

- $1/T$, with the result that two independent bit streams can be transmitted simultaneously and subsequently detected in the receiver.
- In the case of coherent MSK, there are two orthogonal carriers, namely, $\sqrt{2/T_b} \cos(2\pi f_c t)$ and $\sqrt{2/T_b} \sin(2\pi f_c t)$, which are modulated by the two antipodal symbol shaping pulses $\cos(\pi t/2T_b)$ and $\sin(\pi t/2T_b)$, respectively, over $2T_b$ intervals, where T_b is the bit duration. Correspondingly, the receiver uses a coherent phase decoding process over two successive bit intervals to recover the original bit stream.
 - The MSK scheme differs from its counterpart, the QPSK, in that its receiver has *memory*. In particular, the MSK receiver makes decisions based on observations over two successive bit intervals. Thus, although the transmitted signal has a binary format represented by the transmission of two distinct frequencies, the presence of memory in the receiver makes it assume a two-dimensional signal space diagram. There are four message points, depending on which binary symbol (0 or 1) was sent and the past phase history of the FSK signal.

■ BANDWIDTH EFFICIENCY OF M-ARY DIGITAL MODULATION TECHNIQUES

In Table 6.9, we have summarized typical values of power-bandwidth requirements for coherent binary and M -ary PSK schemes, assuming an average probability of symbol error equal to 10^{-4} and the systems operating in identical noise environments. This table shows that, among the family of M -ary PSK signals, QPSK (corresponding to $M = 4$) offers the best trade-off between power and bandwidth requirements. For this reason, we find that QPSK is widely used in practice. For $M > 8$, power requirements become excessive; accordingly, M -ary PSK schemes with $M > 8$ are not as widely used in practice. Also, coherent M -ary PSK schemes require considerably more complex equipment than coherent binary PSK schemes for signal generation or detection, especially when $M > 8$. (Coherent 8-PSK is used in digital satellite communications.)

Basically, M -ary PSK and M -ary QAM have similar spectral and bandwidth characteristics. For $M > 4$, however, the two schemes have different signal constellations. For M -ary PSK the signal constellation is circular, whereas for M -ary QAM it is rectangular. Moreover, a comparison of these two constellations reveals that the distance between the message points of M -ary PSK is smaller than the distance between the message points of M -ary QAM, for a fixed peak transmitted power. This basic difference between the two schemes is illustrated in Figure 6.46 for $M = 16$. Accordingly, in an AWGN channel, M -ary QAM outperforms the corresponding M -ary PSK in error performance for $M > 4$.

TABLE 6.9 Comparison of power-bandwidth requirements for M -ary PSK with binary PSK. Probability of symbol error = 10^{-4}

Value of M	$\frac{(\text{Bandwidth})_{M\text{-ary}}}{(\text{Bandwidth})_{\text{Binary}}}$	$\frac{(\text{Average power})_{M\text{-ary}}}{(\text{Average power})_{\text{Binary}}}$
4	0.5	0.34 dB
8	0.333	3.91 dB
16	0.25	8.52 dB
32	0.2	13.52 dB

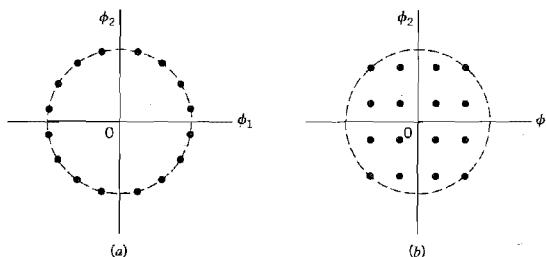


FIGURE 6.46 Signal constellations for (a) M -ary PSK and (b) corresponding M -ary QAM, for $M = 16$.

However, the superior performance of M -ary QAM can be realized only if the channel is free of nonlinearities.

As for M -ary FSK, we find that for a fixed probability of error, increasing M results in a reduced power requirement. However, this reduction in transmitted power is achieved at the cost of increased channel bandwidth. In other words, M -ary FSK behaves in an opposite manner to that of M -ary PSK. We will revisit this issue in an information-theoretical context in Chapter 9, and thereby develop further insight into the contrasting behaviors of M -ary PSK and M -ary FSK.

6.11 Voiceband Modems

The “modem,” a contraction of the term *modulator-demodulator*, is a conversion device that facilitates the transmission and reception of data over the *public switched telephone network* (PSTN).¹³ The data of interest may be digital signals generated by computers or service providers. In such an application, the modulator portion of the modem converts the incoming digital signal into a standard form suitable for transmission over a telephone channel in the PSTN. The demodulator portion of the modem receives the channel output and reconverts it into the original digital signal format. In yet another application, namely, *fax modems*, or more precisely modems with facsimile capability, the data may represent text, graphics, pictures, or combinations thereof. In this latter application, the document of interest is coded into a series of compressed picture elements (pixels), which are then transmitted over the telephone channel by modulating their values according to a predefined modulation standard. When the fax modem is in a receiving mode of operation, the demodulator portion of the modem operates on the received analog signal and decompresses the corresponding binary data representation of the demodulated signal into a near or actual duplicate of the original transmitted image. In what follows, we focus our attention on modems that provide communication between a *user* and an *Internet Service Provider* (ISP) over the PSTN.

Traditionally, the PSTN has been viewed as an analog network. In reality, however, the PSTN as we presently know it has become an almost entirely digital network. In most cases, the only part of the PSTN that has remained analog (and will likely remain so for many years to come) is the local loop, which represents the relatively short connection from a home to the central office. Thus, depending on how the PSTN is used, we may identify two distinct classes of modem configurations, symmetric and asymmetric, as described next.

■ 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 6.47a. (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 6.47a 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 6.47a 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 design 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 6.48a. Let $Q_{1,n}$, $Q_{2,n}$ denote the current value of the most significant

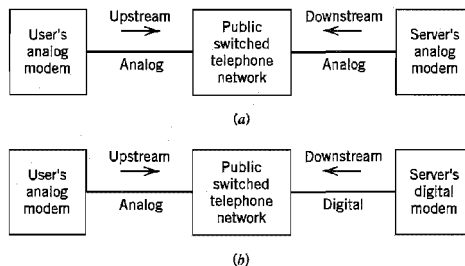


FIGURE 6.47 (a) Environmental overview of symmetric modem configuration: the upstream and downstream data rates are equal. (b) Environmental overview of “asymmetric” modem configuration: data rate downstream is higher than upstream.

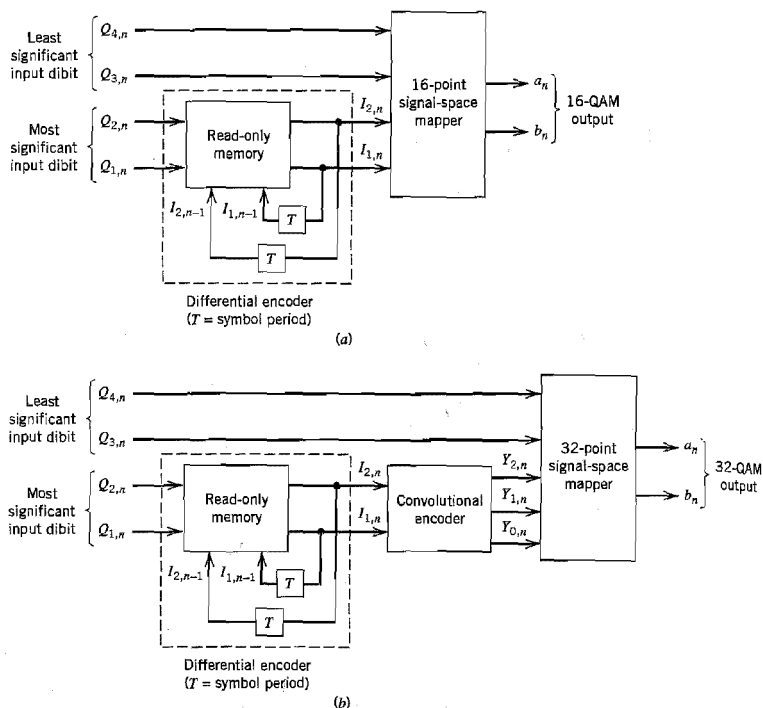


FIGURE 6.48 Block diagrams of V.32 modem. (a) Nonredundant coding. (b) Trellis coding.

input dibit, and let $I_{1,n-1}I_{2,n-1}$ denote the previous value of the corresponding dibit output by the encoder. Then, in response to the dibits $Q_{1,n}Q_{2,n}$ and $I_{1,n-1}I_{2,n-1}$, the differential encoder produces the dibit $I_{1,n}I_{2,n}$, which, in turn, induces a phase change in the modulated signal. This phase change, measured in the counterclockwise direction, is governed by the Gray coding scheme of Table 6.10. Note that the phase change is determined entirely by the input dibit $Q_{1,n}Q_{2,n}$. Insofar as the differential phase modulation is concerned, there is one other matter that needs to be addressed: a code for identifying the four quadrants of the two-dimensional signal space. This second matter is resolved by adopting the Gray coding scheme included in Figure 6.49.

Turning next to the amplitude modulation, a code has to be specified for the four possible values which the least significant input dibit, denoted by $Q_{3,n}Q_{4,n}$, can assume in, say, the first quadrant. This matter is taken care of by adopting the Gray code for the four signal points in the first quadrant shown lightly shaded in Figure 6.49.

The final issue that needs to be resolved is the 90° rotational invariance, which is mandated by the use of differential encoding. This form of invariance means that the overall M -ary QAM constellation looks exactly the same when it is rotated

TABLE 6.10 Phase changes induced by differential encoding in the V.32 modem due to varying input dibits

Current input dibit		Phase change (degrees)
$Q_{1,n}$	$Q_{2,n}$	
0	0	90
0	1	0
1	0	180
1	1	270

through an integer multiple of 90 degrees, regardless of whether it is coded or uncoded; then the receiver can correctly decode the transmitted message sequence when the local oscillator phase differs from the carrier phase by an integer multiple of 90 degrees. This final requirement is satisfied by filling in the Gray codes for the signal points in the remaining three quadrants in the manner shown in Figure 6.49. Dashed arrows are included in Figure 6.49 to illustrate the 90° rotational invariance.

Putting all of these matters together for the combined amplitude and phase modulation, we get the 16-QAM constellation shown previously in Figure 6.17a, which is reproduced here as Figure 6.50a. Correspondingly, the encoding system consists of a differential encoder followed by a 16-point signal-space mapper, as shown in Figure 6.48a. The V.32 modem so configured is said to be nonredundant because, with 16 constellation points, the transmitted 4-bit code word has *no* redundant bits.

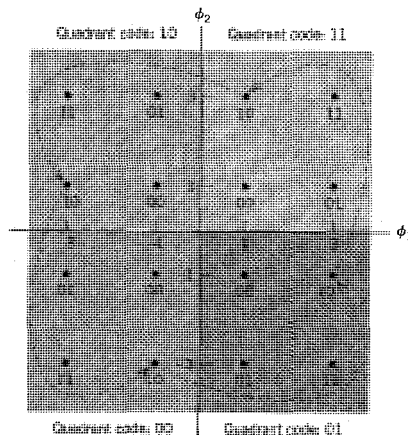


FIGURE 6.49 Illustrating the Gray encoding of the four quadrants and dibits in each quadrant for the V.32 modem. The dashed arrows illustrate the 90° rotational invariance.

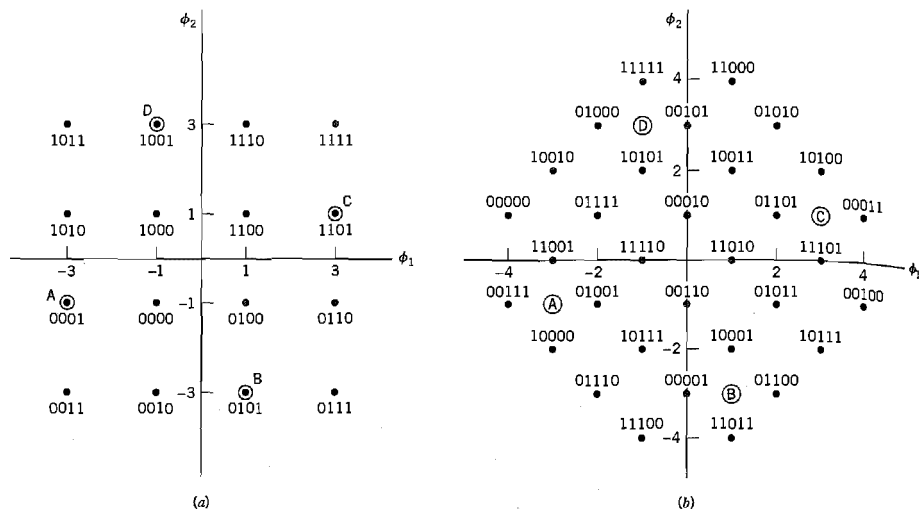


FIGURE 6.50 (a) Signal constellation of V.32 modem using nonredundant coding. (b) Signal constellation of V.32 modem using trellis coding.

As an illustrative example of how this particular V.32 modem operates, let the current group of four input bits be 1001 and the dibit previously output by the modem be 11. For this example, we thus have

$$\begin{aligned} Q_{1,n}Q_{2,n} &= 10 \\ Q_{3,n}Q_{4,n} &= 01 \\ I_{1,n-1}I_{2,n-1} &= 11 \end{aligned}$$

Then in light of the coding scheme for the four quadrants specified in Figure 6.49, the previous output dibit 11 means that the modulator was previously residing in the first quadrant. Because the corresponding input dibit is 10, it follows from Table 6.10 that the modulator experiences a phase change of 180° in the counterclockwise direction, thereby switching its operation into the third quadrant identified by the dibit 00. Finally, with the current value of the least significant dibit $Q_{3,n}Q_{4,n}$ being 01, the modulator outputs a QAM signal whose coordinates are $a_n = -3$ (along the ϕ_1 -axis) and $b_n = -1$ (along the ϕ_2 -axis). This output corresponds to the code word 0001.

When the signal-to-noise ratio is not high enough, the V.32 modem switches to its QPSK mode, operating at the reduced rate of 4,800 b/s. In this latter mode of operation, the four states of the modem are signified by the points labeled A, B, C, and D in Figure 6.50a.

Trellis Coding

Trellis coding is a forward-error correction scheme where coding and modulation are treated as a combined entity rather than as two separate operations. Figure 6.48b

shows the encoding system of the V.32 modem with trellis coding. The incoming data stream is divided into quadbits, but unlike the case of nonredundant coding, they are transmitted over the channel as a 32-QAM signal.

As indicated in Figure 6.48b, the trellis encoder involves the use of a *convolutional encoder*, which operates on the output of the differential encoder. (Convolutional encoders are discussed in Chapter 10.) However, the choice of convolutional encoding is restricted in the V.32 modem to accommodate the use of differential encoding (i.e., 90 degrees rotational invariance). Indeed, this requirement cannot be satisfied by a linear convolutional encoder. Rather, the convolutional encoder must be *nonlinear*;¹⁴ see Problem 10.30.

The data-encoding process in the V.32 modem with trellis coding proceeds in three stages:

1. The differential encoder in Figure 6.48b, in response to the current input dibit $Q_{1,n}Q_{2,n}$ and the previous differentially encoded dibit $I_{1,n-1}I_{2,n-1}$, produces the dibit $I_{1,n}I_{2,n}$.
2. The differentially encoded current dibit $I_{1,n}I_{2,n}$ is input to the convolutional encoder in Figure 6.48b, which produces a three-bit output. One of these bits is a *parity-check bit*, denoted by $Y_{0,n}$. The value of $Y_{0,n}$ depends on the other two bits, $Y_{1,n}$ and $Y_{2,n}$, produced by the convolutional encoder.
3. The bits $Y_{0,n}$, $Y_{1,n}$ and $Y_{2,n}$ produced by the convolutional encoder, together with the least significant input dibit $Q_{3,n}Q_{4,n}$ are applied to the signal-space mapper in Figure 6.48b, which selects one of the states in the 32-point constellation shown in Figure 6.50b as the modem output.

The parity-check bit $Y_{0,n}$ provides a modem with trellis coding better immunity to channel impairments than a V.32 modem with nonredundant coding, an advantage that is gained without an increase in bandwidth requirements. In quantitative terms, trellis coding provides an effective coding gain of 4 dB compared to 16-QAM. *Coding gain* expresses how much more signal energy per data bit is needed by the uncoded modem for the same level of noise performance.

However, for this advantage of trellis coding to be realized in practice, the signal-to-noise ratio must be high enough. Otherwise, the V.32 modem is switched to its QPSK mode of operation, which is signified by the four states labeled A, B, C, and D in Figure 6.50b. In this latter mode of operation, the data rate of the modem is reduced to 4,800 b/s.

■ ASYMMETRIC MODEM CONFIGURATIONS

For a more efficient use of the PSTN, we should treat it as what it really is: an *almost entirely digital network that is nonlinear*. In particular, since the ISP is digitally implemented, the need for analog-to-digital conversion at the ISP modem is eliminated. This means that the communication between the ISP and the PSTN can be entirely digital, as portrayed in Figure 6.47b. However, the user's modem has to remain analog because the local loop is analog. This, in turn, requires the use of analog-to-digital and digital-to-analog conversions each time the user's modem sends signals to and receives signals from the PSTN. The modem configuration depicted in Figure 6.47b is "asymmetric" in that it is possible for the downstream signaling data rate to be much higher than the upstream signaling data rate, as explained next.

As mentioned earlier, a digital PSTN is based on the use of PCM for the transmission of voice signals. Features of the system relevant to the present discussion are as follows (see Chapter 3):

- ▶ Data signaling rate of 64 kb/s, which is made up of a sampling rate of 8 kHz and the representation of each voice sample by an 8-bit code word.
- ▶ Fifteen-segment companding law (e.g., a logarithmic μ -law with $\mu = 255$) for compressing the voice signal at the transmitter and expanding it at the receiver.

From the discussion on PCM presented in Chapter 3 we also recall that quantization only affects analog-to-digital conversion but *not* digital-to-analog conversion. These observations have a profound impact on the optimum strategy for the design of asymmetric modems.

Suppose there is no analog-to-digital conversion between a digital modem at the ISP and the digital portion of the PSTN, and the digitally connected transmitter of the modem is designed to properly use the nonuniformly spaced 256 (discrete) threshold levels of the digital PSTN. Then, since digital-to-analog conversion is completely unaffected by quantization noise, it follows that the information transmitted by the ISP's digital modem reaches the user's analog modem with no loss whatsoever. On the basis of these arguments, in theory, it should be possible to transmit data from the ISP to the user at a rate equal to the 64 kb/s data rate of the digital PSTN. But system limitations inherent to the PSTN reduce the attainable data rate down to 56 kb/s, as explained in the sequel.

Digital Modem

From the description of a PCM voiceband channel presented in Chapter 3, we find that the design of the digital modem is constrained by three factors *not* under our control. The design constraints are:

1. A sampling rate $f_s = 8$ kHz.
2. A set of $M = 256$ allowable threshold levels built into the construction of the compressor (i.e., transmitter portion of the compander).
3. A baseband (antialiasing) filter of about 3.5 kHz bandwidth, built into the front end of the PCM transmitter.

In light of these constraints, we may now state the fundamental philosophy underlying the design of the digital modem as follows:

Design a signal $s(t)$ at the digital modem's input such that each of its samples taken at the rate $f_s = 8$ kHz matches one of the $M = 256$ threshold levels of the compressor, and the transmitted signal satisfies Nyquist's criterion for zero intersymbol interference.

(Nyquist's criterion for zero intersymbol interference was discussed in Chapter 4.)

One Realization of the Digital Modem

A solution to this signal design problem is made particularly difficult by the fact that the PCM transmit filter has a bandwidth of about 3.5 kHz and not 4 kHz (half the sampling rate f_s). The immediate implication of this constraint is that instead of the desired set of 8,000 samples, we can only generate $2 \times 3,500 = 7,000$ *independent* samples every second in accordance with Nyquist's criterion for zero intersymbol interference. How then do we

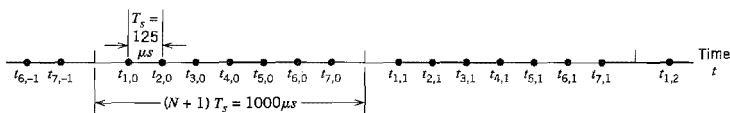


FIGURE 6.51 Group of N uniformly spaced samples, repeating every $(N + 1)T_s$ seconds.

fit 7,000 independent samples per second within the prescribed framework of 8,000 samples per second?

To answer this fundamental question, we make use of the *recurrent nonuniform equivalent* form of the sampling theorem. To be more specific, consider the situation depicted in Figure 6.51, where the samples are divided into groups, with each group containing N uniformly spaced samples, and the groups having a recurrent period of $(N + 1)T_s$ seconds, where $T_s = 1/f_s$. The illustration presented in Figure 6.51 is for the problem at hand: $T_s = 125 \mu\text{s}$ and $N = 7$. The sampling instants in the nonuniform distribution of Figure 6.51 are written as

$$\begin{aligned} t_{k,l} &= t_k + (N + 1)lT_s \\ &= (k - 1)T_s + (N + 1)lT_s, \quad \begin{matrix} k = 1, 2, \dots, N \\ l = 0, \pm 1, \pm 2, \dots \end{matrix} \end{aligned} \quad (6.187)$$

The stage is now set for us to define the band-limited signal $s(t)$ as follows:¹⁵

$$s(t) = \sum_{l=-\infty}^{\infty} \sum_{k=1}^N s(t_{k,l}) \psi_k(t - (N + 1)lT_s) \quad (6.188)$$

where the *interpolation function* $\psi_k(t)$ is itself defined by

$$\psi_k(t) = \text{sinc}\left(\frac{t - t_k}{(N + 1)T_s}\right) \prod_{\substack{q=1 \\ q \neq k}}^N \frac{\sin\left(\frac{\pi}{(N + 1)T_s}(t - t_q)\right)}{\sin\left(\frac{\pi}{(N + 1)T_s}(t_k - t_q)\right)} \quad (6.189)$$

Computing Equation (6.189) for $N = 7$, we obtain the seven *standard pulses* plotted in Figure 6.52, where time is normalized with respect to the sampling period T_s . These pulses exhibit the following properties:

- Each standard pulse is normalized so that we have

$$\psi_k\left(\frac{t_k}{T_s}\right) = \psi_k(k - 1) = 1 \quad \text{for } k = 1, 2, \dots, 7$$

Note, however, that the peak of the k th pulse does *not* occur at time $t_k = (k - 1)T_s$.

- For $k = 1, 2, \dots, 7$ the pulse $\psi_k(t/T_s)$ goes through zero at times $t \neq (k - 1)T_s$ modulo $(N + 1)$, except at those times that are congruent to $t = (-1) \bmod (N + 1)$.

Accordingly, the signaling scheme for the digital modem consists of a recurrent nonuniform pulse amplitude modulation scheme. The amplitudes of seven uniformly spaced samples in each group of eight samples are determined by the incoming data stream and in conformity to the threshold levels of the compressor in the PCM transmitter. In effect, these seven samples are the independent samples that are responsible for carrying the

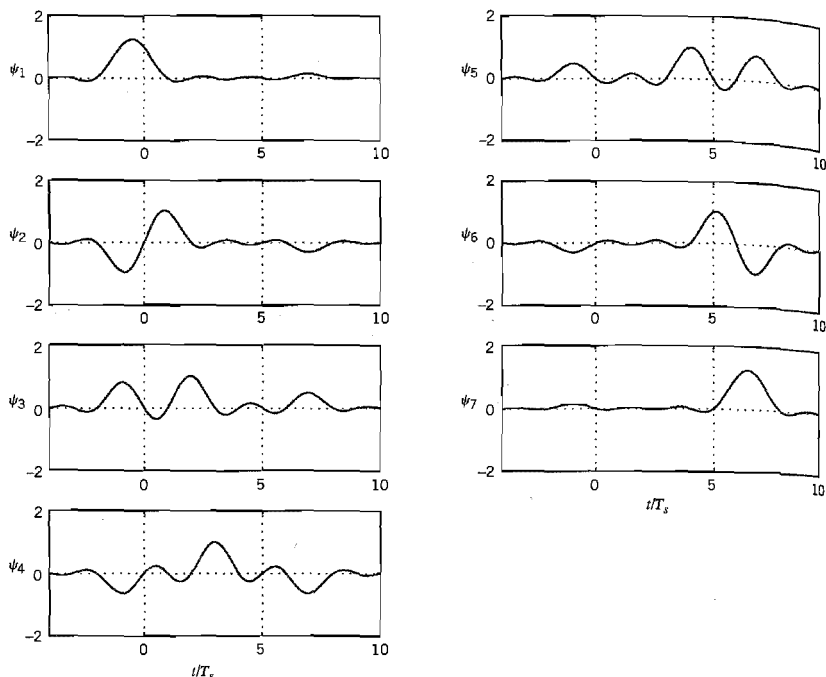


FIGURE 6.52 A digital modem's waveforms of the standard pulses $\psi_k(t)$, $k = 1, 2, \dots, 7$.

incoming data stream across the PSTN every $1,000 \mu\text{s}$. Moreover, they deliver the data to the receiver with *zero* intersymbol interference. The remaining “eighth” samples are completely determined by the independent samples and known beforehand to the system; they do not carry information and are therefore discarded at the receiver. Thus the digital modem is capable of transmitting digital data across the PSTN almost errorless at a rate equal to 56 kb/s, which is calculated as follows:

$$7 \times 1,000 \times \log_2 256 = 56,000 \text{ b/s}$$

One last comment is in order. The standard pulses $\psi_k(t)$ can be constructed so as to decay at a rate faster than $1/t$. To do so, we simply replace the sinc function in Equation (6.189) by a Nyquist pulse with a rolloff in a manner similar to that described in Chapter 4.

Another Realization of the Digital Modem

The kind of digital modem just described is *bidirectional*, assuming that both ends of the data link are analog. However, a simpler solution to the digital modem design problem ensues when one end of the link is digital and asymmetric data rates are possible.¹⁶

Consider what happens when a data sequence consisting of *octets* (i.e., 8-bit code words) arrives at the PSTN. There they will be treated as octets representing speech encoded in accordance with the μ -law or A-law, depending on the part of the world where

the PSTN is located. Consequently, the D/A converter, which drives the analog modem, produces a continuous-time signal defined by

$$s(t) = \sum_k a(c_k)g(t - kT_s) \quad (6.190)$$

where c_k is the k th octet in the data sequence, $a(c_k)$ is the representation level specified by the pertinent companding law, T_s is the sampling interval (equal to $125 \mu\text{s}$), and $g(t)$ is an interpolation function bandlimited to a frequency below $1/2T_s$, or about 4 kHz, to satisfy the reconstruction part of the sampling theorem; see Section 3.2.

In the normal operation of the PSTN, the signal $s(t)$ represents a reconstructed speech signal. However, in the case of input data, $s(t)$ appears like noise. In any event, from a communication theoretical perspective, the signal $s(t)$ in Equation (6.190) may be viewed as a *pulse-amplitude modulated* signal. Herein lies the theoretical basis for the design of the digital modem. Specifically, the design is based on a signal constellation as in an analog modem, except that the constellation is constructed from *one-dimensional PCM symbols* rather than two-dimensional QAM symbols.

Ordinarily, the data rate achievable by a digital modem is limited to about 56 kb/s because of the following factors:

1. The inner levels of the compander in the PSTN are very closely spaced, as shown in Table 3.4; hence they are susceptible to residual intersymbol interference and noise following the modem's equalizer.
2. Least significant bits (LSBs) are robbed from the data stream for various purposes internal to the PSTN; this "bit-robbing" can be as much as (but usually less than) 8 kb/s and always in a periodic pattern.

Analog Modem

Unlike the digital modem, the noise performance of the analog modem is limited essentially by quantization noise in the μ -law or A-law governing the operation of the PCM compander. Typically, the signal-to-noise ratio on a good PCM voiceband channel is on the order of 34 to 38 dB. The other channel impairment that limits the operation of the analog modem is the effect of bandlimiting imposed by the antialiasing and interpolation filters, which, as already mentioned, is typically about 3.5 kHz.

A sophisticated choice for the analog modem is the standard V.34 *modem*, which operates at rates extending up to 33.6 kb/s. The fundamental design philosophy of this modem embodies five distinctive features.¹⁷

1. 960-QAM super-constellation.

The signal constellation is said to be a super- or nested-constellation in that it consists of four constellations: the QAM constellation shown in Figure 6.53 with 240 message points, and its rotated versions through 90, 180, and 270 degrees.

2. Adaptive bandwidth.

The transmitter probes the channel by sending a set of tones, which permits measurement of the signal-to-noise ratio at the channel output as a function of frequency. The modem is thereby enabled to select the appropriate carrier frequency and bandwidth according to the probing results and available symbol rates.

3. Adaptive bit rates.

During the training of the receiver, the bit rate is selected according to the receiver's estimate of the maximum bit rate, which the modem can support at bit error rates as low as 10^{-6} to 10^{-5} .

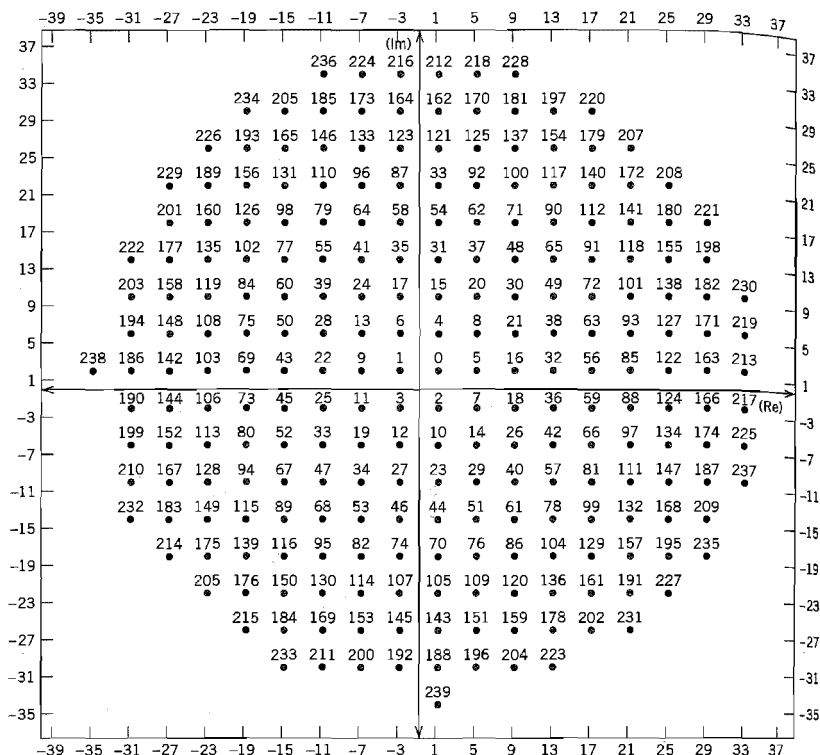


FIGURE 6.53 Quarter-superconstellation of V.34 modem with 240 signal points. The full superconstellation is obtained by combining the rotated versions of these points by 0, 90, 180, and 270 degrees. (Taken from Forney et al., 1996, with permission of the IEEE.)

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. To 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.)

V.90 Modem

The V.90 modem standard embodies digital and analog modems. The digital modem at the ISP end is based on the second realization described earlier; it sends data downstream at the rate of 56 kb/s. The analog modem at the user's end is a V.34 modem standard, transmitting data upstream at the rate of 33.6 kb/s. These two highly different rates confirm the asymmetric nature of the V.90 modem.

The outstanding feature of the V.90 modem, namely, the downstream data rate of 56 kb/s, makes it suitable for use on the Internet for downloading graphics in intensive Web pages, audio, and video at near-ISDN speeds.

6.12 Multichannel Modulation

The *asymmetric digital subscriber line (ADSL)*, described in Section 4.8, is a data transmission system capable of realizing megabit rates over existing twisted-pair telephone lines. Specifically, ADSL runs at a downstream data rate up to 9 Mb/s and an upstream data rate up to 1 Mb/s. These data signaling rates fit the access requirements of the Internet perfectly. (As mentioned in Section 4.8, the upstream bit rate should be about 10 percent of the downstream bit rate for efficient operation of the Internet protocol.) The challenge in designing ADSL is to develop a line code that exploits the information capacity of the channel as fully as possible. The carrierless amplitude phase modulation (CAP), discussed in Section 6.4, provides one approach for solving this difficult passband data transmission problem. Another approach is to use an equally elegant modulation technique called discrete multitone. This latter approach is a form of *multichannel modulation*¹⁸ that allows the modulator characteristics to be a function of measured channel characteristics. It is fitting that we begin the discussion by describing multichannel modulation, which we do in this section, followed by discrete multitone in the next section.

The basic idea of *multichannel modulation* is rooted in a commonly used engineering principle: *divide and conquer*. According to this principle, a difficult problem is solved by dividing it into a number of simpler problems, and then combining the solutions to those simple problems. In the context of our present discussion, the difficult problem is that of data transmission over a wideband channel with severe intersymbol interference, and the simpler problems are exemplified by data transmission over AWGN channels. We may thus summarize the essence of multichannel modulation as follows:

Data transmission over a difficult channel is transformed through the use of advanced signal processing techniques into the parallel transmission of the given data stream over a large number of subchannels, such that each subchannel may be viewed effectively as an AWGN channel.

Naturally, the overall data rate is the sum of the individual data rates over the subchannels operating in parallel.

■ CAPACITY OF AWGN CHANNEL

From the Background and Preview material presented in the opening chapter, we recall that, according to *Shannon's information capacity theorem*, the capacity of an AWGN channel (that is free from intersymbol interference) is defined by

$$C = B \log_2(1 + \text{SNR}) \text{ b/s} \quad (6.191)$$

where B is the channel bandwidth, and SNR denotes the signal-to-noise ratio measured at the channel output. A proof of this important theorem is formally presented in Chapter 9. For now it suffices to say that for a given SNR, we can transmit data over an AWGN channel of bandwidth B at the maximum rate of C bits per second with arbitrarily small probability of error, provided that we employ an encoding system of sufficiently high complexity. Equivalently, we may express the capacity C in bits per transmission or channel use as

$$C = \frac{1}{2} \log_2(1 + \text{SNR}) \quad \text{bits/transmission} \quad (6.192)$$

In practice, we usually find that a physically realizable encoding system must transmit data at a rate R less than the maximum possible rate C for it to be reliable. For an implementable system operating at low enough probability of symbol error, we thus need to introduce a *signal-to-noise ratio gap* or just *gap*, denoted by Γ . The gap is a function of the permissible probability of symbol error P_e and the encoding system of interest. It provides a measure of the “efficiency” of an encoding system with respect to the ideal transmission system of Equation (6.192). With C denoting the capacity of the ideal encoding system and R denoting the capacity of the corresponding implementable encoding system, the gap is defined by

$$\begin{aligned} \Gamma &= \frac{2^{2C} - 1}{2^{2R} - 1} \\ &= \frac{\text{SNR}}{2^{2R} - 1} \end{aligned} \quad (6.193)$$

Equivalently, we may write

$$R = \frac{1}{2} \log_2 \left(1 + \frac{\text{SNR}}{\Gamma} \right) \quad \text{bits/transmission} \quad (6.194)$$

For encoded PAM or QAM operating at $P_e = 10^{-6}$, for example, the gap Γ is constant at 8.8 dB. Through the use of codes (e.g., trellis codes discussed in Chapter 10), the gap Γ may be reduced to as low as 1 dB.

Let P denote the transmitted signal power, and σ^2 denote the channel noise variance measured over the bandwidth B . The signal-to-noise ratio is therefore

$$\text{SNR} = \frac{P}{\sigma^2}$$

where

$$\sigma^2 = N_0 B$$

We may thus finally define the attainable data rate as

$$R = \frac{1}{2} \log_2 \left(1 + \frac{P}{\Gamma \sigma^2} \right) \quad \text{bits/transmission} \quad (6.195)$$

With this formula at hand, we are ready to describe multichannel modulation in quantitative terms.

■ CONTINUOUS-TIME CHANNEL PARTITIONING

Consider a linear wideband channel (e.g., twisted pair) with an arbitrary frequency response $H(f)$. Let the squared magnitude response $|H(f)|$ be approximated by a staircase

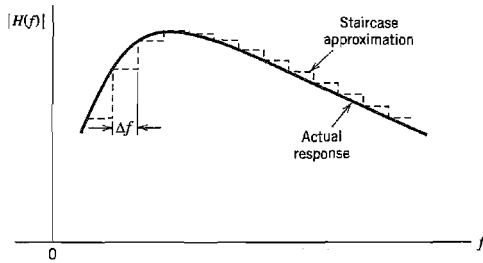


FIGURE 6.54 Staircase approximation of an arbitrary magnitude response $|H(f)|$; only positive-frequency portion of the response is shown.

function as illustrated in Figure 6.54, with Δf denoting the width of each step. In the limit, as the frequency increment Δf approaches zero, the staircase approximation of the channel approaches the actual $H(f)$. Along each step of the approximation, the channel may be assumed to operate as an AWGN channel free from intersymbol interference. The problem of transmitting a single wideband signal is thereby transformed into the transmission of a set of narrowband orthogonal signals. Each narrowband orthogonal signal, with its own carrier, is generated using a spectrally efficient modulation technique such as M -ary QAM, with additive white Gaussian noise being essentially the only primary source of transmission impairment. This, in turn, means that data transmission over each subchannel of bandwidth Δf can be optimized by invoking Shannon's information capacity theorem, with the optimization of each subchannel being performed independently of all the others. Thus, in practical signal-processing terms, *the need for complicated equalization of a wideband channel is replaced by the need for multiplexing and demultiplexing the transmission of the incoming data stream over a large number of narrowband subchannels that are contiguous and disjoint*. Although the resulting complexity of a multicarrier system is indeed high for a large number of subchannels, implementation of the entire system can be accomplished in a cost-effective manner through the use of VLSI technology.

Figure 6.55 shows a block diagram of the multichannel data transmission system in its most basic form. The system is configured here using quadrature-amplitude modulation whose choice is justified by virtue of its spectral efficiency. The incoming binary data stream is first applied to a demultiplexer (not shown in the figure), thereby producing a set of N substreams. Each substream represents a sequence of *two-element subsymbols*, which, for the symbol interval $0 \leq t \leq T$, is denoted by

$$(a_n, b_n), \quad n = 1, 2, \dots, N$$

where a_n and b_n are element values along the two coordinates of subchannel n .

Correspondingly, the passband basis functions of the quadrature-amplitude modulators are defined by the function pairs

$$\{\phi(t) \cos(2\pi f_n t), \phi(t) \sin(2\pi f_n t)\}, \quad n = 1, 2, \dots, N \quad (6.196)$$

where the carrier frequency f_n of the n th modulator is an integer multiple of the symbol rate $1/T$, as shown by

$$f_n = \frac{n}{T}, \quad n = 1, 2, \dots, N$$

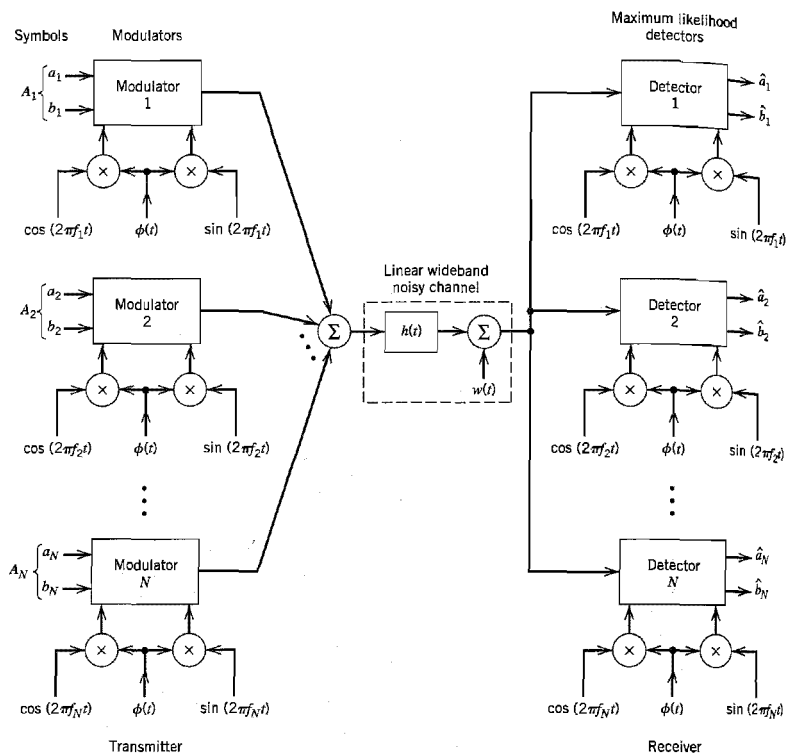


FIGURE 6.55 Block diagram of multichannel data transmission system.

and the low-pass function $\phi(t)$ is the sinc function:

$$\phi(t) = \sqrt{\frac{2}{T}} \operatorname{sinc}\left(\frac{t}{T}\right), \quad -\infty < t < \infty \quad (6.197)$$

The passband basis functions defined here have the following desirable properties (see Problem 6.41 for their proofs):

Property 1

For each n , the two quadrature-modulated sinc functions form an orthogonal pair as shown by

$$\int_{-\infty}^{\infty} (\phi(t) \cos(2\pi f_n t)) (\phi(t) \sin(2\pi f_n t)) dt = 0 \quad \text{for all } n \quad (6.198)$$

This orthogonal relationship provides the basis for formulating the signal constellation for each of the N modulators in the form of a squared lattice.

Property 2

Recognizing that

$$\exp(j2\pi f_n t) = \cos(2\pi f_n t) + j \sin(2\pi f_n t)$$

we may completely redefine the passband basis functions in the complex form

$$\left\{ \frac{1}{\sqrt{2}} \phi(t) \exp(j2\pi f_n t) \right\}, \quad n = 1, 2, \dots, N \quad (6.199)$$

where the factor $1/\sqrt{2}$ has been introduced to ensure that the scaled function $\phi(t)/\sqrt{2}$ has unit energy. Hence, these passband basis functions form an orthonormal set, as shown by

$$\int_{-\infty}^{\infty} \left(\frac{1}{\sqrt{2}} \phi(t) \exp(j2\pi f_n t) \right) \left(\frac{1}{\sqrt{2}} \phi(t) \exp(j2\pi f_k t) \right)^* dt = \begin{cases} 1, & k = n \\ 0, & k \neq n \end{cases} \quad (6.200)$$

where the asterisk denotes complex conjugation.

Equation (6.200) provides the mathematical basis for ensuring that the N modulator-demodulator pairs operate independently of each other.

Property 3

The set of channel-output functions $\{h(t) \star \phi(t)\}$ remains orthogonal for a linear channel with arbitrary impulse response $h(t)$, where \star denotes convolution.

The channel is thus partitioned into a set of independent subchannels operating in continuous time.

Figure 6.55 also includes the structure of the receiver. It consists of a bank of N coherent detectors, with the channel output being simultaneously applied to the detector inputs. Each detector is supplied with a locally generated pair of quadrature modulated sinc functions operating in synchrony with the pair of passband basis function applied to the corresponding modulator in the transmitter.

Each subchannel may have some residual intersymbol interference (ISI). However, as the number of subchannels N approaches infinity, the ISI disappears. Thus, for a sufficiently large N , the bank of coherent detectors in Figure 6.55 operates as *maximum likelihood detectors*, independently of each other and on a subsymbol-by-subsymbol basis.

To define the detector outputs in response to the input subsymbols, we find it convenient to use complex notation. Let A_n denote the subsymbol applied to the n th modulator during the symbol interval $0 \leq t \leq T$:

$$A_n = a_n + jb_n, \quad n = 1, 2, \dots, N \quad (6.201)$$

The corresponding detector output is

$$Y_n = H_n A_n + W_n, \quad n = 1, 2, \dots, N \quad (6.202)$$

where H_n is the complex-valued frequency response of the channel evaluated at the subchannel carrier frequency $f = f_n$:

$$H_n = H(f_n), \quad n = 1, 2, \dots, N \quad (6.203)$$

The W_n is a complex-valued random variable due to the channel noise $w(t)$; the real and imaginary parts of W_n have zero mean and variance $N_0/2$. With knowledge of the mea-

sured frequency response $H(f)$ available, we may therefore use Equation (6.202) to compute a maximum likelihood estimate of the transmitted subsymbol A_n . The estimates $\hat{A}_1, \hat{A}_2, \dots, \hat{A}_N$ so obtained are finally multiplexed to produce the corresponding estimate of the original binary data transmitted during the interval $0 \leq t \leq T$.

To summarize, for a sufficiently large N , we may implement the receiver as an optimum maximum likelihood detector, operating as N subsymbol-by-subsymbol detectors. The reason why it is possible to build a maximum likelihood receiver in such a simple way is the fact that the passband basis functions constitute an orthonormal set, and their orthogonality is maintained for any channel impulse response $h(t)$.

■ GEOMETRIC SIGNAL-TO-NOISE RATIO

In the multichannel transmission system of Figure 6.55, each subchannel is characterized by a SNR of its own. It would be highly desirable to derive a single measure for the performance of the entire system of Figure 6.55.

To simplify the derivation of such a measure, we assume that all of the subchannels in Figure 6.55 are represented by one-dimensional constellations. Then the channel capacity of the entire system in bits per transmission is given by

$$\begin{aligned} R &= \frac{1}{N} \sum_{n=1}^N R_n \\ &= \frac{1}{2N} \sum_{n=1}^N \log_2 \left(1 + \frac{P_n}{\Gamma \sigma_n^2} \right) \\ &= \frac{1}{2N} \log_2 \prod_{n=1}^N \left(1 + \frac{P_n}{\Gamma \sigma_n^2} \right) \\ &= \frac{1}{2} \log_2 \left[\prod_{n=1}^N \left(1 + \frac{P_n}{\Gamma \sigma_n^2} \right) \right]^{1/N} \end{aligned} \quad (6.204)$$

Let $(\text{SNR})_{\text{overall}}$ denote the overall signal-to-noise ratio of the entire system. We may then express R in bits per transmission as

$$R = \frac{1}{2} \log_2 \left(1 + \frac{(\text{SNR})_{\text{overall}}}{\Gamma} \right) \quad (6.205)$$

Comparing Equations (6.205) with (6.204), we may thus write

$$(\text{SNR})_{\text{overall}} = \Gamma \left(\prod_{n=1}^N \left(1 + \frac{P_n}{\Gamma \sigma_n^2} \right)^{1/N} - 1 \right) \quad (6.206)$$

Assuming that $P_n/\Gamma \sigma_n^2$ is high enough to ignore the two unity terms in Equation (6.206), we may approximate the overall SNR as

$$(\text{SNR}) = \prod_{n=1}^N \left(\frac{P_n}{\sigma_n^2} \right)^{1/N} \quad (6.207)$$

We may thus characterize the overall system by a SNR that is the *geometric mean* of the SNRs of the individual subchannels.

The geometric SNR of Equation (6.207) can be improved considerably by distributing the available transmit power among the N subchannels on a nonuniform basis. This objective is attained through the use of loading as discussed next.

■ LOADING OF THE MULTICHANNEL TRANSMISSION SYSTEM

Equation (6.204) for the bit rate of the entire multichannel system ignores the effect of the channel on system performance. To account for this effect, define

$$g_n = |H(f_n)|, \quad n = 1, 2, \dots, N \quad (6.208)$$

Then assuming that the number of subchannels N is large enough, we may assume that g_n is constant over the entire bandwidth Δf assigned to subchannel n for all n . In such a case, we may modify the second line of Equation (6.204) for the overall SNR of the system as

$$R = \frac{1}{2N} \sum_{n=1}^N \log_2 \left(1 + \frac{g_n^2 P_n}{\Gamma \sigma_n^2} \right) \quad (6.209)$$

The g_n^2 and Γ are usually fixed. The noise variance $\sigma_n^2 = \Delta f N_0$ for all n , where Δf is the bandwidth of each subchannel and $N_0/2$ is the noise power spectral density. We may therefore optimize the overall bit rate R through a proper allocation of the total transmit power among the various channels. However, for this optimization to be of practical value, we must maintain the total transmit power at some constant value P , say, as shown by

$$\sum_{n=1}^N P_n = P = \text{constant} \quad (6.210)$$

The optimization we therefore have to deal with is a *constrained optimization problem*, which may be stated as follows:

Maximize the bit rate R for the entire multichannel transmission system through an optimal sharing of the total transmit power P between the N subchannels, subject to the constraint that P is maintained constant.

To solve this optimization problem, we first use the *method of Lagrange multipliers*¹⁹ to set up an objective function that incorporates the constraint of Equation (6.210), as shown by

$$\begin{aligned} J &= \frac{1}{2N} \sum_{n=1}^N \log_2 \left(1 + \frac{g_n^2 P_n}{\Gamma \sigma_n^2} \right) + \lambda \left(P - \sum_{n=1}^N P_n \right) \\ &= \frac{1}{2N} \log_2 e \sum_{n=1}^N \log_e \left(1 + \frac{g_n^2 P_n}{\Gamma \sigma_n^2} \right) + \lambda \left(P - \sum_{n=1}^N P_n \right) \end{aligned} \quad (6.211)$$

where λ is the *Lagrange multiplier*. Hence, differentiating J with respect to P_n , then setting the result equal to zero and finally rearranging terms, we get

$$\frac{\frac{1}{2N} \log_2 e}{P_n + \frac{\Gamma \sigma_n^2}{g_n^2}} = \lambda \quad (6.212)$$

This result indicates that the solution to our constrained optimization problem is to have

$$P_n + \frac{\Gamma \sigma_n^2}{g_n^2} = K \quad \text{for } n = 1, 2, \dots, N \quad (6.213)$$

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 defined 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 sum 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

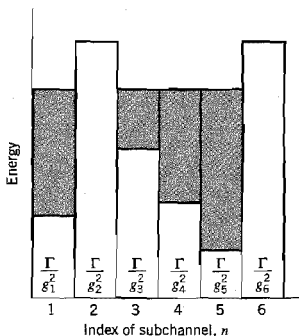


FIGURE 6.56 Water-filling interpretation of the loading problem.

of the entire system, we may proceed as follows. Let the total transmit power be fixed at the constant value P as in Equation (6.210). Let K denote the constant value prescribed for the sum $P_n + \Gamma \sigma_n^2 / g_n^2$ for all n as in Equation (6.213). We may then use this pair of equations to set up the following system of simultaneous equations:

$$\begin{aligned} P_1 + P_2 + \cdots P_N &= P \\ P_1 - K &= -\Gamma \sigma^2 / g_1^2 \\ P_2 - K &= -\Gamma \sigma^2 / g_2^2 \\ &\vdots \\ P_N - K &= -\Gamma \sigma^2 / g_N^2 \end{aligned} \quad (6.214)$$

where we have a total of $(N + 1)$ unknowns and $(N + 1)$ equations to solve for them. We may rewrite this set of simultaneous equations in matrix form as

$$\begin{bmatrix} 1 & 1 & \cdots & 1 & 0 \\ 1 & 0 & \cdots & 0 & -1 \\ 0 & 1 & \cdots & 0 & -1 \\ \vdots & \vdots & & \vdots & \vdots \\ 0 & 0 & \cdots & 1 & -1 \end{bmatrix} \begin{bmatrix} P_1 \\ P_2 \\ \vdots \\ P_N \\ K \end{bmatrix} = \begin{bmatrix} P \\ -\Gamma \sigma^2 / g_1^2 \\ -\Gamma \sigma^2 / g_2^2 \\ \vdots \\ -\Gamma \sigma^2 / g_N^2 \end{bmatrix} \quad (6.215)$$

Premultiplying both sides of Equation (6.215) by the inverse of the $(N + 1)$ -by- $(N + 1)$ matrix on the left-hand side of the equation, we obtain solutions for the unknowns P_1, P_2, \dots, P_N , and K . We should always find that K is positive, but it is possible for some of the P s to be negative. The negative P s are discarded as power cannot be negative.

► EXAMPLE 6.7

Consider a linear channel whose squared magnitude response $|H(f)|^2$ has the piecewise-linear form shown in Figure 6.57. To simplify the example, we set the gap $\Gamma = 1$ and the noise variance $\sigma^2 = 1$. In the situation so described, the application of Equation (6.214) yields

$$\begin{aligned} P_1 + P_2 &= P \\ P_1 - K &= -1 \\ P_2 - K &= -1/l \end{aligned}$$

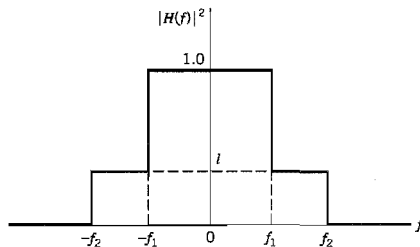


FIGURE 6.57 Squared magnitude response for Example 6.7.

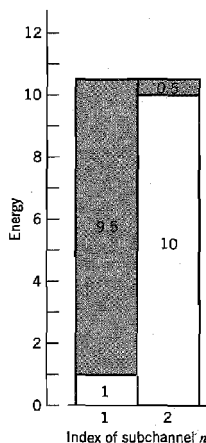


FIGURE 6.58 Water-filling profile for Example 6.7.

where the total transmit power P is normalized with respect to the noise variance. Solving these three simultaneous equations for P_1 , P_2 , and K , we get

$$\begin{aligned} P_1 &= \frac{1}{2} \left(P - 1 + \frac{1}{l} \right) \\ P_2 &= \frac{1}{2} \left(P + 1 - \frac{1}{l} \right) \\ K &= \frac{1}{2} \left(P + 1 + \frac{1}{l} \right) \end{aligned}$$

Since $0 < l < 1$, it follows that $P_1 > 0$, but it is possible for P_2 to be negative. This latter condition can arise if

$$l < \frac{1}{P + 1}$$

But then P_1 exceeds the prescribed value of transmit power P . It follows therefore that in this example the only acceptable solution is to have $1/(P + 1) < l < 1$. Suppose then we have $P = 10$ and $l = 0.1$, for which the solution is

$$\begin{aligned} K &= 10.5 \\ P_1 &= 9.5 \\ P_2 &= 0.5 \end{aligned}$$

The corresponding water-filling picture is portrayed in Figure 6.58.

6.13 Discrete Multitone

The material presented in Section 6.12 provides an insightful introduction to the notion of multichannel modulation. In particular, the continuous-time channel partitioning induced by the passband basis functions of Equation (6.196) or equivalently (6.199) exhibits

a highly desirable property: Orthogonality of the basis functions (and therefore the channel partitioning) is preserved despite their convolution with the impulse response of the channel. However, the system has two shortcomings:

1. The passband basis functions use a sinc function that is nonzero for an infinite time interval, whereas practical considerations favor a finite observation interval.
2. For a finite number of subchannels, N , the system is suboptimal; optimality of the system is assured only when N approaches infinity.

We may overcome these shortcomings by using *discrete multitone* (DMT), the basic idea of which is to transform a wideband channel into a set of N subchannels operating in parallel. What makes DMT distinctive is the fact that the transformation is performed in discrete time as well as discrete frequency. Consequently, the transmitter input-output behavior of the entire communication system admits a linear matrix representation, which lends itself to implementation using the discrete Fourier transform.

To explore this new approach, we first recognize that in a realistic situation the channel has its nonzero impulse response, $h(t)$, essentially confined to a finite interval $[0, T_b]$. So, let the sequence h_0, h_1, \dots, h_ν denote the baseband equivalent impulse response of the channel sampled at the rate $1/T_s$, with

$$T_b = (1 + \nu)T_s \quad (6.216)$$

The sampling rate $1/T_s$ is chosen to be greater than twice the higher frequency component of interest in accordance with the sampling theorem. To continue with the discrete-time description of the system, let $s[n] = s(nT_s)$ denote a sample of the transmitted symbol $s(t)$, $w[n] = w(nT_s)$ denote a sample of the channel noise $w(t)$, and $x[n] = x(nT_s)$ denote the corresponding sample of the channel output (received signal). The channel performs linear convolution on the incoming symbol sequence $\{s[n]\}$ of length N , producing a channel output sequence $\{x[n]\}$ of length $N + \nu$. Extension of the channel output sequence by ν samples compared to the channel input sequence is due to the intersymbol interference produced by the channel.

To overcome the effect of intersymbol interference, we create a cyclically extended *guard interval* whereby each symbol sequence is preceded by a periodic extension of the sequence itself. Specifically, the last ν samples of the symbol sequence are repeated at the beginning of the sequence being transmitted, as shown by

$$s[-k] = s[N - k] \quad \text{for } k = 1, 2, \dots, \nu \quad (6.217)$$

This condition is called a *cyclic prefix*. The *excess bandwidth factor* due to the inclusion of the cyclic prefix is therefore ν/N , where N is the number of transmitted samples after the guard interval.

With the cyclic prefix in place, the matrix description of the channel takes the form

$$\begin{bmatrix} x[N-1] \\ x[N-2] \\ \vdots \\ x[N-\nu-1] \\ x[N-\nu-2] \\ \vdots \\ x[0] \end{bmatrix} = \begin{bmatrix} h_0 & h_1 & h_2 & \cdots & h_{\nu-1} & h_\nu & 0 & \cdots & 0 \\ 0 & h_0 & h_1 & \cdots & h_{\nu-2} & h_{\nu-1} & h_\nu & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & 0 & h_0 & h_1 & \cdots & h_\nu \\ h_\nu & 0 & 0 & \cdots & 0 & 0 & h_0 & \cdots & h_{\nu-1} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ h_1 & h_2 & h_3 & \cdots & h_\nu & 0 & 0 & \cdots & h_0 \end{bmatrix} \begin{bmatrix} s[N-1] \\ s[N-2] \\ \vdots \\ s[N-\nu-1] \\ s[N-\nu-2] \\ \vdots \\ s[0] \end{bmatrix} + \begin{bmatrix} w[N-1] \\ w[N-2] \\ \vdots \\ w[N-\nu-1] \\ w[N-\nu-2] \\ \vdots \\ w[0] \end{bmatrix} \quad (6.218)$$

Equivalently, we may describe the discrete-time representation of the channel in the compact matrix form

$$\mathbf{x} = \mathbf{H}\mathbf{s} + \mathbf{w} \quad (6.219)$$

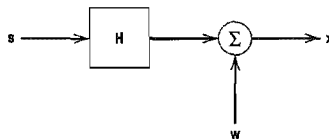


FIGURE 6.59 Discrete-time representation of multichannel data transmission system.

where the transmitted symbol vector \mathbf{s} , the channel noise vector \mathbf{w} , and the received signal vector \mathbf{x} are all N -by-1 vectors which are respectively defined by

$$\mathbf{s} = [s[N-1], s[N-2], \dots, s[0]]^T \quad (6.220)$$

$$\mathbf{w} = [w[N-1], w[N-2], \dots, w[0]]^T \quad (6.221)$$

and

$$\mathbf{x} = [x[N-1], x[N-2], \dots, x[0]]^T \quad (6.222)$$

We may thus depict the discrete-time representation of the channel as in Figure 6.59. The N -by- N channel matrix \mathbf{H} is defined by

$$\mathbf{H} = \begin{bmatrix} h_0 & h_1 & h_2 & \cdots & h_{v-1} & h_v & 0 & \cdots & 0 \\ 0 & h_0 & h_1 & \cdots & h_{v-2} & h_{v-1} & h_v & \cdots & 0 \\ \vdots & \vdots & \vdots & & \vdots & \vdots & \vdots & & \vdots \\ 0 & 0 & 0 & \cdots & 0 & h_0 & h_1 & \cdots & h_v \\ h_v & 0 & 0 & \cdots & 0 & 0 & h_0 & \cdots & h_{v-1} \\ \vdots & \vdots & \vdots & & \vdots & \vdots & \vdots & & \vdots \\ h_1 & h_2 & h_3 & \cdots & h_v & 0 & 0 & \cdots & h_0 \end{bmatrix} \quad (6.223)$$

From this definition, we readily see that the matrix \mathbf{H} has the following structural composition: Every row of the matrix is obtained by applying a right-shift to the previous row by one position, with the added proviso that the rightmost element of the previous row spills over in the shifting process to be “circulated” back to the leftmost element of the new row. Accordingly, the matrix \mathbf{H} is referred to as a *circulant matrix*.

