) bounded and (b) unbounded inputs.



zero probability of overload. This is shown by the shaded area in Figure 4.6(b). The overload point x_{ol} has been defined in the midriser example in Figure 4.6(a) so that $|q| > \Delta/2$ for $|x| > x_{ol}$. The same definition of x_{ol} can also be applied to a midtread quantizer. In general,

$$x_{ol} = \begin{cases} 2^{R-1}\Delta & \text{for a uniform midrise quantizer} \\ (2^{R-1} - 0.5)\,\Delta & \text{for a uniform midtread quantizer} \end{cases} \tag{4.23}$$

An interesting type of non-uniform quantizer is one where step size Δ_k in interval $\mathcal{I}_k$ is an increasing function of k [Figure 4.3 (a) and (c)]. In this case, x_{ol} may be defined so that $|q| > \max\{\Delta_k\}/2$ for $|x| > x_{ol}$.

The total error variance (4.12) can be expressed for any quantizer characteristic as a sum of granular and overload error variances:

$$\sigma_q^2 = \sigma_{q(granular)}^2 + \sigma_{q(overload)}^2 ; \tag{4.24}$$

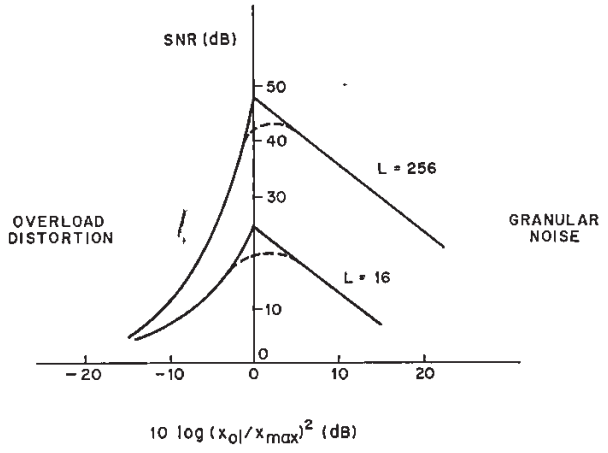
$$\sigma_{q(granular)}^2 = \sum_{k=2}^{L-1} \int_{x_k}^{x_{k+1}} (x - y_k)^2 p_x(x)\,dx + 2\int_{x_L}^{x_{ol}} (x - y_L)^2 p_x(x)\,dx \tag{4.25a}$$

$$\sigma_{q(overload)}^2 = 2\int_{x_{ol}}^{\infty} (x - y_L)^2 p_x(x)\,dx \tag{4.25b}$$

Relative values of granular and overload noise components depend on x_{ol}, and hence on step size.

Figure 4.7 shows $SNR = 10 \log_{10} (\sigma_x^2/\sigma_q^2)$ as a function of x_{ol}/x_{max} for the example of a uniform quantizer (Problem 4.1) with a uniform pdf input

Figure 4.7 Uniform quantizer performance with uniform input pdf.



($|x| \leq x_{max}$), using values $L = 16$ and 256 for illustration. The extreme left of the figure (small x_{ol}) is an overload-distortion-dominated region, while the right ($x_{ol} > x_{max}$) describes the effects of granular errors. The fact that *SNR* decreases more rapidly with increasing overload than with increasing granularity causes an asymmetry about the *SNR*-axis that will be observed repeatedly in similar future illustrations in this book.

The dashed lines in Figure 4.7 indicate qualitative results for unbounded inputs such as those with a Gaussian pdf. In this case, the transition between overload and granular regions will be a smooth one, and the maximum of the characteristic depends on the so-called *loading factor*; see Example 4.4 and Figure 4.8.

Example 4.2. Probability of overload

The probability of overload can easily be calculated for a random variable X with a Gaussian pdf. Following (2.15), the probability that the amplitude is above the overload point is

$$P\{|x| > x_{ol}\} = 2(1-P_x(x_{ol})) = 1 - \text{erf}\,(x_{ol}/(\sqrt{2}\sigma_x)) \tag{4.26}$$

For $x_{ol} = 4\sigma_x$ (4σ-loading), we have a probability of overload given by $1 - \text{erf}\, 2\sqrt{2} = .000064$, a generally negligible value. With $x_{ol} = 2\sigma_x$ (2σ-loading), on the other hand, the overload probability will be $1-\text{erf}\,\sqrt{2} = 0.046$. •

4.3 Uniform Quantization

In *uniform quantizers*, the decision intervals $\mathcal{I}_k$ are of the same length Δ, and the reconstruction levels are the midpoints of the decision intervals. In the case of midrise quantizers,

$$x_1 = -\infty\,; \quad x_{L+1} = \infty\,; \quad x_{k+1} - x_k = \Delta\,; \quad k = 2,\ldots,L-1 \tag{4.27}$$

Figure 4.3(b) has already shown the transfer function of the uniform midriser quantizer for the example of $L=8$.

4.3.1 The Case of Zero or Negligible Overload

We assume an input random variable X of variance σ_x^2 where the amplitudes are bounded and the dynamic range of the input equals that of the quantizer. Following Figure 4.6(a),

$$|x| \leq x_{max} = x_{ol} = L\Delta/2 \tag{4.28}$$

As shown earlier on the basis of a uniform (granular) error pdf in the range $(-\Delta/2, \Delta/2)$ (which is a good assumption for $R \gg 1$),

$$\sigma_q^2 = \Delta^2/12 = (1/3)\, x_{max}^2\, 2^{-2R} \tag{4.29}$$

This error variance can be reduced by a factor of 4 by reducing the step size Δ by a

0111

factor of 2, i.e., by doubling L (for a given x_{max}). Let us define a *loading factor*

$$f_l = x_{ol}/\sigma_x = L\Delta/2\sigma_x \tag{4.30}$$

This determines the half-amplitude range of the quantizer in multiples of the standard deviation of the signal to be quantized. From (4.29) and (4.30),

$$\sigma_x^2/\sigma_q^2 = 3L^2\,\sigma_x^2/x_{max}^2 = 3L^2/f_l^2$$

$$SNR\ (\text{dB}) = 6.02R - 10\log_{10}(f_l^2/3): \quad R \gg 1 \tag{4.31}$$

Example 4.3. Uniform pdf input with $x_{max} = x_{ol}$

$f_l = x_{max}/\sigma_x$, which is equal to $\sqrt{3}$ for the uniform pdf; $SNR\ (\text{dB}) = 6.02\,R$ (4.32)

We will note presently (also see Figure 4.7) that the empirical design $x_{max} = x_{ol}$ is indeed optimal for the uniform pdf. •

Example 4.4. Sine-wave input with $x_{max} = x_{ol}$

$f_l = x_{max}/\sigma_x$, which is equal to $\sqrt{2}$ for the arcsine pdf; $SNR\ (\text{dB}) = 6.02\,R + 1.76$ (4.33)

Quantization of sine-waves is of interest since they are useful as test signals in digital coding systems. Also see Problem 4.3. •

Example 4.5. Unbounded input with negligible overload probability

The formulas derived above may also be used for *unbounded* inputs if x_{ol} is chosen such that the probability of overload $P\{|x| \geq x_{ol}\}$ is sufficiently small so that the corresponding overload errors can be neglected. A typical value of the loading factor is $f_l = 4$ (4σ-loading, as in Example 4.2). In this case,

$$f_l = 4; \qquad SNR\ (\text{dB}) = 6.02\,R - 7.27 \tag{4.34}$$

It will be seen in the next section that this choice of f_l *does not give the minimum overall (i.e., granular plus overload) noise*; the optimum value of f_l is pdf-and R-dependent. For Gaussian pdf's it is a monotonically increasing function of R, with values between 1.6 and 3.8 in the range of $R = 1$ to 8. This implies an overload probability (4.26) that increases with decreasing R. •

4.3.2 PDF-Optimized Uniform Quantizers

The preceding section has shown that it is easy to calculate the SNR under the assumption of zero overload. In general, σ_q^2-minimizing designs imply non-zero overload for any unbounded pdf. We find the optimum or *minimum mean squared error (mmse)* step size, and hence the *mmse uniform quantizer*, by minimizing the error variance as given by (4.12) with the uniform quantizer definitions given by (4.27):

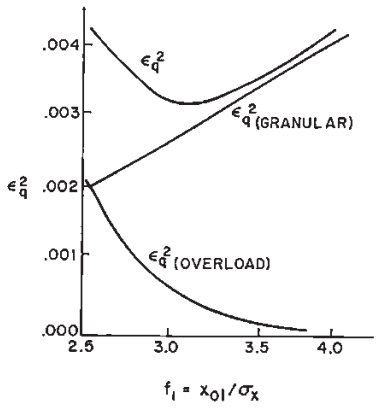
$$x_k = \left(k - \frac{L+2}{2}\right)\Delta \; ; \quad k = 2,3,\ldots,L$$

$$y_k = \left(k - \frac{L+1}{2}\right)\Delta \; ; \quad k = 1,2,\ldots,L \tag{4.35}$$

where we have assumed a midrise quantizer [Figure 4.3(b)] with an even number L of output levels ($x_1 = -\infty$; $x_{L+1} = \infty$). For minimizing σ_q^2 we require that $d\sigma_q^2/d\Delta = 0$. In general it is not possible to calculate the optimum step sizes Δ_{opt} explicitly. The simple cases of $L = 2$ and $L = 3$ will be seen to be exceptions. However, we can solve the minimization problem numerically with a computer [Max, 1960] [Paez and Glisson, 1972]. The resulting step size Δ_{opt} minimizes the total error variance, the sum of $\sigma_{q(granular)}^2$ and $\sigma_{q(overload)}^2$. Figure 4.8 shows an example of the dependence of these terms on f_l. The granular noise variance *increases* with f_l because of increasing step size, and the overload noise variance *decreases* with f_l because the probability of overload becomes smaller. The total noise σ_q^2 has a minimum with respect to f_l (near 3.05 in the 5-bit Gaussian example in the figure). Note that a similar minimum occurred in the uniform pdf example of Figure 4.7; although at a different value of f_l: specifically, at a value of f_l for which $x_{ol} = x_{\max}$ i.e., at $f_l = x_{\max}/\sigma_x = \sqrt{3}$ (see Example 4.3).

Table 4.1 gives the optimum step sizes and the corresponding *SNR*-values max{*SNR*} of pdf-optimized symmetric uniform quantizers for various pdf's. The *SNR* values for the uniform pdf are those of (4.32). The other pdf's are those of unbounded random variables. Large step sizes have to be chosen in order to have a balance between granular noise and overload noise. A study of Table 4.1 reveals that for unbounded inputs the deviation of the *SNR* (in dB) from a simple 6.02 R formula increases with bit rate. In the term ϵ_x^2 of (4.21), this implies an increase of ϵ_x^2 with R. For example, for a gamma pdf we have $\epsilon_x^2 = 2.67$ for $R = 1$ and $\epsilon_x^2 =$

Figure 4.8 Dependence of normalized overload and granular error variances on loading factor $f_l = x_{ol}/\sigma_x$ in the uniform 5-bit quantization of a Gaussian input.



0113

Table 4.1 Optimum step size and max{*SNR*} for uniform symmetric quantizers with different input pdf's (U: Uniform G: Gaussian L: Laplacian Γ: Gamma) [Max, 1960] [Paez and Glisson, 1972].

R (bits/sample)	Δ_{opt}/σ_x				max {*SNR*} (dB)			
	pdf				pdf			
	U	G	L	Γ	U	G	L	Γ
1	1.7320	1.5956	1.4142	1.1547	6.02	4.40	3.01	1.76
2	0.8660	0.9957	1.0874	1.0660	12.04	9.25	7.07	4.95
3	0.4330	0.5860	0.7309	0.7957	18.06	14.27	11.44	8.78
4	0.2165	0.3352	0.4610	0.5400	24.08	19.38	15.96	13.00
5	0.1083	0.1881	0.2800	0.3459	30.10	24.57	20.60	17.49
6	0.0541	0.1041	0.1657	0.2130	36.12	29.83	25.36	22.16
7	0.0271	0.0569	0.0961	0.1273	42.14	35.13	30.23	26.99
8	0.0135	0.0308	0.0549	0.0743	48.17	40.34	35.14	31.89

32.77 for $R=7$. On the other hand, rate distortion theory promises ϵ_q^2 values that are upperbounded by a value of 1.0. In this sense, we conclude that the relative performance of uniform quantizers becomes worse at high bit rates. Finally, we note from Table 4.1 that the step size values (Δ_{opt}/σ_x) can also be used to compute the optimum loading factor $f_{l,opt}$ and hence the *SNR* in (4.31). It turns out that $f_{l,opt}$ in general is not a constant but depends on $R = \log_2 L$. When using the approximate rule-of-thumb formula (4.34), this should be kept in mind. A special case is that of the uniform pdf for which $f_{l,opt} = \sqrt{3}$ for all R. This is easily verified from the numbers in Table 4.1. On the other hand, for the example of a Gaussian input, optimal values of f_l for $R = 1$, 2, 3 and 4 are 1.6, 2.0, 2.3 and 2.7, respectively.

4.3.3 One-Bit Quantization ($R = 1$)

One-bit quantization is depicted in Figure 4.9:

$$x_1 = -\infty; \quad x_2 = 0: \quad x_3 = +\infty$$

$$y_1 = -\delta = -\Delta/2; \quad y_2 = +\delta = \Delta/2 \tag{4.36}$$

The step size Δ and hence the absolute value δ of the reconstruction level can be optimized in closed form. From (4.12),

$$\sigma_q^2 = \sigma_x^2 + \delta^2 - 4\delta \int_0^\infty x p_x(x)dx \tag{4.37}$$

if the pdf is symmetric, and therefore

$$\delta_{opt} = 2 \int_0^\infty x p_x(x)dx \tag{4.38}$$

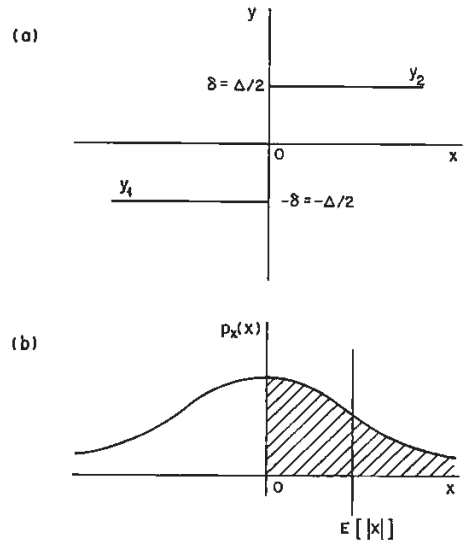


Figure 4.9 (a) One-bit quantization ($R = 1$, $L = 2$); (b) $\delta_{opt} = E[|X|]$ is the center of gravity of the right half of the input pdf, indicated by the shaded area.

$$\min\{\sigma_q^2\} = \sigma_x^2 \min\{\epsilon_q^2\} = \sigma_x^2 - \delta_{opt}^2 \tag{4.39}$$

Note that the magnitude of the optimum reconstruction level is just the conditional mean of the random variable for positive or negative values, or the mean of the absolute value:

$$\delta_{opt} = \Delta_{opt}/2 = E[X|x > 0] = E[|X|] \tag{4.40}$$

The optimum reconstruction level δ_{opt} and the minimum error variance depend on the pdf of the signal to be quantized: to be more exact, on the variance of the signal and on the functional form of the pdf (see Table 4.2 and Problem 4.3). If we had chosen a value $\delta \neq \delta_{opt}$, we would have

$$\sigma_q^2 = \sigma_x^2 + \delta^2 - 2\delta\delta_{opt} = \min\{\sigma_q^2\} + (\delta - \delta_{opt})^2 \tag{4.41}$$

This could happen if we had chosen a specific value δ expecting the signal to have either a different variance than the actual one or a different unit variance pdf. Such effects of *variance mismatch* or *pdf mismatch* on the quantization error variance are small in the case of one-bit quantization. This is not true for quantizers with more than two levels (Section 4.4.3).

Closed-form optimizations can also be given for 3-level quantization (Problems 4.4 and 4.5). It can be shown that for a symmetrical midtread 3-level quantizer (Figure 4.10), the *SNR* gain over the optimized two-level midrise (Figure 4.9) is 1.8 dB for a Gaussian input. This gain is obtained by increasing R from $\log_2 2 = 1$ to $\log_2 3 = 1.59$ bits/sample.

0115

Table 4.2 Two-level quantization; dependence of δ_{opt} and min $\{\epsilon_q^2\}$ on pdf.

pdf	δ_{opt}/σ_x	$\min\{\epsilon_q^2\}$	max{SNR} (in dB)
U	$\sqrt{3}/2$	0.25	6.02
G	$\sqrt{2/\pi}$	0.363	4.40
L	$1/\sqrt{2}$	0.5	3.01
Γ	$1/\sqrt{3}$	0.667	1.76

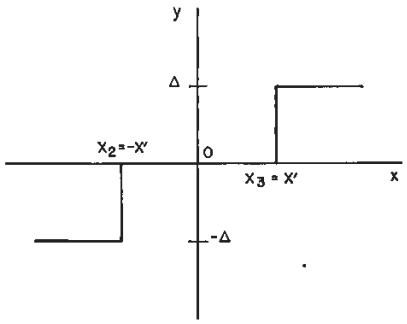


Figure 4.10 Three-level quantization.

Non-symmetric quantizers. The results of Table 4.1 apply to quantization characteristics that are symmetric with respect to zero, as in Figure 4.3(b). For the gamma pdf, as discussed again in Section 4.4, the optimum quantizer is non-symmetric. For example, if $R = 1$ bit/sample, the use of a decision threshold $x_2 = 0$ and reconstruction values $y_1 = -1.51\sigma_x$ and $y_2 = 0.27\sigma_x$, instead of the symmetric design of (4.36) and (4.38), will provide an SNR of 2.23 dB (which is nearly 0.5 dB higher than the value of 1.76 dB in Table 4.1) [Brehm and Trottler, 1983].

4.4 Nonuniform Quantization

Uniform quantization is the most common means for transforming an amplitude-continuous random variable into an amplitude-discrete variable. Equidistant decision and representation levels are simple conceptually, as well as for implementation. However, they do not necessarily represent the most effective conversion; it is possible, in general, to quantize a random variable with a smaller error variance. This can be attained by choosing smaller decision intervals where the probability of occurrence of the random variable X is high, i.e., where the pdf $p_x(x)$ is comparably high, and by choosing larger decision intervals otherwise, as seen in Figure 4.3(a) and (c). Note also that this procedure makes the magnitude of the quantization error depend strongly on that of input x.

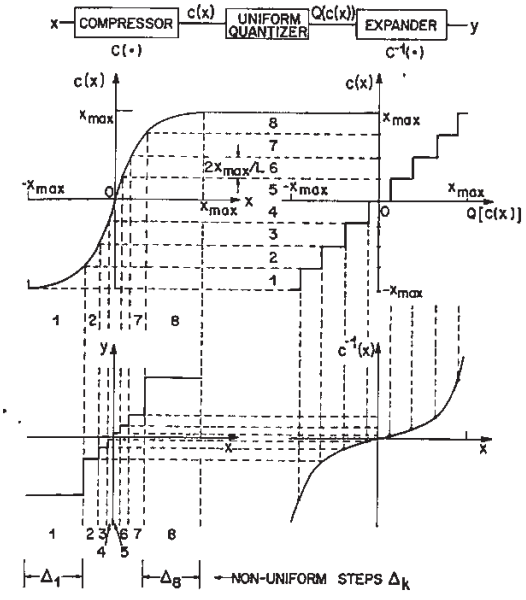
There are two approaches to the problem of finding the sets of optimum decision and representation values x_k, y_k for a given pdf. One is an exact solution based on an iterative procedure, and another is an approximation that is good for quantizers with a large number of levels. Before we go into the details of this analysis, we will note that nonuniform quantization can be achieved by compressing the signal x using a nonuniform *compressor characteristic* $c(\cdot)$, by quantizing the compressed signal $c(x)$ employing a uniform quantizer, and by expanding the quantized version of the compressed signal using a nonuniform transfer characteristic $c^{-1}(\cdot)$ that is inverse to that of the compressor. Figure 4.11 shows a block diagram of this *companding* technique (*com*pressing and ex*pand*ing). The compressor characteristic $c(\cdot)$ is sometimes also called the *companding law*. Implementation of this type of nonuniform quantization is no longer common (See Chapter 5). The companding concept is, however, useful in analyzing and optimizing nonuniform quantizers with a large number of levels.

As in the figure, consider the example of a bounded input in the range $(-x_{max}, x_{max})$. The characteristic $c(x)$ analytically defines non-uniform intervals Δ_k via uniform intervals of length $2x_{max}/L$ each. The slope of the compressor characteristic $c(\cdot)$ in interval $\mathscr{I}_k$ is

$$\frac{dc(x)}{dx} \approx \frac{2x_{max}}{L\Delta_k} \qquad \text{for large } L \tag{4.42}$$

The slope term $dc(x)/dx$ describes the *interval density*, and it is inversely proportional to the local step size in the equivalent non-uniform characteristic

Figure 4.11 The use of a compression function $c(x)$ to realize a nonuniform quantizer characteristic.



$Q(x)$. Thus, with a saturating $c(x)$ characteristic as in Figure 4.11, the equivalent step size is smallest at $x = 0$, and greatest at $x = x_{max}$. *In the case of a uniform quantizer, $c(x) = x$ and $dc(x)/dx = 1$.* As in the sketch of Figure 4.11, a typical nonuniform characteristic has an approximately constant slope around the origin. Since this slope defines the density of inner step sizes, it turns out that the inner step sizes will be approximately uniform. In addition, these step sizes near the origin will be smaller than for a uniform quantizer [$dc(x)/dx=1$]. But for large signal values, we now find larger step sizes and fewer intervals. The term

$$G_C = \frac{dc(x)}{dx}\bigg|_{x \to \infty} \geq 1 \tag{4.43}$$

is called the *companding gain.* We will come back to this term in the section on log-quantizers. The interval density $dc(x)/dx$ can also be used to determine the ratio of maximum to minimum step size (See Example 4.7). From (4.42) and (4.43), this ratio is

$$\frac{\max\{\Delta\}}{\min\{\Delta\}} = \frac{dc(x)/dx\big|_{x \to 0}}{dc(x)/dx\big|_{x \to x_{max}}} = \frac{G_C}{dc(x)/dx\big|_{x \to x_{max}}} \tag{4.44}$$

4.4.1 PDF-Optimized Nonuniform Quantizers: Iterative Solutions

The one-bit example has already shown that a quantizer can be optimized to yield the *minimum mean-squared error (mmse)* min $\{\sigma_q^2\}$. The mmse quantizer is frequently called the *Max quantizer* or *Lloyd-Max quantizer* [Max, 1960] [Lloyd, 1957, 1982]. Explicit solutions are possible in the case of L=2 and 3, as noted earlier. For $L > 2$, the mmse quantizer is nonuniform if the input pdf is itself nonuniform.

In the following we shall consider a stationary zero-mean random sequence $\{X(n)\}$ of variance σ_x^2 and give a non-explicit exact solution for *pdf-optimized quantizers* that minimize σ_q^2. Note that optimization with regard to pdf shape includes a match of quantizer to input variance σ_x^2. Mismatch relative to the shape or variance will be discussed later.

A general expression for quantization error variance has been given in (4.12) for any choice of the decision levels x_k; $k = 2,3,...,L$ and reconstruction levels y_k; $k = 1,2,...,L$. Necessary conditions for a minimum are

$$\frac{\partial \sigma_q^2}{\partial x_k} = 0\,; \quad k = 2,3,...,L; \qquad \frac{\partial \sigma_q^2}{\partial y_k} = 0\,; \quad k = 1,2,...,L$$

Note that these conditions may define a local optimum. However, if $p_x(x)$ is log-concave, i.e., if $\partial^2 \log p_x(x)/\partial x^2$ is negative, these conditions are also sufficient [Fleischer, 1964], and they lead to the important results [Lloyd, 1957,1982] [Max, 1960]:

$$x_{k,opt} = \frac{1}{2}(y_{k,opt}+y_{k-1,opt}); \quad k = 2,3,\ldots,L,$$

$$x_{1,opt} = -\infty\,, \quad x_{L+1,opt} = \infty \tag{4.45a}$$

$$y_{k,opt} = \frac{\int_{x_{k,opt}}^{x_{k+1,opt}} x p_x(x)dx}{\int_{x_{k,opt}}^{x_{k+1,opt}} p_x(x)dx}; \quad k = 1,2,\ldots,L \tag{4.45b}$$

The log-concave property holds for the pdf's **U**, **G** and **L** but not for the gamma pdf **Γ**. In this case, (4.45) will *not* define a global optimum.

The first of these equations shows that the optimum *decision levels are halfway between neighboring reconstruction levels*, and the second equation states that a *reconstruction level should be the centroid of the pdf in the appropriate interval.* Note that this centroid is the mean of the input signal on the given interval. Note also that (4.45) is true for previous results for $L = 2,3$. Various extensions can be drawn from (4.45). First, since the denominator equals $P_k = P\{X \in \mathcal{I}_{k,opt}\}$ and since $p_x(x) = P_k\, p_x(x|X \in \mathcal{I}_{k,opt})$ within interval $\mathcal{I}_{k,opt}$,

$$y_{k,opt} = \int_{-\infty}^{\infty} x\, p_x(x|X \in I_{k,opt})dx = E[X|X \in I_{k,opt}]\,, \tag{4.46}$$

a result that can also be anticipated from estimation theory. Further, the result (4.46) can be rearranged in the form

$$\int_{-\infty}^{\infty} (x-y_{k,opt})p_x(x|X \in I_{k,opt})dx = E[Q|X \in I_{k,opt}] = 0 \tag{4.47a}$$

i.e., the mean of the quantization noise in the interval is zero if the reconstruction level is chosen as the mean of the input in that interval. This last result implies that the unconditional mean of quantization error is zero:

$$E[Q] = \sum_k E[Q|X \in \mathcal{I}_{k,opt}]\, P_k = 0 \tag{4.47b}$$

We can also multiply (4.47) with $y_{k,opt}$ and sum over all indices k, to obtain

$$E[QY] = 0 \tag{4.48}$$

i.e., *the quantization noise is orthogonal to the output of the pdf-optimized quantizer.* However,

$$E[QX] = E[Q(Y+Q)] = E[Q^2] = \min\{\sigma_q^2\} \tag{4.49a}$$

i.e., *input and quantization error are correlated*, so that a simple input-independent additive noise model is invalid. Finally, by an algebraic manipulation that employs (4.48) twice, we can show that

$$\sigma_y^2 = \sigma_x^2 - \min\{\sigma_q^2\} \tag{4.49b}$$

i.e., *the variance of the output is less than that of the random variable at the quantizer input* [Wood, 1969]. This is related doubtless to the non-zero probability of input values greater than the maximum output level y_L. Also, as $L \to \infty$, $\min\{\sigma_q^2\} \to 0$ and $\sigma_y^2 \to \sigma_x^2$. The observation $\sigma_y^2 < \sigma_x^2$ is also true of the rate distortion-optimal $x \to y$ mapping as noted in Appendix D.

Two other corollaries of the above set of results are worth mention:

$$E[XY] = (1 - \epsilon_q^2)\,\sigma_x^2 \tag{4.50}$$

$$P_k E[Q^2 | X \in \mathcal{I}_{k,opt}] = P_j E[Q^2 | X \in \mathcal{I}_{j,opt}] \tag{4.51}$$

The last result shows that *error variance contributions from different intervals are identical*. This is true in spite of varying Δ_k, because of the equalizing role of interval probability values P_j and P_k. This result is formally derived in Section 4.4.2.

From the conditions in (4.45), the optimum values $x_{k,opt}$ and $y_{k,opt}$ can be calculated iteratively. Only positive values have to be calculated if $p_x(x)$ is symmetric. A special procedure is known for Laplacian inputs [Nitadori, 1965] [Noll and Zelinski, 1979].

One begins by choosing a value y_1, solves (4.45b) for x_2, solves (4.45a) for y_2, ..., and solves (4.45a) for y_L. One then checks if y_L is close enough to the right hand term of (4.45b). If not, one perturbs y_1 in an appropriate direction and continues the iteration. Table 4.3 lists the decision and reconstruction levels for various pdf's and for quantizers with $L = 2$, 4, 8, and 16 levels. The table uses a re-numbering scheme for x and y indices for convenience, and shows only positive values, recognizing symmetry about zero. Note that the $x_{k,opt}$, $y_{k,opt}$-values of Table 4.3 have to be multiplied with σ_x for inputs with a non-unity variance. Note also that in the case of a uniform input pdf, the Max quantizer will be uniform as well, as in Section 4.3.

Implied in Table 4.3, as in Table 4.1, is a *symmetry about zero* in the quantizer characteristic. This symmetry is an intuitionally expected result when the input has a pdf that is symmetrical about zero. The four pdf's that we have considered are indeed symmetrical about zero. There are examples of zero-mean symmetrical pdfs for which the mmse quantizer is *not* symmetric. See Section 4.3.3, Problem 4.5 and [Abhaya and Wise, 1981].

Table 4.4 lists signal-to-noise ratios of Max quantizers for $R = 1, 2, \ldots, 7$ bits/sample. The results of the present table should be compared with those of Table 4.1, and with the rate distortion bounds of Table D.1. Notice that gains due to non-uniform quantization are most significant with a very "non-uniform" pdf such as the gamma pdf. With $R = 7$, the gain over uniform quantization is 2.7 dB for a Gaussian pdf, and 7.7 dB for a gamma pdf.

0120

Table 4.3 Optimum decision values x_j and reconstruction values y_j for pdf-optimized nonuniform quantizers (U: Uniform G: Gaussian L: Laplace Γ: Gamma) [Max, 1960] [Paez and Glisson, 1972] [Noll and Zelinski, 1974]). Note that in this table, the quantizer characteristics are symmetrical about zero; $j = 1$ corresponds to the first non-negative value of x or y, and $j > 1$ to succeeding positive values. This notation is different from that in the rest of this chapter; the more usual notation is that of Figure 4.3.

R		1		2		3		4	
pdf	j	$x_{j,\,opt}$	$y_{j,\,opt}$	$x_{j,\,opt}$	$y_{j,\,opt}$	$x_{j,\,opt}$	$y_{j,\,opt}$	$x_{j,\,opt}$	$y_{j,\,opt}$
U	1	0.000	0.866	0.000	0.433	0.000	0.217	0.000	0.109
	2			0.866	1.299	0.433	0.650	0.217	0.326
	3					0.866	1.083	0.433	0.542
	4					1.299	1.516	0.650	0.759
	5							0.866	0.975
	6							1.083	1.192
	7							1.299	1.408
	8							1.516	1.624
G	1	0.000	0.798	0.000	0.453	0.000	0.245	0.000	0.128
	2			0.982	1.510	0.501	0.756	0.258	0.388
	3					1.050	1.344	0.522	0.657
	4					1.748	2.152	0.800	0.942
	5							1.099	1.256
	6							1.437	1.618
	7							1.844	2.069
	8							2.401	2.733
L	1	0.000	0.707	0.000	0.420	0.000	0.233	0.000	0.124
	2			1.127	1.834	0.533	0.833	0.264	0.405
	3					1.253	1.673	0.567	0.729
	4					2.380	3.087	0.920	1.111
	5							1.345	1.578
	6							1.878	2.178
	7							2.597	3.017
	8							3.725	4.432
Γ	1	0.000	0.577	0.000	0.313	0.000	0.155	0.000	0.073
	2			1.268	2.223	0.527	0.899	0.230	0.387
	3					1.478	2.057	0.591	0.795
	4					3.089	4.121	0.051	1.307
	5							1.633	1.959
	6							1.390	2.822
	7							3.422	4.061
	8							5.128	6.195

Figure 4.12 plots the quantity $\Delta SNR = 6.02R - \max\{SNR\}$ as a function of R. Solid curves are for pdf-optimized non-uniform quantizers, while dashed curves are for pdf-optimized uniform quantizers. Note that in the case of a uniform pdf, the optimum quantizer (which is also uniform) attains $\Delta SNR = 0$ for all R; one gains exactly 6.02 dB for every doubling of L. With other pdf's, ΔSNR increases with R, but rapidly attains a pdf-specific asymptote, as given in (4.68); uniform quantizers for these pdf's become increasingly inefficient with increasing R.

0121

Table 4.4 Performance of pdf-optimized quantizers. Values of max{*SNR*} (in dB). (U: Uniform G: Gaussian L: Laplace Γ: Gamma) [Max, 1960] [Paez and Glisson, 1972] [Noll and Zelinski, 1974].

R (bits per sample)	pdf			
	U	G	L	Γ
1	6.02	4.40	3.01	1.76
2	12.04	9.30	7.54	6.35
3	18.06	14.62	12.64	11.52
4	24.08	20.22	18.13	17.07
5	30.10	26.01	23.87	22.85
6	36.12	31.89	29.74	28.73
7	42.14	37.81	35.69	34.67

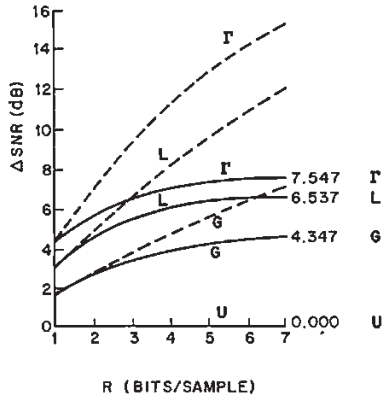


Figure 4.12 Values of $\Delta SNR = 6.02R - \max\{SNR\}$ in pdf-optimized quantizers. Solid and dashed curves are for nonuniform and uniform quantizers, respectively [Noll and Zelinski, 1974].

We will see in Section 4.6 that even smaller values of ΔSNR can be realized with Entropy Coding, if the quantizer input is stationary. On the other hand, with non-stationary inputs, logarithmic non-uniformity (Section 4.5) and adaptive techniques (Section 4.10) provide important means for robust quantization.

4.4.2 PDF-Optimized Nonuniform Quantizers: Approximate Formulas for Large *R*

Consider the high bit rate case where R and therefore L, the number of quantizer steps, are both sufficiently large; we can then assume that the pdf $p_x(x)$ of the input signal is approximately flat in all intervals $\mathcal{I}_k$; $k = 1,2,...,L$, which includes the outermost intervals 1 and L. We also assume that the input is

0122

bounded in practice to a value x_{max}, even if the pdf should have a long tail. We also assume that the pdf is symmetric. Let

$$p_x(x) = constant = p_x(y_k) \quad \text{for } X \in \mathcal{I}_k \tag{4.52}$$

where y_k is a value between x_k and x_{k+1}. Let Δ_k be the length of interval $\mathcal{I}_k$:

$$\Delta_k = x_{k+1} - x_k \tag{4.53}$$

The probability of finding X in the interval $\mathcal{I}_k$ is then

$$P_k = P\{X \in \mathcal{I}_k\} = p_x(y_k)\Delta_k; \qquad \sum_{k=1}^{L} P_k = 1 \tag{4.54}$$

The error variance (4.12) is

$$\sigma_q^2 = \sum_{k=1}^{L} \frac{P_k}{\Delta_k} \int_{x_k}^{x_{k+1}} (x-y_k)^2 dx \tag{4.55}$$

and a necessary condition for a minimum is

$$\frac{\partial \sigma_q^2}{\partial y_k} = 0; \qquad k = 1,2,...,L \tag{4.56}$$

which results in

$$y_k = \frac{1}{2}(x_k + x_{k+1}) \tag{4.57}$$

The representation level y_k lies in the middle of the interval $\mathcal{I}_k$; this is of course a direct consequence of the assumption of a flat pdf in the intervals, and y_k is the centroid of $p_x(x)$ in this interval $\mathcal{I}_k$ as in (4.45b). By combining (4.55) and (4.57) we obtain

$$\sigma_q^2 = \frac{1}{12} \sum_{k=1}^{L} P_k \Delta_k^2 \tag{4.58}$$

with $\Delta_k^2/12$ representing the error variance conditioned on interval $\mathcal{I}_k$. For uniform quantizers ($\Delta_k = \Delta$; all k), (4.58) reduces to $\Delta^2/12$ [as in (4.17)] since the summation over P_k values (4.54) equals 1. Note also that (4.58) does not depend on the y_k-values because these are defined by the decision levels (4.57). Therefore only these decision levels (or, equivalently, the lengths of the intervals) need to be optimized.

A first condition for optimality can be derived as follows. With (4.54) and (4.58),

$$\sigma_q^2 = \frac{1}{12} \sum_{k=1}^{L} p_x(y_k)\Delta_k^3 = \frac{1}{12} \sum_{k=1}^{L} \alpha_k^3 \tag{4.59}$$

0123

However, we have the constraint that

$$\sum_{k=1}^{L} \alpha_k \simeq \int_{-x_{max}}^{x_{max}} \sqrt[3]{p_x(x)}\, dx \simeq \text{constant} \tag{4.60}$$

since the above integral is a function only of its limits. Use of the Lagrange multiplier technique

$$\frac{\partial}{\partial \alpha_k} \left(\sigma_q^2 + \lambda \sum_{k=1}^{L} \alpha_k\right) = 0 \quad \text{for } k = 1,2,\ldots,L$$

therefore results in

$$p_x(y_k)\, \Delta_k^3 = P_k \Delta_k^2 = \textit{constant} \tag{4.61}$$

An interval must be small if the probability is high that this interval is occupied, and vice versa. From (4.58) and (4.61) we see that *all quantizer intervals will produce identical quantization error variance contributions*; also see (4.51).

We shall now derive equivalent results in closed form using the $c(x)$ notation of Figure 4.11. From (4.58) and (4.42),

$$\sigma_q^2 \approx \frac{x_{max}^2}{3L^2} \sum_{k=1}^{L} P_k \left[dc(x)/dx\right]^{-2} \quad \text{for large } L \tag{4.62}$$

Setting $P_k \simeq p_x(x)dx$ after (4.54),

$$\sigma_q^2 \approx \frac{x_{max}^2}{3L^2} \int_{-x_{max}}^{+x_{max}} p_x(x) \left[dc(x)/dx\right]^{-2} dx \quad \text{for large } L \tag{4.63}$$

This result [Bennett, 1948] expresses the error variance in terms of input pdf and the compressor characteristic $c(\cdot)$ of the quantizer. Recall that $c(\cdot)$ determines the decision levels, and that the reconstruction levels are determined by the latter ones via (4.57).

We can now minimize the error variance by searching for the optimum companding law $c(x)$ under the constraint of a fixed area under the interval density:

$$\int_{-x_{max}}^{x_{max}} \left[dc(x)/dx\right] dx = \textit{constant} \tag{4.64}$$

The area under the interval density is a measure of the number of levels and should therefore be a constant. Note that without the constraint, $\sigma_q^2 \to 0$ for $dc(x)/dx \to \infty$. To use Lagrange multipliers, consider

$$p_x(x) \left[(dc(x)/dx)\right]^{-2} + \lambda\, dc(x)/dx \tag{4.65}$$

The derivative with respect to dc/dx is set to zero and λ is eliminated by using (4.64). Then

$$\frac{dc_{opt}(x)}{dx} = \frac{x_{\max}}{\int_0^{x_{\max}} \sqrt[3]{p_x(x)}\,dx} \sqrt[3]{p_x(x)} \tag{4.66}$$

i.e., the optimum interval density is proportional to the cube root of the pdf of the quantizer input as in (4.61). By integrating (4.66) and choosing c such that $c_{opt}(x_{\max}) = x_{\max}$ (a boundary condition in Figure 4.11), we obtain the optimum compressor characteristic

$$c_{opt}(x) = x_{\max} \left[\int_0^x \sqrt[3]{p_x(x)}\,dx\right] \left[\int_0^{x_{\max}} \sqrt[3]{p_x(x)}\,dx\right]^{-1} \tag{4.67}$$

From this result we can find the optimum decision levels $x = x_{k,opt}$ that result when $c_{opt}(x)$ takes on the values $kx_{\max}/L$; $k = 1,2,\ldots, L$.

Also, by combining (4.63) and (4.67), we obtain [Panter and Dite, 1951]

$$\min\{\sigma_q^2\} \approx \frac{2}{3L^2} \left[\int_0^{\infty} \sqrt[3]{p_x(x)}\,dx\right]^3 \quad \text{for large } L \tag{4.68}$$

If the pdf $p_x(x)$ is unbounded, (4.68) describes only the granular contribution for bounded inputs. For large L, SNR values corresponding to (4.68) agree well with the numbers already mentioned for different pdf's in the exact results section (Table 4.4 and Figure 4.12, which actually shows asymptotic values of 6.02−max{SNR}). We must emphasize that (4.68) is valid for quantizers with many levels ($L \geqslant 32$). The assumption used in (4.63) and hence in (4.68), viz., constant input pdf over an interval, does not really hold for the outermost intervals, causing (4.68) to be inexact, especially for small L. Approximate SNR formulas that are closer to the exact result (4.12) have been proposed elsewhere [Algazi, 1966] [Roe, 1964] [Wood, 1969].

In later chapters, we shall mention nonuniform quantizers that minimize *noise audibility* or *noise visibility* rather than *noise variance*. Such quantizers are often designed on the basis of subjective experimentation. In principle, they can also be realized by replacing the probability function in (4.67) and (4.68) by appropriate subjective functions [Rubinstein and Limb, 1978].

Example 4.6. Uniform pdf

Using (4.67) and (4.68),

$$c_{opt}(x) = x\,; \qquad \min\{\sigma_q^2\} = 2^{-2R}\,\sigma_x^2 \tag{4.69}$$ •

Example 4.7. Laplacian pdf

The optimum compressor characteristic can be given in closed form [Smith, 1957] using (4.67):

$$m\text{-}law: \qquad c_{opt}(x) = x_{\max}\ \operatorname{sgn}(x)\ \frac{1 - e^{-m|x|/x_{\max}}}{1 - e^{-m}}\,; \qquad m_{opt} = \frac{\sqrt{2}}{3\sigma_x}\,x_{\max} \tag{4.70}$$

0125

$$\min\{\sigma_q^2\} = (9/2)\, 2^{-2R}\sigma_x^2\,; \quad SNR\,(\text{dB}) = 6.02R - 6.53\,; \quad \max\{\Delta\}/\min\{\Delta\} = e^m \qquad (4.71)$$

See Problem 4.7 for a numerical illustration. For a specified input-clipping level x_{max}, the optimum performance is attained only if m is chosen as in (4.70); this will provide an optimal loading factor $f_{l,opt} = 3m_{opt}/\sqrt{2}$. Equivalently, for a given design of m and x_{max}, the optimal input level is described by $\sigma_{x,opt} = \sqrt{2}\, x_{max}/3m$. The quantizer-input mismatch that results otherwise can lead to drastic reductions in SNR from the value implied in (4.71). See Example 4.8. •

4.4.3 Quantizer Mismatch

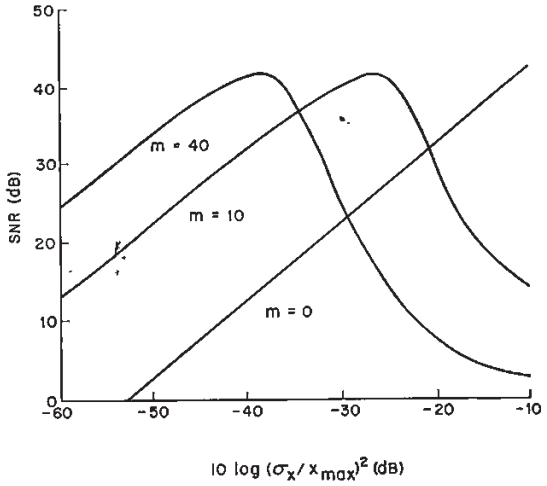
Communication systems may have to carry signals of changing statistics; for example, inputs with different variances in telephony. Therefore, optimum quantizers are often not matched to the statistics of the actual inputs. Mismatches can be in terms of variance (with a known pdf shape at the input), or in terms of pdf shape as well [Mauersberger, 1979].

Of special interest is a *mismatch in variance.* It implies that suboptimum step sizes are employed, and if the input variance is higher than the design value, it changes the ratio of granular noise and overload noise. Figure 4.8 has shown that one-bit quantizers are rather insensitive to changes in variance (or, equivalently, improperly chosen step sizes). Variance mismatches can be much more serious with $R > 1$.

Example 4.8. 8-bit m-law quantization of Laplacian input with unknown variance σ_x^2

We have mentioned already that the best m-law performance is obtained if the Laplacian input has the σ_x value $\sqrt{2}x_{max}/3m$ of (4.70), where x_{max} is the input clipping level, the saturation point in the $c(x)$ characteristic. Figure 4.13 shows the effect of a

Figure 4.13 Performance of 8-bit m-law quantization.



0126

general σ_x value that is not necessarily equal to the optimum. The results of Figure 4.13 are precise ones, following (4.68) and (4.12) rather than (4.63). The $m = 0$ curve refers to uniform quantization. For $m = 10$, the optimum occurs for $f_{l,opt} = 30/\sqrt{2}$ which is the -26 dB point on the abscissa. •

In Figure 4.13, performance is a strong function of input standard deviation σ_x. We will note in Sections 4.5 and 4.10 that with *logarithmic quantization* or *adaptive step size* procedures, a performance that is nearly independent of σ_x can be attained over a wide range of input variance unlike in Figure 4.13.

Example 4.9. 4-bit quantization with known input variance, but mismatched pdf shape

Figure 4.14 shows *SNR*'s of optimum nonuniform and optimum uniform quantizers designed for uniform (U), Gaussian (G), Laplacian (L), and gamma (Γ) pdf's. The *SNR*'s are shown for all combinations of input pdf and quantizer type. Thus, the *SNR* at abscissa point G of the Γ-curve is the value that the gamma-optimized quantizer attains if the pdf of the input is actually Gaussian.

The following conclusions emerge from Figure 4.14:

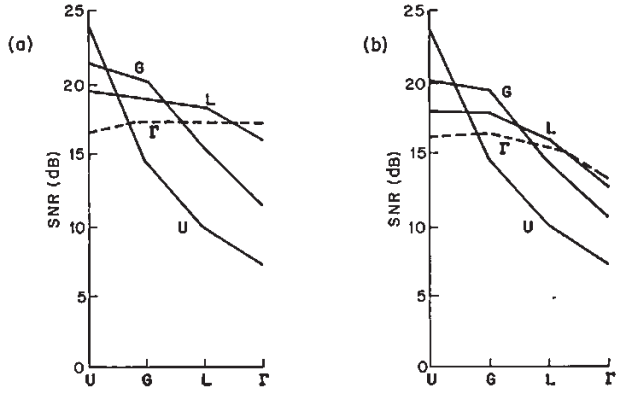
(a) The pdf-mismatch performance of the optimum uniform quantizers is very similar to that of optimum nonuniform quantizers; differences can be significant for $R > 4$.

(b) Uniform quantizers, which are optimal for signals with a uniform pdf, are an extremely bad match for signals with different pdf's. The strong decrease in performance is caused by severe overload effects.

(c) Uniform, Gaussian, and Laplacian quantizers show an improved performance if their inputs have a pdf with shorter tails; this is an improvement obtained even under pdf-mismatch conditions!

(d) Gamma quantizers achieve a signal-to-noise ratio that is nearly independent of the type of pdf. For this reason, the most significant numbers in Table 4.3 are those referring to gamma quantization. The robustness of gamma quantizers is

Figure 4.14 Effects of quantizer mismatch in (a) optimum nonuniform and (b) optimum uniform quantizers. Letters on abscissa represent input pdf and letters on curves represent pdf's used in optimizing the quantizer. (U: Uniform, G: Gaussian, L: Laplace, Γ: Gamma) [Noll and Zelinski, 1976].



0127

related to the fact that the quantization characteristic of the optimum gamma quantizer does not differ much from a logarithmic one. A logarithmic characteristic has the well-known property (Section 4.5) of giving a performance nearly independent of the shape of the input pdf. Logarithmic quantization can also provide a performance that is nearly constant over a very wide variation of input variance. ●

4.5 Logarithmic Quantization

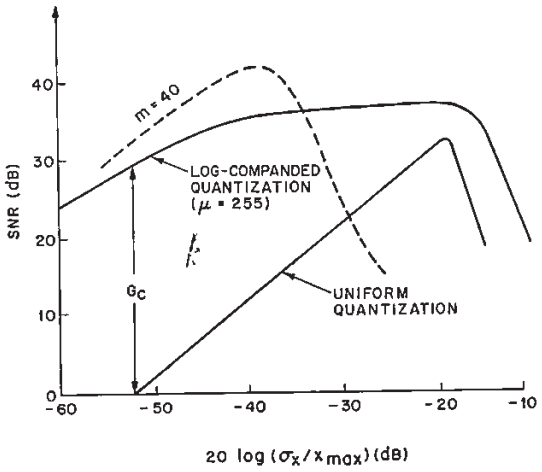
4.5.1 Basic Formulas

We have seen that for uniform quantizers the quantization noise is $\sigma_q^2 = \Delta^2/12$, i.e., it is independent of the variance of the input signal, assuming that the signal does not exceed the range of the quantizer. This implies an *SNR* that decreases with a decrease of the signal level (see Figure 4.13 for $m=0$). In the case of pdf-optimized quantizers we have seen that the *SNR* is only at its maximum for one specific variance; any deviation from this specific value leads to a significant decrease in *SNR* (see Figure 4.13 for $m=10,40$).

In situations such as speech coding, the exact value of the input variance is not known in advance; and in addition, it tends to change with time (Figure 2.6). In such situations, a signal-to-quantization noise ratio that is constant over a broad range of input variances can be obtained by using a logarithmic companding law [Holzwarth, 1949].

The benefits of logarithmic companding are illustrated in the dynamic range curves of Figure 4.15, which refer to the performance of an 8-bit $\mu 255$ quantizer

Figure 4.15 Dependence of *SNR* on σ_x^2 in 8-bit log- and uniform quantization of a bounded Laplacian input. The dashed line refers to a pdf-optimized quantizer.



0128

with a Laplacian input assumed bounded to x_{max}. Note that as in Figures 4.7 and 4.13, the uniform quantizer has a performance that is highly input-dependent. The logarithmic quantizer, on the other hand, has a *dynamic range* of about 30 dB (−15 dB to −45 dB on the abscissa) over which its *SNR* is within 3 dB of the maximum value of 38 dB. Also shown in the figure is the companding gain for small inputs, the parameter G_C of Section 4.4. Note that this gain is about 30 dB, which implies that the smallest step size min $\{\Delta\}$ is about 32 times smaller than the step size of a corresponding uniform quantizer with identical x_{max} and R. We will presently define the $\mu 255$ law and note that indeed $G_C \approx 30$ dB for this companding function. Note also that a pdf-optimized quantizer (broken curve in the figure) has a higher maximum *SNR* than the log-quantizer, but a smaller dynamic range; the pdf-optimized quantizing system is the $m = 40$ system of Figure 4.13.

For large values of x_{max}, or for truncated pdf's, the value of σ_q^2 in (4.63) leads to a signal-to-noise ratio

$$SNR = \frac{\sigma_x^2}{\sigma_q^2} = \frac{3L^2}{x_{max}^2} \frac{\int_{-\infty}^{\infty} x^2 p_x(x)\,dx}{\int_{-\infty}^{\infty} p_x(x) \left[dc(x)/dx\right]^{-2} dx} \tag{4.72}$$

Thus we can obtain a constant *SNR* if the interval density $dc(x)/dx$ is inversely proportional to x:

$$dc(x)/dx = (Kx)^{-1}; \quad K = \text{constant} \tag{4.73}$$

$$SNR = \sigma_x^2/\sigma_q^2 = 3L^2/K^2 x_{max}^2 \tag{4.74}$$

From (4.73),

$$c(x) = K^{-1} \log_e x + \text{constant}; \quad x > 0 \tag{4.75}$$

and because $c(x) = x_{max}$ for $x = x_{max}$ (Figure 4.11),

$$c(x) = \left[\frac{1}{K} \log_e \frac{|x|}{x_{max}} + x_{max}\right] \text{sgn}(x) \tag{4.76}$$

The characteristic (4.76) implies a negative infinite $c(x)$ value at the specific point $x = 0$. In order to attain a practically implementable boundary condition $c(x) = 0$ for $x \to 0$ as in Figure 4.11, we have to correct the companding law. Two proposals will be discussed in the following. In both proposals we find a linearization of $c(\cdot)$ near the origin.

4.5.2 *A*-Law Companding

The *A*-law compression characteristic [Cattermole, 1969] is based on a linear companding characteristic near the origin ($|x| \leq x_{max}/A$) and a logarithmic

0129

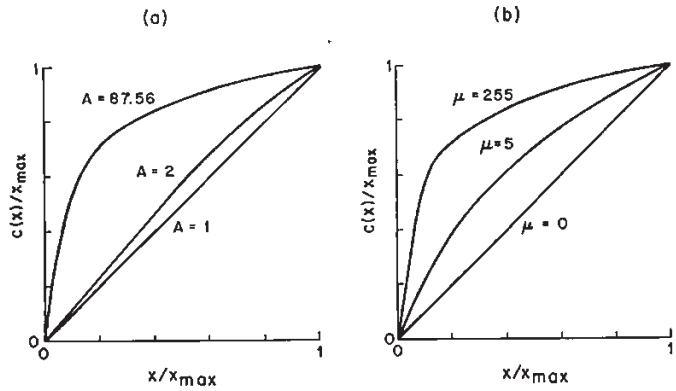


Figure 4.16 Logarithmic compression characteristics (shown for $x > 0$): (a) A-law and (b) μ-law.

companding characteristic for large x [Figure 4.16(a)]:

$$c(x) = \begin{cases} \dfrac{A|x|}{1+\log_e A}\operatorname{sgn}(x) ; & 0 \leqslant \dfrac{|x|}{x_{\max}} \leqslant \dfrac{1}{A} \\ x_{\max}\dfrac{1+\log_e (A|x|/x_{\max})}{1+\log_e A}\operatorname{sgn}(x) ; & \dfrac{1}{A} < \dfrac{|x|}{x_{\max}} \leqslant 1 \end{cases} \tag{4.77}$$

From (4.73), (4.74) and (4.77), we find that for *large signals*,

$$K = (1+\log_e A)/x_{\max}; \quad \sigma_x^2/\sigma_q^2 = 3L^2/(1+\log_e A)^2 \tag{4.78}$$

$$SNR_{A\text{-}law}\ (\text{dB}) = 6.02R + 4.77 - 20\log_{10}(1+\log_e A) \tag{4.79}$$

The companding gain is

$$G_C = dc(x)/dx|_{x=0} = A/(1+\log_e A) \tag{4.80}$$

and, using (4.63), we find that for *small signals*

$$\sigma_x^2/\sigma_q^2 = 3L^2(\sigma_x^2/x_{\max}^2)G_C^2 \tag{4.81}$$

Recall that the *SNR* of a uniform quantizer with negligible overload (4.31) is

$$\sigma_x^2/\sigma_q^2 = 3L^2(\sigma_x^2/x_{\max}^2) ; \qquad |x| \leqslant x_{\max}$$

Comparison of the above two results implies that the step size in the linear part of $c(x)$ is reduced by the companding gain G_C. The equivalent increase of *SNR* for *small signals* is described by

$$SNR_{A\text{-}law}\ (\text{dB}) = SNR_{uniform}\ (\text{dB}) + 20\log_{10} G_C ; \quad |x| \to 0 \tag{4.82}$$

0130

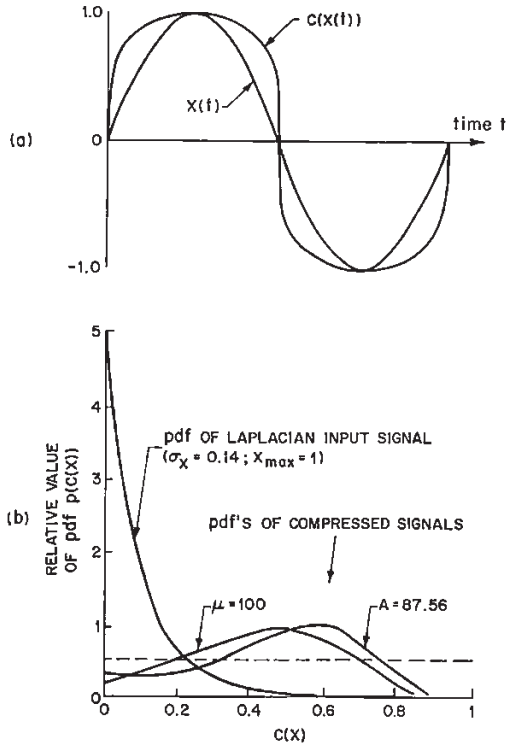
For $A = 87.56$, this increase is 24.08 dB. A uniform quantizer would need four additional bits for the same *SNR* at low levels; although its *SNR* at high levels would be higher than that of the A-law quantizer.

It can be shown (Problem 4.10) that A-law companding provides a step size ratio

$$\max\{\Delta\}/\min\{\Delta\} = A \tag{4.83}$$

The ratio increases linearly with A; recall that in Laplacian-optimized m-law companding, the ratio is an exponential, e^m. Figure 4.16(a) demonstrates A-law compression characteristics and Figure 4.17(a) shows the effect of compression on a sine-wave input. Notice how smaller amplitudes are effectively amplified relative to higher amplitudes. The effect of the logarithmic compressor can also be described by the way it changes the input pdf. The general tendency, for any input pdf, is to obtain an output pdf that more closely resembles the uniform pdf **U**. For example, with a Laplacian input, the effect of μ-law or A-law compression is to produce a

Figure 4.17 (a) Modification of sine-wave by A-law and μ-law compression ($A = 87.56$, $\mu = 255$) [Steele, 1975. Reprinted with permission]. (b) Modification of Laplacian pdf by A-law and μ-law compression ($A = 87.56$, $\mu = 100$) [Sundberg and Rydbeck, 1976. Reprinted with permission].



compressed signal pdf $p[c(x)]$ that is much flatter; with a mode, or $p[c(x)]$-maximizing value that is no longer at $c(x) = 0$ [Figure 4.17(b) and Problem 4.9; also see Problem 4.8]. *SNR* curves for A-law quantization, as a function of input power, will be similar to the μ-law dynamic range curve in Figure 4.15. Further comparisons of A-law and μ-law characteristics will appear in Chapter 5.

Example 4.10. $A = 87.56$ **[European PCM standard (Chapter 5)]**

$$SNR_{A\text{-}law}\ (dB) = 6.02R - 9.99$$

$$G_C = 24.08\ dB\ ; \quad \max\{\Delta\}/\min\{\Delta\} = 87.56 \quad \bullet$$

4.5.3 μ-Law Companding

The μ-law compression characteristic [Holzwarth, 1949] [Panter and Dite, 1951] [Smith, 1957] is

$$c(x) = x_{\max} \frac{\log_e (1+\mu|x|/x_{\max})}{\log_e (1+\mu)} \operatorname{sgn} x \tag{4.84}$$

This characteristic (Figure 4.16(b)) is quasi-logarithmic; it is linear for small x-values [$\log_e (1+ax) \approx ax$]. For *large signals* ($\mu|x| >> x_{\max}$), it is logarithmic:

$$c(x) = \frac{x_{\max}}{\log_e (1+\mu)} \log_e (\mu|x|/x_{\max}) \operatorname{sgn} x \tag{4.85}$$

From (4.73) and (4.85),

$$K = \log_e (1+\mu)/x_{\max} \tag{4.86}$$

and we find, using (4.74), that the *SNR* is again independent of the statistics of the input signal:

$$\sigma_x^2/\sigma_q^2 = 3L^2/[\log_e (1+\mu)]^2 \qquad \text{for } \mu|x| >> x_{\max} \tag{4.87}$$

$$SNR_{\mu\text{-}law}\ (\text{dB}) = 6.02R + 4.77 - 20 \log_{10}(\log_e (1+\mu)) \tag{4.88}$$

For *small signals* we obtain the companding gain

$$G_C = dc(x)/dx|_{x\to 0} = \mu/\log_e (1+\mu) \tag{4.89}$$

It can be shown (Problem 4.11) that μ-law provides a step size ratio

$$\max\{\Delta\}/\min\{\Delta\} = 1+\mu \tag{4.90}$$

Example 4.11. μ**=255 [North American PCM standard (Chapter 5)]**

$$SNR_{\mu\text{-}law}\ (\text{dB}) = 6.02R - 10.1$$

0132

$$G_C = 33.25 \text{ dB} ; \quad \max\{\Delta\}/\min\{\Delta\} = 256 \quad \bullet$$

The log-quantizer results of Examples 4.10 and 4.11 are primarily applicable to speech coding, and they can be summarized, as in (1.6), in the form

$$SNR \text{ (dB)}_{\textit{log-quantized speech}} = 6.02R - 10.1 \tag{4.91a}$$

This should be contrasted with (4.32), the uniform quantizer and uniform pdf result that is more representative of image coding:

$$SNR \text{ (dB)}_{\textit{uniformly quantized image}} = 6.02\ R \tag{4.91b}$$

Image *SNR* results are sometimes defined as peak-to-peak-signal to rms-noise ratios. These numbers are higher than the values in (4.91b) by about 10.8 dB (Problem 5.2).

4.5.4 Quantizer-Input Mismatch

We have shown in Figure 4.15 that the logarithmic quantizer is very insensitive to a *quantizer mismatch relative to variance.* For example, consider a Laplacian pdf. The pdf-optimized quantizer has an *SNR* of $6.02\ R - 6.53$ (dB) as given in (4.71), whereas the log-characteristic (e.g., $\mu = 255$) results in a smaller *SNR* of $6.02\ R - 10.1$ (dB). On the other hand, the pdf-optimized quantizer can be very sensitive to a mismatch in variance or pdf shape (Figures 4.13 and 4.14). It is obvious that the log-quantizer is extremely robust. The log-quantizer is also very insensitive to a mismatch in relative shape of the pdf, for a given variance. Differences in the quantization of different pdf's are only significant in the overload region. This is indicated by the dashed overload-region skirt in Figure 4.15.

The illustration in Figure 4.15 is for the example of $R = 8$ bits/sample, an example of great practical importance (Chapter 5). For much coarser designs such as $R < 4$ bits/sample, the approach of adaptive quantization (Section 4.10) will be seen to be even more efficient than time-invariant log-quantization.

4.6 Entropy-Coded Quantization

4.6.1 Entropy Coding — Concept and Implementation

In our previous discussion we have concentrated on the quantizer characteristic $Q(\cdot)$ and on its corresponding quantization error. An input random variable X is represented at the quantizer output as a discrete-valued random variable $Y = Q(X)$ such that

$$Y = y_k \quad \text{if} \quad X \in \mathcal{I}_k \quad \text{for} \quad k = 1,2,...,L \tag{4.92}$$

Clearly, y_k is a member of an L-ary alphabet of capacity $\log_2 L$ (Appendix C). Actually, it is the L-ary index k of representation level y_k that is transmitted to

0133

the receiver. The channel maps k into k if the channel is error-free and the receiver maps k into representation level y_k.

The L-ary index of the chosen output is typically encoded in a binary format with

$$R \geqslant \log_2 L \quad \text{bits/sample} \tag{4.93}$$

R has to be an integer: the smallest integer that is needed to describe L numbers in a binary format. For example, $R=3$ for $L=8$, and also for $L=5$. When L is not a power of 2, transmission with about $\log_2 L$ bits/sample is still possible by appropriate *block coding* where *sequences* of L-ary indices are converted into sequences of binary digits. For example, consider $5^3=125\sim128=2^7$; there are 125 possible 5-ary letter blocks of length 3; these can be encoded very efficiently by binary sequences that are 7 bits long. Table 4.6 and Problem 4.12 discuss other examples where L is not a power of 2.

The quantizer output can be viewed as a discrete-valued information source (Appendix C) with an L-ary alphabet. The average information content of this source is given by the *zeroth-order entropy* of the quantizer output:

$$H_Q = -\sum_{k=1}^{L} P_k \log_2 P_k \tag{4.94}$$

P_k is the probability of an occurrence of quantizer output y_k:

$$P_k = P\{X \in \mathcal{I}_k\} = P\{Y = y_k\} = \int_{x \in \mathcal{I}_k} p_x(x)dx \tag{4.95}$$

Note that H_Q represents the average information content only if the quantizer outputs are statistically independent. In optimized coding schemes such as DPCM with optimal prediction (Chapter 6), this is always an approximately valid assumption. The importance of H_Q stems from the *noiseless coding theorem* for discrete sources (Appendix C): a discrete-valued source can be encoded with an average bit rate $\bar{R}$ that is arbitrarily close to H_Q:

$$\min\{\bar{R}\} = H_Q + \epsilon \quad \text{bits/sample}; \qquad \text{arbitrary } \epsilon > 0 \tag{4.96}$$

such that perfect reconstruction is possible at the receiver. There are at least two techniques that can realize this ideal, *permutation coding* [Berger, 1982] and the approach used in this book, *entropy coding*. Entropy coding is obtained by means of a *variable length coding* procedure which assigns codewords of variable lengths to the possible outcomes y_k such that highly probable outcomes are assigned shorter codewords, and vice versa. If the entropy coding is ideal, the bit rate at the output of the L-level quantizer can be reduced by

$$\Delta R = \log_2 L - H_Q \tag{4.97}$$

In fact, ΔR is the *redundancy* of the discrete-valued source. For example, if $L = 4$, $P_1 = P_4 = 1/8$ and $P_2 = P_3 = 3/8$, and if we assume statistically

0134

independent quantizer outputs, $\log_2 L = 2$, $H_Q = 1.81$, and $\Delta R = 0.19$ bit/symbol. It is obvious that no reduction is possible if all outcomes are equally probable, i.e., $P_k = 1/L$. Indeed, from (4.94), $H_Q = \log_2 L$ in that case, i.e., the entropy equals the capacity of the alphabet. In all other cases, a savings of ΔR is possible by entropy coding: here, indices of the quantizer outputs which occur with probability P_k; $k = 1,2,...,L$ are coded with codewords of different lengths.

Let y_k be assigned a codeword of n_k bits which may be smaller, equal to or greater than $\log_2 L$. The average length of the binary codewords also defines the *average bit rate*:

$$\bar{R} = \sum_{k=1}^{L} P_k \, n_k \quad \text{bits/sample}$$

and the obtained reduction is

$$\log_2 L - \bar{R} \quad \text{bits/sample} \tag{4.98}$$

The *optimum code* is given for

$$n_k = -\log_2 P_k \tag{4.99}$$

We then have $\bar{R} = H_Q$ and hence the bit rate reduction equals ΔR, as given by (4.97). This optimal code can only be reached if the probabilities satisfy the Shannon-Fano integer constraint

$$P_k = 2^{-n_k} \tag{4.100}$$

In all other cases, the average length $\bar{R}$ that results from entropy coding will be slightly higher than H_Q. Appendix C shows that it may be useful to use entropy coding on sequences (rather than on single outputs) in order to make the term ϵ of (4.96) as small as possible. This is true even for independent, identically distributed samples. The effect of coding in sequences is to provide a better approach to the *Shannon-Fano integer constraint.* Appendix C further demonstrates the constructive procedure of Huffman coding [Huffman, 1952], which seeks to realize the existence results just summarized.

Appendix C also discusses discrete sources with memory. If the source memory is significant, entropy coding can be quite complex since it must be based on processing long blocks and since joint probabilities must be known to construct the codebook. *In the practical approaches emphasized in this book, linear redundancies are removed by linear filtering prior to quantizing.* The quantizer outputs are then uncorrelated and the minimum rate to convey the information about the output sequence to the receiver is again given by the zeroth-order entropy H_Q.

Table 4.5 demonstrates the benefits of entropy coding for an example with $L = 4$ and $H_Q = 1.49$. As discussed in Appendix C, variable-length codes need special properties or structure to be useful. The reader should verify that code EC1 is useless, being *not uniquely decipherable*; code EC2 is decipherable but *non-instantaneous* (in terms of decoding); codes EC3 and EC4 are *instantaneous* in

0135

Table 4.5 Codes for a 4-level quantizer (NBC: Natural Binary Code, FBC: Folded Binary Code, EC1 to EC4: Entropy Codes).

output	P_k	NBC	FBC	EC1	EC2	EC3	EC4
y_1	0.60	00	00	0	0	0	0
y_2	0.25	01	01	1	01	10	10
y_3	0.10	10	11	01	011	110	110
y_4	0.05	11	10	10	111	111	1110
$\bar{R}$		2	2	1.15	1.55	1.55	1.6

that their codewords can be decoded as soon as the last digit of a codeword has been received; and code EC4 is *self-synchronizable* in that each codeword ends with a specific terminating symbol or *comma*, 0.

Buffer Considerations. The source, and hence the quantizer, produces samples at a constant rate. If these samples are mapped into codewords of variable length, these encoder outputs must be initially fed into a buffer, to accommodate a communication channel that accepts symbols at a constant rate. The buffer tends to fill when a succession of long codewords is generated. This is the case if the random variable assumes improbable values on several successive samples. A long buffer must be employed to prevent *buffer overflow* that would cause a loss of samples and hence a large distortion. On the other hand, the buffer tends to empty if highly probable codewords (which have short lengths) are generated with a local probability that exceeds the average probability. This is the situation of *buffer underflow*. In this case the transmitter must resort to *bit stuffing*, and transmit a dummy signal to mark time until it has enough non-trivial information to put out.

It can be shown [Jelinek, 1968] that any buffer will eventually overflow with probability 1 in the case of stationary signals. Nonstationary signals are even more difficult to encode since the local entropy may be quite different from the long term entropy. For example, in image coding, *high entropy* or *high detail* areas of the image cause the buffer to become full. High entropy areas are the *busy* areas characterized by frequent changes of brightness. There is no such problem in *low entropy* or *low detail* areas which are also called the *quiet* areas of the image. Anyway, buffer overflow will occur whenever the information rate is higher than the transmission rate for a long enough period of time. The easiest way of preventing overflow is to have a buffer-feedback control of the quantizer, as in Figure 4.18. In speech applications, where *adaptive quantization* is often employed, the quantizer range is expanded whenever the buffer tends to overflow. It is also possible to conceive designs which change only the quantizer thresholds, and keep the reconstruction values fixed. Such systems have the advantage that the decoder at the receiver will be the same as that of a system without entropy coding. In image coding, the quantizer is typically fixed, and the easiest way to change quantizing characteristics is to switch between two quantizers. We shall mention these entropy coding systems again in Chapter 6, in the context of DPCM coding of speech and images.

0136

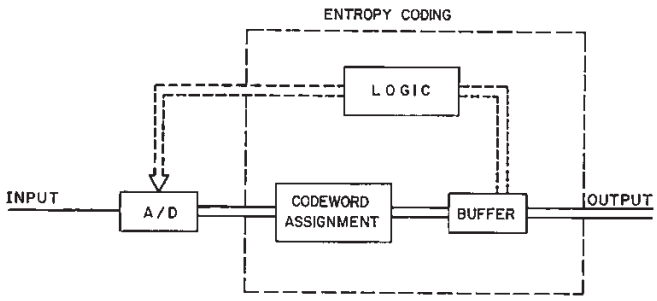


Figure 4.18 Entropy coding with quantizer control.

So far we have mentioned nothing about the receiver buffer. The status of the transmit buffer need not be transmitted provided that the channel is error-free. This is because transmit and receive buffers are strongly related. When the transmit buffer is approximately full the receiver buffer is nearly empty, and vice versa, and the total number of samples in the two buffers equals a constant.

Synchronization Problems. A single channel error can cause loss of codeword synchronization which leads to a long sequence of erroneous receiver outputs. This problem can be minimized by proper code selection which will permit re-acquisition of synchronization quickly. An example is the self-synchronizing code EC4 in Table 4.5. Codeword synchronization is obtained as soon as a 0 appears. In fact, the 0 is the synchronizing symbol or *comma*. Decoding is simply a matter of counting the 1's before the comma.

An alternative solution is a *variable-length input to fixed-length output code.* For this code, each final codeword of given fixed length may represent a different number of source output symbols. Table 4.6 is an example for 3-level quantizers [Cohn and Melsa, 1975] and three possible outputs A, B, and C, where B is the output of highest probability. The code accumulates a sequence of source symbols until a message in the output dictionary is formed. Note that in this code, all channel codewords have the same length so that channel errors do not lead to long losses of codeword synchronization.

Table 4.6 Fixed-length code for variable-length input from ternary source. B: center level of high probability. A,C: outer levels of low probability [Cohn and Melsa, 1975].

Source Word	Codeword
BBB	000
BBA	001
BBC	010
BA	011
BC	100
A	101
C	110

0137

4.6.2 Entropy Coding of Quantizer Outputs

We shall now re-examine the bit rate reduction parameter ΔR in (4.97). We shall show that $\Delta R_{pdf-opt}$, the value of ΔR that results in mmse pdf-optimized quantization (Section 4.4) is *not* the best value achievable by entropy coding; and that at high bit rates, the best value of ΔR is in fact given by $\max\{\Delta R\} = (3/2)\Delta R_{pdf-opt}$. Further, this best configuration will involve a *uniform threshold* quantization characteristic with $L \to \infty$ for any $\bar{R}$; and at high bit rates $\bar{R}$, its performance falls short of the rate distortion bound by 1.53 dB for any input pdf, whereas at low bit rates $\bar{R}$, its performance approaches this bound [Berger, 1982].

Table 4.7 shows the results of Huffman-coding of pdf-optimized minimum mean square error (mmse) quantizers, for the example of a Gaussian input pdf. Note that the entropy H_Q of the quantizer output symbols is only slightly lower than the capacity $\log_2 L$ of the alphabet. We will show presently that the above gap can be made even smaller if the quantizers are re-optimized such that efficient entropy-coding is possible. The re-optimization results in a redistribution of the probabilities P_k in such a way that there are only few symbols with comparatively high values of the probability (and therefore short codewords) and a majority of symbols with small probabilities. The problem of designing an optimum quantizer will be reformulated as that of obtaining decision and reconstruction levels to minimize the entropy, subject to a constraint on the mean square error σ_q^2. Such entropy-coded quantizers are sometimes called *entropy-constrained* quantizers. This is in contrast to *level-constrained* quantizers discussed hitherto, whose bit rate is $R = \log_2 L$.

A quantitative description of entropy coding bounds is given in Figure 4.19(a). For any σ_q^2 and L, there is one quantizer characteristic $Q(\cdot)$ that has minimum output entropy. In Figure 4.19(a) the curve marked $L = 8$ describes the (minimum) H_Q versus σ_q^2 performance of eight-level quantizers. Point a refers to the pdf-optimized mmse quantizer. This is the level-constrained quantizer with $H_Q = 2.825$ (Row 2 of Table 4.7). The output entropy in this case is H_a and the bit rate reduction that is possible with entropy coding is $\Delta R_{pdf-opt}$. We note that for a given L, H_Q can be reduced at the cost of a higher σ_q^2. The absolute minimum is $\min\{H_Q\} = 1$ since we have assumed an even number L of levels, which means that at least one bit is needed to assign codewords to positive and negative outputs. We shall see shortly that smaller values of entropy can be

Table 4.7 Huffman coding of the output of a Gaussian-pdf-optimized quantizer. (H_Q: output entropy; *SNR*: that of Gaussian Max quantizer, in dB; $\bar{R}$: average length of codewords with Huffman coding based on single outputs) [Zelinski and Noll, 1972].

L	$\log_2 L$	H_Q	$\bar{R}$	*SNR*
4	2	1.911	1.989	9.30
8	3	2.825	2.884	14.62
16	4	3.765	3.809	20.22
32	5	4.730	4.771	26.01

0138

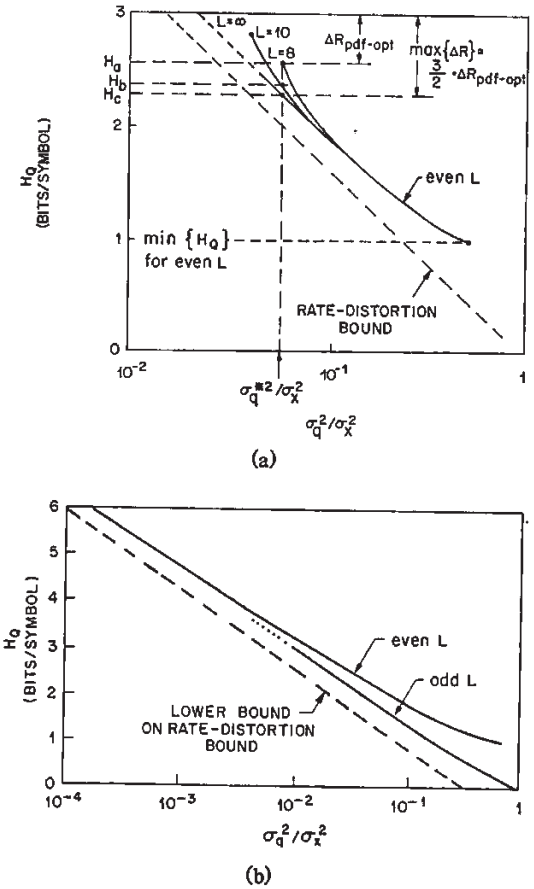


Figure 4.19 H_Q versus σ_q^2/σ_x^2. Results for (a) a quantizer with an even number of levels and a Gaussian-pdf input [Zelinski and Noll, 1972]; and (b) for gamma-pdf inputs [Granzow and Noll, 1983].

realized by employing an odd number of levels L. On the other hand, consider a fixed $\sigma_q^2 = \sigma_q^{*2}$. It is seen that the entropy can be reduced at this distortion level by increasing the number of levels. With $L = 10$, we obtain H_b, and by letting $L \to \infty$, an absolute minimum H_c is obtained. The savings in bit rate is then $\max\{\Delta R\} = \log_2 L - H_c$. Note also that this falls short of the rate distortion bound by 0.25 bit, as shown by the $H_Q = 2^{-2R}\sigma_x^2$ curve in Figure 4.19(a). We will note presently that at high bit rates, $\max\{\Delta R\} = (3/2)\ \Delta R_{pdf-opt}$.

We see that the numbers of quantizer steps can be made arbitrarily high in the case of entropy coding: outer steps will have a very small probability of occurrence so that their contribution to the *average* length of codewords will be very small. On the other hand, for a given entropy, we obtain a decrease in distortion because even very high-valued input samples can be coded without overload distortion. The

0139

following sections will analyze the savings obtainable from entropy coding the quantizer outputs with or without quantizer reoptimization. In the case of high entropies (high average bit rates) we will obtain results in closed form using the large-L approximation (4.63) for σ_q^2. The main result will be that for a given distortion, a *uniform-threshold* quantizer will yield the minimum entropy, and that this minimum is higher than the rate distortion bound (the absolute minimum) by only about 1/4 bit (as in Figure 4.19); a uniform-threshold quantizer is one that has uniform intervals between decision levels. In the case of small values of entropy, we will present results as obtained from iterative computer procedures. The main result is again that of a uniform-threshold quantizer, but analyses will be based on the exact σ_q^2 result (4.12) and solutions will be computer-based and non-explicit.

Entropy Coding at High Average Bit Rates. When the number of quantizer steps is sufficiently high, we may use the approximation formulas of Section 4.4 based on the companding law $c(x)$.

