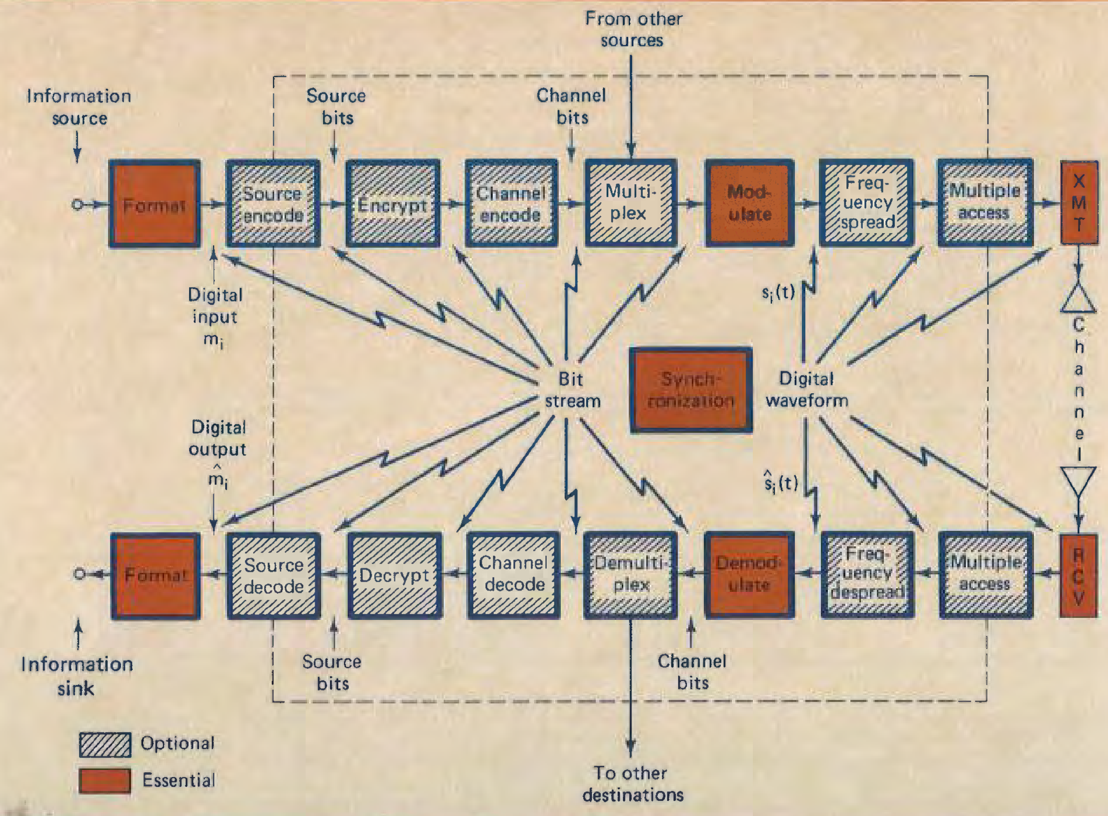
BERNARD SKLAR

DIGITAL COMMUNICATIONS
Fundamentals and Applications



Petitioner's Exhibit 1003
Page 001

DIGITAL COMMUNICATIONS

Fundamentals and Applications

BERNARD SKLAR

The Aerospace Corporation, El Segundo, California
and
University of California, Los Angeles

P T-R Prentice Hall
Englewood Cliffs, New Jersey 07632

Library of Congress Cataloging-in-Publication Data

SKLAR, BERNARD (date)
 Digital communications.

 Bibliography: p.
 Includes index.
 1. Digital communications. I. Title.
TK5103.7.S55 1988 621.38'0413 87-1316
ISBN 0-13-211939-0

Editorial/production supervision and
 interior design: Reynold Rieger
Cover design: Wanda Lubelska Design
Manufacturing buyers: Gordon Osbourne and Paula Benevento

Prentice-Hall, Inc.
A Paramount Communications Company
Englewood Cliffs, New Jersey 07632

Printed in the United States of America

20 19 18 17 16

ISBN 0-13-211939-0

Prentice-Hall International (UK) Limited, *London*
Prentice-Hall of Australia Pty. Limited, *Sydney*
Prentice-Hall Canada Inc., *Toronto*
Prentice-Hall Hispanoamericana, S.A., *Mexico*
Prentice-Hall of India Private Limited, *New Delhi*
Prentice-Hall of Japan, Inc., *Tokyo*
Simon & Schuster Asia Pte. Ltd., *Singapore*
Editora Prentice-Hall do Brasil, Ltda., *Rio de Janeiro*

Contents

Preface

This book is intended to provide a comprehensive coverage of digital communication systems for senior-level undergraduates, first-year graduate studcnts, and practicing engineers. Even though the emphasis of the book is on digital communications, necessary analog fundamentals are included, since analog waveforms are used for the radio transmission of digital signals.

The key feature of a digital communication system is that it deals with a finite set of discrete messages, in contrast to an analog communication system in which messages are defined on a continuum. The objective at the receiver of the digital system is *not* to reproduce a waveform with precision; it is, instead, to determine from a noise-perturbed signal which of the finite set of waveforms had been sent by the transmitter. In fulfillment of this objective, an impressive assortment of signal processing techniques has arisen over the past two decades.

The book develops these important techniques in the context of a unified structure. The structure, in block diagram form, appears at the beginning of each chapter; blocks in the diagram are emphasized, as appropriate, to correspond to the subject of that chapter. Major purposes of the book are (1) to add organization and structure to a field that has grown rapidly in the last two decades, and (2) to ensure awareness of the "big picture" even while delving into the details. The signals and key processing steps are traced from the information source through the transmitter, channel, receiver, and ultimately to the information sink. Signal transformations are organized according to functional classes: formatting and source coding, modulation, channel coding, multiplexing and multiple access, spreading, encryption, and synchronization. Throughout the book, emphasis is placed on system goals and the need to trade off basic system parameters such as signal-to-noise ratio, probability of error, and bandwidth (spectral) expenditure.

ORGANIZATION OF THE BOOK

It is assumed that the reader is familiar with Fourier methods and convolution. Appendix A reviews these techniques, emphasizing those properties that are particularly useful in the study of communication theory. It is also assumed that the reader has a knowledge of basic probability and has some familiarity with random variables. Appendix B builds on these disciplines for a short treatment on statistical decision theory with emphasis on hypothesis testing—so important in the understanding of detection theory. Chapter 1 introduces the overall digital communication system and the basic signal transformations that are highlighted in subsequent chapters. Some basic ideas of random variables and the additive white Gaussian noise (AWGN) model are reviewed. Also, the relationship between power spectral density and autocorrelation, and the basics of signal transmission through linear systems, are established. Chapter 2 covers the signal processing step, known as formatting, the step that renders an information signal compatible with a digital system. Chapter 2 also emphasizes the *transmission* of baseband signals. Chapter 3 deals with bandpass modulation and demodulation techniques. The detection of digital signals in Gaussian noise is stressed, and receiver optimization is examined. Chapter 4 deals with link analysis, an important subject for providing overall system insight; it considers some subtleties usually neglected at the college level. Chapters 5 and 6 deal with channel coding—a cost-effective way of providing improvement in system error performance. Chapter 5 emphasizes linear block coding, and Chapter 6 emphasizes convolutional coding.

Chapter 7 considers various modulation/coding system trade-offs dealing with probability of bit error performance, bandwidth efficiency, and signal-to-noise ratio. Chapter 8 deals with synchronization for digital systems. It covers phase-locked-loop implementation for achieving carrier synchronization; bit synchronization, frame synchronization, and network synchronization; and some fundamentals of synchronization as applied to satellite links.

Chapter 9 treats multiplexing and multiple access. It explores techniques that are available for utilizing the communication resource efficiently. Chapter 10 introduces spread-spectrum techniques and their application in such areas as multiple access, ranging, and interference rejection. This technology is particularly important for most military communication systems. The subject of source coding in Chapter 11 deals with data formatting, as is done in Chapter 2; the main difference between formatting and source coding is that source coding additionally involves data redundancy reduction. Rather than considering source coding immediately after formatting, source coding has purposely been treated in a later chapter. It is felt that the reader should be involved with the fundamental processing steps, such as modulation and channel coding, early in the book, before examining some of the special considerations of source coding. Chapter 12 covers some basic encryption/decryption ideas. It includes some classical encryption concepts, as well as some of the proposals for a class of encryption systems called public key cryptosystems.

If the book is used for a two-term course, a simple partitioning is suggested: the first six chapters to be taught in the first term, and the last six chapters in the second term. If the book is used for a one-term only course, it is suggested that the course material be selected from the following chapters: 1, 2, 3, 4, 5, 6, 8, and 10.

ACKNOWLEDGMENTS

This book is an outgrowth of my teaching activities at the University of California, Los Angeles, and my work in the Communications Division at The Aerospace Corporation. A number of people have contributed in many ways and it is a pleasure to acknowledge them. Dr. Maurice King, my colleague at Aerospace, carefully reviewed and made important contributions to each chapter. His continual assistance has been invaluable. He also contributed Chapter 8, Synchronization. Professor Fred Harris of San Diego State University suggested many improvements and contributed Chapter 11, Source Coding. I want to pay special thanks to Dr. Marvin Simon of the Jet Propulsion Laboratory for providing me with much encouragement and many valuable suggestions.

I also want to thank Professor Jim Omura of UCLA for sharing with me his considerable knowledge of encryption and thereby helping me improve Chapter 12. Professor Raymond Pickholtz of George Washington University gave me lots of beneficial advice throughout the writing process. Professors William Lindsey and Andreas Polydoros of the University of Southern California suggested important improvements. Professor James Modestino of Rensselaer Polytechnic Institute, Dr. Adam Lender of Lockheed Palo Alto Research Laboratory, and Professor Ron Iltis of the University of California, Santa Barbara, each provided valuable reviews. Dr. Todd Citron of Hughes Aircraft, Dr. Joe Odenwalder of MA/COM Linkabit, and Dr. Unjeng Cheng of Axiomatics were extremely helpful in the chapters on channel coding. Mr. Don Martin and Mr. Ned Feldman of The Aerospace Corporation made numerous suggestions and contributions. I also want to pay special thanks to Professor Wayne Stark of the University of Michigan, whose unique critical talents enhanced the manuscript's continuity.

The block diagrams in Figures 1.2 and 1.3, at each chapter opening, and on the cover of the book, first appeared in the two part paper: © 1983 IEEE; B. Sklar, "A Structured Overview of Digital Communications—A Tutorial Review," *IEEE Communications Magazine*, August and October, 1983. Permission from IEEE to reprint these figures throughout the book is gratefully acknowledged.

My students at UCLA and those at Aerospace used early versions of chapters of this book and made many helpful contributions. I am indebted to all those students who have taken my courses and thus helped me with this project. I also want to express my appreciation to my management at Aerospace, Mr. Hal

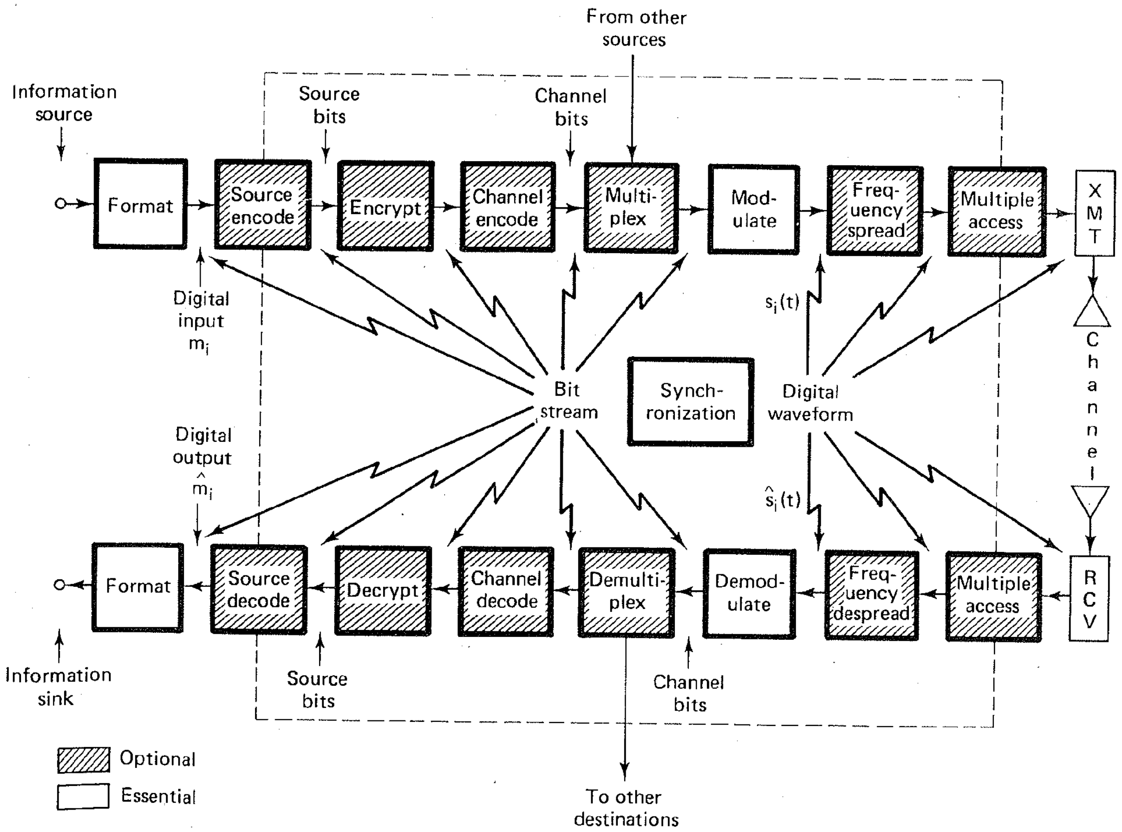
McDonnell and Mr. Fred Jones, for their indulgence and moral support. I want to acknowledge and thank Ms. Cynthia Dickson for her diligence and speed in typing the entire manuscript.

Finally, I want to thank my wife, Gwen, for her very unselfish support, her understanding, and her endurance of the many months I had time for only *one* devotion—the writing of this book.

BERNARD SKLAR
Tarzana, California

CHAPTER 1

Signals and Spectra



This book presents the ideas and techniques fundamental to digital communication systems. Emphasis is placed on system design goals and on the need for tradeoffs among basic system parameters such as signal-to-noise ratio (SNR), probability of error, and bandwidth expenditure. Transmission bandwidth is a finite resource; there is a growing awareness that bandwidth must be conserved, shared, and used efficiently. In general, we shall see that system performance can often be improved through the use of increased transmission bandwidth. However, such an increase is not always possible, because of physical limitations or the constraint of government regulations concerning the allocation and conservation of the usable electromagnetic spectrum.

We shall deal with the transmission of information (voice, video, or data) over a path (channel) that may consist of wires, waveguides, or free space. Frequently, the treatment will be in the context of a satellite communications link. Communication via satellites has two unique characteristics: (1) the ability to cover the globe with a flexibility that cannot be duplicated with terrestrial links, and (2) the availability of bandwidth exceeding anything previously available for intercontinental communications. Until recently, most satellite communication systems have been analog in nature. However, digital communication is becoming increasingly attractive because of the ever-growing demand for data communication and because digital transmission offers data processing options and flexibilities not available with analog transmission.

The principal feature of a digital communication system (DCS) is that during a finite interval of time, it sends a waveform from a finite set of possible waveforms, in contrast to an analog communication system, which sends a waveform from an infinite variety of waveform shapes with theoretically infinite resolution. In a DCS, the objective at the receiver is *not* to reproduce a transmitted waveform with precision; it is, instead, to determine from a noise-perturbed signal which waveform from the finite set of waveforms had been sent by the transmitter. An important measure of system performance in a DCS is the probability of error (P_E).

1.1 DIGITAL COMMUNICATION SIGNAL PROCESSING

1.1.1 Why Digital?

Why are communication systems, military and commercial alike, "going digital"? There are many reasons. The primary advantage is the ease with which digital signals, compared to analog signals, are regenerated. Figure 1.1 illustrates an ideal binary digital pulse propagating along a transmission line. The shape of the waveform is affected by two basic mechanisms: (1) as all transmission lines and circuits have some nonideal transfer function, there is a distorting effect on the ideal pulse; and (2) unwanted electrical noise or other interference further distorts the pulse waveform. Both of these mechanisms cause the pulse shape to degrade as a function of line length, as shown in Figure 1.1. During the time that the transmitted pulse can still be reliably identified (before it is degraded to an ambiguous state by the transmission line), the pulse is amplified by a digital amplifier that recovers its original ideal shape. The pulse is thus "reborn" or regenerated. Circuits that perform this function at regular intervals along a transmission system are called *regenerative repeaters*.

Digital circuits are less subject to distortion and interference than are analog circuits. Since binary digital circuits operate in one of two states, fully on or fully off, to be meaningful a disturbance must be large enough to change the circuit operating point from one state to the other. Such two-state operation facilitates signal regeneration and thus prevents noise and other disturbances from accu-

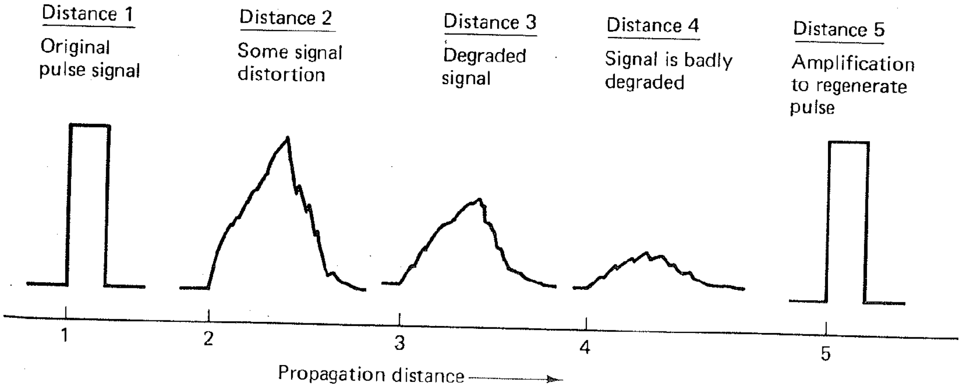


Figure 1.1 Pulse degradation and regeneration.

mulating in transmission. Analog signals, however, are *not* two-state signals; they can take an *infinite variety* of shapes. With analog circuits, even a small disturbance can render the reproduced waveform unacceptably distorted. Once the analog signal is distorted, the distortion cannot be removed by amplification. Since, with analog signals, accumulated noise is irrevocably bound to the signal, analog signals cannot be completely regenerated. Extremely low error rates producing high signal fidelity are possible through error detection and correction with digital techniques, but similar procedures are not available with analog.

There are other important advantages to digital communications. Digital circuits are *more reliable* and can be produced at lower cost than analog circuits. Also, digital hardware lends itself to *more flexible* implementation than analog hardware [e.g., microprocessors, digital switching, and large-scale integrated (LSI) circuits]. The combining of digital signals using time-division multiplexing (TDM) is *simpler* than the combining of analog signals using frequency-division multiplexing (FDM). Different types of digital signals (data, telegraph, telephone, television) can be treated as identical signals in transmission and switching—*a bit is a bit*. Also, for convenient switching, digital messages can be handled in autonomous groups called *packets*. Digital techniques lend themselves naturally to signal processing functions that protect against interference and jamming, or that provide encryption and privacy; such techniques are discussed in Chapters 10 and 12, respectively. Also, much data communication is computer to computer, or digital instrument or terminal to computer. Such digital terminations are naturally best served by digital communication links.

Most system choices entail trade-offs; system options are rarely all good or all bad. Thus far we have discussed only the *benefits* of digital transmission. What do you suppose are the *costs* or *liabilities*? A major disadvantage of digital transmission is that it typically requires a *greater system bandwidth* to communicate the same information in a digital format as compared to an analog format. Throughout this book we emphasize that bandwidth is a valuable resource, not always available. Bandwidth-efficient signaling techniques are discussed in Chapters 2 and 7. Another cost of digital transmission is that digital detection requires system synchronization (Chapter 8), whereas analog signals generally have no such requirement.

1.1.2 Typical Block Diagram and Transformations

The functional block diagram shown in Figure 1.2 illustrates the signal flow through a typical DCS. The upper blocks—format, source encode, encrypt, channel encode, multiplex, modulate, frequency spread, and multiple access—indicate the signal transformations from the source to the transmitter. The lower blocks indicate the signal transformations from the receiver to the sink; the lower blocks essentially reverse the signal processing steps performed by the upper blocks. It used to be that the only blocks within the dashed lines were the *modulator* and *demodulator,* together called a *modem*. During the past two decades, other signal processing functions were frequently incorporated within the same assembly as the modulator and demodulator. Consequently, at present, the term "modem"

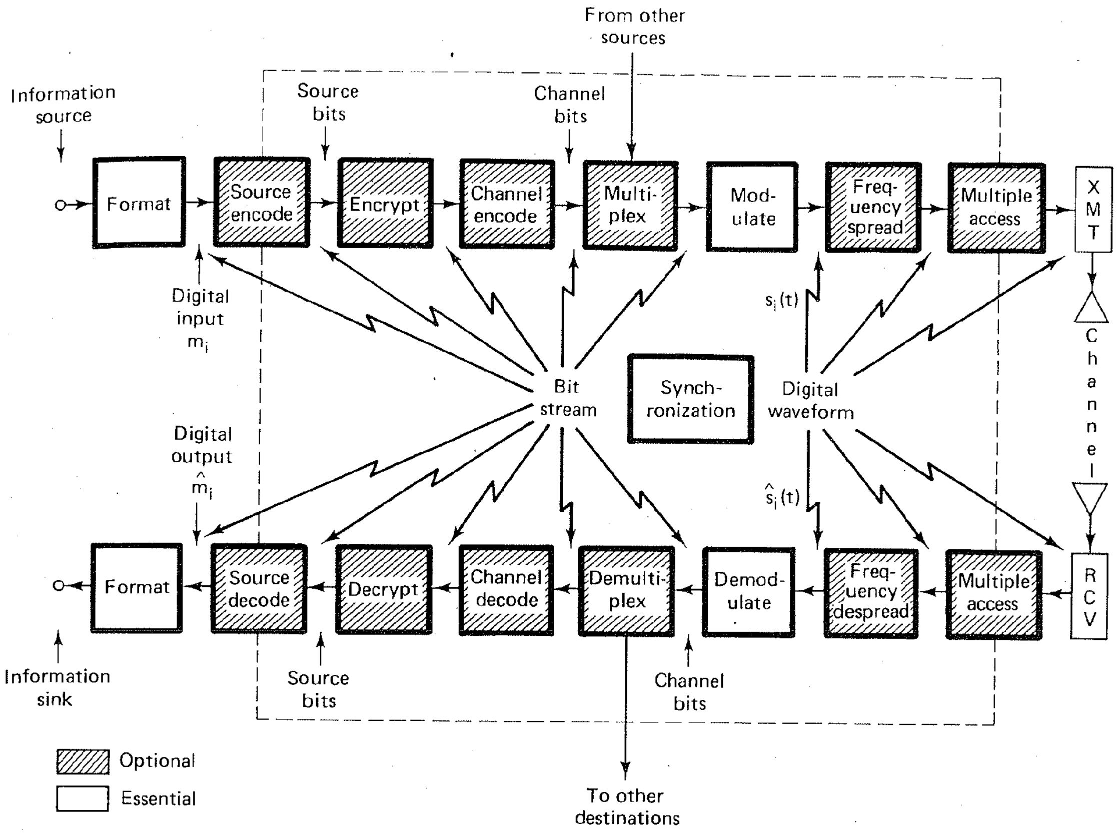


Figure 1.2 Block diagram of a typical digital communication system. (Reprinted with permission from B. Sklar, "A Structured Overview of Digital Communications," *IEEE Commun. Mag.*, August 1983, Fig. 1, p. 5. © 1983 IEEE.)

often encompasses all the processing steps shown within the dashed lines of Figure 1.2; when this is the case, the modem can be thought of as the "brains" of the system. Note that what constitutes a modem is not a precise concept; some of the blocks have purposely been shown *on* the dashed line rather than either inside or outside the modem. The transmitter and receiver can be thought of as the "muscles" of the system. The transmitter usually consists of a frequency up-conversion stage, a high-power amplifier, and an antenna. The receiver portion usually consists of an antenna, a low-noise amplifier (LNA), and a down-converter stage, typically to an intermediate frequency (IF).

Of all the signal processing steps, only formatting, modulation, and demodulation are essential for a DCS; the other processing steps within the modem are design options for specific system needs. *Formatting* transforms the source information into *digital symbols*; it makes the information compatible with the signal processing within a digital communication system. *Modulation* is the process by which the symbols are converted to *waveforms* that are compatible with the transmission channel.

The source encoding step produces analog-to-digital (A/D) conversion (for

↗ compression

analog sources) *and* removes *redundant or unneeded* information. Encryption prevents unauthorized users from understanding messages and from injecting false messages into the system. Channel coding, for a given data rate, can reduce the probability of error (P_E), or reduce the signal-to-noise ratio (SNR) requirement, at the expense of bandwidth or decoder complexity. Channel coding can also reduce the system bandwidth requirement at the expense of SNR or P_E performance. Frequency spreading can produce a signal that is less vulnerable to interference (both natural and intentional) and can be used to enhance the privacy of the communicators. Multiplexing and multiple access procedures combine signals that might have different characteristics or might originate from different sources, so that they can share a portion of the communications resource.

lots of trade offs

The flow of the signal processing steps shown in Figure 1.2 represents a typical arrangement; however, the blocks are sometimes implemented in a different order. For example, multiplexing can take place prior to channel encoding, or prior to modulation, or—with a two-step modulation process (subcarrier and carrier)—it can be performed between the two modulation steps. Similarly, spreading can take place anywhere along the transmission chain; its precise location depends on the particular technique used. Figure 1.2 illustrates the reciprocal aspect of the procedure; any signal processing step that takes place in the transmitting chain must be reversed in the receiving chain. The figure also indicates that from the source to the modulator a message, also called a *baseband signal* or a *bit stream,* is characterized by a sequence of digital symbols. After modulation, the message takes the form of a digitally encoded waveform or *digital waveform.* Similarly, in the reverse direction, a received message appears as a digital waveform until it is demodulated. Thereafter it takes the form of a bit stream for all further signal processing steps. At various points along the signal route, noise corrupts the waveform $s(t)$ so that its reception must be termed an estimate $\hat{s}(t)$. Such noise and its deleterious effects on system performance are considered in Chapter 4.

Figure 1.3 shows the basic signal processing functions, which may be viewed as transformations from one signal space to another. The transformations are classified into seven basic groups:

1. Formatting and source coding
2. Modulation/demodulation
3. Channel coding
4. Multiplexing and multiple access
5. Spreading
6. Encryption
7. Synchronization

Although this organization has some inherent overlap, it provides a useful structure for the book. Beginning with Chapter 2, the seven basic transformations are considered individually. In Chapter 2 we discuss the basic formatting techniques for transforming the source information into digital symbols, as well as

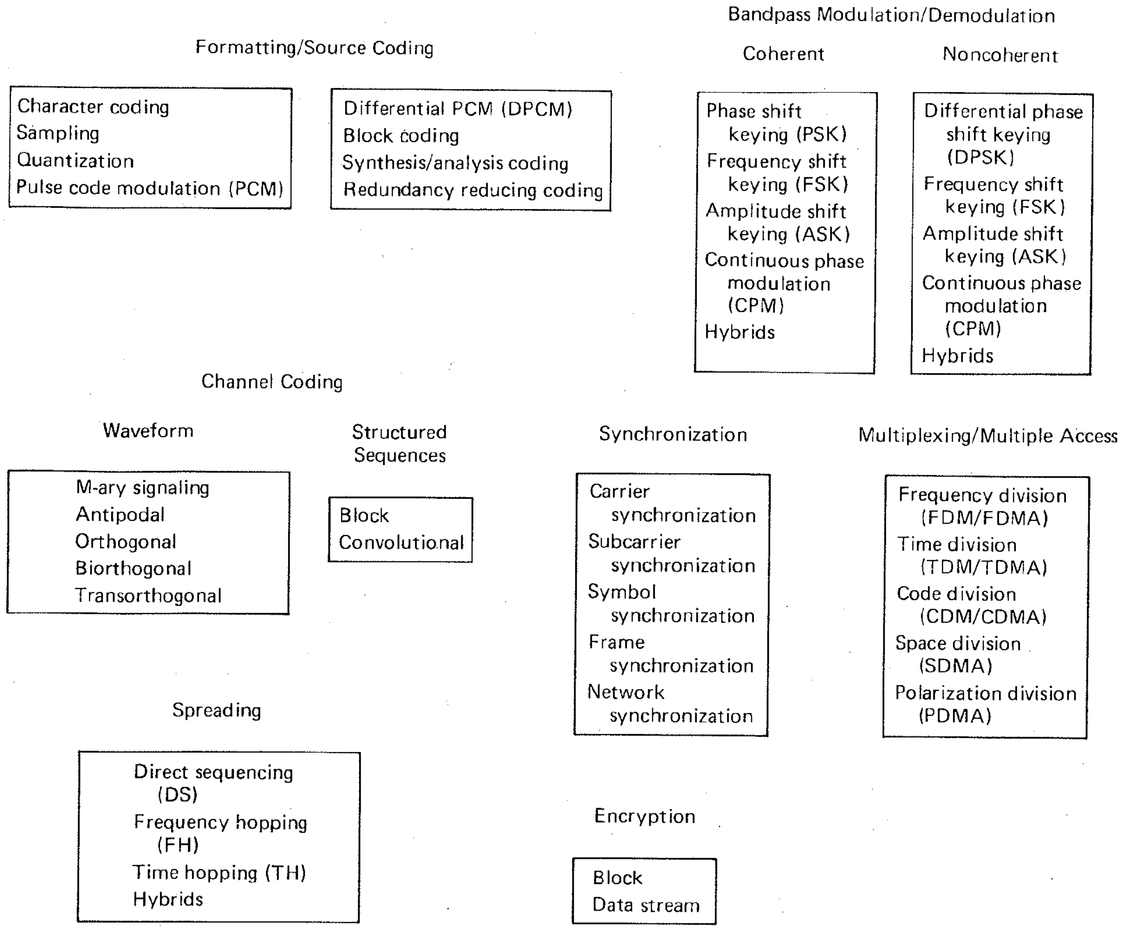


Figure 1.3 Basic digital communication transformations. (Reprinted with permission from B. Sklar, "A Structured Overview of Digital Communications," *IEEE Commun. Mag.*, August 1983, Fig. 2, p. 6. © 1983 IEEE.)

the selection of waveforms for making the symbols compatible with baseband transmission. As seen in Figure 1.3, formatting and source coding are grouped together; they are similar in that they involve data digitization. Since the term "source coding" has taken on the connotation of data redundancy reduction in addition to digitization, it is treated later, as a special formatting case, in Chapter 11.

In Figure 1.3, bandpass modulation/demodulation is partitioned into two basic categories, coherent and noncoherent. The process of *demodulation* involves the detection of the baseband information. Digital demodulation is typically accomplished with the aid of reference waveforms. When the references contain all the signal attributes, particularly phase information, the process is termed *coherent*; when phase information is not used, the process is termed *noncoherent*. Both techniques are detailed in Chapter 3.

Chapter 4 is devoted to link analysis. In the past, this area has received little attention in colleges or in textbooks, probably because it was considered straightforward and not worthy of detailed discussion. However, of the many specifications, analyses, and tabulations that support a developing communication system, link analysis stands out in its ability to provide overall system insight. In Chapter 4 we bring together all the link fundamentals that are essential for the analysis of most communication systems.

Channel coding deals with the techniques used to enhance digital signals so that they are less vulnerable to such channel impairments as noise, fading, and jamming. In Figure 1.3 channel coding is partitioned into two basic categories, waveform coding and structured sequences. *Waveform coding* involves the use of new waveforms, yielding improved detection performance over that of the original waveforms. *Structured sequences* involve the use of redundant bits to determine whether or not an error has occurred due to noise on the channel. One of these techniques, known as automatic repeat request (ARQ), simply recognizes the occurrence of an error and requests that the sender retransmit the message; other techniques, called forward error correction (FEC), are capable of automatically correcting the errors (within specified limitations). Under the heading of structured sequences, we shall discuss the two prevalent techniques, block coding and convolutional coding. In Chapter 5 we consider waveform coding and linear block coding. In Chapter 6 we consider convolutional coding, Viterbi decoding (and other decoding algorithms), hard versus soft decoding procedures, and interleaving and deinterleaving.

In Chapter 7 we summarize the design goals for a communication system and present various modulation and coding trade-offs that need to be considered in the design of a system. We discuss theoretical limitations such as the Nyquist criterion and the Shannon limit. We also examine bandwidth-efficient modulation schemes.

Chapter 8 deals with synchronization. In digital communications, synchronization involves the estimation of both time and frequency. The subject is partitioned as shown in Figure 1.3. Coherent systems need to synchronize their frequency reference with the carrier (and possibly subcarrier) in both frequency and phase. For noncoherent systems, phase synchronization is not needed. The fundamental time-synchronization process is symbol synchronization. The demodulator needs to know when to start and end the symbol detection procedure; a timing error will degrade detection performance. The next time-synchronization level, frame synchronization, allows the reconstruction of the message; and network synchronization allows coordination with other users in order to use the resource efficiently. In Chapter 8 we are concerned with the alignment of the timing of spatially separated periodic processes; the alignment is illustrated for the case of a satellite communications link.

Chapter 9 deals with multiplexing and multiple access. The two terms mean very similar things. Both involve the idea of resource sharing. The main difference between the two is that *multiplexing* takes place locally (e.g., on a printed circuit board, within an assembly, or even within a facility), and *multiple access* takes place remotely (e.g., multiple users share the use of a satellite transponder). Multiplexing involves an algorithm that is known a priori; usually, it is hard-wired into the system. Multiple access, on the other hand, is generally adaptive and may require overhead to enable the algorithm to operate. In Chapter 9 we discuss the classical ways of sharing the resource: frequency division, time division, and code division. We also consider some of the multiple access techniques that havc emerged as a result of satellite communications.

Chapter 10 introduces a transformation of primary importance in military communications called spreading. The chapter deals with the spread-spectrum techniques that are emerging as important for achieving interference protection, privacy, or flexible access of the communications resource.

Chapter 11 treats source coding—techniques that deal with thc task of forming efficient descriptions of source information. Source coding can be applied to digital data and to waveform signals; it can reduce data redundancy and thus reduce data rates. We will see that the advantage of source coding is a reduction of the system resources (i.e., bandwidth) required to describe the information.

The final chapter of the book, Chapter 12, deals with encryption and decryption, whose basic goals are privacy and authentication. *Privacy* refers to preventing unauthorized persons from extracting information (eavesdropping) from the channel. *Authentication* refers to preventing unauthorized persons from injecting spurious signals (spoofing) into the channel. In this chapter we highlight the data encryption standard (DES) and some current ideas for a class of encryption systems called public key cryptosystems.

1.1.3 Basic Digital Communication Nomenclature

Some of the basic digital signal nomenclature that frequently appears in digital communication literature is as follows:

Information source: the device producing information to be communicated by means of the DCS. Information sources can be analog or discrete. The output of an analog source can have any value in a continuous range of amplitudes, whereas the output of a discrete information source takes its value from a finite set. Analog information sources can be transformed into digital sources through the use of sampling and quantization. Sampling and quantization techniques called formatting and source coding (see Figure 1.3) are described in Chapters 2 and 11.

Textual message: a sequence of characters (see Figure 1.4a). For digital transmission, the message will be a sequence of digits or symbols from a finite symbol set or alphabet.

Character: a member of an alphabet or set of symbols (see Figure 1.4b). Characters may be mapped into a sequence of binary digits. There are several standardized codes used for character encoding, including the American Standard Code for Information Interchange (ASCII), Extended Binary Coded Decimal Interchange Code (EBCDIC), Hollerith, Baudot, Murray, and Morse.

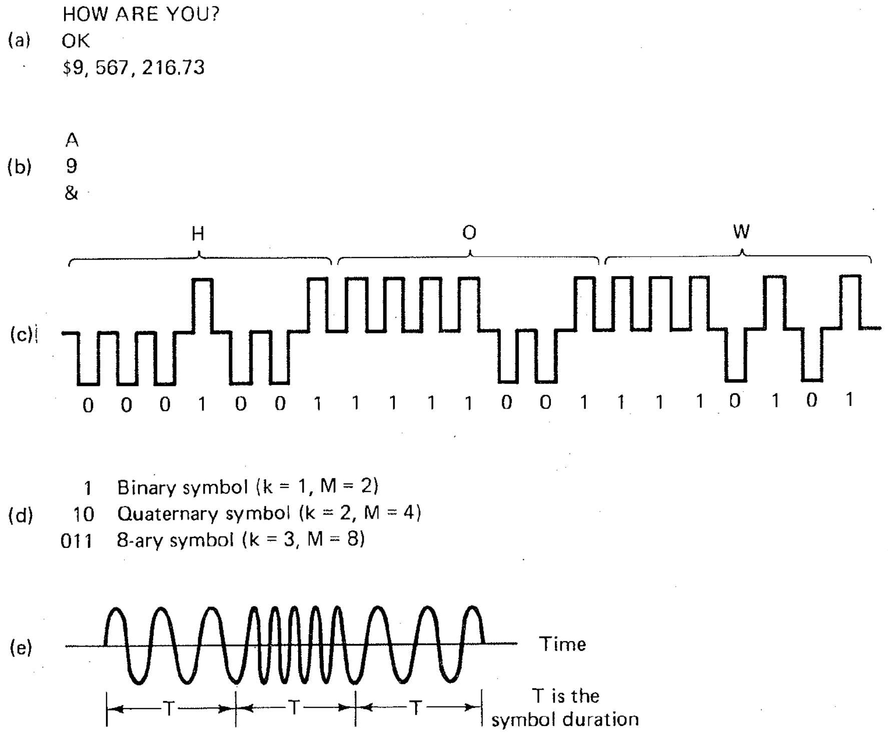


Figure 1.4 Nomenclature examples. (a) Textual messages. (b) Characters. (c) Bit stream (7-bit ASCII). (d) Symbols m_i, $i = 1, \dots, M$, $M = 2^k$. (e) Bandpass digital waveform $s_i(t)$, $i = 1, \dots, M$.

Binary digit (bit): the fundamental information unit for all digital systems. The term *bit* is also used as a unit of information content; this second usage is described in Chapter 7.

Bit stream: a sequence of binary digits (ones and zeros). Sometimes, a sequence of two-level pulses is used as a convenient illustration of the bit stream. The bit stream in Figure 1.4c uses a 7-bit ASCII character code for representing the message "HOW." A bit stream is often termed a *baseband signal,* which implies that its spectral content extends from (or near) dc up to some finite value, usually less than a few megahertz.

Symbol (digital message): groups of k bits considered as a unit or character m_i, from a finite symbol set or alphabet (see Figure 1.4d). The size of the alphabet, M, is $M = 2^k$ (i.e., k is the number of bits in the symbol). For transmission, each m_i symbol ($i = 1, \dots, M$) will be represented by a corresponding waveform $s_1(t)$, $s_2(t)$, . . . , $s_M(t)$. The symbol, m_i, is sent by transmitting the digital waveform, $s_i(t)$, for T seconds, the symbol time duration. The next symbol is sent during the next time interval, T. The fact that the symbol set transmitted by the DCS is finite is a primary difference between a DCS and an analog system. The DCS receiver need only decide which of the M waveforms was transmitted; however, an analog receiver must be capable of accurately estimating a continuous range of waveforms.

Digital waveform: a voltage or current waveform (a pulse for baseband transmission, or a sinusoid for bandpass transmission) that represents a digital symbol. The waveform characteristics (amplitude, width, position for pulses, or amplitude, frequency, phase for sinusoids) allow its identification as one of the symbols in the finite symbol alphabet. Figure 1.4e shows an example of a bandpass digital waveform. Even though the waveform is sinusoidal, and consequently has an analog appearance, it is called a *digital waveform* because it is encoded with digital information. In the figure, during each time interval, T, a preassigned frequency indicates the value of a digit.

Data rate: data rate in bits per second (bits/s) is given by $R = k/T = (1/T) \log_2 M$ bits/s, where k bits identify a symbol from an $M = 2^k$-symbol alphabet, and T is the k-bit symbol duration.

1.1.4 Digital versus Analog Performance Criteria

A principal difference between analog and digital communication systems has to do with the way in which we evaluate their performance. Analog systems draw their waveforms from a continuum, which therefore forms an infinite set; that is, a receiver must deal with an infinite number of possible waveshapes. The figure of merit for the performance of analog communication systems is a fidelity criterion, such as signal-to-noise ratio, percent distortion, or expected mean-square error between the transmitted and received waveforms.

By contrast, a digital communication system transmits signals that represent digits. These digits form a finite set or alphabet, and the set is known a priori to the receiver. A figure of merit for digital communication systems is the probability of incorrectly detecting a digit, or the probability of error (P_E).

1.2 CLASSIFICATION OF SIGNALS

1.2.1 Deterministic and Random Signals

A signal can be classified as *deterministic*, meaning that there is no uncertainty with respect to its value at any time, or as *random*, meaning that there is some degree of uncertainty before the signal actually occurs. Deterministic signals or waveforms are modeled by explicit mathematical expressions, such as $x(t) = 5 \cos 10t$. For a random waveform it is *not* possible to write such an explicit expression. However, when examined over a long period, a random waveform, also referred to as a *random process*, may exhibit certain regularities that can be described in terms of probabilities and statistical averages. Such a model, in the form of a probabilistic description of the random process, is particularly useful for characterizing signals and noise in communication systems.

1.2.2 Periodic and Nonperiodic Signals

A signal $x(t)$ is called *periodic in time* if there exists a constant $T_0 > 0$ such that

$$x(t) = x(t + T_0) \qquad \text{for } -\infty < t < \infty \tag{1.1}$$

where t denotes time. The smallest value of T_0 that satisfies this condition is called the *period* of $x(t)$. The period T_0 defines the duration of one complete cycle of $x(t)$. A signal for which there is no value of T_0 that satisfies Equation (1.1) is called a *nonperiodic signal.*

1.2.3 Analog and Discrete Signals

An *analog signal, $x(t)$*, is a continuous function of time; that is, $x(t)$ is uniquely defined for all t. An electrical analog signal arises when a physical waveform (e.g., speech) is converted into an electrical signal by means of a transducer. By comarison, a *discrete signal, $x(kT)$*, is onc that exists only at discrete times; it is characterized by a sequence of numbers defined for each time, kT, where k is an integer and T is a fixed time interval.

1.2.4 Energy and Power Signals

An electrical signal can be represented as a voltage, $v(t)$, or a current, $i(t)$, with instantaneous power $p(t)$ across a resistor $\mathcal{R}$ defined by

$$p(t) = \frac{v^2(t)}{\mathcal{R}} \tag{1.2}$$

or

$$p(t) = i^2(t)\mathcal{R} \tag{1.3}$$

In communication systems, power is often normalized by assuming $\mathcal{R}$ to be 1 Ω, although $\mathcal{R}$ may be another value in the actual circuit. If the actual value of the power is needed, it is obtained by "denormalization" of the normalized value. For the normalized case, Equations (1.2) and (1.3) have the same form. Therefore, regardless of whether the signal is a voltage or current waveform, the normalization convention allows us to express the instantaneous power as

$$p(t) = x^2(t) \tag{1.4}$$

where $x(t)$ is either a voltage or a current signal. The energy dissipated during the time interval $(-T/2, T/2)$ by a real signal with instantaneous power expressed by Equation (1.4) can then be written as

$$E_x^T = \int_{-T/2}^{T/2} x^2(t)\, dt \tag{1.5}$$

and the average power dissipated by the signal during the interval is

$$P_x^T = \frac{1}{T}\int_{-T/2}^{T/2} x^2(t)\, dt \tag{1.6}$$

The performance of a communication system depends on the detected signal *energy*; higher-energy signals are detected more reliably (with fewer errors) than are lower-energy signals—the transmitted *energy does the work.* On the other hand, power is the *rate* at which energy is delivered. It is important for different reasons. The power determines the voltages that must be applied to a transmitter and the intensities of the electromagnetic fields that one must contend with in radio systems (i.e., fields in waveguides that connect the transmitter to the antenna, and fields around the radiating elements of the antenna).

In analyzing communication signals it is often desirable to deal with the *waveform energy*. We classify $x(t)$ as an *energy signal* if, and only if, it has nonzero but finite energy ($0 < E_x < \infty$) for all time, where

$$\begin{aligned} E_x &= \lim_{T\to\infty} \int_{-T/2}^{T/2} x^2(t)\, dt \\ &= \int_{-\infty}^{\infty} x^2(t)\, dt \end{aligned} \tag{1.7}$$

In the real world we always transmit signals having finite energy ($0 < E_x < \infty$). However, in order to describe *periodic signals,* which by definition [Equation (1.1)] exist for all time and thus have infinite energy, and in order to deal with random signals that have infinite energy, it is convenient to define a class of signals called *power signals.* A signal is defined to be a power signal if, and only if, it has finite but nonzero power ($0 < P_x < \infty$) for all time, where

$$P_x = \lim_{T\to\infty} \frac{1}{T}\int_{-T/2}^{T/2} x^2(t)\, dt \tag{1.8}$$

The energy and power classifications are mutually exclusive. An energy signal has finite energy but *zero average power,* whereas a power signal has finite average power but *infinite energy.* A waveform in a system may be constrained in either its power or energy values. As a general rule, periodic signals and random signals are classified as power signals, while signals that are both deterministic and nonperiodic are classified as energy signals [1, 2].

As mentioned earlier, signal energy and power are both important parameters in specifying a communication system. The classification of a signal as either an energy signal or a power signal is a convenient model to facilitate the mathematical treatment of various signals and noise.

1.2.5 The Unit Impulse Function

A useful function in communication theory is the unit impulse or *Dirac delta function,* $\delta(t)$. The impulse function is an abstraction—an infinitely large amplitude pulse, with zero pulse width, and unity weight (area under the pulse), concentrated at the point where its argument is zero. The unit impulse is characterized by the following relationships:

$$\int_{-\infty}^{\infty} \delta(t)\, dt = 1 \tag{1.9}$$

$$\delta(t) = 0 \qquad \text{for } t \neq 0 \tag{1.10}$$

$$\delta(t) \text{ is unbounded at } t = 0 \tag{1.11}$$

$$\int_{-\infty}^{\infty} x(t)\delta(t - t_0)\, dt = x(t_0) \tag{1.12}$$

The unit impulse function, $\delta(t)$, is not a function in the usual sense. When operations involve $\delta(t)$, the convention is to interpret $\delta(t)$ as a unit-area pulse of finite amplitude and nonzero duration, after which the limit is considered as the pulse duration approaches zero. $\delta(t - t_0)$ can be depicted graphically as a spike located at $t = t_0$ with height equal to its integral or area. Thus $A\delta(t - t_0)$ with A constant represents an impulse function whose area or weight is equal to A, that is zero everywhere except at $t = t_0$.

Equation (1.12) is known as the *sifting* or *sampling property* of the unit impulse function; the unit impulse multiplier selects a sample of the function $x(t)$ evaluated at $t = t_0$.

1.3 SPECTRAL DENSITY

The *spectral density* of a signal characterizes the distribution of the signal's energy or power in the frequency domain. This concept is particularly important when considering filtering in communication systems. We need to be able to evaluate the signal and noise at the filter output. The energy spectral density (ESD) or the power spectral density (PSD) is used in the evaluation.

1.3.1 Energy Spectral Density

The total energy of a real-valued energy signal $x(t)$, defined over the interval $(-\infty, \infty)$, is described by Equation (1.7). Using Parseval's theorem [1], we can relate the energy of such a signal expressed in the time domain to the energy expressed in the frequency domain, as follows:

$$E_x = \int_{-\infty}^{\infty} x^2(t)\, dt = \int_{-\infty}^{\infty} |X(f)|^2\, df \tag{1.13}$$

where $X(f)$ is the Fourier transform of the nonperiodic signal $x(t)$ (for a review of Fourier techniques, see Appendix A). Let $\Psi_x(f)$ denote the squared magnitude spectrum, defined as

$$\Psi_x(f) = |X(f)|^2 \tag{1.14}$$

The quantity $\Psi_x(f)$ is the waveform *energy spectral density* (ESD) of the signal $x(t)$. Therefore, from Equation (1.13), we can express the total energy of the signal $x(t)$ by integrating the spectral density with respect to frequency, as follows:

$$E_x = \int_{-\infty}^{\infty} \Psi_x(f)\, df \tag{1.15}$$

This equation states that the energy of a signal is equal to the area under the $\Psi_x(f)$ versus frequency curve. Energy spectral density describes the signal energy per unit bandwidth measured in joules/hertz. There are equal energy contributions from both positive and negative frequency components, since for a real signal, $x(t)$, $|X(f)|$ is an even function of frequency. Therefore, the energy spectral density is symmetrical in frequency about the origin, and thus the total energy of the signal $x(t)$ can be expressed as

$$E_x = 2 \int_{0}^{\infty} \Psi_x(f)\, df \tag{1.16}$$

1.3.2 Power Spectral Density

The average power, P_x, of a real-valued power signal, $x(t)$, is defined in Equation (1.8). If $x(t)$ is a *periodic signal* with period T_0, it is classified as a power signal. The expression for the average power of a periodic signal takes the form of Equation (1.6), where the time average is taken over the signal period T_0, as follows:

$$P_x = \frac{1}{T_0} \int_{-T_0/2}^{T_0/2} x^2(t)\, dt \tag{1.17a}$$

Parseval's theorem for a real-valued periodic signal [1] takes the form

$$P_x = \frac{1}{T_0} \int_{-T_0/2}^{T_0/2} x^2(t)\, dt = \sum_{n=-\infty}^{\infty} |c_n|^2 \tag{1.17b}$$

where the $|c_n|$ terms are the complex Fourier series coefficients of the periodic signal (see Appendix A).

To apply Equation (1.17b), we need only know the magnitude of the coefficients, $|c_n|$. The *power spectral density* (PSD) function, $G_x(f)$, of the periodic signal, $x(t)$, is a real, even, and nonnegative function of frequency that gives the distribution of the power of $x(t)$ in the frequency domain, defined as

$$G_x(f) = \sum_{n=-\infty}^{\infty} |c_n|^2\, \delta(f - nf_0) \tag{1.18}$$

Equation (1.18) defines the power spectral density of a periodic signal, $x(t)$, as a succession of the weighted delta functions. Therefore, the PSD of a periodic signal is a discrete function of frequency. Using the PSD defined in Equation (1.18), we can now write the average normalized power of a real-valued signal, as follows:

$$P_x = \int_{-\infty}^{\infty} G_x(f)\, df = 2\int_{0}^{\infty} G_x(f)\, df \tag{1.19}$$

Equation (1.18) describes the PSD of periodic (power) signals only. If $x(t)$ is a nonperiodic signal it *cannot* be expressed by a Fourier series, and if it is a nonperiodic power signal (having infinite energy) it *may not* have a Fourier transform. However, we may still express the power spectral density of such signals in the *limiting sense*. If we form a *truncated version*, $x_T(t)$, of the nonperiodic power signal, $x(t)$, by observing it only in the interval $(-T/2, T/2)$, then $x_T(t)$ has finite energy, and has a proper Fourier transform, $X_T(f)$. It can be shown [2] that the power spectral density of the nonperiodic $x(t)$ can then be defined in the limit as

$$G_x(f) = \lim_{T\to\infty} \frac{1}{T} |X_T(f)|^2 \tag{1.20}$$

Example 1.1 Average Normalized Power

(a) Find the average normalized power in the waveform, $x(t) = A \cos 2\pi f_0 t$, using time averaging.

(b) Repeat part (a) using the summation of spectral coefficients.

Solution

(a) Using Equation (1.17a), we have

$$\begin{aligned} P_x &= \frac{1}{T_0}\int_{-T_0/2}^{T_0/2} A^2 \cos^2 2\pi f_0 t\, dt \\ &= \frac{A^2}{2T_0}\int_{-T_0/2}^{T_0/2} (1 + \cos 4\pi f_0 t)\, dt \\ &= \frac{A^2}{2T_0}(T_0) = \frac{A^2}{2} \end{aligned}$$

(b) Using Equations (1.18) and (1.19) gives us

$$G_x(f) = \sum_{n=-\infty}^{\infty} |c_n|^2 \delta(f - nf_0)$$

$$\left.\begin{aligned} c_1 &= c_{-1} = \frac{A}{2} \\ c_n &= 0 \qquad \text{for } n = 0, \pm 2, \pm 3, \ldots \end{aligned}\right\} \quad \text{(see Appendix A)}$$

$$G_x(f) = \left(\frac{A}{2}\right)^2 \delta(f - f_0) + \left(\frac{A}{2}\right)^2 \delta(f + f_0)$$

$$P_x = \int_{-\infty}^{\infty} G_x(f)\, df = \frac{A^2}{2}$$

1.4 AUTOCORRELATION

1.4.1 Autocorrelation of an Energy Signal

Correlation is a matching process; *autocorrelation* refers to the matching of a signal with a delayed version of itself. The autocorrelation function, $R_x(\tau)$, of a real-valued energy signal, $x(t)$, is defined as

$$R_x(\tau) = \int_{-\infty}^{\infty} x(t)x(t + \tau)\,dt \qquad \text{for } -\infty < \tau < \infty \tag{1.21}$$

The autocorrelation function, $R_x(\tau)$, provides a measure of how closely the signal matches a copy of itself as the copy is shifted τ units in time. The variable τ plays the role of a scanning or searching parameter. $R_x(\tau)$ is not a function of time; it is only a function of the time difference, τ, between the waveform and its shifted copy.

The autocorrelative function of a real-valued *energy* signal has the following properties:

1. $R_x(\tau) = R_x(-\tau)$	symmetrical in τ about zero
2. $R_x(\tau) \leq R_x(0)$ for all τ	maximum value occurs at the origin
3. $R_x(\tau) \leftrightarrow \Psi_x(f)$	autocorrelation and ESD form a Fourier transform pair, as designated by the double-headed arrows
4. $R_x(0) = \int_{-\infty}^{\infty} x^2(t)\,dt$	value at the origin is equal to the energy of the signal

If items 1 through 3 are satisfied, $R_x(\tau)$ satisfies the properties of an autocorrelation function. Property 4 can be derived from property 3 and thus need not be included as a basic test.

1.4.2 Autocorrelation of a Periodic (Power) Signal

The autocorrelation function of a real-valued power signal $x(t)$ is defined as

$$R_x(\tau) = \lim_{T\to\infty} \frac{1}{T} \int_{-T/2}^{T/2} x(t)x(t + \tau)\,dt \qquad \text{for } -\infty < \tau < \infty \tag{1.22}$$

When the power signal, $x(t)$, is periodic with period T_0, the time average in Equation (1.22) may be taken over a *single period*, T_0, and the autocorrelation function can be expressed as follows:

$$R_x(\tau) = \frac{1}{T_0} \int_{-T_0/2}^{T_0/2} x(t)x(t + \tau)\,dt \qquad \text{for } -\infty < \tau < \infty \tag{1.23}$$

The autocorrelation function of a real-valued *periodic* signal has properties similar to those of an energy signal, as follows:

1. $R_x(\tau) = R_x(-\tau)$	symmetrical in τ about zero
2. $R_x(\tau) \leq R_x(0)$ for all τ	maximum value occurs at the origin
3. $R_x(\tau) \leftrightarrow G_x(f)$	autocorrelation and PSD form a Fourier transform pair
4. $R_x(0) = \frac{1}{T_0}\int_{-T_0/2}^{T_0/2} x^2(t)\, dt$	value at the origin is equal to the average power of the signal

1.5 RANDOM SIGNALS

Random !!

The main objective of a communication system is the transfer of information over a channel. All useful message signals appear random; that is, the receiver does not know, a priori, which of the possible message waveforms will be transmitted. Also, the noise that accompanies the message signals is due to random electrical signals. Therefore, we need to be able to form efficient descriptions of random signals.

1.5.1 Random Variables

Let a *random variable, X(A)*, represent the functional relationship between a random event, A, and a real number. For notational convenience we shall designate the random variable by X, and let the functional dependence upon A be implicit. The random variable may be discrete or continuous. The *distribution function*, $F_X(x)$, of the random variable, X, is given by

$$F_X(x) = P(X \leq x) \tag{1.24}$$

where $P(X \leq x)$ is the probability that the value taken by the random variable, X, is less than or equal to a real number, x. The distribution function, $F_X(x)$, has the following properties:

1. $0 \leq F_X(x) \leq 1$

2. $F_X(x_1) \leq F_X(x_2)$ if $x_1 \leq x_2$

3. $F_X(-\infty) = 0$

4. $F_X(+\infty) = 1$

Another useful function relating to the random variable, X, is the *probability density function* (pdf), denoted $p_X(x)$, where

$$p_X(x) = \frac{dF_X(x)}{dx} \tag{1.25}$$

As in the case of the distribution function, the pdf is a function of a real number, x. The name "density function" arises from the fact that the probability of the event $x_1 \leq X \leq x_2$ equals

$$\begin{aligned} P(x_1 \leq X \leq x_2) &= P(X \leq x_2) - P(X \leq x_1) \\ &= F_X(x_2) - F_X(x_1) \\ &= \int_{x_1}^{x_2} p_X(x)\, dx \end{aligned}$$

The probability density function has the following properties

1. $p_X(x) \geq 0$

2. $\int_{-\infty}^{\infty} p_X(x)\, dx = F_X(+\infty) - F_X(-\infty) = 1$

Thus, a probability density function is always a nonnegative function with a total area of one. Throughout the book we use the designation, $p_X(x)$, for the probability density function of a *continuous* random variable. For ease of notation, we will often omit the subscript, X, and write simply, $p(x)$. We will use the designation $P(X = x_i)$ for the probability of a random variable, X, where X can take on *discrete* values only.

1.5.1.1 Ensemble Averages

The *mean value*, m_X, or *expected value* of a random variable, X, is defined by

$$m_X = \mathrm{E}\{X\} = \int_{-\infty}^{\infty} x p_X(x)\, dx \tag{1.26}$$

where $\mathrm{E}\{\cdot\}$ is called the *expected value operator*. The *nth moment* of a probability distribution of a random variable, X, is defined by

$$\mathrm{E}\{X^n\} = \int_{-\infty}^{\infty} x^n p_X(x)\, dx \tag{1.27}$$

For the purposes of communication system analysis, the most important moments of X are the first two moments. Thus, $n = 1$ in Equation (1.27) gives m_X as discussed above, whereas $n = 2$ gives the mean-square value of X, as follows:

$$\mathrm{E}\{X^2\} = \int_{-\infty}^{\infty} x^2 p_X(x)\, dx \tag{1.28}$$

We can also define *central moments*, which are the moments of the difference between X and m_X. The second central moment, called the *variance* of X, is defined as

$$\mathrm{var}\,(X) = \mathrm{E}\{(X - m_X)^2\} = \int_{-\infty}^{\infty} (x - m_X)^2 p_X(x)\, dx \tag{1.29}$$

The variance of X is also denoted as σ_X^2, and its square root, σ_X, is called the *standard deviation* of X. Variance is a measure of the "randomness" of the random variable X. By specifying the variance of a random variable, we are constraining the width of its probability density function. The variance and the mean-square value are related by

$$\begin{aligned}\sigma_X^2 &= \mathbf{E}\{X^2 - 2m_X X + m_X^2\} \\ &= \mathbf{E}\{X^2\} - 2m_X \mathbf{E}\{X\} + m_X^2 \\ &= \mathbf{E}\{X^2\} - m_X^2\end{aligned}$$

Thus, the variance is equal to the difference between the mean-square value and the square of the mean.

1.5.2 Random Processes

A random process, $X(A, t)$, can be viewed as a function of two variables, *an event A*, and *time*. Figure 1.5 illustrates a random process. In the figure there are *N sample functions* of time, $\{X_j(t)\}$. Each of the sample functions can be regarded as the output of a different noise generator. For a specific event A_j, we have a

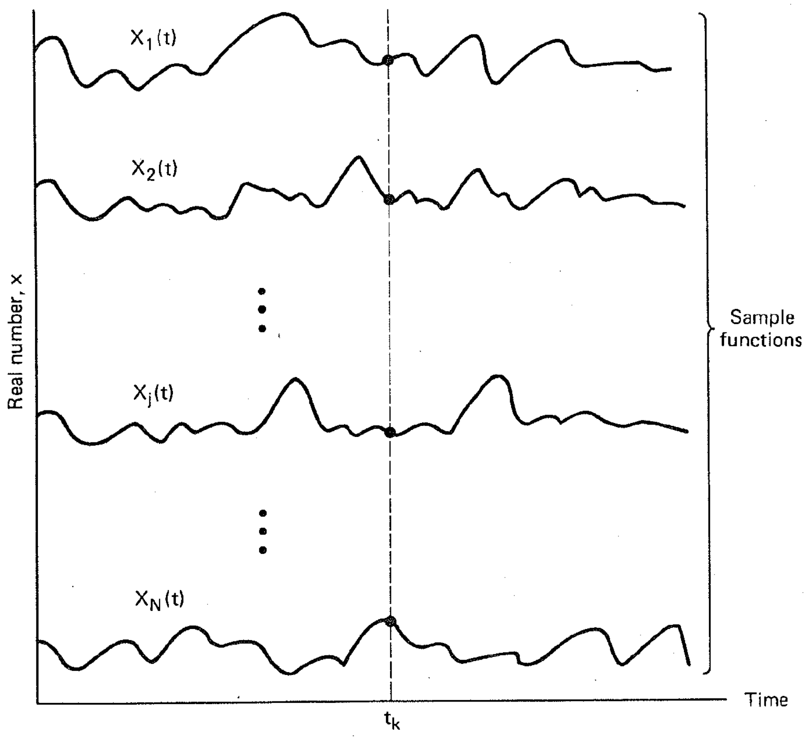


Figure 1.5 Random noise process.

single time function, $X(A_j, t) = X_j(t)$ (i.e., a sample function). The totality of all sample functions is called an *ensemble*. For a specific time t_k, $X(A, t_k)$ is a *random variable* $X(t_k)$, whose value depends on the event. Finally, for a specific event, $A = A_j$ and a specific time $t = t_k$, $X(A_j, t_k)$ is simply a *number*. For notational convenience we shall designate the random process by $X(t)$, and let the functional dependence upon A be implicit.

1.5.2.1 Statistical Averages of a Random Process

Because the value of a random process at any future time is unknown (since the identity of the event A is unknown), a random process whose distribution functions are continuous can be described statistically with a probability density function (pdf). In general the form of the pdf of a random process will be different for different times. In most situations it is not practical to determine empirically the probability distribution of a random process. However, a partial description consisting of the mean and autocorrelation function are often adequate for the needs of communication systems. We define the mean of the random process, $X(t)$, as

$$\mathbf{E}\{X(t_k)\} = \int_{-\infty}^{\infty} x p_{X_k}(x)\, dx = m_X(t_k) \tag{1.30}$$

where $X(t_k)$ is the random variable obtained by observing the random process at time t_k, and the pdf of $X(t_k)$, the density over the ensemble of events at time t_k, is designated $p_{X_k}(x)$.

We define the autocorrelation function of the random process, $X(t)$, to be a function of two variables, t_1 and t_2, as shown by

$$R_X(t_1, t_2) = \mathbf{E}\{X(t_1)X(t_2)\} \tag{1.31}$$

where $X(t_1)$ and $X(t_2)$ are random variables obtained by observing $X(t)$ at times t_1 and t_2. respectively. The autocorrelation function is a measure of the degree to which two time samples of the same random process are related.

1.5.2.2 Stationarity

A random process $X(t)$ is said to be *stationary* in the *strict sense* if none of its statistics are affected by a shift in the time origin. A random process is said to be *wide-sense stationary* (WSS) if two of its statistics, its mean and autocorrelation function, do not vary with a shift in the time origin. Thus, a process is WSS if

$$\mathbf{E}\{X(t)\} = m_X = \text{a constant} \tag{1.32}$$

and

$$R_X(t_1, t_2) = R_X(t_1 - t_2) \tag{1.33}$$

Strict-sense stationary implies wide-sense stationary, but not vice versa. Most of the useful results in communication theory are predicated on random information signals and noise being wide-sense stationary. From a practical point of view it is not necessary for a random process to be stationary for all time, but only for some observation interval of interest.

For stationary processes, the autocorrelation function in Equation (1.33) does not depend on time but only on the difference between t_1 and t_2. That is, all pairs of values of $X(t)$ at points in time separated by $\tau = t_1 - t_2$ have the same correlation value. Thus, for stationary systems, we can denote $R_X(t_1, t_2)$ simply as $R_X(\tau)$.

1.5.2.3 Autocorrelation of a Wide-Sense Stationary Random Process

Just as the variance provides a measure of randomness for random variables, the autocorrelation function provides a similar measure for random processes. For a wide-sense stationary process, the autocorrelation function is only a function of the *time difference* $\tau = t_1 - t_2$, that is,

$$R_X(\tau) = \mathrm{E}\{X(t)X(t + \tau)\} \qquad \text{for } -\infty < \tau < \infty \tag{1.34}$$

For a zero mean WSS processes, $R_X(\tau)$ indicates the extent to which the random values of the process separated by τ seconds in time are statistically correlated. In other words, $R_X(\tau)$ gives us an idea of the frequency response that is associated with a random process. If $R_X(\tau)$ changes slowly as τ increases from zero to some value, it indicates that, on the average, sample values of $X(t)$ taken at $t = t_1$ and $t = t_1 + \tau$ are nearly the same. Thus, we would expect a frequency domain representation of $X(t)$ to contain a preponderance of low frequencies. On the other hand if $R_X(\tau)$ decreases rapidly as τ is increased, we would expect $X(t)$ to change rapidly with time and thereby contain mostly high frequencies.

Properties of the autocorrelation function of a real-valued wide-sense stationary process are:

1. $R_X(\tau) = R_X(-\tau)$	symmetrical in τ about zero
2. $R_X(\tau) \le R_X(0)$ for all τ	maximum value occurs at the origin
3. $R_X(\tau) \leftrightarrow G_X(f)$	autocorrelation and power spectral density form a Fourier transform pair
4. $R_X(0) = \mathrm{E}\{X^2(t)\}$	value at the origin is equal to the average power of the signal

1.5.3 Time Averaging and Ergodicity

To compute m_X and $R_X(\tau)$ by ensemble averaging, we would have to average across all the sample functions of the process and would need to have complete knowledge of the first- and second-order joint probability density functions. Such knowledge is generally not available.

When a random process belongs to a special class, known as an *ergodic process,* its time averages equal its ensemble averages, and the statistical properties of the process can be determined by *time averaging over a single sample function* of the process. For a random process to be ergodic it must be stationary in the strict sense. (The converse is not necessary.) However, for communication systems, where we are satisfied to meet the conditions of wide-sense stationarity, we are interested only in the mean and autocorrelation functions.

We can say that a random process is *ergodic in the mean* if

$$m_X = \lim_{T\to\infty} 1/T \int_{-T/2}^{T/2} X(t)\, dt \tag{1.35}$$

and it is *ergodic in the autocorrelation function* if

$$R_X(\tau) = \lim_{T\to\infty} 1/T \int_{-T/2}^{T/2} X(t)X(t+\tau)\, dt \tag{1.36}$$

Testing for the ergodicity of a random process is usually very difficult. In practice one makes an intuitive judgment as to whether it is reasonable to interchange the time and ensemble averages. A reasonable assumption in the analysis of most communication signals (in the absence of transient effects) is that the random waveforms are ergodic in the mean and the autocorrelation function. Since time averages equal ensemble averages for ergodic processes, fundamental electrical engineering parameters, such as dc value, rms value, and average power can be related to the moments of an ergodic random process. A summary of these relationships is:

What happens with transients

1. The quantity $m_X = \mathbf{E}\{X(t)\}$ is equal to the dc level of the signal.
2. The quantity m_X^2 is equal to the normalized power in the dc component.
3. The second moment of $X(t)$, $\mathbf{E}\{X^2(t)\}$, is equal to the total average normalized power.
4. The quantity $\sqrt{\mathbf{E}\{X^2(t)\}}$ is equal to the root-mean-square (rms) value of the voltage or current signal.
5. The variance, σ_X^2, is equal to the average normalized power in the time-varying or ac component of the signal.
6. If the process has zero mean (i.e., $m_X = m_X^2 = 0$), then $\sigma_X^2 = \mathbf{E}\{X^2\}$, and the variance is the same as the mean-square value, or the variance represents the total power in the normalized load.
7. The standard deviation, σ_X, is the rms value of the ac component of the signal.
8. If $m_X = 0$, then σ_X is the rms value of the signal.

1.5.4 Power Spectral Density of a Random Process

A random process, $X(t)$, can generally be classified as a power signal having a power spectral density (PSD); $G_X(f)$, of the form shown in Equation (1.20). $G_X(f)$ is particularly useful in communications systems, because it describes the distribution of a signal's power in the frequency domain. The PSD enables us to evaluate the signal power that will pass through a network having known frequency characteristics. We summarize the principal features of PSD functions as follows:

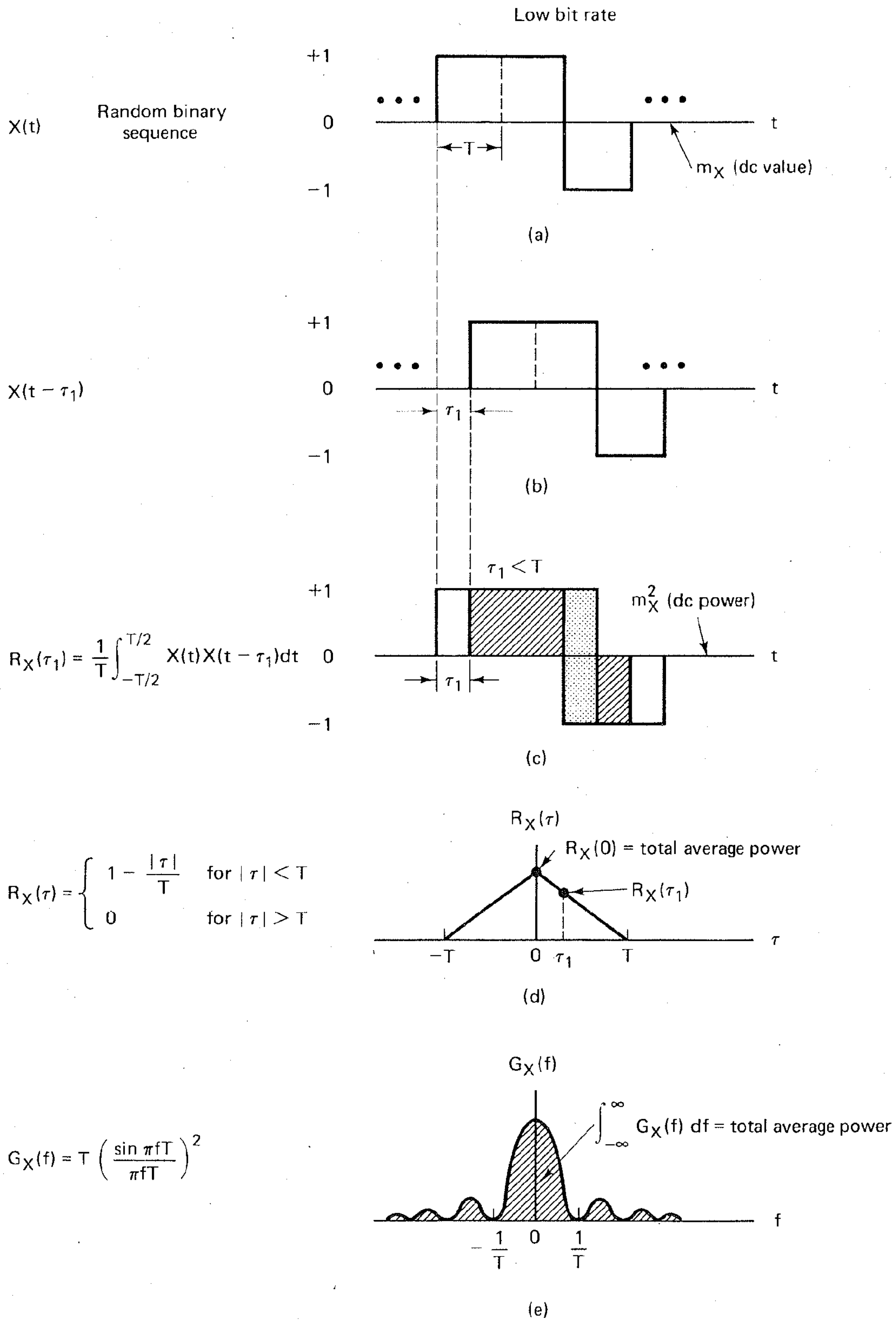
1. $G_X(f) \geq 0$	and is always real valued
2. $G_X(f) = G_X(-f)$	for $X(t)$ real-valued
3. $G_X(f) \leftrightarrow R_X(\tau)$	PSD and autocorrelation form a Fourier transform pair
4. $P_X = \int_{-\infty}^{\infty} G_X(f)\, df$	relationship between average normalized power and PSD

Figure 1.6a illustrates a single sample waveform from a WSS random process, $X(t)$. The waveform is a binary random sequence with unit-amplitude positive and negative (bipolar) pulses. The positive and negative pulses occur with equal probability. The duration of each binary digit is T seconds, and the average or dc value of the random sequence is zero. Figure 1.6b shows the same sequence displaced τ_1 seconds in time; this sequence is therefore denoted $X(t - \tau_1)$. Let us assume that $X(t)$ is ergodic in the autocorrelation function so that we can use time averaging instead of ensemble averaging to find $R_X(\tau)$. The value of $R_X(\tau_1)$ is obtained by taking the product of the two sequences $X(t)$ and $X(t - \tau_1)$ and finding the average value using Equation (1.36). Equation (1.36) is accurate for ergodic processes *only in the limit*. However, integration over an integer number of periods can provide us with an estimate of $R_X(\tau)$. Notice that $R_X(\tau_1)$ can be obtained by a positive or negative shift of $X(t)$. Figure 1.6c illustrates such a calculation, using the single sample sequence (Figure 1.6a) and its shifted replica (Figure 1.6b). The cross-hatched areas under the product curve $X(t)X(t - \tau_1)$ contribute to positive values of the product, and the dotted areas contribute to negative values. The sequences can be further shifted by $\tau_2, \tau_3, \ldots$, each shift yielding a point on the overall autocorrelation function $R_X(\tau)$ shown in Figure 1.6d. Every random bit stream has an autocorrelation plot of the general shape shown in Figure 1.6d. The plot peaks at $R_X(0)$ [the best match occurs when τ equals zero, since $R(\tau) \leq R(0)$ for all τ], and it declines as τ increases. Figure 1.6d shows points corresponding to $R_X(0)$ and $R_X(\tau_1)$.

The analytical expression for the autocorrelation function $R_X(\tau)$ shown in Figure 1.6d, is [1]

$$R_X(\tau) = \begin{cases} 1 - \dfrac{|\tau|}{T} & \text{for } |\tau| \leq T \\ 0 & \text{for } |\tau| > T \end{cases} \tag{1.37}$$

The autocorrelation function allows us to express a random signal's power spectral density directly. Since the PSD and the autocorrelation function are Fourier transforms of each other, the PSD, $G_X(f)$, of the random binary sequence can be found,



Low bit rate
X(t)
Random binary sequence
+1
0
−1
t
T
m_X (dc value)
(a)
$X(t-\tau_1)$
+1
0
−1
t
τ_1
(b)
$\tau_1 < T$
$R_X(\tau_1) = \frac{1}{T}\int_{-T/2}^{T/2} X(t)X(t-\tau_1)dt$
+1
0
−1
t
m_X^2 (dc power)
τ_1
(c)
$R_X(\tau) = \begin{cases} 1 - \frac{|\tau|}{T} & \text{for } |\tau| < T \\ 0 & \text{for } |\tau| > T \end{cases}$
$R_X(\tau)$
$R_X(0)$ = total average power
$R_X(\tau_1)$
τ
−T
0
τ_1
T
(d)
$G_X(f) = T\left(\frac{\sin \pi fT}{\pi fT}\right)^2$
$G_X(f)$
$\int_{-\infty}^{\infty} G_X(f)\, df$ = total average power
f
$-\frac{1}{T}$
0
$\frac{1}{T}$
(e)

Figure 1.6 Autocorrelation and power spectral density.

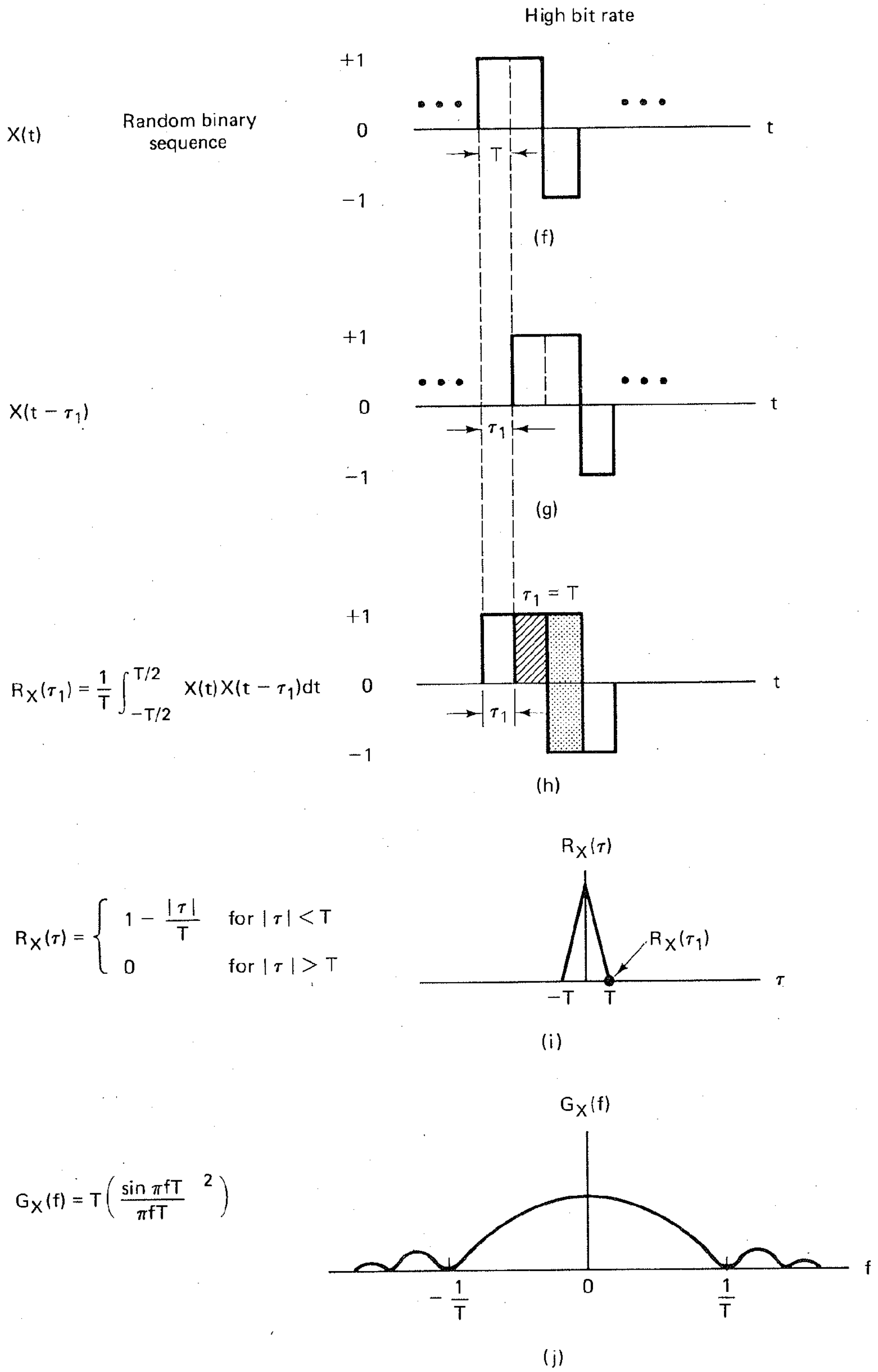

High bit rate
X(t)
Random binary sequence
+1
0
−1
t
T
(f)
$X(t-\tau_1)$
+1
0
−1
t
τ_1
(g)
$R_X(\tau_1) = \frac{1}{T}\int_{-T/2}^{T/2} X(t)X(t-\tau_1)dt$
$\tau_1 = T$
+1
0
−1
t
τ_1
(h)
$R_X(\tau) = \begin{cases} 1 - \frac{|\tau|}{T} & \text{for } |\tau| < T \\ 0 & \text{for } |\tau| > T \end{cases}$
$R_X(\tau)$
$R_X(\tau_1)$
−T
T
τ
(i)
$G_X(f) = T\left(\frac{\sin \pi fT}{\pi fT}^{2}\right)$
$G_X(f)$
$-\frac{1}{T}$
0
$\frac{1}{T}$
f
(j)

Figure 1.6 (*Continued*)

using Table A.1, as the transform of $R_X(\tau)$ in Equation (1.37). $G_X(f)$ is shown below, and its general shape is illustrated in Figure 1.6e.

$$G_X(f) = T\left(\frac{\sin \pi f T}{\pi f T}\right)^2 = T \operatorname{sinc}^2 fT \tag{1.38}$$

where

$$\operatorname{sinc} y = \frac{\sin \pi y}{\pi y} \tag{1.39}$$

Notice that the area under the PSD curve represents the average power in the signal. One convenient measure of *bandwidth* is the width of the main spectral lobe. Figure 1.6e illustrates that the bandwidth of a signal is inversely related to the symbol duration or pulse width. Figures 1.6f–j repeat the steps shown in Figures 1.6a–e, except that the bit duration is shorter. Notice that the shape of the shorter-bit-duration $R_X(\tau)$ is narrower, shown in Figure 1.6i, than it is for the longer-bit-duration $R_X(\tau)$, shown in Figure 1.6d. In Figure 1.6i, $R_X(\tau_1) = 0$; in other words, a shift of τ_1 in the case of the shorter-bit-duration example is enough to produce a zero match, or a complete decorrelation between the shifted sequences. Since the pulse duration, T, is shorter in Figure 1.6f, and the bit rate is higher than in Figure 1.6a, the bandwidth occupancy in Figure 1.6j is greater than the lower-bit-rate bandwidth occupancy shown in Figure 1.6e.