Before proceeding further, it is befitting that we briefly review the discrete Fourier transform and its role in the spectral decomposition of the circulant matrix \mathbf{H} .

■ DISCRETE FOURIER TRANSFORM

Consider the N -by-1 vector \mathbf{x} of Equation (6.222). The *discrete Fourier transform* (DFT) of the vector \mathbf{x} is defined by the N -by-1 vector

$$\mathbf{X} = [X[N-1], X[N-2], \dots, X[0]]^T \quad (6.224)$$

where

$$X[k] = \frac{1}{\sqrt{N}} \sum_{n=0}^{N-1} x[n] \exp\left(-j \frac{2\pi}{N} kn\right), \quad k = 0, 1, \dots, N-1 \quad (6.225)$$

The exponential term $\exp(-j2\pi kn/N)$ is referred to as the *kernel* of the DFT. Correspondingly, the *inverse discrete Fourier transform* (IDFT) of the N -by-1 vector \mathbf{X} is defined by

$$x[n] = \frac{1}{\sqrt{N}} \sum_{k=0}^{N-1} X[k] \exp\left(j \frac{2\pi}{N} kn\right), \quad n = 0, 1, \dots, N-1 \quad (6.226)$$

Although Equations (6.225) and (6.226) appear to be similar, they have different interpretations. Given the signal vector \mathbf{x} , Equation (6.225) provides a spectral representation of the signal computed at a set of discrete frequencies: $f_k = k/N$, which are normalized with respect to the sampling rate. Given the transformed vector \mathbf{X} , Equation (6.226) recovers the original signal vector \mathbf{x} . We may therefore view Equation (6.225) as the *analysis equation* and Equation (6.226) as the *synthesis equation*.

An important property of a circulant matrix, exemplified by the channel matrix \mathbf{H} of Equation (6.223), is that it permits *spectral decomposition* as shown by

$$\mathbf{H} = \mathbf{Q}^{\dagger} \mathbf{\Lambda} \mathbf{Q} \quad (6.227)$$

where the superscript \dagger denotes *Hermitian transposition* (i.e., the combination of complex conjugation and ordinary matrix transposition). Descriptions of the matrices \mathbf{Q} and $\mathbf{\Lambda}$ are presented in the sequel in that order.

The matrix \mathbf{Q} is a square matrix defined in terms of the kernel of the N -point DFT as follows:

$$\mathbf{Q} = \frac{1}{\sqrt{N}} \begin{bmatrix} \exp\left(-j \frac{2\pi}{N} (N-1)(N-1)\right) & \cdots & \exp\left(-j \frac{2\pi}{N} 2(N-1)\right) & \exp\left(-j \frac{2\pi}{N} (N-1)\right) & 1 \\ \exp\left(-j \frac{2\pi}{N} (N-1)(N-2)\right) & \cdots & \exp\left(-j \frac{2\pi}{N} 2(N-2)\right) & \exp\left(-j \frac{2\pi}{N} (N-2)\right) & 1 \\ \vdots & & \vdots & \vdots & \vdots \\ \exp\left(-j \frac{2\pi}{N} (N-1)\right) & \cdots & \exp\left(-j \frac{2\pi}{N} 2\right) & \exp\left(-j \frac{2\pi}{N}\right) & 1 \\ 1 & \cdots & 1 & 1 & 1 \end{bmatrix} \quad (6.228)$$

From this definition, we readily see that the k th element of the N -by- N matrix, \mathbf{Q} , starting from the *bottom right* at $k = 0$ and $l = 0$ and counting up step-by-step, is

$$q_{kl} = \frac{1}{\sqrt{N}} \exp\left(-j \frac{2\pi}{N} kl\right), \quad (k, l) = 0, 1, \dots, N-1 \quad (6.229)$$

The matrix \mathbf{Q} is an *orthonormal matrix* or *unitary matrix* in that it satisfies the condition

$$\mathbf{Q}^{\dagger} \mathbf{Q} = \mathbf{I} \quad (6.230)$$

where \mathbf{I} is the identity matrix. That is, the inverse matrix of \mathbf{Q} is equal to the Hermitian transpose of \mathbf{Q} .

The matrix $\mathbf{\Lambda}$ is a *diagonal matrix* that contains the N discrete Fourier transform values of the sequence h_0, h_1, \dots, h_{N-1} characterizing the channel. Denoting these transform values by $\lambda_{N-1}, \dots, \lambda_1, \lambda_0$, we may express $\mathbf{\Lambda}$ as

$$\mathbf{\Lambda} = \begin{bmatrix} \lambda_{N-1} & 0 & \cdots & 0 \\ 0 & \lambda_{N-2} & \cdots & 0 \\ \vdots & \vdots & & \vdots \\ 0 & 0 & \cdots & \lambda_0 \end{bmatrix} \quad (6.231)$$

(The λ s here are not to be confused with the Lagrange multipliers in Section 6.12.)

The DFT has established itself as one of the principal tools of digital signal processing by virtue of its efficient computation using the *fast Fourier transform (FFT) algorithm*.²⁰

Specifically, the FFT algorithm requires on the order of $N \log_2 N$ operations rather than the N^2 operations for direct computation of the DFT. For efficient implementation of the FFT algorithm, we should choose the block length N an integer power of two. The computational savings obtained by using the FFT algorithm are made possible by exploiting the special structure of the DFT defined in Equation (6.225). Moreover, these savings become more substantial as we increase the data length N .

■ FREQUENCY-DOMAIN DESCRIPTION OF THE CHANNEL

With this brief description of the DFT on hand, we are ready to resume our discussion of discrete multitone. First, we define

$$\mathbf{s} = \mathbf{Q}^T \mathbf{S} \quad (6.232)$$

where \mathbf{S} is the frequency-domain vector representation of the transmitter input. Each element of the N -by-1 vector \mathbf{S} may be viewed as a complex-valued point in a two-dimensional QAM signal constellation. Given the channel output vector \mathbf{x} , we define its corresponding frequency-domain representation as

$$\mathbf{X} = \mathbf{Q}\mathbf{x} \quad (6.233)$$

Using Equations (6.227), (6.232) and (6.233), we may rewrite Equation (6.219) in the equivalent form

$$\mathbf{X} = \mathbf{Q}(\mathbf{Q}^T \mathbf{A} \mathbf{Q} \mathbf{Q}^T \mathbf{S} + \mathbf{W}) \quad (6.234)$$

Hence, using the relation of Equation (6.230), we simply get

$$\mathbf{X} = \mathbf{A}\mathbf{S} + \mathbf{W} \quad (6.235)$$

where

$$\mathbf{W} = \mathbf{Q}\mathbf{w} \quad (6.236)$$

In expanded form, Equation (6.235) reads as

$$X_k = \lambda_k S_k + W_k, \quad k = 0, 1, \dots, N-1 \quad (6.237)$$

where the set of frequency-domain values $\{\lambda_k\}_{k=0}^{N-1}$ is known for a prescribed channel.

For a channel with additive white noise, Equation (6.237) implies that the receiver is composed of a set of independent processors operating in parallel. With the λ_k all known, we may thus use the block of frequency-domain values $\{X_k\}_{k=0}^{N-1}$ to compute estimates of the corresponding transmitted block of frequency domain-values $\{S_k\}_{k=0}^{N-1}$.

■ DFT-BASED DMT SYSTEM

Equations (6.235), (6.225), (6.226), and (6.237) provide the mathematical basis for the implementation of DMT using the DFT. Figure 6.60 illustrates the block diagram of the system derived from these equations and their practical implications.

The transmitter consists of the following functional blocks:

- ▶ *Demultiplexer*, which converts the incoming serial data stream into parallel form.
- ▶ *Constellation encoder*, which maps the parallel data into $N/2$ multibit subchannels with each subchannel being represented by a QAM signal constellation. Bit allocation among the subchannels is also performed here in accordance with a loading algorithm.

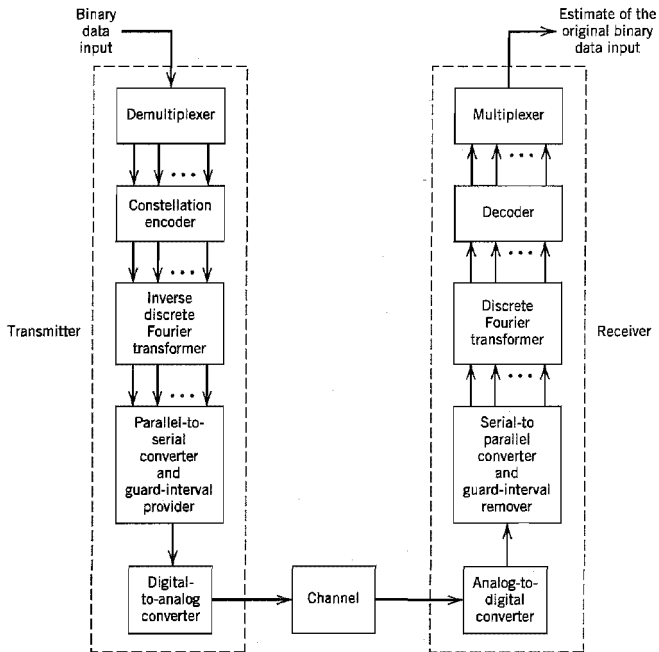


FIGURE 6.60 Block diagram of the discrete-multitone (DMT) data-transmission system.

- ▶ *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.

- ▶ *Decoder*, which uses the DFT output to compute estimates of the original multi-bit subchannel data supplied to the transmitter.
- ▶ *Multiplexer*, which combines the estimates so computed to produce a reconstruction of the transmitted serial data stream.

■ APPLICATIONS OF DMT

An important application of DMT is in the transmission of data over two-way channels. Indeed, DMT has been standardized for use on *asymmetric digital subscriber lines* (ADSLs) using twisted pairs. The ADSL was described in Chapter 4. For example, DMT provides for the transmission of data downstream (i.e., from an Internet service provider to a subscriber) at the DS1 rate of 1.544 Mb/s and the simultaneous transmission of data upstream (i.e., from the subscriber to the Internet service provider) at 160 kb/s. This kind of data transmission capability is well suited for handling data-intensive applications such as video-on-demand.

DMT is also a core technology in implementing the asymmetric *very-high-rate digital subscriber lines*²¹ (VDSL), which differs from all other DSL transmission techniques because of its ability to deliver extremely high data rates. For example, VDSL can provide data rates of 13 to 26 Mb/s downstream and 2 to 3 MB/s upstream over twisted pairs that emanate from an optical network unit and connect to the subscriber over distances less than about 1 km. These high data rates allow the delivery of digital TV, super-fast Web surfing and file transfer, and virtual offices at home.

The use of DMT for ADSL and VDSL provides a number of advantages:

- ▶ *The ability to maximize the transmitted bit rate*, which is provided by tailoring the distribution of information-bearing signals across the channel according to channel attenuation and noise conditions.
- ▶ *Adaptivity to changing line conditions*, which is realized by virtue of the fact that the channel is partitioned into a number of subchannels.
- ▶ *Reduced sensitivity to impulse noise*, which is achieved by spreading its energy over the many subchannels of the receiver. As the name implies, *impulse noise* is characterized by long, quiet intervals followed by narrow pulses of randomly varying amplitude. In an ADSL or VDSL environment, impulse noise arises due to switching transients coupled to wire pairs in the central office and to various electrical devices on the user's premises.

■ COMPARISON OF DIGITAL SUBSCRIBER LINES AND VOICEBAND MODEMS

In Section 6.11 we discussed voiceband modems that are already close to operating at their theoretical limits of 33.6 kb/s upstream and 56 kb/s downstream. In this section we have discussed the application of DMT to VDSLs that can operate at data rates of about 2 to 3 Mb/s upstream and 13 to 26 Mb/s downstream. These two vastly different sets of upstream/downstream data rates prompt the following question: How is it possible for VDSL to operate at rates about three orders of magnitude faster than voiceband modems over the same twisted pairs (i.e., phone lines)? The reason for this vast difference in operating data rates between voiceband modems and VDSLs is not the twisted pairs; rather, it is the *digital switches* built into a public switched telephone network that prevent the transport of broadband data to subscribers (users) via voiceband modems. Simply put, the digital switches treat digital data in the same way as voice signals for which they are primarily designed.

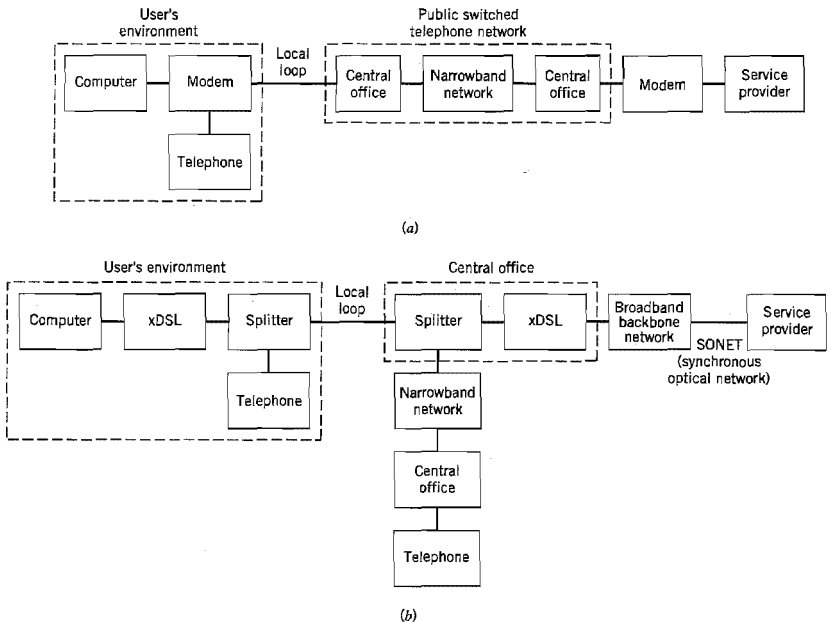


FIGURE 6.61 (a) Voiceband modem environment. (b) xDSL (digital subscriber line) environment, where x stands for “asymmetric” or “very high-rate.”

Figure 6.61 highlights the operational environments of voiceband modems and xDSLs, where x stands for A in ADSL and V in VDSL. In the model of Figure 6.61a pertaining to a voiceband modem, we have a relatively long transmission path between an Internet service provider (ISP) and a subscriber. Most importantly, the transmission path traverses through a narrowband public switched telephone network (PSTN), which limits the available channel bandwidth to about 3.5 kHz. In contrast, in the model of Figure 6.61b pertaining to xDSL, the transmission path accommodates the transport of broadband data between the ISP and subscriber via a broadband integrated services digital network and a relatively short local loop consisting of a twisted pair. The system permits the coexistence of POTS and xDSL signals on the same local loop, which is made possible through the use of a pair of splitters, as indicated in Figure 6.61b; splitters, consisting of bidirectional low-pass and high-pass filters, are discussed in Section 4.8.

■ ORTHOGONAL FREQUENCY DIVISION MULTIPLEXING²²

Discrete multitone is one particular discrete form of multichannel modulation. Another closely related form of this method of modulation is *orthogonal frequency-division multiplexing* (OFDM) that differs from DMT in areas of application and some aspects of its design.

OFDM is used for data transmission over radio broadcast channels and wireless communication channels. This domain of application requires some changes to the design

of the OFDM system. Unlike DMT that uses loading for bit allocation, OFDM uses a fixed number of bits per subchannel. This restriction is made necessary by the fact that a broadcast channel involves one-way transmission, and in a wireless communications environment the channel is varying too rapidly. Accordingly, in both cases it is not feasible for the transmitter to know the channel and how to “load” it.

Thus, the block diagram of Figure 6.60 applies equally to OFDM except for the fact that the signal constellation encoder does not include a loading algorithm for bit allocation. In addition, two other changes have to be made to the design of the system:

- ▶ In the transmitter, an *upconverter* is included after the digital-to-analog converter to translate the transmitted frequency, thereby facilitating the propagation of the transmitted signal over a radio channel.
- ▶ In the receiver, a *downconverter* is included before the analog-to-digital converter to undo the frequency translation that was performed by the upconverter in the transmitter.

Applications of OFDM include the following:

1. *Wireless communications.*

OFDM, combined with coding and interleaving, provides an effective technique to combat multipath fading that is a characteristic feature of wireless communication channels.

2. *Digital audio broadcasting.*

OFDM has been adopted as the standard for digital audio broadcasting in Europe. Here again the system involves the combined use of coding and interleaving.

(Error-control coding and related issues are discussed in Chapter 10.)

6.14 Synchronization

The coherent reception of a digitally modulated signal, irrespective of its form, requires that the receiver be synchronous to the transmitter. We say that two sequences of events (representing a transmitter and a receiver) are *synchronous* relative to each other when the events in one sequence and the corresponding events in the other occur simultaneously. The process of making a situation synchronous, and maintaining it in this condition, is called *synchronization*.²³

From the discussion presented on the operation of digital modulation techniques, we recognize the need for two basic modes of synchronization:

1. When coherent detection is used, knowledge of both the frequency and phase of the carrier is necessary. The estimation of carrier phase and frequency is called *carrier recovery* or *carrier synchronization*.
2. To perform demodulation, the receiver has to know the instants of time at which the modulation can change its state. That is, it has to know the starting and finishing times of the individual symbols, so that it may determine when to sample and when to quench the product-integrators. The estimation of these times is called *clock recovery* or *symbol synchronization*.

These two modes of synchronization can be coincident with each other, or they can occur sequentially one after the other. Naturally, in a noncoherent system, carrier synchronization is of no concern.

Synchronization can be implemented in one of two fundamentally different ways:

1. *Data-aided synchronization.*

In data-aided synchronization systems, a preamble is transmitted along with the data-bearing signal in a time-multiplexed manner on a periodic basis. The preamble contains information about the carrier and symbol timing, which is extracted by appropriate processing of the channel output at the receiver. Such an approach is commonly used in digital satellite and wireless communications, where the motivation is to minimize the time required to synchronize the receiver to the transmitter. Its limitations are two-fold: (1) reduced data-throughput efficiency that is incurred by assigning a certain portion of each transmitted frame to the preamble, and (2) reduced power efficiency by allocating a certain fraction of the transmitted power to the transmission of the preamble.

2. *Nondata-aided synchronization.*

In this second approach, the use of a preamble is avoided, and the receiver has the task of establishing synchronization by extracting the necessary information from the modulated signal. Both throughput and power efficiency are thereby improved but at the expense of an increase in the time taken to establish synchronization.

In any event, synchronization is basically a statistical parameter estimation problem. A principled approach for solving such a problem is *maximum likelihood estimation* (see Section 5.5), which proceeds by first formulating a log-likelihood function of the parameter of interest given the received signal. This formulation is relatively straightforward by treating the channel noise as a Gaussian process. Most important, it requires no prior information about the modulated signal.

In this section we confine our attention to nondata-aided forms of carrier and timing synchronization systems. In this context, we may identify two approaches for solving the synchronization problem, given a modulated signal with suppressed carrier to conserve power:

1. *Classical approach.*

An essential building block in the classical approach to synchronization is the *phase-locked loop*. (The phase-locked loop was discussed in Chapter 2.) Specifically, for carrier recovery the receiver requires the use of a *suppressed-carrier tracking loop* for providing a coherent secondary carrier (subcarrier) reference. For example, we may use a variant of the Costas loop or the *Mth power loop* for *M*-ary PSK. The standard Costas loop for double sideband-suppressed carrier (DSB-SC) modulation was discussed in Chapter 2. As for the *Mth power loop*, it consists of the cascade connection of an *Mth power-law* device, band-pass filter, phase-locked loop, and frequency divider by *M*. The objective here is to exploit the acquisition and tracking properties of the phase-locked loop. For further discussion of the *Mth power loop*, the reader is referred to Problem 6.47.

2. *Algorithmic (modern) approach.*

In the modern approach, the solution to maximum likelihood estimation is formulated in algorithmic form using discrete-time signal processing. Specifically, implementation of the synchronizer is built on an algorithm that provides an estimate of carrier phase or symbol timing on an iteration-by-iteration basis. The processing is performed in the baseband domain to pave the way for the use of discrete-time (digital) signal processing.

In this section we describe the algorithmic approach to synchronization for *M*-ary PSK systems for both carrier recovery and symbol-timing recovery.

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 this 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 $k = 0, 1, \dots, M - 1$:

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

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

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

Equivalently, we may write

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

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

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

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 $0 \leq t \leq T_0$ 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)

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

* 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.

At the receiver the basis functions are defined by

$$\phi_1(t) = \sqrt{\frac{2}{T}} \cos(2\pi f_c t), \quad \tau \leq t \leq T + \tau \quad (6.243)$$

$$\phi_2(t) = \sqrt{\frac{2}{T}} \sin(2\pi f_c t), \quad \tau \leq t \leq T + \tau \quad (6.244)$$

Here it is assumed that the receiver has perfect knowledge of the carrier frequency f_c ; otherwise, a carrier frequency offset has to be included, which complicates the analysis. Accordingly, we may represent the received signal $x(t)$ by the vector

$$\mathbf{x}(\tau) = \begin{bmatrix} x_1(\tau) \\ x_2(\tau) \end{bmatrix} \quad (6.245)$$

where

$$x_i(\tau) = \int_{\tau}^{T+\tau} x(t) \phi_i(t) dt, \quad i = 1, 2 \quad (6.246)$$

In a corresponding fashion, we may express the signal component of $\mathbf{x}(t)$ by the vector

$$s(a_k, \theta, \tau) = \begin{bmatrix} s_1(a_k, \theta, \tau) \\ s_2(a_k, \theta, \tau) \end{bmatrix} \quad (6.247)$$

where a_k is the transmitted symbol and

$$s_i(a_k, \theta, \tau) = \int_{\tau}^{T+\tau} \sqrt{\frac{2E}{T}} \cos(2\pi f_c t + \theta + \alpha_k) \phi_i(t) dt \text{ for } i = 1, 2 \quad (6.248)$$

Assuming that f_c is an integer multiple of the symbol rate $1/T$, we have

$$s_1(a_k, \theta, \tau) = \sqrt{E} \cos(\theta + \alpha_k) \quad (6.249)$$

$$s_2(a_k, \theta, \tau) = -\sqrt{E} \sin(\theta + \alpha_k) \quad (6.250)$$

We may thus write

$$\mathbf{x}_k(\tau) = s(a_k, \theta, \tau) + \mathbf{w} \quad (6.251)$$

where \mathbf{w} is the noise vector

$$\mathbf{w} = \begin{bmatrix} w_1 \\ w_2 \end{bmatrix} \quad (6.252)$$

with

$$w_i = \int_{\tau}^{T+\tau} w(t) \phi_i(t) dt, \quad i = 1, 2 \quad (6.253)$$

The w_i is the sample value of a Gaussian random variable W of zero mean and variance $N_0/2$, where $N_0/2$ is the (two-sided) power spectral density of the channel noise $w(t)$.

The conditional probability density function of the random vector \mathbf{X} , given the transmission of symbol a_k and the occurrence of carrier phase θ and group delay τ , is

$$f_{\mathbf{X}}(\mathbf{x} | a_k, \theta, \tau) = \frac{1}{\pi N_0} \exp\left(-\frac{1}{N_0} \|\mathbf{x}_k(\tau) - s(a_k, \theta, \tau)\|^2\right) \quad (6.254)$$

For $a_k = 0$ the received signal $x(t)$ equals the channel noise $w(t)$, so

$$f_{\mathbf{x}}(\mathbf{x} | a_k = 0) = \frac{1}{\pi N_0} \exp\left(-\frac{1}{N_0} \|\mathbf{x}_k(\tau)\|^2\right) \quad (6.255)$$

Hence we may define the likelihood function for M -ary PSK at the receiver as

$$\begin{aligned} L(a_k, \theta, \tau) &= \frac{f_{\mathbf{x}}(\mathbf{x} | a_k, \theta, \tau)}{f_{\mathbf{x}}(\mathbf{x} | a_k = 0)} \\ &= \exp\left(\frac{2}{N_0} \mathbf{x}_k^T(\tau) s(a_k, \theta, \tau) - \frac{1}{N_0} \|\mathbf{s}(a_k, \theta, \tau)\|^2\right) \end{aligned} \quad (6.256)$$

In M -ary PSK,

$$\|\mathbf{s}(a_k, \theta, \tau)\| = \text{constant}$$

as the message points lie on a circle of radius \sqrt{E} . Hence, ignoring the second term in the exponent, we may simplify the likelihood function as

$$L(a_k, \theta, \tau) = \exp\left(\frac{2}{N_0} \mathbf{x}_k^T(\tau) s(a_k, \theta, \tau)\right) \quad (6.257)$$

Assuming that we transmit a sequence of L_0 statistically independent symbols, namely,

$$\mathbf{a} = [a_0, a_1, \dots, a_{L_0-1}]^T \quad (6.258)$$

the resulting likelihood function is

$$L(\mathbf{a}, \theta, \tau) = \prod_{k=0}^{L_0-1} \exp\left(\frac{2}{N_0} \mathbf{x}_k^T(\tau) s(a_k, \theta, \tau)\right) \quad (6.259)$$

The log-likelihood function is therefore

$$\begin{aligned} l(\mathbf{a}, \theta, \tau) &= \log L(\mathbf{a}, \theta, \tau) \\ &= \frac{2}{N_0} \sum_{k=0}^{L_0-1} \mathbf{x}_k^T(\tau) s(a_k, \theta, \tau) \end{aligned} \quad (6.260)$$

From Equations (6.249) and (6.250) we deduce

$$\begin{aligned} \hat{s}_k(\theta) &= \mathbf{s}(\hat{a}_k, \theta, \tau) \\ &= \sqrt{E} \begin{bmatrix} \cos(\hat{\alpha}_k + \theta) \\ -\sin(\hat{\alpha}_k + \theta) \end{bmatrix}, \quad k = 0, 1, \dots, L_0 - 1 \end{aligned} \quad (6.261)$$

where $\hat{\alpha}_k$ is an estimate of the actual α_k produced at the detector output for the symbol a_k . Correspondingly, we may express the matched filter output as

$$\mathbf{x}_k = \begin{bmatrix} x_{1,k} \\ -x_{2,k} \end{bmatrix}$$

Hence, using this definition and Equation (6.261) in Equation (6.260), we get

$$\begin{aligned} l(\theta) &= \frac{2\sqrt{E}}{N_0} \sum_{k=0}^{L_0-1} [x_{1,k} \cos(\hat{\alpha}_k + \theta) + x_{2,k} \sin(\hat{\alpha}_k + \theta)] \\ &= \frac{2\sqrt{E}}{N_0} \sum_{k=0}^{L_0-1} [(x_{1,k} \cos \hat{\alpha}_k + x_{2,k} \sin \hat{\alpha}_k) \cos \theta \\ &\quad - (x_{1,k} \sin \hat{\alpha}_k - x_{2,k} \cos \hat{\alpha}_k) \sin \theta] \end{aligned} \quad (6.262)$$

Differentiating $l(\theta)$ with respect to θ , we obtain

$$\frac{\partial l(\theta)}{\partial \theta} = -\frac{2\sqrt{E}}{N_0} \sum_{k=0}^{L_0-1} [(x_{1,k} \cos \hat{\alpha}_k + x_{2,k} \sin \hat{\alpha}_k) \sin \theta + (x_{1,k} \sin \hat{\alpha}_k - x_{2,k} \cos \hat{\alpha}_k) \cos \theta] \quad (6.263)$$

We may simplify Equation (6.263) by introducing the following notations:

$$\tilde{x}_k = x_{1,k} + jx_{2,k} \quad (6.264)$$

and

$$\begin{aligned} a_k &= e^{j\alpha_k} \\ &= \cos \alpha_k + j \sin \alpha_k \end{aligned} \quad (6.265)$$

where \tilde{x}_k is the complex envelope (i.e., baseband value) of the matched filter output due to the k th transmitted symbol, and a_k is a symbol indicator in the message constellation of the M -ary PSK. We may thus write

$$\begin{aligned} \text{Re}[\hat{a}_k^* \tilde{x}_k] &= \text{Re}[(\cos \hat{\alpha}_k - j \sin \hat{\alpha}_k)(x_{1,k} + jx_{2,k})] \\ &= x_{1,k} \cos \hat{\alpha}_k + x_{2,k} \sin \hat{\alpha}_k \end{aligned} \quad (6.266)$$

$$\begin{aligned} \text{Im}[\hat{a}_k^* \tilde{x}_k] &= \text{Im}[(\cos \hat{\alpha}_k - j \sin \hat{\alpha}_k)(x_{1,k} + jx_{2,k})] \\ &= -x_{1,k} \sin \hat{\alpha}_k + x_{2,k} \cos \hat{\alpha}_k \end{aligned} \quad (6.267)$$

We may also note from Euler's formula:

$$e^{-j\theta} = \cos \theta - j \sin \theta \quad (6.268)$$

Accordingly, we may rewrite Equation (6.263) in the compact form:

$$\begin{aligned} \frac{\partial l(\theta)}{\partial \theta} &= \frac{2\sqrt{E}}{N_0} \sum_{k=0}^{L_0-1} \{(\text{Re}[\hat{a}_k^* \tilde{x}_k])(\text{Im}[e^{-j\theta}]) + (\text{Im}[\hat{a}_k^* \tilde{x}_k])(\text{Re}[e^{-j\theta}])\} \\ &= \frac{2\sqrt{E}}{N_0} \sum_{k=0}^{L_0-1} \text{Im}[\hat{a}_k^* \tilde{x}_k e^{-j\theta}] \end{aligned} \quad (6.269)$$

where \hat{a}_k is an estimate of a_k , and the asterisk denotes complex conjugation.

■ RECURSIVE ALGORITHM FOR MAXIMUM LIKELIHOOD ESTIMATION OF THE CARRIER PHASE

With the formula of Equation (6.269) for the derivative of the log-likelihood function $l(\theta)$ with respect to the carrier phase θ at hand, we are now ready to formulate an algorithm that seeks to maximize $l(\theta)$. We would like to perform the maximization in an iterative fashion so that the receiver is enabled to respond to the received signal on a symbol-by-symbol basis. To that end, we may build on the following algorithmic idea borrowed from adaptive filtering (see the discussions on the LMS algorithm presented in Chapters 3 and 4):

$$\begin{pmatrix} \text{Updated} \\ \text{estimate} \end{pmatrix} = \begin{pmatrix} \text{Old} \\ \text{estimate} \end{pmatrix} + \begin{pmatrix} \text{Step-size} \\ \text{parameter} \end{pmatrix} \begin{pmatrix} \text{Error} \\ \text{signal} \end{pmatrix} \quad (6.270)$$

where the *error signal*, or the adjustment signal to be more precise, is defined as the instantaneous value of the gradient of the log-likelihood function $l(\theta)$ with respect to θ . Note that the parameter adjustment applied to the old estimate in Equation (6.270) is

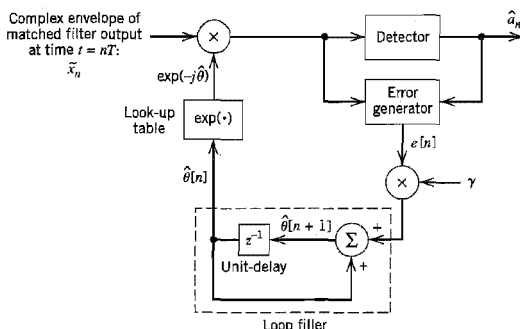


FIGURE 6.62 Recursive Costas loop.

positive as the objective here is to perform *gradient ascent*. From Equation (6.269) we readily see that the error signal (i.e., the instantaneous value of $\partial l(\theta)/\partial \theta$ due to the transmission of a single symbol) is given by

$$e[n] = \text{Im}[\hat{a}_n^* \hat{x}_n e^{-j\hat{\theta}}] \quad (6.271)$$

where the scaling factor $2\sqrt{E}/N_0$ is accounted for in what follows. Also, we have used n in place of k to denote a time step or iteration of the algorithm. Accordingly, we use Equation (6.270) to write

$$\hat{\theta}[n+1] = \hat{\theta}[n] + \gamma e[n] \quad (6.272)$$

where $\hat{\theta}[n]$ is the *old estimate* of the carrier phase θ , $\hat{\theta}[n+1]$ is the *updated estimate* of θ , and γ is the *step-size parameter*; the scaling factor $2\sqrt{E}/N_0$ is absorbed in γ .

Equations (6.271) and (6.272) define the recursive algorithm for phase recovery. This algorithm is implemented using the system shown in Figure 6.62, which may be viewed as a recursive generalization of the Costas loop. We may therefore refer to it as the *recursive Costas loop* for phase synchronization.

The following points should be noted in Figure 6.62:

- ▶ The detector supplies an estimate of the transmitted symbol \hat{a}_n , given the matched filter output.
- ▶ The look-up table supplies the value of $\exp(-j\hat{\theta}[n]) = \cos \hat{\theta}[n] - j \sin \hat{\theta}[n]$ for an input $\hat{\theta}[n]$.
- ▶ The output of the error generator is the error signal $e[n]$.
- ▶ The block labeled z^{-1} is a unit-delay element with the delay equal to the symbol period T .

The recursive Costas loop of Figure 6.62 uses a first-order digital filter. To improve the tracking performance of this synchronization system we may use a second-order digital filter. Figure 6.63 shows an example of a second-order digital filter made up of a cascade of two first-order sections, with ρ as an adjustable *loop parameter*. An important property of a second-order filter used in the Costas loop for phase recovery is that it will eventually lock onto the incoming carrier with no static error, provided that the frequency error between the receiver and transmitter is initially small.

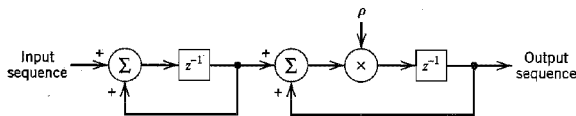


FIGURE 6.63 Second-order digital filter.

■ NONDATA-AIDED RECURSIVE ALGORITHM FOR SYMBOL TIMING

For timing synchronization the only assumption made is that the receiver has knowledge of the carrier frequency f_c . The requirement is to develop an algorithm for recursive estimation of the group delay τ incurred in the course of transmitting the modulated signal through the channel.

Let $L(a_k, \theta, \tau)$ denote the likelihood function of τ , which is also a function of transmitted symbol a_k and carrier phase θ . The likelihood function is defined by Equation (6.257). To proceed further we must remove the dependencies of $L(a_k, \theta, \tau)$ on the transmitted data sequence $\{a_k\}$ and carrier phase θ , as described next.

To remove the dependence on θ we average the likelihood function $L(a_k, \theta, \tau)$, but *not* its logarithm, over all possible values of θ inside the range $[0, 2\pi]$. Assuming that θ is uniformly distributed inside this range, which is usually justifiable, we may write

$$\begin{aligned} L_{av}(a_k, \tau) &= \int_0^{2\pi} L(a_k, \theta, \tau) f_\theta(\theta) d\theta \\ &= \frac{1}{2\pi} \int_0^{2\pi} \exp\left(\frac{2}{N_0} \mathbf{x}_k^T(\tau) \mathbf{s}(a_k, \theta, \tau)\right) d\theta \end{aligned}$$

The exponent in $L(a_k, \theta, \tau)$ is expressed by (see Problem 6.49)

$$\begin{aligned} \frac{2}{N_0} \mathbf{x}_k^T(\tau) \mathbf{s}(a_k, \theta, \tau) &= \frac{2\sqrt{E}}{N_0} \operatorname{Re}[a_k^* \tilde{x}_k(\tau) e^{-j\theta}] \\ &= \frac{2\sqrt{E}}{N_0} \operatorname{Re}[|a_k \tilde{x}_k(\tau)| \exp(j(\arg[\tilde{x}_k(\tau)] - \arg[a_k] - \theta))] \quad (6.273) \\ &= \frac{2\sqrt{E}}{N_0} |a_k \tilde{x}_k(\tau)| \cos(\arg[\tilde{x}_k(\tau)] - \arg[a_k] - \theta) \end{aligned}$$

Hence,

$$\begin{aligned} L_{av}(a_k, \tau) &= \frac{1}{2\pi} \int_0^{2\pi} \exp\left(\frac{2\sqrt{E}}{N_0} |a_k \tilde{x}_k(\tau)| \cos(\arg[\tilde{x}_k(\tau)] - \arg[a_k] - \theta)\right) d\theta \\ &= \frac{1}{2\pi} \int_{-\arg[\tilde{x}_k(\tau)] + \arg[a_k]}^{2\pi - \arg[\tilde{x}_k(\tau)] + \arg[a_k]} \exp\left(\frac{2\sqrt{E}}{N_0} |a_k \tilde{x}_k(\tau)| \cos(\varphi) d\varphi\right) \quad (6.274) \end{aligned}$$

where, in the last line, we have made the substitution

$$\varphi = \arg[\tilde{x}_k(\tau)] - \arg[a_k] - \theta$$

We now invoke the definition of the modified Bessel function of zero order, as shown by (see Appendix 3)

$$I_0(x) = \frac{1}{2\pi} \int_0^{2\pi} e^{x \cos \varphi} d\varphi \quad (6.275)$$

Hence, we may express the average likelihood function $L_{av}(a_k, \tau)$ as

$$L_{av}(a_k, \tau) = I_0 \left(\frac{2\sqrt{E}}{N_0} |a_k \tilde{x}_k(\tau)| \right) \quad (6.276)$$

where $\tilde{x}_k(\tau)$ is the complex envelope of the matched filter output in the receiver due to the k th transmitted symbol a_k . For M -ary PSK, we have

$$|a_k| = 1 \quad \text{for all } k$$

Hence, Equation (6.276) reduces to

$$L_{av}(a_k, \tau) = I_0 \left(\frac{2\sqrt{E}}{N_0} |\tilde{x}_k(\tau)| \right) \quad (6.277)$$

We thus see that averaging the likelihood function over the carrier phase θ has also removed dependence on the transmitted symbol a_k for M -ary PSK.

Finally, taking account of the transmission of L_0 independent symbols $a_0, a_1, \dots, a_{L_0-1}$, we may express the overall likelihood function of τ as

$$\begin{aligned} L_{av}(\tau) &= \prod_{k=0}^{L_0-1} L_{av}(a_k, \tau) \\ &= \prod_{k=0}^{L_0-1} I_0 \left(\frac{2\sqrt{E}}{N_0} |\tilde{x}_k(\tau)| \right) \end{aligned} \quad (6.278)$$

Now we can take the logarithm of $L_{av}(\tau)$ to obtain the log-likelihood function of τ as

$$\begin{aligned} l_{av}(\tau) &= \log L_{av}(\tau) \\ &= \sum_{k=0}^{L_0-1} \log I_0 \left(\frac{2\sqrt{E}}{N_0} |\tilde{x}_k(\tau)| \right) \end{aligned} \quad (6.279)$$

To proceed further, we need to approximate $I_0(\tau)$. To that end we first note that the modified Bessel function $I_0(x)$ may be expanded in a power series as (see Appendix 3)

$$I_0(x) = \sum_{m=0}^{\infty} \frac{\left(\frac{1}{2}x\right)^{2m}}{(m!)^2}$$

For small values of x we may thus approximate $I_0(x)$ as

$$I_0(x) \approx 1 + \frac{x^2}{4}$$

We may further simplify matters by using the approximation

$$\begin{aligned} \log I_0(x) &\approx \log \left(1 + \frac{x^2}{4} \right) \\ &\approx \frac{x^2}{4} \quad \text{for small } x \end{aligned}$$

For the problem at hand, small x corresponds to small signal-to-noise ratio. Under this condition, we may approximate Equation (6.279) as

$$l_{av}(\tau) \approx \frac{E}{N_0^2} \sum_{k=0}^{L_0-1} |\tilde{x}_k(\tau)|^2 \quad (6.280)$$

where, as mentioned earlier, $\tilde{x}_k(\tau)$ is the complex envelope of the matched filter output due to the k th transmitted symbol.

Differentiating $l_{av}(\tau)$ with respect to the group delay τ , we obtain

$$\begin{aligned}\frac{\partial l_{av}(\tau)}{\partial \tau} &= \frac{E}{N_0^2} \sum_{k=0}^{L_0-1} \frac{\partial}{\partial \tau} |\tilde{x}_k(\tau)|^2 \\ &= \frac{2E}{N_0^2} \sum_{k=0}^{L_0-1} \text{Re}[\tilde{x}_k^*(\tau) \tilde{x}_k'(\tau)]\end{aligned}\quad (6.281)$$

where $\tilde{x}_k^*(\tau)$ is the complex conjugate of $\tilde{x}_k(\tau)$ and $\tilde{x}_k'(\tau)$ is its derivative with respect to τ . Accordingly, we may define the error signal for timing recovery as (accounting for the scaling factor $2E/N_0^2$ in what follows)

$$e[n] = \text{Re}[\hat{x}_n^*(\tau) \hat{x}_n'(\tau)]$$

where we have used n in place of k to be consistent with the notation in Figure 6.62. Let $\hat{\tau}_n$ denote the estimate of the unknown delay τ at time $t = nT$. Then, introducing the definitions

$$\tilde{x}_n(\tau) = \tilde{x}(nT + \hat{\tau}_n)$$

and

$$\tilde{x}_n'(\tau) = \tilde{x}'(nT + \hat{\tau}_n)$$

we may reformulate the error signal $e(n)$ as

$$e[n] = \text{Re}[\tilde{x}^*(nT + \hat{\tau}_n) \tilde{x}'(nT + \hat{\tau}_n)] \quad (6.282)$$

Calculation of the error signal $e[n]$ requires the use of two filters:

1. The complex matched filter for generating $\tilde{x}_n(\tau)$.
2. The derivative matched filter for generating $\tilde{x}_n'(\tau)$.

The receiver is already equipped with the first filter. The second one is new. In practice, the additional computational complexity due to the derivative matched filter is objectionable. We may dispense with the need for it by using a finite difference to approximate the derivative $\tilde{x}_n'(\tau)$ as

$$\tilde{x}'(nT + \hat{\tau}_n) \approx \frac{1}{T} \left[\tilde{x}\left(nT + \frac{T}{2} + \hat{\tau}_{n+1/2}\right) - \tilde{x}\left(nT - \frac{T}{2} + \hat{\tau}_{n-1/2}\right) \right] \quad (6.283)$$

where $\hat{\tau}_{n\pm 1/2}$ are the timing estimates computed at $nT \pm T/2$. It is desirable to make one further modification to account for the fact that timing estimates are updated at multiples of the symbol period T and the only available quantities are $\hat{\tau}_n$. Consequently, we replace $\hat{\tau}_{n+1/2}$ by $\hat{\tau}_n$ (which represents the latest estimate of τ) and replace $\hat{\tau}_{n-1/2}$ by $\hat{\tau}_{n-1}$ (which is the estimate of τ before the last one). We may thus rewrite Equation (6.283) as

$$\tilde{x}'(nT + \hat{\tau}_n) \approx \frac{1}{T} \left[\tilde{x}\left(nT + \frac{T}{2} + \hat{\tau}_n\right) - \tilde{x}\left(nT - \frac{T}{2} + \hat{\tau}_{n-1}\right) \right] \quad (6.284)$$

and so finally redefine the error signal as

$$e[n] = \text{Re} \left\{ \tilde{x}^*(nT + \hat{\tau}_n) \left[\tilde{x}\left(nT + \frac{T}{2} + \hat{\tau}_n\right) - \tilde{x}\left(nT - \frac{T}{2} + \hat{\tau}_{n-1}\right) \right] \right\} \quad (6.285)$$

where the scaling factor $1/T$ is also accounted for in what follows.

We are now ready to formulate the recursive algorithm for timing recovery:

$$c[n+1] = c[n] + \gamma e[n] \quad (6.286)$$

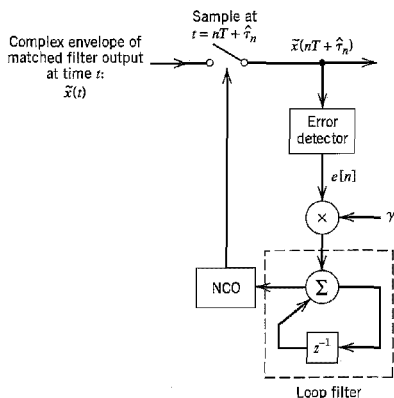


FIGURE 6.64 Nondata-aided early-late delay synchronizer.

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

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

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, $B_L T = 0.02$.

Experiment 1: Carrier Phase Recovery

In order to investigate the phase-acquisition behavior of the recursive Costas loop, we need to have the so-called *S-curve* of the phase-error generator. This is defined as the expectation of the adjustment signal $e[n]$, conditioned on a fixed value of the phase error

$$\varphi = \theta - \hat{\theta}$$

where θ is the actual value of the carrier phase and $\hat{\theta}$ is its estimate. That is,

$$S(\varphi) = E[e[n] | \varphi] \quad (6.288)$$

Experimentally, $S(\varphi)$ is measured by opening the recursive Costas loop of Figure 6.62 and measuring the average of the adjustment signal $e[n]$, as indicated in Figure 6.65.

The implementation procedure consists of the following steps. First, the complex envelope of the received signal is generated, which is given by

$$\tilde{x}_k(t) = \sqrt{\frac{2E}{T}} \exp(-j(2\pi f_c \tau_c - \alpha_k)) g(t - \tau_g) + \tilde{w}(t) \quad (6.289)$$

where $\alpha_k = 0, \pi/2, \pi, 3\pi/4$; τ_c is the carrier delay and τ_g is the group delay; and $\tilde{w}(t)$ is the complex-valued channel noise. The overall channel response $g(t)$ is given by the Nyquist pulse (see Section 4.5)

$$g(t) = \frac{\sin(\pi t/T)}{(\pi t/T)} \cdot \frac{\cos(\pi \alpha t/T)}{1 - 4\alpha^2 t^2/T^2} \quad (6.290)$$

where $\alpha = 0.5$. As pointed out earlier, we assume that the symbol timing (i.e., group delay τ_g) is known, and the problem is to estimate the carrier phase $\theta = -2\pi f_c \tau_c$. The effect of θ is to shift an element of the signal constellation in the manner indicated in Figure 6.66.

Using the experimental procedure described in Figure 6.65, the *S-curve* of the QPSK system may now be measured. Figure 6.67a shows the ideal *S-curve*, assuming an infinitely large signal-to-noise ratio. This curve displays discontinuities at $\varphi = \pm m\pi/4$, where $m = 0, 1, 3, \dots$, because of ambiguity encountered in the detection of the transmitted

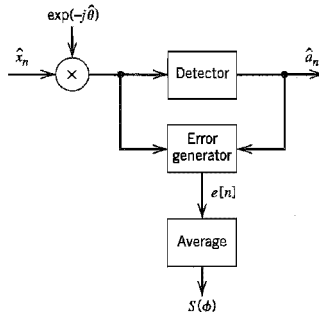


FIGURE 6.65 Scheme for measuring the *S-curve* for carrier phase recovery.

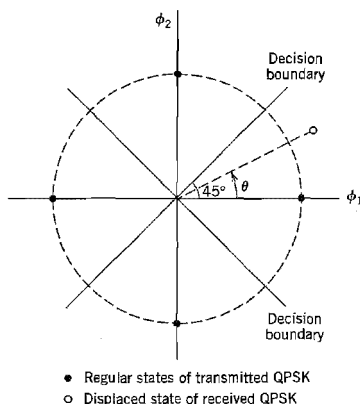


FIGURE 6.66 Illustrating the effect of carrier phase θ on a state of the QPSK signal.

symbol a_k . The presence of channel noise tends to roundoff the discontinuities, as shown in the experimentally measured S -curve of Figure 6.67b. The results presented in Figure 6.67b were obtained for $E/N_0 = 10$ dB. Recall that the in-phase and quadrature components of the narrowband noise have an identical Gaussian distribution with zero mean and the same variance as the original narrowband noise; these two components define $\tilde{w}(t)$.

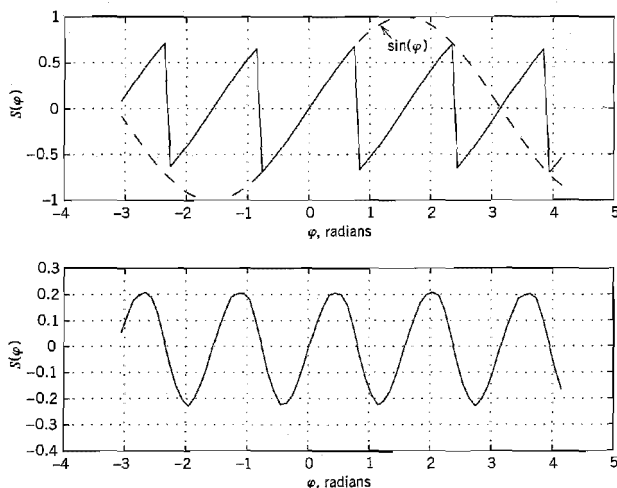


FIGURE 6.67 Performance of recursive Costas loop. (a) S -curve for $(E/N_0) = \infty$. (b) S -curve for $(E/N_0) = 10$ dB.

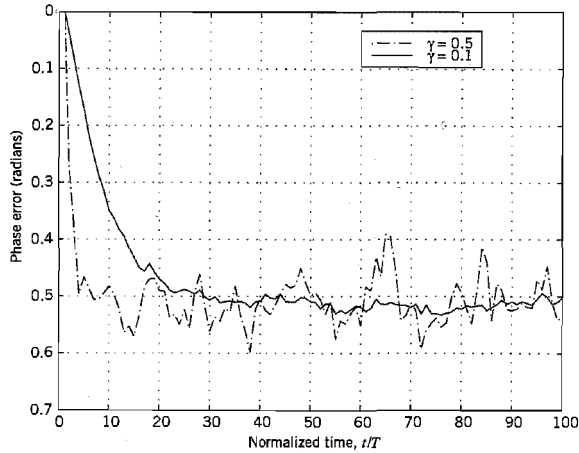


FIGURE 6.68 Effects of varying the step-size parameter on convergence behavior of the recursive Costas loop.

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

$$L_0 \approx \frac{1}{2B_L T} \quad (6.291)$$

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

$$B_L T \approx \frac{\gamma A}{4} \quad (6.292)$$

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.

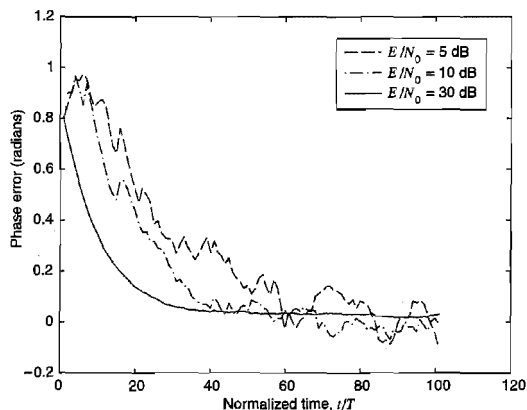


FIGURE 6.69 Convergence behavior of the recursive Costas loop for varying E/N_0 .

Figure 6.70 plots the variance of the phase error (averaged over 100 trials of the experiment) versus E/N_0 (measured in decibels) for $B_L T = 0.02$ and $\gamma = 0.08$. This figure also includes a plot of the *modified Cramér-Rao bound* defined by²⁴

$$\text{MCRB}(\theta) = \frac{1}{2L_0(E/N_0)} \quad (6.293)$$

This bound is a modification of the ordinary Cramér-Rao bound, which is a lower bound on the variance of any *unbiased* estimator. The modification to this bound is made to overcome computational difficulties encountered in practical synchronization problems.

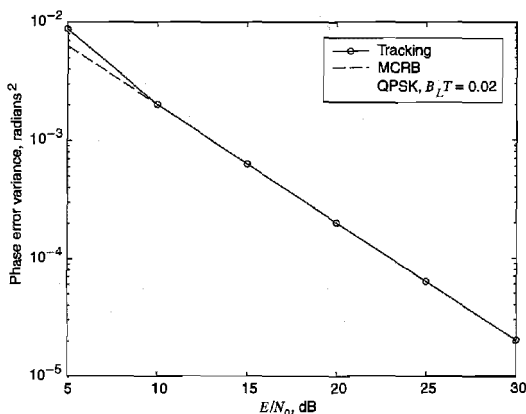


FIGURE 6.70 Comparison of the measured tracking-error variance of the recursive Costas loop against theory for varying E/N_0 .

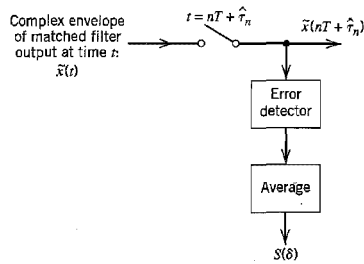


FIGURE 6.71 Scheme for measuring the S -curve for the recursive early-late-delay synchronizer.

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

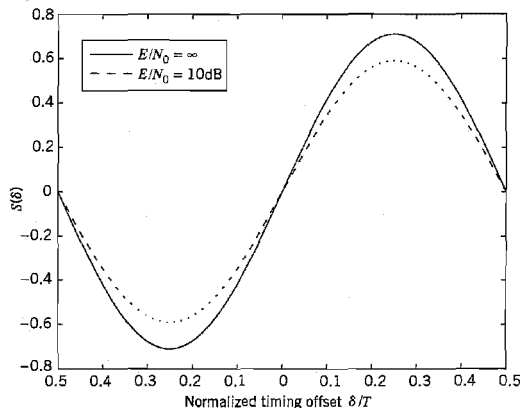


FIGURE 6.72 S -curve of NDA-ELD synchronizer measured under noiseless and noisy conditions.

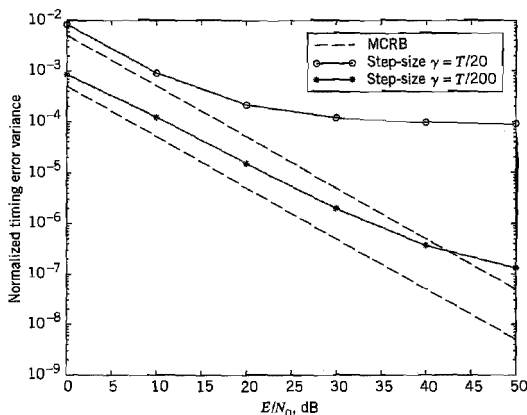


FIGURE 6.73 Comparison of tracking-error variance of the NDA-ELD synchronizer against theory for varying E/N_0 and two step-size parameters.

error is reduced and the range of E/N_0 for which the modified Cramér-Rao bound holds (albeit in an approximate fashion) is enlarged.

6.16 Summary and Discussion

With the basic background theory on optimum receivers of Chapter 5 at our disposal, in this chapter we derived formulas for, or bounds on, the bit error rate for some important digital modulation techniques in an AWGN channel:

1. Phase-shift keying (PSK), represented by
 - ▶ Coherent binary phase-shift keying (BPSK).
 - ▶ Coherent quadriphase shift keying (QPSK) and its variants, namely, the offset QPSK and $\pi/4$ -shifted QPSK.
 - ▶ Coherent M -ary PSK, which includes BPSK and QPSK as special cases with $M = 2$ and $M = 4$, respectively. Coherent M -ary PSK is used in digital satellite communications.
 - ▶ Differential phase-shift keying (DPSK), which may be viewed as the pseudo-non-coherent form of PSK.
2. Coherent M -ary quadrature amplitude modulation (QAM), which is a hybrid form of modulation that combines amplitude and phase-shift keying. For $M = 4$ it includes QPSK as a special case. M -ary QAM is basic to the construction of high-speed voiceband modems.
3. Frequency-shift keying (FSK), represented by
 - ▶ Coherent binary frequency-shift keying.
 - ▶ Coherent forms of minimum shift keying (MSK) and Gaussian minimum shift keying (GMSK); GMSK is basic to the construction of GSM wireless communications.

- Coherent M -ary FSK.
- Noncoherent binary FSK.

In this chapter we also studied two alternative techniques for passband data transmission: carrierless amplitude/phase modulation (CAP) and discrete multitone (DMT). In the case of an AWGN channel, the performance of CAP and DMT are equivalent because the DMT may be viewed as a linear reversible transformation of a single-carrier modulated signal. However, they perform quite differently in a practical setting that deviates from this idealized model.²⁵ DMT has been standardized for use on asymmetric digital subscriber lines (ADSLs) using twisted pairs. CAP, with the use of decision feedback equalization, provides another approach for solving the ADSL problem. CAP is also used for data transmission in local area networks for premises' distribution systems.

DMT is a form of multichannel modulation, and so is orthogonal frequency-division multiplexing (OFDM). The basic difference between DMT and OFDM is that DMT permits the use of loading to optimize information capacity, whereas OFDM does not. This difference arises because of their different domains of application. DMT applies to two-wire channels such as ADSLs, whereas OFDM applies to broadcasting and wireless channels.

Irrespective of the digital modulation system of interest, synchronization of the receiver to the transmitter is essential to the operation of the system. Symbol timing recovery is required whether the receiver is coherent or not. If the receiver is coherent, we also require provision for carrier recovery. In the latter part of the chapter, we discussed non-data-aided synchronizers to cater to these two requirements with emphasis on M -ary phase-shift keying signals in which the carrier is suppressed. The presentation focused on iterative synchronization techniques that are naturally suited for the use of digital signal processing.

NOTES AND REFERENCES

1. For an early tutorial paper reviewing different digital modulation techniques (ASK, FSK, and PSK) based on a geometric viewpoint, see Arthurs and Dym, (1962). See also the following list of books:
 - Anderson (1998, Chapter 3)
 - Benedetto and Biglieri (1999, Chapters 4 and 5)
 - Lee and Messerschmitt (1994, Part II)
 - Proakis (1995, Chapter 5)
 - Sklar (1988, Chapter 3)
 - Viterbi and Omura (1979, pp. 47–127)
2. For an early paper on the offset QPSK, see Gitlin and Ho (1975).
3. The $\pi/4$ -shifted QPSK was first described in the open literature in Akaiwa and Negata (1987).
4. Chennakeshu and Sauliner (1993) use computer simulations to study the performance of $\pi/4$ -shifted QPSK in a digital wireless communications environment. The pulse-shaping signal used in the generation of the $\pi/4$ -shifted QPSK signal is based on the square root raised cosine spectrum (see Problem 4.38). In this latter paper, it is shown that the performance of $\pi/4$ -shifted QPSK may degrade rapidly in such an environment. The differential detector of Figure 6.13 follows Chennakeshu and Sauliner (1993).

