Electrical engineering education has undergone some radical changes during the past couple of decades and continues to do so. A modern undergraduate program in electrical engineering includes the following two introductory courses:

- *Signals and Systems*, which provides a balanced and integrated treatment of continuous-time and discrete-time forms of signals and systems. The Fourier transform (in its different forms), Laplace transform, and z-transform are treated in detail. Typically, the course also includes an elementary treatment of communication systems.
- *Probability and Random Processes*, which develops an intuitive grasp of discrete and continuous random variables and then introduces the notion of a random process and its characteristics.

Typically, these two introductory courses lead to a senior-level course on communication systems.

The fourth edition of this book has been written with this background and primary objective in mind. Simply put, the book provides a modern treatment of communication systems at a level suitable for a one- or two-semester senior undergraduate course. The emphasis is on the statistical underpinnings of communication theory with applications.

The material is presented in a logical manner, and it is illustrated with examples, with the overall aim being that of helping the student develop an intuitive grasp of the theory under discussion. Except for the Background and Preview chapter, each chapter ends with numerous problems designed not only to help the students test their understanding of the material covered in the chapter but also to challenge them to extend this material. Every chapter includes notes and references that provide suggestions for further reading. Sections or subsections that can be bypassed without loss of continuity are identified with a footnote.

A distinctive feature of the book is the inclusion of eight computer experiments using MATLAB. This set of experiments provides the basis of a "Software Laboratory", with each experiment being designed to extend the material covered in the pertinent chapter. Most important, the experiments exploit the unique capabilities of MATLAB in an instructive manner. The MATLAB codes for all these experiments are available on the Wiley Web site: **http://www.wiley.com/college/haykin/.**

The Background and Preview chapter presents introductory and motivational material, paving the way for detailed treatment of the many facets of communication systems in the subsequent 10 chapters. The material in these chapters is organized as follows:

- Chapter 1 develops a detailed treatment of *random*, or *stochastic*, *processes*, with particular emphasis on their partial characterization (i.e., second-order statistics). In effect, the discussion is restricted to wide-sense stationary processes. The correlation

properties and power spectra of random processes are described in detail. Gaussian processes and narrowband noise feature prominently in the study of communication systems, hence their treatment in the latter part of the chapter. This treatment naturally leads to the consideration of the Rayleigh and Rician distributions that arise in a communications environment.

- Chapter 2 presents an integrated treatment of *continuous-wave (CW) modulation* (i.e., *analog communications*) and their different types, as outlined here:
  (i) *Amplitude modulation*, which itself can assume one of the following forms (depending on how the spectral characteristics of the modulated wave are specified):
    - Full amplitude modulation
    - Double sideband-suppressed carrier modulation
    - Quadrature amplitude modulation
    - Single sideband modulation
    - Vestigial sideband modulation

  (ii) *Angle modulation*, which itself can assume one of two interrelated forms:
    - Phase modulation
    - Frequency modulation

  The time-domain and spectral characteristics of these modulated waves, methods for their generation and detection, and the effects of channel noise on their performances are discussed.

- Chapter 3 covers *pulse modulation* and discusses the processes of sampling, quantization, and coding that are fundamental to the digital transmission of analog signals. This chapter may be viewed as the transition from analog to digital communications. Specifically, the following types of pulse modulation are discussed:
  (i) *Analog pulse modulation*, where only time is represented in discrete form; it embodies the following special forms:
    - Pulse amplitude modulation
    - Pulse width (duration) modulation
    - Pule position modulation

  The characteristics of pulse amplitude modulation are discussed in detail, as it is basic to all forms of pulse modulation, be they of the analog or digital type.
  (ii) *Digital pulse modulation*, in which both time and signal amplitude are represented in discrete form; it embodies the following special forms:
    - Pulse-code modulation
    - Delta modulation
    - Differential pulse-code modulation

  In delta modulation, the sampling rate is increased far in excess of that used in pulse-code modulation so as to simplify implementation of the system. In contrast, in differential pulse-code modulation, the sampling rate is reduced through the use of a predictor that exploits the correlation properties of the information-bearing signal.
  (iii) MPEG/audio coding standard, which includes a psychoacoustic model as a key element in the design of the encoder.

- Chapter 4 covers *baseband pulse transmission*, which deals with the transmission of pulse-amplitude modulated signals in their baseband form. Two important issues are discussed: the effects of channel noise and limited channel bandwidth on the performance of a digital communication system. Assuming that the channel noise is additive

CONTENTS

thus represents an efficient use of resources only to the extent that the allocated bandwidth is properly used. Although the telephone network is used to transmit data, voice constitutes the bulk of the network's traffic. Indeed, circuit switching is well suited to the transmission of voice signals, since voice gives rise to a stream traffic and voice conversations tend to be of long duration (about 2 minutes on the average) compared to the time required for setting up the circuit (about 0.1 to 0.5 seconds). Moreover, in most voice conversations, there is information flow for a relatively large percentage of the connection time, which makes circuit switching all the more suitable for voice conversations.

In circuit switching, a communication link is shared between the different sessions using that link on a *fixed* allocation basis. In *packet switching*, on the other hand, the sharing is done on a *demand* basis, so it has an advantage over circuit switching in that when a link has traffic to send, the link may be more fully utilized.

The network principle of packet switching is "store and forward." Specifically, in *packet-switched network*, any message longer than a specified size is subdivided prior to transmission into segments not exceeding the specified size. The segments are commonly referred to as *packets*. The original message is reassembled at the destination on a packet-by-packet basis. The network may be viewed as a distributed pool of *network resources* (i.e., channel bandwidth, buffers, and switching processors) whose capacity is *shared dynamically* by a community of competing hosts wishing to communicate. In contrast, in a circuit-switched network, resources are dedicated to a pair of hosts for the entire period they are in session. Accordingly, packet switching is far better suited to a computer-communication environment in which bursts of data are exchanged between hosts on an occasional basis. The use of packet switching, however, requires that careful *control* be exercised on user demands; otherwise, the network may be seriously abused.

The design of a *data network* (i.e., a network in which the hosts are all made up of computers and terminals) may proceed in an orderly way by looking at the network in terms of a *layered architecture*, regarded as a hierarchy of nested layers. *Layer* refers to a process or device inside a computer system, designed to perform a specific function. Naturally, the designers of a layer will be intimately familiar with its internal details and operation. At the system level, however, a user views the layer merely as a "black box" that is described in terms of inputs, outputs, and the functional relation between outputs and inputs. In a layered architecture, each layer regards the next lower layer as one or more black boxes with some given functional specification to be used by the given higher layer. Thus, the highly complex communication problem in data networks is resolved as a manageable set of well-defined interlocking functions. It is this line of reasoning that has led to the development of the *open systems interconnection* (OSI)[7] *reference model* by a subcommittee of the International Organization for Standardization. The term *open* refers to the ability of any two systems conforming to the reference model and its associated standards to interconnect.

In the OSI reference model, the communications and related-connection functions are organized as a series of layers, or *levels*, with well-defined *interfaces*, and with each layer built on its predecessor. In particular, each layer performs a related subset of primitive functions, and it relies on the next lower layer to perform additional primitive functions. Moreover, each layer offers certain services to the next higher layer and shields the latter from the implementation details of those services. Between each pair of layers, there is an *interface*. It is the interface that defines the services offered by the lower layer to the upper layer.

The OSI model is composed of seven layers, as illustrated in Figure 5; this figure also includes a description of the functions of the individual layers of the model. Layer k on system A, say, communicates with layer k on some other system B in accordance with a

Unfortunately, Shannon's information capacity theorem does not tell us how to design the system. Nevertheless, from a design point of view, the theorem is very valuable for the following reasons:

1. The information capacity theorem provides a *bound* on what rate of data transmission is theoretically attainable for prescribed values of channel bandwidth B and received SNR. On this basis, we may use the ratio

$$\eta = \frac{R}{C}$$

as a measure of the *efficiency* of the digital communication system under study. The closer η is to unity, the more efficient the system is.
2. Equation (1) provides a basis for the trade-off between channel bandwidth B and received SNR. In particular, for a prescribed signaling rate R, we may reduce the required SNR by increasing the channel bandwidth B. Hence the motivation for using a wideband modulated scheme (e.g. pulse code modulation) for improved noise performance.
3. Equation (1) provides an idealized framework for comparing the noise performance of one modulation scheme against another.

A Digital Communication Problem

When we speak of a digital communication system having a low bit error rate, say, the implication is that only a small fraction in a long stream of binary symbols is decoded in error by the receiver. The issue of the receiver determining whether a binary symbol sent over a noisy channel is decoded in error or not is of fundamental importance to the design of digital communication systems. It is therefore appropriate briefly to discuss this basic issue so as to motivate the study of communication systems.

Suppose we have a random binary signal, $m(t)$, consisting of symbols 1 and 0 that are equally likely. Symbol 1 is represented by a constant level $+1$, and symbol 0 is represented by a constant level -1, each of which lasts for a duration T. Such a signal may represent the output of a digital computer or the digitized version of a speech signal. To facilitate the transmission of this signal over a communication channel, we employ a simple modulation scheme known as *phase-shift keying*. Specifically, the information bearing signal $m(t)$ is multiplied by a sinusoidal carrier wave $A_c \cos(2\pi f_c t)$, where A_c is the carrier amplitude, f_c is the carrier frequency, and t is time. Figure 10*a* shows a block diagram of the transmitter, the output of which is defined by

$$s(t) = \begin{cases} A_c \cos(2\pi f_c t) & \text{for symbol 1} \\ -A_c \cos(2\pi f_c t) & \text{for symbol 0} \end{cases} \tag{2}$$

where $0 \le t \le T$. The carrier frequency f_c is a multiple of $1/T$.

The channel is assumed to be distortionless but noisy, as depicted in Figure 10*b*. The received signal $x(t)$ is thus defined by

$$x(t) = s(t) + w(t) \tag{3}$$

where $w(t)$ is the additive channel noise.

The receiver consists of a correlator followed by a decision-making device, as depicted in Figure 10*c*. The *correlator* multiplies the received signal $x(t)$ by a locally generated

that emboldened Shannon to amend the title of his paper to "The Mathematical Theory of Communication" when it was reprinted a year later in a book co-authored with Warren Weaver. It is noteworthy that prior to the publication of Shannon's 1948 classic paper, it was believed that increasing the rate of information transmission over a channel would increase the probability of error; the communication theory community was taken by surprise when Shannon proved that this was not true, provided that the transmission rate was below the channel capacity. Shannon's 1948 paper was followed by some significant advances in coding theory, which include the following:

- Development of the first nontrivial *error-correcting codes* by M. J. E. Golay in 1949 and Richard W. Hamming in 1950.
- Development of *turbo codes* by C. Berrou, A. Glavieux, and P. Thitimajshima in 1993; turbo codes provide near-optimum error correcting coding and decoding performance in the Shannon sense.

The transistor was invented in 1948 by Walter H. Brattain, John Bardeen, and William Shockley at Bell Laboratories. The first silicon integrated circuit (IC) was produced by Robert Noyce in 1958. These landmark innovations in solid-state devices and integrated circuits led to the development of *very-large-scale integrated* (VLSI) circuits and single-chip *microprocessors*, and with them the nature of signal processing and the telecommunications industry changed forever.

The invention of the transistor in 1948 spurred the application of electronics to switching and digital communications. The motivation was to improve reliability, increase capacity, and reduce cost. The first call through a stored-program system was placed in March 1958 at Bell Laboratories, and the first commercial telephone service with digital switching began in Morris, Illinois, in June 1960. The first *T-1 carrier system* transmission was installed in 1962 by Bell Laboratories.

During the period 1943 to 1946, the first electronic digital computer, called the ENIAC, was built at the Moore School of Electrical Engineering of the University of Pennsylvania under the technical direction of J. Presper Eckert, Jr., and John W. Mauchly. However, John von Neumann's contributions were among the earliest and most fundamental to the theory, design, and application of digital computers, which go back to the first draft of a report written in 1945. Computers and terminals started communicating with each other over long distances in the early 1950s. The links used were initially voice-grade telephone channels operating at low speeds (300 to 1200 b/s). Various factors have contributed to a dramatic increase in data transmission rates; notable among them are the idea of *adaptive equalization*, pioneered by Robert Lucky in 1965, and efficient modulation techniques, pioneered by G. Ungerboeck in 1982. Another idea widely employed in computer communications is that of *automatic repeat-request* (ARQ). The ARQ method was originally devised by H. C. A. van Duuren during World War II and published in 1946. It was used to improve radio-telephony for telex transmission over long distances.

From 1950 to 1970, various studies were made on *computer networks*. However, the most significant of them in terms of impact on computer communications was the Advanced Research Project Agency Network (ARPANET), first put into service in 1971. The development of ARPANET was sponsored by the Advanced Research Projects Agency of the U.S. Department of Defense. The pioneering work in *packet switching* was done on ARPANET. In 1985, ARPANET was renamed the *Internet*. The turning point in the evolution of the Internet occurred in 1990 when Tim Berners-Lee proposed a hypermedia software interface to the Internet, which he named the *World Wide Web*.[12] Thereupon, in

radio receiver. A major source of channel noise is *thermal noise*, which is caused by the random motion of the electrons in conductors and devices at the front end of the receiver. We thus find that the received signal is random in nature. Although it is not possible to predict the exact value of the signal in advance, it is possible to describe the signal in terms of statistical parameters such as average power and power spectral density, as discussed in this chapter.

1.2 *Mathematical Definition of a Random Process*

In light of these introductory remarks, it is apparent that random processes have two properties. First, they are functions of time. Second, they are random in the sense that before conducting an experiment it is not possible to exactly define the waveforms that will be observed in the future.

In describing a random experiment it is convenient to think in terms of a sample space. Specifically, each outcome of the experiment is associated with a *sample point*. The totality of sample points corresponding to the aggregate of all possible outcomes of the experiment is called the *sample space*. Each sample point of the sample space is a function of time. The sample space or ensemble composed of functions of time is called a *random* or *stochastic process*.[1] As an integral part of this notion, we assume the existence of a probability distribution defined over an appropriate class of sets in the sample space, so that we may speak with confidence of the probability of various events.

Consider, then, a random experiment specified by the outcomes s from some *sample space* S, by the events defined on the sample space S, and by the probabilities of these

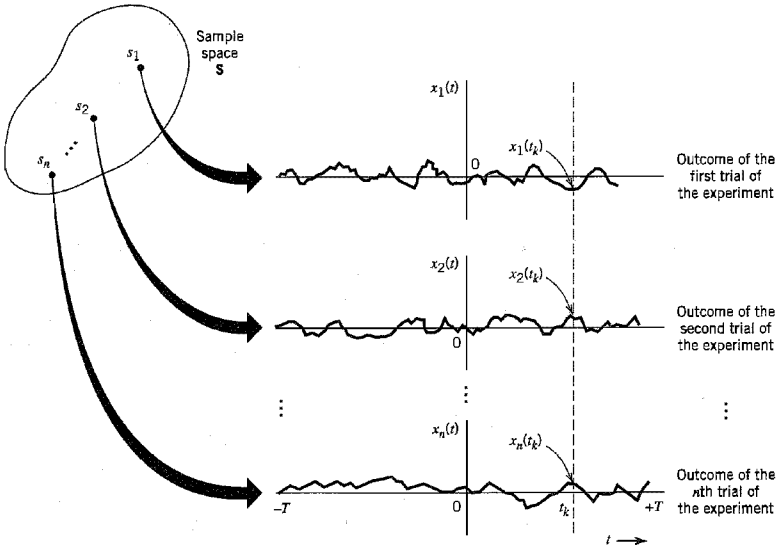

FIGURE 1.1 An ensemble of sample functions.

events. Suppose that we assign to each sample point s a function of time in accordance with the rule:

$$X(t, s), \qquad -T \leq t \leq T \tag{1.1}$$

where $2T$ is the *total observation interval*. For a fixed sample point s_j, the graph of the function $X(t, s_j)$ versus time t is called a *realization* or *sample function* of the random process. To simplify the notation, we denote this sample function as

$$x_j(t) = X(t, s_j) \tag{1.2}$$

Figure 1.1 illustrates a set of sample functions $\{x_j(t) \mid j = 1, 2, \ldots, n\}$. For this figure, we note that for a fixed time t_k inside the observation interval, the set of numbers

$$\{x_1(t_k), x_2(t_k), \ldots, x_n(t_k)\} = \{X(t_k, s_1), X(t_k, s_2), \ldots, X(t_k, s_n)\}$$

constitutes a *random variable*. Thus we have an indexed ensemble (family) of random variables $\{X(t, s)\}$, which is called a *random process*. To simplify the notation, the customary practice is to suppress the s and simply use $X(t)$ to denote a random process. We may now formally define a random process $X(t)$ as *an ensemble of time functions together with a probability rule that assigns a probability to any meaningful event associated with an observation of one of the sample functions of the random process*. Moreover, we may distinguish between a random variable and a random process as follows:

- For a random variable, the outcome of a random experiment is mapped into a number.
- For a random process, the outcome of a random experiment is mapped into a waveform that is a function of time.

1.3 Stationary Processes

In dealing with random processes encountered in the real world, we often find that the statistical characterization of a process is independent of the time at which observation of the process is initiated. That is, if such a process is divided into a number of time intervals, the various sections of the process exhibit essentially the same statistical properties. Such a process is said to be *stationary*. Otherwise, it is said to be *nonstationary*. Generally speaking, a stationary process arises from a stable physical phenomenon that has evolved into a steady-state mode of behavior, whereas a nonstationary process arises from an unstable phenomenon.

To be more precise, consider a random process $X(t)$ that is initiated at $t = -\infty$. Let $X(t_1), X(t_2), \ldots, X(t_k)$ denote the random variables obtained by observing the random process $X(t)$ at times $t_1, t_2, \ldots, t_k$, respectively. The joint distribution function of this set of random variables is $F_{X(t_1), \ldots, X(t_k)}(x_1, \ldots, x_k)$. Suppose next we shift all the observation times by a fixed amount τ, thereby obtaining a new set of random variables $X(t_1 + \tau)$, $X(t_2 + \tau), \ldots, X(t_k + \tau)$. The joint distribution function of this latter set of random variables is $F_{X(t_1+\tau), \ldots, X(t_k+\tau)}(x_1, \ldots, x_k)$. The random process $X(t)$ is said to be *stationary in the strict sense* or *strictly stationary* if the following condition holds:

$$F_{X(t_1+\tau), \ldots, X(t_k+\tau)}(x_1, \ldots, x_k) = F_{X(t_1), \ldots, X(t_k)}(x_1, \ldots, x_k) \tag{1.3}$$

for all time shifts τ, all k, and all possible choices of observation times $t_1, \ldots, t_k$. In other words, a *random process $X(t)$, initiated at time $t = -\infty$, is strictly stationary if the joint distribution of any set of random variables obtained by observing the random process $X(t)$ is invariant with respect to the location of the origin $t = 0$*. Note that the finite-dimensional

This, in turn, implies that *the autocorrelation function of a strictly stationary process depends only on the time difference* $t_2 - t_1$, as shown by

$$R_X(t_1, t_2) = R_X(t_2 - t_1) \qquad \text{for all } t_1 \text{ and } t_2 \tag{1.9}$$

Similarly, the *autocovariance function* of a strictly stationary process $X(t)$ is written as

$$\begin{aligned} C_X(t_1, t_2) &= E[(X(t_1) - \mu_X)(X(t_2) - \mu_X)] \\ &= R_X(t_2 - t_1) - \mu_X^2 \end{aligned} \tag{1.10}$$

Equation (1.10) shows that, like the autocorrelation function, the autocovariance function of a strictly stationary process $X(t)$ depends only on the time difference $t_2 - t_1$. This equation also shows that if we know the mean and autocorrelation function of the process, we can uniquely determine the autocovariance function. The mean and autocorrelation function are therefore sufficient to describe the first two moments of the process.

However, two important points should be carefully noted:

1. The mean and autocorrelation function only provide a *partial description* of the distribution of a random process $X(t)$.
2. The conditions of Equations (1.7) and (1.9), involving the mean and autocorrelation function, respectively, are *not* sufficient to guarantee that the random process $X(t)$ is strictly stationary.

Nevertheless, practical considerations often dictate that we simply limit ourselves to a partial description of the process given by the mean and autocorrelation function. The class of random processes that satisfy Equations (1.7) and (1.9) has been given various names, such as *second-order stationary*, *wide-sense stationary*, or *weakly stationary* processes. Henceforth, we shall simply refer to them as *stationary processes*.[2]

A stationary process is not necessarily strictly stationary because Equations (1.7) and (1.9) obviously do not imply the invariance of the joint (k-dimensional) distribution of Equation (1.3) with respect to the time shift τ for all k. On the other hand, a strictly stationary process does not necessarily satisfy Equations (1.7) and (1.9) as the first- and second-order moments may not exist. Clearly, however, the class of strictly stationary processes with finite second-order moments forms a subclass of the class of all stationary processes.

▪ PROPERTIES OF THE AUTOCORRELATION FUNCTION

For convenience of notation, we redefine the autocorrelation function of a stationary process $X(t)$ as

$$R_X(\tau) = E[X(t + \tau)X(t)] \qquad \text{for all } t \tag{1.11}$$

This autocorrelation function has several important properties:

1. The *mean-square value* of the process may be obtained from $R_X(\tau)$ simply by putting $\tau = 0$ in Equation (1.11), as shown by

$$R_X(0) = E[X^2(t)] \tag{1.12}$$

2. The autocorrelation function $R_X(\tau)$ is an even function of τ, that is,

$$R_X(\tau) = R_X(-\tau) \tag{1.13}$$

This property follows directly from the defining equation (1.11). Accordingly, we may also define the autocorrelation function $R_X(\tau)$ as

$$R_X(\tau) = E[X(t)X(t - \tau)]$$

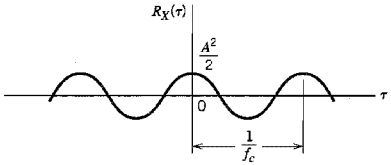

FIGURE 1.5 Autocorrelation function of a sine wave with random phase.

This means that the random variable Θ is equally likely to have any value θ in the interval $[-\pi, \pi]$. Each value of Θ corresponds to a sample in the sample space of the random process $X(t)$.

The process $X(t)$ defined by Equations (1.15) and (1.16) may represent a locally generated carrier in the receiver of a communication system, which is used in demodulation of the received signal. In particular, the random variable Θ represents the phase difference between this locally generated carrier and the sinusoidal carrier wave used to modulate the message signal in the transmitter.

The autocorrelation function of $X(t)$ is

$$\begin{aligned} R_X(\tau) &= E[X(t+\tau)X(t)] \\ &= E[A^2 \cos(2\pi f_c t + 2\pi f_c \tau + \Theta) \cos(2\pi f_c t + \Theta)] \\ &= \frac{A^2}{2} E[\cos(4\pi f_c t + 2\pi f_c \tau + 2\Theta)] + \frac{A^2}{2} E[\cos(2\pi f_c \tau)] \\ &= \frac{A^2}{2} \int_{-\pi}^{\pi} \frac{1}{2\pi} \cos(4\pi f_c t + 2\pi f_c \tau + 2\theta)\, d\theta + \frac{A^2}{2} \cos(2\pi f_c \tau) \end{aligned}$$

The first term integrates to zero, and so we get

$$R_X(\tau) = \frac{A^2}{2} \cos(2\pi f_c \tau) \tag{1.17}$$

which is plotted in Figure 1.5. We see therefore that the autocorrelation function of a sinusoidal wave with random phase is another sinusoid at the same frequency in the "τ domain" rather than the original time domain. ◀

▶ EXAMPLE 1.3 Random Binary Wave

Figure 1.6 shows the sample function $x(t)$ of a process $X(t)$ consisting of a random sequence of *binary symbols* 1 and 0. The following assumptions are made:

1. The symbols 1 and 0 are represented by pulses of amplitude $+A$ and $-A$ volts, respectively, and duration T seconds.
2. The pulses are not synchronized, so the starting time t_d of the first complete pulse for positive time is equally likely to lie anywhere between zero and T seconds. That is, t_d is the sample value of a uniformly distributed random variable T_d, with its probability density function defined by

$$f_{T_d}(t_d) = \begin{cases} \frac{1}{T}, & 0 \le t_d \le T \\ 0, & \text{elsewhere} \end{cases}$$

3. During any time interval $(n-1)T < t - t_d < nT$, where n is an integer, the presence of a 1 or a 0 is determined by tossing a fair coin; specifically, if the outcome is heads,

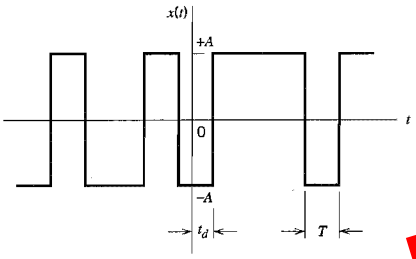

FIGURE 1.6 Sample function of random binary wave.

we have a 1 and if the outcome is tails, we have a 0. These two symbols are thus equally likely, and the occurrence of a 1 or 0 in any one interval is independent of all other intervals.

Since the amplitude levels $-A$ and $+A$ occur with equal probability, it follows immediately that $E[X(t)] = 0$ for all t, and the mean of the process is therefore zero.

To find the autocorrelation function $R_X(t_k, t_i)$, we have to evaluate $E[X(t_k)X(t_i)]$, where $X(t_k)$ and $X(t_i)$ are random variables obtained by observing the random process $X(t)$ at times t_k and t_i, respectively.

Consider first the case when $|t_k - t_i| > T$. Under this condition the random variables $X(t_k)$ and $X(t_i)$ occur in different pulse intervals and are therefore independent. We thus have

$$E[X(t_k)X(t_i)] = E[X(t_k)]E[X(t_i)] = 0, \qquad |t_k - t_i| > T$$

Consider next the case when $|t_k - t_i| < T$, with $t_k = 0$ and $t_i < t_k$. In such a situation we observe from Figure 1.6 that the random variables $X(t_k)$ and $X(t_i)$ occur in the same pulse interval if and only if the delay t_d satisfies the condition $t_d < T - |t_k - t_i|$. We thus obtain the *conditional expectation*:

$$E[X(t_k)X(t_i)|t_d] = \begin{cases} A^2, & t_d < T - |t_k - t_i| \\ 0, & \text{elsewhere} \end{cases}$$

Averaging this result over all possible values of t_d, we get

$$\begin{aligned} E[X(t_k)X(t_i)] &= \int_0^{T-|t_k-t_i|} A^2 f_{T_d}(t_d)\, dt_d \\ &= \int_0^{T-|t_k-t_i|} \frac{A^2}{T}\, dt_d \\ &= A^2\left(1 - \frac{|t_k - t_i|}{T}\right), \qquad |t_k - t_i| < T \end{aligned}$$

By similar reasoning for any other value of t_k, we conclude that the autocorrelation function of a random binary wave, represented by the sample function shown in Figure 1.6, is only a function of the time difference $\tau = t_k - t_i$, as shown by

$$R_X(\tau) = \begin{cases} A^2\left(1 - \frac{|\tau|}{T}\right), & |\tau| < T \\ 0, & |\tau| \geq T \end{cases} \tag{1.18}$$

This result is plotted in Figure 1.7.

In the last integral on the right-hand side of Equation (1.35), define a new variable

$$\tau = \tau_2 - \tau_1$$

Then we may rewrite Equation (1.35) in the form

$$E[Y^2(t)] = \int_{-\infty}^{\infty} df H(f) \int_{-\infty}^{\infty} d\tau_2 h(\tau_2) \exp(j2\pi f\tau_2) \int_{-\infty}^{\infty} R_X(\tau) \exp(-j2\pi f\tau)\, d\tau \quad (1.36)$$

However, the middle integral on the right-hand side in Equation (1.36) is simply $H^*(f)$, the complex conjugate of the frequency response of the filter, and so we may simplify this equation as

$$E[Y^2(t)] = \int_{-\infty}^{\infty} df |H(f)|^2 \int_{-\infty}^{\infty} R_X(\tau) \exp(-j2\pi f\tau)\, d\tau \quad (1.37)$$

where $|H(f)|$ is the *magnitude response* of the filter. We may further simplify Equation (1.37) by recognizing that the last integral is simply the Fourier transform of the autocorrelation function $R_X(\tau)$ of the input random process $X(t)$. This prompts us to introduce the definition of a new parameter

$$S_X(f) = \int_{-\infty}^{\infty} R_X(\tau) \exp(-j2\pi f\tau)\, d\tau \quad (1.38)$$

The function $S_X(f)$ is called the *power spectral density*, or *power spectrum*, of the stationary process $X(t)$. Thus substituting Equation (1.38) into (1.37), we obtain the desired relation:

$$E[Y^2(t)] = \int_{-\infty}^{\infty} |H(f)|^2 S_X(f)\, df \quad (1.39)$$

Equation (1.39) states that *the mean-square value of the output of a stable linear time-invariant filter in response to a stationary process is equal to the integral over all frequencies of the power spectral density of the input process multiplied by the squared magnitude response of the filter.* This is the desired frequency-domain equivalent to the time-domain relation of Equation (1.33).

To investigate the physical significance of the power spectral density, suppose that the random process $X(t)$ is passed through an ideal narrowband filter with a magnitude response centered about the frequency f_c, as shown in Figure 1.9; that is,

$$|H(f)| = \begin{cases} 1, & |f \pm f_c| < \frac{1}{2}\Delta f \\ 0, & |f \pm f_c| > \frac{1}{2}\Delta f \end{cases} \quad (1.40)$$

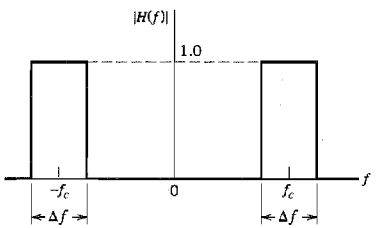

FIGURE 1.9 Magnitude response of ideal narrowband filter.

Property 4

If the random variables $X(t_1), X(t_2), \ldots, X(t_n)$, obtained by sampling a Gaussian process $X(t)$ at times $t_1, t_2, \ldots, t_n$, are uncorrelated, that is,

$$E[(X(t_k) - \mu_{X(t_k)})(X(t_i) - \mu_{X(t_i)})] = 0, \qquad i \neq k$$

then these random variables are statistically independent.

The uncorrelatedness of $X(t_1), \ldots, X(t_n)$ means that the covariance matrix is a diagonal matrix as shown by

$$\mathbf{\Sigma} = \begin{bmatrix} \sigma_1^2 & & & \mathbf{O} \\ & \sigma_2^2 & & \\ & & \ddots & \\ \mathbf{O} & & & \sigma_n^2 \end{bmatrix}$$

where

$$\sigma_i^2 = E[(X(t_i) - E[X(t_i)])^2], \qquad i = 1, 2, \ldots, n$$

Under this condition, the multivariate Gaussian distribution of Equation (1.85) simplifies to

$$f_{\mathbf{X}}(\mathbf{x}) = \prod_{i=1}^{n} f_{X_i}(x_i)$$

where $X_i = X(t_i)$ and

$$f_{X_i}(x_i) = \frac{1}{\sqrt{2\pi}\sigma_i} \exp\left(-\frac{(x_i - \mu_{X_i})^2}{2\sigma_i^2}\right)$$

In words, if the Gaussian random variables $X(t_1), \ldots, X(t_n)$ are uncorrelated, then they are statistically independent, which, in turn, means that the joint probability density function of this set of random variables can be expressed as the product of the probability density functions of the individual random variables in the set.

1.9 *Noise*

The term *noise* is used customarily to designate unwanted signals that tend to disturb the transmission and processing of signals in communication systems and over which we have incomplete control. In practice, we find that there are many potential sources of noise in a communication system. The sources of noise may be external to the system (e.g., atmospheric noise, galactic noise, man-made noise), or internal to the system. The second category includes an important type of noise that arises from *spontaneous fluctuations* of current or voltage in electrical circuits.[6] This type of noise represents a basic limitation on the transmission or detection of signals in communication systems involving the use of electronic devices. The two most common examples of spontaneous fluctuations in electrical circuits are *shot noise* and *thermal noise*, which are described in the sequel.

▪ SHOT NOISE

Shot noise arises in electronic devices such as diodes and transistors because of the discrete nature of current flow in these devices. For example, in a *photodetector* circuit a current

pulse is generated every time an electron is emitted by the cathode due to incident light from a source of constant intensity. The electrons are naturally emitted at random times denoted by τ_k, where $-\infty < k < \infty$. It is assumed that the random emissions of electrons have been going on for a long time. Thus, the total current flowing through the photodetector may be modeled as an infinite sum of current pulses, as shown by

$$X(t) = \sum_{k=-\infty}^{\infty} h(t - \tau_k) \tag{1.86}$$

where $h(t - \tau_k)$ is the current pulse generated at time τ_k. The process $X(t)$ defined by Equation (1.86) is a stationary process called *shot noise*.

The number of electrons, $N(t)$, emitted in the time interval $[0, t]$ constitutes a discrete stochastic process, the value of which increases by one each time an electron is emitted. Figure 1.14 shows a sample function of such a process. Let the mean value of the number of electrons, ν, emitted between time t and $t + t_0$ be defined by

$$E[\nu] = \lambda t_0 \tag{1.87}$$

The parameter λ is a constant called the *rate* of the process. The total number of electrons emitted in the interval $[t, t + t_0]$, that is,

$$\nu = N(t + t_0) - N(t)$$

follows a *Poisson distribution* with a mean value equal to λt_0. In particular, the probability that k electrons are emitted in the interval $[t, t + t_0]$ is defined by

$$P(\nu = k) = \frac{(\lambda t_0)^k}{k!} e^{-\lambda t_0} \qquad k = 0, 1, \ldots \tag{1.88}$$

Unfortunately, a detailed statistical characterization of the shot-noise process $X(t)$ defined in Equation (1.86) is a difficult mathematical task. Here we simply quote the results pertaining to the first two moments of the process:

- The mean of $X(t)$ is

$$\mu_X = \lambda \int_{-\infty}^{\infty} h(t)\, dt \tag{1.89}$$

where λ is the rate of the process and $h(t)$ is the waveform of a current pulse.

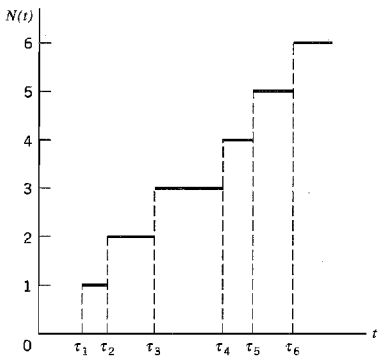

FIGURE 1.14 Sample function of a Poisson counting process.

That is, the autocorrelation function of white noise consists of a delta function weighted by the factor $N_0/2$ and occurring at $\tau = 0$, as in Figure 1.16*b*. We note that $R_W(\tau)$ is zero for $\tau \neq 0$. Accordingly, any two different samples of white noise, no matter how closely together in time they are taken, are uncorrelated. If the white noise $w(t)$ is also Gaussian, then the two samples are statistically independent. In a sense, white Gaussian noise represents the ultimate in "randomness."

Strictly speaking, white noise has infinite average power and, as such, it is not physically realizable. Nevertheless, white noise has simple mathematical properties exemplified by Equations (1.93) and (1.95), which make it useful in statistical system analysis.

The utility of a white noise process is parallel to that of an impulse function or delta function in the analysis of linear systems. Just as we may observe the effect of an impulse only after it has been passed through a system with finite bandwidth, so it is with white noise whose effect is observed only after passing through a similar system. We may state, therefore, that as long as the bandwidth of a noise process at the input of a system is appreciably larger than that of the system itself, then we may model the noise process as white noise.

▶ EXAMPLE 1.10 Ideal Low-Pass Filtered White Noise

Suppose that a white Gaussian noise $w(t)$ of zero mean and power spectral density $N_0/2$ is applied to an ideal low-pass filter of bandwidth B and passband magnitude response of one. The power spectral density of the noise $n(t)$ appearing at the filter output is therefore (see Figure 1.17)

$$S_N(f) = \begin{cases} \dfrac{N_0}{2}, & -B < f < B \\ 0, & |f| > B \end{cases} \tag{1.96}$$

The autocorrelation function of $n(t)$ is the inverse Fourier transform of the power spectral density shown in Figure 1.17*a*:

$$\begin{aligned} R_N(\tau) &= \int_{-B}^{B} \frac{N_0}{2} \exp(j2\pi f\tau)\, df \\ &= N_0 B \operatorname{sinc}(2B\tau) \end{aligned} \tag{1.97}$$

This autocorrelation function is plotted in Figure 1.17*b*. We see that $R_N(\tau)$ has its maximum value of N_0B at the origin, and it passes through zero at $\tau = \pm k/2B$, where $k = 1, 2, 3, \cdots$.

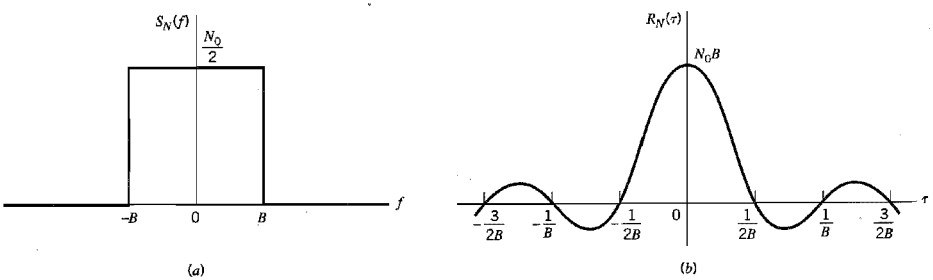

FIGURE 1.17 Characteristics of low-pass filtered white noise. (*a*) Power spectral density. (*b*) Autocorrelation function.

Since the input noise $w(t)$ is Gaussian (by hypothesis), it follows that the band-limited noise $n(t)$ at the filter output is also Gaussian. Suppose now that $n(t)$ is sampled at the rate of $2B$ times per second. From Figure 1.17*b*, we see that the resulting noise samples are uncorrelated and, being Gaussian, they are statistically independent. Accordingly, the joint probability density function of a set of noise samples obtained in this way is equal to the product of the individual probability density functions. Note that each such noise sample has a mean of zero and variance of N_0B. ◀

▶ **EXAMPLE 1.11 Correlation of White Noise with a Sinusoidal Wave**

Consider the sample function

$$w'(t) = \sqrt{\frac{2}{T}} \int_0^T w(t) \cos(2\pi f_c t)\, dt \tag{1.98}$$

which is the output of a correlator with white Gaussian noise $w(t)$ and a sinusoidal wave $\sqrt{2/T}\cos(2\pi f_c t)$ as inputs; the scaling factor $\sqrt{2/T}$ is included here to make the sinusoidal wave have unit energy over the interval $0 \le t \le T$. (This problem was encountered in the Background and Preview chapter but was not elaborated on at that time.) With the noise $w(t)$ having zero mean, it immediately follows that the correlator output $w'(t)$ has zero mean, too. The variance of the correlator output is defined by

$$\begin{aligned}
\sigma^2 &= E\left[\frac{2}{T}\int_0^T \int_0^T w(t_1)\cos(2\pi f_c t_1) w(t_2)\cos(2\pi f_c t_2)\, dt_1\, dt_2\right] \\
&= \frac{2}{T}\int_0^T \int_0^T E[w(t_1)w(t_2)]\cos(2\pi f_c t_1)\cos(2\pi f_c t_2)\, dt_1\, dt_2 \\
&= \frac{2}{T}\int_0^T \int_0^T R_W(t_1, t_2)\cos(2\pi f_c t_1)\cos(2\pi f_c t_2)\, dt_1\, dt_2
\end{aligned}$$

where $R_W(t_1, t_2)$ is the autocorrelation function of the white noise $w(t)$. But from Equation (1.95):

$$R_W(t_1, t_2) = \frac{N_0}{2}\delta(t_1 - t_2)$$

where $N_0/2$ is the power spectral density of the white noise $w(t)$. Accordingly, we may simplify the expression for the variance σ^2 as

$$\sigma^2 = \frac{N_0}{2}\cdot\frac{2}{T}\int_0^T \int_0^T \delta(t_1 - t_2)\cos(2\pi f_c t_1)\cos(2\pi f_c t_2)\, dt_1\, dt_2$$

We now invoke the *sifting property* of the delta function, namely,

$$\int_{-\infty}^{\infty} g(t)\,\delta(t)\, dt = g(0)$$

where $g(t)$ is a continuous function of time, assuming the value $g(0)$ at time $t = 0$. Hence, we may further simplify σ^2 as

$$\begin{aligned}
\sigma^2 &= \frac{N_0}{2}\cdot\frac{2}{T}\int_0^T \cos^2(2\pi f_c t)\, dt \\
&= \frac{N_0}{2}
\end{aligned} \tag{1.99}$$

where it is assumed that the frequency f_c of the sinusoidal wave input is an integer multiple of the reciprocal of T. ◀

$n_I(t)$ or the quadrature noise component $n_Q(t)$ is as shown in Figure 1.21*c*. The autocorrelation function of $n_I(t)$ or $n_Q(t)$ is therefore (see Example 1.10):

$$R_{N_I}(\tau) = R_{N_Q}(\tau) = 2N_0B\ \text{sinc}(2B\tau) \tag{1.104}$$

1.12 *Representation of Narrowband Noise in Terms of Envelope and Phase Components*

In Section 1.11 we considered the representation of a narrowband noise $n(t)$ in terms of its in-phase and quadrature components. We may also represent the noise $n(t)$ in terms of its envelope and phase components as follows:

$$n(t) = r(t)\cos[2\pi f_c t + \psi(t)] \tag{1.105}$$

where

$$r(t) = [n_I^2(t) + n_Q^2(t)]^{1/2} \tag{1.106}$$

and

$$\psi(t) = \tan^{-1}\left[\frac{n_Q(t)}{n_I(t)}\right] \tag{1.107}$$

The function $r(t)$ is called the *envelope* of $n(t)$, and the function $\psi(t)$ is called the *phase* of $n(t)$.

The envelope $r(t)$ and phase $\psi(t)$ are both sample functions of low-pass random processes. As illustrated in Figure 1.18*b*, the time interval between two successive peaks of the envelope $r(t)$ is approximately $1/B$, where $2B$ is the bandwidth of the narrowband noise $n(t)$.

The probability distributions of $r(t)$ and $\psi(t)$ may be obtained from those of $n_I(t)$ and $n_Q(t)$ as follows. Let N_I and N_Q denote the random variables obtained by observing (at some fixed time) the random processes represented by the sample functions $n_I(t)$ and $n_Q(t)$, respectively. We note that N_I and N_Q are independent Gaussian random variables of zero mean and variance σ^2, and so we may express their joint probability density function by

$$f_{N_I,N_Q}(n_I, n_Q) = \frac{1}{2\pi\sigma^2}\exp\left(-\frac{n_I^2 + n_Q^2}{2\sigma^2}\right) \tag{1.108}$$

Accordingly, the probability of the joint event that N_I lies between n_I and $n_I + dn_I$ and that N_Q lies between n_Q and $n_Q + dn_Q$ (i.e., the pair of random variables N_I and N_Q lies jointly inside the shaded area of Figure 1.21*a*) is given by

$$f_{N_I,N_Q}(n_I, n_Q)\,dn_I\,dn_Q = \frac{1}{2\pi\sigma^2}\exp\left(-\frac{n_I^2 + n_Q^2}{2\sigma^2}\right)dn_I\,dn_Q \tag{1.109}$$

Define the transformation (see Figure 1.21*a*)

$$n_I = r\cos\psi \tag{1.110}$$

$$n_Q = r\sin\psi \tag{1.111}$$

In a limiting sense, we may equate the two incremental areas shown shaded in Figures 1.21*a* and 1.21*b* and thus write

$$dn_I\,dn_Q = r\,dr\,d\psi \tag{1.112}$$

observing the output of the filter at some fixed time has a Gaussian distribution. The narrowband nature of the noise means that it may be represented in terms of an in-phase and a quadrature component. These two components are both low-pass, Gaussian processes, each with zero mean and a variance equal to that of the original narrowband noise. Alternatively, a Gaussian narrowband noise may be represented in terms of a Rayleigh-distributed envelope and a uniformly distributed phase. Each of these representations has its own specific area of application, as shown in subsequent chapters of the book.

NOTES AND REFERENCES

1. For a rigorous treatment of random processes, see the classic books of Doob (1953), Loève (1963), and Cramér and Leadbetter (1967).

2. There is another important class of random processes commonly encountered in practice, the mean and autocorrelation function of which exhibit *periodicity* as in

$$\mu_X(t + T) = \mu_X(t)$$
$$R_X(t_1 + T, t_2 + T) = R_X(t_1, t_2)$$

for all t_1 and t_2. A random process $X(t)$ satisfying this pair of conditions is said to be *cyclostationary* (in the wide sense). Modeling the process $X(t)$ as cyclostationary adds a new dimension, namely, period T to the partial description of the process. Examples of cyclostationary processes include a television signal obtained by raster-scanning a random video field, and a modulated process obtained by varying the amplitude, phase, or frequency of a sinusoidal carrier. For detailed discussion of cyclostationary processes, see Franks (1969), pp. 204–214, and the paper by Gardner and Franks (1975).