1.5.5 Noise in Communication Systems

The term *noise* refers to *unwanted* electrical signals that are always present in electrical systems. The presence of noise superimposed on a signal tends to obscure or mask the signal; it limits the receiver's ability to make correct symbol decisions, and thereby limits the rate of information transmission. Noise arises from a variety of sources, both man-made and natural. *Man-made noise* includes such sources as spark-plug ignition noise, switching transients, and other radiating electromagnetic signals. *Natural noise* includes electrical circuit and component noise, atmospheric disturbances, and galactic sources.

Good engineering design can eliminate much of the noise or its undesirable effect through filtering, shielding, the choice of modulation, and the selection of an optimum receiver site. For example, sensitive radio astronomy measurements are typically located at remote desert locations, far from man-made noise sources. However, there is one natural source of noise, called *thermal* or *Johnson noise*, that cannot be eliminated. Thermal noise [4, 5] is caused by the thermal motion of electrons in all dissipative components—resistors, wires, and so on. The same electrons that are responsible for electrical conduction are also responsible for thermal noise.

We can describe thermal noise as a zero-mean *Gaussian* random process. A Gaussian process, $n(t)$, is a random function whose value, n, at any arbitrary time, t, is statistically characterized by the Gaussian probability density function, $p(n)$:

$$p(n) = \frac{1}{\sigma\sqrt{2\pi}} \exp\left[-\frac{1}{2}\left(\frac{n}{\sigma}\right)^2\right] \tag{1.40}$$

where σ^2 is the variance of n. The *normalized* or *standardized Gaussian density function* of a zero-mean process is obtained by assuming that $\sigma = 1$. This normalized pdf is shown sketched in Figure 1.7.

We will often represent a random signal as the sum of a Gaussian noise random variable and a dc signal:

$$z = a + n$$

where z is the random signal, a the dc component, and n the Gaussian noise random variable. The pdf $p(z)$ is then expressed as

$$p(z) = \frac{1}{\sigma\sqrt{2\pi}} \exp\left[-\frac{1}{2}\left(\frac{z-a}{\sigma}\right)^2\right] \tag{1.41}$$

where, as before, σ^2 is the variance of n. The Gaussian distribution is often used as the system noise model because of a theorem, called the *central limit theorem* [3], which states that under very general conditions the probability distribution of the sum of j statistically independent random variables approaches the Gaussian distribution as $j \to \infty$, no matter what the individual distribution functions may be. Therefore, even though individual noise mechanisms might have other than

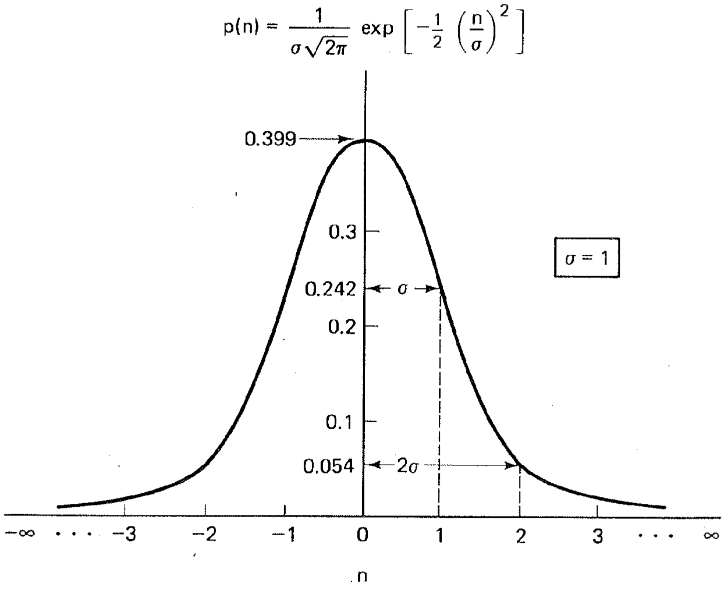


Figure 1.7 Normalized ($\sigma = 1$) Gaussian probability density function.

Gaussian distributions, the aggregate of many such mechanisms will tend toward the Gaussian distribution.

1.5.5.1 White Noise

The primary spectral characteristic of thermal noise is that its power spectral density is *the same* for all frequencies of interest in most communication systems; in other words, a thermal noise source emanates an equal amount of noise power per unit bandwidth at all frequencies—from dc to about 10^{12} Hz. Therefore, a simple model for thermal noise assumes that its power spectral density $G_n(f)$ is flat for all frequencies, as shown in Figure 1.8a, and is denoted as follows:

$$G_n(f) = \frac{N_0}{2} \quad \text{watts/hertz} \tag{1.42}$$

where the factor of 2 is included to indicate that $G_n(f)$ is a *two-sided* power spectral density. When the noise power has such a uniform spectral density, we refer to it as *white noise.* The adjective "white" is used in the sense that white light contains equal amounts of all frequencies within the visible band of electromagnetic radiation.

The autocorrelation function of white noise is given by the inverse Fourier transform of the noise power spectral density (see Table A.1) denoted as follows:

$$R_n(\tau) = \mathcal{F}^{-1}\{G_n(f)\} = \frac{N_0}{2}\,\delta(\tau) \tag{1.43}$$

Thus the autocorrelation of white noise is a delta function weighted by the factor $N_0/2$ and occurring at $\tau = 0$, as seen in Figure 1.8b. Note that $R_n(\tau)$ is zero for $\tau \neq 0$; that is, any two different samples of white noise, no matter how close together in time they are taken, are uncorrelated.

The average power, P_n, of white noise is *infinite* because its bandwidth is infinite. This can be seen by combining Equations (1.19) and (1.42) to yield.

$$P_n = \int_{-\infty}^{\infty} \frac{N_0}{2}\,df = \infty \tag{1.44}$$

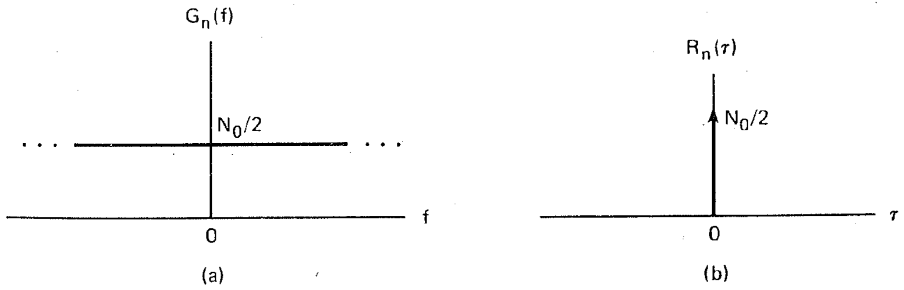


Figure 1.8 (a) Power spectral density of white noise. (b) Autocorrelation function of white noise.

Although white noise is a useful abstraction, no noise process can truly be white; however, the noise encountered in many real systems can be assumed to be approximately white. We can only observe such noise after it has passed through a real system which will have a finite bandwidth. Thus, as long as the bandwidth of the noise is appreciably larger than that of the system, the noise can be considered to have an infinite bandwidth.

The delta function in Equation (1.43) means that the noise signal, $n(t)$, is totally decorrelated from its time-shifted version, for any $\tau > 0$. Equation (1.43) indicates that *any* two different samples of a white noise process are uncorrelated. Since thermal noise is a Gaussian process and the samples are uncorrelated, the noise samples are also independent [3]. Therefore, the effect on the detection process of a channel with *additive white Gaussian noise* (AWGN) is that the noise affects each transmitted symbol *independently.* Such a channel is called a *memoryless channel.* The term "additive" means that the noise is simply superimposed or added to the signal—that there are no multiplicative mechanisms at work.

Since thermal noise is present in all communication systems and is the prominent noise source for most systems, the thermal noise characteristics—additive, white, and Gaussian—are most often used to model the noise in communication systems. Since zero-mean Gaussian noise is completely characterized by its *variance,* this model is particularly simple to use in the detection of signals and in the design of optimum receivers. In this book we shall assume, unless otherwise stated, that the system is corrupted by *additive zero-mean white Gaussian noise,* even though this is sometimes an oversimplification.

1.6 SIGNAL TRANSMISSION THROUGH LINEAR SYSTEMS

Having developed a set of models for signals and noise, we now consider the characterization of systems and their effects on such signals and noise. Since a system can be characterized equally well in the time domain or the frequency domain, techniques will be developed in both domains to analyze the response of a linear system to an arbitrary input signal. The signal, applied to the input of the system, as shown in Figure 1.9, can be described either as a time-domain signal, $x(t)$, or by its Fourier transform, $X(f)$. The use of time-domain analysis yields the time-domain output, $y(t)$, and in the process, $h(t)$, the characteristic or *impulse response* of the network, will be defined. When the input is considered in the frequency domain, we shall define a *frequency transfer function, $H(f)$,* for the system, which will determine the frequency-domain output, $Y(f)$. The system is assumed to be linear and time invariant. It is also assumed that there is no stored energy in the system at the time the input is applied.

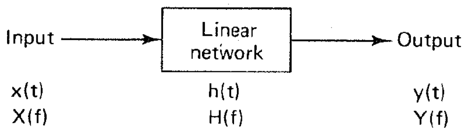


Figure 1.9 Linear system and its key parameters.

1.6.1 Impulse Response

The linear time-invariant system or network illustrated in Figure 1.9 is characterized in the time domain by an impulse response, $h(t)$, which is the response when the input is equal to a unit impulse $\delta(t)$; that is,

$$h(t) = y(t) \qquad \text{when } x(t) = \delta(t) \tag{1.45}$$

The response of the network to an arbitrary input $x(t)$ is then found by the convolution of $x(t)$ with $h(t)$, where $*$ denotes the convolution operation (see Section A.5):

$$y(t) = x(t) * h(t) = \int_{-\infty}^{\infty} x(\tau)h(t - \tau)\, d\tau \tag{1.46}$$

The system is assumed to be *causal,* which means that there can be *no* output prior to the time, $t = 0$, when the input is applied. Therefore, the lower limit of integration can be changed to zero, and we can express the output $y(t)$ as

$$y(t) = \int_{0}^{\infty} x(\tau)h(t - \tau)\, d\tau \tag{1.47}$$

Equations (1.46) and (1.47) are called the *superposition integral* or the *convolution integral.*

1.6.2 Frequency Transfer Function

The frequency-domain output signal, $Y(f)$, is obtained by taking the Fourier transform of both sides of Equation (1.46). Since convolution in the time-domain transforms to multiplication in the frequency domain (and vice versa), Equation (1.46) yields

$$Y(f) = X(f)H(f) \tag{1.48}$$

or

$$H(f) = \frac{Y(f)}{X(f)} \tag{1.49}$$

provided, of course, that $X(f) \neq 0$ for all f. Here $H(f) = \mathcal{F}\{h(t)\}$, the Fourier transform of the impulse response function, is called the *frequency transfer function* or the *frequency response* of the network. In general, $H(f)$ is complex and can be written as

$$H(f) = |H(f)|\, e^{j\theta(f)} \tag{1.50}$$

where $|H(f)|$ is the magnitude response. The phase response, $\theta(f)$, is defined as

$$\theta(f) = \tan^{-1} \frac{\text{Im}\,\{H(f)\}}{\text{Re}\,\{H(f)\}} \tag{1.51}$$

where the terms "Re" and "Im" denote "the real part of" and "the imaginary part of," respectively.

The frequency transfer function of a linear time-invariant network can easily be measured in the laboratory with a sinusoidal generator at the input of the network and an oscilloscope at the output. When the input waveform $x(t)$ is expressed as

$$x(t) = A \cos 2\pi f_0 t$$

the output of the network will be

$$y(t) = A\,|H(f_0)| \cos [2\pi f_0 t + \theta(f_0)] \tag{1.52}$$

The input frequency, f_0, is stepped through the values of interest; at each step, the amplitude and phase at the output are measured.

1.6.2.1 Random Processes and Linear Systems

If a random process forms the input to a time-invariant linear system, the output will also be a random process. That is, each sample function of the input process yields a sample function of the output process. The input power spectral density, $G_X(f)$, and the output power spectral density, $G_Y(f)$, are related as follows:

$$G_Y(f) = G_X(f)\,|H(f)|^2 \tag{1.53}$$

Equation (1.53) provides a simple way of finding the power spectral density out of a time-invariant linear system when the input is a random process.

In Chapters 2 and 3 we consider the detection of signals in Gaussian noise. We will utilize a fundamental property of a Gaussian process applied to a linear system, stated as follows: It can be shown that if a Gaussian process, $X(t)$, is applied to a time-invariant linear filter, the random process, $Y(t)$, developed at the output of the filter is also Gaussian [6].

1.6.3 Distortionless Transmission

What is required of a network for it to behave like an *ideal* transmission line? The output signal from an ideal transmission line may have some time delay compared to the input, and it may have a different amplitude than the input (just a scale change), but otherwise it must have no distortion—it must have the same shape as the input. Therefore, for ideal distortionless transmission, we can describe the output signal as

$$y(t) = Kx(t - t_0) \tag{1.54}$$

where K and t_0 are constants. Taking the Fourier transform of both sides (see Section A.3.1), we write

$$Y(f) = KX(f)e^{-j2\pi f t_0} \tag{1.55}$$

Substituting the expression (1.55) for $Y(f)$ into Equation (1.49), we see that the required system transfer function for distortionless transmission is

$$H(f) = Ke^{-j2\pi f t_0} \tag{1.56}$$

Therefore, to achieve *ideal distortionless transmission*, the overall system response must have a constant magnitude response, and its phase shift must be linear with frequency. It is not enough that the system amplify or attenuate all frequency components equally. All of the signal's frequency components must also arrive with identical time delay in order to add up correctly. Since time delay, t_0, is related to phase shift, θ, and radian frequency, $\omega = 2\pi f$, as follows,

$$t_0 \text{ (seconds)} = \frac{\theta \text{ (radians)}}{2\pi f \text{ (radians/second)}} \tag{1.57}$$

it is clear that phase shift must be proportional to frequency in order for the time delay of all components to be identical. In practice, a signal will be distorted in passing through some parts of a system. Phase or amplitude correction (*equalization*) networks may be introduced elsewhere in the system to correct for this distortion. It is the overall input–output characteristic of the system that determines its performance.

1.6.3.1 Ideal Filter

One cannot build the ideal network described in Equation (1.56). The problem is that Equation (1.56) implies an infinite bandwidth capability, where the bandwidth of a system is defined as the interval of positive frequencies over which the magnitude $|H(f)|$ remains within a specified value. In Section 1.7 various measures of bandwidth are enumerated. As an approximation to the ideal infinite-bandwidth network, let us choose a truncated network that passes, without distortion, all frequency components between f_ℓ and f_u, where f_ℓ is the lower cutoff frequency and f_u is the upper cutoff frequency, as shown in Figure 1.10. Each of these networks is called an *ideal filter*. Outside the range $f_\ell \le f \le f_u$, which is called the *passband*, the ideal filter is assumed to have a response of zero magnitude. The effective width of the passband is specified by the filter bandwidth $W_f = (f_u - f_\ell)$ hertz.

When $f_\ell \neq 0$ and $f_u \neq \infty$, the filter is called a *bandpass filter* (BPF), shown in Figure 1.10a. When $f_\ell = 0$ and f_u has a finite value, the filter is called a *low-pass filter* (LPF), shown in Figure 1.10b. When f_ℓ has a nonzero value and when $f_u \to \infty$, the filter is called a *high-pass filter* (HPF), shown in Figure 1.10c.

Following Equation (1.56), for the ideal low-pass filter transfer function with bandwidth $W_f = f_u$ hertz, shown in Figure 1.10b, we can write the transfer function as follows (letting $K = 1$):

$$H(f) = |H(f)|\, e^{-j\theta(f)} \tag{1.58}$$

where

$$|H(f)| = \begin{cases} 1 & \text{for } |f| < f_u \\ 0 & \text{for } |f| \geq f_u \end{cases} \tag{1.59}$$

and

$$e^{-j\theta(f)} = e^{-j2\pi f t_0} \tag{1.60}$$

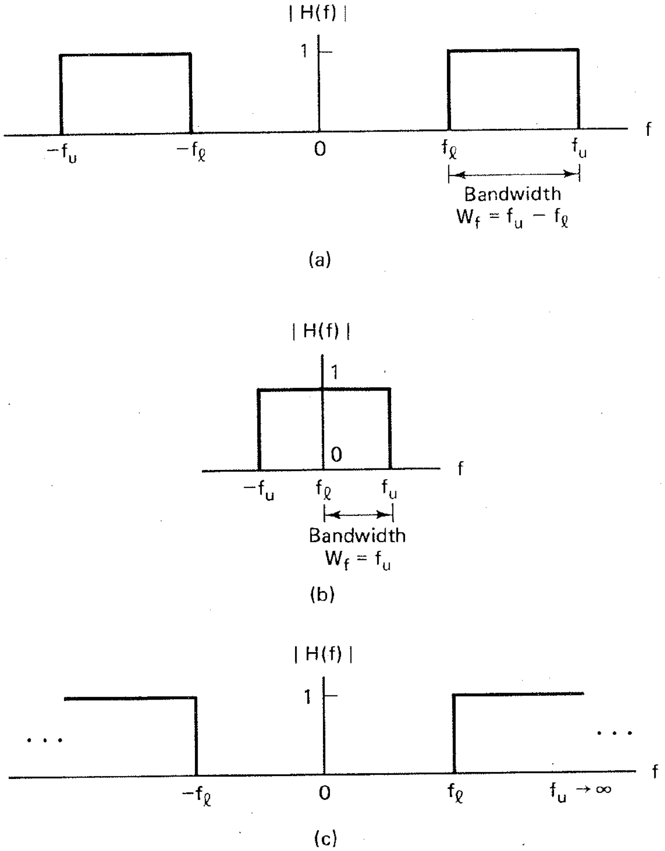


Figure 1.10 Ideal filter transfer function. (a) Ideal bandpass filter. (b) Ideal low-pass filter. (c) Ideal high-pass filter.

The impulse response $h(t)$ of the ideal low-pass filter, illustrated in Figure 1.11, is

$$h(t) = \mathcal{F}^{-1}\{H(f)\} = \int_{-\infty}^{\infty} H(f)e^{j2\pi ft}\,df \tag{1.61}$$

$$= \int_{-f_u}^{f_u} e^{-j2\pi f t_0} e^{j2\pi ft}\,df$$

$$= \int_{-f_u}^{f_u} e^{j2\pi f(t - t_0)}\,df$$

$$= 2f_u \frac{\sin 2\pi f_u(t - t_0)}{2\pi f_u(t - t_0)}$$

$$= 2f_u \operatorname{sinc} 2f_u(t - t_0) \tag{1.62}$$

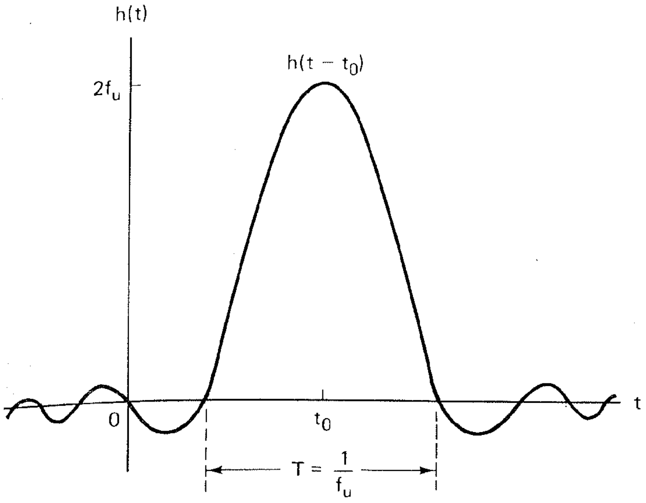


Figure 1.11 Impulse response of the ideal low-pass filter.

where sinc x is as defined in Equation (1.39). The impulse response shown in Figure 1.11 is noncausal, which means that it has a nonzero output prior to the application of an input at time $t = 0$. Therefore, it should be clear that the ideal filter described in Equation (1.58) is not realizable.

Example 1.2 Effect of an Ideal Filter on White Noise

White noise with power spectral density $G_n(f) = N_0/2$, shown in Figure 1.8a, forms the input to the ideal low-pass filter shown in Figure 1.10b. Find the power spectral density, $G_Y(f)$, and the autocorrelation function, $R_Y(\tau)$, of the output signal.

Solution

$$G_Y(f) = G_n(f)\,|H(f)|^2$$

$$= \begin{cases} \dfrac{N_0}{2} & \text{for } |f| < f_u \\ 0 & \text{otherwise} \end{cases}$$

The autocorrelation is the inverse Fourier transform of the power spectral density and is given by (see Table A.1)

$$R_Y(\tau) = N_0 f_u \frac{\sin 2\pi f_u \tau}{2\pi f_u \tau}$$

$$= N_0 f_u \operatorname{sinc} 2 f_u \tau$$

Comparing this result with Equation (1.62), we see that $R_Y(\tau)$ has the same shape as the impulse response of the ideal low-pass filter shown in Figure 1.11. In this example the ideal low-pass filter transforms the autocorrelation function of white noise (defined by the delta function) into a sinc function. After filtering, we no longer have white noise. The output noise signal will have zero correlation with shifted copies of itself, only at shifts of $\tau = n/2f_u$, where n is any integer other than zero.

1.6.3.2 Realizable Filters

The very simplest example of a realizable low-pass filter is made up of resistance ($\mathcal{R}$) and capacitance (C), as shown in Figure 1.12a; it is called an $\mathcal{R}C$ *filter,* and its transfer function can be expressed as [7]

$$H(f) = \frac{1}{1 + j2\pi f\mathcal{R}C} = \frac{1}{\sqrt{1 + (2\pi f\mathcal{R}C)^2}}\, e^{-j\theta(f)} \qquad (1.63)$$

where $\theta(f) = \tan^{-1} 2\pi f\mathcal{R}C$. The magnitude characteristic, $|H(f)|$, and the phase characteristic, $\theta(f)$ are plotted in Figures 1.12b and c, respectively. The low-pass filter bandwidth is defined to be its half-power point; this point is the frequency at which the output signal power has fallen to one-half of its peak value, or the frequency at which the magnitude of the output voltage has fallen to $1/\sqrt{2}$ of its peak value.

The half-power point is generally expressed in decibel (dB) units as the -3-dB point, or the point which is 3 dB down from the peak, where the decibel

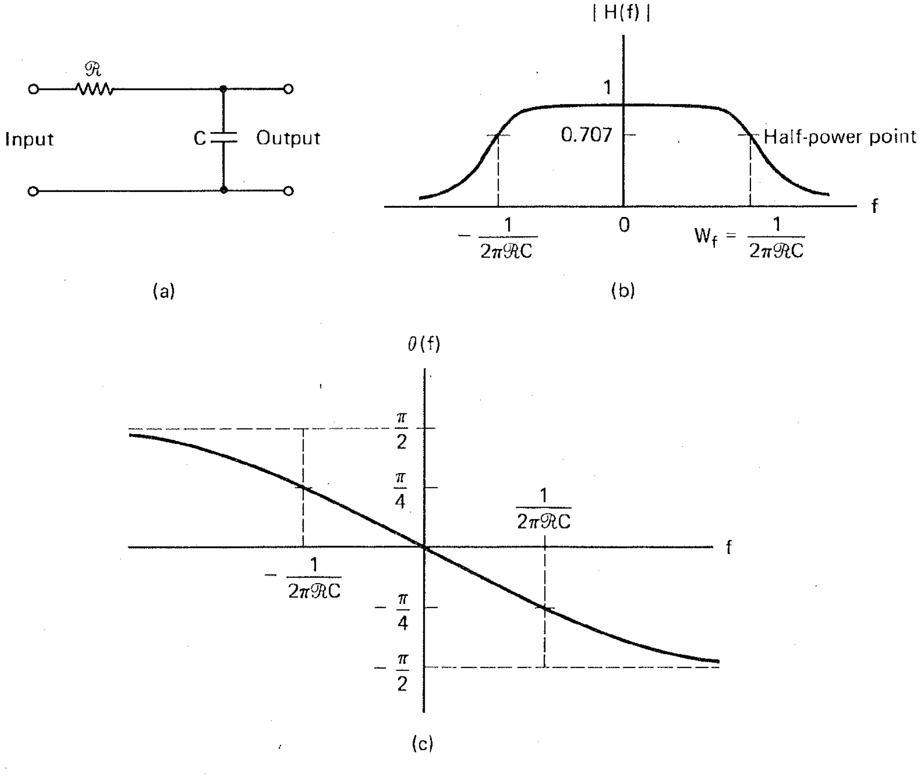


Figure 1.12 $\mathcal{R}C$ filter and its transfer function. (a) $\mathcal{R}C$ filter. (b) Magnitude characteristic of the $\mathcal{R}C$ filter. (c) Phase characteristic of the $\mathcal{R}C$ filter.

is defined as the ratio of two amounts of power, P_1 and P_2, existing at two points. By definition

$$\text{number of dB} = 10 \log_{10} \frac{P_2}{P_1} = 10 \log_{10} \frac{V_2^2/\mathcal{R}_2}{V_1^2/\mathcal{R}_1} \tag{1.64a}$$

where V_1 and V_2 are voltages and $\mathcal{R}_1$ and $\mathcal{R}_2$ are resistances. For communication systems, *normalized power* is generally used for analysis; in this case, $\mathcal{R}_1$ and $\mathcal{R}_2$ are set equal to 1 Ω, so that

$$\text{number of dB} = 10 \log_{10} \frac{P_2}{P_1} = 10 \log_{10} \frac{V_2^2}{V_1^2} \tag{1.64b}$$

The amplitude response, $|H(f)|$, can be expressed in decibels by

$$|H(f)|_{\text{dB}} = 20 \log_{10} \frac{V_2}{V_1} = 20 \log_{10} |H(f)| \tag{1.64c}$$

where V_1 and V_2 are the input and output voltages, respectively, and where the input and output resistances have been assumed equal.

From Equation (1.63) it is easy to verify that the half-power point of the low-pass $\mathcal{R}C$ filter corresponds to $\omega = 1/\mathcal{R}C$ radians per second or $f = 1/(2\pi\mathcal{R}C)$ hertz. Thus the bandwidth W_f in hertz is $1/(2\pi\mathcal{R}C)$. The filter *shape factor* is a measure of how well a realizable filter approximates the ideal filter. It is typically defined as the ratio of the filter bandwidths at the -60-dB and -6-dB amplitude response points. A sharp-cutoff bandpass filter can be made with a shape factor as low as about 2. By comparison, the shape factor of the simple $\mathcal{R}C$ low-pass filter is almost 600.

There are several useful approximations to the ideal low-pass filter characteristic. One of these, the *Butterworth filter,* approximates the ideal low-pass filter with the following function:

$$|H_n(f)| = \frac{1}{\sqrt{1 + (f/f_u)^{2n}}} \qquad n \geq 1 \tag{1.65}$$

where f_u is the upper -3-dB cutoff frequency. The magnitude function, $|H(f)|$, is sketched (single sided) for several values of n in Figure 1.13. Note that as n gets larger, the magnitude characteristics approach that of the ideal filter. Butterworth filters are popular because they are the best approximation to the ideal, in the sense of *maximal flatness* in the filter passband.

Example 1.3 Effect of an $\mathcal{R}C$ Filter on White Noise

White noise with spectral density, $G_n(f) = N_0/2$, shown in Figure 1.8a, forms the input to the $\mathcal{R}C$ filter shown in Figure 1.12a. Find the power spectral density, $G_Y(f)$, and the autocorrelation function, $R_Y(\tau)$, of the output signal.

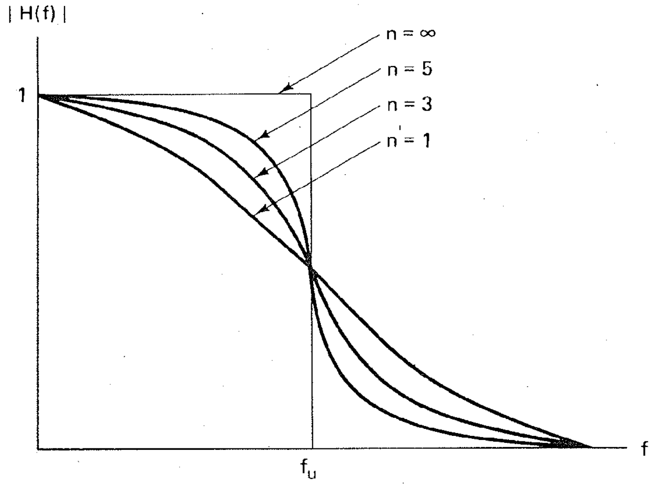


Figure 1.13 Butterworth filter magnitude response.

Solution

$$G_Y(f) = G_n(f)\,|H(f)|^2$$

$$= \frac{N_0}{2}\,\frac{1}{1 + (2\pi f \mathcal{R}C)^2}$$

$$R_Y(\tau) = \mathcal{F}^{-1}\{G_Y(f)\}$$

Using Table A.1, the inverse Fourier transform of $G_Y(f)$ is

$$R_Y(\tau) = \frac{N_0}{4\mathcal{R}C}\exp\left(-\frac{|\tau|}{\mathcal{R}C}\right)$$

As might have been predicted, we no longer have white noise after filtering. The $\mathcal{R}C$ filter transforms the input autocorrelation function of white noise (defined by the delta function) into an exponential function. For a narrowband filter (a large $\mathcal{R}C$ product), the output noise will exhibit higher correlation between noise samples of a fixed time shift than will the output noise from a wideband filter.

1.6.4 Signals, Circuits, and Spectra

Signals have been described in terms of their spectra. Similarly, networks or circuits have been described in terms of their spectral characteristics or frequency transfer functions. How is a signal's bandwidth affected as a result of the signal passing through a filter circuit? Figure 1.14 illustrates two cases of interest. In Figure 1.14a (case 1), the input signal has a narrowband spectrum, and the filter transfer function is a wideband function. From Equation (1.48) we see that the output signal spectrum is simply the product of these two spectra. In Figure 1.14a we can verify that multiplication of the two spectral functions will result in a spectrum with a bandwidth approximately equal to the smaller of the two bandwidths (when one of the two spectral functions goes to zero, the multiplication yields zero). Therefore, for case 1, the output signal spectrum is constrained by

;. The
ed by
ge $\mathcal{R}C$
les of

ks or
uency
signal
st. In
filter
at the
1.14a
t in a
band-
cation
ed by

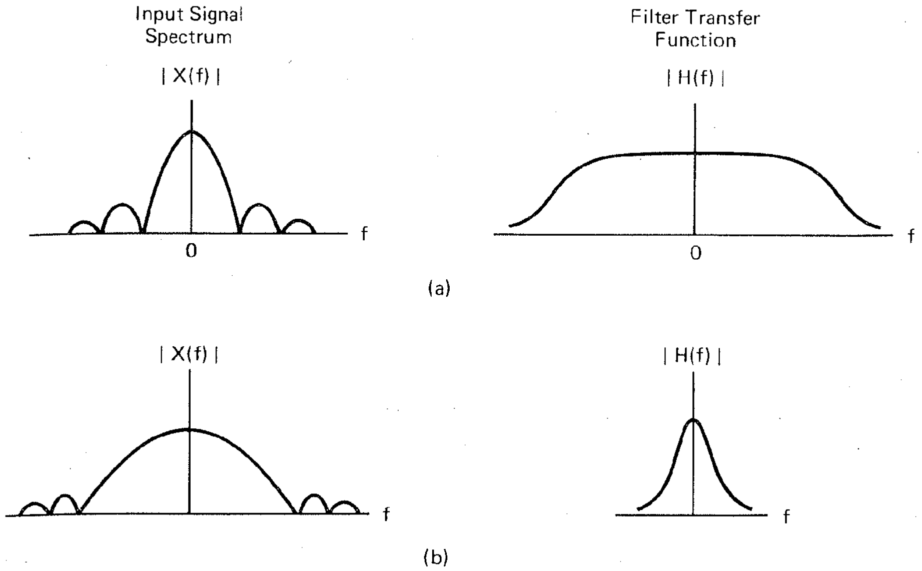


Figure 1.14 Spectral characteristics of the input signal and the circuit contribute to the spectral characteristics of the output signal. (a) Case 1: Output bandwidth is constrained by input signal bandwidth. (b) Case 2: Output bandwidth is constrained by filter bandwidth.

the input signal spectrum alone. Similarly, we see that for case 2, in Figure 1.14b, where the input signal is a wideband signal but the filter has a narrowband transfer function, the bandwidth of the output signal is constrained by the filter bandwidth; the output signal will be a filtered (distorted) rendition of the input signal.

The effect of a filter on a waveform can also be viewed in the time domain. The output, $y(t)$, resulting from convolving an ideal input pulse, $x(t)$ (having amplitude V_m and pulse width T), with the impulse response of a low-pass $\mathcal{R}C$ filter can be written as [8]

$$y(t) = \begin{cases} V_m(1 - e^{-t/\mathcal{R}C}) & \text{for } 0 \leq t \leq T \\ V'_m e^{-(t-T)/\mathcal{R}C} & \text{for } t > T \end{cases} \tag{1.66}$$

where

$$V'_m = V_m(1 - e^{-T/\mathcal{R}C}) \tag{1.67}$$

Let us define the pulse bandwidth, W_p, and the $\mathcal{R}C$ filter bandwidth, W_f, as

$$W_p = \frac{1}{T} \tag{1.68}$$

and

$$W_f = \frac{1}{2\pi\mathcal{R}C} \tag{1.69}$$

The ideal input pulse, $x(t)$, and its magnitude spectrum $|X(f)|$, are shown in Figure 1.15. The $\mathcal{RC}$ filter and its magnitude characteristic, $|H(f)|$, are shown in Figures 1.12a and b, respectively. Following Equations (1.66) to (1.69), three cases are illustrated in Figure 1.16. Example 1 illustrates the case where $W_p \ll W_f$. Notice that the output response, $y(t)$, is a reasonably good approximation of the input pulse, $x(t)$, shown in dashed lines. This represents an example of *good fidelity*. In example 2, where $W_p \simeq W_f$, we can still recognize that a pulse had been transmitted from the output, $y(t)$. Finally, example 3 illustrates the case where $W_p \gg W_f$. Here the presence of the pulse is barely perceptible from the output, $y(t)$. Can you think of an application where the large filter bandwidth or good fidelity of example 1 is called for? A *precise ranging application,* perhaps, where the pulse time of arrival translates into distance, necessitates a pulse with a steep rise time. Which example characterizes the binary digital communications application? *It is example 2.* As we pointed out earlier regarding Figure 1.1, one of the principal features of binary digital communications is that each received pulse

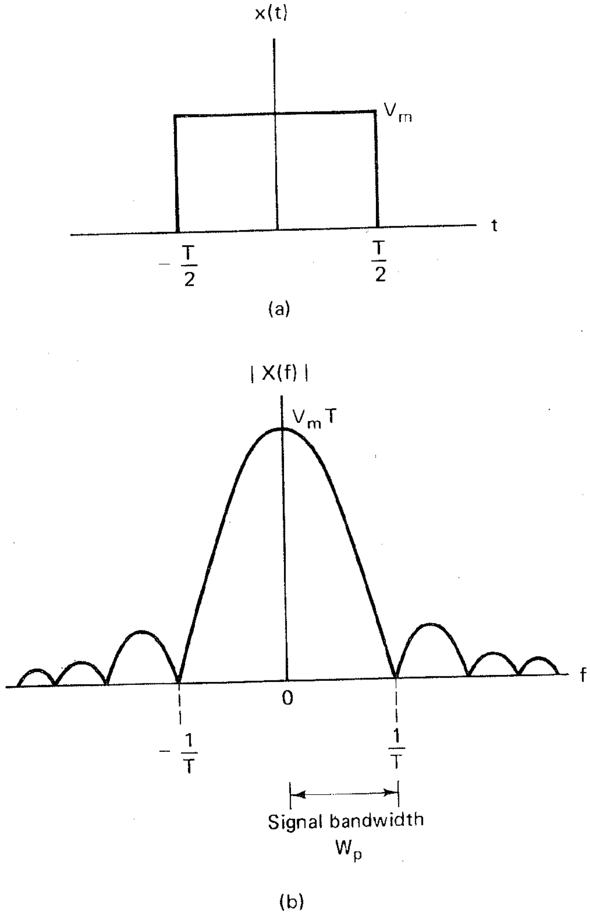


Figure 1.15 (a) Ideal pulse. (b) Magnitude spectrum of the ideal pulse.

ˀigure
igures
ɜs are
Notice
input
ɟelity.
been
where
utput,
good
where
steep
appli-
ɔne of
pulse

l pulse.

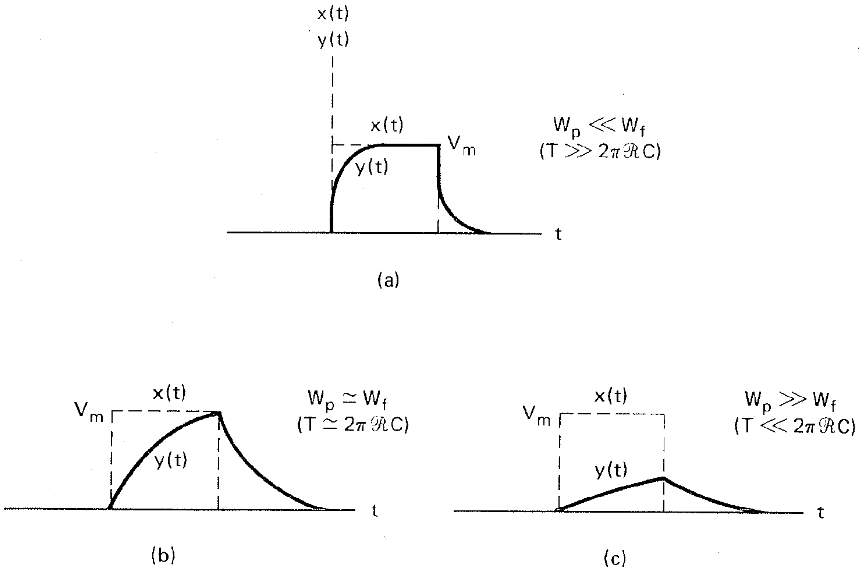


Figure 1.16 Three examples of filtering an ideal pulse. (a) Example 1: Good-fidelity output. (b) Example 2: Good-recognition output. (c) Example 3: Poor-recognition output.

need only be accurately *perceived* as being in one of its two states; a high-fidelity signal need not be maintained. Example 3 has been included for completeness; it would not be used as a design criterion for a practical system.

1.7 BANDWIDTH OF DIGITAL DATA

1.7.1 Baseband versus Bandpass

An easy way to translate the spectrum of a low-pass or baseband signal, $x(t)$, to a higher frequency is to multiply or *heterodyne* the baseband signal with a carrier wave, $\cos 2\pi f_c t$, as shown in Figure 1.17a. The resulting waveform, $x_c(t)$, is called a *double-sideband* (DSB) *modulated signal* and is expressed as

$$x_c(t) = x(t) \cos 2\pi f_c t \tag{1.70}$$

From the frequency shifting theorem (see Section A.3.2) the spectrum of the DSB signal, $x_c(t)$, is given by $X_c(f)$:

$$X_c(f) = \tfrac{1}{2}[X(f - f_c) + X(f + f_c)] \tag{1.71}$$

The magnitude spectrum $|X(f)|$ of the baseband signal, $x(t)$, having a bandwidth f_m, and the magnitude spectrum, $|X_c(f)|$, of the DSB signal, $x_c(t)$, having a bandwidth W_{DSB}, are shown in Figure 1.17b and c, respectively. In the plot of $|X_c(f)|$,

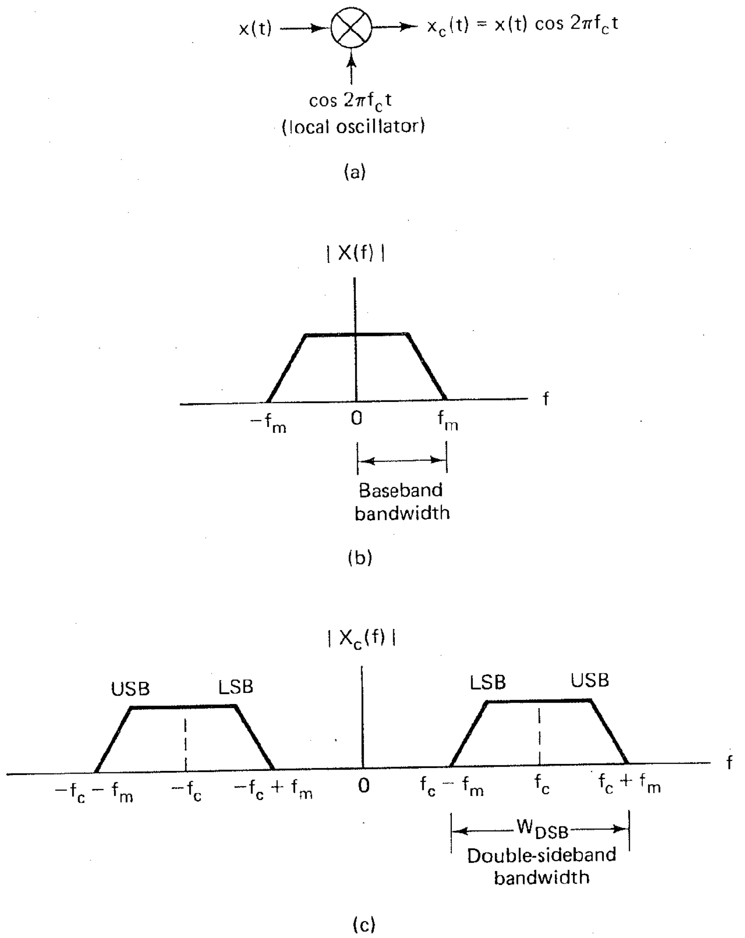


Figure 1.17 Comparison of baseband and double-sideband spectra. (a) Heterodyning. (b) Baseband spectrum. (c) Double-sideband spectrum.

spectral components corresponding to positive baseband frequencies, appear in the range f_c to $(f_c + f_m)$. This part of the DSB spectrum is called the *upper sideband* (USB). Spectral components corresponding to negative baseband frequencies appear in the range $(f_c - f_m)$ to f_c. This part of the DSB spectrum is called the *lower sideband* (LSB). Mirror images of the USB and LSB spectra appear in the negative-frequency half of the plot. The *carrier wave* is sometimes referred to as a *local oscillator* (LO) *signal*, a *mixing signal*, or a *heterodyne signal*. Generally, the carrier wave frequency is much higher than the bandwidth of the baseband signal; that is,

$$f_c \gg f_m$$

From Figure 1.17 we can readily compare the bandwidth f_m, required to transmit the baseband signal, with the bandwidth W_{DSB}, required to transmit the DSB signal; we see that

$$W_{\text{DSB}} = 2f_m \tag{1.72}$$

That is, we need twice as much transmission bandwidth to transmit a DSB version of the signal than we do to transmit its baseband counterpart.

1.7.2 The Bandwidth Dilemma

Many important theorems of communication and information theory are based on the assumption of *strictly bandlimited* channels, which means that no signal power whatever is allowed outside the defined band. We are faced with the dilemma that strictly bandlimited signals are not realizable since they imply signals with infinite duration; nonbandlimited signals, having energy at arbitrarily high frequencies, appear just as unreasonable. It is no wonder that there is no single universal definition of bandwidth.

All bandwidth criteria have in common the attempt to specify a measure of the width, W, of a nonnegative real-valued power spectral density defined for all frequencies $|f| < \infty$. Figure 1.18 illustrates some of the most common definitions of bandwidth; in general, the various criteria are not interchangeable. The single-

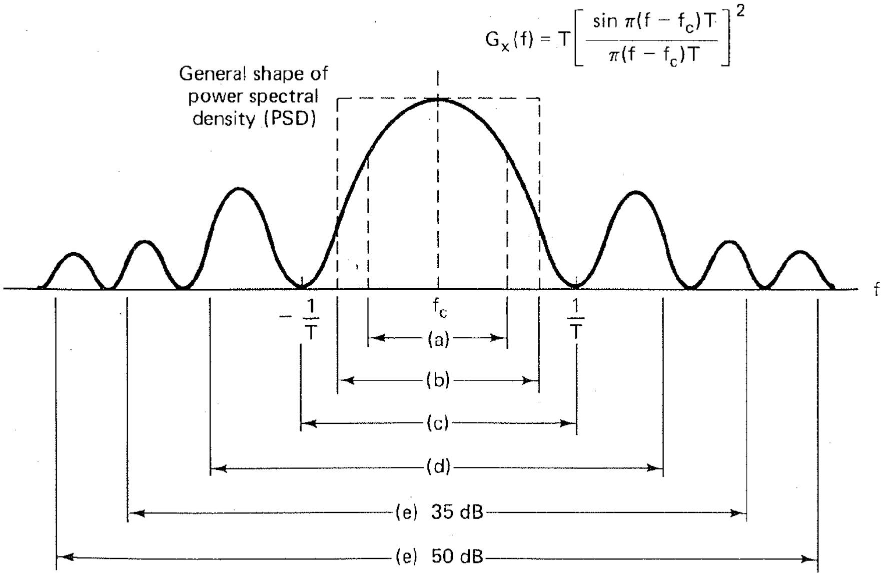

$G_x(f) = T\left[\dfrac{\sin \pi(f - f_c)T}{\pi(f - f_c)T}\right]^2$
General shape of power spectral density (PSD)
$-\frac{1}{T}$
f_c
$\frac{1}{T}$
f
(a)
(b)
(c)
(d)
(e) 35 dB
(e) 50 dB

Figure 1.18 Bandwidth of digital data. (a) Half-power. (b) Noise equivalent. (c) Null to null. (d) 99% of power. (e) Bounded PSD (defines attenuation outside bandwidth) at 35 and 50 dB.

sided power spectral density, $G_x(f)$, for a single heterodyned pulse, $x_c(t)$, takes the analytical form

$$G_x(f) = T\left[\frac{\sin \pi(f - f_c)T}{\pi(f - f_c)T}\right]^2 \tag{1.73}$$

where f_c is the carrier wave frequency and T is the pulse duration. This power spectral density, whose general appearance is sketched in Figure 1.18, also characterizes a *random pulse sequence,* assuming that the averaging time is long relative to the pulse duration. The plot consists of a main lobe and smaller symmetrical sidelobes. The general shape of the plot is valid for most digital modulation formats; some formats, however, do not have well-defined lobes. The bandwidth criteria depicted in Figure 1.18 are as follows:

(a) *Half-power bandwidth.* This is the interval between frequencies at which $G_x(f)$ has dropped to half-power, or 3 dB below the peak value.

(b) *Equivalent rectangular or noise equivalent bandwidth.* The noise equivalent bandwidth was originally conceived to permit rapid computation of output noise power from an amplifier with a wideband noise input; the concept can similarly be applied to a signal bandwidth. The noise equivalent bandwidth W_N of a signal is defined by the relationship $W_N = P_x/G_x(f_c)$, where P_x is the total signal power over all frequencies and $G_x(f_c)$ is the value of $G_x(f)$ at the band center (assumed to be the maximum value over all frequencies).

(c) *Null-to-null bandwidth.* The most popular measure of bandwidth for digital communications is the width of the main spectral lobe, where most of the signal power is contained. This criterion lacks complete generality since some modulation formats lack well-defined lobes.

(d) *Fractional power containment bandwidth.* This bandwidth criterion has been adopted by the Federal Communications Commission (FCC Rules and Regulations Section 2.202) and states that the occupied bandwidth is the band that leaves exactly 0.5% of the signal power above the upper band limit and exactly 0.5% of the signal power below the lower band limit. Thus 99% of the signal power is inside the occupied band.

(e) *Bounded power spectral density.* A popular method of specifying bandwidth is to state that everywhere outside the specified band, $G_x(f)$ must have fallen at least to a certain stated level below that found at the band center. Typical attenuation levels might be 35 or 50 dB.

(f) *Absolute bandwidth.* This is the interval between frequencies, outside of which the spectrum is zero. This is a useful abstraction. However, for all realizable waveforms, the absolute bandwidth is infinite.

Example 1.4 Strictly Bandlimited Signals

The concept of a signal that is strictly limited to a band of frequencies is not realizable. Prove this by showing that a *strictly bandlimited* signal must also be a signal of *infinite time duration.*

takes

(1.73)

›ower
char-
ȝ rela-
:trical
n for-
width

which

ʋalent
›utput
ɔt can
width
: P_x is
$G_x(f)$
cies).
ligital
ɔf the
since

ȝ been
l Reg-
band
it and
9% of

width
fallen
ypical

ide of
for all

izable.
nfinite

Solution

Let $x(t)$ be a signal, with Fourier transform $X(f)$, that is strictly limited to the band of frequencies centered at $\pm f_c$ and of width $2W$. We may express $X(f)$ in terms of an ideal filter transfer function, $H(f)$, illustrated in Figure 1.19a, as follows:

$$X(f) = X'(f)H(f) \tag{1.74}$$

where, $X'(f)$ is the Fourier transform of a signal $x'(t)$, not necessarily bandlimited, where

$$H(f) = \text{rect}\left(\frac{f - f_c}{2W}\right) + \text{rect}\left(\frac{f + f_c}{2W}\right) \tag{1.75}$$

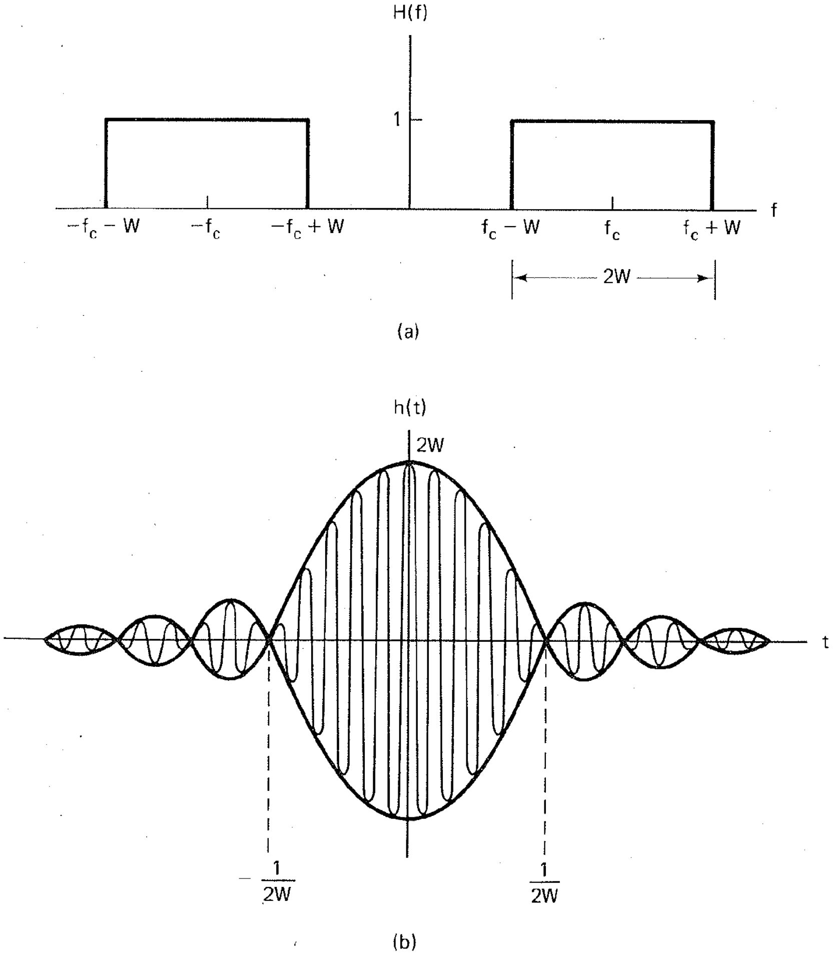


Figure 1.19 Transfer function and impulse response for a strictly bandlimited signal. (a) Ideal bandpass filter. (b) Ideal bandpass impulse response.

and where

$$\text{rect}\left(\frac{f}{2W}\right) = \begin{cases} 1 & \text{for } -W < f < W \\ 0 & \text{for } |f| > W \end{cases}$$

We can express $X(f)$ in terms of $X'(f)$ as

$$X(f) = \begin{cases} X'(f) & \text{for } (f_c - W) \leq |f_c| \leq (f_c + W) \\ 0 & \text{otherwise} \end{cases}$$

Multiplication in the frequency domain, as seen in Equation (1.74), transforms to convolution in the time domain as follows:

$$x(t) = x'(t) * h(t) \tag{1.76}$$

where $h(t)$, the inverse Fourier transform of $H(f)$, can be written as (see Tables A.1 and A.2)

$$h(t) = 2W\,(\text{sinc } 2Wt) \cos 2\pi f_c t$$

and is illustrated in Figure 1.19b. We note that $h(t)$ is of *infinite duration*. It follows, therefore, that $x(t)$ obtained in Equation (1.76) by convolving $x'(t)$ with $h(t)$ is also of infinite duration and therefore is *not realizable*.

1.8 CONCLUSION

In this chapter, the goals of the book have been outlined and the basic nomenclature has been defined. The fundamental concepts of time-varying signals, such as classification, spectral density, and autocorrelation, have been reviewed. Also, random signals have been considered, and white Gaussian noise, the primary noise model in most communication systems, has been characterized, statistically and spectrally. Finally, we have treated the important area of signal transmission through linear systems and have examined some of the realizable approximations to the ideal case. We have also established that the concept of an absolute bandwidth is an abstraction, and that in the real world we are faced with the need to choose a definition of bandwidth that is useful for our particular application. In the remainder of the book, each of the signal processing steps introduced in this chapter will be explored in the context of the typical system block diagram appearing at the beginning of each chapter.

REFERENCES

1. Haykin, S., *Communication Systems,* John Wiley & Sons, Inc., New York, 1983.
2. Shanmugam, K. S., *Digital and Analog Communication Systems,* John Wiley & Sons, Inc., New York, 1979.
3. Papoulis, A., *Probability, Random Variables, and Stochastic Processes,* McGraw-Hill Book Company, New York, 1965.
4. Johnson, J. B., "Thermal Agitation of Electricity in Conductors," *Phys. Rev.*, vol. 32, July 1928, pp. 97–109.

5. Nyquist, H., "Thermal Agitation of Electric Charge in Conductors," *Phys. Rev.*, vol. 32, July 1928, pp. 110–113.

6. Van Trees, H. L., *Detection, Estimation, and Modulation Theory,* Part 1, John Wiley & Sons, New York, 1968.

7. Schwartz, M., *Information Transmission, Modulation, and Noise,* McGraw-Hill Book Company, New York, 1970.

8. Millman, J., and Taub, H., *Pulse, Digital, and Switching Waveforms,* McGraw-Hill Book Company, New York, 1965.

PROBLEMS

1.1. Classify the following signals as energy signals or power signals. Find the normalized energy or normalized power of each.

(a) $x(t) = A \cos 2\pi f_0 t \qquad \text{for } -\infty < t < \infty$

(b) $x(t) = \begin{cases} A \cos 2\pi f_0 t & \text{for } -T_0/2 \le t \le T_0/2, \text{ where } T_0 = 1/f_0 \\ 0 & \text{elsewhere} \end{cases}$

(c) $x(t) = \begin{cases} A \exp(-at) & \text{for } t > 0,\ a > 0 \\ 0 & \text{elsewhere} \end{cases}$

(d) $x(t) = \cos t + 5 \cos 2t \qquad \text{for } -\infty < t < \infty$

1.2. Determine the energy spectral density of a square pulse $x(t) = \text{rect}\,(t/T)$, where $\text{rect}\,(t/T)$ equals 1, for $-T/2 \le t \le T/2$, and equals 0, elsewhere. Calculate the normalized energy E_x in the pulse.

1.3. Find an expression for the average normalized power in a periodic signal in terms of its complex Fourier series coefficients.

1.4. Using time averaging, find the average normalized power in the waveform $x(t) = 10 \cos 10t + 20 \cos 20t$.

1.5. Repeat Problem 1.4 using the summation of spectral coefficients.

1.6. Determine which, if any, of the following functions have the properties of autocorrelation functions. Justify your determination. [*Note:* $\mathcal{F}\{R(\tau)\}$ must be a nonnegative function. Why?]

(a) $x(\tau) = \begin{cases} 1 & \text{for } -1 \le \tau \le 1 \\ 0 & \text{otherwise} \end{cases}$

(b) $x(\tau) = \delta(\tau) + \sin 2\pi f_0 \tau$

(c) $x(\tau) = \exp(|\tau|)$

(d) $x(\tau) = 1 - |\tau| \qquad \text{for } -1 \le \tau \le 1$

1.7. Determine which, if any, of the following functions have the properties of power spectral density functions. Justify your determination.

(a) $X(f) = \delta(f) + \cos^2 2\pi f$

(b) $X(f) = 10 + \delta(f - 10)$

(c) $X(f) = \exp(-2\pi|f - 10|)$

(d) $X(f) = \exp[-2\pi(f^2 - 10)]$

1.8. Find the autocorrelation function of $x(t) = A\cos(2\pi f_0 t + \phi)$ in terms of its period, $T_0 = 1/f_0$. Find the average normalized power of $x(t)$, using $P_x = R(0)$.

1.9. (a) Use the results of Problem 1.8 to find the autocorrelation function, $R(\tau)$, of waveform $x(t) = 10\cos 10t + 20\cos 20t$.

(b) Use the relationship $P_x = R(0)$ to find the average normalized power in $x(t)$. Compare the answer with the answers to Problems 1.4 and 1.5.

1.10. For the function $x(t) = 1 + \cos 2\pi f_0 t$, calculate (a) the average value of $x(t)$; (b) the ac power of $x(t)$; (c) the rms value of $x(t)$.

1.11. Consider a random process given by $X(t) = A\cos(2\pi f_0 t + \phi)$, where A and f_0 are constants and ϕ is a random variable that is uniformly distributed over $(0, 2\pi)$. If $X(t)$ is an ergodic process, the time averages of $X(t)$ in the limit as $t \to \infty$ are equal to the corresponding ensemble averages of $X(t)$.

(a) Use time averaging over an integer number of periods to calculate the approximations to the first and second moments of $X(t)$.

(b) Use Equations (1.26) and (1.28) to calculate the ensemble-average approximations to the first and second moments of $X(t)$. Compare the results with your answers in part (a).

1.12. The Fourier transform of a signal, $x(t)$ is defined by $X(f) = \text{sinc } f$, where the sinc function is as defined in Equation (1.39). Find the autocorrelation function, $R_x(\tau)$, of the signal $x(t)$.

1.13. Use the sampling property of the unit impulse function to evaluate the following integrals.

(a) $\displaystyle\int_{-\infty}^{\infty} \cos 6t\delta(t - 3)\, dt$

(b) $\displaystyle\int_{-\infty}^{\infty} 10\delta(t)(1 + t)^{-1}\, dt$

(c) $\displaystyle\int_{-\infty}^{\infty} \delta(t + 4)(t^2 + 6t + 1)\, dt$

(d) $\displaystyle\int_{-\infty}^{\infty} \exp(-t^2)\delta(t - 2)\, dt$

1.14. Find $X_1(f) * X_2(f)$ for the spectra shown in Figure P1.1.

1.15. The two-sided power spectral density, $G_x(f) = 10^{-6}f^2$, of a waveform $x(t)$ is shown in Figure P1.2.

(a) Find the normalized average power in $x(t)$ over the frequency band from 0 to 10 kHz.

(b) Find the normalized average power contained in the frequency band from 5 to 6 kHz.

1.16. Decibels are logarithmic measures of *power ratios,* as described in Equation (1.64a). Sometimes, a similar formulation is used to express nonpower measurements in decibels (referenced to some designated unit). As an example, calculate how many decibels of hamburger meat you would buy to feed 2 hamburgers each to a group of 100 people. Assume that you and the butcher have agreed on the unit of "$\frac{1}{2}$ pound of meat" (the amount in one hamburger) as a reference unit.

eriod,

(τ), of

ı $x(t)$.

t); (b)

f_0 are
!π). If
equal

ɔprox-

ɔxima-
ı your

e sinc
$R_x(\tau)$,

owing

shown

n 0 to

n 5 to

1.64a).
ents in
many
group
pound

hap. 1

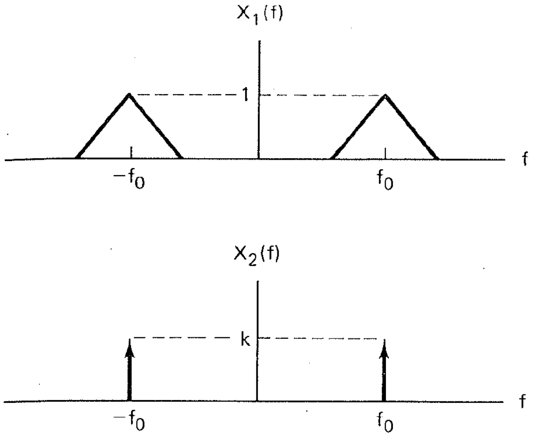


Figure P1.1

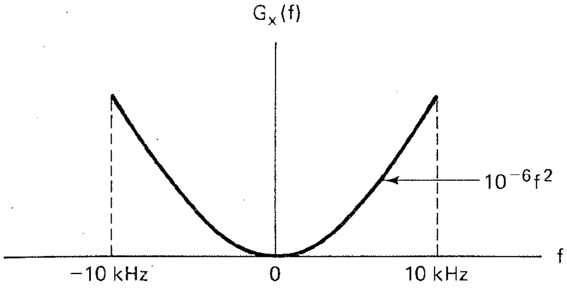


Figure P1.2

1.17. Consider the Butterworth low-pass amplitude response given in Equation (1.65).

(a) Find the value of n so that $|H(f)|^2$ is constant to within ± 1 dB over the range $|f| \leq 0.9f_u$.

(b) Show that as n approaches infinity, the amplitude response approaches that of an ideal low-pass filter.

1.18. Consider the network in Figure 1.9, whose frequency transfer function is $H(f)$. An impulse $\delta(t)$ is applied at the input. Show that the response $y(t)$ at the output is the inverse Fourier transform of $H(f)$.

1.19. An example of a *holding circuit*, commonly used in pulse systems, is shown in Figure P1.3. Determine the impulse response of this circuit.

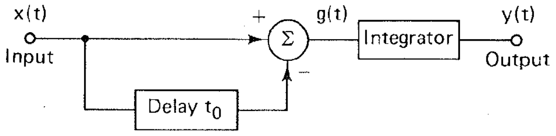


Figure P1.3

1.20. Given the spectrum

$$G_x(f) = 10^{-4}\left\{\frac{\sin[\pi(f - 10^6)10^{-4}]}{\pi(f - 10^6)10^{-4}}\right\}^2$$

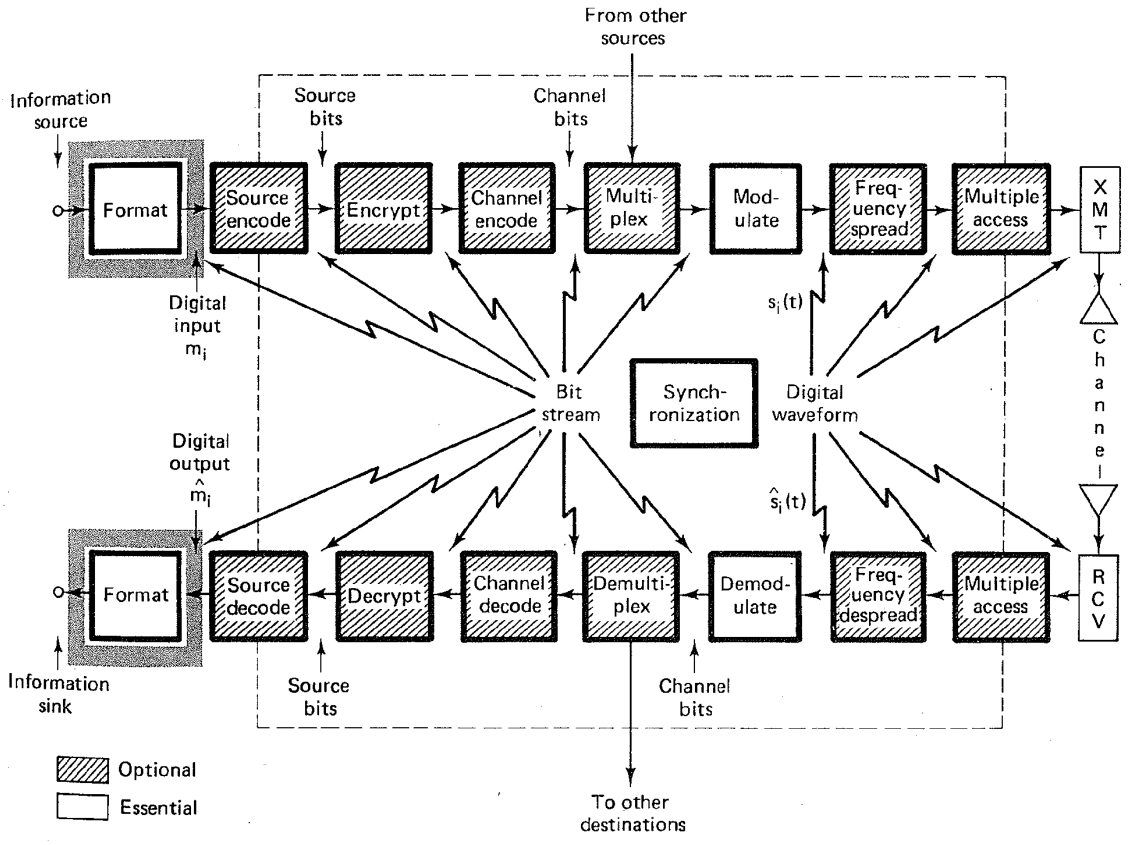
Find the value of the signal bandwidth using the following bandwidth definitions:

(a) Half-power bandwidth.
(b) Noise equivalent bandwidth.
(c) Null-to-null bandwidth.
(d) 99% of power bandwidth.
(e) Bandwidth beyond which the attenuation is 35 dB.
(f) Absolute bandwidth.

ıns:

CHAPTER 2

Formatting and Baseband Transmission



The first essential signal processing step, *formatting*, makes the source signal compatible with digital processing. *Transmit formatting* is a transformation from source information to digital symbols (in the receive chain, formatting is the reverse transformation). When there is data redundancy reduction or data compression, in addition to formatting, the process is termed *source coding*. Some authors consider formatting to be a special case of source coding. We treat formatting (and baseband transmission) in this chapter, and treat source coding as a special case of the *efficient description* of source information in Chapter 11. In Figure 2.1 the main formatting topics are highlighted—character coding, sampling, quantization, and pulse code modulation (PCM).

A signal whose spectrum extends from (or near) dc up to some finite value, usually less than a few megahertz, is called a *baseband* or *low-pass* signal. Such a signal is implied whenever we use the term "information," "message," or "data." For the transmission of baseband signals by a digital communication system, the information is *formatted* so that it is represented by digital symbols. Then, pulse waveforms are assigned that represent these symbols; this step is referred to as *pulse modulation* or *baseband modulation*. These waveforms can then be transmitted over a cable.

Baseband signals are not appropriate for propagation through many transmission media. Baseband signals whose spectrum has been shifted to a frequency band that is more appropriate for propagation through a transmission medium are called *bandpass modulation signals* or simply *bandpass signals*. Bandpass signals have their spectral content clustered in a band of frequencies near a value called

signal
n from
the re-
mpres-
uthors
natting
special
Figure
, quan-

value,
l. Such
;e," or
ication
mbols.
step is
ms can

/ trans-
quency
ium are
signals
: called

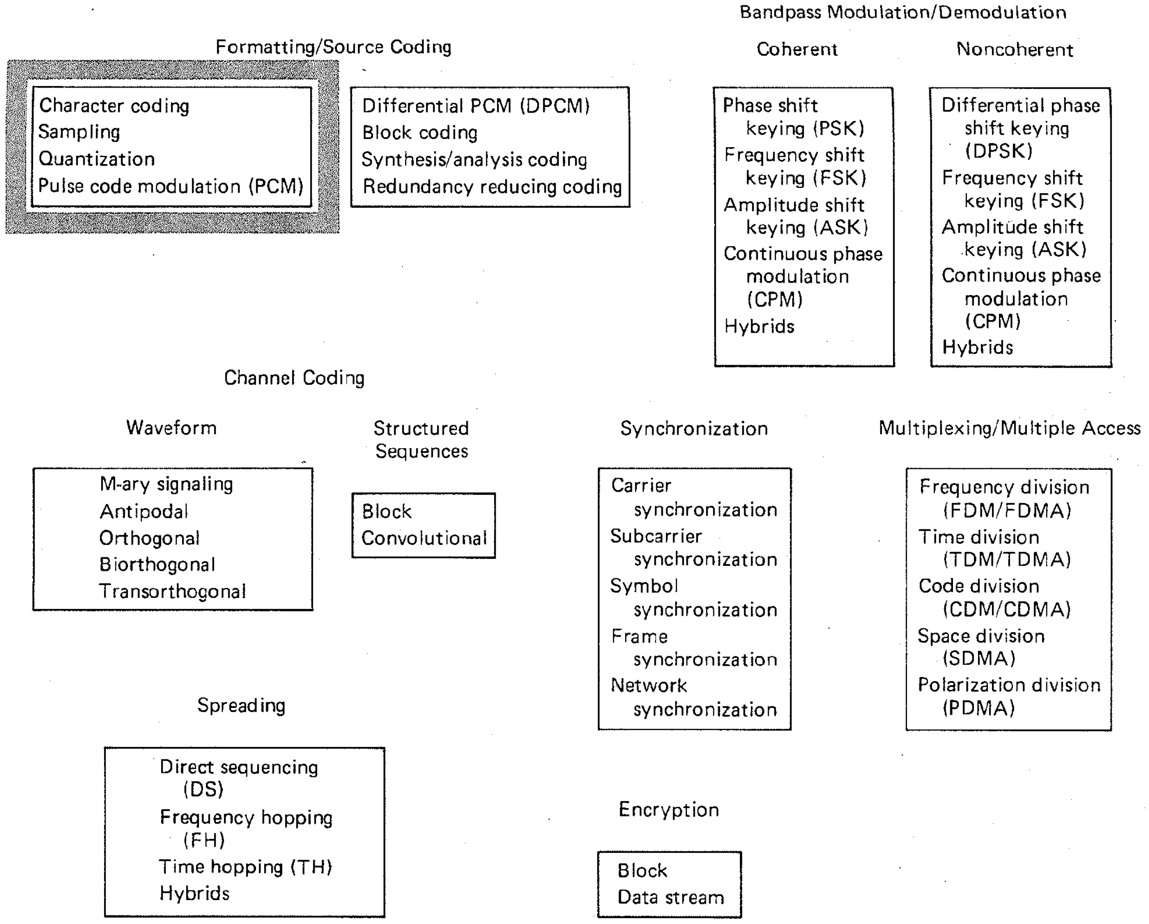


Figure 2.1 Basic digital communication transformations.

the *carrier frequency*. In Chapter 3 we deal with the modulation and demodulation of these bandpass signals.

2.1 BASEBAND SYSTEMS

In Figure 1.2 we presented a block diagram of a typical digital communication system. A version of this functional diagram, focusing primarily on the formatting and transmission of *baseband* signals, is shown in Figure 2.2. Data already in a digital format would bypass the formatting function. Textual information is transformed into binary digits by use of a coder. Analog information is formatted using three separate processes: sampling, quantization, and coding. In all cases, the formatting step results in a sequence of binary digits.

These digits are to be transmitted through a *baseband channel*, such as a pair of wires or a coaxial cable. However, no channel can be used for the transmission of binary digits without first transforming the digits to *waveforms* that are compatible with the channel. For baseband channels, compatible waveforms are pulses.

In Figure 2.2, the conversion from binary digits to pulse waveforms takes place in the block labeled *waveform encoder*, also called a *baseband modulator*. The output of the waveform encoder is typically a sequence of pulses with characteristics that correspond to the binary digits being sent. After transmission through the channel, the received waveforms are detected to produce an estimate of the transmitted digits, and then the final step, (reverse) formatting, recovers an estimate of the source information.

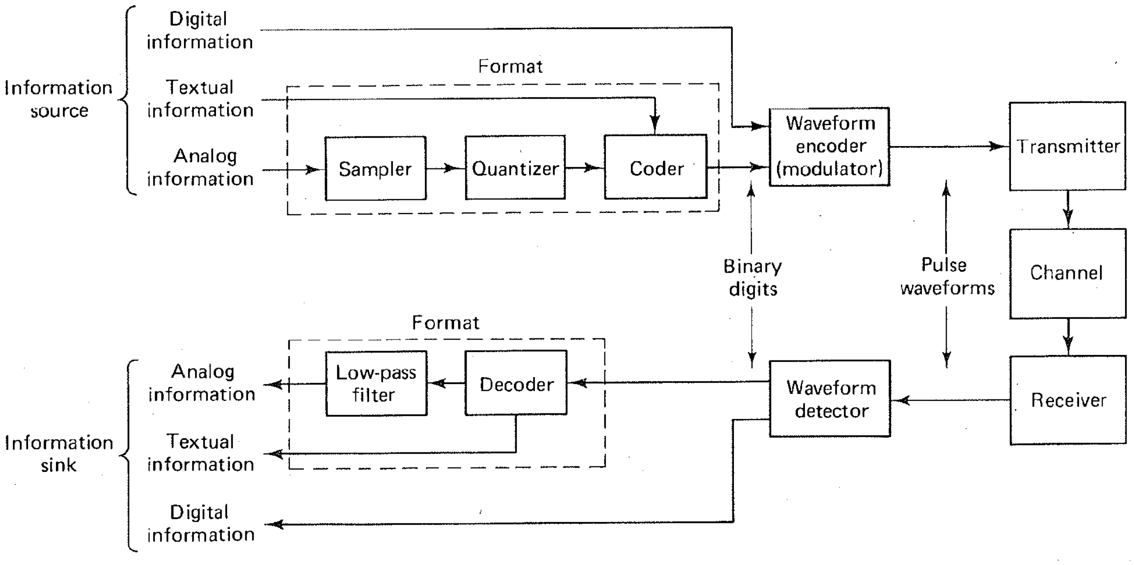


Figure 2.2 Formatting and transmission of baseband signals.

2.2 FORMATTING TEXTUAL DATA (CHARACTER CODING)

The original form of most communicated data (except for computer-to-computer transmissions) is either textual or analog. If the data consist of alphanumeric text, they will be character encoded with one of several standard formats, examples of which are, the American Standard Code for Information Interchange (ASCII), the Extended Binary Coded Decimal Interchange Code (EBCDIC), Baudot, and Hollerith. The textual material is thereby transformed into a digital format. The ASCII format is shown in Figure 2.3; the EBCDIC format is shown in Figure 2.4. The bit numbers signify the order of serial transmission, where bit number 1 is the first signaling element. Character coding, then, is the step that transforms text into binary digits (bits). Sometimes, existing character codes are modified to meet specialized needs. For example, the 7-bit ASCII code (Figure 2.3) can be modified to include an added bit for error detection purposes (see Chapter 5). On the other hand, sometimes the code is truncated to a 6-bit ASCII version, which provides capability for only 64 characters instead of the 128 characters allowed by 7-bit ASCII.

2.3 MESSAGES, CHARACTERS, AND SYMBOLS

Textual messages are comprised of a sequence of alphanumeric characters. When digitally transmitted the characters are first encoded into a sequence of bits, called a *bit stream* or *baseband signal.* Groups of k bits can then be combined to form new digits, or *symbols*, from a finite symbol set or alphabet of $M = 2^k$ such symbols. A system using a symbol set size of M is referred to as an *M-ary system.* The value of k or M represents an important initial choice in the design of any digital communication system. For $k = 1$, the system is termed *binary*, the size of the symbol set is $M = 2$, and the modulator uses one of the two different waveforms to represent the binary "one" and the other to represent the binary "zero." For this special case, the symbol and the bit are the same. For $k = 2$, the system is termed *quaternary* or *4-ary* ($M = 4$). At each symbol time, the modulator uses one of the four different waveforms that represents the symbol. The partitioning of the sequence of message bits is determined by the specification of the symbol set size, M. The following example should help clarify the relationship between the terms "message," "character," "symbol," "bit," and "digital waveform."