5. For a derivation of Equation (6.65), see Cioffi (1998).
6. The derivation of Equation (6.76) was first reported in a 1975 internal Bell Laboratories memorandum authored by Werner. A little later, Falconer (1975) issued another Bell Laboratories memorandum, in which it was pointed out the symbol rotation is not really needed if we did not want to be compatible with existing QAM or other bandpass signals, thereby simplifying the mathematical representation of CAP signals (and therefore their implementation), as shown in Equation (6.77). However, the terminology "CAP" was not coined until 1987 when carrierless amplitude/phase modulation was replaced by CAP by the standards representative, Garry Smith, of Bell Laboratories. The first detailed discussion of CAP in the context of digital subscriber lines was presented in a two-part report by Werner (1992, 1993). In a separate report by Chen, Im, and Werner (1992), the feasibility of CAP for use on digital subscriber lines was studied; see also the book by Chen (1998), pp. 461–473. The application of CAP to local area networks, involving the use of twisted pairs for lengths less than 100 m, is discussed in the paper by Im and Werner (1995); the maximum length of 100 m is specified by a standard for the wiring of premises.
The digital implementation of a baseband equalizer similar to the CAP receiver of Figure 6.24 is discussed in Mueller and Werner (1982).
7. The MSK signal was first described in Doelz and Heald (1961). For a tutorial review of MSK and comparison with QPSK, see Pasupathy (1979). Since the frequency spacing is only half as much as the conventional spacing of $1/T_b$, that is used in the coherent detection of binary FSK signals, this signaling scheme is also referred to as *fast* FSK; see deBuda (1972).
8. For early discussions of Gaussian MSK, see Murota and Hirade (1981) and Ishizuka and Hirade (1980).
9. The analytical specification of the power spectral density of digital FM is difficult to handle, except for the case of a rectangular shaped modulating pulse. The paper by Garrison (1975) presents a procedure based on the selection of an appropriate duration-limited/level-quantized approximation for the modulating pulse. The equations developed therein are particularly suitable for machine computation of the power spectra of digital FM signals; see the book by Stüber (1996).
10. A detailed analysis of the spectra of M -ary FSK for an arbitrary value of frequency deviation is presented in the paper by Anderson and Salz (1965). The results shown plotted in Figure 6.36 represent a special case of a formula derived in that paper for a frequency deviation of $k = 0.5$.
11. The standard method of deriving the bit error rate for noncoherent binary FSK, presented in McDonough and Whalen (1995) and that for differential phase-shift keying presented in Arthurs and Dym (1962), involves the use of the Rician distribution. This distribution arises when the envelope of a sine wave plus additive Gaussian noise is of interest; see Chapter 1 for a discussion of the Rician distribution. The derivations presented in Section 6.6 avoid the complications encountered in the standard method.
12. The optimum receiver for differential phase-shift keying is discussed in Simon and Divsalar (1992).
13. For a technical discussion of various kinds of modems, with emphasis on their operational characteristics, see the books of Lewart (1988) and Hold (1997).
14. In a two-part paper by Wei (1984), differential encoding is applied to convolutional channel coding. Several eight-state convolutional encoders are described therein, which result in codes that are transparent to signal element rotations. In particular, in part II of the paper, Wei describes design rules and procedures for a 90-degree rotationally invariant convolutional code that has been adopted for use in the V.32 modem with trellis coding.

15. Nonuniform sampling of band-limited signals is discussed in the paper by Yen (1956). The main results derived in that paper are contained in four generalized theorems. Equation (6.188) is based on Theorem III of Yen's paper.

In the paper by Kalet, Mazo, and Saltzberg (1993), this particular theorem due to Yen is used to formulate the fundamental philosophy underlying the design of the bidirectional digital modem; see also the patent by Ayanoglu et al. (1995).

16. For a discussion of the second realization of a digital modem, see the article by Humblet and Tzou (1996).
17. For a detailed description of the V.34 high-speed modem standard, see Forney et al. (1996). The trellis codes used in the V.34 modem are due to Wei (1984, 1987).
18. The idea of multichannel modulation may be traced to the early work of Chang (1966), Saltzberg (1967), and Weinstein and Ebert (1971). A mathematical treatment of the optimality of multitone modulation for a linear channel with severe intersymbol interference is presented in Kalet (1989). However, it was the work done by Cioffi and co-workers that led to the standardization of discrete multitone (DMT) for asymmetric digital subscriber lines; for details, see Ruiz et al. (1992), Chow and Cioffi (1995), Section 7.2 of the book by Starr et al. (1999), and Chapter 11 of Cioffi (1998). Problem 6.44 is adapted from Cioffi (1998).
19. The *method of Lagrange multipliers* for determining the extreme values of the function

$$y = f(x)$$

subject to the constraint

$$\varphi(x) = 0$$

follows from the following theorem: A necessary and sufficient condition for an extremum of a continuously differentiable function $f(x)$ is that its differential with respect to x vanishes at the critical (i.e., maximum and minimum) points of the function. Accordingly, at the critical points of $f(x)$ we have

$$\frac{\partial f}{\partial x} dx = 0 \quad (1)$$

Moreover, since $\varphi(x) = 0$, its differential also vanishes as shown by

$$\frac{\partial \varphi}{\partial x} dx = 0 \quad (2)$$

Hence multiplying (2) by some parameter λ and then adding the result to (1), we get

$$\left(\frac{\partial f}{\partial x} + \lambda \frac{\partial \varphi}{\partial x} \right) dx = 0$$

Since dx is an independent increment, we immediately deduce that

$$\frac{\partial}{\partial x} (f(x) + \lambda \varphi(x)) = 0$$

This equation is a mathematical statement of the method of Lagrange multipliers. The parameter λ is called the *Lagrange multiplier*. The material presented in this note follows Sokolnikoff and Redheffer (1966, pp. 341–344).

20. In the discrete Fourier transform (DFT), both the input and the output consist of sequences of numbers defined at uniformly spaced points in time and frequency, respectively. This feature makes the DFT ideally suited for numerical computation using the *fast Fourier transform (FFT) algorithm*. FFT algorithms are efficient because they use a greatly reduced number of arithmetic operations compared to the brute-force computation of the DFT. Basically, an FFT algorithm attains its computational efficiency by following a “divide and conquer” strategy, whereby the original DFT computation is decomposed successively into smaller DFT computations. For the case of an N -point DFT and $N = 2^L$, the FFT algorithm

requires $L = \log_2 N$ stages of computation, with each stage of the computation involving complex multiplications and additions of order N . For detailed discussion of the FFT algorithm, see Oppenheim and Schaffer (1989, Chapter 9).

21. An overview of very-high-rate digital subscriber lines (VDSL) is presented in the paper by Cioffi et al. (1999); this paper also includes a comparative discussion on VDSLs and Voiceband modems.
22. For discussion of OFDM and its applications, see Casas and Leung (1991), LeFloch et al. (1989), and Zou and Wu (1995). For tutorial notes on OFDM and an extensive list of references, see Cimini and Li (1999).
23. For detailed descriptions of phase recovery and symbol-timing recovery using classical synchronization systems, see Stiffler (1971), Lindsey (1972), and Lindsey and Simon (1973, Chapters 2 and 9).

For a modern treatment of synchronization systems with emphasis on the use of discrete-time signal processing algorithms, see Mengali and D'Andrea (1997), Meyr, Moeneclaey, and Fechtel (1998).

24. Equation (6.293) on the modified Cramér-Rao bound for phase recovery is derived in Mengali and D'Andrea (1997).
25. Saltzberg (1998) discusses how the performances of CAP and DMT are affected by channel impairments and system imperfections in the context of ADSL application. The impairments/imperfections considered therein include impulse noise, narrowband interference (e.g., RF ingress from an over-the-air AM radio transmission), timing jitter caused by imperfect synchronization, and system nonlinearities.

PROBLEMS

Amplitude-Shift Keying

- 6.1 In the on-off keying version of an ASK system, symbol 1 is represented by transmitting a sinusoidal carrier of amplitude $\sqrt{2E_b/T_b}$, where E_b is the signal energy per bit and T_b is the bit duration. Symbol 0 is represented by switching off the carrier. Assume that symbols 1 and 0 occur with equal probability.

For an AWGN channel, determine the average probability of error for this ASK system under the following scenarios:

- (a) Coherent reception.
- (b) Noncoherent reception, operating with a large value of bit energy-to-noise spectral density ratio E_b/N_0 .

Note: When x is large, the modified Bessel function of the first kind of zero order may be approximated as follows (see Appendix 3):

$$I_0(x) \approx \frac{\exp(x)}{\sqrt{2\pi x}}$$

Phase-Shift Keying

- 6.2 A PSK signal is applied to a correlator supplied with a phase reference that lies within φ radians of the exact carrier phase. Determine the effect of the phase error φ on the average probability of error of the system.
- 6.3 Consider a phase-locked loop consisting of a multiplier, loop filter, and voltage-controlled oscillator (VCO). Let the signal applied to the multiplier input be a PSK signal defined by

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

where k_b 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 6.15b. Given that message point m_1 was transmitted, show that the approximate formula of Equation (6.47) may be derived directly from Figure 6.15b.
- 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.

- 6.10 The binary data stream 01101000 is applied to a $\pi/4$ -shifted DQPSK modulator that is initially in the state ($\phi_1 = \sqrt{E}$, $\phi_2 = 0$) in Figure 6.11a. Using the relationships between input dibits and carrier-phase shifts summarized in Table 6.2, determine the phase states occupied by the modulator in response to the specified data stream.
- 6.11 Just as in an ordinary QPSK modulator, the output of a $\pi/4$ -shifted DQPSK modulator may be expressed in terms of its in-phase and quadrature components as follows:

$$s(t) = s_I(t) \cos(2\pi f_c t) - s_Q(t) \sin(2\pi f_c t)$$

Formulate the in-phase component $s_I(t)$ and quadrature component $s_Q(t)$ of the $\pi/4$ -shifted DQPSK signal. Hence, outline a scheme for the generation of $\pi/4$ -shifted DQPSK signals.

- 6.12 An interesting property of $\pi/4$ -shifted DQPSK signals is that they can be demodulated using an FM discriminator. Demonstrate the validity of this property. The FM discriminator is discussed in Chapter 2.
- 6.13 Let $\Delta\theta_k$ denote the differentially encoded phase in the $\pi/4$ -shifted DQPSK. The symbol pairs (I , Q) generated by this scheme may be defined as

$$\begin{aligned} I_k &= I_{k-1} \cos(\Delta\theta_k) - Q_{k-1} \sin(\Delta\theta_k) \\ Q_k &= I_{k-1} \sin(\Delta\theta_k) + Q_{k-1} \cos(\Delta\theta_k) \end{aligned}$$

where I_k and Q_k are the in-phase and quadrature components corresponding to the k th symbol. Show that this pair of relations can be expressed simply as

$$\begin{aligned} I_k &= \cos \theta_k \\ Q_k &= \sin \theta_k \end{aligned}$$

where θ_k is the absolute phase angle for the k th symbol.

Quadrature-Amplitude Modulation

- 6.14 Figure 6.53 shows a 240-QAM signal constellation, which may be viewed as an extended form of QAM cross constellation.
- Identify the portion of Figure 6.53 that is a QAM square constellation.
 - Build on part (a) to identify the portion of Figure 6.53 that is a QAM cross constellation.
 - Hence, identify the portion of Figure 6.53 that is an extension to QAM cross constellation.
- 6.15 Determine the transmission bandwidth reduction and average signal energy of 256-QAM, compared to 64-QAM.
- 6.16 Two passband data transmission systems are to be compared. One system uses 16-PSK, and the other uses 16-QAM. Both systems are required to produce an average probability of symbol error equal to 10^{-3} . Compare the signal-to-noise ratio requirements of these two systems.

Carrierless Amplitude/Phase Modulation (CAP)

- 6.17 The two-dimensional CAP and M -ary QAM schemes are closely related. Do the following:
- Given a QAM system, with a prescribed number of amplitude levels, derive the equivalent CAP system.
 - Perform the reverse of part (a).
- 6.18 Show that the power spectral density of a CAP signal with a total of L amplitude levels is defined by

$$S(f) = \frac{\sigma_A^2}{T} |P(f)|^2$$

where $|P(f)|$ is the magnitude spectrum of the passband in-phase pulse $p(t)$; the σ_A^2 is the variance of the complex symbols $A_i = a_i + jb_i$, which is defined by

$$\sigma_A^2 = \frac{1}{L} \sum_{i=1}^L (a_i^2 + b_i^2)$$

- 6.19 You are given the baseband raised-cosine spectrum $G(f)$ pertaining to a certain rolloff factor α . Describe a frequency-domain procedure for evaluating the passband in-phase pulse $p(t)$ and quadrature pulse $\hat{p}(t)$ that characterize the corresponding CAP signal.

Frequency-Shift Keying

- 6.20 The signal vectors s_1 and s_2 are used to represent binary symbols 1 and 0, respectively, in a coherent binary FSK system. The receiver decides in favor of symbol 1 when

$$\mathbf{x}^T s_1 > \mathbf{x}^T s_2$$

where $\mathbf{x}^T s_i$ is the inner product of the observation vector \mathbf{x} and the signal vector s_i , where $i = 1, 2$. Show that this decision rule is equivalent to the condition $x_1 > x_2$, where x_1 and x_2 are the elements of the observation vector \mathbf{x} . Assume that the signal vectors s_1 and s_2 have equal energy.

- 6.21 An FSK system transmits binary data at the rate of 2.5×10^6 bits per second. During the course of transmission, white Gaussian noise of zero mean and power spectral density 10^{-20} W/Hz is added to the signal. In the absence of noise, the amplitude of the received sinusoidal wave for digit 1 or 0 is 1 mV. Determine the average probability of symbol error for the following system configurations:

- (a) Coherent binary FSK
- (b) Coherent MSK
- (c) Noncoherent binary FSK

- 6.22 (a) In a coherent FSK system, the signals $s_1(t)$ and $s_2(t)$ representing symbols 1 and 0, respectively, are defined by

$$s_1(t), s_2(t) = A_c \cos \left[2\pi \left(f_c \pm \frac{\Delta f}{2} \right) t \right], \quad 0 \leq t \leq T_b$$

Assuming that $f_c > \Delta f$, show that the correlation coefficient of the signals $s_1(t)$ and $s_2(t)$ is approximately given by

$$\rho = \frac{\int_0^{T_b} s_1(t)s_2(t) dt}{\int_0^{T_b} s_1^2(t) dt} \approx \text{sinc}(2\Delta f T_b)$$

- (b) What is the minimum value of frequency shift Δf for which the signals $s_1(t)$ and $s_2(t)$ are orthogonal?
 - (c) What is the value of Δf that minimizes the average probability of symbol error?
 - (d) For the value of Δf obtained in part (c), determine the increase in E_b/N_0 required so that this coherent FSK system has the same noise performance as a coherent binary PSK system.
- 6.23 A binary FSK signal with *discontinuous phase* is defined by

$$s(t) = \begin{cases} \sqrt{\frac{2E_b}{T_b}} \cos \left[2\pi \left(f_c + \frac{\Delta f}{2} \right) t + \theta_1 \right] & \text{for symbol 1} \\ \sqrt{\frac{2E_b}{T_b}} \cos \left[2\pi \left(f_c - \frac{\Delta f}{2} \right) t + \theta_2 \right] & \text{for symbol 0} \end{cases}$$

where E_b is the signal energy per bit, T_b is the bit duration, and θ_1 and θ_2 are sample values of uniformly distributed random variables over the interval 0 to 2π . In effect, the two oscillators supplying the transmitted frequencies $f_c \pm \Delta f/2$ operate independently of each other. Assume that $f_c \gg \Delta f$.

- (a) Evaluate the power spectral density of the FSK signal.
 - (b) Show that for frequencies far removed from the carrier frequency f_c , the power spectral density falls off as the inverse square of frequency.
- 6.24 Set up a block diagram for the generation of Sunde's FSK signal $s(t)$ with continuous phase by using the representation given in Equation (6.104), which is reproduced here:

$$s(t) = \sqrt{\frac{2E_b}{T_b}} \cos\left(\frac{\pi t}{T_b}\right) \cos(2\pi f_c t) \mp \sqrt{\frac{2E_b}{T_b}} \sin\left(\frac{\pi t}{T_b}\right) \sin(2\pi f_c t)$$

- 6.25 Discuss the similarities between MSK and offset QPSK, and the features that distinguish them.
- 6.26 There are two ways of detecting an MSK signal. One way is to use a coherent receiver to take full account of the phase information content of the MSK signal. Another way is to use a noncoherent receiver and disregard the phase information. The second method offers the advantage of simplicity of implementation, at the expense of a degraded noise performance. By how many decibels do we have to increase the bit energy-to-noise density ratio E_b/N_0 in the second case so as to realize an average probability of symbol error equal to 10^{-5} in both cases?
- 6.27 (a) Sketch the waveforms of the in-phase and quadrature components of the MSK signal in response to the input binary sequence 1100100010.
(b) Sketch the MSK waveform itself for the binary sequence specified in part (a).
- 6.28 A nonreturn-to-zero data stream (of amplitude levels ± 1) is passed through a low-pass filter whose impulse response is defined by the Gaussian function

$$h(t) = \frac{\sqrt{\pi}}{\alpha} \exp\left(-\frac{\pi^2 t^2}{\alpha^2}\right)$$

where α is a design parameter defined in terms of the filter's 3-dB bandwidth by

$$\alpha = \sqrt{\frac{\log 2}{2}} \frac{1}{W}$$

- (a) Show that the transfer function of the filter is defined by
- $$H(f) = \exp(-\alpha^2 f^2)$$
- Hence demonstrate that the 3-dB bandwidth of the filter is indeed equal to W . You may use Table A6.3 on Fourier-transform pairs.
- (b) Show that the response of the filter to a rectangular pulse of unit amplitude and duration T centered on the origin is defined by Equation (6.135).
- 6.29 Plot the waveform of a GMSK modulator produced in response to the binary sequence 1101000, assuming the use of a gain-bandwidth product $WT_b = 0.3$. Compare your result with that of Example 6.5.
- 6.30 Summarize the similarities and differences between the standard MSK and Gaussian-filtered MSK signals.

Noncoherent Receivers

- 6.31 In Section 6.8 we derived the formula for the bit error rate of noncoherent binary FSK as a special case of noncoherent orthogonal modulation. In this problem we revisit this

issue. As before, we assume that binary symbol 1 represented by signal $s_1(t)$ is transmitted. According to the material presented in Section 6.8, we note the following:

- The random variable L_2 represented by the sample value l_2 of Equation (6.164) is Rayleigh distributed.
- The random variable L_1 represented by the sample value l_1 of Equation (6.170) is Rician distributed.

The Rayleigh and Rician distributions are discussed in Chapter 1. Using the probability distributions defined in that chapter, derive the formula of Equation (6.181) for the BER of noncoherent binary FSK.

- 6.32 Figure P6.32a shows a noncoherent receiver using a matched filter for the detection of a sinusoidal signal of known frequency but random phase, in the presence of additive white Gaussian noise. An alternative implementation of this receiver is its mechanization in the frequency domain as a *spectrum analyzer receiver*, as in Figure P6.32b, where the correlator computes the finite time autocorrelation function $R_x(\tau)$ defined by

$$R_x(\tau) = \int_0^{T-\tau} x(t)x(t+\tau) dt, \quad 0 \leq \tau \leq T$$

Show that the square-law envelope detector output sampled at time $t = T$ in Figure P6.32a is twice the spectral output of the Fourier transformer sampled at frequency $f = f_c$ in Figure P6.32b.

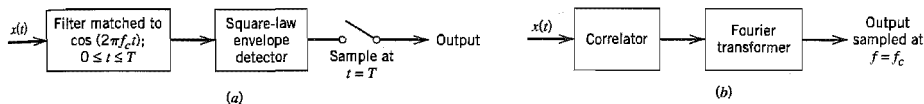


FIGURE P6.32

- 6.33 The binary sequence 1100100010 is applied to the DPSK transmitter of Figure 6.43a.
- (a) Sketch the resulting waveform at the transmitter output.
 - (b) Applying this waveform to the DPSK receiver of Figure 6.43b, show that, in the absence of noise, the original binary sequence is reconstructed at the receiver output.
- 6.34 *Differential M-ary PSK* is the M -ary extension of binary DPSK. The present phase angle θ_n of the modulator at symbol time n is determined recursively by the relation

$$\theta_n = \theta_{n-1} + \left(\frac{2\pi}{M}\right)m_n, \quad \text{modulo } 2\pi$$

where θ_{n-1} is the previous phase angle and $m_n \in \{0, 1, \dots, M-1\}$ is the present modulator input. The probability of symbol error for this M -ary modulation scheme is approximately given by

$$P_e \approx \text{erfc}\left(\sqrt{\frac{2E}{N_0}} \sin\left(\frac{\pi}{2M}\right)\right), \quad M \geq 4$$

where it is assumed that E/N_0 is large.

- (a) Determine the factor by which the transmitted energy per symbol would have to be increased for the differential M -ary PSK to attain the same probability of symbol error as coherent M -ary PSK for $M \geq 4$.
- (b) For $M = 4$, by how many decibels is differential QPSK poorer in performance than coherent QPSK?

Comparison of Digital Modulation Schemes Using a Single Carrier

- 6.35 Binary data are transmitted over a microwave link at the rate of 10^6 b/s, and the power spectral density of the noise at the receiver input is 10^{-10} W/Hz. Find the average carrier power required to maintain an average probability of error $P_e \leq 10^{-4}$ for (a) coherent binary PSK, and (b) DPSK.
- 6.36 The values of E_b/N_0 required to realize an average probability of symbol error $P_e = 10^{-4}$ using coherent binary PSK and coherent FSK (conventional) systems are equal to 7.2 and 13.5, respectively. Using the approximation

$$\operatorname{erfc}(u) \approx \frac{1}{\sqrt{\pi}u} \exp(-u^2)$$

determine the separation in the values of E_b/N_0 for $P_e = 10^{-4}$, using

- (a) Coherent binary PSK and DPSK.
 - (b) Coherent binary PSK and QPSK.
 - (c) Coherent binary FSK (conventional) and noncoherent binary FSK.
 - (d) Coherent binary FSK (conventional) and coherent MSK.
- 6.37 In Section 6.10 we compared the noise performances of coherent binary PSK, coherent binary FSK, QPSK, MSK, DPSK, and noncoherent FSK by using the bit error rate as the basis of comparison. In this problem we take a different viewpoint and use the average probability of symbol error, P_s , to do the comparison. Plot P_s versus E_b/N_0 for each of these schemes and comment on your results.
- 6.38 The *noise equivalent bandwidth* of a bandpass signal is defined as the value of bandwidth that satisfies the relation

$$2BS(f_c) = P/2$$

where $2B$ is the noise equivalent bandwidth centered around the midband frequency f_c , $S(f_c)$ is the maximum value of the power spectral density of the signal at $f = f_c$, and P is the average power of the signal. Show that the noise equivalent bandwidths of binary PSK, QPSK, and MSK are as follows:

Type of Modulation	Noise Bandwidth/Bit Rate
Binary PSK	1.0
QPSK	0.5
MSK	0.62

Note: You may use the definite integrals in Table A6.10. A discussion of noise equivalent bandwidth is presented in Appendix 2.

Voiceband Modems

- 6.39 (a) Refer to the differential encoder used in Figure 6.48a. Table 6.10 defines the phase changes induced in the V.32 modem by varying input dibits. Expand this table by including the corresponding previous and current values of the differential encoder's output. Note that for every input dibit $Q_{1,n}Q_{2,n}$, there are four possible values for the differentially encoded dibit $I_{1,n}I_{2,n}$ and likewise for its previous value $I_{1,n-1}I_{2,n-1}$.
- (b) The current quadbit applied to the V.32 modem with nonredundant coding is 0001. The previous output of the modem is 01. Find the code word output produced by the modem and its coordinates.

- 6.40 The V.32 modem standard with nonredundant coding uses a rectangular 16-QAM constellation. The model specifications are as follows:

Carrier frequency = 1,800 Hz

Symbol rate = 2,400 bauds

Data rate = 9,600 b/s

Calculate (a) the average signal-to-noise ratio, and (b) the average probability of symbol error for this modem, assuming that $E_{av}/N_0 = 20\text{dB}$.

Multichannel Line Codes

- 6.41 Consider the passband basis functions defined in Equation (6.196), where $\phi(t)$ is itself defined by Equation (6.197). Demonstrate the validity of Properties 1, 2, and 3 of these passband basis functions mentioned on pages 434 and 435.
- 6.42 The water-filling solution for the loading problem is defined by Equation (6.213) subject to the constraint of Equation (6.210). Using this pair of relations, formulate a recursive algorithm for computing the allocation of the transmit power P among the N subchannels. The algorithm should start with (a) an initial total or sum *noise-to-signal ratio* $NSR(i) = 0$ for iteration $i = 0$, and (b) the subchannels sorted in terms of those with the smallest power allocation to the largest.
- 6.43 The squared magnitude response of a linear channel, denoted by $|H(f)|^2$, is shown in Figure P6.43. Assume that the gap $\Gamma = 1$ and the noise variance $\sigma_n^2 = 1$ for all subchannels.
- (a) Derive the formulas for the optimum powers P_1 , P_2 , and P_3 allocated to the three subchannels of frequency bands $(0, W_1)$, (W_1, W_2) , and (W_2, W) .
- (b) Given that the total transmit power $P = 10$, $l_1 = 2/3$ and $l_2 = 1/3$, calculate the corresponding values of P_1 , P_2 , and P_3 .

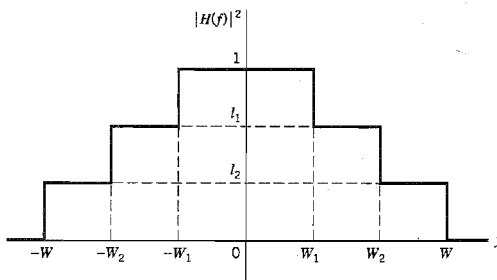


FIGURE P6.43

- 6.44 In this problem we explore the use of *singular value decomposition* (SVD) as an alternative to the discrete Fourier transform for vector coding. This approach avoids the need for a cyclic prefix, with the channel matrix being formulated as

$$\mathbf{H} = \begin{bmatrix} h_0 & h_1 & h_2 & \cdots & h_v & 0 & \cdots & 0 \\ 0 & h_0 & h_1 & \cdots & h_{v-1} & h_v & \cdots & 0 \\ \vdots & \vdots & \vdots & & \vdots & \vdots & & \vdots \\ 0 & 0 & 0 & \cdots & h_0 & h_1 & \cdots & h_v \end{bmatrix}$$

where the sequence b_0, b_1, \dots, b_ν denotes the sampled impulse response of the channel. The SVD of the matrix \mathbf{H} is defined by

$$\mathbf{H} = \mathbf{U}[\mathbf{A} : \mathbf{O}_{N,\nu}]\mathbf{V}^\dagger$$

where \mathbf{U} is an N -by- N unitary matrix and \mathbf{V} is an $(N + \nu)$ -by- $(N + \nu)$ unitary matrix; that is,

$$\mathbf{U}\mathbf{U}^\dagger = \mathbf{I}$$

$$\mathbf{V}\mathbf{V}^\dagger = \mathbf{I}$$

where \mathbf{I} is the identity matrix and the superscript \dagger denotes Hermitian transposition. The \mathbf{A} is an N -by- N diagonal matrix with singular values λ_n , $n = 1, 2, \dots, N$. The $\mathbf{O}_{N,\nu}$ is an N -by- ν matrix of zeros.

- (a) Using this decomposition, show that the N subchannels resulting from the use of vector coding are mathematically described by

$$X_n = \lambda_n A_n + W_n$$

The X_n is an element of the matrix product $\mathbf{U}^\dagger \mathbf{x}$, where \mathbf{x} is the received signal (channel output) vector. The A_n is the n th symbol $a_n + jb_n$ and W_n is a random variable due to channel noise.

- (b) Show that the signal-to-noise ratio for vector coding as described herein is given by

$$(\text{SNR})_{\text{vector coding}} = \Gamma \left(\prod_{n=1}^{N^*} \left(1 + \frac{(\text{SNR})_n}{\Gamma} \right) \right)^{1/(N+\nu)} - \Gamma$$

where N^* is the number of channels for each of which the allocated transmit power is nonnegative, $(\text{SNR})_n$ is the signal-to-noise ratio of subchannel n , and Γ is a prescribed gap.

- (c) As the block length N approaches infinity, the singular values approach the magnitudes of the channel Fourier transform. Using this result, comment on the relationship between vector coding and discrete multitone.

- 6.45 Compare the performance of DMT and CAP with respect to the following channel impairments:

- (a) Impulse noise.
(b) Narrowband interference.

Assume that (1) the DMT has a large number of subchannels, and (2) the CAP system is uncoded and its receiver uses a pair of adaptive filters for implementation.

- 6.46 Orthogonal frequency-division multiplexing may be viewed as a generalization of M -ary FSK. Validate the rationale of this statement.

Synchronization

- 6.47 Figure P6.47 shows the block diagram of a continuous-time M th power loop for phase recovery in an M -ary PSK receiver.

- (a) Show that the output of the M th power-law device contains a tone of frequency Mf_c where f_c is the original carrier.
(b) The oscillator in the phase-locked loop is set to a frequency equal to Mf_c . Justify this choice.
(c) The M th power loop suffers from a phase ambiguity problem in that it exhibits M phase ambiguities in the interval $[0, 2\pi]$. Explain how this problem arises in the M th power loop. How would you overcome the problem?

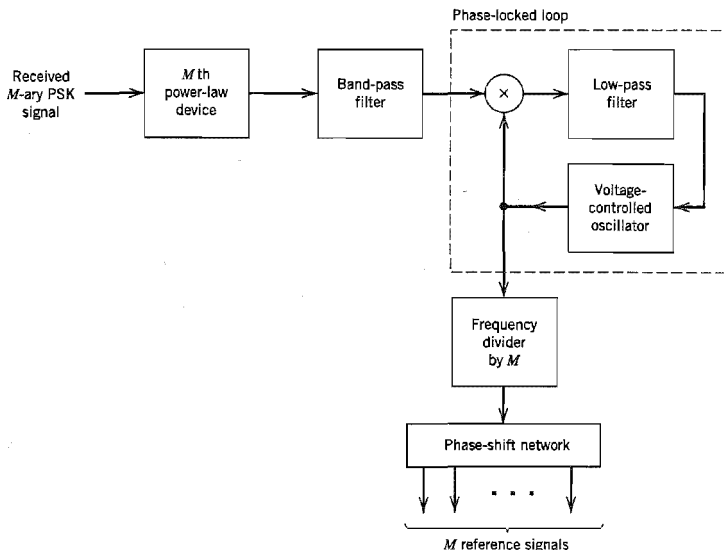


FIGURE P6.47

- 6.48 (a) In the recursive algorithm of Equation (6.272) for phase recovery, the old estimate $\hat{\theta}[n]$ and the updated estimate $\hat{\theta}[n+1]$ of the carrier phase θ are both measured in radians. Discuss the units in which the error signal $e[n]$ and step-size parameter γ are measured.
- (b) In the recursive algorithm of Equation (6.286) for symbol timing recovery, the control signals $c[n]$ and $c[n+1]$ are both dimensionless. Discuss the units in which the error signal $e[n]$ and step-size parameter γ are measured.
- 6.49 Using the definitions of Equations (6.264) and (6.265) for \tilde{x}_k and a_k , respectively, show that the exponent in the likelihood function $L(a_k, \theta, \tau)$ can be expressed as in Equation (6.273).
- 6.50 In Section 6.14 we studied a non-data-aided scheme for carrier phase recovery, based on the log-likelihood function of Equation (6.260). In this problem we explore the use of this equation for *data-aid carrier phase recovery*.
- (a) Consider a receiver designed for a linear modulation system. Given that the receiver has knowledge of a preamble of length L_0 , show that the maximum likelihood estimate of the carrier phase is defined by

$$\hat{\theta} = \arg \left\{ \sum_{k=0}^{L_0-1} a_k^* \tilde{x}(k) \right\}$$

where the preamble $\{a_k\}_{k=0}^{L_0-1}$ is a known sequence of complex symbols, and $\{\tilde{x}(k)\}_{k=0}^{L_0-1}$ is the complex envelope of the corresponding received signal.

- (b) Using the result derived in part (a), construct a block diagram for the maximum likelihood phase estimator.

Computer Experiment

6.51 The purpose of this computer experiment is to compare the effect of a dispersive channel on the waveforms generated by the following passband modulation techniques:

- (a) Binary phase-shift keying (BPSK)
- (b) Quadriphase-shift keying (QPSK)
- (c) Minimum shift keying (MSK)
- (d) Gaussian MSK with time-bandwidth product $WT_b = 0.3$

The channel consists of a band-pass *Butterworth filter* of order $2N = 10$ and 3-dB bandwidth $2B$ centered on the midband frequency f_c . The low-pass equivalent of the channel has the squared magnitude response

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

The channel bandwidth is variable so as to illustrate its effect on the filtered modulated wave.

Assuming the use of a coherent receiver, plot the waveforms of the modulated signals under (a), (b), (c) and (d) for the following channel bandwidths:

- (i) $2B = 12$ kHz
- (ii) $2B = 16$ kHz
- (iii) $2B = 20$ kHz
- (iv) $2B = 24$ kHz
- (v) $2B = 30$ kHz

Comment on your results.

Hint. To perform the computations needed for this experiment, it is advisable to perform the computations in baseband by performing the band-pass to low-pass transformation described in Appendix 2.

SPREAD-SPECTRUM MODULATION

This chapter introduces a modulation technique called spread-spectrum modulation, which is radically different from the modulation techniques that are covered in preceding chapters. In spread-spectrum modulation, channel bandwidth and transmit power are sacrificed for the sake of secure communications.

Specifically, we cover the following topics:

- ▶ *Spreading sequences in the form of pseudo-noise sequences, their properties, and methods of generation.*
- ▶ *The basic notion of spread-spectrum modulation.*
- ▶ *The two commonly used types of spread-spectrum modulation: direct sequence and frequency hopping.*

The material presented in this chapter is basic to wireless communications using code-division multiple access, which is covered in Chapter 8.

7.1 Introduction

A major issue of concern in the study of digital communications as considered in Chapters 4, 5, and 6 is that of providing for the efficient use of bandwidth and power. Notwithstanding the importance of these two primary communication resources, there are situations where it is necessary to sacrifice this efficiency in order to meet certain other design objectives. For example, the system may be required to provide a form of *secure* communication in a *hostile* environment such that the transmitted signal is not easily detected or recognized by unwanted listeners. This requirement is catered to by a class of signaling techniques known collectively as *spread-spectrum modulation*.

The primary advantage of a spread-spectrum communication system is its ability to reject *interference* whether it be the *unintentional* interference by another user simultaneously attempting to transmit through the channel, or the *intentional* interference by a hostile transmitter attempting to jam the transmission.

The definition of spread-spectrum modulation¹ may be stated in two parts:

1. Spread spectrum is a means of transmission in which the data sequence occupies a bandwidth in excess of the minimum bandwidth necessary to send it.
2. The spectrum spreading is accomplished before transmission through the use of a code that is independent of the data sequence. The same code is used in the receiver

(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-hopping 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 noise-like 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 noise-like 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 noise-like 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

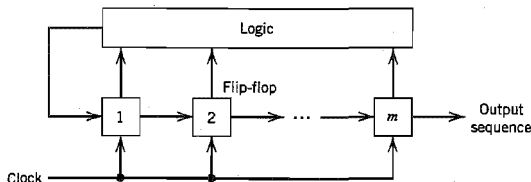


FIGURE 7.1 Feedback shift register.

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

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

where $s_0(k)$ is the input applied to the first flip-flop after the k th clock pulse. According to the configuration described in Figure 7.1, $s_0(k)$ is a Boolean function of the individual states $s_1(k), s_2(k), \dots, s_m(k)$. 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, . . .

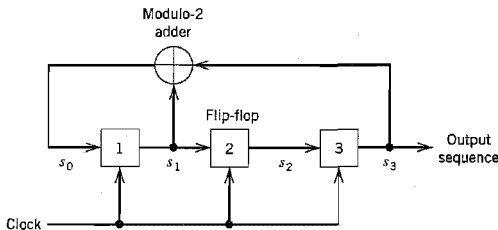


FIGURE 7.2 Maximal-length sequence generator for $m = 3$.

The output sequence (the last position of each state of the shift register) is therefore

00111010 ...

which repeats itself with period $2^3 - 1 = 7$.

Note that the choice of 100 as the initial state is arbitrary. Any of the other six permissible states could serve equally well as an initial state. The resulting output sequence would then simply experience a cyclic shift. ◀

■ PROPERTIES OF MAXIMAL-LENGTH SEQUENCES²

Maximal-length sequences have many of the properties possessed by a truly *random binary sequence*. A random binary sequence is a sequence in which the presence of binary symbol 1 or 0 is equally probable. Some properties of maximal-length sequences are as follows:

1. In each period of a maximal-length sequence, the number of 1s is always one more than the number of 0s. This property is called the *balance property*.
2. Among the runs of 1s and of 0s in each period of a maximal-length sequence, one-half the runs of each kind are of length one, one-fourth are of length two, one-eighth are of length three, and so on as long as these fractions represent meaningful numbers of runs. This property is called the *run property*. By a "run" we mean a subsequence of identical symbols (1s or 0s) within one period of the sequence. The length of this subsequence is the length of the run. For a maximal-length sequence generated by a linear feedback shift register of length m , the total number of runs is $(N + 1)/2$, where $N = 2^m - 1$.
3. The autocorrelation function of a maximal-length sequence is periodic and binary-valued. This property is called the *correlation property*.

The period of a maximum-length sequence is defined by

$$N = 2^m - 1 \quad (7.2)$$

where m is the length of the shift register. Let binary symbols 0 and 1 of the sequence be denoted by the levels -1 and $+1$, respectively. Let $c(t)$ denote the resulting waveform of the maximal-length sequence, as illustrated in Figure 7.3a for $N = 7$. The period of the waveform $c(t)$ is (based on terminology used in subsequent sections)

$$T_b = NT_c \quad (7.3)$$

where T_c is the duration assigned to symbol 1 or 0 in the maximal-length sequence. By definition, the autocorrelation function of a periodic signal $c(t)$ of period T_b is

$$R_c(\tau) = \frac{1}{T_b} \int_{-T_b/2}^{T_b/2} c(t)c(t - \tau) dt \quad (7.4)$$

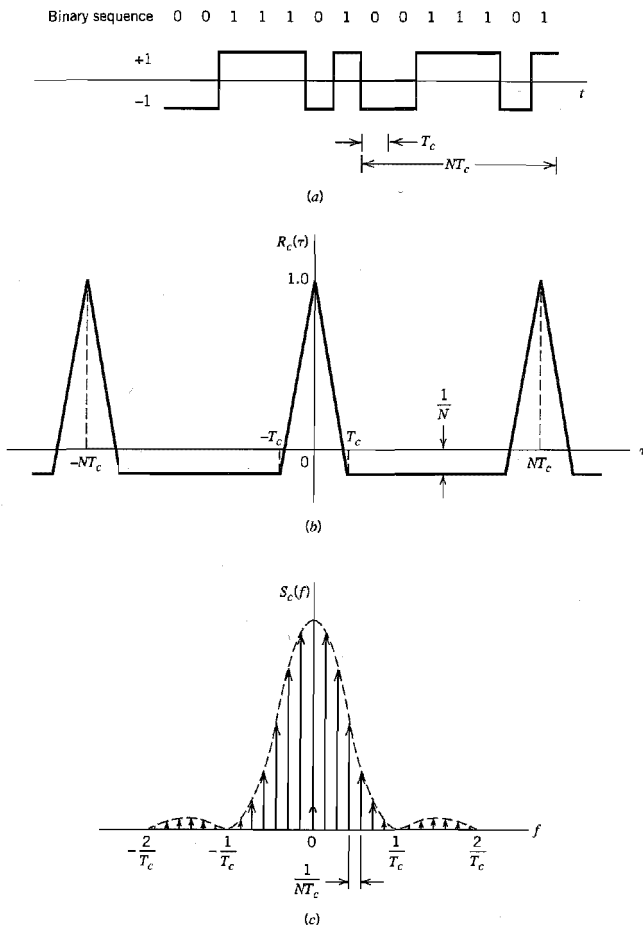


FIGURE 7.3 (a) Waveform of maximal-length sequence for length $m = 3$ or period $N = 7$. (b) Autocorrelation function. (c) Power spectral density. All three parts refer to the output of the feedback shift register of Figure 7.2.

where the lag τ lies in the interval $(-T_b/2, T_b/2)$; Equation (7.4) is a special case of Equation (1.26). Applying this formula to a maximal-length sequence represented by $c(t)$, we get

$$R_c(\tau) = \begin{cases} 1 - \frac{N+1}{NT_c} |\tau|, & |\tau| \leq T_c \\ -\frac{1}{N}, & \text{for the remainder of the period} \end{cases} \quad (7.5)$$

This result is plotted in Figure 7.3b for the case of $m = 3$ or $N = 7$.

From Fourier transform theory we know that periodicity in the time domain is transformed into uniform sampling in the frequency domain. This interplay between the time and frequency domains is borne out by the power spectral density of the maximal-length wave $c(t)$. Specifically, taking the Fourier transform of Equation (7.5), we get the sampled spectrum

$$S_c(f) = \frac{1}{N^2} \delta(f) + \frac{1+N}{N^2} \sum_{\substack{n=-\infty \\ n \neq 0}}^{\infty} \text{sinc}^2\left(\frac{n}{N}\right) \delta\left(f - \frac{n}{NT_c}\right) \quad (7.6)$$

which is plotted in Figure 7.3c for $m = 3$ or $N = 7$.

Comparing the results of Figure 7.3 for a maximal-length sequence with the corresponding results shown in Figure 1.11 for a random binary sequence, we may make the following observations:

- For a period of the maximal-length sequence, the autocorrelation function $R_c(\tau)$ is somewhat similar to that of a random binary wave.
- The waveforms of both sequences have the same envelope, $\text{sinc}^2(fT)$, for their power spectral densities. The fundamental difference between them is that whereas the random binary sequence has a continuous spectral density characteristic, the corresponding characteristic of a maximal-length sequence consists of delta functions spaced $1/NT_c$ Hz apart.

As the shift-register length m , or equivalently, the period N of the maximal-length sequence is increased, the maximal-length sequence becomes increasingly similar to the random binary sequence. Indeed, in the limit, the two sequences become identical when N is made infinitely large. However, the price paid for making N large is an increasing storage requirement, which imposes a practical limit on how large N can actually be made.

■ CHOOSING A MAXIMAL-LENGTH SEQUENCE

Now that we understand the properties of a maximal-length sequence and the fact that we can generate it using a linear feedback shift register, the key question that we need to address is: How do we find the feedback logic for a desired period N ? The answer to this

TABLE 7.1 Maximal-length sequences of shift-register lengths 2–8

Shift-Register Length, m	Feedback Taps
2*	[2, 1]
3*	[3, 1]
4	[4, 1]
5*	[5, 2], [5, 4, 3, 2], [5, 4, 2, 1]
6	[6, 1], [6, 5, 2, 1], [6, 5, 3, 2]
7*	[7, 1], [7, 3], [7, 3, 2, 1], [7, 4, 3, 2], [7, 6, 4, 2], [7, 6, 3, 1], [7, 6, 5, 2], [7, 6, 5, 4, 2, 1], [7, 5, 4, 3, 2, 1]
8	[8, 4, 3, 2], [8, 6, 5, 3], [8, 6, 5, 2], [8, 5, 3, 1], [8, 6, 5, 1], [8, 7, 6, 1], [8, 7, 6, 5, 2, 1], [8, 6, 4, 3, 2, 1]

question is to be found in the theory of error-control codes, which is covered in Chapter 10. The task of finding the required feedback logic is made particularly easy for us by virtue of the extensive tables of the necessary feedback connections for varying shift-register lengths that have been compiled in the literature. In Table 7.1, we present the sets of maximal (feedback) taps pertaining to shift-register lengths $m = 2, 3, \dots, 8$.³ Note that as m increases, the number of alternative schemes (codes) is enlarged. Also, for every set of feedback connections shown in this table, there is an “image” set that generates an identical maximal-length code, reversed in time sequence.

The particular sets identified with an asterisk in Table 7.1 correspond to *Mersenne prime length sequences*, for which the period N is a prime number.

► EXAMPLE 7.2

Consider a maximal-length sequence requiring the use of a linear feedback-shift register of length $m = 5$. For feedback taps, we select the set $[5, 2]$ from Table 7.1. The corresponding configuration of the code generator is shown in Figure 7.4a. Assuming that the initial state is 10000, the evolution of one period of the maximal-length sequence generated by this scheme is shown in Table 7.2a, where we see that the generator returns to the initial 10000 after 31 iterations; that is, the period is 31, which agrees with the value obtained from Equation (7.2).

Suppose next we select another set of feedback taps from Table 7.1, namely, $[5, 4, 2, 1]$. The corresponding code generator is thus as shown in Figure 7.4b. For the initial state 10000, we now find that the evolution of the maximal-length sequence is as shown in Table 7.2b. Here again, the generator returns to the initial state 10000 after 31 iterations, and so it should. But the maximal-length sequence generated is different from that shown in Table 7.2a.

Clearly, the code generator of Figure 7.4a has an advantage over that of Figure 7.4b, as it requires fewer feedback connections. ◀

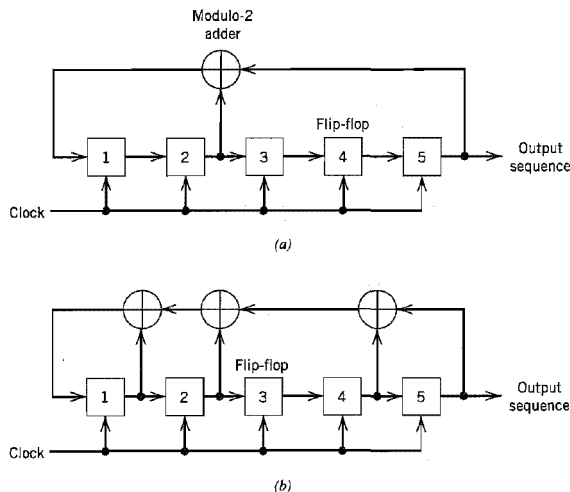


FIGURE 7.4 Two different configurations of feedback shift register of length $m = 5$. (a) Feedback connections $[5, 2]$. (b) Feedback connections $[5, 4, 2, 1]$.

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

Code: 0000101011101100011111001101001

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

7.3 A Notion of Spread Spectrum

An important attribute of spread-spectrum modulation is that it can provide protection against externally generated interfering (jamming) signals with finite power. The jamming signal may consist of a fairly powerful broadband noise or multitone waveform that is directed at the receiver for the purpose of disrupting communications. Protection against jamming waveforms is provided by purposely making the information-bearing signal occupy a bandwidth far in excess of the minimum bandwidth necessary to transmit it. This has the effect of making the transmitted signal assume a noiselike appearance so as to blend into the background. The transmitted signal is thus enabled to propagate through the channel undetected by anyone who may be listening. We may therefore think of spread spectrum as a method of “camouflaging” the information-bearing signal.

One method of widening the bandwidth of an information-bearing (data) sequence involves the use of *modulation*. Let $\{b_k\}$ denote a binary data sequence, and $\{c_k\}$ denote a pseudo-noise (PN) sequence. Let the waveforms $b(t)$ and $c(t)$ denote their respective polar nonreturn-to-zero representations in terms of two levels equal in amplitude and opposite in polarity, namely, ± 1 . We will refer to $b(t)$ as the information-bearing (data) signal, and to $c(t)$ as the PN signal. The desired modulation is achieved by applying the data signal $b(t)$ and the PN signal $c(t)$ to a product modulator or multiplier, as in Figure 7.5a. We know from Fourier transform theory that multiplication of two signals produces a signal whose spectrum equals the convolution of the spectra of the two component signals. Thus, if the message signal $b(t)$ is narrowband and the PN signal $c(t)$ is wideband, the *product (modulated) signal $m(t)$ will have a spectrum that is nearly the same as the wideband PN signal*. In other words, in the context of our present application, the PN sequence performs the role of a *spreading code*.

By multiplying the information-bearing signal $b(t)$ by the PN signal $c(t)$, each information bit is “chopped” up into a number of small time increments, as illustrated in the waveforms of Figure 7.6. These small time increments are commonly referred to as *chips*.

For *baseband* transmission, the product signal $m(t)$ represents the *transmitted signal*. We may thus express the transmitted signal as

$$m(t) = c(t)b(t) \quad (7.7)$$

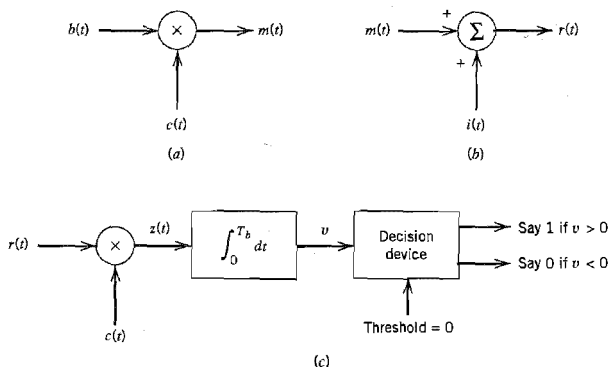


FIGURE 7.5 Idealized model of baseband spread-spectrum system. (a) Transmitter. (b) Channel. (c) Receiver.

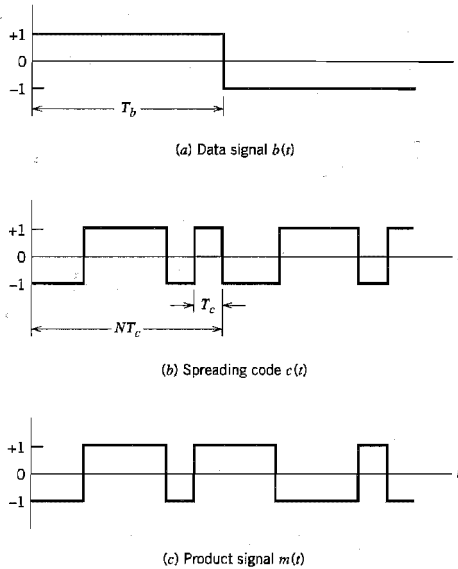


FIGURE 7.6 Illustrating the waveforms in the transmitter of Figure 7.5a.

The received signal $r(t)$ consists of the transmitted signal $m(t)$ plus an additive *interference* denoted by $i(t)$, as shown in the channel model of Figure 7.5b. Hence,

$$\begin{aligned} r(t) &= m(t) + i(t) \\ &= c(t)b(t) + i(t) \end{aligned} \quad (7.8)$$

To recover the original message signal $b(t)$, the received signal $r(t)$ is applied to a *demodulator* that consists of a multiplier followed by an integrator, and a decision device, as in Figure 7.5c. The multiplier is supplied with a locally generated PN sequence that is an exact *replica* of that used in the transmitter. Moreover, we assume that the receiver operates in perfect *synchronism* with the transmitter, which means that the PN sequence in the receiver is lined up exactly with that in the transmitter. The multiplier output in the receiver is therefore given by

$$\begin{aligned} z(t) &= c(t)r(t) \\ &= c^2(t)b(t) + c(t)i(t) \end{aligned} \quad (7.9)$$

Equation (7.9) shows that the data signal $b(t)$ is multiplied *twice* by the PN signal $c(t)$, whereas the unwanted signal $i(t)$ is multiplied only *once*. The PN signal $c(t)$ alternates between the levels -1 and $+1$, and the alternation is destroyed when it is squared; hence,

$$c^2(t) = 1 \quad \text{for all } t \quad (7.10)$$

Accordingly, we may simplify Equation (7.9) as

$$z(t) = b(t) + c(t)i(t) \quad (7.11)$$

We thus see from Equation (7.11) that the data signal $b(t)$ is reproduced at the multiplier output in the receiver, except for the effect of the interference represented by the additive term $c(t)i(t)$. Multiplication of the interference $i(t)$ by the locally generated PN signal $c(t)$ means that the spreading code will affect the interference just as it did the original signal at the transmitter. We now observe that the data component $b(t)$ is narrowband, whereas the spurious component $c(t)i(t)$ is wideband. Hence, by applying the multiplier output to a baseband (low-pass) filter with a bandwidth just large enough to accommodate the recovery of the data signal $b(t)$, most of the power in the spurious component $c(t)i(t)$ is filtered out. The effect of the interference $i(t)$ is thus significantly reduced at the receiver output.

In the receiver shown in Figure 7.5c, the low-pass filtering action is actually performed by the integrator that evaluates the area under the signal produced at the multiplier output. The integration is carried out for the bit interval $0 \leq t \leq T_b$, providing the sample value v . Finally, a decision is made by the receiver: If v is greater than the threshold of zero, the receiver says that binary symbol 1 of the original data sequence was sent in the interval $0 \leq t \leq T_b$, and if v is less than zero, the receiver says that symbol 0 was sent; if v is exactly zero the receiver makes a random guess in favor of 1 or 0.

In summary, the use of a spreading code (with pseudo-random properties) in the transmitter produces a wideband transmitted signal that appears *noiselike* to a receiver that has *no* knowledge of the spreading code. From the discussion presented in Section 7.2, we recall that (for a prescribed data rate) the longer we make the period of the spreading code, the closer will the transmitted signal be to a truly random binary wave, and the harder it is to detect. Naturally, the price we have to pay for the improved protection against interference is increased transmission bandwidth, system complexity, and processing delay. However, when our primary concern is the security of transmission, these are not unreasonable costs to pay.

7.4 Direct-Sequence Spread Spectrum with Coherent Binary Phase-Shift Keying

The spread-spectrum technique described in the previous section is referred to as *direct-sequence spread spectrum*. The discussion presented there was in the context of baseband transmission. To provide for the use of this technique in passband transmission over a satellite channel, for example, we may incorporate *coherent binary phase-shift keying* (PSK) into the transmitter and receiver, as shown in Figure 7.7. The transmitter of Figure 7.7a first converts the incoming binary data sequence $\{b_k\}$ into a polar NRZ waveform $b(t)$, which is followed by two stages of modulation. The first stage consists of a product modulator or multiplier with the data signal $b(t)$ (representing a data sequence) and the PN signal $c(t)$ (representing the PN sequence) as inputs. The second stage consists of a binary PSK modulator. The transmitted signal $x(t)$ is thus a *direct-sequence spread binary phase-shift-keyed* (DS/BPSK) signal. The phase modulation $\theta(t)$ of $x(t)$ has one of two values, 0 and π , depending on the polarities of the message signal $b(t)$ and PN signal $c(t)$ at time t in accordance with the truth table of Table 7.3.

Figure 7.8 illustrates the waveforms for the second stage of modulation. Part of the modulated waveform shown in Figure 7.6c is reproduced in Figure 7.8a; the waveform shown here corresponds to one period of the PN sequence. Figure 7.8b shows the waveform of a sinusoidal carrier, and Figure 7.8c shows the DS/BPSK waveform that results from the second stage of modulation.

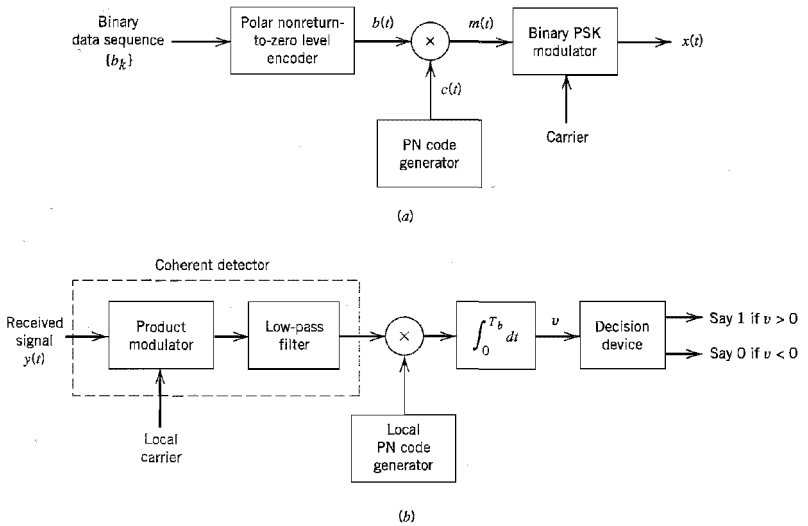


FIGURE 7.7 Direct-sequence spread coherent phase-shift keying. (a) Transmitter. (b) Receiver.

The receiver, shown in Figure 7.7b, 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 7.7a, 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

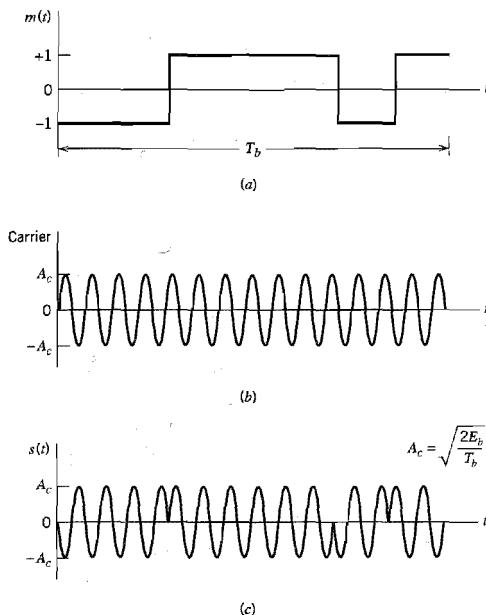


FIGURE 7.8 (a) Product signal $m(t) = c(t)b(t)$. (b) Sinusoidal carrier. (c) DS/BPSK signal.

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

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

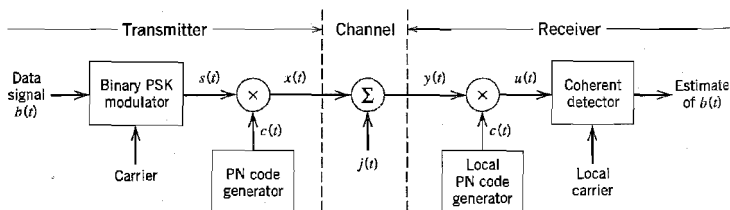


FIGURE 7.9 Model of direct-sequence spread binary PSK system.

where $s(t)$ is the binary PSK signal, and $c(t)$ is the PN signal. In the channel model included in Figure 7.9, the interfering signal is denoted by $j(t)$. This notation is chosen purposely to be different from that used for the interference in Figure 7.5b. The channel model in Figure 7.9 is passband in spectral content, whereas that in Figure 7.5b is in baseband form.

In the receiver, the received signal $y(t)$ is first multiplied by the PN signal $c(t)$ yielding an output that equals the coherent detector input $u(t)$. Thus,

$$\begin{aligned} u(t) &= c(t)y(t) \\ &= c^2(t)s(t) + c(t)j(t) \\ &= s(t) + c(t)j(t) \end{aligned} \quad (7.13)$$

In the last line of Equation (7.13), we have noted that, by design, the PN signal $c(t)$ satisfies the property described in Equation (7.10), reproduced here for convenience:

$$c^2(t) = 1 \quad \text{for all } t$$

Equation (7.13) shows that the coherent detector input $u(t)$ consists of a binary PSK signal $s(t)$ embedded in additive code-modulated interference denoted by $c(t)j(t)$. The modulated nature of the latter component forces the interference signal (jammer) to spread its spectrum such that the detection of information bits at the receiver output is afforded increased reliability.

■ SYNCHRONIZATION

For its proper operation, a spread-spectrum communication system requires that the locally generated PN sequence used in the receiver to despread the received signal be *synchronized* to the PN sequence used to spread the transmitted signal in the transmitter.⁴ A solution to the synchronization problem consists of two parts: *acquisition* and *tracking*. In acquisition, or *coarse* synchronization, the two PN codes are aligned to within a fraction of the chip in as short a time as possible. Once the incoming PN code has been acquired, tracking, or *fine* synchronization, takes place. Typically, PN acquisition proceeds in two steps. First, the received signal is multiplied by a locally generated PN code to produce a measure of *correlation* between it and the PN code used in the transmitter. Next, an appropriate *decision-rule and search strategy* is used to process the measure of correlation so obtained to determine whether the two codes are in synchronism and what to do if they are not. As for tracking, it is accomplished using phase-lock techniques very similar to those used for the local generation of coherent carrier references. The principal difference between them lies in the way in which phase discrimination is implemented.