The entropy of the L-level quantizer output (4.94) can be split into two terms if we make use of the result $P_k = p_x(y_k)\Delta_k$:

$$H_Q = -\sum_{k=1}^{L} p_x(y_k)\Delta_k \log_2 p_x(y_k) - \sum_{k=1}^{L} p_x(y_k)\Delta_k \log_2 \Delta_k \qquad (4.101)$$

For large L and statistically independent inputs, the first term becomes the *differential entropy* of the quantizer input

$$h(X) = E[-\log_2 p_x(X)] = -\int_{-\infty}^{\infty} p_x(x) \log_2 p_x(x)dx \qquad (4.102)$$

The differential entropies are given in Table 4.8 for various pdf's. Using (4.42) to relate Δ_k to $c(x)$, and using (4.101) and (4.102),

$$H_Q = h(X) + \log_2 \frac{L}{2\, x_{\max}} + \int_{-\infty}^{\infty} p_x(x) \log_2 \frac{dc(x)}{dx}\, dx \qquad (4.103)$$

This result gives the entropy for any companding law as given by its interval density dc/dx. It can be verified (Problem 4.13) that in the special case of a uniform pdf input and zero overload, H_Q in (4.103) reduces to $\log_2 L$.

The entropy of the quantizer output can be reduced (by an amount no greater than $\max\{\Delta R\}$ of Table 4.8) by taking into account the fact that one can employ an arbitrary number of quantizer levels in the case of entropy coding; and it can be shown then (Problem 4.14), as in columns 4 and 5 of Table 4.8, that

$$\max\{\Delta R\} = (3/2)\ \Delta R_{pdf-opt} \qquad (4.104)$$

A practical approach will be to seek a minimum of the output entropy for a given distortion σ_q^2; i.e., we choose the companding law $c(x)$ such that H_Q of (4.103) is minimized subject to the constraint of a fixed mse as given by the large-L

Table 4.8 Entropy Coding with different input pdf's (U: Uniform, G: Gaussian, L: Laplace, Γ: Gamma) [Noll and Zelinski, 1978]. C = 0.5772 is the Euler Constant. Values in the last column are maximum *SNR* gains due to entropy coding.

pdf	$h(X)$	ϵ_*^2	$\Delta R_{pdf-opt}$ (bits/sample)	max{ΔR} (bits/sample)	ΔSNR_{EC} = 6.02 max{ΔR} (dB)
U	$\frac{1}{2}\log_2 12\sigma_x^2$	1.00	0.000	0	0
G	$\frac{1}{2}\log_2 2\pi e\sigma_x^2$	2.71	0.312	0.467	2.81
L	$\frac{1}{2}\log_2 2e^2\sigma_x^2$	4.50	0.623	0.934	5.62
Γ	$\frac{1}{2}\log_2 \frac{4\pi}{3e^{C-1}}\sigma_x^2$	5.68	1.138	1.707	10.28

approximation (4.63). We use the Lagrange multiplier technique. Consider

$$p_x(x)\log_2\frac{dc(x)}{dx}+\lambda p_x(x)\left[\frac{dc}{dx}\right]^{-2} \tag{4.105}$$

Setting the derivative with respect to dc/dx to zero,

$$dc_{opt}(x)/dx = (-\lambda/3)^{1/2} = constant \tag{4.106}$$

and with the boundary conditions $c(x_{max}) = x_{max}$ and $c(0) = 0$ of Figure 4.11, we obtain

$$c_{opt}(x) = x \tag{4.107}$$

This means that *uniform quantization* leads to *a minimum of the output entropy for a given distortion*, provided that the step size is small enough to support the assumption of constant pdf used in (4.63). Using (4.107) in (4.103), we obtain [Wood, 1969]

$$\min\{H_Q\} = h(X) - \log_2\Delta = h(X) - (1/2)\log_2 12\sigma_q^2 \tag{4.108}$$

where $\Delta = 2x_{max}/L$ is the quantizer step size and $\sigma_q^2 = \Delta^2/12$.

Comparison with the Rate Distortion Bound. We will now compare this minimum average bit rate $\bar{R} = \min\{H_Q\}$ with the rate distortion lower bound ${}^LR$ that applies to a memoryless source of pdf $p_x(x)$ and differential entropy $h(X) = E[-\log_2 p_x(X)]$ [(D.22) and Table C.2]:

$${}^LR = h(X) - (1/2)\log_2 2\pi e\sigma_q^2 \tag{4.109}$$

$$\min\{H_Q\} = {}^LR + (1/2)\log_2 \pi e/6 = {}^LR + 0.255 \tag{4.110}$$

0141

Thus we find that entropy coding with $\bar{R} = \min\{H_Q\}$ bits/sample based on a uniform quantizer is only $(1/2)\log_2(\pi e/6)$ bits or 0.255 bits above the lower bound, irrespective of the pdf of the signal. This result has been derived for Gaussian signals both experimentally [Goblick and Holsinger, 1967] and theoretically [Gish and Pierce, 1968]. Recall that the above statement assumes very small step sizes and hence high average bit rates. At these high rates the rate distortion function is asymptotically close to the lower bound ${}^{L}R$. The small difference of 0.255 bits translates to an approximate 1.5 dB difference on the basis of the $6.02R$ formulas noted repeatedly in this chapter. The above difference from the theoretical bound is the price paid for having the simplicity of single-letter or memoryless quantization.

In summary, a non-entropy-coded L-level quantizer can perform with an error variance of $\min\{\sigma_q^2\} = \epsilon_c^2\sigma_x^2/L^2$ if it is pdf-optimized [$c(x)=c_{opt}(x)$ of (4.67)]. Its bit rate is then $R = \log_2 L$. A re-optimized (and hence uniform) quantizer can perform with same variance $\min\{\sigma_q^2\}$ with (average) bit rate $\bar{R} = \min\{H_Q\}$ as given in (4.110). The maximum obtainable bit rate reduction with single-letter quantizing systems followed by entropy coding is

$$\max\{\Delta R\} = \log_2 L - \min\{H_Q\} = (1/2)\log_2 12\epsilon_c^2\sigma_x^2 - h(X) \qquad (4.111)$$

These are the values listed in column 5 of Table 4.8. With a Gaussian pdf, the gain is only 0.467 bit/sample. But in the case of the gamma pdf, about 1.7 bits per sample can be saved by entropy coding. Alternatively, for a given average bit rate, entropy coding will increase the SNR by $\Delta SNR_{EC} = 6.02 \max\{\Delta R\}$ dB over that of a non-entropy-coded Max quantizer. These values are also listed in Table 4.8. By comparing these with the values given in Figure 4.12 for large R, we find that SNR values for Gaussian and Laplacian inputs are now much closer to the value of $6.02\,R$; and for gamma sources, they are actually higher than $6.02\,R$, by 2.73 dB. The improvements in entropy coding are, of course, a consequence of employing an arbitrarily high number of quantizer levels. It is important to mention again that these additional levels of the quantizer are *not* used to increase the resolution in the region where the pdf of the input is significant (this would cause a significant increase in entropy), but they are used to assign representation levels to the outer parts of the pdf where they do not contribute significantly to entropy.

The result $\min\{H_Q\} = h(X) - \log_2\Delta$ in (4.108) has a simple information-theoretical interpretation. The differential entropy $h(X)$ of the random variable at the quantizer input is a measure of the average information of the source. The information at the input of the quantizer is partly destroyed by the effect of quantization. However, the quantization error Q is uniformly distributed in the interval $(-\Delta/2,\Delta/2)$ since we have assumed small step sizes Δ, and a high number of levels that prevents overload. Its differential entropy is

$$h(Q) = E[-\log_2 p_q(Q)] = \log_2\Delta \qquad (4.112)$$

Assuming that X and Q are statistically independent, we obtain $\min\{H_Q\}$, the average information of the quantized output, as the difference between $h(X)$ and $h(Q)$. This is the difference in (4.108).

0142

Entropy Coding at Low Average Bit Rates. The discussion of the preceding section was based on the assumption of a sufficiently high (average) bit rate. The derivation used (4.63) which assumed that the pdf is approximately constant over any of the intervals; however, this will not be true if the average bit rate is low. A different approach has to be used in order to find the optimum entropy-encoded quantizers.

We minimize the exact expression for quantization error variance in (4.12), subject to the constraint of a given entropy H_Q (4.94): The necessary conditions for a minimum in error variance are [Berger, 1972] [Zelinski and Noll, 1972]:

$$\frac{\partial \sigma_q^2}{\partial y_k} + \lambda \frac{\partial H_Q}{\partial y_k} = 0 \quad ; \quad k = 1,2,...,L \tag{4.113a}$$

$$\frac{\partial \sigma_q^2}{\partial x_k} + \lambda \frac{\partial H_Q}{\partial x_k} = 0 \quad ; \quad k = 2,3,...,L \tag{4.113b}$$

where λ is the Lagrange multiplier. However, H_Q does not depend on the reconstruction levels y_k; hence, from (4.113a), we obtain again the result of the pdf-optimized quantizer: the reconstruction level should be the centroid of the pdf in its interval. The minimization problem can be solved iteratively on a digital computer. The minimization results in a H_Q-versus-σ_q^2 performance as given for Gaussian memoryless sources in the quantitative result of Figure 4.19(a). Figure 4.19(b) shows the H_Q-versus-σ_q^2 performance for a memoryless gamma source, for entropy-coded quantizers with both even and odd numbers of levels L. Note the significant reduction in distortion, for a given entropy, in the case of odd L. This is because successive samples of the source will have values close to zero with high probability. The midtread characteristic of an odd-L quantizer will then lead to long clusters, or sequences of zero-value outputs that can be assigned the shortest possible codeword of length 1. Longer codewords, signifying sequences including one or more non-zero outputs, will have much smaller probabilities with the gamma pdf input.

Example 4.12. Entropy coding of a gamma source with $R = 1$ bit/sample

Consider a midtread quantizer with an odd number of levels L. As noted in the first row of Table 4.9, an *SNR* of 7.80 dB can be obtained using a step size of $\Delta = 2.003\sigma_x$. Experiments have shown that a quantizer with the specific design of $L = 9$ yields a very close result, 7.79 dB. We will now show how the output of this system can indeed be entropy coded for the rate of 1 bit/sample.

Consider the coding of sequences of three successive output samples. Of the $L^3 = 9^3 = 729$ possible sequences, the sequence of three (midtread) zeros has a probability of 0.536, and it is assigned a codeword of length 1. Note that this nearly satisfies the Shannon-Fano integer constraint (4.100) because $2^{-1} \sim 0.536$. All other codewords have lengths of at least 4 bits. These codewords all have probabilities in the neighborhood of 0.055; once again, this follows the Shannon-Fano integer constraint closely, because $2^{-4} \sim 0.055$. As a result, the encoding system is very efficient. However, to realize a synchronous output rate of 1 bit/sample, buffer lengths needed are in the order of 800 to 1000 bits [Granzow and Noll, 1983]. Also see Problem 4.15. •

0143

Table 4.9 Maximum *SNR*-values (in dB) and corresponding threshold step sizes for entropy-coded memoryless Γ-sources [Granzow and Noll, 1983]. Also included for comparison are *SNR* values with pdf-optimized (Max-) quantizers (Table 4.4) and RDF-bounds (Table D.1).

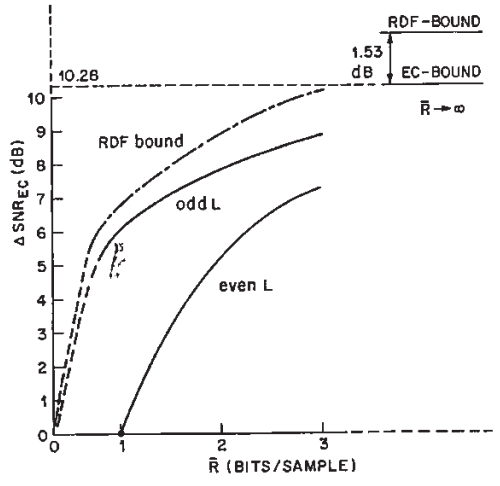
$\bar{R}$	Max - quantizers	Entropy-coded quantizers				RDF-
		L even		L odd		
bits/sample	*SNR*	*SNR*	Δ/σ_x	*SNR*	Δ/σ_x	bounds
1	1.76	1.76	∞	7.80	2.003	8.53
2	6.33	11.60	0.965	14.15	0.816	15.21
3	11.47	18.78	0.400	20.36	0.370	21.59

Table 4.9 lists maximum *SNR* values in the entropy coding of a gamma source, and compares them with the performance of a Max-quantizer, and with rate distortion bounds.

Figure 4.20 shows the gain ΔSNR_{EC} obtained by entropy-coding the Γ-source, instead of using a Max-quantizer. The figure is based on the values of Table 4.9. Note that the maximum gain $\Delta SNR_{EC} = 10.28$ dB of Table 4.8 can be obtained only for $\bar{R} \to \infty$. From that table, it is also clear that L- and G-sources have significantly smaller asymptotic values. Entropy coding with even L-quantizers and finite $\bar{R}$-values have been given elsewhere [Noll and Zelinski, 1978] for U, G, L and Γ sources.

The above bounds for single-letter quantization refer to quantizers with a theoretically infinite number of levels. A finite number of levels has to be used in practical applications. In practice, it has been shown that even for small L values,

Figure 4.20 Maximum gain in *SNR* over Max-quantization due to entropy coding of a memoryless gamma-pdf source [Granzow and Noll, 1983]. The dashed curve is the rate distortion bound, also expressed as a gain over Max-quantizer performance [Noll and Zelinski, 1978].



entropy coding performance is very close to the best possible entropy coding results. See Example 4.12.

Note that the gap between the RDF bound and entropy-coding performance decreases with decreasing R until it vanishes at rate zero. Also, see [Berger, 1982] for Gaussian and Laplacian examples. For large R, there is the 1.53 dB difference implied in the 0.255 bit gap of (4.110). Other sections of this book discuss the bridging of this residual gap by *delayed decision coding* (Chapter 9, for the case of Gaussian and gamma pdf's).

4.7 Statistical Properties of Quantization Errors

Previous sections have concentrated on computing the variance of the quantization error. We recall that, for a given input x, the output y is deterministically given as $y = x-q$, where q is the quantization error. However, if x is a realization of a random variable X, then q can be considered as a realization of a random variable Q. This section will discuss some of the statistical properties of Q; in particular, its pdf and its acf (and hence its psd). The discussions will show that the assumption of a uniform pdf for q is valid if $R \geqslant 2$, and that the assumption of a white quantization error is valid if $R \geqslant 2$, and if the adjacent sample correlation of the input, $\rho_{xx}(1)$, is not too close to + 1.

4.7.1 Probability Density Functions

Figure 4.4 has shown the nature of quantization error. In uniform quantization [Figure 4.4(a)], the input intervals are of constant length Δ. The error is then bounded to $\pm \Delta/2$ provided that the representation values are midway between decision thresholds. It seems reasonable in this case to assume that all amplitude samples within the range of $\pm \Delta/2$ occur with the same probability. This means that the pdf $p_q(q)$ is constant over the interval $-\Delta/2 \leqslant q \leqslant \Delta/2$, as noted in (4.16).

A more rigorous discussion can be used to justify the above model. Consider a (not necessarily uniform) quantizer with $L = 2^R$ levels. The errors produced in each of the decision intervals $\mathcal{I}_k$ contribute to the total quantization error, and the contributions depend on the probability $P_k = P\{X \in \mathcal{I}_k\}$ that we find the amplitude of the input signal in this interval. The total pdf $p_q(\cdot)$ is thus the sum of L conditional pdf's weighted by the corresponding probabilities, since the events $\{X \in \mathcal{I}_j\}$ and $\{X \in \mathcal{I}_k\}$ are mutually exclusive:

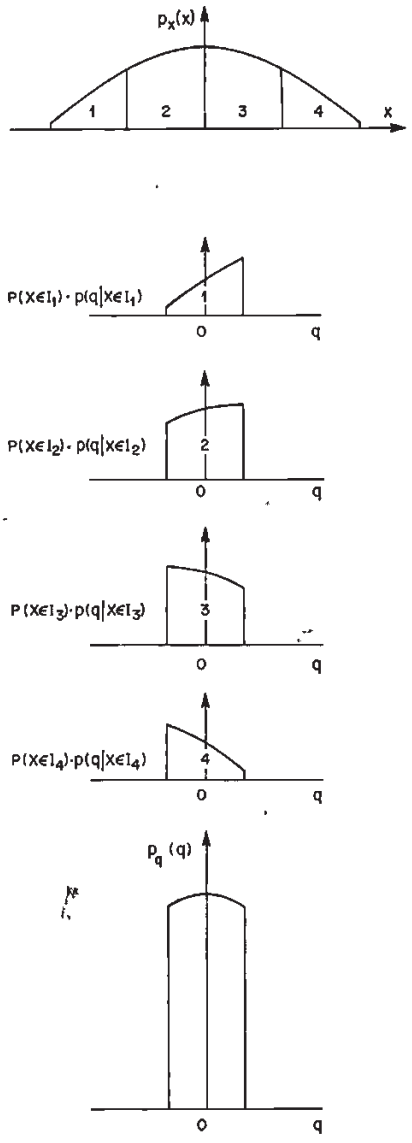
$$p_q(q) = \sum_{k=1}^{L} P_k \, p_q(q|X \in \mathcal{I}_k) \; ; \qquad p_q(q|X \in \mathcal{I}_k) = p_x(q+y_k|X \in \mathcal{I}_k) \quad (4.114)$$

However, from (2.29),

$$p_x(x|X \in \mathcal{I}_k) = \begin{cases} (1/P_k)p_x(x) \; ; & X \in \mathcal{I}_k \\ 0 & \text{otherwise} \end{cases} \quad (4.115)$$

Thus, $p_q(q)$ is just a superposition of sections of the input pdf $p_x(x)$. These sections are given by the decision intervals $\mathcal{I}_k$ and they are recentered by a shifting of $\mathcal{I}_k$ by a distance y_k. Figure 4.21 shows an example: the input has a bounded pdf and it is quantized by a 4-level uniform quantizer. Note that the resulting pdf

Figure 4.21 Justification of the uniform pdf model for quantization errors.



0146

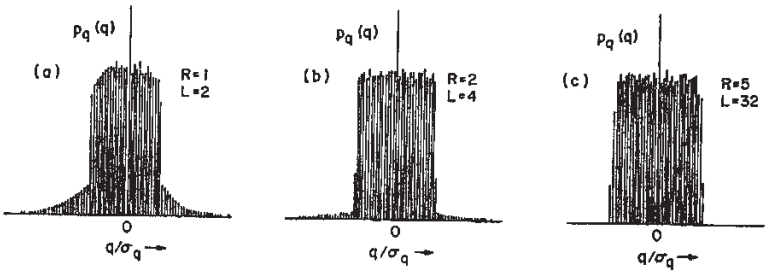


Figure 4.22 Quantization error pdf for Gaussian inputs and uniform quantization with (a) 2, (b) 4 and (c) 32 output levels [Koerber, 1979].

of the quantization error is almost flat-topped even if certain individual sections are not quite flat. In fact, an exactly flat $p_q(q)$ is exactly guaranteed for all symmetric input-pdf's if the pdf within a segment is a straight line, if the quantization is uniform, if the reconstruction levels are the midpoints of the corresponding decision intervals, and if the input is bounded in amplitude. In Figure 4.21, these conditions are not satisfied, and therefore the final pdf is not exactly flat. Even if the input random variable is unbounded, a flat-topped pdf $p_q(q)$ is a useful approximation if the number of output levels is large:

$$p_q(q) \approx (1/\Delta)\text{rect}(q/\Delta) \quad \text{for large } L \tag{4.116}$$

Figure 4.22 shows experimental error pdf's for uniform quantization of a Gaussian input with $L=2,4$ and 32. Note the rapid evolution of a uniform pdf as L increases. Also see Problem 4.16.

A precise condition can be given for a flat-topped error pdf if $L\to\infty$. This result is part of what is sometimes known as the *Quantization Theorem* [Widrow, 1956] [Sripad and Snyder, 1977]. This theorem states conditions under which the pdf of an input can be recovered from that of its quantized version, so that statistical measurements can be meaningfully based on quantized signals.

4.7.2 Power Spectral Densities

Let $Q(m)$ and $Q(n)$ be r.v.'s defining the quantization errors at times m and n. We are interested in the acf $R_{qq}(m-n) = E[Q(m)Q(n)]$ and its Fourier transform, the psd $S_{qq}(e^{j\omega})$. It is clear that these quantities can be determined if the joint pdf $p_{q(m)q(n)}[q(m), q(n)]$ is known. Instead of attempting analytical results, we note some intuitive arguments. Figure 4.23 shows a segment of a random waveform as well as the quantization error waveform. If the number of quantizer levels is sufficiently large, as it is in the figure, there are many changes in sign in the error waveform, and it seems reasonable that error samples taken from it at a sufficiently low rate (as given by the Nyquist frequency illustrated in the figure) are *statistically independent*. This will not be true (i) if the waveform does not change much in amplitude in a Nyquist interval [$\rho_x(1)\to+1$], (ii) if the waveform has periodicities that correlate distant error samples, or (iii) if the

0147

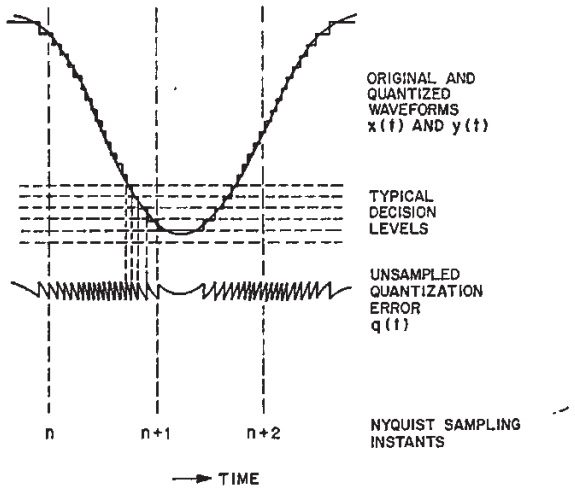


Figure 4.23 Unsampled signal and quantization error waveform [Cattermole, 1969].

quantizer has only a few levels so that error oscillations such as those of Figure 4.23 do not exist.

One of the consequences of statistical independence of successive error samples $q(m)$ and $q(n)$ is that their joint pdf is the product of individual pdf's, so that with the uniform univariate model (4.116),

$$p_{q(m)q(n)}[q(m), q(n)] = \begin{cases} 1/\Delta^2 & -\Delta/2 \leqslant q(m), q(n) \leqslant \Delta/2 \\ 0 & \textit{otherwise} \end{cases} \tag{4.117}$$

which represents a square probability box of constant height $1/\Delta^2$.

Another consequence of statistical independence is that the quantization noise is *white*:

$$R_{qq}(m-n) = \sigma_q^2 \ \delta_{mn} \tag{4.118}$$

We can also invoke a spectral interpretation to support (4.118). The unsampled error waveform has a spectrum that is much broader than the signal spectrum. Sampling of the quantized signal implies an undersampling of the error waveform (see Figure 4.23) and hence aliasing. (Quantizing of the samples gives equivalent results.) The total noise spectrum has many components that would be folded back into the baseband as in Figure 3.15(b). Consequently we have two results: (i) the noise spectrum flattens, and (ii) its baseband power equals the variance of the unsampled error waveform. We have already discussed (ii) in Chapter 3. The flattening of spectrum (i) is shown by the dashed spectrum in Figure 3.15(b). Graphically, it is not very different from the pdf-flattening process in Figure 4.21.

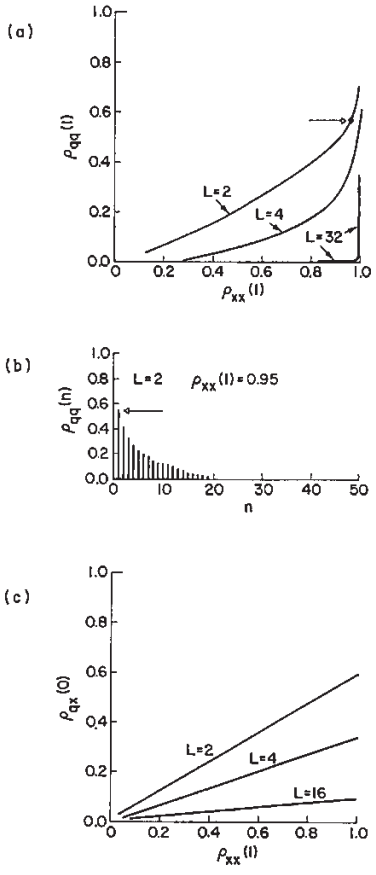
The nature of quantization noise spectrum can also be discussed by looking at the equivalent description by means of quantization noise acf, as in the following example.

Example 4.13. Measurements with an AR(1) Source

Consider a Gaussian AR(1) source with normalized adjacent sample correlation $\rho_{xx}(1)$, and quantizers with numbers of levels $L=2,4,\ldots,32$. Figure 4.24(a) demonstrates that there are significant correlations $\rho_{qq}(1)$ only if the quantization is coarse (say $L \leqslant 4$) and/or $\rho_{xx}(1) \rightarrow 1$. For $\rho_{xx}(1)$ values of practical significance ($\rho_{xx}(1) < 1$), $\rho_{qq}(1)$ values are insignificant for $L=32$.

Figure 4.24(b) shows the entire acf sequence $\rho_{qq}(n)$ for quantization error with $L=2$ and $\rho_{xx}(1)=0.95$. Notice that $\rho_{qq}(n)$ decays exponentially with n; $\rho_{qq}(1)$ is, of course, the value 0.56 on the $\rho_{xx}(1)=0.95$ point of Figure 4.24(a).

Figure 4.24 Quantization of a Gaussian AR(1) input: (a) $\rho_{qq}(1)$ versus $\rho_{xx}(1)$; (b) $\rho_{qq}(n)$ versus n; and (c) $\rho_{qx}(0)$ versus $\rho_{xx}(1)$ [Koerber, 1979].



0149

Figure 4.24(c) shows the crosscorrelation value $\rho_{qx}(0) = E[Q(n)X(n)]/\sigma_q\sigma_x$ as a function of $\rho_{xx}(1)$. In very coarse quantization ($L = 2$), a significant crosscorrelation exists between quantization error and input for $\rho_{xx}(1) \to 1$. Also see Problem 4.17(a). •

4.7.3 The Gain-Plus-Additive-Noise Model

We have noted that it is reasonable to assume that the quantization noise is white and that its pdf is flat. We shall now characterize this noise a little further. Since $Y=X-Q$, by squaring and taking expectations,

$$\sigma_y^2 = \sigma_x^2 + \sigma_q^2 - 2E[XQ] \tag{4.119}$$

A simple *additive input-independent noise* model would imply that

$$E[XQ] = 0; \quad \sigma_y^2 = \sigma_x^2 + \Delta^2/12 \tag{4.120}$$

In the above model, therefore, the variance of the quantizer output is greater than that of the input by the amount $\Delta^2/12$ [Figure 4.25(a)].

We have already seen, however, that with *the noise in pdf-optimized quantizers*, $\sigma_y^2<\sigma_x^2$. In fact, in that case we have shown (4.49) that

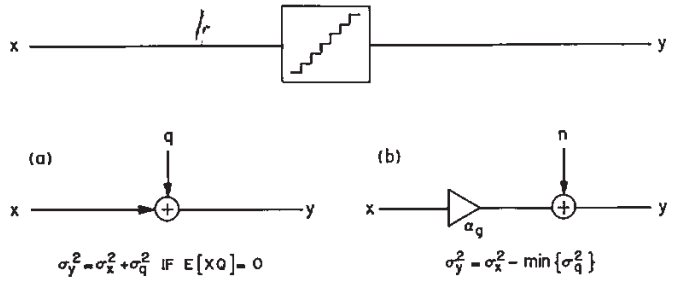
$$E[XQ] = \min\{\sigma_q^2\}; \quad \sigma_y^2 = \sigma_x^2 - \min\{\sigma_q^2\} \tag{4.121}$$

A better model for quantization is therefore not a purely additive independent noise model, but one with a less-than-unity *gain* component and an *additive* component. This is shown in Figure 4.25(b) where the gain is $\alpha_g < 1$, and the additive component is described by a r.v. N. If N is additive and input-independent, $E[XN] = 0$, $E[XQ] = E[X(X-\alpha_g X-N)] = \sigma_x^2(1-\alpha_g)$, and for (4.121) to hold

$$\alpha_g = 1-\min\{\sigma_q^2\}/\sigma_x^2 = 1 - \min\{\epsilon_q^2\} \tag{4.122}$$

Clearly, $\alpha_g \to 1$ as $L \to \infty$. This has been noted in the gain-plus-additive-noise model of rate distortion theory, Appendix D. Also see Problem 4.18 and Figure D.2.

Figure 4.25 Quantization noise models: (a) purely additive noise model; and (b) model with non-unity gain and additive noise.



It is clear from the above that a careful analysis of quantization noise must take into account the fact that the noise is not additive, and not uncorrelated with the input. In particular, at low bit rates, especially at $R = 1$ bit/sample, the errors in quantization must necessarily be considered as belonging to two classes: one caused by a change in level and one caused by uncorrelated noise. It is obvious why these types of distortions must often be considered separately. In speech communication, for example, a change only in level will have very little perceptual significance, unless the change exceeds about 3 dB [Richards, 1973]. In coarse quantization, α_g values can be small enough to cause an attenuation of 2.5 dB; this is seen by using $\min\{\epsilon_q^2\} = 0.25$ from Table 4.2 in (4.122), and computing 20 log α_g. But the main source of perceptual degradation is not this attenuation, but rather the additive noise component N.

4.8 The Use of Dithering in Coarse Quantization

Figure 4.24 has shown that when quantization is coarse (for example, $L=2$), $\rho_{qq}(n)$ and $\rho_{qx}(1)$ values are quite significant, suggesting a colored and input-correlated error sequence. Equation (4.121) has shown that $E[XQ]=\min\{\sigma_q^2\}$; this shows again that quantization error and input are correlated in pdf-optimized quantization, and that this correlation increases as L decreases, causing a higher $\min\{\sigma_q^2\}$. The above autocorrelation and crosscorrelation properties are reflected by perceptually undesirable signal-dependent patterns in the error sequence. By adding appropriate high-frequency signals to a waveform prior to quantization, it is possible to break up these undesirable patterns. The high-frequency perturbation signal can be a pseudorandom signal; it can also be a carefully designed deterministic signal; in either case, it is referred to as a *dither* signal, and the refinement of *dithering* is particularly useful with coarse quantizers [Goodall, 1951] [Jaffe, 1959] [Roberts, 1962] [Furman, 1963]. We shall begin with a study of dithered quantizers from a statistical viewpoint. At a later point in this section, we will discuss the dithering of speech and image signals [Thompson and Sparkes, 1967] [Limb and Mounts, 1969] [Wood and Turner, 1972] [Jayant and Rabiner, 1972] and relate the respective perceptual phenomena to the statistical viewpoint.

Figure 4.26 illustrates the principle of dithering with the example of a 9-sample input sequence and a uniform one-bit midrise quantizer. Figure 4.26(a) shows the conventional two-level output and the conventional reconstruction error $r(n) = x(n)-y(n)$. Note how the error pattern mimics the input in the left and right skirts as well as in the concave segment in the middle. Note also that the output $y(n)$ shows sustained artificial *contours* of amplitude ± 0.5. In Figure 4.26(b), the addition of a high-frequency dither signal $\vartheta(n)$ prior to quantization results in an error sequence that is more oscillatory and less input-like. For example, the extreme right skirt of the error waveform is now falling, unlike the rising input; and the input-like concavity in the error waveform of (a) is also broken up by zero crossings at samples 5 and 6. However, the error waveform in (b) clearly has a larger variance than that in (a). In (c), on the other hand, the additional final operation of dither subtraction at the receiver seems to restore the

0151

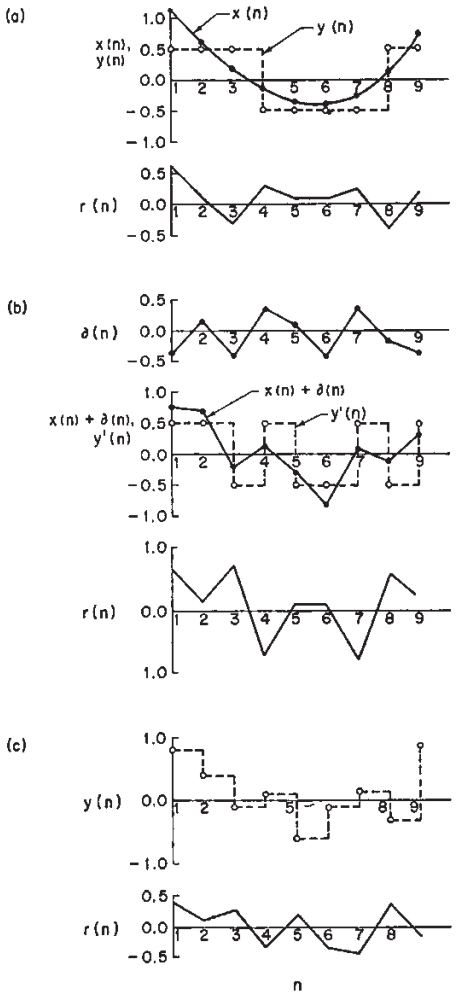


Figure 4.26 Input $x(n)$, output $y(n)$ and reconstruction error $r(n)$ in quantizing systems: (a) without dithering; (b) with dither $\partial(n)$, but without subsequent subtraction of dither; and (c) with dithering, followed by subtraction of dither at the receiver.

variance of the reconstruction error waveform without losing its oscillatory character. The reconstructed output $y(n) = y'(n) - \partial(n)$ (see Figure 4.27) tracks the input very closely, unlike the waveform $y'(n)$; *further, the number of possible levels in* $y(n)$ *is no longer limited to two, the number of actual quantizer levels, as in the output sequences in (a) and (b).* The implementation in (c) requires knowledge of the dither signal at the receiver with full sample-to-sample synchrony with the $\partial(n)$ used at the transmitter. The simpler system of (b) is therefore used

0152

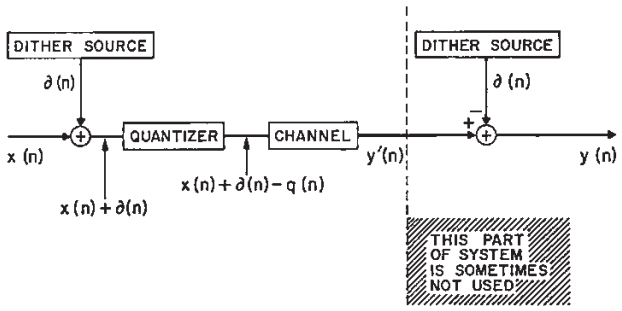


Figure 4.27 Block diagram of a dithered quantization system. The section to the right of the dashed line represents an operation that is sometimes uncritical; for example, in image quantization.

quite often. The increase in error variance in (b) is more than offset by the perceptual gain in terms of a non-input-like error pattern.

Figure 4.27 is a block diagram of a dithered quantizer system. As stated, the subtraction of the dither signal $\partial(n)$ at the receiver is sometimes uncritical and not done, especially in picture quantizers. In such systems, the final output is $y'(n)$, not $y(n)$. Usual operations of codeword assignment and D/A conversion are implied in Figure 4.27.

4.8.1 A Statistical Interpretation

Consider the reconstruction error variance

$$\sigma_r^2 = E[R^2] = \int_x \int_y p_{xy}\,(x,y)\;(x-y)^2\,dxdy \tag{4.123}$$

By setting $x-y = x-Q(x)$, and noting that the integral of the joint pdf over y is $p_x(x)$, (4.123) reduces to the earlier (4.11). For simplicity, $x(n)$, $y(n)$, and later the dither signal $\partial(n)$ will be denoted by x,y and ∂; and subscript indices will be dropped from all pdf's. From earlier discussions, we know that $E[R^2] = \Delta^2/12$ for uniform quantizers (4.17), if there are no overload effects.

Let $E[Y|x]$ denote the conditional mean of the output for a given input x:

$$E[Y|x] = \int_y yp(y|x)\,dy \tag{4.124}$$

It will be interesting to split $E[R^2]$ into three components, each of which involves the conditional mean $E[Y|x]$:

$$\begin{aligned} E[R^2] &= \int_x \int_y p(x,y)\;(x-E[Y|x]+E[Y|x]-y)^2\,dxdy \\ &= \int_x \int_y p(x,y)\;(x-E[Y|x])^2\,dxdy + \int_x \int_y p(x,y)\;(E[Y|x]-y)^2\,dxdy \\ &\quad + \int_x \int_y p(x,y)\;2(x-E[Y|x])(E[Y|x]-y)\,dxdy \\ &= I_1 + I_2 + I_3 \end{aligned} \tag{4.125}$$

The third integral can be rewritten as the sum of two components:

$$2\int_x\int_y p(x,y)\,(x\,E[Y|x]-xy)\,dxdy+2\int_x\int_y p(x,y)\,(y\,E[Y|x]-(E[Y|x])^2)\,dxdy$$

It can be seen that both these components vanish by definition, so that the third integral in (4.125) equals zero. The fact that $E[Y|x]$ is a function only of x enables us to express the first integral as a single integral

$$I_1 = \int_x p(x)\;(x-E[Y|x])^2\,dx \tag{4.126a}$$

The second integral is

$$I_2 = \int_x\int_y p(x)\;p(y|x)\;(y-E[Y|x])^2\,dx\,dy \tag{4.126b}$$

The quantity I_2 has been termed *apparent noise*: it represents the variation in y relative to the conditional mean output value; and the component I_1 has been called the *tonal quality*: it shows the deviation of the conditional mean output value from the true input [Roberts, 1962]. In coarse quantization of an image, for example, a slowly varying or constant gray tone would be displaced, if $I_1 \neq 0$, into a different average shade or tone of gray in the output. Thus, the quantities I_1 and I_2 are descriptors of orthogonal effects, both statistically and perceptually, and since $I_3 = 0$,

$$\sigma_r^2 = E[R^2] = I_1 + I_2 \tag{4.127}$$

In refining coarse quantizers for speech and image inputs, a central notion is that the degradation represented by I_1 is perceptually more annoying than that given by I_2. A good quantizer should seek to decrease I_1, at the cost of I_2, if necessary, while keeping the total error σ_r^2 as low as possible. Before proceeding to show how I_1 can be decreased by dithering, we should mention that the perceptual importance of I_1 is augmented by statistical interpretations. First, $I_1 = 0$ in (4.126b) implies that $E[Y|x] = x$ for all x because the integrand is a product of two nonnegative terms. Thus if $I_1 = 0$, *the expected value of reconstruction equals that of input, for all values thereof.* Note also that $E[Y|x] = x$ implies that $E[R|x] = 0$, because $Y = X - R$. Further, if $\partial(n)$ is a zero-mean, input-independent signal, $E[Y|x] = E[Y'|x]$ and $E[R|x] = E[R'|x]$, where $R' = R + \partial$; so that $I_1 = 0$ implies that $E[Y'|x] = x$ and $E[R'|x] = 0$. Thus, for any given value of the input, the conditional mean values of the reconstruction errors R and R' are both zero. Finally, a value of $I_1 = 0$ *is a necessary and sufficient condition for the reconstruction errors R and R' to be statistically independent of the input X* [Schuchman, 1964]. Therefore the object of dithering will be to render $I_1 = 0$ and equivalently, to make R and R' independent of X.

The fact that $E[R|x] = 0$ with dithering, *for any input x*, is very significant. Recall from Section 4.4 that in a pdf-optimized quantizer without dithering $E[R|\mathcal{I}_k] = 0$, for *any input interval $\mathcal{I}_k$*.

The Effect of Dithering on I_1, I_2 and σ_r^2. To get some perspective about the quantities I_1 and I_2, we shall use the example of a uniform pdf input. As noted in Chapter 2, this is a good model for image inputs.

Example 4.14. Uniform quantization of a bounded input

Consider a midrise R-bit quantizer with step size Δ. Assume for simplicity that the maximum input magnitude is

$$\max\{|x|\} = x_{\max} = (2^{R-1}-1/2)\Delta = x_{ol} - \Delta/2\,, \tag{4.128}$$

a value smaller than that of the bounded input in (4.28) where $x_{\max}$ was equal to x_{ol}. The present special assumption about $|x|_{\max}$ is not critical for realizing the benefits of dithering. It is merely used for analytical simplicity in this example. Figure 4.28 represents the quantization of a random input of the form

$$x = (k+1/2)\,\Delta + s\,; \quad 0 < s < \Delta/2 \tag{4.129}$$

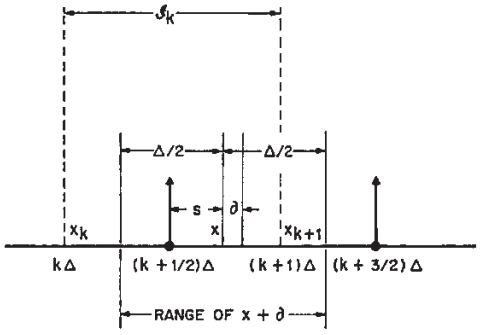
where k is an integer in the range $(-2^{R-1},2^{R-1})$ and the requirement of positive s is for convenience. The ensuing discussion will extend to negative s also, in a very straightforward fashion. The arrows in Figure 4.28 represent adjacent reconstruction values in the quantizer characteristic, and the interval $\mathscr{I}_k$ from earlier notation includes the reconstruction value $(k+1/2)\,\Delta$. Three situations are of interest: (i) quantization without dither (ii) quantization with random dither, and no subsequent subtraction of it at the receiver, and (iii) quantization with pseudo-random dither which is eventually subtracted from the quantizer output, as in Figure 4.27.

i. **Quantization without dither**

From earlier notation, $r(n) = q(n) = s$ showing how the error is strongly dependent on the input. Therefore $\sigma_r^2 = \Delta^2/12$ and $I_2 = 0$ because, for any given value of input $x(n)$, there is no variation in the output $y(n)$, the latter being uniquely specified by the transfer function of the quantizer. From (4.126) and (4.127), therefore,

$$I_2 = 0\,; \quad I_1 = \sigma_r^2 = \Delta^2/12 \tag{4.130}$$

Figure 4.28 Notation used for dither analysis in Example 4.1. Arrows indicate adjacent reconstruction values in the quantizer output.



0155

ii. **Quantization with unsubtracted random dither**

As in Figure 4.28, let the random zero-mean dither ϑ have a range $(-\Delta/2,\Delta/2)$ and a pdf

$$p_\vartheta(\vartheta) = 1/\Delta \text{ rect } (\vartheta/\Delta) \tag{4.131}$$

Two sub-cases can be distinguished, keeping in mind that $s > 0$:

(a) $-\Delta/2 < \vartheta < \Delta/2 - s$ or $x+\vartheta \in \mathscr{I}_k$: In this case the quantizer input is $(x+\vartheta) < (k+1)\Delta$, the output is $(k+1/2)\Delta$ and the reconstruction error is s.

(b) $\Delta/2 - s < \vartheta < \Delta/2$ or $x+\vartheta \in \mathscr{I}_{k+1}$: This is the range of ϑ for which the quantizer output is $(k+3/2)\Delta$, and the reconstruction error is $s - \Delta$. In this case, the effect of dithering is to shift quantizer input from interval $\mathscr{I}_k$ to I_{k+1}, although coder input X is in $\mathscr{I}_k$.

Expressions for $E[Y'|x]$ and reconstruction error variance σ_r^2 result from averaging the results in (a) and (b), first over ϑ and then over the variable s. The probabilities of the ϑ-ranges in (a) and (b) are $(1-s/\Delta)$ and (s/Δ), because of the uniform distribution (4.131). Consequently,

$$E[Y'|s] = (1-s/\Delta)\ (k+1/2)\ \Delta + (s/\Delta)\ (k+3/2)\Delta\ ;$$

$$E[R'^2|s] = (1-s/\Delta)\ (s)^2 + (s/\Delta)\ (s-\Delta)^2$$

With a uniform model for input pdf,

$$p_s(s) = 2/\Delta\ ; \quad 0<s<\Delta/2 \tag{4.132}$$

We now average $E[Y'|s]$ and $E[R'^2|s]$ over r, using (4.132), to obtain

$$E[Y'|s>0] = (k+1/2)\ \Delta + \Delta/4\ ; \quad E[R'^2|s>0] = \Delta^2/6$$

Similarly, it can be shown that for negative s,

$$E[Y'|s<0] = (k+1/2)\ \Delta - \Delta/4\ ; \quad E[R'^2|s<0] = \Delta^2/6$$

so that expected values over all s $(-\Delta/2 \leqslant s \leqslant \Delta/2)$ are

$$E[Y'|x] = (k+1/2)\Delta x\ ; \quad \sigma_r^2 = \Delta^2/6$$

Using (4.126) and (4.127),

$$I_1 = 0\ ; \quad I_2 = \sigma_r^2 = \Delta^2/6 \tag{4.133}$$

Note from (4.130) and (4.133) that unsubtracted dithering decreases I_1 and increases I_2; it also doubles the total error variance. We now examine a means of keeping a zero I_1, while reducing the error variance to the original minimum value of $\Delta^2/12$.

iii. **Quantization with subsequently subtracted pseudo-random dither**

If a known dither signal is eventually subtracted at the receiver from the quantizer output, the analysis of (ii) above will have to include the effect of this subtraction in evaluating $E(Y|s)$ and $E[R^2|s]$. With dither subtraction, the final output decreases

0156

by ϑ, and the reconstruction error consequently increases by the same amount ϑ. Consequently, reconstruction errors for subcases (a) and (b) [defined in (ii)] will be $s + \vartheta$ (instead of s) and $\Delta - s + \vartheta$ (instead of $\Delta - s$), respectively. Note that for the extreme example of $s = \Delta/4$ and $\vartheta = \Delta/2$ [values just satisfying the requirement of subcase (b) in (ii)], the error with dither subtraction is zero; while without dither subtraction, the final error would be $-\Delta/2$.

The properties of a quantizer with dither subtraction can be analyzed as in case (ii), and the results are

$$I_1 = 0\,; \quad I_2 = \sigma_r^2 = \Delta^2/12 \tag{4.134}$$

Table 4.10 Effect of dithering on σ_r^2, I_1 and I_2 for uniform quantization with a non-overloading input $x_{max} = x_{ol} - \Delta/2$.

Quantizing scheme	σ_r^2	I_1	I_2
No dither	$\Delta^2/12$	$\Delta^2/12$	0
Unsubtracted random dither	$\Delta^2/6$	0	$\Delta^2/6$
Subtracted pseudo-random dither	$\Delta^2/12$	0	$\Delta^2/12$

The benefits of dithering, as summarized in Table 4.10, reflect the breaking up of signal-dependent patterns in the granular error region, as already illustrated in Figure 4.26. The most important effect of non-zero overload is to render the zero I_1 values in the table non-zero as well; resulting I_1 values will in fact be equal to $\Delta^2/12$ weighted by the probability of quantizer overload.

Independence of Error and Input with Uniform Dither. We can also show that in cases (ii) and (iii) above, the pdf of reconstruction noise is independent of the value of input. For example, for case (iii), we express the probability distribution function as a weighted sum of two terms, corresponding to subcases (a) and (b):

$$\begin{aligned} P\{R < t\,|s\} = {} & P\{-\Delta/2 \leqslant \vartheta \leqslant \Delta/2 - s\}\, P\{s + \vartheta < t\} \\ & + P\{\Delta/2 - s \leqslant \vartheta \leqslant \Delta/2\}\, P\{s - \Delta + \vartheta < t\} \end{aligned}$$

Using the uniform dither pdf (4.131), and considering a specific fixed value of s,

$$P\{R < t\,|s\} = \frac{\Delta - s}{\Delta} \cdot \frac{t - s + \Delta/2}{\Delta} + \frac{s}{\Delta} \cdot \frac{t - s + \Delta + \Delta/2}{\Delta} = \frac{1}{\Delta}(t + \Delta/2)$$

which depends on t and Δ, but is *independent* of s. In fact, the above distribution is simply that of a uniform pdf with support $(-\Delta/2, \Delta/2)$; the reconstruction error follows this pdf *irrespective* of the value of s, and hence irrespective of the value of x. •

4.8.2 Results for Speech and Images

Speech Quantization. Figure 4.29 shows the effect of dithering on the normalized error-input crosscorrelation $\rho_{rx}(0) = E[R(n)X(n)]/\sigma_r\sigma_x$ in speech quantization.

0157

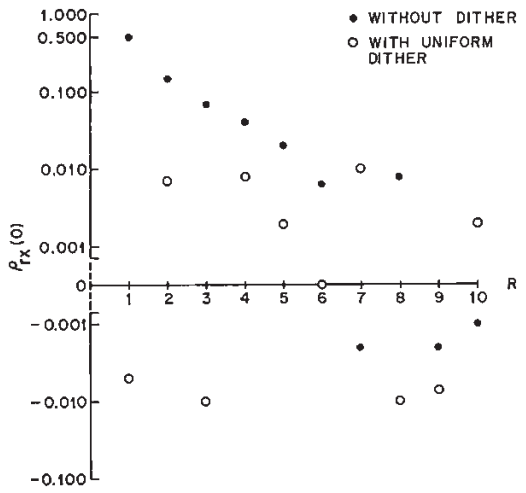


Figure 4.29 Measured signal-to-error correlation $\rho_{rx}(0)$ versus R in uniform quantization of speech. In the dither example, the dither signal was eventually subtracted at the receiver [Jayant and Rabiner;©1972, AT&T].

Note the nonlinearity of the y-axis scale. The effect of dithering is to make the crosscorrelation very small for all values of R, the number of quantizer bits. In the absence of dithering, the magnitude of the crosscorrelation is a monotonically decreasing function of R in the range $R = 1$ to 6; and a significant correlation exists in coarse quantization (say $R \leqslant 2$). The correlation value of 0.5 for $R=1$ is very close to that for the AR(1) input with $R=1$ and the speech-typical $\rho_{xx}(1)=0.85$ [Figure 4.24(c)].

Figure 4.30 shows waveforms of speech input $x(n)$ and reconstruction error $q(n) = r(n) = x(n) - y(n)$ for the three values of R in Figure 4.29, and for the cases of dither (b) and no dither (c). The error waveforms are magnified (by a factor 2^R) to highlight their features. Note that (i) the signal-dependence in $r(n) = q(n)$ is a monotonically decreasing function of R in the absence of dithering; and (ii) dither serves to decorrelate $r(n)$ and $x(n)$ even for the worst case $R=1$.

Perceptual studies of speech dithering with fixed uniform quantizers [Rabiner and Johnson, 1972] serve to supplement the analytical and graphical results noted above. The tests involve a set of ten sentences and several subjects, and the following are the main observations:

(i) strong preferences for dithered quantization for bit rates $R=2$ to 6;

(ii) for $R=2,3$ and 4, a preference for dithered quantization even when it had one less bit per sample than the undithered quantizer;

(iii) an increase in word intelligibility with dithering for $R=4,5$ and 6; and

(iv) a decrease in word intelligibility with dithering for $R = 2$ and 3.

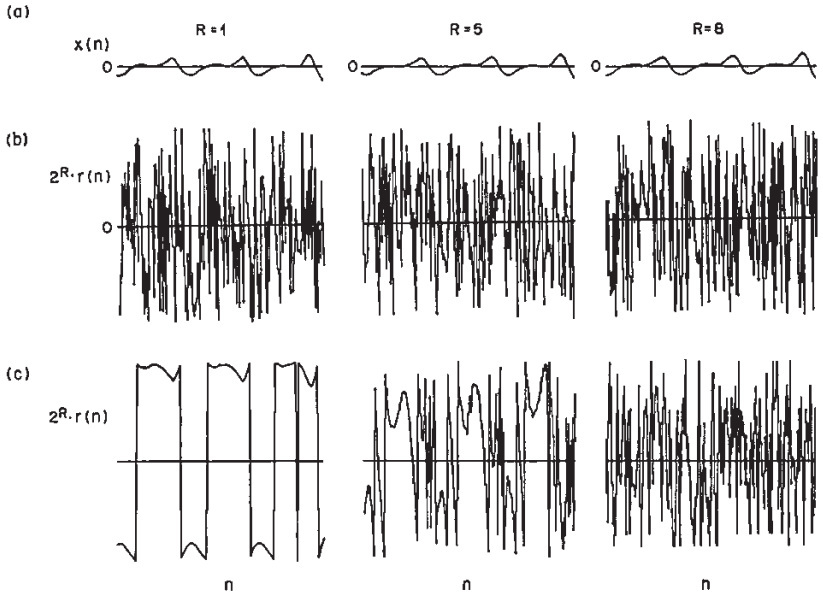
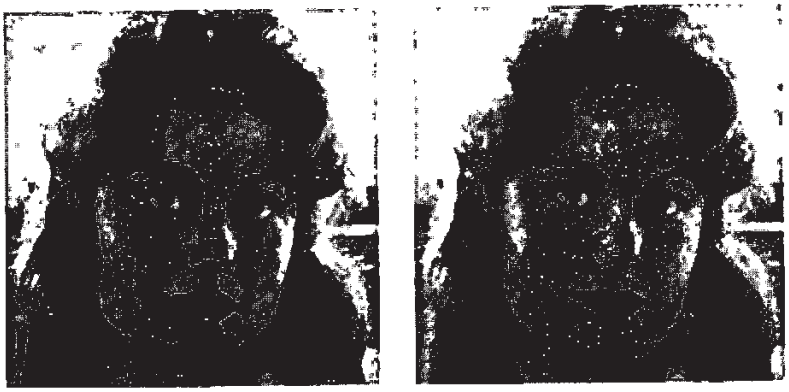


Figure 4.30 Waveforms of (a) speech input $x(n)$; and reconstruction error $r(n)$ in uniform quantization using R bits/sample (b) with dither and (c) without dither. In (b), dithering is followed by eventual subtraction of it at the receiver. The error waveforms have been magnified by a factor 2^R [Jayant and Rabiner;©1972, AT&T].

The difference between *preference* results and *intelligibility* results at $R=2$ and 3 is interesting, although these bit rates are academic for high-quality speech coding (Chapter 5) unless sophisticated redundancy reducing techniques are employed (Chapters 6 to 12). The loss of word intelligibility in (iv) has been attributed to the masking of low-energy, high frequency consonant information by reconstruction noise [Miller and Nicely, 1950]. Dithering tends to whiten the error spectrum and to produce more high-frequency error components than are present in the signal-dependent error signal of an undithered quantizer; and this results in increased consonant masking of low-level consonants by the errors. The masking and intelligibility loss noted above can be reduced in conditional dithering systems [Wood and Turner, 1972] [Chen and Turner, 1973] where the dithering process is halted whenever input speech amplitudes are in the innermost quantizer levels, indicating a high zero-crossing probability that is characteristic of consonant inputs.

Image Quantization. Video examples from a pioneering work on dithering [Roberts, 1962] appear in Figure 4.31. Note that with $R=3$, input-error dependencies in conventional quantization manifest themselves in the form of *contouring*, which is the result of abrupt changes of output level in a slowly changing gray area of the input scene. Such contouring is also indicated from the conventional (undithered) reconstructions of a slowly varying input in Figure



(a) (b)

Figure 4.31 Examples of 3-bit image quantization: (a) without dithering and (b) with dithering. In (b), the dither signal was eventually subtracted at the receiver [Roberts;©1962, IRE].

4.26(a). The use of uniform dither breaks up these contours and provides an image smoothness similar to that in undithered quantization with $R=6$. An important factor in image dithering is the lowpass filtering action of the eye. This gives a smooth appearance to the oscillatory outputs of dithered system reconstructions, even those with unsubtracted dither and consequently higher-than-conventional error variance $\Delta^2/6$ [as in Figure 4.26(b)]. The absence of such lowpass filtering in speech perception makes dither subtraction more critical in speech work; this is one reason why implementations of dithering in video work have been more extensive than in speech coding [Limb, 1969] [Lippel and Kurland, 1971] [Jarvis, Judice and Ninke, 1976].

4.8.3 Useful Forms of the Dither Signal

In the quantization of bounded inputs, where overload effects do not exist, there is a class of dither signals [Schuchman, 1964] that renders the reconstruction error $R(n)$ statistically independent of the input $X(n)$; the most useful member of this class is one whose pdf is uniform over the support $(-\Delta/2\ ,\ \Delta/2)$. This was already demonstrated in Example 4.13. Also, as stated earlier, a necessary and sufficient condition for Q to be independent of X is that I_1 be zero [Schuchman, 1964].

Amplitude-Quantized and Deterministic Dither Signals. Practical systems need to employ amplitude-quantized dither signals. As long as the resolution in the dither signal is fine compared with that of the quantizer input, one expects that the error-decorrelating properties of dither should be unaffected. Indeed, experiments with speech signals indicate that an adequate design is to make the step size in the quantized dither signal no greater than *one-fourth* the step size of the speech quantizer [Jayant and Rabiner, 1972]. In fact, this is precisely the design that was used in the illustration of Figure 4.26 where the quantizer step size was $\Delta = 1$, and

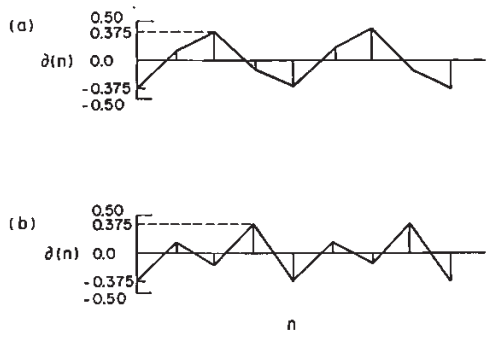


Figure 4.32 Periodic four-level dither sequences [Limb, 1969].

$\partial(n)$ randomly assumed one of four levels ($-3/8$, $-1/8$, $1/8$ and $3/8$), with an adjacent level spacing that was one-fourth of the quantizer step size. It is reasonable to insist, of course, that the dither signal be a symmetrical zero-mean quantity, with a uniform distribution of (discrete) amplitude in the range $(-\Delta/2, \Delta/2)$.

Dither signals can go a step further in terms of simplicity in that they can be *deterministic*. The motivation for a deterministic dither signal, besides simplicity, is to force negative correlations in the quantizer input $x(n) + \partial(n)$. When $x(n)$ is constant, as in the slowly varying areas of a picture, this is simply achieved by adding to it a dither sequence that rapidly oscillates about a zero mean. The ultimate perceptual result is an error waveform that is not only independent of the input, but also spectrally shaped in a perceptually desirable fashion — meaning specifically an error spectrum that is rich in the relatively less visible high-frequency components. Such a spectrum is the result of negative correlations induced in the error sequence. We recall from the simple AR(1) model of Figure 2.16(b) that a negative value for adjacent sample correlation results in a highpass psd. Error shaping can be illustrated by considering the amplitude-quantized deterministic dither sequences of Figure 4.32:

$$\partial(n):\{-3/8,\ 1/8,\ 3/8,\ -1/8...\} \quad \text{and} \quad \partial(n):\{-3/8,\ 1/8,\ -1/8,\ 3/8...\} \qquad (4.135)$$

When used with a constant gray level input $x(n)=0.5$ and a two-level quantizer ($y=1$ if $x>0.5$; $y=0$ otherwise), the respective values of error acf $\rho_{rr}(1)$ are 0 and -1, respectively (Problem 4.19). The dither signal in Figure 4.32(b) [the right-hand half of (4.135)] is the suggested design if one seeks a highpass error spectrum in constant-brightness areas of a picture. A visibility analysis [Limb, 1969] shows that this sequence has perceptual properties that are in fact better than those of random dither.

Dither Matrices. A natural extension of the 1-D sequence (4.136) is to a 2-D deterministic *dither matrix* with an oscillatory, or negatively-correlated $\partial(n)$ pattern in both dimensions. There is a systematic procedure for generating such

dither matrices. One starts with a well designed 2 × 2 matrix such as [Limb, 1969]

$$^2\mathbf{D} = \begin{bmatrix} D_{00} & D_{01} \\ D_{10} & D_{11} \end{bmatrix} = \begin{bmatrix} 0 & 2 \\ 3 & 1 \end{bmatrix} \tag{4.136}$$

and for larger matrices, one uses a recursion relationship of the form [Jarvis, Judice and Ninke, 1976]

$$^n\mathbf{D} = \begin{bmatrix} 4 \cdot {}^{n/2}\mathbf{D} + D_{00}\boldsymbol{\lambda} & 4 \cdot {}^{n/2}\mathbf{D} + D_{01}\boldsymbol{\lambda} \\ 4 \cdot {}^{n/2}\mathbf{D} + D_{10}\boldsymbol{\lambda} & 4 \cdot {}^{n/2}\mathbf{D} + D_{11}\boldsymbol{\lambda} \end{bmatrix}; \quad \begin{matrix} \boldsymbol{\lambda} \text{ is an all} - \textit{one} \\ \text{matrix of order } n/2 \end{matrix} \tag{4.137}$$

For example, the first recursion produces (Problem 4.21) a 4 × 4 matrix

$$^4\mathbf{D} = \begin{bmatrix} 0 & 8 & 2 & 10 \\ 12 & 4 & 14 & 6 \\ 3 & 11 & 1 & 9 \\ 15 & 7 & 13 & 5 \end{bmatrix} \tag{4.138}$$

Dither matrices have a perceptually optimal size $n \times n$ in image coding. Very small n represents inadequate randomness in dithering and insufficient quantization of dither amplitudes; while with very large n, the repetition of dither pattern produces signal-masking effects at a very low spatial frequency (given by f_s/n; f_s being the sampling frequency). A good design is an (8 × 8) dither matrix [Bayer, 1973].

4.9 Transmission Error Effects

In this section, we shall consider the effect of a noisy channel on the uniform and nonuniform quantization systems of Sections 4.3, 4.4 and 4.5. We shall assume that random errors are introduced in the bits that convey information about quantizer output level. As in Figure 1.9, we assume a *binary symmetric channel*, which implies that information about quantizer output level will be transmitted as a sequence of $R = \log_2 L$ bits; and we denote the random *bit error probability*, or *bit error rate*, by p_e. The above channel is assumed to be a *stationary memoryless channel*. *Stationarity* implies that channel properties do not change with time. A *memoryless* channel is one where the mapping between input and output at any given time is independent of previous outcomes.

In what follows we assume statistically independent inversions of message bits, signifying a totally random error pattern. This is a good model not only for channels without memory (where the fates of successive message bits are indeed independent) but also in systems where channel memory is rendered ineffective by bit-rearranging procedures such as interleaving or scrambling—procedures that tend to randomize channel error patterns that may be bursty or clustered [Jayant, 1975].

0162

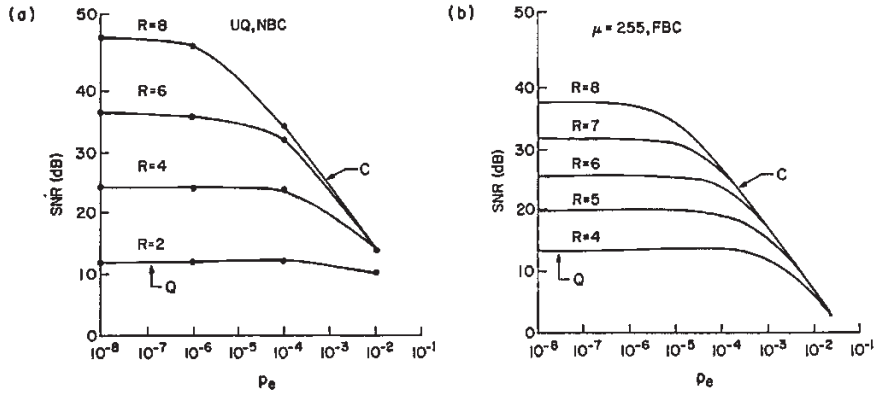


Figure 4.33 Illustration of quantizer-limited performance and channel-limited performance in digital transmission. The two limits are indicated by letters Q and C. (a) Uniform quantization of a uniform pdf input, followed by NBC coding of the output. (b) Quantization of a Laplacian pdf input with log companding ($\mu = 255$) and FBC coding of the output [Rydbeck and Sundberg, 1975]. The NBC and FBC coding systems are defined in Figure 4.35.

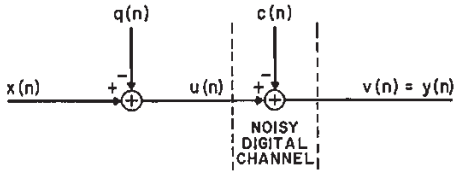
Further, most of our analyses will assume a bit error probability p_e that is small enough to justify consideration of only *single-error* patterns in received codewords.

Before proceeding with channel error analyses, let us put the effects of quantization errors and channel errors in perspective by means of the sketches of Figure 4.33. The results in (a) refer to the uniform quantization of a uniform pdf input. The performance of this system with channel errors will presently be derived (4.174). The results in (b) refer to the $\mu 255$ log-quantization of a Laplacian input [Rydbeck and Sundberg, 1975]. Figures 4.33(a) and (b) both suggest that at high bit rates, the quantizer system is more sensitive to very high values of p_e, and we shall see later that this is typical. The figures also suggest that when we are operating in the *channel-limited region* rather than in the *quantizer-limited region* of p_e, it pays to design the system with a coarser quantizer; and perhaps allocate some of the transmission capacity for explicit bit protection.

4.9.1 Error Definitions and Channel Codes

Referring to Figure 4.34, we find that in the presence of transmission errors, reconstructed output signal $y(n)$ will include the effects of *both quantization error*

Figure 4.34 Transmission of quantized amplitudes over noisy channels. In a noiseless channel, $v(n) = u(n)$.



q(n) and channel error c(n), resulting in a total reconstruction error $r(n)$ that is the sum of $q(n)$ and $c(n)$:

$$q(n) = x(n) - u(n) \tag{4.139a}$$

$$c(n) = u(n) - v(n) \tag{4.139b}$$

$$y(n) = v(n) = u(n) - c(n) = x(n) - q(n) - c(n)$$

$$r(n) = x(n) - y(n) = q(n) + c(n) \tag{4.140}$$

We will see presently that there are important classes of systems in which $\sigma_r^2 = \sigma_q^2 + \sigma_c^2$, or at least approximately so.

For example, refer to the midrise quantizer in Figure 4.35(a) with $L = 8$ levels and $R=3$ bits per codeword. This quantizer maps each input sample $x(n)$ into one of a set of eight rational numbers $u(n) \in \{y_k\}$; $k=1,2,...,8$. The representation level y_k is chosen if $x_{k+1} \geqslant x(n) > x_k$. The index k of input symbol y_k of the transmission system is transmitted to the receiver in a binary format, as the *channel codeword*; the received *channel codeword* is interpreted as one of the eight output symbols $v(n) \in \{z_k\}$; $k=1,2,...,8$, with $z_k = y_k$ [Figure 4.35(b)]. A change in amplitude $\Delta_{kj} = |y_k - z_j|$ results if the transmitted quantizer index k is changed to j because of channel errors. If this happens for input sample n, the corresponding channel error is $c(n) = \Delta_{kj}$.

Several *binary codes* are shown in Figure 4.35(a): the *natural binary code*, NBC; the *folded binary* code, FBC (also known as a *sign-magnitude* code, SMC); the *two's complement* code (TCC), where the codeword for amplitude y complements, bit by bit, the codeword for amplitude $-y$; and the *Gray* or *reflected binary* code (RBC), where every pair of adjacent codewords has exactly one bit position where the codewords differ.

The codewords in NBC are merely the so-called *binary representations* of decimal numbers 0 to 2^R-1:

$$\text{codeword: } [b_1\ b_2\ b_3\ ...\ b_R]\ ; \qquad \text{decimal equivalent: } \sum_{r=1}^{R} 2^{R-r}\, b_r \tag{4.141}$$

For example, if $R=3$, the 3-bit word 110 represents a quantizer output of $4{\cdot}1 + 2{\cdot}1 + 1{\cdot}0 = 6$ in decimal notation, which is y_7 in Figure 4.35. The bit with the highest weighting in (4.141) is b_1, the *most significant bit* (MSB). The *least significant bit* (LSB) is b_R. The most significant bit b_1 in Figure 4.35(a) is a *sign bit*. The remaining bits constitute an $(R-1)$-bit NBC code used to convey magnitude information. Bit b_2 is the *most significant magnitude bit*.

If the channel input is y_2, an error in the most significant bit causes the reception of z_6 with NBC coding (001→101), and of z_7 with FBC (010→110). In general, an error in the most significant bit causes a displacement equal to $2^{R-1}\Delta$ (half the quantizer range) in NBC and TCC, and a displacement that is

0164

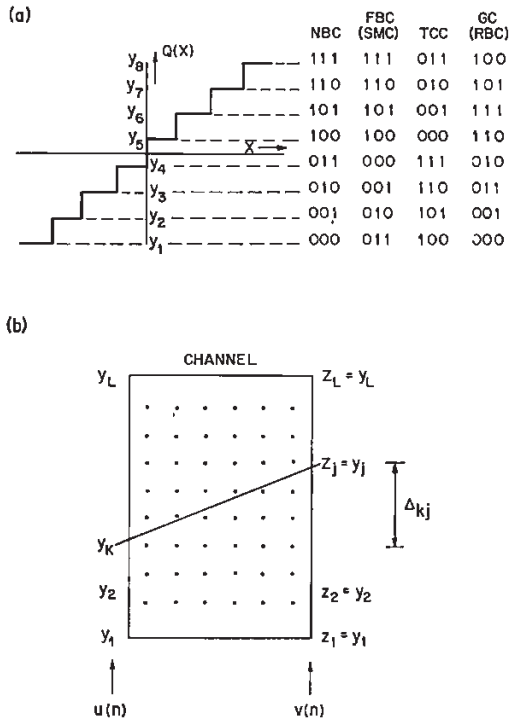


Figure 4.35 (a) Symmetric midrise quantizer with eight output levels, and four binary codes. (b) Effect of transmission errors on y_k.

proportional to input amplitudes in FBC and RBC. If signal amplitudes near zero are more probable than those away from zero (as in speech), the FBC code would clearly lead to a smaller channel error variance than the NBC code (see Problem 4.31).

It is possible to optimize a channel code to minimize channel error variance. Such a code would be constructed so that errors in the most probable codewords make only a small contribution to error variance. This, in turn, is achieved if codewords resulting from an error are close to the original codeword — in other words, this would lead to a *minimum distance code*, MDC. At least for one example of an input pdf with a high probability of small amplitudes, viz., the Laplacian pdf, the FBC is identical to the MDC for $R=2$ and 3 [Sundberg and Rydbeck, 1976].

With an R-bit code, the *probability of no error* in the received codeword is equal to the probability of R correctly received bits:

$$P_0 = (1-p_e)^R \simeq \exp(-p_e R) \quad \text{for small } p_e \tag{4.142}$$

The probability of *one or more errors* in the received codeword is

$$P_{>0} = 1-P_0 \simeq 1-\exp(-Rp_e) \simeq Rp_e \quad \text{for small } p_e \tag{4.143a}$$

The probability of *exactly one error* in the received codeword is the probability of $(R-1)$ correct bits and one incorrect bit (with R possible positions in the codeword):

$$P_1 = \binom{R}{1} p_e(1-p_e)^{R-1} = Rp_e(1-p_e)^{R-1} \tag{4.143b}$$

$$\simeq Rp_e \simeq P_{>0} \quad \text{for small } p_e \tag{4.143c}$$

Thus, for $p_e \to 0$, most error patterns imply single errors per codeword.

A useful tool in channel error analyses is the *Hamming distance* D_{kj}, the number of codeword letters that are different in the representation of intended output level y_k and actual output level y_j. It can be seen (Problem 4.22) that with a binary symmetric channel, and an R-bit code, the conditional probability P_{kj}, the probability that y_j will be received when y_k was sent, is given by

$$P_{kj} = P\{V(n)=y_j | U(n)=y_k\} = p_e^{D_H}(1-p_e)^{R-D_H} \tag{4.144}$$

Note that if $p_e < 0.5$, P_{kj} decreases monotonically with D_{kj}.

The Hamming distance D_{kj} between y_k and y_j depends on the binary coding scheme. For example, from Figure 4.35, D_{26} is 1 for NBC, and 3 for FBC. Based on (4.144), the corresponding values of conditional probability P_{26} are $p_e(1-p_e)^2$ for NBC and $(1-p_e)^3$ for FBC. Also see Problem 4.30. Conditional probabilities P_{kj} are completely described for $k,j = 1,2,...,L$ by a *channel transition matrix*

$$\mathbf{P}_c = \{P_{kj}\}; \quad k,j = 1,2,...,L \tag{4.145}$$

with the special characterizations

$$p_e = 0: \quad \mathbf{P}_c = \mathbf{I}, \tag{4.146}$$

$$p_e = 0.5: \quad \mathbf{P}_c = \{P_{kj}\} = \{2^{-R}\}\ ;\ k, j = 1,2,...,L\ . \tag{4.147}$$

for *error-free* (p_e=0) and *totally noisy* (p_e=0.5) binary channels.

With a noisy channel, the *mean-squared reconstruction error* is given by

$$\sigma_r^2 = E[(X(n)-V(n))^2] = E[Q(n) + C(n)]^2 \tag{4.148}$$

$$= \sigma_q^2 + \sigma_c^2 + 2\sigma_m^2 \tag{4.149}$$

where σ_q^2, σ_c^2 and σ_m^2 are *quantization error*, *channel error* and *mutual error* components defined by

$$\sigma_q^2 = E[(X(n)-U(n))^2] = E[Q^2(n)] \tag{4.150}$$

$$\sigma_c^2 = E[(U(n)-V(n))^2] = E[C^2(n)] \tag{4.151}$$

$$\sigma_m^2 = E[Q(n)\ C(n)] \tag{4.152}$$

The *mutual error* is the crosscorrelation between quantization and channel errors. This mutual correlation arises in the process which maps channel noise into reconstruction noise. The mutual error term in (4.152) will be seen in general to be non-zero, although the noise on the channel that causes channel error $C(n)$ is quantizer-independent.

4.9.2 Reconstruction Error Variance

The total reconstruction error variance (4.148), skipping time index n, is

$$\sigma_r^2 = E[(X-V)^2] = \int_{-\infty}^{\infty}\int_{\infty}^{\infty} (x-v)^2 p_{xv}(x,v)\,dx\,dv \tag{4.153}$$

and since

$$p_{xv}(x,v) = p(v|x)p_x(x)\ ;\ \text{and}\ p(v|x) = \sum_{k=1}^{L} P\{V = y_k|x\}\delta(v-y_k)\ , \tag{4.154}$$

$$\sigma_r^2 = \sum_{j=1}^{L}\sum_{k=1}^{L} P_{kj}\int_{\mathcal{I}_k} (x-y_j)^2 p_x(x)\,dx \tag{4.155}$$

Note that for $p_e = 0$, $\mathrm{P}_c = \{P_{kj}\}$; $k, j = 1,2,...,L = \mathbf{I}$, as in (4.146); for this case, (4.155) reduces to that for the error-free system, the familiar (4.12). The channel contribution to error variance is

$$\begin{aligned}\sigma_c^2 = E[(U-V)^2] &= \sum_{k=1}^{L}\sum_{j=1}^{L} (y_k-y_j)^2 P\{y_k, y_j\} \\ &= \sum_{k=1}^{L}\sum_{j=1}^{L} (y_k-y_j)^2 P_{kj}\int_{\mathcal{I}_k} p_x(x)\,dx\end{aligned} \tag{4.156}$$

Using the quantization error result of (4.12) and the identity $\sum_{j=1}^{L} P_{kj} = 1$, the mutual error term is

$$\sigma_m^2 = E[Q(n)\ C(n)] = \sum_{k=1}^{L}\sum_{j=1}^{L} P_{kj}(y_k-y_j)\int_{\mathcal{I}_k} (x-y_k)p_x(x)\,dx \tag{4.157}$$

The mutual error vanishes if (i) $p_e = 0$ so that $P_{kj} = \delta_{kj}$, or if (ii) the quantizer has been chosen such that its reconstruction values y_k are the centroids of intervals $\mathcal{I}_k$ (4.45b) so that the integral in (4.157) vanishes [Totty and Clark, 1967]. This

0167

is the Max-quantizer condition, and we will denote the quantizer characteristic by $Q_*(\cdot)$ and its mmse by $\min\{\sigma_q^2\} = \sigma_{q*}^2$. Thus

$$\sigma_r^2 = \sigma_{q*}^2 + \sigma_c^2 \qquad \text{for Max–quantizers } Q_*(\cdot) \tag{4.158a}$$

Experience shows that σ_m^2 can be neglected in general for small p_e, even if the quantizer is not $Q_*(\cdot)$; while σ_c^2 depends on the chosen decision and reconstruction values, and (through P_{kj}) on error probability *and* chosen channel code.

The approximate relation

$$\sigma_r^2 = \sigma_q^2 + \sigma_c^2 \tag{4.158b}$$

will be used in many of the later sections. Note that $\sigma_q^2 \geqslant \sigma_{q*}^2$ if the quantizer is non-Max. However, σ_c^2 may be much smaller in that case, so that the overall mse can be smaller than that of (4.158a). Such re-optimizations of the quantizer will be discussed in Section 4.9.4.

4.9.3 One-Bit Quantization

Closed-form solutions are possible in the case of $R=1$. The one-bit quantizer of a zero-mean random variable has only one finite decision level $x_2 = 0$ and two reconstruction values which will be denoted by

$$y_1 = -\delta\,; \quad y_2 = \delta \tag{4.159}$$

The quantization error variance is

$$\sigma_q^2 = 2\int_0^\infty (x-\delta)^2 p_x(x)dx = \sigma_x^2 + \delta^2 - 2\delta\,\delta_*\,; \tag{4.160}$$

$$\delta_* = \left[\int_0^\infty x p_x(x)dx\right] \Big/ \left[\int_0^\infty p_x(x)dx\right] = E[X|x \geqslant 0] \tag{4.161}$$

where δ_* is the mean absolute value of X, equal to the centroid of the area under positive $p_x(x)$. This is the mmse reconstruction value that was pdf-optimal for $p_e = 0$ and was therefore called δ_{opt} in that context (Section 4.3). We have already seen that this value minimizes σ_q^2 and that [see (4.39) and (4.41)]

$$\sigma_{q*}^2 = \min\{\sigma_q^2\} = \sigma_x^2 - \delta_*^2\,; \qquad \sigma_q^2 = \sigma_{q*}^2 + (\delta_* - \delta)^2\,, \tag{4.162}$$

which shows that quantization error variance increases by the square of the deviation of step size δ from δ_*. Values of optimal step size δ_* are listed in Table 4.11 for several $p_x(x)$ models. The δ_* and ϵ_{q*}^2 values are the same as the δ_{opt} and $\min\{\epsilon_q^2\}$ values of Table 4.2.

Table 4.11 Optimum step size, minimum quantization error variance, and channel coefficients of one-bit quantizers. (U: Uniform, G: Gaussian, L: Laplacian, Γ: Gamma).

pdf	δ_*/σ_x	$\epsilon_{q*}^2 = \sigma_{q*}^2/\sigma_x^2$	ξ_*
U	$\sqrt{3}/2$	0.25	3.0
G	$\sqrt{2/\pi}$	0.363	2.55
L	$1/\sqrt{2}$	0.50	2.0
Γ	$1/\sqrt{3}$	0.667	1.33

The *channel error variance* is simply obtained by noting that the magnitude of a possible channel error contribution is $|y_1-y_2| = 2\delta$ and errors $\pm 2\delta$ occur with probability $p_e/2$ each. Thus,

$$\sigma_c^2 = 4\delta^2 p_e/2 + 4\delta^2 p_e/2 = 4\delta^2 p_e \ , \tag{4.163}$$

a variance that increases linearly with p_e and squared step size.