2.3.1 Example of Messages, Characters, and Symbols

Figure 2.5 shows examples of bit stream partitioning, based on the system specification for the values of k and M. The textual message in the figure is the word "THINK." Using 6-bit ASCII character coding (bit numbers 1 to 6 from Figure 2.3) yields a bit stream comprised of 30 bits. In Figure 2.5a, the symbol set size, M, has been chosen to be 8 (each symbol represents an 8-ary digit). The bits are therefore partitioned into groups of three ($k = \log_2 8$); the resulting 10 numbers

Bits				5	0	1	0	1	0	1	0	1
				6	0	0	1	1	0	0	1	1
1	2	3	4	7	0	0	0	0	1	1	1	1
0	0	0	0		NUL	DLE	SP	0	@	P	'	p
1	0	0	0		SOH	DC1	!	1	A	Q	a	q
0	1	0	0		STX	DC2	"	2	B	R	b	r
1	1	0	0		ETX	DC3	#	3	C	S	c	s
0	0	1	0		EOT	DC4	$	4	D	T	d	t
1	0	1	0		ENQ	NAK	%	5	E	U	e	u
0	1	1	0		ACK	SYN	&	6	F	V	f	v
1	1	1	0		BEL	ETB	'	7	G	W	g	w
0	0	0	1		BS	CAN	(	8	H	X	h	x
1	0	0	1		HT	EM	)	9	I	Y	i	y
0	1	0	1		LF	SUB	*	:	J	Z	j	z
1	1	0	1		VT	ESC	+	;	K	[	k	{
0	0	1	1		FF	FS	,	<	L	\	l	|
1	0	1	1		CR	GS	–	=	M	]	m	}
0	1	1	1		SO	RS	.	>	N	Λ	n	~
1	1	1	1		SI	US	/	?	O	—	o	DEL

NUL	Null, or all zeros	DC1	Device control 1
SOH	Start of heading	DC2	Device control 2
STX	Start of text	DC3	Device control 3
ETX	End of text	DC4	Device control 4
EOT	End of transmission	NAK	Negative acknowledge
ENQ	Enquiry	SYN	Synchronous idle
ACK	Acknowledge	ETB	End of transmission block
BEL	Bell, or alarm	CAN	Cancel
BS	Backspace	EM	End of medium
HT	Horizontal tabulation	SUB	Substitute
LF	Line feed	ESC	Escape
VT	Vertical tabulation	FS	File separator
FF	Form feed	GS	Group separator
CR	Carriage return	RS	Record separator
SO	Shift out	US	Unit separator
SI	Shift in	SP	Space
DLE	Data link escape	DEL	Delete

Figure 2.3 Seven-bit American standard code for information interchange (ASCII).

Bits			5	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1
			6	0	0	0	0	1	1	1	1	0	0	0	0	1	1	1	1
			7	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1
			8	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1
1	2	3	4																
0	0	0	0	NUL	SOH	STX	ETX	PF	HT	LC	DEL			SMM	VT	FF	CR	SO	SI
0	0	0	1	DLE	DC1	DC2	DC3	RES	NL	BS	IL	CAN	EM	CC		IFS	IGS	IRS	IUS
0	0	1	0	DS	SOS	FS		BYP	LF	EOB	PRE			SM			ENQ	ACK	BEL
0	0	1	1			SYN		PN	RS	US	EOT					DC4	NAK		SUB
0	1	0	0	SP										¢		<	(	+	!
0	1	0	1	&										!	$	*	)	;	¬
0	1	1	0	–	/										,	%	__	>	?
0	1	1	1											:	#	@	'	=	"
1	0	0	0		a	b	c	d	e	f	g	h	i						
1	0	0	1		j	k	l	m	n	o	p	q	r						
1	0	1	0			s	t	u	v	w	x	y	z						
1	0	1	1																
1	1	0	0		A	B	C	D	E	F	G	H	I						
1	1	0	1		J	K	L	M	N	O	P	Q	R						
1	1	1	0			S	T	U	V	W	X	Y	Z						
1	1	1	1	0	1	2	3	4	5	6	7	8	9						

PF	Punch off
HT	Horizontal tab
LC	Lower case
DEL	Delete
SP	Space
UC	Upper case
RES	Restore
NL	New line
BS	Backspace
IL	Idle
PN	Punch on
EOT	End of transmission
BYP	Bypass
LF	Line feed
EOB	End of block
PRE	Prefix (ESC)
RS	Reader stop
SM	Start message
DS	Digit select
SOS	Start of significance
IFS	Interchange file separator
IGS	Interchange group separator
IRS	Interchange record separator
IUS	Interchange unit separator
Others	Same as ASCII

Figure 2.4 EBCDIC character code set.

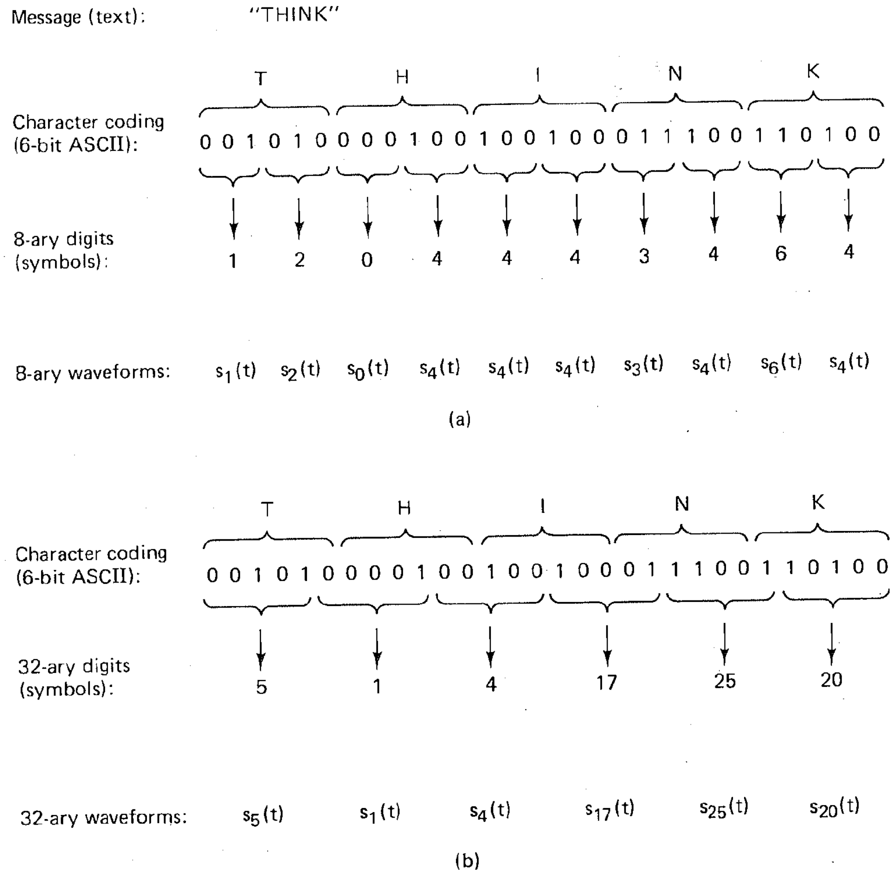


Figure 2.5 Messages, characters, and symbols. (a) 8-ary example. (b) 32-ary example.

represent the 10 octal symbols to be transmitted. The transmitter must have a repertoire of eight waveforms, $s_i(t)$, where $i = 1, \ldots, 8$, to represent the possible symbols, any one of which may be transmitted during a symbol time. The final row of Figure 2.5a lists the 10 waveforms that an 8-ary modulating system transmits to represent the textual message "THINK."

In Figure 2.5b, the symbol set size, M, has been chosen to be 32 (each symbol represents a 32-ary digit). The bits are therefore taken five at a time, and the resulting group of six numbers represent the six 32-ary symbols to be transmitted. Notice that there is no need for the symbol boundaries and the character boundaries to coincide. The first symbol represents $\frac{5}{6}$ of the first character, "T." The second symbol represents the remaining $\frac{1}{6}$ of the character "T" and $\frac{4}{6}$ of the next character, "H," and so on. It is not necessary that the characters be partitioned more aesthetically. The system sees the characters as a string of digits to be transmitted; only the end user (or the user's teleprinter machine) ascribes textual meaning to the final delivered sequence of bits. In this 32-ary case, a transmitter needs a repertoire of 32 waveforms, $s_i(t)$, where $i = 1, \ldots, 32$, one for each possible symbol that may be transmitted. The final row of the figure lists the six waveforms that a 32-ary modulating system transmits to represent the textual message "THINK."

2.4 FORMATTING ANALOG INFORMATION

If the information is analog, it cannot be character encoded as in the case of textual data; the information must first be transformed into a digital format. The process of transforming an analog waveform into a form that is compatible with a digital communication system starts with sampling the waveform to produce a discrete pulse-amplitude-modulated waveform, as described below.

2.4.1 The Sampling Theorem

The link between an analog waveform and its sampled version is provided by what is known as the *sampling process*. This process can be implemented in several ways, the most popular being the *sample-and-hold* operation. In this operation, a switch and storage mechanism (such as a transistor and a capacitor, or a shutter and a filmstrip) form a sequence of samples of the continuous input waveform. The output of the sampling process is called *pulse amplitude modulation* (PAM) because the successive output intervals can be described as a sequence of pulses with amplitudes derived from the input waveform samples. The analog waveform can be approximately retrieved from a PAM waveform by simple low-pass filtering. An important question is: How closely can a filtered PAM waveform approximate the original input waveform? This question can be answered by reviewing the *sampling theorem*, which states [1]: A bandlimited signal having no spectral components above f_m hertz can be determined uniquely by values sampled at uniform intervals of T_s seconds, where

$$T_s \leq \frac{1}{2f_m} \tag{2.1}$$

This particular statement is also known as the *uniform sampling theorem*. Stated another way, the upper limit on T_s can be expressed in terms of the sampling rate, denoted $f_s = 1/T_s$. The restriction, stated in terms of the sampling rate, is known as the *Nyquist criterion*. The statement is

$$f_s \geq 2f_m \tag{2.2}$$

The sampling rate $f_s = 2f_m$ is also called the *Nyquist rate*. The Nyquist criterion is a theoretically sufficient condition to allow an analog signal to be *reconstructed completely* from a set of uniformly spaced discrete-time samples. In the sections that follow, the validity of the sampling theorem is demonstrated using different sampling approaches.

2.4.1.1 Impulse Sampling

Here we demonstrate the validity of the sampling theorem using the frequency convolution property of the Fourier transform. Let us first examine the case of *ideal sampling* with a sequence of unit impulse functions. Assume an analog waveform, $x(t)$, as shown in Figure 2.6a, with a Fourier transform, $X(f)$, which is zero outside the interval $(-f_m < f < f_m)$, as shown in Figure 2.6b. The sampling of $x(t)$ can be viewed as the product of $x(t)$ with a periodic train of unit impulse functions, $x_\delta(t)$, shown in Figure 2.6c and defined as follows:

$$x_\delta(t) = \sum_{n=-\infty}^{\infty} \delta(t - nT_s) \tag{2.3}$$

where T_s is the sampling period and $\delta(t)$ is the unit impulse or Dirac delta function defined in Section 1.2.5. Let us choose $T_s = 1/2f_m$, so that the Nyquist criterion is just satisfied.

The *sifting property* of the impulse function (see Section A.4.1) states that

$$x(t)\delta(t - t_0) = x(t_0)\delta(t - t_0) \tag{2.4}$$

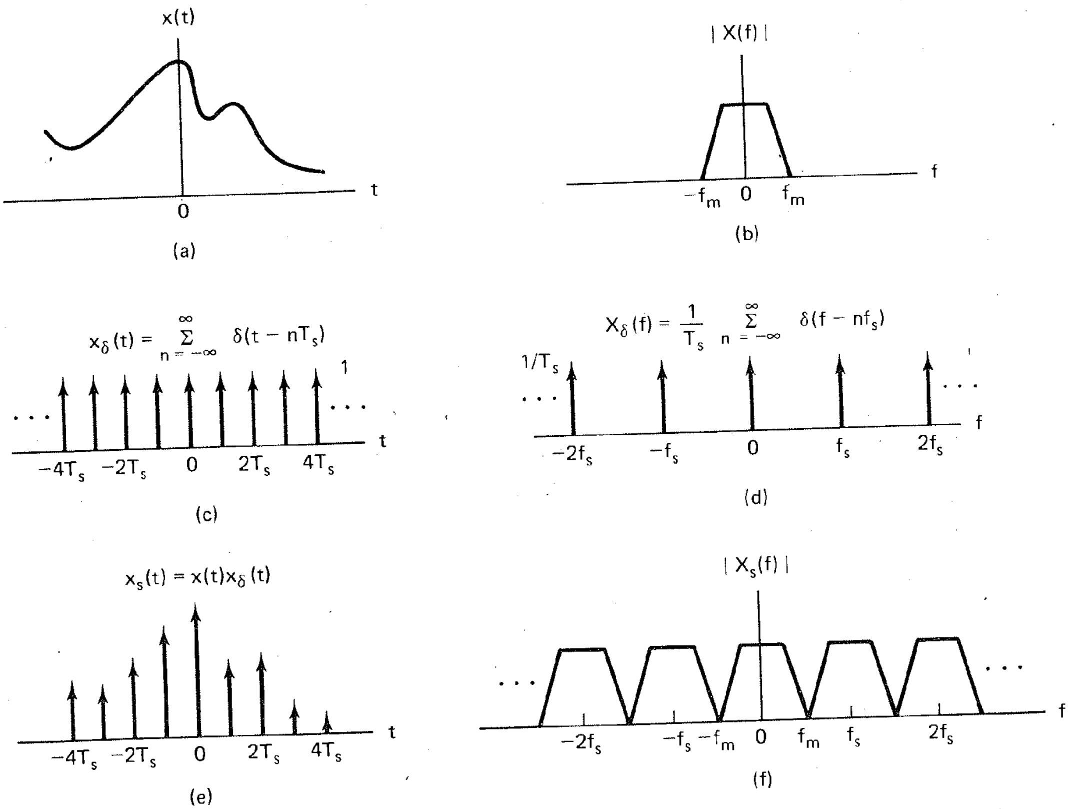


Figure 2.6 Sampling theorem using the frequency convolution property of the Fourier transform.

Using this property, we can see that $x_s(t)$, the sampled version of $x(t)$, shown in Figure 2.6e, is given by

$$\begin{aligned} x_s(t) = x(t)x_\delta(t) &= \sum_{n=-\infty}^{\infty} x(t)\delta(t - nT_s) \\ &= \sum_{n=-\infty}^{\infty} x(nT_s)\delta(t - nT_s) \end{aligned} \tag{2.5}$$

Using the *frequency convolution property* of the Fourier transform (see Section A.5.3), the time-domain product $x(t)x_\delta(t)$ of Equation (2.5) transforms to the frequency-domain convolution $X(f) * X_\delta(f)$, where $X_\delta(f)$ is the Fourier transform of the impulse train $x_\delta(t)$,

$$X_\delta(f) = \frac{1}{T_s} \sum_{n=-\infty}^{\infty} \delta(f - nf_s) \tag{2.6}$$

and where $f_s = 1/T_s$ is the sampling frequency. Notice that the Fourier transform of an impulse train is another impulse train; the values of the periods of the two trains are reciprocally related to one another. Figures 2.6c and d illustrate the impulse train $x_\delta(t)$ and its Fourier transform $X_\delta(f)$, respectively.

Convolution with an impulse function simply shifts the original function, as follows:

$$X(f) * \delta(f - nf_s) = X(f - nf_s) \tag{2.7}$$

We can solve for the transform, $X_s(f)$, of the sampled waveform as follows:

$$\begin{aligned} X_s(f) = X(f) * X_\delta(f) &= X(f) * \left[\frac{1}{T_s} \sum_{n=-\infty}^{\infty} \delta(f - nf_s)\right] \\ &= \frac{1}{T_s} \sum_{n=-\infty}^{\infty} X(f - nf_s) \end{aligned} \tag{2.8}$$

We therefore conclude that within the original bandwidth, the spectrum $X_s(f)$ of the sampled signal $x_s(t)$ is, to within a constant factor $(1/T_s)$, exactly the same as that of $x(t)$. In addition, the spectrum repeats itself periodically in frequency every f_s hertz. The sifting property of an impulse function makes the convolving of an impulse train with another function easy to visualize. The impulses act as sampling functions. Hence, convolution can be performed graphically by sweeping the impulse train, $X_\delta(f)$, in Figure 2.6d past the transform, $|X(f)|$, in Figure 2.6b. This sampling of $|X(f)|$ at each step in the sweep replicates $|X(f)|$ at each of the frequency positions of the impulse train, resulting in $|X_s(f)|$, shown in Figure 2.6f.

When the sampling rate is chosen, as it has been here, such that $f_s = 2f_m$, each spectral replicate is separated from each of its neighbors by a frequency band exactly equal to f_s hertz, and the analog waveform can theoretically be completely recovered from the samples, by the use of filtering. However, a filter with infinitely steep sides would be required. It should be clear that if $f_s > 2f_m$, the replications will move farther apart in frequency, as shown in Figure 2.7a, making it easier to perform the filtering operation. A typical low-pass filter characteristic that might be used to separate the baseband spectrum from those at higher frequencies is shown in the figure. When the sampling rate is reduced, such that $f_s < 2f_m$, the replications will overlap, as shown in Figure 2.7b, and some information will be lost. This phenomenon, the result of undersampling (sampling at too low a rate), is called *aliasing*. The Nyquist rate, $f_s = 2f_m$, is the sampling rate below which aliasing occurs; to avoid aliasing, the Nyquist criterion, $f_s \geq 2f_m$, must be satisfied.

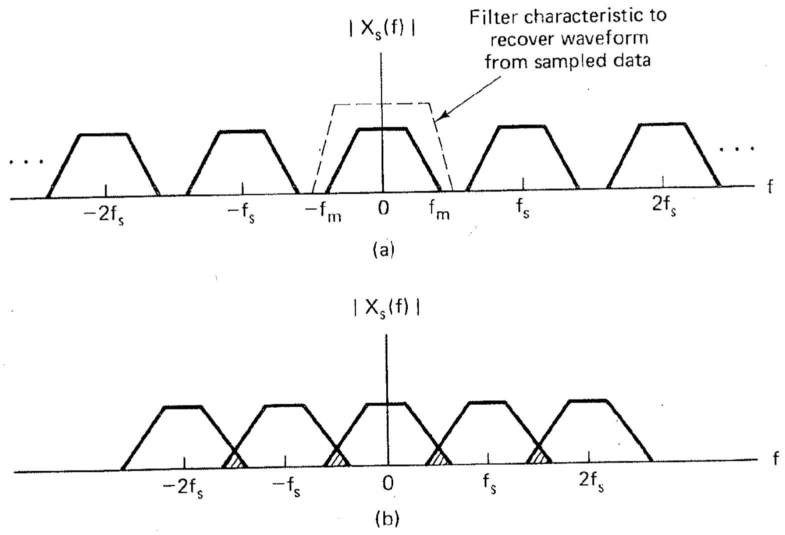


Figure 2.7 Spectra for various sampling rates. (a) Sampled spectrum ($f_s > 2f_m$). (b) Sampled spectrum ($f_s < 2f_m$).

As a matter of practical consideration, neither waveforms of engineering interest nor realizable bandlimiting filters are strictly bandlimited. These signals and filters can, however, be considered to be "essentially" bandlimited. By this we mean that a bandwidth can be determined beyond which the spectral components are attenuated to a level that is considered negligible.

2.4.1.2 Natural Sampling

Here we demonstrate the validity of the sampling theorem using the frequency shifting property of the Fourier transform. Although instantaneous sampling is a convenient model, a more practical way of accomplishing the sampling of a bandlimited analog signal, $x(t)$, is to multiply $x(t)$, shown in Figure 2.8a, by the pulse train or switching waveform, $x_p(t)$, shown in Figure 2.8c. Each pulse in $x_p(t)$ has width T and amplitude $1/T$. Multiplication by $x_p(t)$ can be viewed as the opening and closing of a switch. As before, the sampling frequency is designated f_s, and its reciprocal, the time period between samples, is designated T_s.

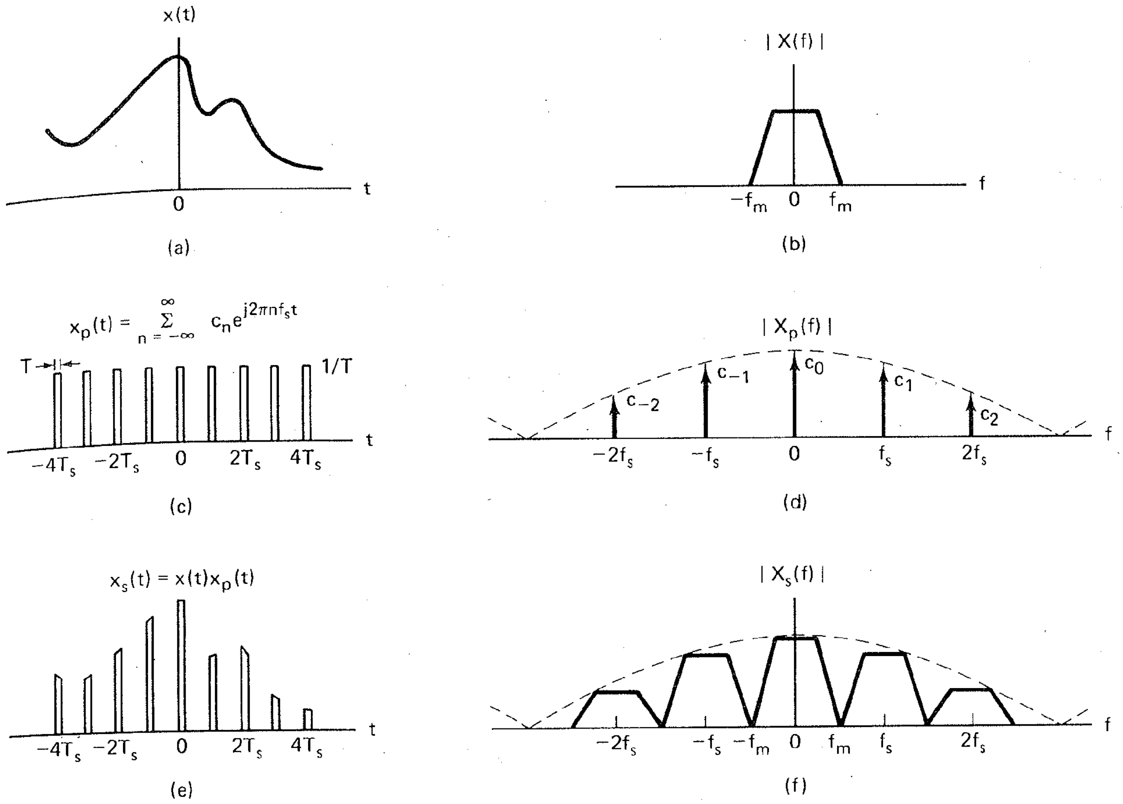

x(t)
0
t
(a)
| X(f) |
$-f_m$ 0 f_m
f
(b)
$x_p(t) = \sum_{n=-\infty}^{\infty} c_n e^{j2\pi n f_s t}$
T
1/T
$-4T_s$ $-2T_s$ 0 $2T_s$ $4T_s$
t
(c)
$| X_p(f) |$
c_{-2} c_{-1} c_0 c_1 c_2
$-2f_s$ $-f_s$ 0 f_s $2f_s$
f
(d)
$x_s(t) = x(t)x_p(t)$
$-4T_s$ $-2T_s$ 0 $2T_s$ $4T_s$
t
(e)
$| X_s(f) |$
$-2f_s$ $-f_s$ $-f_m$ 0 f_m f_s $2f_s$
f
(f)

Figure 2.8 Sampling theorem using the frequency shifting property of the Fourier transform.

The resulting sampled-data sequence, $x_s(t)$, is illustrated in Figure 2.8e and is expressed as

$$x_s(t) = x(t)x_p(t) \tag{2.9}$$

The sampling here is termed *natural sampling*, since the top of each pulse in the $x_s(t)$ sequence retains the shape of its corresponding analog segment during the pulse interval. Using Equation (A.13), we can express the periodic pulse train $x_p(t)$ as a Fourier series in the form

$$x_p(t) = \sum_{n=-\infty}^{\infty} c_n e^{j2\pi n f_s t} \tag{2.10}$$

where the sampling rate, $f_s = 1/T_s$, is chosen equal to $2f_m$, so that the Nyquist criterion is just satisfied. From Equation (A.24), $c_n = (1/T_s)\ \text{sinc}\ (nT/T_s)$, where T is the pulse width, $1/T$ is the pulse amplitude, and

$$\text{sinc}\ y = \frac{\sin \pi y}{\pi y}$$

The envelope of the magnitude spectrum of the pulse train, seen as a dashed line in Figure 2.8d, has the characteristic sinc shape. Combining Equations (2.9) and (2.10), we can express $x_s(t)$ as

$$x_s(t) = x(t) \sum_{n=-\infty}^{\infty} c_n e^{j2\pi n f_s t} \tag{2.11}$$

The transform, $X_s(f)$, of the sampled waveform is found as follows:

$$X_s(f) = \mathcal{F}\left\{x(t) \sum_{n=-\infty}^{\infty} c_n e^{j2\pi n f_s t}\right\} \tag{2.12}$$

For linear systems, we can interchange the operations of summation and Fourier transformation. Therefore, we can write

$$X_s(f) = \sum_{n=-\infty}^{\infty} c_n \mathcal{F}\{x(t)e^{j2\pi n f_s t}\} \tag{2.13}$$

Using the *frequency translation* property of the Fourier transform (see Section A.3.2), we solve for $X_s(f)$ as follows:

$$X_s(f) = \sum_{n=-\infty}^{\infty} c_n X(f - nf_s) \tag{2.14}$$

Similar to the unit impulse sampling case, Equation (2.14) and Figure 2.8f illustrate that $X_s(f)$ is a replication of $X(f)$, periodically repeated in frequency every f_s hertz. In this natural-sampled case, however, we see that $X_s(f)$ is weighted by the Fourier series coefficients of the pulse train, compared to a constant value in the impulse-sampled case. It is satisfying to note that *in the limit*, as the pulse width, T, approaches zero, c_n approaches $1/T_s$ for all n (see the example that follows), and Equation (2.14) converges to Equation (2.8).

Example 2.1 Comparison of Impulse Sampling and Natural Sampling

> Consider a given waveform, $x(t)$, with Fourier transform, $X(f)$. Let $X_{s1}(f)$ be the spectrum of $x_{s1}(t)$, which is the result of sampling $x(t)$ with a unit impulse train $x_\delta(t)$. Let $X_{s2}(f)$ be the spectrum of $x_{s2}(t)$, the result of sampling $x(t)$ with a pulse train, $x_p(t)$, with pulse width, T, amplitude $1/T$ and period, T_s. Show that in the limit, as T approaches zero, $X_{s1}(f) = X_{s2}(f)$.
>
> *Solution*
>
> From Equation (2.8),
>
> $$X_{s1}(f) = \frac{1}{T_s} \sum_{n=-\infty}^{\infty} X(f - nf_s)$$
>
> and from Equation (2.14),
>
> $$X_{s2}(f) = \sum_{n=-\infty}^{\infty} c_n X(f - nf_s)$$

As the pulse width $T \to 0$, and the pulse amplitude approaches infinity (the area of the pulse remains unity), $x_p(t) \to x_\delta(t)$. Using Equation (A.14), we can solve for c_n in the limit as follows:

$$c_n = \lim_{T \to 0} \frac{1}{T_s} \int_{-T_s/2}^{T_s/2} x_p(t) e^{-j2\pi n f_s t}\, dt$$
$$= \frac{1}{T_s} \int_{-T_s/2}^{T_s/2} x_\delta(t) e^{-j2\pi n f_s t}\, dt$$

Since, within the range of integration, $-T_s/2$ to $T_s/2$, the only contribution of $x_\delta(t)$ is that due to the impulse at the origin, we can write

$$c_n = \frac{1}{T_s} \int_{-T_s/2}^{T_s/2} \delta(t) e^{-j2\pi n f_s t}\, dt = \frac{1}{T_s}$$

Therefore, in the limit, $X_{s1}(f) = X_{s2}(f)$ for all n.

2.4.1.3 Sample-and-Hold Operation

The simplest and thus most popular sampling method, *sample and hold*, can be described by the convolution of the sampled pulse train, $[x(t)x_\delta(t)]$, shown in Figure 2.6e, with a unity amplitude rectangular pulse, $p(t)$, of pulse width T_s. This time convolution results in the *flat-top* sampled sequence, $x_s(t)$:

$$\begin{aligned} x_s(t) &= p(t) * [x(t)x_\delta(t)] \\ &= p(t) * \left[x(t) \sum_{n=-\infty}^{\infty} \delta(t - nT_s) \right] \end{aligned} \tag{2.15}$$

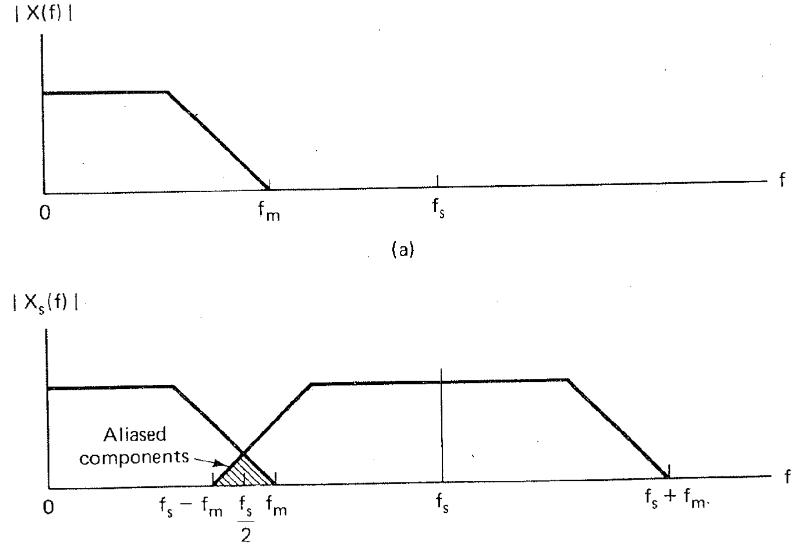
The Fourier transform, $X_s(f)$, of the time convolution in Equation (2.15) is the frequency-domain product between the transform $P(f)$ of the rectangular pulse and the periodic spectrum, shown in Figure 2.6f, of the impulse-sampled data:

$$\begin{aligned} X_s(f) &= P(f)\mathscr{F}\left\{ x(t) \sum_{n=-\infty}^{\infty} \delta(t - nT_s) \right\} \\ &= P(f)\left\{ X(f) * \left[\frac{1}{T_s} \sum_{n=-\infty}^{\infty} \delta(f - nf_s) \right] \right\} \\ &= P(f) \frac{1}{T_s} \sum_{n=-\infty}^{\infty} X(f - nf_s) \end{aligned} \tag{2.16}$$

where $P(f)$ is of the form T_s sinc fT_s. The effect of this product operation results in a spectrum similar in appearance to the natural-sampled example presented in Figure 2.8f. The most obvious effect of the hold operation is the significant attenuation of the higher-frequency spectral replicates (compare Figure 2.8f to Figure 2.6f), which is a desired effect. Additional analog postfiltering is usually required to finish the filtering process by further attenuating the residual spectral components located at the multiples of the sample rate. A secondary effect of the hold operation is the nonuniform spectral gain, $P(f)$, applied to the desired baseband spectrum shown in Equation (2.16). The postfiltering operation can compensate for this attenuation by incorporating the inverse of $P(f)$ over the signal passband.

2.4.2 Aliasing

Figure 2.9 is a detailed view of the positive half of the baseband spectrum and one of the replicates from Figure 2.7b. It illustrates aliasing in the frequency domain. The overlapped region, shown in Figure 2.9b, contains that part of the spectrum which is aliased due to *undersampling*. The aliased spectral components represent ambiguous data that can be retrieved only under special conditions (see Section 11.4.4, on subband coding). In general, the ambiguity is not resolved and the ambiguous data appear in the frequency band between $(f_s - f_m)$ and f_m. Figure 2.10 illustrates that a higher sampling rate, f_s', can eliminate the aliasing by separating the spectral replicates; the resulting spectrum in Figure 2.10b corresponds to the case in Figure 2.7a. Figures 2.11 and 2.12 illustrate two ways of eliminating aliasing using *antialiasing filters*. In Figure 2.11 the analog signal is *prefiltered* so that the new maximum frequency, f_m', is reduced to $f_s/2$ or less. Thus there are no aliased components seen in Figure 2.11b, since $f_s > 2f_m'$. Eliminating the aliasing terms prior to sampling is good engineering practice. When the signal structure is well known, the aliased terms can be eliminated after sampling, with a low-pass filter operating on the sampled data [2]. In Figure 2.12 the aliased components are removed by *postfiltering* after sampling; the filter cutoff frequency, f_m'', removes the aliased components; f_m'' needs to be less than



(a)

(b)

Figure 2.9 Aliasing in the frequency domain. (a) Continuous signal spectrum. (b) Sampled signal spectrum.

n com-
: signal

ım and
quency
: of the
ıonents
ns (see
'ed and
nd f_m.
aliasing
0b cor-
vays of
ignal is
or less.
$> 2f'_m$.
ractice.
ed after
ıre 2.12
e filter
ss than

f

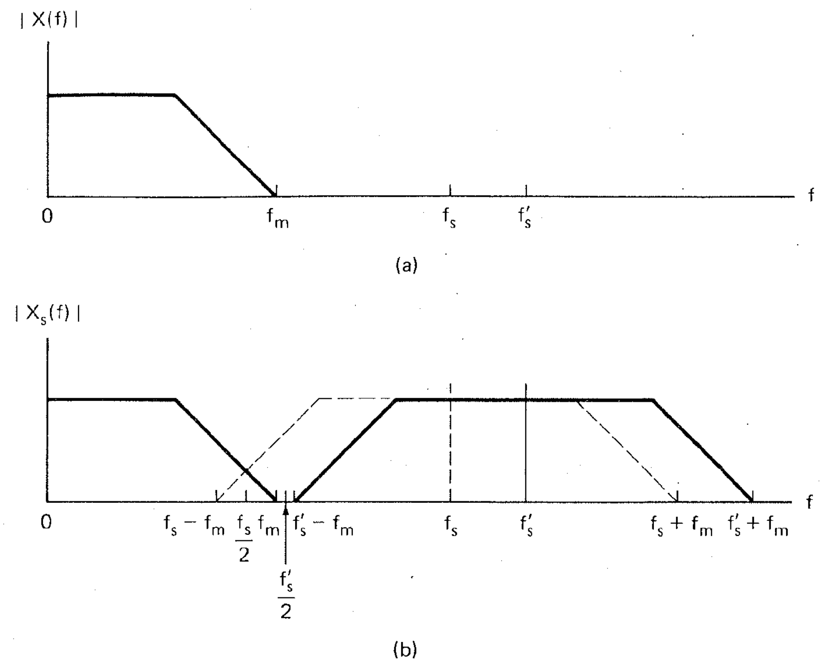


Figure 2.10 Higher sampling rate eliminates aliasing. (a) Continuous signal spectrum. (b) Sampled signal spectrum.

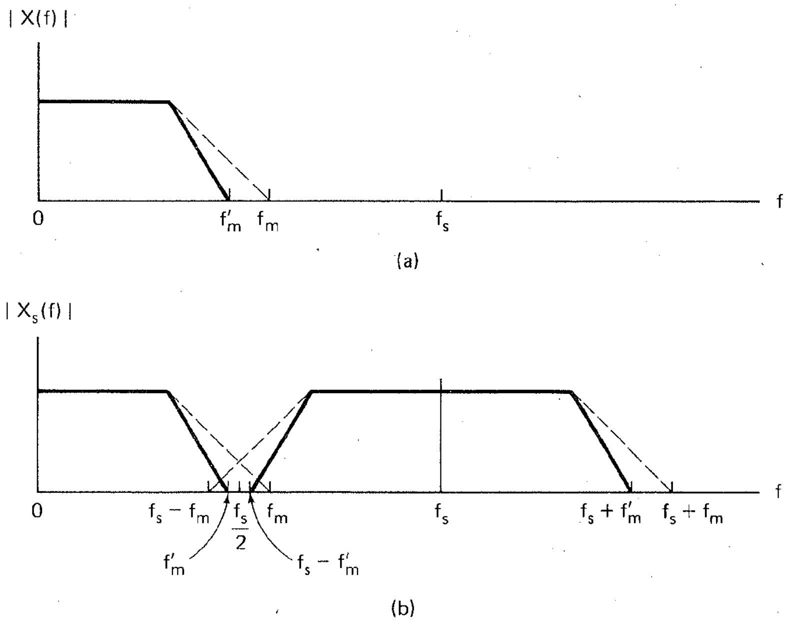


Figure 2.11 Sharper-cutoff filters eliminate aliasing. (a) Continous signal spectrum. (b) Sampled signal spectrum.

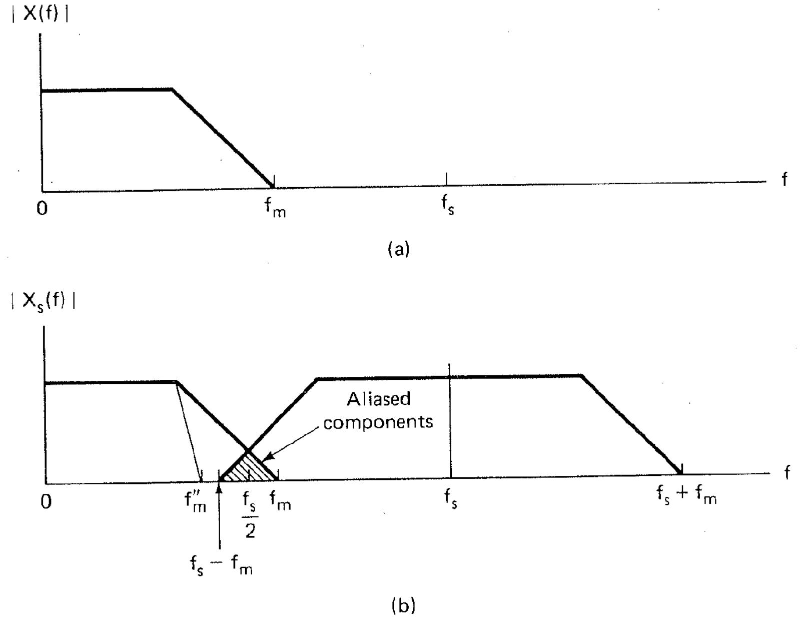


Figure 2.12 Postfilter eliminates aliased portion of spectrum. (a) Continuous signal spectrum. (b) Sampled signal spectrum.

$(f_s - f_m)$. Notice that the filtering techniques for eliminating the aliased portion of the spectrum in Figures 2.11 and 2.12 *will result in a loss* of some of the signal information. For this reason, the sample rate, cutoff bandwidth, and filter type selected for a particular signal bandwidth are all interrelated.

Realizable filters require a nonzero bandwidth for the transition between the passband and the required out-of-band attenuation. This is called the *transition bandwidth*. To minimize the system sample rate, we desire that the antialiasing filter have a small transition bandwidth. Filter complexity and cost rise sharply with narrower transition bandwidth, so a trade-off is required between the cost of a small transition bandwidth and the costs of the higher sampling rate, which are those of more storage and higher transmission rates. In many systems the answer has been to make the transition bandwidth between 10 and 20% of the signal bandwidth. If we account for the 20% transition bandwidth of the antialiasing filter, we have an *engineer's version* of the Nyquist sampling rate:

$$f_s \geq 2.2 f_m \tag{2.17}$$

Figure 2.13 provides some insight into aliasing as seen in the time domain. The sampling instants of the solid-line sinusoid have been chosen so that the sinusoidal signal is undersampled. Notice that the resulting ambiguity allows one to draw a totally different (dashed-line) sinusoid, following the undersampled points.

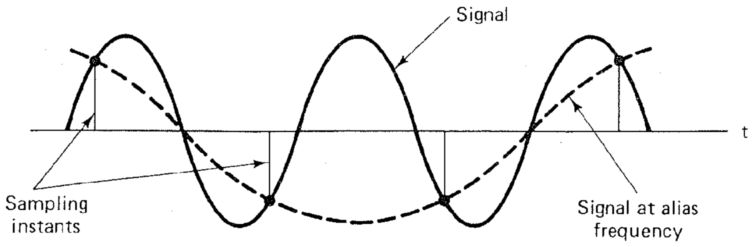


Figure 2.13 Alias frequency generated by sub-Nyquist sampling rate.

Example 2.2 Sampling Rate for a High-Quality Music System

We wish to produce a high-quality digitization of a 20-kHz bandwidth music source. We are to determine a reasonable sample rate for this source. By the engineer's version of the Nyquist rate, in Equation (2.17), the sampling rate should be greater than 44.0 ksamples/s. As a matter of comparison, the standard sampling rate for the compact disc digital audio player is 44.1 ksamples/s, and the standard sampling rate for studio-quality audio is 48.0 ksamples/s.

2.4.3 Signal Interface for a Digital System

Let us examine four ways in which analog source information can be described. Figure 2.14 illustrates the choices. Let us refer to the waveform in Figure 2.14a as the *original analog waveform*. Figure 2.14b represents a sampled version of the original waveform, typically referred to as *natural-sampled data* or PAM

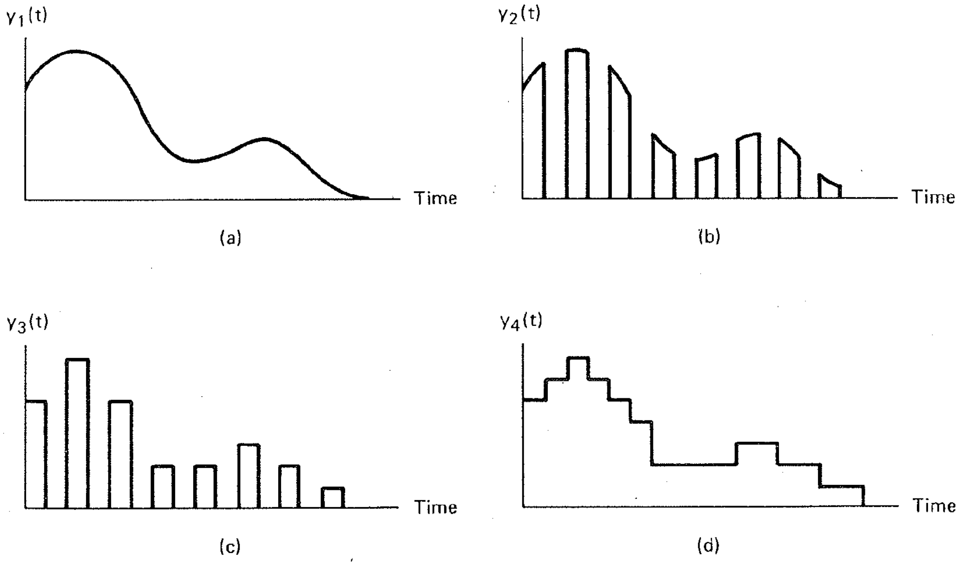


Figure 2.14 Amplitude and time coordinates of source data. (a) Original analog waveform. (b) Natural-sampled data. (c) Quantized samples. (d) Sample and hold.

d portion
he signal
lter type

ween the
ransition
tialiasing
: sharply
the cost
:e, which
tems the
% of the
antialias-

(2.17)

: domain.
that the
llows one
:rsampled

(pulse amplitude modulation). Do you suppose that the sampled data in Figure 2.14b are compatible with a digital system? No, they are not, because the amplitude of each natural sample still has an infinite number of possible values; a digital system deals with a finite number of symbols. Even if the sampling is flat-top sampling, the possible pulse values form an infinite set, since they reflect all the possible values of the continuous analog waveform. Figure 2.14c illustrates the original waveform represented by discrete pulses. Here the pulses have flat tops *and* the pulse amplitude values are limited to a finite set. Each pulse is expressed as a level from a finite number of predetermined levels; each such level can be represented by a symbol from a finite alphabet. The pulses in Figure 2.14c are referred to as *quantized samples*; such a format is the obvious choice for interfacing with a digital system. The format in Figure 2.14d may be construed as the output of a sample-and-hold circuit. When the sample values are quantized to a finite set, this format can also interface with a digital system. After quantization, the analog waveform can still be recovered, but not precisely; improved reconstruction fidelity of the analog waveform can be achieved by increasing the number of quantization levels (requiring increased system bandwidth). Signal distortion due to quantization is treated in the following sections (and in Chapter 11).

2.5 SOURCES OF CORRUPTION

The analog signal recovered from the sampled, quantized, and transmitted pulses will contain corruption from several sources. The sources of corruption are related to (1) sampling and quantizing effects, and (2) channel effects. These effects are considered in the sections that follow.

2.5.1 Sampling and Quantizing Effects

2.5.1.1 Quantization Noise

The distortion inherent in quantization is a round-off or truncation error. The process of encoding the PAM waveform into a quantized waveform involves discarding some of the original analog information. This distortion, introduced by the need to approximate the analog waveform with quantized samples, is referred to as *quantization noise*; the amount of such noise is inversely proportional to the number of levels employed in the quantization process. The signal-to-noise ratio of quantized pulses is treated in Section 2.5.3.

2.5.1.2 Quantizer Saturation

The quantizer (or analog-to-digital converter) allocates L levels to the task of approximating the continuous range of inputs with a finite set of outputs. The range of inputs for which the difference between the input and output is small is called the *operating range* of the converter. If the input exceeds this range, the difference between the input and the output becomes large, and we say that the converter is operating in *saturation*. Saturation errors, being large, are more objectionable than quantizing noise. Generally, saturation is avoided by the use of automatic gain control (AGC), which effectively extends the operating range of the converter. Chapter 11 covers quantizer saturation in greater detail.

2.5.1.3 Timing Jitter

Our analysis of the sampling theorem predicted precise reconstruction of the signal based on uniformly spaced samples of the signal. If there is a slight jitter in the position of the sample, the sampling is no longer uniform. Although exact reconstruction is still possible if the sample positions are accurately known, the jitter is usually a random process and thus the sample positions are not accurately known. The effect of the jitter is equivalent to frequency modulation (FM) of the baseband signal. If the jitter is random, a low-level wideband spectral contribution is induced whose properties are very close to those of the quantizing noise. If the jitter exhibits periodic components, as might be found in data extracted from a tape recorder, the periodic FM will induce low-level spectral lines in the data. Timing jitter can be controlled with very good power supply isolation and stable clock references.

2.5.2 Channel Effects

2.5.2.1 Channel Noise

Thermal noise, interference from other users, and interference from circuit switching transients can cause errors in detecting the pulses carrying the digitized samples. Channel-induced errors can degrade the reconstructed signal quality quite quickly. This rapid degradation of output signal quality with channel-induced errors is called a *threshold effect*. If the channel noise is small, there will be no problem detecting the presence of the waveforms. Thus small noise does not corrupt the reconstructed signals. In this case, the only noise present in the reconstruction is the quantization noise. On the other hand, if the channel noise is large enough to affect our ability to detect the waveforms, the resultant detection error causes reconstruction errors. A large difference in behavior can occur for very small changes in channel noise level.

2.5.2.2 Intersymbol Interference

The channel is always bandlimited. A bandlimited channel disperses or spreads a pulse waveform passing through it (see Section 1.6.4). When the channel bandwidth is much greater than the pulse bandwidth, the spreading of the pulse will be slight. When the channel bandwidth is close to the signal bandwidth, the spreading will exceed a symbol duration and cause signal pulses to overlap. This overlapping is called *intersymbol interference* (ISI). Like any other source of interference, ISI causes system degradation (higher error rates); it is a particularly insidious form of interference because raising the signal power to overcome the interference will not improve the error performance. Details of how ISI is handled are presented in Section 2.11.

2.5.3 Signal-to-Noise Ratio for Quantized Pulses

Figure 2.15 illustrates an L-level linear quantizer for an analog signal with a peak-to-peak voltage range of $V_{pp} = V_p - (-V_p) = 2V_p$ volts. The quantized pulses assume positive and negative values, as shown in the figure. The step size between quantization levels, called the *quantile interval,* is denoted q volts. When the quantization levels are uniformly distributed over the full range, the quantizer is called a *uniform or linear quantizer.* Each sample value of the analog waveform is approximated with a quantized pulse; the approximation will result in an error no larger than $q/2$ in the positive direction or $-q/2$ in the negative direction. The degradation of the signal due to quantization is therefore limited to half a quantile interval, $\pm q/2$ volts.

A useful figure of merit for the uniform quantizer is the quantizer variance (mean-square error assuming zero mean). If we assume that the quantization error, e, is uniformly distributed over a single quantile interval q-wide (i.e., the analog input takes on all values with equal probability), the quantizer error variance is found to be

$$\sigma^2 = \int_{-q/2}^{+q/2} e^2 p(e)\, de \tag{2.18a}$$

$$= \int_{-q/2}^{+q/2} e^2 \frac{1}{q}\, de = \frac{q^2}{12} \tag{2.18b}$$

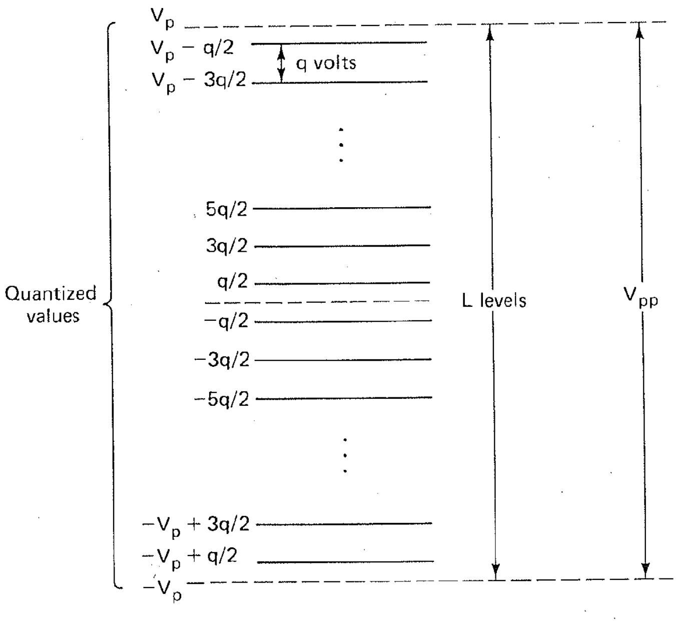


Figure 2.15 Quantization levels.

where $p(e) = 1/q$ is the (uniform) probability density function of the quantization error. The variance, σ^2, corresponds to the *average quantization noise power*. The peak power of the analog signal (normalized to 1 Ω) can be expressed as

$$V_p^2 = \left(\frac{V_{pp}}{2}\right)^2 = \left(\frac{Lq}{2}\right)^2 = \frac{L^2q^2}{4} \tag{2.19}$$

where L is the number of quantization levels. Equations (2.18) and (2.19) combined yield the ratio of peak signal power to average quantization noise power $(S/N)_q$, assuming that there are no errors due to ISI or channel noise:

$$\left(\frac{S}{N}\right)_q = \frac{L^2q^2/4}{q^2/12} = 3L^2 \tag{2.20}$$

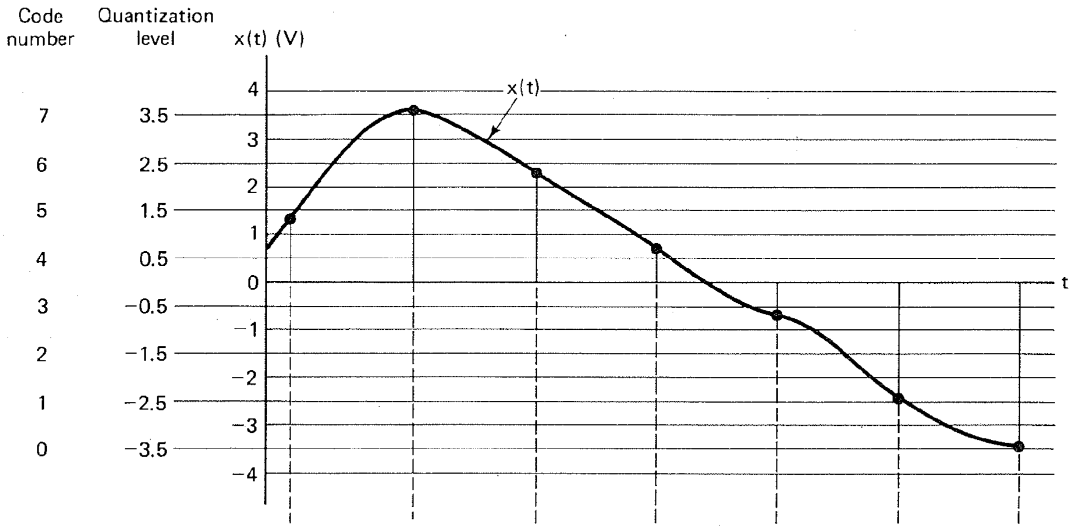
It is intuitively satisfying to see that $(S/N)_q$ improves as a function of the number of quantization levels squared. In the limit (as $L \to \infty$), the signal approaches the PAM format (with no quantization), and the signal-to-quantization noise ratio is infinite; in other words, with an infinite number of quantization levels, there is zero quantization noise.

2.6 PULSE CODE MODULATION

Pulse code modulation (PCM) is the name given to the class of baseband signals obtained from the quantized PAM signals by encoding each quantized sample into a *digital word* [3]. The source information is sampled and quantized to one of L levels; then each quantized sample is digitally encoded into an ℓ-bit ($\ell = \log_2 L$) codeword. For baseband transmission, the codeword bits will then be transformed to pulse waveforms. The essential features of binary PCM are shown in Figure 2.16. Assume that an analog signal, $x(t)$, is limited in its excursions to the range -4 to $+4$ V. The step size between quantization levels has been set at 1 V. Thus eight quantization levels are employed; these are located at -3.5, -2.5, . . . , $+3.5$ V. We assign the code number 0 to the level at -3.5 V, the code number 1 to the level at -2.5 V, and so on, until the level at 3.5 V, which is assigned the code number 7. Each code number has its representation in binary arithmetic, ranging from 000 for code number 0 to 111 for code number 7.

The ordinate in Figure 2.16 is labeled with quantization levels and their code numbers. Each sample of the analog signal is assigned to the quantization level closest to the value of the sample. Beneath the analog waveform, $x(t)$, are seen four representations of $x(t)$, as follows: the natural sample values, the quantized sample values, the code numbers, and the PCM sequence.

Note that in the example of Figure 2.16, each sample is represented by a 3-bit codeword. If the signal, $x(t)$, had been quantized to 16 levels, a 4-bit codeword would be needed to characterize each sample, or if $x(t)$ had been quantized to four levels, a 2-bit codeword would be needed. From Equation (2.20) it can be seen that the greater the number of quantization levels, the lower will be the quantization noise. Hence quantization noise performance can be traded off versus data rate.



Natural sample value	1.3	3.6	2.3	0.7	−0.7	−2.4	−3.4
Quantized sample value	1.5	3.5	2.5	0.5	−0.5	−2.5	−3.5
Code number	5	7	6	4	3	1	0
PCM sequence	101	111	110	100	011	001	000

Figure 2.16 Natural samples, quantized samples, and pulse code modulation. (Reprinted with permission from Taub and Schilling, *Principles of Communication Systems*, McGraw-Hill Book Company, New York, 1971, Fig. 6.5-1, p. 205.)

2.7 UNIFORM AND NONUNIFORM QUANTIZATION

2.7.1 Statistics of Speech Amplitudes

Speech communication is a very important and specialized area of digital communications. Human speech is characterized by unique statistical properties; one such property is illustrated in Figure 2.17. The abscissa represents speech signal magnitudes, normalized to the root-mean-square (rms) value of such magnitudes through a typical communication channel, and the ordinate is probability. For most voice communication channels, very low speech volumes predominate; 50% of the time, the voltage characterizing detected speech energy is less than one-fourth of the rms value. Large amplitude values are relatively rare; only 15% of the time does the voltage exceed the rms value. We see from Equation (2.18b) that the quantization noise depends on the step size (size of the quantile interval). When the steps are uniform in size the quantization is known as *uniform quantization*. Such a system would be wasteful for speech signals; many of the quantizing steps would rarely be used. In a system that uses equally spaced quantization levels, the quantization noise is the same for all signal magnitudes. Therefore, with uniform quantization, the signal-to-noise ratio (SNR) is worse for low-level signals than for high-level signals. *Nonuniform quantization* can provide fine quantization of the weak signals and coarse quantization of the strong signals. Thus in the case of nonuniform quantization, quantization noise can be made

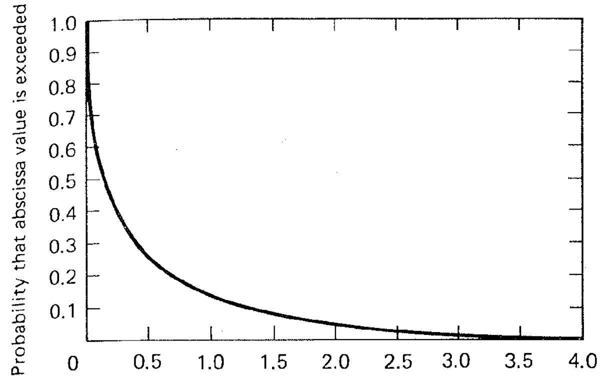


Figure 2.17 Statistical distribution of single-talker speech signal magnitudes.

proportional to signal size. The effect is to improve the overall SNR by reducing the noise for the predominant weak signals, at the expense of an increase in noise for the rarely occurring strong signals. Figure 2.18 compares the quantization of a strong versus a weak signal for uniform and nonuniform quantization. The staircase-like waveforms represent the approximations to the analog waveforms (after quantization distortion has been introduced). The SNR improvement that nonuniform quantization provides for the weak signal should be apparent. Nonuni-

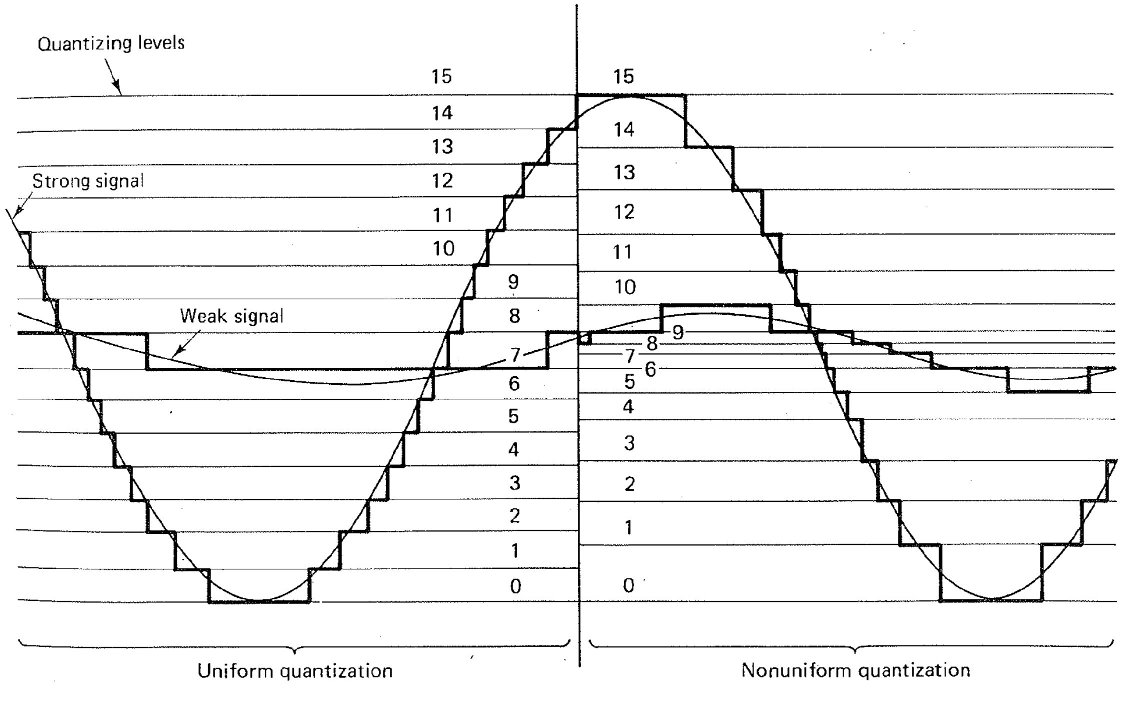


Figure 2.18 Uniform and nonuniform quantization of signals.

form quantization can be used to make the SNR a constant for all signals within the input range. For voicc signals, the typical input signal dynamic range is 40 decibels (dB), where a decibel is defined in terms of the ratio of power P_2 to power P_1:

$$\text{number of dB} = 10 \log_{10} \frac{P_2}{P_1} \tag{2.21}$$

With a uniform quantizer, weak signals would experience a 40-dB-poorer SNR than that of strong signals. The standard telephone technique of handling the large range of possible input signal levels is to use a *logarithmic-compressed* quantizer instead of a uniform one. With such a nonuniform compressor the output SNR is independent of the distribution of input signal levels.

2.7.2 Nonuniform Quantization

One way of achieving nonuniform quantization is to use a nonuniform quantizer characteristic, shown in Figure 2.19a. More often, nonuniform quantization is achieved by first distorting the original signal with a logarithmic compression characteristic, as shown in Figure 2.19b, and then using a uniform quantizer. For small magnitude signals the compression characteristic has a much steeper slope than for large magnitude signals. Thus a given signal change at small magnitudes will carry the uniform quantizer through more steps than the same change at large

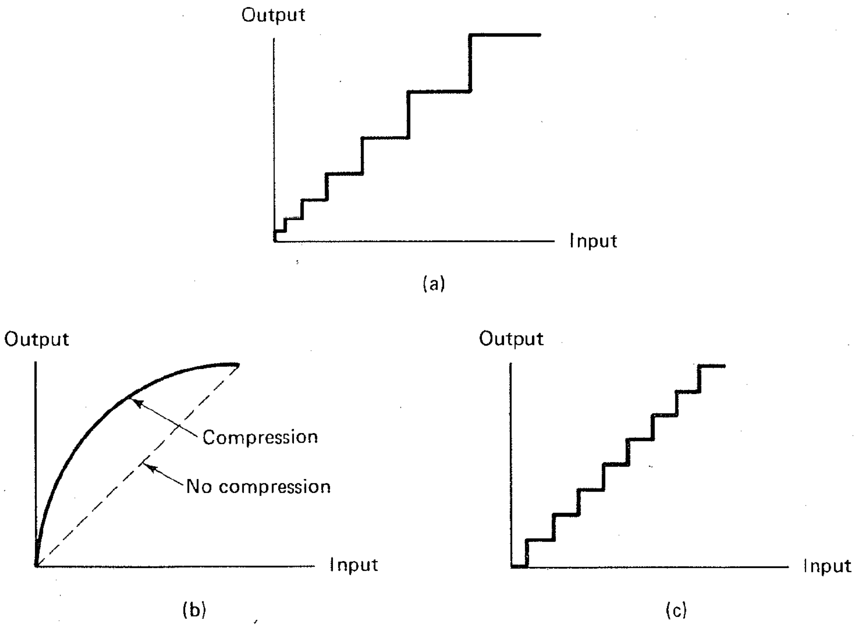


Figure 2.19 (a) Nonuniform quantizer characteristic. (b) Compression characteristic. (c) Uniform quantizer characteristic.

magnitudes. The compression characteristic effectively changes the distribution of the input signal magnitudes so that there is not a preponderance of *low* magnitude signals at the output of the compressor. After compression, the distorted signal is used as the input to a uniform (linear) quantizer characteristic, shown in Figure 2.19c. At the receiver, an inverse compression characteristic, called *expansion*, is applied so that the overall transmission is not distorted. The processing pair (compression and expansion) is usually referred to as *companding*.

2.7.3 Companding Characteristics

The early PCM systems implemented a smooth logarithmic compression function. Today, most PCM systems use a piecewise linear approximation to the logarithmic compression characteristic. In North America a μ-law compression characteristic is used:

$$y = y_{\max} \frac{\log_e[1 + \mu(|x|/x_{\max})]}{\log_e(1 + \mu)} \operatorname{sgn} x \tag{2.22}$$

where

$$\operatorname{sgn} x = \begin{cases} +1 & \text{for } x \geq 0 \\ -1 & \text{for } x < 0 \end{cases}$$

and where μ is a positive constant, x and y represent input and output voltages, and $x_{\max}$ and $y_{\max}$ are the maximum positive excursions of the input and output voltages, respectively. The compression characteristic is shown in Figure 2.20a for several values of μ. The standard value for μ is 255. Notice that $\mu = 0$ corresponds to linear amplification (uniform quantization).

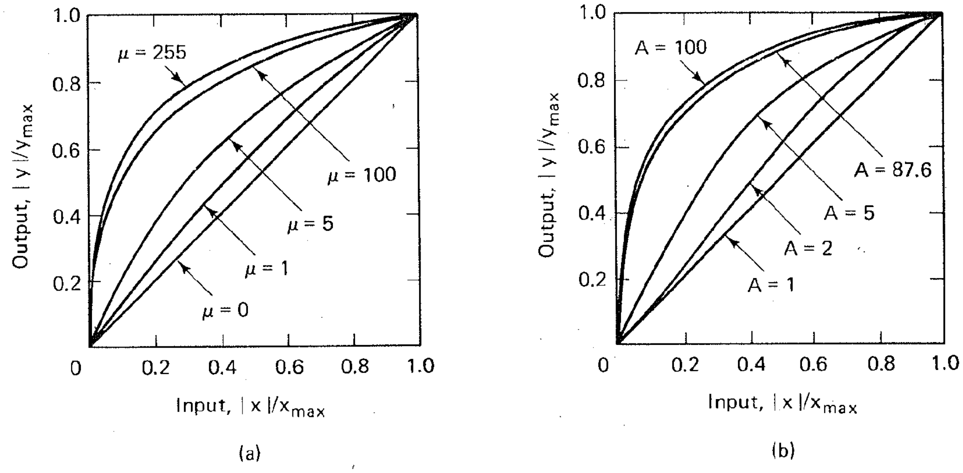


Figure 2.20 Compression characteristics. (a) μ-law characteristic. (b) *A*-law characteristic.

Another compression characteristic, used mainly in Europe, is the A-law characteristic, defined as

$$y = \begin{cases} y_{\max} \dfrac{A(\,|x|\,/x_{\max})}{1 + \log_e A} \operatorname{sgn} x & 0 < \dfrac{|x|}{x_{\max}} \leq \dfrac{1}{A} \\[2ex] y_{\max} \dfrac{1 + \log_e[A(\,|x|\,/x_{\max})]}{1 + \log_e A} \operatorname{sgn} x & \dfrac{1}{A} < \dfrac{|x|}{x_{\max}} < 1 \end{cases} \tag{2.23}$$

where A is a positive constant and x and y are as defined in Equation (2.22). The A-law compression characteristic is shown in Figure 2.20b for several values of A. A standard value for A is 87.6. See Chapter 11 for a more detailed treatment of μ-law and A-law companding characteristics.

2.8 BASEBAND TRANSMISSION

2.8.1 Waveform Representation of Binary Digits

We need to represent PCM binary digits by electrical pulses in order to transmit them through a baseband channel. Such a representation is shown in Figure 2.21. Codeword time slots are shown in Figure 2.21a, where the codeword is a 4-bit representation of each quantized sample. In Figure 2.21b, each binary one is represented by a pulse and each binary zero is represented by the absence of a pulse. Thus a sequence of electrical pulses having the pattern shown in Figure 2.21b can be used to transmit the information in the PCM bit stream, and hence the information in the quantized samples of a message.

At the receiver, a determination must be made as to the presence or absence of a pulse in each bit time slot. It will be shown in Section 2.9 that the likelihood of correctly detecting the presence of a pulse is a function of the pulse energy (or area under the pulse). Thus there is an advantage in making the pulse width, T', in Figure 2.21b as wide as possible. If we increase the pulse width to the maximum possible (equal to the bit time duration, T), we have the waveform shown in Figure 2.21c. Rather than describe this waveform as a sequence of present or absent pulses, we can describe it as a sequence of transitions between two levels. When the waveform occupies the upper voltage level it represents a binary one; when it occupies the lower voltage level it represents a binary zero.

2.8.2 PCM Waveform Types

Figure 2.22 illustrates the most commonly used PCM waveforms. The various waveforms are classified into the following groups:

1. Nonreturn-to-zero (NRZ)
2. Return-to-zero (RZ)
3. Phase encoded
4. Multilevel binary

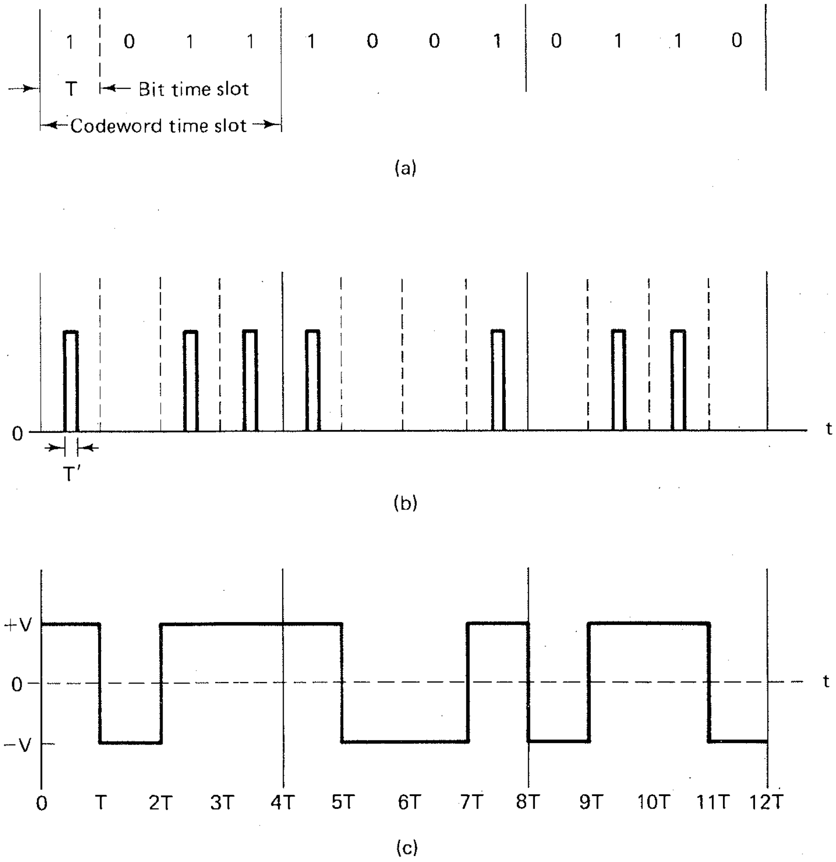


Figure 2.21 Example of waveform representation of binary digits. (a) PCM sequence. (b) Pulse representation of PCM. (c) Pulse waveform (transition between two levels).

The NRZ group is probably the most commonly used PCM waveform. It can be partitioned into the following subgroups: NRZ-L (L for level), NRZ-M (M for mark), and NRZ-S (S for space). NRZ-L is used extensively in digital logic. A binary one is represented by one level and a binary zero is represented by another level. There is a change in level whenever the data change from a one to a zero or from a zero to a one. With NRZ-M, the one, or *mark*, is represented by a change in level, and the zero, or *space*, is represented by no change in level. This is often referred to as *differential encoding*. NRZ-M is used primarily in magnetic tape recording. NRZ-S is the complement of NRZ-M: A one is represented by no change in level, and a zero is represented by a change in level.

The RZ waveforms consist of unipolar-RZ, bipolar-RZ, and RZ-AMI. These codes find application in baseband data transmission and in magnetic recording. With unipolar-RZ, a one is represented by a half-bit-wide pulse, and a zero is represented by the absence of a pulse. With bipolar-RZ, the ones and zeros are represented by opposite-level pulses that are one-half-bit wide. There is a pulse

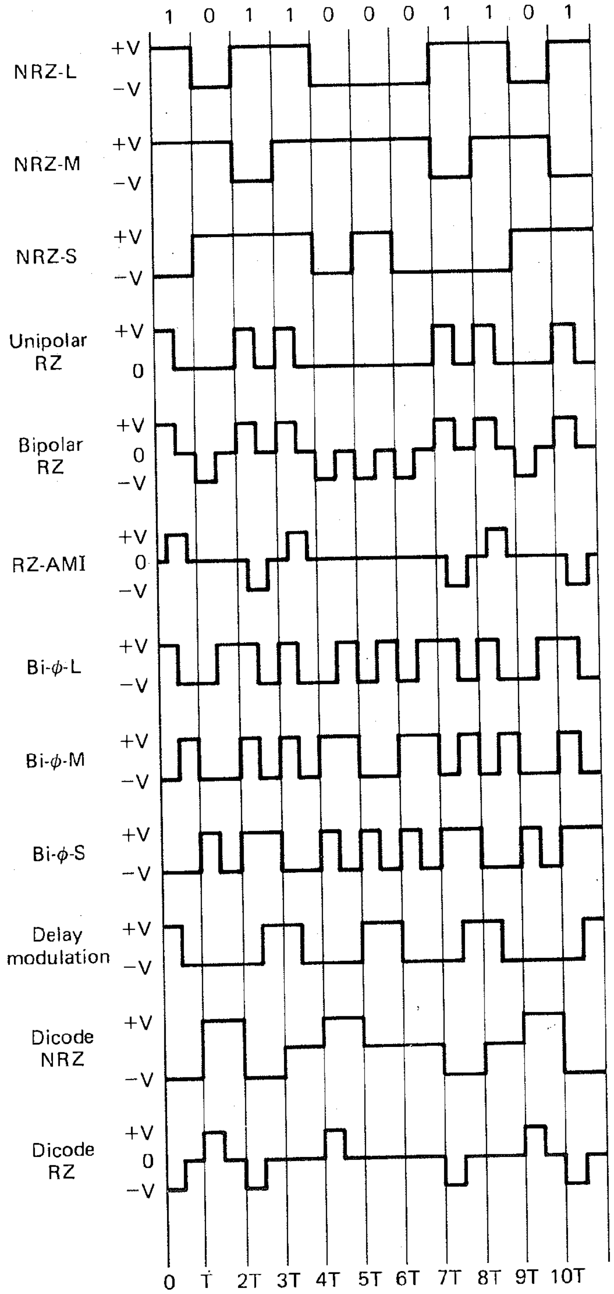


Figure 2.22 Various PCM waveforms.

present in each bit interval. RZ-AMI (AMI for "alternate mark inversion") is the coding scheme most often used in telemetry systems. The ones are represented by equal-amplitude alternating pulses. The zeros are represented by the absence of pulses.

The phase-encoded group consists of bi-ϕ-L (bi-phase-level), better known as *Manchester coding*; bi-ϕ-M (bi-phase-mark); bi-ϕ-S (bi-phase-space); and *delay modulation* (DM), or *Miller coding*. The phase-encoding schemes are used in magnetic recording systems and optical communications and in some satellite telemetry links. With bi-ϕ-L, a one is represented by a half-bit-wide pulse positioned during the first half of the bit interval; a zero is represented by a half-bit-wide pulse positioned during the second half of the bit interval. With bi-ϕ-M, a transition occurs at the beginning of every bit interval. A one is represented by a second transition one-half bit interval later; a zero is represented by no second transition. With bi-ϕ-S, a transition also occurs at the beginning of every bit interval. A one is represented by no second transition; a zero is represented by a second transition one-half bit interval later. With delay modulation [4], a one is represented by a transition at the midpoint of the bit interval. A zero is represented by no transition, unless it is followed by another zero. In this case, a transition is placed at the end of the bit interval of the first zero. Reference to the illustration in Figure 2.22 should help to make these descriptions clear.

Many binary waveforms use three levels, instead of two, to encode the binary data. Bipolar RZ and RZ-AMI belong to this group. The group also contains formats called *dicode* and *duobinary*. With dicode-NRZ, the one-to-zero or zero-to-one data transition changes the pulse polarity; without a data transition, the zero level is sent. With dicode-RZ, the one-to-zero or zero-to-one transition produces a half-duration polarity change; otherwise, a zero level is sent. The three-level duobinary signaling scheme is treated in Section 2.12.

One might ask why there are so many PCM waveforms. Are there really so many unique applications necessitating such a variety of waveforms to represent digits? The reason for the large selection relates to the differences in performance that characterize each waveform [5]. In choosing a coding scheme for a particular application, some of the parameters worth examining are the following:

1. *Dc component*. Eliminating the dc energy from the signal's power spectrum enables the system to be ac coupled. Magnetic recording systems, or systems using transformer coupling, have little sensitivity to very low frequency signal components. Thus low-frequency information could be lost.
2. *Self-Clocking*. Symbol or bit synchronization is required for any digital communication system. Some PCM coding schemes have inherent synchronizing or clocking features that aid in the recovery of the clock signal. For example, the Manchester code has a transition in the middle of every bit interval whether a one or a zero is being sent. This guaranteed transition provides a clocking signal.
3. *Error detection*. Some schemes, such as duobinary, provide the means of detecting data errors without introducing additional error-detection bits into the data sequence.
4. *Bandwidth compression*. Some schemes, such as multilevel codes, increase the efficiency of bandwidth utilization by allowing a reduction in required bandwidth for a given data rate; thus there is more information transmitted per unit bandwidth.

5. *Differential encoding*. This technique is useful because it allows the polarity of differentially encoded waveforms to be inverted without affecting the data detection. In communication systems where waveforms sometimes experience inversion, this is a great advantage. Differential encoding is treated in greater detail in Section 3.6.2.
6. *Noise immunity*. The various PCM waveform types can be further characterized by probability of bit error versus signal-to-noise ratio. Some of the schemes are more immune than others to noise. For example, the NRZ waveforms have better error performance than does the unipolar RZ waveform.

2.8.3 Spectral Attributes of PCM Waveforms

The most common criteria used for comparing PCM waveforms and for selecting one waveform type from the many available are: spectral characteristics, bit synchronization capabilities, error-detecting capabilities, interference and noise immunity, and cost and complexity of implementation. Figure 2.23 shows the spectral characteristics of some of the most popular PCM waveforms. The figure plots power spectral density in watts/hertz versus normalized bandwidth (frequency times pulse width). The spectral characteristic of a PCM waveform establishes the required system bandwidth and indicates how efficiently the bandwidth is being used. Bandwidth efficiency is addressed in detail in Chapter 7. The features that are easily observed in Figure 2.23 are the energy content at low frequency and the bandwidth requirements. Notice that the NRZ and duobinary schemes

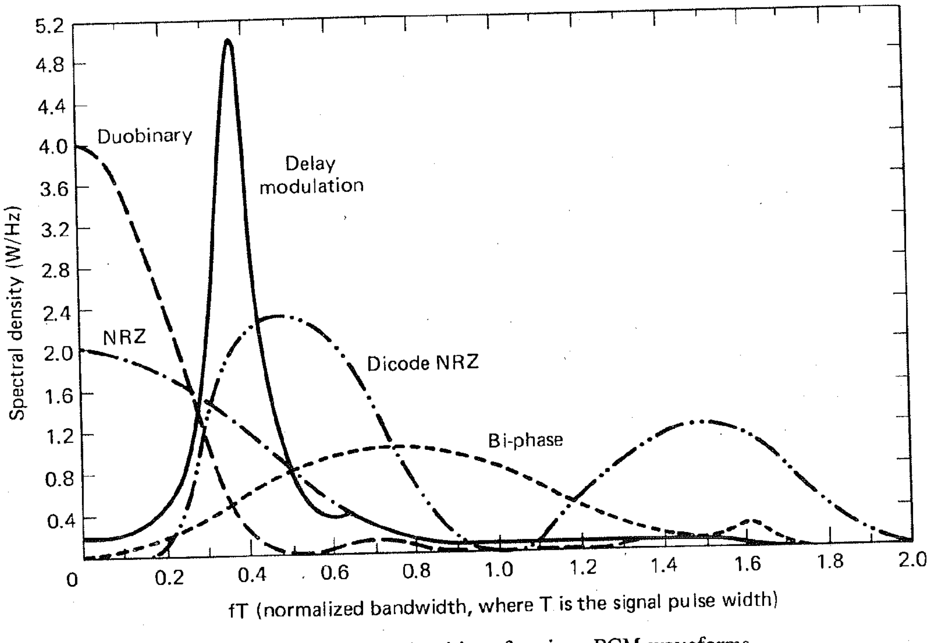


Figure 2.23 Spectral densities of various PCM waveforms.

have large spectral components at low frequency. Notice also that the bi-phase schemes have no energy at dc. However, bi-phasc requires a relatively large system bandwidth, as does the dicode scheme. The methods that are particularly bandwidth efficient are the duobinary and delay modulation. Duobinary signaling is treated in Scction 2.12.