7.5 Signal-Space Dimensionality and Processing Gain

Having developed a conceptual understanding of spread-spectrum modulation and a method for its implementation, we are ready to undertake a detailed mathematical analysis of the technique. The approach we have in mind is based on the signal-space theoretic ideas of Chapter 5. In particular, we develop signal-space representations of the transmitted signal and the interfering signal (jammer).

In this context, consider the set of orthonormal basis functions:

$$\phi_k(t) = \begin{cases} \sqrt{\frac{2}{T_c}} \cos(2\pi f_c t), & kT_c \leq t \leq (k+1)T_c \\ 0, & \text{otherwise} \end{cases} \quad (7.14)$$

$$\tilde{\phi}_k(t) = \begin{cases} \sqrt{\frac{2}{T_c}} \sin(2\pi f_c t), & kT_c \leq t \leq (k+1)T_c \\ 0, & \text{otherwise} \end{cases} \quad (7.15)$$

$$k = 0, 1, \dots, N-1$$

where T_c is the *chip duration*, and N is the number of chips per bit. Accordingly, we may describe the transmitted signal $x(t)$ for the interval of an information bit as follows:

$$\begin{aligned} x(t) &= c(t)s(t) \\ &= \pm \sqrt{\frac{2E_b}{T_b}} c(t) \cos(2\pi f_c t) \\ &= \pm \sqrt{\frac{E_b}{N}} \sum_{k=0}^{N-1} c_k \phi_k(t), \quad 0 \leq t \leq T_b \end{aligned} \quad (7.16)$$

where E_b is the signal energy per bit; the plus sign corresponds to information bit 1, and the minus sign corresponds to information bit 0. The code sequence $\{c_0, c_1, \dots, c_{N-1}\}$ denotes the PN sequence, with $c_k = \pm 1$. The transmitted signal $x(t)$ is therefore N -dimensional in that it requires a minimum of N orthonormal functions for its representation.

Consider next the representation of the interfering signal (jammer), $j(t)$. Ideally, the jammer likes to place all of its available energy in exactly the same N -dimensional signal space as the transmitted signal $x(t)$; otherwise, part of its energy goes to waste. However, the best that the jammer can hope to know is the transmitted signal bandwidth. Moreover, there is no way that the jammer can have knowledge of the signal phase. Accordingly, we may represent the jammer by the general form

$$j(t) = \sum_{k=0}^{N-1} j_k \phi_k(t) + \sum_{k=0}^{N-1} \tilde{j}_k \tilde{\phi}_k(t), \quad 0 \leq t \leq T_b \quad (7.17)$$

where

$$j_k = \int_0^{T_b} j(t) \phi_k(t) dt, \quad k = 0, 1, \dots, N-1 \quad (7.18)$$

and

$$\tilde{j}_k = \int_0^{T_b} j(t) \tilde{\phi}_k(t) dt, \quad k = 0, 1, \dots, N-1 \quad (7.19)$$

Thus the interference $j(t)$ is $2N$ -dimensional; that is, it has twice the number of dimensions required for representing the transmitted DS/BPSK signal $x(t)$. In terms of the represen-

tation given in Equation (7.17), we may express the average power of the interference $j(t)$ as follows:

$$\begin{aligned} J &= \frac{1}{T_b} \int_0^{T_b} j^2(t) dt \\ &= \frac{1}{T_b} \sum_{k=0}^{N-1} j_k^2 + \frac{1}{T_b} \sum_{k=0}^{N-1} \tilde{j}_k^2 \end{aligned} \quad (7.20)$$

Moreover, due to lack of knowledge of signal phase, the best strategy a jammer can apply is to place equal energy in the cosine and sine coordinates defined in Equations (7.18) and (7.19); hence, we may safely assume

$$\sum_{k=0}^{N-1} j_k^2 = \sum_{k=0}^{N-1} \tilde{j}_k^2 \quad (7.21)$$

Correspondingly, we may simplify Equation (7.20) as

$$J = \frac{2}{T_b} \sum_{k=0}^{N-1} j_k^2 \quad (7.22)$$

Our aim is to tie these results together by finding the signal-to-noise ratios measured at the input and output of the DS/BPSK receiver in Figure 7.9. To that end, we use Equation (7.13) to express the coherent detector output as

$$\begin{aligned} v &= \sqrt{\frac{2}{T_b}} \int_0^{T_b} u(t) \cos(2\pi f_c t) dt \\ &= v_s + v_{cj} \end{aligned} \quad (7.23)$$

where the components v_s and v_{cj} are due to the despread binary PSK signal, $s(t)$, and the spread interference, $c(t)j(t)$, respectively. These two components are defined as follows:

$$v_s = \sqrt{\frac{2}{T_b}} \int_0^{T_b} s(t) \cos(2\pi f_c t) dt \quad (7.24)$$

and

$$v_{cj} = \sqrt{\frac{2}{T_b}} \int_0^{T_b} c(t)j(t) \cos(2\pi f_c t) dt \quad (7.25)$$

Consider first the component v_s due to the signal. The despread binary PSK signal $s(t)$ equals

$$s(t) = \pm \sqrt{\frac{2E_b}{T_b}} \cos(2\pi f_c t), \quad 0 \leq t \leq T_b \quad (7.26)$$

where the plus sign corresponds to information bit 1, and the minus sign corresponds to information bit 0. Hence, assuming that the carrier frequency f_c is an integer multiple of $1/T_b$, we have

$$v_s = \pm \sqrt{E_b} \quad (7.27)$$

Consider next the component v_{cj} due to interference. Expressing the PN signal $c(t)$ in the explicit form of a sequence, $\{c_0, c_1, \dots, c_{N-1}\}$, we may rewrite Equation (7.25) in the corresponding form

$$v_{cj} = \sqrt{\frac{2}{T_b}} \sum_{k=0}^{N-1} c_k \int_{kT_c}^{(k+1)T_c} j(t) \cos(2\pi f_c t) dt \quad (7.28)$$

Using Equation (7.14) for $\phi_k(t)$, and then Equation (7.18) for the coefficient j_k , we may redefine v_{cj} as

$$\begin{aligned} v_{cj} &= \sqrt{\frac{T_c}{T_b}} \sum_{k=0}^{N-1} c_k \int_0^{T_b} j(t) \phi_k(t) dt \\ &= \sqrt{\frac{T_c}{T_b}} \sum_{k=0}^{N-1} c_k j_k \end{aligned} \quad (7.29)$$

We next approximate the PN sequence as an *independent and identically distributed (i.i.d.) binary sequence*. We emphasize the implication of this approximation by recasting Equation (7.29) in the form

$$V_{cj} = \sqrt{\frac{T_c}{T_b}} \sum_{k=0}^{N-1} C_k j_k \quad (7.30)$$

where V_{cj} and C_k are random variables with sample values v_{cj} and c_k , respectively. In Equation (7.30), the jammer is assumed to be fixed. With the C_k treated as i.i.d. random variables, we find that the probability of the event $C_k = \pm 1$ is

$$P(C_k = 1) = P(C_k = -1) = \frac{1}{2} \quad (7.31)$$

Accordingly, the mean of the random variable V_{cj} is zero since, for fixed k , we have

$$\begin{aligned} E[C_k j_k | j_k] &= j_k P(C_k = 1) - j_k P(C_k = -1) \\ &= \frac{1}{2} j_k - \frac{1}{2} j_k \\ &= 0 \end{aligned} \quad (7.32)$$

For a fixed vector \mathbf{j} , representing the set of coefficients j_0, j_1, \dots, j_{N-1} , the variance of V_{cj} is given by

$$\text{var}[V_{cj} | \mathbf{j}] = \frac{1}{N} \sum_{k=0}^{N-1} j_k^2 \quad (7.33)$$

Since the *spread factor* $N = T_b/T_c$, we may use Equation (7.22) to express this variance in terms of the average interference power J as

$$\text{var}[V_{cj} | \mathbf{j}] = \frac{JT_c}{2} \quad (7.34)$$

Thus the random variable V_{cj} has zero mean and variance $JT_c/2$.

From Equation (7.27), we note that the signal component at the coherent detector output (during each bit interval) equals $\pm\sqrt{E_b}$, where E_b is the signal energy per bit. Hence, the peak instantaneous power of the signal component is E_b . Accordingly, we may define

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

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

The average signal power at the receiver input equals E_b/T_b . We thus define an *input signal-to-noise ratio* as

$$(\text{SNR})_I = \frac{E_b/T_b}{J} \quad (7.36)$$

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

$$(\text{SNR})_O = \frac{2T_b}{T_c} (\text{SNR})_I \quad (7.37)$$

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

$$\text{PG} = \frac{T_b}{T_c} \quad (7.38)$$

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:

$$10 \log_{10}(\text{SNR})_O = 10 \log_{10}(\text{SNR})_I + 3 + 10 \log_{10}(\text{PG}) \text{ dB} \quad (7.39)$$

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_{ej} produced by external interference denote the sample value of a random variable V_{ej} . Then, from Equations (7.23) and (7.27) we deduce that

$$V = \pm\sqrt{E_b} + V_{ej} \quad (7.40)$$

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

find that (because of the symmetric nature of the problem) the average probability of error P_e is the same as the conditional probability of (say) the receiver making a decision in favor of symbol 1, given that symbol 0 was sent. That is,

$$\begin{aligned} P_e &= P(V > 0 | \text{symbol 0 was sent}) \\ &= P(V_{cj} > \sqrt{E_b}) \end{aligned} \quad (7.41)$$

Naturally, the probability of error P_e depends on the random variable V_{cj} defined by Equation (7.30). According to this definition, V_{cj} is the sum of N identically distributed random variables. Hence, from the *central limit theorem*, we deduce that for large N , the random variable V_{cj} assumes a Gaussian distribution. Indeed, the spread factor or PN sequence length N is typically large in the direct-sequence spread-spectrum systems encountered in practice, under which condition the application of the central limit theorem is justified.

Earlier we evaluated the mean and variance of V_{cj} ; see Equations (7.32) and (7.34). We may therefore state that the equivalent noise component V_{cj} contained in the coherent detector output may be approximated as a Gaussian random variable with zero mean and variance $JT_c/2$, where J is the average interference power and T_c is the chip duration. With this approximation at hand, we may then proceed to calculate the probability of the event $V_{cj} > \sqrt{E_b}$, and thus express the average probability of error in accordance with Equation (7.41) as

$$P_e \approx \frac{1}{2} \operatorname{erfc} \left(\sqrt{\frac{E_b}{JT_c}} \right) \quad (7.42)$$

This simple formula, which invokes the Gaussian assumption, is appropriate for DS/BPSK binary systems with large spread factor N .

■ ANTIJAM CHARACTERISTICS

It is informative to compare Equation (7.42) with the formula for the average probability of error for a coherent binary PSK system reproduced here for convenience of presentation [see Equation (6.20)]

$$P_e = \frac{1}{2} \operatorname{erfc} \left(\sqrt{\frac{E_b}{N_0}} \right) \quad (7.43)$$

Based on this comparison, we see that insofar as the calculation of bit error rate in a direct-sequence spread binary PSK system is concerned, the interference may be treated as wideband noise of power spectral density $N_0/2$, defined by

$$\frac{N_0}{2} = \frac{JT_c}{2} \quad (7.44)$$

This relation is simply a restatement of an earlier result given in Equation (7.34).

Since the signal energy per bit $E_b = PT_b$, where P is the average signal power and T_b is the bit duration, we may express the signal energy per bit-to-noise spectral density ratio as

$$\frac{E_b}{N_0} = \left(\frac{T_b}{T_c} \right) \left(\frac{P}{J} \right) \quad (7.45)$$

Using the definition of Equation (7.38) for the processing gain PG we may reformulate this result as

$$\frac{J}{P} = \frac{PG}{E_b/N_0} \quad (7.46)$$

The ratio J/P is termed the *jamming margin*. Accordingly, the jamming margin and the processing gain, both expressed in decibels, are related by

$$(\text{Jamming margin})_{\text{dB}} = (\text{Processing gain})_{\text{dB}} - 10 \log_{10} \left(\frac{E_b}{N_0} \right)_{\text{min}} \quad (7.47)$$

where $(E_b/N_0)_{\text{min}}$ is the minimum value needed to support a 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, $T_c = 1$ μ s

Hence, using Equation (7.38) we find that the processing gain is

$$PG = 4095$$

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

some jammers of concern, in which case we have to resort to other methods. One such alternative method is to force the jammer to cover a wider spectrum by *randomly hopping* the data-modulated carrier from one frequency to the next. In effect, the spectrum of the transmitted signal is spread *sequentially* rather than instantaneously; the term “sequentially” refers to the pseudo-random-ordered sequence of frequency hops.

The type of spread spectrum in which the carrier hops randomly from one frequency to another is called *frequency-hop (FH) spread spectrum*. A common modulation format for FH systems is that of *M*-ary *frequency-shift keying* (MFSK). The combination of these two techniques is referred to simply as FH/MFSK. (A description of *M*-ary FSK is presented in Chapter 6.)

Since frequency hopping does not cover the entire spread spectrum instantaneously, we are led to consider the rate at which the hops occur. In this context, we may identify two basic (technology-independent) characterizations of frequency hopping:

1. *Slow-frequency hopping*, in which the *symbol rate* R_s of the MFSK signal is an integer multiple of the *hop rate* R_h . That is, several symbols are transmitted on each frequency hop.
2. *Fast-frequency hopping*, in which the hop rate R_h is an integer multiple of the MFSK symbol rate R_s . That is, the carrier frequency will change or hop several times during the transmission of one symbol.

Obviously, slow-frequency hopping and fast-frequency hopping are the converse of one another. In the following, these two characterizations of frequency hopping are considered in turn.

■ SLOW-FREQUENCY HOPPING

Figure 7.10a shows the block diagram of an FH/MFSK transmitter, which involves *frequency modulation* followed by *mixing*. First, the incoming binary data are applied to an *M*-ary FSK modulator. The resulting modulated wave and the output from a digital *frequency synthesizer* are then applied to a mixer that consists of a multiplier followed by a band-pass filter. The filter is designed to select the sum frequency component resulting from the multiplication process as the transmitted signal. In particular, successive *k*-bit segments of a PN sequence drive the frequency synthesizer, which enables the carrier frequency to hop over 2^k distinct values. On a single hop, the bandwidth of the transmitted signal is the same as that resulting from the use of a conventional MFSK with an alphabet of $M = 2^k$ orthogonal signals. However, for a complete range of 2^k frequency hops, the transmitted FH/MFSK signal occupies a much larger bandwidth. Indeed, with present-day technology, FH bandwidths on the order of several GHz are attainable, which is an order of magnitude larger than that achievable with direct-sequence spread spectra. An implication of these large FH bandwidths is that coherent detection is possible only within each hop, because frequency synthesizers are unable to maintain phase coherence over successive hops. Accordingly, most frequency-hop spread-spectrum communication systems use noncoherent *M*-ary modulation schemes.

In the receiver depicted in Figure 7.10b, the frequency hopping is first removed by *mixing* (down-converting) the received signal with the output of a local frequency synthesizer that is synchronously controlled in the same manner as that in the transmitter. The resulting output is then band-pass filtered, and subsequently processed by a *noncoherent M*-ary FSK detector. To implement this *M*-ary detector, we may use a bank of *M* noncoherent matched filters, each of which is matched to one of the MFSK tones. (Noncoherent matched filters are described in Chapter 6.) An estimate of the original symbol transmitted is obtained by selecting the largest filter output.

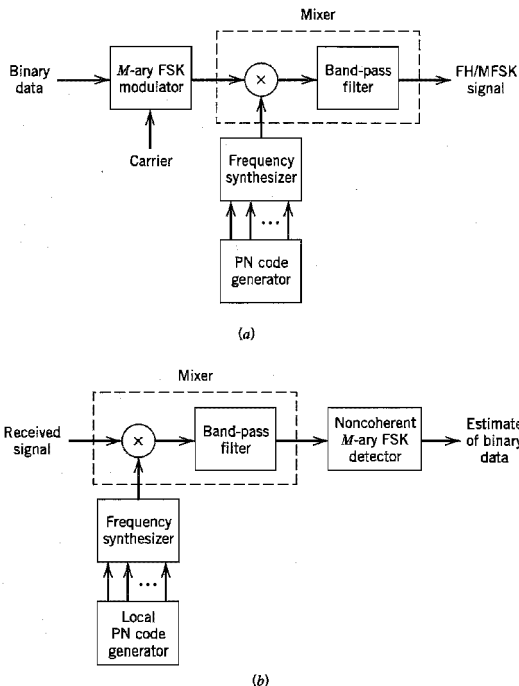


FIGURE 7.10 Frequency-hop spread M -ary frequency-shift keying. (a) Transmitter. (b) Receiver.

An individual FH/MFSK tone of shortest duration is referred to as a *chip*; this terminology should not be confused with that used in Section 7.4 describing DS/BPSK. The *chip rate*, R_c , for an FH/MFSK system is defined by

$$R_c = \max(R_b, R_s) \quad (7.48)$$

where R_b is the *hop rate*, and R_s is the *symbol rate*.

A slow FH/MFSK signal is characterized by having multiple symbols transmitted per hop. Hence, each symbol of a slow FH/MFSK signal is a chip. Correspondingly, in a slow FH/MFSK system, the bit rate R_b of the incoming binary data, the symbol rate R_s of the MFSK signal, the chip rate R_c , and the hop rate R_h are related by

$$R_c = R_s = \frac{R_b}{K} \geq R_h \quad (7.49)$$

where $K = \log_2 M$.

At each hop, the MFSK tones are separated in frequency by an integer multiple of the chip rate $R_c = R_s$, ensuring their orthogonality. The implication of this condition is that any transmitted symbol will not produce any crosstalk in the other $M - 1$ noncoherent matched filters constituting the MFSK detector of the receiver in Figure 7.10b. By “crosstalk” we mean the spillover from one filter output into an adjacent one. The resulting performance of the slow FH/MFSK system is the same as that for the noncoherent detection

of conventional (unhopped) MFSK signals in additive white Gaussian noise. Thus the interfering (jamming) signal has an effect on the FH/MFSK receiver, in terms of average probability of symbol error, equivalent to that of additive white Gaussian noise on a conventional noncoherent M -ary FSK receiver experiencing no interference. On the basis of this equivalence, we may use Equation (6.140) for approximate evaluation of the probability of symbol error in the FH/MFSK system.

Assuming that the jammer decides to spread its average power J over the entire frequency-hopped spectrum, the jammer's effect is equivalent to an AWGN with power spectral density $N_0/2$, where $N_0 = J/W_c$ and W_c is the FH bandwidth. The spread-spectrum system is thus characterized by the *symbol energy-to-noise spectral density ratio*:

$$\frac{E}{N_0} = \frac{P/J}{W_c/R_s} \quad (7.50)$$

where the ratio P/J is the reciprocal of the jamming margin. The other ratio in the denominator of Equation (7.50) is the processing gain of the slow FH/MFSK system, which is defined by

$$\begin{aligned} PG &= \frac{W_c}{R_s} \\ &= 2^k \end{aligned} \quad (7.51)$$

That is, the processing gain (expressed in decibels) is equal to $10 \log_{10} 2^k \approx 3k$, where k is the length of the PN segment employed to select a frequency hop.

This result assumes that the jammer spreads its power over the entire FH spectrum. However, if the jammer decides to concentrate on just a few of the hopped frequencies, then the processing gain realized by the receiver would be less than $3k$ decibels.

▼ EXAMPLE 7.4

Figure 7.11a illustrates the variation of the frequency of a slow FH/MFSK signal with time for one complete period of the PN sequence. The period of the PN sequence is $2^4 - 1 = 15$. The FH/MFSK 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, the carrier is hopped to a new frequency after transmitting two symbols or equivalently, four information bits. Figure 7.11a also includes the input binary data, and the PN sequence controlling the selection of FH carrier frequency. It is noteworthy that although there are eight distinct frequencies available for hopping, only three of them are utilized by the PN sequence.

Figure 7.11b shows the variation of the dehopped frequency with time. This variation is recognized to be the same as that of a conventional MFSK signal produced by the given input data. ◀

■ FAST-FREQUENCY HOPPING

A fast FH/MFSK system differs from a slow FH/MFSK system in that there are multiple hops per M -ary symbol. Hence, in a fast FH/MFSK system, each hop is a chip. In general,

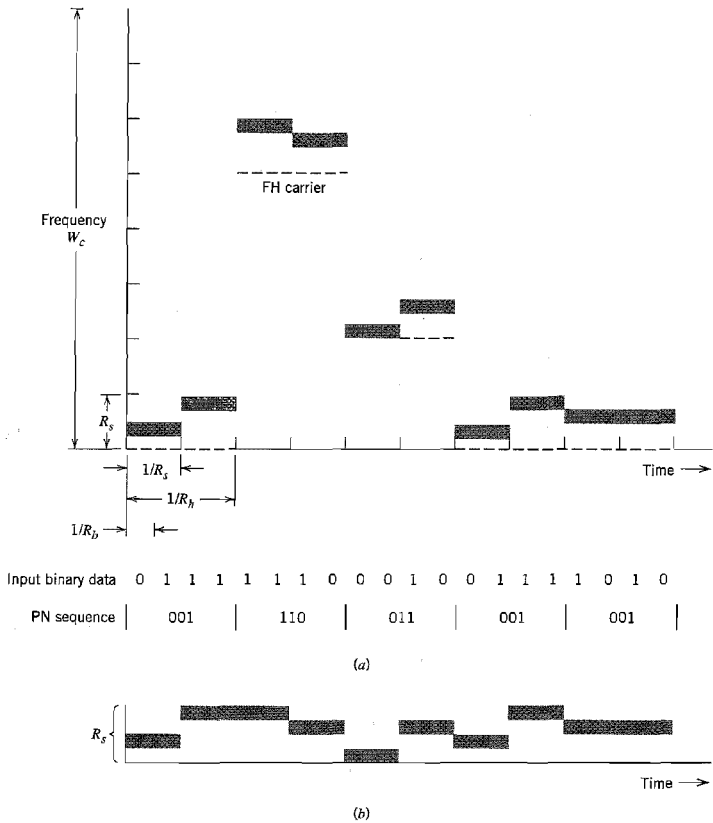


FIGURE 7.11 Illustrating slow-frequency hopping. (a) Frequency variation for one complete period of the PN sequence. (b) Variation of the dehopped frequency with time.

fast-frequency hopping is used to defeat a smart jammer's tactic that involves two functions: measurements of the spectral content of the transmitted signal, and retuning of the interfering signal to that portion of the frequency band. Clearly, to overcome the jammer, the transmitted signal must be hopped to a new carrier frequency *before* the jammer is able to complete the processing of these two functions.

For data recovery at the receiver, noncoherent detection is used. However, the detection procedure is quite different from that used in a slow FH/MFSK receiver. In particular, two procedures may be considered:

1. For each FH/MFSK symbol, separate decisions are made on the K frequency-hop chips received, and a simple rule based on *majority vote* is used to make an estimate of the dehopped MFSK symbol.
2. For each FH/MFSK symbol, likelihood functions are computed as functions of the total signal received over K chips, and the largest one is selected.

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 7.12a 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 the same as the bit rate R_b . After each chip, the carrier frequency of the transmitted MFSK signal is hopped to a different value, except for few occasions when the k -chip segment of the PN sequence repeats itself.

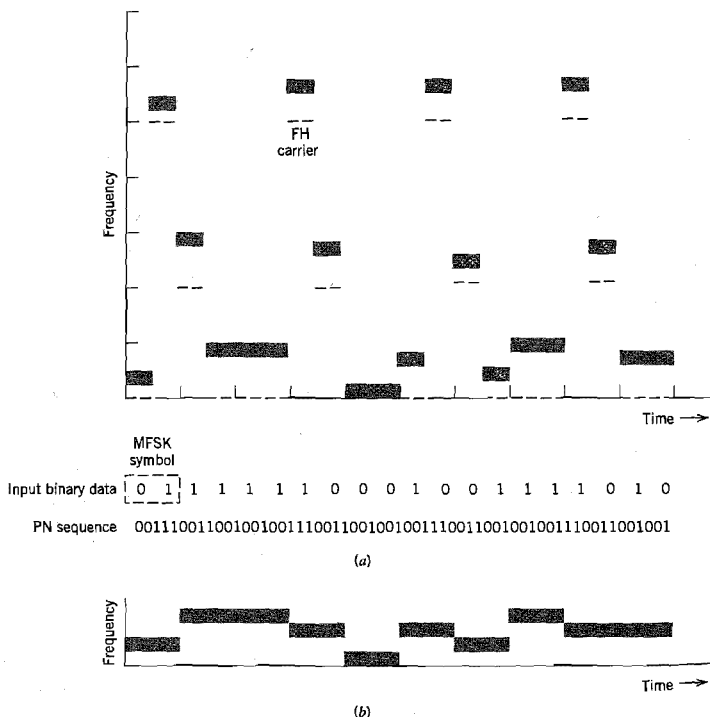



FIGURE 7.12 Illustrating fast-frequency hopping. (a) Variation of the transmitter frequency with time. (b) Variation of the dehopped frequency with time.

Figure 7.12b depicts the time variation of the frequency of the dehopped MFSK signal, which is the same as that in Example 7.4. 

7.8 Computer Experiments: Maximal-Length and Gold Codes

Code-division multiplexing (CDM) provides an alternative to the traditional methods of frequency-division multiplexing (FDM) and time-division multiplexing (TDM). It does not require the bandwidth allocation of FDM (discussed in Chapter 2) nor the time synchronization needed in TDM (discussed in Chapter 3). Rather, users of a common channel are permitted access to the channel through the assignment of a “spreading code” to each individual user under the umbrella of spread-spectrum modulation. The purpose of this computer experiment is to study a certain class of spreading codes for CDM systems that provide a satisfactory performance.

In an ideal CDM system, the cross-correlation between any two users of the system is zero. For this ideal condition to be realized, we require that the cross-correlation function between the spreading codes assigned to any two users of the system be zero for all cyclic shifts. Unfortunately, ordinary PN sequences do not satisfy this requirement because of their relatively poor cross-correlation properties.

As a remedy for this shortcoming of ordinary PN sequences, we may use a special class of PN sequences called *Gold sequences (codes)*,⁵ the generation of which is embodied in the following theorem:

Let $g_1(X)$ and $g_2(X)$ be a preferred pair of primitive polynomials of degree n whose corresponding shift registers generate maximal-length sequences of period $2^n - 1$ and whose cross-correlation function has a magnitude less than or equal to

$$2^{(n+1)/2} + 1 \quad \text{for } n \text{ odd} \quad (7.52)$$

or

$$2^{(n+2)/2} + 1 \quad \text{for } n \text{ even and } n \neq 0 \bmod 4 \quad (7.53)$$

Then the shift register corresponding to the product polynomial $g_1(X) \cdot g_2(X)$ will generate $2^n + 1$ different sequences, with each sequence having a period of $2^n - 1$, and the cross-correlation between any pair of such sequences satisfying the preceding condition.

Hereafter, this theorem is referred to as *Gold's theorem*.

To understand Gold's theorem, we need to define what we mean by a primitive polynomial. Consider a polynomial $g(X)$ defined over a *binary field* (i.e., a finite set of two elements, 0 and 1, which is governed by the rules of binary arithmetic). The polynomial $g(X)$ is said to be an *irreducible polynomial* if it cannot be factored using any polynomials from the binary field. An irreducible polynomial $g(X)$ of degree m is said to be a *primitive polynomial* if the smallest integer m for which the polynomial $g(X)$ divides the factor $X^n + 1$ is $n = 2^m - 1$. Further discussion of this topic is deferred to Chapter 8; in particular, see Example 8.3.

Experiment 1. Correlation Properties of PN Sequences

Consider a pair of shift registers for generating two PN sequences of period $2^7 - 1 = 127$. One feedback shift register has the feedback taps [7, 1] and the other one has the feedback taps [7, 6, 5, 4]. Both sequences have the same autocorrelation function shown in Figure 7.13a, which follows readily from the definition presented in Equation (7.5).

However, the calculation of the cross-correlation function between PN sequences is a more difficult proposition, particularly for large n . To perform this calculation, we resort to the use of computer simulation for varying cyclic shift τ inside the interval $0 < \tau \leq 2^n - 1$. The results of this computation are presented in Figure 7.13b. This figure confirms the poor cross-correlation property of PN sequences compared to their autocorrelation function. The magnitude of the cross-correlation function exceeds 40.

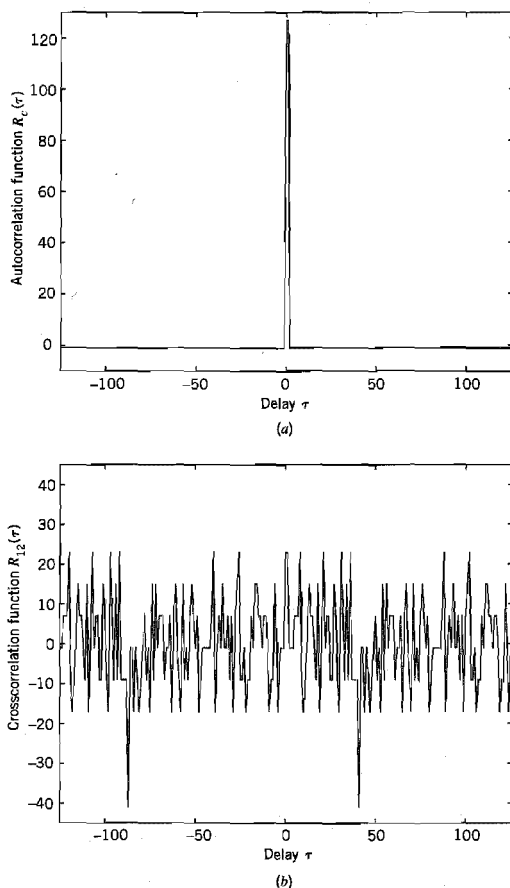


FIGURE 7.13 (a) Autocorrelation function $R_c(\tau)$, and (b) cross-correlation function $R_{12}(\tau)$ of the two PN sequences [7, 1] and [7, 6, 5, 4].

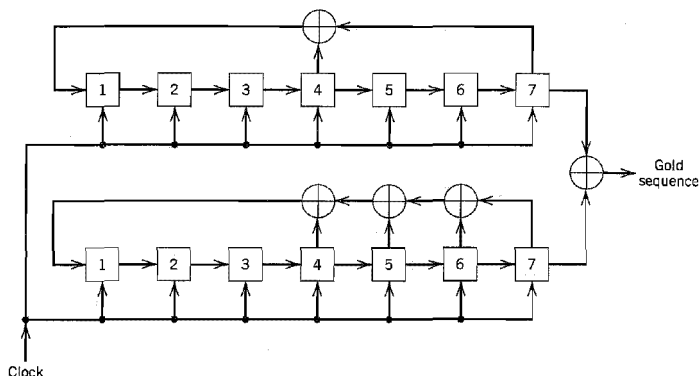


FIGURE 7.14 Generator for a Gold sequence of period $2^7 - 1 = 127$.

Experiment 2. Correlation Properties of Gold Sequences

For our next experiment, we consider Gold sequences with period $2^7 - 1 = 127$. To generate such a sequence for $n = 7$ we need a preferred pair of PN sequences that satisfy Equation (7.52) (n odd), as shown by

$$2^{(n+1)/2} + 1 = 2^4 + 1 = 17$$

This requirement is satisfied by the PN sequences with feedback taps $[7, 4]$ and $[7, 6, 5, 4]$. The Gold-sequence generator is shown in Figure 7.14 that involves the modulo-2 addition of these two sequences. According to Gold's theorem, there are a total of

$$2^n + 1 = 2^7 + 1 = 129$$

sequences that satisfy Equation (7.52). The cross-correlation between any pair of such sequences is shown in Figure 7.15, which is indeed in full accord with Gold's theorem. In particular, the magnitude of the cross-correlation is less than or equal to 17.

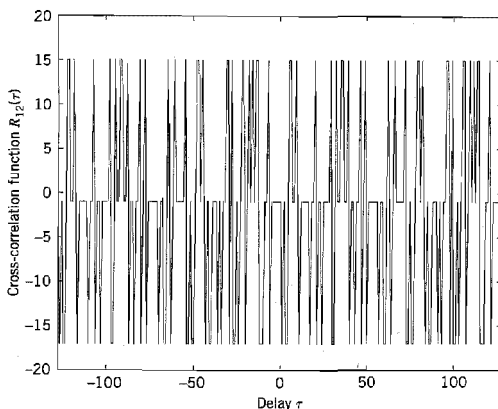


FIGURE 7.15 Cross-correlation function $R_{12}(\tau)$ of a pair of Gold sequences based on the two PN sequences $[7, 4]$ and $[7, 6, 5, 4]$.

7.9 Summary and Discussion

Direct-sequence M-ary phase shift keying (DS/MPSK) and frequency-hop M-ary frequency shift-keying (FH/MFSK) represent two principal categories of spread-spectrum communications. Both of them rely on the use of a pseudo-noise (PN) sequence, which is applied differently in the two categories.

In a DS/MPSK system, the PN sequence makes the transmitted signal assume a noiselike appearance by spreading its spectrum over a broad range of frequencies simultaneously. For the phase-shift keying, we may use binary PSK (i.e., $M = 2$) with a single carrier. Alternatively, we may use QPSK (i.e., $M = 4$), in which case the data are transmitted using a pair of carriers in phase quadrature. (Both PSK and QPSK are discussed in Section 6.3.) The usual motivation for using QPSK is to provide for improved bandwidth efficiency. In a spread-spectrum system, bandwidth efficiency is usually not of prime concern. Rather, the use of QPSK is motivated by the fact that it is less sensitive to some types of interference (jamming).

In an FH/MFSK system, the PN sequence makes the carrier hop over a number of frequencies in a pseudo-random manner, with the result that the spectrum of the transmitted signal is spread in a sequential manner.

Naturally, the direct-sequence and frequency-hop spectrum-spreading techniques may be employed in a single system. The resulting system is referred to as *hybrid DS/FH spread-spectrum system*. The reason for seeking a hybrid approach is that advantages of both the direct-sequence and frequency-hop spectrum-spreading techniques are realized in the same system.

A discussion of spread-spectrum communications would be incomplete without some reference to jammer waveforms. The jammers encountered in practice include the following types:

1. *The barrage noise jammer*, which consists of band-limited white Gaussian noise of high average power. The barrage noise jammer is a brute-force jammer that does not exploit any knowledge of the antijam communication system except for its spread bandwidth.
2. *The partial-band noise jammer*, which consists of noise whose total power is evenly spread over some frequency band that is a subset of the total spread bandwidth. Owing to the smaller bandwidth, the partial-band noise jammer is easier to generate than the barrage noise jammer.
3. *The pulsed noise jammer*, which involves transmitting wideband noise of power

$$J_{\text{peak}} = \frac{J}{p}$$

for a fraction p of the time, and nothing for the remaining fraction $1 - p$ of the time. The average noise power equals J .

4. *The single-tone jammer*, which consists of a sinusoidal wave whose frequency lies inside the spread bandwidth; as such, it is the easiest of all jamming signals to generate.
5. *The multitone jammer*, which is the tone equivalent of the partial-band noise jammer.

In addition to these five, many other kinds of jamming waveforms occur in practice. In any event, there is no single jamming waveform that is worst for all spread-spectrum

systems, and there is no single spread-spectrum system that is best against all possible jamming waveforms.

NOTES AND REFERENCES

1. The definition of spread-spectrum modulation presented in the Introduction is adapted from Pickholtz, Schilling, and Milstein (1982). This paper presents a tutorial review of the theory of spread-spectrum communications.

For introductory papers on the subject, see Viterbi (1979), and Cook and Marsh (1983). For books on the subject, see Dixon (1984), Holmes (1982), Ziemer and Peterson (1985, pp. 327–649), Cooper and McGillem (1986, pp. 269–411), and Simon, Omura, Scholtz, and Levitt (1985, Volumes I, II, and III). The three-volume book by Simon et al. is the most exhaustive treatment of spread-spectrum communications available in the open literature. The development of spread-spectrum communications dates back to about the mid-1950s. For a historical account of these techniques, see Scholtz (1982). This latter paper traces the origins of spread-spectrum communications back to the 1920s. Much of the historical material presented in this paper is reproduced in Chapter 2, Volume I, of the book by Simon et al.

The book edited by Tantaratana and Ahmed (1998) includes introductory and advanced papers on wireless applications of spread-spectrum modulation. The papers are grouped into the following categories: spread-spectrum technology, cellular mobile systems, satellite communications, wireless local area networks, and global positioning systems (GPS).

2. For further details on maximal-length sequences, see Golomb (1964, pp. 1–32), Simon, Omura, Scholtz, and Levitt (1985, pp. 283–295), and Peterson and Weldon (1972). The last reference includes an extensive list of polynomials for generating maximal-length sequences; see also Dixon (1984). For a tutorial paper on pseudo-noise sequences, see Sarwate and Pursley (1980).
3. Table 7.1 is extracted from the book by Dixon (1984, pp. 81–83), where feedback connections of maximal-length sequences are tabulated for shift-register length m extending up to 89.
4. For detailed discussion of the synchronization problem in spread-spectrum communications, see Ziemer and Peterson (1985, Chapters 9 and 10) and Simon et al. (1985, Volume III).
5. The original papers on Gold sequences are Gold (1967, 1968). A detailed discussion of Gold sequences is presented in Holmes (1982).

PROBLEMS

Pseudo-Noise Sequences

- 7.1 A pseudo-noise (PN) sequence is generated using a feedback shift register of length $m = 4$. The chip rate is 10^7 chips per second. Find the following parameters:
 - (a) PN sequence length.
 - (b) Chip duration of the PN sequence.
 - (c) PN sequence period.

- 7.2 Figure P7.2 shows a four-stage feedback shift register. The initial state of the register is 1000. Find the output sequence of the shift register.

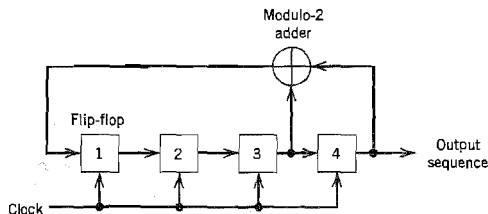


FIGURE P7.2

- 7.3 For the feedback shift register given in Problem 7.2, demonstrate the balance property and run property of a PN sequence. Also, calculate and plot the autocorrelation function of the PN sequence produced by this shift register.
- 7.4 Referring to Table 7.1, develop the maximal-length codes for the three feedback configurations [6, 1], [6, 5, 2, 1], and [6, 5, 3, 2], whose period is $N = 63$.
- 7.5 Figure P7.5 shows the modular multitap version of the linear feedback shift-register shown in Figure 7.4b. Demonstrate that the PN sequence generated by this scheme is exactly the same as that described in Table 7.2b.

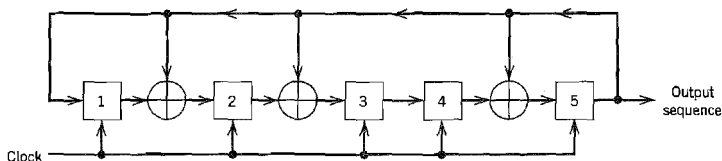


FIGURE P7.5

Direct Sequence/Phase-Shift Keying System

- 7.6 Show that the truth table given in Table 7.3 can be constructed by combining the following two steps:
- The message signal $b(t)$ and PN signal $c(t)$ are added modulo-2.
 - Symbols 0 and 1 at the modulo-2 adder output are represented by phase shifts of 0 and 180 degrees, respectively.
- 7.7 A single-tone jammer

$$j(t) = \sqrt{2J} \cos(2\pi f_j t + \theta)$$

is applied to a DS/BPSK system. The N -dimensional transmitted signal $x(t)$ is described by Equation (7.16). Find the $2N$ coordinates of the jammer $j(t)$.

- 7.8 The processing gain of a spread-spectrum system may be expressed as the ratio of the spread bandwidth of the transmitted signal to the despread bandwidth of the received signal. Justify this statement for the DS/BPSK system.

- 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:
- Processing gain.
 - 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.
- Compute the autocorrelation function of these two sequences, and their cross-correlation function.
 - 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
- 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].
 - Repeat the computation for the PN sequence with feedback taps [5, 2] and the PN sequence with feedback taps [5, 4, 2, 1].
 - 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.

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*

(i.e., total freedom of location is permitted), hence the interest in their use for local area networks (i.e., data networks confined to buildings up to a few kilometers in size) due to significant advantages over conventional cabling: elimination of wiring and rewiring, flexibility of creating new communication services, and mobility of users.

The radio propagation channel characterizing wireless communications deviates from the idealized AWGN channel model due to the presence of *multipath*, which is a non-Gaussian form of signal-dependent phenomenon that arises because of reflections of the transmitted signal from fixed and moving objects. The presence of multipath raises practical difficulties in the use of a radio propagation channel and complicates its mathematical analysis. Simply put, multipath is a physical phenomenon that is intrinsic to the operation of indoor and outdoor forms of wireless communications.

Before proceeding to discuss specific aspects of satellite communications and wireless communications, however, it is appropriate that we begin the discussion by describing multiple-access techniques, which enable different users to simultaneously (or nearly so) access a common channel.

8.2 Multiple-Access Techniques

Multiple access is a technique whereby many subscribers or local stations can share the use of a communication channel at the same time or nearly so, despite the fact that their individual transmissions may originate from widely different locations. Stated in another way, a multiple-access technique permits the communication resources of the channel to be shared by a large number of users seeking to communicate with each other.

There are subtle differences between multiple access and multiplexing that should be noted:

- ▶ Multiple access refers to the remote sharing of a communication channel such as a satellite or radio channel by users in highly dispersed locations. On the other hand, multiplexing refers to the sharing of a channel such as a telephone channel by users confined to a local site.
- ▶ In a multiplexed system, user requirements are ordinarily fixed. In contrast, in a multiple-access system user requirements can change dynamically with time, in which case provisions are necessary for dynamic channel allocation.

For obvious reasons it is desirable that in a multiple-access system the sharing of resources of the channel be accomplished without causing serious interference between users of the system. In this context, we may identify four basic types of multiple access:

1. Frequency-division multiple access (FDMA).

In this technique, disjoint subbands of frequencies are allocated to the different users on a continuous-time basis. In order to reduce interference between users allocated adjacent channel bands, *guard bands* are used to act as buffer zones, as illustrated in Figure 8.1a. These guard bands are necessary because of the impossibility of achieving ideal filtering for separating the different users.

2. Time-division multiple access (TDMA).

In this second technique, each user is allocated the full spectral occupancy of the channel, but only for a short duration of time called a *time slot*. As shown in Figure 8.1b, buffer zones in the form of *guard times* are inserted between the assigned time slots. This is done

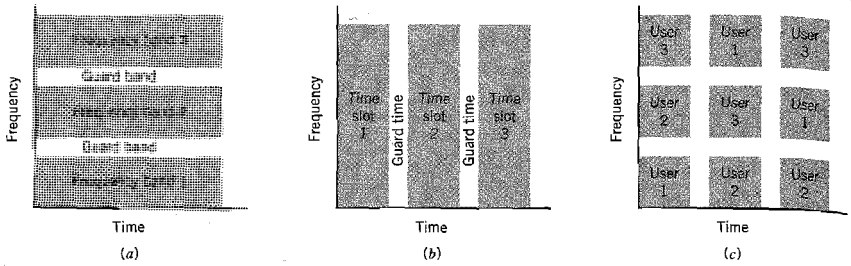


FIGURE 8.1 Illustrating the ideas behind multiple-access techniques. (a) Frequency-division multiple access. (b) Time-division multiple access. (c) Frequency-hop multiple access.

to reduce interference between users by allowing for time uncertainty that arises due to system imperfections, especially in synchronization schemes.

3. Code-division multiple access (CDMA).

In FDMA, the resources of the channel are shared by dividing them along the frequency coordinate into disjoint frequency bands, as illustrated in Figure 8.1a. In TDMA, the resources are shared by dividing them along the time coordinate into disjoint time slots, as illustrated in Figure 8.1b. In Figure 8.1c, we illustrate another technique for sharing the channel resources by using a hybrid combination of FDMA and TDMA, which represents a specific form of code-division multiple access (CDMA). For example, *frequency hopping* may be employed to ensure that during each successive time slot, the frequency bands assigned to the users are reordered in an essentially random manner. To be specific, during time slot 1, user 1 occupies frequency band 1, user 2 occupies frequency band 2, user 3 occupies frequency band 3, and so on. During time slot 2, user 1 hops to frequency band 3, user 2 hops to frequency band 1, user 3 hops to frequency band 2, and so on. Such an arrangement has the appearance of the users playing a game of musical chairs. An important advantage of CDMA over both FDMA and TDMA is that it can provide for *secure* communications. In the type of CDMA illustrated in Figure 8.1c, the frequency hopping mechanism can be implemented through the use of a pseudo-noise (PN) sequence.

4. Space-division multiple access (SDMA).

In this multiple-access technique, resource allocation is achieved by exploiting the spatial separation of the individual users. In particular, *multibeam antennas* are used to separate radio signals by pointing them along different directions. Thus, different users are enabled to access the channel simultaneously on the same frequency or in the same time slot.

These multiple-access techniques share a common feature: allocating the communication resources of the channel through the use of disjointedness (or orthogonality in a loose sense) in time, frequency, or space.

With this background material at hand, we are now ready to discuss some important multiuser communication systems.

8.3 Satellite Communications

In a geostationary satellite communication system,¹ a message signal is transmitted from an earth station via an *uplink* to a satellite, amplified in a *transponder* (i.e., electronic

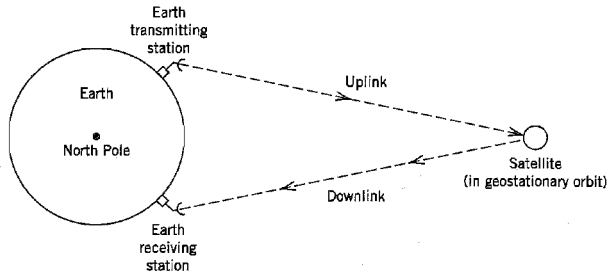


FIGURE 8.2 Satellite communications system.

circuitry) on board the satellite, and then retransmitted from the satellite via a *downlink* to another earth station, as illustrated in Figure 8.2. The most popular frequency band for satellite communications is 6 GHz (C-band) for the uplink and 4 GHz for the downlink. The use of this frequency band offers the following advantages:

- Relatively inexpensive microwave equipment.
- Low attenuation due to rainfall; rainfall is the primary atmospheric cause of signal degradation.
- Insignificant sky background noise; the sky background noise (due to random noise emissions from galactic, solar, and terrestrial sources) reaches its lowest level between 1 and 10 GHz.

However, radio interference limits the applications of communication satellites operating in the 6/4 GHz band, because the transmission frequencies of this band coincide with those used for terrestrial microwave systems. This problem is eliminated in the more powerful “second-generation” communication satellites that operate in the 14/12 GHz band (i.e., Ku-band); moreover, the use of these higher frequencies makes it possible to build smaller and therefore less expensive antennas.

The block diagram of Figure 8.3 shows the basic components of a single transponder channel of a typical communication satellite. Specifically, the receiving antenna output of the uplink is applied to the cascade connection of the following components:

- *Band-pass filter*, designed to separate the received signal from among the different radio channels.
- *Low-noise amplifier*.

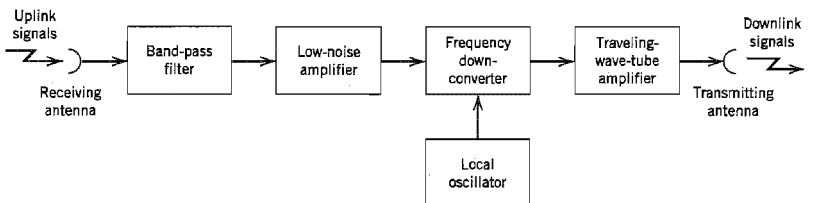


FIGURE 8.3 Block diagram of transponder.

- ▶ *Frequency down-converter*, the purpose of which is to convert the received radio frequency (RF) signal to the desired downlink frequency.
- ▶ *Traveling-wave tube amplifier*, which provides high gain over a wide band of frequencies. In a traveling-wave tube (TWT), an electromagnetic signal travels along a helix (i.e., a spring-shaped coil of wire), while electrons in a high-voltage beam travel through the helix at a speed close to that of the signal wave; the net result is the transfer of power from the electrons to the wave, which grows rapidly as the signal wave travels down the helix.

The channel configuration shown in Figure 8.3 uses a single frequency translation. Other channel configurations do the frequency conversion from the uplink to the downlink frequency in two stages: down-conversion to an intermediate frequency, followed by amplification, and then up-conversion to the desired transmit frequency.

Propagation time delay becomes particularly pronounced in a satellite channel because of the large distances involved. Specifically, speech signals sent by satellite incur a transmission delay of approximately 270 ms. Hence, for speech signals, any impedance mismatch at the receiving end of a satellite link results in an *echo* of the speaker's voice, which is heard back at the transmitting end after a round-trip delay of approximately 540 ms. We may overcome this problem by using an *echo canceller*, which is a device that subtracts an estimate of the echo from the return path; elimination of the echo is performed by means of a special filter that adapts itself to the changing channel characteristics.

The satellite channel is closely represented by an *additive white Gaussian noise (AWGN) model*, which applies to both the uplink and downlink portions of the satellite communication system. Accordingly, much of the material presented in Chapter 6 on passband systems for the transmission of data, with particular reference to phase-shift keying and frequency-shift keying techniques, is directly applicable to digital satellite communications.

A satellite transponder differs from a conventional microwave line-of-sight repeater in that many earth stations can access the satellite from widely different locations on earth at the same time or nearly so. This capability is made possible by using one of the multiple-access techniques discussed in Section 8.2. In this context we may offer the following observations:

- ▶ In a satellite channel, nonlinearity of the transponder is the primary cause of interference between users. To contain this serious problem, the traveling-wave tube amplifier in the transponder is purposely operated below capacity. Consequently, we find that in an FDMA system the power efficiency of the system is reduced because of the necessary *power backoff* of the traveling-wave tube amplifier.
- ▶ In a TDMA system, the users access the satellite transponder one at a time. Accordingly, the satellite transponder is now able to operate close to full power efficiency by permitting the traveling-wave tube amplifier to run into saturation. This, in turn, means that TDMA uses the transponder more efficiently than FDMA, hence its wide use in the implementation of digital satellite communication systems.
- ▶ SDMA operates by exploiting the spatial locations of earth stations, which is achieved by means of *onboard switching*. Specifically, the transponder is equipped with multiple antennas, with the proper antenna beam being selected for radio transmission to the particular earth station demanding use of the transponder.

In addition to multiple access, another capability of a satellite channel is that of *broadcasting* with emphasis on broad area coverage. Here we mention broadcasting satellites, which are characterized by their high power transmission to inexpensive receivers.

This characteristic is exploited in the use of *direct broadcast satellites* (DBS), designed for home reception of television services on a very wide scale. By comparison with the large earth stations used for satellite communications, the earth stations for DBS are very simple and therefore inexpensive.

8.4 Radio Link Analysis

An important issue that arises in the design of satellite communication systems is that of link budget analysis.² As its name implies, a *link budget*, or more precisely “link power budget,” is the totaling of all the gains and losses incurred in operating a communication link. In particular, the balance sheet constituting the link budget provides a detailed accounting of three broadly defined items:

1. Apportionment of the resources available to the transmitter and the receiver.
2. Sources responsible for the loss of signal power.
3. Sources of noise.

Putting all these items together into the link budget, we end up with an *estimation* procedure for evaluating the performance of a radio link, which could be the uplink or downlink of a satellite communication system. Needless to say, the essence of the communication link analysis presented in this section also applies to other radio links that rely on *line of sight* for their operation. It is for this reason the treatment of radio link analysis presented in this section is of a generic nature. The section finishes with an illustrative example on the budget analysis of a downlink of a digital satellite communication system.

From the material presented in Chapter 6 we learned that the performance of a digital communication system, in the presence of channel noise modeled as additive white Gaussian noise, is defined by a formula having the shape of a “waterfall” curve as shown in Figure 8.4. This figure portrays the probability of symbol error, P_e , plotted versus the bit

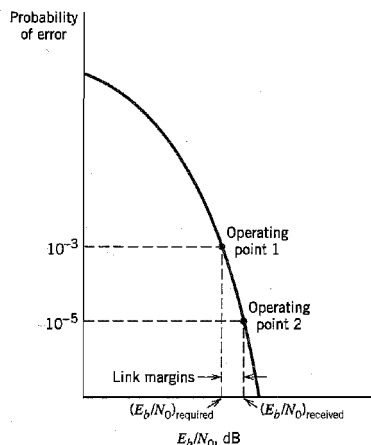


FIGURE 8.4 “Waterfall” curve relating the probability of error to the E_b/N_0 ratio.

energy-to-noise spectral density ratio, E_b/N_0 . Once a modulation scheme has been chosen, the first design task is to specify two particular values of E_b/N_0 as described here:

1. *Required E_b/N_0 .*

Suppose for example, the prescribed probability of symbol error is $P_e = 10^{-3}$. Using the waterfall curve of Figure 8.4 pertaining to the modulation scheme of interest, the E_b/N_0 required to realize the prescribed P_e is determined. Let $(E_b/N_0)_{\text{req}}$ denote the value of E_b/N_0 obtained from this calculation. The prescribed P_e and the calculated $(E_b/N_0)_{\text{req}}$ define a point on the waterfall curve of Figure 8.4, which is designated as operating point 1.

2. *Received E_b/N_0 .*

To assure *reliable* operation of the communication link, the link budget includes a safety measure called the *link margin*. The link margin provides protection against change and the unexpected. Thus the (E_b/N_0) actually received by the system is somewhat larger than $(E_b/N_0)_{\text{req}}$. Let $(E_b/N_0)_{\text{rec}}$ denote the actual or received E_b/N_0 , which defines a second point on the waterfall curve of Figure 8.4, designated as operating point 2. The P_e corresponding to operating point 2 is shown as 10^{-5} in Figure 8.4 merely for the purpose of illustration. In any event, introducing the link margin denoted by M , we may write

$$\left(\frac{E_b}{N_0}\right)_{\text{rec}} = M \left(\frac{E_b}{N_0}\right)_{\text{req}} \quad (8.1)$$

Equivalently, expressing the two E_b/N_0 values of interest in decibels, we may define the link margin as

$$M(\text{dB}) = \left(\frac{E_b}{N_0}\right)_{\text{rec}} (\text{dB}) - \left(\frac{E_b}{N_0}\right)_{\text{req}} (\text{dB}) \quad (8.2)$$

Clearly, the larger we make the link margin M , the more reliable is the communication link. However, the increased reliability of the link is attained at the cost of a higher E_b/N_0 .

■ FREE-SPACE PROPAGATION MODEL

The next step in formulating the link budget is to calculate the received signal power. Naturally, this calculation accounts for all the gains and losses incurred in the transmission and reception of the carrier.

In a radio communication system, the propagation of the modulated signal is accomplished by means of a *transmitting antenna*, the function of which is twofold:

- ▶ To convert the electrical modulated signal into an electromagnetic field. In this capacity, the transmitting antenna acts as an “impedance-transforming” transducer, matching the impedance of the antenna to that of free space.
- ▶ To radiate the electromagnetic energy in desired directions.

At the receiver, we have a *receiving antenna* whose function is the opposite of that of the transmitting antenna: It converts the electromagnetic field into an electrical signal from which the modulated signal is extracted. In addition, the receiving antenna may be required to suppress radiation originating from directions where it is not wanted.

Typically, the receiver is located in the farfield of the transmitting antenna, in which case, for all practical purposes, we may view the transmitting antenna as a fictitious volumeless emitter or *point source*. A complete description of the far field of the point source requires knowledge of the electromagnetic field as a function of both time and space.

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 radio 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 a 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

$$\rho(d) = \frac{P_t}{4\pi d^2} \text{ watts/m}^2 \quad (8.3)$$

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

$$\Phi = d^2 \rho(d) \quad (8.4)$$

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,

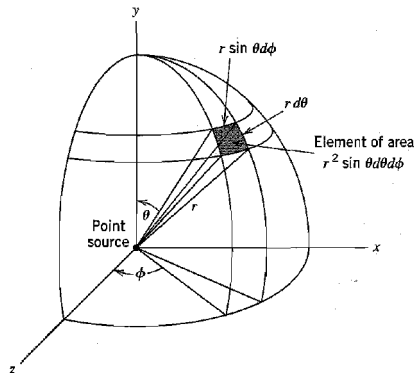


FIGURE 8.5 Illustrating the spherical coordinates of a point source.

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)

$$d\Omega = \sin \theta \, d\theta \, d\phi \quad \text{steradians} \quad (8.5)$$

The total power radiated is therefore

$$P = \int \Phi(\theta, \phi) \, d\Omega \quad \text{watts} \quad (8.6)$$

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

$$\begin{aligned} P_{\text{av}} &= \frac{1}{4\pi} \int \Phi(\theta, \phi) \, d\Omega \\ &= \frac{P}{4\pi} \quad \text{watts/steradian} \end{aligned} \quad (8.7)$$

which represents the radiation intensity that is produced by an isotropic source radiating the same total power P .

Directive Gain, Directivity, and Power Gain³

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

$$\begin{aligned} g(\theta, \phi) &= \frac{\Phi(\theta, \phi)}{P_{\text{av}}} \\ &= \frac{\Phi(\theta, \phi)}{P/4\pi} \end{aligned} \quad (8.8)$$

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

$$G = \eta_{\text{radiation}} D \quad (8.9)$$

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

if any losses are present in the antenna (i.e., $\eta_{\text{radiation}} < 1$). Henceforth, we assume that the antenna is 100 percent efficient and therefore refer only to the power gain of the antenna.

The concept of power gain, which is based on the transmitted power-pattern shape, can be extended to a receiving antenna by virtue of the *reciprocity principle*. An antenna is said to be reciprocal if the transmission medium is linear, passive and isotropic. For a given antenna structure, the power gains of transmitting and receiving antennas are then identical.

The power gain of an antenna is the result of concentrating the power density in a restricted region smaller than 4π steradians, as illustrated in Figure 8.6. In light of the picture portrayed in this figure, we may introduce the following two parameters:

1. *Effective radiated power referenced to an isotropic source (EIRP)*; the EIRP is defined as the product of the transmitted power, P_t , and the power gain of the transmitting antenna, G_t , as shown by

$$\text{EIRP} = P_t G_t \quad \text{watts} \quad (8.10)$$

2. *Antenna beamwidth*, representing a “planar” measure of the antenna’s solid angle of view; the beamwidth, in degrees or radians, is defined as the angle that subtends the two points on the mainlobe of the field-power pattern at which the peak field power is reduced by 3 dBs. The higher the power gain of the antenna, the narrower is the antenna beamwidth.

Another matter of interest discernible from Figure 8.6 is the *sidelobes* of the field-power pattern. Unfortunately, every physical antenna has sidelobes, which are responsible for absorbing unwanted interfering radiations.