3. Traditionally, Equations (1.42) and (1.43) have been referred to in the literature as the Wiener-Khintchine relations in recognition of pioneering work done by Norbert Wiener and A. I. Khintchine; for their original papers, see Wiener (1930) and Khintchine (1934). A discovery of a forgotten paper by Albert Einstein on time-series analysis (delivered at the Swiss Physical Society's February 1914 meeting in Basel) reveals that Einstein had discussed the autocorrelation function and its relationship to the spectral content of a time series many years before Wiener and Khintchine. An English translation of Einstein's paper is reproduced in the *IEEE ASSP Magazine*, vol. 4, October 1987. This particular issue also contains articles by W. A. Gardner and A. M. Yaglom, which elaborate on Einstein's original work.

4. For further details of power spectrum estimation, see Blackman and Tukey (1958), Box and Jenkins (1976), Marple (1987), and Kay (1988).

5. The Gaussian distribution and associated Gaussian process are named after the great mathematician C. F. Gauss. At age 18, Gauss invented *the method of least squares* for finding the best value of a sequence of measurements of some quantity. Gauss later used the method of least squares in fitting orbits of planets to data measurements, a procedure that was published in 1809 in his book entitled *Theory of Motion of the Heavenly Bodies*. In connection with the error of observation, he developed the *Gaussian distribution*. This distribution is also known as the *normal distribution*. Partly for historical reasons, mathematicians commonly use the term normal, while engineers and physicists commonly use the term Gaussian.

6. For a detailed treatment of electrical noise, see Van der Ziel (1970) and the collection of papers edited by Gupta (1977).

   An introductory treatment of shot noise is presented in Helstrom (1990). For a more detailed treatment, see the paper by Yue, Luganani, and Rice (1978).

Thermal noise was first studied experimentally by J. B. Johnson in 1928, and for this reason it is sometimes referred to as the *Johnson noise*. Johnson's experiments were confirmed theoretically by Nyquist (1928).

7. The noisiness of a receiver may also be measured in terms of the so-called *noise figure*. The relationship between the noise figure and the equivalent noise temperature is developed in Chapter 8.
8. The Rayleigh distribution is named after the English physicist J. W. Strutt, Lord Rayleigh.
9. The Rician distribution is named in honor of Stephen O. Rice for the original contribution reported in a pair of papers published in 1944 and 1945, which are reproduced in Wax (1954).
10. The statistical characterization of communication systems presented in this book is confined to the first two moments, mean and autocorrelation function (equivalently, autocovariance function) of the pertinent random process. However, when a random process is transmitted through a nonlinear system, valuable information is contained in higher-order moments of the resulting output process. The parameters used to characterize higher-order moments in the time domain are called cumulants, and their multidimensional Fourier transforms are called *polyspectra*. For a discussion of higher-order cumulants and polyspectra and their estimation, see the paper by Nikias and Raghuveer (1987).

PROBLEMS

Stationarity and Ergodicity

1.1 Consider a random process $X(t)$ defined by

$$X(t) = \sin(2\pi f_c t)$$

in which the frequency f_c is a random variable uniformly distributed over the interval $[0, W]$. Show that $X(t)$ is nonstationary. *Hint:* Examine specific sample functions of the random process $X(t)$ for the frequency $f = W/4$, $W/2$, and W, say.

1.2 Consider the sinusoidal process

$$X(t) = A\cos(2\pi f_c t)$$

where the frequency f_c is constant and the amplitude A is uniformly distributed:

$$f_A(a) = \begin{cases} 1, & 0 \le a \le 1 \\ 0, & \text{otherwise} \end{cases}$$

Determine whether or not this process is strictly stationary.

1.3 A random process $X(t)$ is defined by

$$X(t) = A\cos(2\pi f_c t)$$

where A is a Gaussian-distributed random variable of zero mean and variance σ_A^2. This random process is applied to an ideal integrator, producing the output

$$Y(t) = \int_0^t X(\tau)\, d\tau$$

(a) Determine the probability density function of the output $Y(t)$ at a particular time t_k.

(b) Determine whether or not $Y(t)$ is stationary.

(c) Determine whether or not $Y(t)$ is ergodic.

1.4 Let X and Y be statistically independent Gaussian-distributed random variables, each with zero mean and unit variance. Define the Gaussian process

$$Z(t) = X\cos(2\pi t) + Y\sin(2\pi t)$$

where v is a constant, is applied to the low-pass RC filter of Figure P1.14. Determine the power spectral density and autocorrelation function of the random process at the filter output.

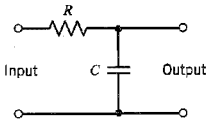

FIGURE P1.14

1.15 A *running integrator* is defined by

$$y(t) = \int_{t-T}^{t} x(\tau)\, d\tau$$

where $x(t)$ is the input, $y(t)$ is the output, and T is the integration period. Both $x(t)$ and $y(t)$ are sample functions of stationary processes $X(t)$ and $Y(t)$, respectively. Show that the power spectral density of the integrator output is related to that of the integrator input as

$$S_Y(f) = T^2 \operatorname{sinc}^2(fT) S_X(f)$$

1.16 A zero-mean stationary process $X(t)$ is applied to a linear filter whose impulse response is defined by a truncated exponential:

$$h(t) = \begin{cases} ae^{-at}, & 0 \le t \le T \\ 0, & \text{otherwise} \end{cases}$$

Show that the power spectral density of the filter output $Y(t)$ is defined by

$$S_Y(f) = \frac{a^2}{a^2 + 4\pi^2 f^2} (1 - 2 \exp(-aT) \cos(2\pi fT) + \exp(-2aT)) S_X(f)$$

where $S_X(f)$ is the power spectral density of the filter input.

1.17 The output of an oscillator is described by

$$X(t) = A \cos(2\pi ft - \Theta)$$

where A is a constant, and f and Θ are independent random variables. The probability density function of Θ is defined by

$$f_\Theta(\theta) = \begin{cases} \dfrac{1}{2\pi}, & 0 \le \theta \le 2\pi \\ 0, & \text{otherwise} \end{cases}$$

Find the power spectral density of $X(t)$ in terms of the probability density function of the frequency f. What happens to this power spectral density when the frequency f assumes a constant value?

1.31 Consider a Gaussian noise $n(t)$ with zero mean and the power spectral density $S_N(f)$ shown in Figure P1.31.

(a) Find the probability density function of the envelope of $n(t)$.

(b) What are the mean and variance of this envelope?

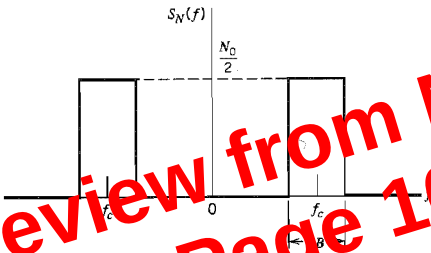

FIGURE P1.31

Computer Experiments

1.32 In this computer experiment we study the statistical characterization of a random process $X(t)$ defined by

$$X(t) = A\cos(2\pi f_c t + \Theta) + W(t)$$

where the phase Θ of the sinusoidal component is a uniformly distributed random variable over the interval $[-\pi, \pi]$, and $W(t)$ is a white Gaussian noise component of zero mean and power spectral density $N_0/2$. The two components of $X(t)$ are statistically independent; hence the autocorrelation function of $X(t)$ is

$$R_X(\tau) = \frac{A^2}{2}\cos(2\pi f_c \tau) + \frac{N_0}{2}\delta(\tau)$$

This equation shows that for $|\tau| > 0$ the autocorrelation function $R_X(\tau)$ has the same sinusoidal waveform as the signal component of $X(t)$.

The purpose of this computer experiment is to perform the computation of $R_X(\tau)$ using two different methods:

(a) *Ensemble averaging.* Generate $M = 50$ randomly picked realizations of the process $X(t)$. Hence compute the product $x(t + \tau)x(t)$ for some fixed time t, where $x(t)$ is a realization of $X(t)$. Repeat the computation of $x(t + \tau)x(t)$ for the M realizations of $X(t)$, and thereby compute the average of these computations over M. Repeat this sequence of computations for different values of τ.

(b) *Time averaging.* Compute the time-averaged autocorrelation function

$$R_x(\tau, T) = \frac{1}{2T}\int_{-T}^{T} x(t + \tau)x(t)\, dt$$

where $x(t)$ is a particular realization of $X(t)$, and $2T$ is the total observation interval. For this computation, use the Fourier-transform pair:

$$R_x(\tau, T) \rightleftharpoons \frac{1}{2T}|X_T(f)|^2$$

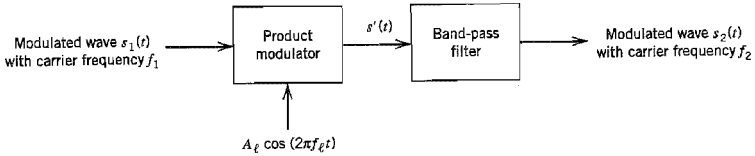

FIGURE 2.16 Block diagram of mixer.

To explain the action of the mixer, consider the situation described in Figure 2.17, where, for the purpose of illustration, it is assumed that the mixer input $s_1(t)$ is an AM signal with carrier frequency f_1 and bandwidth $2W$. Part (*a*) of Figure 2.17 displays the AM spectrum $S_1(f)$ assuming that $f_1 > W$. Part (*b*) of the figure displays the spectrum $S'(f)$ of the resulting signal $s'(t)$ at the product modulator output.

The signal $s'(t)$ may be viewed as the sum of two modulated components: one component represented by the shaded spectrum in Figure 2.17*b*, and the other component represented by the unshaded spectrum in this figure. Depending on whether the incoming carrier frequency f_1 is translated upward or downward, we may identify two different situations, as described here:

Up conversion. In this case the translated carrier frequency f_2 is greater than the incoming carrier frequency f_1, and the required local oscillator frequency f_l is therefore defined by

$$f_2 = f_1 + f_l$$

or

$$f_l = f_2 - f_1$$

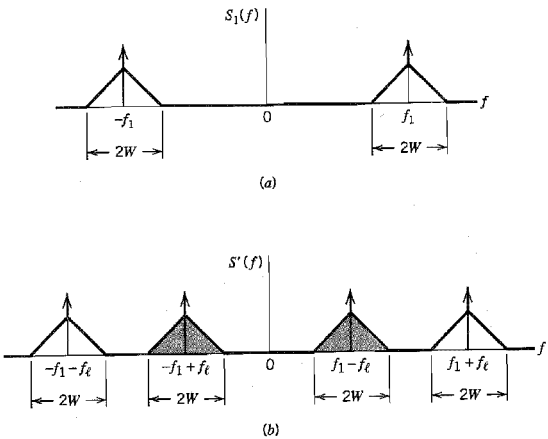

FIGURE 2.17 (*a*) Spectrum of modulated signal $s_1(t)$ at the mixer input. (*b*) Spectrum of the corresponding signal $s'(t)$ at the output of the product modulator in the mixer.

where A_c is the carrier amplitude. A complete oscillation occurs whenever $\theta_i(t)$ changes by 2π radians. If $\theta_i(t)$ increases monotonically with time, the average frequency in Hertz, over an interval from t to $t + \Delta t$, is given by

$$f_{\Delta t}(t) = \frac{\theta_i(t + \Delta t) - \theta_i(t)}{2\pi \Delta t} \tag{2.20}$$

We may thus define the *instantaneous frequency* of the angle-modulated signal $s(t)$ as follows:

$$\begin{aligned} f_i(t) &= \lim_{\Delta t \to 0} f_{\Delta t}(t) \\ &= \lim_{\Delta t \to 0} \left[\frac{\theta_i(t + \Delta t) - \theta_i(t)}{2\pi \Delta t} \right] \\ &= \frac{1}{2\pi} \frac{d\theta_i(t)}{dt} \end{aligned} \tag{2.21}$$

Thus, according to Equation (2.19), we may interpret the angle-modulated signal $s(t)$ as a rotating phasor of length A_c and angle $\theta_i(t)$. The angular velocity of such a phasor is $d\theta_i(t)/dt$ measured in radians per second, in accordance with Equation (2.21). In the simple case of an unmodulated carrier, the angle $\theta_i(t)$ is

$$\theta_i(t) = 2\pi f_c t + \phi_c$$

and the corresponding phasor rotates with a constant angular velocity equal to $2\pi f_c$. The constant ϕ_c is the value of $\theta_i(t)$ at $t = 0$.

There are an infinite number of ways in which the angle $\theta_i(t)$ may be varied in some manner with the message (baseband) signal. However, we shall consider only two commonly used methods, phase modulation and frequency modulation, defined as follows:

1. *Phase modulation* (PM) *is that form of angle modulation in which the angle* $\theta_i(t)$ *is varied linearly with the message signal* $m(t)$, as shown by

$$\theta_i(t) = 2\pi f_c t + k_p m(t) \tag{2.22}$$

The term $2\pi f_c t$ represents the angle of the *unmodulated* carrier; and the constant k_p represents the *phase sensitivity* of the modulator, expressed in radians per volt on the assumption that $m(t)$ is a voltage waveform. For convenience, we have assumed in Equation (2.22) that the angle of the unmodulated carrier is zero at $t = 0$. The phase-modulated signal $s(t)$ is thus described in the time domain by

$$s(t) = A_c \cos[2\pi f_c t + k_p m(t)] \tag{2.23}$$

2. *Frequency modulation* (FM) *is that form of angle modulation in which the instantaneous frequency* $f_i(t)$ *is varied linearly with the message signal* $m(t)$, as shown by

$$f_i(t) = f_c + k_f m(t) \tag{2.24}$$

The term f_c represents the frequency of the unmodulated carrier, and the constant k_f represents the *frequency sensitivity* of the modulator, expressed in Hertz per volt

where we have used the value $\Gamma(3/2) = \sqrt{\pi}/2$. To calculate the mean signal s_o at the detector output, we also need the expectation of $y(t)$. Due to the combined presence of signal and noise, we recall from Section 1.13 that $y(t)$ is Rician distributed, as shown by

$$f_Y(y) = \begin{cases} \dfrac{y}{\sigma_N^2} \exp\left(-\dfrac{y^2 + A_c^2}{2\sigma_N^2}\right) I_0\left(\dfrac{A_c y}{\sigma_N^2}\right) & \text{for } y \geq 0 \\ 0 & \text{otherwise} \end{cases} \tag{2.114}$$

where $I_0(\cdot)$ is the modified Bessel function of the first kind of zero order (see Appendix 3). Hence,

$$E[y(t)] = \int_0^\infty \frac{y^2}{\sigma_N^2} \exp\left(-\frac{y^2 + A_c^2}{2\sigma_N^2}\right) I_0\left(\frac{A_c y}{\sigma_N^2}\right) dy \tag{2.115}$$

Putting $A_c y/\sigma_N^2 = u$ and recognizing that $\rho = A_c^2/2\sigma_N^2$, we may recast this expectation in the form

$$E[y(t)] = \frac{\sigma_N}{(2\rho)^{3/2}} \exp(\quad) \int_0^\infty u^2 \exp\left(-\frac{u^2}{4\rho}\right) I_0(u)\, du \tag{2.116}$$

The integral in Equation (2.116) can be written in a concise form by using *confluent hypergeometric functions*; see Appendix 4. In particular, using the integral representation

$$\int_0^\infty u^{m-1} \exp(-b^2u^2) I_0(u)\, du = \frac{\Gamma(m/2)}{2b^m} \left({}_1F_1\left(\frac{m}{2}; 1; \frac{1}{4b^2}\right)\right) \tag{2.117}$$

with $m = 3$, $\Gamma(m/2) = \sqrt{\pi}/2$ and $b^2 = 1/4\rho$, we may express the expectation of $y(t)$ in terms of the confluent hypergeometric function ${}_1F_1(3/2;1;\rho)$ as

$$E[y(t)] = \sqrt{\frac{\pi}{2}}\, \sigma_N \exp(-\rho) \left({}_1F_1\left(\frac{3}{2}; 1; \rho\right)\right) \tag{2.118}$$

We may further simplify matters by using the following identity:

$$\exp(-u)({}_1F_1(\alpha;\beta;u)) = {}_1F_1(\beta - \alpha;\beta;-u) \tag{2.119}$$

and so finally express the expectation of $y(t)$ in the concise form

$$E[y(t)] = \sqrt{\frac{\pi}{2}}\, \sigma_N \left({}_1F_1\left(-\frac{1}{2}; 1; -\rho\right)\right) \tag{2.120}$$

Thus using Equations (2.113) and (2.120) in Equation (2.106) yields the mean output signal as

$$s_o = \sqrt{\frac{\pi}{2}}\, \sigma_N \left({}_1F_1\left(-\frac{1}{2}; 1; -\rho\right) - 1\right) \tag{2.121}$$

whose dependence on the standard deviation σ_N of the noise $n(t)$ is testimony to the intermingling of signal and noise at the detector output.

Following a similar procedure, we may express the mean-square value of the detector output $y(t)$ as

$$\begin{aligned} E[y^2(t)] &= \int_0^\infty \frac{y^3}{\sigma_N^2} \exp\left(-\frac{y^2 + A_c^2}{2\sigma_N^2}\right) I_0\left(\frac{A_c y}{\sigma_N^2}\right) dy \\ &= 2\sigma_N^2({}_1F_1(-1;1;-\rho)) \end{aligned} \tag{2.122}$$

The discriminator output is therefore

$$\begin{aligned} v(t) &= \frac{1}{2\pi}\frac{d\theta(t)}{dt} \\ &= k_f m(t) + n_d(t) \end{aligned} \tag{2.140}$$

where the noise term $n_d(t)$ is defined by

$$n_d(t) = \frac{1}{2\pi A_c}\frac{d}{dt}\{r(t)\sin[\psi(t) - \phi(t)]\} \tag{2.141}$$

We thus see that provided the carrier-to-noise ratio is high, the discriminator output $v(t)$ consists of the original message signal $m(t)$ multiplied by the constant factor k_f, plus an additive noise component $n_d(t)$. Accordingly, we may use the output signal-to-noise ratio as previously defined to assess the quality of performance of the FM receiver. Before doing this, however, it is instructive to see how we can simplify the expression defining the noise $n_d(t)$.

From the phasor diagram of Figure 2.11, we note that the effect of variations in the phase $\psi(t)$ of the narrowband noise appears referred to the signal term $\phi(t)$. We know that the phase $\psi(t)$ is uniformly distributed over 2π radians. It would therefore be tempting to assume that the phase difference $\psi(t) - \phi(t)$ is also uniformly distributed over 2π radians. If such an assumption were true, then the noise $n_d(t)$ at the discriminator output would be independent of the modulating signal and would depend only on the characteristics of the carrier and narrowband noise. Theoretical considerations show that this assumption is justified provided that the carrier-to-noise ratio is high.[9] Then we may simplify Equation (2.141) as:

$$n_d(t) \simeq \frac{1}{2\pi A_c}\frac{d}{dt}\{r(t)\sin[\psi(t)]\} \tag{2.142}$$

However, from the defining equations for $r(t)$ and $\psi(t)$, we note that the quadrature component $n_Q(t)$ of the filtered noise $n(t)$ is

$$n_Q(t) = r(t)\sin[\psi(t)] \tag{2.143}$$

Therefore, we may rewrite Equation (2.142) as

$$n_d(t) \simeq \frac{1}{2\pi A_c}\frac{dn_Q(t)}{dt} \tag{2.144}$$

This means that *the additive noise $n_d(t)$ appearing at the discriminator output is determined effectively by the carrier amplitude A_c and the quadrature component $n_Q(t)$ of the narrowband noise $n(t)$.*

The output signal-to-noise ratio is defined as the ratio of the average output signal power to the average output noise power. From Equation (2.140), we see that the message component in the discriminator output, and therefore the low-pass filter output, is $k_f m(t)$. Hence, the average output signal power is equal to $k_f^2 P$, where P is the average power of the message signal $m(t)$.

To determine the average output noise power, we note that the noise $n_d(t)$ at the discriminator output is proportional to the time derivative of the quadrature noise component $n_Q(t)$. Since the differentiation of a function with respect to time corresponds to multiplication of its Fourier transform by $j2\pi f$, it follows that we may obtain the noise process $n_d(t)$ by passing $n_Q(t)$ through a linear filter with a frequency response equal to

$$\frac{j2\pi f}{2\pi A_c} = \frac{jf}{A_c}$$

noise deviates appreciably from a linear function of ρ when ρ is about 11 dB. Also when the signal is present, the resulting modulation of the carrier tends to increase the average number of clicks per second. Experimentally, it is found that occasional clicks are heard in the receiver output at a carrier-to-noise ratio of about 13 dB, which appears to be only slightly higher than what theory indicates. Also it is of interest to note that the increase in the average number of clicks per second tends to cause the output signal-to-noise ratio to fall off somewhat more sharply just below the threshold level in the presence of modulation.

From the foregoing discussion we may conclude that threshold effects in FM receivers may be avoided in most practical cases of interest if the carrier-to-noise ratio is equal to or greater than 20 or, equivalently, 13 dB. Thus using Equation (2.154) we find that the loss of message at the discriminator output is negligible if

$$\frac{A_c^2}{2B_T N_0} \geq 20$$

or, equivalently, if the average transmitted power $A_c^2/2$ satisfies the condition

$$\frac{A_c^2}{2} \geq 20B_T N_0 \tag{2.155}$$

To use this formula, we may proceed as follows:

1. For a specified modulation index β and message bandwidth W, we determine the transmission bandwidth of the FM wave, B_T, using the universal curve of Figure 2.26 or Carson's rule.
2. For a specified average noise power per unit bandwidth, N_0, we use Equation (2.155) to determine the minimum value of the average transmitted power $A_c^2/2$ that is necessary to operate above threshold.

■ FM THRESHOLD REDUCTION

In communication systems using frequency modulation, there is particular interest in reducing the noise threshold in an FM receiver so as to satisfactorily operate the receiver with the minimum signal power possible. Threshold reduction in FM receivers may be achieved by using an FM demodulator with negative feedback[12] (commonly referred to as an *FMFB demodulator*), or by using a *phase-locked loop demodulator*. Such devices are referred to as *extended-threshold demodulators*, the idea of which is illustrated in Figure 2.46. The threshold extension shown in this figure is measured with respect to the standard frequency discriminator (i.e., one without feedback).

The block diagram of an FMFB demodulator[13] is shown in Figure 2.47. We see that the local oscillator of the conventional FM receiver has been replaced by a voltage-controlled oscillator (VCO) whose instantaneous output frequency is controlled by the demodulated signal. In order to understand the operation of this receiver, suppose for the moment that the VCO is removed from the circuit and the feedback loop is left open. Assume that a wideband FM signal is applied to the receiver input, and a second FM signal, from the same source but whose modulation index is a fraction smaller, is applied to the VCO terminal of the mixer. The output of the mixer would consist of the difference frequency component, because the sum frequency component is removed by the band-pass filter. The frequency deviation of the mixer output would be small, although the frequency deviation of both input FM waves is large, since the difference between their instantaneous deviations is small. Hence, the modulation indices would subtract and the resulting FM wave at the mixer output would have a smaller modulation index. The FM wave with

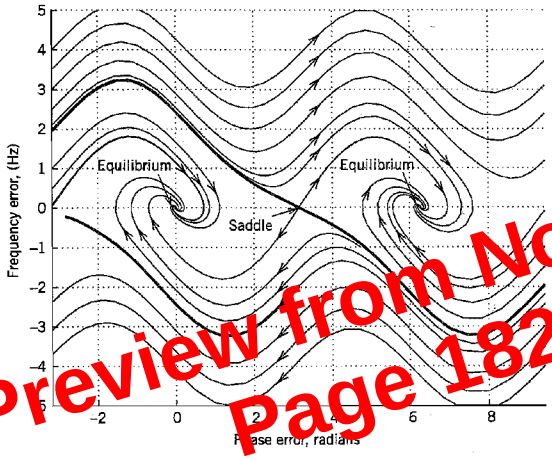

FIGURE 2.54 Phase-plane portrait for critical damping and sinusoidal modulation.

3. For an initial frequency error

$$\frac{1}{K}\frac{d\phi_e}{dt}$$

with an absolute value equal to 2, we have a *saddle point* at $(0, \pi)$ where the slightest perturbation applied to the phase-locked loop causes it to shift to the equilibrium point $(0, 0)$ or $(0, 2\pi)$.

2.15 *Summary and Discussion*

In this chapter we studied the principles of continuous-wave (CW) modulation. This analog form of modulation uses a sinusoidal carrier whose amplitude or angle is varied in accordance with a message signal. We may thus distinguish two families of CW modulation: amplitude modulation and angle modulation.

■ AMPLITUDE MODULATION

Amplitude modulation may itself be classified into four types, depending on the spectral content of the modulated signal. The four types of amplitude modulation and their practical merits are as follows:

1. *Full amplitude modulation (AM), in which the upper and lower sidebands are transmitted in full, accompanied by the carrier wave.*
Accordingly, demodulation of an AM signal is accomplished rather simply in the receiver by using an envelope detector, for example. It is for this reason we find that full AM is commonly used in commercial AM *radio broadcasting*, which involves a single powerful transmitter and numerous receivers that are relatively inexpensive to build.

terms of the peak deviation Δf of the carrier frequency, the delay τ, and the repetition frequency f_0 of the transmitted signal.

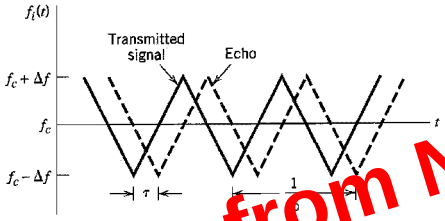

FIGURE P2.25

2.26 The instantaneous frequency of a sine wave is equal to $f_c - \Delta f$ for $|t| \leq T/2$, and f_c for $|t| > T/2$. Determine the spectrum of this frequency-modulated wave. *Hint:* Divide up the time interval of interest into three regions: $-\infty < t < -T/2$, $-T/2 \leq t \leq T/2$, and $T/2 < t < \infty$.

2.27 Single-sideband modulation may be viewed as a hybrid form of amplitude modulation and frequency modulation. Evaluate the envelope and instantaneous frequency of an SSB wave for the following two cases:

(a) When only the upper sideband is transmitted.

(b) When only the lower sideband is transmitted.

2.28 Consider a narrowband FM signal approximately defined by

$$s(t) \simeq A_c \cos(2\pi f_c t) - \beta A_c \sin(2\pi f_c t) \sin(2\pi f_m t)$$

(a) Determine the envelope of this modulated signal. What is the ratio of the maximum to the minimum value of this envelope? Plot this ratio versus β, assuming that β is restricted to the interval $0 \leq \beta \leq 0.3$.

(b) Determine the average power of the narrowband FM signal, expressed as a percentage of the average power of the unmodulated carrier wave. Plot this result versus β, assuming that β is restricted to the interval $0 \leq \beta \leq 0.3$.

(c) By expanding the angle $\theta_i(t)$ of the narrow-band FM signal $s(t)$ in the form of a power series, and restricting the modulation index β to a maximum value of 0.3 radians, show that

$$\theta_i(t) \simeq 2\pi f_c t + \beta \sin(2\pi f_m t) - \frac{\beta^3}{3} \sin^3(2\pi f_m t)$$

What is the power ratio of third harmonic to fundamental component for $\beta = 0.3$?

2.29 The sinusoidal modulating wave

$$m(t) = A_m \cos(2\pi f_m t)$$

is applied to a phase modulator with phase sensitivity k_p. The unmodulated carrier wave has frequency f_c and amplitude A_c.

(a) Determine the spectrum of the resulting phase-modulated signal, assuming that the maximum phase deviation $\beta_p = k_p A_m$ does not exceed 0.3 radians.

(b) Construct a phasor diagram for this modulated signal, and compare it with that of the corresponding narrowband FM signal.

2.30 Suppose that the phase-modulated signal of Problem 2.29 has an arbitrary value for the maximum phase deviation β_p. This modulated signal is applied to an ideal band-pass filter with midband frequency f_c and a passband extending from $f_c - 1.5f_m$ to $f_c + 1.5f_m$.

(b) Illustrate the operation of this demodulator, using the sawtooth wave of Figure P2.24 as the modulating wave.

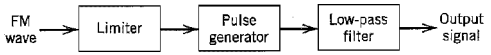

FIGURE P2.41

2.42 Suppose that the received signal in an FM system contains some residual amplitude modulation of positive amplitude $a(t)$, as shown by

$$s(t) = a(t)\cos[2\pi f_c t + \phi(t)]$$

where f_c is the carrier frequency. The phase $\phi(t)$ is related to the modulating signal $m(t)$ by

$$\phi(t) = 2\pi k_f \int_0^t m(\tau)\, d\tau$$

where k_f is a constant. Assume that the signal $s(t)$ is restricted to a frequency band of width B_T, centered at f_c, where B_T is the transmission bandwidth of the FM signal in the absence of amplitude modulation, and that the amplitude modulation is slowly varying compared with $\phi(t)$. Show that the output of an ideal frequency discriminator produced by $s(t)$ is proportional to $a(t)m(t)$. *Hint:* Use the complex notation described in Appendix 2 to represent the modulated wave $s(t)$.

2.43 (a) Let the modulated wave $s(t)$ in Problem 2.42 be applied to a *hard limiter*, whose output $z(t)$ is defined by

$$\begin{aligned} z(t) &= \operatorname{sgn}[s(t)] \\ &= \begin{cases} +1, & s(t) > 0 \\ -1, & s(t) < 0 \end{cases} \end{aligned}$$

Show that the limiter output may be expressed in the form of a Fourier series as follows:

$$z(t) = \frac{4}{\pi} \sum_{n=0}^{\infty} \frac{(-1)^n}{2n+1} \cos[2\pi f_c t(2n+1) + (2n+1)\phi(t)]$$

(b) Suppose that the limiter output is applied to a band-pass filter with a passband magnitude response of one and bandwidth B_T centered about the carrier frequency f_c, where B_T is the transmission bandwidth of the FM signal in the absence of amplitude modulation. Assuming that f_c is much greater than B_T, show that the resulting filter output equals

$$y(t) = \frac{4}{\pi} \cos[2\pi f_c t + \phi(t)]$$

By comparing this output with the original modulated signal $s(t)$ defined in Problem 2.42, comment on the practical usefulness of the result.

2.44 (a) Consider an FM signal of carrier frequency f_c, which is produced by a modulating signal $m(t)$. Assume that f_c is large enough to justify treating this FM signal as a narrowband signal. Find an approximate expression for its Hilbert transform.

(b) For the special case of a sinusoidal modulating wave $m(t) = A_m \cos(2\pi f_m t)$, find the exact expression for the Hilbert transform of the resulting FM signal. For this case, what is the error in the approximation used in part (a)?

2.45 The *single sideband version of angle modulation* is defined by

$$s(t) = \exp[-\hat{\phi}(t)] \cos[2\pi f_c t + \phi(t)]$$

where $\hat{\phi}(t)$ is the Hilbert transform of the phase function $\phi(t)$, and f_c is the carrier frequency.

(a) Show that the spectrum of the modulated signal $s(t)$ contains no frequency components in the interval $-f_c < f < f_c$, and is of infinite extent.

(b) Given that the phase function

$$\phi(t) = \beta \sin(2\pi f_m t)$$

where β is the modulation index and f_m is the modulation frequency, derive the corresponding expression for the modulated wave $s(t)$.

Note: For Problems 2.44 and 2.45, you need to refer to Appendix 2 for a treatment of the Hilbert transform.

Noise in CW Modulation Systems

2.46 A DSB-SC modulated signal is transmitted over a noisy channel, with the power spectral density of the noise being as shown in Figure P2.46. The message bandwidth is 4 kHz and the carrier frequency is 200 kHz. Assuming that the average power of the modulated wave is 10 watts, determine the output signal-to-noise ratio of the receiver.

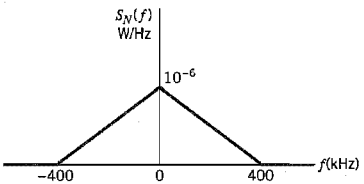

FIGURE P2.46

2.47 Evaluate the autocorrelation functions and cross-correlation functions of the in-phase and quadrature components of the narrowband noise at the coherent detector input for (a) the DSB-SC system, (b) an SSB system using the lower sideband, and (c) an SSB system using the upper sideband.

2.48 In a receiver using coherent detection, the sinusoidal wave generated by the local oscillator suffers from a phase error $\theta(t)$ with respect to the carrier wave $\cos(2\pi f_c t)$. Assuming that $\theta(t)$ is a sample function of a zero-mean Gaussian process of variance σ_θ^2, and that most of the time the maximum value of $\theta(t)$ is small compared with unity, find the mean-square error of the receiver output for DSB-SC modulation. The mean-square error is defined as the expected value of the squared difference between the receiver output and the message signal component of the receiver output.

2.49 Following a procedure similar to that described in Section 2.11 for the DSB-SC receiver, extend this noise analysis to a SSB receiver. Specifically, evaluate the following:

(a) The output signal-to-noise ratio.

(b) The channel signal-to-noise ratio.

Hence, show that the figure of merit for the SSB receiver is exactly the same as that for the DSB-SC receiver. Note that unlike the DSB-SC receiver, the midband frequency of the spectral density function of the narrowband-filtered noise at the front end of the SSB

receiver is offset from the carrier frequency f_c by an amount equal to $W/2$, where W is the message bandwidth.

2.50 Let a message signal $m(t)$ be transmitted using single-sideband modulation. The power spectral density of $m(t)$ is

$$S_M(f) = \begin{cases} a\dfrac{|f|}{W}, & |f| \leq W \\ 0, & \text{otherwise} \end{cases}$$

where a and W are constants. White Gaussian noise of zero mean and power spectral density $N_0/2$ is added to the SSB modulated wave at the receiver input. Find an expression for the output signal-to-noise ratio of the receiver.

2.51 Consider the output of an envelope detector defined by Equation (2.92), which is reproduced here for convenience

$$y(t) = [(A_c[1 + k_a m(t)] + n_I(t)]^2 + n_Q^2(t)]^{1/2}$$

(a) Assume that the probability of the event

$$|n_Q(t)| < |A_c[1 + k_a m(t)]|$$

is equal to or less than δ_1, where $\varepsilon \ll 1$. What is the probability that the effect of the quadrature component $n_Q(t)$ is negligible?

(b) Suppose that k_a is adjusted relative to the message signal $m(t)$ such that the probability of the event

$$A_c[1 + k_a m(t)] + n_I(t) < 0$$

is equal to δ_2. What is the probability that the approximation

$$y(t) \simeq A_c[1 + k_a m(t)] + n_I(t)$$

is valid?

(c) Comment on the significance of the result in part (b) for the case when δ_1 and δ_2 are both small compared with unity.

2.52 An unmodulated carrier of amplitude A_c and frequency f_c and band-limited white noise are summed and then passed through an ideal envelope detector. Assume the noise spectral density to be of height $N_0/2$ and bandwidth $2W$, centered about the carrier frequency f_c. Determine the output signal-to-noise ratio for the case when the carrier-to-noise ratio is high.

2.53 Let R denote the random variable obtained by observing the output of an envelope detector at some fixed time. Intuitively, the envelope detector is expected to be operating well into the threshold region if the probability that the random variable R exceeds the carrier amplitude A_c is 0.5. On the other hand, if this same probability is only 0.01, the envelope detector is expected to be relatively free of loss of message and the threshold effect.

(a) Assuming that the narrowband noise at the detector input is white, zero-mean, Gaussian with spectral density $N_0/2$ and the message bandwidth is W, show that the probability of the event $R \geq A_c$ is

$$P(R \geq A_c) = \exp(-\rho)$$

where ρ is the carrier-to-noise ratio:

$$\rho = \frac{A_c^2}{4WN_0}$$

(b) Using the formula for this probability, calculate the carrier-to-noise ratio when (1) the envelope detector is expected to be well into the threshold region, and (2) it is expected to be operating satisfactorily.

Hence, under the following two conditions:

1. $G(f) = 0$ for $|f| \geq W$
2. $f_s = 2W$

we find from Equation (3.5) that

$$G(f) = \frac{1}{2W} G_\delta(f), \qquad -W < f < W \tag{3.6}$$

Substituting Equation (3.4) into (3.6), we may also write

$$G(f) = \frac{1}{2W} \sum_{n=-\infty}^{\infty} g\left(\frac{n}{2W}\right) \exp\left(-\frac{j\pi nf}{W}\right), \qquad -W < f < W \tag{3.7}$$

Therefore, if the sample values $g(n/2W)$ of a signal $g(t)$ are specified for all n, then the Fourier transform $G(f)$ of the signal is uniquely determined by using the discrete-time Fourier transform of Equation (3.7). Because $g(t)$ is related to $G(f)$ by the inverse Fourier transform, it follows that the signal $g(t)$ is itself uniquely determined by the sample values $g(n/2W)$ for $-\infty < n < \infty$. In other words, the sequence $\{g(n/2W)\}$ has all the information contained in $g(t)$.

Consider next the problem of reconstructing the signal $g(t)$ from the sequence of sample values $\{g(n/2W)\}$. Substituting Equation (3.7) in the formula for the inverse Fourier transform defining $g(t)$ in terms of $G(f)$, we get

$$\begin{aligned} g(t) &= \int_{-\infty}^{\infty} G(f) \exp(j2\pi ft)\, df \\ &= \int_{-W}^{W} \frac{1}{2W} \sum_{n=-\infty}^{\infty} g\left(\frac{n}{2W}\right) \exp\left(-\frac{j\pi nf}{W}\right) \exp(j2\pi ft)\, df \end{aligned}$$

Interchanging the order of summation and integration:

$$g(t) = \sum_{n=-\infty}^{\infty} g\left(\frac{n}{2W}\right) \frac{1}{2W} \int_{-W}^{W} \exp\left[j2\pi f\left(t - \frac{n}{2W}\right)\right] df \tag{3.8}$$

The integral term in Equation (3.8) is readily evaluated, yielding the final result

$$\begin{aligned} g(t) &= \sum_{n=-\infty}^{\infty} g\left(\frac{n}{2W}\right) \frac{\sin(2\pi Wt - n\pi)}{(2\pi Wt - n\pi)} \\ &= \sum_{n=-\infty}^{\infty} g\left(\frac{n}{2W}\right) \operatorname{sinc}(2Wt - n), \qquad -\infty < t < \infty \end{aligned} \tag{3.9}$$

Equation (3.9) provides an *interpolation formula* for reconstructing the original signal $g(t)$ from the sequence of sample values $\{g(n/2W)\}$, with the sinc function $\operatorname{sinc}(2Wt)$ playing the role of an *interpolation function*. Each sample is multiplied by a delayed version of the interpolation function, and all the resulting waveforms are added to obtain $g(t)$.

We may now state the *sampling theorem* for strictly band-limited signals of finite energy in two equivalent parts, which apply to the transmitter and receiver of a pulse-modulation system, respectively:

1. *A band-limited signal of finite energy, which has no frequency components higher than W Hertz, is completely described by specifying the values of the signal at instants of time separated by 1/2W seconds.*
2. *A band-limited signal of finite energy, which has no frequency components higher than W Hertz, may be completely recovered from a knowledge of its samples taken at the rate of 2W samples per second.*

random variable M of continuous amplitude into a discrete random variable V; their respective sample values m and v are related by Equation (3.22). Let the quantization error be denoted by the random variable Q of sample value q. We may thus write

$$q = m - v \tag{3.23}$$

or, correspondingly,

$$Q = M - V \tag{3.24}$$

With the input M having zero mean, and the quantizer assumed to be symmetric as in Figure 3.10, it follows that the quantizer output V and therefore the quantization error Q, will also have zero mean. Thus for a partial statistical characterization of the quantizer in terms of output signal-to-(quantization) noise ratio, we need only find the mean-square value of the quantization error Q.

Consider then an input m of continuous amplitude in the range $(-m_{max}, m_{max})$. Assuming a uniform quantizer of the midrise type illustrated in Figure 3.10, we find that the step-size of the quantizer is given by

$$\Delta = \frac{2m_{max}}{L} \tag{3.25}$$

where L is the total number of representation levels. For a uniform quantizer, the quantization error Q will have its sample values bounded by $-\Delta/2 \le q \le \Delta/2$. If the step-size Δ is sufficiently small (i.e., the number of representation levels L is sufficiently large), it is reasonable to assume that the quantization error Q is a *uniformly distributed* random variable, and the interfering effect of the quantization noise on the quantizer input is similar to that of thermal noise. We may thus express the probability density function of the quantization error Q as follows:

$$f_Q(q) = \begin{cases} \dfrac{1}{\Delta}, & -\dfrac{\Delta}{2} < q \le \dfrac{\Delta}{2} \\ 0, & \text{otherwise} \end{cases} \tag{3.26}$$

For this to be true, however, we must ensure that the incoming signal does *not* overload the quantizer. Then, with the mean of the quantization error being zero, its variance σ_Q^2 is the same as the mean-square value:

$$\begin{aligned} \sigma_Q^2 &= E[Q^2] \\ &= \int_{-\Delta/2}^{\Delta/2} q^2 f_Q(q)\, dq \end{aligned} \tag{3.27}$$

Substituting Equation (3.26) into (3.27), we get

$$\begin{aligned} \sigma_Q^2 &= \frac{1}{\Delta}\int_{-\Delta/2}^{\Delta/2} q^2\, dq \\ &= \frac{\Delta^2}{12} \end{aligned} \tag{3.28}$$

Typically, the L-ary number k, denoting the kth representation level of the quantizer, is transmitted to the receiver in binary form. Let R denote the number of *bits per sample* used in the construction of the binary code. We may then write

$$L = 2^R \tag{3.29}$$

TABLE 3.1 *Signal-to-(quantization) noise ratio for varying number of representation levels for sinusoidal modulation*

Number of Representation Levels, L	*Number of Bits per Sample, R*	*Signal-to-Noise Ratio (dB)*
32	5	31.8
64	6	37.8
128	7	43.8
256	8	49.8

For various values of L and R, the corresponding values of signal-to-noise ratio are given in Table 3.1. From Table 3.1 we can make a quick estimate of the number of bits per sample required for a desired output signal-to-noise ratio when using sinusoidal modulation. ◀

Thus far in this section we have focused on how to characterize memoryless scalar quantizers and assess their performance. In so doing, however, we avoided the optimum design of quantizers, that is, the issue of selecting the representation levels and partition cells so as to minimize the average quantization power for a prescribed number of representation levels. Unfortunately, this optimization problem does not lend itself to a closed-form solution because of the highly *nonlinear* nature of the quantization process. Rather, we have effective algorithms for finding the optimum design in an iterative manner. A well-known algorithm that deserves to be mentioned in this context is the Lloyd-Max quantizer, which is discussed next.

■ CONDITIONS FOR OPTIMALITY OF SCALAR QUANTIZERS

In designing a scalar quantizer the challenge is how to select the representation levels and surrounding partition cells so as to minimize the average quantization power for a fixed number of representation levels.

To state the problem in mathematical terms, consider a message signal $m(t)$ drawn from a stationary process $M(t)$. Let $-A \leq m \leq A$ denote the dynamic range of $m(t)$, which is partitioned into a set of L cells, as depicted in Figure 3.12. The boundaries of the partition cells are defined by a set of real numbers $m_1, m_2, \ldots, m_{L+1}$ that satisfy the following three conditions:

$$\begin{aligned} m_1 &= -A \\ m_{L+1} &= A \\ m_k &\leq m_{k+1} \text{ for } k = 1, 2, \ldots, L \end{aligned}$$

The kth partition cell is defined by

$$\mathcal{I}_k: m_k < m \leq m_{k+1} \text{ for } k = 1, 2, \ldots, L \tag{3.36}$$

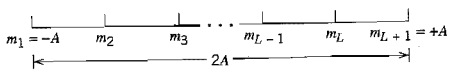

FIGURE 3.12 Illustrating the partitioning of the dynamic range $-A \leq m \leq A$ of a message signal $m(t)$ into a set of L cells.

Example 3.2. These bit streams are called the *primary rate* in the digital hierarchy, because it is the lowest bit rate that exists outside a digital switch. The *digital switch* is a device consisting of memory and logic, the function of which is merely the switching of digital signals, hence the name.

- The *second-level multiplexer* combines four DS1 bit streams to obtain a *digital signal two* (*DS2*) at 6.312 Mb/s.
- The *third-level multiplexer* combines seven DS2 bit streams to obtain a *digital signal three* (*DS3*) at 44.736 Mb/s.
- The *fourth-level multiplexer* combines six DS3 bit streams to obtain a *digital signal four* (*DS4*) at 274.176 Mb/s.
- The *fifth-level multiplexer*, the final one in the hierarchy, combines two DS4 bit streams to obtain a *digital signal five* (*DS5*) at 560.160 Mb/s.