2.9 DETECTION OF BINARY SIGNALS IN GAUSSIAN NOISE

Once the digital symbols are transformed into electrical waveforms, they can then be transmitted through the channel. During a given signaling interval, T, a binary system will transmit one of two waveforms, denoted $s_1(t)$ and $s_2(t)$. The transmitted signal over a symbol interval $(0, T)$ is represented by

$$s_i(t) = \begin{cases} s_1(t) & 0 \le t \le T \quad \text{for a binary 1} \\ s_2(t) & 0 \le t \le T \quad \text{for a binary 0} \end{cases}$$

The signal, $r(t)$, received by the receiver is represented by

$$r(t) = s_i(t) + n(t) \qquad i = 1, 2; \qquad 0 \le t \le T \tag{2.24}$$

where $n(t)$ is a zero-mean additive white Gaussian noise (AWGN) process.

Figure 2.24 highlights the *two separate* steps involved in signal detection. The *first step* consists of reducing the received waveform, $r(t)$ (whether baseband or bandpass), to a *single number*, $z(t = T)$. This operation can be performed by a linear filter followed by a sampler, as shown in block 1 of Figure 2.24, or optimally by a matched filter or correlator, which will be treated in later sections. The initial conditions of the filter or correlator are set to zero just prior to the arrival of each new symbol. At the end of a symbol duration, T, the output of block 1 yields the sample, $z(T)$, sometimes called the *test statistic*. We have assumed that the input noise is a Gaussian random process, and we have stated that the input filter is linear. A linear operation on a Gaussian random process will produce a second Gaussian random process [6]. Thus the filter output noise is Gaussian. If a nonlinear detector is used, the output noise will not be Gaussian

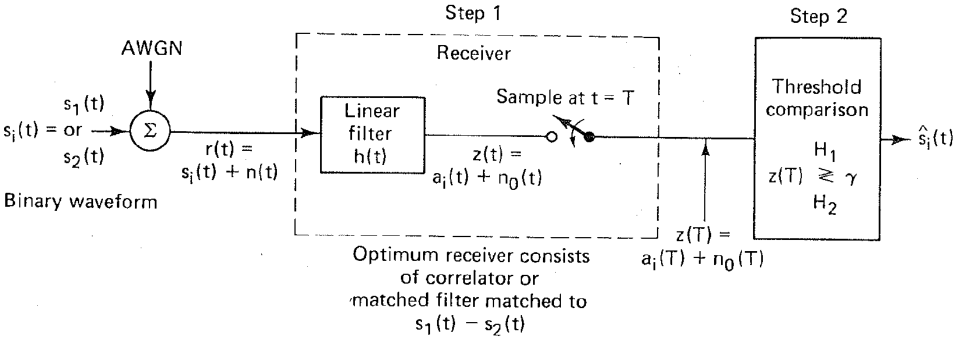


Figure 2.24 Two basic steps in digital signal detection.

and the following analysis will not apply. The output of block 1, sampled at $t = T$, yields

$$z(T) = a_i(T) + n_0(T) \qquad i = 1, 2 \tag{2.25}$$

where $a_i(T)$ is the signal component of $z(T)$ and $n_0(T)$ is the noise component. To shorten the notation, we sometimes write Equation (2.25) as $z = a_i + n_0$. The noise component, n_0, is a zero-mean *Gaussian random variable*, and thus $z(T)$ is a *Gaussian random variable* with a mean of either a_1 or a_2 depending on whether a binary one or binary zero was sent. The probability density function (pdf) of the Gaussian random noise, n_0, can be expressed as

$$p(n_0) = \frac{1}{\sigma_0\sqrt{2\pi}} \exp\left[-\frac{1}{2}\left(\frac{n_0}{\sigma_0}\right)^2\right] \tag{2.26}$$

where σ_0^2 is the noise variance. Thus it follows from Equations (2.25) and (2.26) that the conditional probability density functions (pdfs), $p(z|s_1)$ and $p(z|s_2)$ can be expressed as

$$p(z|s_1) = \frac{1}{\sigma_0\sqrt{2\pi}} \exp\left[-\frac{1}{2}\left(\frac{z - a_1}{\sigma_0}\right)^2\right] \tag{2.27}$$

$$p(z|s_2) = \frac{1}{\sigma_0\sqrt{2\pi}} \exp\left[-\frac{1}{2}\left(\frac{z - a_2}{\sigma_0}\right)^2\right] \tag{2.28}$$

These conditional pdfs are illustrated in Figure 2.25. The rightmost conditional pdf, $p(z|s_1)$, illustrates the probability density of the detector output, $z(T)$, given that $s_1(t)$ was transmitted. Similarly, the leftmost conditional pdf, $p(z|s_2)$, illustrates the probability density of $z(T)$ given that $s_2(t)$ was transmitted. The abscissa, $z(T)$, represents the full range of possible sample output values from block 1 of Figure 2.24.

The *second step* of the signal detection process consists of comparing the test statistic, $z(T)$, to a threshold level, γ, in block 2 of Figure 2.24, in order to estimate which signal, $s_1(t)$ or $s_2(t)$, has been transmitted. The filtering operation in block 1 does not depend on the decision criterion in block 2. Thus the choice of how best to implement block 1 can be independent of the particular decision strategy (choice of the threshold setting, γ).

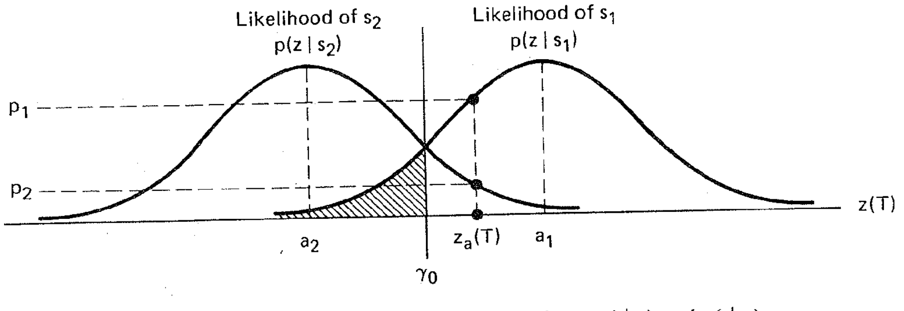


Figure 2.25 Conditional probability density functions: $p(z|s_1)$ and $p(z|s_2)$.

Once a received waveform, $r(t)$, is transformed to a number $z(T)$, the actual shape of the waveform is no longer important; all waveform types that are transformed to the same value of $z(T)$ are identical for detection purposes. We will see in Section 2.9.2 that a *matched filter* receiver in block 1 of Figure 2.24 is one that maps all signals of equal energy into the same point, $z(T)$. Therefore, the *signal energy* (not its shape) is the important paramcter in the detection process. Thus the detection analysis for *baseband signals* is the same as that for *bandpass signals*. The final step in block 2 is to make the decision

$$z(T) \underset{H_2}{\overset{H_1}{\gtrless}} \gamma \tag{2.29}$$

where H_1 and H_2 are the two possible (binary) hypotheses. Choosing H_1 is equivalent to deciding that signal $s_1(t)$ was sent, and choosing H_2 is equivalent to deciding that signal $s_2(t)$ was sent. The inequality relationship indicates that hypothesis H_1 is chosen if $z(T) > \gamma$, and hypothesis H_2 is chosen if $z(T) < \gamma$. If $z(T) = \gamma$, the decision can be an arbitrary one.

2.9.1 Maximum Likelihood Receiver Structure

A popular criterion for choosing the threshold level, γ, for the binary decision is based on minimizing the probability of error. The computation for this *minimum error* value of $\gamma = \gamma_0$ starts with forming an inequality expression between the ratio of conditional probability density functions and the signal a priori probabilities. The conditional density function, $p(z|s_i)$, is also called the *likelihood* of s_i. Thus thc formulation as shown below is called the *likelihood ratio test* (see Appendix B).

$$\frac{p(z|s_1)}{p(z|s_2)} \underset{H_2}{\overset{H_1}{\gtrless}} \frac{P(s_2)}{P(s_1)} \tag{2.30}$$

where $P(s_1)$ and $P(s_2)$ are the a priori probabilities that $s_1(t)$ and $s_2(t)$, respectively, are transmitted, and H_1 and H_2 are the two possible hypotheses. The rule for minimizing the error probability in Equation (2.30) states that we should choose hypothesis H_1 if the ratio of likelihoods is greater than the ratio of a priori probabilities.

It is shown in Section B.3.1 that if $P(s_1) = P(s_2)$, and if the likelihoods, $p(z|s_i)$ $(i = 1, 2)$, are symmetrical, the substitution of Equations (2.27) and (2.28) into (2.30) yields

$$z(T) \underset{H_2}{\overset{H_1}{\gtrless}} \frac{a_1 + a_2}{2} = \gamma_0 \tag{2.31}$$

where a_1 is the signal component of $z(T)$ when $s_1(t)$ is transmitted, and a_2 is the signal component of $z(T)$ when $s_2(t)$ is transmitted. The threshold lcvel, γ_0, represented by $(a_1 + a_2)/2$, is the *optimum threshold* for minimizing the probability of making an incorrect decision for this important special case. This strategy is known as the *minimum error criterion.*

For equally likely signals, the optimum threshold, γ_0, passes through the intersection of the likelihood functions, as shown in Figure 2.25. Thus by following Equation (2.31), the decision stage effectively selects the hypothesis that corresponds to the signal with the *maximum likelihood.* For example, given an arbitrary detector output value, $z_a(T)$, for which there is a nonzero likelihood that $z_a(T)$ belongs to either signal class $s_1(t)$ or $s_2(t)$, one can think of the likelihood test as a comparison of the likelihood values $p(z_a|s_1)$ and $p(z_a|s_2)$. The signal corresponding to the maximum pdf is chosen as the most likely to have been transmitted. In other words, the detector chooses $s_1(t)$ if

$$p(z_a|s_1) > p(z_a|s_2) \tag{2.32}$$

Otherwise, the detector chooses $s_2(t)$. A detector that minimizes the error probability (for the case where the signal classes are equally likely) is also known as a *maximum likelihood detector.*

Figure 2.25 illustrates that Equation (2.32) is just a "common sense" way to make a decision when there exists statistical knowledge of the classes. Given the detector output value, $z_a(T)$, we see in Figure 2.25 that $z_a(T)$ intersects the likelihood of $s_1(t)$ at a value p_1, and it intersects the likelihood of $s_2(t)$ at a value p_2. What is the most reasonable decision for the detector to make? For this example, choosing class $s_1(t)$, which has the greater likelihood, is the most sensible choice. If this was an M-ary instead of a binary example, there would be a total of M likelihood functions representing the M signal classes to which a received signal might belong. The maximum likelihood decision would then be to choose the class that had the greatest likelihood of all M likelihoods. Refer to Appendix B for a review of decision theory fundamentals.

2.9.1.1 Error Probability

For the binary example in Figure 2.25, there are two ways in which errors can occur. An error, e, will occur when $s_1(t)$ is sent, and channel noise results in the receiver output signal, $z(T)$, being less than γ_0. The probability of such an occurrence is

$$P(e|s_1) = P(H_2|s_1) = \int_{-\infty}^{\gamma_0} p(z|s_1)\, dz \tag{2.33}$$

This is illustrated by the shaded area to the left of γ_0 in Figure 2.25. Similarly, an error occurs when $s_2(t)$ is sent, and the channel noise results in $z(T)$ being greater than γ_0. The probability of this occurrence is

$$P(e|s_2) = P(H_1|s_2) = \int_{\gamma_0}^{\infty} p(z|s_2)\, dz \tag{2.34}$$

The probability of an error is the sum of the probabilities of all the ways that an error can occur. For the binary case, we can express the probability of bit error, P_B, as follows:

$$P_B = \sum_{i=1}^{2} P(e, s_i) \tag{2.35}$$

Combining Equations (2.33) to (2.35), we can write

$$P_B = P(e|s_1)P(s_1) + P(e|s_2)P(s_2) \tag{2.36a}$$

or equivalently,

$$P_B = P(H_2|s_1)P(s_1) + P(H_1|s_2)P(s_2) \tag{2.36b}$$

That is, given that signal $s_1(t)$ was transmitted, an error results if hypothesis H_2 is chosen; or given that signal $s_2(t)$ was transmitted, an error results if hypothesis H_1 is chosen. For the case where the a priori probabilities are equal, that is, $P(s_1) = P(s_2) = \frac{1}{2}$,

$$P_B = \tfrac{1}{2}P(H_2|s_1) + \tfrac{1}{2}P(H_1|s_2) \tag{2.37}$$

and because of the symmetry of the probability density functions

$$P_B = P(H_2|s_1) = P(H_1|s_2) \tag{2.38}$$

The probability of a bit error, P_B, is numerically equal to the area under the "tail" of either likelihood function, $p(z|s_1)$ or $p(z|s_2)$, falling on the "incorrect" side of the threshold. We can therefore compute P_B by integrating $p(z|s_1)$ between the limits $-\infty$ and γ_0, or as shown below, by integrating $p(z|s_2)$ between the limits γ_0 and ∞:

$$P_B = \int_{\gamma_0=(a_1+a_2)/2}^{\infty} p(z|s_2)\, dz \tag{2.39}$$

where $\gamma_0 = (a_1 + a_2)/2$ is the optimum threshold from Equation (2.31). Replacing the likelihood $p(z|s_2)$ with its Gaussian equivalent from Equation (2.28), we have

$$P_B = \int_{\gamma_0=(a_1+a_2)/2}^{\infty} \frac{1}{\sigma_0\sqrt{2\pi}} \exp\left[-\frac{1}{2}\left(\frac{z-a_2}{\sigma_0}\right)^2\right] dz \tag{2.40}$$

where σ_0^2 is the variance of the noise out of the correlator.

Let $u = (z - a_2)/\sigma_0$. Then $\sigma_0\, du = dz$ and

$$P_B = \int_{u=(a_1-a_2)/2\sigma_0}^{u=\infty} \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{u^2}{2}\right) du = Q\left(\frac{a_1-a_2}{2\sigma_0}\right) \tag{2.41}$$

where $Q(x)$, called the *complementary error function* or *co-error function*, is a commonly used symbol for the probability under the tail of the Gaussian distribution. It is defined as

$$Q(x) = \frac{1}{\sqrt{2\pi}} \int_x^{\infty} \exp\left(-\frac{u^2}{2}\right) du \tag{2.42}$$

Note that the co-error function is defined in several ways (see Appendix B); however, all definitions are essentially equivalent. $Q(x)$ cannot be evaluated in closed form. It is presented in tabular form in Table B.1. Good approximations to $Q(x)$ by simpler functions can be found in Reference [7]. One such approximation, valid for $x > 3$, is

$$Q(x) \simeq \frac{1}{x\sqrt{2\pi}} \exp\left(-\frac{x^2}{2}\right) \tag{2.43}$$

We have optimized (in the sense of minimizing P_B) the threshold level, γ, but have not optimized the filter in block 1 of Figure 2.24; we next consider optimizing this filter by maximizing the argument of $Q(x)$ in Equation (2.41).

2.9.2 The Matched Filter

A matched filter is a linear filter designed to provide the maximum signal-to-noise power ratio at its output for a given transmitted symbol waveform. Consider that a known signal $s(t)$ plus AWGN, $n(t)$, is the input to a linear, time-invariant filter followed by a sampler, as shown in Figure 2.24. At time $t = T$, the receiver output, $z(T)$, consists of a signal component, a_i, and a noise component, n_0. The variance of the output noise (average noise power) is denoted by σ_0^2, so that the ratio of the instantaneous signal power to average noise power, $(S/N)_T$, at time $t = T$, out of the receiver in block 1, is

$$\left(\frac{S}{N}\right)_T = \frac{a_i^2}{\sigma_0^2} \tag{2.44}$$

We wish to find the filter transfer function, $H_0(f)$, that *maximizes* Equation (2.44). We can express the signal, $a(t)$, at the filter output, in terms of the filter transfer function, $H(f)$ (before optimization), and the Fourier transform of the input signal, as follows:

$$a(t) = \int_{-\infty}^{\infty} H(f)S(f)e^{j2\pi ft}\, df \tag{2.45}$$

where $S(f)$ is the Fourier transform of the input signal, $s(t)$. If the two-sided power spectral density of the input noise is $N_0/2$ watts/hertz, then using Equations (1.19) and (1.53), we can express the output noise power, σ_0^2, as

$$\sigma_0^2 = \frac{N_0}{2} \int_{-\infty}^{\infty} |H(f)|^2\, df \tag{2.46}$$

We then combine Equations (2.44) to (2.46) to express $(S/N)_T$, as follows:

$$\left(\frac{S}{N}\right)_T = \frac{\left| \int_{-\infty}^{\infty} H(f)S(f)e^{j2\pi fT}\, df \right|^2}{N_0/2 \int_{-\infty}^{\infty} |H(f)|^2\, df} \tag{2.47}$$

We next find that value of $H(f) = H_0(f)$ for which the maximum $(S/N)_T$ is achieved, by using *Schwarz's inequality*. One form of the inequality can be stated as

$$\left| \int_{-\infty}^{\infty} f_1(x) f_2(x)\, dx \right|^2 \leq \int_{-\infty}^{\infty} |f_1(x)|^2\, dx \int_{-\infty}^{\infty} |f_2(x)|^2\, dx \tag{2.48}$$

The equality holds if $f_1(x) = kf_2^*(x)$, where k is an arbitrary constant and * indicates complex conjugate. If we identify $H(f)$ with $f_1(x)$ and $S(f)\, e^{j2\pi fT}$ with $f_2(x)$, we can write

$$\left| \int_{-\infty}^{\infty} H(f) S(f) e^{j2\pi fT}\, df \right|^2 \leq \int_{-\infty}^{\infty} |H(f)|^2\, df \int_{-\infty}^{\infty} |S(f)|^2\, df \tag{2.49}$$

Substituting into Equation (2.47) yields

$$\left(\frac{S}{N}\right)_T \leq \frac{2}{N_0} \int_{-\infty}^{\infty} |S(f)|^2\, df \tag{2.50}$$

or

$$\max \left(\frac{S}{N}\right)_T = \frac{2E}{N_0} \tag{2.51}$$

where the energy, E, of the input signal $s(t)$ is

$$E = \int_{-\infty}^{\infty} |S(f)|^2\, df \tag{2.52}$$

Thus the maximum output $(S/N)_T$ depends on the input *signal energy* and the power spectral density of the noise, *not on the particular shape* of the waveform that is used.

The equality in Equation (2.51) holds only if the optimum filter transfer function, $H_0(f)$, is employed, such that

$$H(f) = H_0(f) = kS^*(f) e^{-j2\pi fT} \tag{2.53}$$

or

$$h(t) = \mathcal{F}^{-1}\{kS^*(f) e^{-j2\pi fT}\} \tag{2.54}$$

Since $s(t)$ is a real-valued signal, we can write from Equations (A.29) and (A.31),

$$h(t) = \begin{cases} ks(T - t) & 0 \leq t \leq T \\ 0 & \text{elsewhere} \end{cases} \tag{2.55}$$

Thus the impulse response of a filter that produces the maximum output signal-to-noise ratio is the mirror image of the message signal, $s(t)$, *delayed* by the symbol time duration, T. Note that the delay of T seconds makes Equation (2.55) *causal*; that is, the delay of T seconds makes $h(t)$ a function of positive time in the interval $0 \leq t \leq T$. Without the delay of T seconds, the response, $s(-t)$, is unrealizable because it describes a response as a function of negative time.

2.9.3 Correlation Realization of the Matched Filter

The term *matched filter* is often used synonymously with *product integrator* or *correlator*. Equation (2.55) and Figure 2.26a illustrate the matched filter's basic property: The impulse response of the filter is a delayed version of the mirror image (rotated on the $t = 0$ axis) of the signal waveform. Therefore, if the signal waveform is $s(t)$, its mirror image is $s(-t)$, and the mirror image delayed by T seconds is $s(T - t)$. The output, $z(t)$, of a causal filter can be described in the time domain as the convolution of a received input waveform, $r(t)$, with the impulse response of the filter (see Section A.5):

$$z(t) = r(t) * h(t) = \int_0^t r(\tau)h(t - \tau)\, d\tau \tag{2.56}$$

Substituting $h(t)$ of Equation (2.55) into $h(t - \tau)$ of Equation (2.56) and arbitrarily setting the constant k equal to unity, we get

$$\begin{aligned} z(t) &= \int_0^t r(\tau)s[T - (t - \tau)]\, d\tau \\ &= \int_0^t r(\tau)s(T - t + \tau)\, d\tau \end{aligned} \tag{2.57}$$

When $t = T$, we can write Equation (2.57) as

$$z(T) = \int_0^T r(\tau)s(\tau)\, d\tau \tag{2.58}$$

The operation of Equation (2.58), the product integration of the received signal, $r(t)$, with a replica of the transmitted waveform, $s(t)$, over one symbol interval is known as the *correlation* of $r(t)$ with $s(t)$. Consider that a received signal, $r(t)$, is correlated with each prototype signal, $s_i(t)$ ($i = 1, \ldots, M$), using a bank of M correlators. The signal $s_i(t)$ whose product integration or correlation with $r(t)$ yields the maximum output $z_i(T)$ is the signal that matches $r(t)$ better than all the other $s_j(t), j \neq i$. We will subsequently use this correlation characteristic for the optimum detection of signals.

2.9.3.1 Comparison of Convolution and Correlation

It is important to note that the correlator output and the matched filter output are the same *only at time $t = T$*. For a sine-wave input, the output of the correlator, $z(t)$, is approximately a linear ramp for $0 \leq t \leq T$. However, the matched filter output is approximately a sine-wave amplitude modulated by a linear ramp for $0 \leq t \leq T$. The comparison is shown in Figure 2.26b. To understand the similarities and differences between a matched filter and a product integrator, one might first ask: What are the similarities between *convolution* as expressed in Equation (2.56) and *correlation* as expressed in Equation (2.58)? With correlation, we simply multiply two functions together and integrate (compute the area under their product curve). We are calculating how closely two waveforms *match each other* in a given time period. With convolution, we sweep (step) two functions past one

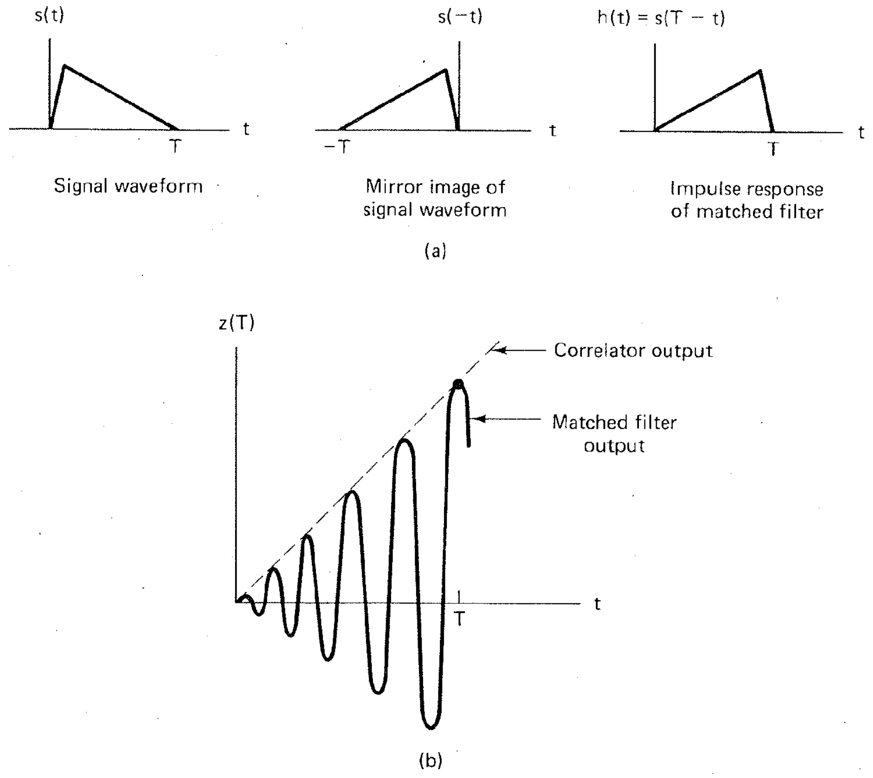


Figure 2.26 Correlator and matched filter. (a) Matched filter characteristic. (b) Comparison of correlator and matched filter outputs.

another and calculate a sequence of correlations (one for each step). The matched filter, used as a demodulator, only utilizes the correlation made at the symbol duration, T. Since the matched filter output and the correlator output are identical at the sampling time $t = T$, the matched filter and correlator functions, pictured in Figure 2.27, are used interchangeably.

2.9.4 Application of the Matched Filter

In Equation (2.41) we found that the optimum decision threshold resulted in $P_B = Q[(a_1 - a_2)/2\sigma_0]$. Finding the optimum threshold alone is not sufficient to optimize the detection process. To minimize P_B, we also need to select an optimum filter to maximize the argument of $Q(x)$. Thus we need to determine the linear filter that maximizes $(a_1 - a_2)/2\sigma_0$, or equivalently, that maximizes

$$\frac{(a_1 - a_2)^2}{\sigma_0^2} \tag{2.59}$$

where $(a_1 - a_2)$ is the difference of the signal components at the filter output, at time $t = T$, and the square of this difference signal is the instantaneous power

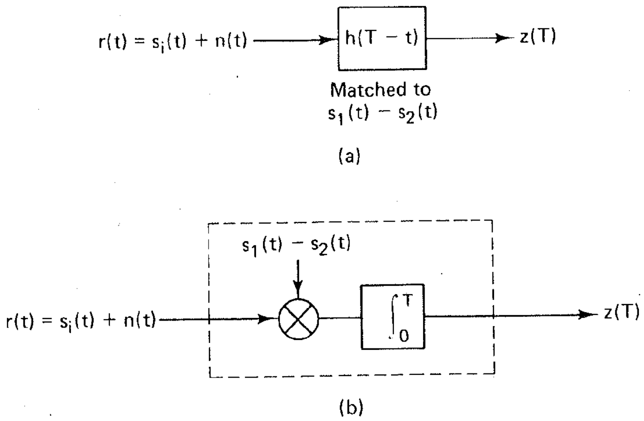


Figure 2.27 Equivalence of matched filter and correlator. (a) Matched filter. (b) Correlator.

of the difference signal. In Section 2.9.2 we described a filter that maximizes the output signal-to-noise ratio—the matched filter. Consider a filter that is *matched* to the input difference signal $[s_1(t) - s_2(t)]$. From Equations (2.44) and (2.51), the ratio of the instantaneous signal power to average noise power, $(S/N)_T$, at time $t = T$ out of this matched filter can be expressed as

$$\left(\frac{S}{N}\right)_T = \frac{(a_1 - a_2)^2}{\sigma_0^2} = \frac{2E_d}{N_0} \tag{2.60}$$

where $N_0/2$ is the two-sided power spectral density of the noise at thc filter input, and E_d is the energy of the difference signal at the filter input:

$$E_d = \int_0^T [s_1(t) - s_2(t)]^2 \, dt \tag{2.61}$$

Thus, using Equations (2.41) and (2.60), we have

$$P_B = Q\left(\sqrt{\frac{E_d}{2N_0}}\right) \tag{2.62}$$

2.9.5 Error Probability Performance of Binary Signaling

2.9.5.1 Unipolar Signaling

Figure 2.28a illustrates an example of a baseband waveform used for unipolar signaling where

$$\begin{aligned} s_1(t) &= A \qquad 0 \le t \le T \qquad \text{for binary 1} \\ s_2(t) &= 0 \qquad 0 \le t \le T \qquad \text{for binary 0} \end{aligned} \tag{2.63}$$

where $A > 0$ is the amplitude of signal $s_1(t)$. Assume that the unipolar signal plus white Gaussian noise is present at the input of a matched filter, with sampling time $t = T$. The correlator detector for such a signal type is shown in Figurc 2.28b. The correlator multiplies and integrates the incoming signal, $r(t)$, with the

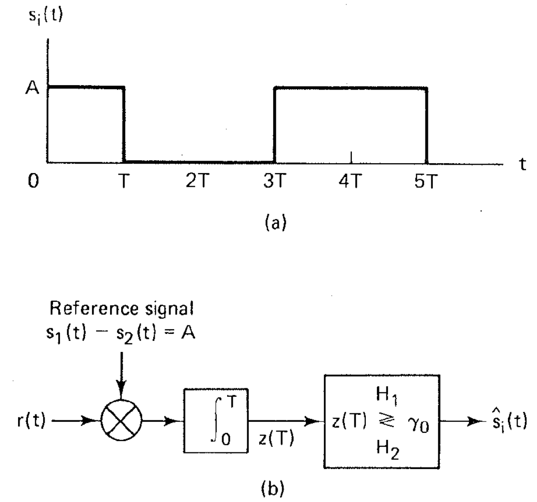


Figure 2.28 Detection of unipolar baseband signaling. (a) Unipolar signaling example. (b) Correlator detector.

difference of the prototype signals $[s_1(t) - s_2(t)] = A$, and after a symbol duration, T, compares the result, $z(T)$, with the threshold, γ_0. When $r(t) = s_1(t) + n(t)$, the signal component, $a_1(T)$, of $z(T)$ is found, using Equation (2.58), to be

$$a_1(T) = \mathbf{E}\{z(T)\} = \mathbf{E}\left\{ \int_0^T A^2 + An(t)\, dt \right\} = A^2T$$

where $\mathbf{E}\{\cdot\}$ is the *expected value operator*. This follows since $\mathbf{E}\{n(t)\} = 0$. Similarly, when $r(t) = s_2(t) + n(t)$, then $a_2(T) = 0$. Thus the optimum threshold is $\gamma_0 = (a_1 + a_2)/2 = \frac{1}{2}A^2T$. If the correlator output, $z(T)$, is greater than γ_0, the signal is declared to be $s_1(t)$; otherwise, it is declared to be $s_2(t)$.

The energy difference signal, from Equation (2.61), is $E_d = A^2T$. Then the bit error performance at the output is obtained from Equation (2.62) as follows:

$$P_B = Q\left(\sqrt{\frac{E_d}{2N_0}}\right) = Q\left(\sqrt{\frac{A^2T}{2N_0}}\right) = Q\left(\sqrt{\frac{E_b}{N_0}}\right) \tag{2.64}$$

where the average energy per bit is $E_b = A^2T/2$.

2.9.5.2 Bipolar Signaling

Figure (2.29a) illustrates an example of a bipolar baseband waveform, where

$$\begin{aligned} s_1(t) &= +A \quad & 0 \le t \le T \quad & \text{for binary 1} \\ s_2(t) &= -A \quad & 0 \le t \le T \quad & \text{for binary 0} \end{aligned} \tag{2.65}$$

Binary waveforms that are the negative of one another, such as the bipolar pair above, where $s_1(t) = -s_2(t)$, are called *antipodal signals*. A correlator receiver for this antipodal type of waveform can be configured as shown in Figure 2.29b. One correlator multiplies and integrates the incoming signal $r(t)$ with the prototype signal, $s_1(t)$; the second correlator multiplies and integrates $r(t)$ with $s_2(t)$. The

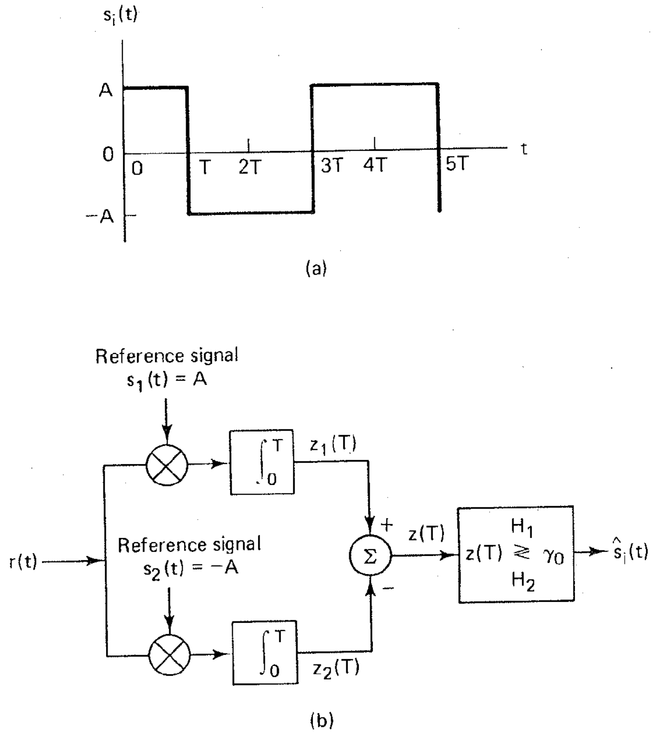


Figure 2.29 Detection of bipolar baseband signaling. (a) Bipolar signaling example. (b) Correlator detector.

correlator outputs are designated $z_i(T)$ $(i = 1, 2)$. The point in the decision space, $z(T)$, is formed from the difference of the correlator outputs, as follows:

$$z(T) = z_1(T) - z_2(T) \tag{2.66}$$

and the decision is made according to Equation (2.31). For antipodal signals, $a_1 = -a_2$; therefore, $\gamma_0 = 0$. Thus if the *test statistic*, $z(T)$, is positive, the signal is declared to be $s_1(t)$, and if it is negative, it is declared to be $s_2(t)$.

The energy difference signal, $E_d = (2A)^2T$. Then the bit error performance from Equation (2.62) is

$$P_B = Q\left(\sqrt{\frac{2A^2T}{N_0}}\right) = Q\left(\sqrt{\frac{2E_b}{N_0}}\right) \tag{2.67}$$

where the average energy per bit is $E_b = A^2T$. Figure 2.30 illustrates curves of P_B versus E_b/N_0 for unipolar and bipolar signaling. In examining the two curves, we can see a 3-dB error performance improvement for bipolar compared to unipolar signaling. This difference could have been predicted by the factor-of-2 difference in the coefficient of E_b in Equation (2.67) compared with Equation (2.64). In Chapter 3 we shall see that the error performance of *bandpass antipodal signaling* (e.g., coherently detected binary phase shift keying) is the same as that for *baseband antipodal signaling* (matched filter reception). Also, we shall see that the error performance of *bandpass orthogonal signaling* (e.g., coherently detected

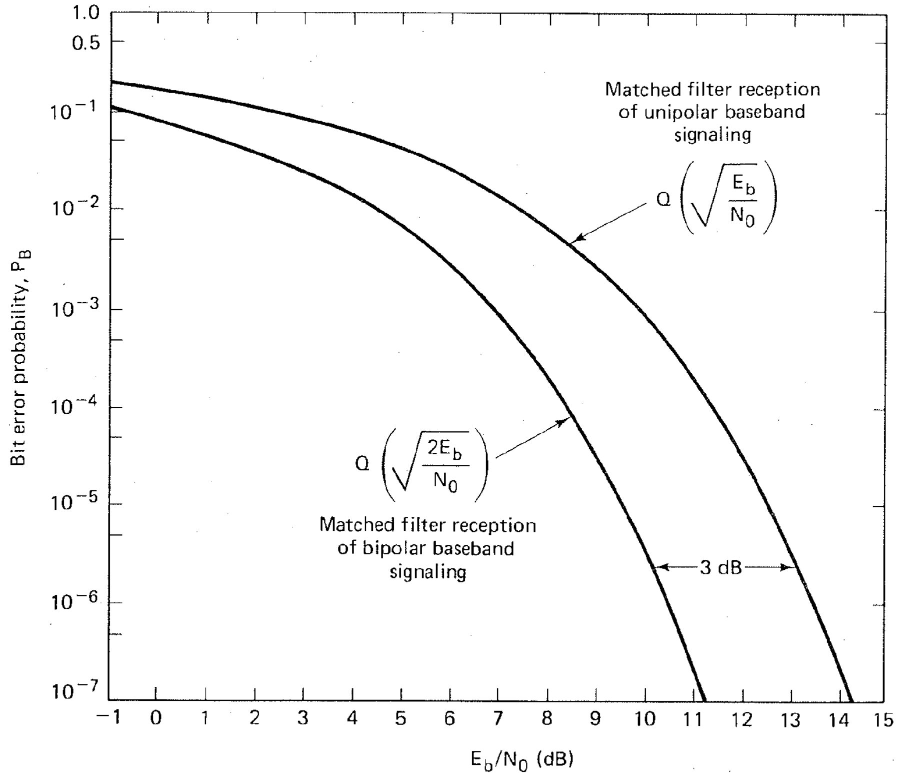


Figure 2.30 Bit error performance of unipolar and bipolar signaling.

frequency shift keying) is the same as that for *baseband unipolar signaling* (matched filter reception).

2.10 MULTILEVEL BASEBAND TRANSMISSION

The system bandwidth required for binary PCM signaling may be very large. What might we do to reduce the required bandwidth? One possibility is to use *multilevel signaling*. Consider a binary PCM bit stream with data rate R bits per second. Instead of transmitting a pulse waveform for each bit, we first partition the data into k-bit groups. We then use $M = 2^k$-level pulses for transmission. Each pulse waveform can now represent a k-bit symbol in a symbol stream of rate R/k symbols per second. Thus multilevel signaling, where $M > 2$, can be used to reduce the number of symbols transmitted per second, or thus to reduce the bandwidth requirements of the channel. Is there a price to be paid for such bandwidth reduction? Of course there is; it is discussed below.

Consider the task that the pulse receiver must perform; it needs to distinguish between the possible levels of each pulse. Can the receiver distinguish among the eight possible levels of each octal pulse in Figure 2.31a as easily as it can distin-

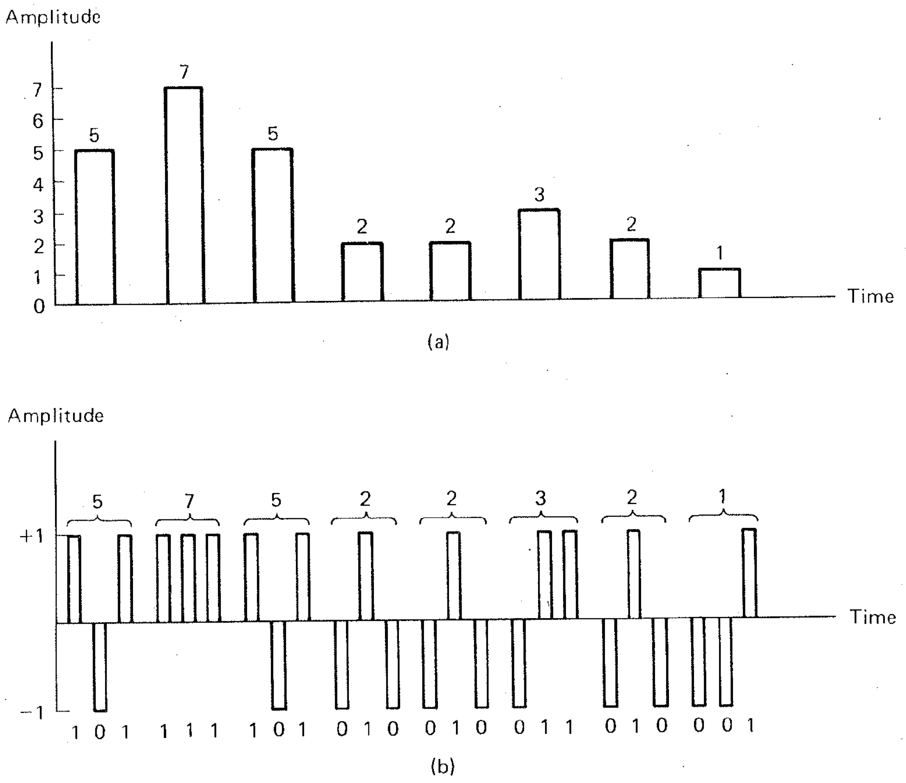


Figure 2.31 Pulse code modulation signaling. (a) Eight-level signaling. (b) Two-level signaling.

guish between the two possible levels of each binary pulse in Figure 2.31b? The transmission of an 8-level (compared to a 2-level) pulse requires a greater amount of energy for equivalent detection performance. (It is the amount of signal energy that determines how reliably a signal will be detected.) For equal average power in the binary and the octal pulses, it is easier to detect the binary pulses because the detector has more signal energy per level for making a binary decision than an 8-level decision. What price does a system designer pay if he or she chooses the transmission waveform to be the easier-to-detect binary PCM, rather than eight-level PCM? The engineer pays the price of needing three times as much system bandwidth for a given data rate, compared to the octal pulses, since each octal pulse must be replaced with three binary pulses (each one-third as wide as the octal pulses). One might ask: Why not use binary pulses with the same pulse duration as the original octal pulses, and suffer the information delay? For some cases this might be appropriate, but for most communication systems, such an increase in delay cannot be tolerated; the six o'clock news *must* be received at six o'clock. In Chapter 7 we examine in detail the trade-off between signal power and system bandwidth.

2.10.1 PCM Word Size

How many bits shall we assign to each analog sample? For digital telephone channels, each speech sample is PCM encoded using 8 bits, yielding 2^8 or 256 levels per sample. The choice of the number of levels, or bits per sample, depends on how much distortion we are willing to tolerate with the PCM format. It is useful to develop a general relationship between the required number of bits per analog sample (the PCM word size) and the allowable quantization distortion. Let the magnitude of the quantization distortion error, $|e|$, be specified not to exceed a fraction, p, of the peak-to-peak analog voltage, V_{pp}, as follows:

$$|e| \leq pV_{\text{pp}} \tag{2.68}$$

Since the quantization error can be no larger than $q/2$, where q is the quantile interval, we can write

$$|e|_{\text{max}} = \frac{q}{2} = \frac{V_{\text{pp}}}{2L} \tag{2.69}$$

where L is the number of quantization levels. Then

$$\frac{V_{\text{pp}}}{2L} \leq pV_{\text{pp}} \tag{2.70}$$

$$2^{\ell} = L \geq \frac{1}{2p} \text{ levels} \tag{2.71}$$

$$\ell \geq \log_2 \frac{1}{2p} \text{ bits} \tag{2.72}$$

It is important that we do not confuse the idea of bits per PCM word, denoted by ℓ in Equation (2.72), with the M-level transmission concept of k data bits per symbol. The following example should clarify the distinction.

Example 2.3 Quantization Levels and Multilevel Signaling

The information in an analog waveform, with maximum frequency $f_m = 3$ kHz, is to be transmitted over an M-level PCM system, where the number of pulse levels is $M = 16$. The quantization distortion is specified not to exceed $\pm 1\%$ of the peak-to-peak analog signal.

(a) What is the minimum number of bits/sample, or bits/PCM word, that should be used in this PCM system?

(b) What is the minimum required sampling rate, and what is the resulting bit transmission rate?

(c) What is the PCM pulse or symbol transmission rate?

In this example we are concerned with two types of *levels*: the number of quantization levels for fulfilling the distortion requirement, and the 16 levels of the multilevel PCM pulses.

Solution

(a) Using Equation (2.72), we calculate

$$\ell \geq \log_2 \frac{1}{0.02} = \log_2 50 \simeq 5.6$$

Therefore, use $\ell = 6$ bits/sample to meet the distortion requirement.

(b) Using the Nyquist sampling criterion, the minimum sampling rate $f_s = 2f_m =$ 6000 samples/second (samples/s). From part (a), each sample will give rise to a PCM word composed of 6 bits. Therefore, the bit transmission rate $R = \ell f_s =$ 36,000 bits/s.

(c) Since multilevel pulses are to be used with $M = 2^k = 16$ levels, $k = \log_2 16 =$ 4 bits/symbol. Therefore, the bit stream will be partitioned into groups of 4 bits to form the new 16-level PCM digits, and the resulting symbol transmission rate R_s is $R/k = 36{,}000/4 = 9000$ symbols/s.

2.11 INTERSYMBOL INTERFERENCE

Figure 2.32a highlights the major filtering aspects of a typical baseband digital system; there are circuit reactances throughout the system—in the transmitter, in the receiver, and in the channel. The pulses at the input might be impulse-like samples, or flat-top samples. In either case, they are low-pass filtered at the transmitter to confine them to some desired bandwidth. Channel reactances can cause amplitude and phase variations that distort the pulses. The receiving filter, called the *equalizing filter*, should be configured to compensate for the distortion

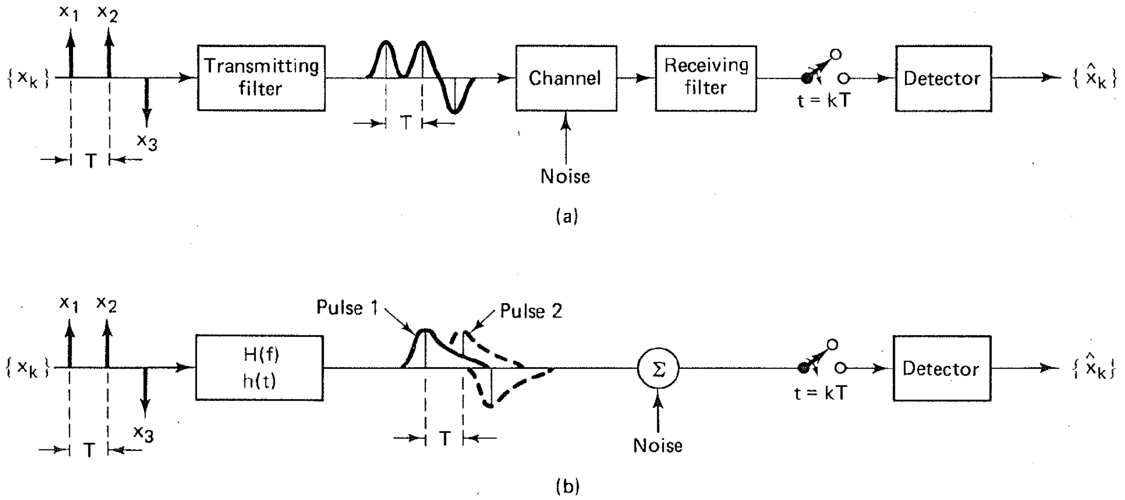


Figure 2.32 Intersymbol interference in the detection process. (a) Typical baseband digital system. (b) Equivalent model.

caused by the transmitter and the channel [8]. In a binary system with a commonly used PCM format, such as NRZ-L, the detector makes symbol decisions by comparing the received bipolar pulses to a threshold; for example, the detector decides that a binary one was sent if the received pulse is positive, and that a binary zero was sent if the received pulse is negative. Figure 2.32b illustrates a convenient model for the system, lumping all the filtering effects into one overall equivalent system transfer function, $H(f)$:

$$H(f) = H_t(f)H_c(f)H_r(f) \tag{2.73}$$

where $H_t(f)$ characterizes the transmitting filter, $H_c(f)$ the filtering within the channel, and $H_r(f)$ the receiving or equalizing filter. The characteristic $H(f)$, then, represents the composite system transfer function due to all of the filtering at various locations throughout the transmitter/channel/receiver chain. Due to the effects of system filtering, the received pulses overlap one another as shown in Figure 2.32b; the tail of one pulse "smears" into adjacent symbol intervals so as to interfere with the detection process; such interference is termed *intersymbol interference* (ISI). Even in the absence of noise, imperfect filtering and system bandwidth constraints lead to ISI. In practice, $H_c(f)$ is usually specified, and the problem remains to determine $H_t(f)$ and $H_r(f)$ such that the ISI of the pulses are minimized at the output of $H_r(f)$.

Nyquist [9] investigated the problem of specifying a received pulse shape so that no ISI occurs at the detector. He showed that the theoretical minimum system bandwidth needed to detect R_s symbols/s, without ISI, is $R_s/2$ hertz. This occurs when the system transfer function, $H(f)$, is made rectangular, as shown in Figure 2.33a. When $H(f)$ is such an ideal filter with bandwidth $1/2T$, its impulse response, the inverse Fourier transform of $H(f)$ (from Table A.1) is $h(t) = \text{sinc}\,(t/T)$, shown in Figure 2.33b. Thus $h(t)$ is the received pulse shape resulting from the application of an impulse at the input of such an ideal system. Nyquist established that if each pulse of a received sequence is of the form $h(t)$, the pulses can be detected without ISI. The bandwidth required to detect $1/T$ such pulses (symbols) per second is equal to $1/2T$; in other words, a system with bandwidth $W = 1/2T = R_s/2$ hertz can support a maximum transmission rate of $2W = 1/T = R_s$ symbols/s (*Nyquist bandwidth constraint*) without ISI. Figure

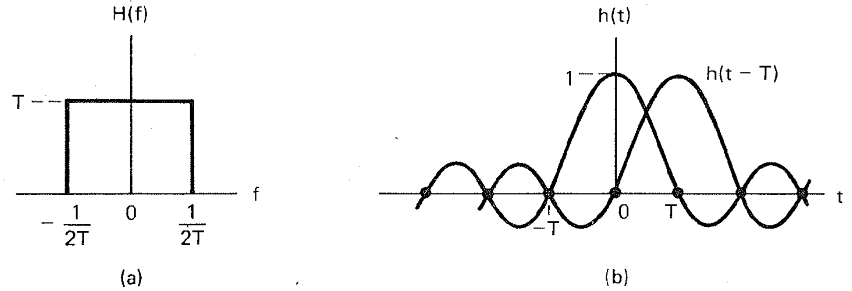


Figure 2.33 Nyquist channels for zero ISI. (a) Rectangular system transfer function $H(f)$. (b) Received pulse shape $h(t) = \text{sinc}\,(t/T)$.

2.33b illustrates how ISI is avoided. The figure shows two successive received pulses, $h(t)$ and $h(t - T)$. Even though $h(t)$ has a long tail, it passes through zero at the instant that $h(t - T)$ is sampled (at $t = T$) and therefore causes no degradation to the detection process. With such an ideal received pulse shape, the maximum possible symbol transmission rate per hertz, called the *symbol-rate packing*, is 2 symbols/s/Hz, without ISI.

What does the Nyquist bandwidth constraint say about the maximum number of *bits*/s/Hz that can be received without ISI? It says nothing about bits, directly. The constraint deals only with pulses or symbols, and the ability to detect their amplitude values without distortion from other pulses. The assignment of how many bits each symbol represents is a separate issue. In theory, each symbol can represent M levels or k bits ($M = 2^k$); as k or M increases in value, so does the complexity of the system. For example, when $k = 6$ bits/symbol, each symbol represents $M = 64$ levels. The number of bits/s/Hz that a system can support is referred to as the *bandwidth efficiency* of the system; this subject is treated separately in Chapter 7.

For most communication systems (with the exception of spread-spectrum systems, covered in Chapter 10), our goal is to reduce the required system bandwidth as much as possible; Nyquist has provided us with a basic limitation to such bandwidth reduction. What would happen if we tried to force a system to operate at smaller bandwidths than the constraint dictates? We would find that restricting the bandwidth would spread the pulses in time; this would degrade the system's error performance, due to the increase in ISI.

2.11.1 Pulse Shaping to Reduce ISI

The Nyquist requirement for a sinc (t/T) received pulse shape is not physically realizable since it dictates a rectangular bandwidth characteristic and an infinite time delay. Also, with such a characteristic, the detection process would be very sensitive to small timing errors. In Figure 2.33b the pulse $h(t)$ has zero value in adjacent pulse times *only* when the sampling is performed at exactly the correct sampling time; timing errors will produce ISI. Therefore, we cannot implement systems using the Nyquist bandwidth; we need to provide some "excess bandwidth" beyond the theoretical minimum. One frequently used system transfer function, $H(f)$, is called the *raised cosine filter*. It can be expressed as

$$H(f) = \begin{cases} 1 & \text{for } |f| < 2W_0 - W \\ \cos^2\left(\dfrac{\pi}{4}\,\dfrac{|f| + W - 2W_0}{W - W_0}\right) & \text{for } 2W_0 - W < |f| < W \\ 0 & \text{for } |f| > W \end{cases} \tag{2.74}$$

where W is the absolute bandwidth, and $W_0 = 1/2T$ represents the minimum Nyquist bandwidth for the rectangular spectrum and the -6-dB bandwidth (or half-amplitude point) for the raised cosine spectrum. The difference $(W - W_0)$ is termed the *excess bandwidth*; notice that $W = W_0$ for the rectangular spectrum.

The *roll-off factor* is defined to be $r = (W - W_0)/W_0$. It represents the excess bandwidth divided by the filter −6-dB bandwidth (i.e., the fractional excess bandwidth). For a given W_0, r specifies the required excess bandwidth (as a fraction of W_0) and characterizes the steepness of the filter roll-off. The raised cosine characteristic is illustrated in Figure 2.34a for roll-off values of $r = 0$, $r = 0.5$, and $r = 1.0$. The $r = 0$ roll-off is the Nyquist minimum-bandwidth case. Notice that when $r = 1.0$, the required excess bandwidth is 100%; a system with such an overall spectral characteristic can provide a symbol rate of R_s symbols/s using a bandwidth of R_s hertz (twice the Nyquist bandwidth), thus yielding a symbol-

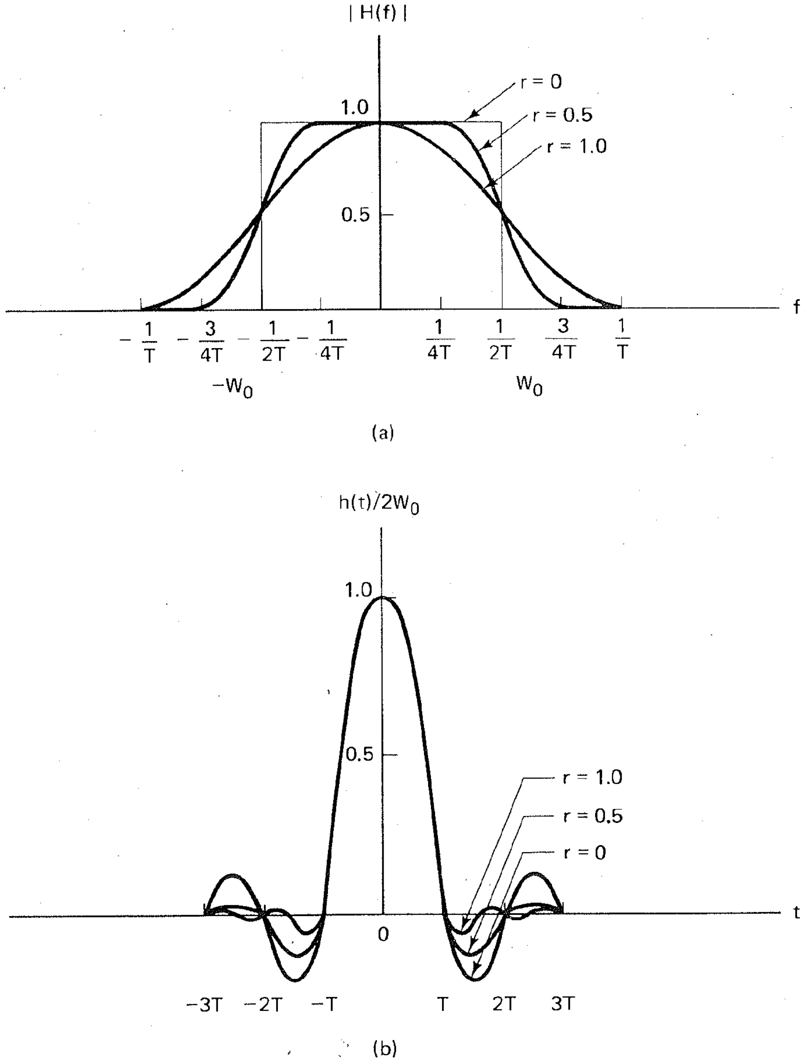


Figure 2.34 Raised cosine filter characteristics. (a) System transfer function. (b) System impulse response.

rate packing of 1 symbol/s/Hz. The corresponding impulse response for the $H(f)$ of Equation (2.74) is

$$h(t) = 2W_0(\text{sinc } 2W_0t) \frac{\cos [2\pi(W - W_0)t]}{1 - 4(W - W_0)t^2} \tag{2.75}$$

The impulse response is shown in Figure 2.34b for $r = 0$, $r = 0.5$, and $r = 1.0$.

Recall that for zero ISI, we shall choose the system received pulse shape to be equal to $h(t)$; we can only do this approximately, since strictly speaking, the raised cosine pulse spectrum is not precisely physically realizable. A realizable frequency characteristic must have a time response that is zero prior to the pulse turn-on time, which is not the case for the family of raised cosine characteristics. These unrealizable filters are *noncausal* (the filter impulse response begins at time $t = -\infty$). However, a delayed version of $h(t)$, say $h(t - t_0)$, may be approximately generated by real filters if the delay t_0 is chosen such that $h(t - t_0) \approx 0$, for $t < 0$. Notice in Figure 2.34b that timing errors will still result in some ISI degradation for $r = 1$. However, the problem is not as serious as it is for $r = 0$, because the tails of the $h(t)$ waveform are of much smaller amplitude for $r = 1$ than they are for $r = 0$.

The Nyquist bandwidth constraint states that the theoretical minimum required system bandwidth, W, for a symbol rate of R_s symbols/s without ISI, is $R_s/2$ hertz. A more general relationship between required bandwidth and symbol transmission rate involves the filter roll-off factor r, and can be stated as

$$W = \tfrac{1}{2}(1 + r)R_s \tag{2.76}$$

Thus with $r = 0$, Equation (2.76) describes the required bandwidth for ideal rectangular filtering, also referred to as *Nyquist filtering*. Bandpass-modulated signals (baseband signals that have been shifted in frequency), such as amplitude shift keying (ASK) and phase shift keying (PSK), require twice the transmission bandwidth of the equivalent baseband signals (see Section 1.7.1). Such frequency-translated signals, occupying twice their baseband bandwidth, are often called double-sideband (DSB) signals. Therefore, for ASK- and PSK-modulated signals, the relationship between the required DSB bandwidth, W_{DSB}, and the symbol transmission rate, R_s, is

$$W_{\text{DSB}} = (1 + r)R_s \tag{2.77}$$

Example 2.4 Bandwidth Requirements

(a) Find the minimum required bandwidth for the baseband transmission of a four-level PCM pulse sequence having a data rate of $R = 2400$ bits/s if the system transfer characteristic consists of a raised cosine spectrum with 100% excess bandwidth ($r = 1$).

(b) The same PCM sequence is modulated onto a carrier wave, so that the baseband spectrum is shifted and centered at frequency f_0. Find the minimum required DSB bandwidth for transmitting the modulated PCM sequence. Assume that the system transfer characteristic is the same as in part (a).

Solution

(a) $M = 2^k$; since $M = 4$ levels, then $k = 2$.

$$\text{Symbol or pulse rate } R_s = \frac{R}{k} = \frac{2400}{2} = 1200 \text{ symbols/s}$$

$$\text{Minimum bandwidth } W = \tfrac{1}{2}(1 + r)R_s = \tfrac{1}{2}(2)(1200) = 1200 \text{ Hz}$$

Figure 2.35a illustrates the baseband PCM received pulse in the time domain—an approximation to the $h(t)$ in Equation (2.75). Figure 2.35b illustrates the Fourier transform of $h(t)$—the raised cosine spectrum. Notice that the required bandwidth, W, starts at zero frequency and extends to $f = 1/T$; it is twice the size of the Nyquist theoretical minimum bandwidth.

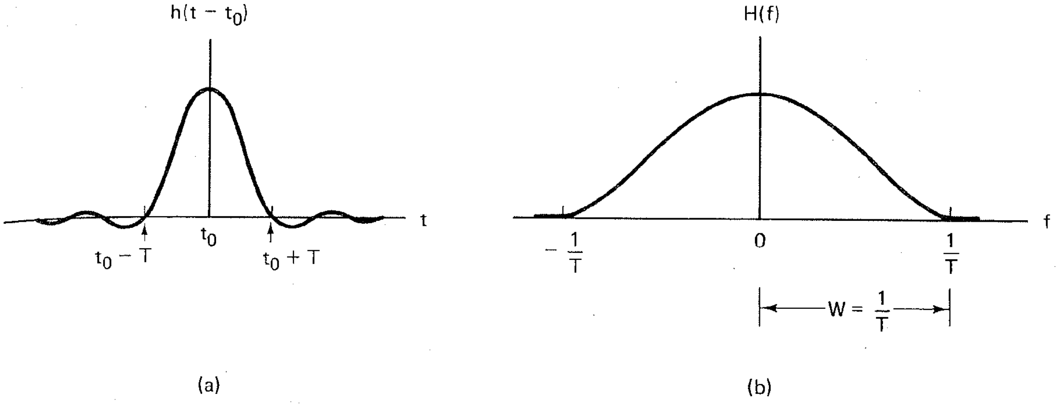


Figure 2.35 (a) Shaped pulse. (b) Baseband raised cosine spectrum.

(b) As in part (a),

$$R_s = 1200 \text{ symbols/s}$$

$$W_{\text{DSB}} = (1 + r)R_s = 2(1200) = 2400 \text{ Hz}$$

Figure 2.36a illustrates the modulated PCM received pulse. This waveform can be viewed as the product of a high-frequency sinusoidal carrier wave and a waveform with the pulse shape of Figure 2.35a. The single-sided spectral plot in Figure 2.36b illustrates that the modulated bandwidth, W_{DSB}, is

$$W_{\text{DSB}} = \left(f_0 + \frac{1}{T}\right) - \left(f_0 - \frac{1}{T}\right) = \frac{2}{T}$$

When the spectrum of Figure 2.35b is shifted up in frequency, the negative and positive halves of the baseband spectrum are shifted up in frequency, thereby doubling the required transmission bandwidth. As the name implies, the DSB signal has two sidebands: the upper sideband (USB), derived from the baseband positive half, and the lower sideband (LSB), derived from the baseband negative half.

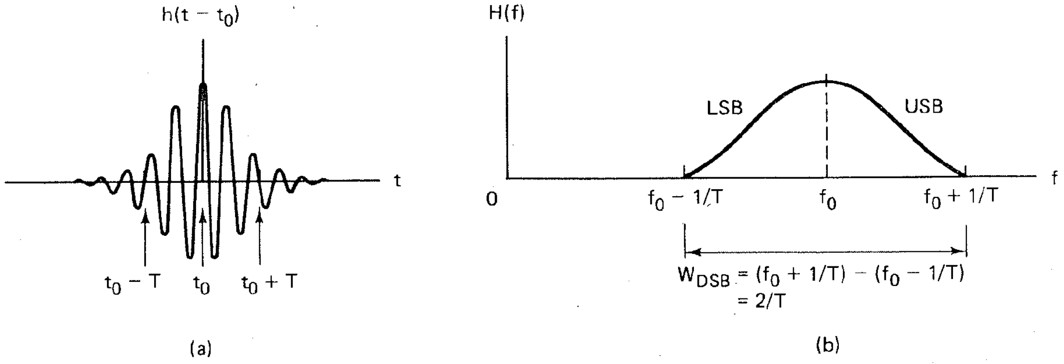


Figure 2.36 (a) Modulated shaped pulse. (b) DSB-modulated raised cosine spectrum.

Example 2.5 Digital Telephone Circuits

Compare the system bandwidth requirements for a 3-kHz analog telephone voice circuit versus a PCM voice circuit. Assume that the sampling rate for the analog-to-digital (A/D) conversion is 8000 samples/s. Also, assume that each voice sample is quantized to one of 256 levels (8-bit quantization).

Solution

The result of the sampling and quantization process yields a PAM signal such that each pulse (symbol) has one of 256 different levels. From Equation (2.76) we can write that the required system bandwidth (without ISI) for R_s symbols/s is

$$W \geq \frac{R_s}{2} \quad \text{hertz}$$

where the equality sign holds true only for Nyquist filtering. For binary PCM, having $L = 256$ levels, each sample is converted to $\ell = \log_2 L = 8$ bits. Therefore, the system bandwidth required to transmit voice using PCM with 8-bit words is

$$W_{\text{PCM}} \geq (\log_2 L)\frac{R_s}{2} \quad \text{hertz}$$

$$\geq \tfrac{1}{2}(8 \text{ bits/symbol})(8000 \text{ symbols/s}) = 32 \text{ kHz}$$

The 3-kHz analog voice circuit will generally require approximately 4 kHz of bandwidth (including some bandwidth separation between channels, called *guard bands*). Therefore, the PCM format using 8-bit quantization requires *at least* eight times the bandwidth required by the analog format.

2.11.2 Equalization

In practical systems, the frequency response of the channel is not known with sufficient precision to allow for a receiver design that will compensate for the intersymbol interference (ISI) for all time. In practice, the filter for handling ISI at the receiver contains various parameters that are adjusted on the basis of measurements of the channel characteristics. The process of thus correcting the channel-induced distortion is called *equalization*. A *transversal filter*—a delay line with T-second taps (where T is the symbol duration)—is a common choice for the *equalizer filter*. The outputs of the taps are amplified, summed, and fed to a decision device. The tap coefficients, c_n, are set to subtract the effects of interference from symbols that are adjacent in time to the desired symbol. Consider that there are $(2N + 1)$ taps with coefficients $c_{-N}, c_{-N+1}, \ldots, c_N$ as shown in Figure 2.37. Output samples, $\{y_k\}$, of the equalizer are then expressed in terms of the input samples, $\{x_k\}$, and tap coefficients as

$$y_k = \sum_{n=-N}^{N} c_n x_{k-n} \qquad k = -2N, \ldots, 2N \tag{2.78}$$

By defining the matrices **y**, **c**, and **x** as

$$\mathbf{y} = \begin{bmatrix} y_{-2N} \\ \vdots \\ y_0 \\ \vdots \\ y_{2N} \end{bmatrix} \qquad \mathbf{c} = \begin{bmatrix} c_{-N} \\ \vdots \\ c_0 \\ \vdots \\ c_N \end{bmatrix} \tag{2.79}$$

$$\mathbf{x} = \begin{bmatrix} x_{-N} & 0 & 0 & \cdots & 0 & 0 \\ x_{-N+1} & x_{-N} & 0 & \cdots & \cdots & \cdots \\ \vdots & & & \vdots & & \vdots \\ x_N & x_{N-1} & x_{N-2} & \cdots & x_{-N+1} & x_{-N} \\ \vdots & & & \vdots & & \vdots \\ 0 & 0 & 0 & \cdots & x_N & x_{N-1} \\ 0 & 0 & 0 & \cdots & 0 & x_N \end{bmatrix} \tag{2.80}$$

we can simplify the computation for $\{y_k\}$ as follows:

$$\mathbf{y} = \mathbf{xc} \tag{2.81}$$

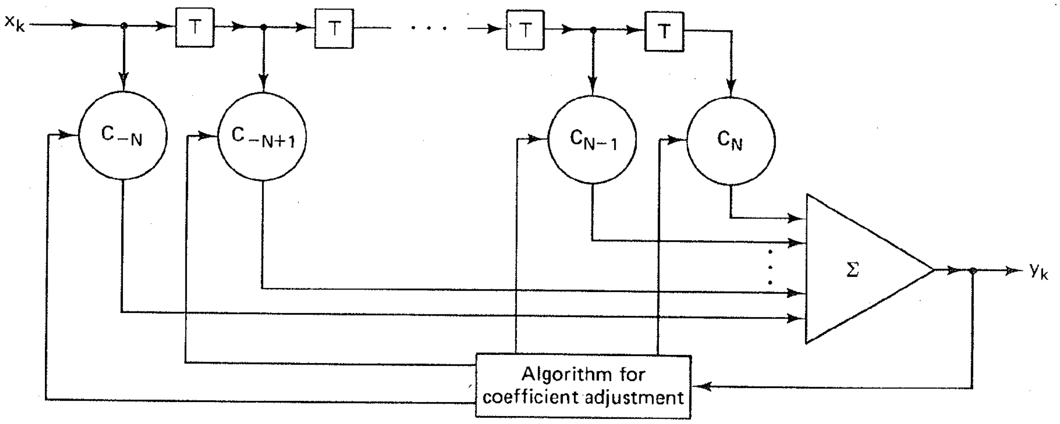


Figure 2.37 Transversal filter.

The criterion for selecting the c_n coefficients is typically based on the minimization of either peak distortion or mean-square distortion. Minimizing peak distortion can be accomplished by selecting the c_n coefficients so that the equalizer output is forced to zero at N sample points on either side of the desired pulse. That is,

$$y_k = \begin{cases} 1 & \text{for } k = 0 \\ 0 & \text{for } k = \pm 1, \pm 2, \ldots, \pm N \end{cases} \tag{2.82}$$

We then solve for c_n by combining Equations (2.79) to (2.81) and solving $2N + 1$ simultaneous equations. Minimizing the mean-square distortion similarly results in $2N + 1$ simultaneous equations.

There are two general types of automatic equalization. The first, *preset equalization*, transmits a training sequence that is compared at the receiver with a locally generated sequence. The differences between the two sequences are used to set the coefficients c_n. With the second method, *adaptive equalization*, the coefficients are continually and automatically adjusted directly from the transmitted data. A disadvantage of preset equalization is that it requires an initial training session, which must be repeated after any break in transmission. Also, a time-varying channel can degrade in ISI since the coefficients are fixed. Adaptive equalization can perform well if the channel error performance is satisfactory. However, if the error performance is poor, received channel errors may not allow the algorithm to converge. A common solution employs preset equalization initially to provide good channel error performance; once normal transmission begins, the system switches to an adaptive algorithm. A significant amount of research and development has taken place in the area of equalization during the past two decades [8, 10, 11].

2.12 PARTIAL RESPONSE SIGNALING

In 1963, Adam Lender [12, 13] showed that it is possible to transmit $2W$ symbols/s with zero ISI, using the theoretical minimum bandwidth of W hertz, without infinitely sharp filters. Lender used a technique called *duobinary signaling*, also referred to by the names *partial response signaling* and *correlative coding*. The basic idea behind the duobinary technique is to introduce some controlled amount of ISI into the data stream rather than trying to eliminate it completely. By introducing correlated interference between the pulses, and by changing the detection procedure, Lender, in effect, "canceled out" the interference at the detector, and thereby achieved the ideal symbol-rate packing of 2 symbols/s/Hz, an amount that had been considered unrealizable.

2.12.1 Duobinary Signaling

To understand how duobinary signaling introduces controlled ISI, let us look at a model of the process. We can think of the duobinary coding operation as if it were implemented as shown in Figure 2.38. Assume that a sequence of binary

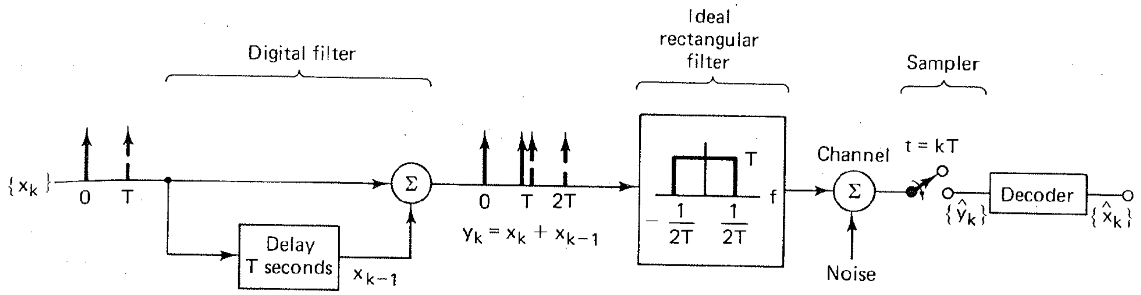


Figure 2.38 Duobinary signaling.

symbols $\{x_k\}$ is to be transmitted at the rate of R symbols/s over a system having an ideal rectangular spectrum of bandwidth $W = R/2 = 1/2T$ hertz. You might ask: How is this rectangular spectrum, in Figure 2.38, different from the unrealizable Nyquist characteristic? It has the same ideal characteristic; but we are not trying to implement the ideal rectangular filter. It is only the part of our equivalent model that is used for developing a filter that is easier to approximate. Before being shaped by the ideal filter, the pulses pass through a simple digital filter, as shown in the figure. The digital filter incorporates a one-digit delay; to each incoming pulse, the filter adds the value of the previous pulse. In other words, for every pulse into the digital filter, we get the summation of two pulses out. Each pulse of the sequence $\{y_k\}$ out of the digital filter can be expressed as

$$y_k = x_k + x_{k-1} \tag{2.83}$$

Hence the $\{y_k\}$ amplitudes are not independent; each y_k digit carries with it the *memory* of the prior digit. The ISI introduced to each y_k digit comes only from the preceding x_{k-1} digit. This correlation between the pulse amplitudes of $\{y_k\}$ can be thought of as the controlled ISI introduced by the duobinary coding. Controlled interference is the essence of this novel technique, because at the detector, such controlled interference can be removed as easily as it was added. The $\{y_k\}$ sequence is followed by the ideal Nyquist filter that does not introduce any ISI. At the receiver sampler, in Figure 2.38, we would expect to recover the sequence $\{y_k\}$, exactly in the absence of noise. Since all systems experience noise contamination, we shall refer to the *received* $\{y_k\}$ as the estimate of $\{y_k\}$ and denote it $\{\hat{y}_k\}$. Removing the controlled interference with the duobinary decoder yields an estimate of $\{x_k\}$ which we shall denote as $\{\hat{x}_k\}$.

2.12.2 Duobinary Decoding

If the binary digit x_k is equal to ± 1, then using Equation (2.83), y_k has one of three possible values: $+2$, 0, or -2. The duobinary code results in a three-level output: in general for M-ary transmission, partial response signaling results in $2M - 1$ output levels. The decoding procedure involves the inverse of the coding procedure, namely, subtracting the x_{k-1} decision from the y_k digit. Consider the following coding/decoding example.

Example 2.6 Duobinary Coding and Decoding

Use Equation (2.83) to demonstrate duobinary coding and decoding for the following sequence: $\{x_k\} = 0\ 0\ 1\ 0\ 1\ 1\ 0$. Consider the first bit of the sequence to be a startup digit, not part of the data.

Solution

Binary digit sequence $\{x_k\}$:	0	0	1	0	1	1	0
Bipolar amplitudes $\{x_k\}$:	-1	-1	$+1$	-1	$+1$	$+1$	-1
Coding rule: $y_k = x_k + x_{k-1}$:		-2	0	0	0	2	0

Decoding decision rule: If $\hat{y}_k = 2$, decide that $\hat{x}_k = +1$ (or binary one)
If $\hat{y}_k = -2$, decide that $\hat{x}_k = -1$ (or binary zero).
If $\hat{y}_k = 0$, decide opposite of the previous decision.

Decoded bipolar sequence $\{\hat{x}_k\}$:	-1	$+1$	-1	$+1$	$+1$	-1
Decoded binary sequence $\{\hat{x}_k\}$:	0	1	0	1	1	0

The decision rule simply implements the subtraction of each $\hat{x}_{k-1}$ decision from each $\hat{y}_k$. One drawback of this detection technique is that once an error is made, it tends to propagate, causing further errors, since present decisions depend on prior decisions. A means of avoiding this error propagation is known as *precoding*.

2.12.3 Precoding

Precoding is accomplished by first differentially encoding the $\{x_k\}$ binary sequence into a new $\{w_k\}$ binary sequence as follows:

$$w_k = x_k \oplus w_{k-1} \tag{2.84}$$

where the symbol $\oplus$ represents modulo-2 addition (equivalent to the logical *exclusive-or* operation) of the binary digits. The rules of modulo-2 addition are as follows:

$$0 \oplus 0 = 0$$

$$0 \oplus 1 = 1$$

$$1 \oplus 0 = 1$$

$$1 \oplus 1 = 0$$

The $\{w_k\}$ binary sequence is then converted to a bipolar pulse sequence, and the coding operation proceeds in the same way as it did in Example 2.6. However, with precoding, the detection process is quite different from the detection of ordinary duobinary, as shown below in Example 2.7. The precoding model is shown in Figure 2.39; in this figure it is implicit that the modulo-2 addition producing the precoded $\{w_k\}$ sequence is performed on the *binary* digits, while the digital filtering producing the $\{y_k\}$ sequence is performed on the *bipolar* pulses.

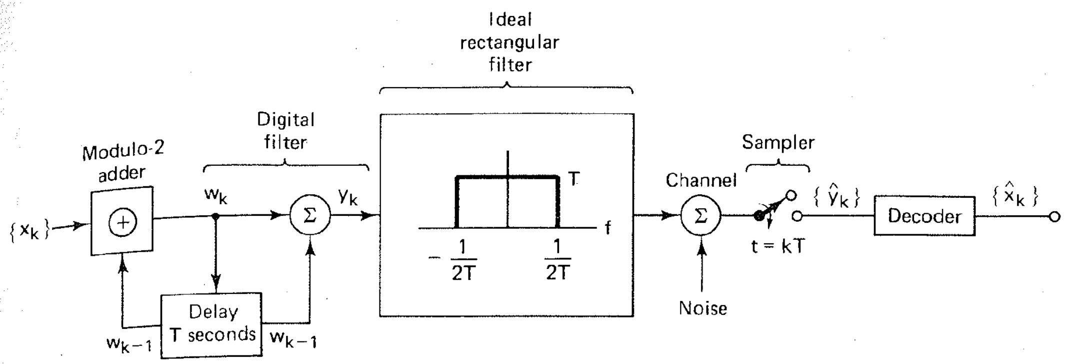


Figure 2.39 Precoded duobinary signaling.

Example 2.7 Duobinary Precoding

Illustrate the duobinary coding and decoding rules when using the differential precoding of Equation (2.84). Assume the same $\{x_k\}$ sequence as that given in Example 2.6.

Solution

Binary digit sequence $\{x_k\}$:	0	0	1	0	1	1	0
Precoded sequence $w_k = x_k \oplus w_{k-1}$:	0	0	1	1	0	1	1
Bipolar sequence $\{w_k\}$:	-1	-1	$+1$	$+1$	-1	$+1$	$+1$
Coding rule: $y_k = w_k + w_{k-1}$:		-2	0	$+2$	0	0	$+2$

Decoding decision rule: If $\hat{y}_k = \pm 2$, decide that $\hat{x}_k$ = binary zero.
If $\hat{y}_k = 0$, decide that $\hat{x}_k$ = binary one.

Decoded binary sequence $\{\hat{x}_k\}$:		0	1	0	1	1	0

The differential precoding enables us to decode the $\{\hat{y}_k\}$ sequence by making a decision on each received sample singly, without resorting to prior decisions which could be in error. The major advantage is that in the event of a digit error due to noise, such an error does not propagate to other digits. Notice that the first bit in the differentially precoded binary sequence $\{w_k\}$ is an arbitrary choice. If the startup bit in $\{w_k\}$ had been chosen to be a binary one instead of a binary zero, the decoded result would have been the same.