Effective Aperture

A term that has a special significance for a receiving antenna is the *effective aperture* of the antenna, which is defined as the ratio of the power available at the antenna terminals to the power per unit area of the appropriately polarized incident electromagnetic wave. The effective aperture, denoted by A , is defined in terms of the antenna’s power gain G as

$$A = \frac{\lambda^2}{4\pi} G \quad (8.11)$$

where λ is the *wavelength* of the carrier. The wavelength λ and frequency f are reciprocally related as

$$\lambda = \frac{c}{f} \quad (8.12)$$

where c is the speed of light (approximately equal to 3×10^8 m/s).

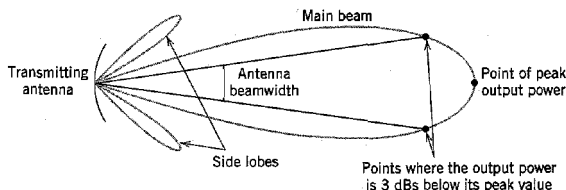


FIGURE 8.6 Illustrating the concentration of power density of a transmitting antenna inside a region smaller than 4π steradians.

The term effective aperture has particular significance in the context of reflector antennas and electromagnetic horns that are characterized by a well-defined aperture. For these antennas, *the ratio of the antenna's effective aperture to its physical aperture is a direct measure of the antenna's aperture efficiency, η_{aperture}* , in radiating power to a desired direction or absorbing power from that direction. Nominal values for the efficiency η_{aperture} of reflector antennas lie in the range of 45 to 75 percent.

Friis Free-Space Equation

With this introductory material on antennas at hand, we are now ready to formulate the basic propagation equation for a radio communication link. Consider a transmitting antenna with an EIRP defined in Equation (8.10). Invoking the inverse-square law of Equation (8.3), we may express the power density of the transmitting antenna as $\text{EIRP}/4\pi d^2$, where d is the distance between the receiving and transmitting antennas. The power P_r absorbed by the receiving antenna is the product of this power density and the antenna's effective area denoted by A_r , as shown by

$$\begin{aligned} P_r &= \left(\frac{\text{EIRP}}{4\pi d^2} \right) A_r \\ &= \frac{P_t G_r A_r}{4\pi d^2} \text{ watts} \end{aligned} \quad (8.13)$$

According to the reciprocity principle, we may use Equation (8.11) to express the effective area of the receiving antenna as

$$A_r = \frac{\lambda^2}{4\pi} G_r$$

where G_r is the power gain of the receiving antenna. Substituting this formula for A_r into Equation (8.13), we may express the received signal power in the equivalent form

$$P_r = P_t G_t G_r \left(\frac{\lambda}{4\pi d} \right)^2 \quad (8.14)$$

Equation (8.14) is called the *Friis free-space equation*.⁴

The *path loss*, PL, representing signal "attenuation" in decibels across the entire communication link, is defined as *the difference (in decibels) between the transmitted signal power P_t and received signal power P_r* , as shown by

$$\begin{aligned} \text{PL} &= 10 \log_{10} \left(\frac{P_t}{P_r} \right) \\ &= -10 \log_{10}(G_t G_r) + 10 \log_{10} \left(\frac{4\pi d}{\lambda} \right)^2 \end{aligned} \quad (8.15)$$

The minus sign associated with the first term in Equation (8.15) signifies the fact that this term represents a "gain." The second term, due to the collection of terms $(4\pi d/\lambda)^2$, is called the *free-space loss*, denoted by $L_{\text{free space}}$. Note that increasing the distance d separating the receiving antenna from the transmitting antenna causes the free-space loss to increase, which, in turn, compels us to operate the radio communication link at lower frequencies so as to maintain the path loss at a manageable level.

The Friis free-space equation enables us to calculate the path loss PL for specified values of power gains G_t and G_r , the carrier wavelength λ , and distance d . To complete

the budget link analysis, we need to calculate the average noise power in the received signal, which is considered next.

■ NOISE FIGURE

To perform noise analysis at the receiver of a communication system, we need a convenient measure of the noise performance of a linear two-port device. One such measure is furnished by the so-called *noise figure*. Consider a linear two-port device connected to a signal source of internal impedance $Z(f) = R(f) + jX(f)$ at the input, as in Figure 8.7. The noise voltage $v(t)$ represents the thermal noise associated with the internal resistance $R(f)$ of the source. The output noise of the device is made up of two contributions, one due to the source and the other due to the device itself. We define *the available output noise power in a band of width Δf centered at frequency f as the maximum average noise power in this band, obtainable at the output of the device*. The maximum noise power that the two-port device can deliver to an external load is obtained when the load impedance is the complex conjugate of the output impedance of the device, that is, when the resistance is matched and the reactance is tuned out. We define *the noise figure of the two-port device as the ratio of the total available output noise power (due to the device and the source) per unit bandwidth to the portion thereof due solely to the source*.

Let the spectral density of the total available noise power of the device output be $S_{NO}(f)$, and the spectral density of the available noise power due to the source at the device input be $S_{NS}(f)$. Also let $G(f)$ denote the *available power gain* of the two-port device, defined as *the ratio of the available signal power at the output of the device to the available signal power of the source when the signal is a sinusoidal wave of frequency f* . Then we may express the noise figure F of the device as

$$F = \frac{S_{NO}(f)}{G(f)S_{NS}(f)} \quad (8.16)$$

If the device were noise free, $S_{NO}(f) = G(f)S_{NS}(f)$, and the noise figure would then be unity. In a physical device, however, $S_{NO}(f)$ is larger than $G(f)S_{NS}(f)$, so that the noise figure is always larger than unity. The noise figure is commonly expressed in decibels, that is, as $10 \log_{10} F$.

The noise figure may also be expressed in an alternative form. Let $P_s(f)$ denote the available signal power from the source, which is the maximum average signal power that can be obtained. For the case of a source providing a single-frequency signal component

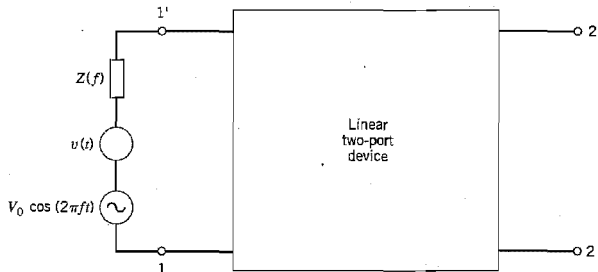


FIGURE 8.7 Linear two-port device.

with open-circuit voltage $V_0 \cos(2\pi ft)$, the available signal power is obtained when the load connected to the source is

$$Z^*(f) = R(f) - jX(f)$$

where the asterisk denotes complex conjugation. Under this condition, we find that

$$\begin{aligned} P_s(f) &= \left[\frac{V_0}{2R(f)} \right]^2 R(f) \\ &= \frac{V_0^2}{4R(f)} \end{aligned} \quad (8.17)$$

The available signal power at the output of the device is therefore

$$P_o(f) = G(f)P_s(f) \quad (8.18)$$

Then, multiplying both the numerator and denominator of the right-hand side of Equation (8.16) by $P_s(f) \Delta f$, we obtain

$$\begin{aligned} F &= \frac{P_s(f)S_{NO}(f) \Delta f}{G(f)P_s(f)S_{NS}(f) \Delta f} \\ &= \frac{P_s(f)S_{NO}(f) \Delta f}{P_o(f)S_{NS}(f) \Delta f} \\ &= \frac{\rho_s(f)}{\rho_o(f)} \end{aligned} \quad (8.19)$$

where

$$\rho_s(f) = \frac{P_s(f)}{S_{NS}(f) \Delta f} \quad (8.20)$$

$$\rho_o(f) = \frac{P_o(f)}{S_{NO}(f) \Delta f} \quad (8.21)$$

We refer to $\rho_s(f)$ as the *available signal-to-noise ratio of the source* and to $\rho_o(f)$ as the *available signal-to-noise ratio at the device output*, both measured in a narrow band of width Δf centered at f . Since the noise figure is always greater than unity, it follows from Equation (8.19) that the signal-to-noise ratio always decreases with amplification, which is a significant result.

The noise figure F is a function of the operating frequency f ; it is therefore referred to as the *spot noise figure*. In contrast, we may define an *average noise figure* F_0 of a two-port device as the ratio of the total noise power at the device output to the output noise power due solely to the source. That is,

$$F_0 = \frac{\int_{-\infty}^{\infty} S_{NO}(f) df}{\int_{-\infty}^{\infty} G(f)S_{NS}(f) df} \quad (8.22)$$

It is apparent that in the case of thermal noise in the input circuit with $R(f)$ constant and constant gain throughout a fixed band with zero gain at other frequencies, the spot noise figure F and the average noise figure F_0 are identical.

Equivalent Noise Temperature

A disadvantage of the noise figure F is that when it is used to compare low-noise devices, the values obtained are all close to unity, which makes the comparison rather

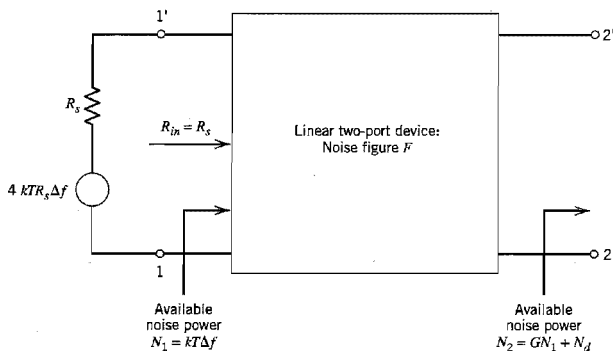


FIGURE 8.8 Linear two-port device matched to the internal resistance of a source connected to the input.

difficult. In such cases, it is preferable to use the *equivalent noise temperature*. Consider a linear two-port device whose input resistance is matched to the internal resistance of the source as shown in Figure 8.8. In this diagram, we have also included the noise voltage generator associated with the internal resistance R_s of the source. The mean-square value of this noise voltage is $4kTR_s\Delta f$, where k is Boltzmann's constant. Hence, the available noise power at the device input is

$$N_1 = kT\Delta f \quad (8.23)$$

Let N_d denote the noise power contributed by the two-port device to the total available output noise power N_2 . We define N_d as

$$N_d = GkT_e\Delta f \quad (8.24)$$

where G is the available power gain of the device and T_e is its equivalent noise temperature. Then it follows that the total output noise power is

$$\begin{aligned} N_2 &= GN_1 + N_d \\ &= Gk(T + T_e)\Delta f \end{aligned} \quad (8.25)$$

The noise figure of the device is therefore (see the output port of Figure 8.8)

$$F = \frac{N_2}{N_2 - N_d} = \frac{T + T_e}{T} \quad (8.26)$$

Solving for the equivalent noise temperature:

$$T_e = T(F - 1) \quad (8.27)$$

The noise figure F is measured under matched input conditions, and with the noise source at temperature T . By convention the temperature T is taken as "room temperature," namely 290 K, where K stands for "degree Kelvin."

Cascade Connection of Two-Port Networks

It is often necessary to evaluate the noise figure of a cascade connection of two-port networks whose individual noise figures are known. Consider Figure 8.9, consisting of a

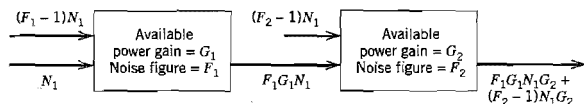


FIGURE 8.9 A cascade of two noisy two-port networks.

pair of two-port networks of noise figures F_1 and F_2 and power gains G_1 and G_2 , connected in cascade. It is assumed that the devices are matched, and that the noise figure F_2 of the second network is defined assuming an input noise power N_1 .

At the input of the first network, we have a noise power N_1 contributed by the source, plus an equivalent noise power $(F_1 - 1)N_1$ contributed by the network itself. The output noise power from the first network is therefore $F_1N_1G_1$. Added to this noise power at the input of the second network, we have the equivalent extra power $(F_2 - 1)N_1$ contributed by the second network itself. The output noise power from this second network is therefore equal to $F_1G_1N_1G_2 + (F_2 - 1)N_1G_2$. We may consider the noise figure F as the ratio of the actual output noise power to the output noise power assuming the networks to be noiseless. We may therefore express the overall noise figure of the cascade connection of Figure 8.9 as

$$\begin{aligned} F &= \frac{F_1G_1N_1G_2 + (F_2 - 1)N_1G_2}{N_1G_1G_2} \\ &= F_1 + \frac{F_2 - 1}{G_1} \end{aligned} \quad (8.28)$$

The result may be readily extended to the cascade connection of any number of two-port networks, as shown by

$$F = F_1 + \frac{F_2 - 1}{G_1} + \frac{F_3 - 1}{G_1G_2} + \frac{F_4 - 1}{G_1G_2G_3} + \dots \quad (8.29)$$

where F_1, F_2, F_3, \dots are the individual noise figures, and G_1, G_2, G_3, \dots are the available power gains, respectively. Equation (8.29) shows that if the first stage of the cascade connection in Figure 8.9 has a high gain, the overall noise figure F is dominated by the noise figure of the first stage.

Correspondingly, we may express the overall equivalent noise temperature of the cascade connection of any number of noisy two-port networks as follows:

$$T_e = T_1 + \frac{T_2}{G_1} + \frac{T_3}{G_1G_2} + \frac{T_4}{G_1G_2G_3} + \dots \quad (8.30)$$

where T_1, T_2, T_3, \dots are the equivalent noise temperatures of the individual networks, and G_1, G_2, G_3, \dots are the available power gains, respectively. Equation (8.30) is known as the *Friis formula*. Here again we note that if the gain G_1 of the first stage is high, the equivalent noise temperature T_e is dominated by that of the first stage.

► EXAMPLE 8.1 Noise Temperature of Earth-Terminal Receiver

Figure 8.10 shows a typical earth-terminal receiver, consisting of a low-noise radio-frequency (RF) amplifier (LNA), frequency down-converter (mixer), and intermediate frequency (IF)

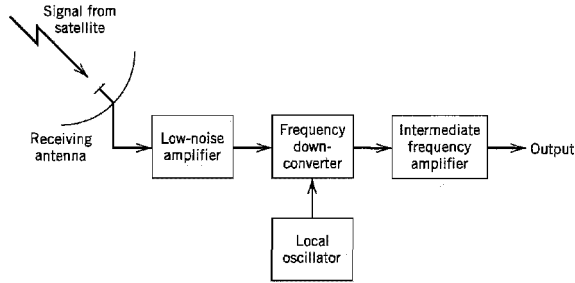


FIGURE 8.10 Block diagram of earth terminal receiver.

amplifier. The equivalent noise temperatures of these components, including the receiving antenna, are

$$\begin{aligned}
 T_{\text{antenna}} &= 50 \text{ K} \\
 T_{\text{RF}} &= 50 \text{ K} \\
 T_{\text{mixer}} &= 500 \text{ K} \\
 T_{\text{IF}} &= 1000 \text{ K}
 \end{aligned}$$

The available power gains of the two amplifiers are

$$\begin{aligned}
 G_{\text{RF}} &= 200 = 23 \text{ dB} \\
 G_{\text{IF}} &= 1000 = 30 \text{ dB}
 \end{aligned}$$

To calculate the equivalent noise temperature of the receiver, we use Equation (8.30), obtaining

$$\begin{aligned}
 T_e &= T_{\text{antenna}} + T_{\text{RF}} + \frac{T_{\text{mixer}} + T_{\text{IF}}}{G_{\text{RF}}} \\
 &= 50 + 50 + \frac{500 + 1000}{200} \\
 &= 107.5 \text{ K}
 \end{aligned}$$

► EXAMPLE 8.2 Downlink Budget Analysis of a Digital Satellite Communication System

In a digital satellite communication system, one of the key elements in the overall design and analysis of the system is the downlink power budget, which is usually more critical than the uplink power budget because of the practical constraints imposed on downlink power and satellite antenna size. The example presented here addresses a sample downlink budget analysis, assuming that any required uplink power (within limits) is available for satisfactory operation of the system.

The critical parameter to be calculated is the *ratio of received carrier power-to-noise spectral density*, denoted by C/N_0 . According to the Friis free-space equation (8.14), the average power received at the earth terminal to the average power P_t transmitted by the satellite is

$$P_r = P_t G_t G_r \left(\frac{\lambda}{4\pi d} \right)^2$$

where, in this example, G_t is the power gain of the satellite antenna, G_r is the power gain of the receiving earth-terminal antenna, λ is the carrier wavelength for the downlink, and d is the distance between the satellite and the earth terminal. Given that the equivalent noise temperature of the system is T_e , we may use Equation (1.94) of Chapter 1 to express the noise spectral density N_0 as kT_e , where k is Boltzmann's constant. Moreover, from Equation (8.10) we note that $P_s G_t$ is equal to the EIRP of the satellite. Hence, dividing P_r by N_0 , we may express the C/N_0 ratio for the downlink as

$$\left(\frac{C}{N_0}\right)_{\text{downlink}} = (\text{EIRP})_{\text{satellite}} \left(\frac{G_r}{T_e}\right)_{\text{earth terminal}} \left(\frac{\lambda}{4\pi d}\right)^2 \frac{1}{k} \quad (8.31)$$

For a given satellite system, the free-space loss $(4\pi d/\lambda)^2$ is a constant. Viewing the system from the earth terminal, we see from Equation (8.31) that the (C/N_0) ratio is proportional to G_r/T_e . The ratio G_r/T_e may therefore be used to assess the "quality" of an earth terminal; it is usually shortened to the G/T ratio, which is referred to as the *figure of merit* of the receiving earth terminal. Thus, rewriting the formula (8.31) for the (C/N_0) ratio measured in decibels, we may express it as the sum of gains and losses as itemized here:

1. $(\text{EIRP})_{\text{satellites}}$ measured in dBW, where dBW denotes decibels referenced to 1 watt, that is, 0 dBW.
2. $(G/T)_{\text{earth terminal}}$ measured in dB/K, where K refers to degree Kelvin.
3. $L_{\text{free space}}$, denoting the free-space loss $10 \log_{10}(4\pi d/\lambda)^2$ in dB.
4. $-10 \log_{10} k$, representing the gain in dBW/K-Hz due to division by the Boltzmann constant $k = 1.38 \times 10^{-23}$ joule/K.

Table 8.1 presents the values of these four terms for the downlink of a typical domestic digital satellite communication system, based on the following:

1. The transponder is operated at its maximum output power (i.e., no power backoff is employed), yielding an EIRP of 46.5 dBW.
2. The receiving earth terminal uses a 2m-dish antenna with a power gain $G = 45$ dB, and the receiver is configured as in Example 8.1 with equivalent temperature $T = 107.5$ K. Hence

$$\begin{aligned} \frac{G}{T} &= 45 - 10 \log_{10} 107.5 \\ &= 45 - 20.3 \\ &= 24.7 \text{ dB/K} \end{aligned}$$

3. The free-space loss is

$$L_{\text{free-space}} = 92.4 + 20 \log_{10} f + 20 \log_{10} d \text{ dB} \quad (8.32)$$

TABLE 8.1 Downlink power budget for Example 8.2

Variable	Value
EIRP	+46.5 dBW
G/T ratio	+24.7 dB/K
Free-space loss	-206 dB
Boltzmann constant	+228.6 dBW/K-Hz
C/N_0	93.8 dB-Hz

where the downlink carrier frequency f is in GHz and the distance d between the satellite and the earth terminal is in kilometers. For a geostationary satellite, the distance between the satellite and an earth terminal lies in the range of 36,000 to 41,000 km. Thus choosing $d = 40,000$ km and assuming $f = 12$ GHz, the use of Equation (8.32) yields

$$\begin{aligned} L_{\text{free-space}} &= 92.4 + 20 \log_{10} 12 + 20 \log_{10} 40,000 \\ &= 92.4 + 21.6 + 92.0 \\ &= 206 \text{ dB} \end{aligned}$$

4. With the Boltzmann constant $k = 1.39 \times 10^{-23}$ joule/K, its contribution to the C/N_0 ratio is

$$\begin{aligned} -10 \log_{10} k &= 10 \log_{10} 1.39 \times 10^{-23} \\ &= 228.6 \text{ dBW/K-Hz} \end{aligned}$$

Totaling the gains and losses, we thus get

$$\left(\frac{C}{N_0} \right)_{\text{downlink}} = 93.8 \text{ dB-Hz}$$

The “received” downlink value of the (C/N_0) ratio may also be expressed in terms of the “required” value of the bit energy-to-noise spectral density ratio, $(E_b/N_0)_{\text{req}}$ dB, at the receiving earth terminal as (see Equation (8.2))

$$\left(\frac{C}{N_0} \right)_{\text{downlink}} = \left(\frac{E_b}{N_0} \right)_{\text{req}} + 10 \log_{10} M + 10 \log_{10} R \text{ dB} \quad (8.33)$$

where $10 \log_{10} M$ is the link margin in decibels, and R is the data rate in b/s. The link margin allows for excess rain losses in propagation and other power degradations. Typically, the link margin is selected as 4 dB for C-band, 6 dB for Ku-band, and higher for the higher K-band frequencies because of the higher rain losses. For operation at the Ku-band frequency of 12 GHz, we choose a link margin of 6 dB. Thus, using the value $C/N_0 = 93.8$ dB-Hz calculated from the link budget, the link margin $10 \log_{10} M = 6$ dB, and assuming $(E_b/N_0)_{\text{req}} = 12.5$ dB, the use of Equation (8.33) yields

$$\begin{aligned} 10 \log_{10} R &= 93.8 - 12.5 - 6 \\ &= 75.3 \end{aligned}$$

Hence,

$$R = 33.9 \text{ Mb/s}$$

Assuming the use of coherent 8-PSK for the transmission of digital data via the satellite, and substituting $(E_b/N_0) = 12.5$ dB in Equation (6.47) of Chapter 6, we find that the probability of symbol error $P_e = 0.6 \times 10^{-3}$.

To summarize, the digital satellite communication system analyzed in this example permits, under the worst operating conditions, data transmission on the downlink at a rate $R = 33.9$ Mb/s and with a probability of symbol error $P_e = 0.6 \times 10^{-3}$, assuming the use of 8-phase PSK. ◀

8.5 Wireless Communications

In this section we study the second type of multiuser radio communication system, namely, *wireless communications*, which is synonymous with *mobile radio*. The term mobile radio is usually meant to encompass indoor or outdoor forms of wireless communications where

a radio transmitter or receiver is capable of being moved, regardless of whether it actually moves or not. Due to the stochastic nature of the mobile radio channel, its characterization mandates the use of practical measurements and statistical analysis. The aim of such an evaluation is to quantify two factors of primary concern:

1. *Median signal strength*, which enables us to predict the minimum power needed to radiate from the transmitter so as to provide an acceptable quality of coverage over a predetermined service area.
2. *Signal variability*, which characterizes the fading nature of the channel.

Our specific interest in wireless communications is in the context of *cellular radio*⁵ that has the inherent capability of building mobility into the telephone network. With such a capability, a user can move freely within a service area and simultaneously communicate with any telephone subscriber in the world. An idealized model of the cellular radio system, illustrated in Figure 8.11, consists of an array of hexagonal *cells* with a *base station* located at the center of each cell; a typical cell has a radius of 1 to 12 miles. The function of the base stations is to act as an interface between *mobile subscribers* and the cellular radio system. The base stations are themselves connected to a *switching center* by dedicated *wirelines*.

The mobile switching center has two important roles. First, it acts as the interface between the cellular radio system and the public switched telephone network. Second, it performs overall supervision and control of the mobile communications. It performs the *latter* function by monitoring the signal-to-noise ratio of a call in progress, as measured at the base station in communication with the mobile subscriber involved in the call. When the SNR falls below a prescribed threshold, which happens when the mobile subscriber leaves its cell or when the radio channel fades, it is switched to another base station. This switching process, called a *handover* or *handoff*, is designed to move a mobile subscriber from one base station to another during a call in a transparent fashion, that is, without interruption of service.

The cellular concept relies on two essential features, as described here:

1. *Frequency reuse*. The term *frequency reuse* refers to the use of radio channels on the same carrier frequency to cover different areas, which are physically separated from

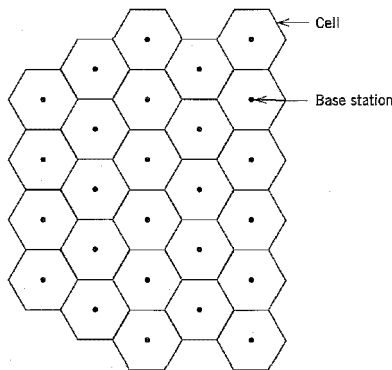


FIGURE 8.11 Idealized model of cellular radio.

each other sufficiently to ensure that *co-channel interference* is not objectionable. Thus, instead of covering an entire local area from a single transmitter with high power at a high elevation, frequency reuse makes it possible to achieve two commonsense objectives: keep the transmitted power from each base station to a minimum, and position the antennas of the base stations just high enough to provide for the area coverage of the respective cells.

2. *Cell splitting.* When the demand for service exceeds the number of channels allocated to a particular cell, cell splitting is used to handle the additional growth in traffic within that particular cell. Specifically, cell splitting involves a revision of cell boundaries, so that the local area formerly regarded as a single cell can now contain a number of smaller cells and use the channel complements of these new cells. The new cells, which have a smaller radius than the original cells, are called *microcells*. The transmitter power and the antenna height of the new base stations are correspondingly reduced, and the same set of frequencies are reused in accordance with a new plan.

For a hexagonal model of the cellular radio system, we may exploit the basic properties of hexagonal cellular geometry to lay out a radio channel assignment plan that determines which channel set should be assigned to which cell. We begin with two integers i and j ($i \geq j$), called *shift parameters*, which are predetermined in some manner. We note that with a hexagonal cellular geometry there are six “chains” of hexagons that emanate from each hexagon and that extend in different directions. Thus, starting with any cell as a reference, we find the nearest *co-channel cells* by proceeding as follows:

- Move i cells along any chain of hexagons, turn counterclockwise 60 degrees, and move j cells along the chain that lies on this new direction. The j th cells so located and the reference cell constitute the set of co-channel cells.

This procedure is repeated for a different reference cell, until all the cells in the system are covered. Figure 8.12 illustrates the application of this procedure for a single reference cell and the example of $i = 2$ and $j = 2$.

In North America, the band of radio frequencies assigned to the cellular system is 800–900 MHz. The subband 824–849 MHz is used to receive signals from the mobile units, and the subband 869–894 MHz is used to transmit signals to the mobile units. The

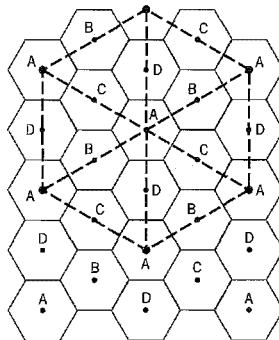


FIGURE 8.12 Illustrating the determination of co-channel cells.

use of these relatively high frequencies has the beneficial feature of providing a good portable coverage by penetrating buildings. In Europe and elsewhere, the base-mobile and mobile-base subbands are reversed.

■ PROPAGATION EFFECTS⁶

The major propagation problems encountered in the use of cellular radio in built-up areas are due to the fact that the antenna of a mobile unit may lie well below the surrounding buildings. Simply put, there is no “line-of-sight” path to the base station. Instead, radio propagation takes place mainly by way of scattering from the surfaces of the surrounding buildings and by diffraction over and/or around them, as illustrated in Figure 8.13. The important point to note from Figure 8.13 is that energy reaches the receiving antenna via more than one path. Accordingly, we speak of a *multipath phenomenon* in that the various incoming radio waves reach their destination from different directions and with different time delays.

To understand the nature of the multipath phenomenon, consider first a “static” multipath environment involving a stationary receiver and a transmitted signal that consists of a narrowband signal (e.g., unmodulated sinusoidal carrier). Let it be assumed that two attenuated versions of the transmitted signal arrive sequentially at the receiver. The effect of the differential time delay is to introduce a relative phase shift between the two components of the received signal. We may then identify one of two extreme cases that can arise:

- The relative phase shift is zero, in which case the two components add constructively, as illustrated in Figure 8.14a.
- The relative phase shift is 180 degrees, in which case the two component add destructively, as illustrated in Figure 8.14b.

We may also use *phasors* to demonstrate the constructive and destructive effects of multipath, as shown in Figures 8.15a and 8.15b, respectively. Note that in the static multipath environment described herein, the amplitude of the received signal does not vary with time.

Consider next a “dynamic” multipath environment in which the receiver is in motion and two versions of the transmitted narrowband signal reach the receiver via paths of

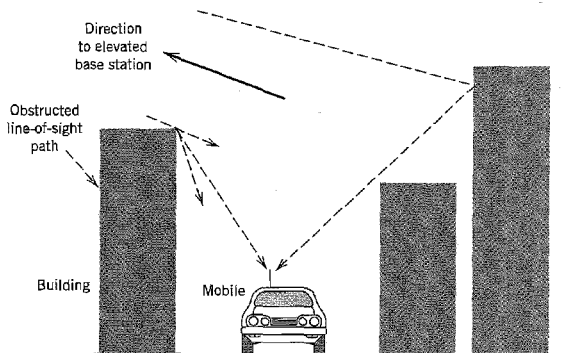


FIGURE 8.13 Illustrating the mechanism of radio propagation in urban areas. (From Parsons, 1992, with permission.)

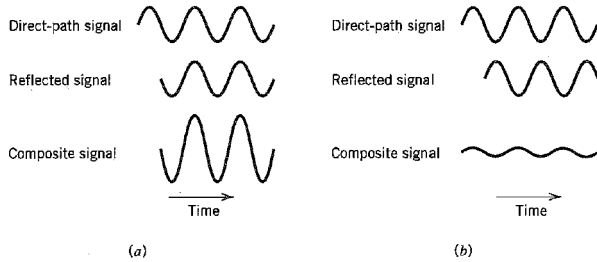


FIGURE 8.14 (a) Constructive and (b) destructive forms of the multipath phenomenon for sinusoidal signals.

different lengths. Due to motion of the receiver, there is a continuous change in the length of each propagation path. Hence, the relative phase shift between the two components of the received signal is a function of spatial location of the receiver. As the receiver moves, we now find that the received amplitude (envelope) is no longer constant as was the case in a static environment; rather, it varies with distance, as illustrated in Figure 8.16. At the top of this figure, we have also included the phasor relationships for the two components of the received signal at various locations of the receiver. Figure 8.16 shows that there is constructive addition at some locations, and almost complete cancellation at some other locations. This phenomenon is referred to as *signal fading*.

In a mobile radio environment encountered in practice, there may of course be a multitude of propagation paths with different lengths, and their contributions to the re-

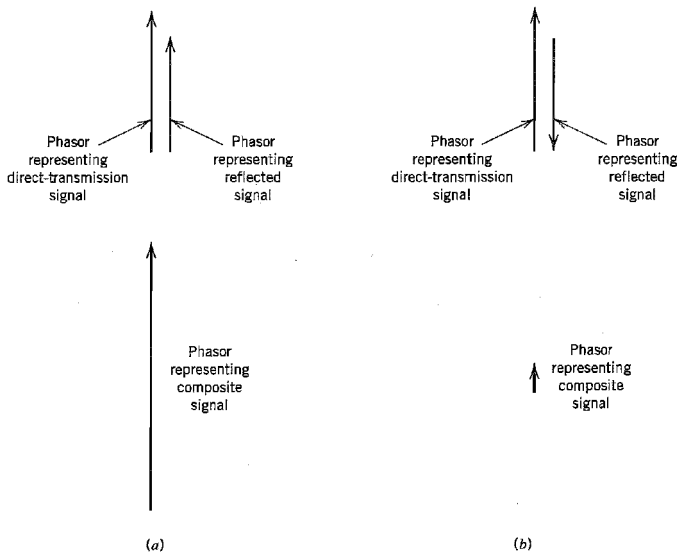


FIGURE 8.15 Phasor representations of (a) constructive and (b) destructive forms of multipath.

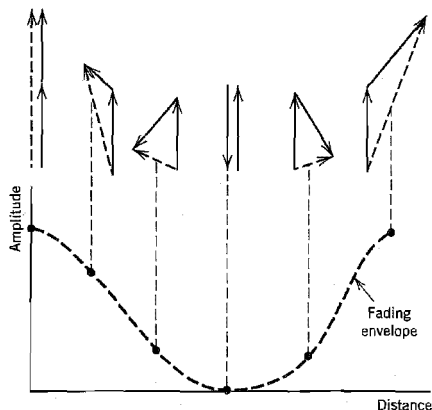


FIGURE 8.16 Illustrating how the envelope fades as two incoming signals combine with different phases. (From Parsons, 1992, with permission.)

ceived signal could combine in a variety of ways. The net result is that the envelope of the received signal varies with location in a complicated fashion, as shown by the experimental record of received signal envelope in an urban area that is presented in Figure 8.17. This figure clearly displays the fading nature of the received signal. The received signal envelope in Figure 8.17 is measured in dBm. The unit dBm is defined as $10 \log_{10}(P/P_0)$, with P denoting the power being measured and $P_0 = 1$ milliwatt. In the case of Figure 8.17, P is the instantaneous power in the received signal envelope.

Signal fading is essentially a *spatial phenomenon* that manifests itself in the time domain as the receiver moves. These variations can be related to the motion of the receiver as follows. To be specific, consider the situation illustrated in Figure 8.18, where the receiver is assumed to be moving along the line AA' with a constant velocity v . It is also

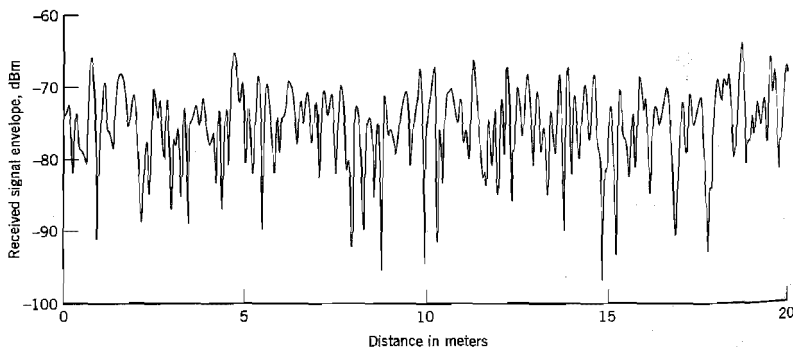


FIGURE 8.17 Experimental record of received signal envelope in an urban area. (From Parsons, 1992, with permission.)

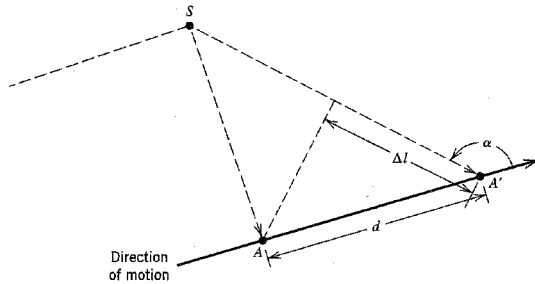


FIGURE 8.18 Illustrating the calculation of Doppler shift.

assumed that the received signal is due to a radio wave from a scatterer labeled S . Let Δt denote the time taken for the receiver to move from point A to A' . Using the notation described in Figure 8.18, the incremental change in the path length of the radio wave is deduced to be

$$\begin{aligned}\Delta l &= d \cos \alpha \\ &= -v \Delta t \cos \alpha\end{aligned}\quad (8.34)$$

where α is the spatial angle between the incoming radio wave and the direction of motion of the receiver. Correspondingly, the change in the phase angle of the received signal at point A' with respect to that at point A is given by

$$\begin{aligned}\Delta \phi &= \frac{2\pi}{\lambda} \Delta l \\ &= -\frac{2\pi v \Delta t}{\lambda} \cos \alpha\end{aligned}\quad (8.35)$$

where λ is the radio wavelength. The apparent change in frequency, or the *Doppler-shift*, is therefore

$$\begin{aligned}\nu &= -\frac{1}{2\pi} \frac{\Delta \phi}{\Delta t} \\ &= \frac{v}{\lambda} \cos \alpha\end{aligned}\quad (8.36)$$

The Doppler-shift ν is positive (resulting in an increase in frequency) when the radio waves arrive from ahead of the mobile unit, and it is negative when the radio waves arrive from behind the mobile unit.

8.6 Statistical Characterization of Multipath Channels

The narrowband characterization of the multipath environment described in Section 8.5 is appropriate for mobile radio transmissions where the signal bandwidth is very small

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 band-pass signal as

$$s(t) = \text{Re}[\tilde{s}(t) \exp(j2\pi f_c t)] \quad (8.37)$$

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

$$h(\tau; t) = \text{Re}[\tilde{h}(\tau; t) \exp(j2\pi f_c t)] \quad (8.38)$$

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

$$\tilde{s}_o(t) = \frac{1}{2} \int_{-\infty}^{\infty} \tilde{s}(t - \tau) \tilde{h}(\tau; t) d\tau \quad (8.39)$$

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

$$\tilde{H}(f; t) = \int_{-\infty}^{\infty} \tilde{h}(\tau; t) \exp(-j2\pi f \tau) d\tau \quad (8.40)$$

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.

For a statistical characterization of the channel, we make the following assumptions:

- ▶ The input delay-spread function $\tilde{h}(\tau; t)$ is a *zero-mean, complex-valued Gaussian process*. Our interest is confined to short-term fading; it is therefore reasonable to assume that $\tilde{h}(\tau; t)$ is also *stationary*. Because Fourier transformation is linear, the time-varying transfer function $\tilde{H}(f; t)$ has similar statistics.
- ▶ The channel is an *uncorrelated scattering channel*, which means that contributions from scatterers with different propagation delays are uncorrelated.

Consider then the autocorrelation function of the input delay-spread function $\tilde{h}(\tau; t)$. Since $\tilde{h}(\tau; t)$ is complex valued, we use the following definition for the autocorrelation function:

$$R_{\tilde{h}}(\tau_1, t_1; \tau_2, t_2) = E[\tilde{h}^*(\tau_1; t_1) \tilde{h}(\tau_2; t_2)] \quad (8.41)$$

where E is the statistical expectation operator, the asterisk denotes complex conjugation, τ_1 and τ_2 are the propagation delays of the two paths involved in the calculation, and t_1 and t_2 are the times at which the outputs of the two paths are observed. Invoking stationarity in the time variable t and uncorrelated scattering in the time-delay variable τ , we may reformulate the autocorrelation function of $\tilde{h}(\tau; t)$ as

$$\begin{aligned} R_{\tilde{h}}(\tau_1, \tau_2; \Delta t) &= E[\tilde{h}^*(\tau_1; t) \tilde{h}(\tau_2; t + \Delta t)] \\ &= r_{\tilde{h}}(\tau_1; \Delta t) \delta(\tau_1 - \tau_2) \end{aligned} \quad (8.42)$$

where Δt is the difference between the observation times, and $\delta(\tau_1 - \tau_2)$ is a delta function. Using τ in place of τ_1 , the remaining function in Equation (8.42) is redefined as

$$r_{\tilde{h}}(\tau; \Delta t) = E[\tilde{h}(\tau; t) \tilde{h}^*(\tau; t + \Delta t)] \quad (8.43)$$

The function $r_{\tilde{h}}(\tau; \Delta t)$ is called the *multipath autocorrelation profile* of the channel.

Consider next a statistical characterization of the channel in terms of the complex-valued, time-varying transfer function $\tilde{H}(f; t)$. Following a formulation similar to that described in Equation (8.41), the autocorrelation function of $\tilde{H}(f; t)$ is defined by

$$R_{\tilde{H}}(f_1, t_1; f_2, t_2) = E[\tilde{H}^*(f_1; t_1) \tilde{H}(f_2; t_2)] \quad (8.44)$$

where f_1 and f_2 represent two frequencies in the spectrum of a transmitted signal. The autocorrelation function $R_{\tilde{H}}(f_1, t_1; f_2, t_2)$ provides a statistical measure of the extent to which the signal is distorted by transmission through the channel. From Equations (8.40), (8.41), and (8.44) we find that the autocorrelation functions $R_{\tilde{H}}(f_1, t_1; f_2, t_2)$ and $R_{\tilde{h}}(\tau_1, t_1; \tau_2, t_2)$ are related by a form of two-dimensional Fourier transformation as follows:

$$R_{\tilde{H}}(f_1, t_1; f_2, t_2) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} R_{\tilde{h}}(\tau_1, t_1; \tau_2, t_2) \exp[j2\pi(f_1\tau_1 - f_2\tau_2)] d\tau_1 d\tau_2 \quad (8.45)$$

Invoking stationarity in the time domain, we may reformulate Equation (8.44) as

$$R_{\tilde{H}}(f_1, f_2; \Delta t) = E[\tilde{H}^*(f_1; t) \tilde{H}(f_2; t + \Delta t)] \quad (8.46)$$

This definition suggests that the autocorrelation function $R_{\tilde{H}}(f_1, f_2; \Delta t)$ may be measured by pairs of spaced tones to carry out cross-correlation measurements on the resulting

channel outputs. Such a measurement presumes stationarity in the time domain. If we also assume stationarity in the frequency domain, we may go one step further and write

$$\begin{aligned} R_{\tilde{H}}(f, f + \Delta f; \Delta t) &= r_{\tilde{H}}(\Delta f; \Delta t) \\ &= E[\tilde{H}^*(f; t) \tilde{H}(f + \Delta f; t + \Delta t)] \end{aligned} \quad (8.47)$$

This specialized form of the autocorrelation function of $\tilde{H}(f; t)$ is in fact the Fourier transform of the multipath autocorrelation profile $r_{\tilde{h}}(\tau; \Delta t)$ with respect to the delay-time variable τ , as shown by

$$r_{\tilde{H}}(\Delta f; \Delta t) = \int_{-\infty}^{\infty} r_{\tilde{h}}(\tau; \Delta t) \exp(-j2\pi\tau \Delta f) d\tau \quad (8.48)$$

The function $r_{\tilde{H}}(\Delta f; \Delta t)$ is called the *spaced-frequency spaced-time correlation function* of the channel.

Finally, we introduce a function $S(\tau; \nu)$ that forms a Fourier-transform pair with the multipath autocorrelation profile $r_{\tilde{h}}(\tau; \Delta t)$ with respect to the variable Δt , as shown by

$$S(\tau; \nu) = \int_{-\infty}^{\infty} r_{\tilde{h}}(\tau; \Delta t) \exp(-j2\pi\nu \Delta t) d(\Delta t) \quad (8.49)$$

and

$$r_{\tilde{h}}(\tau; \Delta t) = \int_{-\infty}^{\infty} S(\tau; \nu) \exp(j2\pi\nu \Delta t) d\nu \quad (8.50)$$

The function $S(\tau; \nu)$ may also be defined in terms of $r_{\tilde{H}}(\Delta f; \Delta t)$ by applying a form of double Fourier transformation: a Fourier transform with respect to the time variable Δt and an inverse Fourier transform with respect to the frequency variable Δf . That is to say,

$$S(\tau; \nu) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} r_{\tilde{H}}(\Delta f; \Delta t) \exp(-j2\pi\nu \Delta t) \exp(j2\pi\tau \Delta f) d(\Delta t) d(\Delta f) \quad (8.51)$$

Figure 8.19 displays the functional relationships between $r_{\tilde{h}}(\tau; \Delta t)$, $r_{\tilde{H}}(\Delta f; \Delta t)$, and $S(\tau; \nu)$ in terms of the Fourier transform and its inverse.

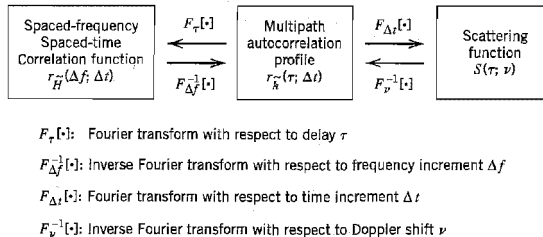


FIGURE 8.19 Functional relationships between the multipath autocorrelation profile $r_{\tilde{h}}(\tau; \Delta t)$, the spaced-frequency spaced-time correlation function $r_{\tilde{H}}(\Delta f; \Delta t)$, and the scattering function $S(\tau; \nu)$.

The function $S(\tau; \nu)$ is called the *scattering function* of the channel. For a physical interpretation of it, consider the transmission of a single tone of frequency f' (relative to the carrier). The complex envelope of the resulting filter output is

$$\tilde{s}_o(t) = \exp(j2\pi f' t) \tilde{H}(f'; t) \quad (8.52)$$

The autocorrelation function of $\tilde{s}_o(t)$ is

$$\begin{aligned} E[\tilde{s}_o^*(t) \tilde{s}_o(t + \Delta t)] &= \exp(j2\pi f' \Delta t) E[\tilde{H}^*(f'; t) \tilde{H}(f'; t + \Delta t)] \\ &= \exp(j2\pi f' \Delta t) r_{\tilde{H}}(0; \Delta t) \end{aligned} \quad (8.53)$$

where, in the last line, we have made use of Equation (8.47). Putting $\Delta f = 0$ in Equation (8.48), and then using Equation (8.50), we may write

$$\begin{aligned} r_{\tilde{H}}(0; \Delta t) &= \int_{-\infty}^{\infty} r_{\tilde{h}}(\tau; \Delta t) d\tau \\ &= \int_{-\infty}^{\infty} \left[\int_{-\infty}^{\infty} S(\tau; \nu) d\tau \right] \exp(j2\pi \nu \Delta t) d\nu \end{aligned} \quad (8.54)$$

Hence, we may view the integral

$$\int_{-\infty}^{\infty} S(\tau; \nu) d\tau$$

as the power spectral density of the channel output relative to the frequency f' of the transmitted tone, and with the Doppler shift ν acting as the frequency variable. Generalizing this result, we may state that the scattering function $S(\tau; \nu)$ provides a statistical measure of the output power of the channel, expressed as a function of the time delay τ and the Doppler shift ν .

■ DELAY SPREAD AND DOPPLER SPREAD

Putting $\Delta t = 0$ in Equation (8.43), we may write

$$\begin{aligned} P_{\tilde{h}}(\tau) &= r_{\tilde{h}}(\tau; 0) \\ &= E[|\tilde{h}(\tau; t)|^2] \end{aligned} \quad (8.55)$$

The function $P_{\tilde{h}}(\tau)$ describes the intensity (averaged over the fading fluctuations) of the scattering process at propagation delay τ . Accordingly, $P_{\tilde{h}}(\tau)$ is called the *delay power spectrum* or the *multipath intensity profile* of the channel. The delay power spectrum may also be defined in terms of the scattering function $S(\tau; \nu)$ by averaging it over all Doppler shifts. Specifically, putting $\Delta t = 0$ in Equation (8.50) and then using the first line of Equation (8.55), we may write

$$P_{\tilde{h}}(\tau) = \int_{-\infty}^{\infty} S(\tau; \nu) d\nu \quad (8.56)$$

Figure 8.20 shows an example of a delay power spectrum that depicts a typical plot of the power spectral density versus excess delay; the excess delay is measured with respect to the time delay for the shortest echo path. Note, as in Figure 8.17, the power is measured in dBm. The “threshold level” included in Figure 8.20 defines the power level below which the receiver fails to operate satisfactorily.

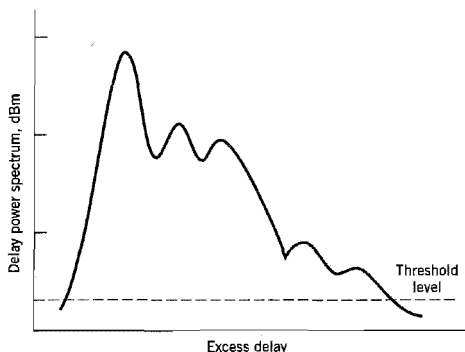


FIGURE 8.20 Example of a power-delay profile for a mobile radio channel. (From Parsons, 1992, with permission.)

Two statistical moments of $P_h(\tau)$ of interest are the *average delay*, τ_{av} , and the *delay spread*, σ_τ . The average delay is defined as the first central moment (i.e., the mean) of $P_h(\tau)$, as shown by

$$\tau_{av} = \frac{\int_0^\infty \tau P_h(\tau) d\tau}{\int_0^\infty P_h(\tau) d\tau} \quad (8.57)$$

The delay spread is defined as the square root of the second central moment of $P_h(\tau)$, as shown by

$$\sigma_\tau = \left(\frac{\int_0^\infty (\tau - \tau_{av})^2 P_h(\tau) d\tau}{\int_0^\infty P_h(\tau) d\tau} \right)^{1/2} \quad (8.58)$$

The reciprocal of the delay spread σ_τ is a measure of the *coherence bandwidth* of the channel, which is denoted by B_c .

Consider next the issue of relating the Doppler effects to time variations of the channel. For this purpose, we first set $\Delta f = 0$, which corresponds to the transmission of a single tone (of some appropriate frequency) over the channel. The spaced-frequency spaced-time correlation function of the channel then reduces to $r_H(0; \Delta t)$. Hence, evaluating the Fourier transform of this function with respect to the time variable Δt , we may write

$$S_H(\nu) = \int_{-\infty}^{\infty} r_H(0; \Delta t) \exp(-j2\pi\nu \Delta t) d(\Delta t) \quad (8.59)$$

The function $S_H(\nu)$ defines the power spectrum of the channel output expressed as a function of the Doppler shift ν ; it is therefore called the *Doppler spectrum* of the channel. The

Doppler spectrum may also be defined in terms of the scattering function by averaging it over all possible propagation delays, as shown by

$$S_{\bar{H}}(\nu) = \int_{-\infty}^{\infty} S(\tau; \nu) d\tau \quad (8.60)$$

The Doppler shift ν may assume positive and negative values with equal likelihood. The mean Doppler shift is therefore zero. The square root of the second moment of the Doppler spectrum is thus defined by

$$\sigma_{\nu} = \left(\frac{\int_{-\infty}^{\infty} \nu^2 S_{\bar{H}}(\nu) d\nu}{\int_{-\infty}^{\infty} S_{\bar{H}}(\nu) d\nu} \right)^{1/2} \quad (8.61)$$

The parameter σ_{ν} provides a measure of the width of the Doppler spectrum; it is therefore called the *Doppler spread* of the channel. The reciprocal of the Doppler spread is called the *coherence time* of the channel, which is denoted by τ_c .

Another useful parameter that is often used in measurements is the *fade rate* of the channel. For a Rayleigh fading channel, the average fade rate is related to the Doppler spread σ_{ν} as

$$f_e = 1.475 \sigma_{\nu}, \text{ crossings per second} \quad (8.62)$$

As the name implies, the fade rate provides a measure of the rapidity of fading of the channel.

Some typical values encountered in a mobile radio environment are as follows:

- ▶ The delay spread, σ_{τ} , amounts to about 20 μ s.
- ▶ The Doppler spread, σ_{ν} , due to the motion of a vehicle may extend up to 40–80 Hz.

■ CLASSIFICATION OF MULTIPATH CHANNELS

The particular form of fading experienced by a multipath channel depends on whether the channel characterization is viewed in the frequency domain or the time domain.

When the channel is viewed in the frequency domain, the parameter of concern is the channel's coherence bandwidth, B_c , which is a measure of the transmission bandwidth for which signal distortion across the channel becomes noticeable. A multipath channel is said to be *frequency selective* if the coherence bandwidth of the channel is small compared to the bandwidth of the transmitted signal. In such a situation, the channel has a filtering effect in that two sinusoidal components, with a frequency separation greater than the channel's coherence bandwidth, are treated differently. If, however, the coherence bandwidth of the channel is large compared to the message bandwidth, the fading is said to be *frequency nonselective*, or *frequency flat*.

When the channel is viewed in the time domain, the parameter of concern is the coherence time, τ_c , which provides a measure of the transmitted signal duration for which distortion across the channel becomes noticeable. The fading is said to be *time selective* if the coherence time of the channel is small compared to the duration of the received signal (i.e., the time for which the signal is in flight). For digital transmission, the received signal's duration is taken as the symbol duration plus the channel's delay spread. If, however, the channel's coherence time is large compared to the received signal duration, the fading is

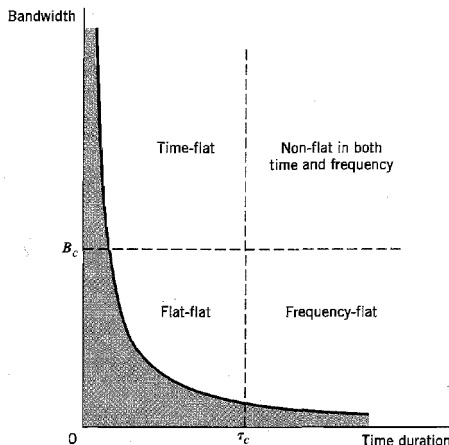


FIGURE 8.21 Illustrating the four classes of multipath channels: τ_c = coherence time, B_c = coherence bandwidth.

said to be *time nonselective*, or *time flat*, in the sense that the channel appears to the transmitted signal as time invariant.

In light of this discussion, we may classify multipath channels as follows:

- ▶ *Flat-flat channel*, which is flat in both frequency and time.
- ▶ *Frequency-flat channel*, which is flat in frequency only.
- ▶ *Time-flat channel*, which is flat in time only.
- ▶ *Nonflat channel*, which is flat neither in frequency nor in time; such a channel is sometimes referred to as a *doubly dispersive channel*.

The classification of multipath channels, based on this approach, is shown in Figure 8.21. The forbidden area, shown shaded in this figure, follows from the inverse relationship that exists between bandwidth and time duration.

8.7 Binary Signaling over a Rayleigh Fading Channel

In Chapter 6, we determined the average probability of symbol error for the transmission of binary data over a channel corrupted by additive white Gaussian noise. In a mobile radio environment, we have an additional effect to consider, namely, the fluctuations in the amplitude and phase of the received signal due to multipath effects. To be specific, consider the transmission of binary data over a Rayleigh fading channel, for which the (low-pass) complex envelope of the received signal is modeled as follows:

$$\tilde{x}(t) = \alpha \exp(-j\phi) \tilde{s}(t) + \tilde{w}(t) \quad (8.63)$$

where $\tilde{s}(t)$ is the complex envelope of the transmitted (band-pass) signal, α is a Rayleigh-distributed random variable describing the attenuation in transmission, ϕ is a uniformly

distributed random variable describing the phase-shift in transmission, and $\tilde{w}(t)$ is a complex-valued white Gaussian noise process. It is assumed that the channel is flat in both time and frequency, so that we can estimate the phase-shift ϕ from the received signal without error. Suppose then that coherent binary phase-shift keying is used to do the data transmission. Under the condition that α is fixed or constant over a bit interval, we may adapt Equation (6.20) of Chapter 6 for the situation at hand by expressing the average probability of symbol error (i.e., bit error rate) due to the additive white Gaussian noise acting alone as follows:

$$P_e(\gamma) = \frac{1}{2} \operatorname{erfc}(\sqrt{\gamma}) \quad (8.64)$$

where γ is an attenuated version of the transmitted signal energy per bit-to-noise spectral density ratio E_b/N_0 , as shown by

$$\gamma = \frac{\alpha^2 E_b}{N_0} \quad (8.65)$$

Now, insofar as a mobile radio channel is concerned, we may view $P_e(\gamma)$ as a conditional probability given that α is fixed. Thus, to evaluate the average probability of symbol error in the combined presence of fading and noise, we must average $P_e(\gamma)$ over all possible values of γ , as shown by

$$P_e = \int_0^\infty P_e(\gamma) f(\gamma) d\gamma \quad (8.66)$$

where $f(\gamma)$ is the probability density function of γ . From Equation (8.65) we note that γ depends on the squared value of α . Since α is Rayleigh distributed, we find that γ has a *chi-square distribution* with two degrees of freedom.⁷ In particular, we may express the probability density function of γ as

$$f(\gamma) = \frac{1}{\gamma_0} \exp\left(-\frac{\gamma}{\gamma_0}\right), \quad \gamma \geq 0 \quad (8.67)$$

The term γ_0 is the *mean value of the received signal energy per bit-to-noise spectral density ratio*, which is defined by

$$\begin{aligned} \gamma_0 &= E[\gamma] \\ &= \frac{E_b}{N_0} E[\alpha^2] \end{aligned} \quad (8.68)$$

where $E[\alpha^2]$ is the mean-square value of the Rayleigh-distributed random variable α . Substituting Equations (8.64) and (8.67) into (8.66), and carrying out the integration, we get the final result

$$P_e = \frac{1}{2} \left(1 - \sqrt{\frac{\gamma_0}{1 + \gamma_0}} \right) \quad (8.69)$$

Equation (8.69) defines the bit error rate for coherent binary phase-shift keying (PSK) over a flat Rayleigh fading channel. Following a similar approach, we may derive the corresponding bit error rates for coherent binary frequency-shift keying (FSK), binary differential phase-shift keying (DPSK), and noncoherent binary FSK. The results of these evaluations are summarized in Table 8.2. In Figure 8.22, we have used the exact formulas of Table 8.2 to plot the bit error rate versus γ_0 expressed in decibels. For the sake of comparison, we have also included in Figure 8.22 plots for the bit error rates of coherent binary PSK and noncoherent binary FSK for a nonfading channel. We see that Rayleigh

TABLE 8.2 Bit error rates for binary signaling over a flat-flat Rayleigh fading channel

Type of Signaling	Exact Formula for the Bit Error Rate P_e	Approximate Formula for the Bit Error Rate, Assuming Large γ_0
Coherent binary PSK	$\frac{1}{2} \left(1 - \sqrt{\frac{\gamma_0}{1 + \gamma_0}} \right)$	$\frac{1}{4\gamma_0}$
Coherent binary FSK	$\frac{1}{2} \left(1 - \sqrt{\frac{\gamma_0}{2 + \gamma_0}} \right)$	$\frac{1}{2\gamma_0}$
Binary DPSK	$\frac{1}{2(1 + \gamma_0)}$	$\frac{1}{2\gamma_0}$
Noncoherent binary FSK	$\frac{1}{2 + \gamma_0}$	$\frac{1}{\gamma_0}$

fading results in a severe degradation in the noise performance of a digital passband transmission system, the degradation being measured in tens of decibels of additional mean signal-to-noise ratio compared to a nonfading channel for the same bit error rate. In particular, for large γ_0 we may derive the approximate formulas given in the last column of Table 8.2, according to which the asymptotic decrease in the bit error rate with the average signal energy per bit-to-noise spectral density ratio γ_0 follows an *inverse* law. This behavior is dramatically different from the case of a nonfading channel, for which the asymptotic decrease in the bit error rate with γ_0 follows an *exponential* law.

The practical implication of this difference is that in a mobile radio environment, we have to provide a large increase in mean signal-to-noise ratio (relative to a nonfading environment), so as to ensure a bit error rate that is low enough for practical use. To meet such a requirement, we have to increase the transmitted power, antenna size, and so on, which can be costly in terms of implementation. Alternatively, we may utilize special modulation and reception techniques that are less vulnerable to fading effects. Among these techniques, the best known and most widely used are the multiple-receiver combining techniques referred to collectively as *diversity*, a brief discussion of which is presented next.