The *mutual error variance* can be derived from (4.157) to be

$$\sigma_m^2 = E[Q(n)C(n)] = 2\delta\, p_e(\delta_*-\delta) \ , \tag{4.164}$$

and it increases linearly with step size deviation $(\delta_*-\delta)$. The *total reconstruction error variance* for the one-bit quantizer, using (4.149), is

$$\sigma_r^2 = \sigma_x^2 + \delta^2 - 2\delta\delta_*(1-2p_e) \tag{4.165}$$

resulting in an optimal step size and minimum variance given by setting $\partial\sigma_r^2/\partial\delta=0$:

$$\delta_{**} = \delta_*(1-2p_e) \ ; \quad \min\{\sigma_r^2\} = \sigma_x^2 - \delta_{**}^2 \tag{4.166}$$

Note the special results for the extreme cases of $p_e = 0$ and 0.5:

$$p_e = 0: \quad \delta_{**} = \delta_* ; \quad \min\{\sigma_r^2\} = \sigma_x^2 - \delta_*^2 \tag{4.167}$$

$$p_e = 0.5: \quad \delta_{**} = 0 ; \quad \min\{\sigma_r^2\} = \sigma_x^2 \tag{4.168}$$

Clearly, a bit error probability of 0.5 suggests a degenerate situation where the optimum strategy is not to transmit at all. Formally, this corresponds to a redesign involving a vanishing quantizer characteristic, i.e., step sizes $\delta_{**} \to 0$. For more realistic channels $(0.5>p_e>0)$, we will note that the redesign (4.164) will involve step sizes δ_{**} that do not tend to zero but are nevertheless smaller than the Max-quantizer values δ_*. The problem in utilizing (4.166) is that in many practical situations, the exact value of p_e may be unknown.

When $\delta \approx \delta_*$, the mutual error variance (4.164) vanishes, and from (4.163)

$$\sigma_r^2 = \sigma_{q*}^2 + \sigma_c^2 = (\epsilon_{q*}^2 + \xi p_e)\sigma_x^2 \ ; \quad \xi = \xi_* = 4\delta_*^2/\sigma_x^2 \tag{4.169}$$

0169

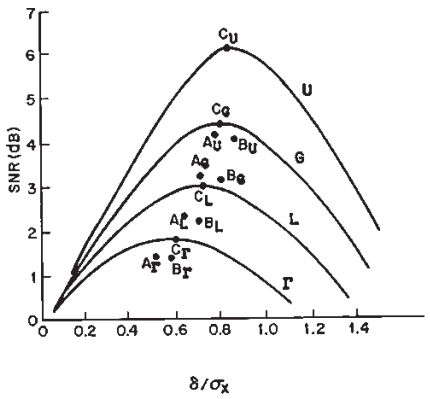


Figure 4.36 One-bit quantizer results. *SNR* versus step size δ for $p_e = 0$, and effects on *SNR* of random bit errors; $p_e = 5 \cdot 10^{-2}$ (U: Uniform, G: Gaussian, L: Laplacian, Γ: Gamma) [Noll, 1976].

The parameter ξ will be called the *channel coefficient* [Noll, 1975]. Its values ξ_* for $\delta = \delta_*$ are listed in Table 4.11. The gamma pdf, which has the highest value of ϵ_{q*}^2 in the table, also has the lowest value of ξ.

As an example for the use of Table 4.11 and (4.169), consider a Laplacian unit variance input for which $\xi = 2.0$. The error variance will increase from 0.5 with an error-free channel (p_e=0) to 0.7 with a bit error rate of 10% (p_e=0.1). An interesting interpretation of ξ_* is that it can be simply related to p_{eqc}, the error probability threshold for which $\sigma_c^2 = \sigma_{q*}^2$ (Problem 4.26):

$$p_{eqc} = (1/\xi_* - 1/4) \tag{4.170}$$

Figure 4.36 shows the dependence of *SNR* on step size for the pdf models of Table 4.11. The pdf models are denoted by usual subscripts U, G, L and Γ. The points labelled C show the optimum step size and corresponding maximum *SNR* for $p_e = 0$. In addition, points A and B show the effect on (σ_x^2/σ_r^2) of 5% (p_e=0.05) random errors; point A corresponds to a quantizer redesign $\delta = \delta_{**}$ based on (4.166), while B is the performance for $\delta = \delta_*$ (optimum step size for p_e=0). Note that the improvement due to quantizer redesign is very small. The gains due to quantizer redesign (step size reduction) will be more significant for higher error rates (Problem 4.28). They will also be seen to be more significant, for any given p_e, for multilevel quantizers ($L > 2$). As mentioned earlier, these gains can only be realized if the bit error rate p_e is indeed known exactly, or at least approximately.

4.9.4 Multilevel Quantizers

Uniform pdf Input. Closed-form solutions for σ_r^2 are generally not available for $R > 1 (L > 2)$. An interesting exception is the transmission error performance of a

0170

coding scheme that transmits signals with a *uniform pdf input* and employs a *uniform quantizer*. In this case, $\epsilon_q^2 = 2^{-2R}$ and $\xi_* = 4(1-\epsilon_q^2)$.

Assume that the step size is $\Delta_* = 2x_{max}/L$, which implies an optimum (Max-) quantizer $Q_*(\cdot)$ for $p_e = 0$ from (4.28). From (4.158a),

$$\sigma_r^2 = \sigma_{q*}^2 + \sigma_c^2 ; \qquad \sigma_{q*}^2 = \sigma_x^2/L^2 = \Delta_*^2/12 \tag{4.171}$$

With NBC coding of the quantizer output (Figure 4.35), an error in the ith most significant bit causes a displacement of $2^{R-i}\Delta_*$. If the error rate is small, the probability of more than one error per codeword is negligible, and single errors in the R bit positions are equally likely, with probability p_e, and

$$E[C^2(n)|U(n)\in\mathcal{I}_k] = \sum_{i=1}^{R} p_e\ (2^{R-i}\Delta_*)^2 = p_e\Delta_*^2(2^{2R}-1)/3 \tag{4.172}$$

because of the independence of the contributions. The quantizer level probabilities P_k are also equal, with probability $1/L$, and

$$\sigma_c^2 = \sum_{k=1}^{L} P_k E[C^2(n)|U(n)\in\mathcal{I}_k] = \Delta_*^2 p_e(2^{2R}-1)/3 \tag{4.173}$$

The reconstruction error variance [Viterbi, 1966] is

$$\sigma_r^2 = \sigma_{q*}^2 + \sigma_c^2 = (2^{-2R} + \xi_* p_e)\sigma_x^2 ; \qquad \xi_* = 4(1-2^{-2R}) \tag{4.174}$$

The channel error contribution σ_c^2 increases linearly with p_e. But it decreases as quantization becomes coarser (Figure 4.33). The special case of $\xi_* = 3$ for $R = 1$ corresponds to the ξ_* value in the first row of Table 4.11.

Re-optimization of Multilevel Quantizers. Gains due to re-optimizing a two-level quantizer were noted to be very modest in Figure 4.36. Re-optimization of a 3-bit (L=8) quantizer is described numerically in Figure 4.37 for the example of a Gaussian input. As with $R = 1$, re-optimization is based on minimizing the sum $\sigma_r^2 = \sigma_q^2 + \sigma_c^2$ by reducing σ_c^2, while using a quantizer with $\sigma_q^2 > \sigma_{q*}^2$. Such re-optimization is possible if p_e is high, and can be considered constant and known.

The figure demonstrates differences for two nonuniform 3-bit quantizers with NBC coding. Note the significant decrease in SNR with increasing p_e for the Max-quantizer $Q_*(\cdot)$. The re-optimized $Q_{**}(\cdot)$ is more robust for high error rates, but it is suboptimal if the actual error rate is small because $Q_{**}(\cdot)$ is optimized for a specific p_e value, namely 2.5%. It is therefore suboptimal for $p_e \neq 0.025$, but it still performs better than Q_* for high values of p_e. A comparison of quantizer characteristics Q_* and Q_{**} would reveal significant shrinkage (about 30%) in quantizer range, as opposed to a much smaller shrinkage (about 5%) in one-bit quantization, as given by the $(1-2p_e)$ factor in (4.166).

In general, if $p_e > 0$, system performance is maximized by setting partial derivatives of σ_r^2 [see (4.155)] (rather than σ_q^2) with respect to x_j and y_j to zero. This leads to an optimum quantizing characteristic $Q_{**}(\cdot)$ [Kurtenbach and Wintz, 1969]:

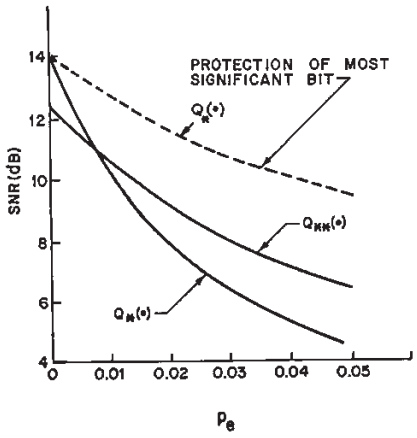


Figure 4.37 *SNR* versus p_e for $R = 3$ and Gaussian inputs, for Max-quantizer Q_* and re-optimized quantizer Q_{**} (optimized for $p_e = 2.5 \cdot 10^{-2}$) [Noll, 1975,II; ©AT&T].

$$y_{k**} = \frac{\sum_{j=1}^{L} P_{kj} \int_{\mathcal{I}_k} x p_x(x)dx}{\sum_{j=1}^{L} P_{kj} \int_{\mathcal{I}_k} p_x(x)dx}; \qquad k = 1,2,...,L$$

$$x_{k**} = \frac{\sum_{j=1}^{L} y_j^2 (P_{k-1,j} - P_{kj})}{2\sum_{j=1}^{L} y_j (P_{k-1,j} - P_{kj})}, \qquad k = 2,3,...,L \tag{4.175}$$

Note again that without errors, earlier designs result, as in (4.45). Also see Problem 4.24.

Channel Coefficients for Multi-level Quantizers. The transition probability P_{kj} was defined in (4.144). Using this expression, the channel error variance σ_c^2 (4.151) can be written as [Noll, 1975,II]

$$\sigma_c^2 = \sigma_x^2 \sum_{j=1}^{R} \xi_j p_e^j \tag{4.176}$$

where ξ_j are *channel coefficients* that reflect the effects on σ_c^2 of the quantizer characteristic, the chosen binary code and the source statistics. These coefficients give only global information about channel error variance. If the individual effects on each possible codeword are needed, *digital error coefficients* must be introduced to take into account all possible error sequences [Rydbeck and Sundberg, 1976].

If p_e and R are sufficiently small, the channel contributes to σ_c^2 only through single errors. With $P_{kj} = p_e(1-p_e)^{R-1} \sim p_e$, terms involving p_e^2, p_e^3, ... can be neglected. The mutual error term can also be neglected as before. Therefore, with $\xi_1 = \xi$, the error variance σ_r^2 reduces to a form similar to the one-bit result (4.169):

$$\sigma_r^2 = \sigma_q^2 + \xi p_e \sigma_x^2 = (\epsilon_q^2 + \xi p_e)\sigma_x^2 \tag{4.177}$$

Note that although (4.177) is functionally similar to (4.169), ξ (as well as σ_q^2) are functions not only of input pdf, but also of R.

Table 4.12 lists channel coefficients for various pdf models, and for two types of quantizers, nonuniform Max-quantizer Q_* and uniform Max-quantizer Q_{L*}. The values for $R = 1$ are those from Table 4.11, and the NBC values for uniform pdf were obtained in closed form from (4.175). Note that coefficients for a uniform quantizer are in general less than those for a nonuniform quantizer. This is due to the fact that displacements Δ_{ij} are in general smaller, for given i and j, with uniform quantizers. For example, with $R = 2$ and mmse quantization of a gamma-input, the output levels in Q_* are proportional to $\pm$ 0.313 and $\pm$ 2.223 (Table 4.3), while in Q_{L*}, they are proportional to $\pm$ 0.5 and $\pm$ 1.5. With a most significant bit in error and an NBC code, the displacement Δ_{ij} is always half the quantizer range, which is 2.536 for Q_* and 2.0 for Q_{L*}. Table 4.12 also indicates that channel coefficients are the least for a uniform pdf for $R > 2$, and that for nonuniform pdf sources and $R \geqslant 2$, FBC outperforms NBC.

In the one-bit example, we saw that quantizer re-optimization decreased the value of ξ by a factor $(1-2p_e)$ (Problem 4.28). This was directly related to quantizer shrinkage by a factor $(1-2p_e)$. Re-optimizations of multibit quantizers lead to similar step size shrinkage. In multibit quantization, it is also more important to choose an appropriate channel code.

Figure 4.38(a) shows ξ_* as a function of R(1 to 8), input pdf and binary code. Note that for the speech-typical gamma pdf, the set of three solid curves shows that the FBC code has better ξ_* values than the NBC code and that the FBC values are very close to those for MDC. In the case of image inputs, a uniform input pdf is typical, and thus the NBC code is optimal, as seen from the pair of dashed curves.

Figure 4.38(a) also shows the values of ξ in log-quantization of a Laplacian input. Results are shown for $R = 8$, FBC and MDC, and for two levels of input power, -17 dB and -40 dB relative to the level that would begin to cause overload

Table 4.12 Channel coefficients ξ_* [Noll, 1976].

R		1	2		3	
pdf			NBC	FBC	NBC	FBC
U		3.0	3.75	4.50	3.94	4.88
G	Q_*	2.55	4.97	4.65	6.91	5.75
	Q_{L*}		4.96	4.57	7.21	5.57
L	Q_*	2.00	7.08	5.30	12.67	7.50
	Q_{L*}		5.91	4.29	11.22	6.39
Γ	Q_*	1.33	9.07	6.28	18.55	9.11
	Q_{L*}		5.68	3.62	13.31	6.64

0173

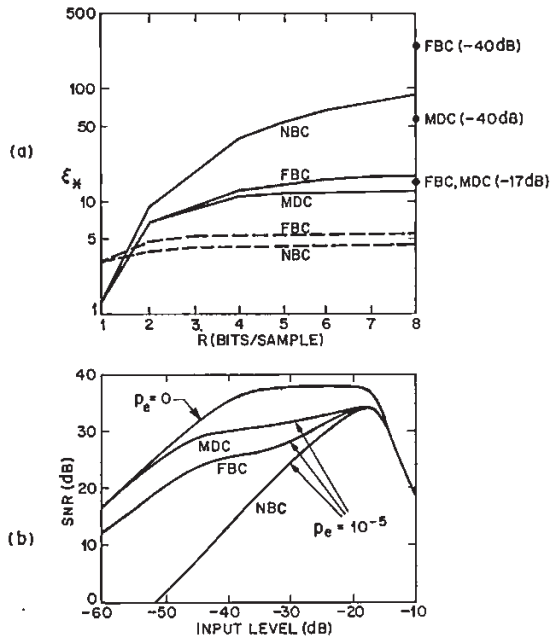


Figure 4.38 Effect of binary code on channel error performance of quantizers. (a) Channel coefficients ξ_* as a function of R, input pdf and binary code [Zelinski;©1979, IEEE] [Rydbeck and Sundberg, 1976]. Solid curves refer to a gamma input pdf and dashed curves to a uniform input pdf. Quantizers are optimized for mmse in each case. The black dots refer to log-quantization of a Laplacian input for two input levels, −40 dB and −17 dB, relative to the level that begins to introduce overload. (b) Dynamic range characteristics of A 87.56 quantization for a Laplacian input, $R = 8$ bits/sample and $p_e = 10^{-5}$ [Rydbeck and Sundberg, 1976. Reprinted with permission].

[Rydbeck and Sundberg, 1976]. With companded quantizers, the importance of selecting a channel code is very significant if the system is underloaded (as shown by the −40 dB example).

The relative performances of NBC, FBC and MDC are also shown by the dynamic range curves of Figure 4.38(b) for the example of A 87.56 quantization, $R = 8$ and $p_e = 10^{-5}$. Note that the range of the Laplacian input signal over which the *SNR* of the companded system is at least 30 dB is 10, 12 and 28 dB respectively, for NBC, FBC and MDC.

Effect of Bit Protection. In multi-bit quantization, it is realistic to protect the more significant bits from transmission error (for example, by the use of increased transmitter power or by redundant transmissions), because most of the contribution to σ_c^2 is due to the large displacements δ_{ij} caused by bit errors in the first and second most significant bits. For example, with $R = 3$ and NBC, protection of the most significant bit will prevent displacements of the form δ_{15}, δ_{26}, δ_{37} and δ_{48} (Figure 4.35). Setting corresponding displacement terms to zero in (4.156) leads to

0174

a smaller value for σ_c^2 and hence for the channel coefficient ξ. Figure 4.37 shows the effect of protecting the most significant bit when the input is Gaussian and $p_e = 0.025$ (Also see Problem 4.29). Finally, there are instances (for example, gamma pdf, FBC, R=4) where protection of the second most significant bit (most significant magnitude bit) is more advantageous than protection of the first most significant (sign) bit.

The reduction in ξ by means of error protection will of course have to be interpreted carefully because the protection of the significant bits involves in general an increase of transmission capacity, or a reduction in the number of quantizer bits for a given total transmission capacity. As seen in Figure 4.33, such reallocation of bits is a very logical thing to do in the channel-limited region of operation.

4.10 Adaptive Quantization

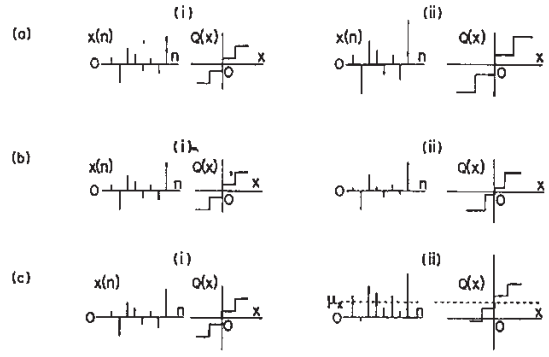
Quantizer adaptations to changing input conditions can take several forms, and three important cases are illustrated in Figure 4.39. In each of these cases, we see two pairs (i) and (ii) of sample input sequence $x(n)$ and an input-matched quantization characteristic $Q(x)$.

In (a) we show adaptation of $Q(x)$ to the *dynamic range* (or *variance*) of $x(n)$. The range of $Q(x)$ as well as that of $x(n)$ increase by a factor of 1.5 as we go from (i) to (ii). There is no change in the shape of the input pdf, and no change of the functional form of $Q(x)$, which happens to be a uniform characteristic.

In (b) we show adaptation of $Q(x)$ to a change of *shape in input pdf*, without any change in dynamic range of either $x(n)$ or $Q(x)$. As we go from (i) to (ii), very small values of $x(n)$ become more likely and $Q(x)$ matches this condition by allowing for a minimum step size in (ii) that is smaller than that in (i). Thus $Q(x)$ is uniform in (i), but nonuniform in (ii).

In (c) we show the result of adaptation of $Q(x)$ to a changing *mean of the input*. As we go from (i) to (ii), the mean changes from zero to a non-zero positive

Figure 4.39 Three ways of adapting $Q(x)$ to changing input statistics, to match (a) input dynamic range or variance; (b) shape of input pdf; and (c) mean value of input.



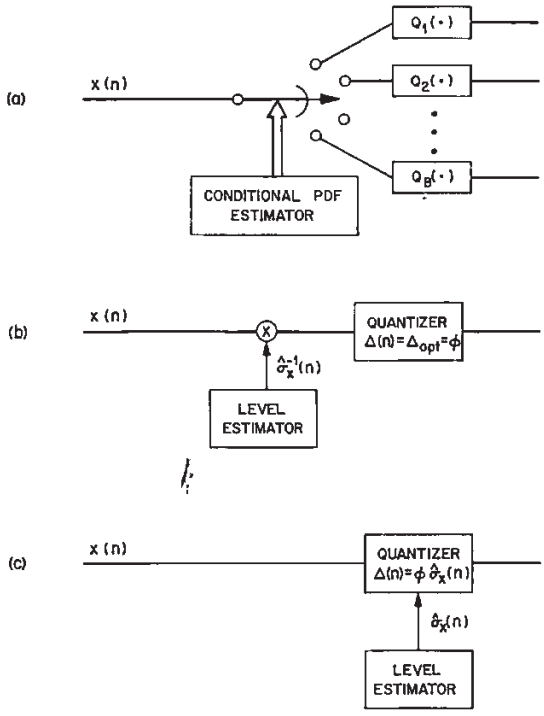
0175

value μ_x, and the quantizer matches this condition by simply shifting or displacing $Q(x)$ vertically, by an amount μ_x. The shape of $Q(x)$ as well as the dynamic range remain unchanged.

The logarithmic companding technique of Section 4.5 was already shown to enhance the performance of a quantizer with an input that has a large dynamic range. A log-quantizer is still *time-invariant*, however; and in that sense, it is not the ideal device for quantizing nonstationary inputs; and although log-quantization has proven to be extremely useful in speech quantization with $R = 8$ bits/sample (Chapter 5), speech coding systems for lower bit rates will be seen invariably to use adaptive quantization of some kind. The better performance of adaptive quantization at low bit rates will be demonstrated by examples with $R \leqslant 4$ bits/sample (Figure 4.43 for $R = 3$, and Figure 6.21). The price paid for using an adaptive algorithm will be a very modest amount of encoding delay, which in speech work can range from a fraction of a millisecond to several milliseconds, depending on the logic.

A logical arrangement for such inputs is *switched quantization*; this consists in providing a bank of B fixed quantizers, and switching among them as appropriate, in response to changing input statistics. As shown in Figure 4.40(a), the quantizers in the bank may be either uniform or nonuniform, zero-mean or non-zero mean,

Figure 4.40 Block diagrams of (a) switched quantization and (b) (c) adaptive quantization.



0176

with large step size or small step size [Schlink, 1972] [Cohen, 1972,1973] [Zschunke, 1977]. The choice of quantizer characteristics is dictated by conditional input statistics; the conditioning is based on the entire past of the input in the ideal case, but in practice only based on the state of recent input or output samples. Switched quantizing systems have been used in both speech and video applications, and gains over fixed quantization have been noted with as few as $B = 2$ quantizers in a bank. In the important special case of *symmetrical zero-mean pdf's* of known shape, adaptation to input statistics consists merely in matching quantizer dynamic range to input variance, as in Figure 4.39(a). With *non-independent inputs*, conditional pdf's have strictly non-zero means in general, even if the unconditional pdf is a zero-mean function, as in Figure 4.39(c). These non-zero means can be simply taken into account by prediction procedures (Chapter 6) that essentially provide quantizer shifts. The zero-mean symmetrical pdf model for quantizer input is therefore quite useful in a wide class of applications.

The *adaptive quantizers* of this section correspond to the limiting case of $B \rightarrow \infty$, at least in principle. We shall concentrate on the important case of a zero-mean input pdf of known shape and unknown level (standard deviation σ_x); and although we mention nonuniform quantizers when appropriate, most of our discussion will relate to the case of a uniform $Q(x)$, with adaptive size, as in Figure 4.39(a). As shown in Figures 4.40(b) and (c), the output of the level estimator in these systems is expected to provide an essentially continuous adaptation of quantization characteristic. In Figure 4.40(b), the goal of this adaptation strategy is to offer a unit level input to a fixed quantizer for all time n. In Figure 4.40(c), the idea is to adapt the quantizer step size to an input-matched value for all n.

We will note presently (Figure 4.43) that adaptive quantization (especially for low values of R) can provide a better quantization error performance than companded quantization if the input is nonstationary. This fact will be significant particularly in the context of coarse quantizers for the low bit rate encoding of speech (Chapters 6 to 8, 11 and 12). There are several other reasons why adaptive quantizers are attractive. First, time-varying quantizing systems can be designed to be even less signal-specific than a time-invariant companded device. Second, there is an increased interest in the coding of individual speech signals, as distinguished from multiplexed inputs; and in a per-channel codec environment, step size adaptations to individual user waveforms (based on their separate histories) is simpler to implement than in a multiplex environment. A third attraction of time-varying quantizers is that they provide *code trees* with less structure than those of time-invariant quantizers. This will be explained presently in the context of Figure 4.42. The desirability of destructuring code trees will be explained further in Chapter 9.

In the interest of uniformity, the term *adaptive quantization* will henceforth refer always to quantizers that operate with a time-varying step size $\Delta(n)$ as in Figure 4.40(c). Furthermore, the adaptive quantizer, at any given time, will be assumed to have a uniform transfer characteristic unless otherwise mentioned. The basic problem then is to adapt step size $\Delta(n)$ to changing input variance $\sigma_x^2(n)$. Ideally, the time-varying step size would continuously follow the value Δ_{opt} that a fixed quantizer would employ if the input were stationary, or at least locally

0177

stationary, and the variance σ_x^2 were known. We know from previous sections that Δ_{opt} is related to σ_x via a parameter, say ϕ, that depends only on input pdf and number of bits per sample R. For example, if $R = 2$, if $p_x(x)$ is Gaussian and if $Q(x)$ is uniform, $\phi = .996 \sim 1.0$ from Table 4.1. The operation of an adaptive quantizer can therefore be defined in the form

$$\Delta(n) = \phi\hat{\sigma}_x(n) \; ; \quad \phi = \text{constant} \tag{4.178}$$

The quantizer loading factor $f_l = \Delta/\sigma_x$ (Section 4.3) therefore tends to be *independent* of σ_x in an ideal adaptive quantizer. A future example (Figure 4.49) will show the dependence of f_l on σ_x in quantizers with different degrees of adaptation.

For a nonstationary input, σ_x is *time-variable*, and the problem of adaptive quantization, according to (4.178), is one of estimating it continuously. The problem is very similar if the input is stationary, or locally stationary, with *unknown* variance σ_x. Subsequent sections will describe how to estimate σ_x or related measures of input level such as $E[|X|]$ or $|x|_{\max} = \max\{|X|\}$; and how these estimates can be derived from observations on $\{x(n)\}$, and very often from observations on the quantizer output $\{y(n)\}$; as the number of bits/sample R increases, estimates of σ_x based on quantizer output samples get increasingly closer to estimates of σ_x based on unquantized input samples. *A significant result of this section is that for an exponentially weighted variance estimate, based on the entire available past of $y(n)$, is exactly equivalent to one that is possible with an explicit memory of only the latest value of $y(n)$ when $R \geqslant 2$.*

A subject of great interest is the distinction between the *static* performance of an adaptation logic [as given by the matching of step size to a locally stationary value of $\sigma_x(n)$], and its *dynamic* performance [as measured by how quickly the changing step size sequence can respond to a changing $\sigma_x(n)$ and exhibit a narrow short-term distribution centered on the local mean value of step size]. Adaptation logics generally exhibit a compromise between the static and dynamic responses, and these two properties have associated perceptual ramifications in speech and image coding.

Our theoretical discussions will refer mainly to the example of independent and identically distributed (i.i.d.) input samples. Many results on level estimation are available for such i.i.d. inputs. Also, in many important situations (such as the differential encoding of speech and video), the quantizer input can be modeled to a fair approximation as an i.i.d. sequence.

4.10.1 "Forward" and "Backward" Adaptations — AQF, AQB

We assume a zero-mean input $x(n)$, and the problem will be one of estimating input level in terms of σ_x or related parameters $E[|X|]$ or $|x|_{\max}$. We shall consider *forward* estimates based on unquantized samples $x(n)$, as well as *backward* estimates derived from samples $y(n)$ of the quantizer output. Respective systems are denoted by AQF (*adaptive quantization with forward estimation*) and AQB (*adaptive quantization with backward estimation*). The AQF system has

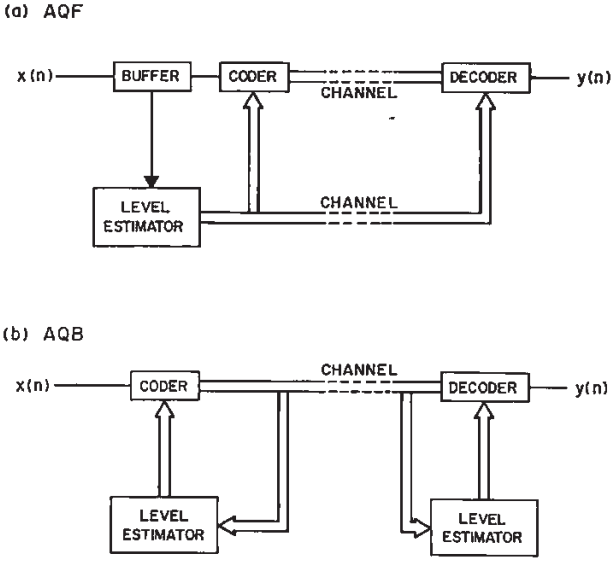
0178

also been known by the term *block companding* [Croisier, 1974]. Block diagrams of these procedures appear in Figure 4.41.

Forward estimates of step size are unaffected by quantization noise. Step size estimates $\Delta(n)$ in AQF are therefore more reliable than those in AQB. The AQF technique implies, however, that level information is to be explicitly transmitted to a remote decoder using typically about 5 to 6 bits per step size sample; and this translates to a corresponding amount of *side information* that needs to be additionally transmitted to the receiver. The AQF technique, however, permits special protection of step size information in transmission because this information is typically only a small fraction of the overall transmission rate; adding a certain percentage of redundancy to this information increases overall transmission rate by a much smaller percentage. Forward estimation also implies an estimation delay in the encoding operation. We shall use equivalent terms "learning period" or "AQF block length" for this parameter N (number of learning samples). The encoding delay, which may be in the order of 16 ms for speech (see Table 4.13), is a significant transmission parameter — unacceptable in some systems, but uncritical in some. Finally, AQF also requires a buffering of "unquantized" input samples — practically, a buffering of samples quantized an order of magnitude finer than the resolution of the AQF system quantizer itself.

The learning period N allowed for a forward level estimator is an interesting issue. When the input signal is locally stationary, the best learning period is a compromise among several factors: (i) the need to minimize the encoding delay, (ii) the need to keep down the information rate in level-transmission and (iii) the need

Figure 4.41 Adaptive quantization with (a) forward estimation of input level (AQF), and (b) backward estimation of input level (AQB).



0179

Table 4.13 *SNR* (dB) values in 3-bit quantization of speech with forward adaptation (AQF) [Noll, 1974].

Nonuniform Quantizers	Nonadaptive	Adaptive $N = 128$ (16 ms)	Adaptive $N = 1024$ (128 ms)
μ-law (μ=100; $x_{max}=8\sigma_x$)	9.5	-	-
Gaussian	7.3	15.0	12.1
Laplacian	9.9	13.3	12.8
Uniform Quantizers			
Gaussian	6.7	14.7	11.3
Laplacian	7.4	13.4	11.5

to track the changing statistics of speech with an appropriate speed. Learning periods in the order of 16 ms ($N = 128$ samples with 8 kHz speech) are appropriate in speech communication [Noll, 1974] [Jayant, 1975]. *SNR* losses in going from 16 ms to, say 128 ms, can be as high as 3 dB. This is seen in the *SNR* results of Table 4.13. As noted in Chapter 2, the input pdf to the quantizer is Gaussian if the AQF block length is short enough to ensure local stationarity (say 16 ms); with a 128 ms block length, the pdf tends to the Laplacian. If the block length is much less than 16 ms, the side information for step size can become excessive (Problem 4.32), and the assumption of a zero-mean input becomes decreasingly valid [Esteban, Menez and Temime, 1976]. In that case, quantizer adaptations should include an adaptation to input mean, as in Figure 4.39(c).

Shorter learning periods such as $N = 16$ have been recommended in image coding applications; the side information in these systems is still kept at a reasonable level by employing a relatively small dictionary of step sizes. In one example, where the AQF technique is used as part of a DPCM coder (Chapter 6), $N = 16$; but the allowed number of step sizes is only four, so that the side information for step size is only 1/8 of a bit per pixel [Habibi and Batson, 1978].

Backward estimators extract step size information $\Delta(n)$ from the recent history of the quantizer output, and thus avoid the problems of estimation delay and level transmission. Quantization noise in estimator samples clearly degrades level-tracking performance in AQB; and this degradation increases with the coarseness of quantization. An adaptive quantizer with backward estimation is a nonlinear system with feedback, and it is not obvious that it will be stable. It can be shown, however, [Stroh, 1974], that the system is indeed stable in at least one sense: if the input $x(n)$ is bounded, so are the backward standard deviation estimate $\hat{\sigma}_x$, step size and the estimation error.

Figure 4.42 shows a speech input $x(n)$, output waveforms $y(n)$ in typical AQF and AQB systems, and also step size variations in AQF and AQB. Note that the AQF system illustrated is *blockwise-adaptive*, or *periodically adaptive*; its step size $\Delta(n)$ gets updated once every (locally stationary) block, and remains constant over the duration of the block. Note also that the AQB is stable in the bounded $\Delta(n)$-sense. The $y(n)$ waveform with the fixed quantizer uses a total of 4 possible amplitudes by definition. On the other hand, $y(n)$ waveforms in AQF and AQB systems have the flexibility to use a much greater number of amplitude levels, as

0180

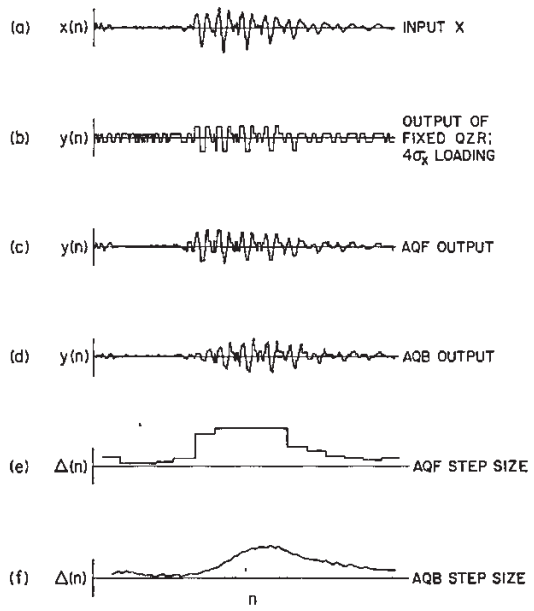


Figure 4.42 Two-bit quantization with fixed and adaptive quantizers: (a) input speech $x(n)$; and output $y(n)$ in (b) fixed quantizer with $4\sigma_x$ loading and in (c) AQF-type and (d) AQB-type adaptive quantizers. Variations of step size $\Delta(n)$ are shown in (e) for AQF and in (f) for AQB [Noll, 1975,I; ©AT&T].

indicated in parts (c) and (d) of the figure. This is the "destructuring" of the code tree that we mentioned earlier. The concept will be re-invoked in Chapter 9.

The results of Figure 4.43 refer to the 3-bit quantization of a bandlimited (300-3400 Hz) speech waveform sampled at 8 kHz [Noll, 1974]. The measures of quantizer performance used in the illustration are *SNR*, $SNR\ (m)$ and $SNRSEG = E[SNR\ (m)]$. The first quantity is the conventional signal-to-quantization-error ratio averaged over the entire speech utterance, and expressed in dB. Temporal variations of signal-to-noise-ratio can be studied by computing the ratio once for every segment of 128 samples (16 ms), to give the short-term quantity $SNR\ (m)$, where m is the segment number. The last measure of performance, *SNRSEG* or the *segmental SNR* (Appendix E) [Noll, 1974], is the average of the $SNR\ (m)$ (dB) values, as computed over the total length of the speech waveform. Bursts of low $SNR\ (m)$ values that are typical in the nonadaptive quantization of low-level speech segments leave the conventional *SNR* unaffected, but they are better reflected in the value of *SNRSEG*.

Figure 4.43 shows plots of short-term input speech $\sigma_x^2(m)$ and $SNR\ (m)$ for several 3-bit quantizers. The nonadaptive quantizer is a fixed nonuniform quantizer matched to the speech pdf. For the specific illustrated waveform, this quantizer in fact performs better, on the average, than a μ-law log-quantizer, and realizes slightly higher values of *SNR* and *SNRSEG*. However, the weak-signal

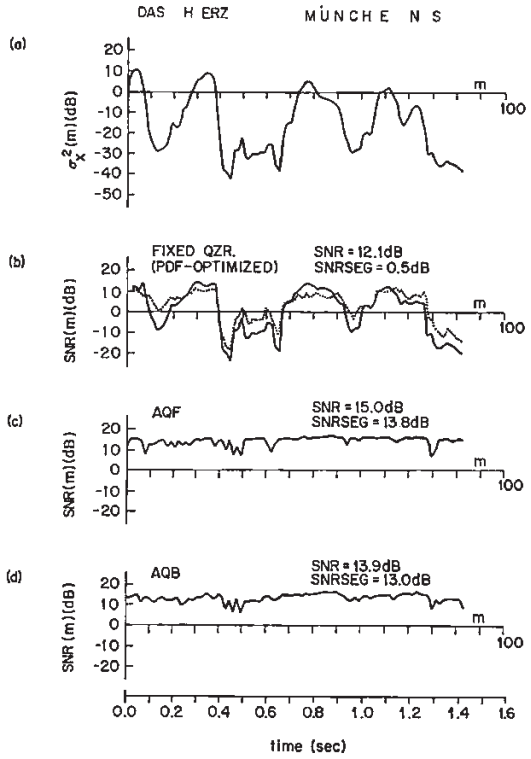


Figure 4.43 Plots of (a) short-term speech variance $\sigma_x^2(m)$ and short-term signal-to-noise ratio $SNR\ (m)$ in 3-bit quantization with (b) time-invariant- (c) AQF- and (d) AQB algorithms [Noll, 1974].

performance is better with the μ-law quantizer. Its $SNR(m)$ characteristic is indicated by the dotted waveform in (b). The adaptive quantizers use nonuniform quantizing characteristics optimal for the Gaussian pdf. The AQF estimator is based on a learning buffer of 128 samples, while the backward estimator uses a one-word memory logic to be described in Section 4.10.3. The benefits of adaptive quantization are evident from the waveforms of Figure 4.43 and the associated *SNR* values that appear alongside those waveforms. Note especially the 13 dB *SNRSEG* gain in adaptive quantization! This is due to the fact that unlike the nonadaptive systems, AQF and AQB systems maintain a fairly constant and *non-negative SNR*-versus-time characteristic for all input speech segments in the illustration.

The Special Case of $R = 1$. In a one-bit quantizer, magnitude information is available only at the quantizer *input*; the output preserves only (two levels of) sign information. For this reason, backward step size adaptation is possible in one-bit

0182

quantization only in the context of highly correlated input sequences. In this case, magnitude information is reflected by the structure in the output bit stream. This structure is indeed the basis of many classes of adaptive deltamodulation (DM-AQB, or simply, ADM) used for both speech and image coding (Chapter 8).

4.10.2 Estimators of Input Variance or Input Level

As mentioned earlier, step size adaptations can follow input variance via

$$\text{AQF:} \quad \Delta(n) = \phi\hat{\sigma}_x(n) \; ; \quad \hat{\sigma}_x(n) = [(1/N)\sum_{i=0}^{N-1} x^2(n+i)]^{1/2} \quad (4.179a)$$

$$\text{AQB:} \quad \Delta(n) = \phi\hat{\sigma}_x(n) \; ; \quad \hat{\sigma}_x(n) = [(1/N)\sum_{i=1}^{N} y^2(n-i)]^{1/2} \quad (4.179b)$$

The input window for variance estimation contains N samples in each case; it extends from input sample n to $n+N-1$ in AQF; and from output sample $n-N$ to $n-1$ in AQB. For a stationary, uncorrelated input, the *learning period* N that is necessary for good estimation is at least an order of magnitude less than the value (say $N = 128$) that is needed for learning the variance of a correlated quasi-stationary input such as 8 kHz sampled speech. Furthermore, the convergence to input variance will be somewhat slower in the AQB mode, where the estimation is based on noisy quantized samples.

The estimator in (4.179b) requires an explicit memory of N samples. We now show that it is possible to estimate the variance with a shorter explicit memory of samples if we allow the x^2 (or y^2) terms in the estimate to be suitably weighted; and that the resulting estimate can be still nearly optimal in a certain sense. The weighting of interest is *exponential* and the magnitude of the weighting decreases monotonically with the age of the x^2 (or y^2) term in question. For example, if the estimate is based on $y(n)$ values,

$$\hat{\sigma}_y^2(n) = (1-\alpha_l)\sum_{i=1}^{\infty} \alpha_l^{i-1} y^2(n-i) \quad ; \quad 0 < \alpha_l < 1 \qquad (4.180)$$

The term α_l^{i-1} provides the desired weighting, and $(1-\alpha_l)$ is simply a normalizing term which makes the sum of weights equal to unity.

Note that for values of α_l close to 1, the weighting of adjacent x samples is very nearly equal; and as $\alpha_l \to 1$ the *learning period* $N = (1+\alpha_l)/(1-\alpha_l)$ (Problem 4.33) tends to ∞. In this sense, an exponentially weighted estimator with $\alpha_l \to 1$ is basically equivalent to the unweighted logic with large N. Values of α_l closer to zero imply an estimator that is based heavily on the most recent samples $y(n-1), y(n-2),\ldots$. In fact, the more nonstationary the input, the closer should α_l be to zero. It can be shown that (4.180) can be written in the recursive form (Problem 4.34)

$$\hat{\sigma}_y^2(n) = \alpha_l\hat{\sigma}_y^2(n-1) + (1-\alpha_l)y^2(n-1) \qquad (4.181a)$$

0183

$$[\hat{\sigma}_y^2(n)/\hat{\sigma}_y^2(n-1)] = \alpha_l + (1-\alpha_l)\ y^2(n-1)/\hat{\sigma}_y^2(n-1) \qquad (4.181b)$$

The step size $\Delta(n)$ is proportional to $\hat{\sigma}_x(n) \simeq \hat{\sigma}_y(n)$, assuming adequately fine quantization. We can therefore interpret the left hand side of (4.181b) as a ratio of successive squared step sizes, or as the square of a *step size multiplier* $M(n-1) = \Delta(n)/\Delta(n-1)$. Note that $y^2(n-1)$ is the corresponding squared output $[H(n-1)\Delta(n-1)/2]^2$ of a midrise quantizer where $\pm H(n-1) = 1,3,5,...$; and assume that the best guess of $\Delta(n-1)$ is $\phi\hat{\sigma}_x(n-1)$ as per (4.178). With these assumptions, it can be shown that the step size multiplier for an adaptive quantizer with a *one-word memory* is

$$M(n-1) = \left[\alpha_l + (1-\alpha_l)\ H^2(n-1)\phi^2/4\right]^{1/2} \qquad (4.181c)$$

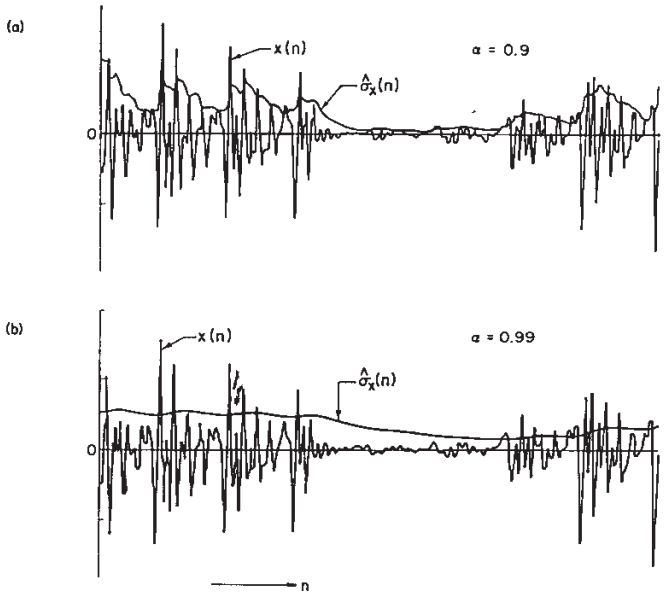
This type of adaptive quantization will be discussed at length in Section 4.10.3.

The parameter α_l determines the *time constant* of adaptation: the time taken for $\hat{\sigma}_y$ to decay to a fraction e^{-1} of the initial value when $y(n)$ has become an all-zero sequence. If the sampling rate is f_s, this time constant will be (Problem 4.35)

$$\tau = 2(f_s \log_e \alpha_l^{-1})^{-1} \qquad (4.182)$$

Figure 4.44 shows the effect of α_l on the speed of adaptation to a speech waveform segment. The value of (a) $\alpha_l = 0.90$ causes fairly rapid changes of $\hat{\sigma}_x$ that follow

Figure 4.44 Speech variance estimates in (a) instantaneous and (b) syllabic adaptation. Values of $\hat{\sigma}_x(n)$ have been magnified for clarity, but the scaling factors are the same in (a) and (b) [Barnwell et al., 1974. Reprinted with permission].



the local peaks in the input waveform, with a time constant $(f_s \log_e \alpha_l^{-1})^{-1}$ of 2.6 ms; this kind of adaptation is called *instantaneous adaptation.* The value of (b) $\alpha_l = 0.99$ causes slower changes in $\hat{\sigma}_x$ that do not follow local waveform peaks, but only their envelope, with a time constant of 25.0 ms; this kind of adaptation is called *syllabic adaptation.* These two types of adaptation have clear distinctions in terms of the *dynamic* and *static* performances mentioned earlier.

Level Estimates Based on Maximum- and Average-Magnitude Statistics. A simple approach to adaptive quantization [Golding and Schultheiss, 1967] is to use local values of peak output magnitude $|y|_{\max} = \max\{|y(n)|\}$ to vary the overload level in image quantization. The adaptations can be done in an AQB mode; prior to the quantization of every horizontal line i, the overload level $|y|_{\max}$ is set equal to the peak output from the N samples of the previous line. In fact, the lower saturation level of the quantizer is also varied independently, and matched to local minima $|y|_{\min} = \min\{|y(n)|\}$, but the adaptation based on maximum value is more critical in that it controls overload errors, which tend to be dominant contributors to total quantization noise. The scheme achieves a typical 2 dB of *SNR* gain in 4-bit quantization, as compared with a nonadaptive scheme.

Adaptive quantizers for speech have also employed the maximum-input criterion [Croll, Moffat and Osborne, 1973] [Duttweiler and Messerschmitt, 1975], [Jayant, 1975] [Noll, 1975], but strictly in an AQF mode. The correlations in typical speech waveforms are much less than those between the adjacent image lines: and this makes backward estimation based on extremal statistics relatively less reliable in speech quantization.

Maximum-magnitude logics can be summarized in the form

$$\text{AQF:}\quad \Delta(n) = \theta_1 |x|_{\max};\quad |x|_{\max} = \max\{|x(n+i)|\};\quad i = 0,1,2,\ldots,N-1 \tag{4.183a}$$

$$\text{AQB:}\quad \Delta(n) = \theta_1 |y|_{\max};\quad |y|_{\max} = \max\{|y(n-i)|\};\quad i = 1,2,\ldots,N \tag{4.183b}$$

where θ_1 is a scaling constant whose best value can usually be related to the constant ϕ of (4.178) for a given pdf [Jayant, 1975]. Adaptive logics based on maximum-magnitude logics are particularly appropriate when an important criterion is the control of overload distortion; and for simultaneous control of overload distortion as well as long term mse in high-quality PCM systems for speech, it is imperative to use extremely small values of learning period (say, 2 ms); the 8 ms that was recommended earlier for variance estimators is more typical of low bit rate differential PCM (Chapters 6 and 8) and transform coding (Chapter 12) systems, where the use of AQ is more common than in PCM.

Another simple indicator of signal level is the *average magnitude* function, and resulting adaptation logics can be summarized in the form

$$\text{AQF:}\quad \Delta(n) = \theta_2 \left[(1/N) \sum_{i=0}^{N-1} |x(n+i)| \right] \tag{4.184a}$$

$$\text{AQB:}\quad \Delta(n) = \theta_2 \left[(1/N) \sum_{i=1}^{N} |y(n-i)| \right] \tag{4.184b}$$

0185

where θ_2 is a pdf-specific scaling factor. For a given pdf, the ratio of θ_2 to the factor ϕ in (4.178) is simply the reciprocal of a corresponding (δ_{opt}/σ_x) value from the theory of one-bit quantization (Table 4.2).

A properly weighted average-magnitude (Problem 4.36) estimate permits a recursive computation, and this makes the average-magnitude function very attractive for practical adaptation procedures [Castellino et al., 1974]. We have seen in (4.181) that a similar recursive realization is possible for a weighted variance estimate as well.

4.10.3 Adaptive Quantization with a One-Word Memory

The recursive variance estimator (4.181b) suggests that effective adaptations can be realized with an explicit memory of only one sample $x(n-1)$ (or a quantized version $y(n-1)$ of it), assuming that all valuable information in the quantizer history is available in condensed form in a slowly varying parameter $\hat{\sigma}_x^2(n-1)$ or the corresponding step size $\Delta(n-1)$. We shall now explore such an adaptive logic. It is instructive to begin with an intuitive description of the adaptation strategy, and introduce the notion of step size multipliers. These multipliers have been introduced already in (4.181) as square roots of the ratio $\hat{\sigma}_x^2(n)/\hat{\sigma}_x^2(n-1)$. This interpretation will apply to any kind of quantizer, but the specific adaptation procedure of this chapter will apply only to quantizers where the number of levels exceeds 4 ($R \geqslant 2$). In such quantizers, the output contains both sign and magnitude information, and the latter part is the one that is exploited in step size modifications. The procedure to be described has been referred to as *adaptive quantization with a one-word memory* [Jayant, 1973]; the term "one word" refers to the latest R-bit output of the quantizer; this is the word that carries information about the latest sample $y(n-1)$. In the case of $R = 1$, more than one word of memory is needed for backward step size adaptation (Chapter 8).

Step Size Adaptations. Figure 4.45 explains adaptive quantization for the example of $R = 3$ bits/sample, $L = 8$, and a set of $L/2 = 4$ step size multipliers M. Only 4 distinct multiplier values are shown in view of the symmetry of zero-mean quantization, and with the notion that step size multipliers should depend only on the magnitude (not the sign) of the quantizer output. If the latest output level from the midrise quantizer is

$$y(n-1) = H(n-1)\,\Delta(n-1)/2; \quad \pm H(n-1) = 1,3,5,\ldots,2^{R-1}; \quad R \geqslant 2 \qquad (4.185a)$$

the adaptation logic derives the next step size $\Delta(n)$ as the product of $\Delta(n-1)$ and a multiplier $M(\cdot)$ that is a time-invariant function of the latest *magnitude index* $|H(n-1)|$:

$$\Delta(n) = M(|H(n-1)|)\;\Delta(n-1) \qquad (4.185b)$$

In the example of Figure 4.45, the four possible values of $|H(n-1)| = 1,3,5$ and 7 lead to the respective multipliers M_1, M_2, M_3 and M_4.

0186

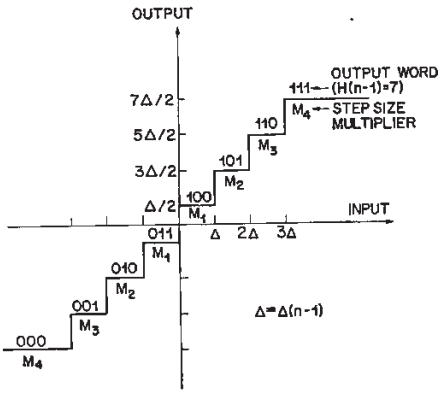


Figure 4.45 Adaptive 3-bit quantization (AQB) with a one-word memory. The figure shows a snapshot of quantizer characteristic $Q(x)$ at time $(n-1)$ [Jayant;©1973, AT&T].

A reasonable design of multipliers is of the form

$$1 > M_1 \leqslant M_2 \leqslant M_3 \leqslant M_4 > 1 \tag{4.186}$$

In this algorithm, an output sample of $\pm 7\Delta(n-1)/2$ (NBC codeword 000 or 111) will indicate that the quantizer is probably in *overload*, and the step size is therefore *increased* by a factor $M_4 > 1$. On the other hand, an output of $\pm \Delta(n-1)/2$ tends to serve as an *underload* cue, and the step size is *decreased* in this case by a factor $M_1 < 1$. Intermediate conditions of quantizer loading suggest suitable intermediate modifications M_2 and M_3. Note that step size increases and decreases imply that the entire transfer function $Q(x)$ is expanded and contracted, like an accordion; and in the example of Figure 4.45, the uniform nature of the quantization characteristic is maintained for all time n as in Figure 4.39(a). Figure 4.45 is actually the snapshot, at time $n-1$, of a variable-range (variable step size) quantizer. As a trivial case, if all the multipliers are unity, the result is a time-invariant uniform quantizer.

It is important that in practice, the adaptations (4.185) are subject to constraints on the maximum and minimum values of step size. Formally,

$$\Delta_{\min} \leqslant \Delta(n) \leqslant \Delta_{\max} \tag{4.187}$$

The value of $\Delta_{\min}$ controls the level of idle-channel noise (granular noise with a zero, or near-zero input), while the value of $\Delta_{\max}$ controls the level of overload distortion. Optimum values of $\Delta_{\min}$ and $\Delta_{\max}$ are functions of R (for example, see Figures 6.12 and 8.20), and examples will be mentioned as appropriate. For a given variance of the coder input, the maximum step size tends to be smaller in DPCM coding (Chapter 6) than in PCM coding (Chapter 5). This is because of the presence of a prediction process in DPCM. This process leads to a quantizer input variance that is usually smaller than the coder input variance, thus permitting smaller values of $\Delta_{\max}$ for a given degree of overload.

0187

The algorithm (4.185) assumes an initial step size $\Delta(0)$. In principle, this can be any empirical value in the allowed range of step size values. It is often meaningful, however, to set $\Delta(0) = \Delta_{min}$. This is particularly appropriate when the beginning portion of the input being quantized is expected to be of low amplitude, as in the silence segments that precede speech activity.

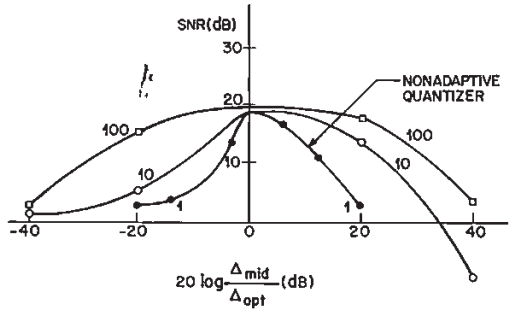
Three other parameters will be of interest in characterizing the adaptation logic:

$$\Delta_{mid} = [\Delta_{min}\Delta_{max}]^{1/2} \; ; \; \Delta_{range} = [\Delta_{max}/\Delta_{min}] \tag{4.188}$$

and, finally, an optimal step size Δ_{opt} which a nonadaptive quantizer would use if the input variance σ_x^2 were constant and known [see (4.178)]. Let us consider that we are quantizing such a stationary input, without knowledge of Δ_{opt} (or σ_x). The object now is to study quantizer performance as a function of its intrinsic range Δ_{range}, and the disposition of its central step size Δ_{mid} with respect to Δ_{opt}. Note that for a nonadaptive logic, $\Delta_{range} = 1$ and $\Delta_{mid} = \Delta$, a time-invariant value.

Results for the 4-bit quantization of a stationary Gaussian input appear in Figure 4.46. The maximum *SNR* is that of a uniform pdf-optimized quantizer (19.38 dB, as in row 4, Table 4.1). The advantage of adaptation consists in increasing the *dynamic range of the quantizer* rather than the maximum possible value of *SNR*. The dynamic range of the quantizer is the range of variation in σ_x which the quantizer can handle; equivalently, the range of Δ_{mid} which a given stationary input can tolerate, for a specified minimum signal-to-noise ratio. Note that with $\Delta_{range} = 100$, which implies a step size dynamic range of $20\log_{10}100 = 40$ dB, the dynamic range of the quantizer, as defined in the last sentence, is also about 40 dB. This is a nearly adequate match to the dynamic range that may be encountered in the speech input of a single speaker (Figure 2.6). Note also that the flattening of the performance curve in Figure 4.46 corresponds to a constant, input-independent loading factor f_l. In the case of nonadaptive quantization, Δ is constant and f_l is inversely proportional to σ_x - by definition (4.30). The asymmetrical *SNR* - versus - step size profile for the nonadaptive case (Δ_{range}=1) in Figure 4.46 should be familiar from earlier discussions (see Figure 4.7).

Figure 4.46 Measured dynamic range curves for a 4-bit adaptive quantizer for a stationary Gaussian AR(1) input with $\rho_1 = 0.5$. The numbers 1, 10 and 100 on the three curves are respective values of Δ_{range} [Jayant;©1973, AT&T].



0188

The value of 100 cited above for Δ_{range} is typical of many quantizing systems, as an order-of-magnitude number. Values which are multiples of 2, such as 128, 256, 512 or 1024 are also frequently encountered. The last value implies a dynamic range of nearly 60 dB. It is also the value used in a standardized DPCM coder design for high quality speech (Section 6.5.3). Transmission error sensitivity, however, is an increasing function of Δ_{range}.

Relative Values of the Step Size Multipliers and the Steady-State Behavior of the Adaptation Logic. More can be said about desirable multipliers than the simple constraints in (4.186). An intuitive stability criterion [Jayant, 1973] would be of the form

$$\prod_{i=1}^{2^R-1} M_i^{P_i} = 1 ; \qquad M_1^{P_1} M_2^{P_2} = 1 \quad \text{if} \quad R = 2 ;$$

$$M_1^{P_1} M_2^{P_2} M_3^{P_3} M_4^{P_4} = 1 \quad \text{if} \quad R = 3 \tag{4.189}$$

where the probability P_i represents the probability of output magnitude level i (and hence the probability of using multiplier M_i) in the *steady-state,* i.e., after the adaptive logic has essentially learned the variance σ_x^2 of a stationary input. The argument is that if a requirement such as (4.189) did not apply, the step size Δ would eventually decay to zero [if the product in (4.189) were less than 1], or blow up [if that product were greater than 1]; and these would represent an unstable adaptive logic in the presence of a stable input.

Example 4.15. 2-bit quantization of Gaussian inputs

Let us illustrate the above notion more quantitatively with the example of $R=2$ for stationary Gaussian inputs. In uniform mmse quantization of a Gaussian input with step size Δ, the probabilities of the two output magnitudes $3\Delta/2$ and $\Delta/2$ are approximately $P_2 = 0.33$ and $P_1 = 0.67$ (a result of setting $\Delta = \Delta_{opt} = 0.996\sigma_x$; see row 2 of Table 4.1 and Problem 4.37). The above probabilities are also the respective multiplier probabilities by definition, so that from (4.189),

$$R = 2: \qquad M_1^{0.67} M_2^{0.33} \sim M_1^2 M_2 = 1 \tag{4.190}$$

One example of a multiplier-pair that closely satisfies (4.190) is $[M_1=0.8; M_2=1.6]$, and a pair that clearly violates the requirement is $[M_1=0.5, M_2=2.0]$. It indeed turns out, in computer simulations with an uncorrelated Gaussian sequence, that the former multiplier-pair outperforms the latter. Corresponding steady-state signal-to-noise ratios differ by about 2 dB [Jayant, 1973]. •

Example 4.16. 3-bit quantization of Gaussian inputs

For the 3-bit quantization of i.i.d. Gaussian sequences, an equation corresponding to (4.190) results from an earlier result ($\Delta_{opt} = 0.586\sigma_x$ if $R=3$; row 3 of Table 4.1), and the resulting probabilities P_1 through P_4 (Problem 4.37):

$$R = 3: \qquad M_1^{0.47} M_2^{0.30} M_3^{0.14} M_4^{0.09} = 1 \tag{4.191}$$ •

The selection of multiplier values for $R = 3$ according to (4.191) is more difficult than for $R = 2$ [from (4.190)]. The problem can be simplified by associating more structure with multiplier sets $\{M_l\}$ and the probabilities P_l; a theoretically, as well as practically, useful way of doing this is by insisting [Mitra, 1974], [Goodman and Gersho, 1974] that multipliers be expressed in the canonical form

$$M_l = \gamma^{k_l}; \quad \gamma > 1, \quad \pm k_i = 0,1,2,... \tag{4.192}$$

so that the constraint (4.189) can be rewritten as

$$\sum_{i=1}^{2^{R-1}} P_i k_i = 0 \quad \text{(for some } \gamma > 1) \tag{4.193}$$

The problem of multiplier selection now becomes one of selecting integer sets $\{k_i\}$ that satisfy (4.193) closely. It is easy to see that with $R = 2$, there is only one simple $\{k_i\}$ pair that satisfies (4.193) for the Gaussian probabilities (0.67 and 0.33); while for the four probabilities for $R = 3$ in (4.191), there are more degrees of freedom, permitting the choice of alternative $\{k_i\}$ sets [Mitra, 1974]. The following numbers are illustrative:

$$R = 3; \quad \{k_i\} = \{-1,-1,2,5\} \quad \text{or} \quad \{-1,0,1,4\} \tag{4.194}$$

The steady-state *SNR* performance of the second multiplier set for $R = 3$ is better by about 0.5 dB [with γ=1.04 in (4.192)]. Compare this with the more significant 2 dB difference for the two multiplier-pairs mentioned earlier for $R = 2$. This difference of 2 dB relates to the fact that one of the multipliers-pairs for $R = 2$ clearly did not satisfy (4.190) or (4.193); while both the multiplier sets in (4.194) satisfy the corresponding stability requirement closely [with respective values of -0.4 and $+0.03$ for the sum in (4.193)].

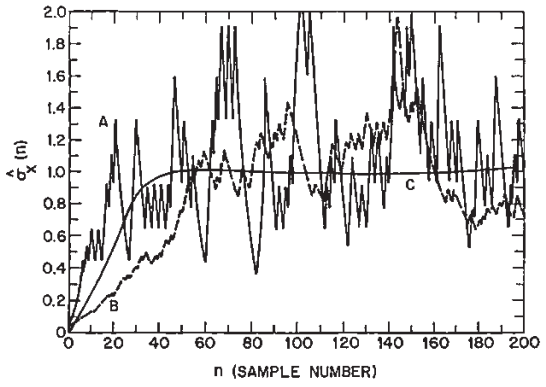
A quite striking property of all multiplier sets discussed in this subsection is that step size *increases* are always designed to be faster than step size *decreases*. The property carries over to larger values of bits ($R > 3$) [Jayant, 1973] and it is related to the fact that the Gaussian pdf leads to a monotonically decreasing set of probabilities P_l in mmse uniform quantization. Fast increases of step size, while preserving such a probability configuration, serve the related physical purpose of mitigating overload noise in relation to granularity. This is desirable from an *SNR* point of view, as noted earlier. Furthermore, fast step size increases and slow decreases are also matched to the envelope characteristics of speech waveforms (Figures 2.1 and 2.6).

We have so far concentrated on the steady-state performance of the adaptation logic, as determined by the steady state probabilities P_l and the constraints they impose on a desirable set of k_l values. The latter, in turn, specify only the relative values of the multipliers M_l, and not their absolute magnitudes. These magnitudes are determined by γ, and the choice of γ describes the dynamic response of the adaptive logic.

Absolute Values of Step Size Multipliers and the Dynamic Response of the Adaptive Logic. When step size adaptations are used for learning the variance of a stationary input, or for tracking a very slowly changing variance, there are many (not unrelated) reasons for selecting multipliers M_l that are close to unity, or equivalently, for selecting a value of γ very close to unity. An interesting observation is that for such γ, the ratio of two multipliers M_i and M_j is linearly related to the difference between integers k_l and k_j. Another property is that as γ tends to unity, the effective variance-estimator weight α in (4.181) tends to unity as well. Finally, as γ approaches 1, the variance of the estimator $\hat{\sigma}_x^2$ (i.e., the oscillation of step size in the steady state) approaches zero [Mitra, 1974]. This is illustrated by the results in Figure 4.47 which refer to 2-bit adaptive quantization of an i.i.d. Gaussian input [Goodman and Gersho, 1974]. Plots A and B show $\hat{\sigma}_x(n)$ as a function of sample number n for two pairs of multiplier values. The multiplier-pairs in these figures correspond to $\gamma = 1.20$ (plot A) and $\gamma = 1.05$ (plot B) and the k-values $\{-1,2\}$ from (4.193). The larger value of $\gamma = 1.20$ provides faster dynamic response. The lower value of $\gamma = 1.05$ provides an estimate with smaller oscillations, or a smaller variance about the mean. Adaptations with differing dynamic responses have also been illustrated earlier in Figure 4.44, for the example of a real-speech input, as a function of α.

Also shown in Figure 4.47 is the nature of step size variation that may characterize a *training-mode* quantizer where all multiplier values converge to 1.0 after the adaptation logic has been trained adequately by a stationary input signal of unknown variance (plot C). When the input of unknown variance is known to be stationary (as in the case of voiceband data waveforms), emphasis is on minimal steady-state oscillation after an appropriate training phase. Step-size multiplier functions for such a system can be expressed in the form $[M(|H(n-1)|)]^{a_n}$ with an appropriately converging $\{a_n\}$ sequence [Gersho and Goodman, 1974]. One

Figure 4.47 Variance estimate $\hat{\sigma}_x^2(n)$ versus n in the adaptive 2-bit quantization of a stationary Gaussian i.i.d. input. Plot A refers to the case of $M_1 = 0.83$, $M_2 = 1.44$, $\gamma = 1.2$; plot B to the case of $M_1 = 0.95$, $M_2 = 1.10$, $\gamma = 1.05$; and plot C to a case where γ decreases with time, and M_1 and M_2 eventually converge to 1.0 [Goodman and Gersho;©1974, IEEE].



example of such a sequence, used in Figure 4.47(c) [Gersho and Goodman, 1974] is

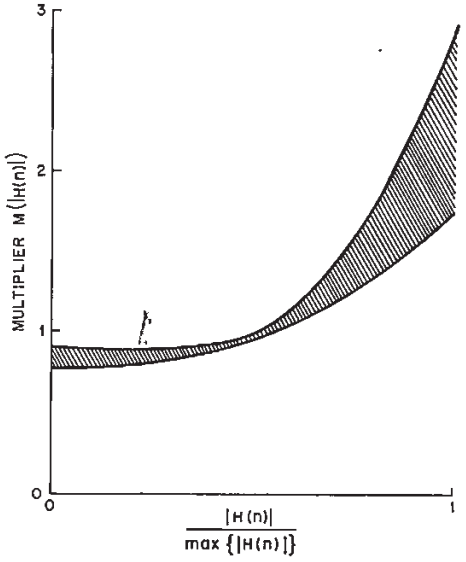
$$\{a_n\}: 1, 1/2, 1/2, 1/4, 1/4, 1/4, 1/8,... \tag{4.195}$$

A somewhat different approach to the quantization of a stationary signal of unknown variance is the *dynamic locking quantizer* described in Section 6.5.3. This system does not need prior knowledge of the fact that the input is stationary; it detects this property automatically, and freezes the step size to an appropriate fixed value.

Adaptive Quantization of Speech with a One-word Memory. Preceding sections have considered adaptive quantization using two different approaches, one exemplified by the recursive variance estimator (4.181), and the other by a set of step size multipliers (4.192). Both formulations have independent means of specifying the dynamic response of the adaptive quantizer (via α and γ respectively), as well its static properties (via the pdf-specific quantities ϕ and k_i). In fact, by equivalent settings of α_l, γ and of ϕ and k_i, very similar multiplier sets result in both approaches (Problem 4.38).

Figure 4.48 and Table 4.14 illustrate multiplier functions recommended [Jayant, 1973] for the adaptive quantization of speech waveforms in PCM (Chapter 5) and first-order prediction error waveforms in DPCM (Chapter 6). Note, in both cases, that step size increases are relatively faster than step size decreases, as noted in earlier discussion.

Figure 4.48 Nature of the step size multiplier function in adaptive quantization of speech with a one-word memory [Jayant;©1973, AT&T].



0192

Table 4.14 Step-size multipliers for AQB with a uniform quantizer and speech inputs [Jayant, 1973].

Coder	PCM			DPCM		
R	2	3	4	2	3	4
M_1	0.60	0.85	0.80	0.80	0.90	0.90
M_2	2.20	1.00	0.80	1.60	0.90	0.90
M_3		1.00	0.80		1.25	0.90
M_4		1.50	0.80		1.75	0.90
M_5			1.20			1.20
M_6			1.60			1.60
M_7			2.00			2.00
M_8			2.40			2.40

Approximations of Multiplier Functions. The number and possible resolution of multiplier values increases with the number R of quantizer bits, and approximate values are often desirable for implementation purposes. Implementations often use a *read-only-memory* (ROM) arrangement where the ROM input address is a function of the quantizer output history ($|H(n-1)|$, for example), and the ROM output directly gives step size. Variance estimations and step size multiplications discussed so far are built into such ROM designs, sometimes with practical approximations.

In variance-estimating quantizers, a useful approximation to optimal design results by constraining the speed parameter in the form

$$\alpha_l = 1 - 2^{-J} \quad ; \quad J = 1,2,... \tag{4.196}$$

The negative power of 2 is easily realizable by a shift register circuit with binary entries.

In implementations based on explicit step size multiplication, multiplier values have already been simplified in the form γ^k. A hardware implementation of a 3-bit quantizer for DPCM uses, for example, $\gamma = \sqrt[4]{2}$ and $\{k\} = -1, -1, 1, 3$ [Cummiskey, Jayant and Flanagan, 1973]. Corresponding M_l values are 0.84, 0.84, 1.19 and 1.68, and these follow the numbers in column 5 of Table 4.14. Further simplifications result by setting some of the integers k equal to zero as in the second set in (4.194). Experiments have shown that useful adaptations, with *SNR* losses often no more than a fraction of 1 dB [Noll, 1974] [Johnston and Goodman, 1978], result even with maximally simplified logics such as the ones below for $R = 3$:

$$k_1 < 0, k_4 > 0; \; k_2 = k_3 = 0 \leftrightarrow (M_1 < 1, M_4 > 1; \; M_2 = M_3 = 1) \tag{4.197}$$

$$k_1 = k_2; \; k_3 = k_4 \leftrightarrow (M_1 = M_2 < 1; \; M_3 = M_4 > 1) \tag{4.198}$$

The multiplier design (4.197) has three distinct multipliers while (4.198) has only two distinct multipliers. The two-multiplier design is the basis of an ROM

0193

implementation of adaptive quantization on a digital signal processor chip [Boddie et al., 1981]. The two-multiplier design also has very good resynchronization properties in situations where step sizes at transmitter and receiver may end up being different, for example, due to transmission errors.

The use of a log-step size concept converts step size *multiplications* to log-step size *additions*, and thus provides another approach for simple implementation. This method will be presently elaborated in (4.199b).

Although most of our discussion has assumed a uniform quantization characteristic $Q(x)$, it is to be expected that pdf-optimized $Q(x)$ functions can be used in adaptive quantization, provided that the pdf at the quantizer input is known, as in Figure 4.40(c); or can be determined from the coder input pdf, as in Figure 4.40(b). Nonuniform $Q(x)$ functions have therefore been proposed for speech coding systems with AQB [Noll, 1974] [Qureshi and Forney, 1975] [Cohn and Melsa, 1975] with noticeable *SNR* gains. It should be mentioned, however, that the perceptual gains due to a nonuniform $Q(x)$ function have usually fallen short of the objectively measured *SNR* gains in AQB systems; and the refinement of a nonuniform $Q(x)$ is of a second order of importance compared with the refinement of a time-varying step size logic.

Robust Adaptive Quantization for Noisy Channels. In Section 4.9, the performance of nonadaptive quantizers with transmission errors was shown to improve as a result of a subdued quantization strategy involving a modified $Q(x)$ characteristic with smaller step sizes. Adaptive quantizers can also be subdued to increase transmission error robustness, and we will show this with the example of a one-word memory adaptation rule.

When a step size sample $\Delta(n)$ is recovered erroneously because of a bit error affecting $y(n-1)$, the effect propagates into future step sizes $\Delta(n+l)$; $l = 1,2,...,$ because of the memory in the logic (4.185b).

An effective means of reducing this error propagation is by using a simple generalization of (4.185b). This leads to the *robust adaptive quantizer* [Goodman and Wilkinson, 1975] [Mitra and Gotz, 1978]:

$$\Delta(n) = M(|H(n-1)|)\Delta^{\beta}(n-1); \quad \beta = 1-\epsilon^2; \epsilon^2 \to 0 \qquad (4.199a)$$

The use of the *leakage factor* $\beta < 1$ helps the system to "forget" the effect of a wrong step-size $\Delta(n-1)$ (Problem 4.39). However, it also implies imperfect, or sluggish adaptations that may slightly diminish coder performance over an error-free channel (and with channels where the error probability p_e is less than the error rate for which a given β is appropriate). But the intent is to improve coder performance with a noisy channel over which the $\beta = 1$ performance can be very inadequate due to step size error propagation. The zero-error-rate performance of the sluggish logic (4.199a) can be boosted somewhat by compensatory devices such as selecting multiplier values farther away from unity (compared to a $\beta=1$ design); or by an *interleaved* adaptation logic where the decay factor β is not used for every sample n, but rather infrequently, in a periodic fashion. The latter refinement also has advantages in terms of finite-precision arithmetic in implementation [Mitra and Gotz, 1978].

An important consequence of (4.199a) is that the multiplier is no longer the ratio of $\Delta(n)$ to $\Delta(n-1)$; instead, it is that ratio modified by a factor $[\Delta(n-1)]^{\beta}$. In other words, the step-size multiplier is not independent of input variance, as implied hitherto in this chapter. In practice, multiplier values may be required to be time-invariant for simplicity. In such applications, the multipliers should at least be re-optimized in some fashion to compensate partially for the compromise of a time-invariant design (See Section 6.5.3).

In log-step size notation, (4.199a) can be rewritten

$$\log \Delta(n) = \beta \log \Delta(n-1) + \log M(|H(n-1)|) \tag{4.199b}$$

With $\beta<1$, (4.199) represents a leaky integration procedure which has been directly utilized in robust quantizer implementations. The time constant of the leaky adaptation logic can be defined as the time taken for a channel-error-induced log-step-size offset to decay to a fraction e^{-1} of its original value. This time constant can be shown to be $(f_s \log_e 1/\beta)^{-1}$ where f_s is the sampling rate (Problem 4.39). As in the case of α (4.196), β values are generally chosen in the form

$$\beta = 1-2^{-J}\ ;\ J = 1,2,... \tag{4.200}$$

for easy implementation.

The performance of the robust adaptive quantizer is further illustrated in Figures 4.49 and 4.50 and Table 4.15.

Dependence of loading factor on β. The following discussion refers to 4-bit adaptive quantization of Gaussian signals with the multiplier set [0.8, 0.8, 0.8, 0.8, 1.25, 1.25, 2.0 and 3.0]. Consider the *loading factor* f_l of (4.30), the ratio of quantizer overload point x_{ol} to input standard deviation σ_x, for $R = 4$ and $x_{ol} = 4\ \Delta_0$; Δ_0 is a mean value of step-size which, along with σ_x, is determined over a sufficiently long segment (say, 1000 of step-size samples of a stationary input). Figure 4.49 plots experimental values of f_l as a function of $\hat{\sigma}_x/\sigma_x$ for different values of β. Three classes of plots are evident: (i) *nonadaptive* quantization (β=0) where f_l is inversely proportional to σ_x by definition, so that a given dB-increase in σ_x causes an identical dB-decrease in f_l, as with the dashed line in the figure; (ii) *perfect adaptation* (β=1) where f_l is independent of σ_x. The value of f_l is 1.23, corresponding to a step size $\Delta_0 = (1.23/4)\ \sigma_x = .31\ \sigma_x$, a value that is very close to the Max-value $0.33\ \sigma_x$ (Row 4 of Table 4.1) for $R = 4$; and (iii) *robust adaptations* ($\beta<1, \beta\rightarrow1$) where f_l does decrease with σ_x, but very slowly (for example, by 4.5 dB for a 40 dB variation of σ_x, with β=63/64).

Example 4.17 Performance of robust adaptive quantizer in 3-bit PCM-AQB coding

Figure 4.50 illustrates output speech waveforms in PCM-AQB systems with $\beta = 1$ and $\beta = 63/64$, for error rates of 0 and 1%, in each case. The step-size multiplier set is [0.8, 1.0, 1.0, 1.5], as in Table 4.14. With $p_e = 0$, there is very little difference between the output waveforms [parts (b) and (c) of the figure] or between *SNR* values, for $\beta = 1$ and $\beta = 63/64$. With $p_e = 1\%$, the robust quantizer ($\beta = 63/64$) mitigates the scale changes in the output waveform due to bit errors [parts (d) and (e) of the figure], and produces substantial improvements in *SNR* and *SNRSEG*. •

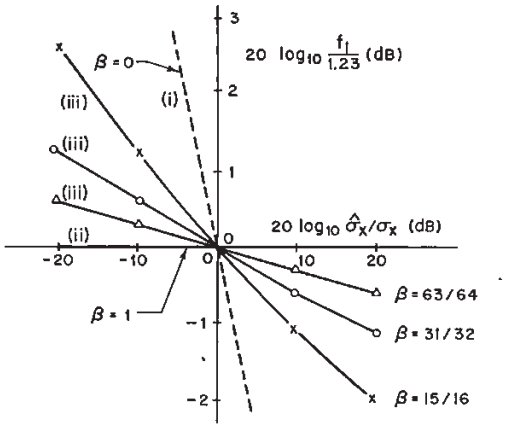
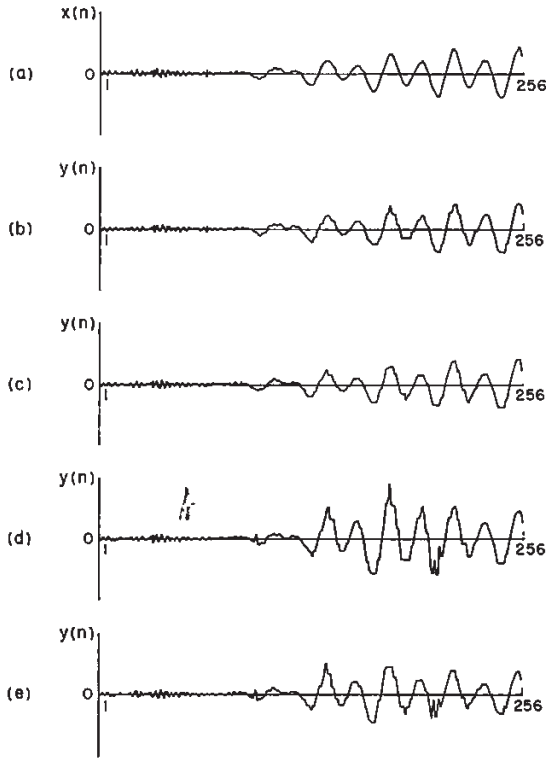


Figure 4.49 Loading factor f_l versus $\hat{\sigma}_x/\sigma_x$ in (i) nonadaptive, (ii) perfectly adaptive and (iii) robust adaptive 4-bit quantizers for a Gaussian input [Goodman and Wilkinson;©1975, IEEE].