Note that the bit rate of a digital signal produced by any one of these multiplexers is slightly higher than the prescribed multiple of the incoming bit rate because of bit stuffing built into the design of each multiplexer. Bit stuffing is discussed in the sequel.

Moreover, it is important to recognize that the functions of a digital transmission facility is merely to carry a bit stream without interpreting what the bits themselves mean. However, the digital switches at the two ends of the facility do have a common understanding of how to interpret the bits within the stream, such as whether the bits represent voice or data, framing format, signaling format, and so on.

There are some basic problems involved in the design of a digital multiplexer, irrespective of its grouping:

1. Digital signals cannot be directly interleaved into a format that allows for their eventual separation unless their bit rates are locked to a common clock. Rather, provision has to be made for *synchronization* of the incoming digital signals, so that they can be properly interleaved.
2. The multiplexed signal must include some form of *framing* so that its individual components can be identified at the receiver.
3. The multiplexer has to handle small variations in the bit rates of the incoming digital signals. For example, a 1000-km coaxial cable carrying 3×10^8 pulses per second will have about one million pulses in transit, with each pulse occupying about one meter of the cable. A 0.01 percent variation in the propagation delay, produced by a 1°F decrease in temperature, will result in 100 fewer pulses in the cable. Clearly, these pulses must be absorbed by the multiplexer.

To tailor the requirements of synchronization and rate adjustment to accommodate small variations in the input data rates, we may use a technique known as *bit stuffing*. The idea here is to have the outgoing bit rate of the multiplexer slightly higher than the sum of the maximum expected bit rates of the input channels by stuffing in additional non-information carrying pulses. All incoming digital signals are stuffed with a number of bits sufficient to raise each of their bit rates to equal that of a locally generated clock. To accomplish bit stuffing, each incoming digital signal or bit stream is fed into an *elastic store* at the multiplexer. The elastic store is a device that stores a bit stream in such a manner that the stream may be read out at a rate different from the rate at which it is read in. At the demultiplexer, the stuffed bits must obviously be removed from the multiplexed signal. This requires a method that can be used to identify the stuffed bits. To illustrate one such method, and also show one method of providing frame synchronization, we describe the signal format of the AT&T *M12 multiplexer*, which is designed to combine four DS1 bit

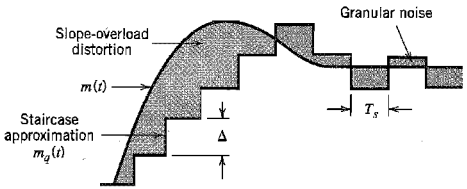

FIGURE 3.24 Illustration of the two different forms of quantization error in delta modulation.

Thus except for the quantization error $q[n-1]$, the quantizer input is a *first backward difference* of the input signal, which may be viewed as a digital approximation to the derivative of the input signal or, equivalently, as the inverse of the digital integration process. If we consider the maximum slope of the original input waveform $m(t)$, it is clear that in order for the sequence of samples $\{m_q[n]\}$ to increase as fast as the input sequence of samples $\{m[n]\}$ in a region of maximum slope of $m(t)$, we require that the condition

$$\frac{\Delta}{T_s} \geq \max\left|\frac{dm(t)}{dt}\right| \tag{3.58}$$

be satisfied. Otherwise, we find that the step-size Δ is too small for the staircase approximation $m_q(t)$ to follow a steep segment of the input waveform $m(t)$, with the result that $m_q(t)$ falls behind $m(t)$, as illustrated in Figure 3.24. This condition is called *slope overload*, and the resulting quantization error is called *slope-overload distortion (noise)*. Note that since the maximum slope of the staircase approximation $m_q(t)$ is fixed by the step size Δ, increases and decreases in $m_q(t)$ tend to occur along straight lines. For this reason, a delta modulator using a fixed step size is often referred to as a *linear delta modulator*.

In contrast to slope-overload distortion, *granular noise* occurs when the step size Δ is too large relative to the local slope characteristics of the input waveform $m(t)$, thereby causing the staircase approximation $m_q(t)$ to hunt around a relatively flat segment of the input waveform; this phenomenon is also illustrated in Figure 3.24. Granular noise is analogous to quantization noise in a PCM system.

We thus see that there is a need to have a large step-size to accommodate a wide dynamic range, whereas a small step size is required for the accurate representation of relatively low-level signals. It is therefore clear that the choice of the optimum step size that minimizes the mean-square value of the quantization error in a linear delta modulator will be the result of a compromise between slope-overload distortion and granular noise. To satisfy such a requirement, we need to make the delta modulator "adaptive," in the sense that the step size is made to vary in accordance with the input signal; this issue is discussed further in a computer experiment presented in Section 3.16.

Delta-Sigma Modulation

As mentioned earlier, the quantizer input in the conventional form of delta modulation may be viewed as an approximation to the *derivative* of the incoming message signal. This behavior leads to a drawback of delta modulation in that transmission disturbances such as noise result in an accumulative error in the demodulated signal. This drawback can be

The LMS algorithm is a *stochastic* adaptive filtering algorithm, stochastic in the sense that, starting from the *initial condition* defined by $\{w_k[0]\}_{k=1}^{p}$, it seeks to find the minimum point of the error surface by following a zig-zag path. Moreover, it never finds this minimum point exactly. Rather, it executes a random motion around the minimum point of the error surface, once steady-state conditions are established.

With this material on linear prediction at hand, we are ready to discuss practical improvements on the performance of pulse-code modulation.

3.14 *Differential Pulse-Code Modulation*

When a voice or video signal is sampled at a rate slightly higher than the Nyquist rate, as usually done in pulse-code modulation, the resulting sampled signal is found to exhibit a high degree of correlation between adjacent samples. The meaning of this high correlation is that, in an average sense, the signal does not change rapidly from one sample to the next, and as a result, the difference between adjacent samples has a variance that is smaller than the variance of the signal itself. When these highly correlated samples are encoded, as in the standard PCM system, the resulting encoded signal contains *redundant information*. This means that symbols that are not absolutely essential to the transmission of information are generated as a result of the encoding process. By removing this redundancy before encoding, we obtain a more efficient coded signal, which is the basic idea behind differential pulse-code modulation.

Now if we know the past behavior of a signal up to a certain point in time, we may use prediction to make an estimate of a future value of the signal as described in Section 3.13. Suppose then a baseband signal $m(t)$ is sampled at the rate $f_s = 1/T_s$ to produce the sequence $\{m[n]\}$ whose samples are T_s seconds apart. The fact that it is possible to predict future values of the signal $m(t)$ provides motivation for the *differential quantization* scheme shown in Figure 3.28*a*. In this scheme, the input signal to the quantizer is defined by

$$e[n] = m[n] - \hat{m}[n] \tag{3.74}$$

which is the difference between the unquantized input sample $m[n]$ and a prediction of it, denoted by $\hat{m}[n]$. This predicted value is produced by using a linear prediction filter whose input, as we will see, consists of a quantized version of the input sample $m[n]$. The difference signal $e[n]$ is the prediction error, since it is the amount by which the prediction filter fails to predict the input exactly. By encoding the quantizer output, as in Figure 3.28*a*, we obtain a variant of PCM known as *differential pulse-code modulation*[10] (DPCM).

The quantizer output may be expressed as

$$e_q[n] = e[n] + q[n] \tag{3.75}$$

where $q[n]$ is the quantization error. According to Figure 3.28*a*, the quantizer output $e_q[n]$ is added to the predicted value $\hat{m}[n]$ to produce the prediction-filter input

$$m_q[n] = \hat{m}[n] + e_q[n] \tag{3.76}$$

Substituting Equation (3.75) into (3.76), we get

$$m_q[n] = \hat{m}[n] + e[n] + q[n] \tag{3.77}$$

However, from Equation (3.74) we observe that the sum term $\hat{m}[n] + e[n]$ is equal to the input sample $m[n]$. Therefore, we may simplify Equation (3.77) as

$$m_q[n] = m[n] + q[n] \tag{3.78}$$

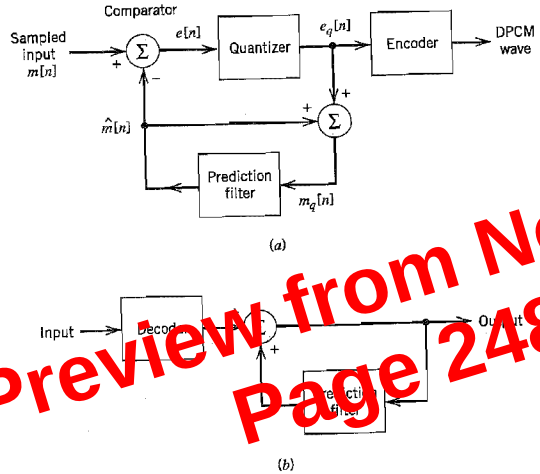

FIGURE 3.28 DPCM system. (*a*) Transmitter. (*b*) Receiver.

which represents a quantized version of the input sample $m[n]$. That is, irrespective of the properties of the prediction filter, the quantized sample $m_q[n]$ at the prediction filter input differs from the original input sample $m[n]$ by the quantization error $q[n]$. Accordingly, if the prediction is good, the variance of the prediction error $e[n]$ will be smaller than the variance of $m[n]$, so that a quantizer with a given number of levels can be adjusted to produce a quantization error with a smaller variance than would be possible if the input sample $m[n]$ were quantized directly as in a standard PCM system.

The receiver for reconstructing the quantized version of the input is shown in Figure 3.28*b*. It consists of a decoder to reconstruct the quantized error signal. The quantized version of the original input is reconstructed from the decoder output using the same prediction filter used in the transmitter of Figure 3.28*a*. In the absence of channel noise, we find that the encoded signal at the receiver input is identical to the encoded signal at the transmitter output. Accordingly, the corresponding receiver output is equal to $m_q[n]$, which differs from the original input $m[n]$ only by the quantization error $q[n]$ incurred as a result of quantizing the prediction error $e[n]$.

From the foregoing analysis we observe that, in a noise-free environment, the prediction filters in the transmitter and receiver operate on the same sequence of samples, $m_q[n]$. It is with this purpose in mind that a feedback path is added to the quantizer in the transmitter, as shown in Figure 3.28*a*.

Differential pulse-code modulation includes delta modulation as a special case. In particular, comparing the DPCM system of Figure 3.28 with the DM system of Figure 3.23, we see that they are basically similar, except for two important differences: the use of a one-bit (two-level) quantizer in the delta modulator and the replacement of the prediction filter by a single delay element (i.e., zero prediction order). Simply put, DM is the 1-bit version of DPCM. Note that unlike a standard PCM system, the transmitters of both the DPCM and DM involve the use of *feedback*.

DPCM, like DM, is subject to slope-overload distortion whenever the input signal changes too rapidly for the prediction filter to track it. Also, like PCM, DPCM suffers from quantization noise.

with $\Delta/2$ for PCM being replaced by Δ for DM.) The DM system is designed to handle analog message signals limited to bandwidth W.

(a) Show that the average quantization noise power produced by the system is

$$N = \frac{4\pi^2 A^2 f_m^2 W}{3 f_s^3}$$

where it is assumed that the step size Δ has been chosen in accordance with the formula used in Problem 3.27 so as to avoid slope overload.

(b) Hence determine the signal-to-(quantization) noise ratio of the DM system for a sinusoidal input.

3.29 Consider a DM system designed to accommodate analog message signals limited to bandwidth $W = 5$ kHz. A sinusoidal test signal of amplitude $A = 1$ volt and frequency $f_m = 1$ kHz is applied to the system. The sampling rate of the system is 50 kHz.

(a) Calculate the step size Δ required to minimize slope overload.

(b) Calculate the signal-to-(quantization) noise ratio of the system for the specified sinusoidal test signal.

For these calculations, use the formula derived for Problems 3.27 and 3.28.

3.30 Consider a low-pass signal with a bandwidth of 3 kHz. A linear delta modulation system, with step size $\Delta = 0.1$V, is used to process this signal at a sampling rate ten times the Nyquist rate.

(a) Evaluate the maximum amplitude of a test sinusoidal signal of frequency 1 kHz, which can be processed by the system without slope-overload distortion.

(b) For the specifications given in part (a), evaluate the output signal-to-noise ratio under (i) prefiltered, and (ii) postfiltered conditions.

Linear Prediction

3.31 A one-step linear predictor operates on the sampled version of a sinusoidal signal. The sampling rate is equal to $10f_0$ where f_0 is the frequency of the sinusoid. The predictor has a single coefficient denoted by w_1.

(a) Determine the optimum value of w_1 required to minimize the prediction error variance.

(b) Determine the minimum value of the prediction error variance.

3.32 A stationary process $X(t)$ has the following values for its autocorrelation function:

$$R_X(0) = 1$$
$$R_X(1) = 0.8$$
$$R_X(2) = 0.6$$
$$R_X(3) = 0.4$$

(a) Calculate the coefficients of an optimum linear predictor involving the use of three unit-delays.

(b) Calculate the variance of the resulting prediction error.

3.33 Repeat the calculations of Problem 3.32, but this time use a linear predictor with two unit-delays. Compare the performance of this second optimum linear predictor with that considered in Problem 3.32.

Differential Pulse-Code Modulation

3.34 A DPCM system uses a linear predictor with a single tap. The normalized autocorrelation function of the input signal for a lag of one sampling interval is 0.75. The predictor is

CHAPTER 4

BASEBAND PULSE TRANSMISSION

This chapter discusses the transmission of digital data over a baseband channel with emphasis on the following topics:

- *The matched filter, which is the optimum system for detecting a known signal in additive white Gaussian noise.*
- *Calculation of the bit error rate due to the presence of channel noise.*
- *Intersymbol interference, which arises when the channel is dispersive as is commonly the case in practice.*
- *Nyquist's criterion for distortionless baseband data transmission.*
- *Correlative-level coding or partial-response signaling for combatting the effects of intersymbol interference.*
- *Digital subscriber lines.*
- *Equalization of a dispersive baseband channel.*
- *The eye pattern for displaying the combined effects of intersymbol interference and channel noise in data transmission.*

4.1 *Introduction*

In Chapter 3 we described techniques for converting an analog information-bearing signal into digital form. There is another way in which digital data can arise in practice: The data may represent the output of a source of information that is inherently discrete in nature (e.g., a digital computer). In this chapter we study the transmission of digital data (of whatever origin) over a *baseband channel*.[1] Data transmission over a band-pass channel using modulation is covered in Chapter 6.

Digital data have a broad spectrum with a significant low-frequency content. Baseband transmission of digital data therefore requires the use of a low-pass channel with a bandwidth large enough to accommodate the essential frequency content of the data stream. Typically, however, the channel is *dispersive* in that its frequency response deviates from that of an ideal low-pass filter. The result of data transmission over such a channel is that each received pulse is affected somewhat by adjacent pulses, thereby giving rise to a common form of interference called *intersymbol interference* (ISI). Intersymbol interference is a major source of bit errors in the reconstructed data stream at the receiver output. To correct for it, control has to be exercised over the pulse shape in the overall system. Thus much of the material covered in this chapter is devoted to *pulse shaping* in one form or another.

where we have made use of the sifting property of the delta function. Since from Equation (4.46) we have $p(0) = 1$, it follows from Equations (4.50) and (4.52) that the condition for zero intersymbol interference is satisfied if

$$\sum_{n=-\infty}^{\infty} P(f - nR_b) = T_b \tag{4.53}$$

We may now state the *Nyquist criterion*[4] *for distortionless baseband transmission* in the absence of noise: *The frequency function $P(f)$ eliminates intersymbol interference for samples taken at intervals T_b provided that it satisfies Equation (4.53).* Note that $P(f)$ refers to the overall system, incorporating the transmit filter, the channel, and the receive filter in accordance with Equation (4.47).

■ IDEAL NYQUIST CHANNEL

The simplest way of satisfying Equation (4.53) is to specify the frequency function $P(f)$ to be in the form of a *rectangular function* as shown by

$$P(f) = \begin{cases} \dfrac{1}{2W}, & -W < f < W \\ 0, & |f| > W \end{cases} = \frac{1}{2W}\,\mathrm{rect}\left(\frac{f}{2W}\right) \tag{4.54}$$

where $\mathrm{rect}(f)$ stands for a *rectangular function* of unit amplitude and unit support centered on $f = 0$, and the overall system bandwidth W is defined by

$$W = \frac{R_b}{2} = \frac{1}{2T_b} \tag{4.55}$$

According to the solution described by Equations (4.54) and (4.55), no frequencies of absolute value exceeding half the bit rate are needed. Hence, from Fourier-transform pair 2 of Table A6.3 we find that a signal waveform that produces zero intersymbol interference is defined by the *sinc function*:

$$p(t) = \frac{\sin(2\pi Wt)}{2\pi Wt} = \mathrm{sinc}(2Wt) \tag{4.56}$$

The special value of the bit rate $R_b = 2W$ is called the *Nyquist rate*, and W is itself called the *Nyquist bandwidth*. Correspondingly, the ideal baseband pulse transmission system described by Equation (4.54) in the frequency domain or, equivalently, Equation (4.56) in the time domain, is called the *ideal Nyquist channel*.

Figures 4.8*a* and 4.8*b* show plots of $P(f)$ and $p(t)$, respectively. In Figure 4.8*a*, the normalized form of the frequency function $P(f)$ is plotted for positive and negative frequencies. In Figure 4.8*b*, we have also included the signaling intervals and the corresponding centered sampling instants. The function $p(t)$ can be regarded as the impulse response of an ideal low-pass filter with passband magnitude response $1/2W$ and bandwidth W. The function $p(t)$ has its peak value at the origin and goes through zero at integer multiples of the bit duration T_b. It is apparent that if the received waveform $y(t)$ is sampled at the

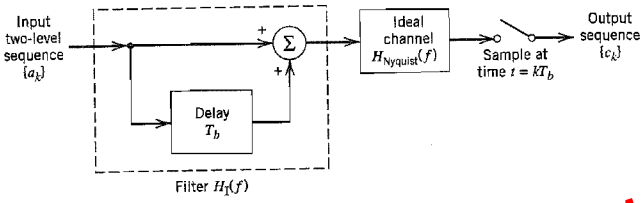

FIGURE 4.11 Duobinary signaling scheme.

input of this filter, we get two unit impulses spaced T_b seconds apart at the filter output. We may therefore express the duobinary coder output c_k as the sum of the present input pulse a_k and its previous value a_{k-1}, as shown by

$$c_k = a_k + a_{k-1} \tag{4.66}$$

One of the effects of the transformation described by Equation (4.66) is to change the input sequence $\{a_k\}$ of uncorrelated two-level pulses into a sequence $\{c_k\}$ of correlated three-level pulses. This correlation between the adjacent pulses may be viewed as introducing intersymbol interference into the transmitted signal in an artificial manner. However, the intersymbol interference so introduced is under the designer's control, which is the basis of correlative coding.

An ideal delay element, producing a delay of T_b seconds, has the frequency response $\exp(-j2\pi fT_b)$, so that the frequency response of the simple delay-line filter in Figure 4.11 is $1 + \exp(-j2\pi fT_b)$. Hence, the overall frequency response of this filter connected in cascade with an ideal Nyquist channel is

$$\begin{aligned} H_I(f) &= H_{\text{Nyquist}}(f)[1 + \exp(-j2\pi fT_b)] \\ &= H_{\text{Nyquist}}(f)[\exp(j\pi fT_b) + \exp(-j\pi fT_b)]\exp(-j\pi fT_b) \\ &= 2H_{\text{Nyquist}}(f)\cos(\pi fT_b)\exp(-j\pi fT_b) \end{aligned} \tag{4.67}$$

where the subscript I in $H_I(f)$ indicates the pertinent class of partial response. For an ideal Nyquist channel of bandwidth $W = 1/2T_b$, we have (ignoring the scaling factor T_b)

$$H_{\text{Nyquist}}(f) = \begin{cases} 1, & |f| \le 1/2T_b \\ 0, & \text{otherwise} \end{cases} \tag{4.68}$$

Thus the overall frequency response of the duobinary signaling scheme has the form of a half-cycle cosine function, as shown by

$$H_I(f) = \begin{cases} 2\cos(\pi fT_b)\exp(-j\pi fT_b), & |f| \le 1/2T_b \\ 0, & \text{otherwise} \end{cases} \tag{4.69}$$

for which the magnitude response and phase response are as shown in Figures 4.12*a* and 4.12*b*, respectively. An advantage of this frequency response is that it can be easily approximated, in practice, by virtue of the fact that there is continuity at the band edges.

From the first line in Equation (4.67) and the definition of $H_{\text{Nyquist}}(f)$ in Equation (4.68), we find that the impulse response corresponding to the frequency response $H_I(f)$

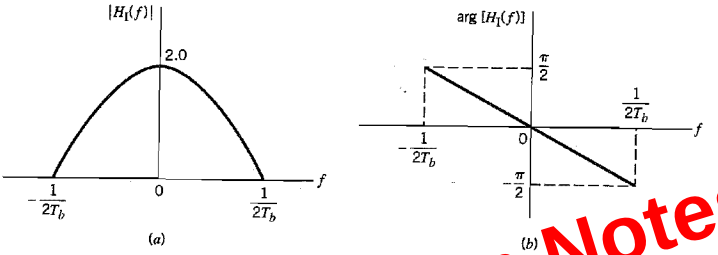

FIGURE 4.12 Frequency response of the duobinary conversion filter. (*a*) Magnitude response. (*b*) Phase response.

consists of two sinc (Nyquist) pulses that are time-displaced by T_b seconds with respect to each other, as shown below (except for a scaling factor):

$$\begin{aligned} h_I(t) &= \frac{\sin(\pi t/T_b)}{\pi t/T_b} + \frac{\sin[\pi(t - T_b)/T_b]}{\pi(t - T_b)/T_b} \\ &= \frac{\sin(\pi t/T_b)}{\pi t/T_b} - \frac{\sin(\pi t/T_b)}{\pi(t - T_b)/T_b} \\ &= \frac{T_b^2 \sin(\pi t/T_b)}{\pi t(T_b - t)} \end{aligned} \tag{4.70}$$

The impulse response $h_I(t)$ is plotted in Figure 4.13, where we see that it has only *two* distinguishable values at the sampling instants. The form of $h_I(t)$ shown here explains why we also refer to this type of correlative coding as partial-response signaling. The response to an input pulse is spread over more than one signaling interval; stated in another way, the response in any signaling interval is "partial." Note also that the tails of $h_I(t)$ decay as $1/|t|^2$, which is a faster rate of decay than the $1/|t|$ encountered in the ideal Nyquist channel.

The original two-level sequence $\{a_k\}$ may be detected from the duobinary-coded sequence $\{c_k\}$ by invoking the use of Equation (4.66). Specifically, let $\hat{a}_k$ represent the *estimate* of the original pulse a_k as conceived by the receiver at time $t = kt_b$. Then, subtracting the previous estimate $\hat{a}_{k-1}$ from c_k, we get

$$\hat{a}_k = c_k - \hat{a}_{k-1} \tag{4.71}$$

It is apparent that if c_k is received without error and if also the previous estimate $\hat{a}_{k-1}$ at time $t = (k - 1)T_b$ corresponds to a correct decision, then the current estimate $\hat{a}_k$ will be

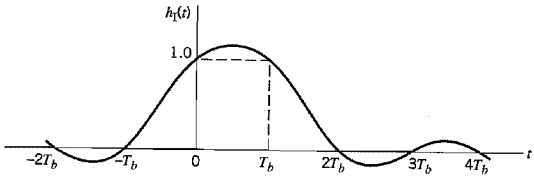

FIGURE 4.13 Impulse response of the duobinary conversion filter.

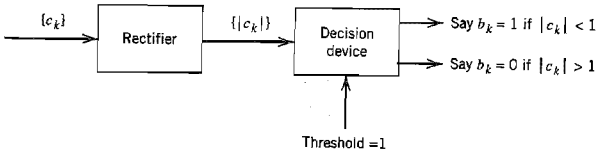

FIGURE 4.15 Detector for recovering original binary sequence from the precoded duobinary coder output.

When $|c_k| = 1$, the receiver simply makes a random guess in favor of symbol 1 or 0. According to this decision rule, the detector consists of a rectifier, the output of which is compared in a decision device to a threshold of 1. A block diagram of the detector is shown in Figure 4.15. A useful feature of this detector is that no knowledge of any input sample other than the present one is required. Hence, error propagation cannot occur in the detector of Figure 4.15.

EXAMPLE 4.3 Duobinary Coding with Precoding

Consider the binary data sequence 0010110. To proceed with the precoding of this sequence, which involves feeding the precoder output back to the input, we add an extra bit to the precoder output. This extra bit is chosen arbitrarily to be 1. Hence, using Equation (4.73), we find that the sequence $\{d_k\}$ at the precoder output is as shown in row 2 of Table 4.1. The polar representation of the precoded sequence $\{d_k\}$ is shown in row 3 of Table 4.1. Finally, using Equation (4.74), we find that the duobinary coder output has the amplitude levels given in row 4 of Table 4.1.

To detect the original binary sequence, we apply the decision rule of Equation (4.76), and so obtain the binary sequence given in row 5 of Table 4.1. This latter result shows that, in the absence of noise, the original binary sequence is detected correctly. ◀

MODIFIED DUOBINARY SIGNALING

In the duobinary signaling technique the frequency response $H(f)$, and consequently the power spectral density of the transmitted pulse, is nonzero at the origin. This is considered to be an undesirable feature in some applications, since many communications channels cannot transmit a DC component. We may correct for this deficiency by using the *class IV partial response* or *modified duobinary* technique, which involves a correlation span of two binary digits. This special form of correlation is achieved by subtracting amplitude-modulated pulses spaced $2T_b$ seconds apart, as indicated in the block diagram of Figure

TABLE 4.1 *Illustrating Example 4.3 on duobinary coding*

Binary sequence $\{b_k\}$		0	0	1	0	1	1	0
Precoded sequence $\{d_k\}$	1	1	1	0	0	1	0	0
Two-level sequence $\{a_k\}$	+1	+1	+1	−1	−1	+1	−1	−1
Duobinary coder output $\{c_k\}$		+2	+2	0	−2	0	0	−2
Binary sequence obtained by applying decision rule of Eq. (4.76)		0	0	1	0	1	1	0

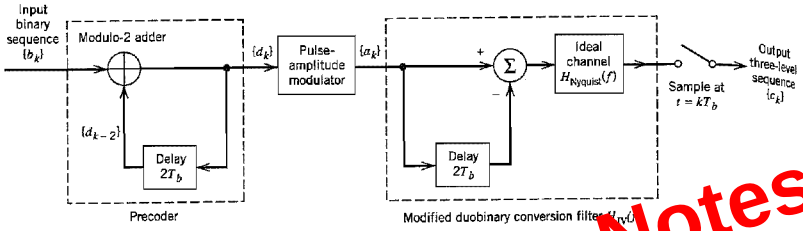

FIGURE 4.16 Modified duobinary signaling scheme.

4.16. The precoder involves a delay of $2T_b$ seconds. The output of the modified duobinary conversion filter is related to the input two-level sequence $\{a_k\}$ at the pulse-amplitude modulator output as follows:

$$c_k = a_k - a_{k-2} \tag{4.77}$$

Here, again, we find that a three-level signal is generated. With $a_k = \pm 1$, we find that c_k takes on one of three values: $+2$, 0, and -2.

The overall frequency response of the delay-line filter connected in cascade with an ideal Nyquist channel, as in Figure 4.16, is given by

$$\begin{aligned} H_{\text{IV}}(f) &= H_{\text{Nyquist}}(f)[1 - \exp(-j4\pi f T_b)] \\ &= 2jH_{\text{Nyquist}}(f)\sin(2\pi f T_b)\exp(-j2\pi f T_b) \end{aligned} \tag{4.78}$$

where the subscript IV in $H_{\text{IV}}(f)$ indicates the pertinent class of partial response and $H_{\text{Nyquist}}(f)$ is as defined in Equation (4.68). We therefore have an overall frequency response in the form of a half-cycle sine function, as shown by

$$H_{\text{IV}}(f) = \begin{cases} 2j\sin(2\pi f T_b)\exp(-j2\pi f T_b), & |f| \le 1/2T_b \\ 0, & \text{elsewhere} \end{cases} \tag{4.79}$$

The corresponding magnitude response and phase response of the modified duobinary coder are shown in Figures 4.17*a* and 4.17*b*, respectively. A useful feature of the modified duobinary coder is the fact that its output has no DC component. Note also that this

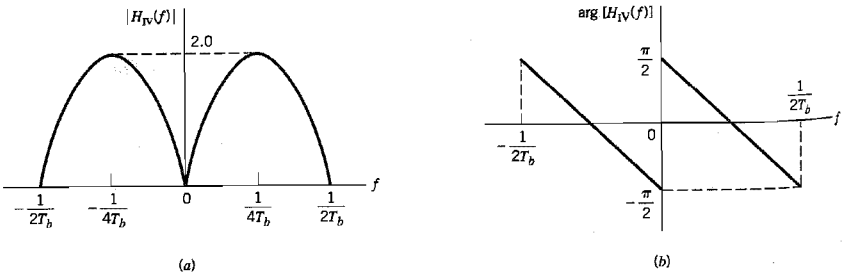

FIGURE 4.17 Frequency response of the modified duobinary conversion filter. (*a*) Magnitude response. (*b*) Phase response.

second form of correlative-level coding exhibits the same continuity at the band edges as in duobinary signaling.

From the first line of Equation (4.78) and the definition of $H_{\text{Nyquist}}(f)$ in Equation (4.68), we find that the impulse response of the modified duobinary coder consists of two sinc (Nyquist) pulses that are time-displaced by $2T_b$ seconds with respect to each other, as shown by (except for a scaling factor)

$$\begin{aligned} h_{\text{IV}}(t) &= \frac{\sin(\pi t/T_b)}{\pi t/T_b} - \frac{\sin[\pi(t - 2T_b)/T_b]}{\pi(t - 2T_b)/T_b} \\ &= \frac{\sin(\pi t/T_b)}{\pi t/T_b} - \frac{\sin(\pi t/T_b)}{\pi(t - 2T_b)/T_b} \\ &= \frac{2T_b^2 \sin(\pi t/T_b)}{\pi(2T_b - t)} \end{aligned} \tag{4.80}$$

This impulse response is plotted in Figure 4.18, which shows that it has *three* distinguishable levels at the sampling instants. Note also that, as with duobinary signaling, the tails of $h_{\text{IV}}(t)$ for the modified duobinary signaling decay as $1/|t|^2$.

To eliminate the possibility of error propagation in the modified duobinary system, we use a precoding procedure similar to that used for the duobinary case. Specifically, prior to the generation of the modified duobinary signal, a modulo-two logical addition is used on signals $2T_b$ seconds apart, as shown by (see the front end of Figure 4.16)

$$\begin{aligned} d_k &= b_k \oplus d_{k-2} \\ &= \begin{cases} \text{symbol 1} & \text{if either symbol } b_k \text{ or symbol } d_{k-2} \text{ (but not both) is 1} \\ \text{symbol 0} & \text{otherwise} \end{cases} \end{aligned} \tag{4.81}$$

where $\{b_k\}$ is the incoming binary data sequence and $\{d_k\}$ is the sequence at the precoder output. The precoded sequence $\{d_k\}$ thus produced is then applied to a pulse-amplitude modulator and then to the modified duobinary conversion filter.

In Figure 4.16, the output digit c_k equals -2, 0, or $+2$, assuming that the pulse-amplitude modulator uses a polar representation for the precoded sequence $\{d_k\}$. Also we find that the detected digit $\hat{b}_k$ at the receiver output may be extracted from c_k by disregarding the polarity of c_k. Specifically, we may formulate the following decision rule:

$$\begin{aligned} &\text{If } |c_k| > 1, && \text{say symbol } b_k \text{ is 1} \\ &\text{If } |c_k| < 1, && \text{say symbol } b_k \text{ is 0} \end{aligned} \tag{4.82}$$

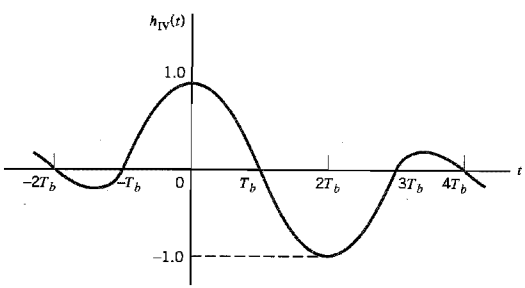

FIGURE 4.18 Impulse response of the modified duobinary conversion filter.

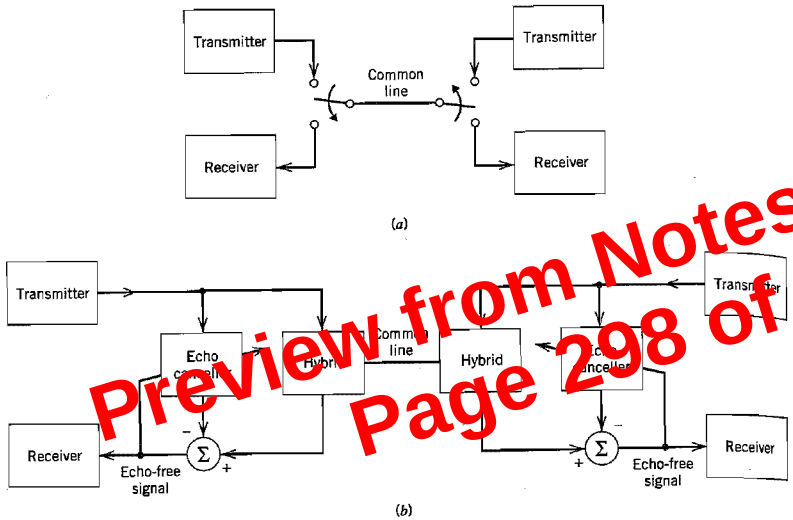

FIGURE 4.22 Full-duplex operation using (*a*) time compression multiplexing, and (*b*) echo-cancellation.

Figure 4.22*a*. To account for propagation time across the line, a guard time is inserted between individual bursts of data. Accordingly, the line rate is slightly greater than twice the data rate.

2. *Echo-cancellation mode*, which supports the simultaneous flow of data along the common line in both directions. For this form of transmission to be feasible, each *transceiver* (transmitter/receiver) includes a hybrid for two purposes: the separation of the transmitted signal from the received signal and the two-to-four-wire conversion, as shown in Figure 4.22*b*. The *hybrid*, or more precisely, the *hybrid transformer*, is basically a bridge circuit with three ports (terminal pairs), as depicted in Figure 4.23. If the bridge is not perfectly balanced, the transmitter port of the hybrid

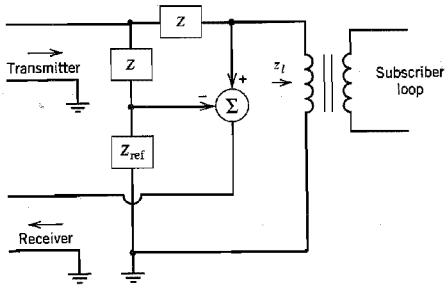

FIGURE 4.23 Simplified circuit of hybrid transformer. For the bridge to be balanced, the reference impedance Z_{ref} should equal the line impedance z_l.

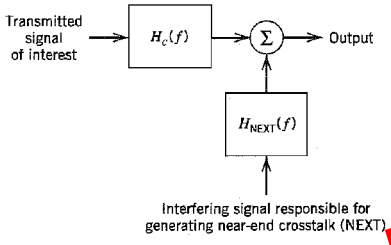

FIGURE 4.25 Model of twisted-pair channel.

2. *Far-end crosstalk (FEXT)*, which is generated by transmitters located further away from the receiver, as illustrated in Figure 4.24*b*.

FEXT naturally suffers the same line loss as the signal, whereas NEXT does not. Accordingly, in the echo cancellation scheme of Figure 4.24*b*, where signals travel in both directions in the cable, NEXT will be much stronger than FEXT. Henceforth, we ignore the effect of FEXT.

Indeed, near-end crosstalk and intersymbol interference are the two most important factors in determining the performance of a digital subscriber loop. Figure 4.25 shows the model of a twisted-pair channel dominated by these two impairments. Since all twisted pairs are usually transmitting similar signals, we may model the NEXT as a signal with the same power spectral density as the transmitted signal passing through a *crosstalk* frequency response $H_{NEXT}(f)$, which is approximated by

$$H_{NEXT}(f) = \beta f^{3/2} \tag{4.87}$$

where β is a constant of the cable. The interesting point to note from Figure 4.25 is that both the transmitted signal and the interfering signal have the same power spectral density; they differ from each other merely in their associated frequency responses, as shown in Equations (4.85) and (4.87), respectively. When the model described herein is used for simulation study, the transmitted signal is represented by a random data sequence, while the interference is represented by a Gaussian noise sequence.

■ LINE CODES FOR DIGITAL SUBSCRIBER LINES

Now that we have identified the major transmission impairments, we may describe the desirable features the spectrum of a transmitted signal should exhibit:

1. The power spectral density of the transmitted signal should be zero at zero frequency, since no DC transmission through a hybrid transformer is possible.
2. The power spectral density of the transmitted signal should be low at high frequencies for the following reasons:
   - Transmission attenuation in a twisted pair is most severe at high frequencies.
   - Crosstalk between adjacent twisted pairs increases dramatically at high frequencies because of increased capacitive coupling. In this regard, recall that the impedance of a capacitor is inversely proportional to frequency.

   To satisfy these desirable properties, we have to be careful in choosing the *line code* that maps the incoming stream of data bits into electrical pulses for transmission on the line. Various possibilities, each with its own advantages and disadvantages, exist

for such a choice. The list of potential candidates for line codes includes the following:

- *Manchester code*, which is simple and has zero DC component. Its disadvantage is the occupation of a large spectrum, which makes it vulnerable to near-end crosstalk and intersymbol interference. (The Manchester code was discussed in Section 3.7.)
- *Modified duobinary code*, which has zero DC, is moderately spectrally efficient, and causes minimal intersymbol interference. However, simulation studies of the crosstalk performance of the modified duobinary code have shown that its immunity to near-end crosstalk and intersymbol interference is about 6 to 10 dB poorer than that of block codes on worst-case subscriber loops. (The modified duobinary code was discussed in Section 4.6.)
- *Bipolar code*, in which successive 1s are represented alternately by positive and negative but equal levels and symbol 0 is represented by a zero level. Bipolar signaling has zero DC. Computer simulations have shown that its near-end crosstalk and intersymbol interference performance is slightly inferior to the modified duobinary code on all digital subscriber loops. (The bipolar code, also known as the *alternate mark inversion (AMI) codes*, was discussed in Section 3.7.)
- *2B1Q code*, which stands for two binary digits encoded into one quaternary symbol. This code is a block code representing a four-level PAM signal, as illustrated in Figure 4.20. Assuming that symbols 1 and 0 are equiprobable, the 2B1Q code has zero DC on the average. Moreover, among all the line codes considered herein, it offers the greatest baud reduction, and the best performance with respect to near-end crosstalk and intersymbol interference.

It is because of the desirable properties of the 2BIQ code compared to the Manchester code, modified duobinary code, the bipolar code, and other line codes not mentioned here,[7] that the 2B1Q code has been adopted as the North American standard for digital subscriber loops.

Using the 2B1Q as the line code and VLSI implementation of a transceiver that incorporates adaptive equalizers and echo cancellers, it is possible to achieve a bit error rate of 10^{-7} operating full duplex at 160 kb/s on the vast majority of twisted-pair subscriber lines. A bit error rate of 10^{-7} with 12 dB noise margin, when 1 percent worst-case NEXT is present, is an accepted performance criterion for digital subscriber lines. *Noise margin* is the amount of receiver noise (including uncancelled echo) that can be tolerated without exceeding the 10^{-7} error rate.

Asymmetric Digital Subscriber Lines

Another important type of DSL is the *asymmetric digital subscriber line* (ADSL), which is a local transmission system designed to simultaneously support three services on a single twisted-wire pair:

1. Data transmission *downstream* (toward the subscriber) at bit rates of up to 9 Mb/s.
2. Data transmission *upstream* (away from the subscriber) at bit rates of up to 1 Mb/s.
3. Plain old telephone service (POTS).

The downstream and upstream bit rates depend on the length of the twisted pair used to do the transmission. The DSL is said to be "asymmetric" because the downstream bit rate is much higher than the upstream bit rate. Analog voice is transmitted at baseband frequencies and combined with the passband transmissions of downstream and upstream

where $c(t) \rightleftharpoons C(f)$, $q(t) \rightleftharpoons Q(f)$, and $R_q \rightleftharpoons S_q(f)$. Solving Equation (4.109) for $C(f)$, we get

$$C(f) = \frac{Q^*(f)}{S_q(f) + \frac{N_0}{2}} \tag{4.110}$$

In Problem 4.33 it is shown that the power spectral density of the sequence $\{q(kT_b)\}$ can be expressed as

$$S_q(f) = \frac{1}{T_b} \sum_k \left| Q\left(f + \frac{k}{T_b}\right) \right|^2 \tag{4.111}$$

which means that the frequency response $C(f)$ of the optimum linear receiver is *periodic* with period T_b. Equation (4.110) suggests the interpretation of the optimum linear receiver as the cascade connection of two basic components:[8]

- A *matched filter* whose impulse response is $q(-t)$, where $q(t) = g(t) \bigstar h(t)$.
- A *transversal (tapped-delay-line) equalizer* whose frequency response is the inverse of the periodic function $S_q(f) + (N_0/2)$.

To implement Equation (4.110) exactly we need an equalizer of infinite length. In practice, we may approximate the optimum solution by using an equalizer with a finite set of coefficients $\{c_k\}_{k=-N}^{N}$, provided N is large enough. Thus the receiver takes the form shown in Figure 4.27. Note that the block labeled z^{-1} in Figure 4.27 introduces a delay equal to T_b, which means that the tap spacing of the equalizer is exactly the same as the bit duration T_b. An equalizer so configured is said to be *synchronous* with the transmitter.

■ PRACTICAL CONSIDERATIONS

The mmse receiver of Figure 4.27 works well in the laboratory, where we have access to the system to be equalized, in which case we may determine a transversal equalizer characterized by the set of coefficients $\{c_k\}_{k=-N}^{N}$, which provides an adequate approximation to the frequency response $C(f)$ of Equation (4.110). In a real-life telecommunications environment, however, the channel is usually time varying. For example, in a public

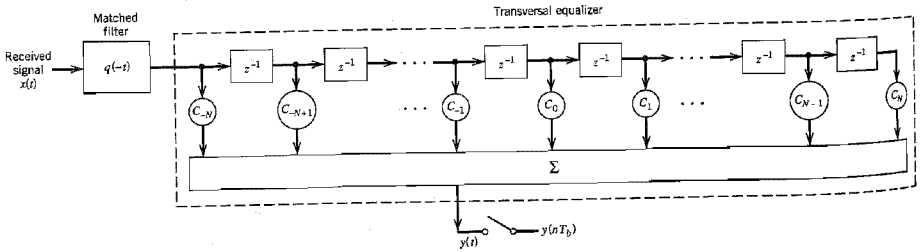

FIGURE 4.27 Optimum linear receiver consisting of the cascade connection of matched filter and transversal equalizer.

When the effect of intersymbol interference is severe, traces from the upper portion of the eye pattern cross traces from the lower portion, with the result that the eye is completely closed. In such a situation, it is impossible to avoid errors due to the combined presence of intersymbol interference and noise in the system.

In the case of an M-ary system, the eye pattern contains $(M - 1)$ eye openings stacked up vertically one on the other, where M is the number of discrete amplitude levels used to construct the transmitted signal. In a strictly linear system with truly random data, all these eye openings would be identical.

In the next two experiments, we use computer simulations to study the eye patterns for a quaternary ($M = 4$) baseband PAM transmission system under noiseless, noisy, and band-limited conditions. The effect of channel nonlinearity on eye patterns is discussed in Problem 4.38.

Experiment 1: Effect of Channel Noise

Figure 4.34 shows the eye diagram of the system under idealized conditions: no channel noise and no bandwidth limitation. The source symbols used are randomly generated on a computer, with raised cosine pulse-shaping. The system parameters used for the generation of the eye diagram are as follows: Nyquist bandwidth $W = 0.5$ Hz, rolloff factor $\alpha = 0.5$, and symbol duration $T = T_b \log_2 M = 2T_b$. The openings in Figure 4.34 are perfect, indicating reliable operation of the system. Note that this figure has $M - 1 = 3$ openings.

Figures 4.34*b* and 4.34*c* show the eye diagrams for the system, but this time with channel noise corrupting the received signal. These two figures were simulated for signal-to-noise ratio SNR = 20 dB and 10 dB, respectively, with the SNR being measured at the channel output. When SNR = 20 dB the effect of channel noise is hardly discernible in Figure 4.34*b*, but when SNR = 10 dB the openings of the eye diagram in Figure 4.34*c* are barely visible.