■ DIVERSITY TECHNIQUES

Diversity may be viewed as a form of redundancy. In particular, if several replicas of the message signal can be transmitted simultaneously over independently fading channels, then there is a good likelihood that at least one of the received signals will not be severely degraded by fading. There are several methods for making such a provision. In the context of our present discussion, the following diversity techniques are of particular interest:

- ▶ Frequency diversity
- ▶ Time (signal-repetition) diversity
- ▶ Space diversity

In *frequency diversity*, the message signal is transmitted using several carriers that are spaced sufficiently apart from each other to provide independently fading versions of

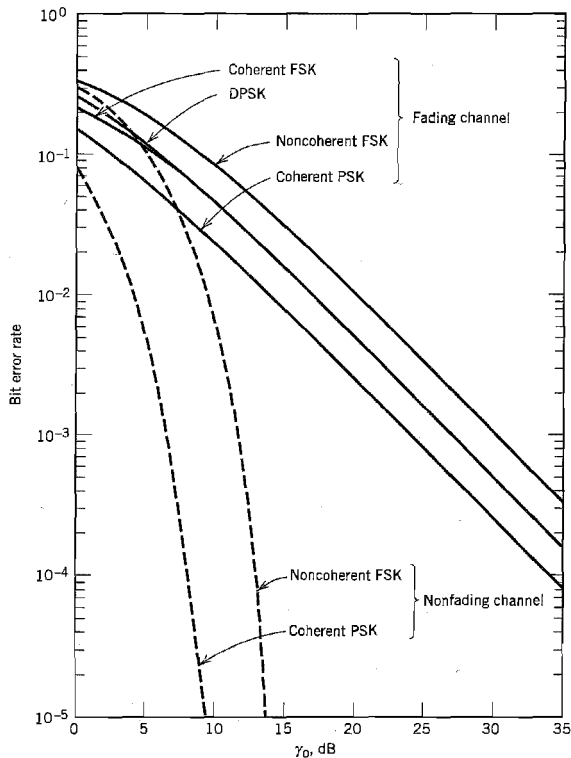


FIGURE 8.22 Performance of binary signaling schemes over a Rayleigh fading channel, shown as continuous curves; the dashed curves pertain to a nonfading channel.

the signal. This may be accomplished by choosing a frequency spacing equal to or larger than the coherence bandwidth of the channel.

In *time diversity*, the same message signal is transmitted in different time slots, with the spacing between successive time slots being equal to or greater than the coherence time of the channel. Time diversity may be likened to the use of a repetition code for error-control coding. (Error-control coding is discussed in Chapter 10.)

In *space diversity*, multiple transmitting or receiving antennas (or both) are used, with the spacing between adjacent antennas being chosen so as to assure the independence of fading events; this may be satisfied by spacing the adjacent antennas by at least seven times the radio wavelength.

Given that by one of these means we create L independently fading channels, we may then use a *linear diversity combining structure* involving L separate receivers, as depicted in Figure 8.23. The system is designed to compensate only for *short-term effects* of a fading channel. Moreover, it is assumed that *noise-free estimates* of the channel attenuation factors $\{\alpha_\ell\}$ and the channel phase-shifts $\{\phi_\ell\}$ are available. Then, the linear combiner achieves optimum performance for binary data transmission (discussed here for

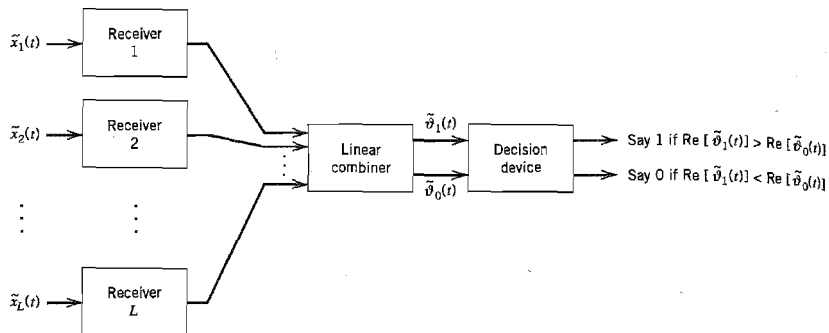


FIGURE 8.23 Block diagram illustrating the space diversity technique.

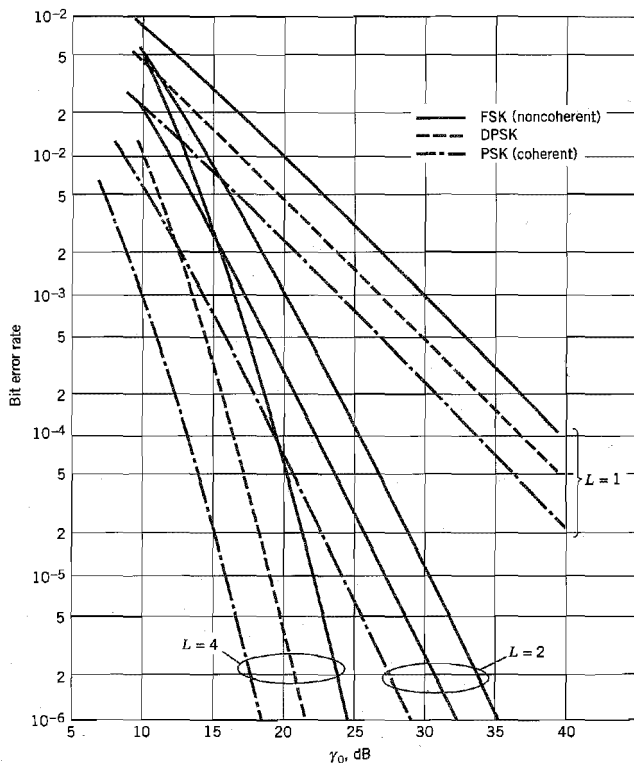


FIGURE 8.24 Performance of binary signaling schemes with diversity. (From Proakis, 1995, with permission of McGraw-Hill.)

the purpose of illustration) by proceeding as follows: The output of the k th matched filter in the ℓ th receiver, $\tilde{v}_{\ell k}(t)$, is multiplied by $\alpha_{\ell} \exp(j\phi_{\ell})$ that represents the complex conjugate of the ℓ th channel gain, where $\ell = 1, 2, \dots, L$, and $k = 0, 1$. Thus, the linear combiner results in two output complex envelopes defined by

$$\tilde{v}_k(t) = \sum_{\ell=1}^L \alpha_{\ell} \exp(j\phi_{\ell}) \tilde{v}_{\ell k}(t), \quad k = 0, 1 \quad (8.70)$$

according to which $\alpha_{\ell} \exp(j\phi_{\ell})$ plays the role of a *weighting factor*. One output complex envelope $\tilde{v}_0(t)$ corresponds to the transmission of symbol 0, and the other $\tilde{v}_1(t)$ corresponds to the transmission of symbol 1. The real parts of $\tilde{v}_0(t)$ and $\tilde{v}_1(t)$ are then used in the decision-making process. The situation described here applies to binary FSK. In the case of binary PSK, only a single matched filter is needed, in which case the linear combiner produces a single output complex envelope. Here again, however, the real part of the combiner output is used in the decision-making process.

In the linear combiner described herein, the “instantaneous” output signal-to-noise ratio (SNR) is the sum of the instantaneous SNRs on the individual diversity branches (channels). This optimum form of a linear combiner is therefore referred to as a *maximal-ratio combiner*; see Problem 8.17.

Figure 8.24 shows the noise performance of coherent binary PSK, binary DPSK, and noncoherent binary FSK for $L = 2, 4$ independently fading channels. For the sake of comparison, we have also included in this figure the corresponding graphs for a fading channel with no diversity (i.e., $L = 1$). Figure 8.24 clearly illustrates the effectiveness of diversity as a means of mitigating the short-term effects of Rayleigh fading.

8.8 TDMA and CDMA Wireless Communication Systems⁸

In wireless communications, as with ordinary telephony, a user would like to talk and listen simultaneously. To cater to this natural desire, some form of duplexing is required. One way in which this requirement can be satisfied is to provide two frequency bands, one for the forward link from the base station to a mobile and the other for the reverse link from the mobile to the base station. As pointed out earlier, in North America the subband 869–894 MHz is used for the forward link, and the subband 824–849 MHz is used for the reverse link. This form of duplexing is called *frequency division duplexing* (FDD). Indeed, FDD is an integral part of the two widely used wireless communication systems summarized in Table 8.3.

The first of these systems, namely, GSM, uses TDMA. From Section 8.2 we recall that in a TDMA system each subscriber is permitted to access the radio channel during a set of predetermined time slots, during which time that particular subscriber will have full use of the channel. Consequently, data are transmitted over the channel in *bursts*, as shown in the *frame structure* of Figure 8.25. The basic frame of GSM is composed of eight 577 μ s slots. The 1-bit *flag* preceding each data burst of 57 bits is used to identify whether the data bits are digitized speech or some other information-bearing signal. The 3 *tail* bits, all logical zeros, are used in convolutional decoding of the channel-encoded data bits. (Convolutional codes are discussed in Chapter 10.) The 26-bit training sequence in the middle of the time slot is used for channel equalization. Finally, the *guard time*, occupying 8.25 bits, is included at the end of each slot to prevent data bursts received at the base

TABLE 8.3 Summary of two widely used wireless communication systems

Item	GSM*	IS-95†	Comments
Number of duplex channels	125	20	CDMA assumes 12.5 MHz in each direction; see the next line
Channel bandwidth (kHz)	200	1,250	
Type of multiple access	TDMA	CDMA	
Access users per channel	8	20 to 35	A TDMA system is <i>deterministic</i> in that the number of access users per channel is defined by the number of available time slots. On the other hand, a CDMA system is <i>interference-limited</i> in that it has a <i>soft</i> limit on the number of access users per channel.
Modulation type	GMSK	BPSK/QPSK	In CDMA, data are modulated as BPSK, but the spreading is QPSK
Data rate (kb/s)	270.833	9.6 or 14.4	
Frame period (ms)	4.615	20	For CDMA, the frame period equals that of the speech codec (coder/decoder)

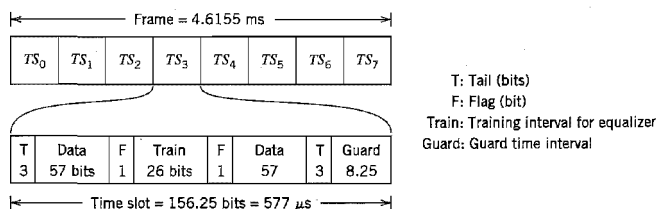
*GSM stands for *Global System for Mobile Communications*; originally, it was introduced as an acronym for *Groupe de travail Spéciale pour les services Mobiles*.

†IS stands for *Interim Standard*.

station from mobiles from overlapping with each other; this is achieved by transmitting no signal at all during the guard time. With each slot consisting of 156.25 bits, of which 40.25 bits are overhead (ignoring the 2 flag bits), the *frame efficiency* of GSM is

$$\left(1 - \frac{40.25}{156.25}\right) \times 100 = 74.24\%$$

The second wireless communication system, IS-95, summarized in Table 8.3 uses CDMA. From Section 8.2 we recall that in CDMA, each subscriber is assigned a distinct spreading code (PN sequence), thereby permitting the subscriber full access to the channel all of the time. Consequently, in a CDMA system we have a new form of interference called *multiple-access interference* (MAI), which arises because of deviation of the spreading codes from perfect orthogonality. A related phenomenon that needs attention is the *near-far problem*, which occurs if the received signals from the mobile units do not have equal power at the base station. In such a situation, the strongest received signal from a mobile user captures the demodulation process at the base station to the detriment of the

**FIGURE 8.25** Frame structure of the GSM wireless communication system.

other users. To overcome the near-far problem, it is customary to use *power control* at the base station, whereby the base station maintains control over the power level of the transmitted signal from every mobile being served by that base station. The use of power control is particularly important in CDMA systems for another reason. A goal of multiple-access systems is to maximize *system capacity*, which is defined as the largest possible number of users that can be reliably served by the system, given prescribed resources. Clearly, system capacity is compromised if each mobile is free to raise its transmitted power level regardless of other users, since that increase in transmitted power will, in turn, raise the level of multiple-access interference in the system. To maximize system capacity, it is therefore essential that each mobile's transmitter be under the control of the serving base station so that the signal-to-interference ratio is maintained at the minimum acceptable level needed for reliable service.

RAKE RECEIVER

A discussion of wireless communications using CDMA would be incomplete without a description of the *RAKE receiver*.⁹ The RAKE receiver was originally developed in the 1950s as a "diversity" receiver designed expressly to equalize the effect of multipath. First, and foremost, it is recognized that useful information about the transmitted signal is contained in the multipath component of the received signal. Thus, taking the viewpoint that multipath may be approximated as a linear combination of differently delayed echoes, the RAKE receiver seeks to combat the effect of multipath by using a correlation method to detect the echo signals individually and then adding them algebraically. In this way, intersymbol interference due to multipath is dealt with by reinserting different delays into the detected echoes so that they perform a constructive rather than destructive role.

Figure 8.26 shows the basic idea behind the RAKE receiver. The receiver consists of a number of *correlators* connected in parallel and operating in a synchronous fashion. Each correlator has two inputs: (1) a delayed version of the received signal and (2) a replica of the pseudo-noise (PN) sequence used as the spreading code to generate the spread-

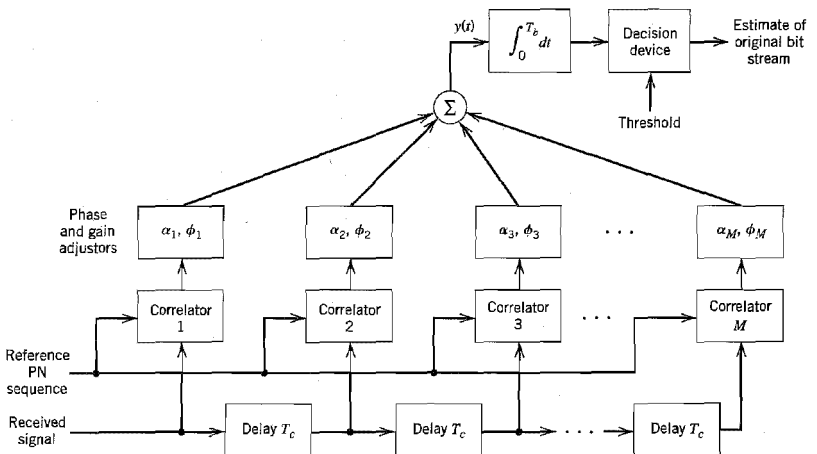


FIGURE 8.26 Block diagram of the RAKE receiver.

spectrum modulated signal at the transmitter. In effect, the PN sequence acts as a “reference signal.” Let the nominal bandwidth of the PN sequence be denoted as $W = 1/T_c$, where T_c is the chip duration. From the discussion of spread-spectrum modulation presented in Chapter 7, we recall that the autocorrelation function of a PN sequence has a single peak of width $1/W$, and it disappears toward zero elsewhere inside one period of the PN sequence (i.e., one symbol period). Thus we need only make the bandwidth W of the PN sequence sufficiently large to “identify” the significant echoes in the received signal. To be sure that the correlator outputs all add constructively, two other operations are performed in the receiver by the functional blocks labeled “phase and gain adjusters”:

1. An appropriate delay is introduced into each correlator output so that the phase angles of the correlator outputs are in agreement with each other.
2. The correlator outputs are weighted so that the correlators responding to strong paths in the multipath environment have their contributions accentuated, while the correlators not synchronizing with any significant path are correspondingly suppressed.

The weighting coefficients, α_k , are computed in accordance with the *maximal ratio combining principle*.¹⁰

The signal-to-noise ratio of a weighted sum, where each element of the sum consists of a signal plus additive noise of fixed power, is maximized when the amplitude weighting is performed in proportion to the pertinent signal strength.

The linear combiner output is

$$y(t) = \sum_{k=1}^M \alpha_k z_k(t) \quad (8.71)$$

where $z_k(t)$ is the phase-compensated output of the k th correlator, and M is the number of correlators in the receiver. Provided we use enough correlators in the receiver to span a region of delays sufficiently wide to encompass all the significant echoes that are likely to occur in the multipath environment, the output $y(t)$ behaves essentially as though there was a single propagation path between the transmitter and receiver rather than a series of multiple paths spread in time.

To simplify the presentation, the receiver of Figure 8.26 assumes the use of binary phase-shift keying in performing spread-spectrum modulation at the transmitter. Thus the final operation performed in Figure 8.26 is that of integrating the linear combiner output $y(t)$ over the bit interval T_b and then determining whether binary symbol 1 or 0 was transmitted in that bit interval.

The RAKE receiver derives its name from the fact that the bank of parallel correlators has an appearance similar to the fingers of a rake. Because spread spectrum modulation is basic to the operation of CDMA wireless communications, it is natural for the RAKE receiver to be central to the design of the receiver used in this type of multiuser radio communication.¹¹

8.9 Source Coding of Speech for Wireless Communications

For the efficient use of channel bandwidth, digital wireless communication systems, be they of the TDMA or CDMA type, rely on the use of *speech coding* to remove almost all

of the natural redundancy in speech, while maintaining a high-quality speech on decoding. The common approach is to use source coding, which, in one form or another, exploits the *linear predictive coding (LPC)* of speech.

In this section, we describe two different techniques for speech coding: multi-pulse excited LPC and code-excited LPC, versions of which are used in GSM and IS-95, respectively. Our treatment of both of these speech coding techniques is in conceptual terms.¹²

■ MULTI-PULSE EXCITED LPC

This form of speech coding exploits the *principle of analysis by synthesis*, which means that the encoder includes a replica of the decoder in its design. Specifically, the encoder consists of three main parts as indicated in Figure 8.27a:

1. *Synthesis filter* for the predictive modeling of speech. It may consist of an all-pole filter (i.e., a filter whose transfer function has poles only), which is designed to model the *short-term* spectral envelope of speech; the term *short-term* refers to the fact that the filter parameters are computed on the basis of predicting the present sample of the speech signal using eight to sixteen previous samples. The synthesis filter may also include a *long-term* predictor for modeling the fine structure of the speech spectrum; in such a case, the long-term predictor is connected in cascade with the short-term predictor. In any event, the function of the synthesis filter is to produce a synthetic version of the original speech that is of high quality.
2. *Excitation generator* for producing the excitation applied to the synthesis filter. The excitation consists of a definite number of pulses every 5 to 15 ms. The amplitudes and positions of the individual pulses are adjustable.
3. *Error minimization* for optimizing the perceptually weighted error between the original speech and synthesized speech. The aim of this minimization is to optimize the amplitudes and positions of the pulses used in the excitation. Typically, a mean-square error criterion is used for the minimization.

Thus, as shown in Figure 8.27a, the three parts of the encoder form a *closed-loop* optimization procedure, which permits the encoder to operate at a bit rate below 16 kb/s, while maintaining high-quality speech.

The encoding procedure itself has two main steps:

- The free parameters of the synthesis filter are computed using the actual speech samples as input. This computation is performed outside the optimization loop over

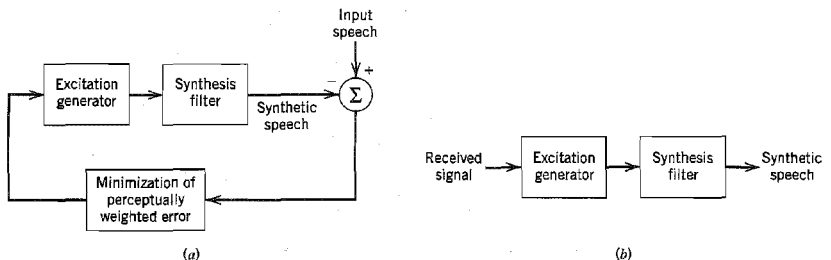


FIGURE 8.27 Multi-pulse excited linear predictive codec. (a) Encoder. (b) Decoder whose input (the received signal) consists of quantized filter parameters and quantized excitation as produced by the encoder.

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 8.27a.

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, albeit 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 8.27b. 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

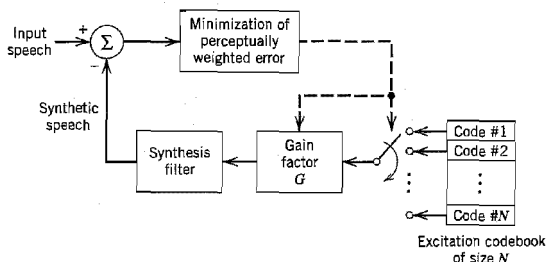


FIGURE 8.28 Encoder of the code-excited linear predictive codec (CELP): the transmitted signal consists of the address of the code selected from the codebook, quantized G , and quantized filter parameters.

the original speech and synthesized speech (i.e., output of the synthesis filter). The address of the stochastic vector selected from the codebook and the corresponding quantized gain factor, together with the quantized filter parameters, constitute the transmitted signal.

An identical copy of the codebook is made available to the decoder, and likewise for the synthesis filter. Hence, given the received signal, the decoder is enabled to parameterize its own synthesis filter and determine the appropriate excitation for the synthesis filter, thereby producing a synthetic version of the original speech signal.

CELP is capable of producing good-quality speech at bit rates below 8 kb/s. However, its computational complexity is intensive because of the exhaustive search of the excitation codebook. In particular, the weighted synthesized speech in the encoder has to be computed for all the entries in the codebook and then compared with the weighted original speech. Nevertheless, real-time implementation of CELP codecs has been made possible by virtue of advances in digital signal processing and VLSI technology.

8.10 Adaptive Antenna Arrays for Wireless Communications¹³

The goal of wireless communications is to allow as many users as possible to communicate reliably without regard to location and mobility. From the discussion presented in Sections 8.5 and 8.6, we find that this goal is seriously impeded by three major channel impairments:

1. *Multipath* can cause severe fading due to phase cancellation between different propagation paths. Fading leads to a reduction in available signal power and therefore a degraded noise performance.
2. *Delay spread* results from differences in propagation delays among the multiple propagation paths. When the delay spread exceeds about 10 percent of the symbol duration, the intersymbol interference experienced by the received signal reaches a significant level, thereby causing a reduction in the attainable data rate.
3. *Co-channel interference* arises in cellular systems where the available frequency channels are divided into different sets, with each set being assigned to a specific cell and with several cells in the system using the same set of frequencies. Co-channel interference limits the *system capacity* (i.e., the largest possible number of users that can be reliably served by the system).

Typically, cellular systems use 120° sectorization at each base station, and only one user accesses a sector of a base station at a given frequency. We may combat the effects of multipath fading and co-channel interference at the base station by using three identical but separate *antenna arrays*, one for each section of the base station. The compensation of delay spread is considered later in the section. Figure 8.29 shows the block diagram of an *array signal processor*, where it is assumed that there are N users whose signals are received at a particular sector of the base station, and the array for that sector consists of M identical antenna elements. A particular user is treated as the one of interest, and the remaining $N-1$ users give rise to co-channel interference. In addition to the co-channel interference, each component of the array signal processor's input is corrupted by additive white Gaussian noise (AWGN). The analysis presented herein is for baseband signals, which, in general, are complex valued. This, in turn, means that both the channel and array signal processor require complex characterizations of their own. The structure depicted in Figure 8.29 is drawn for one output pertaining to the user of

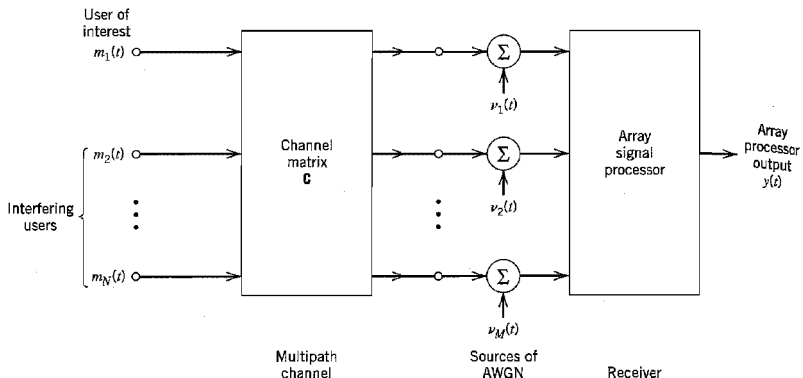


FIGURE 8.29 Block diagram of array signal processor that involves M antenna elements, and that is being driven by a multipath channel.

interest. The array signal processor is duplicated for users at other frequencies at the base station.

The multipath channel is characterized by the channel matrix, which is denoted by \mathbf{C} . The matrix \mathbf{C} has dimensions M -by- N and may therefore be expanded into N column vectors, as shown by

$$\mathbf{C} = [\mathbf{c}_1, \mathbf{c}_2, \dots, \mathbf{c}_N] \quad (8.72)$$

where each column vector is of dimension M .

Given the configuration described in Figure 8.29, the goal is to design a *linear array signal processor* for the receiver, which satisfies two requirements:

1. The co-channel interference produced by the $N-1$ interfering users is cancelled.
2. The output signal-to-noise ratio (SNR) for the user of interest is maximized.

Hereafter, these two requirements are referred to as design requirements 1 and 2.

To proceed with this design task, it is assumed that the multipath channel is described by flat Rayleigh fading. Then, in light of the material presented in Section 8.7, we find that the use of diversity permits the treatment of the column vectors $\mathbf{c}_1, \mathbf{c}_2, \dots, \mathbf{c}_N$ as *linearly independent*, which is justified provided that the spacing between antenna elements of the array is large enough (e.g., seven times the wavelength) for independent fading. To simplify the presentation, we suppose that user 1 is the user of interest and the remaining $N-1$ users are responsible for co-channel interference, as indicated in Figure 8.29. The key design issue is how to find the *weight vector* denoted by \mathbf{w} , which characterizes the array signal processor. To that end, we may proceed as follows:

1. We choose the M -dimensional weight vector \mathbf{w} to be orthogonal to the vectors $\mathbf{c}_2, \dots, \mathbf{c}_N$, which are associated with the interfering users. This choice fulfills design requirement 1 (i.e., cancellation of co-channel interference).
2. To satisfy design requirement 2 (i.e., maximization of the SNR), we will briefly digress from the issue at hand to introduce the notion of a subspace. Given a *vector*

space, or just *space*, formed by a set of linearly independent vectors, a *subspace* of the space is a subset that satisfies two conditions:¹⁴

- (i) If we add any two vectors \mathbf{z}_1 and \mathbf{z}_2 in the subspace, their sum $\mathbf{z}_1 + \mathbf{z}_2$ is still in the subspace.
- (ii) If we multiply any vector \mathbf{z} in the subspace by any scalar a , the multiple $a\mathbf{z}$ is still in the subspace.

Returning to the issue of how to maximize the output SNR for user 1, we first construct a subspace denoted by \mathcal{W} , whose dimension is equal to the difference between the number of antenna elements and the number of interfering users, that is, $M - (N - 1) = M - N + 1$. Next, we project the complex conjugate of the channel vector \mathbf{c}_1 (pertaining to user 1) onto the subspace \mathcal{W} . The projection so computed defines the weight vector \mathbf{w} .

► EXAMPLE 8.3

To illustrate the two-step subspace method for determining the weight vector \mathbf{w} , consider the simple example of a system involving two users characterized by the channel vectors \mathbf{c}_1 and \mathbf{c}_2 , and an antenna array consisting of three elements; that is, $N = 2$ and $M = 3$. Then, for this example, the subspace \mathcal{W} is two-dimensional, as shown by

$$M - N + 1 = 3 - 2 + 1 = 2$$

With user 1 viewed as the user of interest and user 2 viewed as the interferer, we may construct the signal-space diagram shown in Figure 8.30. The subspace \mathcal{W} , shown shaded in this figure, is orthogonal to channel vector \mathbf{c}_2 . The weight vector \mathbf{w} of the array signal processor is determined by the projection of the complex-conjugated channel vector of user 1, that is, \mathbf{c}_1^* , onto the subspace \mathcal{W} , as depicted in Figure 8.30. ▲

The important conclusion drawn from this discussion is that a linear receiver using optimum combining with M antenna elements and involving $N - 1$ interfering users has the same performance as a linear receiver with $M - N + 1$ antenna elements without interference, independent of the multipath environment. For this equivalence to be realized,

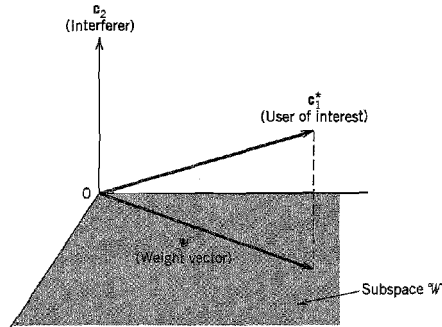


FIGURE 8.30 Signal-space diagram for Example 8.3, involving a user of interest, a single interferer, and an antenna array of 3 elements. The subspace \mathcal{W} , shown shaded, is two-dimensional in this example.

we of course require that $M > N - 1$. Provided that this condition is satisfied, the receiver cancels the co-channel interference with a diversity improvement equal to $M - N + 1$, which represents an N -fold increase in system capacity.

The design of an array signal processor in accordance with the two-step subspace procedure described herein is of the *zero-forcing* kind. We say so because, given M antenna elements, the array has enough degrees of freedom to *force* the output due to the $N - 1$ interfering users represented by the linearly independent channel vectors $\mathbf{c}_2, \dots, \mathbf{c}_M$ to zero so long as M is greater than $N - 1$. Note also that this procedure includes $N = 1$ (i.e., a single user with no interfering users) as a special case. In this case, the channel matrix consists of vector \mathbf{c}_1 , which lies in the subspace \mathcal{W} , and the zero-forcing solution \mathbf{w} equals \mathbf{c}_1^* .

The analysis presented thus far has been entirely of a *spatial* kind, which ignores the effect of delay spread. What if the delay spread is significant compared to the symbol duration and cannot therefore be ignored? Recognizing that delay spread is responsible for intersymbol interference, we may, in light of the material presented in Chapter 4 on the equalization of a telephone channel, incorporate a *linear* equalizer in each antenna branch of the array to compensate for delay spread. The resulting array signal processor takes the form shown in Figure 8.31, which combines temporal and spatial processing.

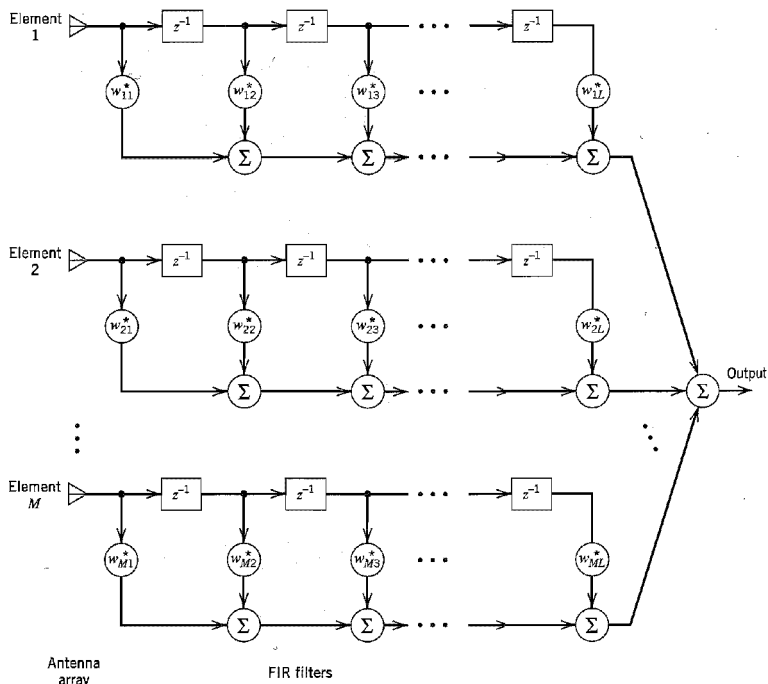


FIGURE 8.31 Baseband space-time processor. The blocks labeled z^{-1} are unit-delay elements with each delay being equal to the symbol period. The filter coefficients are complex valued. The FIR filters are all assumed to be of length L .

Spatial processing is provided by the antenna array, and the temporal processing is provided by a bank of finite-duration impulse response (FIR) filters. For obvious reasons, this structure is called a *space-time processor*.¹⁵

■ ADAPTIVE ANTENNA ARRAY

The subspace design procedure for the array signal processor in Figure 8.29 assumes that the channel impairments are stationary, and that we have knowledge of the channel matrix C . In reality, however, multipath fading, delay spread, and co-channel interference are all nonstationary in their own individual ways. Also, the channel characterization may be unknown. To deal with these practical issues, we need to make the receiving array signal processor in Figure 8.29 adaptive. Bearing in mind the scope of this book, we confine the discussion to adaptive spatial processing, assuming that the delay spread is negligible. We further assume that the multipath fading phenomenon is slow enough to justify the *least-mean-square (LMS) algorithm* to perform the adaptation.

Figure 8.32 shows the structure of an *adaptive antenna array*, where the output of each antenna element is multiplied by an adjustable (controllable) weight, and then the weighted elemental outputs of the array are summed to produce the array output signal. The adaptive antenna array does not require knowledge of the direction of arrival of the desired signal originating from a user of interest as long as the system is supplied with a *reference signal*, which is *correlated* with the desired signal. The output signal of the array is subtracted from the reference signal to generate an *error signal*, which is used to apply the appropriate adjustments to the elemental weights of the array. In this way, a feedback system to control the elemental weights is built into the operation of the antenna array, thereby making it adaptive to changes in the environment. Note that the block diagram of Figure 8.32 is drawn for baseband processing, hence the complex conjugation of the elemental weights. In a practical system, a quadrature hybrid is used for each antenna element of the array to split the complex-valued received signal at each element into two components: one real and the other imaginary. The use of a hybrid has been omitted in Figure 8.32 to simplify the diagram.

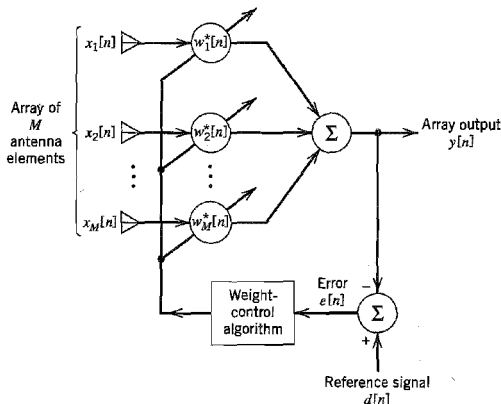


FIGURE 8.32 Block diagram of adaptive antenna array.

To optimize the performance of the adaptive antenna array, it is customary to use the *mean-square error*

$$J = E[|e[n]|^2] \quad (8.73)$$

as the cost function to be *minimized*. The $e[n]$ is the error signal at time $t = nT$, where T is the symbol period and n is an integer serving as discrete time. Minimization of the cost function J suppresses the interfering signals and enhances the desired signal in the array output. However, the LMS algorithm minimizes the instantaneous value of the cost function J and, through successive iterations, it strives to reach the minimum mean-square error (MMSE) (i.e., optimum solution for the elemental weights). In light of the discussion presented in Chapter 4 on temporal equalizers, which carries over to the spatial domain, we may say that an adaptive antenna array based on the minimum mean-square error criterion is highly likely to provide a better solution than one based on the zero-forcing criterion embodied in the two-step subspace method.

Let $x_k[n]$ denote the output of the k th element in the array at discrete time n , and let $w_k[n]$ denote the corresponding value of the weight connected to this element. The output signal of the array (consisting of M antenna elements) is therefore

$$y[n] = \sum_{k=1}^M w_k^*[n]x_k[n] \quad (8.74)$$

where $w_k^*[n]x_k[n]$ is the inner product of the complex-valued quantities $w_k[n]$ and $x_k[n]$. Denoting the reference signal as $d[n]$, we may evaluate the error signal as

$$e[n] = d[n] - y[n] \quad (8.75)$$

Hence, the adjustment applied to the k th elemental weight is

$$\Delta w_k[n] = \mu e^*[n]x_k[n], \quad k = 1, 2, \dots, M \quad (8.76)$$

where μ is the *step-size parameter*, and the updated value of this weight is

$$w_k[n+1] = w_k[n] + \Delta w_k[n], \quad k = 1, 2, \dots, M \quad (8.77)$$

Equations (8.74)–(8.77), in that order, constitute the *complex LMS algorithm*, which includes the LMS algorithm for real signals (studied in Chapters 3 and 4) as a special case. The algorithm is initiated by setting $w_k[0] = 0$ for all k . The derivation of the complex LMS algorithm is posed as Problem 8.19.

The advantages of an adaptive antenna array using the complex LMS algorithm are three-fold:

- ▶ Simplicity of implementation.
- ▶ Linear growth in complexity with the number of antenna elements.
- ▶ Robust performance with respect to disturbances.

However, the system suffers from the following drawbacks:

- ▶ Slow rate of convergence, which is typically ten times the number of weights. This limits the use of the complex LMS algorithm to a slow-fading environment, for which the Doppler spread is small compared to the reciprocal of the duration of the observation interval.
- ▶ Sensitivity of the convergence behavior to variations in the reference signal and co-channel interference powers.

These limitations of the complex LMS algorithm can be overcome by using an algorithm known as *direct matrix inversion* (DMI), which follows directly from the Wiener filter discussed in Chapter 4; see Problem 8.21. Unlike the LMS algorithm, the DMI al-

gorithm operates in the *batch* mode in that the computation of the elemental weights is based on a batch of K snapshots. The batch size K is chosen as a compromise between two conflicting requirements:

- ▶ The size K should be small enough for the batch of snapshots used in the computation to be justifiably treated as pseudo-stationary.
- ▶ The size K should be large enough for the computed values of the elemental weights to approach the MMSE solution.

The DMI algorithm is the optimum combining technique for array antennas currently deployed in many base stations today. The DMI algorithm may be reformulated for recursive computation,¹⁶ if so desired.

When the teletraffic is high, the base stations are ordinarily configured as microcells, which are small cells such as an office floor or a station deployed along a highway with directional antennas. In such a configuration, there are many inexpensive base stations in close proximity to each other. The use of adaptive antenna arrays provides the means for an alternative configuration where there are fewer (but more expensive) base stations and further apart from each other than in the corresponding microcellular system.

8.11 Summary and Discussion

In this chapter, we discussed two important types of multiuser communications: satellite communications and wireless communications. Satellite communication systems offer global coverage, whereas wireless communication systems offer mobility. The global coverage and mobility offered by these two communication systems have profoundly transformed the way we communicate, both locally and globally.

Although satellite communication and wireless communication systems function in entirely different ways, both rely on radio propagation to link the receiver to the transmitter. In satellite communications, we have an *uplink* from an earth terminal to the satellite transponder and a *downlink* from the satellite to another earth terminal. The satellite operates like a repeater in the sky. Moreover, with the satellite positioned in a geostationary orbit, the uplink and downlink operate as line-of-sight paths of fixed lengths. Accordingly, the satellite communication channel, encompassing both of these links, is closely modeled as an *additive white Gaussian noise (AWGN) channel*.

The wireless communication system also has two links of its own: an *uplink*, or *reverse link*, for the mobile-to-base station transmission, and a *downlink*, or *forward link*, for the base station-to-mobile transmission. The base station is fixed, being located at the center or on the edge of a coverage region; it consists of radio channels, and transmitter, and receiver antennas mounted on a tower. Three major sources of degradation in wireless communications, discussed in the chapter, are co-channel interference, fading, and delay spread; the latter two are byproducts of multipath. A common characteristic of these channel impairments is that they are all *signal-dependent phenomena*. Unlike the ubiquitous channel noise, the degrading effects of interference and multipath cannot therefore be combatted by simply increasing the transmitted signal power. Rather, both interference and multipath require the use of specialized techniques, tailor-made to their particular physical characteristics. These specialized techniques include diversity, adaptive array antennas, and the RAKE receiver.

We close the discussion with remarks contrasting wireless communications to wired communications. From Chapter 3 we recall that a major source of concern in wired com-

munication systems is noise; these systems have sufficient channel bandwidth to permit the use of pulse-code modulation (PCM) as the standard method for converting speech into a 64 kb/s stream, which provides the basic data for an almost noise-free performance. In wireless communications, on the other hand, channel bandwidth is a precious resource, the conservation of which necessitates the use of spectrally efficient speech coding techniques to produce toll-quality digitized speech at rates that are a small fraction of the PCM rate. Unfortunately, the waveform coders exemplified by adaptive differential pulse-code modulation, discussed in Chapter 3, do not satisfy this stringent requirement. The preferred approach is to use the spectrally efficient source-coding techniques: multi-pulse excited linear predictive coding (LPC) or its regular-pulse excited variant, and code-excited LPC (CELP); these source coding techniques produce bit rates below 16 kb/s by removing almost all of the natural redundancy in speech, while maintaining high-quality speech, albeit of a synthetic kind. To provide protection against noise, channel coding is used whereby redundant bits are inserted into the transmitted data stream in a controlled manner. The use of channel coding also helps in other ways: It extends the range of low-power handsets as well as battery life. Channel coding is discussed in Chapter 10.

NOTES AND REFERENCES

1. For detailed treatment of satellite communications and related issues, see the following books: Sklar (1988), Pratt and Bostian (1986), Wu (1984), Bhargava et al. (1981), and Spilker, Jr. (1977). The first, third, fourth, and fifth books emphasize the use of satellites for digital communications. The book by Pratt and Bostian presents a broad treatment of satellite communications, emphasizing such diverse topics as radio-wave propagation, antennas, orbital mechanics, signal processing, and radio electronics.
2. Link budget analysis is discussed in the books by Sklar (1988) and Anderson (1999); for satellite communications, it is discussed in Bhargava et al. (1981).
3. For the fundamentals of antennas, see the book by Kraus (1950) and Chapter 11 of the book by Jordan and Balmain (1973).
4. The free-space equation (Equation 8.14) is named in honor of Friis (1946). For the origin of the Friis formula of Equation (8.30), see Friis (1944).
5. For an original treatment of cellular radio, see the paper by MacDonald (1979).
6. For a comprehensive treatment of the mobile radio propagation channel, see the book by Parsons (1992). This book presents the fundamentals of VHF and UHF propagation, propagation over irregular terrain and in built-up areas, and a statistical characterization of the mobile radio channel. The statistical characterization of a mobile radio channel is also discussed in Proakis (1995). This book provides a readable account of the effect of fading on the error performance of Rayleigh fading channels and a good discussion of diversity techniques. For a full treatment of the subject, see Chapters 9–11 by Stein in the book edited by Schwartz, Bennett, and Stein (1966).
7. The chi-square distribution is a special case of the gamma distribution. The probability density function of a *gamma-distributed random variable* X has two parameters: $\alpha > 0$ and $\lambda > 0$; it is defined by

$$f_X(x) = \frac{\lambda(\lambda x)^{\alpha-1} e^{-\lambda x}}{\Gamma(\alpha)}, \quad 0 < x < \infty$$

where $\Gamma(\alpha)$ is the *gamma function*, which is itself defined by

$$\Gamma(\alpha) = \int_0^{\infty} z^{\alpha-1} e^{-z} dz, \quad \alpha > 0$$

The gamma function has the following properties:

$$\Gamma(1/2) = \sqrt{\pi}$$

$$\Gamma(\alpha + 1) = \alpha\Gamma(\alpha), \quad \alpha > 0$$

By letting $\lambda = 1/2$ and $\alpha = k/2$, where k is a positive integer, we get the *chi-square distribution* with $2k$ degrees of freedom, as shown by

$$f_X(x) = \frac{x^{(k-2)/2} e^{-x/2}}{2^{k/2} \Gamma(k/2)}, \quad 0 < x < \infty$$

8. For a survey article on the evolution of wireless communications, see Oliphant (1999). For books on the fundamentals of wireless communication systems, see Steele and Hanzo (1999), Stüber (1996) and Rappaport (1996).
For a detailed description of GSM, see Chapter 8 of the book by Steele and Hanzo (1999). For a detailed description of the IS-95 system, see the handbook by Lee and Miller (1998).
9. The classic paper on the RAKE receiver is due to Price and Green (1958).
10. For the original paper on how to maximize the signal-to-noise ratio realizable from the sum of several noisy signals, see the classic paper by Brennan (1955).
11. The application of the RAKE receiver in CDMA wireless communication systems is discussed in detail in the book by Viterbi (1995).
12. The idea of multi-pulse excitation for speech coding is due to Atal and Remde (1982). Code-excited linear prediction (CELP) of speech was first introduced by Atal and Schroeder (1984). For a detailed mathematical discussion of multi-pulse excited, regular-pulse excited, and code-excited types of speech coding, particularly as they relate to wireless communications, see Chapter 3 in the book edited by Steele and Hanzo (1999).
13. In the wireless communications literature, adaptive antenna arrays are often referred to as *smart antennas*. For an overview of the various issues involved in the use of adaptive antenna arrays for wireless communications, see the article by Winters (1998) and the course notes by Winters (1999). The two-step subspace procedure for designing the array signal processor in Figure 8.29 is based on material presented in Winters (1999). The book by Rappaport (1999) presents a collection of papers on adaptive antenna arrays, which are grouped into algorithms, architectures, hardware applications, channel models, and performance evaluation.
14. The idea of subspace is rooted in matrix algebra. For a discussion of this idea, see Strang (1980) and Stewart (1973). For a discussion of subspace decomposition in the context of statistical signal processing, see Scharf (1991).
15. For tutorial discussions of space-time processing for wireless communications, see the articles by Paulraj and Ng (1998), Paulraj and Papadakis (1997), and Kohno (1998).
16. Recursive implementation of the DMI algorithm leads to a new algorithm commonly referred to as the *recursive least squares (RLS) algorithm*; for a derivation of the RLS algorithm and its variants, see Haykin (1996).

PROBLEMS

Free-Space Propagation

- 8.1 A radio link uses a pair of 2m dish antennas with an efficiency of 60 percent each, as transmitting and receiving antennas. Other specifications of the link are:

Transmitted power = 1 dBw

Carrier frequency = 4 GHz

Distance of the receiver

from the transmitter = 150 m

Calculate (a) the free-space loss, (b) the power gain of each antenna, and (c) the received power in dBW.

- 8.2 Repeat Problem 8.1 for a carrier frequency of 12 GHz.

- 8.3 Equation (8.14) is one formulation of the Friis free-space equation. Show that this equation can also be formulated in the following equivalent forms:

$$(a) P_r = \frac{P_t A_t A_r}{\lambda^2 d^2}$$

$$(b) P_r = \frac{P_t A_t G_r}{4\pi d^2}$$

where P_t is the transmitted power, A_t is the effective area of the transmitting antenna, λ is the carrier wavelength, d is the distance of the receiver from the transmitter, G_r is the power gain of the receiving antenna, A_r is the effective area of the receiving antenna, and P_r is the received power.

Discuss the situations that favor the use of one of these equations over the other.

- 8.4 From the mathematical definition of the free-space loss

$$L_{\text{free space}} = \left(\frac{4\pi d}{\lambda} \right)^2$$

we see that it is dependent on the carrier wavelength λ or frequency f . How can this dependence on wavelength or frequency be justified in physical terms?

- 8.5 In a satellite communication system, the carrier frequency used on the uplink is always higher than the carrier frequency used on the downlink. Justify the rationale for this choice.
- 8.6 A continuous-wave (CW) beacon transmitter is located on a satellite in geostationary orbit. The beacon's 12 GHz output is monitored by an earth station positioned 40,000 km from the satellite. The satellite transmitting antenna is a 1m dish with an aperture efficiency of 70 percent, and the earth station receiving antenna is a 10m dish with an aperture efficiency of 55 percent. Calculate the received power, given that the beacon's output power is 100 mW.

Noise Figure

- 8.7 Consider a 75- Ω resistor maintained at "room temperature" of 290K. Assuming a bandwidth of 1 MHz, calculate the following:

- (a) The root-mean-square (RMS) value of the voltage appearing across the terminals of this resistor due to thermal noise.
- (b) The maximum available noise power delivered to a matched load.

- 8.8 In this problem, we revisit Example 8.1 based on the receiver configuration of Figure 8.10. Suppose that a lossy waveguide is inserted between the receiving antenna and the low-noise amplifier. The waveguide loss is 1 dB, and its physical temperature is 290K. Recalculate the effective noise temperature of the receiver.
- 8.9 Consider the receiver of Figure P8.9, which consists of a lossy waveguide, low-noise RF amplifier, frequency down-converter (mixer), and IF amplifier. The figure includes the noise figures and power gains of these four components. The antenna temperature is 50K.
- Calculate the equivalent noise temperature for each of the four components in Figure P8.9, assuming a room temperature $T = 290\text{K}$.
 - Calculate the effective noise temperature of the whole receiver.

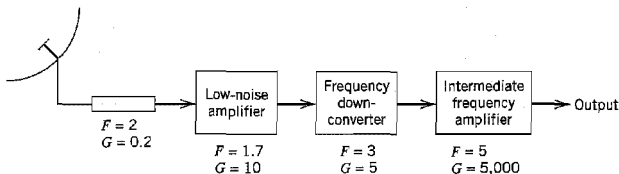


FIGURE P8.9

Budget Link Calculations

- 8.10 In this problem we address the uplink power budget of the digital satellite communication system considered in Example 8.2. The parameters of the link are as follows:

Carrier frequency	= 14 GHz
Power density at the TWT amplifier in saturation	= -81 dBW/m^2
Satellite figure of merit, G/T	= 1.9 dB/K
Distance of the satellite from the transmitting earth terminal	= 40,000 km

- Assuming no power backoff of the TWT, calculate the C/N_0 ratio at the satellite.
 - Given that the data rate in the uplink is the same as that calculated for the downlink in Example 8.2, calculate the probability of symbol error incurred in the uplink allowing for a link margin of 6 dB. Compare your result with that in Example 8.2.
- 8.11 The downlink C/N_0 ratio in a direct broadcast satellite (DBS) system is estimated to be 85 dB-Hz. The specifications of the link are:

Satellite EIRP	= 57 dBW
Downlink carrier frequency	= 12.5 GHz
Data rate	= 10 Mb/s
Required E_b/N_0 at the receiving earth terminal	= 10 dB
Distance of the satellite from the receiving earth terminal	= 41,000 km

Calculate the minimum diameter of the dish antenna needed to provide a satisfactory TV reception, assuming that the dish has an efficiency of 55 percent and it is located alongside the home where the temperature is 310K. For this calculation, assume that the operation of the DBS system is essentially downlink-limited.

Wireless Communications

- 8.12 Both wireless communications and satellite communications rely on radio propagation for their operations. Summarize (a) the similarities of these two multiuser communication systems, and (b) the major differences that distinguish them from each other.
- 8.13 In wireless communication systems, the carrier frequency on the uplink (reverse link) is smaller than the carrier frequency on the downlink (forward link). Justify the rationale for this choice.
- 8.14 Figure P8.14 depicts the direct (line-of-sight) and indirect (reflected) paths of a radio link operating over a plane earth. The heights of the transmitting antenna at the base station and the receiving antenna of a mobile unit are h_b and h_m , respectively. Assume the following:
- The reflection coefficient of the ground is -1 .
 - The distance d between the two antennas is large enough to make the phase difference ϕ between the reflected and direct paths small compared to 1 radian, so that we may set $\sin \phi \approx \phi$.

Hence, show that the received power P_r is given by the approximation

$$P_r \approx P_t G_b G_m \left(\frac{h_b h_m}{d^2} \right)^2$$

where P_t is the transmitted power, and G_b and G_m are the power gains of the transmitting base and mobile antennas, respectively. Compare this result with the Friis free-space equation.

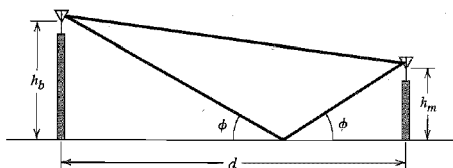


FIGURE P8.14

- 8.15 The *two-path model* defined by the impulse response

$$h(t) = a_1 \delta(t - \tau_1) + a_2 \exp(-j\theta) \delta(t - \tau_2)$$

is frequently used in the analytic treatment of wireless communication systems. The model parameters are the delay times τ_1 and τ_2 , the uniformly distributed phase θ , and the real coefficients a_1 and a_2 .

- (a) Determine (i) the transfer function of the model, and (ii) its power-delay profile.
 - (b) Show that the model exhibits frequency-selective fading due to variations in the coefficients a_1 and a_2 .
- 8.16 In the RAKE receiver illustrated in Figure 8.26, each correlator is synchronized by inserting the right delay into the received signal.
- (a) Show that, in theory, the same result is obtained by inserting the right delay into the reference signal (i.e., pseudo-noise sequence).
 - (b) In practice, the preferred method is to use the procedure described in Figure 8.26. What reason can you suggest for this preference?

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

$$x_j(t) = s_j(t) + n_j(t), \quad j = 1, 2, \dots, N$$

Assume the following:

- The signal components $s_j(t)$ are locally coherent, that is,

$$s_j(t) = z_j m(t), \quad j = 1, 2, \dots, N$$

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,

$$E[n_j(t)n_k(t)] = \begin{cases} \sigma_j^2 & \text{for } k = j \\ 0 & \text{otherwise} \end{cases}$$

The output of the linear combiner is defined by

$$x(t) = \sum_{j=1}^N \alpha_j x_j(t)$$

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

$$(i) \quad (\text{SNR})_O \leq \sum_{j=1}^N (\text{SNR})_j$$

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.

- 8.19 In this problem we derive the complex LMS algorithm. Referring to Figure 8.32 and starting with the instantaneous cost function

$$J = \frac{1}{2} |e[n]|^2$$

where $e[n]$ is the error signal and M is the number of antenna elements, do the following:

- Determine the derivative of the cost function J with respect to the k th elemental weight $w_k[n]$.
- Using the instantaneous derivative $\partial J / \partial w_k[n]$, denoted by $\hat{\nabla} J[k]$, determine the adjustment $\Delta w_k[n]$ made to the k th elemental weight in accordance with the rule

$$\Delta w_k[n] = -\mu \hat{\nabla} J[k]$$

- Verify the composition of the complex LMS algorithm described in Equations (8.75) to (8.77).

Note that $w_k[n]$ is complex valued, and you need to consider its real and imaginary parts separately.

- 8.20 A practical limitation of an adaptive antenna array using the LMS algorithm is the dynamic range over which the array can operate. This limitation is due to the fact that the speed of response of the weights in the LMS algorithm is proportional to the average signal power at the array input.

- Justify the assertion that the dynamic range of average signal power at the array input is proportional to R_b / f_{\max} , where R_b is the data rate in b/s and f_{\max} is the maximum fade rate in Hz.
- Assuming a proportionality factor of 0.2, by which the ratio R_b / f_{\max} is scaled, calculate the dynamic range of an adaptive antenna array using the LMS algorithm for $R_b = 32$ kb/s and $f_{\max} = 70$ Hz. Comment on your result. (The proportionality factor of 0.2 is a reasonable choice for systems using PSK.)

- 8.21 In this problem we derive the direct matrix inversion algorithm for adjusting the weights of an adaptive antenna array. To do so, we revisit the derivation of the Wiener filter presented in Chapter 3.

- Show that

$$\hat{\mathbf{R}}_{\mathbf{x}} \mathbf{w} = \hat{\mathbf{r}}_{\mathbf{x}d}$$

where $\hat{\mathbf{R}}_{\mathbf{x}}$ is an estimate of the correlation matrix of the input vector $\mathbf{x}[k]$:

$$\hat{\mathbf{R}}_{\mathbf{x}} = \frac{1}{K} \sum_{k=1}^K \mathbf{x}[k] \mathbf{x}^H[k]$$

and $\hat{\mathbf{r}}_{\mathbf{x}d}$ is an estimate of the cross-correlation vector between $\mathbf{x}[k]$ and the reference signal $d[k]$:

$$\hat{\mathbf{r}}_{\mathbf{x}d} = \frac{1}{K} \sum_{k=1}^K \mathbf{x}[k] d^*[k]$$

The superscript H in the formula for $\hat{\mathbf{R}}_{\mathbf{x}}$ denotes Hermitian transportation (i.e., transposition and complex conjugation), so $\mathbf{x}[k] \mathbf{x}^H[k]$ denotes the outer product of $\mathbf{x}[k]$ with itself. The summations for both $\hat{\mathbf{R}}_{\mathbf{x}}$ and $\hat{\mathbf{r}}_{\mathbf{x}d}$ are performed over a total of K snapshots, with each snapshot being represented by the pair $\{\mathbf{x}[k], d[k]\}$.

- Using the formulas of part (a), describe an algorithm for computing the weight vector \mathbf{w} , given a data set consisting of K snapshots. Hence demonstrate that the complexity of this algorithm grows as M^3 with the size of the weight vector \mathbf{w} denoted by M .

FUNDAMENTAL LIMITS IN INFORMATION THEORY

Shannon's landmark paper on information theory in 1948, and its refinements by other researchers, were in direct response to the need of electrical engineers to design communication systems that are both efficient and reliable. Efficient communication from a source to a user destination is attained through source coding. Reliable communication over a noisy channel is attained through error-control coding. This chapter addresses these important issues as summarized here:

- ▶ *Entropy as the basic measure of information.*
- ▶ *Source coding theorem and data compaction algorithms.*
- ▶ *Mutual information and its relation to the capacity of a communication channel for information transmission.*
- ▶ *Channel coding theorem as the basis for reliable communication.*
- ▶ *Information capacity theorem as the basis for a tradeoff between channel bandwidth and signal-to-noise ratio.*
- ▶ *Rate-distortion theory for source coding with a fidelity criterion.*

9.1 Introduction

As mentioned in the Background and Preview chapter and reiterated along the way, the purpose of a communication system is to carry information-bearing baseband signals from one place to another over a communication channel. In preceding chapters of the book, we have described a variety of modulation schemes for accomplishing this objective. But what do we mean by the term *information*? To address this issue, we need to invoke *information theory*.¹ This broadly based mathematical discipline has made fundamental contributions, not only to communications, but also to computer science, statistical physics, statistical inference, and probability and statistics.

In the context of communications, information theory deals with mathematical modeling and analysis of a communication system rather than with physical sources and physical channels. In particular, it provides answers to two fundamental questions (among others):

- ▶ What is the irreducible complexity below which a signal cannot be compressed?
- ▶ What is the ultimate transmission rate for reliable communication over a noisy channel?

The answers to these questions lie in the entropy of a source and the capacity of a channel, respectively. *Entropy* is defined in terms of the probabilistic behavior of a source of information; it is so named in deference to the parallel use of this concept in thermodynamics. *Capacity* is defined as the intrinsic ability of a channel to convey information; it is naturally related to the noise characteristics of the channel. A remarkable result that emerges from information theory is that if the entropy of the source is less than the capacity of the channel, then error-free communication over the channel can be achieved. It is therefore befitting that we begin our study of information theory by discussing the relationships among uncertainty, information, and entropy.

9.2 Uncertainty, Information, and Entropy

Suppose that a *probabilistic experiment* involves the observation of the output emitted by a discrete source during every unit of time (signaling interval). The source output is modeled as a discrete random variable, S , which takes on symbols from a fixed finite *alphabet*

$$\mathcal{S} = \{s_0, s_1, \dots, s_{K-1}\} \quad (9.1)$$

with probabilities

$$P(S = s_k) = p_k, \quad k = 0, 1, \dots, K - 1 \quad (9.2)$$

Of course, this set of probabilities must satisfy the condition

$$\sum_{k=0}^{K-1} p_k = 1 \quad (9.3)$$

We assume that the symbols emitted by the source during successive signaling intervals are statistically independent. A source having the properties just described is called a *discrete memoryless source*, memoryless in the sense that the symbol emitted at any time is independent of previous choices.

Can we find a measure of how much information is produced by such a source? To answer this question, we note that the idea of information is closely related to that of uncertainty or surprise, as described next.

Consider the event $S = s_k$, describing the emission of symbol s_k by the source with probability p_k , as defined in Equation (9.2). Clearly, if the probability $p_k = 1$ and $p_i = 0$ for all $i \neq k$, then there is no “surprise,” and therefore no “information,” when symbol s_k is emitted, because we know what the message from the source must be. If, on the other hand, the source symbols occur with different probabilities, and the probability p_k is low, then there is more surprise, and therefore information, when symbol s_k is emitted by the source than when symbol s_i , $i \neq k$, with higher probability is emitted. Thus, the words *uncertainty*, *surprise*, and *information* are all related. Before the event $S = s_k$ occurs, there is an amount of uncertainty. When the event $S = s_k$ occurs there is an amount of surprise. After the occurrence of the event $S = s_k$, there is gain in the amount of information, the essence of which may be viewed as the *resolution of uncertainty*. Moreover, the amount of information is related to the *inverse* of the probability of occurrence.