Figure 4.50 Performance of robust adaptive quantizer in 3-bit PCM-AQB coding of speech: (a) input waveform $x(n)$, from Figure 1.4(a); and outputs $y(n)$ with (b) $\beta = 1$, $p_e = 0$; (c) $\beta = 63/64$, $p_e = 0$; (d) $\beta = 1$, $p_e = 0.01$; and (e) $\beta = 63/64$, $p_e = 0.01$. [Thomas and Noll, 1983].



Example 4.18. Performance of robust adaptive quantizer in 3-bit DPCM-AQB coding

The effect of β on decoded waveform quality is shown in Table 4.15 for the example of DPCM coding of speech (Chapter 6) with adaptive quantization using the multiplier set [0.9, 0.9, 1.25 and 1.75]. The signal-to-noise ratios *SNRT* and *SNRR* refer to values at the transmitter and receiver. These ratios were computed in a simulation with a bandlimited (female) speech input, "The boy was mute about his task". The random bit error probability in the simulation was $p_e = 0.023$. Note that $\beta = 1$ maximizes *SNRT* (= *SNRR* for $p_e = 0$), $\beta = 31/32$ maximizes *SNRR* for $p_e = 0.023$, while $\beta = 63/64$ maximizes [0.5(*SNRT* + *SNRR*)], suggesting a design that may be appropriate in general for a class of channels given by $(0 \leqslant p_e \leqslant 0.023)$.

Table 4.15 Effect of β on speech quality at transmitter (*SNRT*) and at receiver (*SNRR*); R = 3 and $p_e = 2.3\%$ (Multiplier values for step size are 0.9, 0.9, 1.25 and 1.75) [Jayant, 1975].

β	*SNRT* (dB)	*SNRR* (dB)	0.5[*SNRT*+*SNRR*] dB
1	17.9	3.7	10.8
63/64	16.2	6.9	11.6
31/32	14.3	7.1	10.7
15/16	10.7	6.7	8.7
7/8	5.4	4.4	4.9

Problems

(4.1) Consider a uniform R-bit midrise quantizer with $L = 2^R$ levels and a uniform input pdf in the range $(-x_{\max}, x_{\max})$. Let the overload point be x_{ol} as in (4.23). Show that

(a) $\sigma_x^2 = x_{\max}^2/3$

(b) $\sigma_q^2 = x_{ol}^2/3L^2$ if $x_{\max} \leqslant x_{ol}$

(c) $\sigma_q^2 = \dfrac{x_{ol}^2}{3L^2}\left[\dfrac{x_{ol}}{x_{\max}}\right]\left[1+L^2\left(\dfrac{x_{\max}}{x_{ol}}-1\right)^3\right]$ if $x_{\max} > x_{ol}$

(d) Sketch $SNR = 10\log_{10}\sigma_x^2/\sigma_q^2$ as a function of $(x_{\max}/x_{ol})$ for $L = 16$ and $L = 256$. Verify that the result is that of Figure 4.7.

(4.2) Verify analytically that with $L \to \infty$, $(x_{ol}/x_{\max})_{opt} = 1$ for a uniform pdf. Note therefore that the empirical design of Example 4.3 is indeed optimum.

(4.3) One-bit quantization of sine-waves is of interest in the analysis of deltamodulation schemes (which use two-level quantizers and are often tested with sine waves). Verify, using the arcsine pdf of Problem 2.1(a) for instantaneous amplitude x, that

$$\delta_{opt}/\sigma_x = \sqrt{8}/\pi \sim 0.9; \quad \min\{\epsilon_q^2\} = 1 - 8/\pi^2 \sim 0.19$$

Note that the *SNR* is 7.23 dB and that the approximate result of (4.33) yields a slightly higher value. Note also that the performance above is better than that for all other pdf's in Table 4.2. This has to do with the very small peak-to-rms ratio of the arcsine pdf.

0197

(4.4) (a) Consider the midtread 3-level quantizer of Figure 4.10. Consider the uniform case defined by $\Delta = 2x'$. Show that the mean square error is

$$\epsilon_q^2 = \sigma_q^2/\sigma_x^2 = 1 - 8x'\,\phi_{x'}\,(M_{x'} - x')/\sigma_x^2\,;$$

$$\phi_{x'} = P\,\{X > x'\}\,; \qquad \text{and } M_{x'} = E\,[X|X > x']$$

Hence show, using the erf function of (2.15), that for a Gaussian input, σ_q^2 is minimum at $x' \sim 0.6\sigma_x$ and $\min\{\epsilon_q^2\}) \sim 0.23$, indicating a 1.8dB again over optimum two-level quantization of a Gaussian input.

(b) Unlike in the uniform example of Problem 4.4(a), consider a nonuniform characteristic. Show that the mean square error for a Gaussian input is minimized for $\Delta = M_{x'}$ and $\min\{\epsilon_q^2\} = 1 - 2M_{x'}^2\phi_{x'}/\sigma_x^2$ which in turn is minimized by $x' = 0.61\sigma_x$, for a minimum normalized mse of 0.19, an *SNR* gain of 2.8 dB over the optimum two-level case; and a gain of 1 dB over the optimum uniform 3-level case.

(4.5) Consider a r.v. X that assumes values $\pm a$ with probability P_1 each, and a value of zero with probability P_0. For the case of $P_0 = 0.5$ and $P_1 = 0.25$,

(a) show that the *symmetric* 1-bit mmse quantizer is defined by output levels of $\pm 0.5a$, and an mse of $a^2/4$.

(b) show that the *non-symmetric* 1-bit quantizer with output levels $-a/3$ (for $x \leqslant 0$) and $+a$ (for $x > 0$) has a smaller mse, equal to $a^2/6$ [Denker, 1983].

(4.6) Use the Panter-Dite result (4.68) to show that *SNR* values for pdf-optimized 7-bit (128-level) quantizers with

(a) uniform

(b) Gaussian and

(c) Laplacian inputs

are approximately 42, 37.7 and 35.5 dB.

(4.7) (a) Sketch the m-law compression characteristic (4.70) for the specific examples of $m = 0$ and 10.

(b) Verify that for $x = 0.5\,x_{max}$, respective $|c(x)|/x_{max}$ values are 0.5 and 0.99 respectively.

(4.8) For any one-to-one mapping $y = c(x)$, the pdf of y is given, in terms of that of x, by the formula

$$p_Y(y) = \frac{p_x(x)}{\left|\frac{dc(x)}{dx}\right|}\Bigg|_{x = c^{-1}(y)}$$

(a) Show that an input with $p_x(x)$ can be transformed to a new r.v. y with a uniform pdf in the range (0, 1) by the mapping $y = \int_{-\infty}^{x} p_x(x)dx$.

(b) Let X be a Gaussian r.v. $N(0,1)$ and let $Y = -3X+2$. Show that Y is a Gaussian r.v. N (2,3).

0198

(4.9) Derive the pdf of $x' = c(x)$ where $c(x)$ is the μ-law characteristic (4.84) with $\mu = 255$, and the input pdf $p_x(x)$ is Laplacian. To simplify notation, set $x_{max} = 1$ and assume 4σ-loading ($x_{max} = 4\sigma_x$). Note that the pdf $p_{x'}(\cdot)$ is "flatter" or "more uniform" than the input pdf.

(4.10) Show that for A-law companding (4.77), the ratio of maximum to minimum step size (ratio of maximum to minimum interval density dc/dx) is equal to A.

(4.11) Consider the μ-law characteristic (4.84). For large L,

(a) show that the characteristic has a step size ratio

$$\max\{\Delta\}/\min\{\Delta\} = 1+\mu \approx \mu$$

(b) show that the ratio of sample value x to its corresponding step size is

$$|x|/\Delta = \begin{cases} c & \text{if } x \textit{ is large} \\ & ; \; c = L/[2 \log_e (1+\mu)] \\ c\mu|x|/x_{max} & \text{if } x \textit{ is small} \end{cases}$$

(4.12) Consider the variable rate coding of a 3-level quantizer with probabilities P_o, P_i and P_o, where i and o refer to inner and outer levels.

(a) Show that if output codewords have lengths n_o, n_i and n_o, the average codeword length $\bar{n} = n_o - P_i\,(n_o - n_i)$.

(b) Hence, show that if $n_o = 2$, $n_i = 1$, $\bar{n} \leqslant 1.5$ bits/symbol as long as $P_i \geqslant 0.5$.

(4.13) (a) Consider the output entropy result (4.103). For the special case of no companding [$c(x) = x$], a uniform pdf, and zero overload ($f_l = \sqrt{3}$) show that the output entropy reduces to $\log_2 L$.

(b) Consider a uniform quantizer and a Laplacian input. Compute and plot output entropy and mse σ_q^2 versus step size Δ for $L = 4$.

(4.14) Use the results of Section 4.4.2 to show that the output entropy of pdf-optimized quantizers is [Wood, 1969]

$$H_Q|_{pdf-opt} = \frac{2}{3} h(X) - \frac{1}{3} \log_2 \frac{12 \min\{\sigma_q^2\}}{L} \quad \text{for large } L$$

Hence show (as noted in Figure 4.19) that

$$\Delta R|_{pdf-opt} = \log_2 L - H_Q|_{pdf-opt} = (2/3)\max\{\Delta R\}$$

(4.15) Consider a uniform 9-level midtread quantizer with step size 2.0 σ_x (Example 4.12) and a memoryless zero-mean gamma input.

(a) Determine the probabilities of the zero output level y_5 and of its neighboring levels y_4 and y_6.

(b) Determine the probabilities of sequences of length 2, 3 and 4 involving all possible permutations of y_4, y_5 and y_6.

0199

(c) Estimate the entropy of the quantizer output. Compare with the result of Example 4.12.

(4.16) Compute numerically the pdf's of quantization errors for Gaussian inputs and optimum uniform quantization with $L = 2$ and 4 levels. Compare results with those of Figure 4.22.

(4.17) Consider a one-bit quantizer with a Gaussian AR(1) input.

(a) Show that the cross-correlation between input and quantization error is

$$\rho_{xq}(0) = (\sigma_x^2 - \Delta E[|X|])/\sigma_x(\sigma_x^2 + \Delta^2 - 2\Delta E[|X|])^{1/2}.$$

For $\Delta = \Delta_{opt}$, the mmse value of Section 4.3, note that $\rho_{xq}(0) \sim 0.6$, as in Figure 4.24(c). Note also that $\rho_{xq}(0) = 0$ at $\Delta \sim 1.25\,\sigma_x$.

(b) Derive the autocorrelation $\rho_{qq}(1)$ as a function of Δ; show that for $\rho_{xx}(1) = 0.85$, $\rho_{qq}(1)$ equals 0.85 at $\Delta = 0$, 0.65 at $\Delta = \infty$, and a minimum value of 0.45 at $\Delta = 1.25\,\sigma_x$.

(4.18) Consider the gain-and-noise quantization model of Figure 4.25(b) with $\alpha_g = 1 - \min\{\epsilon_q^2\}$ (4.122). Compute α_g values for

(a) $L=2$ and a Gaussian input, as well as for

(b) $L=2,4$, and 8 and a uniform pdf input.

(4.19) Consider the periodic 4-step dither sequences in (4.135) and Figure 4.32. Assume a constant input $x(n)=0.5$ and a quantizer defined by the transfer function: (output = 1 or 0 depending on whether input is greater than or less than 0.5). Show that the following are the respective values of $\rho_{qq}(1)$ for the dither sequences in Figures 4.32(a) and (b):

(a) 0

(b) -1

(4.20) Show that the (4×4) dither matrix obtained by using (4.136) in (4.137) is that in (4.138).

(4.21) Let p_e be the bit error probability in a BSC, and let R be the number of bits in a codeword that represents one of 2^R levels at a quantizer output. Show that

(a) the probability of no error in the reception of a codeword is $P_0 = (1-p_e)^R$,

(b) the probability of at least one error in a codeword is $P_{>0} = 1-(1-p_e)^R$

(c) the probability that exactly E bits are in error is $P_E = \binom{R}{E} p_e^E (1-p_e)^{R-E}$,

(d) the probability that any *given* bit is in error is $P_{1g} = p_e(1-p_e)^{R-1}$

(4.22) Consider a BSC with $p_e=0.05$. Show that the probability of no error in the reception of an R-bit codeword is 0.8145 for $R = 4$ and 0.6634 for $R = 8$.

(4.23) Verify that the results of (4.175) reduce to those of the Max-quantizer if $p_e = 0$.

(4.24) Consider the quantizer reoptimization results (4.175). For a totally noisy channel

0200

with $p_e = 0.5$ and $P_{kj} = 2^{-R}$ (all k,j), show that $x_{k\bullet\bullet} = y_{k\bullet\bullet} = 0$, so that the minimum reconstruction error variance is $\min\{\sigma_r^2\} = \sigma_x^2$.

(4.25) Based on the data of Table 4.11, show that the use of $\delta_\bullet$ instead of $\delta_{\bullet\bullet}$ will result in an increase of σ_r^2 by a factor

$$[\sigma_x^2-\delta_\bullet^2(1-4p_e)/\sigma_x^2-\delta_\bullet^2(1-2p_e)^2] \sim 1+12p_e^2 \text{ for } p_e << 1 \text{ with a uniform pdf}$$

(4.26) Show that the error probability threshold p_{eqc} where channel error variance σ_c^2 equals the minimum quantization error variance $\sigma_{q\bullet}^2$ is given by

$$p_{eqc} = (1/\xi_\bullet - 1/4)$$

with values of 1/12 and 1/2 for uniform and gamma pdf's (Table 4.11).

(4.27) Sketch the transmission error performance (4.174) for the Max-quantization of a uniform pdf input as a function of p_e and L.

(a) Show that this is the result of Figure 4.33.

(b) Show that the critical error rate for which $\sigma_c^2 = \sigma_{q\bullet}^2$ is given by $p_{eqc} = [1/4(L^2-1)]$.

(4.28) Show that the quantizer redesign (4.166) by a factor $(1-2p_e)$ causes a decrease in channel coefficient ξ by the factor $(1-4p_e)$ if $p_e << 1$:

$$\xi_{\bullet\bullet} = \xi_\bullet(1-4p_e)$$

(4.29) For a uniform pdf and an NBC code, show that when $i < R$ most significant bits are error-protected, the resulting channel coefficient is $\xi^{(i)} = 4(2^{-2i}-2^{-2R})$ [Zelinski, 1979].

(4.30) Define the Hamming distance between two codewords as the number of positions in which they differ. For $R=3$ and an FBC code show that the Hamming distance matrix is

$$\begin{bmatrix} 0 & 1 & 1 & 2 & 3 & 2 & 2 & 1 \\ 1 & 0 & 2 & 1 & 2 & 3 & 1 & 2 \\ 1 & 2 & 0 & 1 & 2 & 1 & 3 & 2 \\ 2 & 1 & 1 & 0 & 1 & 2 & 2 & 3 \\ 3 & 2 & 2 & 1 & 0 & 1 & 1 & 2 \\ 2 & 3 & 1 & 2 & 1 & 0 & 2 & 1 \\ 2 & 1 & 3 & 2 & 1 & 2 & 0 & 1 \\ 1 & 2 & 2 & 3 & 2 & 1 & 1 & 0 \end{bmatrix}$$

(4.31) Consider a 2-bit uniform quantizer and symmetrical input pdf models where relative quantizer slot occupancies are simple integers. For example, in an "n:1" pdf, the probability of the inner level is n times that of the outer levels. Assume an input variance of $\sigma_x^2 = 1$, a single bit error per word, and an equal probability of the error being in any of the two positions in the codeword. Show that normalized error variance $\sigma_c^2/\Delta^2 p_e$ for different binary codes are the values listed in the following table. Assuming a mmse step size of $\Delta_{opt} = 0.996$, which is

0201

optimal for a Gaussian input (Table 4.1) and $R = 2$, and assuming that $p_e = 5\%$, show that the normalized σ_c^2 value of 2.5 for the NBC and TCC codes in the table corresponds to a signal-to-channel-noise ratio of 9 dB. Show that the ratio increases to 16 dB for all the pdf's in the table if the most significant bit (MSB) is totally error-protected. Note that the above noise reductions are only upper bounds because error protection is never ideal with realistic constraints on the side information required for it.

pdf	$\sigma_c^2/\Delta^2 p_e$	
	FBC, GC	NBC, TCC
1:1	3.0	2.5
2:1	2.33	2.5
3:1	2.0	2.5

(4.32) Consider an R-bit AQF system for speech with explicit transmission of step size information once every N samples, using an r-bit codeword for step size.

(a) Derive the fractional increase of transmission rate as a function of r, R and N.

(b) Assume that 16 levels of $\hat{\sigma}_x^2$ are allowed and that the 4-bit variance information is redundantly transmitted five times in anticipation of a noisy channel, which introduces random bit errors with a probability of 0.025. Assume that the variance estimate is updated once every 256 samples.

(i) Show that the overall transmission rate is increased by (7.8/R)% due to variance-transmission.

(ii) Show that variance-detection based on a majority-count-in-five logic is going to be erroneous (more than 2 errors in the 5 transmissions of any of the 4 variance bits) with a probability of about 6 in 10000.

(4.33) Show that, for an i.i.d. input, (4.180) is equivalent to (4.179b), with respect to the first and second moments of $\hat{\sigma}_x^2(n)$, if, in (4.179b), the learning period $N = (1+\alpha_l)/(1-\alpha_l)$; $N \gg 1$.

(4.34) Show that the variance estimate (4.180) can be written in the recursive form (4.181a).

(4.35) Show that the time constant τ of the adaptation logic (4.181) is $2(f_s \log_e 1/\alpha)^{-1}$. Hence show that with $f_s = 8$ kHz, τ values of 1/4, 2 and 10 ms can be realized with $\alpha = 0.370$, 0.883, and 0.975, respectively.

(4.36) Consider a weighted average AQB step size rule

$$\Delta_n = \phi_3 E[|Y(n)|] = \theta_2(1-\alpha_l) \sum_{i=1}^{\infty} \alpha_l^{i-1} |y(n-i)|$$

where θ_2 is defined in (4.184), and $0 < \alpha_l < 1$. A good value for speech quantization is $\alpha_l = 0.75$ [Castellino et al., 1974]. Show that Δ_n can be written in a recursive form involving only $E[|Y(n-1)|]$ and $|y(n-1)|$.

0202

(4.37) Consider the mmse uniform midrise quantizer for Gaussian inputs with $R=2$ and 3. Using the results of Table 4.1, and the erfc function in (2.15), show that the probabilities of output magnitudes are

(a) 0.67 and 0.33 ($R=2$) and

(b) 0.47, 0.30, 0.14, and 0.09 ($R=3$).

(4.38) (a) Derive step size multipliers for an adaptive 3-bit quantizer assuming a Gaussian pdf ($\phi=0.586$ for $R=3$) and a value of $\alpha = 0.5$ in (4.181c).

(b) Show that these multipliers correspond closely to those obtained from (4.192) with $\gamma = 1.12$, and $\{k\}$ values from the first set in (4.194), and that both multiplier sets are approximated by $\{M_i\} = 0.9, 0.9, 1.25$ and 1.75. Note that the $H(n-1)\Delta(n-1)/2$ term leading to (4.181c) ignores overload effects; and therefore leads to slightly smaller values for multiplier M_4.

(4.39) Assume that in AQB-type adaptive quantization, the effect of a bit error on $y(n-1)$ is to change the multiplier value from M_t at the transmitter to M_r at the receiver. Assuming no further channel errors up to sample time $n + l$, show that the ratio of received and transmitted versions of $\Delta(n+l) = \Delta_{n+l}$ is

$$(M_r/M_t) \quad \text{if} \quad \beta = 1\,; \quad (M_r/M_t)^{\beta^l} \quad \text{if} \quad \beta < 1$$

Note the log-step size offset is $\beta^l \log_e (M_r/M_t)$. For this to decay to a fraction $(1/e)$ of the original value $\log_e (M_r/M_t)$ at sample n, show that

$$l_{(1/e)} = \frac{1}{f_s \log_e \beta^{-1}} \text{ ms}$$

where f_s is the sampling frequency in kHz. This can be interpreted as the time constant of a leaky integration logic for log step size. Show that if $f_s = 8$ kHz, β values of 31/32 and 63/64 provide time constants of approximately 4 and 8 ms.

(4.40) Consider a robust adaptive quantizer (4.199) for $R=2$ and NBC code. Assume that the input is described by $x(n)=0$ $(n<5)$, $x(n) = 20(4 \leqslant n \leqslant 13)$, $x(n) = 0(13<n)$ and that at $n=0$, $\Delta(0) = \Delta_{\min} = 2$ and output codeword $\{b(0)\} = 00$. Sketch $x(n)$, $y(n)$ and $\Delta(n)$ for (i) $\beta = 1$ and (ii) $\beta = 7/8$.

References

E. Abhaya and G. L. Wise, "Some Notes on Optimal Quantization," Proc. Int. Conf. on Communications, pp. 30.7.1-30.7.5, Denver, June 1981.

V. L. Algazi, "Useful Approximations to Optimum Quantizing," IEEE Trans. on Communications, pp. 297-301, June 1966.

T. P. Barnwell, A. M. Bush, J. B. O'Neal and R. W. Stroh, "Adaptive Differential PCM Speech Transmission," Report No. RADC-TR-74-177, Rome Air Development Center, July 1974.

B. E. Bayer, "An Optimum Method for Two-Level Rendition of Continuous-Tone Pictures," Proc. Int. Conf. on Communications, pp. 26-11 to 26-15, Seattle, June 1973.

W. R. Bennett, "Spectra of Quantized Signals," Bell System Tech. J., pp. 446-472, July 1948.

T. Berger, "Optimum Quantizers and Permutation Codes," IEEE Trans. on Information Theory, pp. 759-765, November 1972.

T. Berger, "Minimum Entropy Quantizers and Permutation Codes," IEEE Trans. on Information Theory, pp. 149-157, March 1982.

J. R. Boddie, J. D. Johnston, C. A. McGonegal, J. W. Upton, D. A. Berkley, R. E. Crochiere and J. L. Flanagan, "Digital Signal Processor: Adaptive Differential Pulse Code Modulation," Bell System Tech. J., pp. 1547-1561, September 1981.

H. Brehm and K. Trottler, "Vector Quantization of Spherically Invariant Random Processes," Proc. European Signal Processing Conference, pp. 383-386, 1983.

P. Castellino, G. Modena, L. Nebbia and C. Scagliola, "Bit Rate Reduction by Automatic Adaptation of Quantizer Step Size in DPCM Systems," Proc. Int. Zurich Seminar on Digital Communications, pp. B6.1-B6.6, March 1974.

K. W. Cattermole, *Principles of Pulse Code Modulation.* Iliffe Books, London, 1969.

M. Chen and L. F. Turner, "Zero Crossing Preservation in the Low-Bit Rate Pseudorandomly Dithered Quantization of Speech Signals," Technical Report, Imperial College of Science and Technology, London, 1973.

F. Cohen, "A Switched Quantizer for Non-linear Coding of Video Signals," Nachrichtentech. Z., pp. 554-559, December 1972.

F. Cohen, "A Switched Quantizer for Markov Sources Applied to Speech Signals," Nachrichtentech. Z., pp. 520-522, November 1973.

D. L. Cohn and J. L. Melsa, "The Residual Encoder: An Improved ADPCM System for Speech Digitization," IEEE Trans. on Communications, September 1975.

A. Croisier, "Progress in PCM and DM: Block Companded Coding of Speech Signals," Int. Zurich Seminar on Digital Commun., pp. B1.1-B1.4, March 1974.

M. G. Croll, M. E. B. Moffat and D. W. Osborne, "Nearly Instantaneous Compander for Transmitting Six Sound Program Signals in a 2.048 Mb/s Multiplex," Electronics Letters, July 1973.

P. Cummiskey, N. S. Jayant and J. L. Flanagan, "Adaptive Quantization in Differential PCM Coding of Speech," Bell System Tech. J., pp. 1105-1118, September 1973.

T. Denker, private communication, 1983.

D. L. Duttweiler and D. G. Messerschmitt, "Nearly Instantaneous Companding for Non-Uniformly Quantized PCM," IEEE Trans. on Communications, pp. 864-873, August 1976.

D. J. Esteban, J. E. Menez and J. Temime, "Optimum One-Bit Quantizing," Proc. 91st Meeting of Acoust. Soc. Am., April 1976.

P. E. Fleischer, "Sufficient Conditions for Achieving Minimum Distortion in a Quantizer," 1964 IEEE Int. Conv. Rec., pt. 1, pp. 104-111, 1964.

G. G. Furman, "Improving the Quantization of Random Signals by Dithering," Rand Corporation Memo RM-3504-PR, May 1963.

A. Gersho, "Principles of Quantization," IEEE Transactions on Circuits and Systems, pp. 427-436, 1978.

A. Gersho, "Asymptotically Optimal Block Quantization," IEEE Trans. on Information Theory, pp. 373-380, July 1979.

A. Gersho and D. J. Goodman, "A Training Mode Adaptive Quantizer," IEEE Trans. on Information Theory, pp. 746-749, November 1974.

H. Gish and J. N. Pierce, "Asymptotically Efficient Quantizing," IEEE Trans. on Information Theory, pp. 676-683, September 1968.

T. J. Goblick and J. L. Holsinger, "Analog Source Digitization: A Comparison of Theory and Practice," IEEE Trans. on Information Theory, pp. 323-326, April 1967.

L. S. Golding and D. M. Schultheiss, "Study of an Adaptive Quantizer, Proc. IEEE, pp. 293-297, March 1967.

W. M. Goodall, "Television by Pulse Code Modulation," Bell System Tech. J., pp. 33-49, January 1951.

D. J. Goodman and A. Gersho, "Theory of an Adaptive Quantizer," IEEE Trans. on Communications, pp. 1037-1045, August 1974.

D. J. Goodman and R. M. Wilkinson, "A Robust Adaptive Quantizer," IEEE Trans. on Communications, pp. 1362-1365, November 1975.

A. Habibi and B. H. Batson, "Potential Digitization/Compression Techniques for Shuttle Video," IEEE Trans. on Communications, pp. 1671-1681, November 1978.

H. Holzwarth, "Pulse Code Modulation und ihre Verzerrungen bei logarithmischer Amplitudenquantelung," Archiv der elektrischen Übertragung, pp. 277-285, January 1949.

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

R. C. Jaffe, "Causal and Statistical Analysis of Dithered Systems Containing Three-Quantizers," S.M. thesis, Dept. Elec. Eng., MIT, August 1959.

J. F. Jarvis, C. N. Judice and W. H. Ninke, "A Survey of Techniques for the Display of Continuous Tone Pictures on Bilevel Displays," Computer Graphics and Image Processing, pp. 13-40, 1976.

N. S. Jayant, "Adaptive Quantization with a One Word Memory," Bell System Tech. J., pp. 1119-1144, September 1973.

N. S. Jayant, "Step-Size Transmitting Differential Coders for Mobile Telephony," Bell System Tech. J., pp. 1557-1581, November 1975.

N. S. Jayant and L. R. Rabiner, "The Application of Dither to the Quantization of Speech Signals," Bell System Tech. J., pp. 1293-1304, July-August 1972.

F. Jelinek, "Buffer Overflow in Variable-Length Coding of Fixed Rate Sources," IEEE Trans. on Information Theory, pp. 490-501, May 1968.

J. D. Johnston and D. J. Goodman, "Multipurpose Hardware for Digital Coding of Audio Signals," IEEE Trans. on Communications, pp. 1785-1788, November 1978.

W. Koerber, "Spectral Analysis of Nonadaptive and Adaptive Coding Schemes" (in German), Masters thesis, University of Bremen, 1979.

A. J. Kurtenbach and P. A. Wintz, "Quantizing for Noisy Channels," IEEE Trans. on Communications, pp. 291-302, April 1969.

J. O. Limb, "Design of Dither Waveforms for Quantized Visual Signals," Bell System Tech. J., pp. 2555-2582, September 1969.

J. O. Limb and F. W. Mounts, "Digital Differential Quantizer for Television," Bell System Tech. J, pp. 2583-2599, September 1969.

B. Lippel and M. Kurland, "The Effect of Dither on Luminance Quantization of Pictures," IEEE Trans. on Communications, pp. 879-888, December 1971.

S. P. Lloyd, "Least Squares Quantization in PCM," Institute of Mathematical Statistics Meeting, Atlantic City, N.J., September 1957; also IEEE Trans. on Information Theory, pp. 129-136, March 1982.

W. Mauersberger, "Experimental Results on the Performance of Mismatched Quantizers," IEEE Trans. on Information Theory, pp. 381-386, July 1979.

J. Max, "Quantizing for Minimum Distortion," IRE Trans. on Information Theory, pp. 7-12, March 1960.

G. A. Miller and P. E. Nicely, "Analysis of Perceptual Confusions among Some English Consonants," J. Acoust. Soc. Am., pp. 338-352, March 1955.

D. Mitra, "Mathematical Analysis of an Adaptive Quantizer," Bell System Tech. J., pp. 867-898, May-June 1974.

D. Mitra and B. Gotz, "An Adaptive PCM System Designed for Noisy Channels and Digital Implementations," Bell System Tech. J., pp. 2727-2763, September 1978.

K. Nitadori, "Statistical Analysis of DPCM," Electron. Commun. (Japan), February 1965.

P. Noll, "Adaptive Quantizing in Speech Coding Systems," Int. Zurich Seminar on Digital Communications, pp. B3.1-B3.6, March 1974.

P. Noll, "A Comparative Study of Various Quantization Schemes for Speech Encoding," Bell System Tech. J., pp. 1597-1614, November 1975.

P. Noll, "Effects of Channel Errors on the Signal-to-Noise Performance of Speech-Encoding Systems," Bell System Tech. J., pp. 1615-1636, November 1975.

P. Noll, unpublished work, Bell Laboratories, 1976.

P. Noll and R. Zelinski, "A Contribution to the Quantization of Memoryless Model Sources" (in German), Technical Report, Heinrich Hertz Institut, Berlin, 1974.

P. Noll and R. Zelinski, "Effects of Quantizer Mismatch," Archiv für Elektronik und Übertragungstechnik, Bd. 30, pp. 373-374, 1976.

P. Noll and R. Zelinski, "Bounds on Quantizer Performance in the Low Bit Rate Region," IEEE Trans. on Communications, pp. 300-305, February 1978.

P. Noll and R. Zelinski, "Comments on Quantizing Characteristics for Signals Having Laplacian Amplitude Probability Density Function," IEEE Trans. on Communications, pp. 1259-1260, August 1979.

M. D. Paez and T. H. Glisson, "Minimum Mean Squared Error Quantization in Speech PCM and DPCM Systems," IEEE Trans. on Communications, pp. 225-230, April 1972.

P. F. Panter and W. Dite, "Quantization Distortion in Pulse Count Modulation with Nonuniform Spacing of Levels," Proc. IRE, pp. 44-48, January 1951.

S. U. H. Qureshi and G. D. Forney, Jr., "A 9.6/16 kb/s Speech Digitizer," Proc. Int. Conf. Commun., pp. 30-31 to 30-36, San Francisco, June 1975.

L. R. Rabiner and J. A. Johnson, "Perceptual Evaluation of the Effects of Dither on Low Bit Rate PCM Systems," Bell System Tech. J, pp. 1487-1494, September 1972.

D. L. Richards, *Telecommunication by Speech; The Transmission Performance of Telephone Networks,* Butterworth, London, 1973.

L. G. Roberts, "Picture Coding Using Pseudo-random Noise," IRE Trans. on Information Theory, pp. 145-154, February 1962.

G. M. Roe, "Quantizing for Minimum Distortion," IEEE Trans. on Information Theory, pp. 384-385, 1964.

C. B. Rubinstein and J. O. Limb, "On the Design of Quantizers for DPCM Coders: The Influence of the Subjective Testing Methodology," IEEE Trans. on Communications, pp. 565-573, May 1978.

N. Rydbeck and C. E. W. Sundberg, "Techniques for Introducing Error-Correcting Codes into TDMA Satellite Communication Systems," Ericsson Technics, pp. 217-246, 1975.

N. Rydbeck and C. E. W. Sundberg, "Analysis of Digital Errors in Nonlinear PCM Systems," IEEE Trans. on Communications, pp. 59-65, January 1976.

W. Schlink, "A Redundancy Reducing PCM System for Speech Signals," Proc. 1972 Zurich Seminar on Communications, pp. F4.1-F4.3, March 1972.

L. Schuchman, "Dither Signals and Their Effect on Quantization Noise," IEEE Trans. on Communication Technology, pp. 162-164, December 1964.

B. Smith, "Instantaneous Companding of Quantized Signals," Bell System Tech. J., pp. 653-709, May 1957.

A. B. Sripad and D. L. Snyder, "A Necessary and Sufficient Condition for Quantization Errors to be Uniform and White," IEEE Trans. on Acoustics, Speech and Signal Processing, pp. 442-449, October 1977.

R. Steele, *Delta Modulation Systems*, John Wiley, New York, 1975.

R. W. Stroh, "Differential PCM with Adaptive Quantization for Voice Communications," Proc. Int. Conf. Communications, pp. 130.1-130.5, June 1974.

C. E. Sundberg and N. Rydbeck, "Pulse Code Modulation with Error-Correcting Codes for TDMA Satellite Communication Systems," Ericsson Technics, pp. 3-56, 1976.

T. Thomas and P. Noll, Technical University of Berlin, unpublished work, 1983.

J. E. Thompson and J. J. Sparkes, "A Pseudo-random Quantizer for Television Signals, Proc. IEEE, pp. 353-355, March 1967.

R. E. Totty and G. C. Clark, "Reconstruction Error in Waveform Transmission," IEEE Trans. on Information Theory (Correspondence), pp. 336-338, April 1967.

A. J. Viterbi, *Principles of Coherent Communications*, McGraw-Hill, New York, 1966.

C. Volmary and P. Noll, Technical University of Berlin, unpublished work, 1983.

B. Widrow, "A Study of Rough Amplitude Quantization by Means of Nyquist Sampling Theory," IRE Trans. on Circuit Theory, pp. 266-276, December 1956.

R. Wood, "On Optimum Quantization," IEEE Trans. on Information Theory, pp. 248-252, March 1969.

R. G. Wood and L. F. Turner, "Pseudorandomly Dithered Quantization of Speech Samples in PCM Transmission System," Proc. IEEE, pp. 512-518, May 1972.

R. Zelinski and P. Noll, "Investigations on Quantization of Memoryless Gaussian Sources" (in German), Technical Report No. 159, Heinrich Hertz Institut, Berlin, 1972.

R. Zelinski, "Effects of Transmission Errors on the Mean-squared Performance of Transform Coding

11

Sub-Band Coding

11.1 Introduction

In the class of *time domain* coding algorithms (Chapters 5 to 10), the input waveform is treated as a single full-band signal; and in predictive coders, redundancy is removed prior to encoding by prediction and inverse filtering. The main differences in the various algorithms are determined by the degree of prediction (Chapters 6 to 8) that is employed, and by whether schemes are adaptive or not. In delayed decision coding (Chapter 9), input structure is exploited by means of a multipath search.

In Chapters 11 and 12 another class of encoding algorithms will be discussed in which the approach is to divide the input signal into a number of separate frequency components, and to encode each of these components separately. This division into frequency components removes the redundancy in the input and provides a set of uncorrelated inputs to the channel. Recall that the action of a DPCM coder is also similar, if not identical. The encoder in that case, when fed by a redundant signal, outputs a sequence of prediction error components that tend to be uncorrelated. The *frequency domain* coding techniques have the advantage that the number of bits used to encode each frequency component can be variable, so that the encoding accuracy is always placed where it is needed in the frequency domain. In fact, bands with little or no energy may not be encoded at all. Variable bit allocation can in principle provide arbitrary forms of noise shaping, a feature that was realized to some extent by noise feedback in the time-domain methods of Chapter 7.

0208

As in the case of time domain techniques, a large variety of frequency domain algorithms, from simple to complex, are available and the main differences are usually determined by the way in which source statistics are modeled, and the degree to which source redundancy is exploited, in the technique. We will begin by describing one technique of lower complexity called *Sub-Band Coding* (SBC) and then proceed to one of higher complexity called *Transform Coding* (TC) (Chapter 12). In the notation of Chapter 1, SBC with fixed bit allocation will be a *medium-complexity* coder and TC with variable bit allocation will be a *high-complexity* coder.

Unless otherwise mentioned, focus in this SBC chapter will be on speech waveforms. In the sub-band coder the speech band is divided into typically four or more sub-bands by a bank of bandpass filters. Each sub-band is, in effect, lowpass translated to zero frequency by a modulation process equivalent to single-side-band amplitude modulation. It is then sampled (or resampled) at its Nyquist rate (twice the width of the band) and digitally encoded with a PCM or DPCM encoder [Crochiere, Webber and Flanagan, 1976] [Esteban and Galand, 1978]. In this process, each sub-band can be encoded according to perceptual criteria that are specific to that band. On reconstruction, the sub-band signals are decoded and modulated back to their original locations. They are then summed to give a close replica of the original speech signal.

Encoding in sub-bands offers several advantages. By appropriately allocating the bits in different bands, the number of quantizer levels and hence reconstruction error variance can be separately controlled in each band, and the shape of the overall reconstruction error spectrum can be controlled as a function of frequency. In the lower frequency bands, where pitch and formant structure must be accurately preserved, a larger number of bits/sample can be used; whereas in upper frequency bands, where fricative and noise-like sounds occur in speech, fewer bits/sample can be used. Further, quantization noise can be contained within bands to prevent masking of a low-level input in one frequency range by quantizing noise in another frequency range. Section 11.2 gives a quantitative demonstration of objective (*SNR*) gains due to sub-band coding.

The most complex part of the coder is the filter bank [Bellanger, Bonnerot and Coudreuse, 1976] [Esteban and Galand, 1977]. With newer filter technologies such as CCD filters and digital filters, this complexity is rapidly being reduced. Also the design technique of *quadrature-mirror filters* (Section 11.4) affords distinct advantages in digital implementation of this coder.

Figure 11.1 illustrates a basic block diagram of the sub-band coder. The coder consists of a bank of M bandpass filters, followed by sub-band encoders which typically are PCM-AQB coders, and a multiplexer. The receiver has the inverse stages of demultiplexing, decoding and bandpass filtering prior to sub-band addition. Unlike the spectrum channel vocoder for synthetic speech [Flanagan et al., 1979] [Rabiner and Schafer, 1978] where the object of the filter-bank is only to preserve information about short-time energy as a function of frequency, the sub-band coder in Figure 11.1 transmits individual time waveforms $x_k(t)$; $k = 1,2,..., M$ and the receiver adds decoder versions $y_k(t)$ phase-synchronously to obtain $y(t)$. The sub-band coder is therefore a waveform-preserving coder.

0209

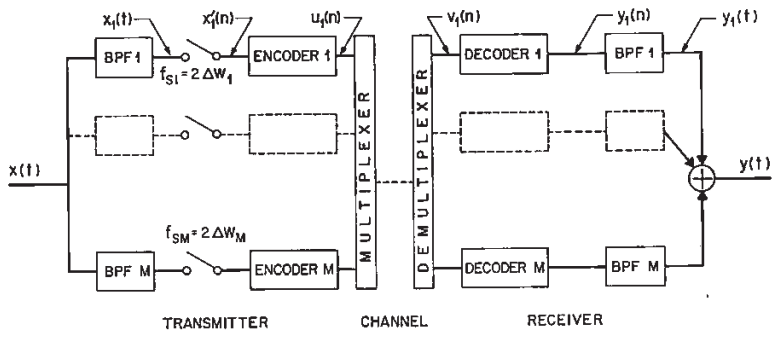


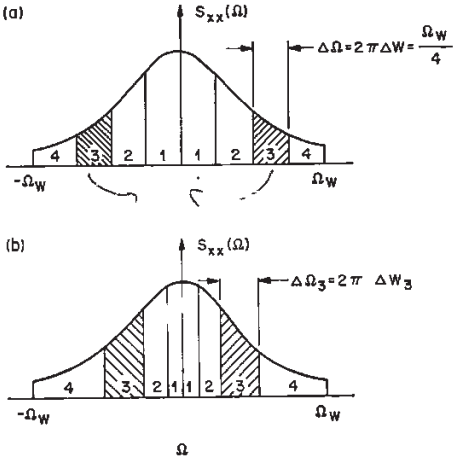
Figure 11.1 Block diagram of sub-band coding (SBC).

In Figure 11.1, sub-band width ΔW_k was a function of sub-band number k, implying *variable-width* sub-bands. The special case of *equal-width* sub-bands is important for implementation (Section 11.4) as well as analytical tractability (Section 11.2). Both types of arrangements will be considered for the sub-band coding of speech (Section 11.5). Figure 11.2 illustrates the two classes of arrangements for the example of $M = 4$. Shaded regions define sub-band number $k = 3$.

In the case of *equal-width* sub-bands

$$\Delta W_k = \Delta W = W/M \; ; \quad k = 1,2,...,M$$
$$\Delta\Omega = 2\pi\Delta W = \Omega_W/M = 2\pi W/M \tag{11.1}$$

Figure 11.2 Division of input spectrum into $M = 4$ sub-bands of (a) constant and (b) variable width.



0210

where W and Ω_W represent the total input bandwidth in Hz and radians/second, respectively. In the case of unequal sub-bands, they are typically made wider as k increases:

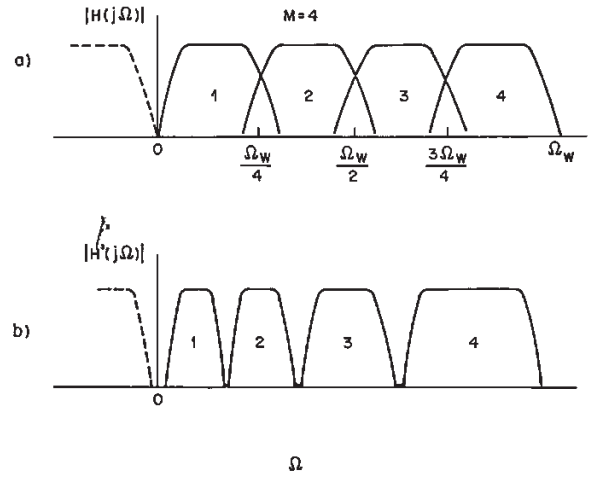
$$\Delta W_{k+1} > \Delta W_k \quad ; \quad k = 1,2,...,M-1 \tag{11.2}$$

The design in (11.2) suggests that the lower frequencies in the speech signal are more carefully isolated or observed than the higher frequencies. This provides a qualitative match to the long-time speech psd [Figure 2.9(a)] and the articulation-index function (Appendix E), both of which are lowpass functions that decrease as frequency increases. The above match is, however, not very critical in the presence of variable bit allocation, which offers the possibility of digitizing sub-bands with varying fidelity (Section 11.3). Indeed, an important filter-bank design (the quadrature-mirror filter bank, Section 11.4) has the defining characteristic that the input psd is split into equal sub-bands.

Figure 11.3 sketches filter-bank amplitude responses that may be appropriate to realize the band-splitting operations shown in Figure 11.2. The observations to be made with Figure 11.3 will also hold for the digital filter banks of Section 11.3, with $H(j\Omega)$ and Ω_W replaced by $H(e^{j\omega})$ and π, respectively. An important distinction in Figure 11.3 is between (a) equal-width and (b) variable-width filters. Another distinction is between filter characteristics that overlap, as in (a), and characteristics that are non-contiguous, as in (b). The in-between situation of exactly contiguous filters is academic because practical implementations involve amplitude responses with finite roll-off characteristics.

The approach in Figure 11.3(b) calls for extremely fast filter roll-offs that minimize inter-band gaps, but it offers the possibility of reduced sampling rates

Figure 11.3 Amplitude responses in filter-banks consisting of four individual bandpass characteristics of (a) equal width and (b) unequal width.



(smaller values of f_{sk}), and hence a lower coding rate I [see (11.3)] for given values of R_k. Inter-band frequency gaps will be non-zero in practical filter designs, and these gaps cause a reverberant quality in the output speech of low bit rate SBC designs, unless the sub-bands can adaptively track regions of significant speech energy, such as formant frequencies in voiced speech [Crochiere and Sambur, 1977]. Discussions of SBC in this book are confined to fixed bands and fixed bit allocations. Sub-band coding systems with adaptive bit allocation perform significantly better because of the dynamic noise-shaping that they provide [Esteban and Galand, 1978] [Grauel, 1980] [Ramstad, 1982] [Heron, Crochiere and Cox, 1983]. However, the higher processing involved in adaptive bit allocation makes it particularly appropriate in the higher-complexity approach of TC (Chapter 12).

11.2 Transmission Rate *I*, *SNR* and Gain Over PCM

In SBC, each sub-band waveform $x_k(t)$ is sampled at a rate f_{sk} and encoded using R_k bits per sub-band sample. The transmission rate in SBC is therefore the sum of the bit rates needed to code individual sub-bands:

$$I = \sum_{k=1}^{M} f_{sk} R_k \quad \text{b/s} \tag{11.3}$$

In the special case of *equal-width* sub-bands,

$$\Delta W_k = W/M \quad \text{for all } k\ ; \qquad f_{sk} = 2\Delta W_k = 2W/M \tag{11.4a}$$

Since individual sub-band k can be sampled at the frequency $2\Delta W_k$ (Section 11.3), (11.3) simplifies to

$$I = \frac{2W}{M} \sum_{k=1}^{M} R_k \quad \text{b/s} \qquad \text{for equal-width bands} \tag{11.4b}$$

Note that (11.4b) reduces to the familiar form $I = 2WR$ for full-band coding if the total number of bits is expressed in the form

$$\sum_{k=1}^{M} R_k = MR \tag{11.5}$$

where R denotes the average number of bits used to encode a full-band sample. The simple equalities in (11.4b) and (11.5) imply that I is proportional to the sum of R_k values. This makes the design of variable bit allocation much simpler than in the general case of unequal-width sub-bands where the relationship between total bit rate I and the individual R_k values is less direct [see (11.3)].

In the following analysis, we assume non-overlapping equal-width sub-bands, so that the variances σ_{xk}^2 of sub-band inputs can be simply added to obtain the variance σ_x^2 of the full-band input. Similarly, variances σ_{rk}^2 of sub-band

0212

reconstruction errors can be added to obtain the variance σ_r^2 of signal reconstruction error.

The final reconstruction error variance is

$$\sigma_{r,SBC}^2 = \sum_{k=1}^{M} \sigma_{rk}^2 \tag{11.6a}$$

We assume error-free transmission, and the use of PCM (or DPCM) coding of individual sub-bands. As a result, for any k, the sub-band reconstruction error variance is $\sigma_{rk}^2 = \sigma_{qk}^2$, a corresponding quantization error variance. Therefore,

$$\sigma_{r,SBC}^2 = \sum_{k=1}^{M} \epsilon_k^2 \, 2^{-2R_k} \, \sigma_{xk}^2 \tag{11.6b}$$

The reconstruction error variance of a conventional (full-band) PCM coder, with bit rate equal to the average bit rate R in (11.5), is given by

$$\sigma_{r,PCM}^2 = \epsilon_x^2 2^{-2R} \sigma_x^2 \tag{11.6c}$$

where R is the number of bits/sample.

The *SNR* improvement G_{SBC} due to sub-band coding is the ratio of (11.6c) to (11.6b). Assuming for simplicity a *constant quantizer performance factor* ($\epsilon_k^2 = \epsilon_x^2$; all k), we obtain a gain over PCM that depends *only* on the bit allocation algorithm:

$$G_{SBC} = \frac{2^{-2R} \, \sigma_x^2}{\sum_{k=1}^{M} \left[2^{-2R_k} \sigma_{xk}^2 \right]} = \frac{2^{-2\sum R_k/M} \sum_{k=1}^{M} \sigma_{xk}^2}{\sum_{k=1}^{M} \left[2^{-2R_k} \sigma_{xk}^2 \right]} \tag{11.7a}$$

$$SNR\,|_{SBC}(\text{dB}) = SNR\,|_{PCM}(\text{dB}) + 10 \log G_{SBC} \tag{11.7b}$$

With a flat spectrum, G_{SBC} can never exceed 1 (Example 11.1). In the case of non-flat spectra, values of $G_{SBC} > 1$ can be realized by *bit allocation* procedures where R_k values are matched to σ_{xk}^2 values in a sense that will be clear from Examples 11.1 and 11.2. The special case of σ_{xk}^2-independent and equal R_k simply leads to $G_{SBC} = 1$ in (11.7). In Chapter 12, we will fully develop a theory of optimum bit allocation.

The sub-band coding gain G_{SBC} is really analogous to the prediction gain G_P in that both of these gains result from the non-flatness of input spectrum. Sub-band coding gain increases as a function of number of bands M; and prediction gain increases with order of prediction. As in prediction, the greatest values of G_{SBC} are realized when spectrum-dependent bit allocation is allowed to be time varying [Stjernvall, 1977] [Esteban and Galand, 1978] [Grauel, 1980] [Ramstad, 1982] [Heron, Crochiere and Cox, 1983]. This will indeed be the approach of Adaptive Transform Coding in Chapter 12.

Example 11.1. Two-band coding of a flat spectrum input

We shall again consider the simple case of equally wide sub-bands of width $W/2$ each. As a result of the flat spectrum,

$$\sigma_{x1}^2 = \sigma_{x2}^2 = \sigma_x^2/2 \tag{11.8a}$$

Using this in (11.7a),

$$G_{SBC} = 2\,\frac{2^{-(R_1+R_2)}}{2^{-2R_1}+2^{-2R_2}} \tag{11.8b}$$

Figure 11.4(a) plots G_{SBC} as a function of R_1 for the example of $R_1 + R_2 = 6$. The maximum value of $G_{SBC} = 1$ occurs for $R_1 = R_2 = 3$. This result can also be seen by identifying the above expression for G_{SBC} as the ratio of geometric mean and arithmetic mean of the terms 2^{-2R_1} and 2^{-2R_2}; this ratio is maximum when the terms are equal, i.e., when $R_1 = R_2$. •

Example 11.2. Two-band coding of the two-level spectrum of Figure 2.24 with $\alpha = 2/17$

From the results of Example 2.12,

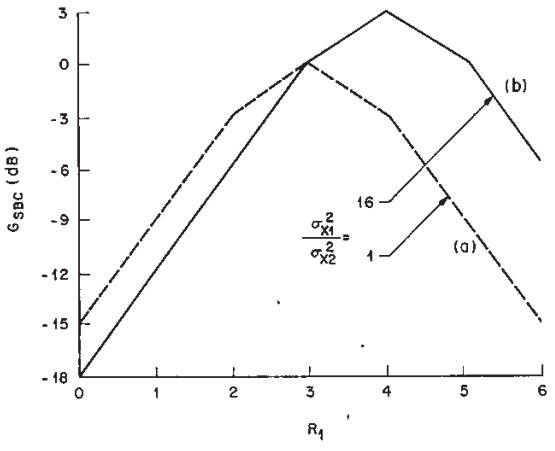
$$\sigma_{x1}^2 = (16/17)\,\sigma_x^2\,; \qquad \sigma_{x2}^2 = (1/17)\,\sigma_x^2 \tag{11.9a}$$

Using this in (11.7a),

$$G_{SBC} = 17\,\frac{2^{-(R_1+R_2)}}{16\cdot 2^{-2R_1}+1\cdot 2^{-2R_2}} \tag{11.9b}$$

Figure 11.4(b) plots G_{SBC} as a function of R_1 for the example of $2R = R_1 + R_2 = 6$. As in Example 11.1, equal bit allocation ($R_1 = R_2 = 3$) results in $G_{SBC} = 1$. But unlike in the flat-spectrum case, the maximum value of G_{SBC} is now 17/8. This maximum occurs if

Figure 11.4 G_{SBC} versus R_1 in two-band SBC schemes with $R = 3$ bits/sample, for (a) a flat-spectrum input, and (b) an input with the two-level psd of Figure 2.24 (with $\alpha = 2/17$).



0214

$R_1 = 4$ and $R_2 = 2$. Note that with this design, the error contributions in both sub-bands (denominator terms in the above expression for G_{SBC}) are equal. In fact, this maximum gain is the inverse of the spectral flatness measure γ_x^2 for the input (Problem 2.23):

$$\gamma_x^2 = [\alpha(2-\alpha)]^{1/2} = \left[\frac{2}{17}\cdot\frac{32}{17}\right]^{1/2} = \frac{8}{17}$$

The equality of maximum gain and γ_x^{-2} will be discussed again in Chapter 12. What is significant for the present discussion is that a two-band SBC procedure is sufficient to utilize all the spectral redundancy of a two-level psd: in other words, sufficient to realize $G_{SBC} = \gamma_x^{-2}$. The exact realization of the maximum possible gain is due to one other property of the input spectrum in question: the ratio of component variances, 16, is an integral power of 2.

The sufficiency of the two-band partition for the two-level spectrum is similar to a result in Chapter 6 where linear prediction of order N was adequate to utilize the *sfm* of an AR(N) process. •

The constraint on $\sum R_k$ used in Examples 11.1 and 11.2 is more meaningful with equal-width sub-bands [where I is directly proportional to $\sum R_k$ (11.4b)] than with variable-width sub-bands [where I is a weighted sum of the R_k (11.3)]. For equal-width sub-bands, the results of Example 11.2 can be generalized to the case of $M > 2$. The gain G_{SBC} in (11.7a) can again be maximized under the constraint of a given number of bits [see (11.5)]. This is equivalent to minimizing

$$\sigma_r^2 = \epsilon_q^2 \sum_{k=1}^{M} 2^{-2R_k}\,\sigma_{xk}^2$$

Using Lagrange multipliers,

$$\frac{\partial}{\partial R_k}\left[\epsilon_q^2 \sum_{k=1}^{M} 2^{-2R_k}\,\sigma_{xk}^2 - \lambda\left(MR - \sum_{k=1}^{M} R_k\right)\right] = 0$$

from which we can express R_k as a function of λ:

$$R_k = \frac{1}{2}\log_2\left(2\epsilon_q^2 \log_e 2\right) + \frac{1}{2}\log_2\frac{\sigma_{xk}^2}{\lambda}$$

Using this result in $MR = \sum R_k$, the *optimum bit allocation* is

$$R_{k,opt} \overset{\text{for}}{=} R + \frac{1}{2}\log_2 \frac{\sigma_{xk}^2}{\left[\prod_{l=1}^{M}\sigma_{xl}^2\right]^{1/M}} \tag{11.10}$$

and from the expression for σ_r^2 above, the *minimum mse* [Goodman, 1967] is

$$\min\{\sigma_r^2\} = M\,\epsilon_q^2 2^{-2R}\left[\prod_{l=1}^{M}\sigma_{xl}^2\right]^{1/M} \tag{11.11}$$

and the *maximum gain* is the ratio of arithmetic mean and geometric mean of the sub-band variances σ_{xl}^2:

$$\max\{G_{SBC}\} = \frac{\sigma_x^2}{M\left[\prod_{l=1}^{M}\sigma_{xl}^2\right]^{1/M}} = \frac{\frac{1}{M}\sum_{l=1}^{M}\sigma_{xl}^2}{\left[\prod_{l=1}^{M}\sigma_{xl}^2\right]^{1/M}} \tag{11.12}$$

The bit allocation in (11.10), which minimizes the mse σ_r^2, will also imply equal values of noise variance for different k, as a result of (11.6b) and (11.10); this is also shown formally in the analysis of Chapter 12. We will also see in Chapter 12 that specific forms of noise-shaping can be realized by bit allocations that minimize certain types of frequency-weighted mean square error.

In coding a signal such as speech whose psd can be approximated as an M-level psd with $M >> 2$ (a generalization of Figure 2.24), SBC performance increases with increasing M. Very large values of M can also take into account the fine structure in the psd due to the pitch period, in the sense of maintaining a locally flat psd within each sub-band. The use of fairly small values such as $M = 4$ is therefore for simplicity, rather than because of a saturation of objective gain G_{SBC} as a function of M. In particular, 4-band SBC coding of speech is significantly better than 2-band SBC coding, both objectively and from a subjective quality viewpoint [Cox, 1981, II]. Improvement of performance with M is maintained at values as high as $M = 16$ [Ramstad, 1982] [Esteban and Galand, 1982].

11.3 The Integer-Band Filter Bank

An important feature in Figure 11.1 is that bandpass filter cutoffs are chosen such that each band can be sampled at twice the corresponding bandwidth

$$f_{sk} = 2\Delta W_k\ ; \qquad k = 1,2,...,M \tag{11.13}$$

rather than at twice the highest frequency of the full-band signal. As discussed in Chapter 3, this is possible in the special situation of *integer-band sampling* [Crochiere and Rabiner, 1983] where the lower cutoff frequency W_{lk} in a sub-band k is an integral multiple of bandwidth (Figure 3.10):

$$W_{lk} = n\,\Delta W_k\ ; \qquad n = 0, 1,2,...; \qquad k = 1,2,3,...,M \tag{11.14}$$

The orders of bandpass filtering and sampling in Figure 11.1 can be reversed. Consider that discrete-time inputs $x(n)$ are available, sampled at the full-band Nyquist rate $f_s = 2W$ where $W = \sum \Delta W_k$, with summation from $k = 1$ to M, is the maximum frequency of the full-band signal.

Let the sub-band width be written in the form

$$\Delta W_k = W/\zeta_k\ ; \qquad k = 1,2,...,M \tag{11.15}$$

$\zeta_k = M$ for all k with *equal-width* sub-bands

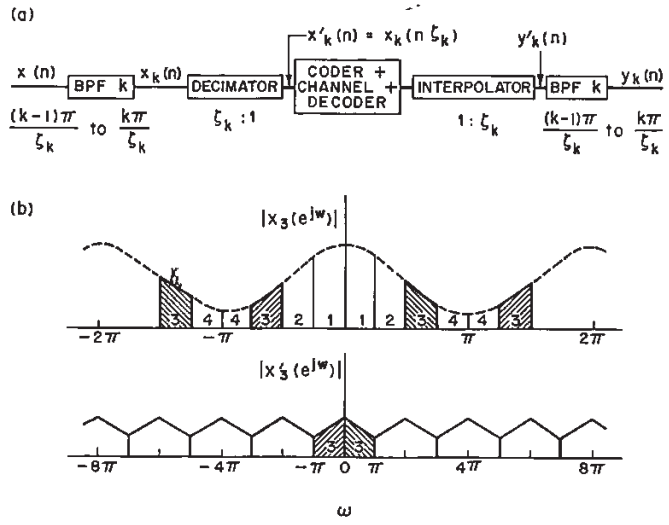
With the sub-band partition of Figures 11.2(a) and 11.3(a), $\zeta_k = 4$ for all k. Some of the speech coding examples mentioned later in this chapter use unequal sub-bands. In these examples, the values of ζ_k range from 4 to 30 (Tables 11.1, 11.2 and 11.3).

Figure 11.5 shows the sequence of filtering and coding operations in SBC, using the general example of sub-band k in Figure 11.5(a) and the special case of $k = 3$ and $\zeta_3 = 4$ in Figure 11.5(b).

The ζ_k:1 *decimator* sub-samples the bandpass output $x_k(n)$ by a factor ζ_k, implying a sampling rate of $f_{sk} = 2W/\zeta_k = 2\Delta W_k$ for sub-band k. This decimation implies a repetition rate of its spectrum that is higher than that of the full-band spectrum by a factor ζ_k. As a result of this, the x-axes in the two figures of (b) differ by a factor $\zeta_k = 4$. One of the repetitions of the spectrum will be in the baseband, so that the decimation effectively translates the lower frequency edge of the bandpass signal band to zero frequency. The 1:ζ_k *interpolator* fills in $(\zeta_k - 1)$ zeros in between every pair of incoming lowpass samples. The kth harmonic of the interpolated baseband is thus effectively bandpass-translated to the appropriate initial bandpass region. The explicit modulation processes mentioned in Section 11.1 are therefore replaced by simpler discrete-time processes of decimation and interpolation. It is assumed that the interpolation process includes an amplitude scaling factor of ζ_k. This maintains the original value of input variance in spite of the zero-valued amplitudes that are introduced in the interpolation process.

The amplitude spectra $|X_3(e^{j\omega})|$ and $|X'_3(e^{j\omega})|$ in Figure 11.5(b) refer to sub-band $k = 3$, with $\zeta_3 = 4$; the illustration is equivalent to the continuous-time case

Figure 11.5 Realization of integer-band sampling with a discrete-time input: (a) block diagram of SBC coding for sub-band k; and (b) original spectrum and resampled spectrum after decimation, for sub-band $k = 3$, with $\zeta_3 = 4$. The baseband spectrum resulting from the decimation is shifted back to the original frequency range of sub-band k after interpolation by a factor ζ_k.



0217

of Figure 3.9, which also used the example of $k = 3$. Note that the spectrum of the decimated sequence has its own frequency scaling. If the procedure of the last two paragraphs is repeated for an even-numbered sub-band ($k = 2$ or $k = 4$), it can be shown that the spectrum gets inverted in the process of lowpass translation to the baseband. This is, however, neutralized by a subsequent inversion in the interpolation process for even k [Crochiere and Rabiner, 1983].

The integer-band constraint in (11.14) is invariably assumed in SBC for the obvious reason of minimizing sub-band sampling frequencies and hence the overall information rate

$$I = \sum_{k=1}^{M} I_k = \sum_{k=1}^{M} f_{sk} R_k = \sum_{k=1}^{M} 2\Delta W_k \, R_k \quad \text{bits/second} \tag{11.16}$$

11.4 Quadrature-Mirror Filter Banks

The overlapping sub-band situation in Figure 11.3(a) suggests that aliasing effects can occur if sub-bands are sampled at $f_{sk} = 2W/M = \Omega_W/\pi M$. This problem is very elegantly tackled in the *quadrature-mirror filter bank* (QMFB) approach of Figure 11.6 [Esteban and Galand, 1977]. This figure shows the division of a full-band signal of maximum radian frequency π into two of equal width by using a constrained pair of lowpass and highpass filters. In the notation of (11.15), $\zeta_1 = \zeta_2 = M = 2$. By repeated subdivisions of resulting sub-bands using QMF filter banks, one can realize an SBC filter bank with M given by a power of 2, such as $M = 4$ as in Figure 11.3(a). Values of M that are not powers of 2 can also be realized by simply ignoring appropriate sub-band branches in the QMF tree (Table 11.3 and Problem 11.2).

The rest of the following discussion refers to the first stage of such a filter bank, involving two sub-bands as in Figure 11.6. When further stages of band-partitioning are introduced, each of the branches in Figure 11.6(a) will be split into further branches, and sampling frequencies will be reduced by factors of two at each stage; but the results of Figure 11.6(b) will apply repeatedly with appropriate redefinitions of the absolute frequencies represented by 0, π and $\pi/2$ in that figure.

Each of the sub-band signals $x_l(n)$ and $x_u(n)$ is resampled by a factor 2:1. This reduction of the sub-band sampling rates is necessary in order to maintain a minimal overall bit rate in encoding these signals. This reduction of sampling rate introduces aliasing terms in each of the sub-band signals because of the finite rate of roll-off in filter responses. For example, in the lower band the signal energy in the frequency range above $\pi/2$ is folded down into the range 0 to $\pi/2$ and appears as *aliasing distortion* in this signal, in the frequency range covered by the hatched region in the left half of Figure 11.6(b). In the above explanation, and in the rest of this section, π is *not* redefined as in Figure 11.5(b). Aliasing also occurs for the upper band in a similar fashion; any signal energy in the frequency range below $\pi/2$ is folded upward into its Nyquist band $\pi/2$ to π; this causes aliasing in the frequency range covered by the hatched area in the right half of Figure 11.6(b). This mutual aliasing of signal energy between the upper and lower sub-bands is

0218

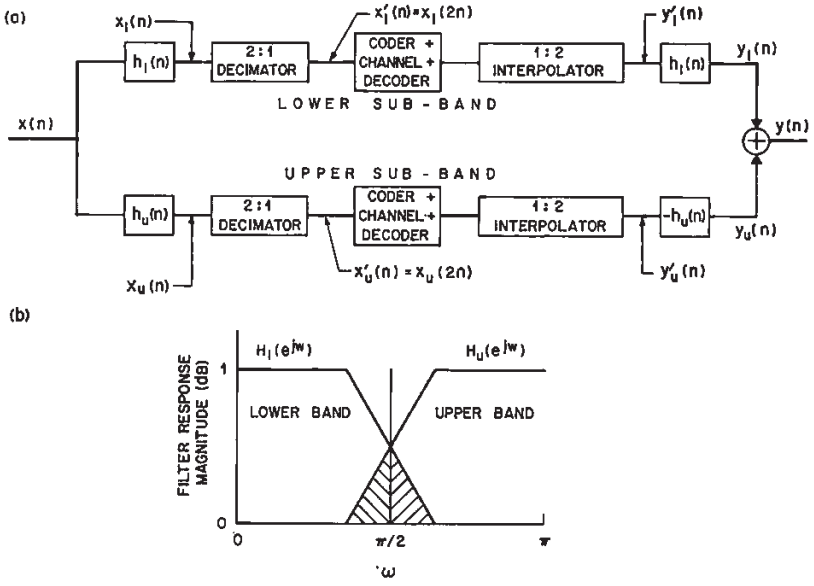


Figure 11.6 Quadrature-mirror filtering for splitting an input into two equal-width sub-bands: (a) implementation; and (b) qualitative illustration of a filter-bank response that provides alias-image cancellation [Esteban and Galand, 1978] [Crochiere, 1981].

sometimes called *interband leakage*. The amount of leakage that occurs between sub-bands is directly dependent on the degree to which the filters $h_l(n)$ and $h_u(n)$ approximate ideal lowpass and highpass filters, respectively.

In the reconstruction process, the sub-band sampling rates are increased by a factor 1:2 by filling in zero-valued samples between each pair of sub-band samples. This introduces a periodic repetition of the signal spectra in the sub-band. For example, in the lower band the signal energy from 0 to $\pi/2$ is symmetrically folded around the frequency $\pi/2$ into the range of the upper band. This unwanted signal energy, referred to as an *image* is mostly filtered out by the lowpass filter $h_l(n)$ in the receiver. This filtering operation effectively interpolates the zero-valued samples that have been inserted between the sub-band signals to values that appropriately represent the desired waveform [Crochiere and Rabiner, 1983]. Similarly, in the upper sub-band signal an image is reflected to the lower sub-band and filtered out by the filter $-h_u(n)$.

The degree to which the above images are removed by the filters $h_l(n)$ and $-h_u(n)$ is determined by the degree to which they approximate ideal lowpass and highpass filters. Because of the special relationship of the sub-band signals in the QMF filter bank, the remaining components of the images can be canceled by aliasing terms introduced in the analysis. This cancellation occurs *after* the addition of the two interpolated sub-band signals $y_l(n)$ and $y_u(n)$, and the cancellation is exact in the absence of coding errors. In the presence of coding, this cancellation is obtained to the level of quantization noise.

If S_l and S_u refer to lower-band and upper-band signals and subscripts A and I denote aliasing and imaging, the adder input $y_l(n)$ in Figure 11.6(a) will consist of the following main components, correct to filter attenuation effects: S_l and S_{uA} in the lower band and S_{lI} and S_{IuA} in the upper band. Similarly, input $y_u(n)$ will consist of components S_{uI} and S_{lIA} in the lower band and S_u and S_{IA} in the upper band. When $y_l(n)$ and $y_u(n)$ are added, components S_{uA} and S_{uI} cancel in the lower band, while components S_{lI} and S_{lIA} cancel in the upper band, leaving S_l and S_{lIA} in the lower band and S_u and S_{IuA} in the higher band. The components of S_{lIA} and S_{IuA} actually belong in the respective bands, and in the case of a QMF design with an allpass characteristic, these components exactly compensate for the in-band attenuations present in S_l and S_u.

One way of obtaining this cancellation property in the QMF filter bank is to use filters $h_l(n)$ and $h_u(n)$, which are respectively symmetrical and anti-symmetrical *finite impulse response* (FIR) designs with even numbers of taps, i.e.,

$$h_l(n) = h_u(n) = 0 \quad \text{for } 0 > n \geqslant N \; ; \tag{11.17}$$

$$h_l(n) = h_l(N-1-n) \; , \quad n = 0,1,\ldots,N/2-1 \; ; \tag{11.18a}$$

$$h_u(n) = -h_u(N-1-n) \; , \quad n = 0,1,\ldots,N/2-1 \tag{11.18b}$$

The cancellation of aliasing effects in the QMF filter bank further requires that the filters in Figure 11.6(a) satisfy the condition [Esteban and Galand, 1977] [Crochiere and Rabiner, 1983]

$$h_u(n) = (-1)^n h_l(n) \; , \quad n = 0,1,\ldots,N-1 \tag{11.19}$$

which is a *mirror image* relationship of the filters, implying symmetry about $\pi/2$, as in Figure 11.6(b). The coefficients of the filters are identical except that their signs alternate. Therefore, both filters can be realized using a single N-tap filter as the starting point.

Further, if the filter-bank output $y(n)$ is desired to be a delayed replica of input $x(n)$ (in the absence of coding errors), the filters $h_l(n)$ and $h_u(n)$ must also satisfy the condition

$$|H_l(e^{j\omega})|^2 + |H_u(e^{j\omega})|^2 = 1 \; , \tag{11.20}$$

where $H_l(e^{j\omega})$ and $H_u(e^{j\omega})$ are the Fourier transforms of $h_l(n)$ and $h_u(n)$, respectively; this is simply the condition for an *allpass* characteristic. If the allpass condition (11.20) is included in the mirror-image design (11.19), the point of intersection of the two filter functions in Figure 11.6(b) will be the -3 dB point for each transfer function.

The filter requirement in (11.20) cannot be met exactly by the mirror image filters of (11.19) except when $N = 2$ and when N approaches infinity. However, it can be very closely approximated for modest values of N. Filter designs which satisfy (11.18) and (11.19) and approximate the condition of (11.20) can be obtained with the aid of an optimization program [Johnston, 1980]. Resulting filter

Table 11.1 Quadrature Mirror Filters of order $N = 32$ and $N = 16$. Listed numbers are values of coefficients $h_l(n)$ for $N/2 \leq n \leq N$. Values of other coefficients follow (11.8) and (11.9) [Johnston, 1980] [Crochiere and Rabiner, 1983; Reprinted with permission].

$N = 32$		$N = 16$
$h(16)$ to $h(23)$	$h(24)$ to $h(31)$	$h(8)$ to $h(15)$
4.6645830E-01	1.7881950E-02	.47211220E 00
1.2846510E-01	−1.7219030E-04	.11786660E 00
−9.9800110E-02	−9.3636330E-03	−.99295500E-01
−3.9244910E-02	1.4272050E-03	−.26275600E-01
5.2909300E-02	4.1581240E-03	.46476840E-01
1.4468810E-02	−1.2601150E-03	.19911500E-02
−3.1155320E-02	−1.3508480E-03	−.20487510E-01
−4.1094160E-03	6.5064660E-04	.65256660E-02

coefficients for N values in the range 8 to 64 have been tabulated [Crochiere, 1981] [Crochiere and Rabiner, 1983].

Table 11.1 lists representative designs for $N = 32$ and $N = 16$. In a QMF tree for $M = 4$ sub-bands, the first subdivision may use filters of order $N = 32$. In view of the 2:1 decimation, a matching design for the second stage of the QMF tree would then be $N = 16$.

Figure 11.7 shows the frequency response characteristics for a 32-tap filter design [Crochiere, 1981]. Figure 11.7(a) shows the magnitude of $H_l(e^{j\omega})$ and $H_u(e^{j\omega})$ expressed in dB as a function of ω. As in the schematic of Figure 11.6(b), note that the roll-off regions of the two responses intersect at the −3dB point for each characteristic. Figure 11.7(b) shows the magnitude of the expression $|H_l(e^{j\omega})|^2 + |H_u(e^{j\omega})|^2$ expressed in dB as a function of ω. As can be seen from Figure 11.7(b), the requirement of (11.20) is satisfied to within ±0.025 dB, which is more than satisfactory for good SBC performance. The reconstruction error in the 32-tap design of Table 11.1 is also ±0.025 dB, but the stop-band attenuation of this filter (measured at the first stop-band peak) is 52 dB, which is much greater than the 37 dB attenuation in the example of Figure 11.7(a). In the 16-tap example of Table 11.1, the reconstruction error is ±0.07 dB, and the stop-band attenuation is 30 dB.

The use of FIR filters has the advantage of linear-phase characteristics which eliminate the problems of group delay distortions. This feature also allows the 2-band design of Figure 11.6 to be conveniently cascaded in three structures (Example 11.5, Problem 11.2) without the need for phase compensation. However, effective FIR designs imply significant coding delays. For example, with the 32-tap design just mentioned, the coding and decoding delays due to the first level of the QMF tree are 4 ms each, assuming 8 kHz input sampling; and subsequent levels of the QMF partition introduce corresponding additional delays. In order to implement SBC systems with smaller values of delay, there has been at least one proposal for a QMF bank based on *infinite impulse response* (IIR) designs [Ramstad and Foss, 1980]. This proposal includes special procedures for mitigating the group delay distortions inherent in IIR designs.

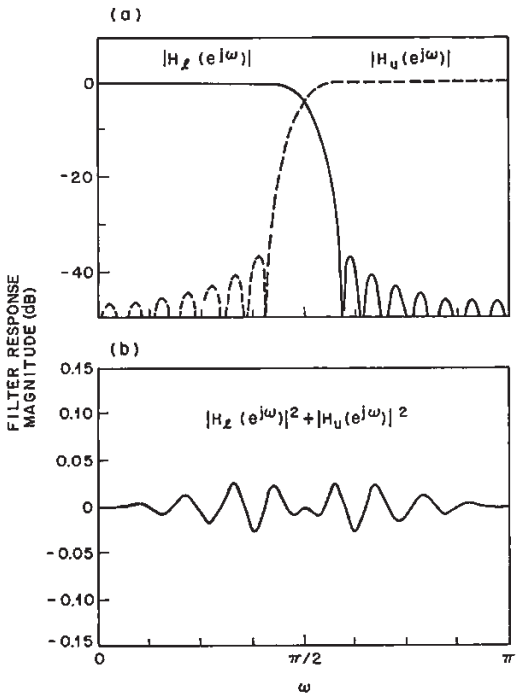


Figure 11.7 Illustration of amplitude-frequency responses in a quadrature mirror filter bank using FIR filters: (a) lowpass and highpass characteristics of individual 32-tap FIR filters; and (b) approximately allpass characteristic of the combination [Crochiere, 1981].

11.5 Sub-Band Coding of Speech

The following examples illustrate the design of fixed bit-allocation SBC systems for speech at bit rates in the range of 9.6 kb/s to 32 kb/s. The 32 kb/s system with fixed bit allocation can provide very high subjective quality, with MOS scores in the order of 4.2; a necessary condition for *toll quality* reproduction of telephone speech, while fixed bit-allocation SBC systems at lower bit rates provide different grades of *communications quality* encoding. Example 11.5 also illustrates the use of the QMF systems discussed in Section 11.4.

Example 11.3. Sub-band coding of speech at 9.6 kb/s

Table 11.2 shows a sub-band partition for a 9.6 kb/s SBC system. The sub-band sampling rates are obtained from an original sampling rate of 9.6 kHz by using ξ_k:1 decimations, with ξ_k values of 20, 10, 9 and 5.

The PCM coders use adaptive quantization (for example, the AQB system with a one-word memory, Chapter 4); and ranges of step size are individually matched to the long-time-averaged variances of individual sub-band signals. This is indicated by the different Δ_{min} values in Table 11.2. Respective Δ_{max} values are typically 256 times greater.

0222

Table 11.2 Sub-band coder designs for 9.6 kb/s [Crochiere, 1977]

Band	$\downarrow_k$ Decimation Factor from 9.6 kHz	Band Edges (Hz)	Sub-band Sampling Rates (Hz)	Relative Δ_{min} Values (dB)	R_k (bits/sample)	I_k (kb/s)
1	20	240-480	480	0.0	3	1.44
2	10	480-960	960	−3.0	3	2.88
3	9	1067-1600	1067	−8.5	2	2.13
4	5	1920-2880	1920	−14.0	1.5	2.88
Sync						0.27
			Total Bit Rate I (kb/s)			9.60
			Typical SNR (dB)			10.8

The bit allocation of (3, 3, 2, 1.5) bits/sample is a perceptually optimized design; it codes lower frequency sub-bands with greater fidelity than the higher frequency sub-bands. The fractional value of $R_4 = 1.5$ bits/sample can be obtained in several ways [Crochiere, 1977]; for example, by the use of $R_4 = 1.0$ and $R_4 = 2.0$ for alternate samples, with appropriate modification of step size logics.

Notice that in Table 11.2, the transmission rate for each band is a rational fraction of total rate I so that sub-band data can be multiplexed into a repetitive framed sequence. The lowest common denominator of these rational fractions, including the fraction of transmission rate reserved for frame synchronization purposes (denoted by "Sync" in Table 11.2) determines the smallest possible frame size for the SBC transmission system [Crochiere, 1977]. In certain applications such as voice storage and computer voice response, synchronization procedures may be built-in, and there may not be any need for transmitting special "Sync" bits. This is indeed the situation assumed in Table 11.4.

The quality of the time-invariant 9.6 kb/s system is limited by a reverberant quality in the output $y(n)$ due to inter-band gaps. Designs without gaps involve smaller values of R_k, greater quantization noise, and even lower quality. On the other hand, 9.6 kb/s SBC systems, as a class, perform much better than full-band fixed-prediction speech coders at 9.6 kb/s; in particular, better than 9.6 kb/s ADM systems. In subjective tests, the 9.6 kb/s SBC system is judged to be equivalent to an ADM system at twice the coding rate, i.e., 19.2 kb/s [Crochiere, 1977]. This advantage is directly related to variable bit allocation, and the fact that the quantization noise in the coding of sub-band k with R_k bits/sample is contained within that band. •

Example 11.4. SBC coding of speech at 16 kb/s

Table 11.3 refers to a 16 kb/s SBC system. The explanation of the numbers in this table is very similar to that for Table 11.2. In perceptual tests, this coder is comparable to 24 kb/s DPCM-AQB. The quality of 16 kb/s SBC with fixed bit allocation is useful for many communications applications, although this SBC output is clearly distinguishable from the original input in side-by-side comparisons.

A comparison of the Δ_{min} values in Tables 11.2 and 11.3 shows that the design of Table 11.3 is a better match to the long-time-averaged psd of speech. Note that in Figure 2.9(a), the psd peaks at a value of frequency that is non-zero, and so does the Δ_k versus k profile in Table 11.3. The behavior with respect to frequency k is monotonic in Table 11.2. •

Table 11.3 Sub-band coder design for 16 kb/s [Crochiere, 1977]

Band	ζ_k Decimation Factor from 10.67 kHz	Band Edges (Hz)	Sub-band Sampling Rates (Hz)	Relative Δ_{min} Values (dB)	R_k (bits/sample)	I_k (kb/s)
1	30	178-356	356	−2.0	4	1.42
2	18	296-593	593	0.0	4	2.37
3	10	533-1067	1067	−6.0	3	3.20
4	5	1067-2133	2133	−11.5	2	4.27
5	5	2133-3200	2133	−18.0	2	4.27
Sync						0.47
			Total Bit Rate I (kb/s)			16.00
			Typical SNR (dB)			13.6

Example 11.5. SBC coding of speech at 16, 24 or 32 kb/s

Table 11.4 shows a general sub-band partitioning framework that can be used for SBC coding at bit rates of 16, 24 or 32 kb/s. The sub-band partitioning is obtained by repeated use of the QMF partitioning principle (Problem 11.2). Note that the 16 and 24 kb/s systems use four sub-bands while the 32 kb/s system uses five sub-bands. The speech bandwidth allowed in this five-band system includes the 3 to 4 kHz range; this is a capability in excess of the 3.2 kHz limit assumed for telephone speech.