2.12.4 Duobinary Equivalent Transfer Function

In Section 2.12.1 we described the duobinary transfer function as a digital filter incorporating a one-digit delay, followed by an ideal rectangular transfer function. Let us now examine an equivalent model. The Fourier transform of a delay can be described as $e^{-j2\pi fT}$ (see Section A.3.1); therefore, the input digital filter of

Figure 2.38 can be characterized with the frequency transfer function, $H_1(f)$, as follows:

$$H_1(f) = 1 + e^{-j2\pi fT} \tag{2.85}$$

The transfer function of the ideal rectangular filter, designated $H_2(f)$, is shown below.

$$H_2(f) = \begin{cases} T & \text{for } |f| < \dfrac{1}{2T} \\ 0 & \text{elsewhere} \end{cases} \tag{2.86}$$

The overall equivalent transfer function $H_e(f)$, of the digital filter cascaded with the ideal rectangular filter is then given by

$$\begin{aligned} H_e(f) &= H_1(f)H_2(f) \qquad \text{for } |f| < \frac{1}{2T} \\ &= (1 + e^{-j2\pi fT})T \\ &= T(e^{j\pi fT} + e^{-j\pi fT})e^{-j\pi fT} \end{aligned} \tag{2.87}$$

$$|H_e(f)| = \begin{cases} 2T \cos \pi fT & \text{for } |f| < \dfrac{1}{2T} \\ 0 & \text{elsewhere} \end{cases} \tag{2.88}$$

Thus $H_e(f)$, the composite transfer function for the cascaded digital and rectangular filters, has a gradual roll-off to the band edge, as can be seen in Figure 2.40a. The transfer function can be approximated by using realizable analog filtering; a separate digital filter is not needed. The duobinary equivalent $H_e(f)$ is called a *cosine filter* [14] (not to be confused with the raised cosine filter described in Section 2.11.1). The corresponding impulse response, $h_e(t)$, found by taking the inverse Fourier transform of $H_e(f)$ in Equation (2.87), is

$$h_e(t) = \text{sinc}\left(\frac{t}{T}\right) + \text{sinc}\left(\frac{t - T}{T}\right) \tag{2.89}$$

and is plotted in Figure 2.40b. For every impulse, $\delta(t)$, at the input of Figure 2.38, the output is $h_e(t)$ with an appropriate polarity. Notice that there are only two nonzero samples, at T-second intervals, giving rise to controlled ISI from the adjacent bit. The introduced ISI is eliminated by use of the decoding procedure discussed in Section 2.12.2. Although the cosine filter is noncausal and therefore nonrealizable, it can be easily approximated. The implementation of the precoded duobinary technique described in Section 2.12.3 can be accomplished by first differentially encoding the binary sequence $\{x_k\}$ into the sequence $\{w_k\}$ (see Example 2.7). The pulse sequence $\{w_k\}$ is then filtered by the equivalent cosine characteristic described in Equation (2.88).

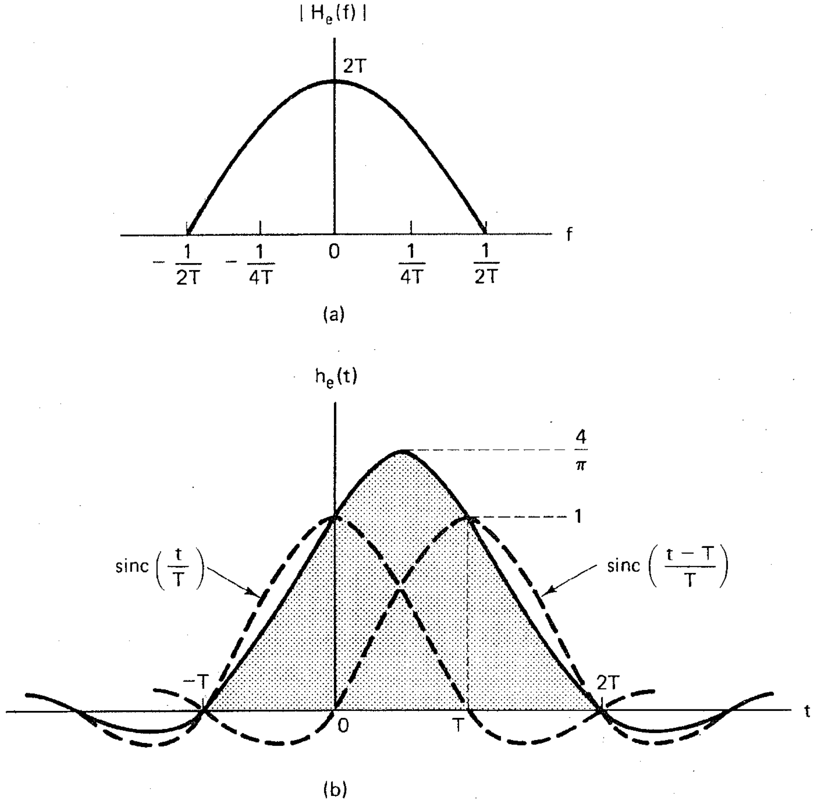


Figure 2.40 Duobinary transfer function and pulse shape. (a) Cosine filter. (b) Impulse response of the cosine filter.

2.12.5 Comparison of Binary with Duobinary Signaling

The duobinary technique introduces correlation between pulse amplitudes, whereas the more restrictive Nyquist criterion assumes that the transmitted pulse amplitudes are independent of one another. We have shown that duobinary signaling can exploit this introduced correlation to achieve zero ISI signal transmission, using a smaller system bandwidth than is otherwise possible. Do we get this performance improvement without paying a price? Such is hardly ever the case with engineering design options; there is almost always a trade-off involved. We saw that duobinary coding requires three levels, compared to the usual two levels for binary coding. Recall our discussion in Section 2.10, where we compared the performance and the required signal power for making eight-level PCM decisions versus two-level PCM decisions. For a fixed amount of signal power, the ease of making reliable decisions is inversely related to the number of levels that must be distinguished in each waveform. Therefore, it should be no surprise that although duobinary signaling accomplishes the zero ISI requirement with minimum bandwidth, duobinary also requires more power than binary signaling, for equivalent performance against noise. For a given probability of bit error (P_B), duobinary signaling requires approximately 2.5 dB greater SNR than binary signaling, while using only $1/(1 + r)$ the bandwidth that binary signaling requires [13], where r is the filter roll-off.

2.12.6 Polybinary Signaling

Duobinary signaling can be extended to more than three digits or levels, resulting in greater bandwidth efficiency; such systems are called *polybinary* [13, 15]. Consider that a binary message with two signaling levels is transformed into a signal with j signaling levels, numbered consecutively from zero to $(j - 1)$. The transformation from binary to polybinary takes place in two steps. First, the original sequence $\{x_k\}$, consisting of binary ones and zeros, is converted into another binary sequence $\{y_k\}$, as follows: The present binary digit of sequence $\{y_k\}$ is formed from the modulo-2 addition of the $(j - 2)$ immediately preceding digits of sequence $\{y_k\}$ and the present digit x_k. For example, let

$$y_k = x_k \oplus y_{k-1} \oplus y_{k-2} \oplus y_{k-3} \tag{2.90}$$

Here x_k represents the input binary digit and y_k the kth encoded digit. Since the expression involves $(j - 2) = 3$ bits preceding y_k, there are $j = 5$ signaling levels. Next, the binary sequence $\{y_k\}$ is transformed into a polybinary pulse train $\{z_k\}$ by adding *algebraically* the present bit of sequence $\{y_k\}$ to the $(j - 2)$ preceding bits of $\{y_k\}$. Therefore, z_k modulo-2 $= x_k$, and the binary elements one and zero are mapped into even- and odd-valued pulses in the sequence $\{z_k\}$. Note that each digit in $\{z_k\}$ can be independently detected despite the strong correlation between bits. The primary advantage of such a signaling scheme is the redistribution of the spectral density of the original sequence $\{x_k\}$, so as to favor the low frequencies, thus improving system bandwidth efficiency.

2.13 CONCLUSION

In this chapter we have considered the first important step in any digital communication system, transforming the source information (both textual and analog) to a form that is compatible with a digital system. We treated various aspects of sampling, quantization (both uniform and nonuniform), and pulse code modulation (PCM). We also considered the selection of PCM waveforms for the transmission of baseband signals through the channel.

We described the detection of binary signals plus Gaussian noise in terms of two basic steps. In the first step the received waveform is reduced to a single number, $z(T)$, and in the second step a decision is made as to which signal was transmitted, on the basis of comparing $z(T)$ to a threshold. We discussed how to best choose this threshold. We also showed that a linear filter known as a matched filter or correlator is the optimum choice for maximizing the output signal-to-noise ratio, and thus minimizing the probability of error.

We defined intersymbol interference (ISI) and explained the importance of Nyquist's work in establishing a theoretical minimum bandwidth for symbol detection without ISI. We also introduced the duobinary concept of adding a controlled amount of ISI to achieve an improvement in bandwidth efficiency at the expense of an increase in power.

REFERENCES

1. Black, H. S., *Modulation Theory*, D. Van Nostrand Company, Princeton, N.J., 1953.
2. Oppenheim, A. V., *Applications of Digital Signal Processing*, Prentice-Hall, Inc., Englewood Cliffs, N.J., 1978.
3. Stiltz, H., ed., *Aerospace Telemetry*, Vol. 1, Prentice-Hall, Inc., Englewood Cliffs, N.J., 1961. p. 179.
4. Hecht, M., and Guida, A., "Delay Modulation," *Proc. IEEE*, vol. 57, no. 7, July 1969, pp. 1314–1316.
5. Deffebach, H. L., and Frost, W. O., "A Survey of Digital Baseband Signaling Techniques," *NASA Technical Memorandum NASATM X-64615*, June 30, 1971.
6. Van Trees, H. L., *Detection, Estimation, and Modulation Theory*, Part 1, John Wiley & Sons, Inc., New York, 1968.
7. Borjesson, P. O., and Sundberg, C. E., "Simple Approximations of the Error Function $Q(x)$ for Communications Applications," *IEEE Trans. Commun.*, vol. COM27, Mar. 1979, pp. 639–642.
8. Proakis, J. G., *Digital Communications*, McGraw-Hill Book Company, New York, 1983.
9. Nyquist, H., "Certain Topics of Telegraph Transmission Theory," *Trans. Am. Inst. Electr. Eng.*, vol. 47, Apr. 1928, pp. 617–644.
10. Korn, I., *Digital Communications*, Van Nostrand Reinhold Company, Inc., New York, 1985.
11. Wu, W. W., *Elements of Digital Satellite Communication*, Computer Science Press, Inc., Rockville, Md., 1984.
12. Lender, A., "The Duobinary Technique for High Speed Data Transmission," *IEEE Trans. Commun. Electron.*, vol. 82, May 1963, pp. 214–218.
13. Lender, A., "Correlative (Partial Response) Techniques and Applications to Digital Radio Systems," in K. Feher, *Digital Communications: Microwave Applications*, Prentice-Hall, Inc., Englewood Cliffs, N.J., 1981, Chap. 7.
14. Couch, L. W., II, *Digital and Analog Communication Systems*, Macmillan Publishing Company, New York, 1982.
15. Lender, A., "Correlative Digital Communication Techniques," *IEEE Trans. Commun. Technol.*, Dec. 1964, pp. 128–135.

PROBLEMS

2.1. You want to transmit the word "HOW" using an 8-ary system.

(a) Encode the word "HOW" into a sequence of bits, using 7-bit ASCII coding, followed by an eighth bit for error detection, per character. The eighth bit is chosen so that the number of ones in the 8 bits is an even number. How many total bits are there in the message?

(b) Partition the bit stream into $k = 3$ bit segments. Represent each of the 3-bit segments as an octal number (symbol). How many octal symbols are there in the message?

(c) If the system were designed with 16-ary modulation, how many symbols would be used to represent the word "HOW"?

(d) If the system were designed with 256-ary modulation, how many symbols would be used to represent the word "HOW"?

2.2. We want to transmit 800 characters/s, where each character is represented by its 7-bit ASCII codeword, followed by an eighth bit for error detection, per character, as in Problem 2.1. A multilevel PCM format with $M = 16$ levels is used.

(a) What is the effective transmitted bit rate?

(b) What is the PCM symbol rate?

2.3. We wish to transmit a 100-character alphanumeric message in 2 s, using 7-bit ASCII coding, followed by an eighth bit for error detection, per character, as in Problem 2.1. A multilevel PCM format with $M = 32$ levels is used.

(a) Calculate the effective transmitted bit rate and the PCM symbol rate.

(b) Repeat part (a) for 16-level PCM, eight-level PCM, four-level PCM, and binary PCM.

2.4. Given an analog waveform that has been sampled at its Nyquist rate, f_s, using natural sampling, prove that a waveform (proportional to the original waveform) can be recovered from the samples, using the recovery techniques shown in Figure P2.1. The parameter mf_s is the frequency of the local oscillator, where m is an integer.

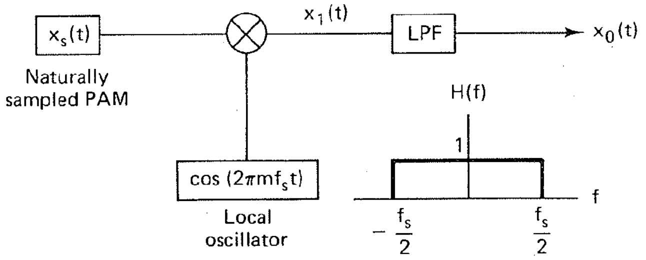


Figure P2.1

2.5. An analog signal is sampled at its Nyquist rate $1/T_s$, and quantized using L quantization levels. The derived digital signal is then transmitted on some channel.

(a) Show that the time duration, T, of one bit of the transmitted binary encoded signal must satisfy $T \leq T_s/(\log_2 L)$.

(b) When is the equality sign valid?

2.6. Determine the number of quantization levels that are implied if the number of bits per sample in a given PCM code is **(a)** 5; **(b)** 8; **(c)** x.

2.7. Determine the minimum sampling rate necessary to sample and perfectly reconstruct the signal $x(t) = \sin(6280t)/(6280t)$.

2.8. Consider an audio signal with spectral components limited to the frequency band 300 to 3300 Hz. Assume that a sampling rate of 8000 samples/s will be used to generate a PCM signal. Assume that the ratio of peak signal power to average quantization noise power at the output needs to be 30 dB.

(a) What is the minimum number of uniform quantization levels needed, and what is the minimum number of bits per sample needed?

(b) Calculate the system bandwidth (as specified by the main spectral lobe of the signal) required for the detection of such a PCM signal.

2.9. A waveform, $x(t) = 10 \cos(1000t + \pi/3) + 20 \cos(2000t + \pi/6)$ is to be uniformly sampled for digital transmission.

(a) What is the maximum allowable time interval between sample values that will ensure perfect signal reproduction?

(b) If we want to reproduce 1 hour of this waveform, how many sample values need to be stored?

2.10. (a) A waveform that is bandlimited to 50 kHz is sampled every 10 μs. Show graphically that these samples uniquely characterize the waveform. (Use a sinusoidal example for simplicity. Avoid sampling at points where the waveform equals zero.)

(b) If samples are taken 30 μs apart instead of 10 μs, show graphically that waveforms other than the original can be characterized by the samples.

2.11. Use the method of convolution to illustrate the effect of undersampling the waveform $x(t) = \cos 2\pi f_0 t$ for a sampling rate of $f_s = \frac{3}{2} f_0$.

2.12. (a) Sketch the complete $\mu = 10$ compression characteristic that will handle input voltages in the range -5 to $+5$ V.

(b) Plot the corresponding expansion characteristic.

(c) Draw a 16-level nonuniform quantizer characteristic that corresponds to the $\mu = 10$ compression characteristic.

2.13. Assume a binary sequence with equally likely binary levels. The sequence can be represented by either a bipolar or a unipolar signal set. Show that if the corresponding bipolar signal and unipolar signal have the same peak-to-peak amplitude separation, the bipolar signal uses less average power than the unipolar signal.

2.14. Assume that in a binary digital communication system, the signal component out of the correlator receiver is $a_i(T) = +1$ or -1 V with equal probability. If the Gaussian noise at the correlator output has unit variance, find the probability of a bit error.

2.15. A bipolar binary signal, $s_i(t)$, is a $+1$- or -1-V pulse during the interval $(0, T)$. Additive white Gaussian noise having two-sided power spectral density of 10^{-3} W/Hz is added to the signal. If the received signal is detected with a matched filter, determine the maximum bit rate that can be sent with a bit error probability of $P_B \leq 10^{-3}$.

2.16. Bipolar pulse signals, $s_i(t)$ $(i = 1, 2)$, of amplitude ±1 V are received in the presence of Gaussian noise with $\sigma^2 = 0.1$ V². Determine the optimum (minimum probability of error) detection threshold, γ_0, for matched filter detection if the a priori probabilities are: **(a)** $P(s_1) = 0.5$; **(b)** $P(s_1) = 0.7$; **(c)** $P(s_1) = 0.2$. **(d)** Explain the effect of the a priori probabilities on the value of γ_0. [*Hint:* Refer to Equations (B.10) to (B.12).]

2.17. A binary communication system transmits signals $s_i(t)$ $(i = 1, 2)$. The receiver test statistic, $z(T) = a_i + n_0$, where the signal component, a_i, is either $a_1 = +1$ or $a_2 = -1$, and the noise component, n_0 is uniformly distributed, yielding the conditional density functions $p(z|s_i)$ given by

$$p(z|s_1) = \begin{cases} \frac{1}{2} & \text{for } -0.2 \leq z \leq 1.8 \\ 0 & \text{otherwise} \end{cases}$$

$$p(z|s_2) = \begin{cases} \frac{1}{2} & \text{for } -1.8 \leq z \leq 0.2 \\ 0 & \text{otherwise} \end{cases}$$

Find the probability of a bit error, P_B, for the case of equally likely signaling and the use of an optimum decision threshold.

2.18. The information in an analog waveform, whose maximum frequency $f_m = 4000$ Hz, is to be transmitted using a 16-level PCM system. The quantization distortion must not exceed ±1% of the peak-to-peak analog signal.

(a) What is the minimum number of bits per sample or bits per PCM word that should be used in this PCM system?

(b) What is the minimum required sampling rate, and what is the resulting bit rate?

(c) What is the PCM pulse or symbol transmission rate?

2.19. (a) What is the theoretical minimum system bandwidth needed for a 10-Mbits/s signal using 16-level PCM without ISI?

(b) How large can the filter roll-off factor be if the allowable system bandwidth is 1.375 MHz?

2.20. A voice signal (300 to 3300 Hz) is digitized such that the quantization distortion $\leq \pm 0.1\%$ of the peak-to-peak signal voltage. Assume a sampling rate of 8000 samples/s and a multilevel PCM format with $M = 32$ levels. Find the theoretical minimum system bandwidth that avoids ISI.

2.21. A binary waveform of 9600 bits/s is converted to an octal waveform that is transmitted over a system having a raised cosine roll-off filter characteristic. The system has a conditioned (equalized) response out to 2.4 kHz.

(a) What is the octal symbol rate?

(b) What is the roll-off factor of the filter characteristic?

2.22. A voice signal in the range 300 to 3300 Hz is sampled at 8000 samples/s. We may transmit these samples directly as PAM, or we may first convert them into codewords using PCM.

(a) What is the minimum system bandwidth required for the detection of PAM with no ISI and with a filter roll-off characteristic of $r = 1$?

(b) Using the same filter roll-off characteristic, what is the minimum bandwidth required for the detection of binary PCM if the samples are quantized to eight levels?

(c) Repeat part (b) using 128 quantization levels.

2.23. A signal in the frequency range 300 to 3300 Hz is limited to a peak-to-peak swing of 10 V. It is sampled at 8000 samples/s and the samples are quantized to 64 evenly spaced levels. Calculate and compare the bandwidths and ratio of peak signal power to rms quantization noise if the quantized samples are transmitted either as binary pulses or as four-level pulses. Assume that the system bandwidth is defined by the main spectral lobe of the signal.

2.24. An analog signal is to be converted to a binary PCM signal and transmitted over a channel that is bandlimited to 100 kHz. Assume that 32 quantization levels are used and that the overall equivalent transfer function is of the raised cosine type with roll-off $r = 0.6$.

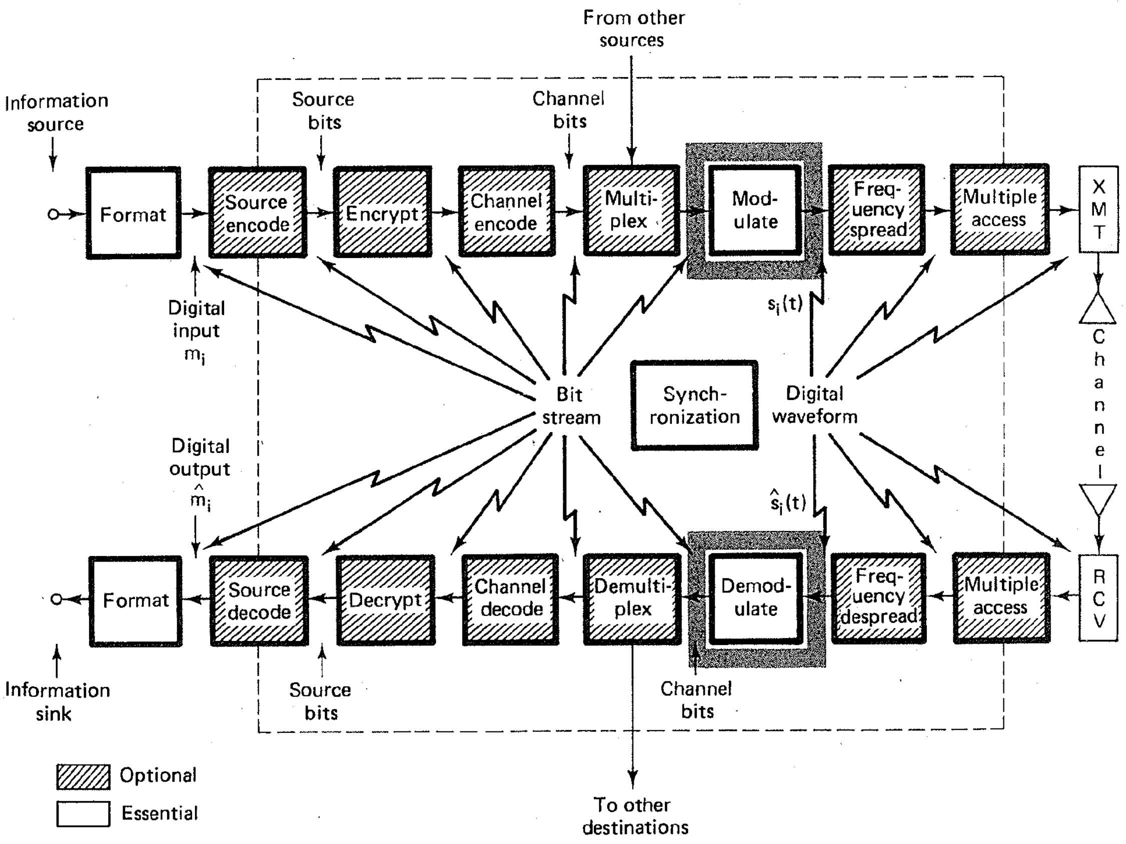
(a) Find the maximum PCM bit rate that can be used by this system without introducing ISI.

(b) Find the maximum signal bandwidth that can be accommodated for the analog signal.

(c) Repeat parts (a) and (b) for an eight-level PCM signal.

CHAPTER 3

Bandpass Modulation and Demodulation



3.1 WHY MODULATE?

Digital modulation is the process by which digital symbols are transformed into waveforms that are compatible with the characteristics of the channel. In the case of baseband modulation, these waveforms are pulses, but in the case of *bandpass modulation* the desired information signal modulates a sinusoid called a *carrier wave,* or simply a *carrier*; for radio transmission the carrier is converted to an electromagnetic (EM) field for propagation to the desired destination. One might ask why it is necessary to use a carrier for the radio transmission of baseband signals. The answer is as follows. The transmission of EM fields through space is accomplished with the use of antennas. To efficiently couple the transmitted EM energy into space, the dimensions of the antenna aperture should be at least as large as the wavelength being transmitted. Wavelength, λ, is equal to c/f, where c, the speed of light, is 3×10^8 m/s. For a baseband signal with frequency f = 3000 Hz, $\lambda = 10^5$ m $\simeq$ 60 miles. To efficiently transmit a 3000-Hz signal through space *without carrier-wave modulation*, an antenna that spans at least 60 miles would be required. Even if we were willing to inefficiently transmit the EM energy with an antenna measuring one-tenth of a wavelength, we are faced with an impossible antenna size. However, if the information to be transmitted is first modulated on a higher frequency carrier, for example a 30-GHz carrier, the equivalent antenna diameter is then less than $\frac{1}{2}$ in. For this reason, carrier-wave or bandpass modulation is an essential step for all systems involving radio transmission.

Bandpass modulation can provide other important benefits in signal transmission. If more than one signal utilizes a single channel, modulation may be used to separate the different signals. Such a technique, known as *frequency-division multiplexing,* is discussed in Chapter 9. Modulation can be used to minimize the effects of interference. A class of such modulation schemes, known as *spread-spectrum modulation*, requires a system bandwidth much larger than the minimum bandwidth that would be required by the message. The trade-off of bandwidth for interference rejection is considered in Chapter 10. Modulation can also be used to place a signal in a frequency band where design requirements, such as filtering and amplification, can be easily met. This is the case when radio-frequency (RF) signals are converted to an intermediate frequency (IF) in a receiver.

3.2 SIGNALS AND NOISE

3.2.1 Noise in Radio Communication Systems

The task of the demodulator or detector is to retrieve the bit stream from the received waveform, as nearly error free as possible, notwithstanding the distortion to which the signal may have been subjected. There are two primary causes for signal distortion. The first is the filtering effects of the transmitter, channel, and receiver discussed in Section 2.11. As described there, a nonideal system transfer function causes symbol "smearing," which can produce *intersymbol interference*.

The second cause for signal distortion is the noise that is produced by a variety of sources, such as galaxy noise, terrestrial noise, amplifier noise, and unwanted signals from other sources. An unavoidable cause of noise is the thermal motion of electrons in any conducting media. This motion produces *thermal noise* in amplifiers and circuits which corrupts the signal in an additive fashion; that is, the received signal, $r(t)$, is the sum of the transmitted signal, $s(t)$, and the thermal noise, $n(t)$. The statistics of thermal noise have been developed using quantum mechanics and are well known [1].

The primary statistical characteristic of thermal noise is that the noise amplitudes are distributed according to a normal or Gaussian distribution, discussed in Section 1.5.5 and shown in Figure 1.7. The probability density function (pdf), $p(n)$, of the zero-mean noise voltage is expressed as

$$p(n) = \frac{1}{\sigma\sqrt{2\pi}} \exp\left[-\frac{1}{2}\left(\frac{n}{\sigma}\right)^2\right] \tag{3.1}$$

where σ^2 is the noise variance. In Figure 1.7 it can be seen that the most probable noise amplitudes are those with small positive or negative values. In theory, the noise can be infinitely large, but very large noise amplitudes are rare.

The primary spectral characteristic of thermal noise is that its two-sided power spectral density, $G_n(f) = N_0/2$, is flat for all frequencies of interest for radio communication systems. In other words, thermal noise, on the average, has just as much power per hertz in low-frequency fluctuations as in high-frequency fluctuations—up to a frequency of about 10^{12} hertz. When the noise power is characterized by a constant power spectral density, as shown in Figure 1.8a, we refer to it as *white noise*. Since thermal noise is present in all communication systems and is the predominant noise source for most systems, the thermal noise characteristics (additive, white, and Gaussian) are most often used to model the noise in the detection process and in the design of optimum receivers.

3.2.2 A Geometric View of Signals and Noise

Let us define an N-dimensional *orthogonal space* as one characterized by a set of N linearly independent functions, $\{\psi_j(t)\}$, called *basis functions*. Any arbitrary function in the space can be generated by a linear combination of these basis functions. The basis functions must satisfy the following conditions:

$$\int_0^T \psi_j(t)\psi_k(t)\,dt = K_j\delta_{jk} \qquad 0 \le t \le T; \quad j, k = 1, \ldots, N \tag{3.2a}$$

$$\delta_{jk} = \begin{cases} 1 & \text{for } j = k \\ 0 & \text{otherwise} \end{cases} \tag{3.2b}$$

where the operator δ_{jk} is called the *Kronecker delta function* and is defined by Equation (3.2b). When the K_j constants are nonzero, the signal space is called *orthogonal*. When the basis functions are normalized so that each $K_j = 1$, the space is called an *orthonormal* space. The principal requirement for orthogonality can be stated as follows: Each $\psi_j(t)$ function of the set of basis functions must be independent of the other members of the set. Each $\psi_j(t)$ must not interfere with any other members of the set in the detection process. From a geometric point of view, each $\psi_j(t)$ is mutually perpendicular to each of the other $\psi_k(t)$ for $j \neq k$. An example of such a space with $N = 3$ is shown in Figure 3.1, where the mutually perpendicular axes are designated $\psi_1(t)$, $\psi_2(t)$, and $\psi_3(t)$. If $\psi_j(t)$ corresponds to a real-valued voltage or current waveform component, associated with a 1-Ω

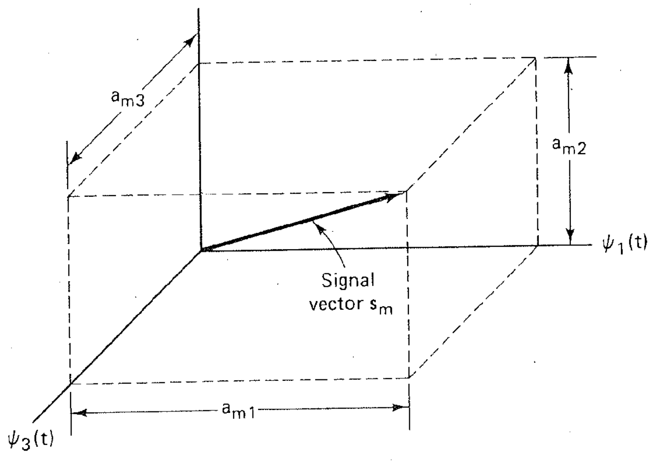


Figure 3.1 Vectorial representation of the signal waveform $s_m(t)$.

resistive load, then using Equations (1.5) and (3.2), the normalized energy in joules dissipated in the load in T seconds, due to ψ_j, is

$$E_j = \int_0^T \psi_j^2(t)\, dt = K_j \tag{3.3}$$

One reason we focus on an *orthogonal signal space* is that Euclidean distance measurements, fundamental to the detection process, are easily formulated in such a space. However, even if the signaling waveforms do not comprise such an orthogonal set, they can be transformed into linear combinations of orthogonal waveforms. It can be shown [2] that *any arbitrary* finite set of waveforms $\{s_i(t)\}$ $(i = 1, \ldots, M)$, where each member of the set is physically realizable and of duration T, can be expressed as a linear combination of N orthogonal waveforms $\psi_1(t), \psi_2(t), \ldots, \psi_N(t)$, where $N \leq M$, such that

$$\begin{aligned}
s_1(t) &= a_{11}\psi_1(t) + a_{12}\psi_2(t) + \cdots + a_{1N}\psi_N(t)\\
s_2(t) &= a_{21}\psi_1(t) + a_{22}\psi_2(t) + \cdots + a_{2N}\psi_N(t)\\
&\vdots \qquad\qquad\qquad\qquad\qquad\qquad\qquad \vdots\\
s_M(t) &= a_{M1}\psi_1(t) + a_{M2}\psi_2(t) + \cdots + a_{MN}\psi_N(t)
\end{aligned}$$

These relationships are expressed in more compact notation as follows:

$$s_i(t) = \sum_{j=1}^{N} a_{ij}\psi_j(t) \qquad \begin{array}{l} i = 1, \ldots, M \\ N \leq M \end{array} \tag{3.4}$$

where

$$a_{ij} = \frac{1}{K_j}\int_0^T s_i(t)\psi_j(t)\, dt \qquad \begin{array}{l} i = 1, \ldots, M; \quad 0 \leq t \leq T \\ j = 1, \ldots, N \end{array} \tag{3.5}$$

The coefficient a_{ij} is the value of the $\psi_j(t)$ component of signal, $s_i(t)$. The form of the $\{\psi_j(t)\}$ is not specified; it is chosen for convenience and will depend on the form of the signal waveforms. The set of signal waveforms, $\{s_i(t)\}$, can be viewed as a set of vectors, $\{\mathbf{s}_i\} = \{a_{i1}, a_{i2}, \ldots, a_{iN}\}$. If, for example, $N = 3$, we may plot the vector, $\mathbf{s}_m$, corresponding to the waveform

$$s_m(t) = a_{m1}\psi_1(t) + a_{m2}\psi_2(t) + a_{m3}\psi_3(t)$$

as a point in a three-dimensional Euclidean space with coordinates (a_{m1}, a_{m2}, a_{m3}), as shown in Figure 3.1. The orientation among the signal vectors describes the relation of the signals to one another (with respect to phase or frequency), and the amplitude of each vector in the set $\{\mathbf{s}_i\}$ is a measure of the signal energy transmitted during a symbol duration. In general, once a set of N orthogonal functions has been adopted, each of the transmitted signal waveforms, $s_i(t)$, is completely determined by the vector of its coefficients

$$\mathbf{s}_i = (a_{i1}, a_{i2}, \ldots, a_{iN}) \qquad i = 1, \ldots, M \tag{3.6}$$

We shall employ the notation of signal vectors, $\{\mathbf{s}\}$, or signal waveforms, $\{\mathbf{s}(t)\}$, as best suits the discussion. A typical detection problem, conveniently viewed in terms of signal vectors, is illustrated in Figure 3.2. Vectors $\mathbf{s}_j$ and $\mathbf{s}_k$ represent *prototype* or *reference signals* belonging to the set of M waveforms, $\{s_i(t)\}$. The receiver knows, a priori, the location in the signal space of each prototype vector belonging to the M-ary set. During the transmission of any signal, the signal is perturbed by noise so that the resultant vector that is actually received is a perturbed version (e.g., $\mathbf{s}_j + \mathbf{n}$ or $\mathbf{s}_k + \mathbf{n}$) of the original one, where $\mathbf{n}$ represents a noise vector. The noise is additive and has a Gaussian distribution; therefore, the resulting distribution of possible received signals is a cluster or cloud of points around $\mathbf{s}_j$ and $\mathbf{s}_k$. The cluster is dense in the center and becomes sparse with increasing distance from the prototype. The arrow marked $\mathbf{r}$ represents a signal vector that might arrive at the receiver during some symbol interval. The task of the receiver is to decide whether $\mathbf{r}$ has a close "resemblance" to the prototype $\mathbf{s}_j$, whether it more closely resembles $\mathbf{s}_k$, or whether it is closer to some other prototype signal in the M-ary set. The measurement can be thought of as a *distance* measurement. The question that the receiver or detector must resolve is: Which of the prototypes within the signal space is *closest* in distance to the received vector, $\mathbf{r}$? The analysis of all demodulation or detection schemes involves this concept of *distance* between a received waveform and a set of possible transmitted waveforms. A simple rule for the detector to follow is to decide that $\mathbf{r}$ belongs to the same class as its nearest neighbor (nearest prototype vector).

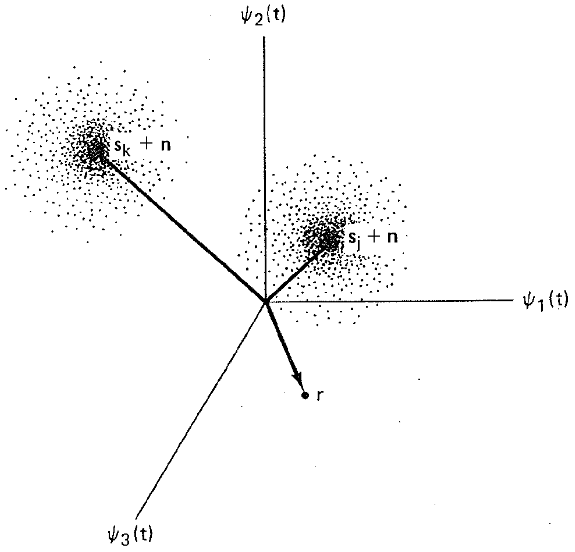


Figure 3.2 Signals and noise in a three-dimensional vector space.

3.2.2.1 Waveform Energy

Using Equations (1.5), (3.4), and (3.2), the normalized energy, E_i, associated with the waveform, $s_i(t)$, over a symbol interval, T, can be expressed in terms of the orthogonal components of $s_i(t)$ as follows:

$$E_i = \int_0^T s_i^2(t)\, dt = \int_0^T \left[\sum_j a_{ij}\psi_j(t)\right]^2 dt \tag{3.7}$$

$$= \int_0^T \sum_j a_{ij}\psi_j(t) \sum_k a_{ik}\psi_k(t)\, dt \tag{3.8}$$

$$= \sum_j \sum_k a_{ij}a_{ik} \int_0^T \psi_j(t)\psi_k(t)\, dt \tag{3.9}$$

$$= \sum_j \sum_k a_{ij}a_{ik}K_j\delta_{jk} \tag{3.10}$$

$$= \sum_{j=1}^{N} a_{ij}^2 K_j \qquad i = 1, \ldots, M \tag{3.11}$$

Equation (3.11) is a special case of Parseval's theorem relating the integral of the square of the waveform, $s_i(t)$, to the sum of the square of the orthogonal series coefficients. If orthonormal functions are used (i.e., $K_j = 1$), the normalized energy over a symbol duration T is given by

$$E_i = \sum_{j=1}^{N} a_{ij}^2 \tag{3.12}$$

If there is equal energy, E, in each of the $s_i(t)$ waveforms, we can write Equation (3.12) in the form

$$E = \sum_{j=1}^{N} a_{ij}^2 \qquad \text{for all } i \tag{3.13}$$

3.2.2.2 Generalized Fourier Transforms

The transformation described by Equations (3.2), (3.4), and (3.5) is referred to as the *generalized Fourier transformation.* In the case of ordinary Fourier transforms, the $\{\psi_j(t)\}$ set is comprised of sine and cosine harmonic functions. But in the case of generalized Fourier transforms, the $\{\psi_j(t)\}$ set is not constrained to any specific form; it must only satisfy the orthogonality statement of Equation (3.2). *Any* arbitrary integrable waveform set, as well as noise, can be represented as a linear combination of orthogonal waveforms through such a generalized Fourier transformation [2]. Therefore, in such an orthogonal space, we are justified in using distance (Euclidean distance) as a decision criterion for the detection of *any* signal set in the presence of AWGN. The most important application of this orthogonal transformation has to do with the way in which signals are actually transmitted and received. The transmission of a nonorthogonal signal set is generally accomplished by the appropriate weighting of the orthogonal carrier components. For example, in Section 3.5.3 we show that multiple phase shift keying (MPSK) signals are fully characterized by weighted sine and cosine components of the carrier.

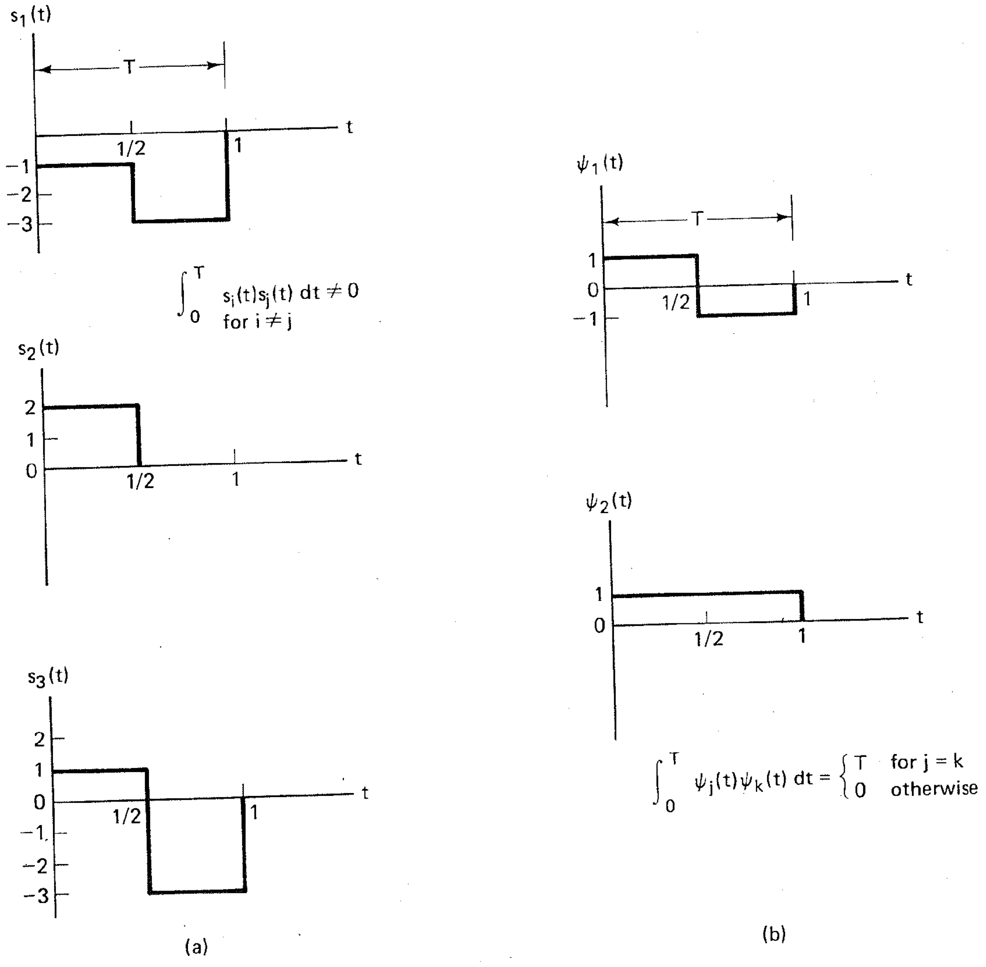
Example 3.1 Orthogonal Representation of Waveforms

Figure 3.3 illustrates the statement that any arbitrary integrable waveform set can be represented as a linear combination of orthogonal waveforms. Figure 3.3a shows a set of three waveforms, $s_1(t)$, $s_2(t)$, $s_3(t)$.

(a) Demonstrate that these waveforms *do not* form an orthogonal set.

(b) Figure 3.3b shows a set of two waveforms, $\psi_1(t)$ and $\psi_2(t)$. Verify that these waveforms form an orthogonal set.

(c) Show how the nonorthogonal waveform set in part (a) can be expressed as a linear combination of the orthogonal set in part (b).



$s_1(t)$
T
t
1/2
1
−1
−2
−3
$\int_0^T s_i(t)s_j(t)\,dt \neq 0$ for $i \neq j$
$s_2(t)$
2
1
0
1/2
1
t
$s_3(t)$
2
1
0
−1
−2
−3
1/2
1
t
(a)
$\psi_1(t)$
T
1
0
−1
1/2
1
t
$\psi_2(t)$
1
0
1/2
1
t
$\int_0^T \psi_j(t)\psi_k(t)\,dt = \begin{cases} T & \text{for } j = k \\ 0 & \text{otherwise} \end{cases}$
(b)

Figure 3.3 Example of an arbitrary signal set in terms of an orthogonal set. (a) Arbitrary signal set. (b) Orthogonal basis functions.

Solution

(a) $s_1(t)$, $s_2(t)$, and $s_3(t)$ are clearly not orthogonal, since they do not meet the requirements of Equation (3.2); that is, the time integrated value (over a symbol duration) of the cross-product of any two of the three waveforms is not zero. Let us verify this for $s_1(t)$ and $s_2(t)$.

$$\int_0^T s_1(t)s_2(t)\,dt = \int_0^{T/2} s_1(t)s_2(t)\,dt + \int_{T/2}^T s_1(t)s_2(t)\,dt$$

$$= \int_0^{T/2} (-1)(2)\,dt + \int_{T/2}^T (-3)(0)\,dt = -T$$

Similarly, the integral over the interval, T, of each of the cross-products $s_1(t)s_3(t)$ and $s_2(t)s_3(t)$ results in nonzero values. Hence the waveform set $\{s_i(t)\}$ $(i = 1, 2, 3)$ in Figure 3.3a is not an orthogonal set.

(b) Using Equation (3.2), we verify that $\psi_1(t)$ and $\psi_2(t)$ form an orthogonal set as follows:

$$\int_0^T \psi_1(t)\psi_2(t)\,dt = \int_0^{T/2} (1)(1)\,dt + \int_{T/2}^T (-1)(1)\,dt = 0$$

(c) We can express the nonorthogonal set $\{s_i(t)\}$ $(i = 1, 2, 3)$ as a linear combination of the orthogonal basis waveforms $\{\psi_j(t)\}$ $(j = 1, 2)$, as follows, by using Equation (3.5), where $K_j = T$:

$$s_1(t) = \psi_1(t) - 2\psi_2(t)$$

$$s_2(t) = \psi_1(t) + \psi_2(t)$$

$$s_3(t) = 2\psi_1(t) - \psi_2(t)$$

These relationships illustrate how an arbitrary waveform set $\{s_i(t)\}$ can be expressed as a linear combination of an orthogonal set $\{\psi_j(t)\}$, as described in Equations (3.4) and (3.5). What are the practical applications for being able to describe $s_1(t)$, $s_2(t)$, and $s_3(t)$, in terms of $\psi_1(t)$, $\psi_2(t)$, and the appropriate coefficients? If we want a system for transmitting waveforms $s_1(t)$, $s_2(t)$, and $s_3(t)$, the transmitter and the receiver need only be implemented using the two basis functions $\psi_1(t)$ and $\psi_2(t)$ instead of the three original waveforms. A convenient way in which an appropriate choice of a basis function set, $\{\psi_j(t)\}$, can be obtained for any given signal set, $\{s_i(t)\}$, is called the *Gram–Schmidt orthogonalization procedure*. It is described in Appendix 4A of Reference [3].

3.2.2.3 Representing White Noise with Orthogonal Waveforms

Additive white Gaussian noise (AWGN) can be expressed as a linear combination of orthogonal waveforms in the same way as signals. For the signal detection problem, the noise can be partitioned into two components,

$$n(t) = \hat{n}(t) + \tilde{n}(t) \tag{3.14}$$

where

$$\hat{n}(t) = \sum_{j=1}^{N} n_j\psi_j(t) \tag{3.15}$$

is taken to be the noise within the signal space, or the projection of the noise components on the signal coordinates $\psi_1(t), \ldots, \psi_N(t)$, and

$$\tilde{n}(t) = n(t) - \hat{n}(t) \tag{3.16}$$

is defined as the noise outside the signal space. In other words, $\tilde{n}(t)$ may be thought of as the noise that is effectively tuned out by the detector. The symbol $\hat{n}(t)$ represents the noise that will interfere with the detection process. We can express the noise waveform, $n(t)$, as follows:

$$n(t) = \sum_{j=1}^{N} n_j \psi_j(t) + \tilde{n}(t) \tag{3.17}$$

where

$$n_j = \frac{1}{K_j} \int_0^T n(t)\psi_j(t)\, dt \qquad \text{for all } j \tag{3.18}$$

and

$$0 = \int_0^T \tilde{n}(t)\psi_j(t)\, dt \tag{3.19}$$

The interfering portion of the noise, $\hat{n}(t)$, expressed in Equation (3.15) will henceforth be referred to simply as $n(t)$. We can express $n(t)$ by a vector of its coefficients similar to the way we did for signals in Equation (3.6).

$$\mathbf{n} = (n_1, n_2, \ldots, n_N) \tag{3.20}$$

where $\mathbf{n}$ is a random vector with zero mean and Gaussian distribution, and where the noise components n_i ($i = 1, \ldots, N$) are independent.

3.2.2.4 Variance of White Noise

White noise is an *idealized process* with two-sided power spectral density equal to a constant, $N_0/2$, for all frequencies from $-\infty$ to $+\infty$. Hence the noise variance (average noise power, since the noise has zero mean) is

$$\sigma^2 = \text{var}\,[n(t)] = \int_{-\infty}^{\infty} \left(\frac{N_0}{2}\right) df = \infty \tag{3.21}$$

Although the variance for AWGN is infinite, the variance for *filtered* AWGN is finite. For example, if AWGN is correlated with one of a set of orthonormal functions $\psi_j(t)$, the variance of the correlator output is given by

$$\sigma^2 = \text{var}\,(n_j) = \mathbf{E}\left\{\left[\int_0^T n(t)\psi_j(t)\, dt\right]^2\right\} = \frac{N_0}{2} \tag{3.22}$$

The proof of Equation (3.22) is given in Appendix C. Henceforth we shall assume that the noise of interest in the detection process is the output noise of a correlator or matched filter with variance $\sigma^2 = N_0/2$ as expressed in Equation (3.22).

3.3 DIGITAL BANDPASS MODULATION TECHNIQUES

Bandpass modulation (either analog or digital) is the process by which an information signal is converted to a sinusoidal waveform; for digital modulation, such a sinusoid of duration T is referred to as a digital symbol. The sinusoid has just three features that can be used to distinguish it from other sinusoids: amplitude, frequency, and phase. Thus bandpass modulation can be defined as the process whereby the amplitude, frequency, or phase of an RF carrier, or a combination of them, is varied in accordance with the information to be transmitted. The general form of the carrier wave, $s(t)$, is as follows:

$$s(t) = A(t) \cos \theta(t) \tag{3.23}$$

where $A(t)$ is the time-varying amplitude and $\theta(t)$ is the time-varying angle. It is convenient to write

$$\theta(t) = \omega_0 t + \phi(t) \tag{3.24}$$

so that

$$s(t) = A(t) \cos [\omega_0 t + \phi(t)] \tag{3.25}$$

where ω_0 is the *radian frequency* of the carrier and $\phi(t)$ is the *phase*. The terms f and ω will each be used to denote frequency. When f is used, frequency in hertz is intended; when ω is used, frequency in radians per second is intended. The two frequency parameters are related by $\omega = 2\pi f$.

The basic *digital modulation/demodulation* types are listed in Figure 3.4. When the receiver exploits knowledge of the carrier's phase to detect the signals, the process is called *coherent detection*; when the receiver does not utilize such phase reference information, the process is called *noncoherent detection*. In digital communications, the terms *demodulation* and *detection* are used somewhat interchangeably, although demodulation emphasizes removal of the carrier, and detection includes the process of symbol decision. In ideal coherent detection, there is available at the receiver a prototype of each possible arriving signal. These prototype waveforms attempt to duplicate the transmitted signal set in every respect, even RF phase. The receiver is then said to be *phase locked* to the incoming signal. During detection, the receiver multiplies and integrates (correlates) the incoming signal with each of its prototype replicas. Under the heading of coherent modulation/demodulation in Figure 3.4 are listed phase shift keying (PSK), frequency shift keying (FSK), amplitude shift keying (ASK), continuous phase modulation (CPM), and hybrid combinations. The basic bandpass modulation formats are discussed in this chapter. Some specialized formats, such as offset quadrature PSK (OQPSK), minimum shift keying (MSK) belonging to the CPM class, and quadrature amplitude modulation (QAM), are treated in Chapter 7.

Noncoherent demodulation refers to systems employing demodulators that are designed to operate without knowledge of the absolute value of the incoming signal's phase; therefore, phase estimation is not required. Thus the advantage of noncoherent over coherent systems is reduced complexity, and the price paid is increased probability of error (P_E). In Figure 3.4 the modulation/demodulation

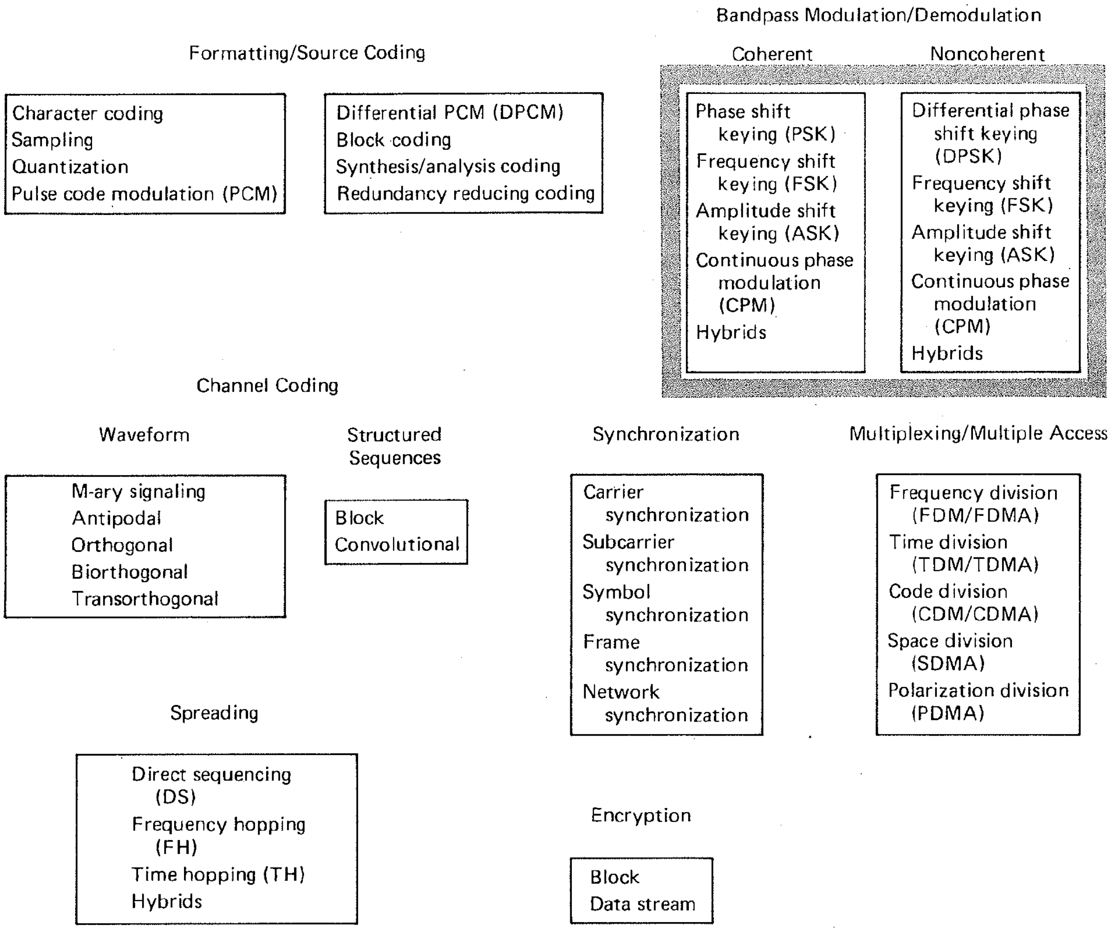


Figure 3.4 Basic digital communication transformations.

types that are listed in the noncoherent column, DPSK, FSK, ASK, CPM, and hybrids, are similar to those listed in the coherent column. We had implied that phase information is not used for noncoherent reception; how do you account for the fact that there is a form of phase shift keying under the noncoherent heading? It turns out that an important form of PSK can be classified as noncoherent (or differentially coherent) since it does not require a reference in phase with the received carrier. This "pseudo-PSK," termed *differential PSK* (DPSK), utilizes phase information of the prior symbol as a phase reference for detecting the current symbol. This is described in Sections 3.6.1 and 3.6.2.

Figure 3.5 illustrates examples of the most common digital modulation formats: PSK, FSK, ASK, and a hybrid combination of ASK and PSK (ASK/PSK or APK). The first column lists the analytic expression, the second is a typical pictorial of the waveform versus time, and the third is a vectorial schematic, with the orthogonal axes labeled $\{\psi_j(t)\}$. In the general M-ary signaling case, the pro-

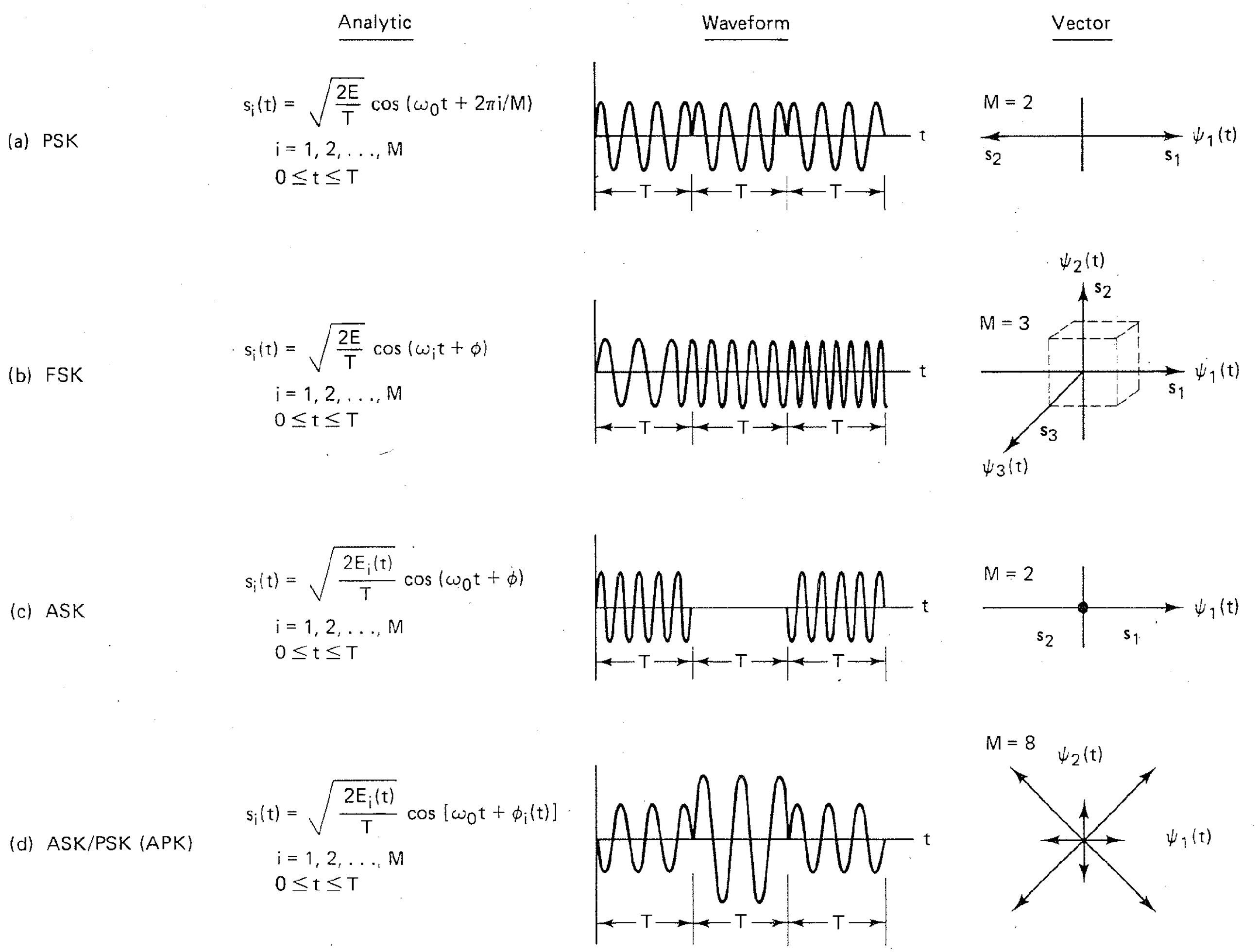


Figure 3.5 Digital modulations. (a) PSK. (b) FSK. (c) ASK. (d) ASK/PSK (APK).

cessor accepts k source bits at a time and instructs the modulator to produce one of an available set of $M = 2^k$ waveform types. Binary modulation, where $k = 1$, is just a special case of M-ary modulation. Each example shown in Figure 3.5 illustrates the set of signal waveforms with a particular value chosen for M.

3.3.1 Phase Shift Keying

Phase shift keying (PSK) was developed during the early days of the deep-space program; PSK is now widely used in both military and commercial communications systems. The general analytic expression for PSK is

$$s_i(t) = \sqrt{\frac{2E}{T}} \cos [\omega_0 t + \phi_i(t)] \qquad \begin{matrix} 0 \le t \le T \\ i = 1, \ldots, M \end{matrix} \tag{3.26}$$

where the phase term, $\phi_i(t)$, will have M discrete values, typically given by

$$\phi_i(t) = \frac{2\pi i}{M} \qquad i = 1, \ldots, M$$

For the binary PSK (BPSK) example in Figure 3.5a, M is 2. The parameter E is symbol energy, T is symbol time duration, and $0 \le t \le T$. In BPSK modulation, the modulating data signal shifts the phase of the waveform, $s_i(t)$, to one of two states, either zero or π (180°). The waveform sketch in Figure 3.5a shows a typical BPSK waveform with its abrupt phase changes at the symbol transitions; if the modulating data stream were to consist of alternating ones and zeros, there would be such an abrupt change at each transition. The signal waveforms can be represented as vectors on a polar plot; the vector length corresponds to the signal amplitude, and the vector direction, for the general M-ary case, corresponds to the signal phase relative to the other $M - 1$ signals in the set. For the BPSK example, the vectorial picture illustrates the two 180° opposing vectors. Signal sets that can be depicted with such opposing vectors are called *antipodal signal sets*.

3.3.2 Frequency Shift Keying

The general analytic expression for FSK modulation is

$$s_i(t) = \sqrt{\frac{2E}{T}} \cos (\omega_i t + \phi) \qquad \begin{matrix} 0 \le t \le T \\ i = 1, \ldots, M \end{matrix} \tag{3.27}$$

where the frequency term, ω_i, will have M discrete values, and the phase term, ϕ, is an arbitrary constant. The FSK waveform sketch in Figure 3.5b illustrates the typical abrupt frequency changes at the symbol transitions. In this example, M has been chosen equal to 3, corresponding to the same number of waveform types (3-ary); note that this $M = 3$ choice for FSK has been selected to emphasize the mutually perpendicular axes. In practice, M is usually a nonzero power of 2 (2, 4, 8, 16, . . .). The signal set is characterized by Cartesian coordinates, such that each of the mutually perpendicular axes represents a sinusoid with a different frequency. As described earlier, signal sets that can be characterized with such mutually perpendicular vectors are called *orthogonal* signals. The required frequency spacing between the orthogonal tones is discussed in Section 3.6.4.

3.3.3 Amplitude Shift Keying

For the ASK example in Figure 3.5c, the general analytic expression is

$$s_i(t) = \sqrt{\frac{2E_i(t)}{T}} \cos (\omega_0 t + \phi) \qquad \begin{array}{l} 0 \le t \le T \\ i = 1, \ldots, M \end{array} \tag{3.28}$$

where the amplitude term, $\sqrt{2E_i(t)/T}$, will have M discrete values, and the phase term, ϕ, is an arbitrary constant. In Figure 3.5c, M has been chosen equal to 2, corresponding to two waveform types. The ASK waveform sketch in the figure can describe a radar transmission example, where the two signal amplitude states would be $\sqrt{2E/T}$ and zero. The vectorial picture utilizes the same phase–amplitude polar coordinates as the PSK example. Here we see a vector corresponding to the maximum-amplitude state, and a point at the origin corresponding to the zero-amplitude state. Binary ASK signaling (also called on–off keying) was one of the earliest forms of digital modulation used in radio telegraphy at the beginning of this century. Simple ASK is no longer widely used in digital communication systems; therefore, it will not be treated in detail.

3.3.4 Amplitude Phase Keying

For the combination of ASK and PSK (APK) example in Figure 3.5d, the general analytic expression

$$s_i(t) = \sqrt{\frac{2E_i(t)}{T}} \cos [\omega_0 t + \phi_i(t)] \qquad \begin{array}{l} 0 \le t \le T \\ i = 1, \ldots, M \end{array} \tag{3.29}$$

illustrates the indexing of both the signal amplitude term and the phase term. The APK waveform picture in Figure 3.5d illustrates some typical simultaneous phase and amplitude changes at the symbol transition times. For this example, M has been chosen equal to 8, corresponding to eight waveforms (8-ary). The figure illustrates a hypothetical eight-vector signal set on the phase–amplitude plane. Four of the vectors are at one amplitude; the other four vectors are at a different amplitude; and each of the vectors is separated by 45°. When the set of M symbols in the two-dimensional signal space are arranged in a rectangular constellation, the signaling is referred to as quadrature amplitude modulation (QAM); examples of QAM are considered in Chapter 7.

The vectorial picture for each of the modulation types described in Figure 3.5 (except the FSK case) is characterized on a plane whose *polar* coordinates represent signal *amplitude* and *phase*. The FSK case is characterized in a *Cartesian* coordinate space, with each axis representing a *frequency tone* ($\cos \omega_i t$) from the M-ary set of orthogonal tones.

3.3.5 Waveform Amplitude Coefficient

The waveform amplitude coefficient appearing in Equations (3.26) to (3.29) has the same general form, $\sqrt{2E/T}$, for all modulation formats. This expression is derived as follows:

$$s(t) = A \cos \omega t \tag{3.30}$$

where A is the peak value of the waveform. Since the peak value of a sinusoidal waveform equals $\sqrt{2}$ times the root-mean-square (rms) value, we can write

$$\begin{aligned} s(t) &= \sqrt{2}A_{\text{rms}} \cos \omega t \\ &= \sqrt{2A_{\text{rms}}^2} \cos \omega t \end{aligned}$$

Assuming the signal to be a voltage or a current waveform, A_{rms}^2 represents average power P (normalized to 1 Ω). Therefore, we can write

$$s(t) = \sqrt{2P} \cos \omega t \tag{3.31}$$

Replacing P watts by E joules/T seconds, we get

$$s(t) = \sqrt{\frac{2E}{T}} \cos \omega t \tag{3.32}$$

We shall use either the amplitude notation, A, in Equation (3.30) or the designation $\sqrt{2E/T}$ in Equation (3.32). Since the *energy* in a signal is the key parameter in determining the error performance of the detection process, it is often more convenient to use the amplitude notation in Equation (3.32) because it facilitates solving directly for the probability of error, P_E, as a function of signal energy.

3.4 DETECTION OF SIGNALS IN GAUSSIAN NOISE

3.4.1 Decision Regions

Consider that the two-dimensional signal space in Figure 3.6 is the locus of the noise-perturbed prototype binary vectors ($\mathbf{s}_1 + \mathbf{n}$) and ($\mathbf{s}_2 + \mathbf{n}$). The noise vector, $\mathbf{n}$, is a zero-mean random vector; hence the received signal vector, $\mathbf{r}$, is a random vector with mean $\mathbf{s}_1$ or $\mathbf{s}_2$. The detector's task after receiving $\mathbf{r}$ is to decide which of the signals, $\mathbf{s}_1$ or $\mathbf{s}_2$, was actually transmitted. The method is usually to decide on the signal classification that yields the minimum expected P_E, although other strategies are possible [4]. For the case where M equals 2, with $\mathbf{s}_1$ and $\mathbf{s}_2$ being equally likely and with the noise being an additive white Gaussian noise (AWGN) process, we will see that the minimum-error decision rule is equivalent to choosing the signal class such that the distance $d(\mathbf{r}, \mathbf{s}_i) = \| \mathbf{r} - \mathbf{s}_i \|$ is minimized, where $\| \mathbf{x} \|$ is called the *norm* or *magnitude* of vector $\mathbf{x}$. This rule is often stated in terms of decision regions. In Figure 3.6, let us construct decision regions in the following

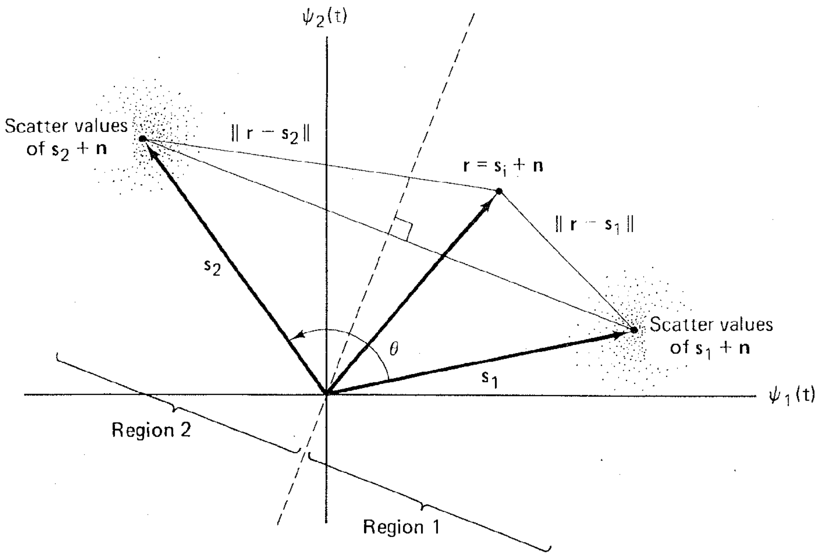


Figure 3.6 Two-dimensional signal space, with arbitrary equal-amplitude vectors $\mathbf{s}_1$ and $\mathbf{s}_2$.

way. Draw a line connecting the tips of the prototype vectors, $\mathbf{s}_1$ and $\mathbf{s}_2$. Next, construct the perpendicular bisector of the connecting line. Notice that this bisector passes through the origin of the space if $\mathbf{s}_1$ and $\mathbf{s}_2$ are equal in amplitude. For this $M = 2$ example in Figure 3.6, the constructed perpendicular bisector represents the locus of points equidistant between $\mathbf{s}_1$ and $\mathbf{s}_2$; hence the bisector describes the boundary between decision region 1 and decision region 2. The *decision rule* for the detector, stated in terms of *decision regions,* is: Whenever the received signal $\mathbf{r}$ is located in region 1, choose signal $\mathbf{s}_1$; when it is located in region 2, choose signal $\mathbf{s}_2$.

3.4.2 Correlation Receiver

In Section 2.9 we treated the detection of *baseband* binary signals in Gaussian noise. Since the detection of *bandpass* signals employs the same concepts, we shall summarize the key findings of that section. We focus particularly on that realization of a matched filter known as a *correlator.* In addition to binary detection, we also consider the more general case of M-ary detection. We assume that the only performance degradation is due to AWGN. The received signal, $r(t)$, is the sum of the transmitted prototype signal plus the random noise:

$$r(t) = s_i(t) + n(t) \qquad \begin{array}{l} 0 \le t \le T \\ i = 1, \ldots, M \end{array} \tag{3.33}$$

Given such a received signal, the detection process consists of *two basic steps.* In the first step, the received waveform, $r(t)$, is reduced to a *single random variable, $z(T)$,* or a *set of random variables, $z_i(T)$* $(i = 1, \ldots, M)$, formed at the output of the correlator(s) at time $t = T$, where T is the symbol duration. In the second step, a symbol decision is made, on the basis of comparing $z(T)$ to a threshold or on the basis of choosing the maximum $z_i(T)$. Step 1 can be thought of as transforming the waveform into a point in the decision space. Step 2 can be thought of as determining *in which decision region* the point is located. For the detector to be optimized (in the sense of minimizing the error probability), it is necessary to optimize the waveform-to-random-variable transformation, by using matched filters or correlators in step 1, and by also optimizing the decision criterion in step 2.

In Sections 2.9.2 and 2.9.3 we found that the matched filter provides the maximum signal-to-noise ratio at the filter output at time $t = T$. We described a correlator as one realization of a matched filter. We can define a *correlation receiver* comprised of M correlators, as shown in Figure 3.7a, that transforms a received waveform, $r(t)$, to a sequence of M numbers or correlator outputs, $z_i(T)$ $(i = 1, \ldots, M)$. Each correlator output is characterized by the following product integration or correlation with the received signal.

$$z_i(T) = \int_0^T r(t)s_i(t)\,dt \qquad i = 1, \ldots, M \tag{3.34}$$

The verb "to correlate" means "to match." The correlators attempt to match the incoming received signal, $r(t)$, with each of the candidate prototype waveforms, $s_i(t)$, known a priori to the receiver. A reasonable decision rule is to choose the waveform, $s_i(t)$, that *matches best* or has the *largest correlation* with $r(t)$. In other words, the decision rule is:

$$\text{Choose the } s_i(t) \text{ whose index corresponds to the max } z_i(T) \tag{3.35}$$

Following Equation (3.4), any signal set, $\{s_i(t)\}$ $(i = 1, \ldots, M)$, can be expressed in terms of some set of basis functions, $\{\psi_j(t)\}$ $(j = 1, \ldots, N)$, where $N \leq M$. Then the bank of M correlators in Figure 3.7a may be replaced with a bank of N correlators, shown in Figure 3.7b, where the set of basis functions $\{\psi_j(t)\}$ form *reference signals*. The decision stage of this receiver consists of logic circuitry for choosing the signal, $s_i(t)$. The choice of $s_i(t)$ is made according to the best match of the coefficients, a_{ij}, seen in Equation (3.4), with the set of outputs $\{z_j(T)\}$. When the prototype waveform set, $\{s_i(t)\}$, is an orthogonal set, the receiver implementation in Figure 3.7a is identical to that in Figure 3.7b (differing perhaps by a scale factor). However, when $\{s_i(t)\}$ is *not* an orthogonal set, the receiver in Figure 3.7b, using N correlators instead of M, with reference signals $\{\psi_j(t)\}$, can represent a cost-effective implementation. We examine such an application for the detection of multiple phase shift keying (MPSK) in Section 3.5.3. For the other applications in this chapter, we shall assume a correlator receiver with reference signals $\{s_i(t)\}$.

In the case of *binary detection*, the correlation receiver can be configured as a single matched filter or product integrator, as shown in Figure 3.8a, with the reference signal being the difference between the binary prototype signals, $s_1(t) - s_2(t)$. The output of the correlator, $z(T)$, is fed directly to the decision stage.

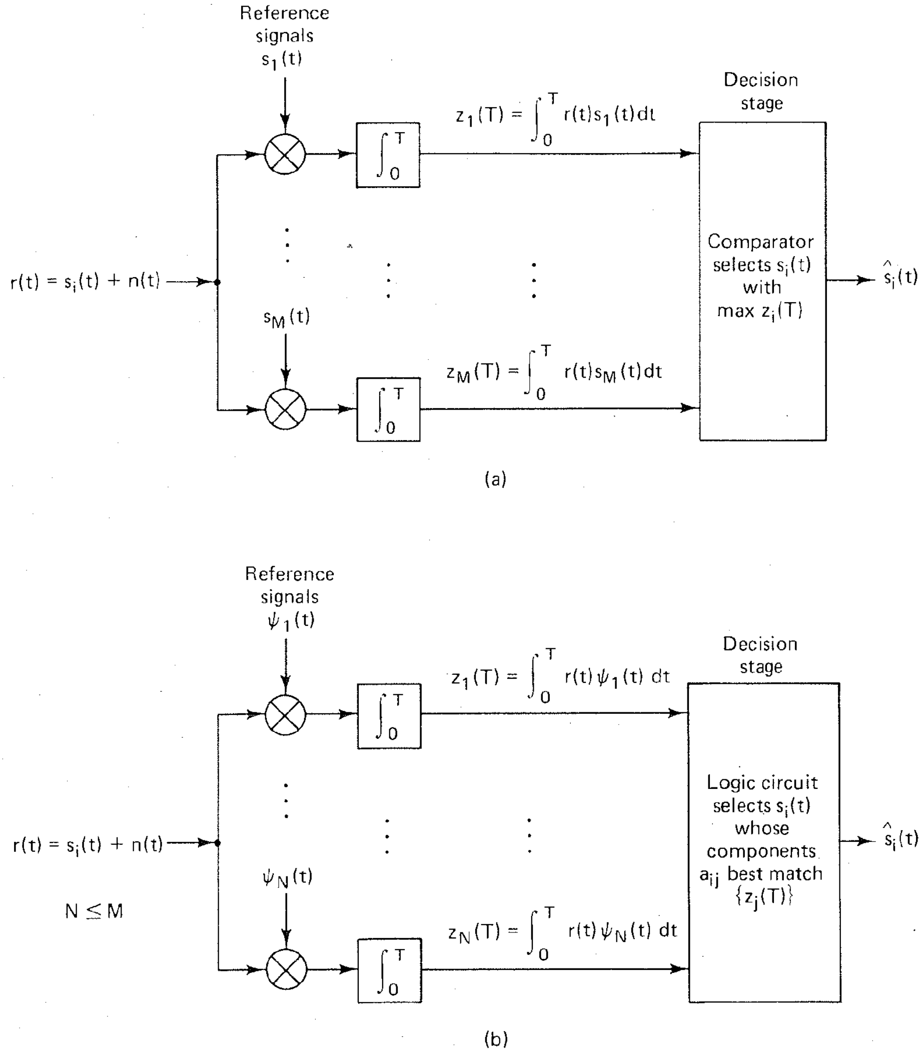
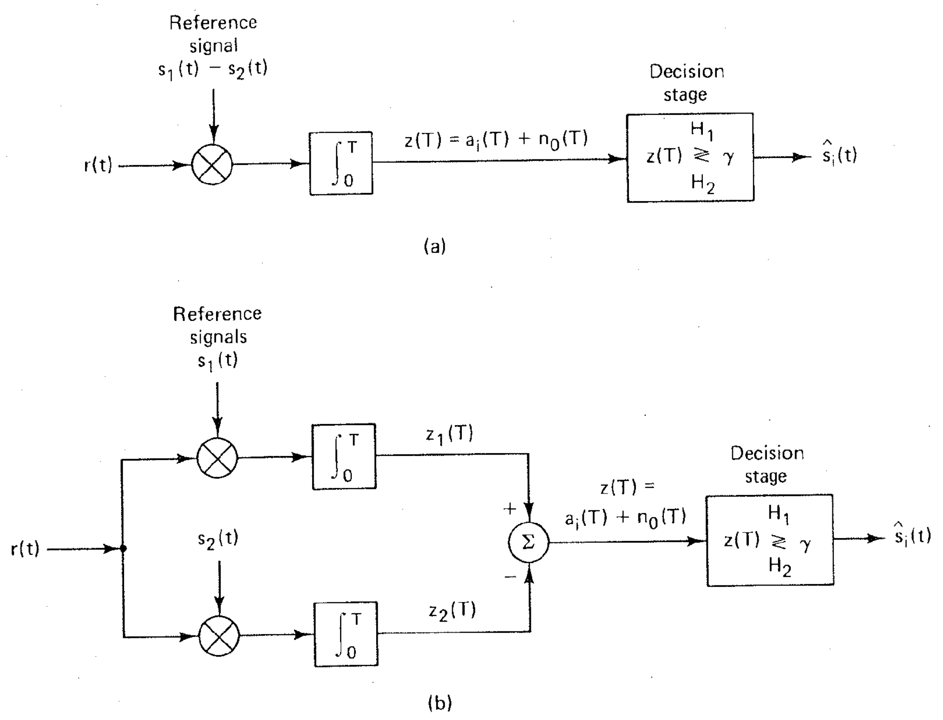

Reference signals
$s_1(t)$
$z_1(T) = \int_0^T r(t)s_1(t)dt$
Decision stage
$\int_0^T$
Comparator selects $s_i(t)$ with max $z_i(T)$
$r(t) = s_i(t) + n(t)$
$\hat{s}_i(t)$
$s_M(t)$
$z_M(T) = \int_0^T r(t)s_M(t)dt$
(a)
Reference signals
$\psi_1(t)$
$z_1(T) = \int_0^T r(t)\psi_1(t)\,dt$
Decision stage
Logic circuit selects $s_i(t)$ whose components a_{ij} best match $\{z_j(T)\}$
$r(t) = s_i(t) + n(t)$
$\hat{s}_i(t)$
$\psi_N(t)$
$N \leq M$
$z_N(T) = \int_0^T r(t)\psi_N(t)\,dt$
(b)

Figure 3.7 (a) Correlator receiver with reference signals $\{s_i(t)\}$. (b) Correlator receiver with reference signals $\{\psi_j(t)\}$.

For binary detection, the correlation receiver can also be drawn, as shown in Figure 3.8b, as two matched filters or product integrators, each of which is matched to one of the prototype reference signals, $s_1(t)$ or $s_2(t)$. The decision stage can then be configured to follow the rule in Equation (3.35), or the correlator outputs, $z_i(T)$ $(i = 1, 2)$, can be differenced to form

$$z(T) = z_1(T) - z_2(T) \tag{3.36}$$

as shown in Figure 3.8b. Then, $z(T)$, called the *test statistic,* is fed to the decision stage, as in the case of the single correlator. In the *absence of noise,* an input



(a)

(b)

Figure 3.8 Binary correlator receiver. (a) Using a single correlator. (b) Using two correlators.

waveform, $s_i(t)$, yields the output, $z(T) = a_i(T)$, a signal-only component. The input noise, $n(t)$, is a Gaussian random process. Since the correlator is a *linear* device, the output noise is also a Gaussian random process [4]. Thus the output of the correlator, sampled at $t = T$, yields

$$z(T) = a_i(T) + n_0(T) \qquad i = 1, 2$$

where $n_0(T)$ is the noise component. To shorten the notation we sometimes express $z(T)$ as $a_i + n_0$. The noise component, n_0, is a zero-mean *Gaussian random variable*, and thus $z(T)$ is a *Gaussian random variable* with a mean of either a_1 or a_2 depending on whether a binary one or binary zero was sent.