We define the amount of information gained after observing the event $S = s_k$, which occurs with probability p_k , as the *logarithmic function*²

$$I(s_k) = \log \left(\frac{1}{p_k} \right) \quad (9.4)$$

This definition exhibits the following important properties that are intuitively satisfying:

1.

$$I(s_k) = 0 \quad \text{for } p_k = 1 \quad (9.5)$$

Obviously, if we are absolutely *certain* of the outcome of an event, even before it occurs, there is *no* information gained.

2.

$$I(s_k) \geq 0 \quad \text{for } 0 \leq p_k \leq 1 \quad (9.6)$$

That is to say, the occurrence of an event $S = s_k$ either provides some or no information, but never brings about a *loss* of information.

3.

$$I(s_k) > I(s_i) \quad \text{for } p_k < p_i \quad (9.7)$$

That is, the less probable an event is, the more information we gain when it occurs.

4. $I(s_k s_l) = I(s_k) + I(s_l)$ if s_k and s_l are statistically independent.

The base of the logarithm in Equation (9.4) is quite arbitrary. Nevertheless, it is the standard practice today to use a logarithm to base 2. The resulting unit of information is called the *bit* (a contraction of *binary digit*). We thus write

$$\begin{aligned} I(s_k) &= \log_2 \left(\frac{1}{p_k} \right) \\ &= -\log_2 p_k \quad \text{for } k = 0, 1, \dots, K-1 \end{aligned} \quad (9.8)$$

When $p_k = 1/2$, we have $I(s_k) = 1$ bit. Hence, *one bit is the amount of information that we gain when one of two possible and equally likely (i.e., equiprobable) events occurs*. Note that the information $I(s_k)$ is positive, since the logarithm of a number less than one, such as a probability, is negative.

The amount of information $I(s_k)$ produced by the source during an arbitrary signaling interval depends on the symbol s_k emitted by the source at that time. Indeed, $I(s_k)$ is a discrete random variable that takes on the values $I(s_0), I(s_1), \dots, I(s_{K-1})$ with probabilities p_0, p_1, \dots, p_{K-1} respectively. The mean of $I(s_k)$ over the source alphabet \mathcal{S} is given by

$$\begin{aligned} H(\mathcal{S}) &= E[I(s_k)] \\ &= \sum_{k=0}^{K-1} p_k I(s_k) \\ &= \sum_{k=0}^{K-1} p_k \log_2 \left(\frac{1}{p_k} \right) \end{aligned} \quad (9.9)$$

The important quantity $H(\mathcal{S})$ is called the *entropy*³ of a discrete memoryless source with source alphabet \mathcal{S} . It is a measure of the *average information content per source symbol*. Note that the entropy $H(\mathcal{S})$ depends only on the probabilities of the symbols in the alphabet \mathcal{S} of the source. Thus the symbol \mathcal{S} in $H(\mathcal{S})$ is not an argument of a function but rather a label for a source.

■ SOME PROPERTIES OF ENTROPY

Consider a discrete memoryless source whose mathematical model is defined by Equations (9.1) and (9.2). The entropy $H(\mathcal{S})$ of such a source is bounded as follows:

$$0 \leq H(\mathcal{S}) \leq \log_2 K \quad (9.10)$$

where K is the *radix* (number of symbols) of the alphabet \mathcal{S} of the source. Furthermore, we may make two statements:

1. $H(\mathcal{S}) = 0$, if and only if the probability $p_k = 1$ for some k , and the remaining probabilities in the set are all zero; this lower bound on entropy corresponds to *no uncertainty*.
2. $H(\mathcal{S}) = \log_2 K$, if and only if $p_k = 1/K$ for all k (i.e., all the symbols in the alphabet \mathcal{S} are *equiprobable*); this upper bound on entropy corresponds to *maximum uncertainty*.

To prove these properties of $H(\mathcal{S})$, we proceed as follows. First, since each probability p_k is less than or equal to unity, it follows that each term $p_k \log_2(1/p_k)$ in Equation (9.9) is always nonnegative, and so $H(\mathcal{S}) \geq 0$. Next, we note that the product term $p_k \log_2(1/p_k)$ is zero if, and only if, $p_k = 0$ or 1. We therefore deduce that $H(\mathcal{S}) = 0$ if, and only if, $p_k = 0$ or 1, that is, $p_k = 1$ for some k and all the rest are zero.

This completes the proofs of the lower bound in Equation (9.10) and statement (1).

To prove the upper bound in Equation (9.10) and statement (2), we make use of a property of the natural logarithm:

$$\log x \leq x - 1, \quad x \geq 0 \quad (9.11)$$

This inequality can be readily verified by plotting the functions $\log x$ and $(x - 1)$ versus x , as shown in Figure 9.1. Here we see that the line $y = x - 1$ always lies above the curve $y = \log x$. The equality holds *only* at the point $x = 1$, where the line is tangential to the curve.

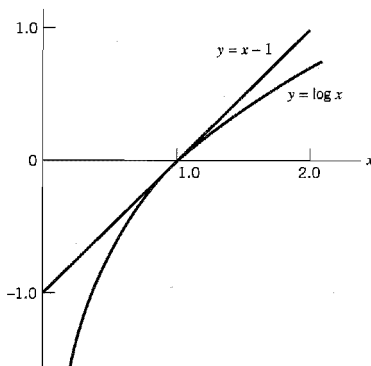


FIGURE 9.1 Graphs of the functions $x - 1$ and $\log x$ versus x .

To proceed with the proof, consider first any two probability distributions $\{p_0, p_1, \dots, p_{K-1}\}$ and $\{q_0, q_1, \dots, q_{K-1}\}$ on the alphabet $\mathcal{S} = \{s_0, s_1, \dots, s_{K-1}\}$ of a discrete memoryless source. Then, changing to the natural logarithm, we may write

$$\sum_{k=0}^{K-1} p_k \log_2 \left(\frac{q_k}{p_k} \right) = \frac{1}{\log 2} \sum_{k=0}^{K-1} p_k \log \left(\frac{q_k}{p_k} \right)$$

Hence, using the inequality of Equation (9.11), we get

$$\begin{aligned} \sum_{k=0}^{K-1} p_k \log_2 \left(\frac{q_k}{p_k} \right) &\leq \frac{1}{\log 2} \sum_{k=0}^{K-1} p_k \left(\frac{q_k}{p_k} - 1 \right) \\ &\leq \frac{1}{\log 2} \sum_{k=0}^{K-1} (q_k - p_k) \\ &\leq \frac{1}{\log 2} \left(\sum_{k=0}^{K-1} q_k - \sum_{k=0}^{K-1} p_k \right) = 0 \end{aligned}$$

We thus have the *fundamental inequality*

$$\sum_{k=0}^{K-1} p_k \log_2 \left(\frac{q_k}{p_k} \right) \leq 0 \quad (9.12)$$

where the equality holds only if $q_k = p_k$ for all k .

Suppose we next put

$$q_k = \frac{1}{K}, \quad k = 0, 1, \dots, K-1 \quad (9.13)$$

which corresponds to an alphabet \mathcal{S} with *equiprobable* symbols. The entropy of a discrete memoryless source with such a characterization equals

$$\sum_{k=0}^{K-1} q_k \log_2 \left(\frac{1}{q_k} \right) = \log_2 K \quad (9.14)$$

Also, the use of Equation (9.13) in Equation (9.12) yields

$$\sum_{k=0}^{K-1} p_k \log_2 \left(\frac{1}{p_k} \right) \leq \log_2 K$$

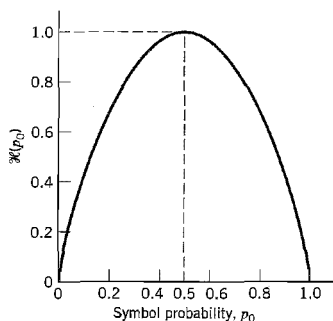
Equivalently, the entropy of a discrete memoryless source with an arbitrary probability distribution for the symbols of its alphabet \mathcal{S} is bounded as

$$H(\mathcal{S}) \leq \log_2 K$$

Thus $H(\mathcal{S})$ is always less than or equal to $\log_2 K$. The equality holds only if the symbols in the alphabet \mathcal{S} are equiprobable, as in Equation (9.13). This completes the proof of Equation (9.10) and statements (1) and (2).

► EXAMPLE 9.1 Entropy of Binary Memoryless Source

To illustrate the properties of $H(\mathcal{S})$, we consider a binary source for which symbol 0 occurs with probability p_0 and symbol 1 with probability $p_1 = 1 - p_0$. We assume that the source is memoryless so that successive symbols emitted by the source are statistically independent.

FIGURE 9.2 Entropy function $\mathcal{H}(p_0)$.

The entropy of such a source equals

$$\begin{aligned} H(\mathcal{S}) &= -p_0 \log_2 p_0 - p_1 \log_2 p_1 \\ &= -p_0 \log_2 p_0 - (1 - p_0) \log_2 (1 - p_0) \text{ bits} \end{aligned} \quad (9.15)$$

from which we observe the following:

1. When $p_0 = 0$, the entropy $H(\mathcal{S}) = 0$; this follows from the fact that $x \log x \rightarrow 0$ as $x \rightarrow 0$.
2. When $p_0 = 1$, the entropy $H(\mathcal{S}) = 0$.
3. The entropy $H(\mathcal{S})$ attains its maximum value, $H_{\max} = 1$ bit, when $p_1 = p_0 = 1/2$, that is, symbols 1 and 0 are equally probable.

The function of p_0 given on the right-hand side of Equation (9.15) is frequently encountered in information-theoretic problems. It is therefore customary to assign a special symbol to this function. Specifically, we define

$$\mathcal{H}(p_0) = -p_0 \log_2 p_0 - (1 - p_0) \log_2 (1 - p_0) \quad (9.16)$$

We refer to $\mathcal{H}(p_0)$ as the *entropy function*. The distinction between Equation (9.15) and Equation (9.16) should be carefully noted. The $H(\mathcal{S})$ of Equation (9.15) gives the entropy of a discrete memoryless source with source alphabet \mathcal{S} . The $\mathcal{H}(p_0)$ of Equation (9.16), on the other hand, is a function of the prior probability p_0 defined on the interval $[0, 1]$. Accordingly, we may plot the entropy function $\mathcal{H}(p_0)$ versus p_0 , defined on the interval $[0, 1]$, as in Figure 9.2. The curve in Figure 9.2 highlights the observations made under points 1, 2, and 3.

■ EXTENSION OF A DISCRETE MEMORYLESS SOURCE

In discussing information-theoretic concepts, we often find it useful to consider *blocks* rather than individual symbols, with each block consisting of n successive source symbols. We may view each such block as being produced by an *extended source* with a source alphabet \mathcal{S}^n that has K^n distinct blocks, where K is the number of distinct symbols in the source alphabet \mathcal{S} of the original source. In the case of a discrete memoryless source, the source symbols are statistically independent. Hence, the probability of a source symbol in \mathcal{S}^n is equal to the product of the probabilities of the n source symbols in \mathcal{S} constituting the particular source symbol in \mathcal{S}^n . We may thus intuitively expect that $H(\mathcal{S}^n)$, the entropy

of the extended source, is equal to n times $H(\mathcal{S})$, the entropy of the original source. That is, we may write

$$H(\mathcal{S}^n) = nH(\mathcal{S}) \quad (9.17)$$

► 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 $\sigma_0, \sigma_1, \dots, \sigma_8$. 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 $p(\sigma_i)$, $i = 0, 1, \dots, 8$	$\frac{1}{16}$	$\frac{1}{16}$	$\frac{1}{8}$	$\frac{1}{16}$	$\frac{1}{16}$	$\frac{1}{8}$	$\frac{1}{8}$	$\frac{1}{8}$	$\frac{1}{4}$

9.3 Source-Coding Theorem

An important problem in communications is the *efficient* representation of data generated by a discrete source. The process by which this representation is accomplished is called *source encoding*. The device that performs the representation is called a *source encoder*. For the source encoder to be *efficient*, we require knowledge of the statistics of the source. In particular, if some source symbols are known to be more probable than others, then we may exploit this feature in the generation of a *source code* by assigning *short* code words to *frequent* source symbols, and *long* code words to *rare* source symbols. We refer to such a source code as a *variable-length code*. The *Morse code* is an example of a variable-length code. In the Morse code, the letters of the alphabet and numerals are encoded into streams of *marks* and *spaces*, denoted as dots “.” and dashes “-”, respectively. In the English language, the letter *E* occurs more frequently than the letter *Q*, for example, so the Morse code encodes *E* into a single dot “.”, the shortest code word in the code, and it encodes *Q* into “- . - .”, the longest code word in the code.

Our primary interest is in the development of an efficient source encoder that satisfies two functional requirements:

1. The code words produced by the encoder are in *binary* form.
2. The source code is *uniquely decodable*, so that the original source sequence can be reconstructed perfectly from the encoded binary sequence.

Consider then the scheme shown in Figure 9.3, which depicts a discrete memoryless source whose output s_k is converted by the source encoder into a block of 0s and 1s, denoted by b_k . We assume that the source has an alphabet with K different symbols, and that the k th symbol s_k occurs with probability p_k , $k = 0, 1, \dots, K - 1$. Let the binary code word assigned to symbol s_k by the encoder have length l_k , measured in bits. We define the average code-word length, \bar{L} , of the source encoder as

$$\bar{L} = \sum_{k=0}^{K-1} p_k l_k \quad (9.18)$$

In physical terms, the parameter \bar{L} represents the *average number of bits per source symbol* used in the source encoding process. Let L_{\min} denote the *minimum* possible value of \bar{L} . We then define the *coding efficiency* of the source encoder as

$$\eta = \frac{L_{\min}}{\bar{L}} \quad (9.19)$$

With $\bar{L} \geq L_{\min}$, we clearly have $\eta \leq 1$. The source encoder is said to be *efficient* when η approaches unity.

But how is the minimum value L_{\min} determined? The answer to this fundamental question is embodied in Shannon's first theorem: the *source-coding theorem*,⁴ which may be stated as follows:

Given a discrete memoryless source of entropy $H(\mathcal{F})$, the average code-word length \bar{L} for any distortionless source encoding scheme is bounded as

$$\bar{L} \geq H(\mathcal{F}) \quad (9.20)$$

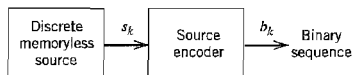


FIGURE 9.3 Source encoding.

(A proof of this theorem for a particular class of source codes is presented in the next section.) According to the source-coding theorem, the entropy $H(\mathcal{S})$ represents a *fundamental limit* on the average number of bits per source symbol necessary to represent a discrete memoryless source in that it can be made as small as, but no smaller than, the entropy $H(\mathcal{S})$. Thus with $L_{\min} = H(\mathcal{S})$, we may rewrite the efficiency of a source encoder in terms of the entropy $H(\mathcal{S})$ as

$$\eta = \frac{H(\mathcal{S})}{\bar{L}} \quad (9.21)$$

9.4 Data Compaction

A common characteristic of signals generated by physical sources is that, in their natural form, they contain a significant amount of information that is *redundant*, the transmission of which is therefore wasteful of primary communication resources. For *efficient* signal transmission, the *redundant information should be removed from the signal prior to transmission*. This operation, with *no* loss of information, is ordinarily performed on a signal in digital form, in which case we refer to it as *data compaction* or *lossless data compression*. The code resulting from such an operation provides a representation of the source output that is not only efficient in terms of the average number of bits per symbol but also exact in the sense that the original data can be reconstructed with no loss of information. The entropy of the source establishes the fundamental limit on the removal of redundancy from the data. Basically, data compaction is achieved by assigning short descriptions to the most frequent outcomes of the source output and longer descriptions to the less frequent ones.

In this section, we discuss some source-coding schemes for data compaction. We begin the discussion by describing a type of source code known as a *prefix code*, which is not only decodable but also offers the possibility of realizing an average code-word length that can be made arbitrarily close to the source entropy.

PREFIX CODING

Consider a discrete memoryless source of alphabet $\{s_0, s_1, \dots, s_{K-1}\}$ and statistics $\{p_0, p_1, \dots, p_{K-1}\}$. For a source code representing the output of this source to be of practical use, the code has to be uniquely decodable. This restriction ensures that for each finite sequence of symbols emitted by the source, the corresponding sequence of code words is different from the sequence of code words corresponding to any other source sequence. We are specifically interested in a special class of codes satisfying a restriction known as the *prefix condition*. To define the prefix condition, let the code word assigned to source symbol s_k be denoted by $(m_{k_1}, m_{k_2}, \dots, m_{k_n})$, where the individual elements m_{k_1}, \dots, m_{k_n} are 0s and 1s, and n is the code-word length. The initial part of the code word is represented by the elements m_{k_1}, \dots, m_{k_i} for some $i \leq n$. Any sequence made up of the initial part of the code word is called a *prefix* of the code word. A *prefix code* is defined as a code in which no code word is the prefix of any other code word.

To illustrate the meaning of a prefix code, consider the three source codes described in Table 9.2. Code I is not a prefix code since the bit 0, the code word for s_0 , is a prefix of 00, the code word for s_2 . Likewise, the bit 1, the code word for s_1 , is a prefix of 11, the code word for s_3 . Similarly, we may show that code III is not a prefix code, but code II is.

To decode a sequence of code words generated from a prefix source code, the *source decoder* simply starts at the beginning of the sequence and decodes one code word at a time. Specifically, it sets up what is equivalent to a *decision tree*, which is a graphical

TABLE 9.2 Illustrating the definition of a prefix code

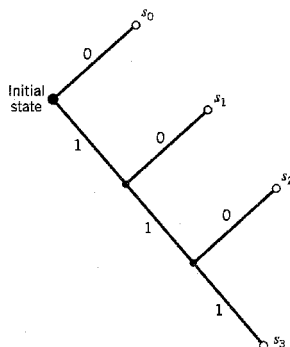
Source Symbol	Probability of Occurrence	Code I	Code II	Code III
s_0	0.5	0	0	0
s_1	0.25	1	10	01
s_2	0.125	00	110	011
s_3	0.125	11	111	0111

portrayal of the code words in the particular source code. For example, Figure 9.4 depicts the decision tree corresponding to code II in Table 9.2. The tree has an *initial state* and four *terminal states* corresponding to source symbols s_0 , s_1 , s_2 , and s_3 . The decoder always starts at the initial state. The first received bit moves the decoder to the terminal state s_0 if it is 0, or else to a second decision point if it is 1. In the latter case, the second bit moves the decoder one step further down the tree, either to terminal state s_1 if it is 0, or else to a third decision point if it is 1, and so on. Once each terminal state emits its symbol, the decoder is reset to its initial state. Note also that each bit in the received encoded sequence is examined only once. For example, the encoded sequence 1011111000 ... is readily decoded as the source sequence $s_1 s_3 s_2 s_0 s_0$... The reader is invited to carry out this decoding.

A prefix code has the important property that it is *always* uniquely decodable. But the converse is not necessarily true. For example, code III in Table 9.2 does not satisfy the prefix condition, yet it is uniquely decodable since the bit 0 indicates the beginning of each code word in the code.

Moreover, if a prefix code has been constructed for a discrete memoryless source with source alphabet $\{s_0, s_1, \dots, s_{K-1}\}$ and source statistics $\{p_0, p_1, \dots, p_{K-1}\}$ and the code word for symbol s_k has length l_k , $k = 0, 1, \dots, K - 1$, then the code-word lengths of the code always satisfy a certain inequality known as the *Kraft-McMillan Inequality*,⁵ as shown by

$$\sum_{k=0}^{K-1} 2^{-l_k} \leq 1 \quad (9.22)$$

**FIGURE 9.4** Decision tree for code II of Table 9.2.

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 II 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 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 \bar{L} , which is bounded as follows:

$$H(\mathcal{S}) \leq \bar{L} < H(\mathcal{S}) + 1 \quad (9.23)$$

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

$$p_k = 2^{-l_k} \quad (9.24)$$

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

$$\bar{L} = \sum_{k=0}^{K-1} \frac{l_k}{2^{l_k}} \quad (9.25)$$

and the corresponding entropy of the source is

$$\begin{aligned} H(\mathcal{S}) &= \sum_{k=0}^{K-1} \left(\frac{1}{2^{l_k}} \right) \log_2(2^{l_k}) \\ &= \sum_{k=0}^{K-1} \frac{l_k}{2^{l_k}} \end{aligned} \quad (9.26)$$

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 $\bar{L} = H(\mathcal{S})$.

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 \bar{L}_n denote the average code-word length of the extended prefix code. For a uniquely decodable code, \bar{L}_n is the smallest possible. From Equation (9.23), we deduce that

$$H(\mathcal{S}^n) \leq \bar{L}_n < H(\mathcal{S}^n) + 1 \quad (9.27)$$

Substituting Equation (9.17) for an extended source into Equation (9.27), we get

$$nH(\mathcal{S}) \leq \bar{L}_n < nH(\mathcal{S}) + 1$$

or, equivalently,

$$H(\mathcal{S}) \leq \frac{\bar{L}_n}{n} < H(\mathcal{S}) + \frac{1}{n} \quad (9.28)$$

In the limit, as n approaches infinity, the lower and upper bounds in Equation (9.28) converge, as shown by

$$\lim_{n \rightarrow \infty} \frac{1}{n} \bar{L}_n = H(\mathcal{S}) \quad (9.29)$$

We may therefore state that by making the order n of an extended prefix source encoder large enough, we can make the code faithfully represent the discrete memoryless source \mathcal{S} as closely as desired. In other words, the average code-word length of an extended prefix code can be made as small as the entropy of the source provided the extended code has a high enough order, in accordance with the source-coding theorem. However, the price we have to pay for decreasing the average code-word length is increased decoding complexity, which is brought about by the high order of the extended prefix code.

■ HUFFMAN CODING

We next describe an important class of prefix codes known as Huffman codes. The basic idea behind *Huffman coding*⁶ is to assign to each symbol of an alphabet a sequence of bits roughly equal in length to the amount of information conveyed by the symbol in question. The end result is a source code whose average code-word length approaches the fundamental limit set by the entropy of a discrete memoryless source, namely, $H(\mathcal{S})$. The essence of the *algorithm* used to synthesize the Huffman code is to replace the prescribed set of source statistics of a discrete memoryless source with a simpler one. This *reduction* process is continued in a step-by-step manner until we are left with a final set of only two source statistics (symbols), for which (0, 1) is an optimal code. Starting from this trivial code, we then work backward and thereby construct the Huffman code for the given source.

Specifically, the Huffman *encoding algorithm* proceeds as follows:

1. The source symbols are listed in order of decreasing probability. The two source symbols of lowest probability are assigned a 0 and a 1. This part of the step is referred to as a *splitting* stage.
2. These two source symbols are regarded as being *combined* into a new source symbol with probability equal to the sum of the two original probabilities. (The list of source symbols, and therefore source statistics, is thereby *reduced* in size by one.) The probability of the new symbol is placed in the list in accordance with its value.
3. The procedure is repeated until we are left with a final list of source statistics (symbols) of only two for which a 0 and a 1 are assigned.

The code for each (original) source symbol is found by working backward and tracing the sequence of 0s and 1s assigned to that symbol as well as its successors.

► EXAMPLE 9.3 Huffman Tree

The five symbols of the alphabet of a discrete memoryless source and their probabilities are shown in the two leftmost columns of Figure 9.5a. Following through the Huffman algorithm, we reach the end of the computation in four steps, resulting in the *Huffman tree* shown in Figure 9.5a. The code words of the Huffman code for the source are tabulated in Figure 9.5b.

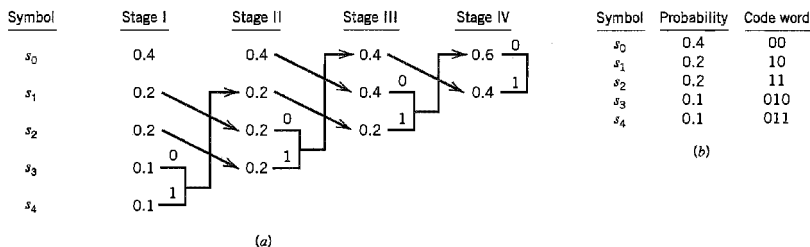


FIGURE 9.5 (a) Example of the Huffman encoding algorithm. (b) Source code.

The average code-word length is therefore

$$\begin{aligned}\bar{L} &= 0.4(2) + 0.2(2) + 0.2(2) + 0.1(3) + 0.1(3) \\ &= 2.2\end{aligned}$$

The entropy of the specified discrete memoryless source is calculated as follows [see Equation (9.9)]:

$$\begin{aligned}H(\mathcal{S}) &= 0.4 \log_2 \left(\frac{1}{0.4} \right) + 0.2 \log_2 \left(\frac{1}{0.2} \right) + 0.2 \log_2 \left(\frac{1}{0.2} \right) \\ &\quad + 0.1 \log_2 \left(\frac{1}{0.1} \right) + 0.1 \log_2 \left(\frac{1}{0.1} \right) \\ &= 0.52877 + 0.46439 + 0.46439 + 0.33219 + 0.33219 \\ &= 2.12193 \text{ bits}\end{aligned}$$

For the example at hand, we may make two observations:

1. The average code-word length \bar{L} exceeds the entropy $H(\mathcal{S})$ by only 3.67 percent.
2. The average code-word length \bar{L} does indeed satisfy Equation (9.23). \triangleleft

It is noteworthy that the Huffman encoding process (i.e., the Huffman tree) is not unique. In particular, we may cite two variations in the process that are responsible for the nonuniqueness of the Huffman code. First, at each splitting stage in the construction of a Huffman code, there is arbitrariness in the way a 0 and a 1 are assigned to the last two source symbols. Whichever way the assignments are made, however, the resulting differences are trivial. Second, ambiguity arises when the probability of a *combined* symbol (obtained by adding the last two probabilities pertinent to a particular step) is found to equal another probability in the list. We may proceed by placing the probability of the new symbol as *high* as possible, as in Example 9.3. Alternatively, we may place it as *low* as possible. (It is presumed that whichever way the placement is made, high or low, it is consistently adhered to throughout the encoding process.) But this time, noticeable differences arise in that the code words in the resulting source code can have different lengths. Nevertheless, the average code-word length remains the same.

As a measure of the variability in code-word lengths of a source code, we define the *variance* of the average code-word length \bar{L} over the ensemble of source symbols as

$$\sigma^2 = \sum_{k=0}^{K-1} p_k (l_k - \bar{L})^2 \quad (9.30)$$

where p_0, p_1, \dots, p_{K-1} are the source statistics, and l_k is the length of the code word assigned to source symbol s_k . It is usually found that when a combined symbol is moved

as high as possible, the resulting Huffman code has a significantly smaller variance σ^2 than when it is moved as low as possible. On this basis, it is reasonable to choose the former Huffman code over the latter.

LEMPEL–ZIV CODING

A drawback of the Huffman code is that it requires knowledge of a probabilistic model of the source; unfortunately, in practice, source statistics are not always known *a priori*. Moreover, in modeling text we find that storage requirements prevent the Huffman code from capturing the higher-order relationships between words and phrases, thereby compromising the efficiency of the code. To overcome these practical limitations, we may use the *Lempel–Ziv algorithm*,⁷ which is intrinsically *adaptive* and simpler to implement than Huffman coding.

Basically, encoding in the Lempel–Ziv algorithm is accomplished by *parsing the source data stream into segments that are the shortest subsequences not encountered previously*. To illustrate this simple yet elegant idea, consider the example of an input binary sequence specified as follows:

000101110010100101 ...

It is assumed that the binary symbols 0 and 1 are already stored in that order in the code book. We thus write

Subsequences stored: 0, 1
Data to be parsed: 000101110010100101 ...

The encoding process begins at the left. With symbols 0 and 1 already stored, the *shortest subsequence* of the data stream encountered for the first time and not seen before is 00; so we write

Subsequences stored: 0, 1, 00
Data to be parsed: 0101110010100101 ...

The second shortest subsequence not seen before is 01; accordingly, we go on to write

Subsequences stored: 0, 1, 00, 01
Data to be parsed: 01110010100101 ...

The next shortest subsequence not encountered previously is 011; hence, we write

Subsequences stored: 0, 1, 00, 01, 011
Data to be parsed: 10010100101 ...

We continue in the manner described here until the given data stream has been completely parsed. Thus, for the example at hand, we get the *code book* of binary subsequences shown in the second row of Figure 9.6.

Numerical positions:	1	2	3	4	5	6	7	8	9
Subsequences:	0	1	00	01	011	10	010	100	101
Numerical representations:			11	12	42	21	41	61	62
Binary encoded blocks:			0010	0011	1001	0100	1000	1100	1101

FIGURE 9.6 Illustrating the encoding process performed by the Lempel–Ziv algorithm on the binary sequence 000101110010100101....

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 uniform 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

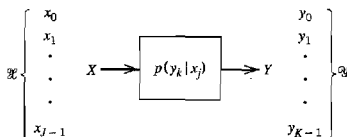


FIGURE 9.7 Discrete memoryless channel.

channel accepts an input symbol X selected from an alphabet \mathcal{X} and, in response, it emits an output symbol Y from an alphabet \mathcal{Y} . The channel is said to be “discrete” when both of the alphabets \mathcal{X} and \mathcal{Y} have *finite* sizes. It is said to be “memoryless” when the current output symbol depends *only* on the current input symbol and *not* any of the previous ones.

Figure 9.7 depicts a view of a discrete memoryless channel. The channel is described in terms of an *input alphabet*

$$\mathcal{X} = \{x_0, x_1, \dots, x_{J-1}\}, \quad (9.31)$$

an *output alphabet*,

$$\mathcal{Y} = \{y_0, y_1, \dots, y_{K-1}\}, \quad (9.32)$$

and a set of *transition probabilities*

$$p(y_k | x_j) = P(Y = y_k | X = x_j) \quad \text{for all } j \text{ and } k \quad (9.33)$$

Naturally, we have

$$0 \leq p(y_k | x_j) \leq 1 \quad \text{for all } j \text{ and } k \quad (9.34)$$

Also, the input alphabet \mathcal{X} and output alphabet \mathcal{Y} need not have the same size. For example, in channel coding, the size K of the output alphabet \mathcal{Y} may be larger than the size J of the input alphabet \mathcal{X} ; thus, $K \geq J$. On the other hand, we may have a situation in which the channel emits the same symbol when either one of two input symbols is sent, in which case we have $K \leq J$.

A convenient way of describing a discrete memoryless channel is to arrange the various transition probabilities of the channel in the form of a matrix as follows:

$$\mathbf{P} = \begin{bmatrix} p(y_0 | x_0) & p(y_1 | x_0) & \cdots & p(y_{K-1} | x_0) \\ p(y_0 | x_1) & p(y_1 | x_1) & \cdots & p(y_{K-1} | x_1) \\ \vdots & \vdots & \ddots & \vdots \\ p(y_0 | x_{J-1}) & p(y_1 | x_{J-1}) & \cdots & p(y_{K-1} | x_{J-1}) \end{bmatrix} \quad (9.35)$$

The J -by- K matrix \mathbf{P} is called the *channel matrix*, or *transition matrix*. Note that each *row* of the channel matrix \mathbf{P} corresponds to a *fixed channel input*, whereas each *column* of the matrix corresponds to a *fixed channel output*. Note also that a fundamental property of the channel matrix \mathbf{P} , as defined here, is that the sum of the elements along any row of the matrix is always equal to one; that is,

$$\sum_{k=0}^{K-1} p(y_k | x_j) = 1 \quad \text{for all } j \quad (9.36)$$

Suppose now that the inputs to a discrete memoryless channel are selected according to the *probability distribution* $\{p(x_j), j = 0, 1, \dots, J-1\}$. In other words, the event that the channel input $X = x_j$ occurs with probability

$$p(x_j) = P(X = x_j) \quad \text{for } j = 0, 1, \dots, J-1 \quad (9.37)$$

Having specified the random variable X denoting the channel input, we may now specify the second random variable Y denoting the channel output. The *joint probability distribution* of the random variables X and Y is given by

$$\begin{aligned} p(x_j, y_k) &= P(X = x_j, Y = y_k) \\ &= P(Y = y_k | X = x_j)P(X = x_j) \\ &= p(y_k | x_j)p(x_j) \end{aligned} \quad (9.38)$$

The *marginal probability distribution* of the output random variable Y is obtained by averaging out the dependence of $p(x_j, y_k)$ on x_j , as shown by

$$\begin{aligned} p(y_k) &= P(Y = y_k) \\ &= \sum_{j=0}^{J-1} P(Y = y_k | X = x_j)P(X = x_j) \\ &= \sum_{j=0}^{J-1} p(y_k | x_j)p(x_j) \quad \text{for } k = 0, 1, \dots, K-1 \end{aligned} \quad (9.39)$$

The probabilities $p(x_j)$ for $j = 0, 1, \dots, J-1$, are known as the *a priori probabilities* of the various input symbols. Equation (9.39) states that if we are given the input *a priori* probabilities $p(x_j)$ and the channel matrix [i.e., the matrix of transition probabilities $p(y_k | x_j)$], then we may calculate the probabilities of the various output symbols, the $p(y_k)$.

► EXAMPLE 9.4 Binary Symmetric Channel

The *binary symmetric channel* is of great theoretical interest and practical importance. It is a special case of the discrete memoryless channel with $J = K = 2$. The channel has two input symbols ($x_0 = 0, x_1 = 1$) and two output symbols ($y_0 = 0, y_1 = 1$). The channel is symmetric because the probability of receiving a 1 if a 0 is sent is the same as the probability of receiving a 0 if a 1 is sent. This conditional probability of error is denoted by p . The *transition probability diagram* of a binary symmetric channel is as shown in Figure 9.8. ◀

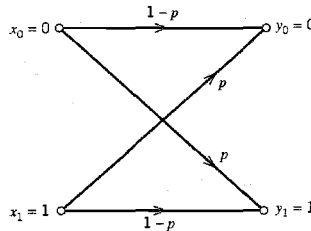


FIGURE 9.8 Transition probability diagram of binary symmetric channel.

It is of interest to relate the transition probability diagram of Figure 9.8 to the conditional probabilities of error p_{10} and p_{01} that were determined for the PCM receiver in Section 3.3. For the case when the binary symbols 0 and 1 are equiprobable, we showed that the optimized values of these two error probabilities are equal. Indeed, recalling the following definitions (using the terminology of Figure 9.8):

$$p_{10} = P(y = 1 | x = 0)$$

and

$$p_{01} = P(y = 0 | x = 1)$$

we immediately see that for the PCM receiver of Figure 3.4:

$$p_{10} = p_{01} = p$$

9.6 Mutual Information

Given that we think of the channel output Y (selected from alphabet \mathcal{Y}) as a noisy version of the channel input X (selected from alphabet \mathcal{X}), and that the entropy $H(\mathcal{X})$ is a measure of the prior uncertainty about X , how can we measure the uncertainty about X after observing Y ? To answer this question, we extend the ideas developed in Section 9.2 by defining the *conditional entropy* of X selected from alphabet \mathcal{X} , given that $Y = y_k$. Specifically, we write

$$H(\mathcal{X} | Y = y_k) = \sum_{j=0}^{J-1} p(x_j | y_k) \log_2 \left[\frac{1}{p(x_j | y_k)} \right] \quad (9.40)$$

This quantity is itself a random variable that takes on the values $H(\mathcal{X} | Y = y_0), \dots, H(\mathcal{X} | Y = y_{K-1})$ with probabilities $p(y_0), \dots, p(y_{K-1})$, respectively. The mean of entropy $H(\mathcal{X} | Y = y_k)$ over the output alphabet \mathcal{Y} is therefore given by

$$\begin{aligned} H(\mathcal{X} | \mathcal{Y}) &= \sum_{k=0}^{K-1} H(\mathcal{X} | Y = y_k) p(y_k) \\ &= \sum_{k=0}^{K-1} \sum_{j=0}^{J-1} p(x_j | y_k) p(y_k) \log_2 \left[\frac{1}{p(x_j | y_k)} \right] \\ &= \sum_{k=0}^{K-1} \sum_{j=0}^{J-1} p(x_j, y_k) \log_2 \left[\frac{1}{p(x_j | y_k)} \right] \end{aligned} \quad (9.41)$$

where, in the last line, we have made use of the relation

$$p(x_j, y_k) = p(x_j | y_k) p(y_k) \quad (9.42)$$

The quantity $H(\mathcal{X} | \mathcal{Y})$ is called a *conditional entropy*. It represents the amount of uncertainty remaining about the channel input after the channel output has been observed.

Since the entropy $H(\mathcal{X})$ represents our uncertainty about the channel input *before* observing the channel output, and the conditional entropy $H(\mathcal{X} | \mathcal{Y})$ represents our uncertainty about the channel input *after* observing the channel output, it follows that the difference $H(\mathcal{X}) - H(\mathcal{X} | \mathcal{Y})$ must represent our uncertainty about the channel input that is resolved by observing the channel output. This important quantity is called the *mutual*

information of the channel. Denoting the mutual information by $I(\mathcal{X}; \mathcal{Y})$, we may thus write

$$I(\mathcal{X}; \mathcal{Y}) = H(\mathcal{X}) - H(\mathcal{X}|\mathcal{Y}) \quad (9.43)$$

Similarly, we may write

$$I(\mathcal{Y}; \mathcal{X}) = H(\mathcal{Y}) - H(\mathcal{Y}|\mathcal{X}) \quad (9.44)$$

where $H(\mathcal{Y})$ is the entropy of the channel output and $H(\mathcal{Y}|\mathcal{X})$ is the conditional entropy of the channel output given the channel input.

■ PROPERTIES OF MUTUAL INFORMATION

The mutual information $I(\mathcal{X}; \mathcal{Y})$ has the following important properties.

Property 1

The mutual information of a channel is symmetric; that is

$$I(\mathcal{X}; \mathcal{Y}) = I(\mathcal{Y}; \mathcal{X}) \quad (9.45)$$

where the mutual information $I(\mathcal{X}; \mathcal{Y})$ is a measure of the uncertainty about the channel input that is resolved by *observing* the channel output, and the mutual information $I(\mathcal{Y}; \mathcal{X})$ is a measure of the uncertainty about the channel output that is resolved by *sending* the channel input.

To prove this property, we first use the formula for entropy and then use Equations (9.36) and (9.38), in that order, to express $H(\mathcal{X})$ as

$$\begin{aligned} H(\mathcal{X}) &= \sum_{j=0}^{J-1} p(x_j) \log_2 \left[\frac{1}{p(x_j)} \right] \\ &= \sum_{j=0}^{J-1} p(x_j) \log_2 \left[\frac{1}{p(x_j)} \right] \sum_{k=0}^{K-1} p(y_k | x_j) \\ &= \sum_{j=0}^{J-1} \sum_{k=0}^{K-1} p(y_k | x_j) p(x_j) \log_2 \left[\frac{1}{p(x_j)} \right] \\ &= \sum_{j=0}^{J-1} \sum_{k=0}^{K-1} p(x_j, y_k) \log_2 \left[\frac{1}{p(x_j)} \right] \end{aligned} \quad (9.46)$$

Hence, substituting Equations (9.41) and (9.46) into Equation (9.43) and then combining terms, we obtain

$$I(\mathcal{X}; \mathcal{Y}) = \sum_{j=0}^{J-1} \sum_{k=0}^{K-1} p(x_j, y_k) \log_2 \left[\frac{p(x_j | y_k)}{p(x_j)} \right] \quad (9.47)$$

From *Bayes' rule* for conditional probabilities, we have [see Equations (9.38) and (9.42)]

$$\frac{p(x_j | y_k)}{p(x_j)} = \frac{p(y_k | x_j)}{p(y_k)} \quad (9.48)$$

Hence, substituting Equation (9.48) into Equation (9.47) and interchanging the order of summation, we may write

$$\begin{aligned} I(\mathcal{X}; \mathcal{Y}) &= \sum_{k=0}^{K-1} \sum_{j=0}^{J-1} p(x_j, y_k) \log_2 \left[\frac{p(y_k | x_j)}{p(y_k)} \right] \\ &= I(\mathcal{Y}; \mathcal{X}) \end{aligned} \quad (9.49)$$

which is the desired result.

Property 2

The mutual information is always nonnegative; that is

$$I(\mathcal{X}; \mathcal{Y}) \geq 0 \quad (9.50)$$

To prove this property, we first note from Equation (9.42) that

$$p(x_j | y_k) = \frac{p(x_j, y_k)}{p(y_k)} \quad (9.51)$$

Hence, substituting Equation (9.51) into Equation (9.47), we may express the mutual information of the channel as

$$I(\mathcal{X}; \mathcal{Y}) = \sum_{j=0}^{J-1} \sum_{k=0}^{K-1} p(x_j, y_k) \log_2 \left(\frac{p(x_j, y_k)}{p(x_j)p(y_k)} \right) \quad (9.52)$$

Next, a direct application of the fundamental inequality [defined by Equation (9.12)] yields the desired result

$$I(\mathcal{X}; \mathcal{Y}) \geq 0$$

with equality if, and only if,

$$p(x_j, y_k) = p(x_j)p(y_k) \quad \text{for all } j \text{ and } k \quad (9.53)$$

Property 2 states that *we cannot lose information, on the average, by observing the output of a channel*. Moreover, the mutual information is zero if, and only if, the input and output symbols of the channel are statistically independent, as in Equation (9.53).

Property 3

The mutual information of a channel is related to the joint entropy of the channel input and channel output by

$$I(\mathcal{X}; \mathcal{Y}) = H(\mathcal{X}) + H(\mathcal{Y}) - H(\mathcal{X}, \mathcal{Y}) \quad (9.54)$$

where the joint entropy $H(\mathcal{X}, \mathcal{Y})$ is defined by

$$H(\mathcal{X}, \mathcal{Y}) = \sum_{j=0}^{J-1} \sum_{k=0}^{K-1} p(x_j, y_k) \log_2 \left(\frac{1}{p(x_j, y_k)} \right) \quad (9.55)$$

To prove Equation (9.54), we first rewrite the definition for the joint entropy $H(\mathcal{X}, \mathcal{Y})$ as

$$\begin{aligned} H(\mathcal{X}, \mathcal{Y}) &= \sum_{j=0}^{J-1} \sum_{k=0}^{K-1} p(x_j, y_k) \log_2 \left[\frac{p(x_j)p(y_k)}{p(x_j, y_k)} \right] \\ &\quad + \sum_{j=0}^{J-1} \sum_{k=0}^{K-1} p(x_j, y_k) \log_2 \left[\frac{1}{p(x_j)p(y_k)} \right] \end{aligned} \quad (9.56)$$

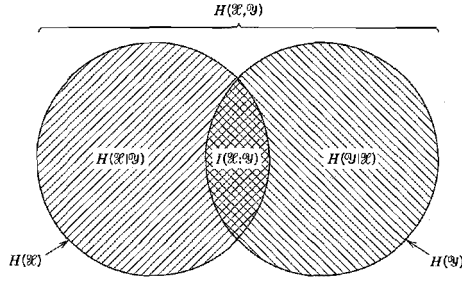


FIGURE 9.9 Illustrating the relations among various channel entropies.

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 diagrammatic 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 $j = 0, 1, \dots, J-1$ and $k = 0, 1, \dots, K-1$. The mutual information of the channel is defined by the first line of Equation (9.49), which is reproduced here for convenience:

$$I(\mathcal{X}; \mathcal{Y}) = \sum_{k=0}^{K-1} \sum_{j=0}^{J-1} p(x_j, y_k) \log_2 \left[\frac{p(y_k | x_j)}{p(y_k)} \right]$$

Here we note that [see Equation (9.38)]

$$p(x_j, y_k) = p(y_k | x_j)p(x_j)$$

Also, from Equation (9.39), we have

$$p(y_k) = \sum_{j=0}^{J-1} p(y_k | x_j)p(x_j)$$

From these three equations we see that it is necessary for us to know the input probability distribution $\{p(x_j) | j = 0, 1, \dots, J-1\}$ so that we may calculate the mutual information $I(\mathcal{X}; \mathcal{Y})$. The mutual information of a channel therefore depends not only on the channel but also on the way in which the channel is used.

The input probability distribution $\{p(x_j)\}$ is obviously independent of the channel. We can then maximize the mutual information $I(\mathcal{X}; \mathcal{Y})$ of the channel with respect to $\{p(x_j)\}$. Hence, we define the channel capacity of a discrete memoryless channel as the maximum mutual information $I(\mathcal{X}; \mathcal{Y})$ in any single use of the channel (i.e., signaling interval), where the maximization is over all possible input probability distributions $\{p(x_j)\}$ on \mathcal{X} . The channel capacity is commonly denoted by C . We thus write

$$C = \max_{\{p(x_j)\}} I(\mathcal{X}; \mathcal{Y}) \quad (9.59)$$

The channel capacity C is measured in *bits per channel use*, or *bits per transmission*.

Note that the channel capacity C is a function only of the transition probabilities $p(y_k | x_j)$, which define the channel. The calculation of C involves maximization of the mutual information $I(\mathcal{X}; \mathcal{Y})$ over J variables [i.e., the input probabilities $p(x_0), \dots, p(x_{J-1})$] subject to two constraints:

$$p(x_j) \geq 0 \text{ for all } j$$

and

$$\sum_{j=0}^{J-1} p(x_j) = 1$$

In general, the variational problem of finding the channel capacity C is a challenging task.

EXAMPLE 9.5 Binary Symmetric Channel (Revisited)

Consider again the *binary symmetric channel*, which is described by the *transition probability diagram* of Figure 9.8. This diagram is uniquely defined by the conditional probability of error p .

The entropy $H(X)$ is maximized when the channel input probability $p(x_0) = p(x_1) = 1/2$, where x_0 and x_1 are each 0 or 1. The mutual information $I(\mathcal{X}; \mathcal{Y})$ is similarly maximized, so that we may write

$$C = I(\mathcal{X}; \mathcal{Y}) |_{p(x_0)=p(x_1)=1/2}$$

From Figure 9.8, we have

$$p(y_0 | x_1) = p(y_1 | x_0) = p$$

and

$$p(y_0 | x_0) = p(y_1 | x_1) = 1 - p$$

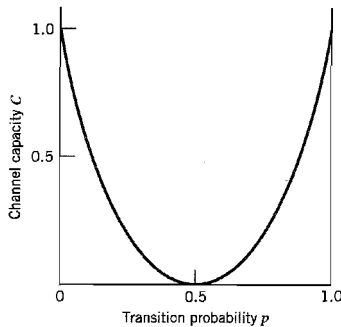


FIGURE 9.10 Variation of channel capacity of a binary symmetric channel with transition probability p .

Therefore, substituting these channel transition probabilities into Equation (9.49) with $J = K = 2$, and then setting the input probability $p(x_0) = p(x_1)$ in accordance with Equation (9.59), we find that the capacity of the binary symmetric channel is

$$C = 1 + p \log_2 p + (1 - p) \log_2(1 - p) \quad (9.60)$$

Using the definition of the entropy function given in Equation (9.16), we may reduce Equation (9.60) to

$$C = 1 - H(p)$$

The channel capacity C varies with the probability of error (transition probability) p in a convex manner as shown in Figure 9.10, which is symmetric about $p = 1/2$. Comparing the curve in this figure with that in Figure 9.2, we may make the following observations:

1. When the channel is *noise free*, permitting us to set $p = 0$, the channel capacity C attains its maximum value of one bit per channel use, which is exactly the information in each channel input. At this value of p , the entropy function $H(p)$ attains its minimum value of zero.
2. When the conditional probability of error $p = 1/2$ due to noise, the channel capacity C attains its minimum value of zero, whereas the entropy function $H(p)$ attains its maximum value of unity; in such a case the channel is said to be *useless*. ◀

9.8 Channel-Coding Theorem

The inevitable presence of *noise* in a channel causes discrepancies (errors) between the output and input data sequences of a digital communication system. For a relatively noisy channel (e.g., wireless communication channel), the probability of error may reach a value as high as 10^{-1} , which means that (on the average) only 9 out of 10 transmitted bits are received correctly. For many applications, this *level of reliability* is unacceptable. Indeed, a probability of error equal to 10^{-6} or even lower is often a necessary requirement. To achieve such a high level of performance, we resort to the use of channel coding.

The design goal of channel coding is to increase the resistance of a digital communication system to channel noise. Specifically, *channel coding* consists of *mapping* the incoming data sequence into a channel input sequence, and *inverse mapping* the channel output sequence into an output data sequence in such a way that the overall effect of

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 purpose 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.

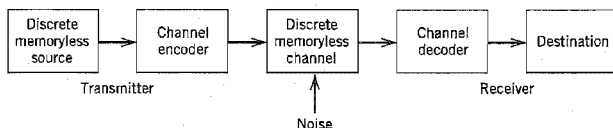


FIGURE 9.11 Block diagram of digital communication system.

Specifically, the *channel coding theorem*⁸ for a discrete memoryless channel is stated in two parts as follows.

- (i) Let a discrete memoryless source with an alphabet \mathcal{S} have entropy $H(\mathcal{S})$ and produce symbols once every T_s seconds. Let a discrete memoryless channel have capacity C and be used once every T_c seconds. Then, if

$$\frac{H(\mathcal{S})}{T_s} \leq \frac{C}{T_c} \quad (9.61)$$

there exists a coding scheme for which the source output can be transmitted over the channel and be reconstructed with an arbitrarily small probability of error. The parameter C/T_c is called the *critical rate*. When Equation (9.61) is satisfied with the equality sign, the system is said to be signaling at the critical rate.

- (ii) Conversely, if

$$\frac{H(\mathcal{S})}{T_s} > \frac{C}{T_c}$$

it is not possible to transmit information over the channel and reconstruct it with an arbitrarily small probability of error.

The channel coding theorem is the single most important result of information theory. The theorem specifies the channel capacity C as a *fundamental limit* on the rate at which the transmission of reliable error-free messages can take place over a discrete memoryless channel. However, it is important to note the following:

- ▶ The channel coding theorem does not show us how to construct a good code. Rather, the theorem should be viewed as an *existence proof* in the sense that it tells us that if the condition of Equation (9.61) is satisfied, then good codes do exist. (Later in Chapter 10 we describe several good codes for discrete memoryless channels.)
- ▶ The theorem does not have a precise result for the probability of symbol error after decoding the channel output. Rather, it tells us that the probability of symbol error tends to zero as the length of the code increases, again provided that the condition of Equation (9.61) is satisfied.

Note also that power and bandwidth constraints were hidden in the discussion presented here. Nevertheless, these two system constraints do actually show up in the channel matrix \mathbf{P} of the discrete memoryless channel. This observation is readily confirmed by linking the results of Example 9.5 on the binary symmetric channel with the noise analysis for the PCM receiver presented in Section 5.3.

■ APPLICATION OF THE CHANNEL CODING THEOREM TO BINARY SYMMETRIC CHANNELS

Consider a discrete memoryless source that emits equally likely binary symbols (0s and 1s) once every T_s seconds. With the source entropy equal to one bit per source symbol (see Example 9.1), the information rate of the source is $(1/T_s)$ bits per second. The source sequence is applied to a channel encoder with code rate r . The channel encoder produces a symbol once every T_c seconds. Hence, the encoded symbol transmission rate is $(1/T_c)$ symbols per second. The channel encoder engages a binary symmetric channel once every T_c seconds. Hence, the channel capacity per unit time is (C/T_c) bits per second, where C

is determined by the prescribed channel transition probability p in accordance with Equation (9.60). Accordingly, the channel coding theorem [part (i)] implies that if

$$\frac{1}{T_s} \leq \frac{C}{T_c} \quad (9.62)$$

the probability of error can be made arbitrarily low by the use of a suitable channel encoding scheme. But the ratio T_c/T_s equals the code rate of the channel encoder:

$$r = \frac{T_c}{T_s} \quad (9.63)$$

Hence, we may restate the condition of Equation (9.62) simply as

$$r \leq C \quad (9.64)$$

That is, for $r \leq C$, there exists a code (with code rate less than or equal to C) capable of achieving an arbitrarily low probability of error.

► EXAMPLE 9.6 Repetition Code

In this example, we present a graphical interpretation of the channel coding theorem. We also bring out a surprising aspect of the theorem by taking a look at a simple coding scheme.

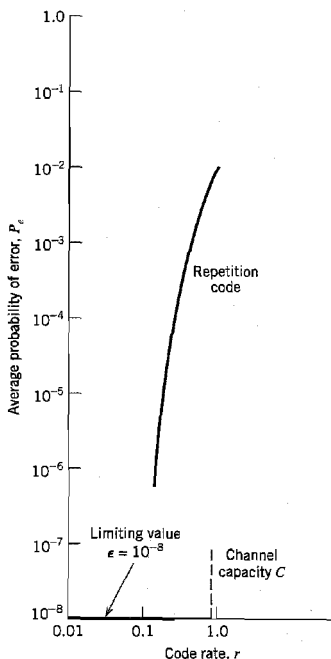


FIGURE 9.12 Illustrating significance of the channel coding theorem.

TABLE 9.3 Average probability of error for repetition code

Code Rate, $r = 1/n$	Average Probability of Error, P_e
1	10^{-2}
$\frac{1}{3}$	3×10^{-4}
$\frac{1}{5}$	10^{-6}
$\frac{1}{7}$	4×10^{-7}
$\frac{1}{9}$	10^{-8}
$\frac{1}{11}$	5×10^{-10}

Consider first a binary symmetric channel with transition probability $p = 10^{-2}$. For this value of p , we find from Equation (9.60) that the channel capacity $C = 0.9192$. Hence, from the channel coding theorem, we may state that for any $\epsilon > 0$ and $r \leq 0.9192$, there exists a code of large enough length n and code rate r , and an appropriate decoding algorithm, such that when the coded bit stream is sent over the given channel, the average probability of channel decoding error is less than ϵ . This result is depicted in Figure 9.12 for the limiting value $\epsilon = 10^{-8}$.

To put the significance of this result in perspective, consider next a simple coding scheme that involves the use of a *repetition code*, in which each bit of the message is repeated several times. Let each bit (0 or 1) be repeated n times, where $n = 2m + 1$ is an odd integer. For example, for $n = 3$, we transmit 0 and 1 as 000 and 111, respectively. Intuitively, it would seem logical to use a *majority rule* for decoding, which operates as follows: *If in a block of n received bits (representing one bit of the message), the number of 0s exceeds the number of 1s, the decoder decides in favor of a 0. Otherwise, it decides in favor of a 1.* Hence, an error occurs when $m + 1$ or more bits out of $n = 2m + 1$ bits are received incorrectly. Because of the assumed symmetric nature of the channel, the average probability of error P_e is independent of the *a priori* probabilities of 0 and 1. Accordingly, we find that P_e is given by (see Problem 9.24)

$$P_e = \sum_{i=m+1}^n \binom{n}{i} p^i (1-p)^{n-i} \quad (9.65)$$

where p is the transition probability of the channel.

Table 9.3 gives the average probability of error P_e for a repetition code, which is calculated by using Equation (9.65) for different values of the code rate r . The values given here assume the use of a binary symmetric channel with transition probability $p = 10^{-2}$. The improvement in reliability displayed in Table 9.3 is achieved at the cost of decreasing code rate. The results of this table are also shown plotted as the curve labeled “repetition code” in Figure 9.12. This curve illustrates the *exchange of code rate for message reliability*, which is a characteristic of repetition codes.

This example highlights the unexpected result presented to us by the channel coding theorem. The result is that it is not necessary to have the code rate r approach zero (as in the case of repetition codes) so as to achieve more and more reliable operation of the communication link. The theorem merely requires that the code rate be less than the channel capacity C . ◀

9.9 Differential Entropy and Mutual Information for Continuous Ensembles

The sources and channels considered in our discussion of information-theoretic concepts thus far have involved ensembles of random variables that are *discrete* in amplitude. In

this section, we extend some of these concepts to *continuous* random variables and random vectors. The motivation for doing so is to pave the way for the description of another fundamental limit in information theory, which we take up in Section 9.10.

Consider a continuous random variable X with the *probability density function* $f_X(x)$. By analogy with the entropy of a discrete random variable, we introduce the following definition:

$$h(X) = \int_{-\infty}^{\infty} f_X(x) \log_2 \left[\frac{1}{f_X(x)} \right] dx \quad (9.66)$$

We refer to $h(X)$ as the *differential entropy* of X to distinguish it from the ordinary or *absolute entropy*. We do so in recognition of the fact that although $h(X)$ is a useful mathematical quantity to know, it is *not* in any sense a measure of the randomness of X . Nevertheless, we justify the use of Equation (9.66) in what follows. We begin by viewing the continuous random variable X as the limiting form of a discrete random variable that assumes the value $x_k = k \Delta x$, where $k = 0, \pm 1, \pm 2, \dots$, and Δx approaches zero. By definition, the continuous random variable X assumes a value in the interval $[x_k, x_k + \Delta x]$ with probability $f_X(x_k) \Delta x$. Hence, permitting Δx to approach zero, the ordinary entropy of the continuous random variable X may be written in the limit as follows:

$$\begin{aligned} H(X) &= \lim_{\Delta x \rightarrow 0} \sum_{k=-\infty}^{\infty} f_X(x_k) \Delta x \log_2 \left(\frac{1}{f_X(x_k) \Delta x} \right) \\ &= \lim_{\Delta x \rightarrow 0} \left[\sum_{k=-\infty}^{\infty} f_X(x_k) \log_2 \left(\frac{1}{f_X(x_k)} \right) \Delta x - \log_2 \Delta x \sum_{k=-\infty}^{\infty} f_X(x_k) \Delta x \right] \\ &= \int_{-\infty}^{\infty} f_X(x) \log_2 \left(\frac{1}{f_X(x)} \right) dx - \lim_{\Delta x \rightarrow 0} \log_2 \Delta x \int_{-\infty}^{\infty} f_X(x) dx \\ &= h(X) - \lim_{\Delta x \rightarrow 0} \log_2 \Delta x \end{aligned} \quad (9.67)$$

where, in the last line, we have made use of Equation (9.66) and the fact that the total area under the curve of the probability density function $f_X(x)$ is unity. In the limit as Δx approaches zero, $-\log_2 \Delta x$ approaches infinity. This means that the entropy of a continuous random variable is infinitely large. Intuitively, we would expect this to be true, because a continuous random variable may assume a value anywhere in the interval $(-\infty, \infty)$ and the uncertainty associated with the variable is on the order of infinity. We avoid the problem associated with the term $\log_2 \Delta x$ by adopting $h(X)$ as a differential entropy, with the term $-\log_2 \Delta x$ serving as reference. Moreover, since the information transmitted over a channel is actually the difference between two entropy terms that have a common reference, the information will be the same as the difference between the corresponding differential entropy terms. We are therefore perfectly justified in using the term $h(X)$, defined in Equation (9.66), as the differential entropy of the continuous random variable X .

When we have a continuous random vector \mathbf{X} consisting of n random variables X_1, X_2, \dots, X_n , we define the differential entropy of \mathbf{X} as the *n-fold integral*

$$h(\mathbf{X}) = \int_{-\infty}^{\infty} f_{\mathbf{X}}(\mathbf{x}) \log_2 \left[\frac{1}{f_{\mathbf{X}}(\mathbf{x})} \right] d\mathbf{x} \quad (9.68)$$

where $f_{\mathbf{X}}(\mathbf{x})$ is the *joint probability density function* of \mathbf{X} .