Experiment 2: Effect of Bandwidth Limitation

Figures 4.35*a* and 4.35*b* show the eye diagrams for the quaternary system using the same parameters as before, but this time under a bandwidth-limited condition and a noiseless channel. Specifically, the channel is now modeled by a low-pass *Butterworth filter*, whose squared magnitude response is defined by

$$|H(f)|^2 = \frac{1}{1 + (f/f_0)^{2N}}$$

where N is the *order* of the filter, and f_0 is its 3-dB cutoff frequency. For the computer experiment described in Figure 4.35*a*, the following values are used:

$$N = 25 \text{ and } f_0 = 0.975 \text{ Hz}$$

The bandwidth required by the PAM trasmission system is computed to be

$$B_T = W(1 + \alpha) = 0.75 \text{ Hz}$$

Although the channel bandwidth (i.e., cutoff frequency) is greater than absolutely necessary, its effect on the passband is observed as a decrease in the size of the eye openings compared to those in Figure 4.34*a*. Instead of the distinct values at time $t = 1$ s (as shown in Figure 4.34*a*), now there is a blurred region.

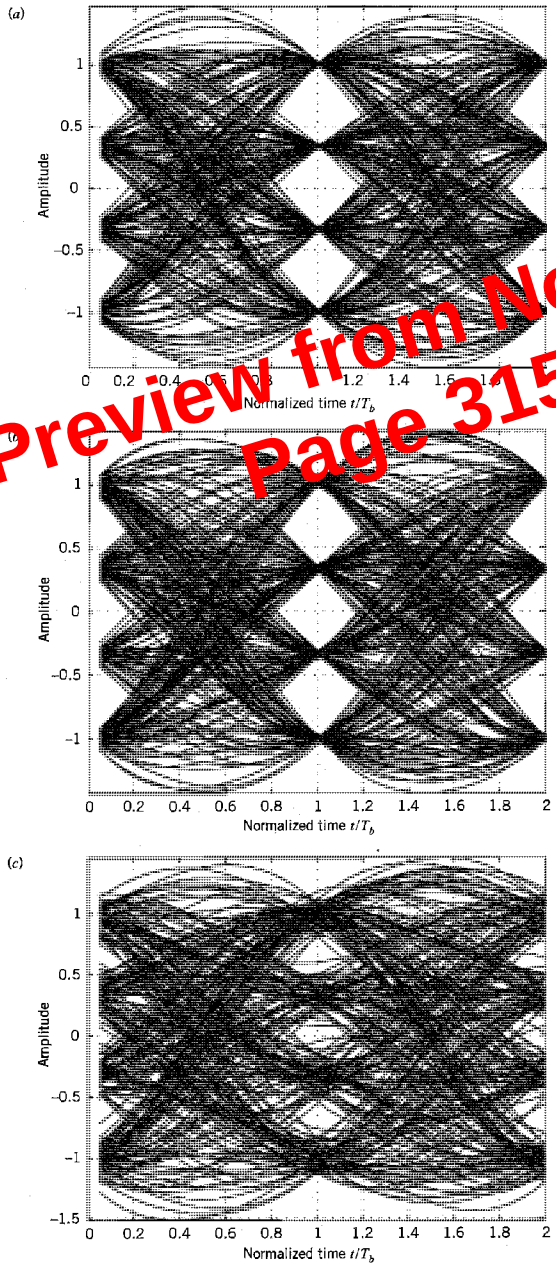

FIGURE 4.34 (*a*) Eye diagram for noiseless quaternary system. (*b*) Eye diagram for quaternary system with SNR = 20 dB. (*c*) Eye diagram for quaternary system with SNR = 10 dB.

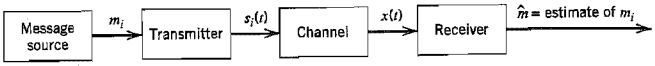

FIGURE 5.1 Block diagram of a generic digital communication system.

The channel is assumed to have two characteristics:

1. The channel is *linear*, with a bandwidth that is wide enough to accommodate the transmission of signal $s_i(t)$ with negligible or no distortion.
2. The channel noise, $w(t)$, is the sample function of a zero-mean white Gaussian noise *process*. The reasons for this second assumption are that it makes receiver calculations tractable, and it is a reasonable description of the type of noise present in many practical communication systems.

We refer to such a channel as an *additive white Gaussian noise* (AWGN) *channel*. Accordingly, we may express the *received signal* as

$$x(t) = s_i(t) + w(t), \qquad \begin{cases} 0 \le t \le T \\ i = 1, 2, \ldots, M \end{cases} \tag{5.3}$$

and thus model the channel as in Figure 5.2.

The receiver has the task of observing the received signal $x(t)$ for a duration of T seconds and making a best *estimate* of the transmitted signal $s_i(t)$ or, equivalently, the symbol m_i. However, owing to the presence of channel noise, this decision-making process is statistical in nature, with the result that the receiver will make occasional errors. The requirement is therefore to design the receiver so as to minimize the *average probability of symbol error*, defined as

$$P_e = \sum_{i=1}^{M} p_i \, P(\hat{m} \neq m_i \,|\, m_i) \tag{5.4}$$

where m_i is the transmitted symbol, $\hat{m}$ is the estimate produced by the receiver, and $P(\hat{m} \neq m_i \,|\, m_i)$ is the conditional error probability given that the ith symbol was sent. The resulting receiver is said to be *optimum in the minimum probability of error* sense.

This model provides a basis for the design of the optimum receiver, for which we will use geometric representation of the known set of transmitted signals, $\{s_i(t)\}$. This method, discussed in Section 5.2, provides a great deal of insight, with considerable simplification of detail.

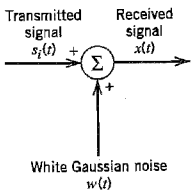

FIGURE 5.2 Additive white Gaussian noise (AWGN) model of a channel.

Case 1

In the first case, we assume that it is possible to perform the mapping from binary to M-ary symbols in such a way that the two binary M-tuples corresponding to any pair of adjacent symbols in the M-ary modulation scheme differ in only one bit position. This mapping constraint is satisfied by using a *Gray code*. When the probability of symbol error P_e is acceptably small, we find that the probability of mistaking one symbol for either one of the two "nearest" symbols is much greater than any other kind of symbol error. Moreover, given a symbol error, the most probable number of bit errors is one, subject to the aforementioned mapping constraint. Since there are $\log_2 M$ bits per symbol, it follows that the average probability of symbol error is related to the bit error rate as follows:

$$\begin{aligned} P_e &= P\left(\bigcup_{i=1}^{\log_2 M} \{i\text{th bit is in error}\}\right) \\ &\leq \sum_{i=1}^{\log_2 M} P(i\text{th bit is in error}) \\ &= \log_2 M \cdot (\text{BER}) \end{aligned} \tag{5.97}$$

We also note that

$$P_e \geq P(i\text{th bit is in error}) = \text{BER} \tag{5.98}$$

It follows therefore that the bit error rate is bounded as follows:

$$\frac{P_e}{\log_2 M} \leq \text{BER} \leq P_e \tag{5.99}$$

Case 2

Let $M = 2^K$, where K is an integer. We assume that all symbol errors are equally likely and occur with probability

$$\frac{P_e}{M-1} = \frac{P_e}{2^K - 1}$$

where P_e is the average probability of symbol error. What is the probability that the ith bit in a symbol is in error? Well, there are 2^{K-1} cases of symbol error in which this particular bit is changed, and there are 2^{K-1} cases in which it is not changed. Hence, the bit error rate is

$$\text{BER} = \left(\frac{2^{K-1}}{2^K - 1}\right) P_e \tag{5.100}$$

or, equivalently,

$$\text{BER} = \left(\frac{M/2}{M-1}\right) P_e \tag{5.101}$$

Note that for large M, the bit error rate approaches the limiting value of $P_e/2$. The same idea described here also shows that bit errors are not independent, since we have

$$P(i\text{th and } j\text{th bits are in error}) = \frac{2^{K-2}}{2^K - 1} P_e \neq (\text{BER})^2$$

5.8 Summary and Discussion

The primary goal of the material presented in this chapter is the formulation of a systematic procedure for the analysis and design of a digital communication receiver in the presence of *additive white Gaussian noise* (AWGN). The procedure, known as *maximum likelihood detection*, decides which particular transmitted symbol is the most likely cause of the noisy signal observed at the channel output. The approach that led to the formulation of the maximum likelihood detector (receiver) is called *signal-space analysis*. The basic idea of the approach is to represent each member of a set of transmitted signals by an N-dimensional vector, where N is the number of orthonormal basis functions needed for a unique geometric representation of the transmitted signals. The set of signal vectors so formed defines a *signal constellation* in an N-dimensional *signal space*.

For a given signal constellation, the average probability of symbol error P_e incurred in maximum likelihood signal detection over an AWGN channel is invariant to rotation of the signal constellation as well as its translation. However, except for a few simple (but important) cases, the numerical calculation of P_e is an impractical proposition. To overcome this difficulty, the customary practice is to resort to the use of bounds that lend themselves to computation in a straightforward manner. In this context, we described the *union bound* that follows directly from the signal-space diagram. The union bound is based on an intuitively satisfying idea: The probability of symbol error P_e is dominated by the nearest neighbors to the transmitted signal. The results obtained using the union bound are usually fairly accurate when the signal-to-noise ratio is high.

With the material on signal-space analysis and related issues on hand, we are well-equipped to study passband data transmission systems, which we do in Chapter 6.

Notes and References

1. The geometric representation of signals was first developed by Kotel'nikov in 1947: V. A. Kotel'nikov, *The Theory of Optimum Noise Immunity* (Dover Publications, 1960), which is a translation of the original doctoral dissertation presented in January 1947 before the Academic Council of the Molotov Energy Institute in Moscow. In particular, see Part II of the book. This method was subsequently brought to fuller fruition in the classic book by Wozencraft and Jacobs (1965). Signal-space analysis is also discussed in Cioffi (1998), Anderson (1999), and Proakis (1995).

2. In Section 5.7, we derived the union bound on the average probability of symbol error; the classic reference for this bound is Wozencraft and Jacobs (1965). For the derivation of tighter bounds, see Viterbi and Omura (1979, pp. 58–59).

3. In Chapter 4, we used the following upper bound on the complementary error function

$$\operatorname{erfc}(u) < \frac{\exp(-u^2)}{\sqrt{\pi}u}$$

For large positive u, a second bound on the complementary error function is obtained by omitting the multiplying factor $1/u$ in the above upper bound, as shown by

$$\operatorname{erfc}(u) < \frac{\exp(-u^2)}{\sqrt{\pi}}$$

It is this second upper bound that is used in Equation (5.97).

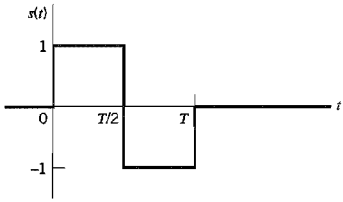

FIGURE P5.13

5.14 In the *Bayes test*, applied to a binary hypothesis-testing problem where we have to choose one of two possible hypotheses H_0 and H_1, we minimize the *risk* R defined by

$$\begin{aligned} R = {} & C_{00}p_0P(\text{say } H_0|H_0 \text{ is true}) \\ & + C_{10}p_0P(\text{say } H_1|H_0 \text{ is true}) \\ & + C_{11}p_1P(\text{say } H_1|H_1 \text{ is true}) \\ & + C_{01}p_1P(\text{say } H_0|H_1 \text{ is true}) \end{aligned}$$

The terms C_{00}, C_{10}, C_{11}, and C_{01} denote the costs assigned to the four possible outcomes of the experiment: The first subscript indicates the hypothesis chosen, and the second the hypothesis that is true. Assume that $C_{10} > C_{00}$ and $C_{01} > C_{11}$. The p_0 and p_1 denote the *a priori* probabilities of hypotheses H_0 and H_1, respectively.

(a) Given the observation vector $\mathbf{x}$, show that the partitioning of the observation space so as to minimize the risk R leads to the *likelihood ratio test*:

$$\begin{aligned} &\text{say } H_0 \text{ if } \Lambda(\mathbf{x}) < \lambda \\ &\text{say } H_1 \text{ if } \Lambda(\mathbf{x}) > \lambda \end{aligned}$$

where $\Lambda(\mathbf{x})$ is the *likelihood ratio*

$$\Lambda(\mathbf{x}) = \frac{f_{\mathbf{X}}(\mathbf{x}|H_1)}{f_{\mathbf{X}}(\mathbf{x}|H_0)}$$

and λ is the *threshold* of the test defined by

$$\lambda = \frac{p_0(C_{10} - C_{00})}{p_1(C_{01} - C_{11})}$$

(b) What are the cost values for which the Bayes' criterion reduces to the minimum probability of error criterion?

Principles of Rotational and Translational Invariance

5.15 Continuing with the four line codes considered in Problem 5.1, identify the line codes that have minimum average energy and those that do not. Compare your answers with the observations made on these line codes in Section 3.7.

5.16 Consider the two constellations shown in Figure 5.11. Determine the orthonormal matrix $\mathbf{Q}$ that transforms the constellation shown in Figure 5.11*a* into the one shown in Figure 5.11*b*.

CHAPTER 6

PASSBAND DATA TRANSMISSION

This chapter builds on the material developed in Chapter 5 on signal-space analysis to discuss the subject of digital data transmission over a band-pass channel, which can be linear or nonlinear. As with analog communications, this mode of data transmission relies on the use of a sinusoidal carrier wave modulated by the data stream.

Specifically, the following topics are covered:

- *Different methods of digital modulation, namely, phase-shift keying, quadrature-amplitude modulation, and frequency-shift keying and their individual variants.*
- *Coherent detection of modulated signals in additive white Gaussian noise, which requires the receiver to be synchronized to the transmitter with respect to both carrier phase and bit timing.*
- *Noncoherent detection of modulated signals in additive white Gaussian noise, disregarding phase information in the received signal.*
- *Modems for the transmission and reception of digital data over the public switched telephone network.*
- *Sophisticated modulation techniques, namely, carrierless amplitude/phase modulation and discrete multitone, for data transmission over a wideband channel with medium to severe intersymbol interference.*
- *Techniques for synchronizing the receiver to the transmitter.*

6.1 Introduction

In *baseband pulse transmission*, which we studied in Chapter 4, a data stream represented in the form of a discrete pulse-amplitude modulated (PAM) signal is transmitted directly over a low-pass channel. In *digital passband transmission*, on the other hand, the incoming data stream is modulated onto a carrier (usually sinusoidal) with fixed frequency limits imposed by a band-pass channel of interest; passband data transmission is studied in this chapter.

The communication channel used for passband data transmission may be a microwave radio link, a satellite channel, or the like. Yet other applications of passband data transmission are in the design of passband line codes for use on digital subscriber loops and orthogonal frequency-division multiplexing techniques for broadcasting. In any event, the modulation process making the transmission possible involves switching (keying) the amplitude, frequency, or phase of a sinusoidal carrier in some fashion in accordance with the incoming data. Thus there are three basic signaling schemes, and they are known as

Returning to the functional model of Figure 6.2, the bandpass communication channel, coupling the transmitter to the receiver, is assumed to have two characteristics:

1. The channel is linear, with a bandwidth that is wide enough to accommodate the transmission of the modulated signal $s_i(t)$ with negligible or no distortion.
2. The channel noise $w(t)$ is the sample function of a white Gaussian noise process of zero mean and power spectral density $N_0/2$.

The assumptions made herein are basically the same as those invoked in Chapter 5 dealing with signal-space analysis.

The receiver, which consists of a *detector* followed by a *signal transmission decoder*, performs two functions:

1. It reverses the operations performed in the transmitter.
2. It minimizes the effect of channel noise on the estimate $\hat{m}$ computed for the transmitted symbol m.

6.3 Coherent Phase-Shift Keying

With the background material on the coherent detection of signals in additive white Gaussian noise that was presented in Chapter 5 at our disposal, we are now ready to study specific passband data transmission systems. In this section we focus on coherent phase-shift keying (PSK) by considering binary PSK, QPSK and its variants, and finish up with M-ary PSK.

Binary Phase-Shift Keying

In a coherent binary PSK system, the pair of signals $s_1(t)$ and $s_2(t)$ used to represent binary symbols 1 and 0, respectively, is defined by

$$s_1(t) = \sqrt{\frac{2E_b}{T_b}}\cos(2\pi f_c t) \tag{6.8}$$

$$s_2(t) = \sqrt{\frac{2E_b}{T_b}}\cos(2\pi f_c t + \pi) = -\sqrt{\frac{2E_b}{T_b}}\cos(2\pi f_c t) \tag{6.9}$$

where $0 \le t \le T_b$, and E_b is the *transmitted signal energy per bit*. To ensure that each transmitted bit contains an integral number of cycles of the carrier wave, the carrier frequency f_c is chosen equal to n_c/T_b for some fixed integer n_c. A pair of sinusoidal waves that differ only in a relative phase-shift of 180 degrees, as defined in Eqautions (6.8) and (6.9), are referred to as *antipodal signals*.

From this pair of equations it is clear that, in the case of binary PSK, there is only one basis function of unit energy, namely,

$$\phi_1(t) = \sqrt{\frac{2}{T_b}}\cos(2\pi f_c t), \qquad 0 \le t < T_b \tag{6.10}$$

Then we may express the transmitted signals $s_1(t)$ and $s_2(t)$ in terms of $\phi_1(t)$ as follows:

$$s_1(t) = \sqrt{E_b}\phi_1(t), \qquad 0 \le t < T_b \tag{6.11}$$

and

$$s_2(t) = -\sqrt{E_b}\phi_1(t), \qquad 0 \le t < T_b \tag{6.12}$$

other hand, if $x_1 < 0$, it decides in favor of symbol 0. If x_1 is exactly zero, the receiver makes a random guess in favor of 0 or 1.

Power Spectra of Binary PSK Signals

From the modulator of Figure 6.4*a*, we see that the complex envelope of a binary PSK wave consists of an in-phase component only. Furthermore, depending on whether we have symbol 1 or symbol 0 at the modulator input during the signaling interval $0 \le t \le T_b$, we find that this in-phase component equals $+g(t)$ or $-g(t)$, respectively, where $g(t)$ is the *symbol shaping function* defined by

$$g(t) = \begin{cases} \sqrt{\dfrac{2E_b}{T_b}}, & 0 \le t \le T_b \\ 0, & \text{otherwise} \end{cases} \tag{6.21}$$

We assume that the input binary wave is random, with symbols 1 and 0 equally likely and the symbols transmitted during the different time slots being statistically independent. In Example 1.6 of Chapter 1 it is shown that the power spectral density of a random binary wave so described is equal to the energy spectral density of the symbol shaping function divided by the symbol duration. The energy spectral density of a Fourier transformable signal $g(t)$ is defined as the squared magnitude of the signal's Fourier transform. Hence, the baseband power spectral density of a binary PSK signal equals

$$\begin{aligned} S_B(f) &= \frac{2E_b \sin^2(\pi T_b f)}{(\pi T_b f)^2} \\ &= 2E_b \operatorname{sinc}^2(T_b f) \end{aligned} \tag{6.22}$$

This power spectrum falls off as the inverse square of frequency, as shown in Figure 6.5.

Figure 6.5 also includes a plot of the baseband power spectral density of a binary FSK signal, details of which are presented in Section 6.5. Comparison of these two spectra is deferred to that section.

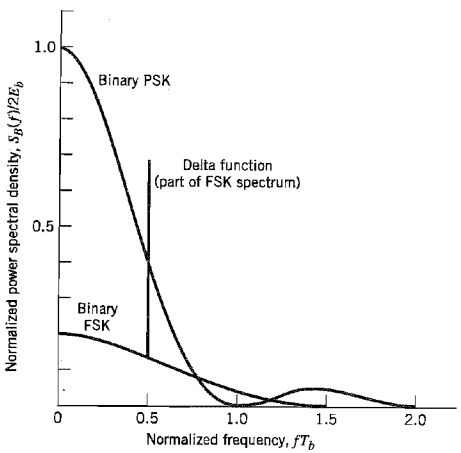

FIGURE 6.5 Power spectra of binary PSK and FSK signals.

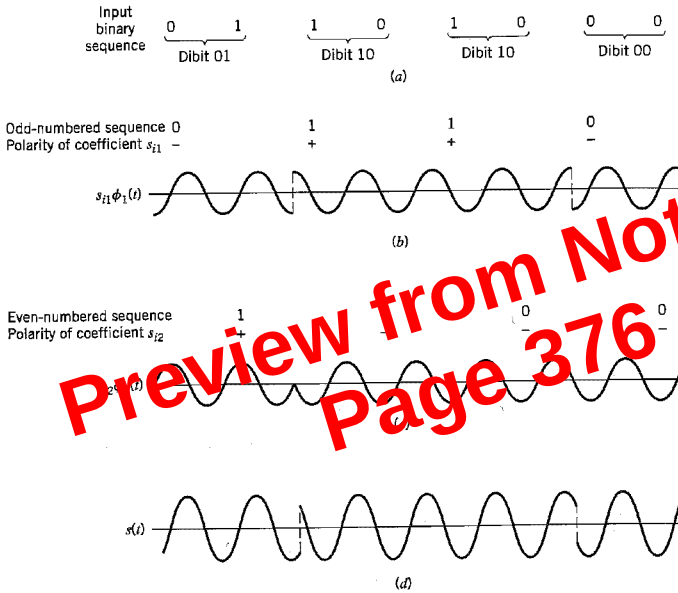

FIGURE 6.7 (*a*) Input binary sequence. (*b*) Odd-numbered bits of input sequence and associated binary PSK wave. (*c*) Even-numbered bits of input sequence and associated binary PSK wave. (*d*) QPSK waveform defined as $s(t) = s_{i1}\phi_1(t) + s_{i2}\phi_2(t)$.

off the appropriate regions. We thus find that the decision regions are quadrants whose vertices coincide with the origin. These regions are marked Z_1, Z_2, Z_3, and Z_4, in Figure 6.6, according to the message point around which they are constructed. ◀

Error Probability of QPSK

In a coherent QPSK system, the received signal $x(t)$ is defined by

$$x(t) = s_i(t) + w(t), \qquad \begin{cases} 0 \le t \le T \\ i = 1, 2, 3, 4 \end{cases} \tag{6.28}$$

where $w(t)$ is the sample function of a white Gaussian noise process of zero mean and power spectral density $N_0/2$. Correspondingly, the observation vector $\mathbf{x}$ has two elements, x_1 and x_2, defined by

$$\begin{aligned} x_1 &= \int_0^T x(t)\phi_1(t)\,dt \\ &= \sqrt{E}\cos\left[(2i-1)\frac{\pi}{4}\right] + w_1 \\ &= \pm\sqrt{\frac{E}{2}} + w_1 \end{aligned} \tag{6.29}$$

bandwidth. For a prescribed performance, QPSK uses channel bandwidth better than binary PSK, which explains the preferred use of QPSK over binary PSK in practice.

Generation and Detection of Coherent QPSK Signals

Consider next the generation and detection of QPSK signals. Figure 6.8*a* shows a block diagram of a typical QPSK transmitter. The incoming binary data sequence is first transformed into polar form by a *nonreturn-to-zero level* encoder. Thus, symbols 1 and 0 are represented by $+\sqrt{E_b}$ and $-\sqrt{E_b}$, respectively. This binary wave is next divided by means of a *demultiplexer* into two separate binary waves consisting of the odd- and even-numbered input bits. These two binary waves are denoted by $a_1(t)$ and $a_2(t)$. We note that in any signaling interval, the amplitudes of $a_1(t)$ and $a_2(t)$ equal s_{i1} and s_{i2}, respectively, depending on the particular dibit that is being transmitted. The two binary waves $a_1(t)$ and $a_2(t)$ are used to modulate a pair of quadrature carriers or orthonormal basis functions: $\phi_1(t)$ equal to $\sqrt{2/T}\cos(2\pi f_c t)$ and $\phi_2(t)$ equal to $\sqrt{2/T}\sin(2\pi f_c t)$. The result is a pair of

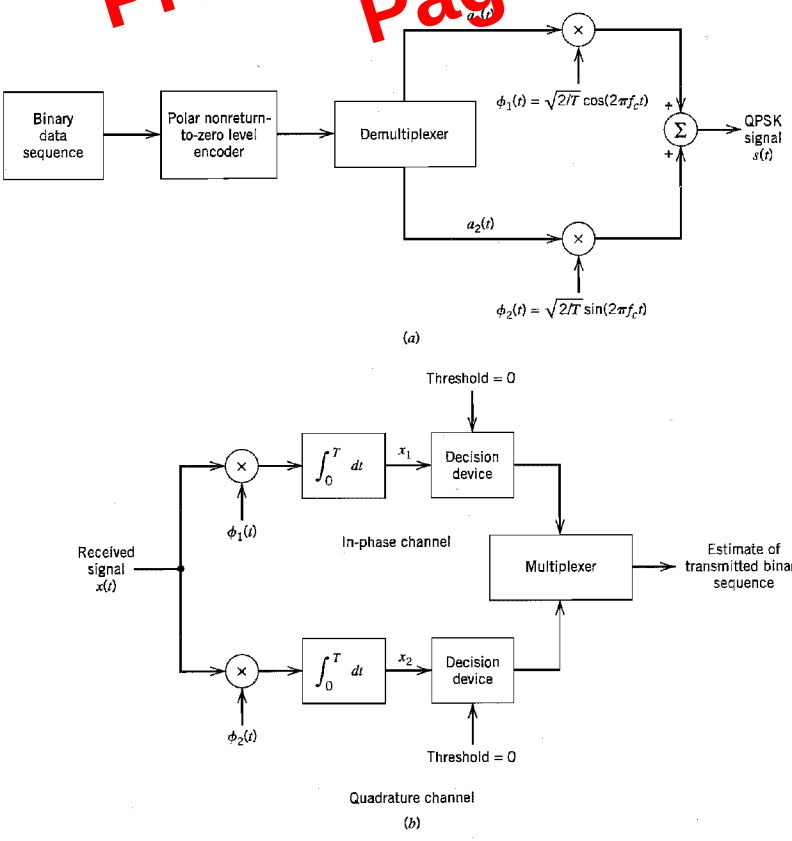

FIGURE 6.8 Block diagrams of (*a*) QPSK transmitter and (*b*) coherent QPSK receiver.

binary PSK signals, which may be detected independently due to the orthogonality of $\phi_1(t)$ and $\phi_2(t)$. Finally, the two binary PSK signals are added to produce the desired QPSK signal.

The QPSK receiver consists of a pair of correlators with a common input and supplied with a locally generated pair of coherent reference signals $\phi_1(t)$ and $\phi_2(t)$, as in Figure 6.8*b*. The correlator outputs x_1 and x_2, produced in response to the received signal $x(t)$, are each compared with a threshold of zero. If $x_1 > 0$, a decision is made in favor of symbol 1 for the in-phase channel output, but if $x_1 < 0$, a decision is made in favor of symbol 0. Similarly, if $x_2 > 0$, a decision is made in favor of symbol 1 for the quadrature channel output, but if $x_2 < 0$, a decision is made in favor of symbol 0. Finally, these two binary sequences at the in-phase and quadrature channel outputs are combined in a *multiplexer* to reproduce the original binary sequence at the transmitter input with the minimum probability of symbol error in an AWGN channel.

Power Spectra of QPSK Signals

Assume that the binary wave at the modulator input is random, with symbols 1 and 0 being equally likely, and with the symbols transmitted during adjacent time slots being statistically independent. We make the following observations pertaining to the in-phase and quadrature components of a QPSK signal:

1. Depending on the dibit sent during the signaling interval $-T_b \le t \le T_b$, the in-phase component equals $+g(t)$ or $-g(t)$, and similarly for the quadrature component. The $g(t)$ denotes the symbol shaping function, defined by

$$g(t) = \begin{cases} \sqrt{\dfrac{E}{T}}, & 0 \le t \le T \\ 0, & \text{otherwise} \end{cases} \tag{6.39}$$

Hence, the in-phase and quadrature components have a common power spectral density, namely, $E\,\text{sinc}^2(Tf)$.

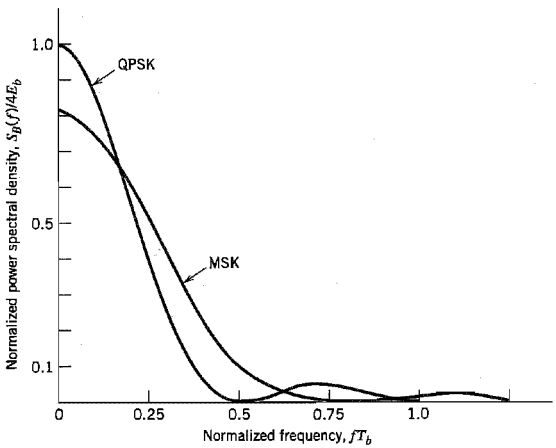

FIGURE 6.9 Power spectra of QPSK and MSK signals.

transmitted signal corresponds to the message point m_1, whose coordinates along the ϕ_1- and ϕ_2-axes are $+\sqrt{E}$ and 0, respectively. Suppose that the ratio E/N_0 is large enough to consider the nearest two message points, one on either side of m_1, as potential candidates for being mistaken for m_1 due to channel noise. This is illustrated in Figure 6.15*b* for the case of $M = 8$. The Euclidean distance of each of these two points from m_1 is (for $M = 8$)

$$d_{12} = d_{18} = 2\sqrt{E}\sin\left(\frac{\pi}{M}\right)$$

Hence, the use of Equation (5.92) of Chapter 5 yields the average probability of symbol error for coherent M-ary PSK as

$$P_e \simeq \operatorname{erfc}\left(\sqrt{\frac{E}{N_0}}\sin\left(\frac{\pi}{M}\right)\right) \tag{6.47}$$

where it is assumed that $M \geq 4$. The approximation becomes extremely tight, for fixed M, as E/N_0 is increased. For $M = 4$, Equation (6.47) reduces to the same form given in Equation (6.34) for QPSK.

Power Spectra of M-ary PSK Signals

The symbol duration of M-ary PSK is defined by

$$T = T_b \log_2 M \tag{6.48}$$

where T_b is the bit duration. Proceeding in a manner similar to that described for a QPSK signal, we may show that the baseband power spectral density of an M-ary PSK signal is given by

$$\begin{aligned} S_B(f) &= 2E\,\operatorname{sinc}^2(Tf) \\ &= 2E_b \log_2 M\,\operatorname{sinc}^2(T_b f \log_2 M) \end{aligned} \tag{6.49}$$

In Figure 6.16, we show the normalized power spectral density $S_B(f)/2E_b$ plotted versus the normalized frequency fT_b for three different values of M, namely, $M = 2, 4, 8$.

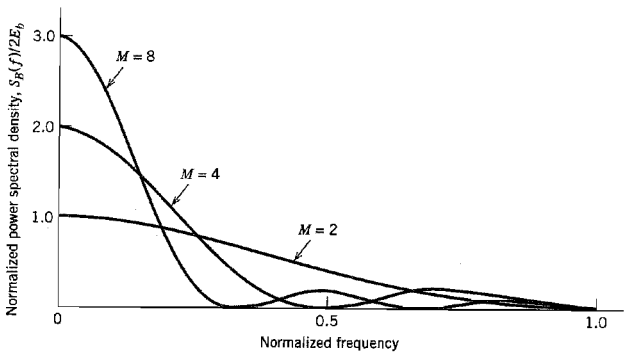

FIGURE 6.16 Power spectra of M-ary PSK signals for $M = 2, 4, 8$.

ization complicates the determination of the probability of symbol error P_e incurred in the use of M-ary QAM characterized by a cross constellation. We therefore simply state the formula for P_e without proof, as shown here

$$P_e \simeq 2\left(1 - \frac{1}{\sqrt{2M}}\right) \operatorname{erfc}\left(\sqrt{\frac{E_0}{N_0}}\right) \quad \text{for high } E_0/N_0 \tag{6.65}$$

which agrees with the formula of Equation (6.61) for a square constellation, except for the inclusion of an extra 0.5 bit per dimension in the constellation.[5] Note also that it is not possible to perfectly Gray code a QAM cross constellation.

Carrierless Amplitude/Phase Modulation

The passband basis functions of equations (6.53) and (6.54) assume the use of a rectangular pulse for the pulse-shaping function. For reasons that will become apparent, we redefine the transmitted M-ary QAM signal of Equation (6.55) in terms of a general pulse-shaping function $g(t)$ as

$$s_k(t) = a_k g(t - kT)\cos(2\pi f_c t) - b_k g(t - kT)\sin(2\pi f_c t), \quad \begin{matrix} 0 \le t \le T \\ k = 0, \pm 1, \pm 2, \ldots \end{matrix} \tag{6.66}$$

It is assumed that carrier frequency f_c has an arbitrary value with respect to the symbol rate $1/T$. On the basis of Equation (6.66), we may express the transmitted M-ary QAM signal $s(t)$ for an infinite succession of symbols as

$$\begin{aligned} s(t) &= \sum_{k=-\infty}^{\infty} s_k(t) \\ &= \sum_{k=-\infty}^{\infty} [a_k g(t - kT)\cos(2\pi f_c t) - b_k g(t - kT)\sin(2\pi f_c t)] \end{aligned} \tag{6.67}$$

This equation shows that for an arbitrary f_c, the passband functions $g(t - kT)\cos(2\pi f_c t)$ and $g(t - kT)\sin(2\pi f_c t)$ are *aperiodic* in that they vary from one symbol to another.

How can we eliminate the time variations of these passband basis functions from symbol to symbol? To answer this question, we find it convenient to change our formalism from real to complex notation. Specifically, we rewrite Equation (6.67) in the equivalent form

$$\begin{aligned} s(t) &= \operatorname{Re}\left\{\sum_{k=-\infty}^{\infty} (a_k + jb_k) g(t - kT)\exp(j2\pi f_c t)\right\} \\ &= \operatorname{Re}\left\{\sum_{k=-\infty}^{\infty} A_k g(t - kT)\exp(j2\pi f_c t)\right\} \end{aligned} \tag{6.68}$$

where A_k is a complex number defined by

$$A_k = a_k + jb_k \tag{6.69}$$

and $\operatorname{Re}\{\cdot\}$ denotes the real part of the complex quantity enclosed inside the braces. Clearly, Equation (6.68) is unchanged by multiplying the summand in this equation by unity ex-

Binary FSK

In a *binary FSK system*, symbols 1 and 0 are distinguished from each other by transmitting one of two sinusoidal waves that differ in frequency by a fixed amount. A typical pair of sinusoidal waves is described by

$$s_i(t) = \begin{cases} \sqrt{\frac{2E_b}{T_b}} \cos(2\pi f_i t), & 0 \le t \le T_b \\ 0, & \text{elsewhere} \end{cases} \tag{6.86}$$

where $i = 1, 2$, and E_b is the transmitted signal energy per bit; the transmitted frequency is

$$f_i = \frac{n_c + i}{T_b} \quad \text{for some fixed integer } n_c \text{ and } i = 1, 2 \tag{6.87}$$

Thus symbol 1 is represented by $s_1(t)$, and symbol 0 by $s_2(t)$. The FSK signal described here is known as *Sunde's FSK*. It is a *continuous-phase signal* in the sense that phase continuity is always maintained, including the over-bit switching times. This form of digital modulation is an example of *continuous-phase frequency-shift keying* (CPFSK), on which we have more to say later on in the section.

From Equations (6.86) and (6.87), we observe directly that the signals $s_1(t)$ and $s_2(t)$ are orthogonal, but not normalized to have unit energy. We therefore deduce that the most useful form for the set of orthonormal basis functions is

$$\phi_i(t) = \begin{cases} \sqrt{\frac{2}{T_b}} \cos(2\pi f_i t), & 0 \le t \le T_b \\ 0, & \text{elsewhere} \end{cases} \tag{6.88}$$

where $i = 1, 2$. Correspondingly, the coefficient s_{ij} for $i = 1, 2$, and $j = 1, 2$ is defined by

$$\begin{aligned} s_{ij} &= \int_0^{T_b} s_i(t)\phi_j(t)\, dt \\ &= \int_0^{T_b} \sqrt{\frac{2E_b}{T_b}} \cos(2\pi f_i t) \sqrt{\frac{2}{T_b}} \cos(2\pi f_j t)\, dt \\ &= \begin{cases} \sqrt{E_b}, & i = j \\ 0, & i \ne j \end{cases} \end{aligned} \tag{6.89}$$

Thus, unlike coherent binary PSK, a coherent binary FSK system is characterized by having a signal space that is two-dimensional (i.e., $N = 2$) with two message points (i.e., $M = 2$), as shown in Figure 6.25. The two message points are defined by the

$$\mathbf{s}_1 = \begin{bmatrix} \sqrt{E_b} \\ 0 \end{bmatrix} \tag{6.90}$$

and

$$\mathbf{s}_2 = \begin{bmatrix} 0 \\ \sqrt{E_b} \end{bmatrix} \tag{6.91}$$

with the Euclidean distance between them equal to $\sqrt{2E_b}$. Figure 6.25 also includes a couple of inserts, which show waveforms representative of signals $s_1(t)$ and $s_2(t)$.

the line joining the two message points. The receiver decides in favor of symbol 1 if the received signal point represented by the observation vector $\mathbf{x}$ falls inside region Z_1. This occurs when $x_1 > x_2$. If, on the other hand, we have $x_1 < x_2$, the received signal point falls inside region Z_2, and the receiver decides in favor of symbol 0. On the decision boundary, we have $x_1 = x_2$, in which case the receiver makes a random guess in favor of symbol 1 or 0.

Define a new Gaussian random variable Y whose sample value y is equal to the difference between x_1 and x_2; that is,

$$y = x_1 - x_2 \tag{6.94}$$

The mean value of the random variable Y depends on which binary symbol was transmitted. Given that symbol 1 was transmitted, the Gaussian random variables X_1 and X_2, whose sample values are denoted by x_1 and x_2, have mean values equal to $\sqrt{E_b}$ and zero, respectively. Correspondingly, the conditional mean of the random variable Y, given that symbol 1 was transmitted, is

$$\begin{aligned} E[Y|1] &= E[X_1|1] - E[X_2|1] \\ &= +\sqrt{E_b} \end{aligned} \tag{6.95}$$

On the other hand, given that symbol 0 was transmitted, the random variables X_1 and X_2 have mean values equal to zero and $\sqrt{E_b}$, respectively. Correspondingly, the conditional mean of the random variable Y, given that symbol 0 was transmitted, is

$$\begin{aligned} E[Y|0] &= E[X_1|0] - E[X_2|0] \\ &= -\sqrt{E_b} \end{aligned} \tag{6.96}$$

The variance of the random variable Y is independent of which binary symbol was transmitted. Since the random variables X_1 and X_2 are statistically independent, each with a variance equal to $N_0/2$, it follows that

$$\begin{aligned} \text{var}[Y] &= \text{var}[X_1] + \text{var}[X_2] \\ &= N_0 \end{aligned} \tag{6.97}$$

Suppose we know that symbol 0 was transmitted. The conditional probability density function of the random variable Y is then given by

$$f_Y(y|0) = \frac{1}{\sqrt{2\pi N_0}} \exp\left[-\frac{(y + \sqrt{E_b})^2}{2N_0}\right] \tag{6.98}$$

Since the condition $x_1 > x_2$, or equivalently, $y > 0$, corresponds to the receiver making a decision in favor of symbol 1, we deduce that the conditional probability of error, given that symbol 0 was transmitted, is

$$\begin{aligned} p_{10} &= P(y > 0 \,|\, \text{symbol 0 was sent}) \\ &= \int_0^\infty f_Y(y|0)\, dy \\ &= \frac{1}{\sqrt{2\pi N_0}} \int_0^\infty \exp\left[-\frac{(y + \sqrt{E_b})^2}{2N_0}\right] dy \end{aligned} \tag{6.99}$$

Put

$$\frac{y + \sqrt{E_b}}{\sqrt{2N_0}} = z \tag{6.100}$$

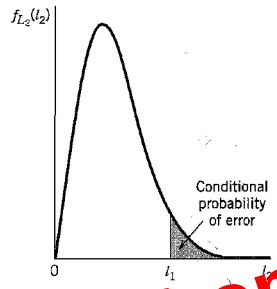

FIGURE 6.41 Calculation of the conditional probability that $l_2 > l_1$, given l_1.

Substituting Equation (6.167) into Equation (6.168) and integrating, we get

$$P(l_2 > l_1 \mid l_1) = \exp\left(-\frac{l_1^2}{N_0}\right) \tag{6.169}$$

Consider next the output amplitude l_1, pertaining to the upper path in Figure 6.39*a*. Since the filter in this path is matched to $s_1(t)$, and it is assumed that $s_1(t)$ is transmitted, it follows that l_1 is due to *signal plus noise*. Let x_{I1} and x_{Q1} denote the components at the output of the matched filter (in the upper path of Figure 6.39*a*) that are in phase and in quadrature with respect to the received signal, respectively. Then from the equivalent structure depicted in Figure 6.39*b*, we see that (for $i = 1$)

$$l_1 = \sqrt{x_{I1}^2 + x_{Q1}^2} \tag{6.170}$$

Figure 6.40*b* presents a geometric interpretation of this relation. Since a Fourier-transformable signal and its Hilbert transform form an orthogonal pair, it follows that x_{I1} is due to signal plus noise, whereas x_{Q1} is due to noise alone. This means that (1) the random variable X_{I1} represented by the sample value x_{I1} is Gaussian distributed with mean $\sqrt{E}$ and variance $N_0/2$, where E is the signal energy per symbol, and (2) the random variable X_{Q1} represented by the sample value x_{Q1} is Gaussian distributed with zero mean and variance $N_0/2$. Hence, we may express the probability density functions of these two independent random variables as follows:

$$f_{X_{I1}}(x_{I1}) = \frac{1}{\sqrt{\pi N_0}} \exp\left(-\frac{(x_{I1} - \sqrt{E})^2}{N_0}\right) \tag{6.171}$$

and

$$f_{X_{Q1}}(x_{Q1}) = \frac{1}{\sqrt{\pi N_0}} \exp\left(-\frac{x_{Q1}^2}{N_0}\right) \tag{6.172}$$

Since the two random variables X_{I1} and X_{Q1} are independent, their joint probability density function is simply the product of the probability density functions given in Equations (6.171) and (6.172).

To find the average probability of error, we have to average the conditional probability of error given in Equation (6.169) over all possible values of l_1. Naturally, this calculation requires knowledge of the probability density function of random variables L_1 represented by sample value l_1. The standard method is now to combine Equations (6.171) and (6.172) to find the probability density function of L_1 due to signal plus noise. However, this leads to rather complicated calculations involving the use of Bessel functions. This

analytic difficulty may be circumvented by the following approach. Given x_{I1} and x_{Q1}, an error occurs when, in Figure 6.39*a*, the lower path's output amplitude l_2 due to noise alone exceeds l_1 due to signal plus noise; from Equation (6.170) we have

$$l_1^2 = x_{I1}^2 + x_{Q1}^2 \tag{6.173}$$

The probability of such an occurrence is obtained by substituting Equation (6.173) into Equation (6.169), as shown by

$$P(\text{error}\,|\,x_{I1}, x_{Q1}) = \exp\left(-\frac{x_{I1}^2 + x_{Q1}^2}{N_0}\right) \tag{6.174}$$

This is now a conditional probability of error, conditioned on the output of the matched filter in the upper path taking on values X_{I1} and X_{Q1}. This conditional probability multiplied by the joint probability density function of X_{I1} and X_{Q1} is the *error density*, *given* x_{I1} *and* x_{Q1}. Since X_{I1} and X_{Q1} are statistically independent, the joint probability density function equals the product of their individual probability density functions. The resulting error-density is a complicated expression in x_{I1} and x_{Q1}. However, the average probability of error, which is the issue of interest, may be obtained in a relatively simple manner. We first use Equations (6.171), (6.172), and (6.174) to evaluate the desired error-density as

$$\begin{aligned} &P(\text{error}\,|\,x_{I1}, x_{Q1}) f_{X_{I1}}(x_{I1}) f_{X_{Q1}}(x_{Q1}) \\ &= \frac{1}{\pi N_0} \exp\left\{-\frac{1}{N_0}\left[x_{I1}^2 + x_{Q1}^2 + (x_{I1} - \sqrt{E})^2 + x_{Q1}^2\right]\right\} \end{aligned} \tag{6.175}$$

Completing the square in the exponent of Equation (6.175), we may rewrite the exponent except for $-1/N_0$ as

$$x_{I1}^2 + x_{Q1}^2 + (x_{I1} - \sqrt{E})^2 + x_{Q1}^2 = 2\left(x_{I1} - \frac{\sqrt{E}}{2}\right)^2 + 2x_{Q1}^2 + \frac{E}{2} \tag{6.176}$$

Next, we substitute Equation (6.176) into Equation (6.175) and integrate the error-density over all x_{I1} and x_{Q1}. We thus evaluate the average probability of error as

$$\begin{aligned} P_e &= \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} P(\text{error}\,|\,x_{I1}, x_{Q1}) f_{X_{I1}}(x_{I1}) f_{X_{Q1}}(x_{Q1})\, dx_{I1}\, dx_{Q1} \\ &= \frac{1}{\pi N_0} \exp\left(-\frac{E}{2N_0}\right) \int_{-\infty}^{\infty} \exp\left[-\frac{2}{N_0}\left(x_{I1} - \frac{\sqrt{E}}{2}\right)^2\right] dx_{I1} \\ &\quad \cdot \int_{-\infty}^{\infty} \exp\left(-\frac{2x_{Q1}^2}{N_0}\right) dx_{Q1} \end{aligned} \tag{6.177}$$

We now use the following two identities:

$$\int_{-\infty}^{\infty} \exp\left[-\frac{2}{N_0}\left(x_{I1} - \frac{\sqrt{E}}{2}\right)^2\right] dx_{I1} = \sqrt{\frac{N_0\pi}{2}} \tag{6.178}$$

and

$$\int_{-\infty}^{\infty} \exp\left(-\frac{2x_{Q1}^2}{N_0}\right) dx_{Q1} = \sqrt{\frac{N_0\pi}{2}} \tag{6.179}$$

The identity of Equation (6.178) is obtained by considering a Gaussian-distributed variable with mean $\sqrt{E/2}$ and variance $N_0/4$, and recognizing that the total area under the curve of a random variable's probability density function equals unity; the identity of Equation

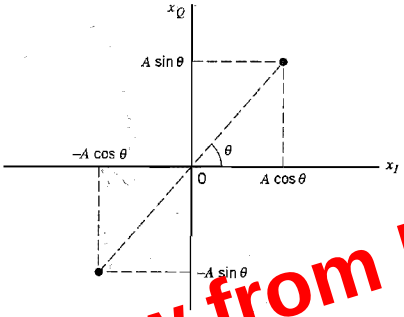

FIGURE 6.44 Signal-space diagram of received DPSK signal.