Figure 11.8 shows sub-band signals $x_k(n)$; $k = 1,2,3,4$ in a 4-band system which uses the four lower bands of Table 11.4. As shown in the table, respective sampling rates are $f_{sk} = 1000$, 1000, 2000 and 2000 Hz. These sampling rates are obtained from an original sampling rate of $f_s = 8000$ Hz by using ζ_k:1 decimations. Respective values of ζ_k, also listed in the table, are 8, 8, 4 and 4. Note that as we move the lowest sub-band (0-500 Hz) to the highest (2000-3000 Hz), the left half of the waveform becomes increasingly more prominent. This is because of the preponderance of high frequency energy in the left half of the original full-band waveform. Notice also that the waveforms in the lower two sub-bands do not look very different. This is because the original full-band waveform in this example has significant energies centered at about 700 Hz, and this is a strong common component in both of the lower two sub-bands, which happen to overlap significantly in the 700 Hz region.

Table 11.4 Sub-band coder designs for 16, 24 or 32 kb/s [Crochiere, 1981].

Band	ζ_k Decimation Factor From 8 kHz	Band Edges (Hz)	Sub-band Sampling Rates (Hz)	R_k (bits/sample) for I (kb/s) = 16	24	32
1	8	0-500	1000	4	5	5
2	8	500-1000	1000	4	5	5
3	4	1000-2000	2000	2	4	4
4	4	2000-3000	2000	2	3	4
5	4	3000-4000	2000	0	0	3

0224

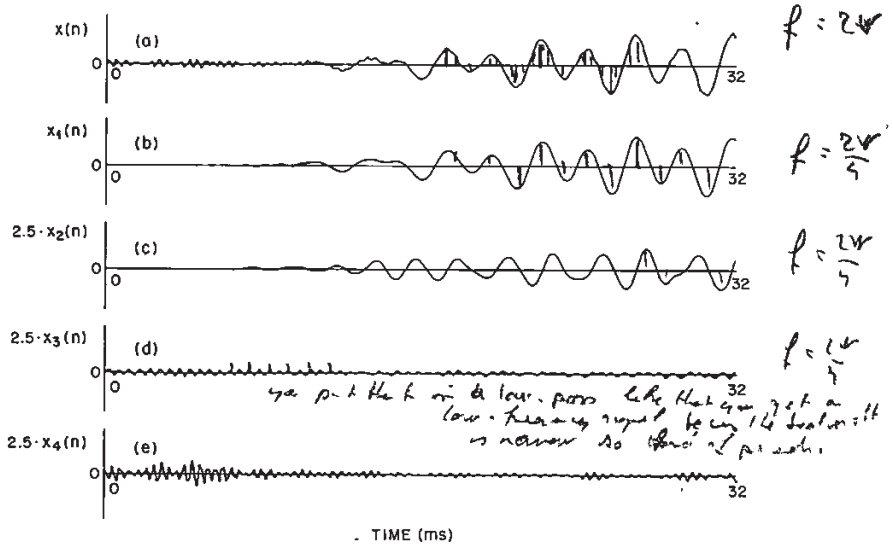


Figure 11.8 (a) Full-band speech waveform of duration 32 ms; and (b)(c)(d)(e) sub-band signals $x_k(n)$ for $k = 1,2,3,4$ in the SBC system of Table 11.4. The sub-band signals $x_2(n)$, $x_3(n)$ and $x_4(n)$ have been magnified by a factor of 2.5 [Cox, 1983].

The full-band signal corresponding to $x_n(k)$ waveforms is a voiced speech segment of duration 32 ms. At the respective Nyquist sampling rates f_{sk}, the sub-band signals exhibit much lower values of adjacent-sample correlation than a full-band speech waveform sampled at its Nyquist rate f_s. The entire waveform $x_1(n)$ in Figure 11.8(b), for example, is sampled only 32 times because of the decimation to 1 kHz. The lower correlations in sub-band signals can also be conjectured from Figure 11.2 where individual sub-bands are seen to have flatter psd's than the full-band speech. (Also, see Problem 11.7). This is the reason why PCM, rather than DPCM, is often assumed in SBC systems, although, with real speech spectra, non-negligible values of prediction gain can be realized even in the case of (individually optimized) fixed predictors of order one.

The nature of the long-time-averaged speech spectrum [Figure 2.9(a)] suggests that the long-term sub-band spectrum can be high-pass in the ranges of sub-bands 1 and 2 of Table 11.4. High-pass spectra in sub-bands can also result from high-frequency pre-emphasis in telephone systems. In fact, in an implementation of the coders of Table 11.4 with telephone speech that includes high-frequency pre-emphasis, the sub-band coders use DPCM-AQB with fixed first-order predictors, that reflect highpass spectra in four out of five sub-bands. Recommended values of coefficient h_1, for pre-emphasized speech, are -0.71, -0.28, -0.31, 0.26 and -0.64 in sub-bands 1 through 5, respectively [Daumer, 1982]. •

With fixed bit allocation, high quality coding with SBC requires a total bit rate in the order of 32 kb/s. In formal subjective testing, the SBC system of Table 11.4 obtains MOS scores (on a scale of 1 to 5), of 4.3, 3.9 and 3.1 at bit rates of 32, 24 and 16 kb/s [Daumer, 1982]. If the bit rate is at least 24 kb/s, there is an

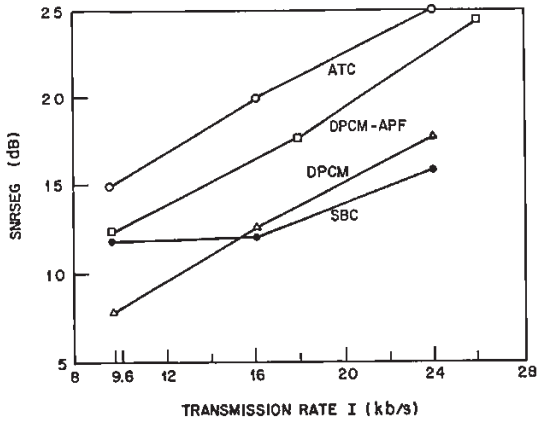
adequate margin of safety for accommodating tandem encodings of 2 to 3 SBC stages with useful final quality.

Although the results of Tables 11.2 to 11.4 refer to specific speech coding systems, they provide general design features that apply to other coding applications as well. An important example is in 9.6 kb/s SBC-TDHS (*time-domain harmonic scaling*) coding systems involving time-compression of input speech by a factor 2:1 prior to sub-band coding at a bit rate slightly under 2(9.6) = 19.2 kb/s [Malah, 1979] [Malah, Crochiere and Cox, 1981] [Crochiere, Cox and Johnston, 1982]. This procedure can result in an improvement over direct SBC at 9.6 kb/s, but the improvement is obtained at the cost of additional complexity, needed for pitch computations that TDHS is based on. Another example of SBC application is the coding of FM-grade audio signals (W = 20 kHz) (Problem 11.3) and AM-grade audio signals (W = 7 kHz). For example, a two-band SBC system with the band partition [0-3500 Hz] [3500 Hz-7000 Hz] is a good digitizer for 7 kHz material at a coding rate of 56 kb/s [Johnston and Crochiere, 1979].

Comparison of DPCM, DPCM-AP, SBC and ATC. Figure 11.9 shows the performance of four speech coders as a function of bit rate I. The set of coders includes two time-domain coders (DPCM coders with and without adaptive prediction (prediction order equal to one and eight, respectively) and two frequency-domain coders, SBC and ATC (Chapter 12).

One significant comparison in the figure is between the *medium complexity* techniques, SBC (with fixed bit allocation) and DPCM (with fixed prediction). The DPCM coder degrades rapidly if $I < 16$ kb/s ($R < 2$ bits/sample) because of poor quantization performance, compounded by the effects of quantization error feedback. The other significant comparison is between the *high complexity* techniques, DPCM-AP (APC) and ATC. In this specific study, neither of the two techniques incorporated pitch structure. It is interesting that ATC has a consistent

Figure 11.9 Segmental *SNR* in time-domain and frequency-domain speech coders as a function of bit rate I [Tribolet et al.;©1979, AT&T].



0226

2 to 3 dB advantage. In another study, involving a subjective comparison of APC and ATC, both using pitch in formation, APC was in fact rated slightly better [Daumer, 1982]. Part of the reason for the very good performance of that APC system was a carefully designed algorithm for noise shaping.

The comparisons between DPCM and DPCM-APF, and between SBC and ATC, are less interesting in that the inferences are well expected in each case. The ATC system can be regarded crudely as the equivalent of an SBC system with two very significant refinements: (a) a much larger number of sub-bands M, and (b) adaptive bit-allocation $\{R_k\}$ that is matched to short-term rather than long-term-averaged speech spectrum. As a consequence of these refinements, ATC systems utilize the spectral non-flatness in speech more completely and adaptively, in a fashion similar to redundancy-removal in a full-fledged APC system with higher-order spectrum prediction. The nonadaptive SBC system utilizes spectral non-flatness only in a non-adaptive, and hence incomplete sense. A special, but academic, input for which SBC provides complete redundancy removal is one with a staircase-type psd, as in Figure 2.24. For that two-level psd example, SBC with $M = 2$ provides a complete utilization of spectral non-flatness (Example 11.3 and Problem 11.6).

Combinations of sub-band coding and adaptive prediction. The performance of sub-band coding can be significantly improved by supplementing it with the time-domain operation of adaptive prediction [Atal and Remde, 1981] [Honda, Kitawaki and Itakura, 1982]. The general philosophy of these hybrid techniques is to split the burden of redundancy removal between the frequency domain and the time domain. The hybrid approach also provides the possibility of noise shaping in both the frequency domain and the time domain. In one system, the latter is provided by dividing basic time blocks of duration in the order of one pitch period into sub-blocks, typically four in number; and allocating quantization bits on the basis of prediction error energy in each sub-interval. This is done separately for each of typically three to four sub-bands. The bit allocation in the time domain provides better encoding of the higher-valued prediction error samples during the onset of a pitch period and mitigates the problem of quantization error feedback, separately in each sub-band. With spectrum prediction of order four and pitch prediction, the sub-band coder with time-frequency bit allocation provides MOS quality scores in the order of 3.5 at 16 kb/s and 4.0 at 24 kb/s [Honda, Kitawaki and Itakura, 1982].

11.6 Transmission Error Effects

Bit errors in an individual sub-band will generate error contributions at the receiver output within that frequency band. Since the channel error contributions in different sub-bands are expected to be uncorrelated, the corresponding error variances will simply add. The results of Section 11.2 and Section 4.9 can therefore be used to derive an expression for channel error variance σ_c^2 in an SBC system. Also, explicit forms of error-protection can be used to mitigate channel

0227

error effects in SBC decoding. Error-protection systems can exploit not only the different sensitivities of various bits in transmitted codewords (as in Section 4.9), but also the different sensitivities of various sub-bands to transmission error effects. These points will be made more quantitative during the discussion of Transform Coding in Chapter 12.

Example 11.6. Transmission error effects in 24 kb/s SBC systems

Figure 11.10 describes the performance of three SBC systems at $I = 24$ kb/s and bit error rate p_e. In coder A, the adaptive quantizers use the one-word memory logic (4.185). In coder B, the adaptive quantizers use the robust version (4.199a) of the adaptation logic, with step size leakage factor $\beta = 31/32$. In coder C, the sign bit (most significant bit) as well as the most significant magnitude bit in the lowest frequency sub-band are ideally error protected. This protection requires an appropriate amount of bit-stealing from the quantizers, in order to maintain the total bit rate at the original value of $I = 24$ kb/s (Problem 11.8). However, the resulting increase of quantization noise is more than compensated for by reductions in channel noise if the error rate p_e is in the order of 10^{-2} or greater. The overall performance is so as to provide acceptable speech outputs at $p_e = 10^{-2}$, but not at $p_e = 10^{-1}$. •

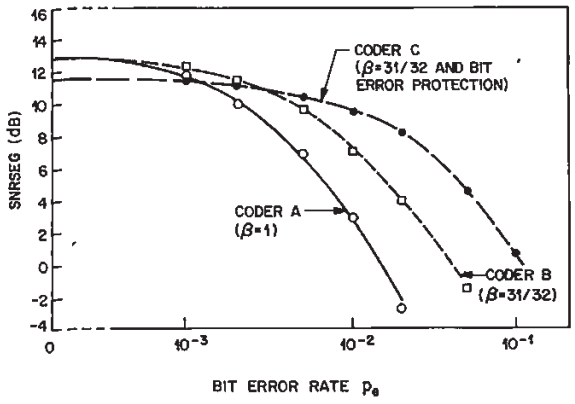


Figure 11.10 Transmission error performance of 16 kb/s SBC systems. Values of segmental *SNR* as a function of bit error rate p_e [Crochiere;©1978, AT&T].

Problems

(11.1) Consider a 4-band SBC system where sub-band partitions are constrained by filter-bank considerations to be either (a) or (b) below:

(a) [0–800] [800–1600] [1600–2400] [2400–3200] Hz

(b) [225–450] [450–900] [1000–1500] [1800–2700] Hz

0228

Ignoring side information for synchronization bits, show that bit allocations that provide 9.6 kb/s with the filter-banks above are respectively

(a) 3, 2, 1, 0 bits/sample

(b) 4, 3, 2, 1 bits/sample

Note that (b) is closer to the design recommended in Table 11.2.

(11.2) It is required to design a four-band filter-bank [0-0.5] [0.5-1.0] [1.0-2.0] [2.0-3.0] kHz for 16 kb/s coding of speech.

(a) Show the band-division tree that realizes this partition using repeated quadrature mirror filtering of the [0-4.0] kHz band.

(b) Assuming that $\max\{R_k\} = 5$ and $\min\{R_k\} = 2$ bits/sample, show that there are at least 2 bit assignments [5,3,2,2] [4,4,2,2] that realize a total bit rate $I = 16$ kb/s. Ignore side information considerations, as in Problem 11.1.

(11.3) Consider a 6-band SBC system for studio-grade music, with sub-bands [0-0.625] [0.625-1.25] [1.25-2.5] [2.5-5.0] [5.0-10.0] [10.0-20.0] kHz. Allowing 6 kb/s for synchronization information, show that the bit allocation (12, 10, 9, 8, 6, 5) bits/sample permits digital transmission over a 256 kb/s channel.

(11.4) Consider a SBC system with M bands of equal width ΔW. Consider the SBC input as a 1-second sequence of $2M$ ΔW samples. Denote variances of individual samples of sub-band k by σ_{xk}^2 ; $k = 1,2,...,M$. Let each of these samples in sub-band k be quantized using R_k bits/sample. Because of a constant sampling rate $2\Delta W$, the total number of samples per second is independent of k. The optimum bit allocation for minimum mse, and the maximum gain G_{SBC} over single-band PCM, are given (on a *per sample* basis) by (11.10) and (11.12).

It is also useful to give results on a *per second* basis. Define $\mathbf{R}$ as the total bit rate (bits/sec) in the SBC system, $\mathbf{R}_k = 2\Delta W R_k$ as the bit rate (bits/second) allocated to encode band k, and $\sigma_{xk}^2 = \sigma_{xk}^2 2\Delta W$ as the power in sub-band k. Show that the formulas (11.10) and (11.12) are equivalent to the pair of formulas

$$\mathbf{R}_{k,opt} = \frac{\mathbf{R}}{M} + \Delta W \log_2 \frac{\sigma_{xk}^2}{\left[\prod_{k=1}^{M} \sigma_{xk}^2\right]^{1/M}} ; \qquad \max\{G_{SBC}\} = \frac{\frac{1}{M}\sum_{k=1}^{M} \sigma_{xk}^2}{\left[\prod_{k=1}^{M} \sigma_{xk}^2\right]^{1/M}}$$

(11.5) Refer to Problem 11.4 and the equal-width four-band partition of Problem 11.1(a). Equate the power $\sigma_{xk}^2 = \sigma_{xk}^2 2\Delta W$ to $S_k 2\Delta W$ where S_k represents the average value of input psd in sub-band k. Use the long-time psd of speech in Figure 2.9(a) together with the formula for $\mathbf{R}_{k,opt}$ in Problem 11.4 to show that the bit allocations in Problem 11.1(a) are indeed nearly optimal for a mmse criterion.

(11.6) Consider the highly structured two-level psd of Figure 2.24 and (2.168).

(a) Consider SBC with $M = 2$ $\Delta\Omega = \Omega_W/2$, and band-edges at 0, $\Omega_W/2$ and Ω_W to follow the shape of the input psd. Use the result of Problem 11.4 to show that the maximum gain in 2-band SBC is

$$\max\{^2G_{SBC}\} = [\alpha(2-\alpha)]^{-1/2}$$

(b) Show that $\max\{^2G_{SBC}\} = \gamma_x^{-2} = \max\{G_{SBC}\}$, the reciprocal of the spectral flatness measure (Problem 2.22) of the process, i.e., SBC with $M = 2$ bands is sufficient to realize the maximum performance for this special psd.

(11.7) Consider the case of M equal-width sub-bands.

(a) Show that the adjacent sample correlation ρ_{1k} for the decimated signal in sub-band k vanishes if the psd in that sub-band is flat-topped.

(b) Determine the four values of ρ_{1k} for the 4-band partition of a signal with an integrated power spectrum and $\psi = 4$.

(11.8) Consider a 4-band SBC system for 24 kb/s speech coding, with the QMF partition [0, 1000] [1000, 2000] [2000, 3000] [3000, 4000] Hz, and with a normal bit allocation of (5, 4, 2, 1) bits/sample for quantization. Consider that this system is adapted to a noisy channel by protecting all bits in the first sub-band using a (12, 8) Hamming code (with 4 redundant bits for every 8 message bits). Show that if the bit allocation for quantization is changed to (4, 3, 2, 1) bits/sample, the total transmission rate is still 24 kb/s.

References

B. S. Atal and J. R. Remde, "Split-Band APC System for Low Bit Rate Encoding of Speech," Proc. ICASSP, April 1981.

H. G. Bellanger, G. Bonnerot, and M. Coudreuse, "Digital Filtering by Polyphase Network: Application to Sample-Rate Alteration and Filter-Banks", IEEE Trans. on Acoustics, Speech and Signal Proc., pp. 252-259, 1976.

R. V. Cox, unpublished work, Bell Laboratories, 1981.

R. V. Cox, "A Comparison of Three Speech Coders to Be Implemented on the Digital Signal Processor," Bell System Tech. J., pp. 1411-1421, September 1981.

R. V. Cox, unpublished work, Bell Laboratories, 1983.

R. E. Crochiere, "On the Design of Sub-Band Coders for Low Bit Rate Speech Communication," Bell System Tech. J., pp. 747-771, May-June 1977.

R. E. Crochiere, "An Analysis of 16 kb/s Sub-band Coder Performance: Dynamic Range, Tandem Connections, and Channel Errors," Bell System Tech., J., pp. 2927-2952, October 1978.

R. E. Crochiere, "Sub-band Coding," Bell System Tech. J., pp. 1633-1654, September 1981.

R. E. Crochiere, R. V. Cox and J. D. Johnston, "Real Time Speech Coding," IEEE Trans. on Communications, pp. 621-634, April 1982.

R. E. Crochiere and L. R. Rabiner, "Interpolation and Decimation of Digital Signals — A Tutorial Review," Proc. IEEE, pp. 300-331, March 1981.

R. E. Crochiere and L. R. Rabiner, *Multirate Digital Processing*, Prentice Hall, 1983.

R. E. Crochiere and M. R. Sambur, "A Variable-Band Coding Scheme for Speech Encoding at 4.8 kb/s", Bell System Tech. J., pp. 771-780, May-June 1977.

R. E. Crochiere, S. A. Webber and J. L. Flanagan, "Digital Coding of Speech in Sub-bands," Bell System Tech. J., pp. 1069-1085, October 1976.

W. R. Daumer, "Subjective Evaluation of Several Efficient Speech Coders," IEEE Trans. on Communications, pp. 662-665, April 1982.

D. Esteban and C. Galand, "Application of Quadrature Mirror Filters to Split Band Voice Coding Schemes," Proc. ICASSP, pp. 191-195, May 1977.

D. Esteban and C. Galand, "32 kb/s CCITT-Compatible Split Band Coding Scheme," Proc. ICASSP, pp. 320-325, 1978.

J. L. Flanagan et al., "Speech Coding", IEEE Trans. on Communications, pp. 710-737, April 1979.

C. Galand and D. Esteban, "16 kb/s Sub-band Coder Incorporating Variable Overhead Information," Proc. ICASSP, pp. 1684-1687, April 1982.

L. M. Goodman, "Channel Encoders," Proc. IEEE, pp. 127-128, 1967.

C. Grauel, "Sub-band Coding with Adaptive Bit Allocation," Signal Proc., 1980.

C. D. Heron, R. E. Crochiere and R. V. Cox, "A 32-Band Sub-Band/Transform Coder Incorporating Vector Quantization for Dynamic Bit Allocation," Proc. ICASSP, pp. 1276-1279, April 1983.

M. Honda, N. Kitawaki and F. Itakura, "Adaptive Bit Allocation Scheme in Predictive Coding of Speech," Proc. ICASSP, pp. 1672-1675, May 1982.

J. D. Johnston, "A Filter Family Designed for Use in Quadrature Mirror Filter Banks," Proc. ICASSP, pp. 291-294, April 1980.

J. D. Johnston and R. E. Crochiere, "An All-Digital 'Commentary-Grade' Sub-band Coder," Audio Eng. Soc. Conv., AES preprint, May 1979.

D. Malah, "Time Domain Harmonic Scaling Algorithms for Bandwidth Reduction and Time Scaling of Speech Signals," IEEE Trans. on Acoustics, Speech and Signal Processing, pp. 121-133, April 1979.

D. Malah, R. E. Crochiere and R. V. Cox, "Performance of Transform and Sub-Band Coding Systems Combined with Harmonic Scaling of Speech," IEEE Trans. on Acoustics, Speech and Signal Proc., pp. 273-283, April 1981.

L. R. Rabiner and R. W. Schafer, *Digital Processing of Speech Signals*, Prentice-Hall, Englewood Cliffs, N.J., 1978.

T. A. Ramstad, "Sub-band Coder with a Simple Bit Allocation Algorithm: A Possible Candidate for Digital Mobile Telephony," Proc. ICASSP, pp. 203-207, May 1982.

T. A. Ramstad and O. Foss, "Sub-band Coder Design Using Recursive Quadrature Mirror Filters," Proc. EUSIPCO, pp. 747-752, 1980.

J. E. Stjernvall, "On Rate and Frequency Allocation in Sub-band Coding of Gaussian Sources", Dept. of Elec. Eng. Report No. 1977-12-05, Linkoping University, Sweden, 1977.

J. M. Tribolet, P. Noll, B. J. McDermott and R. E. Crochiere, "A Comparison of the Performance of Our Low Bit Rate Speech Waveform Coders," Bell System Tech. J., pp. 699-713, March 1979.

12

Transform Coding

12.1 Introduction

We have seen in Chapter 6 that a stationary random sequence $\{X(n)\}$ can be efficiently encoded by predictive coding (DPCM) if its psd $S_{xx}(e^{j\omega})$ is not flat. Specifically, the reconstruction error in predictive encoding is upper-bounded in the form

$$\sigma_r^2 = \epsilon_q^2 \sigma_d^2 \geqslant \epsilon_q^2 \gamma_x^2 \sigma_x^2 \tag{12.1a}$$

where ϵ_q^2 is the quantizer performance factor, σ_d^2 is the variance of the difference (prediction error) signal, and γ_x^2 is the spectral flatness measure:

$$\gamma_x^2 = \exp\left[\frac{1}{2\pi}\int_{-\pi}^{\pi} \log_e S_{xx}(e^{j\omega})\, d\omega\right] / \sigma_x^2 \tag{12.1b}$$

For the example of first-order Gauss-Markov sources (Chapter 2), (12.1b) takes on the special values $(1-\rho^2)$ and $(1-\rho^2)^2$ in one- and two-dimensional cases, respectively, and the prediction gains (gains over PCM), given by the reciprocal of γ_x^2 values, are realized by an optimal first-order predictor in the one-dimensional example (6.24c), and by an optimal three-element predictor in the separable two-dimensional case (6.36d). With higher-order processes, we have seen that the predictor order should be correspondingly higher in order to approach the performance bound (12.1). Finally, we recall that the γ_x^2 bound applies to systems which exploit linear dependencies between input samples. DPCM is an example of

0232

such a system, and the first-order Markov source (with $\rho \rightarrow \pm 1$) is an example where linear dependencies can be simply observed, in terms of very significant correlations between adjacent input samples, or, equivalently, a psd that is lowpass or highpass.

The techniques of the present chapter will represent another approach for the utilization of linear dependencies for efficient digitization, called *Transform Coding* (TC). Its performance will be characterized by a performance gain over PCM, denoted by G_{TC}:

$$SNR\text{ (dB)}|_{TC} = SNR\text{ (dB)}|_{PCM} + 10 \log G_{TC} \tag{12.2}$$

The gain G_{TC} will be a function of the source statistics, the type of linear transform and the transform length N to be defined presently. Later sections of this chapter will include closed-form expressions for G_{TC} for certain model sources, as well as experimental values of G_{TC} for speech and image inputs.

The asymptotic mse performance is theoretically the same for DPCM and TC [Nitadori, 1970]. In the latter case, this performance is expressed by the asymptotic gain ${}^{\infty}G_{TC}$; and its maximum value is equal to the maximum value of ${}^{\infty}G_P$ for a given source. However, important differences exist between DPCM and TC, especially at low bit rates: in terms of subjective quality, matching of algorithm to input statistics, and effects of transmission errors. In the case of video inputs, TC is more robust than DPCM with regard to input statistics and channel errors [Netravali and Limb, 1980]. In the case of an ideal channel and a known source, TC offers a more direct approach to $R(D)$ bounds, in both image and speech applications. The price paid for these advantages is increased encoding delay which, in TC, is proportional to transform length N and, in the case of TC with adaptive bit assignment (ATC), an encoding complexity comparable to that of fully adaptive APC.

Transform coding is a "frequency-domain" approach like SBC; however, the problem of bit allocation, introduced in Chapter 11, will be considered much more carefully in this chapter; and in the case of speech signals, the number of subbands, or equivalently, the number of transform coefficients, will be much greater than four, which was typical in the discussion of Chapter 11. The *number of transform coefficients* will be called N; it will also be referred to as the *order of the transform*; transform coding is also referred to as *block quantization* [Huang and Schultheiss, 1963] and input blocks used for an Nth order transform will in fact be N samples long. By using appropriate values of N and appropriate bit allocation strategies, transform coding procedures for speech and images will be shown to provide *high quality* digitizations at an average rate of $R = 2$ bits/sample (Figures 1.5 and 1.6) and *acceptable* digitizations at $R = 1$ bit/sample.

Variable bit allocation is always assumed in this chapter; and for simplicity, the resulting *average bit rate* R will be referred to simply as the *bit rate* R.

Transform Coding may also be used for the efficient encoding of sequences which are not successive samples of a waveform, but samples of N correlated sources: for example, the outputs of N parallel filters in a speech vocoder [Kramer and Mathews, 1956]. In all of the rest of this chapter, however, we will be

concerned with the use of TC for encoding sequences of successive samples of a single waveform.

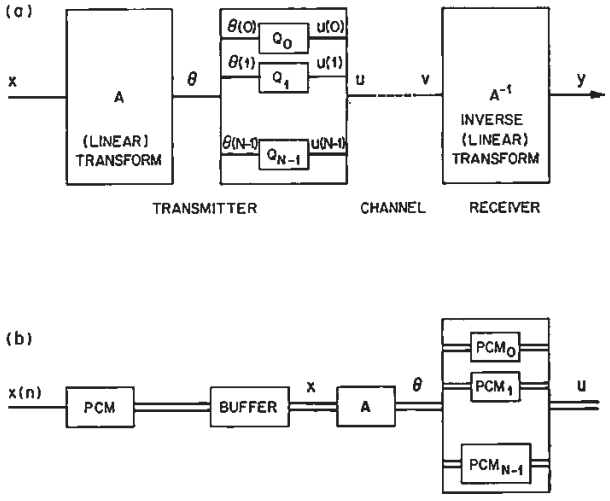
The efficiency of a transform coding system will depend on the type of linear transform (Sections 12.5 and 12.6) and the nature of bit allocation for quantizing transform coefficients (Section 12.4). Most practical systems will be based on sub-optimal approaches for transform operation as well as bit allocation.

Figure 12.1(a) is a block diagram of transform coding. It is a waveform digitizing procedure where a block of N input samples $x(n)$ is linearly transformed into a set of N *transform coefficients* $\theta(n)$ (for example, a set of Fourier coefficients). The coefficients are then quantized for transmission, and a reconstruction of $x(n)$ is obtained at the receiver using an inverse transform operation on quantized coefficients. Figure 12.1(b) shows how the above coding scheme is implemented. The input block could be one of high resolution PCM samples; for example, 8-bit resolution for images, and 12-bit resolution for speech. The output of the coder is the combination of the outputs of N PCM coders that convey quantized coefficient information, typically with a much lower total bit rate than what is present at the coder input.

In principle, the set of N transform coefficients can be quantized with lower average mse by the use of vector quantization or the related methods of Chapter 9. In the rest of this chapter, however, we shall follow the simpler practice of a scalar quantizer bank as in Figure 12.1. We will however briefly consider simple differential techniques for efficient quantization of correlated sets of coefficients in so-called hybrid coding systems (Section 12.6.5).

A crucial part of Transform Coding is a bit-allocation algorithm that provides the possibility of quantizing some coefficients "frequency components" more finely

Figure 12.1 (a) Block diagram of transform coding (TC); and (b) implementation of the encoder with a bank of PCM coders.



than others. The criterion for bit allocation can be the minimization of the mean squared value of reconstruction error. More generally, procedures may involve frequency-weighted reconstruction errors and adaptive bit-allocation for nonstationary inputs.

12.2 Linear Transforms of Order $N = 2$

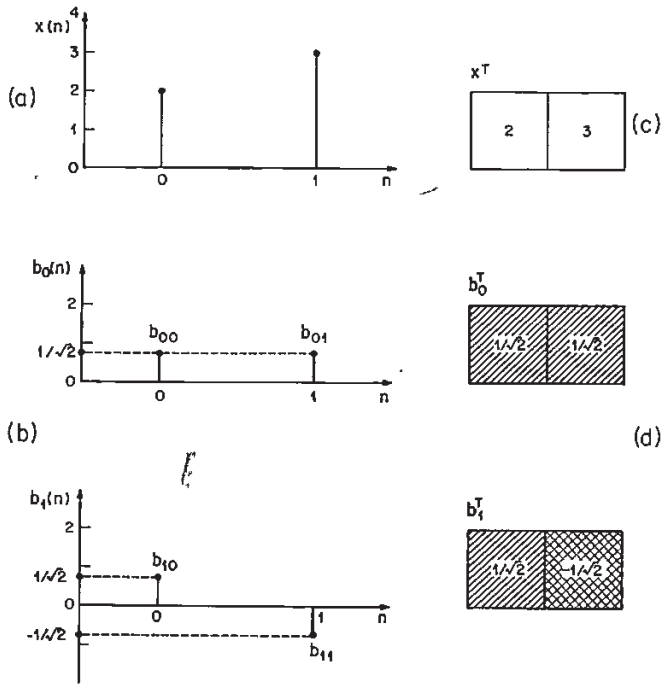
A simple way of introducing transform coding or block quantization is with reference to the deterministic "lowpass" input sequence $\{x(n)\}$ of Figure 12.2:

$$x(0) = 2\,; \qquad x(1) = 3 \tag{12.3}$$

Clearly, this sequence {2,3} can be *approximately* represented in the form {2.5,2.5}. For reasons that will be apparent as formalism develops, we shall write the approximating sequence in the form

$$\{2.5,2.5\} = \frac{5}{\sqrt{2}}\left\{\frac{1}{\sqrt{2}},\frac{1}{\sqrt{2}}\right\} = \frac{5}{\sqrt{2}}\{b_{00},b_{01}\} \tag{12.4}$$

Figure 12.2 (a) Input sequence of length two; and (b) a set of two basis sequences used to represent (a); (c) and (d) are matrix representations of (a) and (b).



0235

The input $\{x(n)\}$ can be *exactly* described if (12.4) is corrected by adding to it a sequence $\{-0.5, +0.5\}$:

$$\{-0.5,+0.5\} = \frac{-1}{\sqrt{2}}\{\frac{1}{\sqrt{2}},\frac{-1}{\sqrt{2}}\} = \frac{-1}{\sqrt{2}}\{b_{10},b_{11}\}. \tag{12.5}$$

The b-sequences in (12.4) and (12.5) can be regarded as *basis sequences* or *basis vectors* $\mathbf{b}_0$ and $\mathbf{b}_1$ used to represent $\{x(n)\}$ by a weighted linear combination, and the *weights*, or *transform* coefficients $5/\sqrt{2}$ and $-1/\sqrt{2}$, can be expressed in the form

$$\theta(0) = \frac{5}{\sqrt{2}} = \frac{1}{\sqrt{2}}\Big[x(0)+x(1)\Big]; \quad \theta(1) = \frac{-1}{\sqrt{2}} = \frac{1}{\sqrt{2}}\Big[x(0)-x(1)\Big] \tag{12.6a}$$

The values of transform coefficients indicate the similarity (in sign as well as in relative magnitude) between the input $\{x(n)\}$ and the two basis vectors.

The input $\{x(n)\}$ can be recovered by an inverse operation

$$x(0) = \frac{1}{\sqrt{2}}\Big[\theta(0)+\theta(1)\Big]; \quad x(1) = \frac{1}{\sqrt{2}}\Big[\theta(0)-\theta(1)\Big] \tag{12.6b}$$

Note that $\theta(0)$ and $\theta(1)$ represent the sum and difference of the two input samples, and hence the powers of low frequency and high frequency components in $\{x(n)\}$. For minimizing the mean squared error in the reconstruction of the lowpass signal (12.3), one expects that the coefficient $\theta(0)$ is more crucial than $\theta(1)$ (Problem 12.5). This isolation of low and high frequency components is precisely the motivation for the *transform* representation (12.6).

Succeeding sections will show that transform representations such as (12.6) can be a basis for utilizing linear redundancies for the efficient coding of inputs with non-flat spectra, and that the upper bound for transformation *(transform coding)* gain is the same spectral flatness measure γ_x^2 that bounds *prediction gain.* A significant point of difference is that predictive systems utilize input dependencies in a recursive fashion while transform systems do so in a nonrecursive fashion, based on observations of N-blocks of $\{x(n)\}$. For speech and image sources, which can be described best by autoregressive models, predictive systems approach asymptotic performance faster (in terms of the order of prediction) than transform systems (in terms of order of transform, or block length).

In addition to providing components with variances that are different in general, transform operations such as (12.6) provide a representation in terms of uncorrelated coefficients. It is easy to see in fact that for a stationary random sequence $\{X(n)\}$ transformed by (12.6a),

$$E[\Theta(0)\Theta(1)] = E[X(0)^2] - E[X(1)^2] = 0 \tag{12.7}$$

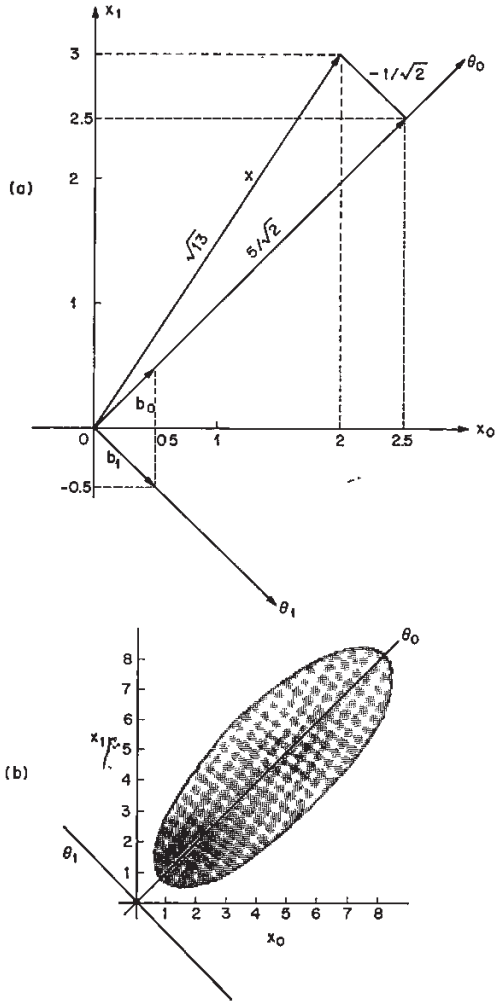
Recall that predictive encoding also aims to represent correlated inputs in terms of less correlated, ideally uncorrelated prediction error sequences.

A geometric interpretation. The decorrelation provided in a transform representation is well illustrated geometrically. Consider the sequence (12.3) as a

0236

point in the two-dimensional (x_0,x_1) domain of Figure 12.3. Also shown in the figure are *orthogonal* vectors $\mathbf{b}_0$ and $\mathbf{b}_1$ corresponding to the basis sequences of (12.4) and (12.5). We will see presently [in (12.22)] that these vectors are also *orthonormal*. The vectors $\mathbf{b}_0$ and $\mathbf{b}_1$ are along axes on which $x_0 = x_1$ and $x_0 = -x_1$, respectively. They define a new (θ_0,θ_1) domain obtained by rotating and reflecting the (x_0,x_1) domain. The input point (2,3) can be represented in the

Figure 12.3 Geometrical interpretations for (a) the transform representation of the input {2,3}; and (b) a population of positively correlated input sequences of length 2.



(θ_0,θ_1) domain by traversing $\theta(0) = +5/\sqrt{2}$ units in the $\mathbf{b}_0$ direction (starting from the origin), and traversing $\theta(1) = -1/\sqrt{2}$ unit in the $\mathbf{b}_1$ direction.

Let us now consider a population or ensemble of input sequences of length 2. Assume for illustration that each of these sequences exhibits a positive adjacent sample correlation as in Figure 12.2(a). The members of such an ensemble of $(x(0), x(1))$-pairs will occupy the shaded ellipse-like region of Figure 12.3(b). Points in this region have the property that their (x_0,x_1) coordinates are highly correlated, while the (θ_0,θ_1) coordinates are not. Given the θ_0 coordinate $\theta(0)$ of a sample point, one expects the $\theta(1)$ value to be smaller in magnitude, but one cannot predict the sign of $\theta(1)$. In other words, $x(1)$ can be greater or less than $x(0)$ with equal probability. In the specific example of (12.6a), $\theta(1)$ was of course negative.

Vector and marix notations. The following notation is very useful in representing transforms of arbitrary order N:

$$\boldsymbol{\theta} = \mathbf{A}\mathbf{x}\,; \quad \mathbf{x} = \mathbf{B}\boldsymbol{\theta}\,; \quad \mathbf{B} = \mathbf{A}^{-1} \tag{12.8}$$

where $\mathbf{A}$ is a *transform matrix* and $\mathbf{B}$ is the *inverse transform matrix* (Section 12.3).

Input and coefficient vectors are represented as $(1\times N)$ matrices for 1-D sequences

$$\mathbf{x}^T = \Big(x(0),x(1),\ldots,x(N-1)\Big)\,; \quad \boldsymbol{\theta}^T = \Big(\theta(0),\theta(1),\ldots,\theta(N-1)\Big)\,, \tag{12.9}$$

as $(N\times N)$ matrices for 2-D sequences, and as appropriate tensors for higher dimensions.

For $N = 2$ we will see (Section 12.5) that, most generally,

$$\mathbf{A} = \mathbf{B} = \begin{bmatrix} a & \sqrt{1-a^2} \\ \sqrt{1-a^2} & -a \end{bmatrix} \tag{12.10a}$$

For the important special case of $a = 1/\sqrt{2}$ (Problem 12.1), the inverse transform $\mathbf{B}$ equals $\mathbf{A}$:

$$\mathbf{A} = \mathbf{B} = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}. \tag{12.10b}$$

The basis vectors $\mathbf{b}_0$ and $\mathbf{b}_1$ are the columns of $\mathbf{B}$:

$$\mathbf{b}_0^T = (b_{00},b_{10}) = \left(\frac{1}{\sqrt{2}}\,,\,\frac{1}{\sqrt{2}}\right)\,; \quad \mathbf{b}_1^T = (b_{01},b_{11}) = \left(\frac{1}{\sqrt{2}}\,,\,\frac{-1}{\sqrt{2}}\right) \tag{12.11}$$

The inverse transform $\mathbf{x} = \mathbf{B}\boldsymbol{\theta}$ has an explicit interpretation

$$x(0) = b_{00}\theta(0) + b_{01}\theta(1)\,; \quad x(1) = b_{10}\theta(0) + b_{11}\theta(1) \tag{12.12}$$

or, the *basis vector* interpretation

$$\mathbf{x} = \theta(0)\mathbf{b}_0 + \theta(1)\mathbf{b}_1 \tag{12.13}$$

which tells us that $\mathbf{x}$ is the weighted sum of the two basis vectors, and that the weights are just the transform coefficients. In the example of Figure 12.2,

$$\mathbf{x} = \begin{bmatrix} 2 \\ 3 \end{bmatrix}; \quad \theta = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \begin{bmatrix} 2 \\ 3 \end{bmatrix} = \frac{1}{\sqrt{2}} \begin{bmatrix} 5 \\ -1 \end{bmatrix}; \tag{12.14a}$$

$$\mathbf{x} = \frac{5}{\sqrt{2}}\mathbf{b}_0 - \frac{1}{\sqrt{2}}\mathbf{b}_1 \tag{12.14b}$$

However, *transform coding* implies the use of quantized versions $u(i) = Q[\theta(i)]$ of the coefficients, and hence an output

$$\mathbf{y} = Q[\theta(0)]\mathbf{b}_0 + Q[\theta(1)]\mathbf{b}_1 = u(0)\mathbf{b}_0 + u(1)\mathbf{b}_1 \tag{12.14c}$$

which will be different from $\mathbf{x}$. This will cause a reconstruction error $\mathbf{r} = \mathbf{x} - \mathbf{y}$ (Section 12.3).

The example of $N = 2$ is too simple to be of practical significance in typical speech and image applications. But the discussion of this simple case has already provided significant insight. It has shown that Transform Coding is different in structure from Sub-band Coding but yet similar to it in the qualitative sense that different frequency components in the input are separated. The example of $N = 2$ has also shown the importance of bit allocation, and the fact that the channel input, as in the case of DPCM and SBC, is a sequence of quantities that are less correlated than the input samples themselves.

Section 12.3 will formalize an $N \times N$ transform in greater detail, and Section 12.4 will discuss variances, which leads in turn to the important issue of error-minimizing bit allocation. Sections 12.5 and 12.6 will discuss optimum (input decorrelating) and several suboptimum transforms. For the special case of $N = 2$, all of these transforms reduce to the one defined in (12.10). Section 12.7 will discuss adaptive bit allocation and resulting techniques for adaptive transform coding (ATC), and Section 12.8 will describe the effects of transmission errors.

12.3 $N \times N$ Transforms

Basic transform definitions have been introduced in Chapter 2 and in Sections 12.1 and 12.2. This section will provide a recapitulation, and also point out classes of one- and two-dimensional transforms that are important for digital coding of waveforms.

12.3.1 One-Dimensional Transforms

We extend the notation of Section 12.2 to define the Nth order linear transform of a one-dimensional sequence

$$\{x(n)\}; \quad n = 0,1,\ldots,N-1 \tag{12.15}$$

0239

The transform is given by

$$\theta(k) = \sum_{n=0}^{N-1} x(n)a(k, n) \qquad \text{for } k = 0,1,...,N-1 \tag{12.16a}$$

where $a(k,n)$ is a *forward transformation kernel*, and $\theta(k)$ are the *transform coefficients*. The inverse transform that recovers the input sequence is

$$x(n) = \sum_{k=0}^{N-1} \theta(k)b(k, n) \qquad \text{for } n = 0,1,...,N-1 \tag{12.16b}$$

where $b(k, n)$ is the *inverse transformation kernel* (see Appendix A).

In matrix notation (Appendix B),

$$\boldsymbol{\theta} = \mathbf{A}\mathbf{x}\,; \qquad \mathbf{x} = \mathbf{B}\boldsymbol{\theta}\,; \qquad \mathbf{B} = \mathbf{A}^{-1} \tag{12.17}$$

with $\mathbf{x}$ and $\boldsymbol{\theta}$ as in (12.9) and

$$\mathbf{A} = \{a(m, n)\}_{m, n=0,1,...,N-1}\,; \qquad \mathbf{B} = \{b(m, n)\}_{m, n=0,1,...,N-1} \tag{12.18}$$

For simplicity, we use the notation

$$\mathbf{B} = \{\mathbf{b}_k\}_{k=0,1,...,N-1}\,; \qquad \mathbf{b}_k = \{b(m, k)\}_{m=0,1,...,N-1} \tag{12.19}$$

where the $\mathbf{b}_k$ are *basis vectors*. We also note, as a generalization of (12.13), that

$$\mathbf{x} = \mathbf{B}\boldsymbol{\theta} = \sum_{k=0}^{N-1} \theta(k)\mathbf{b}_k \tag{12.20a}$$

Hence $\mathbf{x}$ is *the weighted sum of basis vectors, where the weights are just the values of the transform coefficients* ($\mathbf{x}$ can also be considered as a point in an N-dimensional $\boldsymbol{\theta}$-space, generalizing on Figure 12.2). In *transform coding*, output $\mathbf{y}$ will also be a weighted sum of basis vectors. But the weights will be quantized versions of $\theta(k)$ (Section 12.4):

$$\mathbf{y} = \sum_{k=0}^{N-1} u(k)\mathbf{b}_k; \qquad u(k) = Q[\theta(k)] \tag{12.20b}$$

The matrix $\mathbf{A}$ can be complex in general, as in some later subsections. For the time being, let $\mathbf{A}$ be real; the class of *orthogonal* transforms is defined by

$$\mathbf{A}^{-1} = \mathbf{A}^T \tag{12.21}$$

which implies that

$$\mathbf{A}^T\mathbf{A} = \mathbf{A}\mathbf{A}^T = \mathbf{I} \tag{12.22}$$

where $\mathbf{I}$ is the identity matrix of order N. A real matrix is orthogonal if and only

0240

if its columns and rows form an *orthonormal* set (Problem 12.1). From the above, we find that the inverse transform is just the transpose of **A**:

$$\mathbf{B} = \mathbf{A}^{-1} = \mathbf{A}^T \tag{12.23}$$

This implies that the basis vectors are rows of **A** (see Figure 12.4) and that

$$\mathbf{b}_i^T \cdot \mathbf{b}_j = \delta_{ij} \tag{12.24}$$

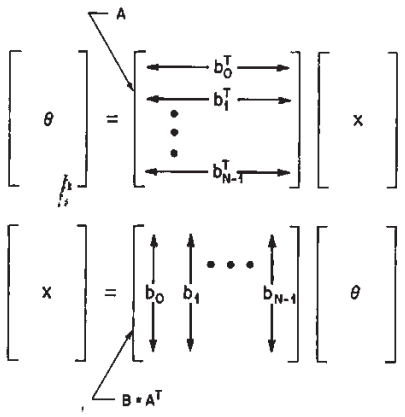
For example, with the basis vectors in (12.11),

$$\mathbf{b}_0^T \cdot \mathbf{b}_0 = \mathbf{b}_1^T \cdot \mathbf{b}_1 = 1\,; \quad \mathbf{b}_0^T \cdot \mathbf{b}_1 = \mathbf{b}_1^T \cdot \mathbf{b}_0 = 0 \tag{12.25}$$

Orthogonality is clearly a necessary property for basis vectors that are used to decompose an input into uncorrelated components in an N-dimensional space, as in Figure 12.2 for $N = 2$. *Orthonormality* of basis vector is a stronger property; it leads to transforms which are necessary in transform coding to make the average sum of the variances of the elements of $\boldsymbol{\theta}$ equal to σ_x^2, the variance of the elements of **x**; this also means that the average reconstruction error variance equals the error variance introduced in the quantization of transform coefficients (Section 12.3).

Figure 12.5 shows basis vectors for several one-dimensional transforms of order $N = 8$. In each of the four sets of basis vectors, all vectors except those in the first row are zero-mean sequences. The basis vectors in the first row are constant-amplitude sequences in all cases except the *Karhunen Loeve Transform* (KLT). The basis vectors in the *Discrete Fourier Transform* (DFT) are simply sines and cosines (fundamental frequency and multiples thereof) appropriate to $N = 8$. The basis vectors in the *Discrete Walsh Hadamard Transform* (DWHT) are square waves of different *sequencies* (Section 12.6), while the *Discrete Cosine Transform* (DCT) basis vectors are cosine functions with special phase and amplitude patterns (Section 12.6). The basis vectors in DFT, DWHT and DCT are input-signal-

Figure 12.4 Transform matrix **A**, inverse $\mathbf{B} = \mathbf{A}^{-1} = \mathbf{A}^T$, and basis vectors $\mathbf{b}_k$.



0241

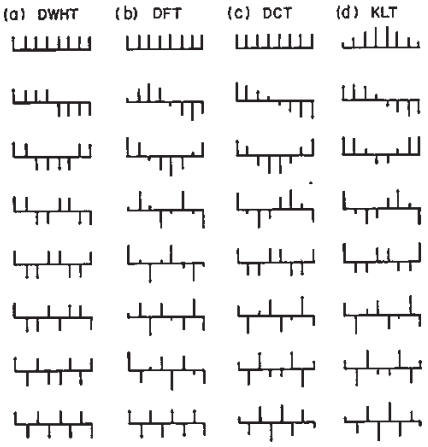


Figure 12.5 Basis vectors for $N = 8$ in (a) DWHT, (b) DFT, (c) DCT and (d) KLT. The input-dependent KLT set in (d) is derived from an AR(8) modeling of long-time-averaged speech statistics [Zelinski and Noll, 1975. Reprinted with permission].

independent. By contrast, the KLT basis vectors, also called *eigenvectors*, are defined by input statistics, as described in Section 12.5. In the specific example of Figure 12.5, the KLT basis vectors are appropriate to long-term speech statistics. Basic vectors for other values of N can be found in coding literature (for example, [Campanella and Robinson, 1971]).

Input-dependent transforms are the hardest to implement, but they also have the best input-decorrelating and variance-ordering properties, as explained in Section 12.4. KLT basis vectors also exhibit the greatest possible resemblance to waveform segments that are typical of the input class whose statistics are used in the KLT derivation. This will be reflected in the fact that the dimensionality N needed for a given representation error is least when the representation is in terms of KLT basis vectors (eigenvectors). Section 12.4 will discuss representations based on retaining only a small fraction of eigenvectors for the reconstruction in (12.16b).

12.3.2 Two-Dimensional Transforms

Let us begin with a straightforward generalization of (12.16):

$$\theta(k, l) = \sum_{m=0}^{N-1} \sum_{n=0}^{N-1} x(m, n)\, a(k, l, m, n)\ ;$$

$$x(m, n) = \sum_{k=0}^{N-1} \sum_{l=0}^{N-1} \theta(k, l)\, b(k, l, m, n) \qquad (12.26)$$

The above equations describe the transform of N^2-point sequences with transformation kernels $a(\cdot)$ and $b(\cdot)$ that are described by N^4 elements in general. For a 2-dimensional image input, the N^2 values of $x(m,n)$ are usually the elements

0242

of a square array, a subimage of size $N \times N$. Typical arrays in image coding use $N = 4$, 8 or 16. The partitioning into subimages is particularly efficient in cases where correlations are localized to neighboring pixels, and where structural details tend to cluster. Equation (12.26) does not preclude the arrangement of the same N^2 input points as a single $(1 \times N^2)$ vector.

Separable Two-Dimensional Transforms. The image transforms that we will study will invariably assume a $N \times N$ square sub-image. In addition, we will only consider the simple case where the transform kernels $a(\cdot)$ and $b(\cdot)$ are *separable* into kernels signifying separate horizontal (row) and vertical (column) operations:

$$a(k, l, m, n) = a_v(k, m)\, a_h(l, n)\ ;$$
$$b(k, l, m, n) = b_v(k, m)\, b_h(l, n) \tag{12.27}$$

The two-dimensional transform to obtain $\theta(k,l)$ [Equation (12.26)] can now be conveniently performed in two steps, each of which involves a one-dimensional transform operation. The first step uses $a_h(\cdot)$ for operation on row m to obtain the lth transform coefficient $\theta[m,l)$ [compare with (12.16a)], while the second step uses $a_v(\cdot)$ to provide the one-dimensional transform of column l:

$$\theta(k, l) = \sum_{m=0}^{N-1} a_v(k, m) \sum_{n=0}^{N-1} x(m, n)\, a_h(l, n) = \sum_{m=0}^{N-1} a_v(k, m)\, \theta(m, l) \tag{12.28}$$

$\mathbf{X}$ and $\boldsymbol{\Theta}$ are arrays which have as their elements $x(m, n)$ and $\theta(k, l)$, respectively. We may write

$$\mathbf{A}_h = \{a_h(m,n)\}_{m,n=0,1,\ldots,N-1}; \qquad \mathbf{A}_v = \{a_v(m,n)\}_{m,n=0,1,\ldots,N-1} \tag{12.29}$$

so that

$$\boldsymbol{\Theta} = \mathbf{A}_v \mathbf{X} \mathbf{A}_h^T \tag{12.30}$$

For *symmetrical* kernels,

$$\mathbf{A}_v = \mathbf{A}_h = \mathbf{A}\ ; \qquad \boldsymbol{\Theta} = \mathbf{A}\mathbf{X}\mathbf{A}^T \tag{12.31}$$

and with $\mathbf{A}^{-1} = \mathbf{A}^T$,

$$\mathbf{X} = \mathbf{A}^T \boldsymbol{\Theta} \mathbf{A} \tag{12.32}$$

Note that $\mathbf{A}$ is simply the 1-D transform (12.31) and that the matrix operation of (12.32) is possible only with separable transforms. With general non-separable transforms, 2-D operations would involve tensor operations, rather than the matrix operations of (12.32). Also, as a result of a separable $b(\cdot)$ kernel, image $\mathbf{X}$ can be expressed as a superposition of *basis images* $\mathbf{B}_{kl}$ [see (12.26)]:

$$\mathbf{X} = \sum_{k=0}^{N-1} \sum_{l=0}^{N-1} \theta(k, l) \mathbf{B}_{kl}\ ; \qquad \mathbf{B}_{kl} = \mathbf{b}_k \mathbf{b}_l^T \tag{12.33a}$$

The second part of (12.33a) is a good working definition for a separable transform.

0243

The basis image is the two-dimensional counterpart of the basis vector in (12.20a). In *transform coding*, the reconstructed image $\mathbf{Y}$ will be superposition of basis images $\mathbf{B}_{kl}$ which have been weighted by quantized versions of $\theta(k,l)$:

$$\mathbf{Y} = \sum_{k=0}^{N-1}\sum_{l=0}^{N-1} Q[\theta(k,l)]\mathbf{B}_{kl} = \sum_{k=0}^{N-1}\sum_{l=0}^{N-1} u(k,l)\mathbf{B}_{kl} \tag{12.33b}$$

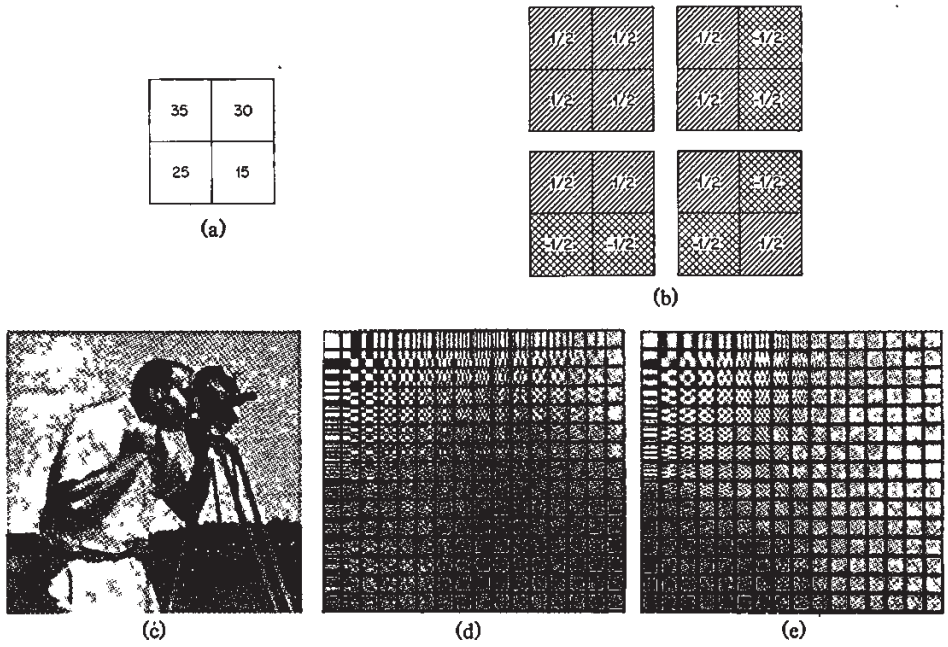
Figure 12.6(a) is an example of a (2×2) image [Problem 12.6(a)] and Figure 12.6(b) is a separable (2×2) transform with a set of four basis images $\mathbf{B}_{00}$, $\mathbf{B}_{01}$, $\mathbf{B}_{10}$ and $\mathbf{B}_{11}$. By assigning amplitude levels of 1/2 and −1/2 with single-hatched and cross-hatched areas in Figure 12.6(b), the 2×2 basis images $\mathbf{B}_{00}$, $\mathbf{B}_{01}$, $\mathbf{B}_{10}$ and $\mathbf{B}_{11}$ are seen to be those in (12.34):

$$\mathbf{B}_{00} = \frac{1}{2}\begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix} \qquad \mathbf{B}_{01} = \frac{1}{2}\begin{bmatrix} 1 & -1 \\ 1 & -1 \end{bmatrix}$$

$$\mathbf{B}_{10} = \frac{1}{2}\begin{bmatrix} 1 & 1 \\ -1 & -1 \end{bmatrix} \qquad \mathbf{B}_{11} = \frac{1}{2}\begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix} \tag{12.34}$$

Note that the basis images are given by $\mathbf{B}_{kl} = \mathbf{b}_k\mathbf{b}_l^T$, as in (12.11) and (12.30).

Figure 12.6 2-D transforms: (a) is a 2×2 input and (b) is a set of four basis images that may be used to represent (a); (c) is the image of "cameraman," which may be represented by sets of (16×16) basis images in (d) DWHT or (e) KLT [Wintz;©1972, IEEE].



The intensity levels of binary images are usually assigned values of 1 (for *white*) and 0 (for *black*). This is in fact the explicit notation of Figure 2.2 and Chapter 10. However, if binary intensities are associated with basis images, the two intensity levels are 1 and −1, as in (12.34).

Higher order transforms are useful in the representation of real images such as that in Figure 12.6(c) (Problem 12.2). Basis images of (16×16) arrays are shown in Figure 12.6(d) and (e). If the (256×256) image of Figure 12.6(c) is divided into 256 sub-images of size (16×16) each, these subimages can be described as a superposition of the basis images in Figures 12.6(d) or (e). The basis images in DWHT have black-or-white components as in Figure 12.6(b), while the basis images in KLT have elements representing all shades of gray. In each superposition, the intensities of the basis images add in proportion to the weighting provided by the respective transform coefficient (see 12.33a). Since the values of these coefficients are not constrained to be binary, the above process can generate *gray* levels of intensity even if the sub-images have only two levels of intensity (white and black), as in DWHT.

Consider next the DWHT basis vectors $\mathbf{b}_k$ of length $N = 8$ in Figure 12.5. We can obtain the first $8\times8 = 64$ basis images of Figure 12.12 [a lower-dimensional version of Figure 12.6(d)] by using these 8 vectors in $\mathbf{b}_k\mathbf{b}_l^T$ (12.33) (Problem 12.3).

KLT basis images are also called *eigenimages*. They are specific to a given description of input statistics. In the example of Figure 12.6(e), they are derived from a two-dimensional $AR(2)$ modeling of the input process in Figure 12.6(c). This model uses measured correlation coefficients of 0.883 and 0.780 in the horizontal and vertical directions. We will see shortly that for a given input process, out of all basis image sets, the eigenimage set leads to the smallest reconstruction error for a given dimensionality ($N\times N$) of representation, i.e., a given number of basis images. Section 12.5 will demonstrate image reconstructions based on retaining only a small fraction of eigenimages in the representation (12.33).

12.3.3 Complex Transforms

Complex kernels and coefficients will be treated at some length in the DFT discussion of Section 12.5. For completeness, however, we note the following generalization rightaway. *Unitary* transforms (Appendix B) are described by

$$\mathbf{B} = \mathbf{A}^{-1} = \mathbf{A}^{*T}; \quad \boldsymbol{\theta} = \mathbf{A}\mathbf{x}; \quad \mathbf{x} = \mathbf{A}^{*T}\boldsymbol{\theta} \tag{12.35}$$

and they are the counterpart of orthogonal transforms in the real-valued case. As in that case, columns and rows of the $\mathbf{A}$ and $\mathbf{B}$ matrices in (12.35) are orthonormal. Separable two-dimensional transforms with symmetrical kernels are described in general by

$$\boldsymbol{\Theta} = \mathbf{A}\mathbf{X}\mathbf{A}^{*T}; \quad \mathbf{X} = \mathbf{A}^{*T}\boldsymbol{\Theta}\mathbf{A} \tag{12.36}$$

12.4 Optimum Bit Allocation and Zonal Sampling

This section will show two results basic to Transform Coder (TC) performance. We first show that for orthogonal (in general, unitary) transforms, the reconstruction error variance for an input block is the sum of error variances in the coding (quantization) of individual transform coefficients. We then show that Transform Coding, which is based on the quantization of these coefficients, can provide an *SNR* gain over PCM, denoted by G_{TC}, whose maximum value is equal to the ratio of arithmetic to geometric means of transform coefficient variances. Sections 12.5 and 12.6 will describe transforms that seek to make this ratio as large as possible. The maximum gain over PCM is achieved in the case of *optimum bit allocation* for the quantization of transform coefficients. We will also discuss one example of suboptimum bit allocation, called *zonal sampling*.

12.4.1 Optimum Bit Allocation

Reconstruction Error. We will begin with a one-dimensional input for simplicity. Refer to the Transform Coder block diagram of Figure 12.1. We arrange N successive signal samples $\{x(n)\}$; $n=0,1,...,N-1$ into a block or vector $\mathbf{x}$. The vector $\mathbf{x}$ is linearly transformed using an orthogonal (in general, unitary) matrix $\mathbf{A}$. The elements of $\boldsymbol{\theta} = \mathbf{Ax}$, i.e., the transform coefficients, are *independently* quantized using a bank of N quantizers Q_i; $i = 0,1,...,N-1$. The vector $\mathbf{u} = Q[\boldsymbol{\theta}]$ of quantized coefficients is transmitted to the receiver and its received version $\mathbf{v}$ (which equals $\mathbf{u}$ with error-free transmission) is inverse-transformed using $\mathbf{A}^{-1}$:

$$\mathbf{y} = \mathbf{A}^{-1}\mathbf{u} = \mathbf{A}^T\mathbf{u} = \sum_{k=0}^{N-1} u(k)\mathbf{b}_k \tag{12.37}$$

The elements of $\mathbf{y}$ are the reconstructed output samples. It can be shown [Huang and Schultheiss, 1963] that $\mathbf{A}^{-1}$, which was assumed for $\mathbf{B}$ in (12.17), is in fact the best inverse transform in the mse sense for Gaussian inputs and Max-quantizers.

We will now compute error variances. In order to do so, we assume that $\mathbf{x}$ is taken from a zero-mean stationary scalar source. Successive vectors thus constitute a random vector source $\mathbf{X}$ whose elements are random variables $X(n)$; $n = 0,1,...,N-1$.

As noted in Section 12.2, orthogonal transforms are variance-preserving. We will hence show that if $\mathbf{A}$ is orthogonal ($\mathbf{A}^{-1}=\mathbf{A}^T$), the average sum of variances $E[\Theta^2(k)] = \sigma_k^2$ of the transform coefficients $\theta(k)$ equals the variance of the input:

$$\begin{aligned}\frac{1}{N}\sum_{k=0}^{N-1}\sigma_k^2 &= \frac{1}{N}\sum_{k=0}^{N-1}E[\Theta^2(k)] = \frac{1}{N}E[\boldsymbol{\Theta}^T\boldsymbol{\Theta}] = \frac{1}{N}E[\mathbf{X}^T\mathbf{A}^T\mathbf{A}\mathbf{X}] \\ &= \frac{1}{N}E[\mathbf{X}^T\mathbf{X}] = \frac{1}{N}\sum_{k=0}^{N-1}E[X^2(k)] = \frac{1}{N}\sum_{k=0}^{N-1}\sigma_x^2 = \sigma_x^2\end{aligned} \tag{12.38}$$

We will next relate the total reconstruction error between input $\mathbf{X}$ and output $\mathbf{Y}$ to the errors introduced by the individual quantizers Q_i. Let

0246

$$\sigma_{rk}^2 = E[(X(k)-Y(k))^2]; \quad k = 0,1,...,N-1 \tag{12.39}$$

be the mse of element k of $\mathbf{X}$. The average reconstruction error variance is

$$\sigma_r^2 = \frac{1}{N}\sum_{k=0}^{N-1}\sigma_{rk}^2 = \frac{1}{N}E[(\mathbf{X}-\mathbf{Y})^T(\mathbf{X}-\mathbf{Y})] \tag{12.40}$$

Clearly, $\mathbf{r} = \mathbf{x} - \mathbf{y}$ is the N-block of error samples and (12.40) is the variance, as averaged over that block. Using (12.37), the corresponding relation $\mathbf{x} = \mathbf{A}^{-1}\boldsymbol{\theta}$ for unquantized quantities, and assuming that $\mathbf{A}^{-1} = \mathbf{A}^T$, (12.40) can be rewritten as

$$\sigma_r^2 = \frac{1}{N}E[(\boldsymbol{\Theta}-\mathbf{U})^T(\boldsymbol{\Theta}-\mathbf{U})] = \frac{1}{N}E[\mathbf{Q}^T\mathbf{Q}] = \frac{1}{N}\sum_{k=0}^{N-1}\sigma_{qk}^2 = \sigma_q^2 \tag{12.41}$$

where $\boldsymbol{q} = \boldsymbol{\theta} - \mathbf{u}$ is the block of quantization errors, and $\sigma_{qk}^2 = E[(\Theta(k)-U(k))^2]$. In other words, *for orthogonal transforms, the reconstruction error variance in TC equals that introduced by the set of coefficient quantizers.*

We may also describe the reconstruction error vector $\mathbf{r}$ as a superposition of basis vectors weighted by the corresponding quantization errors. From (12.20a) and (12.37),

$$\mathbf{r} = \sum_{k=0}^{N-1} q(k)\mathbf{b}_k \tag{12.42}$$

The result $\sigma_r^2 = \sigma_q^2$ can also be established from (12.42).

The results in (12.40) and (12.41) can be straightforwardly extended to include complex entries in $\mathbf{A}$ and to two-dimensional signals where (12.40) and (12.41) generalize to averages taken over N^2 elements.

The error components in (12.41) can be explicitly written, as in Chapter 4, in the form

$$\sigma_{qk}^2 = \epsilon_{qk}^2\sigma_k^2 \approx \epsilon_\epsilon^2 2^{-2R_k}\sigma_k^2 \tag{12.43}$$

where $\sigma_k^2 = E[\Theta^2(k)]$ is the variance of the input to quantizer k, and ϵ_{qk}^2 is the corresponding quantizer performance factor. This factor depends on the pdf of the input (kth transform coefficient) and on the quantizer characteristic, especially on the number of levels therein. However, ϵ_{qk}^2 is approximately reduced by a factor of four if the number of levels is doubled, which leads to the approximation in (12.43). The quantity ϵ_ϵ^2 can be regarded as a variable correction factor that takes into account the performance of practical quantizers. We recall from Chapter 4 that minimum values of ϵ_ϵ^2, for sufficiently fine quantization without entropy coding, are respectively 1.0, 2.7, 4.5, and 5.7 for Max quantizers with uniform, Gaussian, Laplacian and gamma distributions at the input. We also recall that ϵ_ϵ^2 values less than 1 can be realized by entropy coding under certain conditions. We shall use the approximation in (12.43) in the rest of this chapter, although it is quite coarse for low values of R_k.

Rewriting (12.43) as

$$R_k = \frac{1}{2}\log_2\epsilon_\epsilon^2 + \frac{1}{2}\log_2\frac{\sigma_k^2}{\sigma_{qk}^2}; \quad k = 0,1,...,N-1 \tag{12.44}$$

provides an expression for the rate R_k (bits/sample) needed for coefficient $\theta(k)$ of variance σ_k^2 if an average MSE σ_{qk}^2 is not to be exceeded. The second term in the right-half of (12.44) is a minimum rate from rate distortion bounds for statistically independent Gaussian random variables (Appendix D), and the first term is the above mentioned correction for practical quantizers.

Values of σ_k^2 are in general different and so are the individual rates R_k. In fact, the problem of optimizing a Transform Coding scheme can be stated as that of finding an orthogonal (in general, unitary) matrix, and then of finding a distribution (allocation) of bits R_k such that the average coefficient error variance

$$\sigma_q^2 = \frac{1}{N} \sum_{k=0}^{N-1} \sigma_{qk}^2 \tag{12.45}$$

is minimized with the constraint of a given *average* bit rate

$$R = \frac{1}{N} \sum_{k=0}^{N-1} R_k = \text{constant} \tag{12.46}$$

A later subsection will generalize (12.45) to include frequency-weighted (k-weighted) error criteria.

Bit Allocation for $N = 2$. Assuming an appropriate transform, the problem is to minimize

$$\sigma_r^2 = \sigma_q^2 = \frac{1}{2} (\sigma_{q0}^2 + \sigma_{q1}^2) \tag{12.47}$$

given that

$$R_0 + R_1 = 2R \tag{12.48}$$

The problem reduces to one of minimizing σ_r^2 with respect to R_0 because $R_1 = 2R - R_0$. Using (12.43) and (12.47),

$$\sigma_r^2 = \frac{1}{2} \epsilon_*^2 [2^{-2R_0} \sigma_0^2 + 2^{2R_0 - 4R} \sigma_1^2] \tag{12.49}$$

To minimize (12.49) using bit allocation, we set its (partial) derivative with respect to R_0 to zero:

$$\frac{\partial \sigma_r^2}{\partial R_0} = 2[-\sigma_0^2 \, 2^{-2R_0} + \sigma_1^2 \, 2^{2R_0 - 4R}] = 0 \tag{12.50}$$

so that

$$\sigma_0^2 / 2^{2R_0} = \sigma_1^2 / 2^{2R_1} \; ; \quad 2R_0 = \log_2 \sigma_0^2 + c \; ; \quad 2R_1 = \log_2 \sigma_1^2 + c \tag{12.51}$$

where c is a constant. From (12.48),

0248

$$R = \frac{1}{2}\log_2(\sigma_0\sigma_1) + \frac{c}{2} \tag{12.52}$$

We express c in terms of R using (12.52), and use it to rewrite (12.51) as

$$R_0 = R + \frac{1}{2}\log_2\frac{\sigma_0}{\sigma_1}; \quad R_1 = R - \frac{1}{2}\log_2\frac{\sigma_0}{\sigma_1} \tag{12.53}$$

which expresses the desired *bit allocation*. The second derivative of σ_q^2 with respect to R_0 is always positive so that the extremum provided by (12.53) is indeed a minimum (not maximum) value of σ_q^2. (Also see Problem 12.5 for a numerical example).

Inspection of (12.51) and (12.43) reveals that

$$\sigma_{q_0}^2 = \sigma_{q_1}^2 \tag{12.54}$$

so that the bit allocation procedure that minimizes σ_r^2 also provides equal quantization errors in the quantizing of the individual transform coefficients. This implies a reconstruction noise spectrum that is independent of coefficient index k. [In DFT (Section 12.6.2) this provides a reconstruction noise spectrum that is flat as a function of frequency.] Finally, setting (12.53) in (12.43),

$$\sigma_{q1}^2 = \epsilon_*^2 2^{-2R}(\sigma_0\sigma_1)$$

and because of (12.43), (12.47) and (12.53),

$$\sigma_{r,TC}^2 = \sigma_q^2 = \sigma_{q0}^2 = \sigma_{q1}^2 = \epsilon_*^2 2^{-2R}(\sigma_0\sigma_1) \tag{12.55}$$

We will now compare this with the average reconstruction error in PCM [R-bit quantization of $x(n)$]. Assuming an identical quantizer performance factor ϵ_*^2,

$$\sigma_{r,PCM}^2 = \sigma_q^2 = \epsilon_*^2\, 2^{-2R}\sigma_x^2 \tag{12.56}$$

Also, from (12.38), $\sigma_x^2 = (\sigma_0^2 + \sigma_1^2)/2$, so that the gain due to Transform Coding with $N = 2$ is

$$\max\{{}^2G_{TC}\} = \frac{\sigma_{r,PCM}^2}{\sigma_{r,TC}^2} = \frac{(\sigma_0^2+\sigma_1^2)/2}{(\sigma_0^2\sigma_1^2)^{1/2}} = \frac{\sigma_x^2}{(\sigma_0^2\sigma_1^2)^{1/2}}, \tag{12.57}$$

the *ratio of arithmetic to geometric means of the coefficient variances* σ_0^2 *and* σ_1^2. This is a crucial result in Transform Coding theory and it will be generalized to $N > 2$ in the next subsection. Unless otherwise mentioned, maximum values of G_{TC} imply maximization with respect to bit allocation, as in (12.57), and not maximization with respect to linear transform type.

Bit Allocation for $N > 2$. The problem once again is to minimize, by means of an optimum bit allocation R_k; $k = 0,1,\ldots,N-1$, the reconstruction error (12.45) subject to the constraint (12.46). Using the Lagrange multiplier method,

$$\frac{\partial}{\partial R_k}\left[\sigma_q^2 - \lambda\left(R - \frac{1}{N}\sum_{k=0}^{N-1} R_k\right)\right] = 0; \quad k = 0, 1, \ldots, N-1 \tag{12.58}$$

with

$$\sigma_q^2 = \frac{1}{N}\sum_{k=0}^{N-1} \sigma_{qk}^2 = \frac{1}{N}\epsilon^2 \sum_{k=0}^{N-1} 2^{-2R_k}\sigma_k^2 , \tag{12.59}$$

we arrive at an optimum bit allocation given by

$$R_k = R + \frac{1}{2}\log_2 \frac{\sigma_k^2}{\left[\prod_{j=0}^{N-1} \sigma_j^2\right]^{1/N}} \tag{12.60}$$

that depends only on the distribution of coefficient variances, as with $N = 2$. Clearly, (12.60) reduces to (12.53) for that special case. Note, from both of these equations, that the number of quantizer levels 2^{2R_k} is proportional to the variance σ_k^2 [Huang and Schultheiss, 1963] [Segall, 1976]. We will shortly note a graphical interpretation of (12.60) in Figure 12.8. This will also suggest a simple way of implementing the optimal bit allocation.

As with $N = 2$, we find that the bit allocation (12.60) also implies identical error variances in coefficient quantization:

$$\min\{\sigma_r^2\} = \min\{\sigma_q^2\} = \sigma_{qk,opt}^2 = \epsilon^2 2^{-2R}\left[\prod_{j=0}^{N-1} \sigma_j^2\right]^{1/N}; \quad k = 0,1,\ldots,N-1 \tag{12.61}$$

The above result can be easily interpreted for the example of uniform quantizers operating without overload. Identical quantization error variances result if *all quantizers have the same step size* Δ, and hence the same $\sigma_q^2 = \Delta^2/12$. But then a coefficient with higher variance (Figure 12.7b) needs *more levels* than a coefficient of lower variance (Figure 12.7a), for a match in each case, of dynamic ranges of quantizer and input; and this is indeed guaranteed by the bit allocation result (12.60).

An orthogonal transform implies [see (12.38)] that

$$\sigma_x^2 = \frac{1}{N}\sum_{k=0}^{N-1} \sigma_k^2 \tag{12.62}$$

so that the gain over PCM is

$$\max\{{}^N G_{TC}\} = \frac{\sigma_{r,PCM}^2}{\min\{\sigma_{r,TC}^2\}} = \frac{\frac{1}{N}\sum_{k=0}^{N-1} \sigma_k^2}{\left[\prod_{k=0}^{N-1} \sigma_k^2\right]^{1/N}}; \tag{12.63a}$$

$$SNR\,|_{TC}(\text{dB}) = SNR\,|_{PCM}(\text{dB}) + 10\log {}^N G_{TC} \tag{12.63b}$$

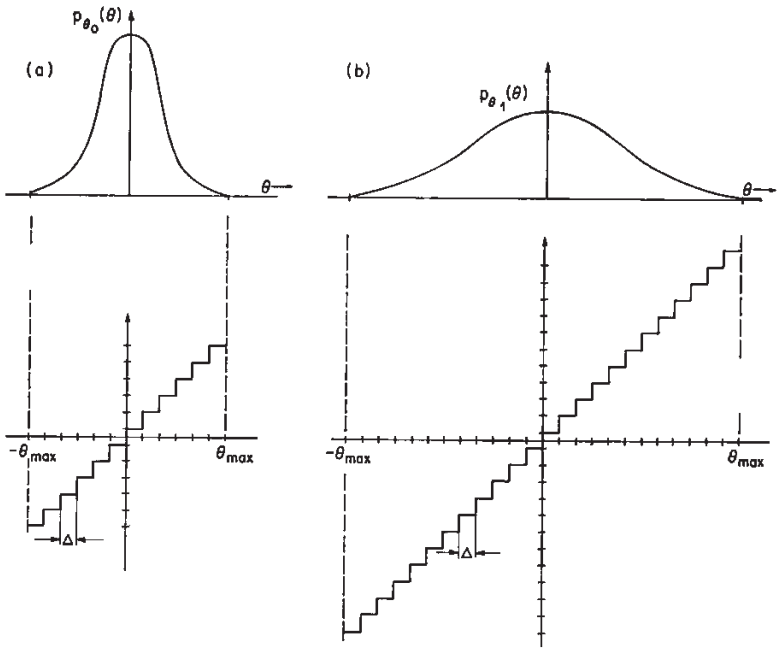


Figure 12.7 Uniform quantization of coefficients (a) θ_0 and (b) θ_1 without overload. The identical step size Δ ensures identical error variances σ_{q0}^2 and σ_{q1}^2, and the differing numbers of quantization levels ensure a match, in each case, of dynamic ranges of quantizer and input.