3.4.2.1 Binary Decision Threshold

For the random variable, $z(T)$, Figure 3.9 illustrates the two conditional probability density functions (pdfs), $p(z|s_1)$ and $p(z|s_2)$, with mean value of a_1 and a_2, respectively (these pdfs are also called the *likelihood* of s_1 and the *likelihood*

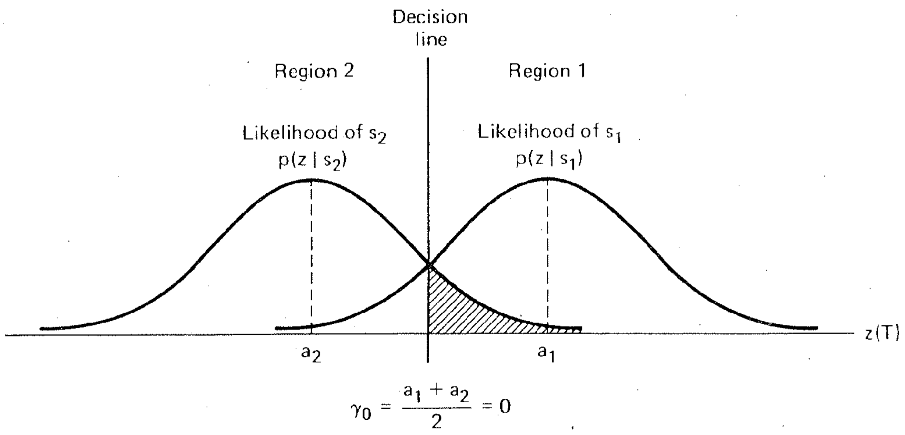


Figure 3.9 Conditional probability density functions: $p(z|s_1)$, $p(z|s_2)$.

of s_2, respectively):

$$p(z|s_1) = \frac{1}{\sigma_0\sqrt{2\pi}} \exp\left[-\frac{1}{2}\left(\frac{z - a_1}{\sigma_0}\right)^2\right] \tag{3.37a}$$

$$p(z|s_2) = \frac{1}{\sigma_0\sqrt{2\pi}} \exp\left[-\frac{1}{2}\left(\frac{z - a_2}{\sigma_0}\right)^2\right] \tag{3.37b}$$

where σ_0^2 is the noise variance. In Figure 3.9 the rightmost likelihood, $p(z|s_1)$, illustrates the probability density of the detector output, $z(T)$, given that $s_1(t)$ was transmitted. Similarly, the leftmost likelihood $p(z|s_2)$, illustrates the probability density of $z(T)$ given that $s_2(t)$ was transmitted. The abscissa, $z(T)$, represents the full range of possible sample output values from the correlation receiver in Figure 3.8.

With regard to optimizing the binary decision threshold for deciding in which region a received signal is located, we found in Section 2.9.1 that the *minimum error* criterion for equally likely binary signals corrupted by Gaussian noise can be stated as follows:

$$z(T) \underset{H_2}{\overset{H_1}{\gtrless}} \frac{a_1 + a_2}{2} = \gamma_0 \tag{3.38}$$

where a_1 is the signal component of $z(T)$ when $s_1(t)$ is transmitted, and a_2 is the signal component of $z(T)$ when $s_2(t)$ is transmitted. The threshold level, γ_0, represented by $(a_1 + a_2)/2$, is the *optimum threshold* for minimizing the probability of making an incorrect decision given equally likely signals and symmetrical likelihoods. The decision rule in Equation (3.38) states that hypothesis H_1 should be selected [equivalent to deciding that signal $s_1(t)$ was sent] if $z(T) > \gamma_0$, and hypothesis H_2 should be selected [equivalent to deciding that $s_2(t)$ was sent] if $z(T) < \gamma_0$. If $z(T) = \gamma_0$, the decision can be an arbitrary one. For equal-energy, equally likely antipodal signals, where $s_1(t) = -s_2(t)$ and $a_1 = -a_2$, the optimum decision rule becomes

$$z(T) \underset{H_2}{\overset{H_1}{\gtrless}} \gamma_0 = 0 \qquad (3.39a)$$

or

$$\begin{array}{ll} \text{decide } s_1(t) & \text{if } z_1(T) > z_2(T) \\ \text{decide } s_2(t) & \text{otherwise} \end{array} \qquad (3.39b)$$

In the next section we illustrate the use of correlators and matched filters for the coherent detection of PSK and FSK modulation. In later sections we consider noncoherent detection, and we treat the error performance of various modulation types.

3.5 COHERENT DETECTION

3.5.1 Coherent Detection of PSK

The detector shown in Figure 3.7 can be used for the coherent detection of any digital waveforms. Such a correlating detector is often referred to as a *maximum likelihood detector*. Consider the following binary PSK (BPSK) example. Let

$$s_1(t) = \sqrt{\frac{2E}{T}} \cos(\omega_0 t + \phi) \qquad 0 \le t \le T \qquad (3.40a)$$

$$s_2(t) = \sqrt{\frac{2E}{T}} \cos(\omega_0 t + \phi + \pi)$$

$$= -\sqrt{\frac{2E}{T}} \cos(\omega_0 t + \phi) \qquad 0 \le t \le T \qquad (3.40b)$$

$$n(t) = \text{zero-mean white Gaussian random process}$$

where the phase term, ϕ, is an arbitrary constant, so that the analysis is unaffected by setting $\phi = 0$. The parameter, E, is the signal energy per symbol, and T is the symbol duration. For this antipodal case, only a single basis function is needed. If an orthonormal signal space is assumed in Equations (3.4) and (3.5) (i.e., $K_j = 1$), we can express a basis function, $\psi_1(t)$, as follows:

$$\psi_1(t) = \sqrt{\frac{2}{T}} \cos \omega_0 t \qquad \text{for } 0 \le t \le T \qquad (3.41)$$

Thus we may express the transmitted signals $s_i(t)$ in terms of $\psi_1(t)$ and the coefficients $a_{i1}(t)$:

$$s_i(t) = a_{i1}\psi_1(t) \tag{3.42a}$$

$$s_1(t) = a_{11}\psi_1(t) = \sqrt{E}\psi_1(t) \tag{3.42b}$$

$$s_2(t) = a_{21}\psi_1(t) = -\sqrt{E}\psi_1(t) \tag{3.42c}$$

Assume that $s_1(t)$ was transmitted. Then the expected values of the product integrators in Figure 3.7b, with reference signals $\psi_1(t)$ and $-\psi_1(t)$, are found as follows:

$$\mathrm{E}\{z_1|s_1\} = \mathrm{E}\left\{\int_0^T \sqrt{E}\psi_1^2(t) + n(t)\psi_1(t)\,dt\right\} \tag{3.43a}$$

$$\mathrm{E}\{z_2|s_1\} = \mathrm{E}\left\{\int_0^T -\sqrt{E}\psi_1^2(t) - n(t)\psi_1(t)\,dt\right\} \tag{3.43b}$$

$$\mathrm{E}\{z_1|s_1\} = \mathrm{E}\left\{\int_0^T \frac{2}{T}\sqrt{E}\cos^2\omega_0 t + n(t)\sqrt{\frac{2}{T}}\cos\omega_0 t\,dt\right\} = \sqrt{E} \tag{3.44a}$$

$$\mathrm{E}\{z_2|s_1\} = \mathrm{E}\left\{\int_0^T -\frac{2}{T}\sqrt{E}\cos^2\omega_0 t - n(t)\sqrt{\frac{2}{T}}\cos\omega_0 t\,dt\right\} = -\sqrt{E} \tag{3.44b}$$

where $\mathrm{E}\{\cdot\}$ denotes the ensemble average, referred to as the *expected value*. Equation (3.44) follows because $\mathrm{E}\{n(t)\} = 0$. The decision stage must decide which signal was transmitted by determining its location within the signal space. For this example, the choice of $\psi_1(t) = \sqrt{2/T}\cos\omega_0 t$ normalizes $\mathrm{E}\{z_i(T)\}$ to be $\pm\sqrt{E}$. The prototype signals $\{s_i(t)\}$ are the same as the reference signals $\{\psi_j(t)\}$ except for the normalizing scale factor. The decision stage chooses the signal with the largest value of $z_i(T)$. Thus, the received signal in this example is judged to be $s_1(t)$. The error performance for such coherently detected BPSK systems is treated in Section 3.7.1.

3.5.2 Sampled Matched Filter

In Section 2.9.2 we discussed the basic characteristic of the matched filter—namely, that its impulse response is a delayed version of the mirror image (rotated on the $t = 0$ axis) of the input signal waveform. Therefore, if the signal waveform is $s(t)$, its mirror image is $s(-t)$ and the mirror image delayed by T seconds is $s(T - t)$. The impulse response, $h(t)$, of a filter matched to $s(t)$ is then described by

$$h(t) = \begin{cases} s(T - t) & 0 \le t \le T \\ 0 & \text{elsewhere} \end{cases} \tag{3.45}$$

Figure 3.10a illustrates how a matched filter can be implemented using digital hardware. The input signal, $r(t)$, is comprised of the prototype signal, $s(t)$, plus noise, $n(t)$. The bandwidth of the signal is $W = 1/2T_s$, where the Nyquist sampling

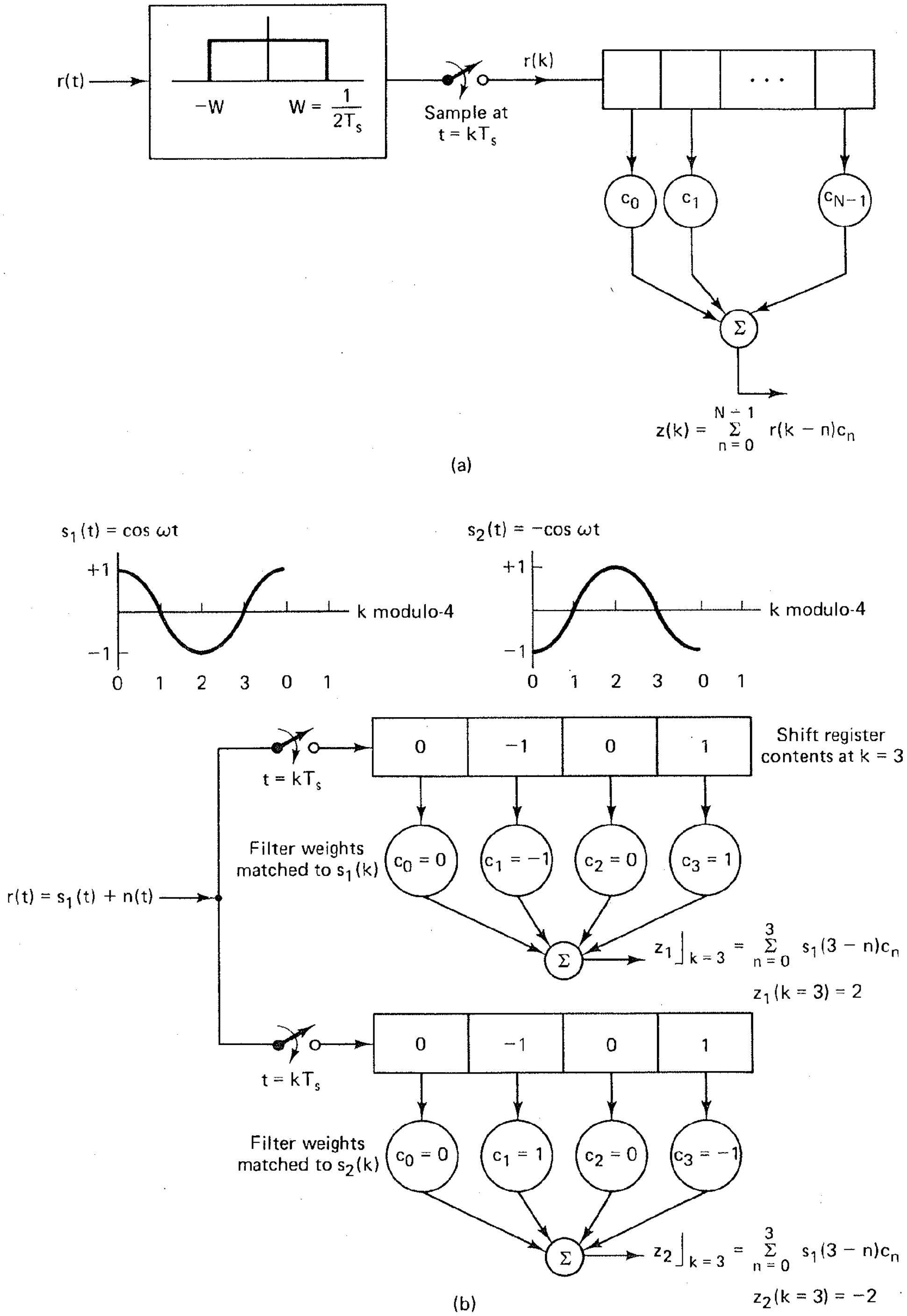


Figure 3.10 (a) Sampled matched filter. (b) Sampled matched filter detection example, in the absence of noise.

rate $f_s = 2W = 1/T_s$; hence the sampling interval is equal to T_s. At the clock times of $t = kT_s$, the analog signal is sampled and the samples are shifted into the register of Figure 3.10a from left to right. The shift register with its coefficients c_0 to c_{N-1} approximate a matched filter. Once the received signal has been sampled, the continuous time notation t is changed to kT_s or simply k to reflect the sampled notation

$$r(k) = s(k) + n(k) \qquad k = 0, 1, \ldots$$

where k represents a sample index. The output, $z(k)$, of the sampled matched filter, at a time corresponding to the kth sample is

$$z(k) = \sum_{n=0}^{N-1} r(k - n)c_n \qquad k = 0, 1, \ldots, \text{modulo-}N \tag{3.46}$$

where x modulo-y is defined as the remainder of dividing x by y. For the binary demodulation application, $z_i(k)(i = 1, 2)$ outputs are compared to a threshold at each value of $k = N - 1$ corresponding to the end of a symbol. The c_n values are the filter weights constituting the filter impulse response that is matched to the signal, where n is the index of the weights and the register stages (from left to right) and k is the index of the samples as they are produced by the sampler. One can see the similarity between the convolution integral of Equation (2.56) and the summation of Equation (3.46), especially with regard to the mirror-image rotation of one of the functions prior to multiplication. Since we assume the noise to have zero mean, the expected value of a received sample for the binary case is expressed as

$$\mathrm{E}\{r(k)\} = s_i(k) \qquad i = 1, 2 \tag{3.47}$$

If $s_1(t)$ had been transmitted, the expected matched filter outputs would be

$$\mathrm{E}\{z_i(k)\} = \sum_{n=0}^{N-1} s_1(k - n)c_n \tag{3.48}$$

where the filter weights, c_n, are matched to the corresponding $s_i(k)$ for each branch.

Example 3.2 Sampled Matched Filter

Consider the BPSK waveform set

$$s_1(t) = \cos \omega t$$

and

$$s_2(t) = -\cos \omega t$$

Illustrate how a *sampled* matched filter or correlator, as shown in Figure 3.10a, can be used to detect a received signal, say $s_1(t)$, from the BPSK waveform set, in the absence of noise.

Solution

First, the waveform is sampled so that $s_1(t)$ is transformed into the set of samples, $\{s_1(k)\}$. The sampled matched filter receiver will be shown with two branches, following the analog implementation in Figure 3.8b. The top branch is made up of shift registers and coefficients matched to the $\{s_1(k)\}$ sample points. The bottom branch is similarly matched to the $\{s_2(k)\}$ sample points. The four equally spaced sample points ($k = 0, 1, 2, 3$) for each of the $\{s_i(k)\}$ are as follows (see Figure 3.10b):

$$s_1(k = 0) = 1, \qquad s_1(k = 1) = 0, \qquad s_1(k = 2) = -1, \qquad s_1(k = 3) = 0$$

$$s_2(k = 0) = -1, \qquad s_2(k = 1) = 0, \qquad s_2(k = 2) = 1, \qquad s_2(k = 3) = 0$$

The c_n coefficients represent the delayed mirror-image rotation of the signal to which the filter is matched. Therefore, $c_n = s_i(N - 1 - n)$, where $n = 0, \ldots, N - 1$, and we can write $c_0 = s_i(3)$, $c_1 = s_i(2)$, $c_2 = s_i(1)$, $c_3 = s_i(0)$. It is here that the reader can gain some insight as to why the convolution operation (with its mirror-image rotation) results in the appropriate lining up of the received signal samples with the weights (reference signal).

Consider the top branch in Figure 3.10b. At the $k = 0$ clock time, the first sample, $s_1(k = 0) = 1$, enters the leftmost stage of each register. At the next clock time, the second sample, $s_1(k = 1) = 0$, enters the leftmost stage of each register; at this same time the first sample, $s_1(k = 0) = 1$, has been shifted to the next right stage in each register, and so on. At the $k = 3$ clock time the sample, $s_1(k = 3) = 0$, enters the leftmost stage; by this time the first sample, $s_1(k = 0) = 1$, has been shifted into the rightmost stage. The four signal samples are now located in the registers in mirror-image arrangement compared to the way the prototype waveform, $s_1(t)$, is drawn in Figure 3.10b. The task of the demodulator is to find the best match to the incoming signal; the demodulator matches the reference coefficients of each branch with the incoming signal samples, in the order in which the samples arrive. Hence the convolution operation is an appropriate expression for describing the alignment of the incoming waveform samples with the reference coefficients, to maximize the correlation in the proper branch.

3.5.3 Coherent Detection of Multiple Phase Shift Keying

Figure 3.11 illustrates the signal space for a multiple phase shift keying (MPSK) signal set; the figure describes a four-level (4-ary) PSK or quadriphase shift keying (QPSK) example ($M = 4$). Binary source digits are collected two at a time, and for each symbol interval the two sequential digits instruct the modulator as to which of the four waveforms to produce. For typical coherent M-ary PSK (MPSK) systems, $s_i(t)$ can be expressed as

$$s_i(t) = \sqrt{\frac{2E}{T}} \cos\left(\omega_0 t - \frac{2\pi i}{M}\right) \qquad \begin{array}{l} 0 \le t \le T \\ i = 1, \ldots, M \end{array} \tag{3.49}$$

where E is the energy content of $s_i(t)$ over each symbol duration T, and ω_0 is the carrier frequency. If an orthonormal signal space is assumed in Equations (3.4) and (3.5), we can choose a convenient set of axes, as follows:

$$\psi_1(t) = \sqrt{\frac{2}{T}} \cos \omega_0 t \tag{3.50a}$$

$$\psi_2(t) = \sqrt{\frac{2}{T}} \sin \omega_0 t \tag{3.50b}$$

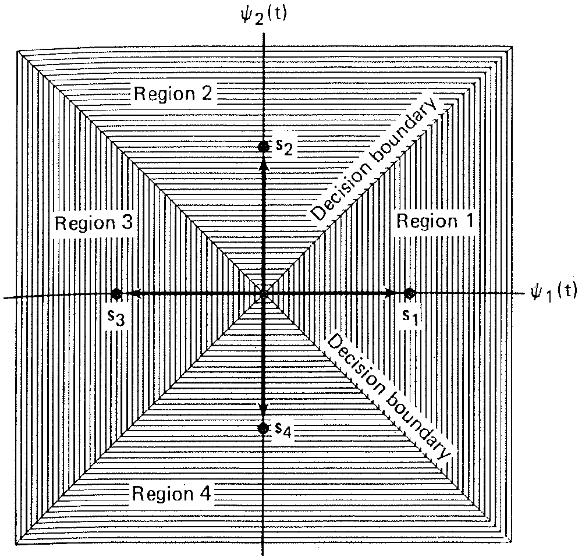


Figure 3.11 Signal space and decision regions for a QPSK system.

where the amplitude $\sqrt{2/T}$ has been chosen to normalize the expected output of the detector, as was done in Section 3.5.1. Now $s_i(t)$ can be written in terms of these orthonormal coordinates, giving

$$s_i(t) = a_{i1}\psi_1(t) + a_{i2}\psi_2(t) \qquad \begin{matrix} 0 \le t \le T \\ i = 1, \ldots, M \end{matrix} \tag{3.51a}$$

$$= \sqrt{E}\cos\left(\frac{2\pi i}{M}\right)\psi_1(t) + \sqrt{E}\sin\left(\frac{2\pi i}{M}\right)\psi_2(t) \tag{3.51b}$$

Notice that Equation (3.51) describes a set of M multiple phase waveforms (intrinsically nonorthogonal) in terms of only two orthogonal carrier-wave components. The $M = 4$ (QPSK) case is unique among MPSK signal sets in the sense that the QPSK waveform set is represented by a combination of antipodal and orthogonal members. The decision boundaries partition the signal space into $M = 4$ regions; the construction is similar to the procedure outlined in Section 3.4.1 and Figure 3.6 for $M = 2$. The decision rule for the detector (see Figure 3.11) is to decide that $s_1(t)$ was transmitted if the received signal vector falls in region 1, that $s_2(t)$ was transmitted if the received signal vector falls in region 2, and so on. In other words, the decision rule is to choose the ith waveform if $z_i(T)$ is the largest of the correlator outputs (seen in Figure 3.7).

The form of the correlator shown in Figure 3.7a implies that there are always M product correlators used for the demodulation of MPSK signals. The figure infers that for each of the M branches, a reference signal with the appropriate phase shift is configured. In practice, the implementation of an MPSK demodulator follows Figure 3.7b, requiring only $N = 2$ product integrators regardless of the size of the signal set M. The savings in implementation is possible because any arbitrary integrable waveform set can be expressed as a linear combination of orthogonal waveforms, as shown in Section 3.2.2. Figure 3.12 illustrates such a demodulator. The received signal, $r(t)$, can be expressed by combining Equations (3.50) and (3.51) as follows:

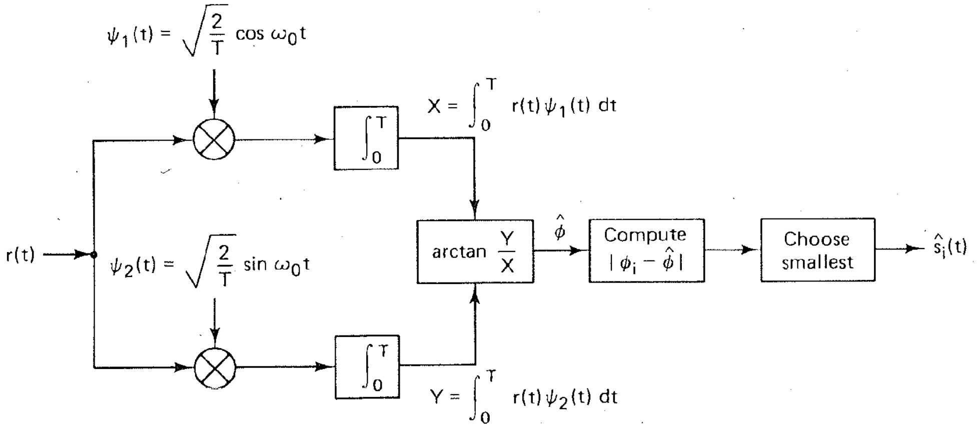
$$r(t) = \sqrt{\frac{2E}{T}}(\cos\phi_i \cos\omega_0 t + \sin\phi_i \sin\omega_0 t) + n(t) \qquad \begin{matrix} 0 \le t \le T \\ i = 1, \ldots, M \end{matrix} \tag{3.52}$$

where $\phi_i = 2\pi i/M$, and $n(t)$ is a zero-mean white Gaussian noise process. Notice in Figure 3.12 that there are only two reference waveforms or basis functions, $\psi_1(t) = \sqrt{2/T}\cos\omega_0 t$ for the upper correlator and $\psi_2(t) = \sqrt{2/T}\sin\omega_0 t$ for the lower correlator. The upper correlator computes

$$X = \int_0^T r(t)\psi_1(t)\,dt \tag{3.53}$$

and the lower correlator computes

$$Y = \int_0^T r(t)\psi_2(t)\,dt \tag{3.54}$$



$\psi_1(t) = \sqrt{\frac{2}{T}}\cos\omega_0 t$
$X = \int_0^T r(t)\psi_1(t)\,dt$
$\int_0^T$
r(t)
$\psi_2(t) = \sqrt{\frac{2}{T}}\sin\omega_0 t$
arctan $\frac{Y}{X}$
$\hat{\phi}$
Compute $|\phi_i - \hat{\phi}|$
Choose smallest
$\hat{s}_i(t)$
$\int_0^T$
$Y = \int_0^T r(t)\psi_2(t)\,dt$

Figure 3.12 Demodulator for MPSK signals.

Figure 3.13 illustrates that the computation of the received phase angle $\hat{\phi}$ can be accomplished by computing the arctan of Y/X, where X can be thought of as the in-phase component of the received signal, Y is the quadrature component, and $\hat{\phi}$ is a noisy estimate of the transmitted ϕ_i. In other words, the upper correlator of Figure 3.12 produces an output X, the magnitude of the in-phase projection of the vector **r**, and the lower correlator produces an output Y, the magnitude of the quadrature projection of the vector **r**. The X and Y outputs of the correlators feed into the block marked arctan (Y/X). The resulting value of the angle $\hat{\phi}$ is compared with each of the stored prototype phase angles, ϕ_i. The demodulator selects the ϕ_i that is closest to the angle $\hat{\phi}$. In other words, the demodulator computes $|\phi_i - \hat{\phi}|$ for each of the ϕ_i prototypes and chooses the ϕ_i yielding the smallest output.

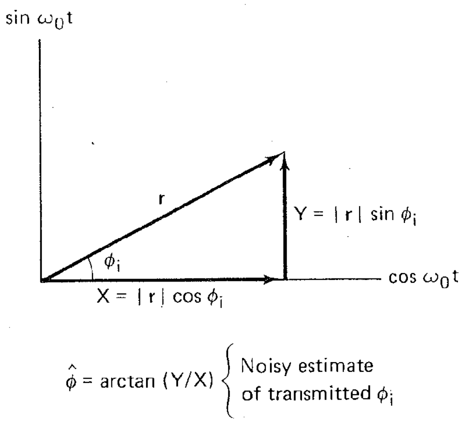


Figure 3.13 In-phase and quadrature components of the received signal vector **r**.

3.5.4 Coherent Detection of FSK

FSK modulation is characterized by the information being contained in the frequency of the carrier. A typical set of FSK signal waveforms was described in Equation (3.27) as

$$s_i(t) = \sqrt{\frac{2E}{T}} \cos(\omega_i t + \phi) \qquad \begin{array}{l} 0 \leq t \leq T \\ i = 1, \ldots, M \end{array}$$

where E is the energy content of $s_i(t)$ over each symbol duration T, and $(\omega_{i+1} - \omega_i)$ is typically assumed to be an integral multiple of π/T. The phase term, ϕ, is an arbitrary constant and can be set equal to zero. Assuming that the basis functions $\psi_1(t)$, $\psi_2(t)$, . . . , $\psi_N(t)$ form an orthonormal set, the most useful form for $\{\psi_j(t)\}$ is shown below.

$$\psi_j(t) = \sqrt{\frac{2}{T}} \cos \omega_j t \qquad j = 1, \ldots, N \tag{3.55}$$

where, as before, the amplitude $\sqrt{2/T}$ normalizes the expected output of the detector. From Equation (3.5) we can write

$$a_{ij} = \int_0^T \sqrt{\frac{2E}{T}} \cos(\omega_i t) \sqrt{\frac{2}{T}} \cos \omega_j t \, dt \tag{3.56}$$

Therefore,

$$a_{ij} = \begin{cases} \sqrt{E} & \text{for } i = j \\ 0 & \text{otherwise} \end{cases} \tag{3.57}$$

In other words, the ith prototype signal vector is located on the ith coordinate axis at a displacement $\sqrt{E}$ from the origin of the signal space. In this scheme, for the general M-ary case, the distance between any two prototype signal vectors $\mathbf{s}_i$ and $\mathbf{s}_j$ is constant:

$$d(\mathbf{s}_i, \mathbf{s}_j) = \| \mathbf{s}_i - \mathbf{s}_j \| = \sqrt{2E} \qquad \text{for } i \neq j \tag{3.58}$$

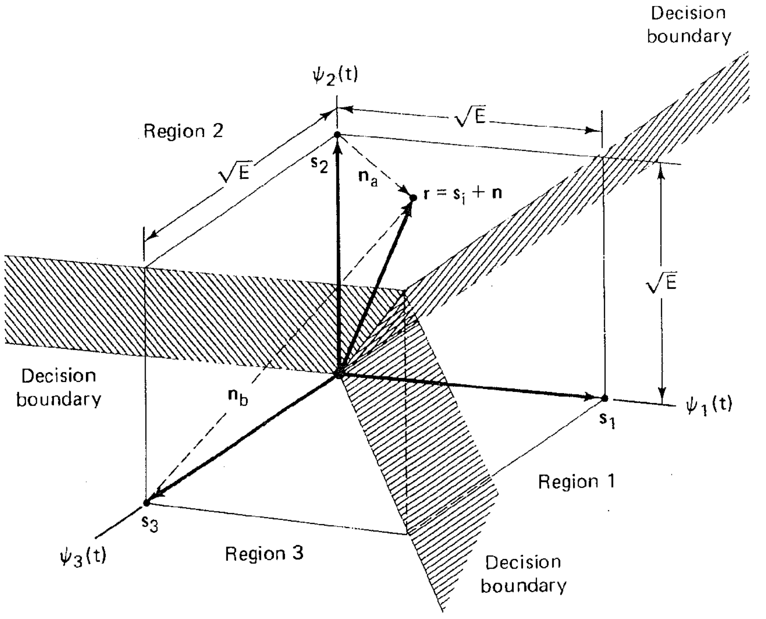


Figure 3.14 Partitioning the signal space for a 3-ary FSK signal.

Figure 3.14 illustrates the prototype signal vectors and the decision regions for a 3-ary ($M = 3$) coherently detected FSK system. As in the PSK case, the signal space is partitioned into M distinct regions, each containing one prototype signal vector; here, because the decision region is three-dimensional, the decision boundaries are planes instead of lines. The optimum decision rule is to decide that the transmitted signal belongs to the class whose index corresponds to the region where the received signal is found. In Figure 3.14, a received signal vector $\mathbf{r}$ is shown in region 2. Using the decision rule stated above, the detector classifies $\mathbf{r}$ as signal $\mathbf{s}_2$. Since the noise is a Gaussian random vector, there is a probability greater than zero that $\mathbf{r}$ could have been produced by some signal other than $\mathbf{s}_2$. For example, if the transmitter had sent $\mathbf{s}_2$, then $\mathbf{r}$ would be the sum of signal plus noise, $\mathbf{s}_2 + \mathbf{n}_a$, and the decision to choose $\mathbf{s}_2$ is correct; however, if the transmitter had actually sent $\mathbf{s}_3$, then $\mathbf{r}$ would be the sum of signal plus noise, $\mathbf{s}_3 + \mathbf{n}_b$ and the decision to select $\mathbf{s}_2$ is an error. The error performance of coherently detected FSK systems is treated in Section 3.7.3.

3.6 NONCOHERENT DETECTION

3.6.1 Detection of Differential PSK

The name *differential PSK* (DPSK) sometimes needs clarification because two separate aspects of the modulation/demodulation format are being referred to: the encoding procedure and the detection procedure. The term *differential encoding* refers to the procedure of encoding the data differentially; that is, the presence of a binary one or zero is manifested by the symbol's similarity or difference when compared to the preceding symbol. The term *differentially coherent detection* of differentially encoded PSK, the usual meaning of DPSK, refers to a detection scheme often classified as noncoherent because it does not require a reference in phase with the received carrier. Sometimes, differentially encoded PSK is *coherently* detected. This will be discussed in Section 3.7.2.

With noncoherent systems, no attempt is made to determine the actual value of the phase of the incoming signal. Therefore, if the transmitted waveform is

$$s_i(t) = \sqrt{\frac{2E}{T}} \cos\left[\omega_0 t + \theta_i(t)\right] \qquad \begin{array}{l} 0 \le t \le T \\ i = 1, \ldots, M \end{array}$$

the received signal can be characterized by

$$r(t) = \sqrt{\frac{2E}{T}} \cos\left[\omega_0 t + \theta_i(t) + \alpha\right] + n(t) \qquad \begin{array}{l} 0 \le t \le T \\ i = 1, \ldots, M \end{array} \tag{3.59}$$

where α is an arbitrary constant and is typically assumed to be a random variable uniformly distributed between zero and 2π, and $n(t)$ is an AWGN process.

For coherent detection, matched filters (or their equivalents) are used; for noncoherent detection, this is not possible because the matched filter output is a function of the unknown angle α. However, if we assume that α varies slowly relative to two period times $(2T)$, the phase difference between two successive waveforms, $\theta_j(T_1)$ and $\theta_k(T_2)$ is independent of α, that is,

$$\left[\theta_k(T_2) + \alpha\right] - \left[\theta_j(T_1) + \alpha\right] = \theta_k(T_2) - \theta_j(T_1) = \phi_i(T_2) \tag{3.60}$$

The basis for *differentially coherent detection* of differentially encoded PSK (DPSK) is as follows. The carrier phase of the previous signaling interval can be used as a phase reference for demodulation. Its use requires *differential encoding* of the message sequence at the transmitter since the information is carried by the difference in phase between two successive waveforms. To send the ith message $(i = 1, 2, \ldots, M)$, the present signal waveform must have its phase advanced by $\phi_i = 2\pi i/M$ radians over the previous waveform. The detector, in general, calculates the coordinates of the incoming signal by correlating it with locally generated waveforms such as $\sqrt{2/T} \cos \omega_0 t$ and $\sqrt{2/T} \sin \omega_0 t$. The detector then measures the angle between the currently received signal vector and the previously received signal vector, as illustrated in Figure 3.15.

In general, DPSK signaling performs less efficiently than PSK, because the errors in DPSK tend to propagate (to adjacent symbol times) due to the correlation between signaling waveforms. One way of viewing the difference between PSK and DPSK is that the former compares the received signal with a clean reference; in the latter, however, two noisy signals are compared with each other. We might say that there is twice as much noise associated with DPSK signaling compared to PSK signaling. Consequently, as a first guess, we might estimate that DPSK manifests a degradation of approximately 3 dB when compared with PSK; this degradation decreases rapidly with increasing signal-to-noise ratio. The trade-off

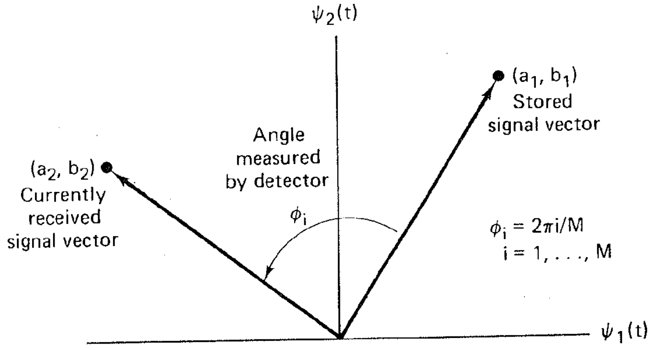

$\psi_2(t)$
(a_1, b_1) Stored signal vector
Angle measured by detector
(a_2, b_2) Currently received signal vector
ϕ_i
$\phi_i = 2\pi i/M$
$i = 1, \ldots, M$
$\psi_1(t)$

Figure 3.15 Signal space for DPSK.

for this performance loss is reduced system complexity. The error performance for the detection of DPSK is treated in Section 3.7.5.

3.6.2 Binary Differential PSK Example

The essence of differentially coherent detection in DPSK is that the identity of the data is inferred from the changes in phase from symbol to symbol. Therefore, since the data are detected by differentially examining the waveform, the transmitted waveform would first be encoded in a differential fashion. Figure 3.16a illustrates a differential encoding of a binary message data stream, $m(k)$, where k is the sample time index. The differential encoding starts (third row in the figure) with the first bit of the code bit sequence, $c(k = 0)$, chosen arbitrarily (here taken to be a one). Then the sequence of encoded bits, $c(k)$, can, in general, be encoded in one of two ways:

$$c(k) = c(k - 1) \oplus m(k) \tag{3.61}$$

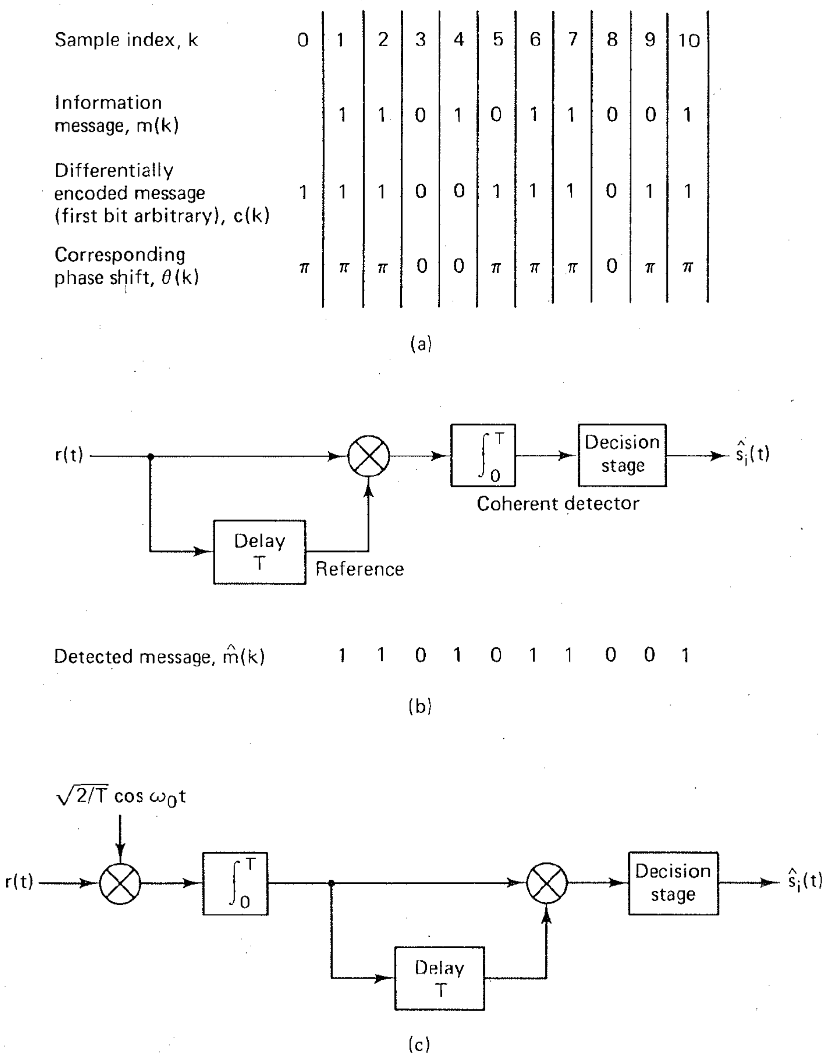
or

$$c(k) = \overline{c(k - 1) \oplus m(k)} \tag{3.62}$$

where the symbol $\oplus$ represents modulo-2 addition (defined in Section 2.12.3) and the overbar denotes complement. In Figure 3.16a the differentially encoded message was obtained by using Equation (3.62). In other words, the present code bit, $c(k)$, is a one if the message bit, $m(k)$, and the prior coded bit, $c(k - 1)$, are the same, otherwise, $c(k)$ is a zero. The fourth row translates the coded bit sequence, $c(k)$, into the phase shift sequence, $\theta(k)$, where a one is characterized by a 180° phase shift, and a zero is characterized by a 0° phase shift.

Figure 3.16b illustrates the binary DPSK detection scheme in block diagram form. Notice that the basic product integrator of Figure 3.7 is the essence of this detection process; as with coherent PSK, we are still attempting to correlate a received signal with a rcference. The interesting difference here is that the reference signal is simply a delayed version of the received signal. In other words, during each symbol time, we are matching a received symbol with the prior symbol and looking for a correlation or an anticorrelation (180° out of phase).

Consider the received signal with phase shift sequence, $\theta(k)$, entering the



Sample index, k	0	1	2	3	4	5	6	7	8	9	10
Information message, m(k)		1	1	0	1	0	1	1	0	0	1
Differentially encoded message (first bit arbitrary), c(k)	1	1	1	0	0	1	1	1	0	1	1
Corresponding phase shift, $\theta(k)$	π	π	π	0	0	π	π	π	0	π	π

(a)

Detected message, $\hat{m}(k)$ 1 1 0 1 0 1 1 0 0 1

(b)

(c)

Figure 3.16 Differential PSK (DPSK). (a) Differential encoding. (b) Differentially coherent detection. (c) Optimum differentially coherent detection.

detector of Figure 3.16b, in the absence of noise. The phase, $\theta(k = 1)$, is matched with $\theta(k = 0)$; they have the same value, π; hence the first bit of the detected output is $\hat{m}(k = 1) = 1$. Then $\theta(k = 2)$ is matched with $\theta(k = 1)$; again they have the same value, and $\hat{m}(k = 2) = 1$. Then $\theta(k = 3)$ is matched with $\theta(k = 2)$; they are different, so that $\hat{m}(k = 3) = 0$, and so on.

It must be pointed out that the detector in Figure 3.16b is suboptimum [5] in the sense of error performance. The optimum differential detector for DPSK requires a reference carrier in frequency but not necessarily in phase with the received carrier. Hence the optimum differential detector is shown in Figure 3.16c [6]. Its performance is treated in Section 3.7.5.

3.6.3 Noncoherent Detection of FSK

A detector for the noncoherent detection of FSK waveforms described by Equation (3.27) can be implemented with correlators similar to those shown in Figure 3.7. However, the hardware must be configured as an *energy detector*, without exploiting phase measurements. For this reason, the noncoherent detector typically requires twice as many channel branches as the coherent detector. Figure 3.17 illustrates the in-phase (I) and quadrature (Q) channels used to detect a binary FSK (BFSK) signal set noncoherently. Notice that the upper two branches are configured to detect the signal with frequency ω_1; the reference signals are $\sqrt{2/T}\cos\omega_1 t$ for the I branch and $\sqrt{2/T}\sin\omega_1 t$ for the Q branch. Similarly, the lower two branches are configured to detect the signal with frequency ω_2; the reference signals are $\sqrt{2/T}\cos\omega_2 t$ for the I branch and $\sqrt{2/T}\sin\omega_2 t$ for the Q branch. Imagine that the received signal $r(t)$, by chance alone, is exactly of the form $\cos\omega_1 t + n(t)$; that is, the phase is exactly zero, and thus the signal component of the received signal exactly matches the top-branch reference signal with regard to frequency and phase. In that event, the product integrator of the top branch should yield the maximum output. The second branch should yield a near-zero output (integrated zero-mean noise) since its reference signal $\sqrt{2/T}\sin\omega_1 t$

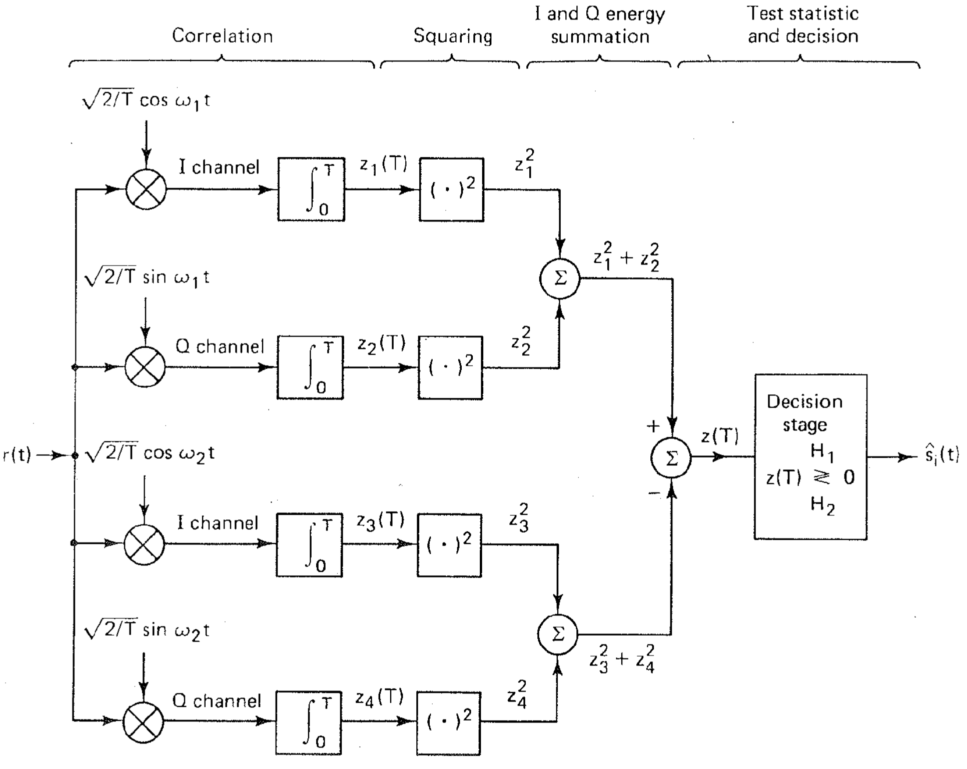


Figure 3.17 Quadrature receiver.

is orthogonal to the signal component of $r(t)$. The third and fourth branches should also yield near-zero outputs since their ω_2 reference signals are also orthogonal to the signal component of $r(t)$.

Now, imagine a different scenario; suppose that by chance alone, the received signal, $r(t)$, is of the form $\sin \omega_1 t + n(t)$. In that event, the second branch in Figure 3.17 should yield the maximum output, while the others should yield near-zero outputs. In actual practice, the most likely scenario is that $r(t)$ is of the form $\cos (\omega_1 t + \phi) + n(t)$; that is, the incoming signal will *partially* correlate with the $\cos \omega_1 t$ reference and *partially* correlate with the $\sin \omega_1 t$ reference. Now it should be obvious why a noncoherent quadrature receiver uses twice as many branches as a coherent one; the receiver knows nothing about the incoming signal's phase. The receiver essentially resolves the signal into an I component and a Q (90° out of phase) component. In Figure 3.17 the blocks following the product integrators perform a squaring operation to prevent the appearance of any negative values. Then for each of the signal types in the set (two in this binary example) the energy from the I and Q channels is added. The final stage forms the test statistic, $z(T)$, and chooses the signal with frequency ω_1 or the signal with frequency ω_2 depending on which pair of energy detectors yielded the maximum output.

Another possible implementation for noncoherent FSK detection uses bandpass filters, centered at $f_i = \omega_i/2\pi$, with bandwidth, $W_f = 1/T$, followed by *envelope detectors*, as shown in Figure 3.18. An envelope detector consists of a rectifier and a low-pass filter. The detectors are matched to the *signal envelopes* and not to the signals themselves. The phase of the carrier is of no importance in defining the envelope; hence no phase information is used. In the case of binary FSK, the decision as to whether a one or a zero was transmitted is made on the basis of which of two envelope detectors has the largest amplitude at the moment of measurement. Similarly, for a multiple frequency shift keying (MFSK) system, the decision as to which of M signals was transmitted is made on the basis of which of the M envelope detectors has the maximum output.

Even though the envelope detector block diagram of Figure 3.18 looks func-

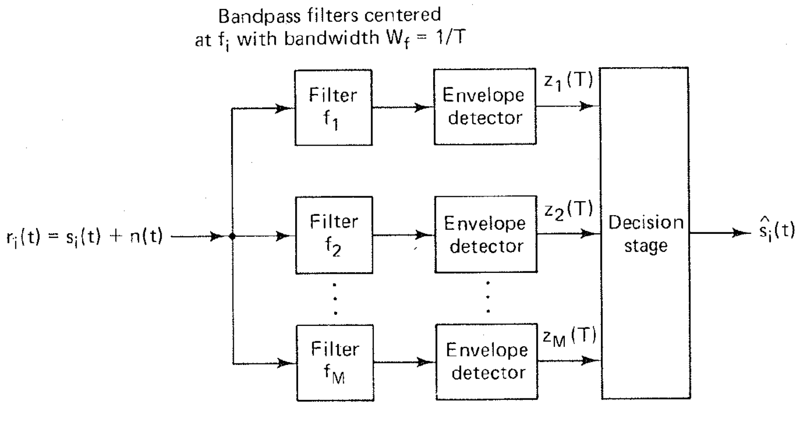


Figure 3.18 Noncoherent detection of FSK using envelope detectors.

tionally simpler than the quadrature receiver of Figure 3.17, the use of filters usually results in the envelope detector design having greater weight and cost than the quadrature receiver. Quadrature receivers can be implemented digitally; thus, with the advent of large-scale integrated (LSI) circuits, they are often the preferred choice for noncoherent detectors. The detector in Figure 3.18 can also be implemented digitally by performing discrete Fourier transformations instead of using analog filters, but such a design is usually more complex than a digital implementation of the quadrature receiver.

3.6.4 Minimum Required Tone Spacing for Noncoherent Orthogonal FSK Signaling

Frequency shift keying is usually implemented as orthogonal signaling where each tone (sinusoid) in the signal set cannot interfere with any of the other tones. In order for the signal set to be orthogonal, any pair of adjacent tones must have a frequency separation of a multiple of $1/T$ hertz. A tone with frequency f_i, that is switched on for a symbol duration of T seconds and then switched off, such as the FSK tone described in Equation (3.27), can be analytically described by

$$s_i(t) = (\cos 2\pi f_i t)\ \text{rect}\ (t/T)$$

$$\text{where rect}\ (t/T) = \begin{cases} 1 & \text{for } -T/2 \leq t \leq T/2 \\ 0 & \text{for } |t| > T/2 \end{cases}$$

The Fourier transform of $s_i(t)$, from Table A.1, is

$$\mathcal{F}\{s_i(t)\} = T\ \text{sinc}\ (f - f_i)T$$

where the sinc function is as defined in Equation (1.39). The spectra of two such adjacent tones, tone 1 with frequency f_1 and tone 2 with frequency f_2, are plotted in Figure 3.19.

In order that the two tones not interfere with each other during detection, the peak of the spectrum of tone 1 must coincide with one of the zero crossings of the spectrum of tone 2 and similarly, the peak of the tone 2 spectrum must coincide with one of the zero crossings of the tone 1 spectrum. The frequency difference between the center of the spectral main lobe and the first zero crossing represents the *minimum required spacing*. This corresponds to a minimum tone separation of $1/T$ hertz.

Example 3.3 Minimum Tone Spacing for Noncoherent Orthogonal FSK

Consider two waveforms $\cos(2\pi f_1 t + \phi)$ and $\cos 2\pi f_2 t$ to be used for *noncoherent* FSK signaling, where $f_1 > f_2$. The symbol rate is equal to $1/T$ symbols/s, where T is the symbol duration and ϕ is a constant arbitrary angle from 0 to 2π.

(a) Prove that the minimum tone spacing for *noncoherently detected* orthogonal FSK signaling is $1/T$.

(b) What is the minimum tone spacing for *coherently detected* orthogonal FSK signaling?

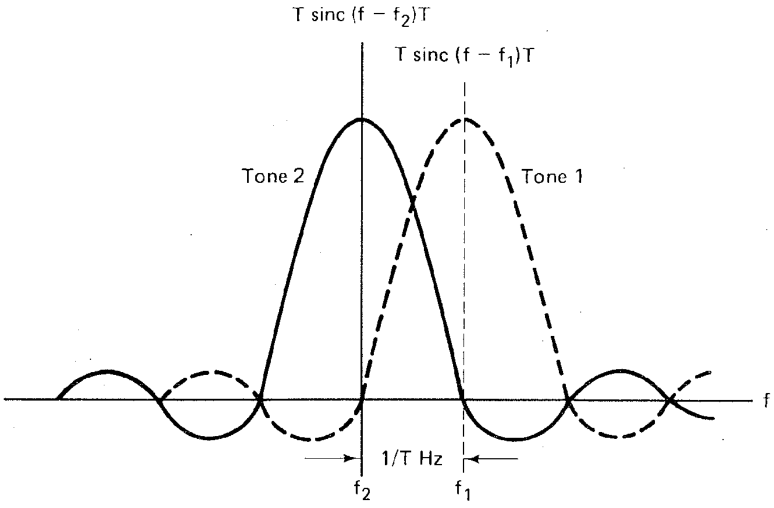


Figure 3.19 Minimum tone spacing for noncoherently detected orthogonal FSK signaling.

Solution

(a) For the two waveforms to be orthogonal, they must fulfill the orthogonality constraint of Equation (3.2):

$$\int_0^T \cos(2\pi f_1 t + \phi)\cos 2\pi f_2 t\, dt = 0 \tag{3.63}$$

Using the basic trigonometric identities shown in Equations (D.6) and (D.1) to (D.3), we can write Equation (3.63) as

$$\cos\phi \int_0^T \cos 2\pi f_1 t \cos 2\pi f_2 t\, dt - \sin\phi \int_0^T \sin 2\pi f_1 t \cos 2\pi f_2 t\, dt = 0 \tag{3.64}$$

$$\cos\phi \int_0^T [\cos 2\pi(f_1 + f_2)t + \cos 2\pi(f_1 - f_2)t]\, dt - \sin\phi \int_0^T [\sin 2\pi(f_1 + f_2)t + \sin 2\pi(f_1 - f_2)t]\, dt = 0 \tag{3.65}$$

$$\cos\phi \left[\frac{\sin 2\pi(f_1 + f_2)t}{2\pi(f_1 + f_2)} + \frac{\sin 2\pi(f_1 - f_2)t}{2\pi(f_1 - f_2)}\right]_0^T + \sin\phi \left[\frac{\cos 2\pi(f_1 + f_2)t}{2\pi(f_1 + f_2)} + \frac{\cos 2\pi(f_1 - f_2)t}{2\pi(f_1 - f_2)}\right]_0^T = 0 \tag{3.66}$$

$$\cos\phi \left[\frac{\sin 2\pi(f_1 + f_2)T}{2\pi(f_1 + f_2)} + \frac{\sin 2\pi(f_1 - f_2)T}{2\pi(f_1 - f_2)}\right] + \sin\phi \left[\frac{\cos 2\pi(f_1 + f_2)T - 1}{2\pi(f_1 + f_2)} + \frac{\cos 2\pi(f_1 - f_2)T - 1}{2\pi(f_1 - f_2)}\right] = 0 \tag{3.67}$$

We can assume that $f_1 + f_2 \gg 1$ and can thus make the following approximation:

$$\frac{\sin 2\pi(f_1 + f_2)T}{2\pi(f_1 + f_2)} \cong \frac{\cos 2\pi(f_1 + f_2)T}{2\pi(f_1 + f_2)} \cong 0 \tag{3.68}$$

Then, combining Equations (3.67) and (3.68), we can write

$$\cos \phi \sin 2\pi(f_1 - f_2)T + \sin \phi [\cos 2\pi(f_1 - f_2)T - 1] \simeq 0 \tag{3.69}$$

Note that for arbitrary ϕ, the terms in Equation (3.69) can sum to zero only when $\sin 2\pi(f_1 - f_2)T = 0$, and simultaneously $\cos 2\pi(f_1 - f_2)T = 1$.

Since

$$\sin x = 0 \qquad \text{for } x = n\pi$$

and

$$\cos x = 1 \qquad \text{for } x = 2k\pi$$

where n and k are integers, then both $\sin x = 0$ and $\cos x = 1$ occur simultaneously when $n = 2k$. From Equation (3.69), for arbitrary ϕ, we can therefore write:

$$\begin{aligned} 2\pi(f_1 - f_2)T &= 2k\pi \\ f_1 - f_2 &= \frac{k}{T} \end{aligned} \tag{3.70}$$

Thus the minimum tone spacing for *noncoherent* FSK signaling occurs for $k = 1$:

$$f_1 - f_2 = \frac{1}{T} \tag{3.71}$$

(b) To find the minimum tone spacing for *coherent* FSK, where the angle ϕ is zero, we simply rewrite Equation (3.69) with $\phi = 0$, which gives

$$\sin 2\pi(f_1 - f_2)T = 0 \tag{3.72}$$

$$f_1 - f_2 = \frac{n}{2T} \tag{3.73}$$

Thus the minimum tone spacing for *coherent* FSK signaling occurs for $n = 1$ as follows:

$$f_1 - f_2 = \frac{1}{2T} \tag{3.74}$$

Therefore, for the same symbol rate, coherently detected FSK can occupy less bandwidth than noncoherently detected FSK and still retain orthogonal signaling. We can say that coherent FSK is more *bandwidth efficient.* The subject of bandwidth efficiency is addressed in greater detail in Chapter 7.

3.7 ERROR PERFORMANCE FOR BINARY SYSTEMS

3.7.1 Probability of Bit Error for Coherently Detected BPSK

An important measure of performance used for comparing digital modulation schemes is the probability of error, P_E. For the correlator or matched filter detector, the calculations for obtaining P_E can be viewed geometrically (see Figure 3.6). They involve finding the probability that given a particular transmitted signal vector, say $\mathbf{s}_1$, the noise vector, $\mathbf{n}$, will give rise to a received signal falling outside region 1. The probability of the detector making an incorrect decision is termed the *probability of symbol error* (P_E). It is often convenient to specify system performance by the probability of bit error (P_B), even when decisions are made on the basis of symbols for which $M > 2$. The relationship between P_B and P_E is treated in Section 3.9.3 for orthogonal signaling and in Section 3.9.4 for multiple phase signaling.

For convenience, this section is restricted to the coherent detection of BPSK modulation. For this case the symbol error probability is the bit error probability. Assume that the signals are equally likely. Also assume that when signal, $s_i(t)$ ($i = 1, 2$), is transmitted, the received signal, $r(t)$, is equal to $s_i(t) + n(t)$, where $n(t)$ is an AWGN process. The antipodal signals, $s_1(t)$ and $s_2(t)$, can be characterized in a one-dimensional signal space as described in Section 3.5.1, where

$$\begin{aligned} s_1(t) &= \sqrt{E}\psi_1(t) \\ & \qquad\qquad\qquad 0 \le t \le T \\ s_2(t) &= -\sqrt{E}\psi_1(t) \end{aligned} \tag{3.75}$$

The decision stage of the detector will choose the $s_i(t)$ with the largest correlator output $z_i(T)$, or in this case of equal-energy antipodal signals, the detector, using the decision rule in Equation (3.39a), decides

$$\begin{array}{ll} s_1(t) & \text{if } z(T) > \gamma_0 = 0 \\ s_2(t) & \text{otherwise} \end{array} \tag{3.76}$$

Two types of errors can be made, as shown in Figure 3.9: The first type of error takes place if signal $s_1(t)$ is transmitted but the noise is such that the detector measures a negative value for $z(T)$ and chooses hypothesis H_2 [the hypothesis that signal $s_2(t)$ was sent]. The second type of error takes place if signal $s_2(t)$ is transmitted but the detector measures a positive value for $z(T)$ and chooses hypothesis H_1 [the hypothesis that signal $s_1(t)$ was sent].

To calculate the probability of a bit error, P_B, for this binary *minimum error* detector, we use the relationships developed in Section 2.9, starting with Equation (2.36b):

$$P_B = P(H_2|s_1)P(s_1) + P(H_1|s_2)P(s_2) \tag{3.77}$$

For the case when the a priori probabilities are equal, that is, $P(s_1) = P(s_2) = \frac{1}{2}$, we can write

$$P_B = \tfrac{1}{2}P(H_2|s_1) + \tfrac{1}{2}P(H_1|s_2) \tag{3.78}$$

Because of the symmetry of the probability density functions in Figure 3.9, we can also write

$$P_B = P(H_2|s_1) = P(H_1|s_2) \tag{3.79}$$

Thus the probability of a bit error, P_B, is numerically equal to the area under the "tail" of either pdf, $p(z|s_1)$ or $p(z|s_2)$, that falls on the "incorrect" side of the threshold. We can therefore compute P_B by integrating $p(z|s_1)$ between the limits $-\infty$ and γ_0, or as shown below, by integrating $p(z|s_2)$ between the limits γ_0 and ∞.

$$P_B = \int_{\gamma_0=(a_1+a_2)/2}^{\infty} p(z|s_2)\, dz \tag{3.80}$$

where the likelihoods, $p(z|s_i)$ $(i = 1, 2)$, are Gaussian functions with mean value, a_i, and the optimum threshold, γ_0, as shown in Section B.3.1, is equal to $(a_1 + a_2)/2$. The area-related probability of bit error, P_B, is seen to be the shaded area in Figure 3.9. It is shown in Section B.3.2 that Equation (3.80) reduces to

$$P_B = \int_{(a_1-a_2)/2\sigma_0}^{\infty} \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{u^2}{2}\right) du = Q\left(\frac{a_1 - a_2}{2\sigma_0}\right) \tag{3.81}$$

where σ_0 is the standard deviation of the noise out of the correlator. The function, $Q(x)$, called the *complementary error function* or *co-error function*, is defined as

$$Q(X) = \frac{1}{\sqrt{2\pi}} \int_x^{\infty} \exp\left(-\frac{u^2}{2}\right) du \tag{3.82}$$

and is described in greater detail in Sections 2.9 and B.3.2.

For equal-energy antipodal signaling, such as the BPSK format in Equation (3.75), the receiver output signal components are $a_1 = \sqrt{E_b}$ when $s_1(t)$ is sent and $a_2 = -\sqrt{E_b}$ when $s_2(t)$ is sent, where E_b is the signal energy per binary symbol. For AWGN we can replace the noise variance, σ_0^2, out of the correlator with $N_0/2$ (see Appendix C), so that we can rewrite Equation (3.81) as follows:

$$P_B = \int_{\sqrt{2E_b/N_0}}^{\infty} \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{u^2}{2}\right) du \tag{3.83}$$

$$= Q\left(\sqrt{\frac{2E_b}{N_0}}\right) \tag{3.84}$$

This result could also have been obtained by noting that the energy difference, E_d, between the *antipodal signal vectors,* $\mathbf{s}_1$ and $\mathbf{s}_2$, with amplitudes of $\pm\sqrt{E_b}$, as seen in Figure 3.20a, can be computed as the square of the distance between

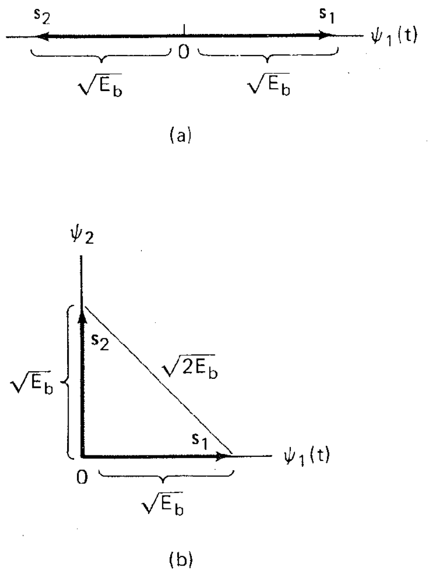


Figure 3.20 Binary signal vectors. (a) Antipodal. (b) Orthogonal.

the heads of the antipodal vectors, or in terms of the waveforms

$$E_d = \int_0^T [s_1(t) - s_2(t)]^2 \, dt \tag{3.85}$$

$$= \int_0^T s_1^2(t) \, dt + \int_0^T s_2^2(t) \, dt - 2 \int_0^T s_1(t)s_2(t) \, dt \tag{3.86}$$

Assuming equal energy signals,

$$E_b = \int_0^T s_1^2(t) \, dt = \int_0^T s_2^2(t) \, dt \tag{3.87}$$

$$E_d = 2E_b - 2E_b\rho = 2E_b(1 - \rho) \tag{3.88}$$

where

$$\rho = \frac{1}{E_b} \int_0^T s_1(t)s_2(t) \, dt \tag{3.89}$$

is the time cross-correlation coefficient and E_b is the average energy of the binary signals, $s_1(t)$ and $s_2(t)$. The correlation coefficient, ρ, is a measure of similarity between the two signals, $s_1(t)$ and $s_2(t)$, such that

$$-1 \le \rho \le 1 \tag{3.90}$$

In terms of signal vectors, the cross-correlation coefficient can be written

$$\rho = \cos \theta \tag{3.91}$$

where θ is the angle between the two signal vectors $\mathbf{s}_1$ and $\mathbf{s}_2$ (see Figure 3.6). In Equation (2.62), we developed an expression for the probability of bit error in terms of the energy difference between the two binary signals, as follows:

$$P_B = Q\left(\sqrt{\frac{E_d}{2N_0}}\right) \tag{3.92}$$

Substituting Equation (3.88) into Equation (3.92), we get

$$P_B = Q\left[\sqrt{\frac{E_b(1-\rho)}{N_0}}\right] \tag{3.93}$$

For $\rho = 1$ (or $\theta = 0$), the signals are perfectly correlated (identical). For $\rho = -1$ (or $\theta = \pi$), the signals are anticorrelated (antipodal). Since the binary PSK signals are antipodal, we can set $\rho = -1$, and Equation (3.93) is then identical to Equation (3.84).

Note that the bit error probability, P_B, for the coherent detection of bandpass antipodal signals, as seen in Equation (3.84), is the same as the P_B for the matched filter detection of baseband antipodal (bipolar) signals in Equation (2.67).

3.7.1.1 The Basic SNR Parameter for Digital Communication Systems

The parameter E_b/N_0 in Equation (3.84) can be expressed as the ratio of average signal power to average noise power, S/N (or SNR). By introducing the signal bandwidth W, we can write the following identities, showing the relationship between E_b/N_0 and SNR for binary signals.

$$\frac{E_b}{N_0} = \frac{ST}{N_0} = \frac{S}{RN_0} = \frac{SW}{RN_0W} = \frac{S}{N}\left(\frac{W}{R}\right) \tag{3.94}$$

where

S = average modulating signal power

T = bit time duration

$R = 1/T$ = bit rate

$N = N_0W$

Analysis similar to that used for developing P_B in Equations (3.84) and (3.93) is used in finding the P_B expressions for other types of modulation. Figure 3.21 illustrates the "waterfall-like" shape of most probability of error curves in the field of digital communications. The curve describes a system's error probability performance in terms of available E_b/N_0. For $E_b/N_0 \geq x_0$, $P_E \leq P_0$. The dimensionless ratio E_b/N_0 is a standard quality measure for digital communications system performance. Note that optimum digital signal detection implies a correlator (or matched filter) implementation, in which case the signal bandwidth is equal to the noise bandwidth. Often we are faced with a system model for which this is not the case; in practice, we include a factor in the required E_b/N_0 that accounts for such suboptimal detection performance. Required E_b/N_0 can be considered a metric that characterizes the performance of one system versus another;

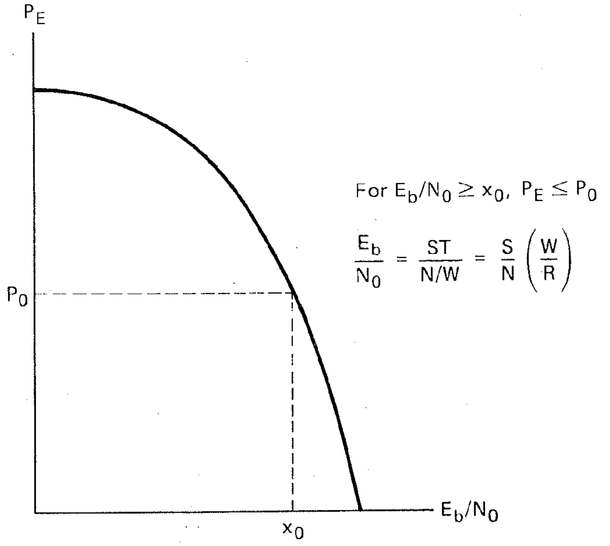


Figure 3.21 General shape of the P_E versus E_b/N_0 curve.

the *smaller* the required E_b/N_0, the *more efficient* is the system modulation and detection process for a given probability of error. Figure 3.22 is a plot comparing the bit error probability, P_B, of several binary modulation/demodulation types. The P_B for coherent detection of PSK, as shown in Equation (3.84), is plotted as the leftmost P_B curve.

Example 3.4 Bit Error Probability for BPSK Signaling

Find the bit error probability for a BPSK system with a bit rate of 1 Mbit/s. The received waveforms, $s_1(t) = A \cos \omega_0 t$ and $s_2(t) = -A \cos \omega_0 t$, are coherently detected with a matched filter. The value of A is 10 mV. Assume that the single-sided noise power spectral density is $N_0 = 10^{-11}$ W/Hz and that signal power and energy per bit are normalized relative to a 1-Ω load.

Solution

$$A = \sqrt{\frac{2E_b}{T}} = 10^{-2}\text{ V} \qquad T = \frac{1}{R} = 10^{-6}\text{ s}$$

Thus

$$E_b = \frac{A^2}{2} T = 5 \times 10^{-11}\text{ J} \qquad \text{and} \qquad \sqrt{\frac{2E_b}{N_0}} = 3.16$$

$$P_B = Q\left(\sqrt{\frac{2E_b}{N_0}}\right) = Q\,(3.16)$$

Using Table B.1 or Equation (2.43), we obtain

$$P_B = 8 \times 10^{-4}$$

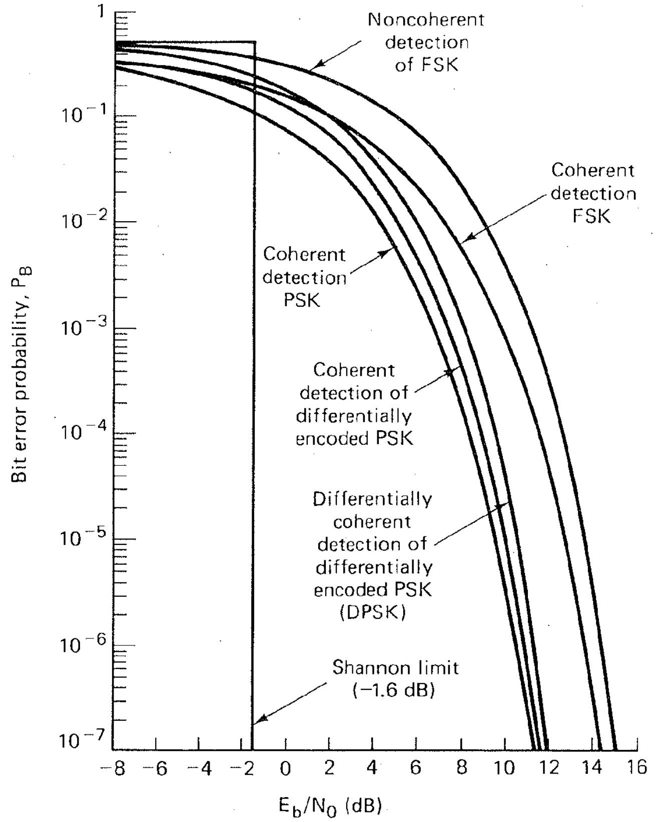


Figure 3.22 Bit error probability for several types of binary systems.

3.7.2 Probability of Bit Error for Coherently Detected Differentially Encoded PSK

Channel waveforms sometimes experience inversion; for example, when using a coherent reference generated by a phase-locked loop (see Chapter 8), one may have phase ambiguity. If the carrier phase were reversed in a DPSK modulation application, what would be the effect on the message? The only effect would be an error in the bit during which inversion occurred or the bit just after inversion, since the message information is encoded in the similarity or difference between adjacent symbols. The similarity or difference quality remains unchanged if the carrier is inverted. Sometimes, systems are *differentially encoded* and *coherently detected*, simply to avoid these phase ambiguities.

The probability of bit error for coherently detected, differentially encoded PSK is given by [7]

$$P_B = 2Q\left(\sqrt{\frac{2E_b}{N_0}}\right)\left[1 - Q\left(\sqrt{\frac{2E_b}{N_0}}\right)\right] \tag{3.95}$$

This relationship is plotted in Figure 3.22. Notice that there is a slight degradation of error performance compared to the coherent detection of PSK. This is due to the differential encoding since any single detection error results in two decision errors. Error performance for the more popular differentially coherent detection (DPSK) is covered in Section 3.7.5.

3.7.3 Probability of Bit Error for Coherently Detected FSK

Equations (3.83) and (3.84) describe the probability of bit error for coherent antipodal signals. A more general treatment for binary coherent signals (not limited to antipodal signals) yields the following equation for P_B [8]:

$$P_B = \frac{1}{\sqrt{2\pi}} \int_{\sqrt{(1-\rho)E_b/N_0}}^{\infty} \exp\left(-\frac{u^2}{2}\right) du \tag{3.96}$$

From Equation (3.91), $\rho = \cos\theta$ is the time cross-correlation coefficient between signal $s_1(t)$ and $s_2(t)$, where θ is the angle between signal vectors $\mathbf{s}_1$ and $\mathbf{s}_2$ (see Figure 3.6). For antipodal signals such as BPSK, $\theta = \pi$, thus $\rho = -1$.

For orthogonal signals such as binary FSK (BFSK), $\theta = \pi/2$, since the $\mathbf{s}_1$ and $\mathbf{s}_2$ vectors are perpendicular to each other; thus $\rho = 0$, as can be verified with Equation (3.89), and Equation (3.96) can then be written

$$P_B = \frac{1}{\sqrt{2\pi}} \int_{\sqrt{E_b/N_0}}^{\infty} \exp\left(-\frac{u^2}{2}\right) du = Q\left(\sqrt{\frac{E_b}{N_0}}\right) \tag{3.97}$$

where the *co-error function, $Q(x)$*, is defined in Equation (3.82). The result could also have been obtained by noting that the energy difference between the orthogonal signal vectors, $\mathbf{s}_1$ and $\mathbf{s}_2$, with amplitudes of $\sqrt{E_b}$, as shown in Figure 3.20b, can be computed as the square of the distance between the heads of the orthogonal vectors, to be $E_d = 2E_b$. Using this result in Equation (3.92) yields the same result as in Equation (3.97). Equation (3.97) is plotted in Figure 3.22 (coherent detection of FSK). If we compare Equation (3.97) with Equation (3.84), we can see that 3 dB (a factor of 2) more E_b/N_0 is required for BFSK to provide the same performance as BPSK. It should not be surprising that the performance of BFSK signaling is worse than BPSK signaling, since for a given signal power, orthogonal vectors are spaced closer to one another than antipodal vectors.

The bit error probability, P_B, for the coherent detection of orthogonal bandpass signals as seen in Equation (3.97) is the same as the P_B for the matched filter detection of baseband unipolar signals in Equation (2.64). As mentioned earlier, the details of on–off keying (OOK) are not treated in this book. However, it is worth noting that the P_B, described in Equation (3.97), is also identical to the error performance for the coherent detection of OOK signaling (matched filter reception).

3.7.4 Probability of Bit Error for Noncoherently Detected FSK

Consider the equally likely binary FSK signal set, $\{s_1(t)\}$, defined in Equation (3.27) as follows:

$$s_i(t) = \sqrt{\frac{2E}{T}} \cos(\omega_i t + \phi) \qquad 0 \le t \le T, \qquad i = 1, 2$$

The phase term, ϕ, is unknown and assumed constant. The detector is characterized by $M = 2$ channels of bandpass filters and envelope detectors, as shown in Figure 3.18. The input to the detector consists of the received signal, $r(t) = s_i(t) + n(t)$, where $n(t)$ is a white Gaussian noise process with two-sided power spectral density, $N_0/2$. Assume that $s_1(t)$ and $s_2(t)$ are separated in frequency sufficiently that they have negligible overlap. We start the probability of error, P_B, computation the same way that we did for coherently detected PSK, with Equation (3.78).

$$\begin{aligned} P_B &= \tfrac{1}{2}P(H_2|s_1) + \tfrac{1}{2}P(H_1|s_2) \\ &= \frac{1}{2}\int_0^T p(z|s_1)\,dz + \frac{1}{2}\int_0^T p(z|s_2)\,dz \end{aligned} \tag{3.98}$$

For the binary case, the *test statistic*, $z(T)$, is defined by $z_1(T) - z_2(T)$. Assume that the bandwidth of the filter, W_f, is $1/T$, so that the envelope of the FSK signal is (approximately) preserved at the filter output. If there was no noise at the receiver, the value of $z(T) = \sqrt{2E/T}$ when $s_1(t)$ is sent, and $z(T) = -\sqrt{2E/T}$ when $s_2(t)$ is sent. Because of this symmetry, the optimum threshold is $\gamma_0 = 0$. The pdf $p(z|s_1)$ is similar to $p(z|s_2)$; that is,

$$p(z|s_1) = p(-z|s_2) \tag{3.99}$$

Therefore, we can write

$$P_B = \int_0^T p(z|s_2)\,dz \tag{3.100}$$

or

$$P_B = P(z_1 > z_2|s_2) \tag{3.101}$$

where z_1 and z_2 denote the outputs $z_1(T)$ and $z_2(T)$ from the envelope detectors shown in Figure 3.18. For the case where the tone $s_2(t) = \cos \omega_2 t$ is sent, such that $r(t) = s_2(t) + n(t)$, the output, $z_1(T)$, is a *Gaussian noise random variable only*; it has no signal component. A Gaussian distribution into the *nonlinear envelope detector* yields a Rayleigh distribution at the output [8], so that

$$p(z_1|s_2) = \begin{cases} \dfrac{z_1}{\sigma_0^2} \exp\left(-\dfrac{z_1^2}{2\sigma_0^2}\right) & z_1 \ge 0 \\ 0 & z_1 < 0 \end{cases} \tag{3.102}$$

where σ_0^2 is the noise at the filter output. On the other hand, $z_2(T)$ has a Rician distribution, since the input to the lower envelope detector is a sinusoid plus noise [8]. The pdf, $p(z_2|s_2)$, is written as

$$p(z_2|s_2) = \begin{cases} \dfrac{z_2}{\sigma_0^2} \exp\left[-\dfrac{(z_2^2 + A^2)}{2\sigma_0^2}\right] I_0\left(\dfrac{z_2 A}{\sigma_0^2}\right) & z_2 \geq 0 \\ 0 & z_2 < 0 \end{cases} \tag{3.103}$$

where $A = \sqrt{2E/T}$, and as before, σ_0^2 is the noise at the filter output. The function $I_0(x)$, known as the modified zero-order Bessel function of the first kind [9], is defined as

$$I_0(x) = \frac{1}{2\pi} \int_0^{2\pi} \exp\,(x \cos \theta)\, d\theta \tag{3.104}$$

When $s_2(t)$ is transmitted, the receiver makes an error whenever the envelope sample $z_1(T)$ obtained from the upper channel (due to noise alone) exceeds the envelope sample $z_2(T)$ obtained from the lower channel (due to signal plus noise). Thus the probability of this error can be obtained by integrating $p(z_1|s_2)$ with respect to z_1 from z_2 to infinity, and then averaging over all possible values of z_2. That is,

$$P_B = P(z_1 > z_2|s_2)$$

$$= \int_0^\infty p(z_2|s_2) \left[\int_{z_2}^\infty p(z_1|s_2)\, dz_1\right] dz_2 \tag{3.105}$$

$$= \int_0^\infty \frac{z_2}{\sigma_0^2} \exp\left[-\frac{(z_2^2 + A^2)}{2\sigma_0^2}\right] I_0\left(\frac{z_2 A}{\sigma_0^2}\right) \left[\int_{z_2}^\infty \frac{z_1}{\sigma_0^2} \exp\left(-\frac{z_1^2}{2\sigma_0^2}\right) dz_1\right] dz_2 \tag{3.106}$$

where $A = \sqrt{2E/T}$ and where the inner integral is the conditional probability of an error for a fixed value of z_2, given that $s_2(t)$ was sent, and the outer integral averages this conditional probability over all possible values of z_2. This integral can be evaluated [10], to yield

$$P_B = \frac{1}{2} \exp\left(-\frac{A^2}{4\sigma_0^2}\right) \tag{3.107}$$

Using Equation (1.19), we can express the filter output noise, σ_0^2, as

$$\sigma_0^2 = 2\left(\frac{N_0}{2}\right) W_f \tag{3.108}$$

where $G_n(f) = N_0/2$ and W_f is the filter bandwidth. Thus Equation (3.107) becomes

$$P_B = \frac{1}{2} \exp\left(-\frac{A^2}{4 N_0 W_f}\right) \tag{3.109}$$

Equation (3.109) indicates that the error performance depends on the bandpass filter bandwidth, and that P_B becomes smaller as W_f is decreased. The result is valid only when the intersymbol interference (ISI) is negligible. The minimum W_f allowed (i.e., for no ISI) is obtained from Equation (2.77) with the filter roll-off factor $r = 0$. Thus $W_f = R$ bits/s $= 1/T$, and we can write Equation (3.109) as

$$P_B = \frac{1}{2} \exp\left(-\frac{A^2T}{4N_0}\right) \tag{3.110}$$

$$= \frac{1}{2} \exp\left(-\frac{E_b}{2N_0}\right) \tag{3.111}$$

where $E_b = (1/2)A^2T$ is the energy per bit. When comparing the error performance of noncoherent FSK with coherent FSK (see Figure 3.22), it is seen that for the same P_B, noncoherent FSK requires approximately 1 dB more E_b/N_0 than that for coherent FSK (for $P_B \leq 10^{-4}$). The noncoherent receiver is easier to implement, since coherent reference signals need not be generated. Therefore, almost all FSK receivers use noncoherent detection. It can be seen in the following section that when comparing noncoherent FSK to noncoherent DPSK, the same 3-dB difference occurs as for the comparison between coherent FSK and coherent PSK.