Thus, the *optimum receiver*[12] for differentially coherent detection of binary DPSK is as shown in Figure 6.43*b*, which follows directly from Equation (6.185). This implementation merely requires that *sample* values be stored, thereby avoiding the need for fancy delay lines that may be needed otherwise. The equivalent receiver implementation that tests squared elements as in Equation (6.186) is more complicated, but its use makes the *analysis* easier to handle in that the two signals to be considered are orthogonal over the interval $(0, 2T_b)$; hence, the noncoherent orthogonal demodulation analysis applies.

6.10 *Comparison of Digital Modulation Schemes Using a Single Carrier*

PROBABILITY OF ERROR

In Table 6.8 we have summarized the expressions for the bit error rate (BER) for coherent binary PSK, conventional coherent binary FSK with one-bit decoding, DPSK, noncoherent binary FSK, coherent QPSK, and coherent MSK, when operating over an AWGN channel. In Figure 6.45 we have used the expressions summarized in Table 6.8 to plot the BER as a function of the signal energy per bit-to-noise spectral density ratio, E_b/N_0.

TABLE 6.8 *Summary of formulas for the bit error rate of different digital modulation schemes*

Signaling Scheme	*Bit Error Rate*
(a) Coherent binary PSK Coherent QPSK Coherent MSK	$\frac{1}{2}\operatorname{erfc}(\sqrt{E_b/N_0})$
(b) Coherent binary FSK	$\frac{1}{2}\operatorname{erfc}(\sqrt{E_b/2N_0})$
(c) DPSK	$\frac{1}{2}\exp(-E_b/N_0)$
(d) Noncoherent binary FSK	$\frac{1}{2}\exp(-E_b/2N_0)$

SYMMETRIC MODEM CONFIGURATIONS

The simplest approach to the design of modems is to treat the entire PSTN as a linear analog network, as indicated in Figure 6.47*a*. (Recall from Chapter 3 that the PSTN is almost entirely digital due to the use of pulse-code modulation (PCM) for the transmission of voice signals.) In such a setting, analog-to-digital and digital-to-analog conversions are needed whenever the modems send signals to and receive signals from the PSTN. The modem configuration depicted in Figure 6.47*a* exhibits "symmetry" in that both modems are identical and the data rate *downstream* (from the ISP to the user) is exactly the same as the data rate *upstream* (from the user to the ISP).

The symmetric modem configuration of Figure 6.47*a* embodies a large number of modem types, ranging in data rate from 300 b/s to 36,600 b/s as summarized in Table A6.7 on a selection of standard modems. The designs of modems began with frequency-shift keying, which catered to relatively low data rates. As the demand for data transmission over telephone channels increased, increasingly more sophisticated modulation techniques were employed to better use the information capacity of the telephone channel.

Consider, for example, the popular V.32 modem *standard* that has the following characteristics:

Carrier frequency = 1,800 Hz
Modulation rate = 2,400 bauds
Data rate = 9,600 b/s

The signaling data rate of 9,600 b/s assumes a high signal-to-noise ratio. The V.32 standard specifies two alternative modulation schemes:

Nonredundant coding. Under this scheme, the incoming data stream is divided into quadbits (i.e., groups of four successive bits) and then transmitted over the telephone channel as 16-QAM. In each quadbit, the most significant input dibit undergoes phase modulation, whereas the least significant input dibit undergoes amplitude modulation. Discussing the phase modulation first, practical considerations favor the use of differential phase modulation for the receiver need only be concerned with the detection of phase charges. This matter is taken care of by using a *differential encoder*, which consists of a read-only memory and a couple of delay units, as shown in Figure 6.48*a*. Let $Q_{1,n}Q_{2,n}$ denote the current value of the most significant

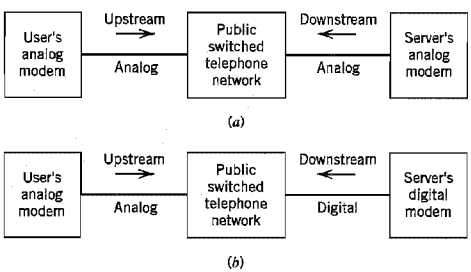

FIGURE 6.47 (*a*) Environmental overview of symmetric modem configuration: the upstream and downstream data rates are equal. (*b*) Environmental overview of "asymmetric" modem configuration: data rate downstream is higher than upstream.

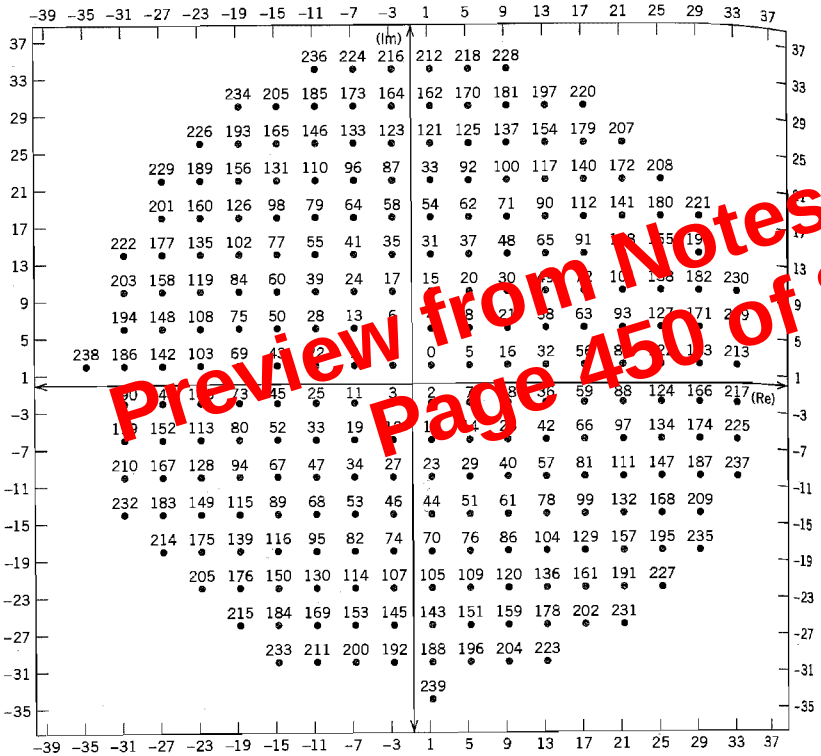

FIGURE 6.53 Quarter-superconstellation of V.34 modem with 240 signal points. The full superconstellation is obtained by combining the rotated versions of these points by 0, 90, 180, and 270 degrees. (Taken from Forney et al., 1996, with permission of the IEEE.)

4. *Trellis coding.*
This error-control coding technique is used to provide an effective coding gain of about 3.6 dB; there is an optional more powerful trellis code with an effective coding gain of about 4.7 dB.

5. *Decision feedback equalization.*
To make full use of the available telephone channel bandwidth, including frequencies near the band edges where there can be attenuation as much as 10 to 20 dB, a decision feedback equalizer (DFE) is used. (The DFE is discussed in Chapter 4.) However, it is not a straightforward matter to combine coding with DFE because decision feedback requires immediate decisions, whereas coding inherently involves decoding delay. The overcome this problem, the feedback section of the DFE is moved to the transmitter, which is made possible through the use of the Tomlinson-Harashima precoding. (This form of equalization via precoding is discussed briefly in Note 12 of Chapter 4.)

where K is a prescribed constant under the designer's control. That is, the sum of the transmit power and the noise variance (power) scaled by the ratio Γ/g_n^2 must be maintained constant for each subchannel. The process of allocating the transmit power P to the individual subchannels so as to maximize the bit rate of the entire multichannel transmission system is called *loading*.

■ WATER-FILLING INTERPRETATION OF THE OPTIMIZATION PROBLEM

In solving the constrained optimization problem just described, two conditions must be satisfied, namely, Equations (6.210) and (6.213). The optimum solution so obtained has an interesting interpretation as illustrated in Figure 6.56 for $N = 6$, assuming that the gap Γ is constant over all the subchannels. To simplify the illustration in Figure 6.56 we have set $\sigma_n^2 = N_0 \Delta f = 1$, that is, the average noise power is unity for all N subchannels. Referring to this figure, we may now make the following observations:

- The amount of power P_n allocated to channel n and the scaled noise power Γ/g_n^2 satisfies the constraint of Equation (6.213) for four of the subchannels for a prescribed value of the constant K.
- The sum of power allocations to these four subchannels consumes all the available transmit power, maintained at the constant value P.
- The remaining two subchannels have been eliminated from consideration because they would each require negative power to satisfy Equation (6.213) for the prescribed value of the constant K; this condition is clearly unacceptable.

The interpretation illustrated in Figure 6.56 prompts us to refer to the optimum solution of Equation (6.213), subject to the constraint of Equation (6.210), as the *water-filling solution*. This terminology follows from analogy of our optimization problem with a fixed amount of water (standing for transmit power) being poured into a container with a number of connected regions, each having a different depth (standing for noise power). The water distributes itself in such a way that a constant water level is attained across the whole container. We have more to say on the water-filling interpretation of information capacity in Chapter 9.

Returning to the task of how to allocate the fixed transmit power P among the various subchannels of a multichannel transmission system so as to optimize the bit rate

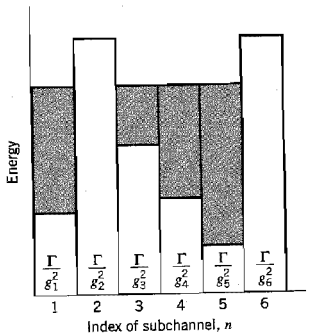

FIGURE 6.56 Water-filling interpretation of the loading problem.

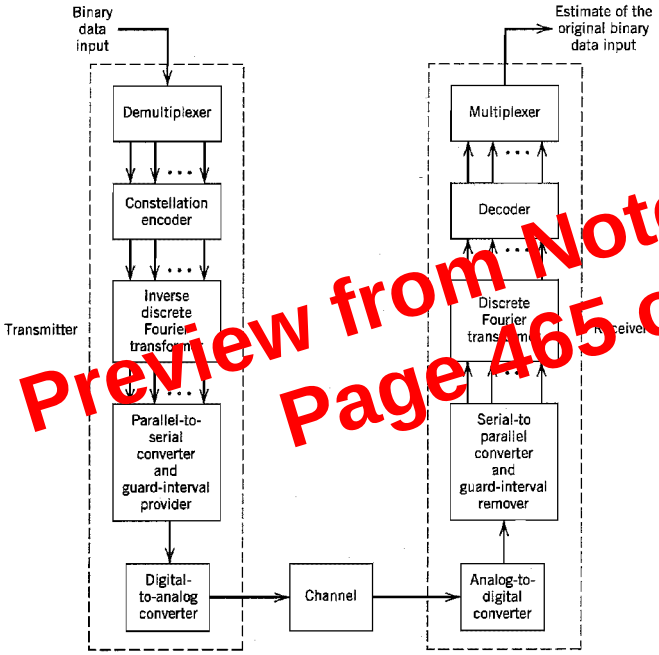

FIGURE 6.60 Block diagram of the discrete-multitone (DMT) data-transmission system.

- *Inverse discrete Fourier transformer (IDFT)*, which transforms the frequency-domain parallel data at the constellation encoder output into parallel time-domain data. For efficient implementation of the IDFT using the fast Fourier transform (FFT) algorithm, we need to choose $N = 2^k$ where k is a positive integer.
- *Parallel-to-serial converter*, which converts the parallel time-domain data into serial form. Guard intervals stuffed with cyclic prefixes are inserted into the serial data on a periodic basis before conversion into analog form.
- *Digital-to-analog converter (DAC)*, which converts the digital data into analog form ready for transmission over the channel.

Typically, the DAC includes a transmit filter. Accordingly, the time function $h(t)$ should be redefined as the combined impulse response of the cascade connection of the transmit filter and the channel.

The receiver performs the inverse operations of the transmitter, as described here:

- *Analog-to-digital converter (ADC)*, which converts the analog channel output into digital form.
- *Serial-to-parallel converter*, which converts the resulting bit stream into parallel form. Before this conversion takes place, the guard intervals (cyclic prefixes) are removed.
- *Discrete Fourier transformer (DFT)*, which transforms the time-domain parallel data into frequency-domain parallel data; as with the IDFT, the FFT algorithm is used to implement the DFT.

The approach taken in the exposition is sequential in that timing recovery is performed *before* phase recovery. The reason for so doing is that if we know the group delay incurred by transmission through the channel, then one sample per symbol at the matched filter output in the receiver is sufficient for estimating the unknown carrier phase. Moreover, the computational complexity of the receiver is minimized by using synchronization algorithms that operate at the symbol rate $1/T$.

■ DECISION-DIRECTED RECURSIVE ALGORITHM FOR PHASE RECOVERY

As remarked earlier, the first important step in solving the synchronization problem is to formulate the log-likelihood function for the carrier phase θ given the Gaussian noise-contaminated received signal. Let $l(\theta)$ denote the log-likelihood function, which serves as the objective function for estimating θ. The next step is to determine the derivative of $l(\theta)$ with respect to θ. The final step is to formulate a recursive (iterative) algorithm for computing a maximum likelihood estimate of the unknown θ in a step-by-step manner.

*Evaluation of $\partial l(\theta)/\partial\theta$**

Let $s_k(t)$ denote the transmitted signal for symbol $k = 0, 1, \ldots, M-1$:

$$s_k(t) = \sqrt{\frac{2E}{T}}\cos(2\pi f_c t + \alpha_k), \qquad 0 \le t \le T \tag{6.238}$$

where E is the symbol energy, T is the symbol period, and

$$\alpha_k = 0, \frac{2\pi}{M}, \ldots, (M-1)\frac{2\pi}{M} \tag{6.239}$$

Equivalently, we may write

$$s_k(t) = \sqrt{\frac{2E}{T}}\cos(2\pi f_c t + \alpha_k)g(t) \tag{6.240}$$

where $g(t)$ is the shaping pulse, namely, a rectangular pulse of unit amplitude and duration T. Let τ_c denote the carrier (phase) delay, and τ_g denote the envelope (group) delay, both of which are introduced by the channel. By definition, τ_c affects the carrier and τ_g affects the envelope. Then the received signal is

$$\begin{aligned} x(t) &= \sqrt{\frac{2E}{T}}\cos(2\pi f_c(t-\tau_c) + \alpha_k)g(t-\tau_g) + w(t) \\ &= \sqrt{\frac{2E}{T}}\cos(2\pi f_c t + \theta + \alpha_k)g(t-\tau_g) + w(t) \end{aligned} \tag{6.241}$$

where $w(t)$ is the channel noise and θ is defined as $-2\pi f_c\tau_c$ to be consistent with the notation in Section 6.6. Both the carrier phase θ and group delay τ_g are unknown. However, it is assumed that they remain constant over the observation interval $0 \le t \le T_0$ or through the transmission of $L_0 = T_0/T$ symbols. Equivalently, we may write (using τ in place of τ_g to simplify matters)

$$x(t) = \sqrt{\frac{2E}{T}}\cos(2\pi f_c t + \theta + \alpha_k) + w(t), \qquad \tau \le t \le T + \tau \tag{6.242}$$

*A reader who is not interested in the formal derivation of $\partial l(\theta)/\partial\theta$ may omit this subsection and move onto the next subsection without loss of continuity.

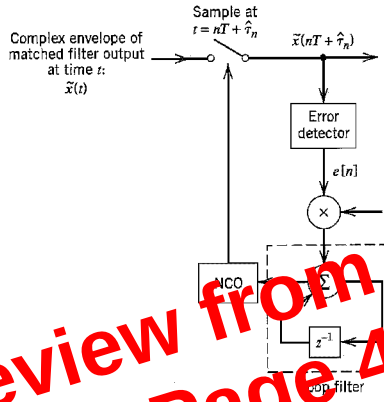

FIGURE 6.64 Nondata-aided early-late delay synchronizer.

where γ is the step-size parameter in which $2E/N_0^2$ and $1/T$ are absorbed, and the error signal $e[n]$ is defined by Equation (6.285). The $c[n]$ is a real number employed as the control for the frequency of an oscillator, referred to as a *number-controlled oscillator* (NCO). The scheme for implementing the timing recovery algorithm of Equations (6.285) and (6.286) is shown in Figure 6.64. This scheme is analogous to the continuous-time version of the early-late gate synchronizer widely used for timing recovery. It is thus referred to as a *nondata-aided early-late delay (NDA-ELD) synchronizer*. At every iteration, it works on three successive samples of the matched filter output, namely, $\tilde{x}\left(nT + \frac{T}{2} + \hat{\tau}_n\right)$, $\tilde{x}\left(nT + \hat{\tau}_n\right)$ and $\tilde{x}\left(nT + \frac{T}{2} - \hat{\tau}_{n-1}\right)$. The first sample is early and the last one is late, both with respect to the middle one.

Note that we could have simplified the derivations presented in this section by using the band-pass to complex low-pass transformation described in Appendix 2. We did not do so merely for the sake of simplifying the understanding of the material presented here.

6.15 *Computer Experiments: Carrier Recovery and Symbol Timing*

In this section we illustrate the operations of the recursive Costas loop and nondata-aided early-late delay synchronizer by considering a coherent QPSK system with the following specifications:

(i) Channel response: raised cosine (Nyquist) with rolloff factor $\alpha = 0.5$.

(ii) Loop filter: first-order digital filter with its transfer function defined by

$$H(z) = \frac{1}{z - (1 - \gamma A)} \tag{6.287}$$

where γ is the step-size parameter and A is a parameter to be defined.

(iii) Loop bandwidth, $B_L = 2\%$ of the symbol rate $1/T$; that is, $B_L T = 0.02$.

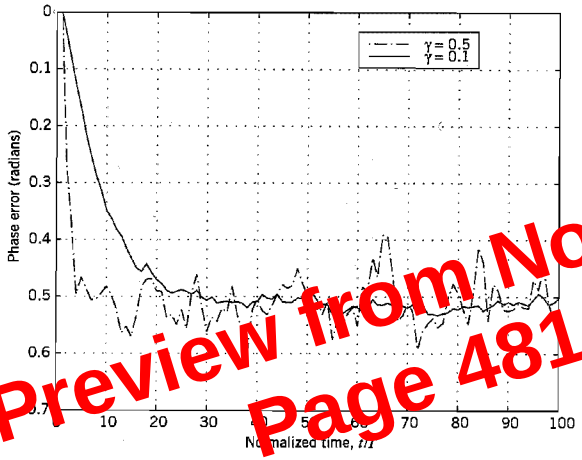

FIGURE 6.68 Effects of varying the step-size parameter on convergence behavior of the recursive Costas loop.

When steady-state conditions have been established, the estimated phase $\hat{\theta}$ will fluctuate around the true value θ. The extent of these fluctuations depends on the step-size parameter γ and the received signal-to-noise ratio:

(i) Figure 6.68 plots the phase error φ versus the normalized time t/T for two different values of step-size parameter γ, namely, 0.1 and 0.5, and fixed $E/N_0 = 20$ dB. This figure clearly shows that the smaller we make γ the smaller the steady-state fluctuations in the phase error φ will be. However, this improvement is attained at the expense of a slower rate of convergence of the algorithm. The number of iterations needed by the algorithm to reach steady-state is approximately given by

$$L_0 \simeq \frac{1}{2B_L T} \tag{6.291}$$

The normalized bandwidth $B_L T$ is itself approximately given by

$$B_L T \simeq \frac{\gamma A}{4} \tag{6.292}$$

where A is the slope of the S-curve measured at the origin. For $\gamma = 0.1$, and $B_L T = 0.02$, Equation (6.291) yields $L_0 = 25$ iterations, which checks with the solid curve plotted in Figure 6.68. Moreover, from Equations (6.291) and (6.292) we see that L_0 is inversely proportional to γ, which again checks with the results presented in Figure 6.68.

(ii) Figure 6.69 plots the phase error φ versus the normalized time t/T for three different values of E/N_0, namely, 5, 10, and 30 dB, and fixed $\gamma = 0.08$. We now see that the larger we make the signal-to-noise ratio, the smaller the steady-state fluctuations in the phase error φ will be. Moreover, the rate of convergence of the algorithm also improves with increased signal-to-noise ratio, which is intuitively satisfying.

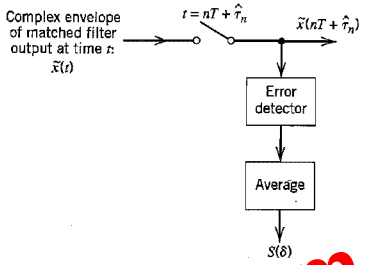

FIGURE 6.71 Scheme for measuring the *S*-curve for the recursive early-late-delay synchronizer.

In any event, the experimental and theoretical results presented in Figure 6.70 are in very close agreement for $(E/N_0) = 10$ dB.

Experiment 2: Symbol Timing Recovery

To measure the *S*-curve for the nondata-aided early-late delay synchronizer for symbol timing recovery, we may use the experimental set-up shown ion Figure 6.71, where the δ in $S(\delta)$ refers to the timing offset. The *S*-curve so measured is plotted in Figure 6.72 for $E/N_0 = 10$ dB and $E/N_0 = \infty$.

Figure 6.73 plots the normalized value of the experimentally measured symbol timing error versus E/N_0 for two different values of step-size parameter γ, namely, $T/20$ and $T/200$. This figure also includes theoretical plots of the corresponding modified Cramér-Rao bound of Equation (6.293) adapted for symbol-timing error. From the results presented here, we observe that as the step-size parameter γ is reduced, the normalized timing

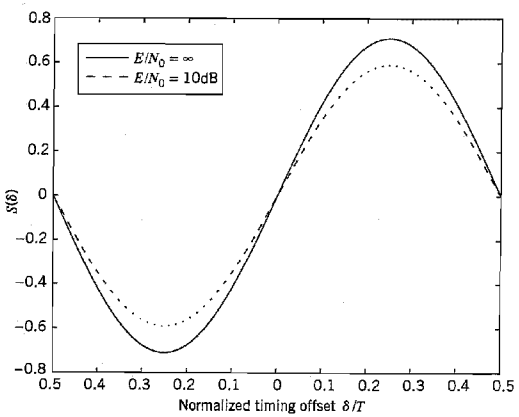

FIGURE 6.72 *S*-curve of NDA-ELD synchronizer measured under noiseless and noisy conditions.

where k_p is the phase sensitivity, and the data signal $m(t)$ takes on the value $+1$ for binary symbol 1 and -1 for binary symbol 0. The VCO output is

$$r(t) = A_c \sin[2\pi f_c t + \theta(t)]$$

(a) Evaluate the loop filter output, assuming that this filter removes only modulated components with carrier frequency $2f_c$.

(b) Show that this output is proportional to the data signal $m(t)$ when the loop is phase locked, that is, $\theta(t) = 0$.

6.4 The signal component of a coherent PSK system is defined by

$$s(t) = A_c k \sin(2\pi f_c t) \pm A_c\sqrt{1 - k^2} \cos(2\pi f_c t)$$

where $0 \leq t \leq T_b$, and the plus sign corresponds to symbol 1 and the minus sign corresponds to symbol 0. The first term represents a carrier component included for the purpose of synchronizing the receiver to the transmitter.

(a) Draw a signal-space diagram for the scheme described here. What observations can you make about this diagram?

(b) Show that, in the presence of additive white Gaussian noise of zero mean and power spectral density $N_0/2$, the average probability of error is

$$P_e = \frac{1}{2}\operatorname{erfc}\left(\sqrt{\frac{E_b}{N_0}(1 - k^2)}\right)$$

where

$$E_b = \frac{1}{2}A_c^2 T_b$$

(c) Suppose that 10 percent of the transmitted signal power is allocated to the carrier component. Determine the E_b/N_0 required to realize a probability of error equal to 10^{-4}.

(d) Compare this value of E_b/N_0 with that required for a conventional PSK system with the same probability of error.

6.5 (a) Given the input binary sequence 1100100010, sketch the waveforms of the in-phase and quadrature components of a modulated wave obtained by using the QPSK based on the signal set of Figure 6.6.

(b) Sketch the QPSK waveform itself for the input binary sequence specified in part (a).

6.6 Let P_{eI} and P_{eQ} denote the probabilities of symbol error for the in-phase and quadrature channels of a narrowband digital communication system. Show that the average probability of symbol error for the overall system is given by

$$P_e = P_{eI} + P_{eQ} - P_{eI}P_{eQ}$$

6.7 Equation (6.47) is an approximate formula for the average probability of symbol error for coherent M-ary PSK. This formula was derived using the union bound in light of the signal-space diagram of Figure 6.15*b*. Given that message point m_1 was transmitted, show that the approximate formula of Equation (6.47) may be derived directly from Figure 6.15*b*.

6.8 Find the power spectral density of an offset QPSK signal produced by a random binary sequence in which symbols 1 and 0 (represented by ± 1) are equally likely, and the symbols in different time slots are statistically independent and identically distributed.

6.9 Vestigial sideband modulation (VSB), discussed in Chapter 2, offers another modulation method for passband data transmission.

(a) In particular, a digital VSB transmission system may be viewed as a time-varying one-dimensional system operating at a rate of $2/T$ dimensions per second, where T is the symbol period. Justify the validity of this statement.

(b) Show that digital VSB is indeed equivalent in performance to the offset QPSK.

(operating in synchronism with the transmitter) to despread the received signal so that the original data sequence may be recovered.

Although standard modulation techniques such as frequency modulation and pulse-code modulation do satisfy part 1 of this definition, they are not spread-spectrum techniques because they do not satisfy part 2 of the definition.

Spread-spectrum modulation was originally developed for military applications, where resistance to jamming (interference) is of major concern. However, there are civilian applications that also benefit from the unique characteristics of spread-spectrum modulation. For example, it can be used to provide *multipath rejection* in a ground-based mobile radio environment. Yet another application is in *multiple-access* communications in which a number of independent users are required to share a common channel without an external synchronizing mechanism; here, for example, we may mention a ground-based radio environment involving mobile vehicles that must communicate with a central station. More is said about this latter application in Chapter 8.

In this chapter we discuss principles of spread-spectrum modulation, with emphasis on direct-sequence and frequency-hop techniques. In a *direct-sequence spread-spectrum* technique, two stages of modulation are used. First, the incoming data sequence is used to modulate a wideband code. This code transforms the narrowband data sequence into a noiselike wideband signal. The resulting wideband signal undergoes a second modulation using a phase-shift keying technique. In a *frequency-hop spread-spectrum* technique, on the other hand, the spectrum of a data-modulated carrier is widened by changing the carrier frequency in a pseudo-random manner. For their operation, both of these techniques rely on the availability of a noiselike spreading code called a *pseudo-random* or *pseudo-noise sequence*. Since such a sequence is basic to the operation of spread-spectrum modulation, it is logical that we begin our study by describing the generation and properties of pseudo-noise sequences.

7.2 *Pseudo-Noise Sequences*

A *pseudo-noise (PN) sequence* is a periodic binary sequence with a noiselike waveform that is usually generated by means of a *feedback shift register*, a general block diagram of which is shown in Figure 7.1. A feedback shift register consists of an ordinary *shift register* made up of m flip-flops (two-state memory stages) and a *logic circuit* that are interconnected to form a multiloop *feedback* circuit. The flip-flops in the shift register are regulated by a single timing *clock*. At each pulse (tick) of the clock, the *state* of each flip-flop is shifted to the next one down the line. With each clock pulse the logic circuit computes a

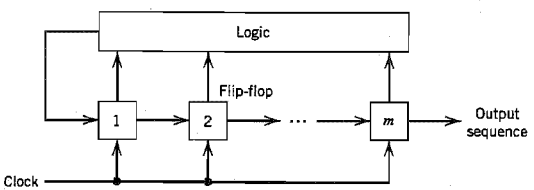

FIGURE 7.1 Feedback shift register.

Boolean function of the states of the flip-flops. The result is then fed back as the input to the first flip-flop, thereby preventing the shift register from emptying. The PN sequence so generated is determined by the length m of the shift register, its initial state, and the feedback logic.

Let $s_j(k)$ denote the state of the jth flip-flop after the kth clock pulse; this state may be represented by symbol 0 or 1. The state of the shift register after the kth clock pulse is then defined by the set $\{s_1(k), s_2(k), \ldots, s_m(k)\}$, where $k \geq 0$. For the initial state, k is zero. From the definition of a shift register, we have

$$s_j(k+1) = s_{j-1}(k), \qquad \begin{cases} k \geq 0 \\ 1 \leq j \leq \end{cases} \tag{7.1}$$

where $s_0(k)$ is the input applied to the first flip-flop after the kth clock pulse. According to the configuration described in Figure 7.1, $s_0(k)$ is a Boolean function of the individual states $s_1(k), s_2(k), \ldots, s_m(k)$. For a specified length m, this Boolean function uniquely determines the subsequent sequence of states and therefore the PN sequence produced at the output of the final flip-flop in the shift register. With a total number of m flip-flops, the number of possible states of the shift register is at most 2^m. It follows therefore that the PN sequence generated by a feedback shift register must eventually become *periodic* with a period of at most 2^m.

A feedback shift register is said to be *linear* when the feedback logic consists entirely of *modulo-2 adders*. In such a case, the *zero state* (e.g., the state for which all the flip-flops are in state 0) is *not* permitted. We say so because for a zero state, the input $s_0(k)$ produced by the feedback logic would be 0, the shift register would then continue to remain in the zero state, and the output would therefore consist entirely of 0s. Consequently, the period of a PN sequence produced by a linear feedback shift register with m flip-flops cannot exceed $2^m - 1$. When the period is exactly $2^m - 1$, the PN sequence is called a *maximal-length-sequence* or simply *m-sequence*.

EXAMPLE 7.1

Consider the linear feedback shift register shown in Figure 7.2, involving three flip-flops. The input s_0 applied to the first flip-flop is equal to the modulo-2 sum of s_1 and s_3. It is assumed that the initial state of the shift register is 100 (reading the contents of the three flip-flops from left to right). Then, the succession of states will be as follows:

100, 110, 111, 011, 101, 010, 001, 100, ...

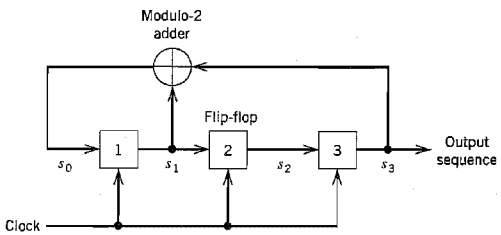

FIGURE 7.2 Maximal-length sequence generator for $m = 3$.

TABLE 7.2a *Evolution of the maximal-length sequence generated by the feedback-shift register of Fig. 7.4a*

Feedback Symbol	State of Shift Register					Output Symbol
	1	0	0	0	0	
0	0	1	0	0	0	0
1	1	0	1	0	0	0
0	0	1	0	1	0	0
1	1	0	1	0	1	0
1	1	1	0	1	0	1
1	1	1	1	0	1	0
0	0	1	1	1	0	1
1	1	0	1	1	1	0
1	1	1	0	1	1	1
0	0	1	1	0	1	1
0	0	0	1	1	0	1
0	0	0	0	1	1	0
1	1	0	0	0	1	1
1	1	1	0	0	0	1
1	1	1	1	0	0	0
1	1	1	1	1	0	0
1	1	1	1	1	1	0
0	0	1	1	1	1	1
0	0	0	1	1	1	1
1	1	0	0	1	1	1
1	1	1	0	0	1	1
0	0	1	1	0	0	1
1	1	0	1	1	0	0
0	0	1	0	1	1	0
0	0	0	1	0	1	1
1	1	0	0	1	0	1
0	0	1	0	0	1	0
0	0	0	1	0	0	1
0	0	0	0	1	0	0
0	0	0	0	0	1	0
1	1	0	0	0	0	1

Code: 000010101110110001111001101001

TABLE 7.2b *Evolution of the maximal-length sequence generated by the feedback-shift register of Fig. 7.4b*

Feedback Symbol	State of Shift Register					Output Symbol
	1	0	0	0	0	
1	1	1	0	0	0	0
0	0	1	1	0	0	0
1	1	0	1	1	0	0
0	0	1	0	1	1	0
1	1	0	1	0	1	1
0	0	1	0	1	0	1
0	0	0	1	0	1	0
1	1	0	0	1	0	1
0	0	1	0	0	1	0
0	0	0	1	0	0	1
0	0	0	0	1	0	0
1	1	0	0	0	1	0
0	0	1	0	0	0	1
1	1	0	1	0	0	0
1	1	1	0	1	0	0
1	1	1	1	0	1	0
1	1	1	1	1	0	1
1	1	1	1	1	1	0
0	0	1	1	1	1	1
1	1	0	1	1	1	1
1	1	1	0	1	1	1
0	0	1	1	0	1	1
0	0	0	1	1	0	1
1	1	0	0	1	1	0
1	1	1	0	0	1	1
1	1	1	1	0	0	1
0	0	1	1	1	0	0
0	0	0	1	1	1	0
0	0	0	0	1	1	1
0	0	0	0	0	1	1
1	1	0	0	0	0	1

Code: 0000110101001000101111101100111

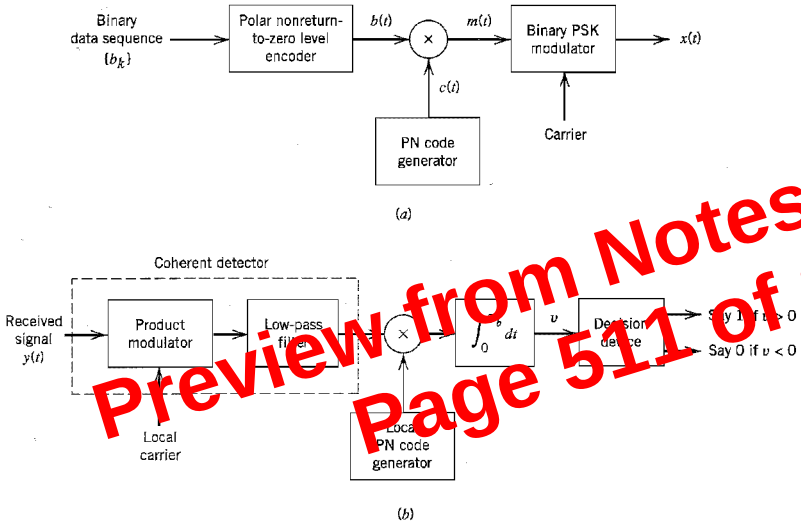

FIGURE 7.7 Direct-sequence spread coherent phase-shift keying. (*a*) Transmitter. (*b*) Receiver.

The receiver, shown in Figure 7.7*b*, consists of two stages of demodulation. In the first stage, the received signal $y(t)$ and a locally generated carrier are applied to a product modulator followed by a low-pass filter whose bandwidth is equal to that of the original message signal $m(t)$. This stage of the demodulation process reverses the phase-shift keying applied to the transmitted signal. The second stage of demodulation performs spectrum despreading by multiplying the low-pass filter output by a locally generated replica of the PN signal $c(t)$, followed by integration over a bit interval $0 \leq t \leq T_b$, and finally decision-making in the manner described in Section 7.3.

Model for Analysis

In the normal form of the transmitter, shown in Figure 7.7*a*, the spectrum spreading is performed prior to phase modulation. For the purpose of analysis, however, we find it more convenient to interchange the order of these operations, as shown in the model of

TABLE 7.3 *Truth table for phase modulation* $\theta(t)$, *radians*

		Polarity of Data Sequence $b(t)$ *at Time* t	
		+	−
Polarity of PN sequence $c(t)$ *at time* t	+	0	π
	−	π	0

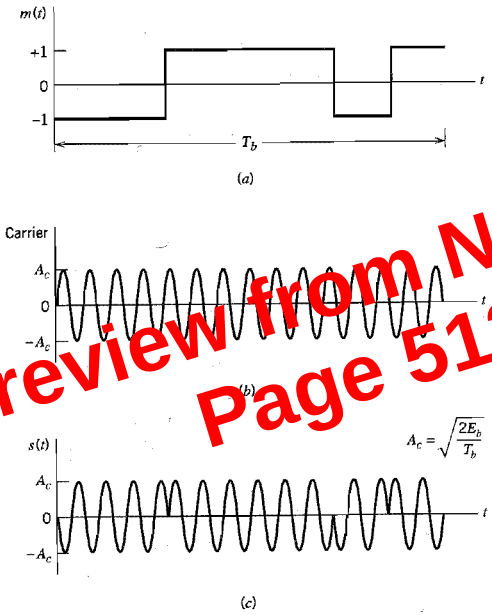

FIGURE 7.8 (*a*) Product signal $m(t) = c(t)b(t)$. (*b*) Sinusoidal carrier. (*c*) DS/BPSK signal.

Figure 7.9. We are permitted to do this because the spectrum spreading and the binary phase-shift keying are both linear operations; likewise for the phase demodulation and spectrum despreading. But for the interchange of operations to be feasible, it is important to synchronize the incoming data sequence and the PN sequence. The model of Figure 7.9 also includes representations of the channel and the receiver. In this model, it is assumed that the interference $j(t)$ limits performance, so that the effect of channel noise may be ignored. Accordingly, the channel output is given by

$$\begin{aligned} y(t) &= x(t) + j(t) \\ &= c(t)s(t) + j(t) \end{aligned} \tag{7.12}$$

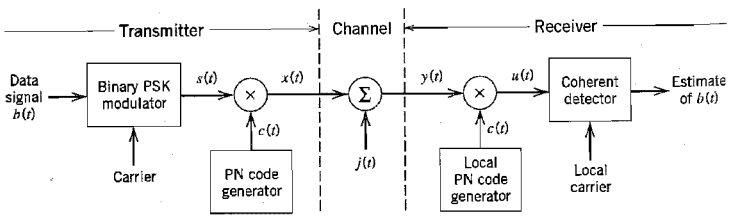

FIGURE 7.9 Model of direct-sequence spread binary PSK system.

the *output signal-to-noise ratio* as the instantaneous peak power E_b divided by the variance of the equivalent noise component in Equation (7.34). We thus write

$$(\text{SNR})_O = \frac{2E_b}{JT_c} \tag{7.35}$$

The average signal power at the receiver input equals E_b/T_b. We thus define an *input signal-to-noise* ratio as

$$(\text{SNR})_I = \frac{E_b/T_b}{J} \tag{7.36}$$

Hence, eliminating E_b/J between Equations (7.35) and (7.36), we may express the output signal-to-noise ratio in terms of the input signal-to-noise ratio as

$$(\text{SNR})_O = \frac{2T_b}{T_c}(\text{SNR})_I \tag{7.37}$$

It is customary practice to express signal-to-noise ratios in decibels. To that end, we introduce a term called the *processing gain* (PG), which is defined as *the gain in SNR obtained by the use of spread spectrum.* Specifically, we write

$$\text{PG} = \frac{T_b}{T_c} \tag{7.38}$$

which represents the gain achieved by processing a spread-spectrum signal over an unspread signal. We may thus write Equation (7.37) in the equivalent form:

$$10\log_{10}(\text{SNR})_O = 10\log_{10}(\text{SNR})_I + 3 + 10\log_{10}(\text{PG}) \text{ dB} \tag{7.39}$$

The 3-dB term on the right-hand side of Equation (7.39) accounts for the gain in SNR that is obtained through the use of coherent detection (which presumes exact knowledge of the signal phase by the receiver). This gain in SNR has nothing to do with the use of spread spectrum. Rather, it is the last term, $10\log_{10}(\text{PG})$, that accounts for the processing gain. Note that both the processing gain PG and the spread factor N (i.e., PN sequence length) equal the ratio T_b/T_c. Thus, the longer we make the PN sequence (or, correspondingly, the smaller the chip time T_c is), the larger will the processing gain be.

7.6 Probability of Error

Let the coherent detector output v in the direct-sequence spread BPSK system of Figure 7.9 denote the sample value of a random variable V. Let the equivalent noise component v_{cj} produced by external interference denote the sample value of a random variable V_{cj}. Then, from Equations (7.23) and (7.27) we deduce that

$$V = \pm\sqrt{E_b} + V_{cj} \tag{7.40}$$

where E_b is the transmitted signal energy per bit. The plus sign refers to sending symbol (information bit) 1, and the minus sign refers to sending symbol 0. The decision rule used by the coherent detector of Figure 7.9 is to declare that the received bit in an interval $(0, T_b)$ is 1 if the detector output exceeds a threshold of zero, and that it is 0 if the detector output is less than the threshold; if the detector output is exactly zero, the receiver makes a random guess in favor of 1 or 0. With both information bits assumed equally likely, we

Using the definition of Equation (7.38) for the processing gain PG we may reformulate this result as

$$\frac{J}{P} = \frac{\text{PG}}{E_b/N_0} \tag{7.46}$$

The ratio J/P is termed the *jamming margin*. Accordingly, the jamming margin and the processing gain, both expressed in decibels, are related by

$$(\text{Jamming margin})_{\text{dB}} = (\text{Procesing gain})_{\text{dB}} - 10 \log_{10}\left(\frac{E_b}{N_0}\right)_{\text{min}} \tag{7.47}$$

where $(E_b/N_0)_{\text{min}}$ is the minimum value needed to support the prescribed average probability of error.

EXAMPLE 7.3

A spread-spectrum communication system has the following parameters:

Information bit duration, $T_b = 4.095$ ms

PN chip duration, $T_c = 1\ \mu$s

Hence, using Equation (7.38) we find that the processing gain is

$$\text{PG} = 4095$$

Correspondingly, the required period of the PN sequence is $N = 4095$, and the shift-register length is $m = 12$.

For a satisfactory reception, we may assume that the average probability of error is not to exceed 10^{-5}. From the formula for a coherent binary PSK receiver, we find that $E_b/N_0 = 10$ yields an average probability of error equal to 0.387×10^{-5}. Hence, using this value for E_b/N_0, and the value calculated for the processing gain, we find from Equation (7.47) that the jamming margin is

$$\begin{aligned}(\text{Jamming margin})_{\text{dB}} &= 10 \log_{10} 4095 - 10 \log_{10}(10) \\ &= 36.1 - 10 \\ &= 26.1 \text{ dB}\end{aligned}$$

That is, information bits at the receiver output can be detected reliably even when the noise or interference at the receiver input is up to 409.5 times the received signal power. Clearly, this is a powerful advantage against interference (jamming), which is realized through the clever use of spread-spectrum modulation. ◀

7.7 *Frequency-Hop Spread Spectrum*

In the type of spread-spectrum systems discussed in Section 7.4, the use of a PN sequence to modulate a phase-shift-keyed signal achieves *instantaneous* spreading of the transmission bandwidth. The ability of such a system to combat the effects of jammers is determined by the processing gain of the system, which is a function of the PN sequence period. The processing gain can be made larger by employing a PN sequence with narrow chip duration, which, in turn, permits a greater transmission bandwidth and more chips per bit. However, the capabilities of physical devices used to generate the PN spread-spectrum signals impose a practical limit on the attainable processing gain. Indeed, it may turn out that the processing gain so attained is still not large enough to overcome the effects of

A receiver based on the second procedure is optimum in the sense that it minimizes the average probability of symbol error for a given E_b/N_0.