If we retain the possibility of different quantization performance factors, the above expression will be modified by a term $\epsilon_{\theta,PCM}^2/\epsilon_{\theta,TC}^2$. For example, if the effect of transforming is one of changing a uniform distribution to one that is Laplacian, the effect on G_{TC} would be a factor of (1/4.5), or about 6.5 dB less than what an arithmetic-to-geometric mean ratio in (12.63) would suggest. The situation of uniformly distributed $X(k)$ and Laplacian $\Theta(k)$ is a good model for image coding (however, see [Murakami, Hatori and Yamamoto, 1982]). Although each $\theta(n)$ is a weighted sum of N values of $x(n)$; $n = 0$ to $N-1$, the pdf of the transform coefficients tends to be non-Gaussian because the $x(n)$-components in the sum are not statistically independent. At any rate, the 6.5 dB loss above relates to the fact that inputs with a nonuniform pdf are more difficult to quantize than those with a uniform pdf. There are two cases where $\epsilon_{\theta,PCM}^2 = \epsilon_{\theta,TC}^2$. One is if $X(k)$ is Gaussian, in which case $\Theta(k)$ is Gaussian as well. The other case is if quantization is logarithmic; in this case the pdf is irrelevant.

It is interesting to compare (12.63) to the maximum gain in DPCM (6.19)

$$\max\{G_{DPCM}\} = \sigma_x^2/\eta_x^2 = (\gamma_x^2)^{-1} \tag{12.64}$$

where σ_x^2 is identical to the arithmetic mean in (12.63) assuming unitary

transforms, η_x^2 is the minimum prediction error variance and γ_x^2 is the spectral flatness measure. We will show in Section 12.5 that

$$\max\{G_{TC}\} = \max\{G_{DPCM}\} \tag{12.65}$$

where the first maximum is over all bit allocations and transform types, and the second maximum is over all prediction algorithms. The maximum values are also those reached with infinite values, in general, for length of transform and order of prediction.

The Non-Negative Integer Constraint for R_k. Practical quantizers for coefficient quantization have the constraint of having to use non-negative integers for the number of bits R_k obtained in the bit allocation procedure (12.60) (Problems 12.5 and 12.7). A weaker but generally less practical constraint is to allow values of R_k for which corresponding values of 2^{R_k}, the number of quantizer *levels*, is an integral value.

Particularly significant is the possibility of negative R_k values for coefficients whose variances are sufficiently smaller than the geometric mean, as given by (12.60). A typical bit reassignment procedure consists in setting $R_k = 0$ in these cases, implying no transmission of the corresponding coefficient, and a corresponding typical error of $\sigma_{qk}^2 = \sigma_k^2$. Clearly, when negative R_k values are increased to zero, the nonzero values of R_j; $j \neq k$ will also have to be altered to maintain a given value of R. Several re-optimization procedures are possible [Huang and Schultheiss, 1963] [Wintz, 1972] [Zelinski and Noll, 1975] [Segall, 1976]. In some of these, use is also made of the fact that R_k values exceeding an appropriate $R_{\max}$ are unwarranted from perceptual considerations, leading to a second type of reassignment, that of surplus quantizer bits. These bits can, of course, be used to transmit coefficients that would otherwise never be coded, or merely to round negative R_k values to zero.

Zero-level quantization of transform coefficients ($R_i = 0$) is reminiscent of an information theoretical result, the RDF for i.i.d. Gaussian samples given by

$$R_k = \max\{0, (1/2)\log_2(\sigma_k^2/D)\} \tag{12.66}$$

and the *water-filling procedure* (Figure D.4) whereby input (spectral) components lower in variance than an average distortion D are excluded from transmission. The points of difference are that (a) Gaussian samples cannot be quantized with $(1/2)\log_2(\sigma_k^2/D)$ bits, but with a number of bits defined by a practical quantizer, and that (b) water-filling procedures in Transform Coding are generally not implemented on eigenvalues (coefficients of the optimum KLT, Section 12.5), but on coefficients from suboptimal transformations (Section 12.6).

A Graphical Interpretation. We conclude this section with a graphical description of a practical bit allocation rule that is based on adjusting the value of σ_q^2 such that the sum of R_i values equals NR. The dashed horizontals in Figure 12.8(a) represent allocation thresholds κ_k for choosing the bit allocation R_k. For example, if $\log_2\sigma_k^2$ lies between κ_3 and κ_4, we assign $R_k = 4$. The thresholds are spaced 6 dB

0252

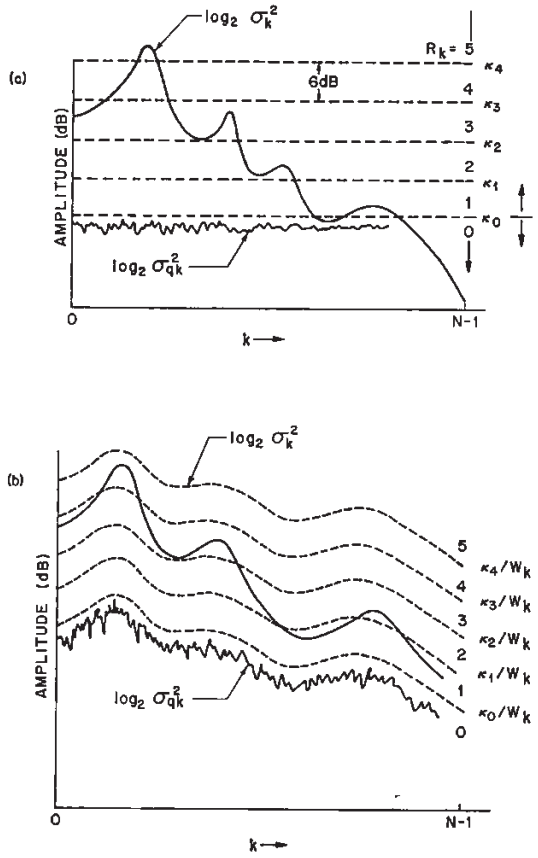


Figure 12.8 A graphical illustration of bit allocation for obtaining (a) white and (b) frequency-weighted quantization noise spectra [Tribolet and Crochiere;©1979, IEEE].

apart. Thus for every 6 dB that σ_k^2 is increased, one more bit is added to the quantizer. Clearly, the noise also remains flat across k values ($\sigma_{qk}^2 = \sigma_{ql}^2$). Two exceptions to the rule occur. Values of $\log_2 \sigma_k^2$ below κ_0 are assigned zero (not negative) bits in this example, and all values of $\log_2 \sigma_k^2$ above, say κ_4, are assigned $R_{\max} = 5$ bits in this example. If the resulting total number of bits assigned in this way is less (or greater) than NR, then the levels of the thresholds κ_k (proportional to σ_{qk}^2) are uniformly decreased (or increased), respectively, while always maintaining the 6 dB spacings. The process is repeated until $\sum R_k = NR$. A somewhat similar procedure for bit allocation is described in [Ramstad, 1982].

White Noise Property of the Reconstruction Error. We have shown that the variance σ_r^2 of the reconstruction error is

$$\sigma_r^2 = \frac{1}{N} E[(\boldsymbol{\theta}-\mathbf{U})^T(\boldsymbol{\theta}-\mathbf{U})] = \frac{1}{N} E[\mathbf{Q}^T\mathbf{Q}] = \sigma_q^2 \tag{12.67}$$

provided that $\mathbf{A}^T\mathbf{A} = \mathbf{I}$ [see (12.41)]. We show now that the individual reconstruction errors for $x(n)$ samples within the input N-blocks are not correlated, i.e., the reconstruction error is white noise as in PCM or DPCM. The block of quantization errors $\mathbf{Q}$ is inversely transformed at the decoder to $\mathbf{A}^T\mathbf{Q}$. Therefore, the covariance matrix of the reconstruction error is

$$\mathbf{R}_{rr} = E[\mathbf{A}^T\mathbf{Q}(\mathbf{A}^T\mathbf{Q})^T] = \mathbf{A}^T E[\mathbf{Q}\mathbf{Q}^T]\mathbf{A} = \mathbf{A}^T\mathbf{R}_{qq}\mathbf{A} \tag{12.68a}$$

$\mathbf{R}_{qq}$ has as its elements the crosscorrelation between quantization errors and these crosscorrelations can be considered to be zero since even for correlated samples, quantization errors tend to be much less correlated. This was illustrated in Figure 4.24(b) for the example of an AR(1) input to the quantizer. In the present context, the transform coefficients are much less correlated than the inputs (and not correlated at all in the case of an optimum KLT transform; see Section 12.5). Hence $\mathbf{R}_{qq}$ is diagonal. In the case of an *optimum* bit allocation (12.61) we have seen that all quantization error variances are to be of the same value σ_q^2. Hence

$$\mathbf{R}_{qq} = \sigma_q^2\mathbf{I} \quad \text{and} \quad \mathbf{R}_{rr} = \sigma_q^2\mathbf{A}^T\mathbf{I}\mathbf{A} = \sigma_q^2\mathbf{I} \tag{12.68b}$$

All reconstruction errors in a block have the same variance σ_q^2 and are not correlated. Note that this result is not obvious since $\mathbf{Q}$ is actually linearly transformed by $\mathbf{A}^T$. A later section will show that the quantizer inputs, or the coefficients, are themselves uncorrelated within a block if the transform $\mathbf{A}$ is of a special form called the KLT (Section 12.5).

Bit Allocations That Minimize Frequency-Weighted Reconstruction Error. Figure 12.8(a) depicted a bit allocation framework that is appropriate to a flat reconstruction error spectrum as in (12.61). Figure 12.8(b) refers to a similar bit allocation algorithm, but with the important difference that the decision thresholds are no longer flat with respect to frequency (coefficient number). In other words, each threshold κ_k is weighted by a certain W_k in anticipation of an overall error spectrum that is desired to be non-flat, specifically a spectrum whose shape follows the quantity W_k^{-1}; $k=0,1,\ldots,N-1$. The error weighting provided by W_k is referred to in general as *frequency-weighting*, although the latter term is precise only in the special case of DFT (Section 12.6), where each index k corresponds to a specific *frequency* component. Frequency-weighted TC algorithms have been beneficial in speech coding [Zelinski and Noll, 1975, 1977] [Tribolet and Crochiere, 1979] as well as image coding [O'Neal and Natarajan, 1977].

The optimum bit allocation is obtained (Problem 12.9) by replacing the σ_k^2 terms in (12.60) by $W_k\sigma_k^2$ terms; and the error spectrum is described by a constant $W_k\sigma_{qk}^2$ rule. In place of (12.61), the weighted-noise contributions are now equal for all coefficients. We have noted a similar result in NFC (Chapter 7).

An interesting class of weighting functions (Appendix E) [Tribolet and Crochiere, 1979] is

0254

$$W_k = (\sigma_k^2)^\lambda \tag{12.69}$$

The case of $\lambda = 0$ provides a white error spectrum ($\sigma_{qk}^2 = \sigma_{ql}^2$) as noted earlier, while $\lambda = -1$ provides a constant bit assignment ($R_k = R_l$) (Problem 12.10). In this case, we find that the ratio of input signal variance and error variance is constant for each coefficient, i.e., the error spectrum has the shape of the input spectrum itself (Figure 12.25).

12.4.3 Zonal Sampling

Bit allocation as discussed hitherto is a complex strategy, especially if it is adaptive. It involves quantizers with different numbers of levels. It also involves reassignment procedures in general, because of the nonzero constraint for R_k.

A simpler strategy is possible in the case of inputs where coefficient variance distributions are known, a priori, to have significant structure. This strategy, known as *zonal sampling*, can be implemented very simply if the input psd is highly structured, as in the two-level example of Figure 2.24. This is exactly what was done in the SBC coding of a two-band signal (Example 11.1) with $R_2 = 0$. More generally, in the case of an input that is expected to be lowpass, if basis vectors are chosen so that low indices of vectors indicate low frequencies, it is possible to obtain useful bit rate reduction by simply retaining $n < N$ coefficients in the transform representation. In reconstructing an approximation to the input, coefficients not included in this subset or zone are set to zero in the simplest case. The term *zonal sampling* originates from 2-D coding where only transform coefficients in a certain geometric zone [for example, the upper left quadrant; see Figure 12.6(d) and (e)] are retained. In keeping with the intent of simplicity, zonal sampling procedures employ a single quantizer with a fixed number of levels for the n in-zone coefficients, and are associated with a straightforward bit rate reduction by a factor of N/n.

We shall now derive an expression for the truncation error in a zonal sampling transform coder. In an actual zonal sampling system, this truncation error will be only one of the components of final reconstruction error. The other component will be the sum of quantization errors for the n retained coefficients.

Let $\mathbf{x}$ be described by a superposition of orthonormal basis vectors $\mathbf{b}_k$ as in (12.20):

$$\mathbf{x} = \sum_{k=0}^{N-1} \theta(k)\mathbf{b}_k \tag{12.70}$$

Let $\mathbf{x}_{zs}$ be the approximation obtained by superposing a subset of $n < N$ basis vectors with the same weights (coefficients) that they had in (12.70):

$$\mathbf{x}_{zs} = \sum_{k=0}^{n-1} \theta(k)\mathbf{b}_k \tag{12.71}$$

The quantities $\mathbf{x}$ and $\mathbf{x}_{zs}$ can be interpreted as points in N- or n-dimensional space, and they can be viewed as realizations of random vectors $\mathbf{X}$ and $\mathbf{X}_{zs}$, respectively.

0255

The mean-square truncation error or zonal sampling error is

$$\sigma_{zs}^2 = E\left[\frac{1}{N}(\mathbf{X}-\mathbf{X}_{zs})^T(\mathbf{X}-\mathbf{X}_{zs})\right] \tag{12.72}$$

However,

$$\mathbf{x} - \mathbf{x}_{zs} = \sum_{k=0}^{N-1} t(k)\mathbf{b}_k ;$$

$$t(k) = \begin{cases} 0 & k = 0,1,\ldots,n-1 \\ \theta(k) & k = n,n+1,\ldots,N-1 \end{cases} \tag{12.73}$$

and assuming an orthonormal basis vector set

$$\mathbf{b}_k^T \cdot \mathbf{b}_l = \delta_{kl} , \tag{12.74}$$

we obtain

$$\sigma_{zs}^2 = \frac{1}{N} E\left[\sum_{k=0}^{N-1} T^2(k)\right] = \frac{1}{N}\sum_{k=n}^{N-1} \sigma_k^2 \tag{12.75}$$

where $\sigma_k^2 = E[\Theta^2(k)]$. Thus the total truncation error $N\sigma_{zs}^2$ of the N-block equals the sum of the variances of the discarded coefficients.

A generalization to two dimensions leads to the results

$$\sigma_{zs}^2 = \frac{1}{N^2}\sum_{k=n}^{N-1}\sum_{l=n}^{N-1}\sigma_{kl}^2 ; \quad \sigma_{kl}^2 = E[\Theta^2(k,l)] . \tag{12.76}$$

The transform which minimizes the mean square error is the one that packs the maximum amount of variance in the first n coefficients in one-dimensional TC, and the first n^2 coefficients in two-dimensional TC. The KLT of Section 12.5 possesses this important property.

Figure 12.9 shows the effect of 4:1 zonal sampling for image inputs on the signal-to-prefiltering-distortion ratio $10\log(\sigma_x^2/\sigma_{zs}^2)$, as a function of subimage size $N\times N$, for sizes ranging from (4×4) to (256×256). Note that the results show a substantial loss due to 4:1 prefiltering when the subimage size is as small as 4×4, and also predict a further rapid loss in truncation error performance in (2×2) coding. (Also see Problem 12.12). Results are based on a first-order Markov process model with equal vertical and horizontal adjacent sample correlations $\rho_v = \rho_h = 0.95$. The ranking of the four alternative transform procedures in Figure 12.9 anticipates a similar observation in Sections 12.5 and 12.6.

The asymptotes in Figure 12.9 range from 20 dB to 24 dB. Perceptually, the degradations represented by these values tend to be less serious than objectively equivalent degradations due to quantization noise. This is because zonal sampling

0256

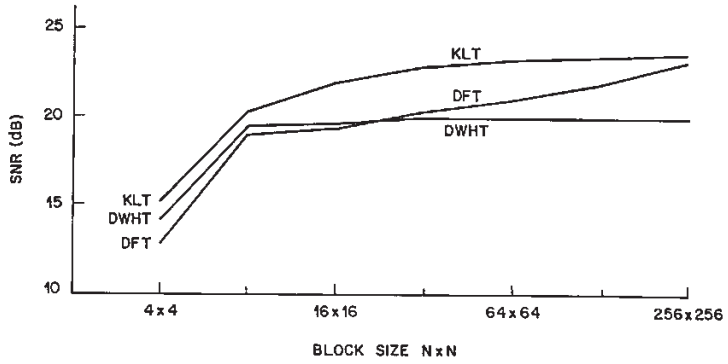


Figure 12.9 Signal-to-truncation error ratio with 4:1 zonal sampling ($n = N^2/4$) in $N \times N$ image transforms, as a function of block size $N \times N$, for an image with $\rho_v = \rho_h = 0.95$ [Pratt, 1978. Reprinted with permission].

is essentially a controlled prefiltering distortion; and unless n is too small, this degradation is perceptually more acceptable than noise that is external to the signal. The nature of prefiltering distortion was in fact discussed earlier in the speech example of Figure 3.16.

In the zonal sampling results of Figure 12.9, the reconstruction error does not include the effect of quantizing retained coefficients. Instead, it is assumed that $R_k = 0$ for rejected coefficients and $R_k = \infty$ for retained coefficients. In *zonal coding*, one assumes that retained coefficients are appropriately quantized. A simple numerical example of zonal coding appears in Problem 12.7. Results for real images will appear in Section 12.5 for DWHT and DFT procedures.

Best results are obtained when the n retained coefficients are not fixed a priori as in zonal sampling, but are adaptively chosen to match input statistics. One form of such adaptive coding is *threshold coding* [Pratt, 1978] [Wintz, 1972] where a coefficient has to exceed a pre-specified or adaptively determined threshold before being included in the transmitter output. This is very closely related to the adaptive bit allocation procedures of Section 12.7. In both threshold coding and adaptive bit allocation coding, one needs side information that informs the receiver about transmitted coefficient indices k and corresponding quantization bits used, R_k.

12.5 Optimum Transform (KLT)

Equations (12.57) and (12.63) show that an optimum matrix A is one that distributes coefficient variances σ_k^2 in such a way as to minimize their geometric mean. The minimum geometric mean result extends to two dimensions with coefficient variances σ_{kl}^2. We will now describe a matrix L with that property.

For any A, the coefficient variances are the diagonal elements of $\mathbf{R}_{00} = \{R_{00}(k,l)\}$; $k,l = 0,1,...,N-1$:

$$\sigma_k^2 = E[\Theta^2(k)] = R_{\theta\theta}(k,k) \tag{12.77}$$

$$\mathbf{R}_{\theta\theta} = E[\boldsymbol{\Theta}\boldsymbol{\Theta}^T] = E[\mathbf{A}\mathbf{X}\mathbf{X}^T\mathbf{A}^T] = \mathbf{A}\mathbf{R}_{xx}\mathbf{A}^T \tag{12.78}$$

We are interested in the product of the diagonal elements of $\mathbf{R}_{\theta\theta}$. From matrix theory [Bellman, 1960],

$$|\mathbf{R}_{\theta\theta}| \leqslant \prod_{k=0}^{N-1} \sigma_k^2 \tag{12.79}$$

for any matrix, while for an orthogonal matrix A,

$$|\mathbf{R}_{\theta\theta}| = |\mathbf{A}\mathbf{R}_{xx}\mathbf{A}^T| = |\mathbf{R}_{xx}||\mathbf{A}\mathbf{A}^T| = |\mathbf{R}_{xx}||\mathbf{I}| = |\mathbf{R}_{xx}| \tag{12.80}$$

and hence

$$\left[\prod_{k=0}^{N-1} \sigma_k^2\right]^{1/N} \geqslant |\mathbf{R}_{\theta\theta}|^{1/N} = |\mathbf{R}_{xx}|^{1/N} \tag{12.81}$$

Minimization of the geometric mean of coefficient variances is provided by a (unitary) matrix A that leads to an equality in (12.79). We shall now describe this optimum transform matrix. We have already shown that corresponding to any $\mathbf{R}_{xx}$, there is a set of *eigenvalues* λ_k; $k=0,1,\ldots,N-1$, and *eigenvectors* $\mathbf{l}_k$; $k=0,1,\ldots,N-1$, defined by

$$\mathbf{R}_{xx}\mathbf{l}_k = \lambda_k \mathbf{l}_k \tag{12.82}$$

Since $\mathbf{R}_{xx}$ is a real symmetric matrix, it follows that the eigenvalues are real, and that there are exactly N eigenvectors which are orthogonal and can be chosen to be orthonormal:

$$\mathbf{l}_k^T \cdot \mathbf{l}_l = \delta_{kl} \tag{12.83}$$

The (discrete) *Karhunen-Loeve Transform* L (*KLT*- or *eigenvector*- or *Hotelling* transform [Hotelling, 1933] [Karhunen, 1947]) is defined by the fact that its *rows* are the above eigenvectors of $\mathbf{R}_{xx}$:

$$\mathbf{b}_k = \mathbf{l}_k \text{ for } \mathbf{L} \tag{12.84}$$

It follows that L is orthogonal, and we have (see 12.20a)

$$\boldsymbol{\theta} = \mathbf{L}\mathbf{x}\,; \qquad \mathbf{x} = \mathbf{L}^T\boldsymbol{\theta} = \sum_{k=0}^{N-1} \theta(k)\mathbf{l}_k \tag{12.85}$$

Note that input vector x is a superposition of eigenvectors which are derived from the statistics of the input (via $\mathbf{R}_{xx}$), and which represent "typical" sequences of the given input sequence. (See the examples of Figures 12.5 and 12.6). Since these basis vectors are signal-dependent, fewer of these vectors can be used on the

0258

average, to approximate the given input, as compared to representations using signal-independent basis vectors. Recall from (12.78) that

$$\mathbf{R}_{\theta\theta} = \mathbf{L}\mathbf{R}_{xx}\mathbf{L}^T \tag{12.86}$$

From (12.82) and (12.84),

$$\mathbf{R}_{\theta\theta} = \begin{bmatrix} \mathbf{l}_0^T \\ \mathbf{l}_1^T \\ \vdots \\ \mathbf{l}_{N-1}^T \end{bmatrix} [\lambda_0\mathbf{l}_0 \quad \lambda_1\mathbf{l}_1 \quad \ldots\lambda_{N-1}\mathbf{l}_{N-1}]$$

$$= [\lambda_j \mathbf{l}_i^T \cdot \mathbf{l}_j^T]_{i,j} = [\lambda_j \delta_{jl}]$$

$$= \begin{bmatrix} \lambda_0 & 0 & 0 & \cdots & 0 \\ 0 & \lambda_1 & 0 & \cdots & \cdot \\ 0 & 0 & \lambda_2 & & \\ \cdot & & & & \\ \cdot & & & & \\ 0 & \cdot & \cdot & \cdot & \lambda_{N-1} \end{bmatrix} = \boldsymbol{\lambda} \tag{12.87}$$

and, from (12.86) and (12.82),

$$\mathbf{L}\mathbf{R}_{xx} = \boldsymbol{\lambda}\mathbf{L} \tag{12.88}$$

Note that the elements of $\mathbf{R}_{\theta\theta}$ are $E[\Theta(i)\Theta(j)]$. Therefore, it turns out that the KLT diagonalizes $\mathbf{R}_{\theta\theta}$, i.e., the transform coefficients are uncorrelated. This supports our approach to quantize these coefficients independently. Note that the uncorrelated nature of the transform coefficients does not imply that their variances are unrelated. For example, the magnitudes of adjacent transform coefficients tend to exhibit closeness in speech and image transforms; this will be illustrated later in the short-time spectra of Figure 12.22a.

Note also that

$$E[\Theta^2(k)] = \lambda_k \; ; \qquad k = 0,1,\ldots,N-1 \tag{12.89}$$

i.e., *the eigenvalues are the variances of the transform coefficients* in the case of KLT. In this ideal case, the individual quantizers of Figure 8.1 quantize linearly independent random variables $\Theta(k)$ of variances λ_k.

The transform $\mathbf{L}$ not only diagonalizes $\mathbf{R}_{\theta\theta}$ but also minimizes the geometric mean of the variances. Indeed, since $\mathbf{R}_{\theta\theta}$ is diagonal, its determinant is just the product of the elements on the main diagonal:

$$|\mathbf{R}_{\theta\theta}| = \prod_{k=0}^{N-1} \lambda_k \tag{12.90}$$

which is the equality in (12.79). Therefore, the KLT results in the least value of the geometric mean of coefficient variances and corresponds to a maximum transform gain, for a given N, of

$$\max_{\mathbf{A}}\{{}^{N}G_{TC}\} = {}^{N}G_{KLT} = \frac{\frac{1}{N}\sum_{k=0}^{N-1}\lambda_k}{\left[\prod_{k=0}^{N-1}\lambda_k\right]^{1/N}} = \frac{\sigma_x^2}{\left[\prod_{k=0}^{N-1}\lambda_k\right]^{1/N}} \tag{12.91a}$$

From (12.61),

$$\min\{\sigma_r^2\} = \epsilon_i^2 2^{-2R}\left[\prod_{k=0}^{N-1}\lambda_k\right]^{1/N} \tag{12.91b}$$

Compare this result with the rate distortion bound for transmitting correlated Gaussian variables in N-blocks:

$$D = 2^{-2R}\left[\prod_{k=0}^{N-1}\lambda_k\right]^{1/N}$$

The geometric mean is the entropy power of Gaussian N-blocks. It turns out that the KLT coding scheme is close to the bound, the difference being zero if a quantizer could be realized with $\epsilon_i^2 = 1$, conditions for which were discussed in Chapter 4 and Section 12.4. We note finally that the denominator in (12.91) can be determined from the correlation matrix $\mathbf{R}_{xx}$ directly from (12.90) and (12.80):

$$\left[\prod_{k=0}^{N-1}\lambda_k\right]^{1/N} = |\mathbf{R}_{\theta\theta}|^{1/N} = |\mathbf{R}_{xx}|^{1/N} \tag{12.92}$$

We will show presently (12.112b) that $|\mathbf{R}_{xx}|$ can be determined from prediction error variances.

Determination of Eigenvalues and Eigenvectors. Let $\mathbf{l}$ be an eigenvector of $\mathbf{R}_{xx}$. The corresponding eigenvalue λ is then defined by (12.82):

$$\mathbf{R}_{xx}\mathbf{l} = \lambda\mathbf{l} \quad \text{or} \quad (\mathbf{R}_{xx}-\lambda\mathbf{I})\mathbf{l} = 0 \tag{12.93a}$$

This is a linear homogeneous system of equations for the unknown λ. In order to have a nonzero eigenvector it is both necessary and sufficient that

$$|\mathbf{R}_{xx}-\lambda\mathbf{I}| = 0 \tag{12.93b}$$

Equation (12.93b) is called the *characteristic equation* of $\mathbf{R}_{xx}$. The determinant above, when expanded, is a polynomial in λ of degree N, and it will have N zeros λ_l with corresponding eigenvectors $\mathbf{l}_l$. The eigenvectors are of course determined only to within a scalar multiple. $\mathbf{R}_{xx}$ is a real symmetric matrix. It can be shown

then that the characteristic equation has N distinct and real zeros (in general, there may be multiplicities), and consequently it has exactly N distinct eigenvectors. In general, it is very difficult to find eigenvalues of a matrix and numerical techniques have to be used. Also, in many cases $\mathbf{R}_{xx}$ is singular, and eigenvectors cannot be uniquely defined [Landau and Slepian, 1971]. What is more significant is that, if $\mathbf{R}_{xx}$ is not known a priori, estimation of its elements requires observations and computations involving an input block of length N' that is an order of magnitude greater than N, for reliable acf estimation. Examples listed below represent relatively simple inputs for which closed-form solutions are available.

Determination of KLT Basis Vectors from R_{xx}. As in Section 12.2, we begin with the simple example of a transform of order 2.

Example 12.1. The case of $N = 2$

Consider a stationary source with variance of unity and normalized autocorrelation ρ between neighboring samples

$$\mathbf{R}_{xx} = \begin{bmatrix} 1 & \rho \\ \rho & 1 \end{bmatrix} \tag{12.94}$$

As shown in Example 2.2, the eigenvalues λ are obtained from the characteristic equation $|\mathbf{R}_{xx} - \lambda \mathbf{I}| = 0$:

$$\begin{vmatrix} 1-\lambda & \rho \\ \rho & 1-\lambda \end{vmatrix} = 0 ; \tag{12.95}$$

$$\lambda_0 = 1 + \rho ; \quad \lambda_1 = 1 - \rho \tag{12.96}$$

so that

$$\prod_{l=0}^{N-1} \lambda_l = \lambda_0 \lambda_1 = 1 - \rho^2 = |\mathbf{R}_{xx}| \tag{12.97}$$

and

$$\mathbf{R}_{\theta\theta} = \begin{bmatrix} \lambda_0 & 0 \\ 0 & \lambda_1 \end{bmatrix} = \begin{bmatrix} 1+\rho & 0 \\ 0 & 1-\rho \end{bmatrix} \tag{12.98}$$

The transform coding gain is

$$G_{KLT} = \frac{[(1+\rho)+(1-\rho)]/2}{[(1+\rho)(1-\rho)]^{1/2}} = \frac{1}{\sqrt{1-\rho^2}} \tag{12.99}$$

To find the eigenvectors, $\mathbf{b}_l = \mathbf{l}_l$, we use (12.82). For $l = 0$,

$$\begin{bmatrix} 1 & \rho \\ \rho & 1 \end{bmatrix} \begin{bmatrix} l_0(0) \\ l_0(1) \end{bmatrix} = (1+\rho) \begin{bmatrix} l_0(0) \\ l_0(1) \end{bmatrix} ; \quad l_0(0) = l_0(1) \tag{12.100a}$$

Similarly for $i = 1$,

$$l_1(0) = -l_1(1) \qquad (12.100b)$$

However, $\mathbf{l}_0$ and $\mathbf{l}_1$ are the rows of $\mathbf{L}$ and for orthonormality, $\mathbf{L}\mathbf{L}^T = \mathbf{I}$. Therefore,

$$l_0^2(0) + l_0^2(1) = l_1^2(0) + l_1^2(1) = 1 \qquad (12.101)$$

From (12.100) and (12.101), the optimum transform is given by

$${}^2\mathbf{L} = {}^2\mathbf{L}^T = {}^2\mathbf{L}^{-1} = \frac{1}{\sqrt{2}}\begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \qquad (12.102)$$

Note that (12.102) also results from setting $a = 1/\sqrt{2}$ in the matrix of (12.10a). •

We have already noted in earlier sections that a matrix $\mathbf{A}$ of the form (12.102) leads to a transform coding gain (12.99). It is significant that the gain noted in these earlier sections is indeed the maximum possible gain for $N = 2$. Recall, from Section 12.2, that it is the sum and the difference of neighboring samples which are to be quantized. It is also significant (Section 12.6) that properties (12.99) and (12.102) are shared by so-called suboptimum transforms, for $N = 2$. Finally, it is interesting to note that for $N = 2$, the KLT is input-independent, in that the matrix $\mathbf{L}$ has no entries depending on ρ. In general, however, for $N > 2$, the basis vectors in $\mathbf{L}$ are input- or $\mathbf{R}_{xx}$-dependent.

KLT Basis Vectors for $N > 2$. The computations of KLT basis vectors for $N > 2$ are invariably tedious. In the following, we will simply mention two examples where the eigenvectors can be derived explicitly.

Example 12.2. Moving Average process

Consider the MA(1) process (2.122):

$$X(n) = Z(n) + a_1 Z(n-1) \qquad (12.103)$$

The process has a psd (2.126)

$$S_{xx}(e^{j\omega}) = (1 + 2\rho\cos\omega)\sigma_x^2 ; \qquad (12.104)$$

$$\rho = a_1/(1+a_1^2) \qquad (12.105)$$

and the eigenvalues (2.127) are simply values of the psd at equidistant *eigenfrequencies* in the interval $(0,\pi)$ given by $(k+1)\pi/(N+1)$; $k = 0,1,\ldots,N-1$. The eigenvectors (2.129a) are samples of sine waves with these eigenfrequencies. •

Example 12.3. Autoregressive process

Consider the AR(1) process as in (2.96):

$$X(n) = \rho X(n-1) + Z(n) ; \qquad (12.106)$$

It can be shown for $\rho \to 1$ that the eigenfrequencies Ω_k are asymptotically uniformly distributed in $(0,\pi)$ and the eigenvalues are the psd values at these eigenfrequencies. The

0262

Ω_k values can be obtained as roots of a polynomial [Bellman, 1960]. In addition, it can be shown that the eigenvectors are sampled sine waves with unequal frequency spacing (except if $\rho \to 1$), and phases ϕ_k depending on Ω_k:

$$\lambda_k = S_{xx}(e^{j\Omega_k}) = \frac{(1-\rho^2)\sigma_x^2}{1+\rho^2-2\rho\cos\Omega_k} \quad k = 0,1,\ldots,N-1 \tag{12.107}$$

$$\mathbf{l}_k = \{\cos n\Omega_k - \phi_k\}_{n=1,2,\ldots,N} \quad \text{for} \quad k = 0,1,\ldots,N-1 \tag{12.108}$$

$$\phi_k = \arctan\frac{1-\rho\cos\Omega_k}{\rho\sin\Omega_k} \tag{12.109}$$

The Ω_k values are obtained as roots of the polynomial

$$\frac{\sin(\Omega_k(N+1))}{\sin\Omega_k} - 2\rho\frac{\sin(\Omega_k N)}{\sin\Omega_k} + \rho^2\frac{\sin(\Omega_k(N-1))}{\sin\Omega_k} = 0 \quad \text{for } k = 0,1,\ldots,N-1 \tag{12.110}$$

Table 12.1 Eigenvectors and eigenvalues for KLT coding of an AR(1) process; $N = 8$ and $\rho = 0.95$.

Eigenvector Matrix								Eigenvalues
−.3383	−.3512	−.3599	−.3642	−.3642	−.3599	−.3512	−.3383	7.030
.4809	.4204	.2860	.1013	−.1013	−.2860	−.4204	−.4809	0.575
.4665	.2065	−.1789	−.4557	−.4557	−.1789	.2065	.4665	0.168
.4226	−.0854	−.4865	−.2783	.2783	.4865	.0854	−.4226	0.082
.3602	−.3468	−.3558	.3513	.3513	−.3558	−.3468	.3602	0.051
−.2833	.4882	−.0942	−.4154	.4154	.0942	−.4882	.2833	0.037
.1952	−.4623	.4603	−.1904	−.1904	.4603	−.4623	.1952	0.030
.0996	−.2786	.4156	−.4896	.4896	−.4156	.2786	−.0996	0.027

Table 12.1 shows the eigenvectors and eigenvalues for an AR(1) process with $\rho = 0.95$, for a block length of $N = 8$. Note that the first eigenvalue λ_0 is much larger than the remaining ones. The eigenvalues for $N = 2$ are simply $1+\rho = 1.95$ and $1-\rho = 0.05$. Respective values of ${}^{N}G_{TC}$ are 5 dB for $N = 2$ and 7 dB for $N = 8$ (Problem 12.19). •

Example 12.4. AR(10) modeling of speech

The first 10 long-time-averaged acf values for a speech signal can be used to develop an AR(10) model. Its maximum prediction gain ${}^{\infty}G_P = {}^{10}G_P$ is 9.5 dB. The model can then be used to compute $\mathbf{R}_{xx}$ for any N. KLT basis vectors and transform codings gains can then be computed. Values of ${}^{N}G_{TC}$ are shown in Figure 12.10 as a function of N. Transform coding results have also been shown for the suboptimum transforms to be discussed in Section 12.6. Note that the KLT gain G_{KLT} is upper-bounded by the maximum value of optimum prediction gain, 9.5 dB.

The basis vectors for this example have already been shown for $N = 8$ in Figure 12.5. Note that the basis vectors, especially the first basis vector $\mathbf{b}_0$, differ considerably from one transform to another. For example, the eigenvector $\mathbf{l}_0$ ($\mathbf{b}_0$ for KLT) is more typical, by definition, for any block of eight successive speech samples $\mathbf{X}$ than the $\mathbf{b}_0$ vector of DFT, DCT, or DWHT, implying that $\mathbf{X}$ usually has no zero-frequency component.

Results from real speech are similar to those of Figure 12.10. Note from this figure that the increase in gain in going from $N = 2$ to $N = 8$ is about 5 dB. A similar gain results for an AR(1) process of adjacent sample correlation $\rho = 0.95$ (Problem 12.19). Also, in Figure 12.10, the gain at $N = 8$ is only 2.5 dB short of the maximum possible gain of 9.5 dB. •

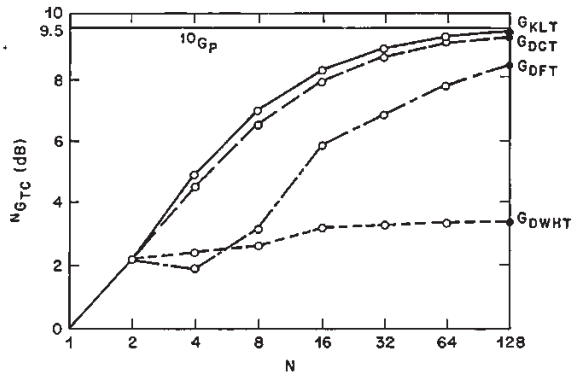


Figure 12.10 Transform coding gain ${}^{N}G_{TC}$ versus N from an AR(10) model of speech signals. The DFT and DWHT are, respectively, frequency-ordered and sequency-ordered transforms [Zelinski and Noll, 1975. Reprinted with permission].

We shall now discuss the effect of N analytically. Results will only apply for KLT, although suboptimal transforms can approach KLT performance, as seen in the example of Figure 12.10.

***G_{KLT}* as a Function of *N*.** Recall from (12.63), with $\epsilon^2_{r,PCM} = \epsilon^2_{r,TC}$ that

$$\sigma^2_{r,TC} = \sigma^2_{r,PCM}/G_{TC}$$

We have seen in (12.91) that

$$\max_{\mathbf{A}}\{{}^{N}G_{TC}\} = {}^{N}G_{KLT} = \frac{\sigma_x^2}{|{}^{N}\mathbf{R}_{xx}|^{1/N}} \tag{12.111}$$

where ${}^{N}G_{KLT}$ is assumed to include the results of optimum bit allocation for any given N. Also, from linear prediction theory (6.48),

$$\frac{|{}^{N}\mathbf{R}_{xx}|}{|{}^{N-1}\mathbf{R}_{xx}|} = \min\{{}^{N-1}\sigma_d^2\}\ ; \tag{12.112a}$$

and hence [Davisson, 1972]

$$|{}^{N}\mathbf{R}_{xx}| = \prod_{k=0}^{N-1} \min\{{}^{k}\sigma_d^2\} = \prod_{k=0}^{N-1} \left(\sigma_x^2/\,{}^{k}G_P\right) \tag{12.112b}$$

where left superscripts refer to covariance matrix or prediction order, and $\min\{\sigma_d^2\}$ and G_P denote prediction error variance and prediction gain, with ${}^{0}G_P = 1$. Using (12.112b) in (12.111),

$$\max_{\mathbf{A}}\{{}^{N}G_{TC}\} = {}^{N}G_{KLT} = \left[\prod_{k=0}^{N-1} {}^{k}G_P\right]^{1/N} \tag{12.113}$$

This relation shows that *the maximum performance of TC is the geometric mean of the first N prediction gains.* Clearly, from (12.113),

$$\max_{\mathbf{A}}\{{}^{N}G_{TC}\} = {}^{N}G_{KLT} < {}^{N}G_{P} \tag{12.114}$$

For example, with a first-order Markov source with adjacent correlation ρ [Noll, 1977],

$${}^{1}G_{P} = {}^{2}G_{P} = \ldots = {}^{\infty}G_{P} = (1-\rho^2)^{-1} \tag{12.115a}$$

$${}^{N}G_{KLT} = \left[\prod_{k=1}^{N-1}(1-\rho^2)^{-1}\right]^{1/N} = (1-\rho^2)^{-\frac{N-1}{N}} \tag{12.115b}$$

As special cases of (12.115b),

$${}^{1}G_{KLT} = 1 \;\; ; \;\; {}^{2}G_{KLT} = (1-\rho^2)^{-1/2} \tag{12.116}$$

The above results show that transform coding gain (as a function of block length) increases more slowly than the prediction gain in DPCM (as a function of predictor order). A similar observation holds in the adaptive coding of speech signals using TC (Figure 12.26) and DPCM (Figure 6.20). PCM performance is represented by ${}^{0}G_{P} = {}^{1}G_{TC} = 1$.

The above comparison of TC and DPCM has a simple qualitative explanation. In a DPCM coder with explicit memory (predictor order) N, the effective memory is much greater than N, because of the propagation of memory in the prediction loop. In TC, however, the effective memory is no greater than the explicit memory (block length) N. Thus, in order to utilize a given extent of memory in the speech process, DPCM needs a much smaller explicit memory than TC. The containing of memory within a block in TC is, however, an advantage in the context of transmission errors (Section 12.8).

Asymptotic Performance. Because of a preponderance of asymptotic terms in (12.113) as $N \to \infty$, the limiting value of (12.113) is

$$\max\{G_{TC}\} = \lim_{N\to\infty} {}^{N}G_{KLT} = {}^{\infty}G_{KLT} = {}^{\infty}G_{P} = \frac{\sigma_x^2}{\min\{{}^{\infty}\sigma_d^2\}} = (\gamma_x^2)^{-1} \tag{12.117}$$

where $\min\{{}^{\infty}\sigma_d^2\}$ is the minimum prediction error and γ_x^2 is the spectral flatness measure.

This result can also be derived by noting that

$$\left[\prod_{i=0}^{N-1}\lambda_i\right]^{1/N} = e^{1/N\sum \log_e \lambda_i} \tag{12.118}$$

and that (2.65)

$$\lim_{N\to\infty}\frac{1}{N}\sum_{i=0}^{N-1}\log_e\lambda_i=\frac{1}{2\pi}\int_{-\pi}^{\pi}\log_e S_{xx}(e^{j\omega})d\omega$$

The above integral is simply $\gamma_x^2\sigma_x^2$ and (12.117) follows.

In summary, for any block-size N, the KLT is optimum. This transform also yields uncorrelated transform coefficients. However, successive transform coefficients of the same index are still correlated, although there is no correlation *within* a block. Finally, in order to realize the spectral flatness measure bound, an infinite block length $N\to\infty$ is needed, in general, in the KLT.

The Optimum Transform for Zonal Sampling. We have noted that the KLT maximizes transform coding performance for any N. It can be shown that the KLT also minimizes zonal sampling error σ_{zs}^2 for any block length N and any value $n\leqslant N$ for the number of coefficients included in the zonal sampling process. Assuming zero mean inputs, the average zonal sampling truncation error is (12.75)

$$\sigma_{zs}^2=\frac{1}{N}\sum_{k=n}^{N-1}E[\Theta^2(k)] \tag{12.119}$$

Let us minimize (12.119) with respect to transform $\mathbf{A}$ subject to the constraint of orthonormality. We minimize

$$\sigma^2=\sigma_{zs}^2+\sum_{k=n}^{N-1}\lambda_k'(\mathbf{a}_k^T\mathbf{a}_k-1) \tag{12.120}$$

where λ_k''s are Lagrange multipliers. From (12.119) and (12.78)

$$\sigma^2=\frac{1}{N}\sum_{k=n}^{N-1}\mathbf{a}_k^T\mathbf{R}_{xx}\mathbf{a}_k+\sum_{k=n}^{N-1}\lambda_k'(\mathbf{a}_k^T\mathbf{a}_k-1) \tag{12.121}$$

The gradient $\nabla\sigma^2$ of (12.121) with respect to $\mathbf{a}_k$ has to be set to zero, the gradient being

$$[\nabla\sigma^2]^T=\left[\frac{\partial\sigma^2}{\partial a_k(0)},\frac{\partial\sigma^2}{\partial a_k(1)},\cdots,\frac{\partial\sigma^2}{\partial a_k(N-1)}\right] \tag{12.122}$$

Note that for any $\mathbf{f}$ and $\mathbf{F}$,

$$\nabla(\mathbf{f}^T\mathbf{F}\mathbf{f})=\mathbf{F}\nabla(\mathbf{f}^T\mathbf{f})=2\mathbf{F}\mathbf{f} \tag{12.123}$$

Applying (12.123) with $\mathbf{F}=\mathbf{R}_{xx}$ and $\mathbf{f}=\mathbf{a}$, and carrying out the operation $\nabla\sigma^2=0$, we find that

$$\mathbf{R}_{xx}\mathbf{a}_k=\lambda_k\mathbf{a}_k \tag{12.124}$$

0266

By definition, (12.124) can hold only for

$$\lambda_k' = \lambda_k; \quad \mathbf{a}_k = \mathbf{l}_k \tag{12.125}$$

where λ_k are the eigenvalues of $\mathbf{R}_{xx}$, and $\mathbf{l}_k$ are the eigenvectors of $\mathbf{R}_{xx}$. Thus the minimization of zonal sampling error is obtained for any n and N with $\mathbf{A} = \mathbf{L}$, the eigenvector (KLT) matrix. The minimum value of zonal sampling error is

$$\min\{\sigma_{zs}^2\} = \frac{1}{N}\sum_{k=n}^{N-1}\mathbf{l}_k^T\mathbf{R}_{xx}\mathbf{l}_k = \frac{1}{N}\sum_{k=n}^{N-1}\lambda_k \tag{12.126}$$

The KLT packs most of the energy into the lower-index coefficients so that the error due to truncation is smaller than with other transforms.

Two-Dimensional KLT. As noted in (12.30), the case of interest is a separable transform of the form

$$\boldsymbol{\Theta} = \mathbf{L}_v\mathbf{X}\mathbf{L}_h^T \tag{12.127}$$

The matrices $\mathbf{L}_v$ and $\mathbf{L}_h$ are computed as in the one-dimensional case, via the KLT basis vector definition (12.88)

$$\mathbf{R}_h = \mathbf{L}_h^T\boldsymbol{\lambda}_h\mathbf{L}_h\,; \quad \mathbf{R}_v = \mathbf{L}_v^T\boldsymbol{\lambda}_v\mathbf{L}_v\,, \tag{12.128}$$

assuming that horizontal and vertical statistics can be separated in the form

$$R_{xx}(k,l) = R_h(k)R_v(l) \tag{12.129}$$

as in the first-order Markov model with separable acf (2.136):

$$\mathbf{R}_h = \{\rho_h^{|l-j|}\}_{l,\,j=0,1,\ldots,N-1} \quad ; \quad \mathbf{R}_v = \{\rho_v^{|l-j|}\}_{l,\,j=0,1,\ldots,N-1} \tag{12.130}$$

When the acf is separable as above, eigenmatrices can be formed from outer products of eigenvectors of the acf in horizontal and vertical directions [Habibi, 1971] [Wintz, 1972].

The above model has been used in deriving the basis images of size 16×16 for the cameraman example in Figure 12.6. Figure 12.11 shows the distribution of variances of coefficients of index 1 to 256, the distributions being statistics taken over 256 subimages of the cameraman image. The monotonic decrease of coefficient variance σ_k^2 with index k is quite evident. It implies that the lowpass basis images (Figure 12.6) get relatively greater weights in image reconstruction. Notice that the packing of energy into low-order coefficients is maximal for the KLT. However, all the transforms in Figure 12.11 have this energy-packing property to different extents, and we will see further examples of zonal coding where this property is utilized. If only 128 coefficients are retained (2:1 zonal sampling), the mse is between 0.34% (KLT) and 0.49% (DWHT), implying respective signal-to-zonal-sampling-error ratios of 25 and 23 dB. In both cases, no visible degradation occurs. If only 64 coefficients are retained (4:1 zonal sampling),

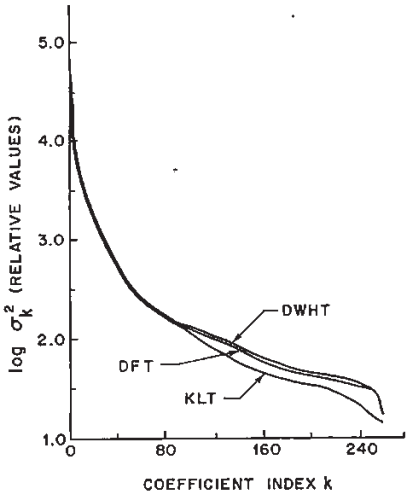


Figure 12.11 Distribution of coefficient variances in 16×16 transform coding of the cameraman of Figure 12.6. Variances are averaged over the 256 sub-images and ordered according to their magnitudes [Wintz;©1972, IEEE].

a blurring effect is visible with all transforms. The hierarchy of coefficients implied in the variance-ordering process is also useful in *embedded coding* and *progressive transmission* systems [Knowlton, 1980]. In these systems, although all coefficients are transmitted to a receiver under ideal conditions, greater priority is given to the transmission of the more important coefficients in the coefficient hierarchy.

12.6 Suboptimum Transforms (DHT, DWHT, DFT, DCT)

The KLT computations (12.85) and (12.86) requires N^2 multiply/add operations. The two stage transform (12.127) needs $2N^3$ multiply-add operations, with each stage needing N^3 (N^2 linear products involving N elements each). The computational problem gets to be simpler with the suboptimum transforms to be discussed next; indeed, the KLT is usually used only to bound the performances of these suboptimal but practical algorithms.

We have already compared the performance of KLT with that of other transforms in Figures 12.9 to 12.11. It is clear that the KLT always performs best. However, for large transform sizes N, the performance of the DFT approaches that of the KLT in both 1-D and 2-D examples; and for very small transform sizes, the mse performances of different transforms do not differ very significantly. Indeed, as seen in Figure 12.10, all mse values are identical for $N = 2$. In this case, all transform matrices, including the input-dependent KLT matrix, simplify to the very familiar matrix (12.10b). Large values of N (say, $N = 128$ or 256) are necessary to realize significant gains in speech coding (as suggested in Figure 12.10).

Transform coding gains saturate faster with N in 2-D image coding. Objective SNR gains tend to saturate after the transform size exceeds (8×8). However, from subjective considerations, a (4×4) transform is in general quite adequate [Netravali and Limb, 1980].

The input-independent suboptimum transforms of this section do not diagonalize covariance matrices. By the same fact, as explained in Section 12.4, they do not minimize the geometric mean of coefficient variances, or provide the best possible distribution of variances for zonal sampling with a given proportion of retained coefficients. However, the input independence of these transforms also makes them simpler in that their basis vectors are predetermined; fast algorithms (such as FFT) exist and aid the implementation of these transforms; and finally, although the gains over PCM for these transforms are strictly upper-bounded by those of the KLT, the gains are often significant enough to warrant their use in digital coding, especially if these applications include refinements such as zonal sampling or adaptive bit allocation. Note that although the design of these transforms is input-independent, their performance (as given by $\mathbf{R}_{\theta\theta}$) will depend on $\mathbf{R}_{xx}$.

The *Discrete Walsh Hadamard* and *Discrete Hadamard* transforms (DWHT and DHT), the *Discrete Fourier* transform (DFT) and the *Discrete Cosine* transform (DCT) are important examples of suboptimal transforms. Several fast algorithms are known for implementing them efficiently [Singleton, 1969] [Ahmed and Rao, 1975] [Elliott and Rao, 1982]

Before we proceed to discuss the suboptimal transforms, let us note for completeness that PCM coding is equivalent to quantization of transform coefficients resulting from the *identity transform*

$$\mathbf{I} = \begin{bmatrix} 1 & 0 & 0 & . & . \\ 0 & 1 & 0 & & \\ 0 & 0 & 1 & & \\ . & . & . & & \\ . & . & . & . & 1 \end{bmatrix} = \{\delta_{ij}\}_{i,j=0,1,\ldots,N-1} \tag{12.131}$$

None of the basis vectors in $\mathbf{I}$ can be expected to be typical for an input sequence. All basis vectors will have to be used to reconstruct an input. The weights (transform coefficients) have equal variances, and hence $G_{TC} = 1$ in this case.

12.6.1 Discrete Hadamard and Walsh Hadamard Transforms (DHT and DWHT)

Walsh Hadamard and Hadamard transforms are characterized by symmetrical orthogonal matrices with unit magnitude elements and orthonormal rows (basis vectors). *Discrete Hadamard Transform* (DHT) and *Discrete Walsh Hadamard Transform* (DWHT) are denoted by matrices $\mathbf{H}$ and $\mathbf{H}'$ respectively:

$$\begin{aligned} &\text{DHT: } \boldsymbol{\theta} = \mathbf{H}\mathbf{x}\,; \quad \mathbf{x} = \mathbf{H}\boldsymbol{\theta} \\ &\text{DWHT: } \boldsymbol{\theta} = \mathbf{H}'\mathbf{x}\,; \quad \mathbf{x} = \mathbf{H}'\boldsymbol{\theta} \end{aligned} \tag{12.132}$$

Note that forward and inverse transforms **A** and **B** are equal in both cases of (12.132), and equal to **H** and **H'** respectively.

The *Hadamard transform* of order 2 is

$$^2\mathbf{H} = \frac{1}{\sqrt{2}}\begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \tag{12.133}$$

which is identical to the KLT matrix for $N = 2$. It is also the familiar matrix (12.10b).

Hadamard matrices of higher order are given by the recursive rule [Shum, Elliot and Brown, 1973]

$$^{2N}\mathbf{H} = \frac{1}{\sqrt{2}}\begin{bmatrix} ^N\mathbf{H} & ^N\mathbf{H} \\ ^N\mathbf{H} & -^N\mathbf{H} \end{bmatrix} \tag{12.134}$$

This rule ensures Hadamard matrices of size 2^p for positive integral p, although Hadamard matrices of other orders (size 0 modulo 4) may exist. From (12.133) and (12.134),

$$^4\mathbf{H} = \frac{1}{2}\begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \end{bmatrix} \tag{12.135a}$$

Walsh transforms, which differ from $^N\mathbf{H}$ with regard to ordering of rows, exist for any integral N. But N is invariably a power of 2 in most applications and the term Walsh Hadamard transform is generally used in such cases to describe any transform that has basis vectors with elements ± 1. Let us consider a *sequency-ordered Walsh Hadamard* matrix $^4\mathbf{H}'$ that differs from $^4\mathbf{H}$ in terms of row ordering:

$$^4\mathbf{H}' = \frac{1}{2}\begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \\ 1 & -1 & 1 & -1 \end{bmatrix} \tag{12.135b}$$

The *sequency* of a ± 1 sequence is defined as half the number of sign changes in one period of the sequence [Harmuth, 1972]. If the period of a basis vector is considered to be the transform size N, the matrices $^4\mathbf{H}$ and $^4\mathbf{H}'$ are characterized by basis vector sequencies of (0,3/2,1/2,1) and (0,1/2,1,3/2). Similarly, the DWHT basis vectors in Figure 12.5(a) have sequencies in the range 0 to 7/2. The term *sequency* (of a rectangular basis waveform $\mathbf{b}_j$) has a connotation that is similar to *frequency* (of a trigonometric sine or cosine waveform, as in DFT).

With a lowpass input and a DWHT matrix of the type ${}^{4}\mathbf{H}'$, one expects transform coefficient variances that monotonically decrease with coefficient index. Recall that the jth coefficient reflects the correlation between the input and the jth basis vector. The coefficients would not be monotonically ordered with a Hadamard transform of the type ${}^{4}\mathbf{H}$, but the coding performances of ${}^{4}\mathbf{H}$ and ${}^{4}\mathbf{H}'$ would be identical because the geometric means of coefficient variances would be identical.

Note also that the matrices (12.135a) and (12.135b) are both symmetric. This property extends to all higher-order matrices ${}^{N}\mathbf{H}$ and ${}^{N}\mathbf{H}'$.

Figure 12.12 shows a set of 64 basis images appropriate to the DWHT matrix ${}^{8}\mathbf{H}'$. These subimages can also be derived from the rectangular basis waveforms given in Figure 12.5(a) (Problem 12.3).

The sequency-ordered Hadamard transform DWHT and its inverse can be explicitly described in the form

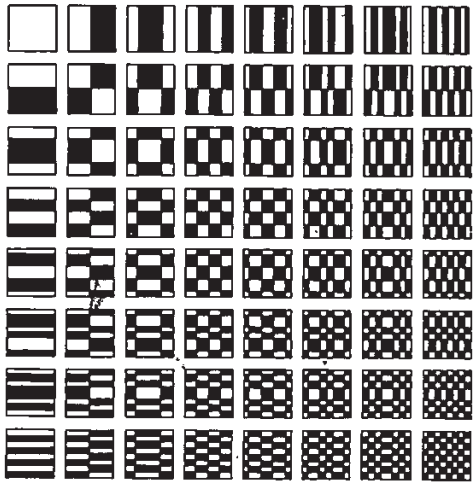
$$\theta(k) = \sum_{n=0}^{N-1} x(n)h(k,n) \; ; \qquad x(n) = \sum_{k=0}^{N-1} \theta(k)h(k,n) \tag{12.136}$$

with forward and inverse transformation kernels that are equal, and given by

$$h(m,n) = a(m,n) = b(m,n) = \frac{1}{\sqrt{N}}(-1)^{\alpha(m,m)} \tag{12.137}$$

where $\alpha(m,n)$ is a function of m and n, assuming values ± 1 [Wintz, 1972] [Pratt, 1978].

Figure 12.12 The set of 64 basis images in 8×8 DWHT. White and black areas represent intensities of +1/8 and −1/8 [Habibi and Robinson;©1974, IEEE].



The comparison of transforms in Figures 12.9 and 12.10 has shown that DWHT is not as effective as transforms with sine-cosine (DFT) and DCT basis vectors. In fact, DWHT performance is not even asymptotically optimum. The suboptimality of the transform is reflected in the fact that it does not diagonalize covariance matrices which are Toeplitz. However, the Hadamard transform does diagonalize a *dyadic* matrix (Problem 12.23). Although suboptimal from the point of view of mse, the rectangular bases in DHT and DWHT suggest that these transforms should be particularly effective in digitizing images which have many edges and discontinuities. In the specific example of intrafield composite coding of color images, simple examples of 2-D DWHT provide better performance than third-order 2-D DPCM (Chapter 6) if the bit rates are in the order of $R = 2$ bits/pixel or less [Murakami, Hatori and Yamamoto, 1982].

Coding systems based on two-dimensional DHT and DWHT have been extensively studied because of the simplicity that is related to the fact that all elements in ${}^{N}\mathbf{H}$ and ${}^{N}\mathbf{H}'$ are ± 1; resulting multiply operations are therefore simply additions with sign changes when appropriate.

2-D DWHT Coding. A discussion of 2-D DWHT coding was already provided in Figure 12.9, in the specific context of zonal sampling. This section will consider this transform in some more detail, with special reference to 2 bits/sample coding.

Example 12.5. 4 × 4 DWHT coding

We have mentioned earlier that 4×4 is an adequate transform size from a perceptual viewpoint. Results from 4×4 DWHT coding are shown in Figure 12.13. Figure 12.13(a) is an original 10-bit PCM representation. Figure 12.13(b) shows a reconstruction based on 16:1 zonal sampling ($R_i = 0$; $0 < i \leq 15$), and Figure 12.13(c) is the result of DWHT coding using the bit allocation procedure shown in Table 12.2, with $R = 2$ bits/sample.

Figure 12.13 Results from 4×4 coding: (a) original 10-bit PCM picture; (b) reconstruction using only the first coefficient $\theta(0)$ in a 4×4 transform (16:1 zonal sampling); and (c) reconstruction based on the bit-allocation procedure of Table 12.2, with $R = 2$ bits/sample [Landau and Slepian;©1971, AT&T].



(a) (b) (c)

Table 12.2 Bit allocation in a 4×4 DWHT coding system with $R = 2$ bits/sample [Landau and Slepian;©1971, AT&T].

Coefficient Number i	Coefficient Variance σ_i^2	Quantizer Bits R_i
0	1.000	6
1	0.098	4
2	0.087	4
3	0.035	3
4	0.038	3
5	0.051	3
6	0.048	3
7	0.034	2
8	0.024	2
9	0.024	2
10	0.020	0
11	0.022	0
12	0.019	0
12	0.015	0
14	0.016	0
15	0.014	0

The dynamic range of coefficient variances in the second column of Table 12.2 is 10 log (1.000/0.014) = 18.4 dB. This is less than the dynamic range of variances to be expected in a KLT. The smaller the dynamic range, the smaller is the ratio of arithmetic mean to geometric mean in (12.63a), and therefore the smaller is the transform coding gain G_{TC}. The same result is true in speech coding. With AR(10) modeling of speech, the dynamic range of coefficient variances is about 30 dB for the DWHT, and about 40 dB for the KLT and the DCT [Zelinski and Noll, 1975].

12.6.2 Discrete Fourier Transform (DFT)

This transform is defined by

$$\theta(k) = \frac{1}{\sqrt{N}} \sum_{n=0}^{N-1} x(n) e^{-j2\pi kn/N}; \quad k = 0, 1, 2, \ldots, N-1$$

$$x(n) = \frac{1}{\sqrt{N}} \sum_{k=0}^{N-1} \theta(k) e^{j2\pi kn/N}; \quad n = 0, 1, 2, \ldots, N-1 \tag{12.138}$$

$$\text{or } \boldsymbol{\theta} = \mathbf{F}\mathbf{x}; \quad \mathbf{x} = \mathbf{F}^{*}\boldsymbol{\theta}; \quad \mathbf{F} = \left\{ \frac{1}{\sqrt{N}} e^{-j2\pi kn/N} \right\}_{k,\, n = 0,1,\ldots,N-1} \tag{12.139}$$

The inverse transformation kernel is the conjugate transpose of the forward kernel [as in (12.35)]. In addition, because of the symmetry of F,

$$\mathbf{F}^{-1} = \mathbf{F}^{*T} = \mathbf{F}^{*} \tag{12.140}$$

For $N = 2$,

$$^2\mathbf{F} = \frac{1}{\sqrt{2}} \begin{bmatrix} e^0 & e^0 \\ e^0 & e^{-j\pi} \end{bmatrix} = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \tag{12.141}$$

which is identical to the KLT matrix of size 2. For $N = 4$,

$$^4\mathbf{F} = \frac{1}{2} \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -j & -1 & j \\ 1 & -1 & 1 & -1 \\ 1 & j & -1 & -j \end{bmatrix}; \quad ^4\mathbf{F}^{-1} = {}^4\mathbf{F}^* = \frac{1}{2} \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & j & -1 & -j \\ 1 & -1 & 1 & -1 \\ 1 & -j & -1 & j \end{bmatrix} \tag{12.142}$$

The columns of $\mathbf{F}^{-1}$ are the unitary basis vectors of the DFT:

$$\mathbf{b}_k = \left\{ \frac{1}{\sqrt{N}} e^{j2\pi kl/N} \right\}_{l=0,1,\ldots,N-1}; \quad \mathbf{b}_k^{*T} \cdot \mathbf{b}_l = \delta_{kl} \tag{12.143}$$

Note that (12.142) is symmetric, as in the case of the DWHT matrices $^4\mathbf{H}$ and $^4\mathbf{H}'$. The general matrix $^N\mathbf{F}$ is also symmetric.

Although $\mathbf{F}$ and $\mathbf{F}^*$ have complex elements of the form $(a+jb)$, a *conjugate symmetry* property applies to the DFT of *real-valued inputs*, and as a consequence the DFT representation involves a total of only N (not $2N$) elements. The conjugate symmetry is described by

$$\theta(k) = \theta^*(N-k); \quad k = 1,2,\ldots,(N/2)-1 \tag{12.144}$$

so that

$$\mathrm{Re}\{\theta(k)\} = \mathrm{Re}\{\theta(N-k)\}; \quad k = 1,2,\ldots,(N/2)-1 \tag{12.145}$$

$$\mathrm{Im}\{\theta(k)\} = -\mathrm{Im}\{\theta(N-k)\}; \quad k = 1,2,\ldots,(N/2)-1 \tag{12.146a}$$

$$\mathrm{Im}\{\theta(0)\} = \mathrm{Im}\{\theta(N/2)\} = 0 \tag{12.146b}$$

In the case of even N, the total number of nonredundant real-valued numbers in the DFT representation is simply N, since $\theta(0)$ and $\theta(N/2)$ are real-valued, and the remaining $N-2$ components have a total of $N-2$ nonredundant numbers. Therefore, for N input samples, there are exactly N unique numbers, representing the transform coefficients that have to be quantized and transmitted.

Example 12.6. 1-D DFT; $N = 4$

$$\mathbf{x}^T = (3, -1, 4, 2); \quad \theta^T = (4, -1/2 + 3j/2, 3, -1/2 - 3j/2) \tag{12.147}$$

The DFT representation has $N = 4$ non-redundant numbers, $-1/2$, $3/2$, 4 and 3. The DFT pair (12.147) is shown in Figure 12.14(a). Notice that the *amplitude spectrum* $|\theta(k)|$, and hence the *energy spectrum* $|\theta(k)|^2$, is symmetrical about $k = N/2 = 2$, except for the $k = 0$ term. •

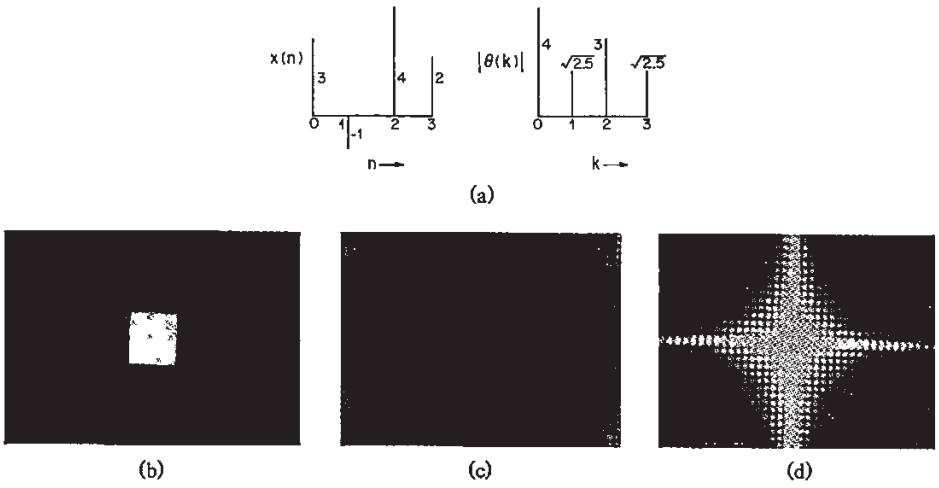


Figure 12.14 DFT examples: (a) 1-D input $x(n)$ and DFT coefficients $\theta(k)$; (b) 2-D input; and DFT coefficient images (c) without spectrum shifting, and (d) with spectrum shifting [Gonzalez and Wintz, 1977. Reprinted with permission].

Example 12.7. A 2-D square input

The square input in Figure 12.14(b) has a $\sin x/x$-type DFT in both spatial dimensions. In the left part of Figure 12.14(c), the origin for the DFT is in the upper left corner of the image, and this is consistent with matrix notation. In Figure 12.14(d), the origin of the DFT, i.e., the zero frequency, has been shifted to the geometric center of the image. This makes the $\sin x/x$ behavior more visible. Both representations include symmetries about $N/2$ as in Example 12.6. Also, in both DFT representations, the luminance of the DFT image has been displayed to be $\log|1+\theta(k,l)|$ rather than $\theta(k,l)$; this is done to accommodate the dynamic range of the DFT terms, which is much larger than the dynamic range of luminance that can be effectively displayed in the illustration. •

The complex basis vectors $\mathbf{b}_k$ in DFT can be decomposed into sine and cosine components. For example, the vector $(1,j,-1,-j)$ in ${}^4\mathbf{F}$ can be decomposed into real and imaginary components $(1,0,-1,0)$ and $(0,1,0,-1)$, which can be interpreted as a sine-cosine basis pair. This observation leads to the possibility of a slightly different Fourier transform $\mathbf{F}'$.

Frequency-Ordered and Real-Valued Fourier Matrix $\mathbf{F}'$. We briefly discuss a transform DF'T which provides only real components in

$$\boldsymbol{\theta}' = \mathbf{F}'\mathbf{x} \tag{12.148}$$

We use the mapping

$$\theta'(0) = \mathrm{Re}\{\theta(0)\} = \theta(0); \quad \theta'(1) = \sqrt{2}\,\mathrm{Im}\{\theta(1)\}; \quad \theta'(2) = \sqrt{2}\,\mathrm{Re}\{\theta(1)\};$$
$$\ldots\ldots \theta'(N-3) = \sqrt{2}\,\mathrm{Im}\{\theta((N/2)-1)\}; \quad \theta'(N-2) = \sqrt{2}\,\mathrm{Re}\{\theta((N/2)-1)\};$$

$$\theta'(N-1) = \mathrm{Re}\{\theta(N/2)\} = \theta(N/2) \tag{12.149}$$

The scaling factors and the ordering have been chosen such that the basis vectors are orthonormal and frequency-ordered [Zelinski and Noll, 1975]. For computation purposes, θ' can be obtained via an FFT that yields θ followed by rescaling and ordering.

The basis vector $\mathbf{b}'_k$ corresponding to (12.149) form a real-valued Fourier matrix $\mathbf{F}'$ which does not have the redundancies caused by the conjugate symmetry of the DFT. Furthermore, the set $\mathbf{b}'_k$ is ordered in accordance with the familiar notion of increasing frequency, beginning with a zero frequency vector $\mathbf{b}'_0$. This has already been shown in Figure 12.5 for the example of $N = 8$.

Example 12.8. 1-D DFT; $N = 4$

$$^4\mathbf{F}' = \frac{1}{2}\begin{bmatrix} 1 & 1 & 1 & 1 \\ 0 & -\sqrt{2} & 0 & +\sqrt{2} \\ \sqrt{2} & 0 & -\sqrt{2} & 0 \\ 1 & -1 & 1 & -1 \end{bmatrix}$$

$$\mathbf{x}^T = (3, -1, 4, 2); \quad \theta'^T = (4, +3/\sqrt{2}, -1/\sqrt{2}, 3) \tag{12.150}$$ •

Implementation of the DFT and IDFT. DFT representations have two important advantages. The first is that a sine-cosine basis space is a very familiar concept. It provides a natural framework for optimizations of transform coder design with inputs where the perceptual effects of signal distortion are best understood in the frequency domain. A second advantage has to do with computation. A direct evaluation of an N-point DFT requires about N^2 complex multiply-add operations; but if N is a power of 2, the Fast Fourier Transform (FFT) method needs in the order of $N \log_2 N$ operations [Elliott and Rao, 1982]. With $N = 1024$, the FFT is about one hundred times faster than a direct computation. The savings in computation time are approximately 32 and 18 for $N = 256$ and 128, respectively.

In the case of real-valued inputs, further reductions are possible by exploiting the symmetry properties of the DFT. Two N-point real sequences can be considered real and imaginary parts of a complex sequence and the $2N$-point sequence can be transformed with a single transform. A simple even-odd separation recovers the real and imaginary parts of the two spectra. This procedure involves a total number of operations in the order of $(N \log_2 N)/4$ [Narasimha and Peterson, 1978].

The FFT algorithm can also be used to evaluate the inverse (IDFT) operation $\mathbf{x} = \mathbf{F}^*\theta$. Indeed, by computing the DFT of the complex conjugate of θ, i.e., the DFT of θ^*, we obtain $\mathbf{x}^* = \mathbf{F}\theta^*$, and we obtain $\mathbf{x}$ by complex-conjugating the result. We finally note that FFT's usually compute a DFT that has no scaling factor and an IDFT that has a scaling by $1/N$. Resulting basis vectors are orthogonal but not orthonormal.

Performance of DFT, KLT and DCT with an AR(1) Input. Figure 12.15 illustrates transform coder performance for a first-order Gauss Markov or AR(1) input. With such an input, since transform coefficients are also Gaussian, the quantizer

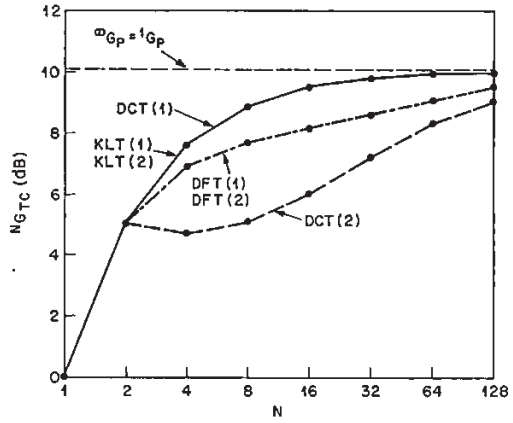


Figure 12.15 Transform coding gains for a first-order Gauss-Markov source with (1) $\rho = 0.95$; and (2) $\rho = -0.95$ [Zelinski and Noll, 1975. Reprinted with permission].

performance factor ϵ_q^2 indeed does not have to be modified, and gains based on coefficient variance distributions, as derived in Sections 12.3 and 12.4, are true gains over PCM. The maximum performance in Figure 12.15 corresponds to ${}^{\infty}G_P = {}^{1}G_P = (1-\rho^2)^{-1} = \gamma_x^{-2}$ [see (12.115a)]. For $\rho = +0.95$, this value is 10.25, or approximately 10.11 dB. This is the upper bound for any transform. For $N = 2$, the transform coding gain is ${}^{2}G_{TC} = (1-\rho^2)^{-1/2}$ or 5.06 dB (12.116).

Note that the gap between G_P and G_{TC} becomes smaller with increasing N but the approach to G_P depends on the transform. The increase of gain in DFT is slower than that in KLT, which in the example of a positive $\rho = 0.95$ almost equals that of DCT, with a difference of less than 0.1 dB. With a negative $\rho = -0.95$, however, DFT is significantly better than DCT. It is interesting that the performance of DFT does not depend on the sign of ρ, but only on its magnitude. The KLT performance is identical for positive and negative ρ's, but note that the KLT matrix has to be reoptimized accordingly. We recall that the eigenvectors of an AR(1) process are sines (or cosines) of appropriate frequency and phase given by (12.109a). DFT basis vectors are suboptimum; they have frequencies which are multiples of a fundamental frequency. For $N \to \infty$, the eigenfrequencies in the KLT are uniformly distributed, as if they were multiples of a fundamental, and the DFT is therefore asymptotically optimum. For finite N it can be shown [Pearl, 1973] that

$$G_{KLT} - G_{DFT} \sim 1/\sqrt{N} \tag{12.151}$$

The suboptimality of the DFT is related to the fact that it does not diagonalize covariance matrices which are Toeplitz. However, the DFT does diagonalize a *circular* acf matrix where all rows are circularly shifted versions of the first row (Problem 12.29). The above result is not surprising. The DFT representation introduces periodicities with period N in all sequences (input as well as transform coefficients). The inverse DFT generates a sequence

$$c(n) = x(n); \quad n = 0,1,\ldots,N-1 \tag{12.152}$$

which equals the original sequence within the N-block but which does not equal the input outside the block. Instead,

$$c(n) = c(n+N) \tag{12.153}$$

As a result of this periodicity,

$$\begin{aligned} \mathbf{R}_{cc}(k) &= E[C(n)C(n+k)] = E[C(n+N)C(n+k)] \\ &= R_{cc}(N-k) \; ; \; k = 0,1,\ldots,N-1 \end{aligned} \tag{12.154}$$

which is a *circular* acf of with period N. Clearly, as N increases, the period in the acf function $\mathbf{R}_{cc}$ becomes longer, the non-Toeplitz structure in $\mathbf{R}_{xx}$ becomes less evident, and the suboptimality of the DFT for a Toeplitz covariance matrix becomes less, as suggested in (12.151). The effect can be explained in terms of end effects induced by forced periodicities and consequent discontinuities at the beginning and end of each N-block. Similar (end) effects limit the performance of DCT transforms. In the case of DCT, it will be shown (Section 12.6.3) that end effects are less harmful than in DFT for a given N. DWHT transforms do not have deviations from Toeplitz at the ends of the matrix but within the matrix. Larger N's do not shift effects to the ends. Therefore, there is no asymptotically optimum performance.

Two-Dimensional DFT. In general, a 2-D transform requires premultiplication of the input with a fourth-order tensor. However, the DFT kernel

$$\frac{1}{N} e^{-j2\pi(km+ln)/N} \; ; \quad k,l = 0,1,\ldots,N-1 \tag{12.155}$$

is separable and symmetric (see Section A.2.1), so that, with (12.36) and (12.140),

$$\boldsymbol{\Theta} = \mathbf{FXF} \; ; \quad \mathbf{X} = \mathbf{F}^*\boldsymbol{\Theta}\mathbf{F}^* \tag{12.156}$$

The operations in (12.156) can be performed with $2N^2\log_2 N$ instead of $2N^3$ complex multiply/add operations if an FFT algorithm is applied. ($N \log_2 N$ operations for one row of X, $N^2\log_2 N$ operations for $\mathbf{XF}$, and $2N^2\log_2 N$ operations for $\mathbf{FXF}$). Also, as in the one-dimensional case, although complex elements are involved in the matrix $\mathbf{F}$, the two-dimensional representation requires only N^2 numbers that have to be quantized and transmitted (Problem 12.25). Redundant components are indicated by cross-hatched regions in the $N = 256$ example of Figure 12.16.

As in the one-dimensional case, there is a suboptimality in DFT performance that can be described in terms of un-diagonalized covariances or, equivalently, in terms of end effects due to DFT-induced periodicities in the IDFT output. These end effects refer, in general, to all four edges of a subimage and the intensity of these effects diminishes with increasing subimage size.