As mentioned earlier, the details of on–off keying (OOK) are not treated in this book. However, it is worth noting that the bit error probability, P_B, described in Equation (3.111) is identical to the P_B for the noncoherent detection of OOK signaling.

3.7.5 Probability of Bit Error for DPSK

Let us define a BPSK signal set

$$\begin{aligned} x_1(t) &= \sqrt{\frac{2E}{T}} \cos(\omega_0 t + \phi) \qquad 0 \leq t \leq T \\ x_2(t) &= \sqrt{\frac{2E}{T}} \cos(\omega_0 t + \phi \pm \pi) \qquad 0 \leq t \leq T \end{aligned} \tag{3.112}$$

A characteristic of DPSK is that there are no fixed decision regions in the signal space. Instead, the decision is based on the phase difference between successively received signals. Then for DPSK signaling we are really transmitting each bit with the binary signal pair

$$\begin{aligned} s_1(t) &= (x_1, x_1) \quad \text{or} \quad (x_2, x_2) \qquad 0 \leq t \leq 2T \\ s_2(t) &= (x_1, x_2) \quad \text{or} \quad (x_2, x_1) \qquad 0 \leq t \leq 2T \end{aligned} \tag{3.113}$$

where (x_i, x_j) $(i, j = 1, 2)$ denotes $x_i(t)$ followed by $x_j(t)$ defined in Equation (3.112). The first T seconds of each waveform are actually the last T seconds of the previous waveform. Note that $s_1(t)$ and $s_2(t)$ can each have either of two possible forms and that $x_1(t)$ and $x_2(t)$ are antipodal signals. Thus the correlation between $s_1(t)$ and $s_2(t)$ for *any combination* of forms can be written as

$$\begin{aligned} z(2T) &= \int_0^{2T} s_1(t)s_2(t)\,dt \\ &= \int_0^T [x_1(t)]^2\,dt - \int_0^T [x_1(t)]^2\,dt = 0 \end{aligned} \tag{3.114}$$

Therefore, pairs of DPSK signals can be represented as orthogonal signals $2T$ seconds long. Detection could correspond to noncoherent envelope detection with four channels matched to each of the possible envelope outputs, as shown in Figure 3.23a. Since the two envelope detectors representing each symbol are negatives of each other, the envelope sample of each will be the same. Hence we can implement the detector as a single channel for $s_1(t)$ matched to either (x_1, x_1) or (x_2, x_2), and a single channel for $s_2(t)$ matched to either (x_1, x_2) or (x_2, x_1), as shown in Figure 3.23b. The DPSK detector is therefore reduced to a standard two-channel noncoherent detector. In reality, the filter can be matched to the difference signal so that only one channel is necessary. For orthogonal signals, this operates with the bit error probability in Equation (3.111). Since the DPSK signals have a bit interval of $2T$, the $s_i(t)$ signals defined in Equation (3.113) have

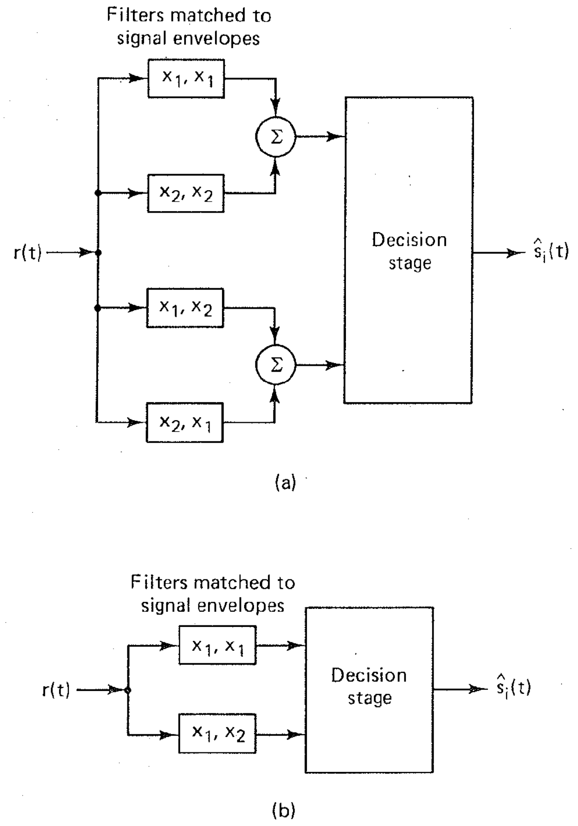


Figure 3.23 DPSK detection. (a) Four-channel differentially coherent detection of binary DPSK. (b) Equivalent two-channel detector for binary DPSK.

twice the energy of a signal defined over a single-symbol duration. Thus we may write P_B as

$$P_B = \frac{1}{2} \exp\left(-\frac{E_b}{N_0}\right) \tag{3.115}$$

Equation (3.115) is seen plotted in Figure 3.22, designated as differentially coherent detection of differentially encoded PSK, or simply DPSK. This expression is valid for the optimum DPSK detector shown in Figure 3.16c. For the detector shown in Figure 3.16b, the error probability will be slightly inferior to that given in Equation (3.115) [5]. When comparing the error performance of Equation (3.115) with that of coherent PSK (see Figure 3.22), it is seen that for the same P_B, DPSK requires approximately 1 dB more E_b/N_0 than does BPSK (for $P_B \le 10^{-4}$). It is easier to implement a DPSK system than a PSK system, since the DPSK receiver does not need phase synchronization. For this reason, DPSK, although less efficient than PSK, is sometimes the preferred choice between the two.

3.7.6 Comparison of Bit Error Performance for Various Modulation Types

The P_B expressions for the best known of the binary modulation schemes discussed above are listed in Table 3.1 and are illustrated in Figure 3.22. For $P_B = 10^{-4}$, it can be seen that there is approximately a 4-dB difference between the best (coherent PSK) and the worst (noncoherent FSK) that were discussed here. In some cases, 4 dB is a small price to pay for the implementation simplicity gained in going from coherent PSK to noncoherent FSK; however, for other cases, even a 1-dB saving is worthwhile. There are other considerations besides P_B and system complexity; for example, in some cases (such as a randomly fading channel), a noncoherent system is more desirable because there may be difficulty in establishing and maintaining a coherent reference. Signals that can withstand significant degradation before their ability to be detected is affected are clearly desirable in military and space applications.

TABLE 3.1 Probability of Error for Selected Binary Modulation Schemes

Modulation	P_B
Coherent PSK	$Q\left(\sqrt{\frac{2E_b}{N_0}}\right)$
Noncoherent DPSK	$\frac{1}{2}\exp\left(-\frac{E_b}{N_0}\right)$
Coherent FSK	$Q\left(\sqrt{\frac{E_b}{N_0}}\right)$
Noncoherent FSK	$\frac{1}{2}\exp\left(-\frac{1}{2}\frac{E_b}{N_0}\right)$

3.8 *M*-ARY SIGNALING AND PERFORMANCE

3.8.1 Ideal Probability of Bit Error Performance

The typical probability of error versus E_b/N_0 curve was shown to have a waterfall-like shape in Figure 3.21. The probability of bit error (P_B) characteristics of various binary modulation schemes in AWGN also display this shape, as shown in Figure 3.22. What should an *ideal* P_B versus E_b/N_0 curve look like? Figure 3.24 displays the ideal characteristic as the *Shannon limit.* The limit represents the threshold E_b/N_0 below which reliable communication cannot be maintained. Shannon's work is described in greater detail in Chapter 7.

We can describe the ideal curve in Figure 3.24 as follows. For all values of E_b/N_0 above the Shannon limit, P_B is zero. Once E_b/N_0 is reduced below the Shannon limit, P_B degrades to the worst-case value of $\frac{1}{2}$. (Note that $P_B = 1$ is not the worst case for binary signaling, since that value is just as good as $P_B = 0$; if the probability of making a bit error is 100%, the bit stream could simply be inverted to retrieve the correct data.) It should be clear, by comparing the typical P_B curve with the ideal one in Figure 3.24 that the large arrow in the figure describes the desired direction of movement to achieve improved P_B performance.

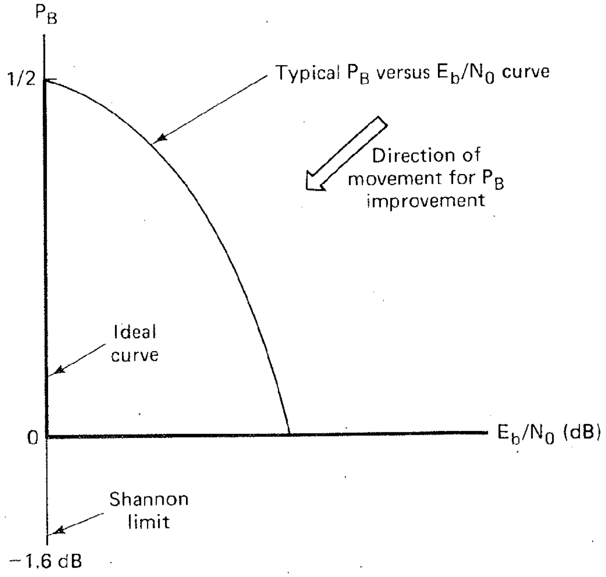


Figure 3.24 Ideal P_B versus E_b/N_0 curve.

3.8.2 *M*-ary Signaling

Let us review *M*-ary signaling. The processor considers k bits at a time. It instructs the modulator to produce one of $M = 2^k$ waveforms; binary signaling is the special case where $k = 1$. Does *M*-ary signaling improve or degrade performance? Be careful with your answer—the question is a loaded one. Figure 3.25 illustrates the probability of bit error, $P_B(M)$, versus E_b/N_0 for coherently detected *orthog-*

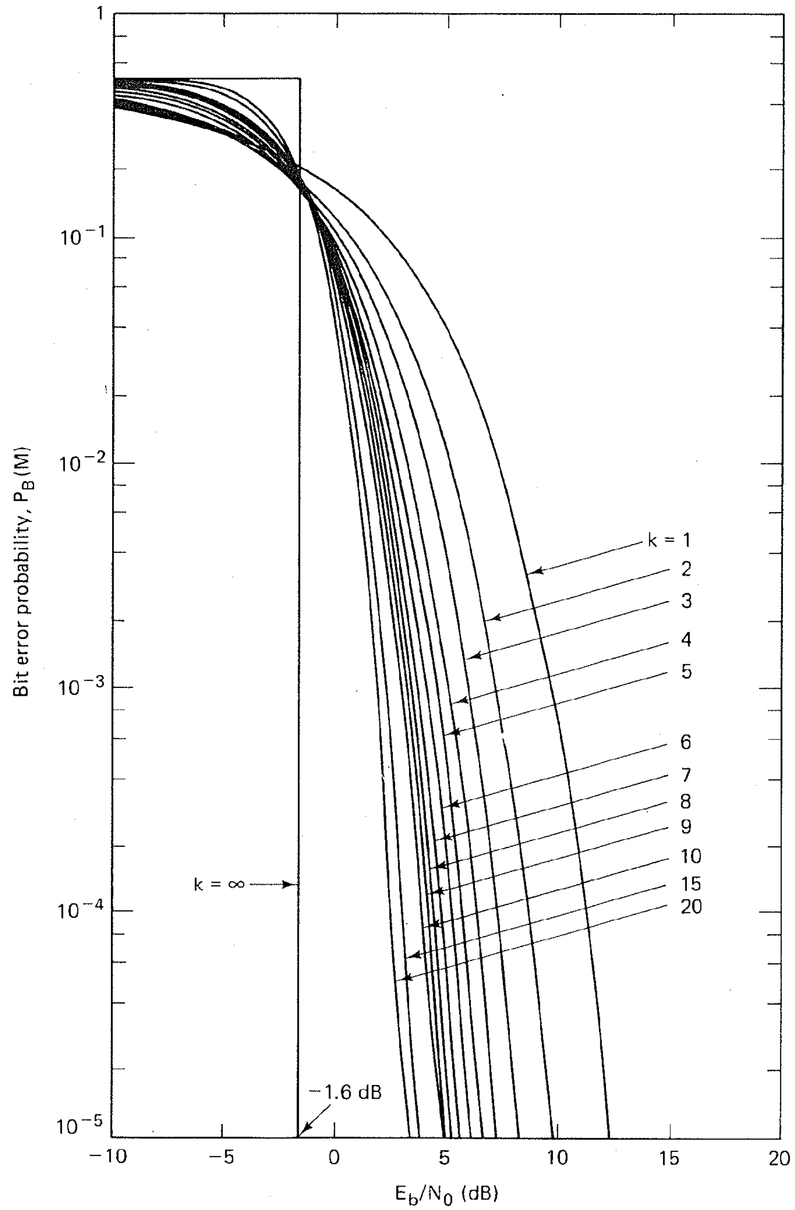


Figure 3.25 Bit error probability for coherently detected M-ary orthogonal signaling. (Reprinted from W. C. Lindsey and M. K. Simon, *Telecommunication Systems Engineering*, Prentice-Hall, Inc., Englewood Cliffs, N.J., 1973, courtesy of W. C. Lindsey and Marvin K. Simon.)

onal M-ary signaling over a Gaussian channel. Figure 3.26 similarly illustrates $P_B(M)$ versus E_b/N_0 for coherently detected *multiple phase* M-ary signaling over a Gaussian channel. In which direction do the curves move as the value of k (or M) increases? From Figure 3.24 we know the directions of curve movement for

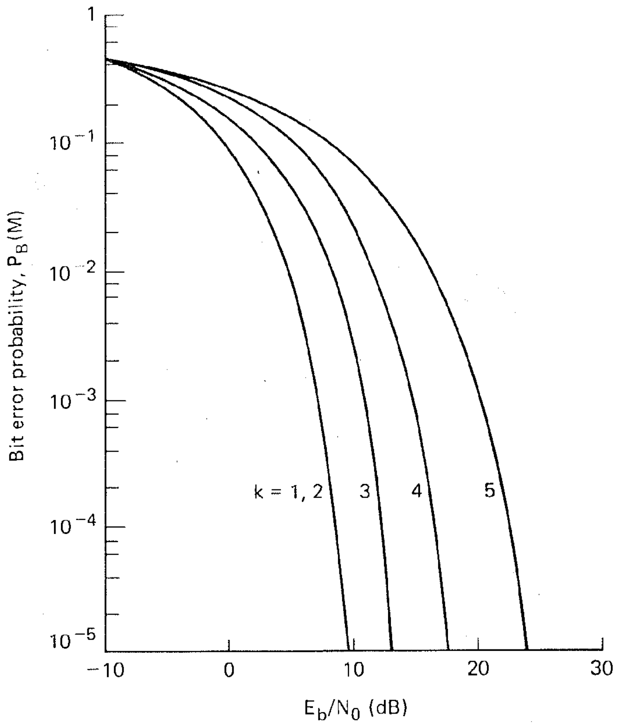


Figure 3.26 Bit error probability for coherently detected multiple phase signaling.

improved and degraded error performance. In Figure 3.25, as k increases, the curves move in the direction of improved error performance. In Figure 3.26, as k increases, the curves move in the direction of degraded error performance. Such movement tells us that M-ary signaling produces improved error performance with orthogonal signaling and degraded error performance with multiple phase signaling. Can that be true? Why would anyone ever use multiple phase PSK signaling if it provides degraded error performance compared to binary PSK signaling? It *is* true, and many systems do use multiple phase signaling. The question, as stated, is loaded because it implies that error probability versus E_b/N_0 is the *only* performance criterion; there are many others (e.g., bandwidth, power, throughput, complexity), but in Figures 3.25 and 3.26, error performance is the characteristic that stands out explicitly.

A performance characteristic that is not explicitly seen in Figures 3.25 and 3.26 is the required system bandwidth. For the curves characterizing M-ary orthogonal signals in Figure 3.25, as k increases, the required bandwidth also increases. For the M-ary multiple phase curves in Figure 3.26, as k increases, a larger bit rate can be transmitted within the same bandwidth. In other words, for a fixed data rate, the required bandwidth is decreased. Therefore, *both* the orthogonal and multiple phase error performance curves tell us that M-ary signaling represents a vehicle for performing a system trade-off. In the case of orthogonal signaling, error performance improvement can be achieved at the expense of bandwidth. In the case of multiple phase signaling, bandwidth performance can be achieved at the expense of error performance. Error performance versus bandwidth performance, a fundamental communications trade-off, is treated in greater detail in Chapter 7.

3.8.3 Vectorial View of MPSK Signaling

Figure 3.27 illustrates MPSK signal sets for $M = 2, 4, 8$, and 16. In Figure 3.27a we see the binary ($k = 1$, $M = 2$) antipodal vectors $\mathbf{s}_1$ and $\mathbf{s}_2$ positioned 180° apart. Thc decision boundary is drawn so as to partition the signal space into two regions. On the figure is also shown a noise vector **n** equal in magnitudc to $\mathbf{s}_1$. The figure establishes the magnitude and orientation of the minimum energy noise vector that would cause the detector to make a symbol error.

In Figure 3.27b we see the 4-ary ($k = 2$, $M = 4$) vectors positioned 90° apart. Thc dccision boundaries (only one line is drawn) divide the signal space into four regions. Again a noise vector **n** is drawn (from the head of a signal vector, normal to the closest decision boundary) to illustrate the minimum energy noise vector that would cause the detector to make a symbol error. Notice that the minimum energy noise vector of Figure 3.27b is smaller than that of Figure 3.27a, illustrating that the 4-ary system is more vulnerable to noise than the 2-ary systcm (signal energy being equal for each case). As we move on to Figure 3.27c for the 8-ary case and Figure 3.27d for the 16-ary case, it should be clear that for multiple phase signaling, as M increases, we are crowding more signal vectors into the signal plane. As the vectors are moved closer together, a smaller amount of noise energy is required to cause an error.

Figure 3.27 adds some insight as to why the curves of Figure 3.26 behave as they do as k is increased. Figure 3.27 also provides some insight into a basic trade-off in multiple phase signaling. Crowding more signal vectors into the signal space is tantamount to increasing the data rate without increasing the system bandwidth (the vectors are all confined to the same plane). In other words, we have increased the bandwidth utilization at the expense of error performance.

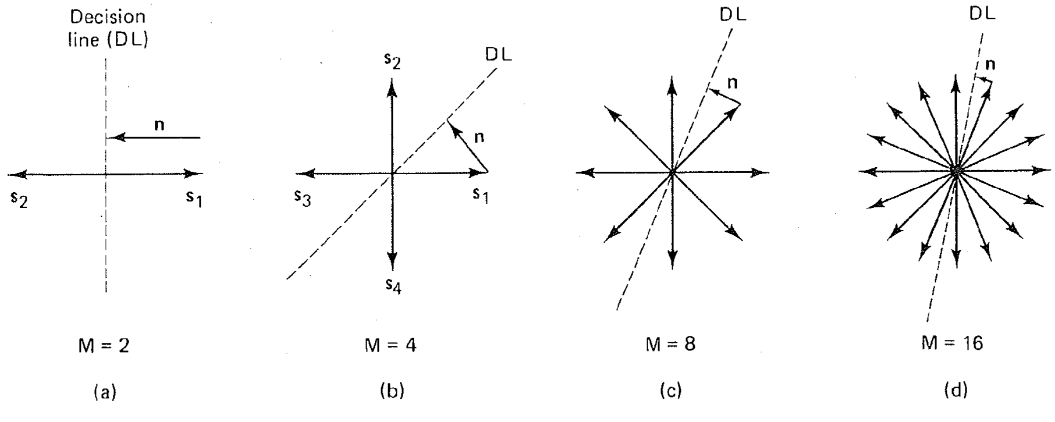


Figure 3.27 MPSK signal sets for $M = 2, 4, 8, 16$.

Look at Figure 3.27d, where the error performance is worse than any of the other examples in Figure 3.27. How might we "buy back" the degraded error performance; that is, what can we trade-off so that the distance between neighboring signal vectors in Figure 3.27d is increased to that in Figure 3.27a? We can increase the signal strength (make the signal vectors larger) until the minimum distance from the head of a signal vector to a decision line equals the length of the noise vector in Figure 3.27a. Therefore, in a multiple phase system, as M is increased, we can either achieve improved bandwidth performance at the expense of error performance, or if we increase the E_b/N_0 so that the error probability is not degraded, we can achieve improved bandwidth performance at the expense of increasing E_b/N_0.

3.8.4 BPSK and QPSK Have the Same Bit Error Probability

In Equation (3.94) we stated the general relationship between E_b/N_0 and S/N_0 for binary transmission, as follows:

$$\frac{E_b}{N_0} = \frac{S}{N_0}\left(\frac{1}{R}\right) \tag{3.116}$$

where S is the average signal power and R is the bit rate. A BPSK signal with the available E_b/N_0 found from Equation (3.116) will perform with a P_B that can be read from the $k = 1$ curve in Figure 3.26. QPSK can be characterized as two orthogonal BPSK channels. The QPSK bit stream is usually partitioned into an even and odd (I and Q) stream; each new stream modulates an orthogonal component of the carrier at half the bit rate of the original stream. The I stream modulates the cos $\omega_0 t$ term and the Q stream modulates the sin $\omega_0 t$ term. If the magnitude of the original QPSK vector has the value A, the magnitude of the I and Q component vectors each has a value of $A/\sqrt{2}$, as shown in Figure 3.28. Thus, each of the quadrature BPSK signals has half of the average power of the original QPSK signal. Hence if the original QPSK waveform has a bit rate of R bits/s and an average power of S watts, the quadrature partitioning results in each

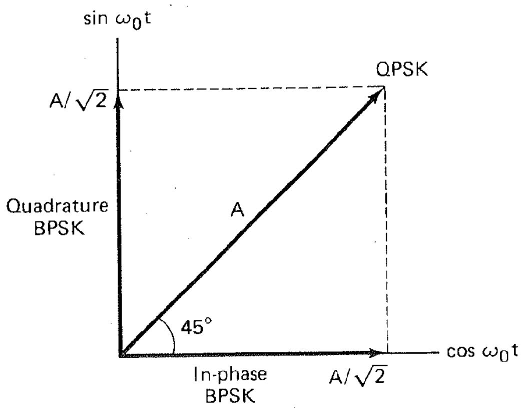


Figure 3.28 In-phase and quadrature BPSK components of QPSK signaling.

of the BPSK waveforms having a bit rate of $R/2$ bits/s and an average power of $S/2$ watts.

Therefore, the E_b/N_0 characterizing each of the orthogonal BPSK channels, comprising the QPSK signal, is equivalent to the E_b/N_0 in Equation (3.116) since it can be written as

$$\frac{E_b}{N_0} = \frac{S/2}{N_0}\left(\frac{W}{R/2}\right) = \frac{S}{N_0}\left(\frac{1}{R}\right) \tag{3.117}$$

Thus each of the orthogonal BPSK channels, and hence the composite QPSK signal, is characterized by the same E_b/N_0 and hence the same P_B performance as a BPSK signal. The natural orthogonality of the 90° phase shifts between adjacent QPSK symbols results in the *bit error probabilities* being equal for both BPSK and QPSK signaling. It is important to note that the *symbol error probabilities* are *not* equal for BPSK and QPSK signaling. The relationship between bit error probability and symbol error probability is treated in Sections 3.9.3 and 3.9.4.

3.8.5 Vectorial View of MFSK Signaling

In Section 3.8.3, Figure 3.27 provides some insight as to why the error performance of MPSK signaling degrades as k (or M) increases. It would be useful to have a similar vectorial illustration for the error performance of MFSK signaling as seen in the curves of Figure 3.25. Since the MFSK signal space is characterized by M mutually perpendicular axes, we can only conveniently illustrate the cases, $M = 2$ and $M = 3$. In Figure 3.29a we see the binary orthogonal vectors $\mathbf{s}_1$ and $\mathbf{s}_2$ positioned 90° apart. The decision boundary is drawn so as to partition the signal space into two regions. On the figure is also shown a noise vector $\mathbf{n}$, which represents the minimum noise vector that would cause the detector to make an error.

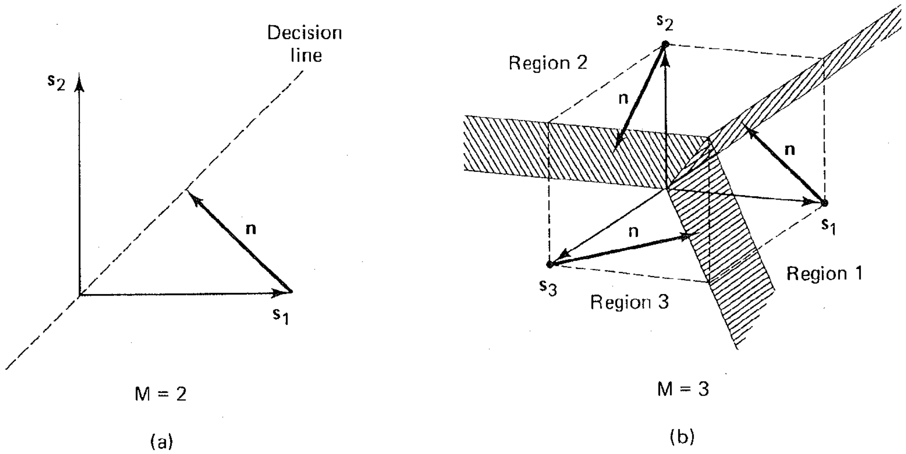


Figure 3.29 MFSK signal sets for M = 2, 3.

In Figure 3.29b we see a 3-ary signal space with axes positioned 90° apart. Here decision planes partition the signal space into three regions. Noise vectors **n** are shown added to each of the prototype signal vectors $\mathbf{s}_1$, $\mathbf{s}_2$, and $\mathbf{s}_3$; each noise vector illustrates an example of the minimum noise energy that would cause the detector to make a symbol error. The minimum noise energy vectors in Figure 3.29b are the same length as the noise vector in Figure 3.29a. In Section 3.5.4 we stated that the distance between any two prototype signal vectors $\mathbf{s}_i$ and $\mathbf{s}_j$ in an M-ary orthogonal space is constant. It follows that the minimum distance between a prototype signal vector and any of the decision boundaries remains fixed as M increases. Unlike the case of MPSK signaling, where adding new signals to the signal set makes the signals vulnerable to smaller noise vectors, here in the case of MFSK signaling, adding new signals to the signal set does *not* make the signals vulnerable to smaller noise vectors.

It would be convenient to illustrate the point by drawing higher-dimensional orthogonal spaces, but of course this is not possible. We can only use our "mind's eye" to understand that increasing the signal set, M, by adding additional axes, where each new axis is mutually perpendicular to all the others, does not crowd the signal set more closely together; thus a transmitted signal from an orthogonal set is *not* more vulnerable to a noise vector when the set is increased in size. In fact, we see from Figure 3.25 that as k increases, the bit error performance improves.

Understanding the error performance improvement of orthogonal signaling, as illustrated in Figure 3.25, is facilitated by comparing the probability of symbol error (P_E) versus unnormalized SNR, with P_E versus E_b/N_0. Figure 3.30 represents a set of P_E performance curves plotted against unnormalized SNR for coherent FSK signaling. Here we see that P_E degrades as M is increased. Didn't we say that an orthogonal signal is *not* made more vulnerable to a given noise vector, as the orthogonal set is increased in size? It is correct that for orthogonal signaling, with a given SNR it takes a fixed-size noise vector to perturb a transmitted signal into an error region; the signals do not become vulnerable to smaller noise vectors as M increases. However, as M increases, more neighboring decision regions are introduced; thus the number of ways in which a symbol error can be made increases. Figure 3.30 reflects the degradation in P_E versus unnormalized SNR as M is increased; there are $(M - 1)$ ways to make an error. Examining performance under the condition of a fixed SNR (as M increases) is not very useful for digital communications. A fixed SNR means a fixed amount of energy per symbol; thus as M increases, there is a fixed amount of energy to be apportioned over a larger number of bits, or there is less energy per bit. The most useful way of comparing one digital system with another is on the basis of *bit-normalized SNR* or E_b/N_0. The error performance improvement with increasing M, seen in Figure 3.25, manifests itself only when error probability is plotted against E_b/N_0. For this case, as M increases, the required E_b/N_0 (to meet a given error probability) is reduced for a fixed SNR; therefore, we need to map the Figure 3.30 plot into a new plot, similar to Figure 3.25, where the abscissa represents E_b/N_0 instead of SNR. Figure 3.31 illustrates such a mapping; it demonstrates that curves manifesting degraded P_E with increasing M (such as Figure 3.30) are

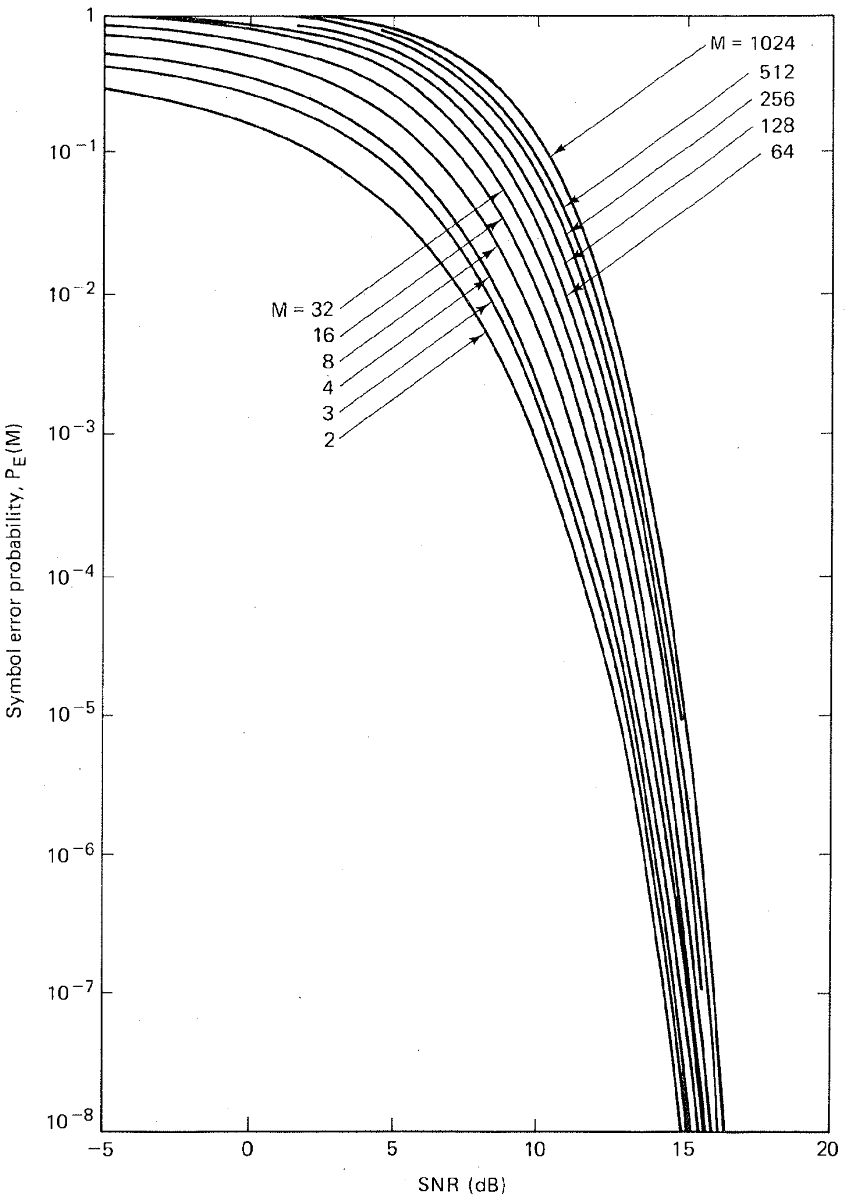


Figure 3.30 Symbol error probability versus SNR for coherent FSK signaling. (From Bureau of Standards, *Technical Note 167*, March 1963.) (Reprinted from *Central Radio Propagation Laboratory Technical Note 167*, March 25, 1963, Fig. 1, p. 5, courtesy of National Bureau of Standards.)

transformed into curves manifesting improved P_E with increasing M. The basic mapping relationship is expressed in Equation (3.94):

$$\frac{E_b}{N_0} = \frac{S}{N}\left(\frac{W}{R}\right)$$

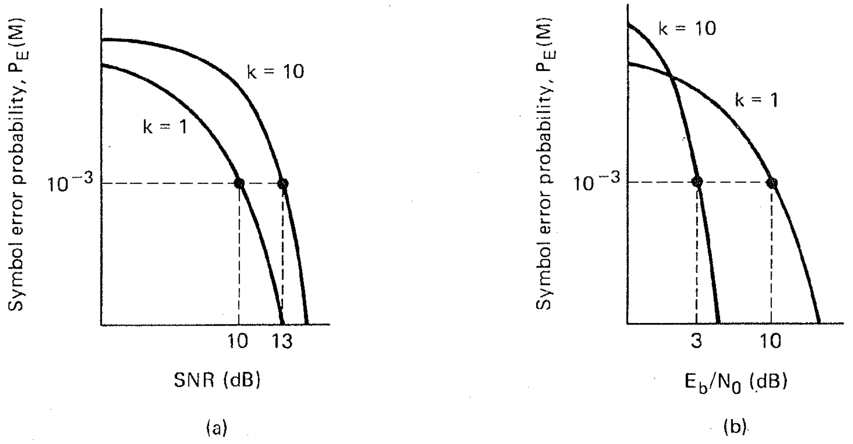


Figure 3.31 Mapping P_E versus SNR into P_E versus E_b/N_0 for orthogonal signaling. (a) Unnormalized. (b) Normalized.

where W is the detection bandwidth. Since

$$R = \frac{\log_2 M}{T} = \frac{k}{T}$$

where T is the symbol duration, we can then write

$$\frac{E_b}{N_0} = \frac{S}{N}\left(\frac{WT}{\log_2 M}\right) = \frac{S}{N}\left(\frac{WT}{k}\right) \tag{3.118}$$

For FSK signaling the detection bandwidth, W in hertz, is typically equal in value to the symbol rate $1/T$, in other words, $WT \simeq 1$. Therefore,

$$\frac{E_b}{N_0} \simeq \frac{S}{N}\left(\frac{1}{k}\right) \tag{3.119}$$

Figure 3.31 illustrates the mapping from P_E versus SNR to P_E versus E_b/N_0 for coherently detected M-ary orthogonal signaling. In Figure 3.31a, on the $k = 1$ curve is shown an operating point corresponding to $P_E = 10^{-3}$ and SNR $= 10$ dB. On the $k = 10$ curve is shown an operating point at the same $P_E = 10^{-3}$ but with SNR $= 13$ dB (approximate values taken from Figure 3.30). Here we clearly see the degradation in error performance as k increases. Consider the same $k = 1$ and $k = 10$ cases mapped onto the Figure 3.31b plane, where the abscissa is E_b/N_0. The $k = 1$ case looks exactly the same as it does in Figure 3.31a. But for the $k = 10$ case, the required E_b/N_0 is obtained from Equation (3.119) as follows: $E_b/N_0 = 20(\frac{1}{10}) = 2$ (3 dB), thus showing the error performance improvement as k is increased. In digital communication systems, error performance is almost always considered in terms of E_b/N_0, since such a measurement makes for a meaningful comparison between one system's performance and another. Therefore, the curves of Figures 3.30 and 3.31a are hardly ever seen.

3.9 SYMBOL ERROR PERFORMANCE FOR *M*-ARY SYSTEMS ($M > 2$)

3.9.1 Probability of Symbol Error for MPSK

For large energy-to-noise ratios, the symbol error performance, $P_E(M)$, for equally likely coherently detected M-ary PSK signaling can be expressed [9] as follows:

$$P_E(M) \simeq 2Q\left(\sqrt{\frac{2E_s}{N_0}}\sin\frac{\pi}{M}\right) \tag{3.120}$$

where $P_E(M)$ is the probability of symbol error, $E_s = E_b(\log_2 M)$ is the energy per symbol, and $M = 2^k$ is the size of the symbol set. The $P_E(M)$ performance curves for coherently detected MPSK signaling are plotted versus E_b/N_0 in Figure 3.32.

The symbol error performance for differentially coherent detection of M-

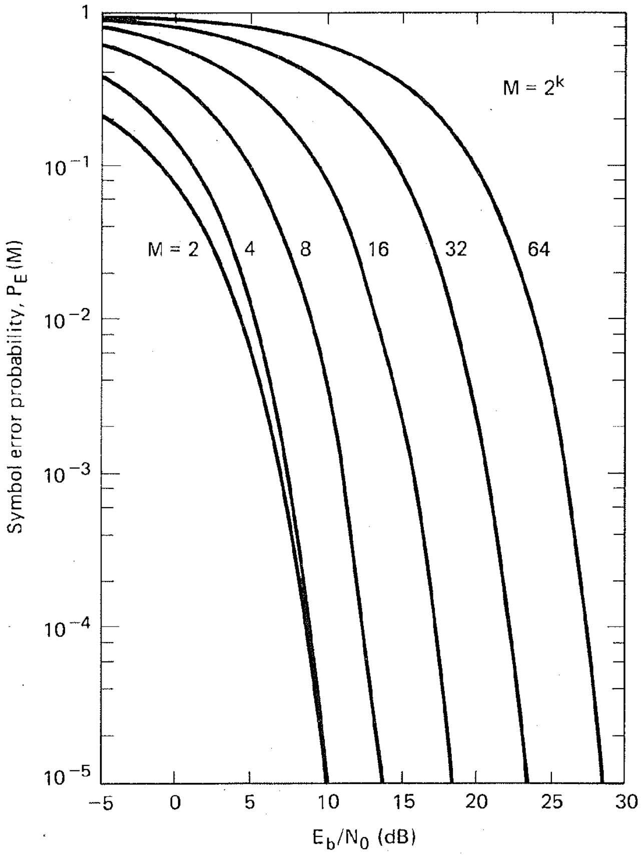


Figure 3.32 Symbol error probability for coherently detected multiple phase signaling. (Reprinted from W. C. Lindsey and M. K. Simon, *Telecommunication Systems Engineering*, Prentice-Hall, Inc., Englewood Cliffs, N.J., 1973, courtesy of W. C. Lindsey and Marvin K. Simon.)

ary DPSK (for large E_s/N_0) is similarly expressed [9] as

$$P_E(M) \simeq 2Q\left(\sqrt{\frac{2E_s}{N_0}}\sin\frac{\pi}{\sqrt{2}M}\right) \tag{3.121}$$

3.9.2 Probability of Symbol Error for MFSK

The symbol error performance $P_E(M)$, for equally likely *coherently* detected M-ary orthogonal signaling can be upper bounded [7] as follows:

$$P_E(M) \leq (M - 1)Q\left(\sqrt{\frac{E_s}{N_0}}\right) \tag{3.122}$$

where $E_s = E_b(\log_2 M)$ is the energy per symbol and M is the size of the symbol set. The $P_E(M)$ performance curves for coherently detected M-ary orthogonal signaling are plotted versus E_b/N_0 in Figure 3.33.

The symbol error performance for equally likely *noncoherently* detected M-ary orthogonal signaling is [11]

$$P_E(M) = \frac{1}{M}\exp\left(-\frac{E_s}{N_0}\right)\sum_{j=2}^{M}(-1)^j\binom{M}{j}\exp\left(\frac{E_s}{jN_0}\right) \tag{3.123}$$

where

$$\binom{M}{j} = \frac{M!}{j!\,(M-j)!} \tag{3.124}$$

is the standard binomial coefficient yielding the number of ways in which j symbols out of M may be in error. Note that for the binary case, Equation (3.123) reduces to

$$P_B = \frac{1}{2}\exp\left(-\frac{E_b}{2N_0}\right) \tag{3.125}$$

which is the same result as that described by Equation (3.111). The $P_E(M)$ performance curves for noncoherently detected M-ary orthogonal signaling are plotted versus E_b/N_0 in Figure 3.34. If we compare this noncoherent orthogonal $P_E(M)$ performance with the corresponding $P_E(M)$ results for the coherent detection of orthogonal signals in Figure 3.33, it can be seen that for $k > 7$, there is a negligible difference. An upper bound for coherent as well as noncoherent reception of orthogonal signals is [11]

$$P_E(M) < \frac{M-1}{2}\exp\left(-\frac{E_s}{2N_0}\right) \tag{3.126}$$

where E_s is the energy per symbol and M is the size of the symbol set.

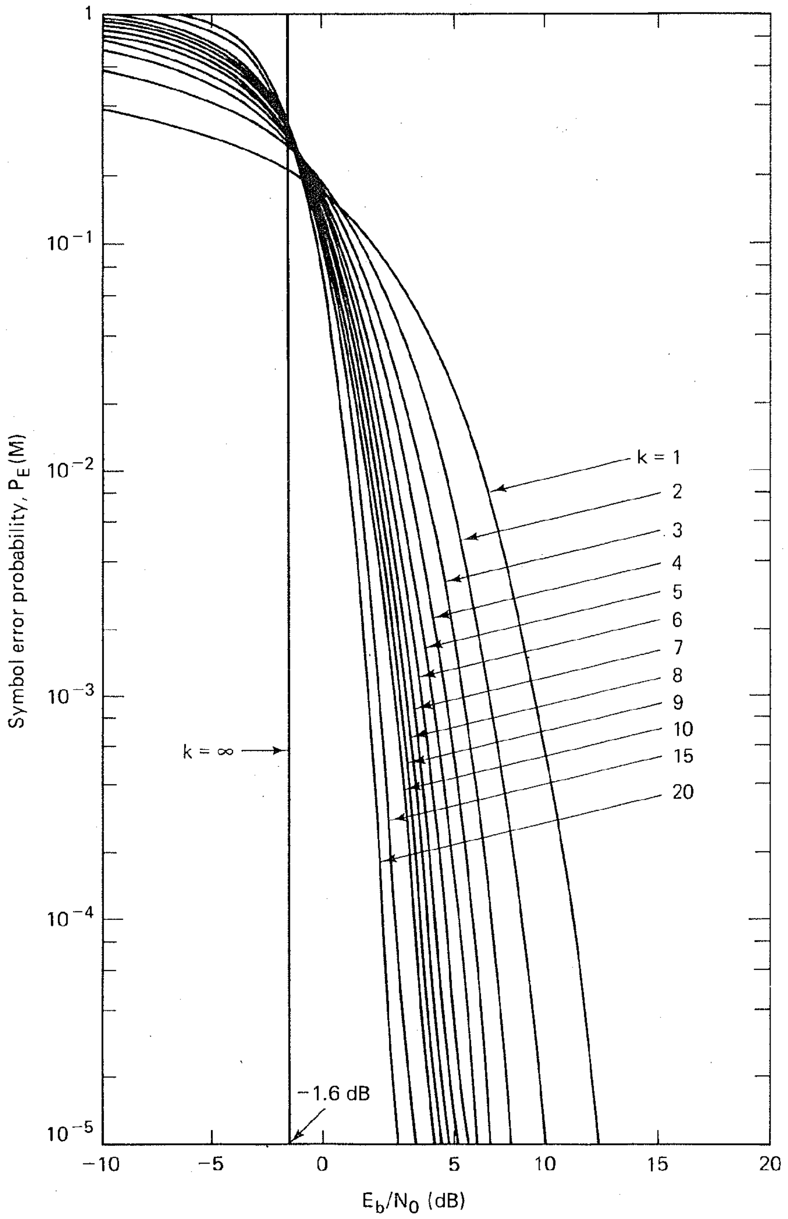


Figure 3.33 Symbol error probability for coherently detected *M*-ary orthogonal signaling. (Reprinted from W. C. Lindsey and M. K. Simon, *Telecommunication Systems Engineering*, Prentice-Hall, Inc., Englewood Cliffs, N.J., 1973, courtesy of W. C. Lindsey and Marvin K. Simon.)

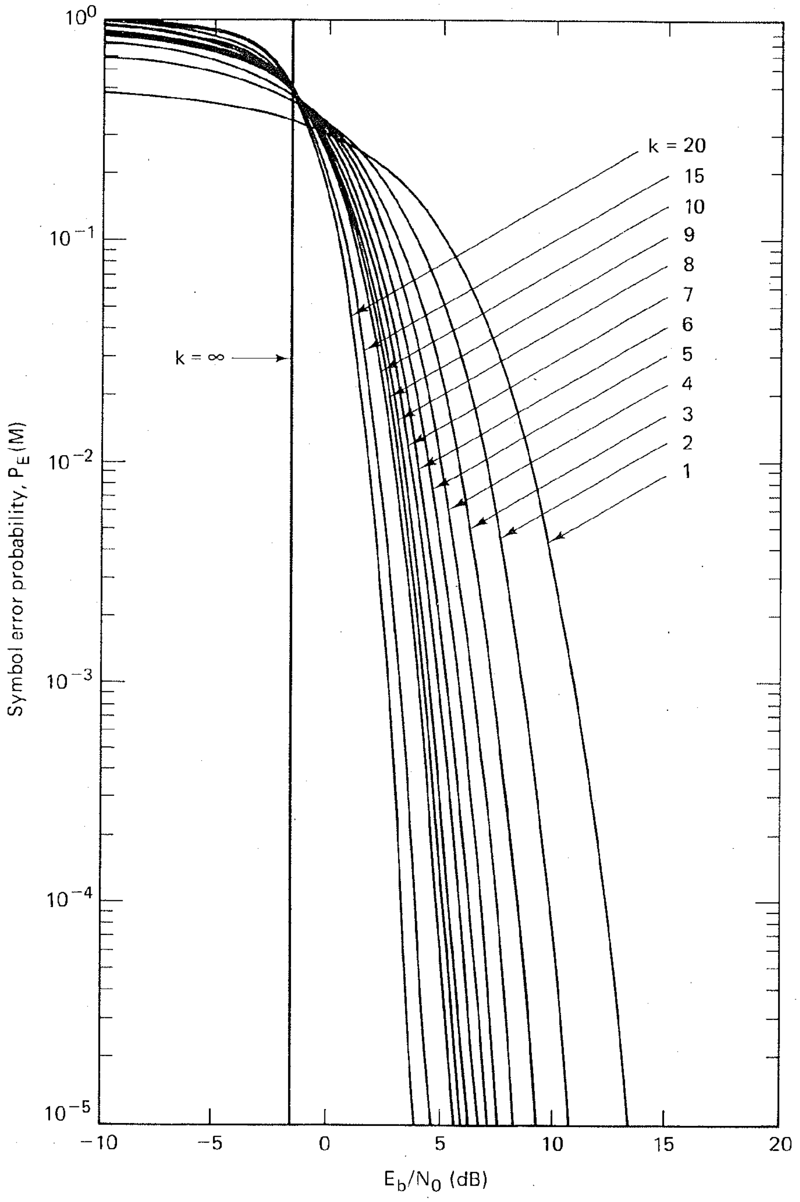


Figure 3.34 Symbol error probability for noncoherently detected M-ary orthogonal signaling. (Reprinted from W. C. Lindsey and M. K. Simon, *Telecommunication Systems Engineering*, Prentice-Hall, Inc., Englewood Cliffs, N.J., 1973, courtesy of W. C. Lindsey and Marvin K. Simon.)

3.9.3 Bit Error Probability versus Symbol Error Probability for Orthogonal Signals

It can be shown [11] that the relationship between probability of bit error (P_B) and probability of symbol error (P_E) for an M-ary orthogonal signal set is

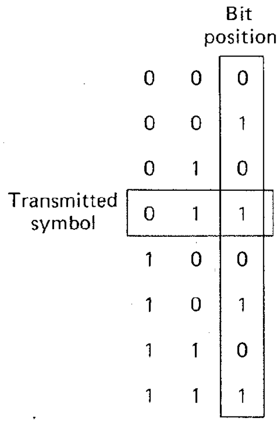
$$\frac{P_B}{P_E} = \frac{2^{k-1}}{2^k - 1} = \frac{M/2}{M - 1} \tag{3.127}$$

In the limit as k increases we get

$$\lim_{k\to\infty} \frac{P_B}{P_E} = \frac{1}{2}$$

A simple example will make Equation (3.127) intuitively acceptable. Figure 3.35 describes an octal message set. The message symbols (assumed equally likely) are to be transmitted on orthogonal waveforms such as FSK. With orthogonal signaling, a decision error willl transform the correct signal into any one of the ($M - 1$) incorrect signals with equal probability. The example in Figure 3.35 indicates that the symbol comprised of bits 0 1 1 was transmitted. An error might occur in any one of the other $2^k - 1 = 7$ symbols, with equal probability. Notice that just because a symbol error is made does not mean that all the bits within the symbol will be in error. In Figure 3.35, if the receiver decides that the transmitted symbol is the bottom one listed, comprised of bits 1 1 1, two of the three transmitted symbol bits will be correct; only one bit will be in error. It should be apparent that P_B will be less than or equal to P_E.

Consider any of the bit-position columns in Figure 3.35. For each bit position, the digit occupancy consists of 50% ones and 50% zeros. In the context of the first bit position (rightmost column) and the transmitted symbol, how many ways are there to cause an error to the binary one? There are $2^{k-1} = 4$ ways (four places where zeros appear in the column) that a bit error can be made; it is the same for each of the columns. The final relationship, P_B/P_E, for orthogonal sig-



			Bit position
	0	0	0
	0	0	1
	0	1	0
Transmitted symbol	0	1	1
	1	0	0
	1	0	1
	1	1	0
	1	1	1

Figure 3.35 Example of P_B versus P_E.

naling, in Equation (3.127), is obtained by forming the following ratio: the number of ways that a bit error can be madc (2^{k-1}) divided by the number of ways that a symbol error can be made ($2^k - 1$). For the Figure 3.35 example, P_B/P_E = 4/7.

3.9.4 Bit Error Probability versus Symbol Error Probability for Multiple Phase Signaling

For the case of MPSK signaling, P_B is less than or equal to P_E, just as in the case of MFSK signaling. However, there is an important difference. For orthogonal signaling, selecting any one of the $(M - 1)$ erroneous symbols is equally likely. In the case of MPSK signaling, each signal vector is not equidistant from all of the others. Figure 3.36a illustrates an 8-ary decision space with the pie-shaped regions denoted by the 8-ary symbols in binary notation. If symbol (0 1 1) is transmitted, it is clear that should an error occur, the transmitted signal will most likely be mistaken for one of its closest neighbors, (0 1 0) or (1 0 0). The likelihood that (0 1 1) would get mistaken for (1 1 1) is relatively remote. If the assignment of bits to symbols follows the binary sequence shown in the symbol decision regions of Figure 3.36a, some symbol errors will usually result in two or more bit errors, even with a large signal-to-noise ratio.

For nonorthogonal schemes, such as MPSK signaling, one often uses a binary-to M-ary code such that binary sequences corresponding to adjacent symbols (phase shifts) differ in only one bit position; thus when an M-ary symbol error occurs, it is more likely that only one of the k input bits will be in error. A code that provides this desirable feature is the Gray code [9]; Figure 3.36b illustrates the bit-to-symbol assignment using a Gray code for 8-ary PSK. Here it can be seen that neighboring symbols differ from one another in only one bit position. Therefore, the occurrence of a multibit error, for a given symbol error, is much

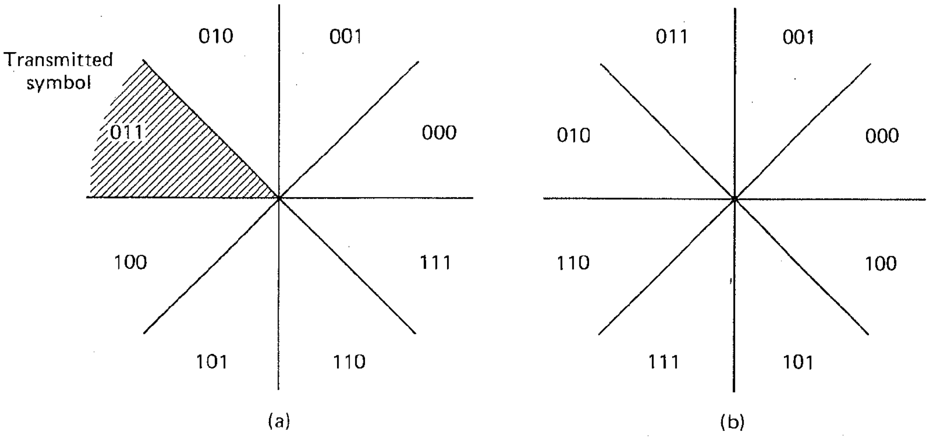


Figure 3.36 Binary-coded versus Gray-coded decision regions in an MPSK signal space. (a) Binary coded. (b) Gray coded.

reduced compared to the uncoded binary assignment seen in Figure 3.36a. Utilizing the Gray code assignment, it can be shown [7] that

$$P_B \simeq \frac{P_E}{\log_2 M} = \frac{P_E}{k} \quad (\text{for } P_E \ll 1) \tag{3.128}$$

Recall from Section 3.8.4 that BPSK and QPSK signaling have the same bit error probability. Here in Equation (3.128) we verify that they do not have the same symbol error probability. For BPSK, $P_E = P_B$. However, for QPSK, $P_E \simeq 2P_B$.

An exact closed-form expression for the bit error probability, P_B, of 8-ary PSK, together with tight upper and lower bounds on P_B for M-ary PSK with larger M, may be found in Lee [12].

3.9.5 Effects of Intersymbol Interference

In the previous sections and in Chapter 2 we have treated the detection of signals in the presence of AWGN under the assumption that there is no intersymbol interference (ISI). Thus the analysis has been straightforward, since the zero-mean AWGN process is characterized by its variance alone. In practice we find that ISI is often a second source of interference which must be accounted for. As explained in Section 2.11, ISI can be generated by the use of bandlimiting filters at the transmitter output, in the channel, or at the receiver input. The result of this additional interference is to degrade the error probabilities for coherent as well as for noncoherent reception. Analysis involving ISI in addition to AWGN is much more complicated since it involves the impulse response of the channel. The subject will not be treated here; however, for those readers interested in the details of the analysis, References [13–18] should prove interesting.

3.10 CONCLUSION

We have catalogued some basic bandpass digital modulation formats, particularly phase shift keying (PSK) and frequency shift keying (FSK). We have considered a geometric view of signal vectors and noise vectors, particularly antipodal and orthogonal signal sets. This geometric view allows us to consider the detection problem in the light of an orthogonal signal space and signal regions. This view of the space, and the effect of noise vectors causing transmitted signals to be received in the incorrect region, facilitates the understanding of the detection problem and the performance of various modulation and demodulation techniques. In Chapter 7 we reconsider the subjects of modulation and demodulation, and we investigate some bandwidth-efficient modulation techniques.

REFERENCES

1. Nyquist, H., "Thermal Agitation of Electric Charge in Conductors," *Phys. Rev.*, vol. 32, July 1928, pp. 110–113.

2. Arthurs, E., and Dym, H., "On the Optimum Detection of Digital Signals in the Presence of White Gaussian Noise—A Geometric Interpretation of Three Basic Data Transmission Systems," *IRE Trans. Commun. Syst.*, December 1962.
3. Wozencraft, J. M., and Jacobs, I. M., *Principles of Communication Engineering,* John Wiley & Sons, Inc., New York, 1965.
4. Van Trees, H. L., *Detection, Estimation, and Modulation Theory,* Part 1, John Wiley & Sons, Inc., New York, 1968.
5. Park, J. H., Jr., "On Binary DPSK Detection," *IEEE Trans. Commun.*, vol. COM26, no. 4, Apr. 1978, pp. 484–486.
6. Ziemer, R. E., and Peterson, R. L., *Digital Communications and Spread Spectrum Systems,* Macmillan Publishing Company, Inc., New York, 1985.
7. Lindsey, W. C., and Simon, M. K., *Telecommunication Systems Engineering,* Prentice-Hall, Inc., Englewood Cliffs, N.J., 1973.
8. Whalen, A. D., *Detection of Signals in Noise,* Academic Press, Inc., New York, 1971.
9. Korn, I., *Digital Communications,* Van Nostrand Reinhold Company, Inc., New York, 1985.
10. Couch, L. W. II, *Digital and Analog Communication Systems,* Macmillan Publishing Company, New York, 1983.
11. Viterbi, A. J., *Principles of Coherent Communications,* McGraw-Hill Book Company, New York, 1966.
12. Lee, P. J., "Computation of the Bit Error Rate of Coherent *M*-ary PSK with Gray Code Bit Mapping," *IEEE Trans. Commun.*, vol. COM34, no. 5, May 1986, pp. 488–491.
13. Hoo, E. Y., and Yeh, Y. S., "A New Approach for Evaluating the Error Probability in the Presence of Intersymbol Interference and Additive Gaussian Noise," *Bell Syst. Tech. J.*, vol. 49, Nov. 1970, pp. 2249–2266.
14. Shimbo, O., Fang, R. J., and Celebiler, M., "Performance of *M*-ary PSK Systems in Gaussian Noise and Intersymbol Interference," *IEEE Trans. Inf. Theory,* vol. IT19, Jan. 1973, pp. 44–58.
15. Prabhu, V. K., "Error Probability Performance of *M*-ary CPSK Systems with Intersymbol Interference," *IEEE Trans. Commun.*, vol. COM21, Feb. 1973, pp. 97–109.
16. Yao, K., and Tobin, R. M., "Moment Space Upper and Lower Error Bounds for Digital Systems with Intersymbol Interference," *IEEE Trans. Inf. Theory,* vol. IT22, Jan. 1976, pp. 65–74.
17. King, M. A., Jr., "Three Dimensional Geometric Moment Bounding Techniques," *J. Franklin Inst.*, vol. 309, no. 4, Apr. 1980, pp. 195–213.
18. Prabhu, V. K., and Salz, J., "On the Performance of Phase-Shift Keying Systems," *Bell Syst. Tech. J.*, vol. 60, Dec. 1981, pp. 2307–2343.

PROBLEMS

3.1. Determine whether or not $s_1(t)$ and $s_2(t)$ are orthogonal over the interval $(-1.5T_2 < t < 1.5T_2)$, where $s_1(t) = \cos(2\pi f_1 t + \phi_1)$, $s_2(t) = \cos(2\pi f_2 t + \phi_2)$, and $f_2 = 1/T_2$ for the following cases.

(a) $f_1 = f_2$ and $\phi_1 = \phi_2$

(b) $f_1 = \frac{1}{3} f_2$ and $\phi_1 = \phi_2$

(c) $f_1 = 2f_2$ and $\phi_1 = \phi_2$
(d) $f_1 = \pi f_2$ and $\phi_1 = \phi_2$
(e) $f_1 = f_2$ and $\phi_1 = \phi_2 + \pi/2$
(f) $f_1 = f_2$ and $\phi_1 = \phi_2 + \pi$

3.2. (a) Show that the three functions illustrated in Figure P3.1 are pairwise orthogonal over the interval $(-2, 2)$.
(b) Determine the value of the constant, A, that makes the set of functions in part (a) an orthonormal set.
(c) Express the following waveform, $x(t)$, in terms of the orthonormal set of part (b).

$$x(t) = \begin{cases} 1 & \text{for } 0 \le t \le 2 \\ 0 & \text{otherwise} \end{cases}$$

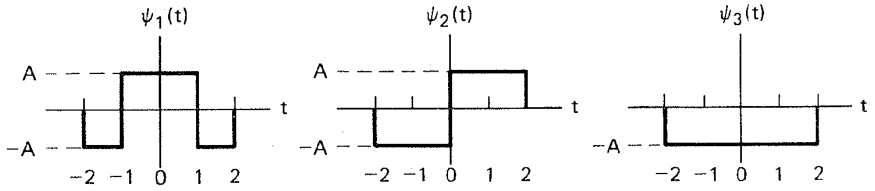


Figure P3.1

3.3. Consider the functions

$$\psi_1(t) = \exp(-|t|) \quad \text{and } \psi_2 = 1 - A\exp(-2|t|)$$

Determine the constant, A, such that $\psi_1(t)$ and $\psi_2(t)$ are orthogonal over the interval $(-\infty, \infty)$.

3.4. Find the expected number of bit errors made in one day by the following continuously operating coherent BPSK receiver. The data rate is 5000 bits/s. The input digital waveforms are $s_1(t) = A \cos \omega_0 t$ and $s_2(t) = -A \cos \omega_0 t$, where $A = 1$ mV and the single-sided noise power spectral density is $N_0 = 10^{-11}$ W/Hz. Assume that signal power and energy per bit are normalized relative to a 1-Ω resistive load.

3.5. A continuously operating coherent BPSK system makes errors at the average rate of 100 errors per day. The data rate is 1000 bits/s. The single-sided noise power spectral density is $N_0 = 10^{-10}$ W/Hz.
(a) If the system is ergodic, what is the average bit error probability?
(b) If the value of received average signal power per bit is adjusted to be 10^{-6} W, will this received power be adequate to maintain the error probability found in part (a)?

3.6. If a system's main performance criterion is bit error probability, which of the following two modulation schemes would be selected for an AWGN channel? Show computations.

Binary coherent orthogonal FSK with $E_b/N_0 = 12$ dB

Binary noncoherent orthogonal FSK with $E_b/N_0 = 14$ dB

3.7. If a system's main performance criterion is bit error probability, which of the following two modulation schemes would be selected for an AWGN channel? Show computations.

Binary noncoherent orthogonal FSK with $E_b/N_0 = 13$ dB

Binary coherent PSK with $E_b/N_0 = 8$ dB

3.8. The bit stream

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

is to be transmitted using DPSK modulation. Show four different differentially encoded sequences that can represent the data sequence above, and explain the algorithm that generated each.

3.9. (a) Calculate the minimum required bandwidth for a noncoherently detected orthogonal binary FSK system. The higher-frequency signaling tone is 1 MHz and the symbol duration is 1 ms.

(b) What is the minimum required bandwidth for a noncoherent MFSK system having the same symbol duration?

3.10. Consider a BPSK system with equally likely waveforms $s_1(t) = \cos \omega_0 t$ and $s_2(t) = -\cos \omega_0 t$. At the matched filter detector, the $s_1(t)$ reference is $\cos (\omega_0 t + \phi)$, where ϕ is a phase error. Calculate the value of the phase error that would increase the probability of bit error from 2.0×10^{-3} to 2.5×10^{-3} relative to no phase error for an AWGN channel.

3.11. Find the probability of bit error, P_B, for the coherent matched filter detection of the equally likely binary FSK signals

$$s_1(t) = 0.5 \cos 2000\pi t$$

$$s_2(t) = 0.5 \cos 2020\pi t$$

where the two-sided AWGN power spectral density is $N_0/2 = 0.0001$. Assume that the symbol duration is $T = 0.01$ s.

3.12. Find the optimum (minimum probability of error) threshold, γ_0, for detecting the equally likely signals $s_1(t) = \sqrt{2E/T} \cos \omega_0 t$ and $s_2(t) = \sqrt{\frac{1}{2}E/T} \cos (\omega_0 t + \pi)$ in AWGN, using a correlator receiver as shown in Figure 3.7b. Assume a reference signal of $\psi_1(t) = \sqrt{2/T} \cos \omega_0 t$.

3.13. A system using matched filter detection of equally likely BPSK signals, $s_1(t) = \sqrt{2E/T} \cos \omega_0 t$ and $s_2(t) = \sqrt{2E/T} \cos (\omega_0 t + \pi)$, operates in AWGN with a received E_b/N_0 of 6.8 dB. Assume that $\mathrm{E}\{z(T)\} = \pm\sqrt{E}$.

(a) Find the minimum probability of bit error, P_B, for this signal set and E_b/N_0.

(b) If the decision threshold is $\gamma = 0.1\sqrt{E}$, find P_B.

(c) The threshold of $\gamma = 0.1\sqrt{E}$ is optimum for a particular set of a priori probabilities, $P(s_1)$ and $P(s_2)$. Find the values of these probabilities (refer to Section B.2).

3.14. A binary source with equally likely symbols controls the switch position in a transmitter operating over an AWGN channel, as shown in Figure P3.2. The noise has two-sided spectral density $N_0/2$. Assume antipodal signals of time duration T seconds and energy E joules. The system clock produces a clock pulse every T seconds, and the binary source rate is $1/T$ bits/s. Under *normal* operation, the switch is up when the source produces a binary zero, and it is down when the source produces a binary one. However, the switch is *faulty*. With probability, p, it will be thrown in the wrong direction during a given T-second interval. The presence of a switch error during any interval is independent of the presence of a switch error at any other time. Assume that $\mathrm{E}\{z(T)\} = \pm\sqrt{E}$.

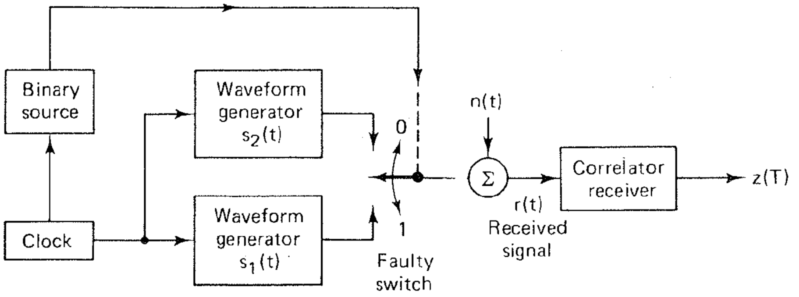


Figure P3.2

(a) Sketch the conditional probability functions, $p(z|s_1)$ and $p(z|s_2)$.

(b) The correlator receiver observes $r(t)$ in the interval $(0, T)$. Sketch the block diagram of an optimum receiver for minimizing the bit error probability when it is known that the switch is faulty with probability, p.

(c) Which one of the following two systems would you prefer to have?

$$p = 0.1 \quad \text{and} \quad \frac{E_b}{N_0} = \infty$$

$$p = 0 \quad \text{and} \quad \frac{E_b}{N_0} = 7 \text{ dB}$$

3.15. (a) Consider a 16-ary PSK system with symbol error probability, $P_E = 10^{-5}$. A Gray code is used for the symbol to bit assignment. What is the approximate bit error probability?

(b) Repeat part (a) for a 16-ary orthogonal FSK system.

3.16. Consider a coherent orthogonal MFSK system with $M = 8$ having the equally likely waveforms $s_i(t) = A \cos 2\pi f_i t$, $i = 1, \ldots, M$, $0 \leq t \leq T$, where $T = 0.2$ ms. The received carrier amplitude, A, is 1 mV, and the two-sided AWGN spectral density, $N_0/2$, is 10^{-11} W/Hz. Calculate the probability of bit error, P_B.

3.17. A bit error probability of $P_B = 10^{-3}$ is required for a system with a data rate of 100 kbits/s to be transmitted over an AWGN channel using coherently detected MPSK modulation. The system handwidth is 50 kHz. Assume that the filter has a roll-off characteristic of $r = 1$ and that a Gray code is used for the symbol to bit assignment.

(a) What E_s/N_0 is required for the specified P_B?

(b) What E_b/N_0 is required?

3.18. A differentially coherent MPSK system operates over an AWGN channel with an E_b/N_0 of 10 dB. What is the symbol error probability for $M = 8$ and equally likely symbols?

3.19. If a system's main performance criterion is bit error probability, which of the following two modulation schemes would be selected for transmission over an AWGN channel? Show computations.

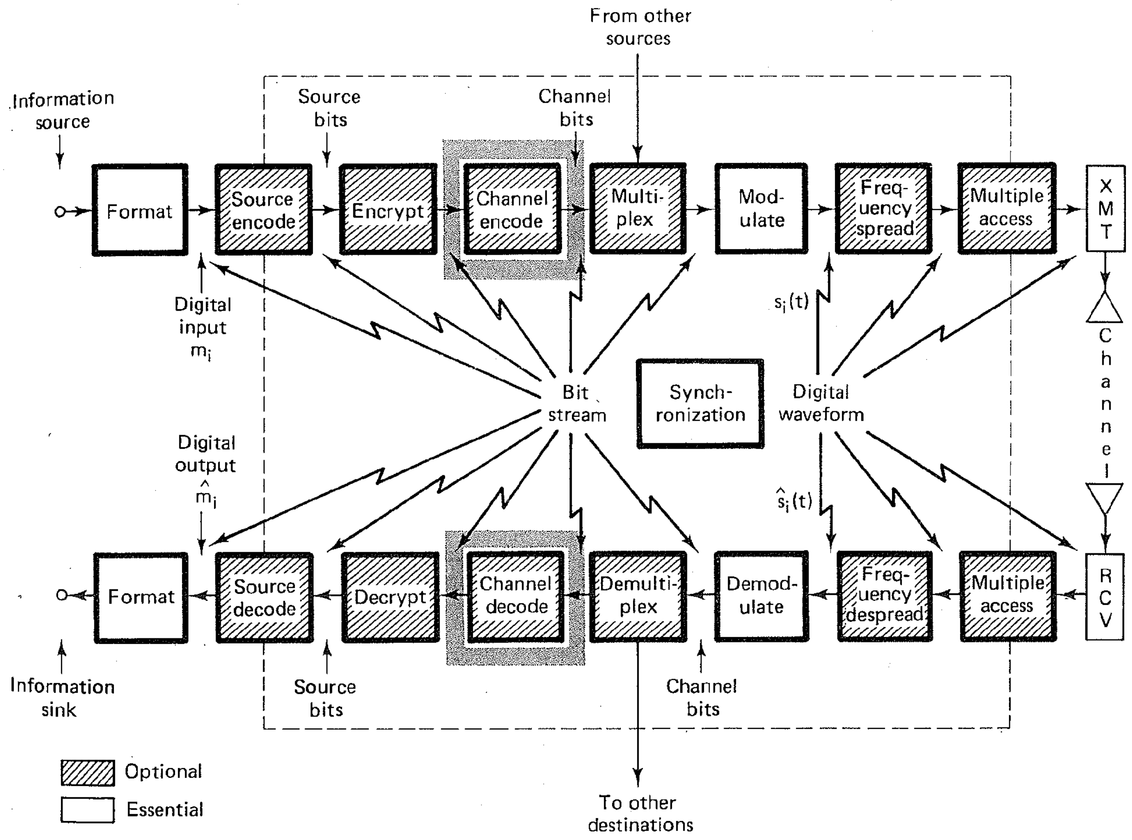
$$\text{coherent 8-ary orthogonal FSK with } \frac{E_b}{N_0} = 8 \text{ dB}$$

$$\text{coherent 8-ary PSK with } \frac{E_b}{N_0} = 13 \text{ dB}$$

(Assume that a Gray code is used for the MPSK symbol-to-bit assignment.)

CHAPTER 5

Channel Coding: Part 1



Channel coding refers to the class of signal transformations designed to improve communications performance by enabling the transmitted signals to better withstand the effects of various channel impairments, such as noise, fading, and jamming. Usually, the goal of channel coding is to reduce the probability of bit error (P_B), or to reduce the required E_b/N_0, at the cost of expending more bandwidth than would otherwise be necessary. The exceptions to this are the combined modulation and coding techniques for bandlimited channels described in Chapter 7. Why do you suppose channel coding has become such a popular way to provide performance improvement? The use of large-scale integrated (LSI) circuits has made it possible to provide as much as an 8-dB performance improvement through coding, at much less cost than through the use of other methods such as higher-power transmitters or larger antennas.

5.1 WAVEFORM CODING

Channel coding can be partitioned into two study areas, waveform (or signal design) coding and structured sequences (or structured redundancy), as shown in Figure 5.1. *Waveform coding* deals with transforming waveforms into "better waveforms," to make the detection process less subject to errors. *Structured sequences* deals with transforming data sequences into "better sequences," having structured redundancy (redundant bits). The redundant bits can then be used for the detection and correction of errors. The encoding procedure provides the coded signal (whether waveforms or structured sequences) with better distance

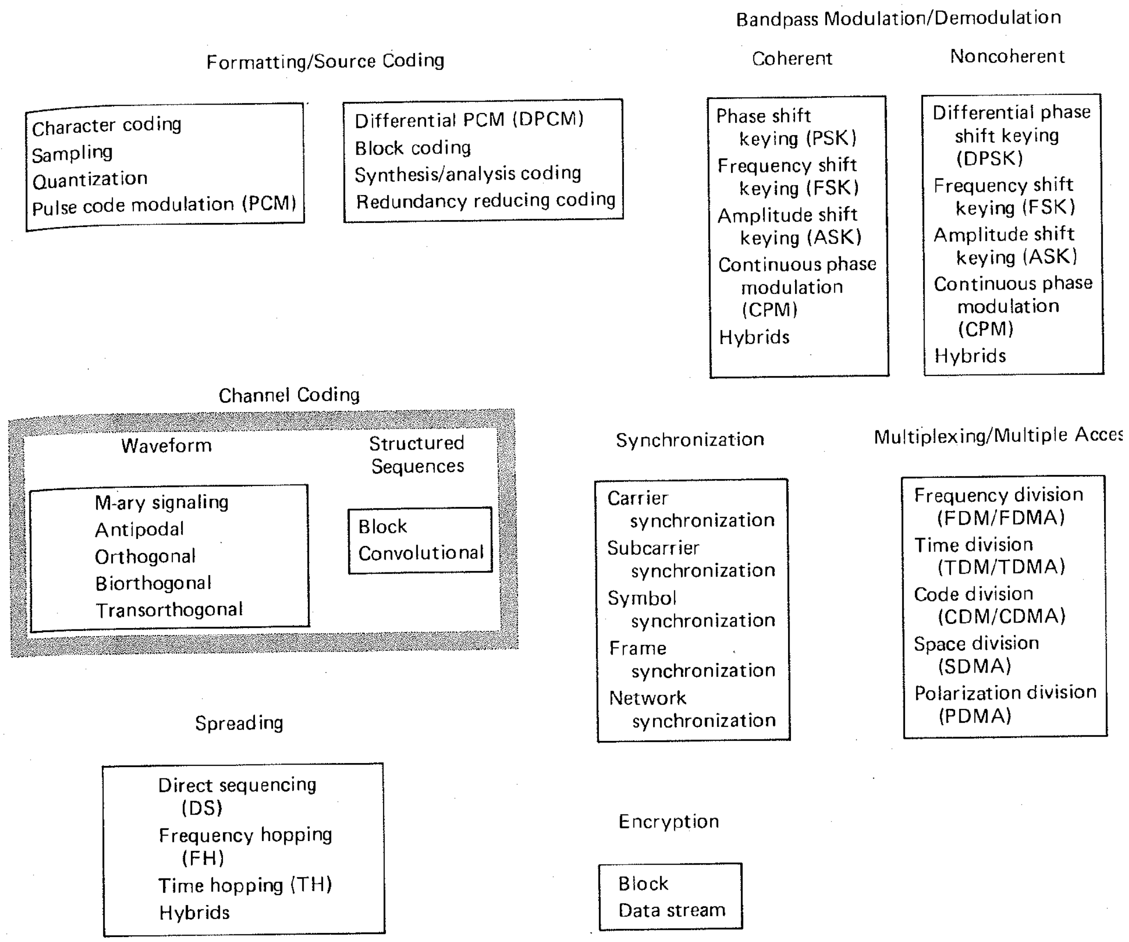


Figure 5.1 Basic digital communication transformations.

properties than those of their uncoded counterparts. First, we consider some waveform coding techniques. Then, starting with Section 5.3, we treat the morc popular subject of structured sequences.

5.1.1 Antipodal and Orthogonal Signals

Antipodal and orthogonal signals have been discussed in Chapter 3; we shall repeat the paramount features of these signal classes. The example shown in Figure 5.2 illustrates the analytical representation, $s_1(t) = -s_2(t) = \sin \omega_0 t$, $0 \leq t \leq T$, of an antipodal signal set, as well as its waveform representation and its vectorial representation. What are some synonyms or analogies that are used to describe *antipodal signals*? We can say that such signals are mirror images, or that one signal is the negative of the other, or that the signals are 180° apart.

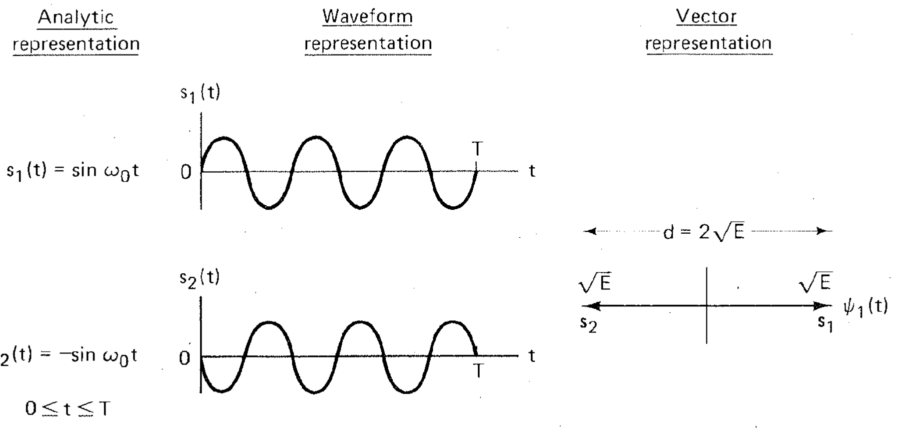


Figure 5.2 Example of an antipodal signal set.