▶ EXAMPLE 7.5

Figure 7.12*a* illustrates the variation of the transmitted frequency of a fast FH/MFSK signal with time. The signal has the following parameters:

Number of bits per MFSK symbol	$K = 2$
Number of MFSK tones	$M = 2^K = 4$
Length of PN segment per hop	$k = 3$
Total number of frequency hops	$2^k = 8$

In this example, each MFSK symbol has the same number of bits and chips, that is, the chip rate R_c is identical to the bit rate R_b. After each chip, the carrier frequency of the transmitted MFSK signal is hopped to a different value except for the occasions when the k-chip segment of the PN sequence repeats itself.

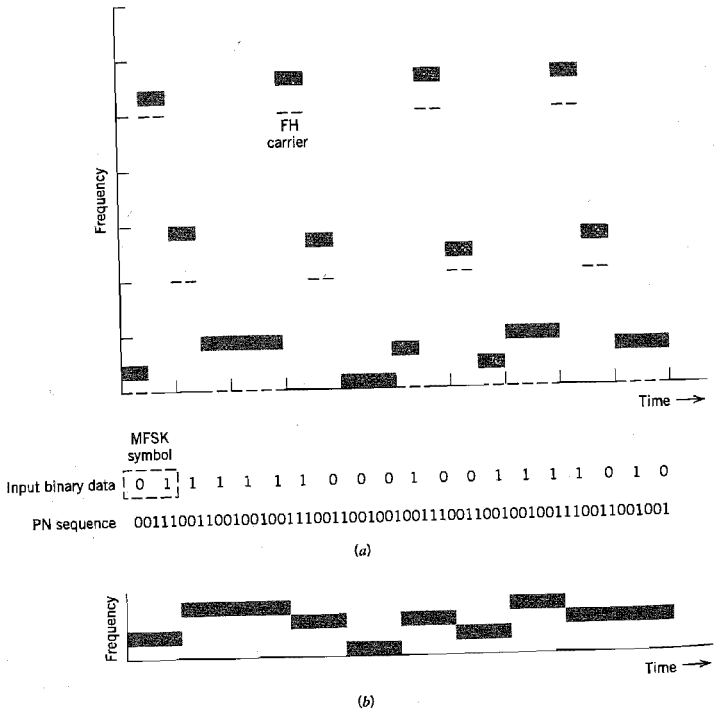

FIGURE 7.12 Illustrating fast-frequency hopping. (*a*) Variation of the transmitter frequency with time. (*b*) Variation of the dehopped frequency with time.

7.9 A direct-sequence spread binary phase-shift keying system uses a feedback shift register of length 19 for the generation of the PN sequence. Calculate the processing gain of the system.

7.10 In a DS/BPSK system, the feedback shift register used to generate the PN sequence has length $m = 19$. The system is required to have an average probability of symbol error due to externally generated interfering signals that does not exceed 10^{-5}. Calculate the following system parameters in decibels:

(a) Processing gain.

(b) Antijam margin.

7.11 In Section 7.5, we presented an analysis on the signal-space dimensionality and processing gain of a direct sequence spread-spectrum system using binary phase-shift keying. Extend the analysis presented therein to the case of such a system using quadriphase-shift keying.

Frequency-Hop Spread Spectrum

7.12 A slow FH/MFSK system has the following parameters:

Number of bits per MFSK symbol = 4

Number of MFSK symbols per hop = 5

Calculate the processing gain of the system.

7.13 A fast FH/MFSK system has the following parameters:

Number of bits per MFSK symbol = 4

Number of hops per MFSK symbol = 4

Calculate the processing gain of the system.

Computer Experiments

7.14 Consider two PN sequences of period $N = 63$. One sequence has the feedback taps [6, 1] and the other sequence has the feedback taps [6, 5, 2, 1], which are picked in accordance with Table 7.1.

(a) Compute the autocorrelation function of these two sequences, and their cross-correlation function.

(b) Compare the cross-correlation function computed in part (a) with the cross-correlation function between the sequence [6, 5 2, 1] and its mirror image [6, 5, 4, 1].

Comment on your results.

7.15 (a) Compute the partial cross-correlation function of a PN sequence with feedback taps [5, 2] and its image sequence defined by the feedback taps [5, 3].

(b) Repeat the computation for the PN sequence with feedback taps [5, 2] and the PN sequence with feedback taps [5, 4, 2, 1].

(c) Repeat the computation for the PN sequence with feedback taps [5, 4, 3, 2] and the PN sequence with feedback taps [5, 4, 2, 1].

The feedback taps [5, 2], [5, 4, 3, 2], and [5, 4, 2, 1] are possible taps for a maximal-length sequence of period 31, in accordance with Table 7.1.

CHAPTER 8

MULTIUSER RADIO COMMUNICATIONS

As its name implies, multiuser communications refers to the simultaneous use of a communication channel by a number of users. In this chapter, we discuss multiuser communication systems that rely on radio propagation for linking the receivers to the transmitters.

In particular, we focus on the following topics:

- *Multiple-access techniques, which are basic to multiuser communication systems.*
- *Satellite communications, offering global coverage.*
- *Radio link analysis, highlighting the roles of transmitting and receiving antennas and free-space propagation.*
- *Wireless communications with emphasis on mobility and the multipath phenomenon.*
- *Speech coding for wireless communications.*
- *Adaptive antennas for wireless communications.*

8.1 *Introduction*

Much of the material on communication theory presented in earlier chapters has been based on a particular idealization of the communication channel, namely, a *channel model limited in bandwidth and corrupted by additive white Gaussian noise* (AWGN). The *classical communication theory* so developed is mathematically elegant, providing a sound introduction to the ever-expanding field of communication systems. An example of a physical channel that is well represented by such a model is the satellite communications channel. It is therefore befitting that the first type of multiuser communications discussed in this chapter is *satellite communications*.

A satellite communication system in geostationary orbit relies on line-of-sight radio propagation for the operation of its uplink from an earth terminal to the transponder and the downlink from the transponder to another earth terminal. Thus the discussion of satellite communications naturally leads to the analysis of radio propagation in free space, linking a receiving antenna to a transmitting antenna.

The use of satellite communications offers *global coverage*. The other multiuser communication system studied in this chapter, namely, *wireless communications*, offers *mobility* which, in conjunction with existing telephone networks and satellite communication systems, permits a mobile unit to communicate with anyone, anywhere in the world. Another characteristic feature of wireless communication systems is that they are *tetherless*

However, insofar as link calculations are concerned, such a complete knowledge is not necessary. Rather, it is sufficient to merely specify the variation of the power density for the antenna.

By definition, the *Poynting vector* or *power density is the rate of energy flow per unit area*; it has the dimensions of watts per square meter. The treatment of the transmitting antenna as a point source greatly simplifies matters in that the power density of a point source has only a radial component; that is, the radiated energy streams from the source along radial lines.

It is useful to have a "reference" antenna against which the performance of the transmitting and receiving antennas can be compared. The customary practice is to assume that the reference antenna is an *isotropic source*, defined as an *omnidirectional (i.e., completely nondirectional) antenna that radiates uniformly in all directions*. An isotropic source is hypothetical because, in reality, all radiating antennas have some directivity however small. Nonetheless, the notion of an isotropic source is useful, especially for gain comparison purposes.

Consider then an isotropic source radiating total power denoted by P_t, measured in watts. The radiated power passes uniformly through a sphere of surface area $4\pi d^2$, where d is the distance (in meters) from the source. Hence, the power density, denoted by $\rho(d)$, at any point on the surface of the sphere is given by

$$\rho(d) = \frac{P_t}{4\pi d^2} \quad \text{watts/m}^2 \tag{8.3}$$

Equation (8.3) states that the power density varies inversely as the square of the distance from a point source. This statement is the familiar *inverse-square law* that governs the propagation of electromagnetic waves in free space.

Multiplying the power density $\rho(d)$ by the square of the distance d at which it is measured, we get a quantity called *radiation intensity* denoted by Φ. We may thus write

$$\Phi = d^2\rho(d) \tag{8.4}$$

Whereas the power density $\rho(d)$ is measured in watts per square meter, the radiation intensity Φ is measured in watts per unit solid angle (watts per steradian).

In the case of a typical transmitting or receiving radio antenna, the radiation intensity is a function of the spherical coordinates θ and ϕ defined in Figure 8.5. Thus, in general,

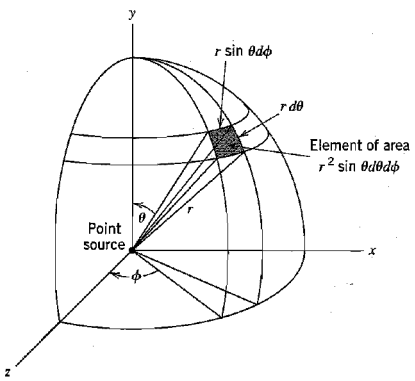

FIGURE 8.5 Illustrating the spherical coordinates of a point source.

we may express the radiation intensity as $\Phi(\theta, \phi)$, and so speak of a *radiation-intensity pattern*. The power radiated inside an infinitesimal solid angle $d\Omega$ is given by $\Phi(\theta, \phi)\, d\Omega$, where (referring to Figure 8.5)

$$d\Omega = \sin\theta\, d\theta\, d\phi \quad \text{steradians} \tag{8.5}$$

The total power radiated is therefore

$$P = \int \Phi(\theta, \phi)\, d\Omega \quad \text{watts} \tag{8.6}$$

which is a mathematical statement of the *power theorem*. In words, the power theorem states that if the radiation-intensity pattern $\Phi(\theta, \phi)$ is known for all values of angle pair (θ, ϕ), then the total power radiated is given by the integral of $\Phi(\theta, \phi)$ over a solid angle of 4π steradians. The *average* power radiated per unit solid angle is

$$P_{av} = \frac{1}{4\pi}\int \Phi(\theta, \phi)\, d\Omega = \frac{P}{4\pi} \quad \text{watts/steradian} \tag{8.7}$$

which represents the radiation intensity that is produced by an isotropic source radiating the same total power P.

Directive Gain, Directivity, and Power Gain[3]

Now the ability of an antenna to concentrate the radiated power in a given direction as in the case of the transmitting antenna or, conversely, to effectively absorb the incident power from that direction as in the case of the receiving antenna, is specified in terms of its directive gain or directivity. For a direction specified by the angle pair (θ, ϕ), the *directive gain* of an antenna, denoted by $g(\theta, \phi)$ is defined as *the ratio of the radiation intensity in that direction to the average radiated power*, as shown by

$$g(\theta, \phi) = \frac{\Phi(\theta, \phi)}{P_{av}} = \frac{\Phi(\theta, \phi)}{P/4\pi} \tag{8.8}$$

The *directivity* of an antenna, denoted by D, is defined as *the ratio of the maximum radiation intensity from the antenna to the radiation intensity from an isotropic source.* That is, the directivity D is the maximum value of the directive gain $g(\theta, \phi)$. Thus, whereas the directive gain of the antenna is a function of the angle pair (θ, ϕ), the directivity D is a constant that has been maximized for a particular direction.

The definition of directivity is based on the shape of the radiation-intensity pattern $\Phi(\theta, \phi)$; as such, it does not involve the effect of antenna imperfections due to dissipation loss and impedance mismatch. A quantity called *power gain* does involve the radiation efficiency of the antenna. Specifically, the power gain of an antenna, denoted by G, is defined as *the ratio of the maximum radiation intensity from the antenna to the radiation intensity from a lossless isotropic source, under the constraint that the same input power is applied to both antennas.* Specifically, using $\eta_{\text{radiation}}$ to denote the *radiation efficiency factor* of the antenna, we may relate the power gain G to the directivity D as

$$G = \eta_{\text{radiation}} D \tag{8.9}$$

Thus, the power gain of an antenna over a lossless isotropic source equals the directivity if the antenna is 100 percent efficient (i.e., $\eta_{\text{radiation}} = 1$), but it is less than the directivity

compared to the reciprocal of the spread in propagation path delays. Multipath in such an environment results in two effects: rapid fading of the received signal envelope and a spread in Doppler shifts in the received spectrum. Real-life signals radiated in a mobile radio environment may, however, occupy a bandwidth wide enough to require more detailed considerations of the effects of multipath propagation on the received signal. In this section, we present a statistical characterization of a mobile radio channel.*

Consider a mobile radio channel with multiple propagation paths. In accordance with the complex notation described in Appendix 2, we may express the transmitted bandpass signal as

$$s(t) = \text{Re}[\tilde{s}(t) \exp(j2\pi f_c t)] \tag{8.37}$$

where $\tilde{s}(t)$ is the complex (low-pass) envelope of $s(t)$, and f_c is a nominal carrier frequency. Since the channel is time varying due to multipath effects, the impulse response of the channel is delay dependent and therefore a time-varying function. Let the impulse response of the channel be expressed as

$$h(\tau;t) = \text{Re}[\tilde{h}(\tau;t) \exp(j2\pi f_c t)] \tag{8.38}$$

where $\tilde{h}(\tau;t)$ is the (low-pass) complex impulse response of the channel, and τ is a delay variable. The complex impulse response $\tilde{h}(\tau;t)$ is called the *input delay-spread function* of the channel. The (low-pass) complex envelope of the channel output is defined by the convolution integral

$$\tilde{s}_o(t) = \frac{1}{2}\int_{-\infty}^{\infty} \tilde{s}(t-\tau)\tilde{h}(\tau;t)\, d\tau \tag{8.39}$$

where the scaling factor $\frac{1}{2}$ is the result of using complex notation.

In general, the behavior of a mobile radio channel can be described only in statistical terms. For analytic purposes, the delay-spread function $\tilde{h}(\tau;t)$ may thus be modeled as a zero-mean complex-valued Gaussian process. Then, at any time t the envelope $|\tilde{h}(\tau;t)|$ is Rayleigh distributed, and the channel is referred to as a *Rayleigh fading channel*. When, however, the mobile radio environment includes *fixed* scatterers, we are no longer justified in using a zero-mean model to describe the input delay-spread function $\tilde{h}(\tau;t)$. In such a case, it is more appropriate to use a Rician distribution to describe the envelope $|\tilde{h}(\tau;t)|$, and the channel is referred to as a *Rician fading channel*. The Rayleigh and Rician distributions for a real-valued random process were considered in Chapter 1. In the discussion presented in this chapter, we consider only a Rayleigh fading channel.

The *time-varying transfer function* of the channel is defined as the Fourier transform of the input delay-spread function $\tilde{h}(\tau;t)$ with respect to the delay variable τ, as shown by

$$\tilde{H}(f;t) = \int_{-\infty}^{\infty} \tilde{h}(\tau;t) \exp(-j2\pi f\tau)\, d\tau \tag{8.40}$$

where f denotes the frequency variable. The time-varying transfer function $\tilde{H}(f;t)$ may be viewed as a frequency transmission characteristic of the channel.

*Readers who are not interested in the mathematical details pertaining to the statistical characterization of fading multipath channels, may skip the material presented in this section, except for the subsection on the classification of multipath channels at the end of the section.

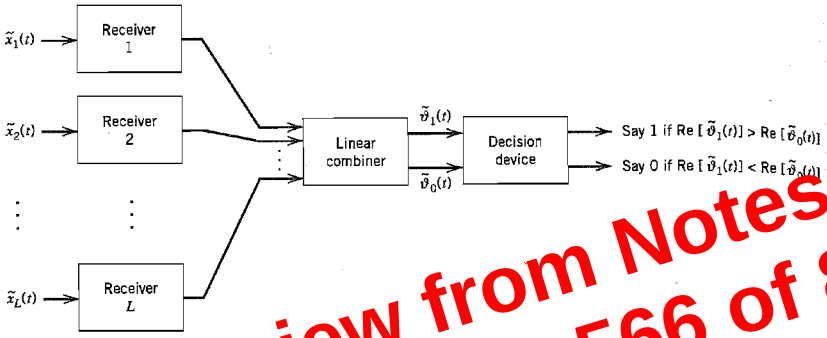

FIGURE 8.23 Block diagram illustrating the space diversity technique.

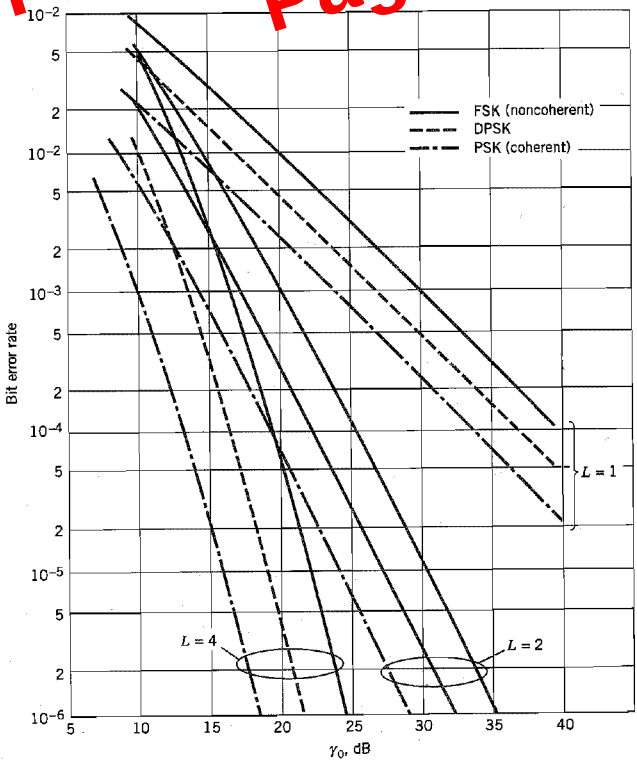

FIGURE 8.24 Performance of binary signaling schemes with diversity. (From Proakis, 1995, with permission of McGraw-Hill.)

a period of 10 to 30 ms, during which the speech signal is treated as pseudo-stationary.

- The optimum excitation for the synthesis filter is computed by minimizing the perceptually weighted error with the loop closed as in Figure 8.27*a*.

Thus the speech samples are divided into *frames* (10 to 30 ms long) for computing the filter parameters, and each frame is divided further into *subframes* (5 to 15 ms) for optimizing the excitation. The quantized filter parameters and quantized excitation constitute the transmitted signal.

Note that by first permitting the filter parameters to vary from one frame to the next, and then permitting the excitation to vary from one subframe to the next, the encoder is enabled to track the nonstationary behavior of speech on a batch-by-batch basis.

The decoder, located in the receiver, consists simply of two parts: excitation generator and synthesis filter, as shown in Figure 8.27*b*. These two parts are identical to the corresponding ones in the encoder. The function of the decoder is to use the received signal to produce a synthetic version of the original speech signal. This is achieved by passing the decoded excitation through the synthesis filter whose parameters are set equal to those in the encoder.

To reduce the computational complexity of the codec (i.e., contraction of coder/decoder), the intervals between the individual pulses in the excitation are constrained to assume a common value. The resulting analysis-by-synthesis codec is said to have a *regular-pulse excitation.*

■ CODE-EXCITED LPC

Figure 8.28 shows the block diagram of the *code-excited LPC*, commonly referred to as CELP. The distinguishing feature of CELP is the use of a predetermined *codebook* of stochastic (zero-mean white Gaussian) vectors as the source of excitation for the synthesis filter. The synthesis filter itself consists of two all-pole filters connected in cascade, one of which performs short-term prediction and the other performs long-term prediction.

As with the multi-pulse excited LPC, the free parameters of the synthesis filter are computed first, using the actual speech samples as input. Next, the choice of a particular vector (code) stored in the excitation codebook and the gain factor G in Figure 8.28 is optimized by minimizing the average power of the perceptually weighted error between

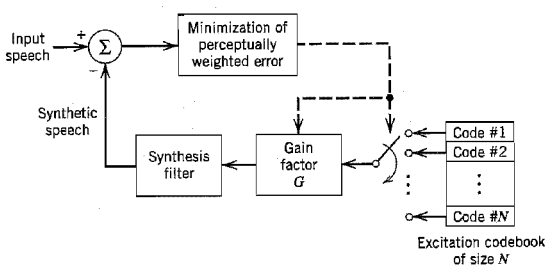

FIGURE 8.28 Encoder of the code-excited linear predictive codec (CELP): the transmitted signal consists of the address of the code selected from the codebook, quantized G, and quantized filter parameters.

8.17 In this problem we study the maximal-ratio combining diversity scheme. To proceed, consider a set of noisy signals $\{x_j(t)\}_{j=1}^{N}$, where $x_j(t)$ is defined by

$$x_j(t) = s_j(t) + n_j(t), \qquad j = 1, 2, \ldots, N$$

Assume the following:

- The signal components $s_j(t)$ are locally coherent, that is,

$$s_j(t) = z_j m(t), \qquad j = 1, 2, \ldots, N$$

where the z_j are positive real numbers, and $m(t)$ denotes a message signal with unit power.

- The noise components $n_j(t)$ have zero mean, and they are statistically independent, that is,

$$E[n_j(t)n_k(t)] = \begin{cases} \sigma_j^2 & \text{for } k = \\ & \text{otherwise} \end{cases}$$

The output of the linear combiner is defined by

$$y(t) = \sum_{j=1}^{N} \alpha_j x_j(t)$$

where the parameters α_j are to be determined.

(a) Show that the output signal-to-noise ratio is

$$(\text{SNR})_O = \frac{\left(\sum_{j=1}^{N} \alpha_j z_j\right)^2}{\sum_{j=1}^{N} \alpha_j^2 \sigma_j^2}$$

(b) Set

$$u_j = \alpha_j \sigma_j$$

$$v_j = \frac{z_j}{\sigma_j}$$

and reformulate the expression for $(\text{SNR})_O$. Hence, applying the Schwarz inequality to this reformulation, show that

(i) $(\text{SNR})_O \leq \sum_{j=1}^{N} (\text{SNR})_j$

where $(\text{SNR})_j = z_j^2/\sigma_j^2$.

(ii) The optimum values of the combiner's coefficients are defined by

$$\alpha_j = \frac{z_j}{\sigma_j^2}$$

in which case the Schwarz inequality is satisfied with the equality sign.

The Schwarz inequality is discussed in Section 5.2.

Adaptive Antenna Arrays

8.18 Consider the array signal processor of Figure 8.29 where there are only two users ($N = 2$) and the array consists of two elements ($M = 2$). Construct the subspace $\mathcal{W}$ for this problem. Hence, using a signal-space diagram, illustrate the computation of the weight characterizing the array signal processor.

of the extended source, is equal to n times $H(\mathcal{S})$, the entropy of the original source. That is, we may write

$$H(\mathcal{S}^n) = nH(\mathcal{S}) \tag{9.17}$$

▶ EXAMPLE 9.2 Entropy of Extended Source

Consider a discrete memoryless source with source alphabet $\mathcal{S} = \{s_0, s_1, s_2\}$ with respective probabilities

$$p_0 = \frac{1}{4}$$
$$p_1 = \frac{1}{4}$$
$$p_2 = \frac{1}{2}$$

Hence, the use of Equation (9.9) yields the entropy of the source as

$$\begin{aligned} H(\mathcal{S}) &= p_0 \log_2\left(\frac{1}{p_0}\right) + p_1 \log_2\left(\frac{1}{p_1}\right) + p_2 \log_2\left(\frac{1}{p_2}\right) \\ &= \frac{1}{4}\log_2(4) + \frac{1}{4}\log_2(4) + \frac{1}{2}\log_2(2) \\ &= \frac{3}{2} \text{ bits} \end{aligned}$$

Consider next the second-order extension of the source. With the source alphabet $\mathcal{S}$ consisting of three symbols, it follows that the source alphabet $\mathcal{S}^2$ of the extended source has nine symbols. The first row of Table 9.1 presents the nine symbols of $\mathcal{S}^2$, denoted as σ_0, $\sigma_1, \ldots, \sigma_8$. The second row of the table presents the composition of these nine symbols in terms of the corresponding sequences of source symbols s_0, s_1, and s_2, taken two at a time. The probabilities of the nine source symbols of the extended source are presented in the last row of the table. Accordingly, the use of Equation (9.9) yields the entropy of the extended source as

$$\begin{aligned} H(\mathcal{S}^2) &= \sum_{i=0}^{8} p(\sigma_i) \log_2 \frac{1}{p(\sigma_i)} \\ &= \frac{1}{16}\log_2(16) + \frac{1}{16}\log_2(16) + \frac{1}{8}\log_2(8) + \frac{1}{16}\log_2(16) \\ &\quad + \frac{1}{16}\log_2(16) + \frac{1}{8}\log_2(8) + \frac{1}{8}\log_2(8) + \frac{1}{8}\log_2(8) + \frac{1}{4}\log_2(4) \\ &= 3 \text{ bits} \end{aligned}$$

We thus see that $H(\mathcal{S}^2) = 2H(\mathcal{S})$ in accordance with Equation (9.17). ◀

TABLE 9.1 *Alphabet particulars of second-order extension of a discrete memoryless source*

Symbols of $\mathcal{S}^2$	σ_0	σ_1	σ_2	σ_3	σ_4	σ_5	σ_6	σ_7	σ_8
Corresponding sequences of symbols of $\mathcal{S}$	s_0s_0	s_0s_1	s_0s_2	s_1s_0	s_1s_1	s_1s_2	s_2s_0	s_2s_1	s_2s_2
Probability $p(\sigma_i)$, $i = 0, 1, \ldots, 8$	$\frac{1}{16}$	$\frac{1}{16}$	$\frac{1}{8}$	$\frac{1}{16}$	$\frac{1}{16}$	$\frac{1}{8}$	$\frac{1}{8}$	$\frac{1}{8}$	$\frac{1}{4}$

where the factor 2 refers to the radix (number of symbols) in the binary alphabet. It is important to note, however, that the Kraft–McMillan inequality does *not* tell us that a source code is a prefix code. Rather, it is merely a condition on the code-word lengths of the code and not on the code words themselves. For example, referring to the three codes listed in Table 9.2, we note the following:

- Code I violates the Kraft–McMillan inequality; it cannot therefore be a prefix code.
- The Kraft–McMillan inequality is satisfied by both codes II and III; but only code III is a prefix code.

Prefix codes are distinguished from other uniquely decodable codes by the fact that the end of a code word is always recognizable. Hence, the decoding of a prefix code can be accomplished as soon as the binary sequence representing a source symbol is fully received. For this reason, prefix codes are also referred to as *instantaneous codes*.

Given a discrete memoryless source of entropy $H(\mathcal{S})$, a prefix code can be constructed with an average code-word length $\overline{L}$, which is bounded as follows:

$$H(\mathcal{S}) \leq \overline{L} < H(\mathcal{S}) + 1 \tag{9.23}$$

The left-hand bound of Equation (9.23) is satisfied with equality under the condition that symbol s_k is emitted by the source with probability

$$p_k = 2^{-l_k} \tag{9.24}$$

where l_k is the length of the code word assigned to source symbol s_k. We then have

$$\sum_{k=0}^{K-1} 2^{-l_k} = \sum_{k=0}^{K-1} p_k = 1$$

Under this condition, the Kraft–McMillan inequality of Equation (9.22) tells us that we can construct a prefix code, such that the length of the code word assigned to source symbol s_k is $-\log_2 p_k$. For such a code, the average code-word length is

$$\overline{L} = \sum_{k=0}^{K-1} \frac{l_k}{2^{l_k}} \tag{9.25}$$

and the corresponding entropy of the source is

$$\begin{aligned} H(\mathcal{S}) &= \sum_{k=0}^{K-1} \left(\frac{1}{2^{l_k}}\right) \log_2(2^{l_k}) \\ &= \sum_{k=0}^{K-1} \frac{l_k}{2^{l_k}} \end{aligned} \tag{9.26}$$

Hence, in this special (rather meretricious) case, we find from Equations (9.25) and (9.26) that the prefix code is *matched* to the source in that $\overline{L} = H(\mathcal{S})$.

But how do we match the prefix code to an arbitrary discrete memoryless source? The answer to this problem lies in the use of an *extended code*. Let $\overline{L}_n$ denote the average code-word length of the extended prefix code. For a uniquely decodable code, $\overline{L}_n$ is the smallest possible. From Equation (9.23), we deduce that

$$H(\mathcal{S}^n) \leq \overline{L}_n < H(\mathcal{S}^n) + 1 \tag{9.27}$$

Substituting Equation (9.17) for an extended source into Equation (9.27), we get

$$nH(\mathcal{S}) \leq \overline{L}_n < nH(\mathcal{S}) + 1$$

The first row shown in this figure merely indicates the numerical positions of the individual subsequences in the code book. We now recognize that the first subsequence of the data stream, 00, is made up of the concatenation of the *first* code book entry, 0, with itself; it is therefore represented by the number 11. The second subsequence of the data stream, 01, consists of the *first* code book entry, 0, concatenated with the *second* code book entry, 1; it is therefore represented by the number 12. The remaining subsequences are treated in a similar fashion. The complete set of numerical representations for the various subsequences in the code book is shown in the third row of Figure 9.6. As a further example illustrating the composition of this row, we note that the subsequence 010 consists of the concatenation of the subsequence 01 in position 4 and symbol 0 in position 1; hence, the numerical representation 41. The last row shown in Figure 9.6 is the binary encoded representation of the different subsequences of the data stream.

The last symbol of each subsequence in the code book (i.e., the second row of Figure 9.6) is an *innovation symbol*, which is so called in recognition of the fact that its appendage to a particular subsequence distinguishes it from all previous subsequences stored in the code book. Correspondingly, the last bit of each uncoded block of bits in the binary encoded representation of the data stream (i.e., the fourth row in Figure 9.6) represents the innovation symbol for the particular subsequence under consideration. The remaining bits provide the equivalent binary representation of the "pointer" to the *root subsequence* that matches the one in question except for the innovation symbol.

The decoder is just as simple as the encoder. Specifically, it uses the pointer to identify the root subsequence and then appends the innovation symbol. Consider, for example, the binary encoded block 1101 in position 9. The last bit, 1, is the innovation symbol. The remaining bits, 110, point to the root subsequence 10 in position 6. Hence, the block 1101 is decoded into 101, which is correct.

From the example described here, we note that, in contrast to Huffman coding, the Lempel–Ziv algorithm uses fixed-length codes to represent a variable number of source symbols; this feature makes the Lempel–Ziv code suitable for synchronous transmission. In practice, fixed blocks of 12 bits long are used, which implies a code book of 4096 entries.

For a long time, Huffman coding was unchallenged as the algorithm of choice for data compaction. However, the Lempel–Ziv algorithm has taken over almost completely from the Huffman algorithm. The Lempel–Ziv algorithm is now the standard algorithm for file compression. When it is applied to ordinary English text, the Lempel–Ziv algorithm achieves a compaction of approximately 55 percent. This is to be contrasted with a compaction of approximately 43 percent achieved with Huffman coding. The reason for this behavior is that, as mentioned previously, Huffman coding does not take advantage of the intercharacter redundancies of the language. On the other hand, the Lempel–Ziv algorithm is able to do the best possible compaction of text (within certain limits) by working effectively at higher levels.

9.5 *Discrete Memoryless Channels*

Up to this point in the chapter, we have been preoccupied with discrete memoryless sources responsible for information generation. We next consider the issue of information transmission, with particular emphasis on reliability. We start the discussion by considering a discrete memoryless channel, the counterpart of a discrete memoryless source.

A *discrete memoryless channel* is a statistical model with an input X and an output Y that is a *noisy* version of X; both X and Y are random variables. Every unit of time, the

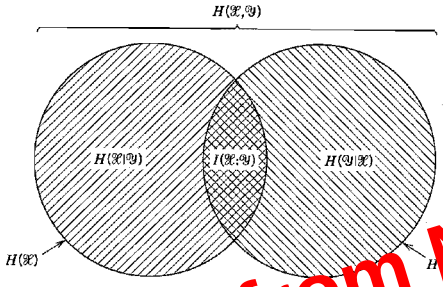

FIGURE 9.9 Illustrating the relation among various channel entropies.

The first double summation term on the right-hand side of Equation (9.56) is recognized as the negative of the mutual information of the channel, $I(\mathcal{X}; \mathcal{Y})$, previously given in Equation (9.52). As for the second summation term, we manipulate it as follows:

$$\begin{aligned}\sum_{j=0}^{J-1}\sum_{k=0}^{K-1} p(x_j, y_k)\log_2\left[\frac{1}{p(x_j)p(y_k)}\right] &= \sum_{j=0}^{J-1}\log_2\left[\frac{1}{p(x_j)}\right]\sum_{k=0}^{K-1} p(x_j, y_k) \\ &\quad + \sum_{k=0}^{K-1}\log_2\left[\frac{1}{p(y_k)}\right]\sum_{j=0}^{J-1} p(x_j, y_k) \\ &= \sum_{j=0}^{J-1} p(x_j)\log_2\left[\frac{1}{p(x_j)}\right] \\ &\quad + \sum_{k=0}^{K-1} p(y_k)\log_2\left[\frac{1}{p(y_k)}\right] \\ &= H(\mathcal{X}) + H(\mathcal{Y})\end{aligned} \tag{9.57}$$

Accordingly, using Equations (9.52) and (9.57) in Equation (9.56), we get the result

$$H(\mathcal{X}, \mathcal{Y}) = -I(\mathcal{X}; \mathcal{Y}) + H(\mathcal{X}) + H(\mathcal{Y}) \tag{9.58}$$

Rearranging terms in this equation, we get the result given in Equation (9.54), thereby confirming Property 3.

We conclude our discussion of the mutual information of a channel by providing a diagramatic interpretation of Equations (9.43), (9.44), and (9.54). The interpretation is given in Figure 9.9. The entropy of channel input X is represented by the circle on the left. The entropy of channel output Y is represented by the circle on the right. The mutual information of the channel is represented by the overlap between these two circles.

9.7 Channel Capacity

Consider a discrete memoryless channel with input alphabet $\mathcal{X}$, output alphabet $\mathcal{Y}$, and transition probabilities $p(y_k|x_j)$, where $j = 0, 1, \ldots, J-1$ and $k = 0, 1, \ldots, K-1$. The mutual information of the channel is defined by the first line of Equation (9.49), which is reproduced here for convenience:

$$I(\mathcal{X}; \mathcal{Y}) = \sum_{k=0}^{K-1}\sum_{j=0}^{J-1} p(x_j, y_k)\log_2\left[\frac{p(y_k|x_j)}{p(y_k)}\right]$$

channel noise on the system is minimized. The first mapping operation is performed in the transmitter by a *channel encoder*, whereas the inverse mapping operation is performed in the receiver by a *channel decoder*, as shown in the block diagram of Figure 9.11; to simplify the exposition, we have not included source encoding (before channel encoding) and source decoding (after channel decoding) in Figure 9.11.

The channel encoder and channel decoder in Figure 9.11 are both under the designer's control and should be designed to optimize the overall reliability of the communication system. The approach taken is to introduce *redundancy* in the channel encoder so as to reconstruct the original source sequence as accurately as possible. Thus, in a rather loose sense, we may view channel coding as the *dual* of source coding, in that the former introduces controlled redundancy to improve reliability, whereas the latter reduces redundancy to improve efficiency.

The subject of channel coding is treated in detail in Chapter 10. For the purposes of our present discussion, it suffices to confine our attention to *block codes*. In this class of codes, the message sequence is subdivided into sequential blocks each *k* bits long, and each *k*-bit block is mapped into an *n*-bit block, where $n > k$. The number of redundant bits added by the encoder to each transmitted block is $n - k$ bits. The ratio k/n is called the *code rate*. Using r to denote the code rate, we may thus write

$$r = \frac{k}{n}$$

where, of course, r is less than unity. For a prescribed k, the code rate r (and therefore the system's coding efficiency) approaches zero as the block length n approaches infinity.

The accurate reconstruction of the original source sequence at the destination requires that the *average probability of symbol error* be arbitrarily low. This raises the following important question: Does there exist a channel coding scheme such that the probability that a message bit will be in error is less than any positive number ϵ (i.e., as small as we want it), and yet the channel coding scheme is efficient in that the code rate need not be too small? The answer to this fundamental question is an emphatic "yes." Indeed, the answer to the question is provided by Shannon's second theorem in terms of the channel capacity C, as described in what follows. Up until this point, *time* has not played an important role in our discussion of channel capacity. Suppose then the discrete memoryless source in Figure 9.11 has the source alphabet $\mathcal{S}$ and entropy $H(\mathcal{S})$ bits per source symbol. We assume that the source emits symbols once every T_s seconds. Hence, the *average information rate* of the source is $H(\mathcal{S})/T_s$ bits per second. The decoder delivers decoded symbols to the destination from the source alphabet $\mathcal{S}$ and at the same source rate of one symbol every T_s seconds. The discrete memoryless channel has a channel capacity equal to C bits per use of the channel. We assume that the channel is capable of being used once every T_c seconds. Hence, the *channel capacity per unit time is* C/T_c bits per second, which represents the maximum rate of information transfer over the channel. We are now ready to state Shannon's second theorem, known as the channel coding theorem.

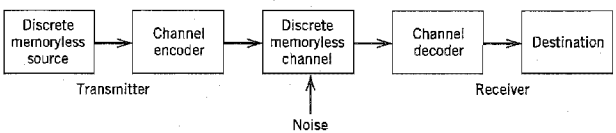

FIGURE 9.11 Block diagram of digital communication system.

the mutual information between X_k and Y_k. We may then define the information capacity of the channel as

$$C = \max_{f_{X_k}(x)} \{I(X_k; Y_k) : E[X_k^2] = P\} \tag{9.87}$$

where the maximization is performed with respect to $f_{X_k}(x)$, the probability density function of X_k.

The mutual information $I(X_k; Y_k)$ can be expressed in one of the two equivalent forms shown in Equation (9.81). For the purpose at hand, we use the second line of this equation and so write

$$I(X_k; Y_k) = h(Y_k) - h(Y_k | X_k) \tag{9.88}$$

Since X_k and N_k are independent random variables, and their sum equals Y_k, as in Equation (9.84), we find that the conditional differential entropy of Y_k, given X_k, is equal to the differential entropy of N_k (see Problem 9.28):

$$h(Y_k | X_k) = h(N_k) \tag{9.89}$$

Hence, we may rewrite Equation (9.88) as

$$I(X_k; Y_k) = h(Y_k) - h(N_k) \tag{9.90}$$

Since $h(N_k)$ is independent of the distribution of X_k, maximizing $I(X_k; Y_k)$ in accordance with Equation (9.87) requires maximizing $h(Y_k)$, the differential entropy of sample Y_k of the received signal. For $h(Y_k)$ to be maximum, Y_k has to be a Gaussian random variable (see Example 9.8). That is, the samples of the received signal represent a noiselike process. Next, we observe that since N_k is Gaussian by assumption, the sample X_k of the transmitted signal must be Gaussian too. We may therefore state that the maximization specified in Equation (9.87) is attained by choosing the samples of the transmitted signal from a noiselike process of average power P. Correspondingly, we may reformulate Equation (9.87) as

$$C = I(X_k; Y_k) : X_k \text{ Gaussian}, \qquad E[X_k^2] = P \tag{9.91}$$

where the mutual information $I(X_k; Y_k)$ is defined in accordance with Equation (9.90).

For the evaluation of the information capacity C, we proceed in three stages:

1. The variance of sample Y_k of the received signal equals $P + \sigma^2$. Hence, the use of Equation (9.76) yields the differential entropy of Y_k as

$$h(Y_k) = \tfrac{1}{2} \log_2[2\pi e(P + \sigma^2)] \tag{9.92}$$

2. The variance of the noise sample N_k equals σ^2. Hence, the use of Equation (9.76) yields the differential entropy of N_k as

$$h(N_k) = \tfrac{1}{2} \log_2(2\pi e\sigma^2) \tag{9.93}$$

3. Substituting Equations (9.92) and (9.93) into Equation (9.90) and recognizing the definition of information capacity given in Equation (9.91), we get the desired result:

$$C = \frac{1}{2} \log_2\left(1 + \frac{P}{\sigma^2}\right) \text{ bits per transmission} \tag{9.94}$$

With the channel used K times for the transmission of K samples of the process $X(t)$ in T seconds, we find that the information capacity per unit time is (K/T) times the result

given in Equation (9.94). The number K equals $2BT$, as in Equation (9.83). Accordingly, we may express the information capacity in the equivalent form:

$$C = B \log_2\left(1 + \frac{P}{N_0 B}\right) \text{ bits per second} \tag{9.95}$$

where we have used Equation (9.85) for the noise variance σ^2.

Based on the formula of Equation (9.95), we may now state Shannon's third (and most famous) theorem, the *information capacity theorem*,[10] as follows:

The information capacity of a continuous channel of bandwidth B hertz, perturbed by additive white Gaussian noise of power spectral density $N_0/2$ and limited in bandwidth to B, is given by

$$C = B \log_2\left(1 + \frac{P}{N_0 B}\right) \text{ bits per second}$$

where P is the average transmitted power.

The information capacity theorem is one of the most remarkable results of information theory for, in a single formula, it highlights most vividly the interplay among three key system parameters: channel bandwidth, average transmitted power (or, equivalently, average received signal power), and noise power spectral density at the channel output. The dependence of information capacity C on channel bandwidth B is *linear*, whereas its dependence on signal-to-noise ratio P/N_0B is *logarithmic*. Accordingly, *it is easier to increase the information capacity of a communication channel by expanding its bandwidth than increasing the transmitted power for a prescribed noise variance.*

The theorem implies that, for given average transmitted power P and channel bandwidth B, we can transmit information at the rate of C bits per second, as defined in Equation (9.95), with arbitrarily small probability of error by employing sufficiently complex encoding systems. It is not possible to transmit at a rate higher than C bits per second by any encoding system without a definite probability of error. Hence, the channel capacity theorem defines the *fundamental limit* on the rate of error-free transmission for a power-limited, band-limited Gaussian channel. To approach this limit, however, the transmitted signal must have statistical properties approximating those of white Gaussian noise.

Sphere Packing[11]

To provide a plausible argument supporting the information capacity theorem, suppose that we use an encoding scheme that yields K code words, one for each sample of the transmitted signal. Let n denote the length (i.e., the number of bits) of each code word. It is presumed that the coding scheme is designed to produce an acceptably low probability of symbol error. Furthermore, the code words satisfy the power constraint; that is, the average power contained in the transmission of each code word with n bits is nP, where P is the average power per bit.

Suppose that any code word in the code is transmitted. The received vector of n bits is Gaussian distributed with mean equal to the transmitted code word and variance equal to $n\sigma^2$, where σ^2 is the noise variance. With high probability, the received vector lies inside a sphere of radius $\sqrt{n\sigma^2}$, centered on the transmitted code word. This sphere is itself contained in a larger sphere of radius $\sqrt{n(P + \sigma^2)}$, where $n(P + \sigma^2)$ is the average power of the received vector.

where the factor 1/2 accounts for the fact that Δf applies to both positive and negative frequencies. All the N subchannels are independent of one another. Hence the total capacity of the overall channel is approximately given by the summation

$$\begin{aligned} C &\simeq \sum_{k=1}^{N} C_k \\ &= \frac{1}{2} \sum_{k=1}^{N} \Delta f \log_2\left(1 + \frac{P_k}{\sigma_k^2}\right) \end{aligned} \tag{9.118}$$

The problem we have to address is to maximize the overall information capacity C subject to the constraint:

$$\sum_{k=1}^{N} P_k = P \quad \text{(constant)} \tag{9.119}$$

The usual procedure to solve a constrained optimization problem is to use the *method of Lagrange multipliers* (see Note 19 in Chapter 6. To proceed with this optimization, we first define an objective function that incorporates both the information capacity C and the constraint [i.e., Equations (9.118) and (9.119)], as shown by

$$J = \frac{1}{2} \sum_{k=1}^{N} \Delta f \log_2\left(1 + \frac{P_k}{\sigma_k^2}\right) + \lambda\left(P - \sum_{k=1}^{N} P_k\right) \tag{9.120}$$

where λ is the Lagrange multiplier. Next, differentiating the objective function J with respect to P_k and setting the result equal to zero, we obtain

$$\frac{\Delta f \log_2 e}{P_k + \sigma_k^2} - \lambda = 0$$

To satisfy this optimizing solution, we impose the following requirement:

$$P_k + \sigma_k^2 = K \Delta f \qquad \text{for } k = 1, 2, \ldots, N \tag{9.121}$$

where K is a constant that is the same for all k. The constant K is chosen to satisfy the average power constraint.