Example 12.9. 16×16 DFT coding of images

Figure 12.17 shows examples of 16 × 16 coding using the image in Figure 12.6(c) as input. The distribution of variances has already been given in Figure 12.11. Figures

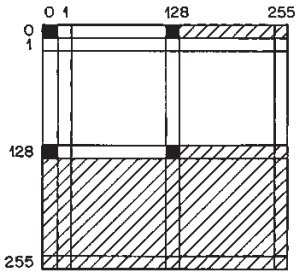
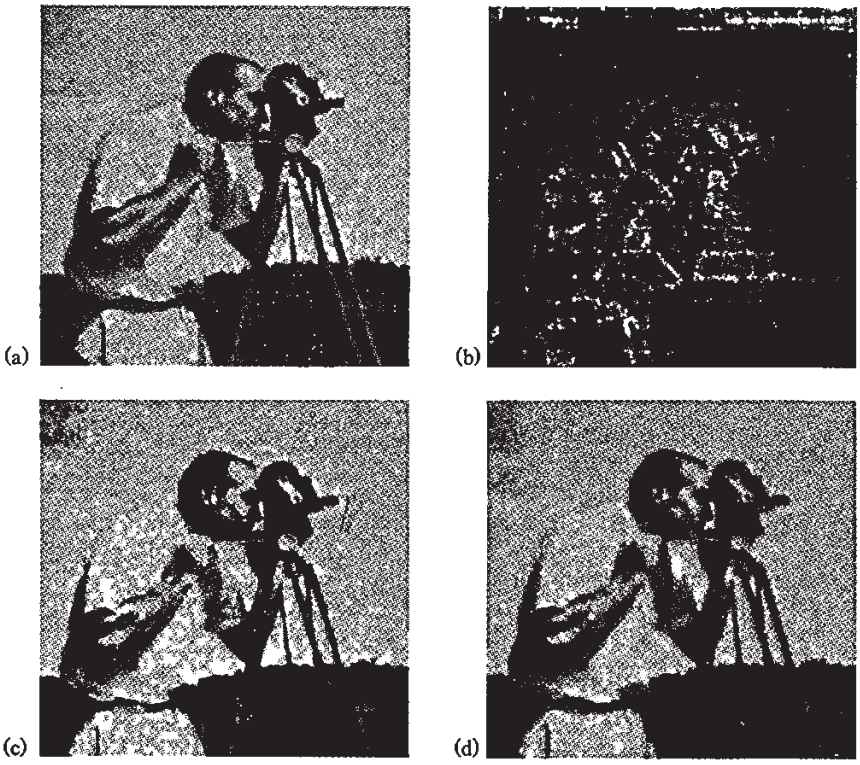


Figure 12.16 Nonredundant (blank), redundant (crosshatched), and real (darkened) areas in two-dimensional DFT with $N = 256$ [Pratt, 1978. Reprinted with permission].

Figure 12.17 An example of 16×16 DFT coding with the image of Figure 12.7(c) as input: (a) reconstructed image in zonal sampling (retaining of $n = 128$ out of 256 coefficients); (b) error in (a) versus the original, scaled by a factor of eight (black refers to zero error and white to an error of 32 gray levels); (c) reconstruction based on coding the 128 coefficients in zonal sampling with a set of 4-level nonuniform quantizers, optimized for the pdf of each transform coefficient ($R = 2$ bits/sample); and (d) reconstruction based on a quantizing system that uses variance-adaptive bit allocation (Section 12.7.2) and an average bit rate of $R = 2$ bits/sample [Wintz;©1972, IEEE].



12.17(a) and (b) show the output and the errors in a 2:1 zonal sampling scheme where only 128 of the most important coefficients are retained. For lowpass images (which are typical) the coefficients with the higher variances are those found in the upper left corner of an unshifted coefficient image array such as Figure 12.14(c). Figure 12.17(b) shows how high frequency edge-information is lost due to the truncation error caused by zonal sampling; the variance of this error is 0.45% (SNR = 23.5 dB). Figures 12.17(c) and (d) show the effect of coding the 128 retained coefficients with a total of 256 bits, using a set of 128 mmse four-level quantizers in (c), and a variance-adaptive bit allocation system (Section 12.7.2) in (d). The quantization error variances in (c) and (d) are 2.1% and 0.78% (SNR = 17 dB and 21 dB, respectively). Note that in (c), the quantization error variance is an order of magnitude higher than the zonal sampling error variance in (a) and (b). With adaptive bit allocation, the variances of zonal sampling and quantization errors are in the same order of magnitude. The quantization error variance can be lowered by at least another order of magnitude by the use of entropy coding [Wintz, 1972]. •

12.6.3 Discrete Cosine Transform (DCT)

The DCT is an orthonormal transform

$$\boldsymbol{\theta} = \mathbf{C}\mathbf{x}\,; \qquad \mathbf{x} = \mathbf{C}^{-1}\boldsymbol{\theta} \tag{12.157a}$$

defined by [Ahmed, Natarajan and Rao, 1974] [Ahmed and Rao, 1975]

$$\theta(k) = \sqrt{\frac{2}{N}}\,\alpha(k)\sum_{n=0}^{N-1} x(n)\cos\frac{(2n+1)k\pi}{2N}\,; \qquad k = 0,1,\ldots,N-1 \tag{12.157b}$$

$$\alpha(0) = 1/\sqrt{2} \text{ and } \alpha(k) = 1; \qquad k \neq 0$$

$$x(n) = \sqrt{\frac{2}{N}}\sum_{k=0}^{N-1} \alpha(k)\theta(k)\cos\frac{(2n+1)k\pi}{2N}\,; \qquad n = 0,1,\ldots,N-1 \tag{12.157c}$$

Note that correct to a scaling factor, forward and backward (inverse) transforms have identical transformation kernels. A comparison of the last equation with (12.20a) shows that the basis vectors are sampled cosines which have phase shifts that are not given by an alternating 0 and $\pi/2$ pattern as in (the sines and cosines of) DFT. Comparing (12.157c) and (12.20a), the DCT basis vectors are

$$\mathbf{b}_k = \left\{\sqrt{\frac{2}{N}}\,\alpha(k)\cos\frac{(2n+1)k\pi}{2N}\right\}_{n=1,2,\ldots,N-1} \text{ for } k = 0,1,\ldots,N-1 \tag{12.158}$$

With $\Omega_k = k\pi/N$ in (12.108), the basis vectors can be rewritten to be just the eigenvectors of an AR(1) process with correlation approaching unity. The DCT is thus close to the optimum for AR(1) processes with correlation $\rho \to 1$ [Jain, 1976].

For $N = 2$, the DCT is identical to the optimum KLT in (12.102):

$${}^2\mathbf{C} = \begin{bmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \\ \cos\frac{\pi}{4} & \cos\frac{3\pi}{4} \end{bmatrix} = \frac{1}{\sqrt{2}}\begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} = {}^2\mathbf{L} \tag{12.159}$$

For $N = 4$

$$^4\mathbf{C} = \frac{1}{2}\begin{bmatrix} 1 & 1 & 1 & 1 \\ a & b & -b & -a \\ 1 & -1 & -1 & 1 \\ b & -a & a & -b \end{bmatrix} \quad ; \quad \begin{matrix} a = \sqrt{2}\cos\frac{\pi}{8} = 1.306 \\ b = \sqrt{2}\cos\frac{3\pi}{8} = 0.541 \end{matrix} \tag{12.160}$$

Note that $^4\mathbf{C}$ is *not* symmetric, unlike $^4\mathbf{H}$ in (12.135) and $^4\mathbf{F}$ in (12.142).

The near-optimality of DCT for AR(10)-modeled speech statistics has been seen from the KLT and DCT basis vectors of Figure 12.5 for $N = 8$; and its optimality for an AR(1) process with $\rho = 0.95$ has been seen in Figure 12.15.

The attractiveness of the DCT is two-fold: (a) it is nearly optimal for AR(1) processes with high positive values of adjacent-sample correlation, and (b) it can be computed via the DFT using an FFT algorithm [Ahmed and Rao, 1975] [Chen, Smith and Fralick, 1976] [Elliott and Rao, 1982].

The feature (a) is very important since many signals can be described fairly accurately by AR(1)-statistics. In all other cases, the DCT does not diagonalize Toeplitz matrices; but it diagonalizes matrices whose elements at the end of the *main* diagonal are slightly perturbed (i.e., they deviate from the Toeplitz structure). The suboptimality of the DCT for negatively correlated sequences is shown in Figure 12.15 and Problem 12.31.

The fast computation procedure mentioned in (b) consists of extending the input block of N samples to a $2N$-block with even symmetry, taking a $2N$-point DFT, and saving N terms in it. The DFT of a real and symmetric sequence contains only real coefficients corresponding to the cosine terms of the series. The extension is defined as

$$x'(n) = \begin{cases} x(n) & n = 0,1,\ldots,N-1 \\ x(2N-1-n) & n = N,N+1,\ldots,2N-1 \end{cases} \tag{12.161a}$$

as shown in Figure 12.18. The $2N$-DFT of $x'(n)$ is

$$\theta'(k) = \frac{1}{\sqrt{2N}} \sum_{n=0}^{2N-1} x'(n) e^{-j(2\pi kn/2N)}$$

$$= \frac{1}{\sqrt{2N}} e^{jk\pi/2N} \sum_{n=0}^{N-1} x(n) \cos\left[\frac{(2n+1)k\pi}{2N}\right] \tag{12.161b}$$

By comparing it with (12.157b), it can be seen that

$$\theta(k) = \alpha(k) e^{-jk\pi/2N} \theta'(k) \tag{12.161c}$$

Notice that periodic extensions which are inherent in DFT operations have smaller end-effects (caused by discontinuity at the border of one block and its repetition) in the case of the $2N$-extension than DFT operations based on the N-block. In DFT, quantization effects cause an exchange of distortion components between the left

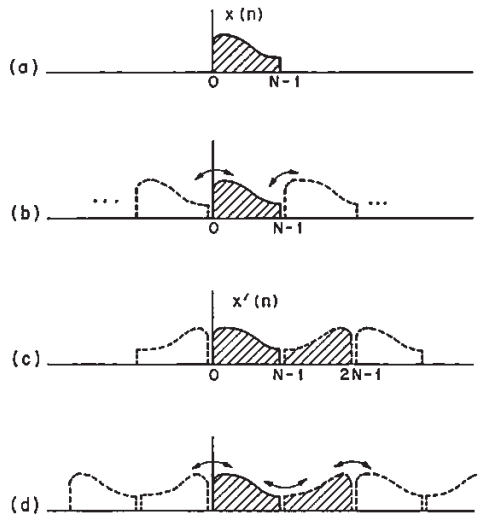


Figure 12.18 Illustration of end effects in DFT and DCT: (a) input block of length N; (b) end effects in DFT analysis/synthesis; (c) equivalent $2N$-point data block for DCT; and (d) end effects in DCT analysis/synthesis [Tribolet and Crochiere;©1979, IEEE].

and right boundaries. In DCT, such an exchange does not occur [Tribolet and Crochiere, 1979].

From the above discussion, it also follows that the DCT can be obtained with $2N \log_2 N$ multiply-add operations. Slightly more efficient procedures are described in the literature [Chen, Smith and Fralick, 1976] [Wang, 1983].

Two-Dimensional DCT. In two dimensions, the DCT can be expressed in the form [Pratt, 1978] [Gonzalez and Wintz, 1977]

$$\theta(k,l) = \frac{2}{N}\alpha(k)\alpha(l)\sum_{m=0}^{N-1}\sum_{n=0}^{N-1} x(m,n)\cos\frac{\pi k(2m+1)}{2N}\cos\frac{\pi l(2n+1)}{2N} \qquad (12.162a)$$

$$x(m,n) = \frac{2}{N}\sum_{k=0}^{N-1}\sum_{l=0}^{N-1}\alpha(k)\alpha(l)\theta(k,l)\cos\frac{\pi k(2m+1)}{2N}\cos\frac{\pi l(2n+1)}{2N} \qquad (12.162b)$$

where $x(m,n)$ is a $N\times N$ field, k, l, m, n all range from 0 to $N-1$, and $\alpha(0) = 1/\sqrt{2}$; $\alpha(j) = 1$; $j \neq 0$. Note that the transformation kernels are separable, so that the 2-D DCT can be conveniently performed in two steps, each of which involves a 1-D DCT. As in the one-dimensional case (Figure 12.18), DCT coding with an even-symmetry end extension has fewer edge-effect problems than DFT image coding.

0282

12.6.4 Three-Dimensional Transform Coding of Images

Moving images can be represented as a sequence of still images presented at adequately frequent intervals (for example, 30 frames or 60 interlaced fields). There is a high *temporal* correlation between successive frames that can be exploited to reduce the bit rate beyond what is possible in schemes that only utilize *spatial* correlations within single frames. Such *interframe* coding, of course, necessitates the storage of one or more frames, as well as the use of special vector processing algorithms [Natarajan and Ahmed, 1977]. To keep the total number of transformed pixels down to a manageable value, it is usual to employ 4×4 (rather than, say an 8×8 or 16×16) subimages in any given frame. An interframe transform coder using 4 frames will then involve a pixel-cube containing $4 \times 4 \times 4 = 64$ elements.

Formally, a 3-D transform, which is a generalization of the 2-D version (12.26), can be described as follows. A sequence of image frames $\mathbf{X}(k,l,m)$ is 3-D transformed in blocks of $K \times L \times M$ pixels with transform coefficients

$$\theta(u, v, w) = \sum_{k=0}^{K-1} \sum_{l=0}^{L-1} \sum_{m=0}^{M-1} x(k, l, m) a(u, v, w; k, l, m) \qquad (12.163)$$

and inverse-transformed by

$$x(k, l, m) = \sum_{k=0}^{K-1} \sum_{l=0}^{L-1} \sum_{m=0}^{M-1} \theta(u, v, w) b(u, v, w; k, l, m) \qquad (12.164)$$

In the case of separable transforms,

$$a(u, v, w; k, l, m) = a(u, k) a(v, l) a(w, m) \qquad (12.165)$$

where the $a(\cdot, \cdot)$ are *transform kernels*, elements of basis vectors of 1-D transforms (see Section 12.3). As an example, using the DCT is nearly equivalent to the optimum (KLT) if the statistics of the image can indeed be described by separable first-order Markov processes with correlations $\rho \to +1$, along rows, columns and temporal axes. A $(K \times L \times M)$ transform with 32:1 zonal sampling and an 8-bit representation for in-zone coefficients ($R = 0.25$ bit/pixel) provides *SNR* values of 13 and 18 dB with $4 \times 4 \times 4$ and $32 \times 32 \times 4$ transforms [Roese, Pratt and Robinson, 1977].

12.6.5 Hybrid Schemes involving TC and DPCM Coding

Both TC and DPCM are based on source decorrelation prior to quantization, suggesting that in a J-dimensional approach ($J > 1$), both of these techniques can be used in different dimensions to exploit respective correlations. The use of DPCM in a given dimension can alleviate storage requirements, and also provide more efficient decorrelation. We recall that AR(M) processes can be decorrelated by Mth order predictors. Other advantages of a hybrid approach can include

implementation advantages and robustness to transmission errors. A general block diagram of a hybrid TC/DPCM coder appears in Figure 12.19.

The diagram shows that hybrid coding can be applied to both intraframe and interframe systems. We will presently discuss one example of each of these types of systems. The interframe hybrid coder is in general more complex, but it offers advantages in bit rate compression as well as transmission error robustness [Roese, Pratt and Robinson, 1977] [Kamanagar and Rao, 1981] [Ploysongsang and Rao, 1982]. The intraframe hybrid coder with DPCM in the vertical direction has the traditional problem of transmission error streaks that is characteristic of 1-D DPCM (Chapter 6).

The intraframe TC/DPCM coder [Habibi, 1974] [Habibi and Robinson, 1974] [Habibi, 1981] uses an N-block 1-D transform along an image line l, and the transform coefficients $\theta_l(k)$; $k=0,1,...,N-1$, are separately DPCM-encoded with a bank of N first-order DPCM coders which use respective reconstructed coefficients $\tilde{\theta}_{l-1}(k)$ from the previous line $l-1$ for predictions. The predicted value is $\hat{\theta}_l(k)$ and the quantized quantity is the interline difference

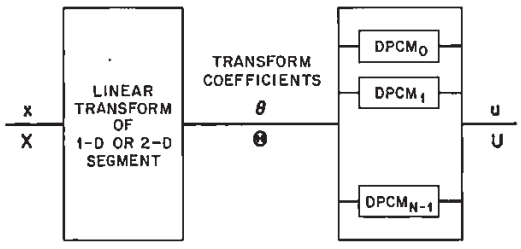
$$d_l(k) = \theta_l(k) - h(k)\tilde{\theta}_{l-1}(k) \quad ; \quad k = 0,1,...,N-1 \tag{12.167}$$

where $h(k)$ is an appropriate predictor coefficient chosen as described in the DPCM section. The reconstructed values $\tilde{\theta}_l(k)$ are used at the receiver for the inverse transform operations. The DPCM encoder utilizes interline image correlations, as reflected in correlations between successive transform coefficients of identical index. The number of bits assigned to the DPCM encoders are ideally set proportional to the logarithm of the respective difference signal variances:

$$\sigma_{dk}^2 = \sigma_k^2(1-\rho_k^2) \quad ; \quad k = 0,1,...,N-1 \tag{12.168}$$

where σ_k^2 and ρ_k are the variance and adjacent-line correlation for transform coefficient of index k.

Figure 12.19 Block diagram of hybrid TC/DPCM coder. The upper row of symbols represents the case where 1-D transform coding of a segment of an image line is followed by DPCM coding of the coefficient vector, based on coefficients of an adjacent line. The lower row of symbols represents the case where 2-D transform coding of a subimage is followed by DPCM coding of the coefficient matrix, using coefficients of an adjacent frame.



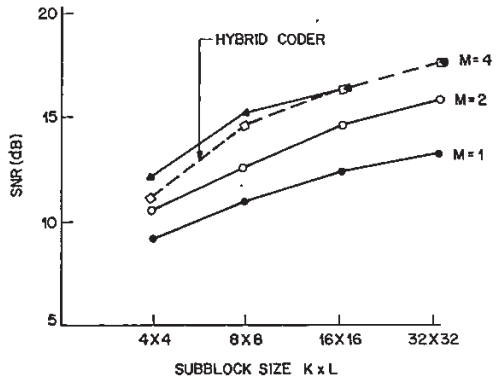


Figure 12.20 Comparison of 3-D DCT and hybrid coding based on 2-D DCT/DPCM. The input is a 3-D Markov source with equal acf values of $\rho = 0.95$ in all three dimensions. The average bit rate in both schemes is 0.25 bits/sample [Roese, Pratt and Robinson; ©1977, IEEE].

In the interframe TC/DPCM coder, spatial correlations are utilized by a 2-D transform, and temporal correlations by first-order DPCM, thus avoiding one-step transform processing of $N \times N \times N$ blocks (Figure 12.19). Figure 12.20 provides a comparison of two interframe approaches—one based on 3-D DCT and the other based on a hybrid approach, with 2-D DCT followed by DPCM [Roese, Pratt and Robinson, 1977]. The input is a 3-D Markov source with equal acf values of $\rho = 0.95$ in horizontal, vertical and temporal directions. With 32:1 zonal sampling and 8-bits per retained coefficient, all the schemes represent an average bit rate of $R = 0.25$ bit/pixel. The 3-D DCT coder requires a $16 \times 16 \times 4$ transform ($K = L = 16$, $M = 4$), i.e., a storage of 4 frames for an *SNR* of about 17 dB. The hybrid coder with $K = L = 16$ reaches this performance with an explicit memory of only one frame for DPCM prediction. The $M = 1$ characteristic in the figure describes the performance of an intraframe zonal-sampling DCT system.

With real images, the *SNR* performances of the 3-D TC and 2-D TC/DPCM coders are in general very different for a given dimensionality. The hybrid approach has the advantage that adaptive bit allocation (see Section 12.7.4) is simpler to optimize in 2-D transform coding and also in the 1-D DPCM stage. A 16×16×16 2-D TC/DPCM coder can provide high quality digitizations of (256×256) black-and-white images at bit rates in the order of 0.5 to 1 bit/pixel (Example 12.11).

12.7 Adaptive Transform Coding of Speech and Images

We will now re-examine the components of Figure 12.1 with emphasis on quantization and adaptive bit allocation. Although transform coding systems can be made adaptive in many different ways [Habibi, 1977], the term *Adaptive Transform Coding* (ATC) in this book will be reserved for systems that use

adaptive bit allocation. The use of adaptive bit allocation will lead to significant increases in *SNR* as well as perceived quality for both speech and images. Important limits on ATC performance are those set by boundary effects due to the block operations inherent in Transform Coding. We will discuss these effects as and when appropriate.

12.7.1 Quantization and Bit Allocation Strategies

Earlier chapters have discussed quantization systems at length. We have seen that quantizer performance can be maximized by means of nonuniform quantizer characteristics, adaptive logics for step size, entropy coding or a suitable combination of such features.

A transform coefficient can be regarded as a random variable. The coefficient is a linear combination of N^J random variables, where J is the transform dimensionality. The pdf of the coefficient therefore depends on input statistics, the type of transform, its size N^J, and in general, also on coefficient index. We note, however, that the pdf's will be strictly Gaussian for Gaussian inputs, while for other inputs, there is a Gaussian tendency for $N \rightarrow \infty$.

In *image coding* with finite N, Gaussian or Laplacian models have been found to be adequate for all except the first coefficient for which several alternative models have been proposed [Tescher, 1979] [Murakami, Hatori and Yamamoto, 1982]. Information about coefficient pdf can be utilized fully only if the quantizer bank of Figure 12.1 consists of a multiplicity of quantizers.

In *speech coding* systems, adaptive step size quantizers are usually critical to handle nonstationarity. As discussed in earlier sections [Figures 4.40(b) and (c)], there are two equivalent approaches, one based on scaling quantizer inputs to provide unit variance, and one based on explicit adaptations of the quantizer step size. In either approach, the scaling depends on information about input variance. This information is either explicitly transmitted (AQF) or recovered reliably from examining a recent sequence of quantizer outputs (AQB). The AQB procedure is very appropriate in the context of sub-band coding (Chapter 11). But ATC systems for speech depend on explicit transmission of variance information anyway, because of the need for adaptive bit allocation; and therefore AQF procedures become a more natural choice for step size adaptation.

An important design parameter in both speech and image coding is the number of quantizer levels as a function of coefficient index. In zonal sampling, where the emphasis is on simplicity, the number of quantizer levels is usually held constant; this leads to a simple system, involving a single shared quantizer. The coefficients can, of course, be scaled to unit variance before quantization, provided that individual variances are known a priori or can be reliably estimated. In a zonal coding or variable bit allocation system, the number of quantizer levels is variable. But if these numbers are constrained to be powers of 2, a shared-quantizer procedure is still possible; all coefficients are first quantized to $2^{R_{\max}}$ levels where $R_{\max}$ is the maximum number of bits allowed in the allocation procedure; subsequently, for coefficient k, the least significant $(R_{\max} - R_k)$ bits in its representation are simply discarded.

0286

12.7.2 Adaptive Bit Allocation

We have already discussed one class of approaches to variable bit allocation — a procedure called *zonal coding*, which uses a priori bit allocations based on log-variance [see (12.60)]; or a simpler procedure based on a priori allocations of zero bits to out-of-zone coefficients in *zonal sampling*, and a constant number of bits to in-zone coefficients. The latter approach can be made more efficient by basing the in-zone coefficient selections on observed magnitudes of coefficients in a particular image rather than a priori. This adaptive procedure, which prevents the suppression of significant frequency components that may lie outside predetermined zones, would be called *threshold sampling*. It is more complex than zonal sampling in that the indices of chosen coefficients have to be transmitted to the decoder, by means of some form of addressing such as run-length coding. It is also possible to keep the total number of bits per block constant by using a variable threshold for transmitted coefficient magnitudes.

In the most general design, the matrix **A** can also be adaptive [Tasto and Wintz, 1971]. However, in most image work, and in the discussions of this chapter, the term ATC will signify variable bit allocation only. If the input is nonstationary, the bit allocation procedure will have to be adapted to local statistics, especially because the transform type, i.e., matrix **A**, itself is usually fixed. Adaptive bit allocation plays a role in TC systems that is comparable to that of adaptive prediction in DPCM (APC). As in that context, adaptations can be accomplished in three ways — partial adaptation as provided by (a) input classifications followed by switching to one of a small number of typical bit allocation rules [Tasto and Wintz, 1971] [Zelinski and Noll, 1975] [Guinlett, 1975]; or fully adaptive bit allocation based on a formal tracking of input spectra (coefficient variances, in general) by (b) *backward* or (c) *forward* estimation procedures.

In the *backward* estimation procedure, the variance of the kth coefficient is estimated either (i) from a time history of the quantized value of *that coefficient over successive blocks*, or (ii) from a recursive relation, *within a given block*, involving the variance estimate for the $(k-1)$th coefficient and its actual quantized value $u(k-1)$. The first coefficient $k = 0$, or the first k_1 coefficients in general, may be nonadaptively quantized, for example, with a very high number of bits, say eight. To start off the recursive estimation procedure reliably, the success of procedure (ii) implies that the variances show a smooth variation with respect to k. Frequency-ordered DFT's and sequency-ordered DWHT's are very helpful in achieving this goal. In image coding, this may involve converting the 2-D coefficients into a specially scanned 1-D array [Cox and Tescher, 1976]. An adaptive procedure of type (ii) above is the following:

$$\hat{\sigma}_k^2 = w\hat{\sigma}_{k-1}^2 + (1-w)u^2(k-1)\ ; \qquad k = 2,3,\ldots,N^2-1 \tag{12.169}$$

$\hat{\sigma}_k^2$ is the estimated variance of quantized coefficient $u(k)$ and w is a weighting term such as 0.75 [Habibi, Samulon and Fultz, 1977]. The average brightness coefficient $\theta(0)$ is quantized nonadaptively with an 8 bit uniform quantizer, $\hat{\sigma}_1^2$ is

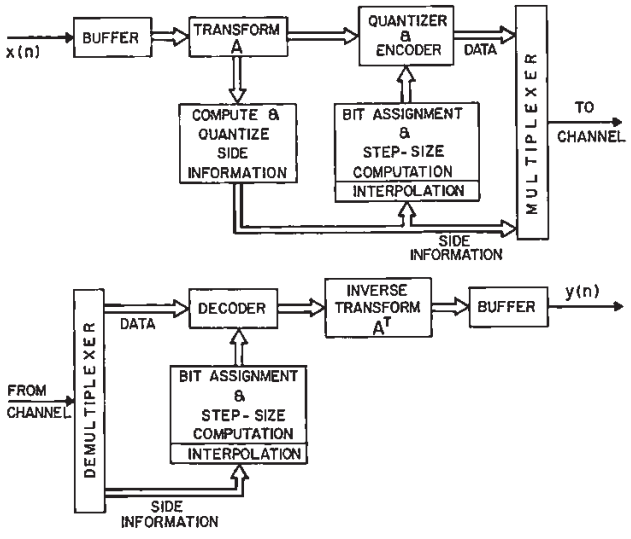
computed as the mean-squared value of the next four coefficients [$\hat{\sigma}_1^2 = \{\theta^2(1) + \theta^2(2) + \theta^2(3) + \theta^2(4)\}/4$] and serves as a starting value of the above recursion. Since this recursion has to be done both at coder and decoder, side information about $\hat{\sigma}_1^2$ has to be explicitly transmitted. Bit allocations follow rules of Section 12.3. Gains over a nonadaptive bit allocation procedure are in excess of 3 dB for $R = 1$ and 2 bits/pixel in 16 × 16 WHT coding [Habibi, Samulon and Fultz, 1977].

In *forward* estimation schemes for variable bit allocation, the actual variance distributions cannot be transmitted if the side information is to be kept small. Instead, the *envelope of the variance distribution*, as described by a few parameters, is used as a coarse description. The envelope parameters are quantized and transmitted as side information; variances are approximately recalculated from this envelope (both at the transmitter and receiver) to compute log-variance-based bit allocations as in Section 12.4. This method is described in detail in the next section.

12.7.3 Adaptive Transform Coding of Speech

A block diagram of an ATC coder with a side information branch appears in Figure 12.21 [Zelinski and Noll, 1977] [Tribolet and Crochiere, 1979] [Cox and Crochiere, 1981] [Crochiere, Cox and Johnston, 1982]. The side information can additionally be used to compute information about the step sizes of different coefficient quantizers, an important requirement in the coding of nonstationary inputs such as speech.

Figure 12.21 Block diagram of an ATC coder with side information transmission for bit allocation and quantizer step size [Zelinski and Noll;©1977, IEEE].

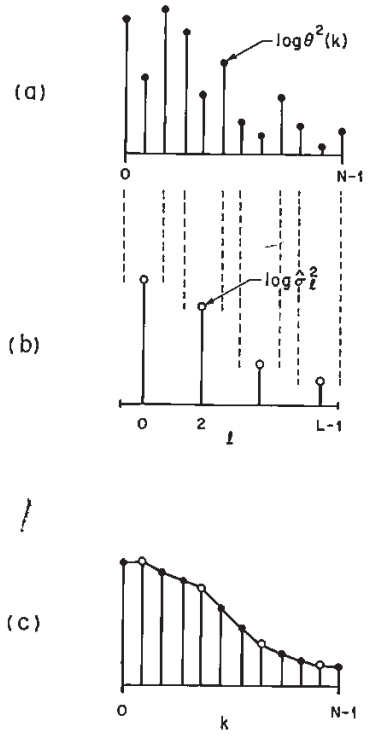


0288

Bit Allocation. Figure 12.22(a) illustrates the squared magnitude of transformed coefficients as a function of coefficient index. Note that although an efficient linear transform decorrelates the coefficients, the actual magnitudes of adjacent coefficients are not unrelated. Rather, they exhibit a slowly varying envelope, perturbed by local, but uncorrelated variations of magnitude.

In one example of speech coding with $N = 128$, sets of 8 to 16 neighboring transform coefficients are therefore squared and averaged, and the averaged values (the L values in Figure 12.22b) are used as support values of an envelope, and recalculations of intermediate individual values $\hat{\sigma}_l^2$ use geometric interpolations, which are equivalent to linear interpolations in the log-domain. The interpolation in the log-domain ensures that low-level coefficients, which are perceptually important, are properly encoded. The encoding of side information requires at least 2 kb/s. A more efficient, but also more complex way of transmitting side information results from using predictor coefficients, or equivalents thereof. These can specify the variance distribution via an autoregressive model (Chapter 6).

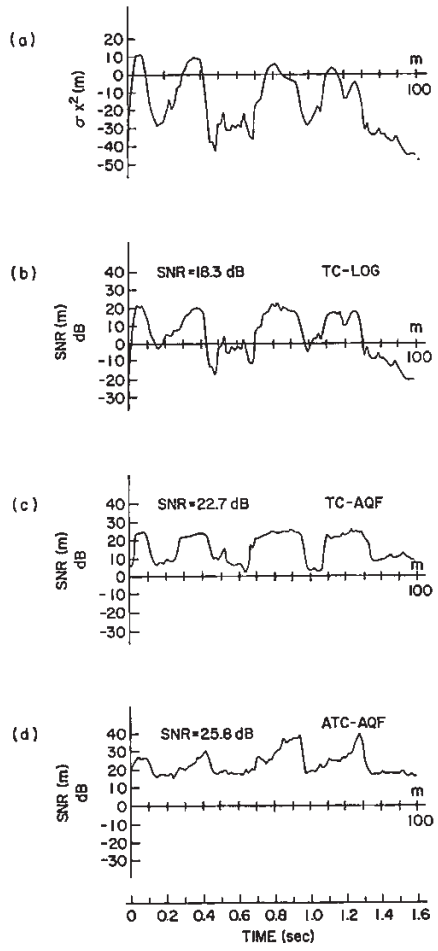
Figure 12.22 Adaptive bit allocation in transform coding of speech using forward estimation of variance: (a) actual squared amplitudes of transform coefficients; (b) support values of envelope obtained by averaging sets of N/L adjacent squared values of (a); and (c) envelope obtained from straight-line interpolation in the log-domain between support values [Zelinski and Noll;©1977, IEEE].



The benefits of adaptive bit allocation are seen in terms of *SNR* results in Figure 12.23, measured with a speech input of duration 1.6 seconds, and a bit rate of $R = 3$ bits/sample.

The figures compare (a) the time-dependence of short-term variance $\sigma_x^2(m)$, where m is the segment number, with (b)(c)(d) the *SNR*-versus-time performance *SNR(m)* of three different TC schemes. The segment length is 16 ms in each case. Part (b) shows that a fixed quantizer, even with logarithmic characteristics, leads to a relatively poor performance. The TC-AQF coder of part (c) employs a forward-

Figure 12.23 Illustration of the benefits of adaptive bit allocation in transform coding of speech at 24 kb/s. Plots of (a) short-time segment variance and short-time *SNR* with (b) TC and log-quantizer, (c) TC-AQF and (d) ATC-AQF [Zelinski and Noll;©1977, IEEE].



0290

adaptive quantizer, again with a fixed bit allocation and achieves an *SNR* gain of over 4 dB. A further increase of 3 dB in *SNR* is obtained with the fully adaptive scheme of part (d), which includes variable bit allocation. These results can be compared with the DPCM-AQF-APF (APC) result (*SNR* = 22.5 dB) of Figure 6.19 for the same speech input. Note that TC-AQF has an *SNR* very similar to that of DPCM-AQF-APF, but ATC-AQF is 3 dB better. Neither the APC coder nor the ATC coder in question employ pitch prediction. The benefits of pitch prediction in DPCM have been discussed in Chapter 6. The benefits of pitch information in ATC will be discussed in the next section.

Side Information Algorithms for ATC Coding of Speech. Figure 12.24 shows a voiced-speech DCT spectrum, two procedures for bit allocation and corresponding reconstructed spectra.

In the first procedure, the side information is based on a smooth spectrum that does not include the effect of pitch-induced fine structure in the input spectrum. This smooth spectrum is obtained following the procedure of Figure 12.22, and it is shown by the dashed lines in Figure 12.24(b)(ii). A consequence of such smoothing is increased zero bit allocation at high frequencies [>2.2 kHz in Figure 12.24(b)], with all available bits getting used to describe the low frequencies. There are many efficient techniques for the derivation of such a smooth spectrum; for example, techniques based on all-pole models.

If, on the other hand, the side information includes the fine structure due to pitch, there are more zero-bit allocations at lower frequencies corresponding to gaps between pitch teeth, and this releases bits that get allocated to high frequency parts in the spectrum [for example, the frequencies in the 2.7 to 3.3 kHz range in Figure 12.24(c)]. The best results in ATC are obtained using this more 'speech-specific', unsmoothed-spectrum-based procedure in Figure 12.24(c), although the greater amount of side information needed in this procedure decreases the value of R, the average number of bits/sample for the quantization of transform coefficients. The 'speech-specific' procedure is also more complex. It involves pitch determination, generation of a pitch pattern in the frequency domain, and multiplication with the smooth spectrum. The resulting spectrum with fine structure is seen in Figure 12.24(c)(ii). The dashed line spectrum in this figure is used for bit allocation as well as for the determination of quantizer step sizes. The improvement due to this method is seen in the reconstructed spectra of Figures 12.24(b) and (c). The speech-specific procedure can be realized very efficiently in systems which transmit a crude version of the so-called *cepstrum* information for providing bit allocation information [Cox and Crochiere, 1981] [Crochiere, Cox and Johnston, 1982] [Zelinski, 1982].

At 16 kb/s and above, both of the above techniques have a similar, if not identical quality. At bit rates under 16 kb/s, the speech-specific algorithm shows significant gains in perceptual performance.

The nature of bit allocation can also be manipulated with a frequency-weighted error criteria as in Figure 12.8(b). Best results for speech are obtained when the error weighting parameter λ in (12.69) is in the order of -0.3; this represents a situation that is midway between the extremes of $\lambda = 0$ [Figure 12.8(a) with white

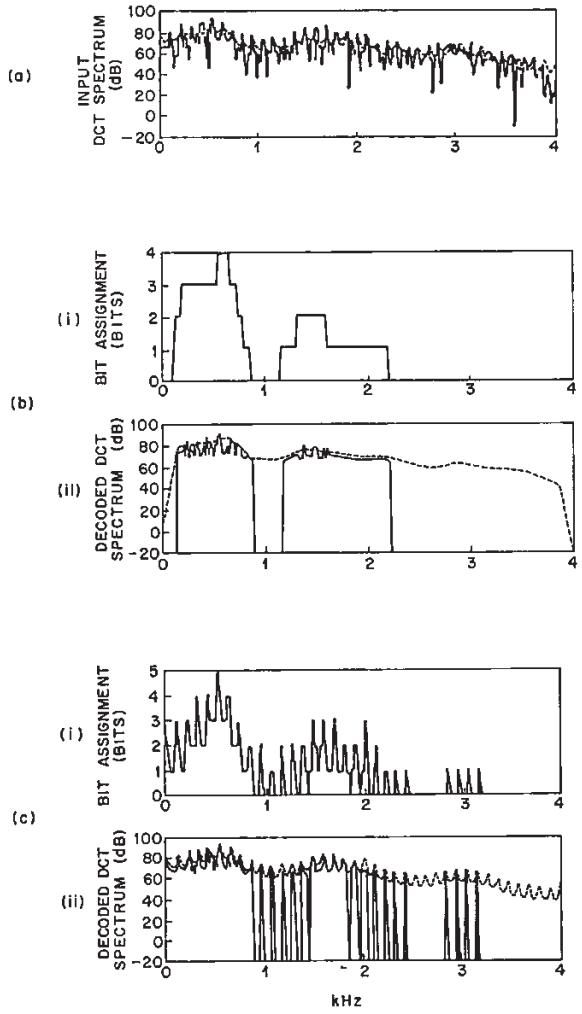


Figure 12.24 (a) Example of DCT spectrum in voiced speech; and bit allocations based on (b) smoothed and (c) unsmoothed short-time spectra. The dashed lines in b(ii) and c(ii) describe the coefficient variance distribution that is used for bit allocation [Tribolet and Crochiere;©1979, IEEE].

error spectrum] and $\lambda = -1$ [constant bit assignment, input psd-type error spectrum]. Indeed, the $\lambda = -0.3$ shaping is very close to that of Figure 12.8(b). The result of optimum error weighting is sketched in Figure 12.25. The non-zero error-weighting is a convenient way of utilizing masking phenomena in perception. One such phenomenon is that the human ear does not perceive noise in a given frequency band if it is more than 20 dB below the signal level in that band.

0292

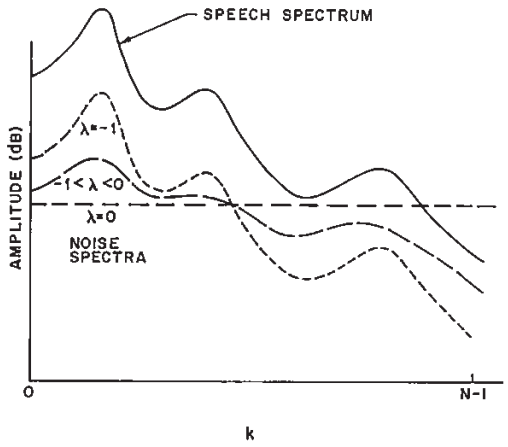


Figure 12.25 The psd of transform coding noise as a function of error-weighting parameter λ in (12.69) [Tribolet and Crochiere;©1979, IEEE].

Switched Bit Allocation. A simpler procedure for bit allocation is one based on *switching*, once for every block, to one of B possible predetermined bit assignment patterns. This approach is similar to switched prediction and vector quantization search techniques, as discussed in the context of APC (Section 6.5.1). The approach has been used in image coding [Tasto and Wintz, 1971] as well as speech coding [Zelinski and Noll, 1979]. The switching may be based on a simple block-classification procedure such as the number of zero crossings per block. With $B = 8$, this type of bit allocation provides a gain over TC-AQF that is about 2 dB, as against the 3 dB gain in ATC-AQF [Zelinski and Noll, 1979]. With a greater value of B, such as $B = 32$, and a more sophisticated classification procedure such as the matching of short-term spectrum to one of B predetermined typical spectral templates, it is possible to obtain a performance that is extremely close to that of ATC-AQF [Heron, Crochiere and Cox, 1982].

Choice of Transform. Figure 12.26 shows the performance of an adaptive coder with different transforms for the three speech sounds (*m*, *ü*, *sch*). It demonstrates once again the near-optimality of the DCT, especially for a voiced input (with adjacent sample correlation $\rho_1 \rightarrow +1$). The results in Figure 12.26 are for no error weighting ($\gamma = 0$) and the smooth-spectrum-based bit allocation procedure in Figure 12.24(b). The maximum TC gains $\max_k\{G_{TC}\}$ refer to KLT with $N \rightarrow \infty$. Note that the greatest value of G_{TC} (for a nasal sound *m*) is equivalent to more than a 4-bit gain over PCM, and the worst case gain (for sound *sch*) is nearly equivalent to a 1-bit advantage.

We have noted earlier that DCT coding offers better end-effect performance than DFT. In speech coding, the choice of the analysis window for the DCT is important in further controlling periodic block-boundary effects which tend to cause a characteristic *burbling distortion* at low bit rates. Trapezoidal windows have

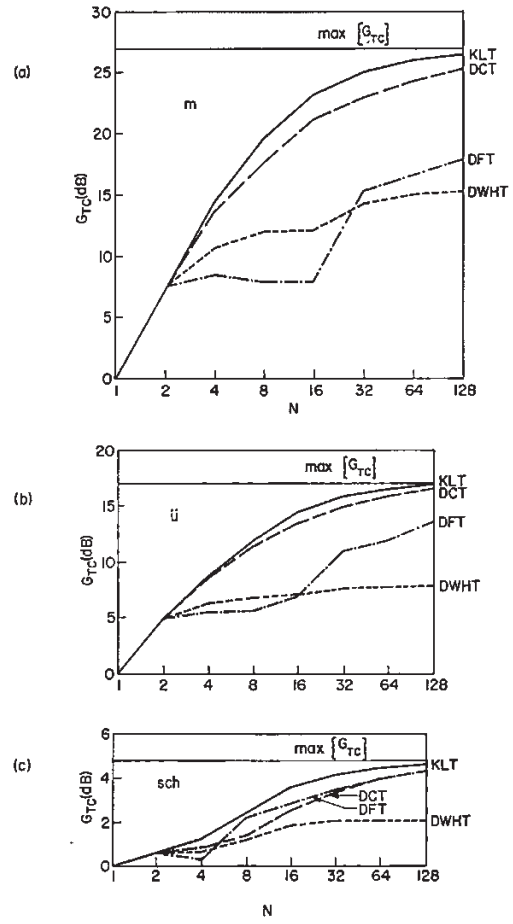


Figure 12.26 Adaptive transform coding gain G_{TC} with different speech inputs [Zelinski and Noll;©1977, IEEE].

been found very useful for low bit rate transform coding of speech [Tribolet and Crochiere, 1979]. By allowing a small overlap between the successive blocks, such that the sum of the overlapped window heights is always unity, an additional reduction of end effect noise is achieved without significantly lowering the number of bits available for encoding each block. As the number of quantizer bits approaches values of $R = 1$ or less, DCT coding degradation takes the form of a *tonal noise*. This is due to the perceptually conspicuous nature of rapid block-to-block changes in bit allocations to adjacent frequency bands.

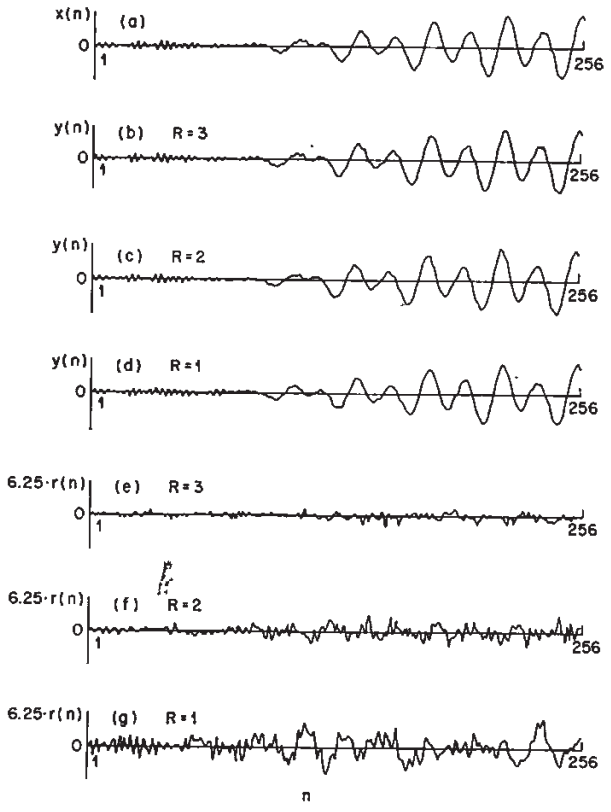
Figure 11.9 has demonstrated the excellent performance of ATC as compared with other redundancy-removing coders for speech waveforms, on the basis of the segmental measure *SNRSEG*. The performance of ATC as a function of bit rate is

0294

further described by the waveforms in Figure 12.27 for $R = 3$, 2 and 1 bit/sample (bit rates $I = 24$, 16 and 8 kb/s, not including side information). The 2-bit output in part (c) of this figure is seen to reproduce the input much better, for example, than the 2-bit DPCM-AQB waveform of Figure 6.12(d). In fact, the noise power in 2-bit ATC is very similar to that in 3-bit DPCM-AQB [Figures 12.27(f) and 6.12(f)]. In addition, the greater speech-like structure in the ATC noise waveform leads to an additional perceptual advantage in the ATC system. Although the reconstruction waveform in 1-bit ATC looks very smooth [Figure 12.27(d)], its quality is limited by distortions that are perceptually extremely obvious.

The error waveform of 24 kb/s ATC implies a value of *SNR* that is much less than that of 8-bit log-PCM. This can be seen from the relative magnitudes in respective waveforms of reconstruction error [Figures 12.27(b) and 1.4(b)]. It is also clear from the *SNR* results of Figure 12.23, which are more than 12 dB lower

Figure 12.27 Adaptive transform coding of speech waveforms: (a) input $x(n)$; (b)(c)(d) reconstruction error waveforms $r(n)$ with (b) $R = 3$, (c) $R = 2$ and (d) $R = 1$ bit per sample. The error waveforms are magnified by a factor of 6.25. The input (a) is the same as that in Figure 1.4 [Cox, 1981].



than the 38 dB *SNR* (Figures 5.7 and 5.10) of 8-bit log-PCM. Subjectively, however, 24 kb/s or 3 bit/sample ATC is very close to 8 bit/sample log-PCM. This is confirmed by MOS quality measurements. With the speech-specific design of Figure 12.24(c), the MOS, on a scale of 1 to 5, was found to be 4.1 at a total bit rate of 24 kb/s, 3.8 at at 16 kb/s and 2.4 at 9.6 kb/s [Daumer, 1982]. This is very close to the variation of MOS with bit rate for the *high complexity* coder of Figure 1.5.

12.7.4 Adaptive Transform Coding of Images

Previous sections have shown examples of transform coded images, including illustrations of zonal coding and/or adaptive bit allocation (Figures 12.13 and 12.17). With adaptive bit allocation and DCT, high quality digitizations (*SNR* approaching 30 dB) are possible in the coding of still images at average rates of $R \simeq 2$ bits/sample, and medium-quality (*SNR* in the range 20 to 25 dB) digitizations are possible at R values of about 1 bit/sample [Mauersberger, 1980]. As in DPCM, interframe techniques are needed to produce high-quality reconstructions of moving images at bits rates in the order of 2 bits/sample and less [Natarajan and Ahmed, 1977].

Example 12.10. Interframe 3-D DCT coding with 1 bit/sample

Figure 12.28 shows the distribution of coefficient variances in a 3D-DCT example. The transformed subimages are chosen to be $4 \times 4 \times 4$ blocks (four frames with 16 pixels each), and in Figure 12.28, coefficients with indices 16 to 63 represent temporal information. Table 12.3 gives a bit allocation procedure. Note that the total number of bits, with summation over 16 coefficients and four frames, is 64, resulting in an average bit rate of 1 bit/sample. The quantizers used for the 64 coefficients have pdf-optimized nonuniform characteristics, with a Laplacian pdf model for all except the first coefficient, which is modeled by a uniform pdf. Figure 12.29 shows original images and reconstructions for

Figure 12.28 Interframe transform coding; DCT coefficient variances in four successive frames [Natarajan and Ahmed;©1977, IEEE].

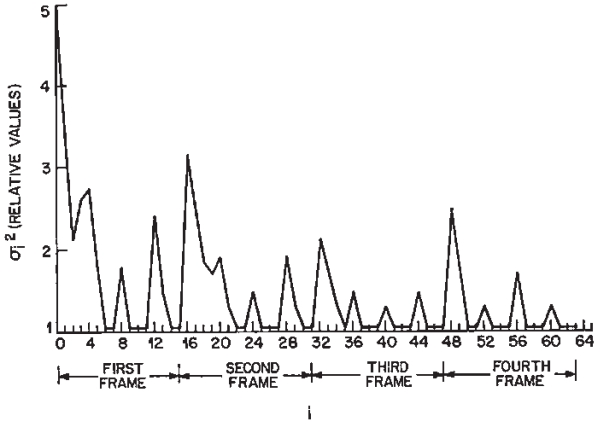


Table 12.3 Bit allocation in a 4×4×4 DCT coder with $R = 1$ bit/pixel. Each column represents R_i values for the 16 coefficients of a given frame [Natarajan and Ahmed;©1977, IEEE].

Frame Number			
1	2	3	4
7	5	3	2
4	3	0	0
2	0	0	0
3	0	0	0
5	4	3	0
3	2	0	0
2	0	0	0
2	0	0	0
3	3	2	0
2	0	0	0
0	0	0	0
0	0	0	0
2	0	0	0
0	0	0	0
2	0	0	0
0	0	0	0

Figure 12.29 Interframe transform coding; (a)(b) frames one and four; (c)(d) the same frames reconstructed using 1 bit/pixel DCT [Natarajan and Ahmed;©1977, IEEE].



(a)



(c)



(b)



(d)

frames 1 and 4. The measured *SNR* is 23 dB with the low-activity image represented in Figure 12.29, and less with an image with more violent motion [Natarajan and Ahmed, 1977]. But in each case, there are no visible distortions or motion degradations in the reconstructed moving images, based on comparison with black-and-white originals with 6 bit PCM samples.

Example 12.11 Interframe 2-D DCT/DPCM coding.

We discussed a nonadaptive hybrid coder in Section 12.6.5 with special reference to Markov inputs. The performance of this coder at low bit rates is less impressive with real images because of their nonstationarity and temporal characteristics. However, with adaptive bit allocation in the 2-D TC system, and adaptive prediction in the temporal DPCM system, a 16×16×16 hybrid coder realizes *SNR* values as high as 37 dB at $R = 1$ bit/pixel, with no degradation of perceived image quality. Coder performance is however image-and-motion-dependent, and certain classes of moving images in broadcast-standard color television may require bit rates as high as 2 bits/sample for high-quality encoding [Kamanagar and Rao, 1981]. The coder is robust with random bit error rates of up to 10^{-4} [Roese, Pratt and Robinson, 1977]. •

12.8 Transmission Error Effects

As in the case of earlier chapters, we shall consider transmission error effects on mse performance as well as on the subjective quality of digitized speech and images.

12.8.1 Effect on Reconstruction Error Variance

Recall that in PCM (Chapters 4 and 5), the reconstruction error in the presence of transmission errors had the form $r(n) = q(n) + c(n)$, with a variance

$$\sigma^2_{r,PCM} = \sigma_q^2 + \sigma_c^2 = (\epsilon_q^2 + \epsilon_c^2)\sigma_x^2 \; ; \quad \epsilon_c^2 = \xi p_e \text{ with single errors} \qquad (12.170a)$$

provided that $c(n)$ and $q(n)$ are uncorrelated (as with Max-quantizers). We recall that p_e is the bit error rate and ξ is the channel coefficient, a function of bit rate R, pdf of the quantizer input, and the binary code.

In DPCM (Chapter 6), the reconstruction error was $r(n) = q(n) + c(n) * g(n)$ with a variance

$$\sigma^2_{r,DPCM} = (\epsilon_q^2 + \alpha\epsilon_c^2)\sigma_x^2/G_P \; ; \quad \alpha = \sum_{k=0}^{\infty} h^2_{DEC}(k) \geq 1 \qquad (12.170b)$$

where $h_{DEC}(k)$ is the impulse response of the decoder filter. The term α reflects error propagation; but the channel error contribution $\sigma^2_{c,DPCM} = \alpha\epsilon_c^2/G_P$ equals $\sigma^2_{c,PCM}$ if one uses a mmse predictor for which $\max\{G_P\} = \alpha$ [see (6.132)]:

$$\sigma^2_{c,DPCM} = \sigma^2_{c,PCM} \qquad \text{with mmse prediction} \qquad (12.170c)$$

We will now show that a similar result holds in the case of TC.

The reconstructed output $\mathbf{y}$ in TC can be described as a superposition of weighted basis vectors (or images) in the form (12.20b):

$$\mathbf{y} = \sum_{k=0}^{N-1} v(k)\mathbf{b}_k \tag{12.171}$$

where $v(k)$ are received versions of quantized transform coefficients. With channel errors, $v(k) = u(k) - c(k)$ and the output will differ from the intended output $\sum u(k)\mathbf{b}_k$ by a term

$$\mathbf{r} = \mathbf{x} - \mathbf{y} = \sum_{k=0}^{N-1} q(k)\mathbf{b}_k + \sum_{k=0}^{N-1} c(k)\mathbf{b}_k \tag{12.172}$$

i.e., by a superposition of basis vectors weighted by the sum of quantization error and channel error components in $v(k)$. The summations in (12.172) indicate that channel errors propagate beyond a single sample; they affect, to different extents, all the reconstructions within a block of length N (or subimage of size $N \times N$ in a 2-D transform).

Assuming uncorrelated error components $q(k)$ and $c(k)$ and orthonormal basis vectors ($\mathbf{b}_i^T\mathbf{b}_j = \delta_{ij}$), the reconstruction error variance is

$$\sigma_{r,TC}^2 = \frac{1}{N} E[(\mathbf{X} - \mathbf{Y})^T(\mathbf{X} - \mathbf{Y})] = \sigma_q^2 + \sigma_c^2 \tag{12.173}$$

with σ_q^2 following (12.61) and with

$$\sigma_{c,TC}^2 = \frac{1}{N} \sum_{k=0}^{N-1} E[C^2(k)] = \frac{1}{N} \sum_{k=0}^{N-1} \sigma_{ck}^2 \tag{12.174}$$

The channel error variance σ_{ck}^2 that is added to the kth transform coefficient $\theta(k)$ of variance σ_k^2 can be given (as in 12.170a) by

$$\sigma_{ck}^2 = \xi p_e \sigma_k^2 \quad \text{for all } k \tag{12.175}$$

Here we have assumed that all channel coefficients ξ are of same value, i.e., they do not depend on the bit rate R_k for the given sample. Earlier results (Chapter 4) show that this approximation is valid *as long as R_k is not very small.* From (12.174) and (12.175), we obtain the result [Zelinski, 1979]

$$\sigma_{c,TC}^2 = \frac{1}{N} \xi p_e \sum_{k=0}^{N-1} \sigma_k^2 = \xi p_e \sigma_x^2 = \sigma_{c,PCM}^2 \tag{12.176}$$

i.e., channel errors in TC produce the same error variance in the output as in PCM. This result is the counterpart of (12.170c). There, as in the derivation leading to (12.176), it is assumed that all ξ values are equal. Note also that the assumption of a constant ξ leads to an expression for $\sigma_{c,TC}^2$ that is independent of bit allocation and transform matrix. The above result should be carefully interpreted; as in DPCM, the channel error variance, *relative* to quantization error variance, has actually *increased*; in TC, this increase is by a factor G_{TC}.

12.8.2 Results for Speech and Images

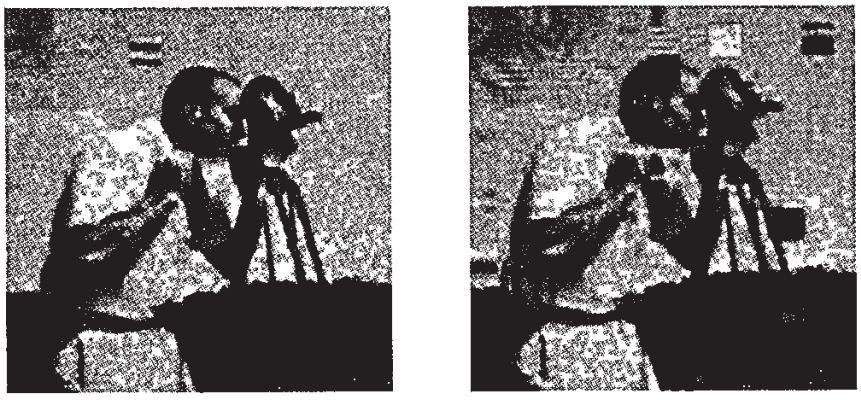
Subjectively, TC is more robust to transmission errors than PCM, in both speech and image coding. Tolerable random error rates are in the order of 10^{-3} and 10^{-4} for low bit rate speech and video in the absence of explicit error protection.

In ATC coding of speech, the error-sensitive part of the coder output is the side information that carries bit allocation information. In low bit rate ATC (say, at 16 kb/s), the side information can constitute up to about 20% of the total coder output. If this side information is not error-protected, output speech is poor in quality at $p_e = 10^{-3}$ [Cox, 1980]. The speech is intelligible, but is degraded by screeches, clicks, pops and other loud sounds. With $p_e = 10^{-2}$, there is significant loss of intelligibility as well. On the other hand, in a system where about 1 kb/s (Problem 12.33) is dedicated to protection of side information, output speech quality is good at $p_e = 10^{-3}$ and acceptable even at $p_e = 10^{-2}$.

With nonadaptive transform coders, the effect of a bit error does not propagate beyond the block because only the samples within a transform subblock depend on a wrongly decoded coefficient. The exact nature of the degradation depends upon the type of transform used and the coefficients that are received in error.

In image coding, errors in lower frequency coefficients are more visible than higher frequency coefficients due to the fact that above a certain frequency the sensitivity of the eye decreases with frequency. Also, the probability of errors occurring in the lower frequency coefficients is higher due to larger number of bits that are required to code them. As the block size decreases, the averaging property (i.e., spreading of error over the entire block) decreases, and the transmission errors appear as blotches in the picture. Results for 16 × 16 DFT coding with 2:1 zonal sampling and $R = 1$ appear in Figure 12.30 for $p_e = 10^{-3}$ and $p_e = 10^{-2}$. Note

Figure 12.30 Effects of random transmission errors in 2-D transform coding with DFT-based 2:1 zonal sampling, $R = 1$ bit/sample and p_e values of (a) 10^{-3} and (b) 10^{-2}. The uncoded 8-bit PCM image is shown in Figure 12.6(c) [Wintz;©1972, IEEE].



(a) (b)

the effects of superimposed basis vectors as indicated by (12.172). Note also that effects do not propagate beyond the transform sub-block. Error visibility is much less than the streaks that result in 1-D DPCM (Figure 6.33) at similar error rates, but it is comparable to the error visibility in well-designed 2-D DPCM (Figure 6.34). Also, as in DPCM, error concealment procedures can be devised that significantly enhance reconstructed quality with only a slight increase of total transmission rate; or with no increase of total transmission rate in the case of techniques that exploit the natural redundancies in images [Wong and Steele, 1978].

Effect of Error-Protection on TC and PCM. The result (12.176) indicates that TC has no advantage over PCM with noisy channels. However, we will show next that TC has an advantage over PCM if explicit error-protection is employed. The advantage comes out of the possibility of protecting the more important coefficients more carefully, a possibility already noted in Section 11.6.

Consider a PCM system where the most significant bits are ideally error-protected. Let e be the number of protected bits per sample. An analysis of channel coefficients [Zelinski, 1979] shows that channel error variance is approximately reduced by a factor of 4 for each protected bit:

$$\sigma_{c,PCM}^2 \simeq \xi 2^{-2e} p_e \sigma_x^2 ; \qquad e = 0,1,...,R \tag{12.177}$$

This result holds for the NBC and Gray code, and also for the FBC code with the exception of its first bit. The above PCM system needs a side information for error protection that depends on e. In a TC system with an identical fraction of side information, Ne bits per *block* are available to protect the transform coefficients. However, instead of allocating e of these protection bits to each coefficient, we can now make use of the hierarchy of coefficient variances σ_k^2 and assign e_k bits to coefficient of index k; and we can thereby minimize

$$\sigma_{c,TC}^2 = \frac{1}{N} \xi p_e \sum_{k=0}^{N-1} 2^{-2e_k} \sigma_k^2 \tag{12.178a}$$

under the constraint

$$e = \frac{1}{N} \sum_{k=0}^{N-1} e_k = \text{constant} \tag{12.178b}$$

The derivation of optimum bit allocation for error protection as well as the gain over an error-protected system follow arguments very similar to those used for deriving the bit allocation that minimized quantization error variance in Section 12.4. The end results are also very similar. The optimum number of protection bits [Zelinski, 1979] is

$$e_k = e + \frac{1}{2} \log_2 \frac{\sigma_k^2}{\left[\prod_{j=0}^{N-1} \sigma_j^2\right]^{1/N}} \tag{12.179}$$

Using (12.178a),

$$\left.\frac{\sigma_{c,PCM}^2}{\sigma_{c,TC}^2}\right|_{e>0} \sim G_{TC} \tag{12.180}$$

The channel error variance in TC can be made smaller than that in a comparable PCM scheme by a factor of G_{TC} if the same (average) number of bits is error-protected. An ideally error-protected TC system can therefore reduce both quantization error and channel error by the same factor G_{TC}, so that

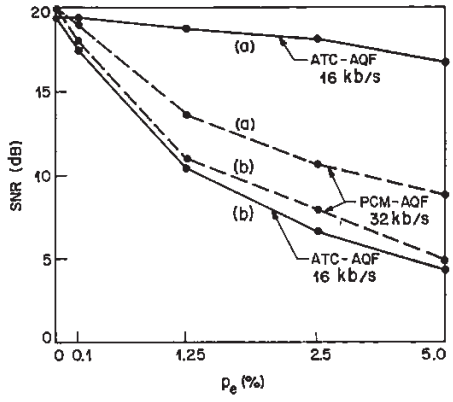
$$\sigma_r^2 = (\epsilon_q^2 + \epsilon_c^2)\sigma_x^2/G_{TC} \tag{12.181}$$

The analytical discussion above has assumed nonadaptive coding and an mse criterion for simplicity. It has been shown that, in the absence of error protection, the mse due to channel errors is the same in TC and PCM, while with ideal error protection, TC offers significantly better transmission error performance [Zelinski, 1979]. The following example shows that the results also apply to adaptive schemes.

Example 12.12. Error-protected ATC and PCM systems for speech

Figure 12.31 compares *SNR*-versus-p_e results with a real speech input, for 32 kb/s PCM-AQF and 16 kb/s ATC, both of which use a block length of $N = 128$. The measured value of G_{TC} is 12 dB. Without error protection, both schemes perform within 1 dB of each other. But with error protection of 1 bit/sample (PCM) and of 128 bits/block [ATC with optimum protection allocation (12.179)], PCM performance improves by about 6 dB as expected from (12.177), while TC performance improves by an *additional* 10.8 to 12.5 dB as predicted by (12.180). Note that Figure 12.31 has assumed error-free transmission of side information for simplicity. Results are therefore optimistic representations of performance. •

Figure 12.31 *SNR* versus p_e in PCM-AQF and ATC-AQF speech-coding systems (a) with error protection and (b) without error protection. In (a), the protection is provided for 1 bit in each sample in PCM, and for a total of 128 bits in a 128-sample block in ATC [Zelinski and Noll, 1975].



Problems

(12.1) Show that for $N = 2$ the most general orthonormal transform is that given by (12.10a).

(12.2) Consider the (2×2) subimage

$$\mathbf{X} = \begin{bmatrix} x(0,0) & x(0,1) \\ x(1,0) & x(1,1) \end{bmatrix}$$

and the separable transform based on (12.30) with $\mathbf{A}_v$ and $\mathbf{A}_h$ based on (12.10b).

(a) Show that the coefficient matrix is

$$\boldsymbol{\Theta} = \frac{1}{2} \begin{bmatrix} x(0,0)+x(0,1)+x(1,0)+x(1,1) & x(0,0)-x(0,1)+x(1,0)-x(1,1) \\ x(0,0)+x(0,1)-x(1,0)-x(1,1) & x(0,0)-x(0,1)-x(1,0)+x(1,1) \end{bmatrix}$$

Note that the coefficients can be interpreted respectively as scaled values of average pixel intensity, average intensity difference between columns, average intensity difference between rows, and finally, a difference between row and column differences.

(b) Use (12.33a) to show that the basis images implied by $\mathbf{A}_v$ and $\mathbf{A}_h$ above are those in (12.34) and Figure 12.6(b).

(12.3) It has been stated in Section 12.3 that the first 64 DWHT basis images of Figure 12.12 can be obtained by using the 8 DWHT vectors from Figure 12.5(a) in (12.33a). Show this result for the basis image B_{43} (column 5, row 4) of Figure 12.12.

(12.4) Consider the orthogonal transform in (12.10). Assume that $\sigma_x^2 = 1$.

(a) Show that for an adjacent sample correlation of $\rho_{xx}(1) = \rho$, the geometric mean of coefficient variances is $[1-4a^2(1-a^2)\rho^2]^{1/2}$.

(b) Show that the geometric mean is minimized [and therefore ${}^2G_{TC}$ in (12.57) is maximized] when $a = 1/\sqrt{2}$, which leads to the well-known transform (12.10b).

(c) For the above value of a, show that

$${}^2G_{TC} = (1-\rho^2)^{-1/2}$$

Compare the TC gain above with DPCM and D*PCM gains for a first-order predictor:

$${}^1G_{DPCM} = (1-\rho^2)^{-1}; \quad {}^1G_{D*PCM} = (1-\rho^2)^{-1/2}$$

(12.5) Consider the transform (12.10b), and an input with variance σ_x^2 and $\rho_{xx}(1) = \rho = 0.85$.

(a) Show that the coefficient variances are

$$\sigma_0^2 = E[\Theta^2(0)] = \sigma_x^2(1+\rho) = 1.85\sigma_x^2; \quad \sigma_1^2 = E[\Theta^2(1)] = \sigma_x^2(1-\rho) = 0.15\sigma_x^2.$$

0303

(b) Show that $E[\Theta(0)\Theta(1)] = 0$.

(c) For an average rate $R = 1.5$, and $\sigma_x^2 = \epsilon_q^2 = 1$, show that a minimum reconstruction error variance of 0.065 is realized with $R_0 = 2.41$ and $R_1 = 0.59$, and that this corresponds to $\max\{G_{TC}\} = 2.78$ dB.

(d) Show that if $\sigma_x^2 = 1$, the values of error variances corresponding to $R_0 = 3,2,1,0$ and $R_1 = 3 - R_0$, are 0.089, 0.077, 0.236 and 0.926, respectively.

(12.6) (a) Consider a two-dimensional image source with adjacent sample correlation ρ in both horizontal and vertical directions. Show that with a "half-transform" $\boldsymbol{\Theta} = \mathbf{AX}$ where $\mathbf{A}$ is the matrix (12.10b), the ratio of arithmetic and geometric means of coefficient variances is $(1-\rho^2)^{-1/2}$, the one-dimensional result ${}^2G_{TC}$ in Problem 12.4.

(b) Show that the arithmetic to geometric mean ratio increases to $(1-\rho^2)^{-1}$ with a "full-transform" $\boldsymbol{\Theta} = \mathbf{AXA}^T$ as in (12.31), assuming a separable acf model (with a the correlation ρ^2 between diagonally opposite elements of $\mathbf{X}$).

(c) Show that for $\rho = 0.95$, the arithmetic-to-geometric mean ratios in (a) and (b) are $\sqrt{10.26}$ and 10.26 respectively.

(d) Consider the transform $\boldsymbol{\Theta} = \mathbf{AXA}^T$ as in (b) but an isotropic acf model with a correlation between diagonally opposite elements of $\rho^{\sqrt{2}}$. Show that the arithmetic-to-geometric mean ratio for $\rho = 0.95$ is 6.4, reflecting a 2 dB loss in performance compared to the separable acf case.

(12.7) Assume a quantizer performance factor of $\epsilon_q^2 = 1$, an average rate of $R = 2$, $\sigma_x^2 = 1$ and $\rho = 0.95$ for a (2×2) image. Show that error-minimizing bit assignments for the examples (a), (b) and (d) in Problem 12.6 are

$$\begin{bmatrix} 3 & 3 \\ 1 & 1 \end{bmatrix} \quad \begin{bmatrix} 4 & 2 \\ 2 & 0 \end{bmatrix} \quad \text{and} \quad \begin{bmatrix} 3 & 2 \\ 2 & 1 \end{bmatrix}$$

with corresponding reconstruction error variances of (a) 22/1024, (b) 8/1024 and (c) 20/1024. The bit allocation (b) above implies that $\theta(0,0)$ is quantized using 4 bits, $\theta(0,1)$ and $\theta(1,0)$ with 2 bits each, and $\theta(1,1)$ is not transmitted at all.

(12.8) Consider the two-level psd of Figure 2.24. Show that the transform-coding gain with $N = 2$ would be

$${}^2G_{TC} = \left[1 - \left[\frac{2}{\pi}(1-\alpha)\right]^2\right]^{-1/2}$$

Note that this is less than the asymptotic gain $[\alpha(2-\alpha)]^{-1/2}$ [Problem 2.22(c)] for all $0 < \alpha < 1$. Discuss the special cases of $\alpha = 0$ and $\alpha = 1$.

(12.9) Consider the problem of minimizing a frequency-weighted reconstruction error variance

$$\sigma_q^2 = \frac{1}{N} \sum_{k=0}^{N-1} W_k \sigma_{qk}^2$$

subject to an average bit rate constraint R as in (12.46). Show that the bit allocation that minimizes the above expression is

$$R_k = R + \frac{1}{2}\log_2 \frac{W_k\sigma_k^2}{\left[\prod_{j=0}^{N-1} W_j\sigma_j^2\right]^{1/N}}\ ; \quad k = 0,1,...,N-1$$

with an error spectrum described by

$$W_k\sigma_{qk}^2 = W_l\sigma_{ql}^2 \quad ; \quad k,l = 0,1,...N-1$$

In the above design, the *weighted*-noise variances are equal for all coefficients.

(12.10) (a) Show that, with $W_k = (\sigma_k^2)^\lambda$ in (12.69) and $N = 2$, the optimum bit allocation is given by

$$R_0 = R + \frac{\lambda+1}{2}\log_2 \frac{\sigma_0}{\sigma_1}$$

(b) For the example of Problem 12.5, show that $\lambda = -0.5$ leads to a bit assignment (R_0=1.96, $R_1 = 1.04$), while $\lambda = +0.5$ leads to ($R_0 = 2.88$, $R_1 = 0.12$).

(12.11) Consider the four-sample input sequence

$$\{x(n)\} = \{17,19,21,18\}$$

Determine the output sequence of a transform coding system based on the Discrete Walsh Hadamard transform $^4\mathbf{H}'$, and a zonal coding procedure where the first two transform coefficients are quantized with zero error, and the last two coefficients are discarded, and replaced by zero values at the decoder.

(12.12) Consider the 2×2 image $\mathbf{X}$ of Figure 12.6(a):

$$\mathbf{X} = \begin{bmatrix} 35 & 30 \\ 25 & 15 \end{bmatrix}$$

and the two-dimensional transform (12.30) $\boldsymbol{\Theta} = \mathbf{A}_v\mathbf{X}\mathbf{A}_h^T$ where $\mathbf{A}_v$ and $\mathbf{A}_h$ are defined by (12.10b). Assume a zonal sampling procedure that zeroes second-column coefficients a priori, and represents the first-column coefficients with negligible quantization error:

$$\hat{\boldsymbol{\Theta}} = \begin{bmatrix} \Theta(0,0) & 0 \\ \Theta(1,0) & 0 \end{bmatrix}$$

(a) Show that the reconstructed image is

$$\hat{\mathbf{X}} = \begin{bmatrix} 32.5 & 32.5 \\ 20.0 & 20.0 \end{bmatrix}$$

0305

(b) Show that the average reconstruction error is 15.625, or nearly 28% of input variance, assuming a mean of 26.25 for $\mathbf{X}$. Derive that result by a direct comparison of $\mathbf{X}$ and $\hat{\mathbf{X}}$.

(c) Derive the above value of σ_{zz}^2 by using (12.76).

While the above example illustrates the nature of zonal sampling in a simple fashion, it should be recognized as a purely deterministic illustration. The average reconstruction error of 15.625, for example, is simply an average of the error in representing the four components of the specific inputs $\mathbf{X}$ of that problem, not an expectation over a statistically characterized ensemble of (2×2) images.

(12.13) Assume a separable image model and adjacent sample correlation ρ in both horizontal and vertical directions. With $\sigma_x^2 = 1$, show that the variances of coefficients in a 2×2 transform based on (12.31) and (12.10b) are the entries in

$$\begin{bmatrix} (1+\rho)^2 & 1-\rho^2 \\ 1-\rho^2 & (1-\rho)^2 \end{bmatrix}$$

(12.14) Assuming that the quantizer performance given by $\sigma_{qi}^2 = 2^{-2R_i}\sigma_i^2$, and a value of $\rho = 0.95$, derive the bit allocations and reconstruction error variances given in the following table, for PCM, TC-ZS (2:1 zonal sampling) and ATC, using the image transform of Example 12.13. (The first two values of R in each set refer to the bits allocated to the first column coefficients).

Coding Scheme	$(R = 2)$ R_i-Values	$mse = \sigma_q^2$	$(R = 1)$ R_i-values	$mse = \sigma_q^2$
PCM	(2,2,2,2)	64/1024	(1,1,1,1)	256/1024
2:1 Zonal Sampling with $n = N/2 = 2$	(4,4,0,0)	30/1024	(2,2,0,0)	88/1024
Optimum Bit Allocation	(4,2,2,0)	8/1024	(3,1,0,0)	48/1024

For $R = 1$, show also that 4:1 zonal sampling with a bit allocation of (4,0,0,0) gives an mse of (56/1024) as against (88/1024) for 2:1 zonal sampling.

(12.15) For inputs of block length $N = 2$, consider 2:1 zonal sampling in the *time domain*: $y(0) = x(0)$; (a) $y(1) = 0$, or (b) $y(1) = y(0)$, i.e., the untransmitted sample is set at the receiver either (a) to zero, or (b) to the value of its predecessor. Show that the respective reconstruction error variances are (a) $\sigma_x^2/2$ and (b) $\sigma_x^2(1-\rho)$, where $\rho = R_{xx}(1)/\sigma_x^2$.

Now use $\mathbf{A}$ of (12.10b), set $\theta(0) = \theta(0)$ and set (c) $\theta(1) = \theta(0)$, or (d) $\theta(1) = 0$ (2:1 zonal sampling in the *transform domain*). Show that the respective error variances are σ_x^2 and $\sigma_x^2(1-\rho)/2$.

Note that the results hold for any input with variance σ_x^2 and acf $\rho_{xx}(1) = \rho$. Note also that with the zon. coding in (d), a gain over (a) and (b) results for any $\rho > 0$. In particular, with $\rho = 0.95$, show that the schemes (a), (b), (c), (d) have *SNR* values of 3, 13, 0 and 16 dB, respectively.

(12.16) Consider inputs of block length $N = 2$, assume 2:1 zonal sampling, and a process with a two-level psd (Figure 2.24). Verify that the error variance is

$$\sigma_{zs}^2 = \frac{1}{2}\left[1 - \frac{2}{\pi}(1-\alpha)\right]\sigma_x^2$$

Note that for the 2:1 zonal sampling of a white input ($\alpha = 1$), $\sigma_{zs}^2 = \sigma_x^2/2$.

(12.17) Consider the non-Toeplitz matrix

$$\begin{bmatrix} 1 & 2 \\ 4 & 3 \end{bmatrix}$$

(a) Show that its eigenvalues are $\lambda_0 = -1$ and $\lambda_1 = 5$.

(b) Compute the orthonormal eigenvectors.

(12.18) Simplify the results (12.108), (12.109) and (12.110) for the case of $N = 2$.

(12.19) Show that the transform coding gain for the numbers in Table 12.1 for $N = 8$ exceeds that for $N = 2$ in (12.99) by 4 dB. Also see Figure 12.10.

(12.20) Consider an MA(1) source.

(a) Compute the KLT matrix for $N = 3$.

(b) Verify that the transform coefficient variances are

$$\sigma_0^2 = \sigma_x^2(1 + \sqrt{2}\rho)\ ; \quad \sigma_1^2 = \sigma_x^2; \quad \sigma_2^2 = \sigma_x^2(1 - \sqrt{2}\rho)$$

(c) Hence show that the transform coding gain is $(1 - 2\rho^2)^{-1/3}$.

(12.21) Use the results of Example 12.1 and Table 12.1 to discuss the effects of 2:1 zonal sampling for $N = 2$ and 8, and 4:1 zonal sampling for $N = 8$. Calculate the value of $SNR = \sigma_x^2/\sigma_{zs}^2$ in each case.

(12.22) Consider an input with $\sigma_x^2 = 1$ and acf values $\rho_{xx}(n) = \rho_n$; $n = 0,1,2,3$, and a linear transform of it using $^4\mathbf{H}$ (12.135a).

(a) Show that the coefficient variances are $(\rho_0+1.5\rho_1+\rho_2+0.5\rho_3)$, $(\rho_0-1.5\rho_1+\rho_2-0.5\rho_3)$, $(\rho_0+0.5\rho_1-\rho_2-0.5\rho_3)$ and $(\rho_0-0.5\rho_1-\rho_2+0.5\rho_3)$.

(b) Show therefore that G_{TC} values for a (i) speech input (ρ_0=1.0, ρ_1=0.825, ρ_2=0.562 and ρ_3=0.31) and a (ii) voiceband data signal (ρ_0=1.0,ρ_1=0.15, ρ_2=−0.75 and ρ_3=−0.26) are 2.0 and 1.6, respectively.

(c) Show also that corresponding gains are (i) 1.77 and (ii) 1.01 for the $^2\mathbf{H}$ transform (12.132).

(12.23) Consider the transform $^4\mathbf{H}$ in (12.135a).

(a) Show that $^4\mathbf{H}$ does not diagonalize the Toeplitz matrix defined by $\mathbf{R}_{xx}(i,j) = \rho^{|i-j|}$.

(b) Show, however, that $^4\mathbf{H}$ diagonalizes a *dyadic* covariance matrix

$$\mathbf{R}_{dd} = \begin{bmatrix} \rho_0 & \rho_1 & \rho_2 & \rho_3 \\ \rho_1 & \rho_0 & \rho_3 & \rho_2 \\ \rho_2 & \rho_3 & \rho_0 & \rho_1 \\ \rho_3 & \rho_2 & \rho_1 & \rho_0 \end{bmatrix}$$

and that diagonal elements in the output acf are $(\rho_0+\rho_1+\rho_2+\rho_3)$, $(\rho_0-\rho_1+\rho_2-\rho_3)$, $(\rho_0+\rho_1-\rho_2-\rho_3)$ and $(\rho_0-\rho_1-\rho_2+\rho_3)$.

(12.24) Show that in DFT representation

$$\sum_{k=0}^{N-1} |\theta(k)|^2 = \sum_{k=0}^{N-1} x^2(k)$$

Note that the right-hand term defines the energy in the sequence $x(n)$.

(12.25) Show the following property of conjugate symmetry for 2-D DFT:

$$\theta(k,l) = \theta^*\ (mN-k,\ nN-l)$$

with $m,n = 0,1,2....$ Show that there are $(N^2/2)+2$ non-redundant complex coefficients and N^2 non-redundant numbers in the 2-D DFT representation.

(12.26) Use the mmse rule (12.60) to derive the optimum bit allocation $\{R_k\}$ for the coefficient variances listed in Table 12.2. Compare the result with the bit-allocation rule used in Table 12.2.

(12.27) Give the form of the basic vectors $\mathbf{b}_k'$ corresponding to the real-valued Fourier matrix $\mathbf{F}'$ and show their orthonormality.