The example shown in Figure 5.3 illustrates an orthogonal signal set. We know that sin x and cos x are orthogonal functions; similarly, sin mx and sin nx, where m and n are integers and $m \neq n$, are also orthogonal functions (see Section A.2.1). In Figure 5.3 we have chosen a pulse waveform example because it provides a clearer picture of orthogonality. The pulse waveform is described by

$$\begin{aligned} s_1(t) &= p(t) \qquad & 0 \le t \le T \\ s_2(t) &= p\left(t - \frac{T}{2}\right) \qquad & 0 \le t \le T \end{aligned} \tag{5.1}$$

where $p(t)$ is a pulse with duration $\tau = T/2$, and T is the symbol duration. In general, a set of equal energy signals $s_i(t)$, where $i = 1, 2, \ldots, M$, constitutes

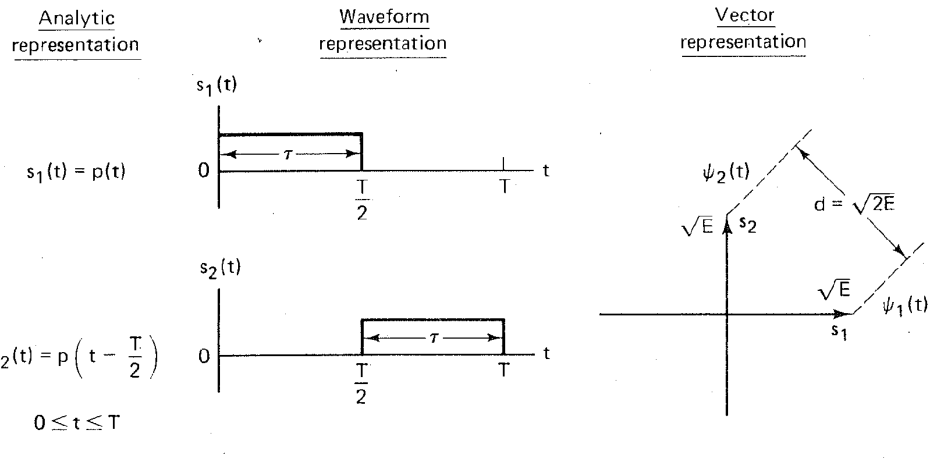


Figure 5.3 Example of a binary orthogonal signal set.

an orthogonal set, if, and only if,

$$z_{ij} = \frac{1}{E}\int_0^T s_i(t)s_j(t)\,dt = \begin{cases} 1 & \text{for } i = j \\ 0 & \text{otherwise} \end{cases} \tag{5.2}$$

where z_{ij} is called the *cross-correlation coefficient*, and where E is the signal energy expressed as

$$E = \int_0^T s_i^2(t)\,dt \tag{5.3}$$

The waveform representation in Figure 5.3 illustrates that $s_1(t)$ and $s_2(t)$ cannot interfere with one another because they are disjoint in time. The vectorial representation illustrates the perpendicular relationship between orthogonal signals. Let us consider some alternative descriptions of orthogonal signals or vectors. We can say that the inner or dot product of two different vectors in the orthogonal set must equal zero. In a two- or three-dimensional Cartesian coordinate space, we can describe the signal vectors, geometrically, as being mutually perpendicular to one another. We can say that one vector has zero projection on the other, or that one signal cannot interfere with the other, since they do not share the same *signal space*.

5.1.2 *M*-ary Signaling

With M-ary signaling, the processor accepts k data bits at a time. It then instructs the modulator to produce one of $M = 2^k$ waveforms; binary signaling is the special case where $k = 1$. For $k > 1$, M-ary signaling, as described in Chapter 3, can be regarded as a *waveform coding* procedure. For orthogonal signaling (e.g., MFSK), as k increases there will be an improved error performance or a reduction in required E_b/N_0, at the expense of bandwidth; nonorthogonal signaling (e.g., MPSK) can manifest improved bandwidth efficiency, at the expense of degraded error performance or an increase in required E_b/N_0. By the appropriate choice of signal waveforms, one can trade off error performance versus E_b/N_0 performance, versus bandwidth efficiency. Such trade-offs are treated in greater detail in Chapter 7.

5.1.3 Waveform Coding with Correlation Detection

Waveform coding procedures transform a waveform set into an improved waveform set. The improved waveform set can then be used to provide improved P_B compared to the original set. The most popular of such *waveform codes* are referred to as *orthogonal* and *biorthogonal codes*. The encoding procedure endeavors to make each of the waveforms in the coded signal set as unalike as possible; the goal is to render the cross-correlation coefficient, z_{ij}, among all pairs of signals, as described in Equation (5.2), as small as possible. The smallest possible value of the cross-correlation coefficient occurs when the signals are anticorrelated ($z_{ij} = -1$); however, this can be achieved only when the number of symbols in the set is two ($M = 2$) and the symbols are *antipodal*. In general, it is possible to make all the cross-correlation coefficients equal to zero [1]. The set is then said to be *orthogonal.* Antipodal signal sets are optimum in the sense that each signal is most distant from the other signal in the set; this is seen in Figure 5.2 where the distance, d, between signal vectors is seen to be $d = 2\sqrt{E}$, where E represents the signal energy during a symbol duration T, as expressed in Equation (5.3). Compared to antipodal signals, the distance properties of orthogonal signal sets can be thought of as "second best" (for a given level of waveform energy). In Figure 5.3 the distance between the orthogonal signal vectors is seen to be $d = \sqrt{2E}$.

The *cross-correlation* between two signals is a measure of the *distance* between the signal vectors. The smaller the cross-correlation, the more distant are the vectors from each other. This can be verified in Figure 5.2, where the antipodal signals (whose $z_{ij} = -1$) are represented by vectors that are most distant from each other, and in Figure 5.3, where the orthogonal signals (whose $z_{ij} = 0$) are represented by vectors that are closer to one another than the antipodal vectors. It should be obvious that the distance between two identical waveforms (whose $z_{ij} = 1$) is zero.

Figure 5.4 illustrates the replacement of a 2-bit data set with an improved (orthogonal) codeword set. Both the original data set and the codeword replacement set are comprised of the binary digits (1, 0). Also shown in the figure is the

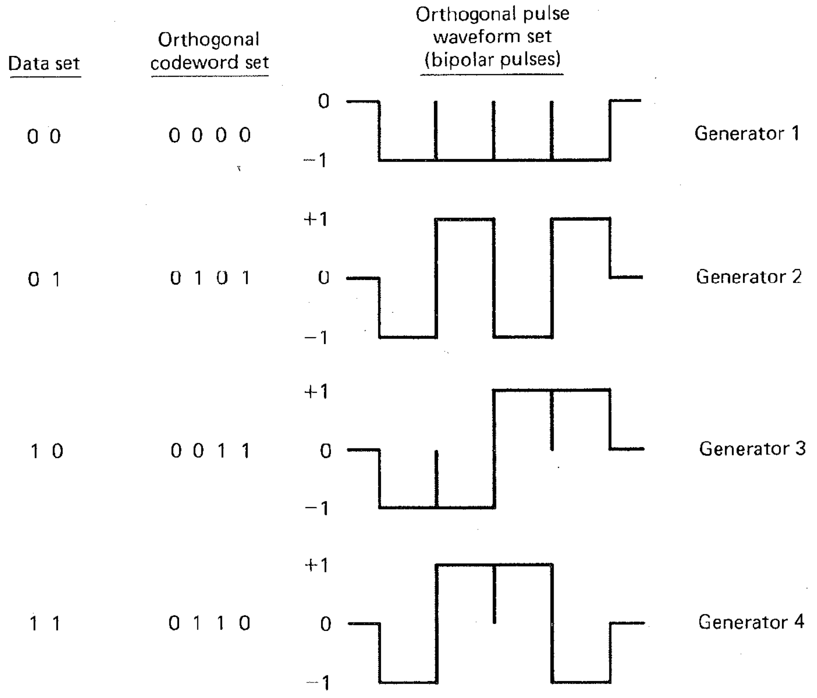


Figure 5.4 Replacement of data set with orthogonal codeword set and waveform set.

waveform set comprised of bipolar pulses (+1, −1) that represents the codeword set. Equation (5.2) is stated in terms of waveforms. However, when the waveform set, $\{s_i(t)\}$, is represented by binary digits, it is easy to show that Equation (5.2) can be simplified as follows:

$$z_{ij} = \frac{\text{number of digit agreements} - \text{number of digit disagreements}}{\text{total number of digits}} \tag{5.4}$$

$$z_{ij} = \begin{cases} 1 & \text{for } i = j \\ 0 & \text{otherwise} \end{cases}$$

where $i, j = 1, \ldots, M$, and M is the size of the codeword set. Using Equation (5.4), one can quickly verify that the codeword set in Figure 5.4 is orthogonal. Transmitting data with such an orthogonal set in place of the original data set results in larger distances among signaling waveforms, and thus yields better error performance for a given SNR.

Consider a set of $M = 2^k$ messages that are to be transmitted, using PSK modulation, over a channel disturbed by additive white Gaussian noise (AWGN). The transmitter shown in Figure 5.5, stores or generates the M pulse waveforms of the type shown in Figure 5.4. A message is transmitted by selecting one of the M waveform generators to *phase modulate* the carrier, such that the phase ($\phi_j = 0$ or π) of the carrier during each bit time, $0 \le t \le T_b$, corresponds to the amplitudes ($j = -1$ or 1) of the generating pulse waveform. At the receiver in Figure 5.6 the noisy signal is demodulated to baseband and fed to the M correlators (or matched filters). Correlation is performed over a codeword duration, $0 \le t \le T$, where $T = (\log_2 M)T_b = kT_b$. With orthogonally coded waveforms, in the absence of noise, the outputs of all correlators, except the one corresponding to the transmitted codeword, are zero.

5.1.4 Orthogonal Codes

A 1-bit data set can be transformed, using *orthogonal* codewords of two-digits each, described by the matrix $\mathbf{H}_1$ as follows:

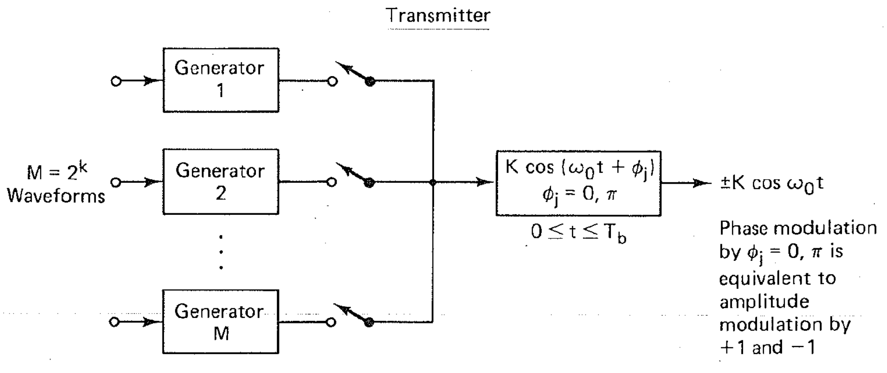


Figure 5.5 Waveform-encoded phase coherent system (transmitter).

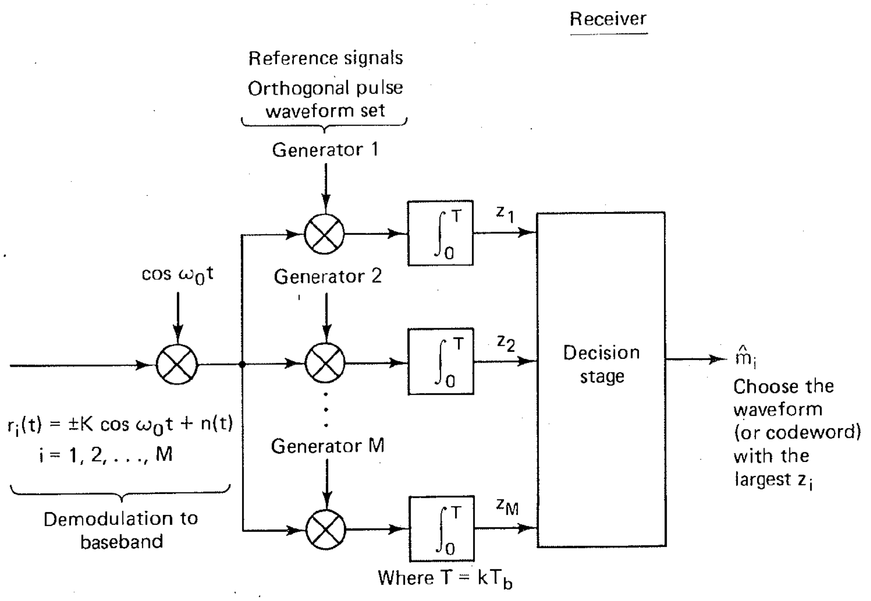


Figure 5.6 Waveform coding with correlation detection.

$$\begin{array}{c c}
\underline{\text{Data set}} & \underline{\text{Orthogonal codeword set}} \\
\begin{matrix} 0 \\ 1 \end{matrix} & \mathbf{H}_1 = \begin{bmatrix} 0 & 0 \\ 0 & 1 \end{bmatrix}
\end{array}$$

For this, and the following examples, use Equation (5.4) to verify the orthogonality of the codeword set. To encode a 2-bit data set, we extend the foregoing set both horizontally and vertically, creating matrix $\mathbf{H}_2$.

$$\begin{array}{c c}
\underline{\text{Data set}} & \underline{\text{Orthogonal codeword set}} \\
\begin{matrix} 0 & 0 \\ 0 & 1 \\ 1 & 0 \\ 1 & 1 \end{matrix} & \mathbf{H}_2 = \left[\begin{array}{cc:cc} 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 \\ \hdashline 0 & 0 & 1 & 1 \\ 0 & 1 & 1 & 0 \end{array}\right] = \begin{bmatrix} \mathbf{H}_1 & \mathbf{H}_1 \\ \mathbf{H}_1 & \overline{\mathbf{H}_1} \end{bmatrix}
\end{array}$$

The lower right quadrant is the complement of the prior codeword set. We continue the same construction rule to obtain an orthogonal set $\mathbf{H}_3$ for a 3-bit data set.

$$\begin{array}{ccc}
\text{Data set} & & \text{Orthogonal codeword set} \\
\begin{array}{c} 0\;0\;0 \\ 0\;0\;1 \\ 0\;1\;0 \\ 0\;1\;1 \\ 1\;0\;0 \\ 1\;0\;1 \\ 1\;1\;0 \\ 1\;1\;1 \end{array} & \mathbf{H}_3 = & \left[\begin{array}{cccc:cccc}
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 \\
0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 \\
0 & 1 & 1 & 0 & 0 & 1 & 1 & 0 \\
\hdashline
0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\
0 & 1 & 0 & 1 & 1 & 0 & 1 & 0 \\
0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 \\
0 & 1 & 1 & 0 & 1 & 0 & 0 & 1
\end{array}\right] = \begin{bmatrix} \mathbf{H}_2 & \mathbf{H}_2 \\ \mathbf{H}_2 & \overline{\mathbf{H}_2} \end{bmatrix}
\end{array}$$

In general, we can construct a codeword set, $\mathbf{H}_k$, of dimension $2^k \times 2^k$, called a *Hadamard matrix*, for a k-bit data set from the $\mathbf{H}_{k-1}$ matrix, as follows:

$$\mathbf{H}_k = \begin{bmatrix} \mathbf{H}_{k-1} & \mathbf{H}_{k-1} \\ \mathbf{H}_{k-1} & \overline{\mathbf{H}_{k-1}} \end{bmatrix}$$

Each pair of words in each codeword set, $\mathbf{H}_1$, $\mathbf{H}_2$, $\mathbf{H}_3$, . . . , $\mathbf{H}_k$, . . . , has as many digit agreements as disagreements [2]. Hence, in accordance with Equation (5.4), $z_{ij} = 0$ (for $i \neq j$), and each of the sets is orthogonal.

Just as M-ary signaling with an orthogonal modulation format (such as MFSK) improves the P_B performance, waveform coding with an orthogonally constructed signal set, in combination with correlation detection, produces *exactly the same* improvement. For equally likely, equal-energy orthogonal signals, the probability of codeword (symbol) error can be upper bounded, as follows [2]:

$$P_E(k) \leq (2^k - 1)Q\left(\sqrt{\frac{kE_b}{N_0}}\right) \tag{5.5}$$

where $Q(x)$ is defined in Equation (2.42). For fixed k, as E_b/N_0 is increased, the bound becomes increasingly tight. For $P_E(k) \leq 10^{-3}$, Equation (5.5) is a good approximation of the error probability. The relationship between $P_B(k)$ and $P_E(k)$ given in Equation (3.127) is repeated here:

$$\frac{P_B(k)}{P_E(k)} = \frac{2^{k-1}}{2^k - 1} \tag{5.6}$$

Combining Equations (5.5) and (5.6), the probability of bit error can be bounded as follows:

$$P_B(k) \leq (2^{k-1})Q\left(\sqrt{\frac{kE_b}{N_0}}\right) \tag{5.7}$$

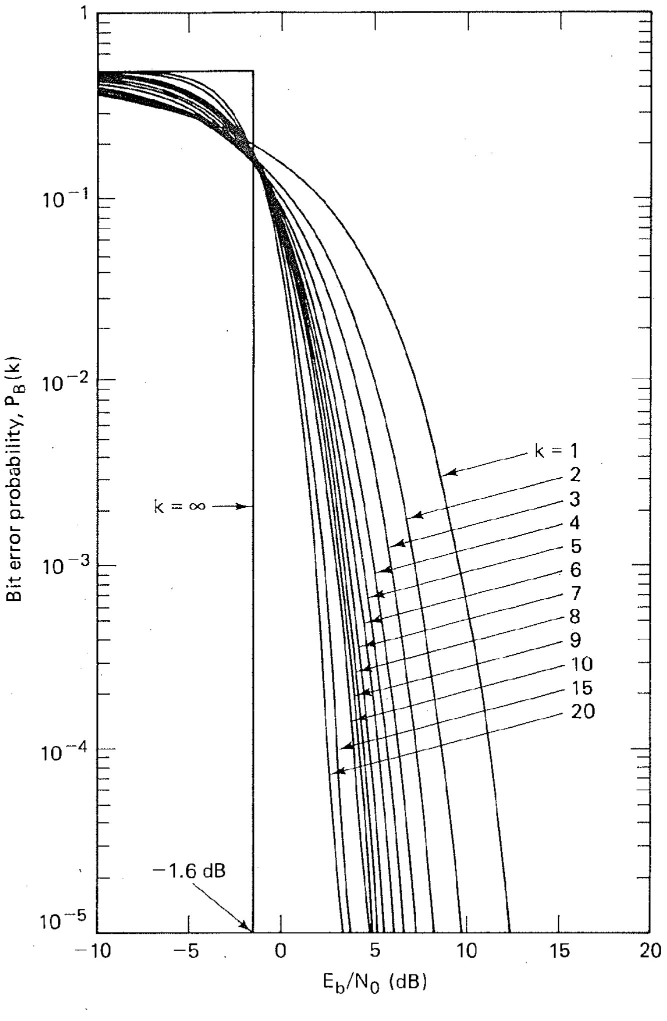


Figure 5.7 Coherent detection of orthogonally coded transmission. (Reprinted from W. C. Lindsey and M. K. Simon, *Telecommunication Systems Engineering*, Prentice-Hall, Inc., Englewood Cliffs, N.J., 1973, courtesy of W. C. Lindsey and Marvin K. Simon.)

$P_B(k)$ is plotted in Figure 5.7 for various values of k; the uncoded case corresponds to the $k = 1$ curve. The performance improvement for $k > 1$ should be obvious. The curves are identical to the orthogonal signaling performance (such as FSK) of Figure 3.25. What price do we pay for this improvement? We need to expend more transmission bandwidth. The orthogonal codes can be described as having $(2^k - k)$ redundant digits. For example, the orthogonal $\mathbf{H}_3$ matrix above reassigns 3-bit messages into 8-bit codewords, resulting in five redundant digits. Therefore, the bandwidth is increased by $\frac{8}{3}$ or, in general, by $2^k/k$. For orthogonal codes, the required transmission bandwidth increases exponentially with k. Compared to structured sequences, this type of coding *does not utilize bandwidth efficiently*.

5.1.5 Biorthogonal Codes

A *biorthogonal* signal set of M total signals or codewords can be obtained from an orthogonal set of $M/2$ signals by augmenting it with the negative of each signal, as follows:

$$\mathbf{B}_k = \left[\begin{array}{c} \mathbf{H}_{k-1} \\ \hline \mathbf{H}_{k-1} \end{array}\right]$$

For example, a 3-bit data set can be transformed into a biorthogonal codeword set as follows:

$$\begin{array}{c c c}
\underline{\text{Data set}} & & \underline{\text{Biorthogonal codeword set}} \\
0\ 0\ 0 & & \left[\begin{array}{cccc} 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 \\ 0 & 1 & 1 & 0 \\ \hdashline 1 & 1 & 1 & 1 \\ 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 0 \\ 1 & 0 & 0 & 1 \end{array}\right] \\
0\ 0\ 1 & & \\
0\ 1\ 0 & & \\
0\ 1\ 1 & \mathbf{B}_3 = & \\
1\ 0\ 0 & & \\
1\ 0\ 1 & & \\
1\ 1\ 0 & & \\
1\ 1\ 1 & &
\end{array}$$

The biorthogonal set is really two sets of orthogonal codes such that each codeword in one set has its antipodal codeword in the other set. The biorthogonal set consists of a *combination of orthogonal and antipodal* signals. With respect to z_{ij} of Equations (5.2) or (5.4), biorthogonal codes can be characterized as

$$z_{ij} = \begin{cases} 1 & \text{for } i = j \\ -1 & \text{for } i \neq j, \, |i - j| = \dfrac{M}{2} \\ 0 & \text{for } i \neq j, \, |i - j| \neq \dfrac{M}{2} \end{cases} \tag{5.8}$$

One advantage of a biorthogonal code over an orthogonal one for the same data set, is that the biorthogonal code requires *one-half* as many bits per codeword (compare the $\mathbf{B}_3$ matrix with the $\mathbf{H}_3$ matrix). Thus the bandwidth requirements for

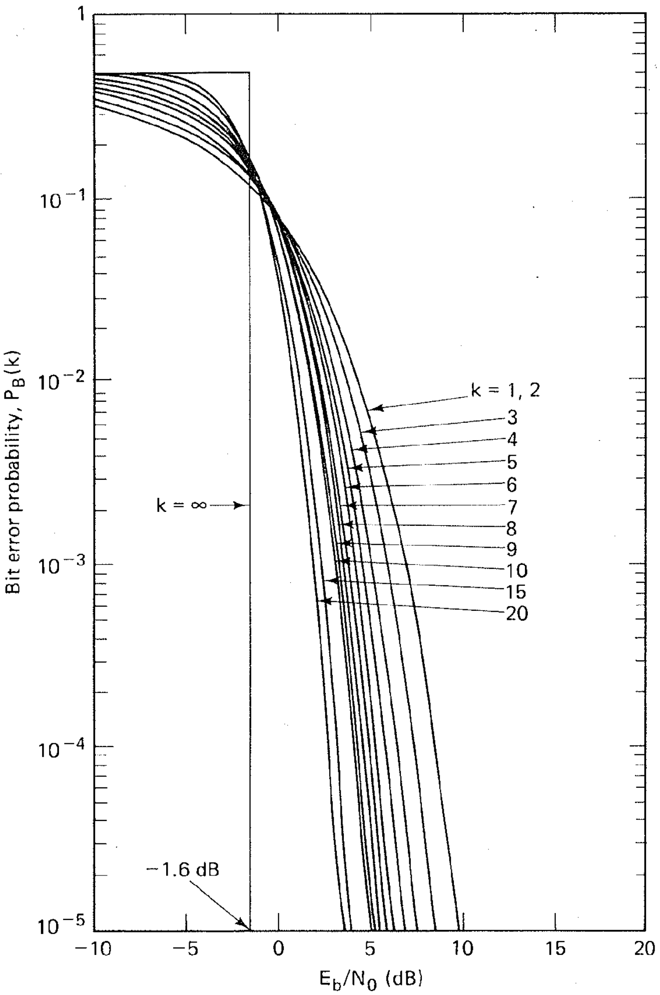


Figure 5.8 Coherent detection of biorthogonally coded transmission. (Reprinted from W. C. Lindsey and M. K. Simon, *Telecommunication Systems Engineering*, Prentice-Hall, Inc., Englewood Cliffs, N.J., 1973, courtesy of W. C. Lindsey and Marvin K. Simon.)

biorthogonal codes are one-half the requirements for comparable orthogonal ones. Since antipodal signal vectors have better distance properties than orthogonal ones, it should come as no surprise that biorthogonal codes perform slightly better than orthogonal ones. For equally likely, equal-energy biorthogonal signals, the probability of codeword (symbol) error can be upper bounded, as follows [2]:

$$P_E(k) \leq (2^k - 2)Q\left(\sqrt{\frac{kE_b}{N_0}}\right) + Q\left(\sqrt{\frac{2kE_b}{N_0}}\right) \tag{5.9}$$

which becomes increasingly tight for fixed k as E_b/N_0 is increased. $P_B(k)$ is a complicated function of $P_E(k)$; we can approximate it with the relationship [2]

$$P_B(k) \simeq \frac{P_E(k)}{2}$$

The approximation is quite good for $k > 3$. Therefore, we can write

$$P_B(k) \lesssim \tfrac{1}{2}\left[(2^k - 2)Q\left(\sqrt{\frac{kE_b}{N_0}}\right) + Q\left(\sqrt{\frac{2kE_b}{N_0}}\right)\right] \tag{5.10}$$

The P_B performance of these biorthogonal codes, shown in Figure 5.8, offers improved performance, compared to the performance of the orthogonal codes shown in Figure 5.7, and requires only *half the bandwidth* of orthogonal codes.

5.1.6 Transorthogonal (Simplex) Codes

A code generated from an orthogonal set by deleting the first digit of each codeword is called a *transorthogonal* or *simplex code*. Such a code is characterized by

$$z_{ij} = \begin{cases} 1 & \text{for } i = j \\ \dfrac{-1}{M - 1} & \text{for } i \neq j \end{cases} \tag{5.11}$$

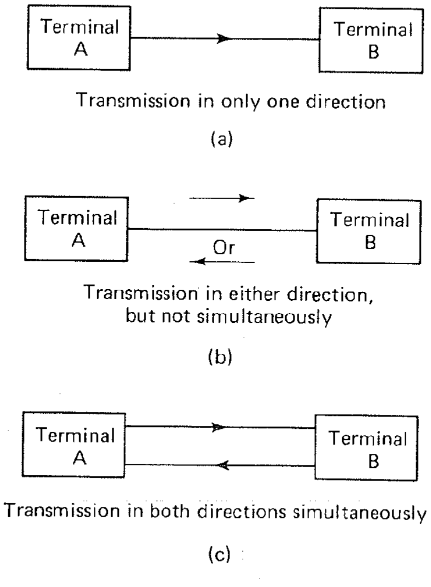
A simplex code represents the *minimum energy* equivalent (in the error probability sense) of the equally likely orthogonal set. In comparing the error performance of orthogonal, biorthogonal, and simplex codes, we can state that simplex coding requires the minimum E_b/N_0 for a specified symbol error rate. However, for a *large value of k*, all three schemes are *essentially identical* in error performance. Biorthogonal coding requires half the bandwidth of the others. However, for each of these codes, bandwidth requirements (and system complexity) grow exponentially with the value of k; therefore, such coding schemes are attractive only when large bandwidths are available. When bandwidth is not plentiful, the structured redundancy techniques (see Section 5.3 through Chapter 6) are more attractive [3]. When bandwidth is *very scarce*, the so-called combined modulation and coding techniques for bandlimited channels are most promising (see Sections 7.10.6 and 7.10.7).

5.2 TYPES OF ERROR CONTROL

Before we discuss the details of structured redundancy, let us describe the two basic ways such redundancy is used for controlling errors. The first, *error detection and retransmission*, utilizes *parity bits* (redundant bits added to the data) to detect that an error has been made. The receiving terminal does not attempt to correct the error; it simply requests the transmitter to retransmit the data. Notice that a two-way link is required for such dialogue between the transmitter and receiver. The second type of error control, *forward error correction* (FEC), requires a one-way link only, since in this case the parity bits are designed for both the detection and correction of errors. We shall see that not all error patterns can be corrected; error-correcting codes are classified according to their error-correcting capabilities.

5.2.1 Terminal Connectivity

Communication terminals are often classified according to their connectivity with other terminals. The possible connections, shown in Figure 5.9, are termed *simplex* (not to be confused with the simplex or transorthogonal codes), *half-duplex*, and *full-duplex*. The simplex connection, in Figure 5.9a, is a one-way link. Transmissions are made from terminal A to terminal B only, never in the reverse direction. The half-duplex connection, in Figure 5.9b, is a link whereby transmissions may be made in either direction but not simultaneously. Finally, the full-duplex connection, in Figure 5.9c, is a two-way link, where transmissions may proceed in both directions simultaneously.



Transmission in only one direction

(a)

Transmission in either direction, but not simultaneously

(b)

Transmission in both directions simultaneously

(c)

Figure 5.9 Terminal connectivity classifications. (a) Simplex. (b) Half-duplex. (c) Full-duplex.

5.2.2 Automatic Repeat Request

When the error control consists of error detection only, the communication system generally needs to provide a means of alerting the transmitter that an error has been detected and that a retransmission is necessary. Such error control procedures are known as *automatic repeat request* or automatic retransmission query (ARQ) methods. Figure 5.10 illustrates three of the most popular ARQ procedures. In each of the diagrams, time is advancing from left to right. The first procedure, called *stop-and-wait ARQ*, is shown in Figure 5.10a. It requires a half-duplex connection only, since the transmitter waits for an acknowledgment (ACK) of each transmission before it proceeds with the next transmission. In the figure, the third transmission block is received in error; therefore, the receiver responds with a negative acknowledgment (NAK), and the transmitter retransmits this third

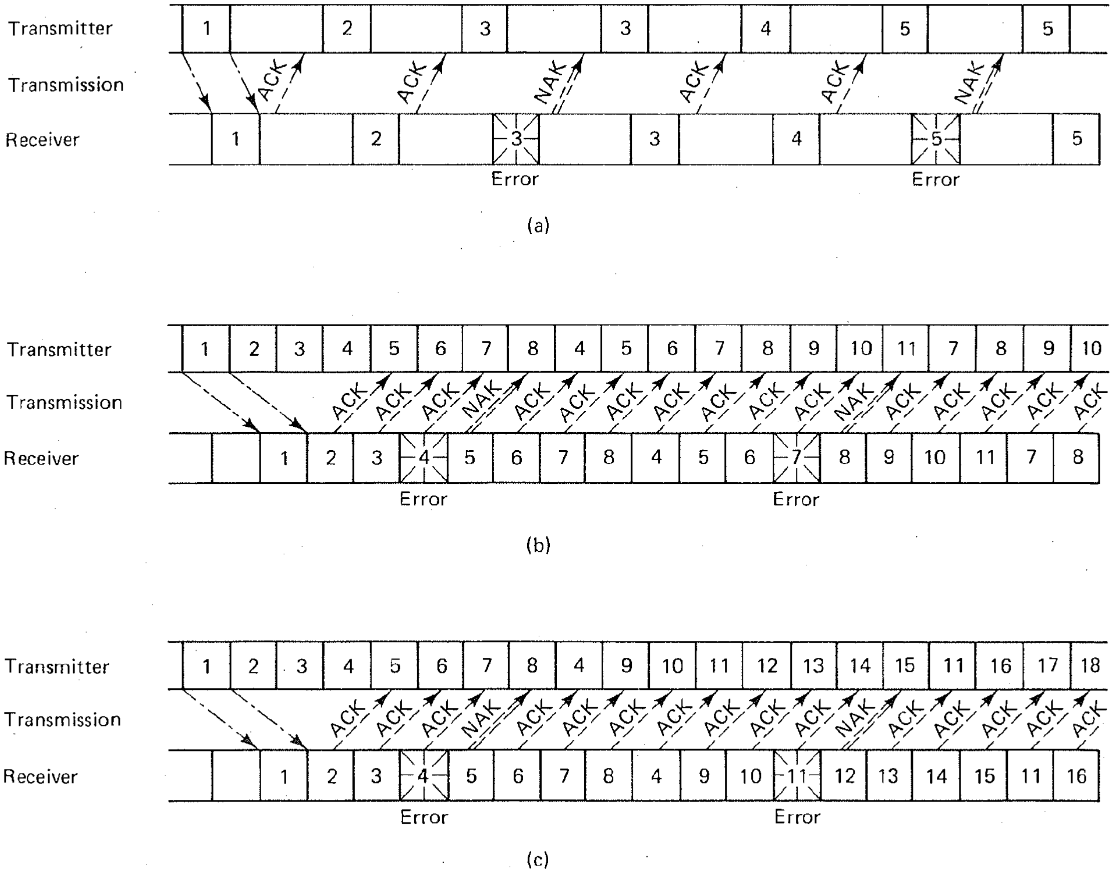


Figure 5.10 Automatic repeat request (ARQ). (a) Stop-and-wait ARQ (half-duplex). (b) Continuous ARQ with pullback (full-duplex). (c) Continuous ARQ with selective repeat (full-duplex).

message block before transmitting the next in the sequence. The second ARQ procedure, called *continuous ARQ with pullback*, is shown in Figure 5.10b. Here a full-duplex connection is necessary. Both terminals are transmitting simultaneously; the transmitter is sending message data and the receiver is sending acknowledgment data. Notice that a sequence number has to be assigned to each block of data. Also, the ACKs and NAKs need to reference such numbers, or else there needs to be a priori knowledge of the propagation delays so that the transmitter knows which messages are associated with which acknowledgments. In the example of Figure 5.10b there is a fixed separation of four blocks between the message being transmitted and the acknowledgment being simultaneously received. For example, when message 8 is being sent, a NAK corresponding to the corrupted message 4 is being received. In this ARQ procedure, the transmitter "pulls back" to the message in error and retransmits all message data, starting with the corrupted message. The final method, called *continuous ARQ with selective repeat*, is shown in Figure 5.10c. Here, as with the second ARQ procedure, a full-duplex connection is needed. However, in this procedure, only the corrupted message is repeated; then the transmitter continues the transmission sequence where it had left off instead of repeating any subsequent correctly received messages.

The choice of which ARQ procedure to choose is a trade-off between the requirements for efficient utilization of the communications resource and the need to provide full-duplex connectivity. The half-duplex connectivity required in Figure 5.10a is less costly than full-duplex; the associated inefficiency can be measured by the blank time slots. The more efficient utilization illustrated in Figures 5.10b and c requires the more costly full-duplex connectivity.

The major advantage of ARQ over forward error correction (FEC) is that error detection requires much simpler decoding equipment and much less redundancy than does error correction. Also, ARQ is adaptive in the sense that information is retransmitted only when errors occur. On the other hand, FEC may be desirable in place of, or in addition to, error detection, for any of the following reasons:

1. A reverse channel is not available or the delay with ARQ would be excessive.
2. The retransmission strategy is not conveniently implemented.
3. The expected number of errors, without corrections, would require excessive retransmissions.

5.3 STRUCTURED SEQUENCES

In Section 3.8 we considered digital signaling by means of $M = 2^k$ signal waveforms (M-ary signaling), where each waveform contains k bits of information. We saw that in the case of orthogonal M-ary signaling, we can decrease P_B by increasing M (expanding the bandwidth). Similarly, in Section 5.1 we showed that it is possible to decrease P_B by encoding k binary digits into one of M orthogonal codewords. The major disadvantage with such orthogonal coding techniques is the associated inefficient use of bandwidth. The required transmission bandwidth grows exponentially with k for an orthogonal set of $M = 2^k$ waveforms. In this and subsequent sections we abandon the need for antipodal or orthogonal properties and focus on a class of encoding procedures known as *parity-check codes.* Such channel coding procedures are classified as *structured sequences* because they represent methods of inserting structured redundancy into the source data so that the presence of errors can be detected or the errors corrected. Structured sequences are partitioned into two important subcategories as shown in Figure 5.1: *block coding* and *convolutional coding.* Block coding (primarily) is treated in this chapter, and convolutional coding is treated in Chapter 6. These techniques allow us to attain a P_B performance comparable to waveform encoding techniques but with lower bandwidth requirements. The codewords of these codes (structured sequences) are usually *nonorthogonal* [3].

5.3.1 Channel Models

5.3.1.1 Discrete Memoryless Channel

A *discrete memoryless channel* (DMC) is characterized by a discrete input alphabet, a discrete output alphabet, and a set of conditional probabilities, $P(j|i)$ $(1 \leq i \leq M,\ 1 \leq j \leq Q)$, where i represents a modulator M-ary input symbol, j represents a demodulator Q-ary output symbol, and $P(j|i)$ is the probability of receiving j given that i was transmitted. Each output symbol of the channel depends only on the corresponding input, so that for a given input sequence $\mathbf{U} = u_1, u_2, \ldots, u_m, \ldots, u_N$ the conditional probability of a corresponding output sequence $\mathbf{Z} = z_1, z_2, \ldots, z_m, \ldots, z_N$ may be expressed as

$$P(\mathbf{Z}|\mathbf{U}) = \prod_{m=1}^{N} P(z_m|u_m) \tag{5.12}$$

In the event that the channel *has memory* (i.e., noise or fading that occurs in bursts), the conditional probability of the sequence $\mathbf{Z}$ would need to be expressed as the *joint* probability of all the elements of the sequence. Equation (5.12) expresses the *memoryless* condition of the channel. Since the channel noise in a memoryless channel is defined to affect each symbol independently of all the other symbols, the conditional probability of $\mathbf{Z}$ is seen as the product of the independent element probabilities.

5.3.1.2 Binary Symmetric Channel

A *binary symmetric channel* (BSC) is a special case of a DMC; the input and output alphabet sets consist of the binary elements (0 and 1). The conditional probabilities are symmetric:

$$\begin{aligned} P(0|1) &= P(1|0) = p \\ P(1|1) &= P(0|0) = 1 - p \end{aligned} \tag{5.13}$$

Equation (5.13) states the channel *transition probabilities*. That is, given that a channel symbol was transmitted, the probability that it is received in error is p (related to the symbol energy), and the probability that it is received correctly is $(1 - p)$. Since the demodulator output consists of the discrete elements 0 and 1, the demodulator is said to make a firm or *hard decision* on each symbol. A commonly used code system consists of BPSK modulated coded data, hard decision demodulated. Then the channel symbol error probability is found using the methods discussed in Section 3.7.1 and Equation (3.84) to be

$$p = Q\left(\sqrt{\frac{2E_c}{N_0}}\right)$$

where E_c/N_0 is the channel symbol energy per noise density, and $Q(x)$ is defined in Equation (2.42).

When such hard decisions are used in a binary coded system, the demodulator feeds the two-valued *code symbols* or *channel bits* to the decoder. Since the decoder then operates on the hard decisions made by the demodulator, decoding with a BSC channel is called *hard-decision decoding*.

5.3.1.3 Gaussian Channel

We can generalize our definition of the DMC to channels with alphabets that are not discrete. An example is the *Gaussian channel* with a discrete input alphabet and a continuous output alphabet over the range $(-\infty, \infty)$. The channel adds noise to the symbols. Since the noise is a Gaussian random variable, with zero mean and variance σ^2, the resulting probability density function (pdf) of the received random variable z, conditioned on the symbol u_k (the likelihood of u_k), can be written

$$p(z|u_k) = \frac{1}{\sigma\sqrt{2\pi}} \exp\left[\frac{-(z - u_k)^2}{2\sigma^2}\right] \tag{5.14}$$

for all z, where $k = 1, 2, \ldots, M$. For this case, *memoryless* has the same meaning as it does in Section 5.3.1.1, and Equation (5.12) can be used to obtain the conditional probability for the sequence, **Z**.

When the demodulator output consists of a continuous alphabet or its quantized approximation (with greater than two quantization levels), the demodulator is said to make *soft decisions*. In the case of a coded system, the demodulator feeds such quantized code symbols to the decoder. Since the decoder then operates on the soft decisions made by the demodulator, decoding with a Gaussian channel is called *soft-decision decoding*.

In the case of a hard-decision channel, we are able to characterize the detection process with a channel symbol error probability. However, in the case of a soft-decision channel, the detector makes the kind of decisions (soft decisions) that cannot be labeled as correct or incorrect. Thus, since there are no firm decisions, there cannot be a probability of making an error; the detector can only formulate a family of conditional probabilities or likelihoods of the different symbol types.

It is possible to design decoders using soft decisions, but block code soft-decision decorders are substantially more complex than hard-decision decoders; therefore, block codes are usually implemented with hard-decision decoders. For convolutional codes, both hard- and soft-decision implementations are equally popular. In this chapter we consider that the channel is a binary symmetric channel (BSC), and hence the decoder employs hard decisions. In Chapter 6 we further discuss channel models, as well as hard- versus soft-decision decoding for convolutional codes.

5.3.2 Code Rate and Redundancy

In the case of block codes, the source data are segmented into blocks of k data bits, also called information bits or message bits; each block can represent any one of 2^k distinct messages. The encoder transforms each k-bit data block into a larger block of n bits, called code bits or channel symbols. The $(n - k)$ bits, which the encoder adds to each data block, are called *redundant bits*, *parity bits*, or *check bits*; they carry no new information. The code is referred to as an (n, k) code. The ratio of redundant bits to data bits, $(n - k)/k$, within a block is called the *redundancy* of the code, and the ratio of data bits to total bits, k/n, is called the *code rate*. The code rate can be thought of as the portion of a code bit that constitutes information. For example, in a rate $\frac{1}{2}$ code, each code bit carries $\frac{1}{2}$ bit of information.

In this chapter and Chapter 6 we consider those coding techniques that provide redundancy by increasing the required transmission bandwidth. For example, an error control technique that employs a rate 1/2 code (100% redundancy) will require double the bandwidth of an uncoded system. However, if a rate 3/4 code is used, the redundancy is 33% and the bandwidth expansion is only 4/3. In Chapter 7 we consider modulation/coding techniques for bandlimited channels where complexity, instead of bandwidth, is traded for error performance improvement.

5.3.3 Parity-Check Codes

5.3.3.1 Single-Parity-Check Code

Parity-check codes use linear sums of the information bits, called *parity symbols* or *parity bits*, for error detection or correction. A single-parity check code is constructed by adding a single-parity bit to a block of data bits. The parity bit takes on the value of one or zero as needed to ensure that the summation of all the bits in the codeword yields an even (or odd) result. The summation operation is performed using modulo-2 arithmetic (exclusive-or logic), as described in Section 2.12.3. If the added parity is designed to yield an even result, the method is termed *even parity*, and if designed to yield an odd result, it is termed *odd*

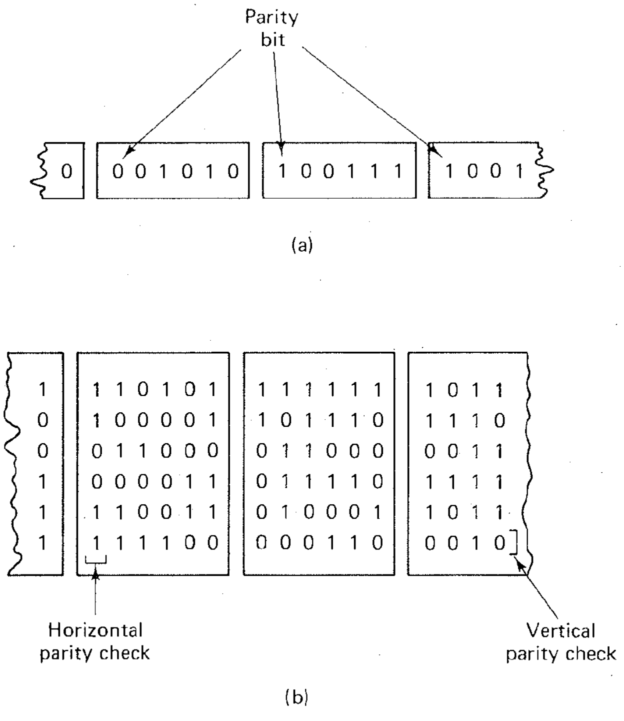


Figure 5.11 Parity checks for serial and parallel transmission. (a) Serial transmission. (b) Parallel transmission.

parity. Figure 5.11a illustrates a serial data transmission (the rightmost bit is the earliest bit). A single-parity bit is added (the leftmost bit in each block) to yield even parity.

At the receiving terminal, the decoding procedure consists of testing that the modulo-2 sum of the codeword bits yields a zero result (even parity). If the result is found to be one instead of zero, the codeword is known to contain errors. The rate of the code can be expressed as $k/(k + 1)$. Do you suppose the decoder can automatically *correct* a digit that is received in error? No, it cannot. It can only *detect* the presence of an odd number of bit errors (if an even number of bits are inverted, the parity test will appear correct; this represents the case of an *undetected error*). Assuming that all bit errors are equally likely and occur independently, we can write the probability of j errors occurring in a block of n symbols as

$$P(j, n) = \binom{n}{j} p^j (1 - p)^{n-j} \tag{5.15}$$

where p is the probability that a *channel symbol* is received in error, and where

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

is the number of various ways in which j bits out of n may be in error. Thus for a single-parity error-detection code, the probability of an undetected error, P_{nd}, within a block of n bits is computed, as follows:

$$P_{\text{nd}} = \sum_{j=1}^{\substack{n/2 \text{ (for } n \text{ even)} \\ (n-1)/2 \text{ (for } n \text{ odd)}}} \binom{n}{2j} p^{2j}(1 - p)^{n-2j} \tag{5.16}$$

Example 5.1 Even-Parity Code

Configure a (4, 3) even-parity error-detection code such that the parity symbol appears as the leftmost symbol of the codeword. Which error patterns can the code detect? Compute thc probability of an undetected message error, assuming that all symbol errors are independent events and that the probability of a channel symbol error is $p = 10^{-3}$.

Solution

Message	Parity	Codeword
000	0	0 000
100	1	1 100
010	1	1 010
110	0	0 110
001	1	1 001
101	0	0 101
011	0	0 011
111	1	1 111

parity message

The code is capable of detecting all single- and triple-error patterns. The probability of an undetected error is equal to the probability that two or four errors occur anywhere in a codeword.

$$\begin{aligned} P_{\text{nd}} &= \binom{4}{2} p^2(1 - p)^2 + \binom{4}{4} p^4 \\ &= 6p^2(1 - p)^2 + p^4 \\ &= 6p^2 - 12p^3 + 7p^4 \\ &= 6(10^{-3})^2 - 12(10^{-3})^3 + 7(10^{-3})^4 \simeq 6 \times 10^{-6} \end{aligned}$$

5.3.3.2 Rectangular Code

A *rectangular code*, also called a *product code*, can be thought of as a parallel data transmission, depicted in Figure 5.11b. First we form a rectangle of message bits comprised of M rows and N columns; then a horizontal parity check is appended to each row and a vertical parity check is appended to each column, resulting in an augmented array of dimensions $(M + 1) \times (N + 1)$. The rate of the rectangular code, k/n, can then be written as

$$\frac{k}{n} = \frac{MN}{(M + 1)(N + 1)} \tag{5.17}$$

How much more powerful is the rectangular code than the single-parity code, which is only capable of error detection? Notice that any single bit error will cause a parity check failure in one of the array columns *and* in one of the array rows. Therefore, the rectangular code can correct a single error pattern since the error is uniquely located at the intersection of the error-detecting row and the error-detecting column. For the example shown in Figure 5.11b, the array dimensions are $M = N = 5$; therefore, the figure depicts a (36, 25) code that can correct a single error located anywhere in the 36 bit positions. For an error-correcting block code, we compute the probability that the decoded block has an uncorrected error by accounting for all the ways in which a *message error* can be made. Starting with the probability of j errors in a block of n symbols, expressed in Equation (5.15), we can write the probability of a message error, also called a *block error* or *word error*, P_M, for a code that can correct all t and fewer error patterns:

$$P_M = \sum_{j=t+1}^{n} \binom{n}{j} p^j (1 - p)^{n-j} \tag{5.18}$$

where p is the probability that a *channel symbol* is received in error. For the example in Figure 5.11b, the code can correct all single error patterns ($t = 1$) within the rectangular block of $n = 36$ bits. Hence the summation in Equation (5.18) starts with $j = 2$:

$$P_M = \sum_{j=2}^{36} \binom{36}{j} p^j (1 - p)^{36-j} \tag{5.19}$$

When p is reasonably small, the first term in the summation is the dominant one; we can therefore write for this (36, 25) rectangular code example

$$P_M \approx \binom{36}{2} p^2 (1 - p)^{34}$$

The *bit error probability*, P_B, depends on the particular code and decoder. An approximation for P_B is given in Section 5.5.3.

5.3.4 Coding Gain

Figure 5.12 illustrates the probability of bit error, P_B, versus E_b/N_0 for coherent binary PSK modulation in combination with examples of various (n, k) codes over a Gaussian channel. The (1, 1) curve illustrates the uncoded PSK performance, while the (24, 12) and (127, 92) curves illustrate coded PSK performance using block codes with $(n - k) = 12$ parity bits and 35 parity bits, respectively. From Figure 3.24 we know in which direction the waterfall-like curves move, corresponding to P_B performance improvement. Look at the various curves in Figure 5.12. Can you explain why the coded curves (to which we attribute P_B performance improvement) appear to be moving in the wrong direction when compared with the uncoded curve? Where does the strength of the code manifest itself? The curves in Figure 5.12 indicate that the strength of a code is seen only after an E_b/N_0 threshold has been exceeded (approximately 5.5 dB in this example). For values of E_b/N_0 less than the threshold, the coding manifests itself only as *over-*

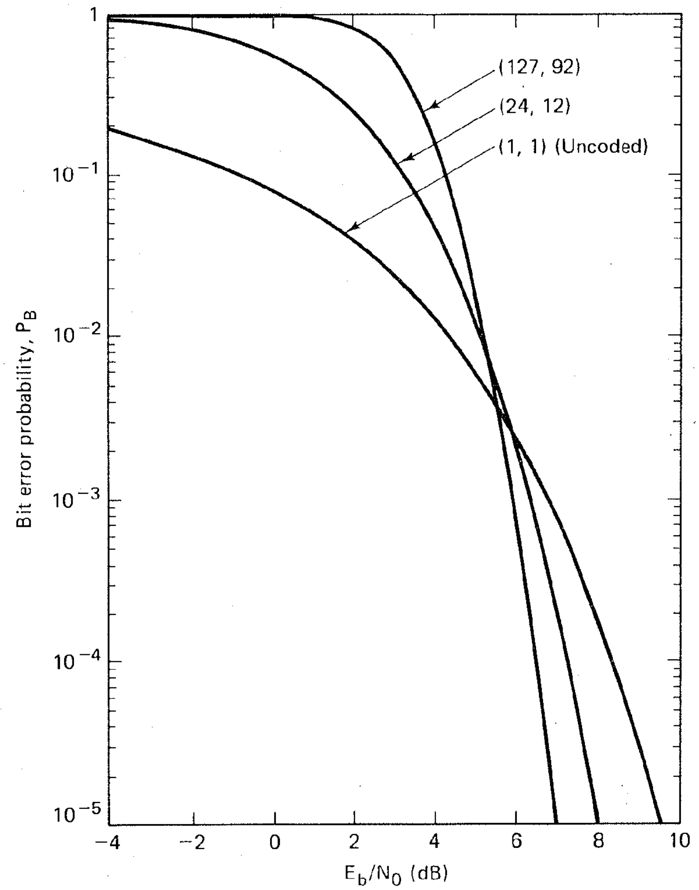


Figure 5.12 Coded versus uncoded bit error performance for coherent PSK with various (n, k) codes.

head bits resulting in *reduced energy per bit*, compared to the uncoded case; before the threshold is exceeded, the redundant bits are simply "excess baggage" without the ability to improve performance. Once the threshold is exceeded, the performance improvement of the code more than compensates for the reduction in energy per coded bit. Therefore, in Figure 5.12, once the threshold value of $E_b/N_0 = 5.5$ dB is exceeded, the relative positions of the curves reverse themselves compared to their positions at less-than-threshold E_b/N_0. *Coding gain* is defined as the reduction, expressed in decibels, in the required E_b/N_0 to achieve a specified error performance of an error-correcting coded system over an uncoded one with the same modulation. For example, in Figure 5.12, for $P_B = 10^{-5}$, the (24, 12) code has a coding gain of about 1.5 dB.

Example 5.2 Coded versus Uncoded Performance

Compare the message error probability for a communications link with and without the use of error-correction coding. Assume that the uncoded transmission characteristics are: BPSK modulation, Gaussian noise, $S/N_0 = 43{,}776$, data rate $R = 4800$ bits/s. For the coded case, also assume the use of a (15, 11) error-correcting code that is capable of correcting any single-error pattern within a block of 15 bits. Consider that the demodulator makes hard decisions and thus feeds the demodulated code bits directly to the decoder, which in turn outputs an estimate of the original message.

Solution

Following Equation (3.84), let $p_u = Q\sqrt{2E_b/N_0}$ and $p_c = Q\sqrt{2E_c/N_0}$ be the uncoded and coded channel symbol error probabilities, respectively, where E_b/N_0 is uncoded bit energy per noise spectral density and E_c/N_0 is the coded bit energy per noise spectral density.

Without coding

$$\frac{E_b}{N_0} = \frac{S}{RN_0} = 9.12 \text{ (9.6 dB)}$$

$$p_u = Q\left(\sqrt{\frac{2E_b}{N_0}}\right) = Q(\sqrt{18.24}) = 1.02 \times 10^{-5} \tag{5.20}$$

where the following approximation of $Q(x)$ from Equation (2.43) was used:

$$Q(x) \simeq \frac{1}{x\sqrt{2\pi}} \exp\left(\frac{-x^2}{2}\right) \qquad \text{for } x > 3$$

The probability that the uncoded message block, P_M^u, will be received in error is 1 minus the product of the probabilities that each bit will be detected correctly. Thus

$$\begin{aligned} P_M^u &= 1 - (1 - p_u)^k \\ &= 1 - \underbrace{(1 - p_u)^{11}}_{\substack{\text{probability that all}\\ \text{11 bits in uncoded}\\ \text{block are correct}}} \qquad \underbrace{= 1.12 \times 10^{-4}}_{\substack{\text{probability that at}\\ \text{least 1 bit out of}\\ \text{11 is in error}}} \end{aligned} \tag{5.21}$$

With coding:

The channel symbol rate, sometimes called the coded bit rate, R_c is 15/11 times the data bit rate.

$$R_c = 4800 \times \tfrac{15}{11} \simeq 6545 \text{ bps}$$

$$\frac{E_c}{N_0} = \frac{S}{R_c N_0} = 6.688 \text{ (8.25 dB)}$$

The E_c/N_0 for each code bit is less than that for the uncoded bit because the channel bit rate has increased but the transmitter power is assumed to be fixed.

$$p_c = Q\left(\sqrt{\frac{2E_c}{N_0}}\right) = Q(\sqrt{13.38}) = 1.36 \times 10^{-4} \tag{5.22}$$

It can be seen by comparing the results of Equation (5.20) with (5.22) that the channel bit error probability has degraded. More bits must be detected during the same time interval, and with the same available power; the performance improvement due to the coding *is not yet apparent.* We now compute the coded message error rate, P_M^c, using Equation (5.18).

$$P_M^c = \sum_{j=2}^{n=15} \binom{15}{j} (p_c)^j (1 - p_c)^{15-j}$$

The summation is started with $j = 2$ since the code corrects all single errors within a block of $n = 15$ bits. A good approximation is obtained by using only the first term of the summation. For p_c we use the value calculated in Equation (5.22):

$$P_M^c = \binom{15}{2} (p_c)^2 (1 - p_c)^{13} = 1.94 \times 10^{-6} \qquad (5.23)$$

By comparing the results of Equation (5.21) with (5.23), it is seen that the probability of message error has improved by a factor of 58 due to the error-correcting code used in this example.

5.4 LINEAR BLOCK CODES

Linear block codes (such as the one in Example 5.2) are a class of parity check codes that can be characterized by the (n, k) notation described earlier. The encoder transforms a block of k message digits (a message vector) into a longer block of n codeword digits (a code vector), constructed from a given alphabet of elements. When the alphabet consists of two elements (0 and 1), the code is a binary code comprised of binary digits (bits). Our discussion of linear block codes is restricted to binary codes, unless otherwise noted.

The k-bit messages form 2^k distinct message sequences referred to as *k-tuples* (sequences of k digits). The n-bit blocks can form as many as 2^n distinct sequences, referred to as *n-tuples.* The encoding procedure assigns to each of the 2^k message k-tuples *one* of the 2^n n-tuples. A block code represents a one-to-one assignment, whereby the 2^k message k-tuples are *uniquely* mapped into a new set of 2^k codeword n-tuples; the mapping can be accomplished via a look-up table. For *linear codes,* the mapping transformation is, of course, *linear.*

5.4.1 Vector Spaces

The set of all binary n-tuples, V_n, is called a *vector space* over the binary field of two elements (0 and 1). The binary field has two operations, addition and multiplication, such that the results of all operations are in the same set of two elements. The arithmetic operations of addition and multiplication are defined by the conventions of the algebraic field [4]. For example, in a binary field, the rules of addition and multiplication are as follows:

Addition	Multiplication
$0 \oplus 0 = 0$	$0 \cdot 0 = 0$
$0 \oplus 1 = 1$	$0 \cdot 1 = 0$
$1 \oplus 0 = 1$	$1 \cdot 0 = 0$
$1 \oplus 1 = 0$	$1 \cdot 1 = 1$

The addition operation, designated with the symbol $\oplus$, is the same modulo-2 operation described in Section 2.12.3.

5.4.2 Vector Subspaces

A subset S of the vector space V_n is called a *subspace* if the following two conditions are met:

1. The all-zeros vector is in S.
2. The sum of any two vectors in S is also in S (known as the *closure property*).

These properties are fundamental for the algebraic characterization of *linear block codes*. Suppose that $\mathbf{V}_i$ and $\mathbf{V}_j$ are two codewords (also called code vectors) in an

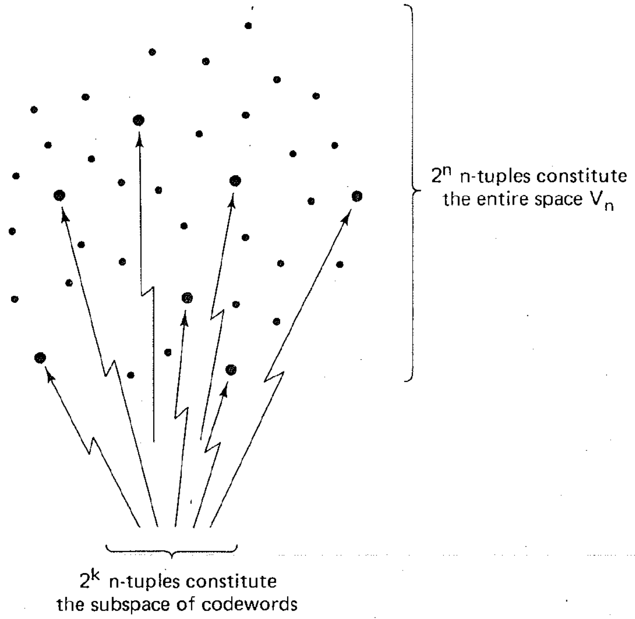

2^n n-tuples constitute the entire space V_n
2^k n-tuples constitute the subspace of codewords

Figure 5.13 Linear block-code structure.

(n, k) binary block code. The code is said to be *linear* if, and only if, $(\mathbf{V}_i \oplus \mathbf{V}_j)$ is also a code vector. A linear block code, then, is one in which vectors outside the subspace cannot be created by the addition of legitimate code vectors (members of the subspace).

For example, the vector space V_4 is totally populated by the following 2^4 = sixteen 4-tuples:

```
0000  0001  0010  0011  0100  0101  0110  0111

1000  1001  1010  1011  1100  1101  1110  1111
```

An example of a subset of V_4 that forms a subspace is

```
0000  0101  1010  1111
```

It is easy to verify that the addition of any two vectors in the subspace can only yield one of the other members of the subspace. A set of 2^k n-tuples is called a *linear block code* if, and only if, it is a subspace of the vector space V_n of all n-tuples. Figure 5.13 illustrates, with a simple geometric analogy, the structure behind linear block codes. We can imagine the vector space V_n comprised of 2^n n-tuples. Within this vector space there exists a subset of 2^k n-tuples comprising a subspace. These 2^k vectors or points, shown "sprinkled" among the more numerous 2^n points, represent the legitimate or allowable codeword assignments. A message is encoded into one of the 2^k allowable code vectors and then transmitted. Because of noise in the channel, a perturbed version of the code vector (one of the other 2^n vectors in the n-tuple space) may be received. If the perturbed vector is not too unlike (not too distant from) the valid code vector, the decoder can decode the message correctly. The basic goals in choosing a particular code, similar to the goals in selecting a set of modulation waveforms, can be stated in the context of Figure 5.13 as follows:

1. We want to strive for coding efficiency by packing the V_n space with as many code vectors as possible. This is tantamount to saying that we only want to expend a *small amount of redundancy* (excess bandwidth).
2. We want the code vectors to be as *far apart from one another* as possible, so that even if the vectors experience some corruption during transmission, they may still be correctly decoded, with a high probability.

5.4.3 A (6, 3) Linear Block Code Example

Examine the following coding assignment that describes a (6, 3) code. There are $2^k = 2^3 = 8$ message vectors, and therefore eight code vectors. There are $2^n = 2^6$ = sixty-four 6-tuples in the V_6 vector space.

Message vector	Code vector
0 0 0	0 0 0 0 0 0
1 0 0	1 1 0 1 0 0
0 1 0	0 1 1 0 1 0
1 1 0	1 0 1 1 1 0
0 0 1	1 0 1 0 0 1
1 0 1	0 1 1 1 0 1
0 1 1	1 1 0 0 1 1
1 1 1	0 0 0 1 1 1

It is easy to check that the eight code vectors shown above form a subspace of V_6 (the all-zcros vector is present, and the sum of any two code vectors yields another code vector member of the subspace). Therefore, these code vectors represent a *linear block code*, as defined in Section 5.4.2.

5.4.4 Generator Matrix

If k is large, a *table look-up* implementation of the encoder becomes prohibitive. For a (127, 92) code there are 2^{92} or approximately 5×10^{27} code vectors. If the encoding procedure consists of a simple look-up table, imagine the size of the memory necessary to contain such a large number of code vectors. Fortunately, it is possible to reduce complexity by generating the required code vectors as needed, instead of storing them.

Since a set of code vectors that forms a linear block code is a k-dimensional subspace of the n-dimensional binary vector space ($k < n$), it is always possible to find a set of n-tuples, fewer than 2^k, that can generate all the 2^k member vectors of the subspace. The generating set of vectors is said to *span* the subspace. The smallest *linearly independent* set that spans the subspace is called a *basis* of the subspace, and the number of vectors in this basis set is the dimension of the subspace. Any basis set of k linearly independent n-tuples $\mathbf{V}_1, \mathbf{V}_2, \ldots, \mathbf{V}_k$ can be used to generate the required linear block code vectors, since each code vector is a linear combination of $\mathbf{V}_1, \mathbf{V}_2, \ldots, \mathbf{V}_k$. That is, each of the set of 2^k code vectors $\mathbf{U}$ can be described by

$$\mathbf{U} = m_1\mathbf{V}_1 + m_2\mathbf{V}_2 + \cdots + m_k\mathbf{V}_k$$

where m_i = (0 or 1) are the message digits and $i = 1, \ldots, k$.

In general, we can define a *generator matrix* by the following $k \times n$ array:

$$\mathbf{G} = \begin{bmatrix} \mathbf{V}_1 \\ \mathbf{V}_2 \\ \vdots \\ \mathbf{V}_k \end{bmatrix} = \begin{bmatrix} v_{11} & v_{12} & \cdots & v_{1n} \\ v_{21} & v_{22} & \cdots & v_{2n} \\ \vdots & & & \\ v_{k1} & v_{k2} & \cdots & v_{kn} \end{bmatrix} \tag{5.24}$$

Code vectors, by convention, are usually designated as row vectors. Thus, the message **m**, a sequence of k message bits, is shown below as a row vector ($1 \times k$ matrix having one row and k columns).

$$\mathbf{m} = m_1, m_2, \ldots, m_k$$

The generation of the code vector, **U**, is written in matrix notation as the product of **m** and **G**, as follows:

$$\mathbf{U} = \mathbf{mG} \tag{5.25}$$

where, in general, the matrix multiplication $\mathbf{C} = \mathbf{AB}$ is performed in the usual way by using the rule

$$c_{ij} = \sum_{k}^{n} a_{ik} b_{kj} \qquad i = 1, \ldots, l \quad j = 1, \ldots, m$$

where **A** is an $l \times n$ matrix, **B** is an $n \times m$ matrix, and the result **C** is an $l \times m$ matrix. For the example introduced in the preceding section, we can fashion a generator matrix as follows:

$$\mathbf{G} = \begin{bmatrix} \mathbf{V}_1 \\ \mathbf{V}_2 \\ \mathbf{V}_3 \end{bmatrix} = \begin{bmatrix} 1 & 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 & 0 & 1 \end{bmatrix} \tag{5.26}$$

where $\mathbf{V}_1$, $\mathbf{V}_2$, and $\mathbf{V}_3$ are three *linearly independent vectors* (a subset of the eight code vectors) that can generate all the code vectors. Notice that the sum of any two generating vectors does not yield any of the other generating vectors (opposite of closure). Let us generate the code vector for the message vector 1 1 0, using the generator matrix of Equation (5.26).

$$\mathbf{U} = [1 \quad 1 \quad 0] \begin{bmatrix} \mathbf{V}_1 \\ \mathbf{V}_2 \\ \mathbf{V}_3 \end{bmatrix} = 1 \cdot \mathbf{V}_1 + 1 \cdot \mathbf{V}_2 + 0 \cdot \mathbf{V}_3$$

$$= 1\,1\,0\,1\,0\,0 + 0\,1\,1\,0\,1\,0 + 0\,0\,0\,0\,0\,0$$

$$= 1\;0\;1\;1\;1\;0 \quad \text{(code vector for the message vector 1 1 0)}$$

Thus the code vector corresponding to a message vector is a linear combination of the rows of **G**. Since the code is totally defined by **G**, the encoder need only store the k rows of G instead of the total 2^k vectors of the code. For this example notice that the generator array of dimension 3×6 replaces the original code vector array of dimension 8×6, representing a reduction in system complexity.

5.4.5 Systematic Linear Block Codes

A systematic (n, k) linear block code is a mapping from a k-dimensional message vector to an n-dimensional code vector in such a way that part of the sequence generated coincides with the k message digits. The remaining $(n - k)$ digits are parity digits. A systematic linear block code will have a generator matrix of the form

$$\mathbf{G} = \left[\begin{array}{c|c} \mathbf{P} & \mathbf{I}_k \end{array} \right]$$

$$= \begin{bmatrix} p_{11} & p_{12} & \cdots & p_{1,(n-k)} & 1 & 0 & \cdots & 0 \\ p_{21} & p_{22} & \cdots & p_{2,(n-k)} & 0 & 1 & \cdots & 0 \\ \vdots & & & & & & & \vdots \\ p_{k1} & p_{k2} & \cdots & p_{k,(n-k)} & 0 & 0 & \cdots & 1 \end{bmatrix} \tag{5.27}$$

where **P** is the parity array portion of the generator matrix, $p_{ij} = (0 \text{ or } 1)$, and $\mathbf{I}_k$ is the $k \times k$ identity matrix (ones on the main diagonal and zeros elsewhere). Notice that with this systematic generator, the encoding complexity is further reduced since it is not necessary to store the identity matrix portion of the array. By combining Equations (5.25) and (5.27), each code vector is expressed as follows:

$$u_1, u_2, \ldots, u_n = [m_1, m_2, \ldots, m_k] \times \begin{bmatrix} p_{11} & p_{12} & \cdots & p_{1,(n-k)} & 1 & 0 & \cdots & 0 \\ p_{21} & p_{22} & \cdots & p_{2,(n-k)} & 0 & 1 & \cdots & 0 \\ \vdots & & & & & & & \vdots \\ p_{k1} & p_{k2} & \cdots & p_{k,(n-k)} & 0 & 0 & \cdots & 1 \end{bmatrix}$$

where

$$\begin{aligned} u_i &= m_1 p_{1i} + m_2 p_{2i} + \cdots + m_k p_{ki} && \text{for } i = 1, \ldots, (n-k) \\ &= m_{i-n+k} && \text{for } i = (n-k+1), \ldots, n \end{aligned}$$

Given the message k-tuple

$$\mathbf{m} = m_1, m_2, \ldots, m_k$$

and the general code vector n-tuple

$$\mathbf{U} = u_1, u_2, \ldots, u_n$$

the systematic code vector can be expressed as

$$\mathbf{U} = \underbrace{p_1, p_2, \ldots, p_{n-k},}_{\text{parity bits}} \underbrace{m_1, m_2, \ldots, m_k}_{\text{message bits}} \tag{5.28}$$

where

$$\begin{aligned} p_1 &= m_1 p_{11} + m_2 p_{21} + \cdots + m_k p_{k1} \\ p_2 &= m_1 p_{12} + m_2 p_{22} + \cdots + m_k p_{k2} \\ p_{n-k} &= m_1 p_{1,(n-k)} + m_2 p_{2,(n-k)} + \cdots + m_k p_{k,(n-k)} \end{aligned} \tag{5.29}$$

Systematic code vectors are sometimes written so that the message bits occupy the left-hand portion of the code vector and the parity bits occupy the right-hand portion. This reordering has no effect on the error detection or error correction properties of the code, and will not be considered further.

For the (6, 3) code example in Section 5.4.3, the code vectors are described as follows:

$$\mathbf{U} = [m_1, m_2, m_3] \left[\begin{array}{ccc:ccc} 1 & 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 & 0 & 1 \end{array}\right] \tag{5.30}$$

$$\qquad\qquad\qquad\qquad \underbrace{\hphantom{1\ 0\ 1}}_{\mathbf{P}} \quad \underbrace{\hphantom{0\ 0\ 1}}_{\mathbf{I}_3}$$

$$\mathbf{U} = \underbrace{m_1 + m_3}_{u_1}, \underbrace{m_1 + m_2}_{u_2}, \underbrace{m_2 + m_3}_{u_3}, \underbrace{m_1}_{u_4}, \underbrace{m_2}_{u_5}, \underbrace{m_3}_{u_6}, \tag{5.31}$$

Equation (5.31) gives us some insight regarding the structure of linear block codes. We see that the redundant digits are produced in a variety of ways. The first parity bit is the sum of the first and third message bits; the second parity bit is the sum of the first and second message bits, and the third parity bit is the sum of the second and third message bits. Intuition tells us that such structure, compared to single-parity checks or simple digit-repeat procedures, may provide greater ability to detect and correct errors.

5.4.6 Parity-Check Matrix

Let us define a matrix, **H**, called the *parity-check matrix*, that will enable us to decode the received vectors. For each $(k \times n)$ generator matrix, **G**, there exists an $(n - k) \times n$ matrix, **H**, such that the rows of **G** are orthogonal to the rows of **H**; that is $\mathbf{GH}^T = \mathbf{0}$, where $\mathbf{H}^T$ is the *transpose* of **H**, and **0** is a $k \times (n - k)$ all-zeros matrix. $\mathbf{H}^T$ is an $n \times (n - k)$ matrix whose rows are the columns of **H** and whose columns are the rows of **H**. To fulfill the orthogonality requirements, the components of the **H** matrix are written

$$\mathbf{H} = [\mathbf{I}_{n-k} \;\vdots\; \mathbf{P}^T] \tag{5.32}$$

Hence, the $\mathbf{H}^T$ matrix is written

$$\mathbf{H}^T = \left[\begin{array}{c} \mathbf{I}_{n-k} \\ \hdashline \mathbf{P} \end{array}\right] \tag{5.33a}$$

$$= \begin{bmatrix} 1 & 0 & \cdots & 0 \\ 0 & 1 & \cdots & 0 \\ \vdots & & & \\ 0 & 0 & \cdots & 1 \\ p_{11} & p_{12} & \cdots & p_{1,(n-k)} \\ p_{21} & p_{22} & \cdots & p_{2,(n-k)} \\ \vdots & & & \\ p_{k1} & p_{k2} & \cdots & p_{k,(n-k)} \end{bmatrix} \tag{5.33b}$$

1. No column of **H** can be all zeros, or else an error in the corresponding code vector position would not affect the syndrome and would be undetectable.
2. All columns of **H** must be unique. If two columns of **H** were identical, errors in these two corresponding code vector positions would be indistinguishable.

Example 5.3 Syndrome Test

Suppose that code vector $\mathbf{U} = 1\ 0\ 1\ 1\ 1\ 0$ from the example in Section 5.4.3 is transmitted and the vector $\mathbf{r} = 0\ 0\ 1\ 1\ 1\ 0$ is received; that is, the leftmost bit is received in error. Find the syndrome vector value $\mathbf{S} = \mathbf{r}\mathbf{H}^T$ and verify that it is equal to $\mathbf{e}\mathbf{H}^T$.

Solution

$$\mathbf{S} = \mathbf{r}\mathbf{H}^T$$

$$= [0\quad 0\quad 1\quad 1\quad 1\quad 0]\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 1 & 0 \\ 0 & 1 & 1 \\ 1 & 0 & 1 \end{bmatrix}$$

$$= [1,\quad 1 + 1,\quad 1 + 1] = [1\quad 0\quad 0] \qquad \text{(syndrome of corrupted code vector)}$$

Next, we verify that the syndrome of the corrupted code vector is the same as the syndrome of the error pattern that caused the error.

$$\mathbf{S} = \mathbf{e}\mathbf{H}^T = [1\quad 0\quad 0\quad 0\quad 0\quad 0]\mathbf{H}^T = [1\quad 0\quad 0] \qquad \text{(syndrome of error pattern)}$$

5.4.8 Error Correction

We have detected a single error and have shown that the syndrome test performed on either the corrupted code vector, or on the error pattern that caused it, yields the same syndrome. This should be a clue that we not only can detect the error, but since there is a one-to-one correspondence between correctable error patterns and syndromes, we can correct such error patterns. Let us arrange the 2^n n-tuples that represent possible received vectors in an array, called the *standard array*, such that the first row contains all the code vectors, starting with the all-zeros vector, and the first column contains all the correctable error patterns. Recall from the basic properties of linear codes (see Section 5.4.2) that the all-zeros vector must be a member of the codeword set. Each row, called a *coset*, consists of an error pattern in the first column, called the *coset leader*, followed by the code vectors perturbed by that error pattern. The standard array format for an (n, k) code is as follows:

$$\begin{array}{lllllllll}
\mathbf{U}_1 & \mathbf{U}_2 & \cdots & \mathbf{U}_i & \cdots & \mathbf{U}_{2^k} \\
\mathbf{e}_2 & \mathbf{U}_2 + \mathbf{e}_2 & \cdots & \mathbf{U}_i + \mathbf{e}_2 & \cdots & \mathbf{U}_{2^k} + \mathbf{e}_2 \\
\mathbf{e}_3 & \mathbf{U}_2 + \mathbf{e}_3 & \cdots & \mathbf{U}_i + \mathbf{e}_3 & \cdots & \mathbf{U}_{2^k} + \mathbf{e}_3 \\
\vdots & \vdots & & \vdots & & \\
\mathbf{e}_j & \mathbf{U}_2 + \mathbf{e}_j & \cdots & \mathbf{U}_i + \mathbf{e}_j & \cdots & \mathbf{U}_{2^k} + \mathbf{e}_j \\
\vdots & \vdots & & \vdots & & \\
\mathbf{e}_{2^{n-k}} & \mathbf{U}_2 + \mathbf{e}_{2^{n-k}} & \cdots & \mathbf{U}_i + \mathbf{e}_{2^{n-k}} & \cdots & \mathbf{U}_{2^k} + \mathbf{e}_{2^{n-k}}
\end{array} \qquad (5.38)$$

The array contains all 2^n n-tuples in the space V_n (each n-tuple appears in *only one* location). Each coset consists of 2^k n-tuples. Therefore, there are $(2^n/2^k) = 2^{n-k}$ cosets. Suppose that a code vector $\mathbf{U}_i$ is transmitted over a noisy channel. If the error pattern caused by the channel is a coset leader, the received vector will be decoded correctly into the transmitted code vector $\mathbf{U}_i$. If the error pattern is not a coset leader, an erroneous decoding will result.

5.4.8.1 The Syndrome of a Coset

If $\mathbf{e}_j$ is the coset leader or error pattern of the jth coset, then $\mathbf{U}_i + \mathbf{e}_j$ is an n-tuple in this coset. The syndrome of this n-tuple can be written

$$\mathbf{S} = (\mathbf{U}_i + \mathbf{e}_j)\mathbf{H}^T = \mathbf{U}_i\mathbf{H}^T + \mathbf{e}_j\mathbf{H}^T$$

Since $\mathbf{U}_i$ is a code vector, $\mathbf{U}_i\mathbf{H}^T = \mathbf{0}$, and we can write, as in Equation (5.37)

$$\mathbf{S} = (\mathbf{U}_i + \mathbf{e}_j)\mathbf{H}^T = \mathbf{e}_j\mathbf{H}^T \tag{5.39}$$

From Equation (5.39) it is clear that all members of a coset have the *same syndrome*, and in fact, the syndrome is used to estimate the error pattern. The syndrome for every coset is different.

5.4.8.2 Error Correction Decoding

The procedure for error correction decoding proceeds as follows:

1. Calculate the syndrome of $\mathbf{r}$ using $\mathbf{S} = \mathbf{r}\mathbf{H}^T$.
2. Locate the coset leader (error pattern), $\mathbf{e}_j$, whose syndrome equals $\mathbf{r}\mathbf{H}^T$.
3. This error pattern is assumed to be the corruption caused by the channel.
4. The corrected received vector, or code vector, is identified as $\mathbf{U} = \mathbf{r} + \mathbf{e}_j$. We can say that we retrieve the valid code vector by subtracting out the identified error; in modulo-2 arithmetic the operation of subtraction is identical to that of addition.

5.4.8.3 Locating the Error Pattern

Returning to the example of Section 5.4.3, we arrange the 2^6 = sixty-four 6-tuples in a standard array as shown in Figure 5.14. The valid code vectors are the eight vectors in the first row, and the *correctable error patterns* are the eight *coset leaders* in the first column. Notice that all 1-bit error patterns are correctable. Also notice that after exhausting all 1-bit error patterns, there remains some error-correcting capability since we have not yet accounted for all sixty-four 6-tuples. There is one unassigned coset leader; therefore, there remains the capability of correcting one additional error pattern. We have the flexibility of choosing this error pattern to be any of the n-tuples in the remaining coset. In Figure 5.14 this final correctable error pattern is chosen, somewhat arbitrarily, to be the 2-bit error pattern 0 1 0 0 0 1. Decoding will be correct if, and only if, the error pattern caused by the channel is one of the coset leaders.

We now determine the syndrome corresponding to each of the correctable

000000	110100	011010	101110	101001	011101	110011	000111
000001	110101	011011	101111	101000	011100	110010	000110
000010	110110	011000	101100	101011	011111	110001	000101
000100	110000	011110	101010	101101	011001	110111	000011
001000	111100	010010	100110	100001	010101	111011	001111
010000	100100	001010	111110	111001	001101	100011	010111
100000	010100	111010	001110	001001	111101	010011	100111
010001	100101	001011	111111	111000	001100	100010	010110

Figure 5.14 Example of a standard array for a (6, 3) code.

error sequences by computing $\mathbf{e}_j\mathbf{H}^T$ for each coset leader, as follows:

$$\mathbf{S} = \mathbf{e}_j \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 1 & 0 \\ 0 & 1 & 1 \\ 1 & 0 & 1 \end{bmatrix}$$

The results are listed in Table 5.1. Since each syndrome in the table is unique, the decoder can identify the error pattern **e** to which it corresponds.

TABLE 5.1 Syndrome Look-Up Table

Error pattern	Syndrome
0 0 0 0 0 0	0 0 0
0 0 0 0 0 1	1 0 1
0 0 0 0 1 0	0 1 1
0 0 0 1 0 0	1 1 0
0 0 1 0 0 0	0 0 1
0 1 0 0 0 0	0 1 0
1 0 0 0 0 0	1 0 0
0 1 0 0 0 1	1 1 1

5.4.8.4 Error Correction Example

As outlined in Section 5.4.8.2, we receive the vector **r** and calculate its syndrome using $\mathbf{S} = \mathbf{r}\mathbf{H}^T$. We then use the syndrome look-up table (Table 5.1), developed in the preceding section, to find the corresponding error pattern. This error pattern is an estimate of the error, and we denote it $\hat{\mathbf{e}}$. The decoder then adds $\hat{\mathbf{e}}$ to **r** to obtain an estimate of the transmitted code vector $\hat{\mathbf{U}}$.

$$\hat{\mathbf{U}} = \mathbf{r} + \hat{\mathbf{e}} = (\mathbf{U} + \mathbf{e}) + \hat{\mathbf{e}} = \mathbf{U} + (\mathbf{e} + \hat{\mathbf{e}}) \tag{5.40}$$