Inserting the defining values of Equations (9.115) and (9.116) in the optimizing condition of Equation (9.121), simplifying, and rearranging terms, we get

$$S_X(f_k) = K - \frac{S_N(f_k)}{|H(f_k)|^2}, \qquad k = 1, 2, \ldots, N \tag{9.122}$$

Let $\mathcal{F}_A$ denote the frequency range for which the constant K satisfies the condition

$$K \geq \frac{S_N(f)}{|H(f)|^2}$$

Then, as the incremental frequency interval Δf is allowed to approach zero and the number of subchannels N goes to infinity, we may use Equation (9.122) to formally state that the power spectral density of the input ensemble that achieves the optimum information capacity is a nonnegative quantity defined by

$$S_X(f) = \begin{cases} K - \dfrac{S_N(f)}{|H(f)|^2} & \text{for } f \in \mathcal{F}_A \\ 0 & \text{otherwise} \end{cases} \tag{9.123}$$

Since the average power of a random process is the total area under the curve of the power spectral density of the process, we may express the average power of the channel input $x(t)$ as

$$P = \int_{f \in \mathcal{F}_A} \left(K - \frac{S_N(f)}{|H(f)|^2} \right) df \tag{9.124}$$

For a prescribed P and specified $S_N(f)$ and $H(f)$, the constant K is the solution to Equation (9.124).

The only thing that remains for us to do is to find the optimum information capacity. Substituting the optimizing solution of Equation (9.121) into Equation (9.118) and then using the defining values of Equations (9.115) and (9.116), we obtain

$$C \simeq \frac{1}{2} \sum_{k=1}^{N} \Delta f \log_2 \left(K \frac{|H(f_k)|^2}{S_N(f_k)} \right)$$

When the incremental frequency interval Δf is allowed to approach zero, this equation takes the limiting form:

$$C = \frac{1}{2} \int_{-\infty}^{\infty} \log_2 \left(K \frac{|H(f)|^2}{S_N(f)} \right) df \tag{9.125}$$

where the constant K is chosen as the solution to Equation (9.124) for a prescribed input signal power P.

▪ WATER-FILLING INTERPRETATION OF THE INFORMATION CAPACITY THEOREM

Equations (9.123) and (9.124) suggest the picture portrayed in Figure 9.21. Specifically, we make the following observations:

- The appropriate input power spectral density $S_X(f)$ is described as the bottom regions of the function $S_N(f)/|H(f)|^2$ that lie below the constant level K, which are shown shaded.
- The input power P is defined by the total area of these shaded regions.

The spectral domain picture portrayed here is called the *water-filling (pouring) interpretation* in the sense that the process by which the input power is distributed across

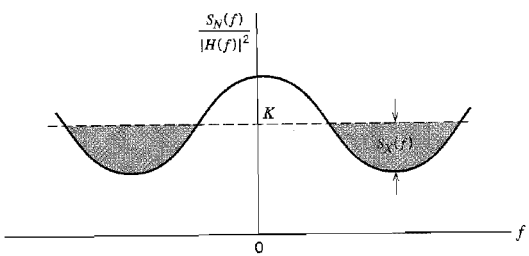

FIGURE 9.21 Water-filling interpretation of information-capacity theorem for a colored noisy channel.

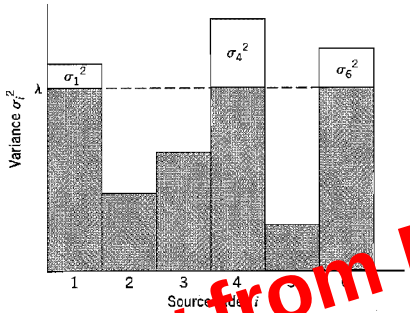

FIGURE 9.23 Reverse water-filling picture for a set of parallel Gaussian processes.

▶ EXAMPLE 9.15 Set of Parallel Gaussian Sources

Consider next a set of N independent Gaussian random variables $\{X_i\}_{i=1}^N$, where X_i has zero mean and variance σ_i^2. Using the distortion measure

$$d = \sum_{i=1}^{N} (x_i - \hat{x}_i)^2$$

and building on the result of Example 9.14, we may express the rate distortion function for the set of parallel Gaussian sources described here as

$$R(D) = \sum_{i=1}^{N} \frac{1}{2} \log\left(\frac{\sigma_i^2}{D_i}\right) \tag{9.134}$$

where D_i is itself defined by

$$D_i = \begin{cases} \lambda & \text{if } \lambda < \sigma_i \\ \sigma_i^2 & \text{if } \lambda \geq \sigma_i^2 \end{cases} \tag{9.135}$$

and the constant λ is chosen so as to satisfy the condition

$$\sum_{i=1}^{N} D_i = D \tag{9.136}$$

Equations (9.135) and (9.136) may be interpreted as a kind of "water-filling in reverse," as illustrated in Figure 9.23. First, we choose a constant λ and only the subset of random variables whose variances exceed the constant λ. No bits are used to describe the remaining subset of random variables whose variances are less than the constant λ. ◀

9.14 *Data Compression*

Rate distortion theory naturally leads us to consider the idea of *data compression* that involves a purposeful or unavoidable reduction in the information content of data from a continuous or discrete source. Specifically, we may think of a *data compressor*, or *signal compressor*, as a device that supplies a code with the least number of symbols for the representation of the source output, subject to a permissible or acceptable *distortion*. The data compressor thus retains the essential information content of the source output by blurring fine details in a deliberate but controlled manner. Accordingly, data compression

9.20 Figure 9.10 depicts the variation of the channel capacity of a binary symmetric channel with the transition probability p. Use the results of Problem 9.19 to explain this variation.

9.21 Consider the binary symmetric channel described in Figure 9.8. Let p_0 denote the probability of sending binary symbol $x_0 = 0$, and let $p_1 = 1 - p_0$ denote the probability of sending binary symbol $x_1 = 1$. Let p denote the transition probability of the channel.

(a) Show that the mutual information between the channel input and channel output is given by

$$I(\mathcal{X}; \mathcal{Y}) = \mathcal{H}(z) - \mathcal{H}(p)$$

where

$$H(z) = z \log_2\left(\frac{1}{z}\right) + (1 - z) \log_2\left(\frac{1}{1 - z}\right)$$

$$z = p_0 p + (1 - p_0)(1 - p)$$

and

$$H(p) = p \log_2\left(\frac{1}{p}\right) + (1 - p) \log_2\left(\frac{1}{1 - p}\right)$$

(b) Show that the value of p_0 that maximizes $I(\mathcal{X}; \mathcal{Y})$ is equal to 1/2.

(c) Hence, show that the channel capacity equals

$$C = 1 - H(p)$$

9.22 Two binary symmetric channels are connected in cascade, as shown in Figure P9.22. Find the overall channel capacity of the cascaded connection, assuming that both channels have the same transition probability diagram shown in Figure 9.8.

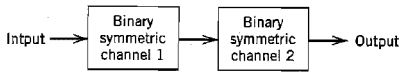

FIGURE P9.22

9.23 The *binary erasure channel* has two inputs and three outputs as described in Figure P9.23. The inputs are labeled 0 and 1, and the outputs are labeled 0, 1, and e. A fraction α of the incoming bits are erased by the channel. Find the capacity of the channel.

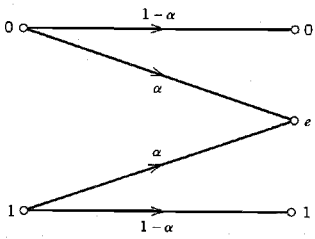

FIGURE P9.23

CHAPTER 10

ERROR-CONTROL CODING

This chapter is the natural sequel to the preceding chapter on Shannon's information theory. In particular, in this chapter we present error-control coding techniques that provide different ways of implementing Shannon's channel coding theorem. Each error-control coding technique involves the use of a channel encoder in the transmitter and a decoding algorithm in the receiver.

The error-control coding techniques described herein include the following important classes of codes:

- *Linear block codes.*
- *Cyclic codes.*
- *Convolutional codes.*
- *Compound codes exemplified by turbo codes and low-density parity-check codes, and their irregular variants.*

10.1 *Introduction*

The task facing the designer of a digital communication system is that of providing a cost-effective facility for transmitting information from one end of the system at a rate and a level of reliability and quality that are acceptable to a user at the other end. The two key system parameters available to the designer are transmitted signal power and channel bandwidth. These two parameters, together with the power spectral density of receiver noise, determine the signal energy per bit-to-noise power spectral density ratio E_b/N_0. In Chapter 6, we showed that this ratio uniquely determines the bit error rate for a particular modulation scheme. Practical considerations usually place a limit on the value that we can assign to E_b/N_0. Accordingly, in practice, we often arrive at a modulation scheme and find that it is not possible to provide acceptable data quality (i.e., low enough error performance). For a fixed E_b/N_0, the only practical option available for changing data quality from problematic to acceptable is to use *error-control coding*.

Another practical motivation for the use of coding is to reduce the required E_b/N_0 for a fixed bit error rate. This reduction in E_b/N_0 may, in turn, be exploited to reduce the required transmitted power or reduce the hardware costs by requiring a smaller antenna size in the case of radio communications.

Error control[1] for data integrity may be exercised by means of *forward error correction* (FEC). Figure 10.1*a* shows the model of a digital communication system using such an approach. The discrete source generates information in the form of binary symbols. The *channel encoder* in the transmitter accepts message bits and adds *redundancy* according to a prescribed rule, thereby producing encoded data at a higher bit rate. The *channel*

a "sliding window" equal in duration to its own memory. This, in turn, means that in a convolutional code, unlike a block code, the channel encoder accepts message bits as a continuous sequence and thereby generates a continuous sequence of encoded bits at a higher rate.

In the model depicted in Figure 10.1*a*, the operations of channel coding and modulation are performed separately in the transmitter; likewise for the operations of detection and decoding in the receiver. When, however, bandwidth efficiency is of major concern, the most effective method of implementing forward error-control correction coding is to combine it with modulation as a single function, as shown in Figure 10.1*b*. In such an approach, coding is redefined as a process of imposing certain patterns on the transmitted signal.

■ AUTOMATIC-REPEAT REQUEST

Feed-forward error correction (FEC) relies on the controlled insertion of redundancy in the transmitted code word for both the *detection and correction* of errors incurred during the course of transmission over a noisy channel. Irrespective of whether the decoding of the received code word is successful, no further processing is performed at the receiver. Accordingly, channel coding techniques suitable for FEC require only a *one-way link* between the transmitter and receiver.

There is another approach known as *automatic-repeat request (ARQ)*[2] for solving the error-control problem. The underlying philosophy of ARQ is quite different from that of FEC. Specifically, ARQ uses redundancy merely for the purpose of *error detection*. Upon the detection of an error in a transmitted code word, the receiver requests a repeat transmission of the corrupted code word, which necessitates the use of a *return path* (i.e., a feedback channel). As such, ARQ can be used only on *half-duplex* or *full-duplex links*. In a half-duplex link, data transmission over the link can be made in either direction but *not* simultaneously. On the other hand, in a full-duplex link, it is possible for data transmission to proceed over the link in both directions simultaneously.

A half-duplex link uses the simplest ARQ scheme known as the *stop-and-wait strategy*. In this approach, a block of message bits is encoded into a code word and transmitted over the channel. The transmitter then stops and waits for feedback from the receiver. The feedback signal can be acknowledgment of a correct receipt of the code word or a request for transmission of the code word because of an error in its decoding. In the latter case, the transmitter resends the code word in question before moving onto the next block of message bits.

The idling problem in stop-and-wait ARQ results in reduced data throughput, which is alleviated in another type of ARQ known as *continuous ARQ with pullback*. This second strategy uses a full-duplex link, thereby permitting the receiver to send a feedback signal while the transmitter is engaged in sending code words over the forward channel. Specifically, the transmitter continues to send a succession of code words until it receives a request from the receiver (on the feedback channel) for a retransmission. At that point, the transmitter stops, pulls back to the particular code word that was not decoded correctly by the receiver, and retransmits the complete sequence of code words starting with the corrupted one.

In a refined version of continuous ARQ known as the *continuous ARQ with selective repeat*, data throughout is improved further by only retransmitting the code word that was received with detected errors. In other words, the need for retransmitting the successfully received code words following the corrupted code word is eliminated.

EXAMPLE 10.1 Repetition Codes

Repetition codes represent the simplest type of linear block codes. In particular, a single message bit is encoded into a block of n identical bits, producing an $(n, 1)$ block code. Such a code allows provision for a variable amount of redundancy. There are only two code words in the code: an all-zero code word and an all-one code word.

Consider, for example, the case of a repetition code with $k = 1$ and $n = 5$. In this case, we have four parity bits that are the same as the message bit. Hence, the identity matrix $\mathbf{I}_k = 1$, and the coefficient matrix P consists of a 1-by-4 vector that has 1 for all of its elements. Correspondingly, the generator matrix equals a row vector of all 1s, as shown by

$$\mathbf{G} = [1 \quad 1 \quad 1 \quad 1 \vdots 1]$$

The transpose of the coefficient matrix P, namely $\mathbf{P}^T$, consists of a 4-by-1 vector that has 1 for all of its elements. The identity matrix $\mathbf{I}_{n-k}$ consists of a 4-by-4 matrix. Hence, the parity-check matrix equals

$$\mathbf{H} = \begin{bmatrix} 1 & 0 & 0 & 0 \vdots 1 \\ 0 & 1 & 0 & 0 \vdots 1 \\ 0 & 0 & 1 & 0 \vdots 1 \\ 0 & 0 & 0 & 1 \vdots 1 \end{bmatrix}$$

Since the message vector consists of a single binary symbol, 0 or 1, it follows from Equation (10.13) that there are only two code words: 00000 and 11111 in the (5, 1) repetition code, as expected. Note also that $\mathbf{HG}^T = 0$, modulo-2, in accordance with Equation (10.15). ◀

SYNDROME: DEFINITION AND PROPERTIES

The generator matrix G is used in the encoding operation at the transmitter. On the other hand, the parity-check matrix H is used in the decoding operation at the receiver. In the context of the latter operation, let r denote the 1-by-n *received vector* that results from sending the code vector c over a noisy channel. We express the vector r as the sum of the original code vector c and a vector e, as shown by

$$\mathbf{r} = \mathbf{c} + \mathbf{e} \tag{10.17}$$

The vector e is called the *error vector* or *error pattern*. The ith element of e equals 0 if the corresponding element of r is the same as that of c. On the other hand, the ith element of e equals 1 if the corresponding element of r is different from that of c, in which case an error is said to have occurred in the ith location. That is, for $i = 1, 2, \ldots, n$, we have

$$e_i = \begin{cases} 1 & \text{if an error has occurred in the } i\text{th location} \\ 0 & \text{otherwise} \end{cases} \tag{10.18}$$

The receiver has the task of decoding the code vector c from the received vector r. The algorithm commonly used to perform this decoding operation starts with the computation of a 1-by-$(n - k)$ vector called the *error-syndrome vector* or simply the *syndrome*.[3] The importance of the syndrome lies in the fact that it depends only upon the error pattern.

Given a 1-by-n received vector r, the corresponding syndrome is formally defined as

$$\mathbf{s} = \mathbf{rH}^T \tag{10.19}$$

Accordingly, the syndrome has the following important properties.

Property 1

The syndrome depends only on the error pattern, and not on the transmitted code word.

To prove this property, we first use Equations (10.17) and (10.19), and then Equation (10.16) to obtain

$$\begin{aligned} \mathbf{s} &= (\mathbf{c} + \mathbf{e})\mathbf{H}^T \\ &= \mathbf{c}\mathbf{H}^T + \mathbf{e}\mathbf{H}^T \\ &= \mathbf{e}\mathbf{H}^T \end{aligned} \tag{10.20}$$

Hence, the parity-check matrix H of a code permits us to compute the syndrome s, which depends only upon the error pattern e.

Property 2

All error patterns that differ by a code word have the same syndrome.

For k message bits, there are 2^k distinct code vectors denoted as $\mathbf{c}_i$, $i = 0, 1, \ldots, 2^k - 1$. Correspondingly, for any error pattern e, we define the 2^k distinct vectors $\mathbf{e}_i$ as

$$\mathbf{e}_i = \mathbf{e} + \mathbf{c}_i, \qquad i = 0, 1, \ldots, 2^k - 1 \tag{10.21}$$

The set of vectors $\{\mathbf{e}_i, i = 0, 1, \ldots, 2^k - 1\}$ so defined is called a *coset* of the code. In other words, a coset has exactly 2^k elements that differ at most by a code vector. Thus, an (n, k) linear block code has 2^{n-k} possible cosets. In any event, multiplying both sides of Equation (10.21) by the matrix $\mathbf{H}^T$, we get

$$\begin{aligned} \mathbf{e}_i\mathbf{H}^T &= \mathbf{e}\mathbf{H}^T + \mathbf{c}_i\mathbf{H}^T \\ &= \mathbf{e}\mathbf{H}^T \end{aligned} \tag{10.22}$$

which is independent of the index i. Accordingly, we may state that each coset of the code is characterized by a unique syndrome.

We may put Properties 1 and 2 in perspective by expanding Equation (10.20). Specifically, with the matrix H having the systematic form given in Equation (10.14), where the matrix P is itself defined by Equation (10.8), we find from Equation (10.20) that the $(n - k)$ elements of the syndrome s are linear combinations of the n elements of the error pattern e, as shown by

$$\begin{aligned} s_0 &= e_0 + e_{n-k}p_{00} + e_{n-k+1}p_{10} + \cdots + e_{n-1}p_{k-1,0} \\ s_1 &= e_1 + e_{n-k}p_{01} + e_{n-k+1}p_{11} + \cdots + e_{n-1}p_{k-1,1} \\ &\vdots \\ s_{n-k-1} &= e_{n-k-1} + e_{n-k}p_{0,n-k-1} + \cdots + e_{n-1}p_{k-1,n-k-1} \end{aligned} \tag{10.23}$$

This set of $(n - k)$ linear equations clearly shows that the syndrome contains information about the error pattern and may therefore be used for error detection. However, it should be noted that the set of equations is *underdetermined* in that we have more unknowns than equations. Accordingly, there is *no* unique solution for the error pattern. Rather, there are 2^n error patterns that satisfy Equation (10.23) and therefore result in the same syndrome, in accordance with Property 2 and Equation (10.22). In particular, with 2^{n-k} possible syndrome vectors, the information contained in the syndrome s about the error pattern e is *not* enough for the decoder to compute the exact value of the transmitted code vector. Nevertheless, knowledge of the syndrome s reduces the search for the true error

$\mathbf{c}_1 = \mathbf{0}$	$\mathbf{c}_2$	$\mathbf{c}_3$	$\cdots$	$\mathbf{c}_i$	$\cdots$	$\mathbf{c}_{2^k}$
$\mathbf{e}_2$	$\mathbf{c}_2 + \mathbf{e}_2$	$\mathbf{c}_3 + \mathbf{e}_2$	$\cdots$	$\mathbf{c}_i + \mathbf{e}_2$	$\cdots$	$\mathbf{c}_{2^k} + \mathbf{e}_2$
$\mathbf{e}_3$	$\mathbf{c}_2 + \mathbf{e}_3$	$\mathbf{c}_3 + \mathbf{e}_3$	$\cdots$	$\mathbf{c}_i + \mathbf{e}_3$	$\cdots$	$\mathbf{c}_{2^k} + \mathbf{e}_3$
$\vdots$	$\vdots$	$\vdots$		$\vdots$		$\vdots$
$\mathbf{e}_j$	$\mathbf{c}_2 + \mathbf{e}_j$	$\mathbf{c}_3 + \mathbf{e}_j$	$\cdots$	$\mathbf{c}_i + \mathbf{e}_j$	$\cdots$	$\mathbf{c}_{2^k} + \mathbf{e}_j$
$\vdots$	$\vdots$	$\vdots$		$\vdots$		$\vdots$
$\mathbf{e}_{2^{n-k}}$	$\mathbf{c}_2 + \mathbf{e}_{2^{n-k}}$	$\mathbf{c}_3 + \mathbf{e}_{2^{n-k}}$		$\mathbf{c}_i + \mathbf{e}_{2^{n-k}}$		$\mathbf{c}_{2^k} + \mathbf{e}_{2^{n-k}}$

FIGURE 10.7 Standard array for an (n, k) block code.

represent the cosets of the code, and their first element $\mathbf{e}_2, \ldots, \mathbf{e}_{2^{n-k}}$ are called *coset leaders*.

For a given channel, the probability of decoding error is minimized when the most likely error patterns (i.e., those that have the largest probability of occurrence) are chosen as the coset leaders. In the case of a binary symmetric channel, the smaller the Hamming weight of an error pattern the more likely it is to occur. Accordingly, the standard array should be constructed with each coset leader having the minimum Hamming weight in its coset.

We may now describe a decoding procedure for a linear block code:

1. For the received vector $\mathbf{r}$, compute the syndrome $\mathbf{s} = \mathbf{r}\mathbf{H}^T$.
2. Within the coset characterized by the syndrome $\mathbf{s}$, identify the coset leader (i.e., the error pattern with the largest probability of occurrence); call it $\mathbf{e}_0$.
3. Compute the code vector

$$\mathbf{c} = \mathbf{r} + \mathbf{e}_0 \tag{10.26}$$

as the decoded version of the received vector $\mathbf{r}$.

This procedure is called *syndrome decoding*.

▶ EXAMPLE 10.2 Hamming Codes[4]

Consider a family of (n, k) linear block codes that have the following parameters:

Block length:	$n = 2^m - 1$
Number of message bits:	$k = 2^m - m - 1$
Number of parity bits:	$n - k = m$

where $m \geq 3$. These are the so-called Hamming codes.

Consider, for example, the (7, 4) Hamming code with $n = 7$ and $k = 4$, corresponding to $m = 3$. The generator matrix of the code must have a structure that conforms to Equation (10.12). The following matrix represents an appropriate generator matrix for the (7, 4) Hamming code:

$$\mathbf{G} = \left[\begin{array}{ccc:cccc} 1 & 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 & 1 \end{array}\right] = [\underbrace{\mathbf{P}}_{} : \underbrace{\mathbf{I}_k}_{}]$$

advantage of cyclic codes over most other types of codes is that they are easy to encode. Furthermore, cyclic codes possess a well-defined mathematical structure, which has led to the development of very efficient decoding schemes for them.

A binary code is said to be a *cyclic code* if it exhibits two fundamental properties:

1. *Linearity property:* The sum of any two code words in the code is also a code word.
2. *Cyclic property:* Any cyclic shift of a code word in the code is also a code word.

Property 1 restates the fact that a cyclic code is a linear block code (i.e., it can be described as a parity-check code). To restate Property 2 in mathematical terms, let the n-tuple $(c_0, c_1, \ldots, c_{n-1})$ denote a code word of an (n, k) linear block code. This code is a cyclic code if the n-tuples

$$(c_{n-1}, c_0, \ldots, c_{n-2}),$$
$$(c_{n-2}, c_{n-1}, \ldots, c_{n-3}),$$
$$\vdots$$
$$(c_1, c_2, \ldots, c_{n-1}, c_0)$$

are all code words in the code.

To develop the algebraic properties of cyclic codes, we use the elements $c_0, c_1, \ldots, c_{n-1}$ of a code word to define the *code polynomial*

$$c(X) = c_0 + c_1X + c_2X^2 + \cdots + c_{n-1}X^{n-1} \tag{10.27}$$

where X is an indeterminate. Naturally, for binary codes, the coefficients are 1s and 0s. Each power of X in the polynomial $c(X)$ represents a one-bit *shift* in time. Hence, multiplication of the polynomial $c(X)$ by X may be viewed as a shift to the right. The key question is: How do we make such a shift *cyclic*? The answer to this question is addressed next.

Let the code polynomial $c(X)$ be multiplied by X^i, yielding

$$\begin{aligned} X^ic(X) &= X^i(c_0 + c_1X + \cdots + c_{n-i-1}X^{n-i-1} + c_{n-i}X^{n-i} \\ &\quad + \cdots + c_{n-1}X^{n-1}) \\ &= c_0X^i + c_1X^{i+1} + \cdots + c_{n-i-1}X^{n-1} + c_{n-i}X^n \\ &\quad + \cdots + c_{n-1}X^{n+i-1} \\ &= c_{n-i}X^n + \cdots + c_{n-1}X^{n+i-1} + c_0X^i + c_1X^{i+1} \\ &\quad + \cdots + c_{n-i-1}X^{n-1} \end{aligned} \tag{10.28}$$

where, in the last line, we have merely rearranged terms. Recognizing, for example, that $c_{n-i} + c_{n-i} = 0$ in modulo-2 addition, we may manipulate the first i terms of Equation (10.28) as follows:

$$\begin{aligned} X^ic(X) &= c_{n-i} + \cdots + c_{n-1}X^{i-1} + c_0X^i + c_1X^{i+1} + \cdots + c_{n-i-1}X^{n-1} \\ &\quad + c_{n-i}(X^n + 1) + \cdots + c_{n-1}X^{i-1}(X^n + 1) \end{aligned} \tag{10.29}$$

Next, we introduce the following definitions:

$$\begin{aligned} c^{(i)}(X) &= c_{n-i} + \cdots + c_{n-1}X^{i-1} + c_0X^i + c_1X^{i+1} \\ &\quad + \cdots + c_{n-i-1}X^{n-1} \end{aligned} \tag{10.30}$$

$$q(X) = c_{n-i} + c_{n-i+1}X + \cdots + c_{n-1}X^{i-1} \tag{10.31}$$

good cyclic codes, whereas some of them generate bad cyclic codes. The issue of how to select generator polynomials that produce good cyclic codes is very difficult to resolve. Indeed, coding theorists have expended much effort in the search for good cyclic codes.

Generator and Parity-Check Matrices

Given the generator polynomial $g(X)$ of an (n, k) cyclic code, we may construct the generator matrix $\mathbf{G}$ of the code by noting that the k polynomials $g(X), Xg(X), \ldots, X^{k-1}g(X)$ span the code. Hence, the n-tuples corresponding to these polynomials may be used as rows of the k-by-n generator matrix $\mathbf{G}$.

However, the construction of the parity-check matrix $\mathbf{H}$ of the cyclic code from the parity-check polynomial $h(X)$ requires special consideration, as described here. Multiplying Equation (10.42) by $a(x)$ and then using Equation (10.35), we obtain

$$c(X)h(X) = a(X) + X^n a(X) \tag{10.43}$$

The polynomials $c(X)$ and $h(X)$ are themselves defined by Equations (10.27) and (10.40), respectively, which means that their product on the left-hand side of Equation (10.43) contains terms with powers extending up to $n + k - 1$. On the other hand, the polynomial $a(X)$ has degree $k - 1$ or less, the implication of which is that the powers of $X^k, X^{k+1}, \ldots, X^{n-1}$ do *not* appear in the polynomial on the right-hand side of Equation (10.43). Thus, setting the coefficients of $X^k, X^{k-1}, \ldots, X^{n-1}$ in the expansion of the product polynomial $c(X)h(X)$ equal to zero, we obtain the following set of $n - k$ equations:

$$\sum_{i=j}^{j+k} c_i h_{k+j-i} = 0 \qquad \text{for } 0 \le j \le n - k - 1 \tag{10.44}$$

Comparing Equation (10.44) with the corresponding relation of Equation (10.16), we may make the following important observation: The coefficients of the parity-check polynomial $h(X)$ involved in the polynomial multiplication described in Equation (10.44) are arranged in *reversed* order with respect to the coefficients of the parity-check matrix $\mathbf{H}$ involved in forming the inner product of vectors described in Equation (10.16). This observation suggests that we define the *reciprocal of the parity-check polynomial* as follows:

$$\begin{aligned} X^k h(X^{-1}) &= X^k\left(1 + \sum_{i=1}^{k-1} h_i X^{-i} + X^{-k}\right) \\ &= 1 + \sum_{i=1}^{k-1} h_{k-i} X^i + X^k \end{aligned} \tag{10.45}$$

which is also a factor of $X^n + 1$. The n-tuples pertaining to the $(n - k)$ polynomials $X^k h(X^{-1}), X^{k+1}h(X^{-1}), \ldots, X^{n-1}h(X^{-1})$ may now be used in rows of the $(n - k)$-by-n parity-check matrix $\mathbf{H}$.

In general, the generator matrix $\mathbf{G}$ and the parity-check matrix $\mathbf{H}$ constructed in the manner described here are not in their systematic forms. They can be put into their systematic forms by performing simple operations on their respective rows, as illustrated in Example 10.3.

Encoder for Cyclic Codes

Earlier we showed that the encoding procedure for an (n, k) cyclic code in systematic form involves three steps: (1) multiplication of the message polynomial $m(X)$ by X^{n-k}, (2) di-

Let $q(X)$ denote the quotient and $s(X)$ denote the remainder, which are the results of dividing $r(X)$ by the generator polynomial $g(X)$. We may therefore express $r(X)$ as follows:

$$r(X) = q(X)g(X) + s(X) \tag{10.47}$$

The remainder $s(X)$ is a polynomial of degree $n - k - 1$ or less, which is the result of interest. It is called the *syndrome polynomial* because its coefficients make up the $(n - k)$-by-1 syndrome s.

Figure 10.9 shows a *syndrome calculator* that is identical to the encoder of Figure 10.8 except for the fact that the received bits are fed into the $(n - k)$ stages of the feedback shift register from the left. As soon as all the received bits have been shifted into the shift register, its contents define the syndrome s.

The syndrome polynomial $s(X)$ has the following useful properties that follow from the definition given in Equation (10.47):

1. *The syndrome of a received word polynomial is also the syndrome of the corresponding error polynomial.*

Given that a cyclic code with polynomial $c(X)$ is sent over a noisy channel, the received word polynomial is defined by

$$r(X) = c(X) + e(X) \tag{10.48}$$

where $e(X)$ is the *error polynomial*. Equivalently, we may write

$$e(X) = r(X) + c(X) \tag{10.49}$$

Hence, substituting Equations (10.35) and (10.47) into (10.49), we get

$$e(X) = u(X)g(X) + s(X) \tag{10.50}$$

where the quotient is $u(X) = a(X) + q(X)$. Equation (10.50) shows that $s(X)$ is also the syndrome of the error polynomial $e(X)$. The implication of this property is that when the syndrome polynomial $s(X)$ is nonzero, the presence of transmission errors in the received word is detected.

2. *Let $s(X)$ be the syndrome of a received word polynomial $r(X)$. Then, the syndrome of $Xr(X)$, a cyclic shift of $r(X)$, is $Xs(X)$.*

Applying a cyclic shift to both sides of Equation (10.47), we get

$$Xr(X) = Xq(X)g(X) + Xs(X) \tag{10.51}$$

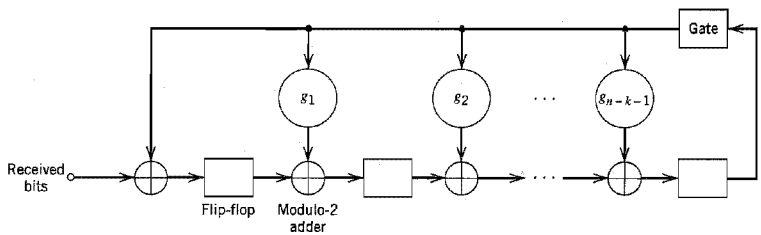

FIGURE 10.9 Syndrome calculator for (n, k) cyclic code.

TABLE 10.5 *CRC codes*

Code	*Generator Polynomial, g(X)*	n − k
CRC-12 code	$1 + X + X^2 + X^3 + X^{11} + X^{12}$	12
CRC-16 code (USA)	$1 + X^2 + X^{15} + X^{16}$	16
CRC-ITU code	$1 + X^5 + X^{12} + X^{16}$	16

Table 10.5 presents the generator polynomials of three CRC codes that have become international standards. All three codes contain $1 + X$ as a prime factor. The CRC-12 code is used for 6-bit characters, and the other two codes are used for 8-bit characters. CRC codes provide a powerful method of error detection for use in automatic-repeat request (ARQ) strategies discussed in Section 10.1, and digital subscriber lines discussed in Chapter 4.

Bose–Chaudhuri–Hocquenghem (BCH) Codes

One of the most important and powerful classes of linear-block codes are *BCH codes*, which are cyclic codes with a wide variety of parameters. The most common binary BCH codes, known as *primitive BCH codes*, are characterized for any positive integers m (equal to or greater than 3) and t [less than $(2^m - 1)/2$] by the following parameters:

$$\begin{aligned} &\text{Block length:} && n = 2^m - 1 \\ &\text{Number of message bits:} && k \geq n - mt \\ &\text{Minimum distance:} && d_{\min} \geq 2t + 1 \end{aligned}$$

Each BCH code is a *t-error correcting code* in that it can detect and correct up to t random errors per code word. The Hamming single-error correcting codes can be described as BCH codes. The BCH codes offer flexibility in the choice of code parameters, namely, block length and code rate. Furthermore, for block lengths of a few hundred bits or less, the BCH codes are among the best known codes of the same block length and code rate.

A detailed treatment of the construction of BCH codes is beyond the scope of our present discussion. To provide a feel for their capability, we present in Table 10.6, the code parameters and generator polynomials for binary block BCH codes of length up to $2^5 - 1$. For example, suppose we wish to construct the generator polynomial for (15, 7)

TABLE 10.6 *Binary BCH codes of length up to $2^5 - 1$*

n	k	t	*Generator Polynomial*									
7	4	1									1	011
15	11	1									10	011
15	7	2								111	010	001
15	5	3							10	100	110	111
31	26	1									100	101
31	21	2							11	101	101	001
31	16	3						1	000	111	110	101	111
31	11	5				101	100	010	011	011	010	101
31	6	7	11	001	011	011	110	101	000	100	111	

Notation: n = block length
k = number of message bits
t = maximum number of detectable errors

The high-order coefficients of the generator polynomial $g(X)$ are at the left.

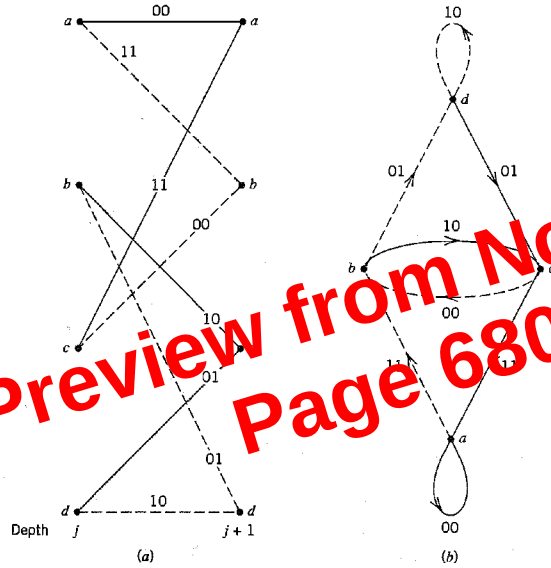

FIGURE 10.16 (*a*) A portion of the central part of the trellis for the encoder of Figure 10.13*a* (*b*) State diagram of the convolutional encoder of Figure 10.13*a*.

agrees exactly with our previous result. Thus, the input–output relation of a convolutional encoder is also completely described by its state diagram.

10.6 *Maximum Likelihood Decoding of Convolutional Codes*

Now that we understand the operation of a convolutional encoder, the next issue to be considered is the decoding of a convolutional code. In this section we first describe the underlying theory of maximum likelihood decoding, and then present an efficient algorithm for its practical implementation.

Let $\mathbf{m}$ denote a *message vector*, and $\mathbf{c}$ denote the corresponding *code vector* applied by the encoder to the input of a discrete memoryless channel. Let $\mathbf{r}$ denote the *received vector*, which may differ from the transmitted code vector due to channel noise. Given the received vector $\mathbf{r}$, the decoder is required to make an *estimate* $\hat{\mathbf{m}}$ of the message vector. Since there is a one-to-one correspondence between the message vector $\mathbf{m}$ and the code vector $\mathbf{c}$, the decoder may equivalently produce an estimate $\hat{\mathbf{c}}$ of the code vector. We may then put $\hat{\mathbf{m}} = \mathbf{m}$ if and only if $\hat{\mathbf{c}} = \mathbf{c}$. Otherwise, a *decoding error* is committed in the receiver. The *decoding rule* for choosing the estimate $\hat{\mathbf{c}}$, given the received vector $\mathbf{r}$, is said to be optimum when the *probability of decoding error* is minimized. From the material presented in Chapter 6, we may state that for equiprobable messages, the probability of decoding error is minimized if the estimate $\hat{\mathbf{c}}$ is chosen to maximize the *log-likelihood function*. Let $p(\mathbf{r}|\mathbf{c})$ denote the conditional probability of receiving $\mathbf{r}$, given that $\mathbf{c}$ was sent.

The log-likelihood function equals $\log p(\mathbf{r}|\mathbf{c})$. The *maximum likelihood decoder* or decision rule is described as follows:

$$\text{Choose the estimate } \hat{\mathbf{c}} \text{ for which the log-likelihood function } \log p(\mathbf{r}|\mathbf{c}) \text{ is maximum.} \tag{10.56}$$

Consider now the special case of a binary symmetric channel. In this case, both the transmitted code vector $\mathbf{c}$ and the received vector $\mathbf{r}$ represent binary sequences of length N, say. Naturally, these two sequences may differ from each other in some locations because of errors due to channel noise. Let c_i and r_i denote the ith elements of $\mathbf{c}$ and $\mathbf{r}$, respectively. We then have

$$p(\mathbf{r}|\mathbf{c}) = \prod_{i=1}^{N} p(r_i|c_i) \tag{10.57}$$

Correspondingly, the log-likelihood is

$$\log p(\mathbf{r}|\mathbf{c}) = \sum_{i=1}^{N} \log p(r_i|c_i) \tag{10.58}$$

Let the transition probability $p(r_i|c_i)$ be defined as

$$p(r_i|c_i) = \begin{cases} p, & \text{if } r_i \neq c_i \\ 1-p, & \text{if } r_i = c_i \end{cases} \tag{10.59}$$

Suppose also that the received vector $\mathbf{r}$ differs from the transmitted code vector $\mathbf{c}$ in exactly d positions. The number d is the *Hamming distance* between vectors $\mathbf{r}$ and $\mathbf{c}$. Then, we may rewrite the log-likelihood function in Equation (10.58) as

$$\begin{aligned} \log p(\mathbf{r}|\mathbf{c}) &= d \log p + (N-d)\log(1-p) \\ &= d \log\left(\frac{p}{1-p}\right) + N \log(1-p) \end{aligned} \tag{10.60}$$

In general, the probability of an error occurring is low enough for us to assume $p < 1/2$. We also recognize that $N \log(1-p)$ is a constant for all $\mathbf{c}$. Accordingly, we may restate the maximum-likelihood decoding rule for the binary symmetric channel as follows:

$$\text{Choose the estimate } \hat{\mathbf{c}} \text{ that minimizes the Hamming distance between the received vector } \mathbf{r} \text{ and the transmitted vector } \mathbf{c}. \tag{10.61}$$

That is, for the binary symmetric channel, the maximum-likelihood decoder reduces to a *minimum distance decoder*. In such a decoder, the received vector $\mathbf{r}$ is compared with each possible transmitted code vector $\mathbf{c}$, and the particular one closest to $\mathbf{r}$ is chosen as the correct transmitted code vector. The term "closest" is used in the sense of minimum number of differing binary symbols (i.e., Hamming distance) between the code vectors under investigation.

The Viterbi Algorithm[9]

The equivalence between maximum likelihood decoding and minimum distance decoding for a binary symmetric channel implies that we may decode a convolutional code by choosing a path in the code tree whose coded sequence differs from the received sequence in the fewest number of places. Since a code tree is equivalent to a trellis, we may equally limit our choice to the possible paths in the trellis representation of the code. The reason for preferring the trellis over the tree is that the number of nodes at any level of the trellis

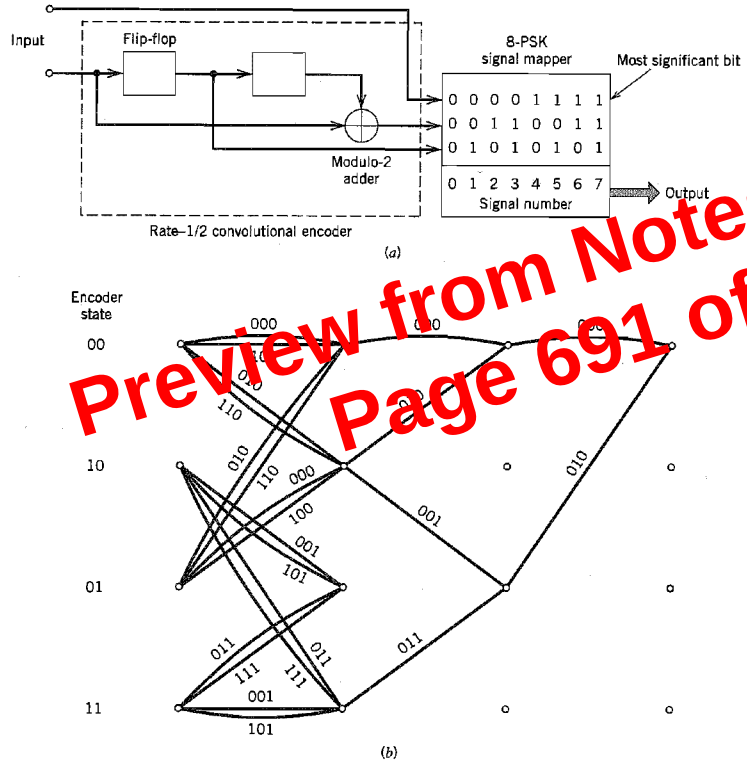

FIGURE 10.22 (*a*) Four-state Ungerboeck code for 8-PSK; the mapper follows Figure 10.20. (*b*) Trellis of the code.

sponding trellis of the code is shown in Figure 10.22*b*, which has four states. Note that the most significant bit of the incoming binary word is left uncoded. Therefore, each branch of the trellis may correspond to two different output values of the 8-PSK modulator or, equivalently, to one of the four 2-point subsets shown in Figure 10.20. The trellis of Figure 10.22*b* also includes the minimum distance path.

The scheme of Figure 10.23*a* depicts another Ungerboeck 8-PSK code for transmitting 2 bits/sample; it is next in the level of complexity. This second scheme uses a rate-2/3 convolutional encoder. Therefore, the corresponding trellis of the code has eight states, as shown in Figure 10.23*b*. In this case, both bits of the incoming binary word are encoded. Hence, each branch of the trellis corresponds to a specific output value of the 8-PSK modulator. The trellis of Figure 10.23*b* also includes the minimum distance path.

Figures 10.22*b* and 10.23*b* also include the encoder states. In Figure 10.22, the state of the encoder is defined by the contents of the two-stage shift register. On the other hand, in Figure 10.23, it is defined by the content of the single-stage (top) shift register followed by that of the two-stage (bottom) shift register.

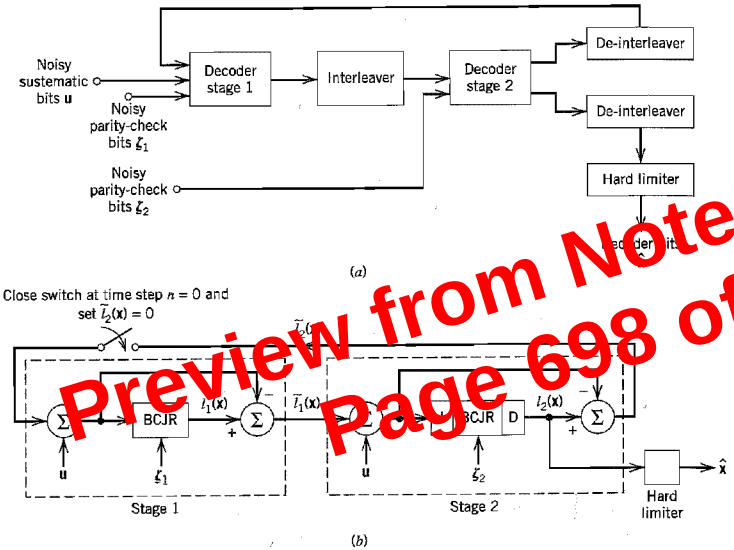

FIGURE 10.28 (*a*) Block diagram of turbo decoder. (*b*) Extrinsic form of turbo decoder, where I stands for interleaver, D for de-interleaver, and BCJR for BCJR algorithm for log-MAP decoding.

Each of the two decoding stages uses a *BCJR algorithm*,[17] which was originally invented by Bahl, Cocke, Jelinek, and Raviv (hence the name) to solve a *maximum a posteriori probability (MAP) detection* problem. The BCJR algorithm differs from the Viterbi algorithm in two fundamental respects:

1. The BCJR algorithm is a *soft input–soft output* decoding algorithm with two recursions, one forward and the other backward, both of which involve soft decisions. In contrast, the Viterbi algorithm is a *soft input–hard output* decoding algorithm, with a single forward recursion involving soft decisions; the recursion ends with a hard decision, whereby a particular survivor path among several ones is retained. In computational terms, the BCJR algorithm is therefore more complex than the Viterbi algorithm because of the backward recursion.
2. The BCJR algorithm is a MAP decoder in that it minimizes the bit errors by estimating the *a posteriori* probabilities of the individual bits in a code word; to reconstruct the original data sequence, the soft outputs of the BCJR algorithm are hard-limited. On the other hand, the Viterbi algorithm is a maximum likelihood sequence estimator in that it maximizes the likelihood function for the whole sequence, not each bit. As such, the average bit error rate of the BCJR algorithm can be slightly better than the Viterbi algorithm; it is never worse.