(12.28) Consider the input of Problem 12.23. Assume that a DFT (12.142) is used instead of the DWHT.

(a) Show that the coefficient variances $E[\Theta^2(k)]$ are $(4+6\rho_1+4\rho_2+2\rho_3)$, $(4-4\rho_2)$, $(4-6\rho_1+4\rho_2-2\rho_3)$ and $(4-4\rho_2)$.

(b) Hence show that the variance geometric means for the (i) speech and (ii) voiceband data examples are 0.57 and 0.64, respectively, corresponding to G_{TC} values of 1.75 and 1.6. Note that in the speech example, the performance of $^4\mathbf{F}$ is worse than that of $^4\mathbf{H}$ and nearly equal to that of $^2\mathbf{F}$. (The $N = 2$ transform $^2\mathbf{H} = {}^2\mathbf{F}$ had gains of 1.7 and 1.01 for speech and data signal statistics).

(12.29) (a) Show that $^4\mathbf{F}$ in (12.142) does not diagonalize the Toeplitz matrix defined by $\mathbf{R}_{xx}(i,j) = \rho^{|i-j|}$.

(b) Show, however, that $^4\mathbf{F}$ diagonalizes a *circular* covariance matrix $\mathbf{R}_{cc}$ in the form

$$\mathbf{R}_{cc} = \begin{bmatrix} \rho_0 & \rho_1 & \rho_2 & \rho_1 \\ \rho_1 & \rho_0 & \rho_1 & \rho_2 \\ \rho_2 & \rho_1 & \rho_0 & \rho_1 \\ \rho_1 & \rho_2 & \rho_1 & \rho_0 \end{bmatrix}$$

Show that resulting diagonal elements in the coefficient acf matrix are $(\rho_0+2\rho_1+\rho_2)$, $(\rho_0-\rho_2)$, $(\rho_0-2\rho_1+\rho_2)$ and $(\rho_0-\rho_2)$. Note that $\mathbf{R}_{cc}$ is not Toeplitz; the deviation is in the top right and bottom left corners. Note also that all rows are circularly shifted versions of the first row.

(12.30) Consider the input of Problem 12.22 with $\sigma_x^2 = 1$, and the transform $^4\mathbf{C}$ in (12.160).

(a) Show that coefficient variances are $(4+6.3\rho_1+3.4\rho_2+1.1\rho_3)$, $(4+2.6\rho_1-3.4\rho_2-2.6\rho_3)$, $(4-2.6\rho_1-3.4\rho_2+2.6\rho_2)$ and $(4-6.3\rho_1+3.4\rho_2-1.1\rho_3)$.

(b) Hence show that the gains G_{TC} for the (i) speech and (ii) data signal examples in Problem 12.22 are 2.2 and 1.4, respectively. Note that the performance of DCT exceeds that of DHT and DFT for the speech example. Note also that the acf values of speech resemble an AR(1)-acf sequence ρ^n; $n = 0,1,2,3$ (with ρ=0.825), while the acf values of the data signal do not.

(12.31) Show that 2-D DCT can be obtained by sequential row and column operations involving 1-D transforms in the form $\boldsymbol{\Theta} = \mathbf{CXC}^T$ where $\mathbf{C}$ is that defined in (12.157a).

(12.32) Assume that the short-term spectrum of a speech signal can be approximately reconstructed by specifying pitch, spectral envelope and absolute spectrum level information using 32 bits of information, once every 16 ms. The spectral envelope information may be conveyed in one of several equivalent ways, such as AR-parameter, or *parcor* values.

(a) Consider that the above specification is used as side information for predictor updates in a 1 bit/sample DPCM-APF system. If the total bit rate I of the system should be 8 kb/s, and further, if the speech bandwidth W is to be 0.45 f_s, where f_s is the sampling rate, determine the maximum permissible value of W in the above system.

(b) Consider next that the short-term spectral representation is used for bit allocation updates in a DFT-ATC system. Let the bit rate I be 8 kb/s as in (a), but let the values of bandwidth and sampling rate be constrained to be full-bandwidth designs W = 3.2kHz and f_s = 8kHz. Determine the maximum permissible value of R, the average number of bits available to quantize the DFT coefficients.

(12.33) Consider a 16 kb/s ATC coder. Assume that side information is transmitted in a total of 88 bits per 30 ms block of speech [Cox, 1980].

(a) Show that at $p_e = 10^{-3}$, the probability of one or more bit errors per side-information block is 0.0843. Assuming that any error can result in a lost block, show that one in every 12 blocks (or approximately 3 blocks per second) are effectively lost. Show also that if an error correcting code could correct 3 or fewer errors per block, the probability of losing a block is about 10^{-5}, or one lost block per hour. A coding theory bound says that 28 parity bits will suffice to correct up to 3 errors if there are 88 information bits. If a (116,28) code is employed for the side information, show that check bits account for 933 bps, and the bit rate available for coefficient quantization drops from 13.1 to about 12.1 kb/s.

(b) At $p_e = 10^{-2}$, show that about 3% of all side information will be lost with the (116,28) scheme in (a). If up to 7 errors can be corrected, side information will once again be well protected. From a coding theory bound, use of 61 parity bits will protect at least 7 errors. Show that the resulting (149,61) code for side information reduces the bit rate available for coefficient quantization to about 11.0 kb/s.

References

N. Ahmed and K. R. Rao, *Orthogonal Transforms for Digital Signal Processing*, Springer-Verlag, New York, 1975.

N. Ahmed, T. Natarajan and K. Rao, "Discrete Cosine Transform," IEEE Trans. on Computers, pp. 90-93, January 1974.

R. Bellman, *Introduction to Matrix Analysis*, McGraw-Hill, New York, 1960.

T. Berger, *Rate Distortion Theory*, Prentice-Hall, Englewood Cliffs, N.J., 1971.

S. J. Campanella and G. S. Robinson, "A Comparison of Orthogonal Transformations for Digital Speech Processing," IEEE Trans. on Communication Technology, pp. 1045-1049, December 1971.

W. Chen, C. H. Smith and S. Fralick, "A Fast Computational Algorithm for the Discrete Cosine Transform," IEEE Trans. on Communications, pp. 1004-1009, September 1977.

R. V. Cox, unpublished work, Bell Laboratories, 1980.

R. V. Cox, unpublished work, Bell Laboratories, 1981.

R. V. Cox and R. E. Crochiere, "Real-Time Simulation of Adaptive Transform Coding," IEEE Trans. on Acoustics, Speech and Signal Proc., pp. 147-154, April 1981.

R. V. Cox and A. G. Tescher, "Generalized Adaptive Transform Coding," Proc. Picture Coding Symposium, Asilomar, Calif., January 1976.

R. E. Crochiere, R. V. Cox and J. D. Johnston, "Real-Time Speech Coding," IEEE Trans. on Communications, pp. 621-634, April 1982.

W. R. Daumer, "Subjective Evaluation of Several Efficient Speech Coders," IEEE Trans. on Communications, pp. 655-662, April 1982.

L. D. Davisson, "Rate Distortion Theory and Applications," Proc. IEEE, pp. 800-808, July 1972.

D. F. Elliott and K. R. Rao, *Fast Transforms — Algorithms, Analyses and Applications*, Academic Press, New York, 1982.

R. C. Gonzalez and P. Wintz, *Digital Image Processing*, Addison-Wesley, Reading, Mass., 1977.

J. I. Guinlett, "Use of Activity Classes in Adaptive Transform Image Coding," IEEE Trans. on Communications, pp. 785-786, July 1975.

A. Habibi, "Comparison of nth Order DPCM Encoder with Linear Transformations and Block Quantization Techniques," IEEE Trans. on Communication Technology, pp. 948-956, February 1971.

A. Habibi, "Hybrid Coding of Pictorial Data," IEEE Trans. on Communications, pp. 614-624, May 1974.

A. Habibi, "Survey of Adaptive Image Coding Techniques," IEEE Trans. on Communications, pp. 1275-1284, November 1977.

A. Habibi, "An Adaptive Strategy for Hybrid Image Coding," IEEE Trans. on Communications, pp. 1736-1753, December 1981.

A. Habibi and G. S. Robinson, "A Survey of Digital Picture Coding," IEEE Computer, pp. 22-34, May 1974.

A. Habibi, A. S. Samulon and G. L. Fultz, "Adaptive Coding of MSS Imagery," Proc. Nat. Telecom. Conf., pp. 10.2.1-10.2.8, 1977.

H. Harmuth, *Transmission of Information by Orthogonal Functions*, Springer-Verlag, New York, 1972.

C. Heron, R. E. Crochiere and R. V. Cox, "A 32-band Sub-band/Transform Coder Incorporating Vector Quantization for the Transmission of Side Information," Proc. ICASSP, April 1983.

H. Hotelling, "Analysis of a Complex of Statistical Variables into Principal Components," J. Educational Psychology, pp. 417-441, 498-520, 1933.

Y. Huang and P. M. Schultheiss, "Block Quantization of Correlated Gaussian Random Variables," IEEE Trans. on Communication Systems, pp. 289-296, September 1963.

A. Jain, "A Fast Karhunen-Loeve Transform for a Class of Random Processes," IEEE Trans. on Communications, pp. 1023-1029, September 1976.

F. A. Kamanagar and K. R. Rao, "Interfield Hybrid Coding of Component Color Television Signals," IEEE Trans. on Communications, pp. 1740-1753, December 1981.

H. Karhunen, "Uber lineare Methoden in der Wahrscheinlichkeitsrechnung," Ann. Acad. Sci. Fenn., Ser. A.I.37, Helsinki, Finland, 1947.

K. Knowlton, "Progressive Transmission of Grey-Scale and Binary Pictures by Simple, Efficient and Lossless Encoding Schemes," Proc. IEEE, pp. 885-896, July 1980.

H. P. Kramer and M. V. Mathews, "A Linear Encoding for Transmitting as Set of Correlated Signals," IRE Trans. on Information Theory, pp. 41-46, September 1956.

H. J. Landau and D. Slepian, "Some Computer Experiments in Picture Processing for Bandwidth Reduction," Bell System Tech. J., pp. 1525-1540, May-June 1971.

W. Mauersberger, "Adaptive Transformationskodierung von digitalisierten Bildsignalen," Thesis, RWTH Aachen, 1980.

H. Murakami, Y. Hatori and H. Yamamoto, "Comparison between DPCM and Hadamard Transform Coding in the Composite Coding of the NTSC Color TV Signal," IEEE Trans. on Communications, pp. 469-479, March 1982.

M. J. Narasimha and A. M. Peterson, "On the Computation of the Discrete Cosine Transform," IEEE Trans. on Communications, pp. 934-936, June 1978.

T. R. Natarajan and N. Ahmed, "On Interframe Transform Coding," IEEE Trans. on Communications, pp. 1323-1329, November 1977.

A. N. Netravali and J. O. Limb, "Picture Coding: A Review," Proc. IEEE, pp. 366-406, March 1980.

K. Nitadori, "Linear transform coding and predictive coding" (in Japanese), Trans. *IECE Japan*, February 1970.

P. Noll, "Adaptive Transform Coding of Speech Signals," Proc. Int. Conf. on Communications, pp. 13.5-306—13.5-309, 1977.

J. B. O'Neal and T. R. Natarajan, "Coding Isotropic Images," IEEE Trans. on Information Theory, pp. 697-707, November 1977.

J. Pearl, "On Coding and Filtering Stationary Signals by Discrete Fourier Transforms," IEEE Trans. on Information Theory, pp. 229-232, March 1973.

A. Ploysongsang and K. R. Rao, "DCT/DPCM Processing of NTSC Composite Video Signal," IEEE Trans. on Communications, pp. 541-549, March 1982.

W. Pratt, *Digital Image Processing*, John Wiley, New York, 1978.

W. K. Pratt, J. Kane and H. C. Andrews, "Hadamard Transform Image Coding," Proc. IEEE, pp. 58-68, January 1969.

J. A. Roese, W. K. Pratt and G. S. Robinson, "Interframe Cosine Transform Image Coding," IEEE Trans. on Communications, pp. 1329-1339, November 1977.

A. Segall, "Bit Allocation and Encoding for Vector Sources," IEEE Trans. on Information Theory, pp. 162-169, March 1976.

F. Shum, A. Elliot and W. Brown, "Speech Processing with Walsh-Hadamard Transforms," IEEE Trans. on Audio and Electroacoustics, pp. 174-179, June 1973.

M. Tasto and P. A. Wintz, "Image Coding by Adaptive Block Quantization," IEEE Trans. on Communications, pp. 957-971, December 1971.

A. G. Tescher, "Transform Image Coding," in *Advances in Electron Physics*, Supplement 12, Academic Press, New York, 1979.

J. Tribolet and R. E. Crochiere, "Frequency Domain Coding of Speech," IEEE Trans. on Acoustics, Speech and Signal Proc., pp. 512-530, October 1979.

Z. Wang, "Reconsideration of a Fast Computational Algorithm for the Discrete Cosine Transform," IEEE Transaction on Communications, pp. 121-123, January 1983.

P. A. Wintz, "Transform Picture Coding," Proc. IEEE, pp. 809-820, July 1972.

W. C. Wong and R. Steele, "Partial Correction of Transmission Errors in Walsh Transform Image Coding Without Recourse to Error Correction Coding," Electronics Lett., pp. 298-300, May 1978.

R. Zelinski, "On Some Asymptotic Properties of Orthogonal Transforms for Diagonalizing of Toeplitz Matrices," (in German), Technical Report No. 190, Heinrich Hertz Institut, Berlin, 1976.

R. Zelinski, "Effects of Transmission Errors on the Mean-Squared Performance of Transform Coding Systems," IEEE Trans. on Acoustics, Speech and Signal Proc., pp. 121-133, April 1979.

R. Zelinski, "An Adaptive Transform Coding System Based on Cepstral Control and Entropy Coding," Frequenz, pp. 193-198, 1982.

R. Zelinski and P. Noll, "Adaptive Block Quantization of Speech Signals" (in German), Technical Report No. 181, Heinrich Hertz Institut, Berlin, 1975.

R. Zelinski and P. Noll, "Adaptive Transform Coding of Speech Signals," IEEE Trans. on Acoustics, Speech and Signal Proc., pp. 299-309, August 1977.

R. Zelinski and P. Noll, "Approaches to Adaptive Transform Speech Coding at Low Bit Rates," IEEE Trans. on Acoustics, Speech and Signal Processing, pp. 89-95, February 1979.

Computer Projects

(1) Generation and Analysis of an AR(1) Sequence

Consider the i.i.d. Gaussian process $\{Z(n)\}$ with mean zero and variance 10000.

(a) Using a Gaussian random number generator program, simulate and plot the random sequence

$$\{Z(n)\}\,; \quad n = 1,2,...,640$$

(b) Compute and plot the AR(1) sequence defined by

$$X(n) = 0.9X(n-1) + Z(n)$$

Compare with Fig. 2.16(a)(i).

(c) Compute and plot the acf $R_{xx}(k)$ for $k = 1,2,...,50$. Compare with Fig. 2.16(a)(ii).

(d) Use the approximation $R_{xx}(k) = 0$ for $k > 50$; use this to compute and plot the psd $S_{xx}(e^{j\omega})$. Compare with Figure 2.16(a)(iii).

(e) Compute the psd using (i) the periodogram (2.7) and (ii) the average of 5 periodograms, one for each of 5 nonoverlapping segments of $Z(n)$. Compare with Figure 2.16(a)(iii).

(f) Compute the spectral flatness measure using the results of (e). Compare with the theoretical value of $(1-0.9^2) = 0.19$ for the AR(1) process.

(2) Design and Analysis of a 2-bit Max-Quantizer

(a) Write an iterative program to design a mmse non-uniform quantizer for a Laplacian pdf input, with $R = 2$ bits/sample. Verify that the results are those of Table 4.3.

0313

(b) Using the decision and reconstruction values from the iterative design, evaluate the *SNR*, and verify that the value is the result of Table 4.4.

(3) Quantization with Entropy Coding

Simulate the uniform-threshold quantization of a gamma pdf input. Use large values of L (both odd and even) and optimum reconstruction values given by (4.45b).

(a) Plot the entropy H_Q versus step size Δ.

(b) Plot the normalized error variance σ_q^2/σ_x^2 versus step size Δ.

(c) Plot the entropy as a function of normalized error variance. Compare with Figure 4.19(b).

(d) Repeat (a) (b) and (c) for a Laplacian input.

(4) Statistical Properties of Quantization Error

Quantize the samples of the AR(1) sequence in Project (1)(b) using mmse Max-quantizers with number of levels $L = 2$, 4 and 32. ($R = 1$, 2 and 5 bits per sample).

(a) Plot the reconstruction error sequences $r(n) = q(n)$ scaled by a factor 2^R, as in Figure 4.30.

(b) Compute, in each case, the error variances σ_r^2, σ_{rg}^2 and σ_{ro}^2, where g and o stand for granular noise and overload distortion. Also, compute the probabilities of overload.

(c) Compute the pdf of reconstruction error in each case. Compare with Figure 4.22.

(d) Compute and plot the normalized acf $\rho_{rr}(n)$ for the case of $L = 2$. Compare with the result of Figure 4.24(b). [Note that $\rho_{xx}(1) = 0.95$ in that example].

(e) Determine the values of the normalized crosscorrelation $\rho_{rx}(0)$ for $L = 2$ and 4. Compare with the values for $L = 2$ and $\rho_{xx}(1) = 0.9$ in Figure 4.24(c).

(f) Repeat (a) (d) and (e) for the case of $L = 2$ and a dithered quantization system. Use a dither signal with a uniform pdf in the range $(-\Delta/2, \Delta/2)$, and assume that the dither signal is eventually subtracted at the receiver.

(g) Repeat (a) (d) and (e) for the case of $L = 2$ and an oscillating two-level dither sequence $\{-3\Delta/8, \Delta/8, -\Delta/8, 3\Delta/8, \ldots\}$.

(5) Comparison of 2-bit Quantizers

(a) Compute and plot a locally stationary Gauss-Markov sequence of length 640 samples, with $\rho_1 = 0.85$ and σ_x values of 1, 10, 100, 50 and 10 for five continuous segments of 128 samples each.

(b) Consider a 2 bit/sample Lloyd-Max nonuniform quantizer based on the long-time (640 sample-) averaged input variance, and the quantizer design of Table 4.3. Compute and plot the quantized output waveform, and the reconstruction error waveform magnified by a factor of 5. Determine the *SNR*, the five values of local *SNR* (one for each segment of 128 samples), and the overall segmental *SNR*, *SNRSEG*.

(c) Repeat (b) for a coding system based on μ-law companding ($\mu = 255$; $x_{max} = 200$) and a 2 bit/sample uniform quantizer designed by assuming a uniform pdf in $(-x_{max}, x_{max})$.

0314

(d) Repeat (b) for an adaptive quantizer with a one-word memory, with $M_1 = 0.8$, $M_2 = 1.6$, and initial step size of 1.

(e) Repeat (b) for a forward-adaptive quantization system with a block length of 128 samples and an mmse step-size given by $0.996\,\sigma_x(m)$; $m = 1,2,\ldots,5$.

(6) Effects of Transmission Errors on 4-bit Quantizers

(a) Simulate an AR(1) sequence, as in Project 1(b), but for $n = 1, 2, \ldots, 64000$.

(b) Quantize the input using a 4 bit/sample AQB system with the multipliers of Table 4.14, a uniform quantization characteristic, an initial step-size of 1, and minimum and maximum step sizes of 1 and 512. Compute the values of *SNR* and *SNRSEG*, and plot the output $y(n)$.

(c) Repeat (b) for bit error rates of 10^{-4}, 10^{-3}, 10^{-2} and 10^{-1}, assuming an FBC code in each case. Compare with the characteristic of Figure 4.33(a) for $R = 2$.

(d) Repeat (b) and (c) using the robust adaptive quantizer of (4.199a) with $\beta = 31/32$.

(7) Toll Quality PCM Coding

(a) Simulate the digitally linearizable 8 bit/sample A87.56 PCM coding system of Table 5.3 for the 640-sample Gauss-Markov input of Project 5(a). Compute the *SNR*, the five values of local *SNR* (one for each segment of 128 samples), and the overall segmental *SNR*, *SNRSEG*. Plot the output and reconstruction error magnified by a factor of 100, as in Figure 1.4.

(b) Repeat (a) with a bit error rate of 10^{-2}.

(c) Enhance the output by (b) by using the following procedure for error detection followed by interpolation. For error detection, declare the sample at time n to be in error if the difference $y(n) - y_c(n-1)$ exceeds $\pm\lambda\sigma_{y'}$ where $y_c(n-1)$ is the correct (or corrected) previous sample, and $\sigma_{y'}$ is the standard deviation of the difference signal in the error-free condition (a). Use the first-order interpolation algorithm $y_c(n) = 0.5\,y(n-1) + 0.5\,y(n+1)$ to correct a sample suspected to be in error. Determine the *SNR*-maximizing value of λ, and compare the corresponding output, *SNR* and *SNRSEG* with the results of (a) and (b).

(8) Analysis of Linear Prediction

Consider the acf data in the last column of Table 2.2.

(a) Compute optimum prediction- and parcor- coefficients $\mathbf{h}$ and $\boldsymbol{\kappa}$ for mmse linear prediction of order $N = 1, 2, \ldots, 8$.

(b) Compute the corresponding values of prediction gain ${}^{N}G_P$.

(c) Use the predictors of (a) to compute a set of autoregressive estimates of the psd that correspond to the acf data of Table 2.2. Compare with the long-time-averaged speech spectra in Figure 2.9(a).

(9) Comparison of 2-bit DPCM Techniques

(a) Compute and plot a locally stationary Gauss-Markov AR(1) sequence of length 640 samples, with $\sigma_x = 100$ and ρ_1 values of 0.9, 0.95, 0.85, 0 and -0.5 for five continuous segments of 128 samples each.

(b) Consider a 2 bit/sample adaptive quantizer with a one-word memory with $M_1 = 0.8$, $M_2 = 1.6$, a uniform quantization characteristic, an initial step-size of 10, and minimum and maximum step sizes of 1 and 512. Compute and plot the quantized 2-bit PCM output $y(n)$ and the reconstruction error waveform magnified by a factor of 5. Determine the *SNR*, the five values of local *SNR* (one for each segment of 128 samples), and the overall segmental *SNR*, *SNRSEG*.

(c) Repeat (b) for 2-bit DPCM-AQB with $h_1 = 0.85$.

(d) Repeat (b) for 2-bit DPCM-AQB-APF assuming that h_1 is known and set equal to the appropriate ρ_1.

(e) Repeat (b) for 2-bit DPCM-AQB-APB based on the polarity-correlation gradient search

$$h_1(n+1) = h_1(n) + c\ sgn\, d(n)\ sgn\, y(n-1)$$

and a value of c that maximizes *SNRSEG*.

(f) Repeat for a 4 bit/sample DPCM-AQB-APB system. Use the AQB system of Table 4.14 and the APB system of Section 6.5.3.

(10) Comparison of Intraframe Predictors

(a) Generate a 32×32 2-D field based on an autoregressive model and a separable acf with $\rho_v = \rho_h = 0.95$, and a unit-variance innovations process (2.138).

(b) Simulate a 3 bit/sample DPCM system based on first order prediction, using the previous horizontal neighbor and prediction coefficient 0.95. Compute the *SNR*.

(c) Repeat with first-order prediction based on the previous vertical neighbor.

(d) Repeat with 2-D prediction of order three, with coefficients 0.95, 0.95 and -0.95^2 (Example 6.5; case of $J = 2$) for horizontal, vertical and diagonal neighbors. Compare the *SNR* (dB) gain over (b) or (c) with the factor of two improvement expected in the separable acf model.

(11) Noise Feedback Coding

Consider the AR(1) input of Project 1(b).

(a) Simulate a 3 bit/sample DPCM-AQF-APF coder with $\Delta = 0.996\ \sigma_x = 99.6$ (which is the mmse step size, ignoring quantization error feedback), and first-order prediction with $h_1 = 0.9$. Compute the *SNR* and plot the psd $S_{rr}(e^{j\omega})$ of reconstruction noise.

(b) Repeat (a) for D*PCM with an mmse design of h_1 (7.18).

(c) Repeat (a) for the one-tap NFC coder of Figure 7.6.

(12) Deltamodulation with F=2

Consider the AR(1) input of Project 1(b).

(a) Use a lowpass filter program to bandlimit the sequence to a frequency slightly less than one-half of the sampling rate. Oversample the output of the filter by a factor 2.

(b) Determine *SNRF*-maximizing step-size in LDM coding, assuming that the LDM staircase output is filtered by the lowpass filter in (a).

0316

(c) Plot $y(n)$, $y_F(n)$, $r(n)$ and $r_F(n)$ using the best design in (b), as in Figure 8.3.

(d) Compute the values of *SNR* and *SNRF*. Compare the value of filtering gain G_F with the theoretical estimate of 10 log F = 3 dB.

(e) Repeat (c) for a bit error rate of 10^{-2}.

(f) Repeat (c) and (d) for ADM with a one-bit memory, with multiplier $M_1 = M_2^{-1} = 1.2$, an initial step size of 10, and minimum and maximum step sizes of 1 and 512.

(13) Generalized Lloyd-Max Vector Quantizer

(a) Write an iterative, generalized Lloyd-Max program to design a vector quantizer for a Gaussian pdf input, with $N = 8$ and $R = 1$ bit/sample. Compute the *SNR*. Verify that the result is that of Table 9.1.

(b) Repeat for $N = 2$ and $R = 1,2,...$ for a uniform pdf in (0,1). Verify that the partitioning of the 2-D space by the VQ algorithm evolves toward a hexagonal lattice for large values of R.

(14) Sub-band Coding (M=4; R=2 bits/sample)

Simulate a four-band, 2 bit/sample SBC system for the input of Project 5(a), using a QMF tree with 32-tap and 16-tap filter components of Table 11.1 for the first and second stages of the tree. Use the AQB system of Project 5(d) for the quantization of the sub-band signals.

(a) Compute the *SNR* for each band and the overall *SNR*.

(b) Plot the input $x(n)$, and the output $y(n)$, as well as the sub-band outputs $y_k(n)$; $k = 1$ to 4.

(c) Compute the variances and normalized adjacent sample correlations in individual sub-band signals $x_k(n)$.

(d) Use the variances in (c) to obtain a mmse bit allocation. Use this to repeat (a) and (b). Compare the *SNR* gain over the fixed bit allocation system with the theoretical maximum gain of 0.19^{-1} mentioned in Project 1(f).

(15) Eigenvectors and Eigenvalues

Consider a zero-mean AR(1) process with $\rho = 0.95$ and variance $\sigma_x^2 = 1$. Use an eigenvector computation program to compute eigenvectors and eigenvalues for $N = 2$ and 8 (See (12.102) and Table 12.1). Note that the sum of eigenvalues is equal to $N\sigma_x^2 = N$ in each case.

(16) Transform Coding (R=2 bits/sample)

Simulate a 2 bit/sample TC system for the input of Project 1(b), using the DCT, uniform quantizers and values of block length $N = 4$, 32 and 128.

(a) Compute the *SNR* and *SNRSEG*, and plot the output $y(n)$ for each value of N.

(b) Compute the variances of the transform coefficients and determine the mmse bit allocation for each value of N.

(c) Repeat (a) for a zonal coding system that retains the first $N/2$ coefficients and quantizes them using 4 bits/sample.

(d) Repeat (a) for an ATC system using the mmse bit allocation obtained in (b).

Compare the gain in *SNR* with the theoretical value obtained from the variance distribution in (b).

(17) Speech input

Repeat relevant parts of Projects 1 through 16 for the speech input of Table CP.1. Compare the 16 kb/s speech outputs in Projects 2, 5, 9 and 12 with those resulting from simulations of the toll-quality systems in Projects 7 and 9(f). [In the projects involving adaptive quantization, maintain the ratio of 512 between maximum and minimum step size, but change the values of initial, minimum and maximum step sizes as needed, to provide a good match to the amplitude levels in Table CP.1].

(18) Image input

Repeat relevant parts of Projects 10, 13(a) and 16 for the image input of Table CP.2. Assume a (8×8) 2D approach in the ATC and VQ projects.

Table CP.1 The speech sequence of Figure 1.4(a). The eight columns represent contiguous segments of 4 ms duration. The total duration is 32 ms (256 samples at 8 kHz).

−120	−166	144	−1381	−3150	418	1860	−3690
186	1087	363	−899	3013	−603	2621	−3565
−348	−497	−341	−1012	−2460	−2202	2790	−2645
−517	−991	−203	−637	−2001	−3100	3551	−1439
555	1034	373	−73	−1129	−3740	3473	−79
−5	606	106	−1	−5	−4572	1997	1097
−434	−1205	−98	594	1023	−5140	837	1682
17	523	86	1112	1795	−4530	380	2373
595	644	227	867	2596	−3996	−1169	3352
−225	−706	54	1369	3491	−3005	−3620	3431
−473	205	88	1585	3482	−1731	−4224	3091
189	320	37	968	3114	−308	−5309	3414
237	−642	−136	1164	2508	1260	−6060	2477
−3	512	337	1420	2256	2898	−5987	450
−188	294	359	709	1581	4169	−5841	−891
117	−892	−281	403	231	5079	−4288	−1903
−249	3	−4	402	−612	5698	−3035	−3887
12	688	772	181	−666	5376	−1301	−5615
617	−525	317	105	−1222	4730	703	−5664
−932	−298	−34	220	−1628	3589	2439	−6152
−92	412	427	540	−1490	2403	4205	−6432
970	−9	610	882	−980	760	5564	−5684
−497	−331	296	693	−747	−658	6520	−4636
−592	9	318	538	−402	−1734	6693	−2713
873	−79	90	503	284	−2531	5896	−569
−175	11	−192	304	827	−3103	4825	1284
−314	80	−17	−296	1530	−3249	3262	3138
178	−505	−583	−1051	2143	−2651	1413	4577
−249	−200	−1108	−1640	2147	−1758	−429	5609
149	449	−863	−2066	2117	−870	−1887	6757
451	−196	−946	−2706	1361	−17	−2974	7069
−884	−538	−1505	−3267	1078	659	−3409	6248

Table CP.2 The 32×32 image sequence of Figure CP.1(b). The 32 rows represent scan lines with 32 pixels per scan line. Intensity levels of 255 and 0 represent pure white and pure black [R. Baker, 1983].

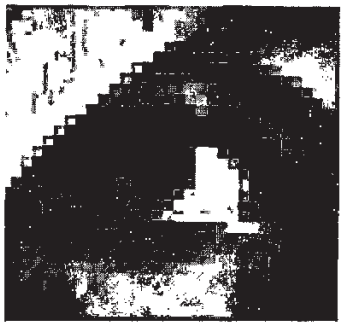
118	113	85	101	161	151	151	163	177	174	165	167	131	124	147	158	181	189	166	148	144	147	145	123	128	129	138	143	148	149	148	156
118	85	101	151	140	144	157	162	178	173	176	169	134	108	159	173	183	170	154	140	138	141	125	129	134	134	144	147	151	152	154	154
89	90	141	146	140	138	145	166	178	172	176	172	149	107	159	180	172	153	141	133	127	128	128	131	135	139	146	149	147	153	153	158
84	141	158	134	138	138	143	155	176	177	173	174	163	134	167	169	148	148	135	126	134	140	134	133	139	140	147	150	150	152	154	160
129	161	140	144	147	137	140	156	179	181	172	177	167	162	170	160	148	132	126	134	138	140	138	139	141	143	148	157	153	152	156	155
171	145	152	160	153	141	138	157	177	179	177	178	170	169	164	124	84	97	113	120	127	135	139	141	144	149	153	153	156	158	156	155
154	147	162	170	151	133	129	155	179	183	185	180	179	159	80	42	58	74	81	95	104	106	114	128	141	147	151	157	160	157	153	154
150	161	174	180	143	133	138	168	179	186	186	185	144	72	67	75	75	69	64	66	54	63	75	86	109	130	145	155	162	156	154	154
163	170	186	174	135	133	149	174	180	186	183	131	91	101	116	114	114	105	93	81	61	34	36	36	53	70	106	130	150	152	153	157
168	183	188	155	129	144	166	182	188	179	113	84	102	115	123	117	113	130	127	109	88	74	54	45	36	32	41	67	112	138	145	145
178	190	179	141	135	159	181	189	173	94	82	93	93	107	105	106	72	118	131	117	90	97	63	63	56	50	47	51	71	104	138	138
186	190	167	147	150	174	192	158	74	66	74	70	73	79	80	75	26	67	116	80	61	66	68	88	68	76	65	69	68	91	116	126
129	186	170	157	169	190	155	55	54	55	51	59	37	34	35	31	16	30	47	24	20	25	52	87	65	92	85	81	82	89	102	116
190	178	174	170	189	133	43	33	38	32	22	27	23	22	20	19	21	25	26	25	16	11	14	28	85	106	92	91	89	93	104	111
186	179	179	182	141	42	32	31	19	20	21	23	25	19	15	17	16	38	67	57	28	18	14	13	27	75	102	97	90	92	100	107
187	179	188	145	42	41	33	23	18	21	20	20	19	18	16	25	28	70	151	150	88	36	13	12	14	42	88	99	89	90	99	101
187	192	154	55	52	43	26	18	12	27	50	30	35	28	9	30	48	57	174	188	168	127	43	25	59	41	79	88	82	86	96	101
192	172	70	65	54	34	22	19	15	35	73	49	47	63	27	41	67	40	174	191	189	176	110	38	76	65	64	92	76	84	90	100
177	90	71	83	73	56	50	27	14	38	72	89	39	54	59	55	31	98	199	199	197	193	148	65	54	83	58	72	74	80	84	94
92	78	84	95	93	89	81	60	17	25	69	87	83	47	31	43	95	177	193	193	189	178	136	95	66	87	75	75	86	78	84	94
82	86	82	87	105	106	99	84	54	50	64	81	101	97	90	117	167	184	191	196	191	159	109	115	92	78	95	81	78	85	91	91
88	90	96	91	107	114	110	104	94	72	67	69	83	100	116	128	139	156	164	164	158	158	133	113	95	95	95	87	81	81	90	93
95	94	100	99	106	110	111	112	107	92	99	91	71	82	62	91	103	102	110	100	126	145	151	155	141	110	87	87	86	90	88	93
98	101	110	104	110	112	112	110	113	113	113	99	84	90	70	97	106	99	89	92	124	120	130	131	118	103	86	84	92	96	89	89
98	103	114	111	115	116	120	113	108	112	119	107	107	104	95	99	106	107	106	120	126	129	124	118	122	119	89	87	97	99	92	90
102	111	117	122	122	119	122	123	124	121	117	110	114	111	108	111	113	122	125	131	136	137	133	135	137	111	95	95	100	100	92	87
98	108	118	122	124	125	132	133	132	132	129	121	124	129	124	131	135	139	142	145	140	144	134	141	132	107	93	99	106	99	88	90
102	105	117	119	126	127	135	142	135	139	148	148	140	139	141	140	150	151	148	145	140	147	139	141	124	106	102	111	107	99	93	91
104	103	112	120	125	127	133	145	135	141	148	153	148	143	142	144	148	148	145	143	148	147	142	131	121	109	104	110	102	96	89	92
97	105	111	121	120	127	132	142	145	144	150	153	149	156	153	151	148	144	145	153	153	151	140	129	120	114	107	110	107	99	91	96
100	105	112	119	119	126	132	141	147	148	149	158	151	145	156	155	157	155	151	152	154	149	143	133	120	115	118	112	108	100	95	95
95	102	112	111	121	126	131	140	143	148	150	157	156	154	154	155	153	154	155	156	150	142	135	128	126	117	113	111	108	103	94	94

Figure CP.1 (a) the "woman with hat" image; and (b) a subimage which includes the woman's left eye, eyebrow and a nice diagonal edge formed by a portion of the brim of her hat [R. Baker, 1983].

(a)



(b)



Index

A-B preference test, 659
A-to-D conversion, 86, 115, 222, 666
Adaptation time constant, 197, 414
Adaptive bit allocation, 290, 490, 565
Adaptive deltamodulation (ADM), 376, 400, 501
Adaptive DPCM (ADPCM), 308
Adaptive postfiltering, 340, 400
Adaptive prediction (AP), 31, 252, 266, 274, 290, 292
Adaptive predictive coding (APC), 290, 315–17, 354, 368, 456, 565
Adaptive quantization (AQ), 16, 188
  backward estimation (AQB), 191, 278, 280, 295, 401, 564
  forward estimation (AQF), 191, 227, 401, 415, 564, 580
  instantaneous adaptation, 197
  one-word memory, 199–200
  robust algorithm, 207, 216
  static and dynamic performance, 191, 198, 202
  syllabic adaptation, 197
Adaptive transform coding (ATC), 563
Adaptive vector quantization, 461
Additive white Gaussian noise channel, 630
Adjacent sample correlation, 36, 71–73, 83
A-law companding, 142–45, 668
Aliasing, 102–9, 496
All-pass characteristic, 381, 498
All-pole filter, 604
All-pole predictor, 254, 259, 275, 293, 325, 434, 457
All-pole process, 62, 275
All-zero filter, 604
All-zero predictor, 254, 306, 325, 434–35
All-zero process, 62
Amplitude quantization, 16, 17, 115–220
Amplitude spectrum, 600
Analog channel, 3, 19
Analog-to-digital conversion, 86, 115, 222, 666
Analog multiplexing, 223
Analysis-by-synthesis coding, 10, 460
Aperture coding, 480
AR(1) process, 62–64, 262, 272, 510, 626
  delayed decision coding, 452
  distortion rate bounds, 645
  DM noise model, 381
  DPCM input, 334, 339
  D*PCM input, 361
  DPCM noise model, 279
  DPCM predictor, 278
  NFC input, 368

quantizer input, 162
transform coding, 541, 555
two-dimensional, 73
two-level inputs, 469, 473
AR(2) process, 66, 397, 523
AR(8) process, 520
AR(10) process, 542, 551
Articulation index, 107, 489, 654, 663
Asynchronous deltamodulation, 479
Audio, 8, 9, 247, 248
Autocorrelation function (acf), 27
of AR(1) process, 64
of bandlimited white spectrum, 78
in continuous-time, 74
effect of linear system, 52
of integrated power spectrum, 75, 384
interframe acf of images, 39
intraframe acf of images, 37, 38, 83
long-time-averaged acf of speech, 36, 37
of MA(1) process, 70
with oversampling, 384
of raised cosine spectrum, 78
of second-order Butterworth spectrum, 77
of symmetrical psd, 114
of 2-D processes, 71, 73
variance-normalized, 35, 50
Autocorrelation matrix, 50, 51
AR(1) process, 65
determinant, 270
diagonalization, 59
inverse, 82
MA(1) process, 70
Autocorrelation method for pitch, 318
Autocorrelation solution in prediction, 297
Autocovariance function, 49
Autoregressive process, 54
continuous-time, 76
first order, 63, 262
image model, 73, 523
second order, 66
speech model, 67, 397, 520, 542, 551
spectrum model, 276
Average magnitude difference function (AMDF), 319

Backward adaptive prediction (APB), 292, 303, 307
Backward adaptive quantization (AQB), 191, 278, 280, 295, 401, 564
Bandlimited signal, 26, 87
Bandpass filtered speech, 36, 271, 294
Bandpass signal, 26, 96–98
Bandwidth, 8, 9, 77
Basis images, 521, 522, 549
Basis sequences, 513
Basis vectors, 518
Binary codes, 177, 178, 187, 215, 222, 232
Binary digit, 2, 4, 19
Binary entropy function, 472, 618
Binary symmetric channel, 18, 175, 616
Binary tree search, 303, 440–41
Bit allocation, 322 (*also see* Optimum- and Variable bit allocation)
Bit error rate, 17, 629
effect on DM, 246, 418–20
effect on DPCM, 246, 308, 311, 326, 327, 332, 333, 335, 338
effect on D*PCM, 335
effect on PCM, 244, 245, 246, 247
effect on quantization, 176, 185, 209
effect on SBC, 506
effect on TC, 578, 580
Bit rate, 2, 9
audio, 13, 228
and coder complexity, 13
digital hierarchies, 14, 228, 283, 308
image coding, 13, 227, 283, 287, 288, 576
in quantization, 116
speech coding, 11, 13, 283, 308, 369, 502, 574
sub-band coding, 490, 496, 504
videophone, 13, 324
Bit stealing, 229, 246, 248, 289, 506
Block companding, 192
Block mode in coding, 431
Block quantization, 511
Broadcast quality, 12, 287, 660
Broadcast television, 227

CCIR image quality standards, 660
CCITT standards:
DPCM speech coding, 308, 311
graphics, 466, 475–77
image noise weighting, 653
PCM speech coding, 228, 241
transmission errors, 244
Channel capacity, 616, 628
Channel coding, 20
Channel coefficient, 183–86, 214, 243, 244, 329, 576
Channel decoder, 4
Channel encoder, 4

0321

Channel error variance, 179, 180, 182, 184, 185
in DM, 417
in DPCM, 329, 330
in TC, 577, 579
Channel matrix, 179, 627
Characteristic equation, 53–54, 538, 612
Circuit switching, 15
Circular acf matrix, 555
Clipping, 225, 318
Clustering technique, 438
C-message weighting, 652
Codebook coding, 428, 432, 436, 437, 453, 458
Code conversion, 235–39, 400, 671
Coder complexity, 2, 9
and bit rate, 13
in delayed decision coding, 431, 437
in DPCM, 252, 307, 308
and encoding delay, 13, 241
and $R(D)$ function, 22
and speech quality, 11, 12
Coder memory, 15, 16, 440
Code tree, 432, 433, 449, 622
Color video, 9, 227, 287, 576
Comma free code, 622
Communications quality, 10, 283, 317, 373, 500
Companding, 130
A-law, 142, 145
digitally linearizable, 229, 234
effect on pdf, 144
gain, 131, 230
logarithmic, 141, 142
μ-law, 145, 146
m-law, 138, 139
nearly instanteous, 229
piecewise linear, 229, 234
Complementary code tree, 449
Complex transform, 523
Component color coding, 9, 227
Composite color coding, 287
Compression function, 130, 138, 153
Computational complexity, 440–42
Conditional entropy, 619
Conditional replenishment, 322
Constant factor deltamodulation (CFDM), 404–7
Continuous amplitude channel, 630
Continuous amplitude source, 2, 22, 23, 616, 624, 625
Continuous deltamodulation, 410
Continuously variable slope deltamodulation (CVSD), 410–15
Continuous time system, 606
Contouring, 175, 227, 285
Corner frequency, 77
Counting-type A/D, 669
Covariance solution in prediction, 296, 297
Cramer's rule, 270
Crosscorrelation function, 52
Cutoff frequency, 77, 109

D/A conversion, 86, 116, 222, 666, 672
DDC (*see* Delayed decision coding)
Decimation, 495
Decision thresholds, 116, 134
Delayed decision coding (DDC), 428–64
APC, 456
DPCM, 452
Deltamodulation (DM), 16, 295, 372–472
A/D conversion, 376, 399, 671
delayed decision, 429, 433, 434, 456
double integration, 461
oversampling, 77
sigma ($\sum$DM), 399, 671
slope-anticipating, 429
Delta sequence modulation, 440, 454
Demodulator, 4
Deterministic population in DDC, 429, 437, 452, 457
Diagnostic Rhyme Test, 317, 663
Differential entropy, 45, 617, 624
Differential PCM (DPCM), 16, 252–350, 543, 555
closed-loop, 254
delayed decision, 434, 453–56, 461
image coding, 271, 272, 283, 320, 321, 550
speech coding, 202, 210, 271, 281, 291, 293, 295, 307, 313
Digital audio, 227, 247, 504, 507
Digital channel, 3–5, 18, 21, 22
Digital coding, 1–25
advantages, 1
analysis-synthesis, 10
basic processes, 2
information-lossy, 2, 16

0322

information-preserving, 2, 16
of speech and images, 11, 13
theoretical limits, 20
waveform-preserving, 9
Digital communication, 3, 4
network hierarchies, 14, 15
Digital hierarchy, 14
Digitally controlled deltamodulation (DCDM), 410
Digital speech interpolation (DSI), 12, 289
Digital-to-analog conversion, 86, 116, 222, 666, 672
Digit-at-a-time A/D, 667
Direct feedback coder, 363
Discrete-amplitude source, 22, 45, 148, 616, 617, 619
Discrete cosine transform (DCT), 520, 542, 555, 558, 572, 574
Discrete Fourier transform (DFT), 59, 520, 542, 551, 554, 555, 557, 572, 599
Discrete Hadamard transform (DHT), 547
Discrete memoryless channel, 616, 626
Discrete Walsh Hadamard transform (DWHT), 520, 522, 542, 547, 550, 551, 572
Distant-sample based prediction, 252, 256, 312–24
Distortion, 1, 632–49
Distortion rate bounds, 445, 446, 452, 454, 538, 632–49
Distortion rate function (DRF), 21, 22, 24, 58, 632–49
Dithering, 164–75
effect on reconstruction error, 165, 172
subjective effect in image coding, 173
subjective effect in speech coding, 171
D(*R*), DRF (*see* Distortion rate bounds and function)
Dither matrix, 175, 213
DM (*see* Deltamodulation)
Dot product, 609
Double-integration deltamodulation, 393–98, 455
DPCM (*see* Differential PCM)
D*PCM, 334, 352, 356, 369
Dual-mode coding, 288
Dyadic acf matrix, 550
Dynamic locking quantizer, 205, 308–9
Dynamic range:
of coefficient variance, 551
of quantizer, 188, 201, 255, 529
of quantizer input, 188, 529
of speech, 34
of step size, 201, 232–34, 309, 405, 407, 409, 412, 416

Edge-adaptive coding, 400, 403
Edge-adaptive DPCM, 286
Edge busyness, 258, 285
Edge rendition, 266, 283, 408, 430
Eigenfrequencies, 54, 65
Eigenimages, 523
Eigenvalues, 51, 54, 59, 536, 614, 646
of AR(1) process, 65, 541
of MA(1) process, 70
Eigenvectors, 53, 54, 520, 536, 541, 614
of AR(1) process, 65
of MA(1) process, 70
Embedded coding, 289, 546
Empty cell problem in VQ, 438
Encoding delay, 13, 622
in APF, 293
in AQF, 192
and complexity, 13
in delayed decision, 428, 433
in interframe coding, 320
in transform coding, 511
and waveform properties, 15
End effects in TC, 560
Energy spectrum, 600
Entropy, 469
Entropy coding, 146–58, 621
of APC-NF output, 369
of DPCM output, 287, 288, 295
gamma pdf input, 152, 156, 157
Gaussian pdf input, 152
high average bit rates, 153
low average bit rates, 156
of quantizer output, 151
Entropy power, 45, 276, 624, 625
Equal-width sub-bands, 488, 489
Equivalent *SNR*, 404, 659
Equivocation, 628
Ergodicity, 31
Error detection, 247, 248, 327, 337, 338, 475
Error function, 45
Error protection, 18, 19
in DPCM, 337, 338
with explicit redundancy, 246, 337, 338

Error protection, (*cont.*)
with natural redundancy, 247, 337
in PCM, 246, 247
of quantizer output, 185, 187, 188, 214
in SBC, 506, 508
in TC, 578–80
Error streaks in images, 327, 419
Estimators of input variance, 196, 198
Exchange of spatial and temporal resolution, 322
Exhaustive search, 430, 441
Explicit coding of reconstruction noise, 290, 398
Exponential adaptation, 402–6

Facsimile, 2, 9, 322, 465–85
Fast Fourier transform (FFT), 554, 559
Filtered reconstruction error, 375
Filtering gain, 379–81
Fine quantization, 257, 278, 279
Finite impulse response (FIR), 498, 604
Finite state vector quantizer (FSVQ), 439
First-order interpolation, 247, 248, 265
First-order Markov source, 63, 620 (*also see* AR(1) process)
First-order prediction:
in delayed decision, 457
in DM, 373, 417
in DPCM and D*PCM, 261, 262, 278, 283, 316, 327, 330, 335, 358, 359
in SBC, 503
Five-point quality scale, 660
Fixed bit allocation, 487
Fixed predictor, 252, 271, 272, 276, 287, 293, 295
Folded binary code (FBC), 177, 187, 215
Forward-adaptive prediction (APF), 292, 295, 307
Forward transformation kernel, 518
Fourier transform, 598
Four-sigma loading, 45, 125
Fourth-order image prediction, 264, 477
Fractional bit rates, 429
Frame period, 15, 30, 39, 101, 320
Frequency division multiplex (FDM), 113
Frequency domain coding, 16, 58, 486, 511
Frequency-ordered Fourier matrix, 553
Frequency response, 603, 605
Frequency weighted reconstruction error:
ATC coding of speech, 569–70
in delayed decision, 460
in NFC, 351, 355, 366–68
rate distortion theory, 642–47, 652
in transform coding, 532
Frequency weighted *SNR*, 652
Frequency weighting function, 363

Gain over PCM:
in DM, 295, 379–81, 386, 411
in DPCM, 253, 259, 271, 272, 282, 294, 295, 335, 386
in SBC, 491, 494
in TC, 511, 527, 543, 572
Gamma source, 34, 154, 156, 157, 182, 445, 446, 450, 454, 640
Gaussian source, 24, 34, 151, 154,162, 182, 445, 447, 450, 452, 454, 640
Generalized Lloyd-Max quantizer, 438
Geometric interpretation of TC, 515
Geometric interpretation of VQ, 438
Geometric pdf of run length, 470
Granular noise, 121
in DM, 377, 390
in DPCM, 258, 282, 283, 285
in RLC, 480
Graphical interpretation of bit allocation, 531
Graphics, 2, 9, 322, 465–85
Gray code, 177, 215, 671
Gray level image, 30

Half-whitening, 357, 358, 360
Hamming distance, 179, 214, 635
High complexity coder, 11, 13, 317, 324, 369, 487, 504, 574
High detail image, 30, 37, 38, 41
High information deltamodulation (HIDM), 403
Highpass filter, 497, 500
Highpass signal, 40, 41, 64, 70, 273, 277, 381
High quality coding, 13, 376, 429, 511, 576 (*also see* Broadcasting quality; Toll quality
Hotelling transform, 536
Huffman coding, 148, 622
Hybrid coding:
APC/SBC, 505
DPCM/PCM, 328, 336
intraframe/interframe, 322
TC/DPCM, 561–63, 576
Hybrid companding, 400, 412, 414

Identity transform, 547
Idle channel noise, 233
  in DM, 391, 392
  in PCM, 242, 249
Image activity, 323
Image classification, 461
Image coding, 1, 12
  delayed decision, 460, 461
  DM, 403, 407
  DPCM, 283, 320
  high-quality, 12, 13, 287, 376, 429, 511, 576, 660
  PCM, 226, 227
  TC, 563, 574, 575
Image enhancement, 327, 338
Image field, 101, 102
Image frame, 102
Image impairment, 324, 660
Image sampling, 9
Imaging error in SBC, 497
Impairment scale, 660
Impulse response, 52, 602, 605
  AR(1) process, 63
  DPCM decoder, 330, 332–33
  MA(1) process, 69
  NFC decoder, 356
In-band error, 375
Incremental mode in coding, 431
Independent, identically distributed (i.i.d.) inputs, 629, 635
Infinite impulse response (IIR), 499, 604
Information theoretical limits, 20–21, 616–49
Innovations, 58, 60, 287, 433, 440,451
Innovations tree, trellis, 439
Instantaneous adaptation, 197
  in DM, 387, 400, 401, 409, 415
  in DPCM, 282
Integer-band sampling, 98, 494
Integrated power spectrum, 74
  bandlimited, 76
  in deltamodulation, 384, 385, 388, 418
  prefiltering error with, 107
Integrated services digital network (ISDN), 15, 228, 468
Intelligibility, 10, 659, 662
  in APC, 317
  in DM, 419–20
  in PCM, 225
Interband leakage, 497
Interfield predictor, 287
Interframe coding, 16
Interframe predictor, 252, 262, 287, 291, 320, 561, 563, 574
Interlacing of image fields, 101
Interpolation, 86
  cubic spline, 113
  image, 419
  si function, 91, 93
  staircase, 91, 94, 112
  sub-band coding, 495
Interpolative A/D converter, 669
Intraframe coding, 16
Intraframe predictor, 252, 262, 265, 271–72, 284, 287, 321
Inverse DFT (IDFT), 599
Inverse Fourier transform, 599
Inverse matrix, 612
Inverse transformation kernel, 518
Irrelevancy, 16, 17, 322
Isopreference, 661
Isotropic acf model, 71, 264–66
Iterative population in DDC, 429, 437, 446, 447, 458

J, K, L algorithm, 408

Kahrunen Loeve transform (KLT), 58, 520, 522, 535–46, 572, 646
Kalman filtering, 307

Laplacian source, 34, 447, 574, 640
Lattice solution, 298
Lattice structures for VQ, 439
Leaky integration, 257, 335
Leaky prediction, 266
Learning period in APF, 293
Learning period in AQ, 192, 196, 215
Least mean square (LMS) algorithm, 305
Least significant bit (LSB), 177, 246, 248
Level-at-a-time A/D, 668–69
Levinson's recursion, 269
Linear adaptation, 402, 409, 411
Linear deltamodulation (LDM), 376
Linear estimation, 49
Linear PCM, 234–39
Linear predictor, 252, 259, 267
  of order 1, 2 and 3, 261–66
  of order *N*, 267–79, 295–97
Linear system, 52, 56, 61, 598
Linear transform, 53, 59, 512
  of order 2, 513, 539
  or order *N*, 517

Line averaging, 337
Line replacement, 327, 475
List coding, 432
Lloyd-Max quantizer, 131, 132, 134, 181, 185, 309, 438
Log-area coefficients, 300
Logarithmic quantization, 141–46, 259
Low activity image, 576
Low complexity coder, 11, 13, 241, 280, 372
Low detail image, 30, 37, 38, 41
Lowpass filter, 497, 500
Lowpass signal, 26, 36, 37, 40, 41, 64, 70, 88, 277, 381, 387

***M*-algorithm, 440, 443, 449, 456, 461**
MA(1) process, 54, 68, 69, 540
Matrix multiplication, 611
Matrix operations, 609
Max quantizer, 131, 132, 134, 181, 185, 309, 438
Mean opinion score (MOS), 2, 10, 658, 660
  adaptive transform coding of speech, 574
  APC-NF speech coding, 369
  APC speech coding, 317
  DM coding of speech, 373, 413, 419
  DPCM image coding, 283, 287, 337
  DPCM speech coding, 308, 311
  motion compensated coding, 324
  PCM speech coding, 241
  speech coding, 11
  sub-band coding of speech, 503, 505
  tandem coding, 242, 243, 308, 311
  transmission errors, 244–46, 308, 311
Mean square error (mse), 6, 120
Mean square value, 31, 44
  root, 44
Mean value, 31, 44
  conditional, 48
  effect of linear system, 52
Median filtering, 221, 249, 337
Medium complexity coder, 11, 13, 310, 316, 487, 504
Memoryless channel, 328
Memoryless source, 24, 621
Method of steepest descent, 303, 304
Metric tree, 445
Minimum distance algorithm, 438
Minimum distance code (MDC), 178, 187
Minimum mean square error (mmse), 6, 120,650
mmse prediction, 260, 268, 275, 330, 358, 626
Mismatched predictor, 271–74, 276, 291
Mismatched quantizer, 128, 139, 140, 146
Mobile radio, 337
Model sources, 27
  continuous time, 73
  pdf, 33,34
  psd, 58, 68
  2-D acf, 70
Modified Huffman coding (MHC), 475
Modified rhyme test (MRT), 663
Modulated noise reference unit (MNRU), 659
Modulation, 4, 19
Most significant bit (MSB), 177, 185, 244, 337, 338
Motion compensated coding, 312, 321, 323
Movement detection, 324
Moving average process, 54, 62, 68, 69, 540
mu-law (μ-law) companding, 145–46, 295, 668
Multidimensional coding, 662
Multipath search coding, 428 (*also see* Delayed decision coding)
Multiple threshold A/D, 671
Multiplicative noise, 659
Multipulse excited APC, 459
Mutual error variance, 179, 329
Mutual information, 22, 617, 627, 632, 636
μ-law companding, (*see* mu-law)

Narrow band noise, 240, 650
Natural binary code (NBC), 177, 187, 215, 243, 668, 671
N-block entropy, 638, 645
N-block DRF, 638, 645
Nearest neighbor interpolation, 221, 322, 337
Nearest neighbor vector, 438
Near-sample based prediction, 312
NFC (*see* Noise feedback coding)
Noise accumulation, 359
Noise entropy, 628
Noise feedback coding (NFC), 313, 317, 351–71, 647
Noise feedback PCM, 366
Noise filtering gain, 379
Noiseless coding, 466, 621
Noise masking, 282, 285, 354, 369, 570
Noise measurement, 239–41, 375

Noise shaping, 317, 354, 363, 368, 369, 505
Noise visibility, 138
Noise weighting filter, 355, 647, 652, 653
Noisy channel coding theorem, 628, 638
Nonadaptive prediction, 281, 293, 295, 316
Nonadaptive quantization, 201, 209, 376
Non-exhaustive search, 430, 443
Nonlinear estimation, 49
Non-mmse distortion measure, 430, 655
Non-negative integer constraint, 530
Non-singular matrix, 51
Nonstationarity, 34
  of speech spectrum, 291, 292, 305
  of speech variance, 34, 35
Nonuniform quantization, 129–41, 285
Normal equations, 269, 297
Notch-filter method for noise measurement, 240
N-th order deltamodulation (DM), 398
Nyquist sampling, 7, 9, 77, 90, 101, 374, 387, 487

Objective measures of performance, 651–57
One-bit coding, 127, 128, 156, 181, 182, 210, 260, 372, 436
One-sided polarity correlation, 298
Open-loop DPCM, 256, 352
Optimal channel mapping, 637, 639, 643
Optimal (mmse) prediction, 257
  of AR(1) process, 276, 278
  in D*PCM, 358
  of order 1, 261
  of order *N*, 269
Optimum bit allocation, 493, 524, 527, 528, 532
Optimum step size, 127
  in deltamodulation, 318, 383
  in one-bit quantization, 128
  with transmission errors, 182, 185
  in uniform quantization, 127
Orthogonality principle, 262, 268, 269, 306
Orthogonal transform, 518
Orthogonal vectors, 515
Orthonormal vectors, 515
Overload, 121
  amplitude, 225
  probability of, 122
  slope, 258, 378, 390
Oversampling, 77, 94, 373, 374, 417

Packet switching, 15, 289
Parallel A/D conversion, 671
Parallel D/A conversion, 672, 673
Parcor coefficients, 58, 298, 299
Parseval's theorem, 61, 603
Path error metric, 442
PCM (*see* Pulse code modulation)
Perceptual measures of performance, 7, 658–65
Perfect integration, 257
Periodicity, 312
Phase shift keying (PSK), 18, 19
Phase spectrum, 600
Phonetically balanced speech inputs, 663
Pitch measurement, 317–20
Pitch period, 15, 28, 29, 38, 314
Pitch prediction, 252, 312–17, 369
Pitch structure, 569
Planar predictor, 266
Polarity correlation, 298, 306, 318
Pole-zero predictor, 309, 325
Polynomial predictor, 272
Population of codebook, tree, and trellis, 429, 437
Positive definite acf matrix, 51
Positive semi-definite acf matrix, 67
Postfilter in NFC, 353, 364
Power spectral density (psd), 27
  and acf, 39, 55
  of AR(1) process, 64
  of AR(2) process, 66
  in continuous-time, 73
  and DFT, 39
  effect of linear system, 56
  estimation, 39
  of images, 40
  of integrated power spectrum, 75
  of MA(1) process, 70
  of raised cosine spectrum, 70, 78
  of second-order Butterworth spectrum, 77
  of speech, 40, 41
  of 2-D sources, 70
  of two-level psd, 79
  of white spectrum, 78
Power transfer factor, 329, 357, 417, 603, 605
Prediction coefficients, 58, 63, 260, 267, 269
  AR(1) source, 276, 278
  noisy input, 279
  in pitch prediction, 315
  with transmission errors, 333–35

Prediction error, 253, 274
filter, 275
in images, 284, 478
in speech, 291–92
variance, 257, 273
Prediction gain, 253, 259, 271, 272, 290, 291, 293, 295, 313, 330, 339, 510
Predictive coding, 17, 24, 26, 57, 58, 252, 510
Predictor re-optimization, 334
Pre-emphasized speech, 273, 503
Prefiltering, 3, 87
error, 87, 106, 107, 113
ideal, 105
of images, 107
in NFC, 353, 364
of speech, 107
Prefix condition, 622
Probability density function (pdf), 31
arcsine, 80
conditional pdf, 48, 80
of difference signal, 33
and distribution function, 43
first-order pdf, 50
of gamma source, 33, 259
of Gaussian source, 32, 33, 45, 259
of image amplitudes, 32
joint pdf, 46
of Laplacian source, 33
marginal pdf, 47
multivariate Gaussian, 52
Rayleigh, 80
of speech amplitudes, 32
triangular, 80
of uniform source, 33
Progressive transmission, 474, 546
Psophometric weighting, 653
Psychometric function, 661
Pulse code modulation (PCM), 7, 11, 13, 16, 221–51
audio, 224, 227, 228, 247
and DPCM, 256, 295
high quality, 8, 13, 224, 225
images, 13, 224, 226, 227
logarithmic, 11, 225
in SBC, 487
SNR in, 222
speech, 13, 225, 228–43
in TC, 512

Q-measure of quality, 651
Quadrature mirror filter, 487, 496
Quality, 659 (*also see* Objective measures; Subjective measures)
Quantization, 7, 115–220
adaptive, 188–210
A-law, 142–45
approximate formulas, 135–38
bit rate in, 116
of bounded input, 120
dithered, 164–75
entropy-coded, 146–58
of gamma pdf, (*see* Gamma source)
of Gaussian pdf, (*see* Gaussian source)
instantaneous, 115
iterative solutions, 131–35
of Laplacian pdf, (*see* Laplacian source)
Lloyd-Max, 131
logarithmic, 141–46
Max, 131
m-law, 138, 139, 211
mu-law (μ-law), 145–46, 176, 187
of narrow-band noise, 240, 241
one-bit, 127, 210
pdf-optimized nonuniform, 129–31
pdf-optimized uniform, 126, 127
of sine-wave input, 125, 240, 241
SNR in, 121
three-level, 129
of uniform pfd, (*see* Uniform pdf source)
vector, 116, 428–64
Quantization characteristic, 116
extension of, 287
midrise, 116, 242
midtread, 116, 242
non-symmetric, 129, 211
nonuniform, 116
of segmented *A*-law, 232
of segmented μ-law, 233
uniform, 116, 117
uniform threshold, 153
Quantization error, 118
acf and psd of, 161
and channel error, 176, 177
as function of interval, 137
gain-plus-additive-noise model, 163
orthogonality to input, 132
pdf of, 158–60
and reconstruction error in DM, 374

0328

and reconstruction error in DPCM, 257
and reconstruction error in NFC, 353
and reconstruction error in TC, 525
statistics with AR(1) input, 162
unconditional mean of, 132
variance, 119, 133, 137, 166, 179
waveform, 161
Quantization error feedback, 260, 278, 279, 285, 334, 353, 358, 411, 451, 453, 504, 505
Quantization interval, 116
Quantizer-input mismatch, 128, 139, 140, 146
Quantizer loading factor, 124, 125, 139, 208
Quantizer performance factor, 120, 257, 259, 491, 525
Quantizer reoptimization, 182, 184

***R*-2*R* resistor ladder D/A, 673**
Raised cosine spectrum, 70, 78, 399
Ramp comparison A/D converter, 668
Random field, 43
Random processes, 27, 43, 52, 55, 70
Random search, 440
Random sequence, 43
Random variables, 43, 47, 48
Raster scanning, 101
Rate distortion bound, 365, 428, 430
Rate distortion function (RDF), 21, 628, 632, 639
and entropy coding, 154, 155, 157
qualitative curves, 22
and quantizer performance, 121
and water filling, 530
R(*D*), RDF (*see* Rate distortion bound and function)
Real-valued Fourier matrix, 553
Reconstruction error, 6
in DM, 375, 381, 407
in DPCM, 257, 260, 279, 280–82, 312
filtered, 375
images, 407
in PCM, 8
in quantization, 118, 122, 165, 172, 177
spectra, 260, 381, 532, 642, 644
speech waveforms, 280, 375, 573
in TC, 573
Reconstruction error variance, 6
in continuous time, 111
in DM, 379, 381, 382
with DM transmission errors, 417
in DPCM, 253, 259
with DPCM transmission errors, 329, 330
in D*PCM, 357
with D*PCM transmission errors, 361
in NFC, 365
with PCM transmission errors, 243
and prefiltering error variance, 87
in SBC, 491, 493
in TC, 525, 527
with TC transmission errors, 577
with transmission errors, 180–86
Reconstruction filtering, 3, 86
casual, 92
continuous time, 111
ideal lowpass, 92, 93
Reconstruction level, 116, 134
Redundancy, 16, 17, 428, 618, 634
of discrete-amplitude source, 147
explicit, 246
and irrelevancy, 16, 17
natural, 247
and spectral flatness, 58
temporal, 322
use in error protection, 246, 247
Reflected binary code (RBC), 177
Relative address coding (RAC), 478
Relative element address designate (READ) coding, 478
Re-optimization for noisy channels, 182, 184, 334
RLC (*see* Run length coding)
Robust adaptive quantizer, 283, 506
Robust LMS algorithm, 306
Root mean square (rms) value, 31, 44
Run-length coding, 37, 322, 465–85

Sample-and-hold, 87–88
Sampling, 3, 26, 86–114
of bandpass signals, 96–98
errors in, 102
ideal, 92
integer-band, 97, 98
of moving images, 101
nonuniform, 94
rate, 5
theorems, 89, 90–91, 94, 98, 101
of 3-D signals, 101–2

Sampling, (*cont.*)
of 2-D signals, 101
uniform, 86
SBC (*see* Sub-band coding)
Search algorithms, 429, 439
Second order prediction, 263–65, 457
Segmental *SNR*, 194–95, 282, 457–58, 504
Self information, 617
Separable acf, 71, 264–66, 545
Separable transforms, 521
Sequency-ordered DWHT, 548
Sequentially adaptive predictor, 303
Serial D/A conversion, 674
Shannon-Fano integer constraint, 148
Shannon limit, 18, 631
Side information:
in APC, 315, 317, 369
in APF, 293, 298–303, 342
in AQF, 192, 215, 228, 248
in conditional replenishment coding, 322, 343
for error protection, 337, 338
in multipulse speech coding, 460
in TC, 569, 578
si function, 76, 91, 101, 111–12, 387, 607
Sigma deltamodulation, 399, 671
Signal-to-channel noise ratio, 19, 631
Signal-to-noise ratio (*SNR*), 2, 4, 5, 6, 655
in DM, 375, 379, 380–85, 397–98, 411
in DPCM, 243, 259, 282, 295
with DPCM transmission errors, 335
in D*PCM, 335, 357
gain over PCM (*see* Filtering gain; Prediction gain; Sub-band coding; Transform coding gain)
in log-quantization of speech, 145, 146
in NFC, 365
in PCM, 8, 222, 239–41
with PCM transmission errors, 243
in quantization of bounded input, 121, 125
in quantization of sine-wave, 125
refinements of, 7, 650–57
in SBC, 491
segmental, 194, 195, 282, 457, 458, 504
in TC, 528
with transmission errors, 176
and transmission rate, 5
in uniform quantization of images, 146
Sign-magnitude code, 177
Sine integral function, 76
Sine-wave input, 80, 125, 241, 389, 397–98, 411, 413, 650
Single-integration DM, 373–76, 394–98
Single-letter distortion, 633
Single-path search, 429
Slope-anticipating DM, 429
Slope-loading factor, 380, 388, 391
Slope overload, 258, 260, 282–85, 377–78, 390, 480
Soft-zone decision, 247
Song deltamodulator, 407
Source coding, 20
Source entropy, 616, 632
Source with memory, 619, 625, 641
Space shuttle DM, 409
Spatial correlation, 561
Spatial detail, 321
Spatial resolution, 322
Spectral flatness measure (sfm), 24, 57, 58, 253, 260, 493, 510, 530, 555, 625, 634, 644
of AR(1) process, 64
of AR(2) process, 67
of continuous time process, 83
of images, 253
of MA(1) process, 70
and power transfer factor, 61
of speech, 253
of 2-D AR(1) process, 73
Spectral moments, 391
Spectrum estimation, 276
Spectrum prediction, 312, 313
Speech, 2, 8
acf, 36
bandwidth, 9
pdf, 32, 33
psd, 40, 41, 277
sampling rate, 9
sfm, 253
waveforms, 28, 277
Speech clipping, 225, 318–19
Speech coding, 1
communications-quality, 10, 283, 317, 373, 500
high-quality, 10, 13, 228, 241, 308–11, 317
subjective quality in, 11, 241
synthetic-quality, 11
toll-quality, 10, 241, 311
variable-rate, 288, 289

0330

Speech enhancement, 247, 248, 337, 338
Speech intelligibility (*see* Intelligibility)
Speech masking, 487
Speech spectra, 503
Square matrix, 610
Stabilizing action of delayed decision, 429, 451, 460
Standard deviation, 31, 44
Stationarity, 31, 40, 45, 52, 60, 293
Statistical block protection coding, 337
Step-size asymmetry, 293
Step-size multipliers in AQB, 197, 200–208, 308, 402
Stochastic populations, 429, 437, 446, 449, 451, 457
Storage, 2, 248, 283, 372, 432, 461
Studio television, 227
Sub-band coding, 16, 31, 112, 486–509
Subjective measures of performance, 658–65
Subjective quality, 9, 658
  of APC coding with pitch prediction, 317
  of APC-NFC coding, 369
  of APC-SBC coding, 505
  and bit rate, 9
  versus bit error rate, 245, 246
  versus bit rate, 10, 11
  and coder complexity, 9, 241
  in DPCM-AQB and log-PCM speech, 282
  of DPCM-AQB-APB speech, 308, 311
  of motion compensated image coding, 324
Subjective *SNR*, 659
Subjective testing, 658
Suboptimal transforms, 546–63
Subsampling in images, 321–23
Successive approximation A/D, 667
Switched bit allocation, 571
Switched prediction, 265, 274, 287, 291, 301, 316
Switched quantization, 189, 190, 254, 286
Syllabic adaptation, 197, 282, 378, 400, 410
Symmetric matrix, 610
Synthetic quality, 11

Table look-up quantization, 409, 434
Tandem coding, 228, 242, 249, 308, 311
TC(*see* Transform coding)
Temporal correlation, 561
Temporal resolution, 322
Tenth-order prediction, 293, 317, 457
Third-order image prediction, 264, 510
Third-order prediction, 265–66, 287, 293, 316, 317
Three-level quantization, 129, 211, 369
Threshold coding, 535
Threshold sampling, 565
Time assignment speech interpolation (TASI), 12, 289
Time division multiplexing (TDM), 15, 223, 229
Time-domain coding, 16, 58, 486
Time-domain harmonic scaling (TDHS), 12, 504
Toeplitz distribution theorem, 55
Toeplitz matrix, 52, 269, 297, 550, 555, 559
Toll quality, 10, 228, 241, 308, 310, 311, 500, 660
Training in VQ, 437
Training mode quantizer, 204
Transfer function, 60, 603, 605
  of DM network, 395–97
  of DPCM network, 267, 272, 275, 277, 313, 329, 352, 354
Transform coding, 16, 17, 24, 27, 31, 57, 58, 510–90, 646
Transform coding gain, 511, 527–30, 542, 543, 555, 572
Transform coefficients, 58, 511, 512
Transition probability, 469, 620
Transmission, 2, 14, 15, 17
Transmission errors, 17 (*also see* Bit error rate)
  effect of binary code, 177, 187
  effect on DPCM images, 266
  effect on D*PCM, 361
  effect on images in DPCM,PCM, 327, 332
  effect on PCM images, 245
  effect on PCM speech, 244
  effect on quantization, 116, 175–88, 243
  effect on speech in DPCM,PCM, 326
  error propagation in DPCM, 325, 326, 343
  robust prediction, 306, 417
  robust quantization, 207
  subjective effects, 578
Tree coding, 16, 428, 432, 436, 447, 454, 457, 461
Tree depth, 433
Trellis coding, 16, 429, 434, 436, 450, 454, 457
Trellis depth, 435
Trellis intensity, 435

0331

Truncation error, 535
Two-channel DM, 416
Two-dimensional DCT, 560
Two-dimensional DFT, 556, 601
Two-dimensional linear system, 605
Two'dimensional prediction, 264–66, 287, 327, 331, 477
Two-dimensional transform, 520
Two-dimensional z-transform, 601
Two-level images (*see* Graphics)
Two-level psd, 79, 492
Two's complement code, 177, 215
Two-sided polarity correlation, 298, 306, 318, 342

Uniform pdf source, 34, 123, 125, 176, 187, 447, 640
Unitary transform, 523
Unit delay predictor, 257
Unvoiced speech, 28, 40, 41, 277, 292, 316

Variable bit allocation, 59, 322, 486, 551, 557, 571
Variable-length coding, 289, 323, 469, 473, 622
Variable-width sub-bands, 488, 489
Variance, 6, 31, 44
Vector DPCM, 439
Vector quantization, 16, 301, 303, 428–64
Video, 2, 9, 101
Videoconferencing, 324
Videophone, 9, 13, 77, 321–24, 387
Viterbi algorithm, 443
Vocoders, 10
Voiceband data, 78, 204, 273, 274, 399
Voiced speech, 28, 40, 41, 68, 277, 291, 316
Voice storage, (*see* Storage)

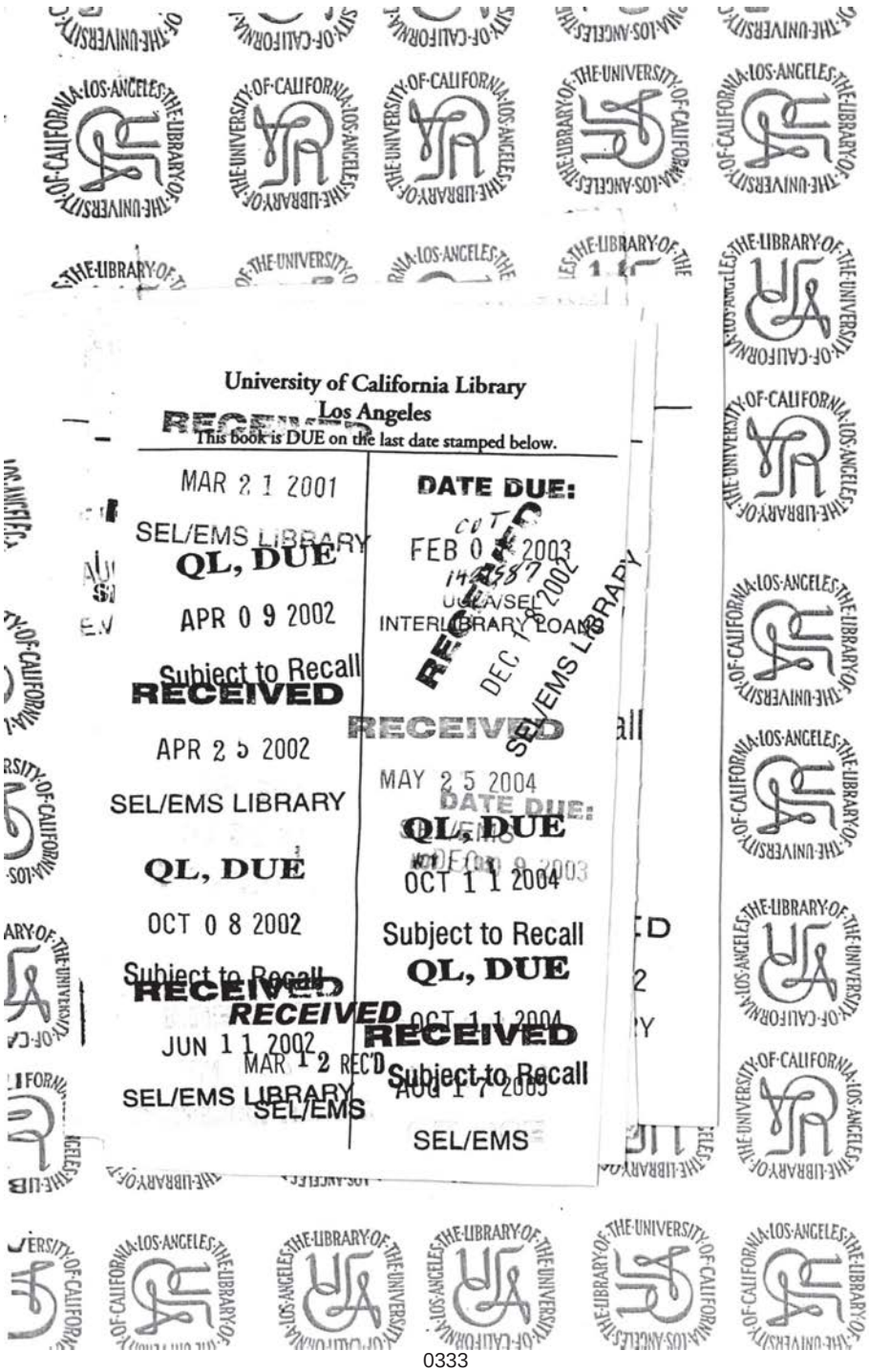
Water-filling procedure, 530, 641
Waveform coding, 1–25
Waveform decoder, 3
Waveform encoder, 2
Waveform-preserving coder, 9, 487
Waveforms, 2, 26–85
  amplitude versus time plots, 27
  gray-level image, 30
  multi-dimensional, 31
  properties, 15
  speech, 28
  two-level image, 29
Weighted resistor D/A, 672
Whitening filter, 59, 275, 357, 358, 359
White noise, 57, 60, 78, 275, 387
Wiener Hopf equation, 269, 303
Wiener Khinchine theorem, 56
Wiener process, 66
Word-at-a-time A/D, 671

Yule-Walker equations, 269

Zero-mean signal, 6, 31, 33, 49
Zero-memory quantization, 1, 16, 115–46, 428
Zero-order extrapolation, 247
Zonal coding, 565
Zonal sampling, 533, 535, 544, 550, 557, 565
z-transform, 599

6C623
14

0332


University of California Library
Los Angeles
This book is DUE on the last date stamped below.
MAR 2 1 2001
SEL/EMS LIBRARY
QL, DUE
APR 0 9 2002
Subject to Recall
RECEIVED
APR 2 5 2002
SEL/EMS LIBRARY
QL, DUE
OCT 0 8 2002
Subject to Recall
RECEIVED
JUN 1 1 2002
SEL/EMS LIBRARY
DATE DUE:
FEB 0 4 2003
UCLA/SEL
INTERLIBRARY LOAN
RECEIVED
DEC 18 2002
SEL/EMS LIBRARY
RECEIVED
MAY 2 5 2004
DATE DUE:
QL, DUE
OCT 1 1 2004
Subject to Recall
QL, DUE
OCT 1 1 2004
RECEIVED
MAR 1 2 REC'D
Subject to Recall
AUG 1 7 2005
SEL/EMS