Most important, formulation of the BCJR algorithm rests on the fundamental assumptions that (1) the channel encoding, namely, the convolutional encoding performed in the transmitter, is modeled as a *Markov process*, and (2) the channel is memoryless. In the context of our present discussion, the Markovian assumption means that if a code can be repre-

where $\mathbf{s}(t)$ and $\boldsymbol{\lambda}(t)$ are both M-by-1 vectors. Then, for a rate $1/n$ linear convolutional code with feedback as in the RSC code, the probability that a symbol "1" was the message bit is given by

$$P(x(t) = 1 \,|\, \mathbf{y}) = \sum_{s \in \mathcal{F}_A} \lambda_s(t) \tag{10.78}$$

where $\mathcal{F}_A$ is the set of transitions that correspond to a symbol "1" at the input, and $\lambda_s(t)$ is the s-component of $\boldsymbol{\lambda}(t)$.

Define the *forward estimation* of state probabilities as the M-by-1 vector

$$\boldsymbol{\alpha}(t) = P(\mathbf{s}(t) \,|\, \mathbf{y}_{(1,t)}) \tag{10.79}$$

where the observation vector $\mathbf{y}_{(1,t)}$ is defined above. Also define the *backward estimation* of state probabilities as the M-by-1 vector

$$\boldsymbol{\beta}(t) = P(\mathbf{s}(t) \,|\, \mathbf{y}_{(t,k)}) \tag{10.80}$$

where

$$\mathbf{y}_{(t,k)} = [y(t), y(t+1), \ldots, y(k)]$$

The vectors $\boldsymbol{\alpha}(t)$ and $\boldsymbol{\beta}(t)$ are estimates of the state probabilities at time t based on the past and future data, respectively. We may then formulate the *separability theorem* as follows:

> The state probabilities at time t are related to the forward estimator $\boldsymbol{\alpha}(t)$ and backward estimator $\boldsymbol{\beta}(t)$ by the vector
>
> $$\boldsymbol{\lambda}(t) = \frac{\boldsymbol{\alpha}(t) \cdot \boldsymbol{\beta}(t)}{\| \boldsymbol{\alpha}(t) \cdot \boldsymbol{\beta}(t) \|_1} \tag{10.81}$$
>
> where $\boldsymbol{\alpha}(t) \cdot \boldsymbol{\beta}(t)$ is the vector product of $\boldsymbol{\alpha}(t)$ and $\boldsymbol{\beta}(t)$, and $\| \boldsymbol{\alpha}(t) \cdot \boldsymbol{\beta}(t) \|_1$ is the L_1 norm of this vector product.

The *vector product* $\boldsymbol{\alpha}(t) \cdot \boldsymbol{\beta}(t)$ (not to be confused with the inner product) is defined in terms of the individual elements of $\boldsymbol{\alpha}(t)$ and $\boldsymbol{\beta}(t)$ by

$$\boldsymbol{\alpha}(t) \cdot \boldsymbol{\beta}(t) = \begin{bmatrix} \alpha_1(t)\beta_1(t) \\ \alpha_2(t)\beta_2(t) \\ \vdots \\ \alpha_M(t)\beta_M(t) \end{bmatrix} \tag{10.82}$$

and the L_1 *norm* of $\boldsymbol{\alpha}(t) \cdot \boldsymbol{\beta}(t)$ is defined by

$$\| \boldsymbol{\alpha}(t) \cdot \boldsymbol{\beta}(t) \|_1 = \sum_{m=1}^{M} \alpha_m(t)\beta_m(t) \tag{10.83}$$

The separability theorem says that the state distribution at time t given the past is independent of the state distribution at time t given the future, which is intuitively satisfying recalling the Markovian assumption for channel encoding, which is basic to the BCJR algorithm. Moreover, this theorem provides the basis of a simple way of combining the forward and backward estimates to obtain a complete description of the state probabilities.

To proceed further, let the state transition probability at time t be defined by

$$\gamma_{m',m}(t) = P(s(t) = m, \, y(t) \,|\, s(t-1) = m') \tag{10.84}$$

In the vertical step, we may also update the *pseudo-posterior probabilities*:

$$P_j^0 = \alpha_j p_j^0 \prod_{i \in \mathcal{B}(j)} Q_{ij}^0$$

$$P_j^1 = \alpha_j p_j^1 \prod_{i \in \mathcal{B}(j)} Q_{ij}^1$$

where α_j is chosen to make

$$P_j^0 + P_j^1 = 1$$

The quantities obtained in the vertical step are used to compute a tentative estimate $\hat{\mathbf{c}}$. If the condition $\hat{\mathbf{c}}\mathbf{A}^T = \mathbf{0}$ is satisfied, the decoding algorithm is terminated. Otherwise, the algorithm goes back to the horizontal step. If after a maximum number of iterations (e.g., 100 or 200) there is no valid decoding, a decoding failure is declared. The decoding procedure described herein is a special case of the general low-complexity *sum-product algorithm*.

Simply stated, the sum-product algorithm passes probabilistic quantities between the check nodes and variable nodes of the bipartite graph. By virtue of the fact that each parity-check constraint can be represented by a simple convolutional coder with one bit of memory, we find that LDPC decoders are simpler to implement than turbo decoders, as stated earlier.

In terms of performance, however, we may say the following in light of experimental results reported in the literature: Regular LDPC codes do not appear to come as close to Shannon's limit as do their turbo code counterparts.

10.11 *Irregular Codes*

The turbo codes discussed in Section 10.8 and the LDPC codes discussed in Section 10.10 are both regular codes, each in its own individual way. The error-correcting performance of both of these codes over a noisy channel can be improved substantially by using their respective irregular forms.

In a standard turbo code with its encoder as shown in Figure 10.25, the interleaver maps each systematic bit to a unique input bit of convolutional encoder 2. In contrast, *irregular turbo codes*[22] use a special design of interleaver that maps some systematic bits to multiple input bits of the convolutional encoder. For example, each of 10 percent of the systematic bits may be mapped to eight inputs of the convolutional encoder instead of

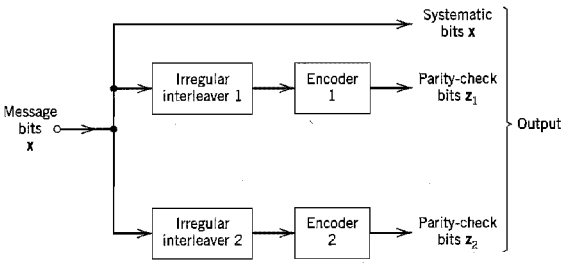

FIGURE 10.32 Block diagram of irregular turbo encoder.

is a nonnegative real number that is less than or equal to unity. Axiom (iii) states that the probability of the union of two mutually exclusive events is the sum of the probabilities of the individual events. These three axioms are sufficient to deal with experiments with finite sample spaces.

Although the axiomatic approach to probability theory is abstract in nature, all three axioms have relative-frequency interpretations of their own. Axiom (ii) corresponds to Equation (A1.1). Axiom (i) corresponds to the limiting case of Equation (A1.1) when the event A occurs in all the n trials. To interpret axiom (iii), we note that if event A occurs $N_n(A)$ times in n trials and event B occurs $N_n(B)$ times, then the union event "A or B" occurs in $N_n(A) + N_n(B)$ trials (since A and B can never occur on the same trial). Hence, $N_n(A + B) = N_n(A) + N_n(B)$, and so we have

$$\frac{N_n(A + B)}{n} = \frac{N_n(A)}{n} + \frac{N_n(B)}{n}$$

which has a mathematical form similar to that of axiom (iii).

Axioms (i), (ii), and (iii) constitute an implicit definition of probability. We may use these axioms to develop some other basic properties of probability, as described next.

Property 1

$$P(\overline{A}) = 1 - P(A) \tag{A1.6}$$

where $\overline{A}$ (denoting "not A") is the complement of event A.

The use of this property helps us investigate the *nonoccurrence of an event*. To prove it, we express the sample space S as the union of two mutually exclusive events A and $\overline{A}$:

$$\mathsf{S} = A + \overline{A}$$

Then, the use of axioms (i) and (iii) yields

$$1 = P(A) + P(\overline{A})$$

from which Equation (A1.6) follows directly.

Property 2

If M mutually exclusive events $A_1, A_2, \ldots, A_M$ have the exhaustive property

$$A_1 + A_2 \cdots + A_M = \mathsf{S} \tag{A1.7}$$

then

$$P(A_1) + P(A_2) + \cdots + P(A_M) = 1 \tag{A1.8}$$

To prove this property, we first use axiom (i) in Equation (A1.7), and so write

$$P(A_1 + A_2 + \cdots + A_M) = 1$$

Next, we generalize axiom (iii) by writing

$$P(A_1 + A_2 + \cdots + A_M) = P(A_1) + P(A_2) + \cdots + P(A_M)$$

We next consider a more general situation. Let X denote a random variable, and let $g(X)$ denote a function of X defined on the real line. The quantity obtained by letting the argument of the function $g(X)$ be a random variable is also a random variable, which we denote as

$$Y = g(X) \tag{A1.31}$$

To find the expected value of the random variable Y, we could of course find the probability density function $f_Y(y)$ and then apply the standard formula

$$E[Y] = \int_{-\infty}^{\infty} y f_Y(y)\, dy$$

A simpler procedure, however, is to write

$$E[g(X)] = \int_{-\infty}^{\infty} g(x) f_X(x)\, dx \tag{A1.32}$$

Indeed, Equation (A1.32) may be viewed as generalizing the concept of expected value to an arbitrary function $g(X)$ of a random variable X.

■ MOMENTS

For the special case of $g(X) = X^n$, using Equation (A1.32) we obtain the nth *moment* of the probability distribution of the random variable X; that is,

$$E[X^n] = \int_{-\infty}^{\infty} x^n f_X(x)\, dx \tag{A1.33}$$

By far the most important moments of X are the first two moments. Thus putting $n = 1$ in Equation (A1.33) gives the mean of the random variable as shown in Eq. (A1.27), whereas putting $n = 2$ gives the *mean-square value* of X:

$$E[X^2] = \int_{-\infty}^{\infty} x^2 f_X(x)\, dx \tag{A1.34}$$

We may also define *central moments*, which are simply the moments of the difference between a random variable X and its mean μ_X. Thus, the nth central moment is

$$E[(X - \mu_X)^n] = \int_{-\infty}^{\infty} (x - \mu_X)^n f_X(x)\, dx \tag{A1.35}$$

For $n = 1$, the central moment is, of course, zero, whereas for $n = 2$ the second central moment is referred to as the *variance* of the random variable X, which is written as

$$\text{var}[X] = E[(X - \mu_X)^2] = \int_{-\infty}^{\infty} (x - \mu_X)^2 f_X(x)\, dx \tag{A1.36}$$

The variance of a random variable X is commonly denoted as σ_X^2. The square root of the variance, namely, σ_X, is called the *standard deviation* of the random variable X.

The variance σ_X^2 of a random variable X in some sense is a measure of the variable's "randomness." By specifying the variance σ_X^2, we essentially constrain the effective width of the probability density function $f_X(x)$ of the random variable X about the mean μ_X.

unaffected by transmission through the device. Such an ideal device is referred to as a *Hilbert transformer*.

Properties of the Hilbert Transform

The Hilbert transform differs from the Fourier transform in that it operates exclusively in the time domain. It has a number of useful properties, some of which are listed next. The signal $g(t)$ is assumed to be real valued, which is the usual domain of application of the Hilbert transform. For this class of signals, we may state the following:

1. A signal $g(t)$ and its Hilbert transform $\hat{g}(t)$ have the same magnitude spectrum.
2. If $\hat{g}(t)$ is the Hilbert transform of $g(t)$, then the Hilbert transform of $\hat{g}(t)$ is $-g(t)$.
3. A signal $g(t)$ and its Hilbert transform $\hat{g}(t)$ are orthogonal over the entire time interval $(-\infty, \infty)$, as shown by

$$\int_{-\infty}^{\infty} g(t)\hat{g}(t)\,dt = 0$$

Proofs of these properties are left as exercises for the reader; the proofs follow from Equations (A2.31), (A2.32) and (A2.35).

A2.4 *Complex Representation of Signals and Systems*

Pre-Envelope

Consider a real-valued signal $g(t)$. We define the *pre-envelope*, or *analytic signal*, of the signal $g(t)$ as the complex-valued function

$$g_+(t) = g(t) + j\hat{g}(t) \tag{A2.36}$$

where $\hat{g}(t)$ is the Hilbert transform of $g(t)$. We note that the given signal $g(t)$ is the real part of the pre-envelope $g_+(t)$, and the Hilbert transform of the signal is the imaginary part of the pre-envelope. Just as the use of phasors simplifies manipulations of alternating currents and voltages, so we find that the pre-envelope is particularly useful in handling band-pass signals and systems.

One of the important features of the pre-envelope $g_+(t)$ is the behavior of its Fourier transform. Let $G_+(f)$ denote the Fourier transform of $g_+(t)$. Then we may write

$$G_+(f) = G(f) + \text{sgn}(f)G(f)$$

from which we readily find that

$$G_+(f) = \begin{cases} 2G(f), & f > 0 \\ G(0), & f = 0 \\ 0, & f < 0 \end{cases} \tag{A2.37}$$

where $G(0)$ is the value of $G(f)$ at frequency $f = 0$. This means that the pre-envelope of a signal has no frequency content (i.e., its Fourier transform vanishes) for all negative frequencies.

Since both $g_I(t)$ and $g_Q(t)$ are low-pass signals limited to the band $-W \leq f \leq W$, they may be derived from the band-pass signal $g(t)$ using the scheme shown in Figure A2.6*a*. Both low-pass filters in this figure are identical, each of which has a bandwidth equal to W. To reconstruct $g(t)$ from its in-phase and quadrature components, we may use the scheme shown in Figure A2.6*b*.

The two schemes shown in Figure A2.6 are basic to the study of *linear modulation systems*. The multiplication of the low-pass in-phase component $g_I(t)$ by $\cos(2\pi f_c t)$ and the multiplication of the low-pass quadrature component $g_Q(t)$ by $\sin(2\pi f_c t)$ represent linear forms of modulation. Given that the carrier frequency f_c is sufficiently large, the resulting band-pass function $g(t)$ defined in Equation (A2.45) is referred to as a *passband signaling waveform*. Correspondingly, the mapping from $g_I(t)$ and $g_Q(t)$ into $g(t)$ is known as *passband modulation*.

Equation (A2.44) is the Cartesian form of expressing the complex envelope $\tilde{g}(t)$. Alternatively, we may express it in the polar form

$$\tilde{g}(t) = a(t) \exp[j\phi(t)] \tag{A2.46}$$

where $a(t)$ and $\phi(t)$ are both real-valued low-pass functions. Based on this polar representation, the original band-pass signal $g(t)$ is defined by

$$g(t) = a(t) \cos[2\pi f_c t + \phi(t)] \tag{A2.47}$$

We refer to $a(t)$ as the *natural envelope* or simply the *envelope* of the band-pass signal $g(t)$ and to $\phi(t)$ as the *phase* of the signal. Equation (A2.47) represents a *hybrid form of amplitude modulation and angle modulation*; indeed, it includes amplitude modulation, frequency modulation, and phase modulation as special cases.

From this discussion it is apparent that, whether we represent a band-pass (modulated) signal $g(t)$ in terms of its in-phase and quadrature components as in Equation (A2.45) or in terms of its envelope and phase as in Equation (A2.47), the information content of the signal $g(t)$ is completely preserved in the complex envelope $\tilde{g}(t)$.

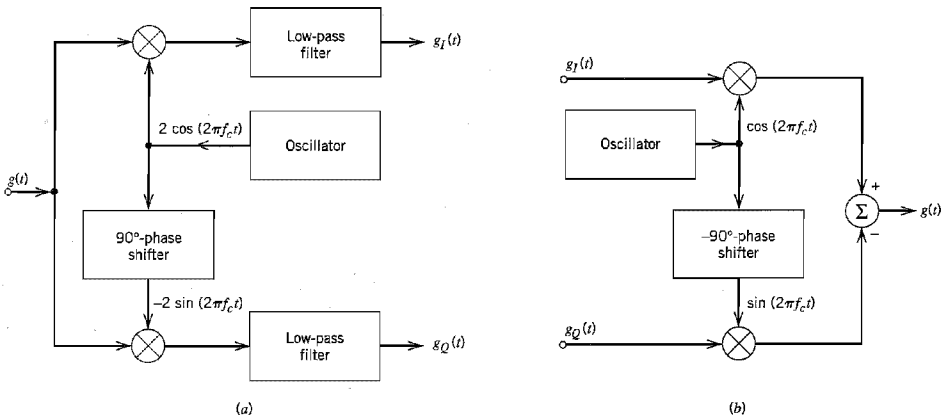

FIGURE A2.6 (*a*) Scheme for deriving the in-phase and quadrature components of a band-pass signal. (*b*) Scheme for reconstructing the band-pass signal from its in-phase and quadrature components.

shifting it to the origin and then scaling it by the factor 2. To determine the complex impulse response $h(t)$, we take the inverse Fourier transform of $\tilde{H}(f)$, obtaining

$$\tilde{h}(t) = \int_{-\infty}^{\infty} \tilde{H}(f) \exp(j2\pi ft)\, df \tag{A2.56}$$

The representations just described for band-pass signals and systems provide the basis of an efficient method for determining the output of a band-pass system driven by a band-pass signal. We assume that the spectrum of the input signal $x(t)$ and the frequency response $H(f)$ of the system are both centered around the same frequency. In practice, there is no need to consider a situation in which the carrier frequency of the input signal is not aligned with the midband frequency of the band-pass system, since we have considerable freedom in choosing the carrier or midband frequency. Thus, changing the carrier frequency of the input signal by an amount Δf_c simply corresponds to absorbing (or removing) the factor $\exp(\pm j2\pi \Delta f_c t)$ into the complex envelope of the input signal or the complex impulse response of the band-pass system. We are therefore justified in proceeding on the assumption that $X(f)$ and $H(f)$ are both centered around f_c. Suppose then we use $y(t)$ to denote the output signal of the system. It is clear that $y(t)$ is also a band-pass signal, so that we may represent it in terms of its low-pass complex envelope $\tilde{y}(t)$, as follows:

$$y(t) = \mathrm{Re}[\tilde{y}(t) \exp(j2\pi f_c t)] \tag{A2.57}$$

The output signal $y(t)$ is related to the input signal $x(t)$ and impulse response $h(t)$ of the system in the usual way by the convolution integral

$$y(t) = \int_{-\infty}^{\infty} h(\tau)x(t - \tau)\, d\tau \tag{A2.58}$$

In terms of pre-envelopes, we have $h(t) = \mathrm{Re}[h_+(t)]$ and $x(t) = \mathrm{Re}[x_+(t)]$. We may therefore rewrite Equation (A2.58) in terms of the pre-envelopes $x_+(t)$ and $h_+(t)$ as follows:

$$y(t) = \int_{-\infty}^{\infty} \mathrm{Re}[h_+(\tau)]\, \mathrm{Re}[x_+(t - \tau)]\, d\tau \tag{A2.59}$$

To proceed further, we make use of a basic property of pre-envelopes that is described by the following relation (presented here without proof):

$$\int_{-\infty}^{\infty} \mathrm{Re}[h_+(\tau)]\, \mathrm{Re}[x_+(\tau)]\, d\tau = \frac{1}{2}\mathrm{Re}\left[\int_{-\infty}^{\infty} h_+(\tau)x_+^*(\tau)\, d\tau\right] \tag{A2.60}$$

where we have used τ as the integration variable to be consistent with that in Equation (A2.59). Next, we note that using $x(-\tau)$ in place of $x(\tau)$ has the effect of removing the complex conjugation on the right-hand side of Equation (A2.60). Hence, bearing in mind the algebraic difference between the argument of $x_+(\tau)$ in Equation (A2.60) and that of $x_+(t - \tau)$ in Equation (A2.59), and using the relationship between the pre-envelope and complex envelope of a band-pass function, we get

$$\begin{aligned} y(t) &= \frac{1}{2}\mathrm{Re}\left[\int_{-\infty}^{\infty} h_+(\tau)x_+(t - \tau)\, d\tau\right] \\ &= \frac{1}{2}\mathrm{Re}\left[\int_{-\infty}^{\infty} \tilde{h}(\tau) \exp(j2\pi f_c\tau)\tilde{x}(t - \tau) \exp(j2\pi f_c(t - \tau))\, d\tau\right] \\ &= \frac{1}{2}\mathrm{Re}\left[\exp(j2\pi f_c t) \int_{-\infty}^{\infty} \tilde{h}(\tau)\tilde{x}(t - \tau)\, d\tau\right] \end{aligned} \tag{A2.61}$$

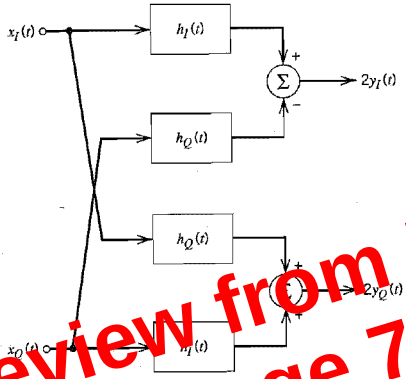

FIGURE A2.8 Block diagram illustrating the relationship between the in-phase and quadrature components of the response of a band-pass filter and those of the input signal.

and for the quadrature component $y_Q(t)$ the relation

$$2y_Q(t) = h_Q(t) \bigstar x_I(t) + h_I(t) \bigstar x_Q(t) \tag{A2.68}$$

Thus, for the purpose of evaluating the in-phase and quadrature components of the complex envelope $\tilde{y}(t)$ of the system output, we may use the *low-pass equivalent model* shown in Figure A2.8. All the signals and impulse responses shown in this model are real-valued low-pass functions. Accordingly, this equivalent model provides a practical basis for the efficient simulation of band-pass filters or communication channels on a digital computer.

To sum up, the procedure for evaluating the response of a band-pass system (with mid-band frequency f_c) to an input band-pass signal (of carrier frequency f_c) is as follows:

1. The input band-pass signal $x(t)$ is replaced by its complex envelope $\hat{x}(t)$, which is related to $x(t)$ by
$$x(t) = \text{Re}[\hat{x}(t) \exp(j2\pi f_c t)]$$
2. The band-pass system, with impulse response $h(t)$, is replaced by a low-pass analog, which is characterized by a complex impulse response $\tilde{h}(t)$ related to $h(t)$ by
$$h(t) = \text{Re}[\tilde{h}(t) \exp(j2\pi f_c t)]$$
3. The complex envelope $\tilde{y}(t)$ of the output band-pass signal $y(t)$ is obtained by convolving $\tilde{h}(t)$ with $\tilde{x}(t)$, as shown by
$$2\tilde{y}(t) = \tilde{h}(t) \bigstar \tilde{x}(t)$$
4. The desired output $y(t)$ is finally derived from the complex envelope $\tilde{y}(t)$ by using the relation
$$y(t) = \text{Re}[\tilde{y}(t) \exp(j2\pi f_c t)]$$

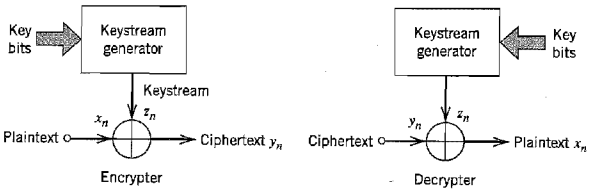

FIGURE A5.4 Binary additive stream cipher.

We thus see that in binary additive stream ciphers, identical devices can be used to perform encryption and decryption, as shown in Figure A5.4. The secret key is chosen according to some probability distribution. To provide secure encryption, the keystream should resemble a coin-tossing (i.e., completely random) sequence as closely as possible.

Block ciphers are normally designed in such a way that a small change in an input block of plaintext produces a major change in the resulting output. This *error propagation* property of block ciphers is valuable in authentication in that it makes it improbable for an enemy cryptanalyst to modify encrypted data, unless knowledge of the key is available. On the other hand, a binary additive stream cipher has *no* error propagation; the decryption of a distorted bit in the ciphertext affects only the corresponding bit of the resulting output.

Stream ciphers are generally better suited for the secure transmission of data over error-prone communication channels; they are used in applications where high data rates are a requirement (as in secure video, for example) or when a minimal transmission delay is essential.[5]

■ REQUIREMENT FOR SECRECY

In cryptography, a fundamental assumption is that an enemy cryptanalyst has knowledge of the entire mechanism used to perform encryption, except for the secret key. We may identify the following forms of attack that may be attempted by the enemy cryptanalyst, depending on the availability of additional knowledge:

1. *Ciphertext-only attack* is a cryptanalytic attack in which the enemy cryptanalyst has access to part or all of the ciphertext.
2. *Known-plaintext attack* is a cryptanalytic attack in which the enemy cryptanalyst has knowledge of some ciphertext–plaintext pairs formed with the actual secret key.
3. *Chosen-plaintext attack* is a cryptanalytic attack in which the enemy cryptanalyst is able to submit any chosen plaintext message and receive in return the correct ciphertext for the actual secret key.
4. *Chosen-ciphertext attack* is a cryptanalytic attack in which the enemy cryptanalyst is able to choose an arbitrary ciphertext and find the correct result for its decryption.

A ciphertext-only attack occurs frequently in practice. In this form of attack, an enemy cryptanalyst uses only knowledge of the statistical structure of the language in use (e.g., in English the letter *e* occurs with a probability of 13 percent, and the letter *q* is always followed by *u*) and knowledge of some probable words (e.g., a letter probably begins with "Dear Sir/Madam:"). A known-plaintext attack may take place by virtue of the standard computer formats used in programming languages and data generation. In any case, the ciphertext-only attack is viewed as the weakest threat to which a crypto-

Role of Data Compression in Cryptography

Lossless data compression or data compaction is a useful tool in cryptography. We say this because data compaction removes redundancy, thereby increasing the unicity distance N_0 in accordance with Equation (A5.11). To exploit this idea, data compaction is used prior to encryption in the transmitter, and the redundant information is reinserted after decryption in the receiver; the net result is that the authorized user at the receiver output sees no difference, and yet the information transmission has been made more secure. It would be tempting to consider the use of perfect data compaction to remove all redundancy, thereby transforming a message source into a completely random source and resulting in $N_0 = \infty$ with any key size. Unfortunately, we do not have a device capable of performing perfect data compaction on realistic message sources, nor is it likely that there will ever be such a device. It is therefore futile to rely on data compaction alone for data security. Nevertheless, limited data compaction tends to increase security, which is the reason why cryptographers like to use data compression as a useful trick.

Diffusion and Confusion

In the Shannon model of cryptography, two methods suggest themselves as general principles to guide the design of practical ciphers. The methods are called *diffusion* and *confusion*, the aims of which (by themselves or together) are to frustrate a statistical analysis of ciphertext by the enemy and therefore make it extremely difficult to break the cipher.

In the method of diffusion, the statistical structure of the plaintext is hidden by spreading out the influence of a single bit in the plaintext over a large number of bits in the ciphertext. This spreading has the effect of forcing the enemy to intercept a tremendous amount of material for the determination of the statistical structure of the plaintext, since the structure is evident only in many blocks, each one of which has a very small probability of occurrence. In the method of confusion, the data transformations are designed to complicate the determination of the way in which the statistics of the ciphertext depend on the statistics of the plaintext. Thus, a good cipher uses a combination of diffusion and confusion.

For a cipher to be of practical value, however, it must not only be difficult to break the cipher by an enemy cryptanalyst, but also it should be easy to encrypt and decrypt data given knowledge of the secret key. We may satisfy these two design objectives using a *product cipher*, based on the notion of "divide and conquer." Specifically, the implementation of a strong cipher is accomplished as a succession of simple component ciphers, each of which contributes a modest amount of diffusion and confusion to the overall makeup of the cipher. Product ciphers are often built using substitution ciphers and transposition ciphers as basic components; these simple ciphers are described next.

1. *Substitution cipher.*

In a substitution cipher each letter of the plaintext is replaced by a fixed substitute, usually also a letter from the same alphabet, with the particular substitution rule being determined by the secret key. Thus the plaintext

$$\mathbf{X} = (x_1, x_2, x_3, x_4, \ldots)$$

where $x_1, x_2, x_3, \ldots$ are the successive letters, is transformed into the ciphertext

$$\begin{aligned} \mathbf{Y} &= (y_1, y_2, y_3, y_4, \ldots) \\ &= (f(x_1), f(x_2), f(x_3), f(x_4), \ldots) \end{aligned} \tag{A5.15}$$

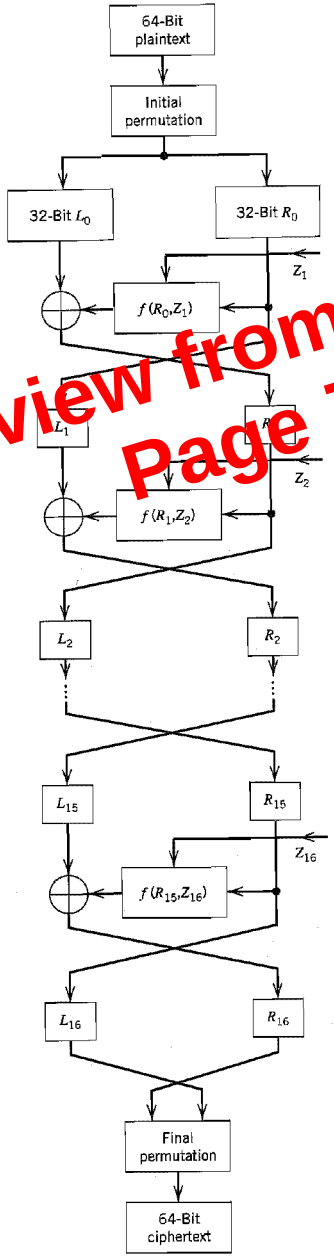

FIGURE A5.8 Data encryption standard. (From Diffie and Hellman, 1979, with permission of the IEEE.)

The notion emerging from the description of a public-key cryptographic system presented herein is that the keys come in inverse pairs (i.e., public key and private key), and that each pair of keys has two basic properties:

1. *Whatever message is encrypted with one of the keys can be decrypted with the other key.*
2. *Given knowledge of the public key, it is computationally infeasible to find the secret key.*

The use of public-key cryptography as described herein makes it possible to solve the secrecy problem as follows. Subscribers to a secure communication system list their public keys in a "telephone directory" along with their name and address. A subscriber can then send a private message to another subscriber simply by looking up the public key of the addressee and using the key to encrypt the message. The encrypted message (i.e., ciphertext) can only be read by the holder of that particular public key. In fact, should the original message (i.e., plaintext) be lost, even its sender would find it extremely difficult to recover the message from the ciphertext.

The key management of public-key cryptography makes it well suited for the development of large, secure communication networks. Indeed, it has evolved from a simple concept to a mainstay of cryptographic technology.

■ DIFFIE–HELLMAN PUBLIC KEY DISTRIBUTION

In a simple and yet elegant system known as the *Diffie–Hellman public key-distribution system*, use is made of the fact that it is easy to calculate a discrete exponential but difficult to calculate a discrete logarithm. To be more specific, consider the *discrete exponential function*

$$Y = \alpha^X \bmod p \qquad \text{for } 1 \le X \le p - 1 \tag{A5.24}$$

where the arithmetic is performed modulo-p. The α is an integer that should be *primitive* (i.e., all powers of α generate all the elements mod p relatively prime to $p - 1$). Correspondingly, X is referred to as the *discrete logarithm* of Y to the base α, mod p, as shown by

$$X = \log_\alpha Y \bmod p \qquad \text{for } 1 \le Y \le p - 1 \tag{A5.25}$$

The calculation of Y from X is easy, using the trick of square-and-multiply. For example, for $X = 16$ we have

$$Y = \alpha^{16} = \{[(\alpha^2)^2]^2\}^2$$

On the other hand, the problem of calculating X from Y is much more difficult.

In the Diffie–Hellman public key-distribution system, all users are presumed to know both α and p. A user i, say, selects an independent random number X_i uniformly from the set of integers $\{1, 2, \ldots, p\}$ that is kept as a *private secret*. But the discrete exponential

$$Y_i = \alpha^{X_i} \bmod p \tag{A5.26}$$

is deposited in a *public directory* with the user's name and address. Every other user of the system does the same thing. Now, suppose that users i and j wish to communicate

GLOSSARY

Conventions and Notations

1. The symbol | | means the absolute value, or magnitude, of the complex quantity contained within.
2. The symbol arg( ) means the phase angle of the complex quantity contained within.
3. The symbol Re[ ] means the "real part of," and Im[ ] means the "imaginary part of."
4. Unless stated otherwise, the natural logarithm is denoted by log. Logarithms to bases 2 and 10 are denoted by $\log_2$ and $\log_{10}$, respectively.
5. The use of an asterisk as a superscript denotes complex conjugate (e.g., x^* is the complex conjugate of x).
6. The symbol $\rightleftharpoons$ indicates a Fourier transform pair, e.g., $g(t) \rightleftharpoons G(f)$, where a lowercase letter denotes the time function and a corresponding uppercase letter denotes the frequency function.
7. The symbol $F[\ ]$ indicates the Fourier-transform operation, e.g., $F[g(t)] = G(f)$, and the symbol $F^{-1}[\ ]$ indicates the inverse Fourier-transform operation, e.g., $F^{-1}[G(f)] = g(t)$.
8. The symbol $\bigstar$ denotes convolution, e.g.,

$$x(t) \bigstar h(t) = \int_{-\infty}^{\infty} x(\tau)h(t - \tau)\, d\tau$$

9. The symbol $\oplus$ denotes modulo-2 addition, except in Chapter 10 where binary arithmetic is used and modulo-2 addition is denoted by an ordinary plus sign throughout that chapter.
10. The use of subscript T_0 indicates that the pertinent function $g_{T_0}(t)$, say, is a periodic function of time t with period T_0.
11. The use of a hat over a function indicates one of two things:
    (a) the Hilbert transform of a function, e.g., the function $\hat{g}(t)$ is the Hilbert transform of $g(t)$, or
    (b) the estimate of an unknown parameter, e.g., the quantity $\hat{\alpha}(\mathbf{x})$ is an estimate of the unknown parameter α, based on the observation vector $\mathbf{x}$.
12. The use of a tilde over a function indicates the complex envelope of a narrowband signal, e.g., the function $\tilde{g}(t)$ is the complex envelope of the narrowband signal $g(t)$. The exception to this convention is in Section 10.8, where, in the description of turbo decoding, the tilde is used to signify extrinsic information and thereby distinguish it from log-likelihood ratio.
13. The use of subscript + indicates the pre-envelope of a signal, e.g., the function $g_+(t)$ is the pre-envelope of the signal $g(t)$. We may thus write $g_+(t) = g(t) + j\hat{g}(t)$, where $\hat{g}(t)$ is the Hilbert transform of $g(t)$. The use of subscript − indicates that $g_-(t) = g(t) - j\hat{g}(t) = g_+^*(t)$.
14. The use of subscripts I and Q indicates the in-phase and quadrature components of a narrowband signal, a narrowband random process, or the impulse response of a narrow-band filter, with respect to the carrier $\cos(2\pi f_c t)$.

15. For a low-pass message signal, the highest frequency component or message bandwidth is denoted by W. The spectrum of this signal occupies the frequency interval $-W \leq f \leq W$ and is zero elsewhere. For a band-pass signal with carrier frequency f_c, the spectrum occupies the frequency intervals, $f_c - W \leq f \leq f_c + W$ and $-f_c - W \leq f \leq -f_c + W$, and so $2W$ denotes the bandwidth of the signal. The (low-pass) complex envelope of this band-pass signal has a spectrum that occupies the frequency interval $-W \leq f \leq W$.

    For a lowpass filter, the bandwidth is denoted by B. A common definition of filter bandwidth is the frequency at which the magnitude response of the filter drops by 3 dB below the zero-frequency value. For a band-pass filter centered on carrier frequency f_c the bandwidth is denoted by $2B$, centered on f_c. The complex low-pass equivalent of this band-pass filter has a bandwidth equal to B.

    The transmission bandwidth of a communication channel, required to transmit a modulated wave, is denoted by B_T.

16. Random variables or random vectors are uppercase (e.g., X or $\mathbf{X}$), and their sample values are lowercase (e.g., x or $\mathbf{x}$).

17. A vertical bar in an expression means "given that," e.g., $f_X(x|H_0)$ is the probability density function of the random variable X, given that hypothesis H_0 is true.

18. The symbol $E[\ ]$ means the expected value of the random variable enclosed within; the E acts as an operator.

19. The symbol var[ ] means the variance of the random variable enclosed within.

20. The symbol cov[ ] means the covariance of the two random variables enclosed within.

21. The average probability of symbol error is denoted by P_e.

    In the case of binary signaling techniques, p_{10} denotes the conditional probability of error given that symbol 0 was transmitted, and p_{01} denotes the conditional probability of error given that symbol 1 was transmitted. The *a priori* probabilities of symbols 0 and 1 are denoted by p_0 and p_1, respectively.

22. The symbol $\langle\ \rangle$ denotes the time average of the sample function enclosed within.

23. Boldface letter denotes a vector or matrix. The inverse of a square matrix $\mathbf{R}$ is denoted by $\mathbf{R}^{-1}$. The transpose of a vector $\mathbf{w}$ is denoted by $\mathbf{w}^T$. The Hermitian transpose of a complex-valued vector $\mathbf{x}$ is denoted by $\mathbf{x}^H$; Hermitian transposition involves both transposition and complex conjugation.

24. The length of a vector $\mathbf{x}$ is denoted by $\|\mathbf{x}\|$. The Euclidean distance between the vectors $\mathbf{x}_i$ and $\mathbf{x}_j$ is denoted by $d_{ij} = \|\mathbf{x}_i - \mathbf{x}_j\|$.

25. The inner product of two real-valued vectors $\mathbf{x}$ and $\mathbf{y}$ is denoted by $\mathbf{x}^T\mathbf{y}$; their outer product is denoted by $\mathbf{x}\mathbf{y}^T$. If the vectors $\mathbf{x}$ and $\mathbf{y}$ are complex valued, their inner product is $\mathbf{x}^H\mathbf{y}$, and their outer product is $\mathbf{x}\mathbf{y}^H$.

26. The vector product of two M-by-1 vectors $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$ is an M-by-1 vector defined by

$$\boldsymbol{\alpha} \cdot \boldsymbol{\beta} = \begin{bmatrix} \alpha_1\beta_1 \\ \alpha_2\beta_2 \\ \vdots \\ \alpha_M\beta_M \end{bmatrix}$$

    where α_k and β_k are the kth elements of $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$, respectively. The L_1 norm of the vector product $\boldsymbol{\alpha} \cdot \boldsymbol{\beta}$ is defined by

$$\|\boldsymbol{\alpha} \cdot \boldsymbol{\beta}\|_1 = \sum_{m=1}^{M} \alpha_m\beta_m$$

PCM:	pulse-code modulation
PDM:	pulse-duration modulation
PG:	processing gain
PLL:	phase-locked loop
PN:	pseudo-noise
POTS:	plain old telephone service
PPM:	pulse-position modulation
PSK:	phase-shift keying
PSTN:	public switched telephone network
PWM:	pulse-width modulation
QAM:	quadrature amplitude modulation
QoS:	quality of service
QPSK:	quadriphase-shift keying
RF:	radio frequency
rms:	root-mean-square
RS:	Reed-Solomon
RS-232	Recommended standard-232 (port)
RSA:	Rivest-Shamir-Adelman
RSC:	recursive systematic convolutional (code)
RZ:	return-to-zero
s:	second
SDH:	synchronous digital hierarchy
SDMA:	space-division multiple access
SDR:	signal-to-distortion ratio
SNR:	signal-to-noise ratio
SONET:	synchronous optical network
STFT:	short-time Fourier transform
STM:	synchronous transfer mode
TC:	time compression
TCM:	trellis-coded modulation
TDM:	time-division multiplexing
TDMA:	time-division multiple access
TV:	television
UHF:	ultra high frequency
V:	volt
VCO:	voltage-controlled oscillator
VHF:	very high frequency
VLSI:	very-large-scale integration
W:	watt
WDM:	wavelength division multiplexing

L.R. Bahl, J. Cocke, F. Jelinek, and J. Raviv, "Optimal decoding of linear codes for minimizing symbol error rate," *IEEE Transactions on Information Theory*, vol. IT-20, pp. 284–287, 1974.

G. Battail, "Coding for the Gaussian channel: the promise of weighted output decoding," *International J. Satellite Communications*, vol. 7, pp. 183–192, 1989.

G. Battail, "Pondération des symbols décodés par l'algorithme de Viterbi," *Ann. Télécommunication*, vol. 42, pp. 31–38, 1987.

E. Bedrosian, "The analytic signal representation of modulated waveforms," *Proceedings of the IRE*, vol. 50, pp. 2071–2076, 1962.

P.A. Bello, "Characterization of randomly time-variant linear channels," *IEEE Transactions on Communication Systems*, vol. CS-11, pp. 360–393, 1963.

S. Benedetto and G. Montorsi, "Unveiling turbo codes: Some results on parallel concatenated coding schemes," *IEEE Transactions on Information Theory*, vol. 42, pp. 409–428, 1996.

W.R. Bennett, "Spectra of quantized signals," *Bell System Tech. J.*, vol. 27, pp. 446–472, 1948.

N. Benvenuto, et al., "The 32 kb/s ADPCM coding standard," *AT&T Technical Journal*, vol. 65, pp. 12–22, Sept./Oct. 1986.

C. Berrou and A. Glavieux, "Near optimum error correcting coding and decoding: turbo codes," *IEEE Transactions on Communications*, vol. 44, no. 10, pp. 1261–1271, 1996.

C. Berrou and A. Glavieux, "Reflections on the Prize Paper: Near optimum error-correcting coding and decoding turbo codes," *IEEE Information Theory Society Newsletter*, vol. 48, no. 2, p. 1 and pp. 24–31, June 1998.

C. Berrou, A. Glavieux, and P. Thitmajshima, "Near Shannon limit error-correction coding and decoding: turbo codes," *International Conference on Communications*, pp. 1064–1090, Geneva, Switzerland, May 1993.

V.K. Bhargava, "Forward error correction schemes for digital communications," *IEEE Communications Magazine*, vol. 21, no. 1, pp. 11–19, 1983.

R.C. Bose and D.K. Ray-Chaudhuri, "On a class of error correcting binary group codes," *Information and Control*, vol. 3, pp. 68–79, 1960.

K. Brandenburg and G. Stoll, "ISO-MPEG-1 Audio: A generic standard for coding of high-quality digital audio," *Journal of the Audio Engineering Society*, vol. 42, pp. 780–792, 1994.

D.G. Brennan, "Linear diversity combining techniques," *Proceedings of the IRE*, vol. 47, pp. 1075–1102, 1959.

A. Buzo, A.H. Gray, Jr., R.M. Gray, and J.D. Markel, "Speech coding based upon vector quantization," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. ASSP-28, pp. 562–574, 1980.

C.R. Cahn, "Combined digital phase and amplitude modulation communication systems," *IRE Transactions on Communication Systems*, vol. CS-8, pp. 150–155, 1960.

J.R. Carson, "Notes on the theory of modulation," *Proceedings of the IRE*, vol. 10, pp. 57–64, 1922.

J.R. Carson and T.C. Fry, "Variable frequency electric circuit theory with application to the theory of frequency modulation," *Bell System Tech. J.*, vol. 16, pp. 513–540, 1937.

E.F. Casas and C. Leung, "OFDM for data communication over mobile radio FM channels," *IEEE Transactions on Communications*, vol. 39, pp. 783–793, 1991.

J.G. Chaffee, "The application of negative feedback to frequency-modulation systems," *Bell System Tech. J.*, vol. 18, pp. 404–437, 1939.

R.W. Chang, "Synthesis of band-limited orthogonal signals for multichannel data transmission," *Bell System Tech. J.*, vol. 45, pp. 1775–1796, 1996.

W.Y. Chen, G.H. Im, and J.J. Werner, "Design of digital carrierless AM/PM transceivers," *Standard Project, T1E1.4/92-149, AT&T and Bellcore*, August 19, 1992.

S. Chennakeshu and G.J. Sauliner, "Differential detection of $\pi/4$-shifted-DQPSK for digital cellular radio," *IEEE Transactions on Vehicular Technology*, vol. 42, pp. 46–57, 1993.

J.M. Cioffi, V. Oksman, J.-J. Werner, T. Pollet, P.M.P. Spruyt, J.S. Chow, and K.S. Jacobsen, "Very-high-speed digital subscriber lines," *IEEE Communications Magazine*, vol. 37, pp. 72–79, April, 1999.

L.J. Cimini, Jr., and Y. Li, "Orthogonal frequency division multiplexing for wireless communica-