► EXAMPLE 9.7 Uniform Distribution

Consider a random variable X uniformly distributed over the interval $(0, a)$. The probability density function of X is

$$f_X(x) = \begin{cases} \frac{1}{a}, & 0 < x < a \\ 0, & \text{otherwise} \end{cases}$$

Applying Equation (9.66) to this distribution, we get

$$\begin{aligned} h(X) &= \int_0^a \frac{1}{a} \log(a) dx \\ &= \log a \end{aligned} \quad (9.69)$$

Note that $\log a < 0$ for $a < 1$. Thus this example shows that, unlike a discrete random variable, the differential entropy of a continuous random variable can be negative. ◀

► EXAMPLE 9.8 Gaussian Distribution

Consider an arbitrary pair of random variables X and Y , whose probability density functions are respectively denoted by $f_Y(x)$ and $f_X(x)$ where x is merely a dummy variable. Adapting the fundamental inequality of Equation (9.12) to the situation at hand, we may write⁹

$$\int_{-\infty}^{\infty} f_Y(x) \log_2 \left(\frac{f_X(x)}{f_Y(x)} \right) dx \leq 0 \quad (9.70)$$

or, equivalently,

$$-\int_{-\infty}^{\infty} f_Y(x) \log_2 f_Y(x) dx \leq -\int_{-\infty}^{\infty} f_Y(x) \log_2 f_X(x) dx \quad (9.71)$$

The quantity on the left-hand side of Equation (9.71) is the differential entropy of the random variable Y ; hence,

$$h(Y) \leq -\int_{-\infty}^{\infty} f_Y(x) \log_2 f_X(x) dx \quad (9.72)$$

Suppose now the random variables X and Y are described as follows:

- The random variables X and Y have the *same mean* μ and the *same variance* σ^2 .
- The random variable X is *Gaussian distributed* as shown by

$$f_X(x) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right) \quad (9.73)$$

Hence, substituting Equation (9.73) into Equation (9.72), and changing the base of the logarithm from 2 to $e = 2.7183$, we get

$$h(Y) \leq -\log_2 e \int_{-\infty}^{\infty} f_Y(x) \left(-\frac{(x-\mu)^2}{2\sigma^2} - \log(\sqrt{2\pi}\sigma) \right) dx \quad (9.74)$$

We now recognize the following properties of the random variable Y (given that its mean is μ and its variance is σ^2):

$$\begin{aligned} \int_{-\infty}^{\infty} f_Y(x) dx &= 1 \\ \int_{-\infty}^{\infty} (x-\mu)^2 f_Y(x) dx &= \sigma^2 \end{aligned}$$

We may therefore simplify Equation (9.74) as

$$b(Y) \leq \frac{1}{2} \log_2(2\pi e\sigma^2) \quad (9.75)$$

The quantity on the right-hand side of Equation (9.75) is in fact the differential entropy of the Gaussian random variable X :

$$b(X) = \frac{1}{2} \log_2(2\pi e\sigma^2) \quad (9.76)$$

Finally, combining Equations (9.75) and (9.76), we may write

$$b(Y) \leq b(X), \quad \begin{cases} X: \text{Gaussian random variable} \\ Y: \text{another random variable} \end{cases} \quad (9.77)$$

where equality holds if, and only if, $Y = X$.

We may now summarize the results of this important example as two entropic properties of a Gaussian random variable:

1. For a finite variance σ^2 , the Gaussian random variable has the largest differential entropy attainable by any random variable.
2. The entropy of a Gaussian random variable X is uniquely determined by the variance of X (i.e., it is independent of the mean of X).

Indeed, it is because of Property 1 that the Gaussian channel model is so widely used as a conservative model in the study of digital communication systems. ◀

■ MUTUAL INFORMATION

Consider next a pair of continuous random variables X and Y . By analogy with Equation (9.47), we define the *mutual information* between the random variables X and Y as follows:

$$I(X; Y) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} f_{X,Y}(x, y) \log_2 \left[\frac{f_X(x|y)}{f_X(x)} \right] dx dy \quad (9.78)$$

where $f_{X,Y}(x, y)$ is the joint probability density function of X and Y , and $f_X(x|y)$ is the conditional probability density function of X , given that $Y = y$. Also, by analogy with Equations (9.45), (9.50), (9.43), and (9.44) we find that the mutual information $I(X; Y)$ has the following properties:

$$1. I(X; Y) = I(Y; X) \quad (9.79)$$

$$2. I(X; Y) \geq 0 \quad (9.80)$$

$$3. I(X; Y) = b(X) - b(X|Y) \\ = b(Y) - b(Y|X) \quad (9.81)$$

The parameter $b(X)$ is the differential entropy of X ; likewise for $b(Y)$. The parameter $b(X|Y)$ is the *conditional differential entropy* of X , given Y ; it is defined by the double integral (see Equation (9.41))

$$b(X|Y) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} f_{X,Y}(x, y) \log_2 \left[\frac{1}{f_X(x|y)} \right] dx dy \quad (9.82)$$

The parameter $h(Y|X)$ is the conditional differential entropy of Y , given X ; it is defined in a manner similar to $h(X|Y)$.

9.10 Information Capacity Theorem

In this section, we use the idea of mutual information to formulate the information capacity theorem for *band-limited, power-limited Gaussian channels*. To be specific, consider a zero-mean stationary process $X(t)$ that is band-limited to B hertz. Let X_k , $k = 1, 2, \dots, K$, denote the continuous random variables obtained by uniform sampling of the process $X(t)$ at the Nyquist rate of $2B$ samples per second. These samples are transmitted in T seconds over a noisy channel, also band-limited to B hertz. Hence, the number of samples, K , is given by

$$K = 2BT \quad (9.83)$$

We refer to X_k as a sample of the *transmitted signal*. The channel output is perturbed by *additive white Gaussian noise* (AWGN) of zero mean and power spectral density $N_0/2$. The noise is band-limited to B hertz. Let the continuous random variables Y_k , $k = 1, 2, \dots, K$ denote samples of the received signal, as shown by

$$Y_k = X_k + N_k, \quad k = 1, 2, \dots, K \quad (9.84)$$

The noise sample N_k is Gaussian with zero mean and variance given by

$$\sigma^2 = N_0 B \quad (9.85)$$

We assume that the samples Y_k , $k = 1, 2, \dots, K$ are statistically independent.

A channel for which the noise and the received signal are as described in Equations (9.84) and (9.85) is called a *discrete-time, memoryless Gaussian channel*. It is modeled as in Figure 9.13. To make meaningful statements about the channel, however, we have to assign a *cost* to each channel input. Typically, the transmitter is *power limited*; it is therefore reasonable to define the cost as

$$E[X_k^2] = P, \quad k = 1, 2, \dots, K \quad (9.86)$$

where P is the *average transmitted power*. The *power-limited Gaussian channel* described herein is of not only theoretical but also practical importance in that it models many communication channels, including line-of-sight radio and satellite links.

The *information capacity* of the channel is defined as the maximum of the mutual information between the channel input X_k and the channel output Y_k over all distributions on the input X_k that satisfy the power constraint of Equation (9.86). Let $I(X_k; Y_k)$ denote

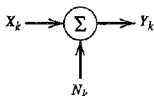


FIGURE 9.13 Model of discrete-time, memoryless Gaussian channel.

the mutual information between X_k and Y_k . We may then define the information capacity of the channel as

$$C = \max_{f_{X_k}(x)} \{I(X_k; Y_k) : E[X_k^2] = P\} \quad (9.87)$$

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

$$I(X_k; Y_k) = h(Y_k) - h(Y_k | X_k) \quad (9.88)$$

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):

$$h(Y_k | X_k) = h(N_k) \quad (9.89)$$

Hence, we may rewrite Equation (9.88) as

$$I(X_k; Y_k) = h(Y_k) - h(N_k) \quad (9.90)$$

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

$$C = I(X_k; Y_k) : X_k \text{ Gaussian, } E[X_k^2] = P \quad (9.91)$$

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

$$h(Y_k) = \frac{1}{2} \log_2 [2\pi e(P + \sigma^2)] \quad (9.92)$$

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

$$h(N_k) = \frac{1}{2} \log_2 (2\pi e\sigma^2) \quad (9.93)$$

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:

$$C = \frac{1}{2} \log_2 \left(1 + \frac{P}{\sigma^2} \right) \text{ bits per transmission} \quad (9.94)$$

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:

$$C = B \log_2 \left(1 + \frac{P}{N_0 B} \right) \text{ bits per second} \quad (9.95)$$

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 PACKING¹¹

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.

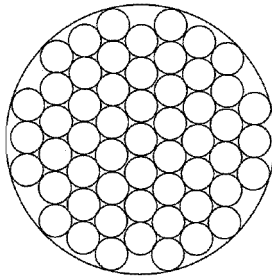


FIGURE 9.14 The sphere-packing problem.

We may thus visualize the picture portrayed in Figure 9.14. With everything inside a small sphere of radius $\sqrt{n}\sigma^2$ assigned to the code word on which it is centered, it is reasonable to say that when this particular code word is transmitted, the probability that the received vector will lie inside the correct “decoding” sphere is high. The key question is: How many decoding spheres can be packed inside the larger sphere of received vectors? In other words, how many code words can we in fact choose? To answer this question, we first recognize that the volume of an n -dimensional sphere of radius r may be written as $A_n r^n$, where A_n is a scaling factor. We may therefore make the following statements:

- The volume of the sphere of received vectors is $A_n [n(P + \sigma^2)]^{n/2}$.
- The volume of the decoding sphere is $A_n (n\sigma^2)^{n/2}$.

Accordingly, it follows that the maximum number of *nonintersecting* decoding spheres that can be packed inside the sphere of possible received vectors is

$$\frac{A_n [n(P + \sigma^2)]^{n/2}}{A_n (n\sigma^2)^{n/2}} = \left(1 + \frac{P}{\sigma^2}\right)^{n/2} = 2^{(n/2) \log_2(1 + P/\sigma^2)} \quad (9.96)$$

Taking the logarithm of this result to base 2, we readily see that the maximum number of bits per transmission for a low probability of error is indeed as defined previously in Equation (9.94).

► EXAMPLE 9.9 Reconfiguration of Constellation for Reduced Power

To illustrate the idea of sphere packing, consider the 64-QAM square constellation of Figure 9.15a. The figure depicts two-dimensional nonintersecting decoding spheres centered on the message points in the constellation. In trying to pack the decoding spheres as tightly as possible while maintaining the same Euclidean distance between the message points as before, we obtain the alternative constellation shown in Figure 9.15b. With a common Euclidean distance between the message points, the two constellations of Figure 9.15 produce approximately the same bit error rate, assuming the use of a high enough signal-to-noise ratio over an AWGN channel; see, for example, Equation (5.95). However, comparing these two constellations, we find that the sum of squared Euclidean distances from the message points to the origin in Figure 9.15b is smaller than that in Figure 9.15a. It follows therefore that the tightly packed constellation of Figure 9.15b has an advantage over the square constellation

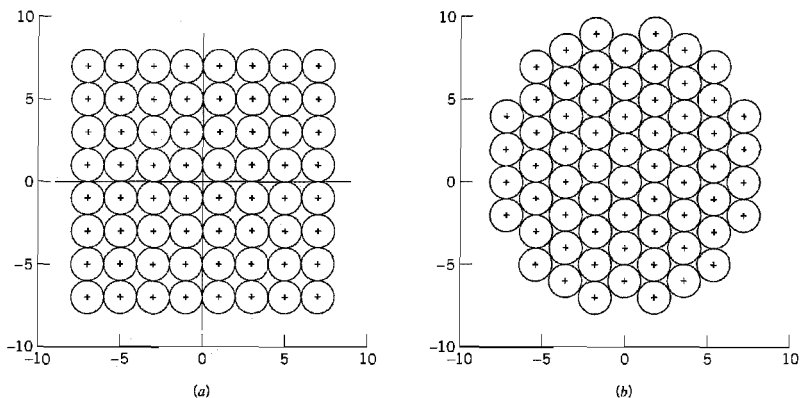


FIGURE 9.15 (a) Square 64-QAM constellation. (b) The most tightly coupled alternative to that of part a.

of Figure 9.15a: a smaller transmitted average signal energy per symbol for the same bit error rate on an AWGN channel. \blacktriangleleft

9.11 Implications of the Information Capacity Theorem

Now that we have an intuitive feel for the information capacity theorem, we may go on to discuss its implications in the context of a Gaussian channel that is limited in both power and bandwidth. For the discussion to be useful, however, we need an ideal framework against which the performance of a practical communication system can be assessed. To this end, we introduce the notion of an *ideal system* defined as one that transmits data at a bit rate R_b equal to the information capacity C . We may then express the average transmitted power as

$$P = E_b C \quad (9.97)$$

where E_b is the transmitted energy per bit. Accordingly, the ideal system is defined by the equation

$$\frac{C}{B} = \log_2 \left(1 + \frac{E_b}{N_0} \frac{C}{B} \right) \quad (9.98)$$

Equivalently, we may define the *signal energy-per-bit to noise power spectral density ratio* E_b/N_0 in terms of the ratio C/B for the ideal system as

$$\frac{E_b}{N_0} = \frac{2^{C/B} - 1}{C/B} \quad (9.99)$$

A plot of bandwidth efficiency R_b/B versus E_b/N_0 is called the *bandwidth-efficiency diagram*. A generic form of this diagram is displayed in Figure 9.16, where the curve labeled

“capacity boundary” corresponds to the ideal system for which $R_b = C$. Based on Figure 9.16, we can make the following observations:

1. For *infinite bandwidth*, the ratio E_b/N_0 approaches the limiting value

$$\begin{aligned} \left(\frac{E_b}{N_0}\right)_{\infty} &= \lim_{B \rightarrow \infty} \left(\frac{E_b}{N_0}\right) \\ &= \log 2 = 0.693 \end{aligned} \quad (9.100)$$

This value is called the *Shannon limit* for an AWGN channel, assuming a code rate of zero. Expressed in decibels, it equals -1.6 dB. The corresponding limiting value of the channel capacity is obtained by letting the channel bandwidth B in Equation (9.95) approach infinity; we thus find that

$$\begin{aligned} C_{\infty} &= \lim_{B \rightarrow \infty} C \\ &= \frac{P}{N_0} \log_2 e \end{aligned} \quad (9.101)$$

where e is the base of the natural logarithm.

2. The *capacity boundary*, defined by the curve for the critical bit rate $R_b = C$, separates combinations of system parameters that have the potential for supporting error-free transmission ($R_b < C$) from those for which error-free transmission is not possible ($R_b > C$). The latter region is shown shaded in Figure 9.16.
3. The diagram highlights potential *trade-offs* among E_b/N_0 , R_b/B , and probability of symbol error P_e . In particular, we may view movement of the operating point along

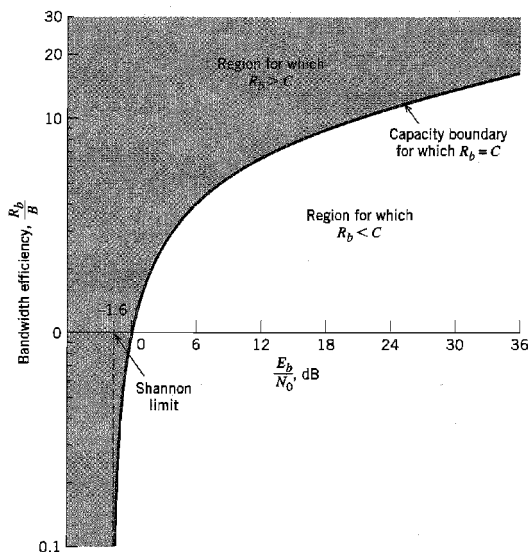


FIGURE 9.16 Bandwidth-efficiency diagram.

a horizontal line as trading P_e versus E_b/N_0 for a fixed R_b/B . On the other hand, we may view movement of the operating point along a vertical line as trading P_e versus R_b/B for a fixed E_b/N_0 .

► EXAMPLE 9.10 *M*-ary PCM

In this example, we look at an *M*-ary PCM system in light of the channel capacity theorem under the assumption that the system operates above the error threshold. That is, the average probability of error due to channel noise is negligible.

We assume that the *M*-ary PCM system uses a code word consisting of *n* code elements, each having one of *M* possible discrete amplitude levels; hence the name “*M*-ary.” From Chapter 3 we recall that for a PCM system to operate above the error threshold, there must be provision for a noise margin that is sufficiently large to maintain a negligible error rate due to channel noise. This, in turn, means there must be a certain separation between these *M* discrete amplitude levels. Call this separation $k\sigma$, where k is a constant and $\sigma^2 = N_0B$ is the noise variance measured in a channel bandwidth *B*. The number of amplitude levels *M* is usually an integer power of 2. The average transmitted power will be least if the amplitude range is symmetrical about zero. Then the discrete amplitude levels, normalized with respect to the separation $k\sigma$, will have the values $\pm 1/2, \pm 3/2, \dots, \pm (M-1)/2$. We assume that these *M* different amplitude levels are equally likely. Accordingly, we find that the average transmitted power is given by

$$\begin{aligned} P &= \frac{2}{M} \left[\left(\frac{1}{2} \right)^2 + \left(\frac{3}{2} \right)^2 + \dots + \left(\frac{M-1}{2} \right)^2 \right] (k\sigma)^2 \\ &= k^2 \sigma^2 \left(\frac{M^2 - 1}{12} \right) \end{aligned} \quad (9.102)$$

Suppose that the *M*-ary PCM system described herein is used to transmit a message signal with its highest frequency component equal to *W* hertz. The signal is sampled at the Nyquist rate of $2W$ samples per second. We assume that the system uses a quantizer of the midrise type, with *L* equally likely representation levels. Hence, the probability of occurrence of any one of the *L* representation levels is $1/L$. Correspondingly, the amount of information carried by a single sample of the signal is $\log_2 L$ bits. With a maximum sampling rate of $2W$ samples per second, the maximum rate of information transmission of the PCM system, measured in bits per second, is given by

$$R_b = 2W \log_2 L \text{ bits per second} \quad (9.103)$$

Since the PCM system uses a code word consisting of *n* code elements, each having one of *M* possible discrete amplitude values, we have M^n different possible code words. For a unique encoding process, we require

$$L = M^n \quad (9.104)$$

Clearly, the rate of information transmission in the system is unaffected by the use of an encoding process. We may therefore eliminate *L* between Equations (9.103) and (9.104) to obtain

$$R_b = 2Wn \log_2 M \text{ bits per second} \quad (9.105)$$

Equation (9.102) defines the average transmitted power required to maintain an *M*-ary PCM system operating above the error threshold. Hence, solving this equation for the number of discrete amplitude levels, *M*, we get

$$M = \left(1 + \frac{12P}{k^2 N_0 B} \right)^{1/2} \quad (9.106)$$

where $\sigma^2 = N_0 B$ is the variance of the channel noise measured in a bandwidth B . Therefore, substituting Equation (9.106) into Equation (9.105), we obtain

$$R_b = W \log_2 \left(1 + \frac{12P}{k^2 N_0 B} \right) \quad (9.107)$$

The channel bandwidth B required to transmit a rectangular pulse of duration $1/2nW$ (representing a code element in the code word) is given by (see Chapter 3)

$$B = \kappa n W$$

where κ is a constant with a value lying between 1 and 2. Using the minimum possible value $\kappa = 1$, we find that the channel bandwidth $B = nW$. We may thus rewrite Equation (9.107) as

$$R_b = B \log_2 \left(1 + \frac{12P}{k^2 N_0 B} \right) \quad (9.108)$$

The *ideal system* is described by Shannon's channel capacity theorem, given in Equation (9.95). Hence, comparing Equation (9.108) with Equation (9.95), we see that they are identical if the average transmitted power in the PCM system is increased by the factor $k^2/12$, compared with the ideal system. Perhaps the most interesting point to note about Equation (9.108) is that the form of the equation is right: *Power and bandwidth in a PCM system are exchanged on a logarithmic basis, and the information capacity C is proportional to the channel bandwidth B .* ◀

► EXAMPLE 9.11 M-ary PSK and M-ary FSK

In this example, we compare the bandwidth-power exchange capabilities of M -ary PSK and M -ary FSK signals in light of Shannon's information capacity theorem. Consider first a coherent M -ary PSK system that employs a *nonorthogonal* set of M phase-shifted signals for the transmission of binary data. Each signal in the set represents a symbol with $\log_2 M$ bits. Using the definition of null-to-null bandwidth, we may express the bandwidth efficiency of M -ary PSK as follows [see Equation (6.51)]:

$$\frac{R_b}{B} = \frac{\log_2 M}{2}$$

In Figure 9.17*a*, we show the operating points for different numbers of phase levels $M = 2, 4, 8, 16, 32, 64$. Each point corresponds to an average probability of symbol error $P_e = 10^{-5}$. In the figure we have also included the capacity boundary for the ideal system. We observe from Figure 9.17*a* that as M is increased, the bandwidth efficiency is improved, but the value of E_b/N_0 required for error-free transmission moves away from the Shannon limit.

Consider next a coherent M -ary FSK system that uses an *orthogonal* set of M frequency-shifted signals for the transmission of binary data, with the separation between adjacent signal frequencies set at $1/2T$, where T is the symbol period. As with the M -ary PSK, each signal in the set represents a symbol with $\log_2 M$ bits. The bandwidth efficiency of M -ary FSK is as follows [see Equation (6.143)]:

$$\frac{R_b}{B} = \frac{2 \log_2 M}{M}$$

In Figure 9.17*b*, we show the operating points for different numbers of frequency levels $M = 2, 4, 8, 16, 32, 64$ for an average probability of symbol error $P_e = 10^{-5}$. In the figure, we have also included the capacity boundary for the ideal system. We see that increasing M in (orthogonal) M -ary FSK has the opposite effect to that in (nonorthogonal) M -ary PSK. In particular, as M is increased, which is equivalent to increased bandwidth requirement, the operating point moves closer to the Shannon limit. ◀

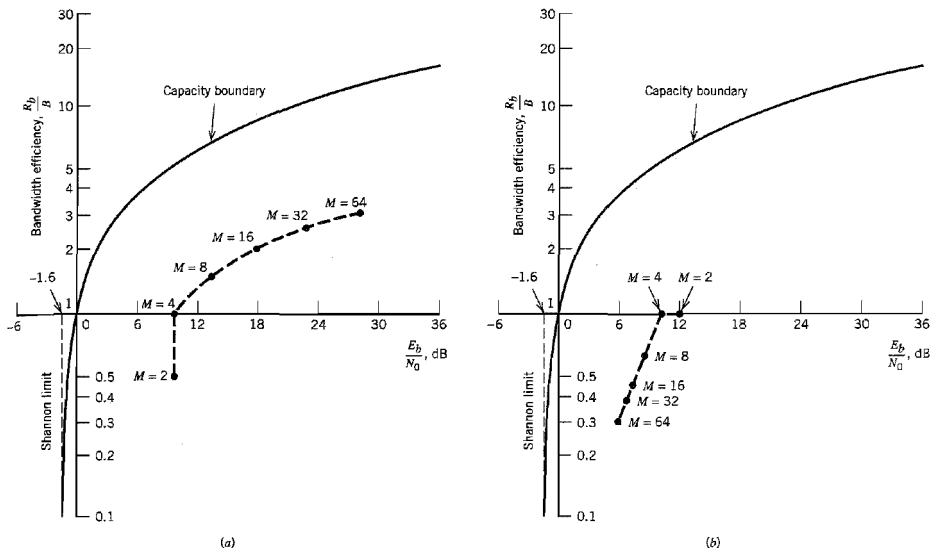


FIGURE 9.17 (a) Comparison of M -ary PSK against the ideal system for $P_e = 10^{-5}$ and increasing M . (b) Comparison of M -ary FSK against the ideal system for $P_e = 10^{-5}$ and increasing M .

EXAMPLE 9.12 Capacity of Binary-Input AWGN Channel

In this example, we investigate the capacity of an AWGN channel using *encoded* binary antipodal signaling (i.e., levels -1 and $+1$ for binary symbols 0 and 1 , respectively). In particular, we address the issue of determining the minimum achievable bit error rate as a function of E_b/N_0 for varying code rate r . It is assumed that the binary symbols 0 and 1 are equiprobable.

Let the random variables X and Y denote the channel input and channel output, respectively; X is a discrete variable, whereas Y is a continuous variable. In light of the second line of Equation (9.81), we may express the mutual information between the channel input and channel output as

$$I(X; Y) = h(Y) - h(Y|X)$$

The second term, $h(Y|X)$, is the conditional differential entropy of the channel output Y , given the channel input X . By virtue of Equations (9.89) and (9.93), this term is just the entropy of a Gaussian distribution. Hence, using σ^2 to denote the variance of the channel noise, we may write

$$h(Y|X) = \frac{1}{2} \log_2(2\pi\sigma^2)$$

Next, the first term, $h(Y)$, is the differential entropy of the channel output Y . With the use of binary antipodal signaling, the probability density function of Y , given $X = x$, is a mixture of two Gaussian distributions with common variance σ^2 and mean values -1 and $+1$, as shown by

$$f_Y(y_i|x) = \frac{1}{2} \left[\frac{\exp(-(y_i + 1)^2/2\sigma^2)}{\sqrt{2\pi}\sigma} + \frac{\exp(-(y_i - 1)^2/2\sigma^2)}{\sqrt{2\pi}\sigma} \right] \quad (9.109)$$

Hence, we may determine the differential entropy of Y using the formula

$$h(Y) = - \int_{-\infty}^{\infty} f_Y(y_i | x) \log_2[f_Y(y_i | x)] dy_i$$

where $f_Y(y_i | x)$ is defined by Equation (9.109). From the formulas of $h(Y | X)$ and $h(Y)$, it is clear that the mutual information is solely a function of the noise variance σ^2 . Using $M(\sigma^2)$ to denote this functional dependence, we may thus write

$$I(X; Y) = M(\sigma^2)$$

Unfortunately, there is no closed formula that we can derive for $M(\sigma^2)$ because of the difficulty of determining $h(Y)$. Nevertheless, the differential entropy $h(Y)$ can be well approximated using *Monte Carlo integration*, which is straightforward to program on a digital computer; see Problem 9.36.

Because symbols 0 and 1 are equiprobable, it follows that the channel capacity C is equal to the mutual information between X and Y . Hence, for error-free data transmission over the AWGN channel, the code rate r must satisfy the condition

$$r < M(\sigma^2) \quad (9.110)$$

A robust measure of the ratio E_b/N_0 is

$$\frac{E_b}{N_0} = \frac{P}{N_0 r} = \frac{P}{2\sigma^2 r}$$

where P is the average transmitted power, and $N_0/2$ is the two-sided power spectral density of the channel noise. Without loss of generality, we may set $P = 1$. We may then express the noise variance as

$$\sigma^2 = \frac{N_0}{2E_b r} \quad (9.111)$$

Substituting Equation (9.111) into (9.110) and rearranging terms, we get the desired relation:

$$\frac{E_b}{N_0} = \frac{1}{2rM^{-1}(r)} \quad (9.112)$$

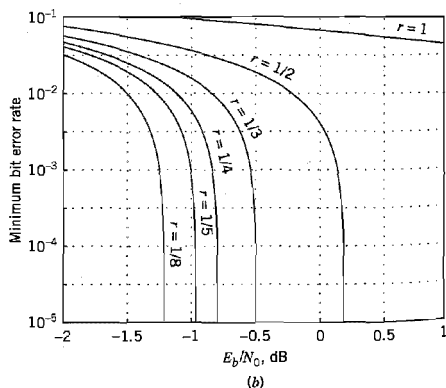
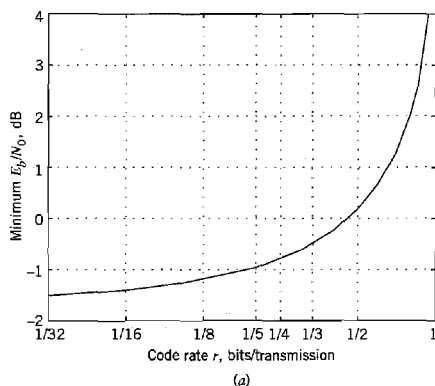


FIGURE 9.18 Binary antipodal signaling over an AWGN channel. (a) Minimum E_b/N_0 versus the code rate r . (b) Minimum bit error rate (BER) versus E_b/N_0 for varying code rate r .

where $M^{-1}(r)$ is the *inverse* of the mutual information between the channel input and output, expressed as a function of the code rate r .

Using the Monte Carlo method to estimate the differential entropy $h(Y)$ and therefore $M^{-1}(r)$, the plots of Figure 9.18 are computed.¹² Figure 9.18a plots the minimum E_b/N_0 versus the code rate r for error-free communication. Figure 9.18b plots the minimum achievable bit error rate versus E_b/N_0 with the code rate r as a running parameter. From Figure 9.18 we may draw the following conclusions:

- ▶ For uncoded binary signaling (i.e., $r = 1$), an infinite E_b/N_0 is required for error-free communication, which agrees with what we know about uncoded data transmission over an AWGN channel.
- ▶ The minimum E_b/N_0 decreases with decreasing code rate r , which is intuitively satisfying. For example, for $r = 1/2$, the minimum value of E_b/N_0 is slightly less than 0.2 dB.
- ▶ As r approaches zero, the minimum E_b/N_0 approaches the limiting value of -1.6 dB, which agrees with the Shannon limit derived earlier; see Equation (9.100). ◀

9.12 Information Capacity of Colored Noise Channel¹³

The information capacity theorem as formulated in Equation (9.95) applies to a band-limited white noise channel. In this section, we extend Shannon's information capacity theorem to the more general case of a *nonwhite*, or *colored*, noise channel. To be specific, consider the channel model shown in Figure 9.19a where the transfer function of the channel is denoted by $H(f)$. The channel noise $n(t)$, which appears additively at the channel output, is modeled as the sample function of a stationary Gaussian process of zero mean and power spectral density $S_N(f)$. The requirement is twofold:

1. Find the input ensemble, described by the power spectral density $S_X(f)$, that maximizes the mutual information between the channel output $y(t)$ and the channel input $x(t)$, subject to the constraint that the average power of $x(t)$ is fixed at a constant value P .
2. Hence, determine the optimum information capacity of the channel.

This problem is a constrained optimization problem. To solve it, we proceed as follows:

- ▶ Because the channel is linear, we may replace the model of Figure 9.19a with the equivalent model shown in Figure 9.19b. From the viewpoint of the spectral characteristics of the signal plus noise measured at the channel output, the two models of Figure 9.19 are equivalent, provided that the power spectral density of the noise

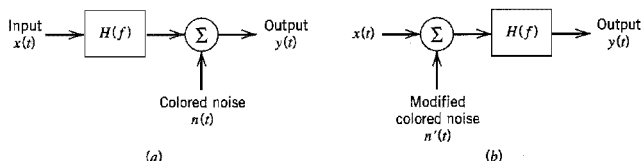


FIGURE 9.19 (a) Model of band-limited, power-limited noisy channel. (b) Equivalent model of the channel.

$n'(t)$ in Figure 9.19b is defined in terms of the power spectral density of the noise $n(t)$ in Figure 9.19a as

$$S_{N'}(f) = \frac{S_N(f)}{|H(f)|^2} \quad (9.113)$$

where $|H(f)|$ is the magnitude response of the channel.

- To simplify the analysis, we use the “principle of divide and conquer” in a manner similar to that described in Section 6.12. Specifically, the channel is divided into a large number of adjoining frequency slots, as illustrated in Figure 9.20. The smaller we make the incremental frequency interval Δf of each subchannel, the better is this approximation.

The net result of these two points is that the original model of Figure 9.19a is replaced by the parallel combination of a finite number of subchannels, N , each of which is corrupted essentially by “band-limited white Gaussian noise.”

The k th subchannel in the approximation to the model of Figure 9.19b is described by

$$y_k(t) = x_k(t) + n_k(t), \quad k = 1, 2, \dots, N \quad (9.14)$$

The average power of the signal component $x_k(t)$ is

$$P_k = S_X(f_k) \Delta f, \quad k = 1, 2, \dots, N \quad (9.115)$$

where $S_X(f_k)$ is the power spectral density of the input signal evaluated at the frequency $f = f_k$. The variance of the noise component $n_k(t)$ is

$$\sigma_k^2 = \frac{S_N(f_k)}{|H(f_k)|^2} \Delta f, \quad k = 1, 2, \dots, N \quad (9.116)$$

where $S_N(f_k)$ and $|H(f_k)|$ are the noise spectral density and the channel’s magnitude response evaluated at the frequency f_k , respectively. The information capacity of the k th subchannel is

$$C_k = \frac{1}{2} \Delta f \log_2 \left(1 + \frac{P_k}{\sigma_k^2} \right), \quad k = 1, 2, \dots, N \quad (9.117)$$

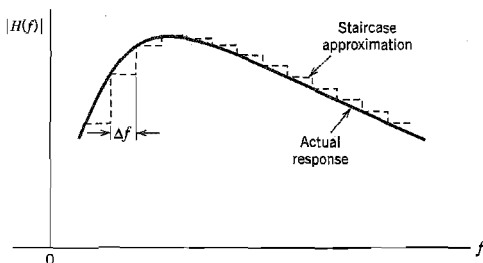


FIGURE 9.20 Staircase approximation of an arbitrary magnitude response $|H(f)|$; only positive-frequency portion of the response is shown.

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

$$\begin{aligned} C &\simeq \sum_{k=1}^N C_k \\ &= \frac{1}{2} \sum_{k=1}^N \Delta f \log_2 \left(1 + \frac{P_k}{\sigma_k^2} \right) \end{aligned} \quad (9.118)$$

The problem we have to address is to maximize the overall information capacity C subject to the constraint:

$$\sum_{k=1}^N P_k = P = \text{constant} \quad (9.119)$$

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

$$J = \frac{1}{2} \sum_{k=1}^N \Delta f \log_2 \left(1 + \frac{P_k}{\sigma_k^2} \right) + \lambda \left(P - \sum_{k=1}^N P_k \right) \quad (9.120)$$

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:

$$P_k + \sigma_k^2 = K \Delta f \quad \text{for } k = 1, 2, \dots, N \quad (9.121)$$

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

$$S_X(f_k) = K - \frac{S_N(f_k)}{|H(f_k)|^2}, \quad k = 1, 2, \dots, N \quad (9.122)$$

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

$$S_X(f) = \begin{cases} K - \frac{S_N(f)}{|H(f)|^2} & \text{for } f \in \mathcal{F}_A \\ 0 & \text{otherwise} \end{cases} \quad (9.123)$$

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

$$P = \int_{f \in \mathcal{F}_A} \left(K - \frac{S_N(f)}{|H(f)|^2} \right) df \quad (9.124)$$

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

$$C \approx \frac{1}{2} \sum_{k=1}^N \Delta f \log_2 \left(K \frac{|H(f_k)|^2}{S_N(f_k)} \right)$$

When the incremental frequency interval Δf is allowed to approach zero, this equation takes the limiting form:

$$C = \frac{1}{2} \int_{-\infty}^{\infty} \log_2 \left(K \frac{|H(f)|^2}{S_N(f)} \right) df \quad (9.125)$$

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

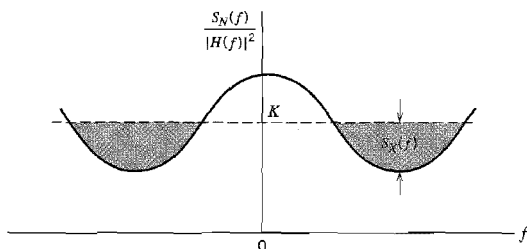


FIGURE 9.21 Water-filling interpretation of information-capacity theorem for a colored noisy channel.

the function $S_X(f)/|H(f)|^2$ is identical to the way in which water distributes itself in a vessel.

Consider now the idealized case of a band-limited signal in additive white Gaussian noise of power spectral density $N(f) = N_0/2$. The transfer function $H(f)$ is that of an ideal band-pass filter defined by

$$H(f) = \begin{cases} 1, & 0 \leq f_c - \frac{B}{2} \leq |f| \leq f_c + \frac{B}{2} \\ 0, & \text{otherwise} \end{cases}$$

where f_c is the midband frequency and B is the channel bandwidth. For this special case, Equations (9.124) and (9.125) reduce to, respectively,

$$P = 2B \left(K - \frac{N_0}{2} \right)$$

and

$$C = B \log_2 \left(\frac{2K}{N_0} \right)$$

Hence, eliminating K between these two equations, we get the standard form of Shannon's capacity theorem, defined by Equation (9.95).

► EXAMPLE 9.13 Capacity of NEXT-Dominated Channel

From the discussion presented in Section 4.8, we recall that a major channel impairment in digital subscriber lines is near-end crosstalk (NEXT). The power spectral density of this crosstalk may be taken as

$$S_N(f) = |H_{\text{NEXT}}(f)|^2 S_X(f) \quad (9.126)$$

where $S_X(f)$ is the power spectral density of the transmitted signal and $H_{\text{NEXT}}(f)$ is the transfer function that couples adjacent twisted pairs. The only constraint we have to satisfy in this example is that the power spectral density function $S_X(f)$ be *nonnegative for all f* . Substituting Equation (9.126) into (9.123), we readily find that this condition is satisfied by solving for K as

$$K = \left(1 + \frac{|H_{\text{NEXT}}(f)|^2}{|H(f)|^2} \right) S_X(f)$$

Finally, using this result in Equation (9.125), we find that the capacity of the NEXT-dominated digital subscriber channel is given by

$$C = \frac{1}{2} \int_{\mathcal{F}_A} \log_2 \left(1 + \frac{|H(f)|^2}{|H_{\text{NEXT}}(f)|^2} \right) df$$

where \mathcal{F}_A is the set of positive and negative frequencies for which $S_X(f) > 0$.

9.13 Rate Distortion Theory

In Section 9.3 we introduced the source coding theorem for a discrete memoryless source, according to which the average code-word length must be at least as large as the source entropy for perfect coding (i.e., perfect representation of the source). However, in many practical situations there are constraints that force the coding to be imperfect, thereby

resulting in unavoidable *distortion*. For example, constraints imposed by a communication channel may place an upper limit on the permissible code rate and therefore average code-word length assigned to the information source. As another example, the information source may have a continuous amplitude as in the case of speech, and the requirement is to quantize the amplitude of each sample generated by the source to permit its representation by a code word of finite length as in pulse-code modulation. In such cases, the problem is referred to as *source coding with a fidelity criterion*, and the branch of information theory that deals with it is called *rate distortion theory*.¹⁴ Rate distortion theory finds applications in two types of situations:

- Source coding where the permitted coding alphabet cannot exactly represent the information source, in which case we are forced to do lossy *data compression*.
- Information transmission at a rate greater than channel capacity.

Accordingly, rate distortion theory may be viewed as a natural extension of Shannon's coding theorems.

■ RATE DISTORTION FUNCTION

Consider a discrete memoryless source defined by an M -ary alphabet $X: \{x_i | i = 1, 2, \dots, M\}$, which consists of a set of statistically independent symbols together with the associated symbol probabilities $\{p_i | i = 1, 2, \dots, M\}$. Let R be the average code rate in bits per code word. The representation code words are taken from another alphabet $Y: \{y_j | j = 1, 2, \dots, N\}$. The source coding theorem states that this second alphabet provides a perfect representation of the source provided that $R > H$, where H is the source entropy. But if we are forced to have $R < H$, then there is unavoidable distortion and therefore loss of information.

Let $p(x_i, y_j)$ denote the joint probability of occurrence of source symbol x_i and representation symbol y_j . From probability theory, we have

$$p(x_i, y_j) = p(y_j | x_i) p(x_i) \quad (9.127)$$

where $p(y_j | x_i)$ is a transition probability. Let $d(x_i, y_j)$ denote a measure of the cost incurred in representing the source symbol x_i by the symbol y_j ; the quantity $d(x_i, y_j)$ is referred to as a *single-letter distortion measure*. The statistical average of $d(x_i, y_j)$ over all possible source symbols and representation symbols is given by

$$\bar{d} = \sum_{i=1}^M \sum_{j=1}^N p(x_i) p(y_j | x_i) d(x_i, y_j) \quad (9.128)$$

Note that the average distortion \bar{d} is a nonnegative continuous function of the transition probabilities $p(y_j | x_i)$ that are determined by the source encoder-decoder pair.

A conditional probability assignment $p(y_j | x_i)$ is said to be *D-admissible* if and only if the average distortion \bar{d} is less than or equal to some acceptable value D . The set of all D -admissible conditional probability assignments is denoted by

$$P_D = \{p(y_j | x_i) : \bar{d} \leq D\} \quad (9.129)$$

For each set of transition probabilities, we have a mutual information

$$I(X; Y) = \sum_{i=1}^M \sum_{j=1}^N p(x_i) p(y_j | x_i) \log \left(\frac{p(y_j | x_i)}{p(y_j)} \right) \quad (9.130)$$

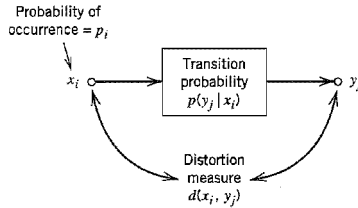


FIGURE 9.22 Summary of rate distortion theory.

A rate distortion function $R(D)$ is defined as the smallest coding rate possible for which the average distortion is guaranteed not to exceed D . Let P_D denote the set to which the conditional probability $p(y_j | x_i)$ belongs for a prescribed D . Then, for a fixed D we write¹⁵

$$R(D) = \min_{p(y_j | x_i) \in P_D} I(X; Y) \quad (9.131)$$

subject to the constraint

$$\sum_{j=1}^N p(y_j | x_i) = 1 \quad \text{for } i = 1, 2, \dots, M \quad (9.132)$$

The rate distortion function $R(D)$ is measured in units of bits if the base-2 logarithm is used in Equation (9.130). Intuitively, we expect the distortion D to decrease as the rate distortion function $R(D)$ is increased. We may say conversely that tolerating a large distortion D permits the use of a smaller rate for coding and/or transmission of information.

Figure 9.22 summarizes the main parameters of rate distortion theory. In particular, given the source symbols $\{x_i\}$ and their probabilities $\{p_i\}$ and given a definition of the single-letter distortion measure $d(x_i, y_j)$, the calculation of the rate distortion function $R(D)$ involves finding the conditional probability assignment $p(y_j | x_i)$ subject to certain constraints imposed on $p(y_j | x_i)$. This is a variational problem, the solution of which is unfortunately not straightforward in general.

EXAMPLE 9.14 Gaussian Source

Consider a discrete-time, memoryless Gaussian source with zero mean and variance σ^2 . Let x denote the value of a sample generated by such a source. Let y denote a quantized version of x that permits a finite representation of it. The squared error distortion

$$d(x, y) = (x - y)^2$$

provides a distortion measure that is widely used for continuous alphabets. The rate distortion function for the Gaussian source with squared error distortion, as described herein, is given by

$$R(D) = \begin{cases} \frac{1}{2} \log \left(\frac{\sigma^2}{D} \right), & 0 \leq D \leq \sigma^2 \\ 0, & D > \sigma^2 \end{cases} \quad (9.133)$$

In this case, we see that $R(D) \rightarrow \infty$ as $D \rightarrow 0$, and $R(D) = 0$ for $D = \sigma^2$.

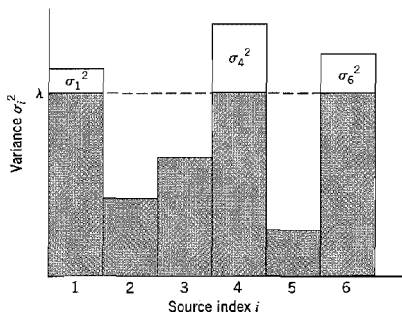


FIGURE 9.23 Reverse water-filling picture for a set of parallel Gaussian processes.

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

$$R(D) = \sum_{i=1}^N \frac{1}{2} \log \left(\frac{\sigma_i^2}{D_i} \right) \quad (9.134)$$

where D_i is itself defined by

$$D_i = \begin{cases} \lambda & \text{if } \lambda < \sigma_i^2 \\ \sigma_i^2 & \text{if } \lambda \geq \sigma_i^2 \end{cases} \quad (9.135)$$

and the constant λ is chosen so as to satisfy the condition

$$\sum_{i=1}^N D_i = D \quad (9.136)$$

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

is a *lossy* operation in the sense that the source entropy is reduced (i.e., information is lost), irrespective of the type of source being considered.

In the case of a discrete source, the reason for using data compression is to encode the source output at a rate smaller than the source entropy. By so doing, the source coding theorem is violated, which means that exact reproduction of the original data is *no longer* possible.

In the case of a continuous source, the entropy is infinite, and therefore a signal compression code must always be used to encode the source output at a finite rate. Consequently, it is impossible to digitally encode an analog signal with a finite number of bits without producing some distortion. This statement is in perfect accord with the idea of pulse-code modulation, which was studied in Chapter 3. There it was shown that quantization, which is basic to the analog-to-digital conversion process in pulse-code modulation, always introduces distortion (known as quantization noise) into the transmitted signal. A quantizer may therefore be viewed as a signal compressor.

The uniform and nonuniform quantizers considered in Chapter 3 are said to be *scalar quantizers* in the sense that they deal with samples of the analog signal (i.e., continuous source output) one at a time. Each sample is converted into a quantized value, with the conversion being independent from sample to sample. A scalar quantizer is a rather simple signal compressor, which makes it attractive for practical use. Yet it can provide a surprisingly good performance; this is especially so if nonuniform quantization is used.

There is another class of quantizers known as *vector quantizers* that use blocks of consecutive samples of the source output to form vectors, each of which is treated as a single entity. The essential operation in a vector quantizer is the quantization of a random vector¹⁶ by encoding it as a binary code word. The vector is encoded by comparing it with a *codebook* consisting of a set of stored reference vectors known as *code vectors* or *patterns*. Each pattern in the codebook is used to represent input vectors that are identified by the encoder to be similar to the particular pattern, subject to the maximization of an appropriate fidelity criterion. The encoding process in a vector quantizer may thus be viewed as a *pattern matching operation*.

Let N be the number of code vectors in the codebook, k be the dimension of each vector (i.e., the number of samples in each pattern), and r be the coded transmission rate in bits per sample. These three parameters are related as follows:

$$r = \frac{\log_2 N}{k} \quad (9.137)$$

Then, assuming that the size of the code book is sufficiently large, the signal-to-quantization noise ratio (SNR) for the vector quantizer is given by

$$10 \log_{10}(\text{SNR}) = 6 \left(\frac{\log_2 N}{k} \right) + C_k \text{ dB} \quad (9.138)$$

where C_k is a constant (expressed in dB) that depends on the dimensions k . According to Equation (9.138), the SNR for a vector quantizer increases approximately at the rate of $6/k$ dB for each doubling of the codebook size. Equivalently, we may state that the SNR increases by 6 dB per unit increase in rate (bits per sample) as in the standard PCM using a uniform scalar quantizer. The advantage of the vector quantizer over the scalar quantizer is that its constant term C_k has a higher value, because the vector quantizer optimally exploits the correlations among the samples constituting a vector. Specifically, the constant C_k increases with the dimension k , approaching the ultimate rate-distortion limit for a

given source of information. However, the improvement in SNR is attained at the cost of increased encoding complexity, which grows exponentially with the dimension k for a specified rate r . Unfortunately, this is the main obstacle to the wide use of vector quantization in practice. Nevertheless, in certain applications, the issue of computational complexity is mitigated by exploiting the capability of VLSI technology to concentrate a highly complex signal processor on a silicon chip. For example, that is precisely what is done in the use of code-excited linear predictive (CELP) modeling of speech in wireless communication systems of the CDMA type, namely, the IS-95 system. From the description of CELP presented in Section 8.9, it is clear that the CELP modeling of speech is an example of vector quantization.

9.15 Summary and Discussion

In this chapter we established four fundamental limits on different aspects of a communication system. The limits are embodied in the source coding theorem, the channel coding theorem, the information capacity theorem, and the rate distortion function.

The *source coding theorem*, Shannon's first theorem, provides the mathematical tool for assessing *data compaction*, that is, *lossless compression* of data generated by a discrete memoryless source. The theorem tells us that we can make the average number of binary code elements (bits) per source symbol as small as, but no smaller than, the entropy of the source measured in bits. The *entropy* of a source is a function of the probabilities of the source symbols that constitute the alphabet of the source. Since entropy is a measure of uncertainty, the entropy is maximum when the associated probability distribution generates maximum uncertainty.

The *channel coding theorem*, Shannon's second theorem, is both the most surprising and the single most important result of information theory. For a *binary symmetric channel*, the channel coding theorem tells us that for any *code rate* r less than or equal to the *channel capacity* C , codes do exist such that the average probability of error is as small as we want it. A binary symmetric channel is the simplest form of a discrete memoryless channel. It is symmetric because the probability of receiving a 1 if a 0 is sent is the same as the probability of receiving a 0 if a 1 is sent. This probability, the probability that an error will occur, is termed a *transition probability*. The transition probability p is determined not only by the additive noise at the channel output but also by the kind of receiver used. The value of p uniquely defines the channel capacity C .

Shannon's third remarkable theorem, the *information capacity theorem*, tells us that there is a maximum to the rate at which any communication system can operate reliably (i.e., free of errors) when the system is constrained in power. This maximum rate is called the *information capacity*, measured in bits per second. When the system operates at a rate greater than the information capacity, it is condemned to a high probability of error, regardless of the choice of signal set used for transmission or the receiver used for processing the received signal.

Finally, the rate distortion function provides the mathematical tool for signal compression (i.e., solving the problem of source coding with a fidelity criterion): The rate distortion function can be applied to a discrete as well as continuous memoryless source.

When the output of a source of information is compressed in a lossless manner, the resulting data stream usually contains redundant bits. These redundant bits can be removed by using a lossless algorithm such as Huffman coding or the Lempel–Ziv algorithm for data compaction. We may thus speak of data compression followed by data compaction as two constituents of the *dissection of source coding*, which is so called because it refers

exclusively to the sources of information. In some source coding applications, we have a third constituent, namely, *data encryption*, which follows data compaction. The purpose of data encryption is to disguise the data (bit) stream in such a way that it has no meaning to an unauthorized receiver. Some basic aspects of *cryptography*, which encompasses both encryption and decryption, follow quite naturally from information theory, as discussed in Appendix 5. Other issues relating to cryptography are also discussed in that appendix.

One last comment is in order. Shannon's information theory, as presented in this chapter, has been entirely in the context of memoryless sources and channels. The theory can be extended to deal with sources and channels with *memory*, in which case a symbol of interest depends on preceding symbols; however, the level of exposition needed to do this is beyond the scope of this book.¹⁷

NOTES AND REFERENCES

1. According to Lucky (1989), the first mention of the term *information theory* by Shannon occurs in a 1945 memorandum entitled "A Mathematical Theory of Cryptography." It is rather curious that the term was never used in the classic 1948 paper by Shannon, which laid down the foundations of information theory. For an introductory treatment of information theory, see Chapter 2 of Lucky (1989) and the paper by Wyner (1981); see also the books of Adámek (1991), Hamming (1980), and Abramson (1963). For more advanced treatments of the subject, see the books of Cover and Thomas (1991), Blahut (1987), McEliece (1977), and Gallager (1968). For a collection of papers on the development of information theory (including the 1948 classic paper by Shannon), see Slepian (1974). For a collection of the papers published by Shannon, see Sloane and Wyner (1993).
2. The use of a logarithmic measure of information was first suggested by Hartley (1928); however, Hartley used logarithms to base 10.
3. In statistical physics, the entropy of a physical system is defined by (Reif, 1967, p. 147)

$$\mathcal{S} = k \log \Omega$$

where k is Boltzmann's constant, Ω is the number of states accessible to the system, and \log denotes the natural logarithm. This entropy has the dimensions of energy because its definition involves the constant k . In particular, it provides a *quantitative measure of the degree of randomness of the system*. Comparing the entropy of statistical physics with that of information theory, we see that they have a similar form. For a detailed discussion of the relation between them, see Pierce (1961, pp. 184–207) and Brillouin (1962).

4. For the original proof of the source coding theorem, see Shannon (1948). A general proof of the source coding theorem is also given in the following books: Viterbi and Omura (1979, pp. 13–19), McEliece (1977, Chapter 3), and Gallager (1968, pp. 38–55). The source coding theorem is also referred to in the literature as the *noiseless coding theorem*, noiseless in the sense that it establishes the condition for error-free encoding to be possible.
5. For proof of the Kraft–McMillan inequality, see Cover and Thomas (1991, pp. 82–84), Blahut (1990, pp. 298–299), and McEliece (1977, pp. 239–240). For a proof of Equation (9.23), see Cover and Thomas (1991, pp. 87–88), Blahut (1990, pp. 300–301), and McEliece (1977, pp. 241–242).
6. The Huffman code is named after its inventor: D. A. Huffman (1952). For a readable account of Huffman coding and its use in data compaction, see Adámek (1991).
7. The original papers on the Lempel–Ziv algorithm are Ziv and Lempel (1977, 1978). For readable descriptions of the Lempel–Ziv algorithm, see Lucky (1989, pp. 118–122), Blahut

(1990, pp. 314–319), and Gitlin, Hayes, and Weinstein (1992, pp. 120–122). For the application of the Lempel–Ziv algorithm to the compaction of English text, see Lucky (1989, pp. 122–128) and the paper by Welch (1984); see also the review paper by Weiss and Shremp (1993).

8. The channel coding theorem is also known as the *noisy coding theorem*. The original proof of the theorem is given in Shannon (1948). A proof of the theorem is also presented in Hamming (1980, Chapters 9 and 10) in sufficient detail so that a general appreciation of relevant results is developed. The second part of the theorem is referred to in the literature as *the converse to the coding theorem*. A proof of this theorem is presented in the following references: Viterbi and Omura (1979, pp. 28–34) and Gallager (1968, pp. 76–82).
9. The quantity

$$\int_{-\infty}^{\infty} f_Y(x) \log_2 \left(\frac{f_X(x)}{f_Y(x)} \right) dx$$

on the left-hand side of Equation (9.70) is called *relative entropy* or the *Kullback–Leibler divergence* between the probability density functions $f_X(x)$ and $f_Y(x)$; see Kullback (1968).

10. Shannon's information capacity theorem is also referred to in the literature as the *Shannon–Hartley law* in recognition of early work by Hartley on information transmission (Hartley, 1928). In particular, Hartley showed that the amount of information that can be transmitted over a given channel is proportional to the product of the channel bandwidth and the time of operation.
11. A lucid exposition of sphere packing is presented in Cover and Thomas (1991, pp. 242–243); see also Wozencraft and Jacobs (1965, pp. 323–341).
12. Parts *a* and *b* of Figure 9.18 follow the corresponding parts of Figure 6.2 in the book by Frey (1998).
13. For a rigorous treatment of the information capacity of a colored noisy channel, see Gallager (1968). The idea of replacing the channel model of Figure 9.19*a* with that of Figure 9.19*b* is discussed in Gitlin, Hayes, and Weinstein (1992).
14. For a complete treatment of rate distortion theory, see the book by Berger (1971); this subject is also treated in somewhat less detail in Cover and Thomas (1991), McEliece (1977), and Gallager (1968).
15. For the derivation of Equation (9.131), see Cover and Thomas (1991, p. 345). An algorithm for computation of the rate distortion function $R(D)$ defined in Equation (9.131) is described in Blahut (1987, pp. 220–221) and Cover and Thomas (1991, pp. 364–367).
16. For the early papers on vector quantization, see Gersho (1979) and Linde, Buzo, and Gray (1980). For a tutorial review of vector quantization, see Gray (1984). Equation (9.138), defining the SNR for a vector quantizer, is discussed in Gersho and Cuperman (1983). For a complete treatment of vector quantization, see the book by Gersho and Gray (1992).
17. For detailed discussion of discrete channels with memory, see Gallager (1968, pp. 97–112) and Ash (1965, pp. 211–229).

PROBLEMS

Entropy

- 9.1 Let p denote the probability of some event. Plot the amount of information gained by the occurrence of this event for $0 \leq p \leq 1$.

- 9.2 A source emits one of four possible symbols during each signaling interval. The symbols occur with the probabilities:

$$p_0 = 0.4$$

$$p_1 = 0.3$$

$$p_2 = 0.2$$

$$p_3 = 0.1$$

Find the amount of information gained by observing the source emitting each of these symbols.

- 9.3 A source emits one of four symbols s_0, s_1, s_2 , and s_3 with probabilities $1/3, 1/6, 1/4$, and $1/4$, respectively. The successive symbols emitted by the source are statistically independent. Calculate the entropy of the source.
- 9.4 Let X represent the outcome of a single roll of a fair die. What is the entropy of X ?
- 9.5 The sample function of a Gaussian process of zero mean and unit variance is uniformly sampled and then applied to a uniform quantizer having the input-output amplitude characteristic shown in Figure P9.5. Calculate the entropy of the quantizer output.

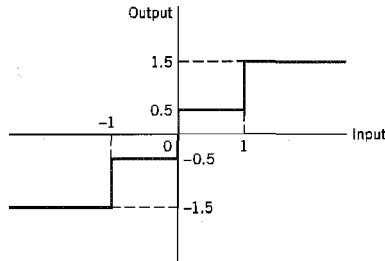


FIGURE P9.5

- 9.6 Consider a discrete memoryless source with source alphabet $\mathcal{S} = \{s_0, s_1, \dots, s_{K-1}\}$ and source statistics $\{p_0, p_1, \dots, p_{K-1}\}$. The n th extension of this source is another discrete memoryless source with source alphabet $\mathcal{S}^n = \{\sigma_0, \sigma_1, \dots, \sigma_{M-1}\}$, where $M = K^n$. Let $P(\sigma_i)$ denote the probability of σ_i .

(a) Show that

$$\sum_{i=0}^{M-1} P(\sigma_i) = 1$$

which is to be expected.

(b) Show that

$$\sum_{i=0}^{M-1} P(\sigma_i) \log_2 \left(\frac{1}{p_{i_k}} \right) = H(\mathcal{S}^n), \quad k = 1, 2, \dots, n$$

where p_{i_k} is the probability of symbol s_{i_k} , and $H(\mathcal{S})$ is the entropy of the original source.

(c) Hence, show that

$$\begin{aligned} H(\mathcal{S}^n) &= \sum_{i=0}^{M-1} P(\sigma_i) \log_2 \frac{1}{P(\sigma_i)} \\ &= nH(\mathcal{S}) \end{aligned}$$

- 9.7 Consider a discrete memoryless source with source alphabet $\mathcal{S} = \{s_0, s_1, s_2\}$ and source statistics $\{0.7, 0.15, 0.15\}$.
- Calculate the entropy of the source.
 - Calculate the entropy of the second-order extension of the source.
- 9.8 It may come as a surprise, but the number of bits needed to store text is much less than that required to store its spoken equivalent. Can you explain the reason for it?

Data Compaction

- 9.9 Consider a discrete memoryless source whose alphabet consists of K equiprobable symbols.
- Explain why the use of a fixed-length code for the representation of such a source is about as efficient as any code can be.
 - What conditions have to be satisfied by K and the code-word length for the coding efficiency to be 100 percent?
- 9.10 Consider the four codes listed below:

Symbol	Code I	Code II	Code III	Code IV
s_0	0	0	0	00
s_1	10	01	01	01
s_2	110	001	011	10
s_3	1110	0010	110	110
s_4	1111	0011	111	111

- Two of these four codes are prefix codes. Identify them, and construct their individual decision trees.
 - Apply the Kraft–McMillan inequality to codes I, II, III, and IV. Discuss your results in light of those obtained in part (a).
- 9.11 Consider a sequence of letters of the English alphabet with their probabilities of occurrence as given here:

Letter	a	i	l	m	n	o	p	y
Probability	0.1	0.1	0.2	0.1	0.1	0.2	0.1	0.1

Compute two different Huffman codes for this alphabet. In one case, move a combined symbol in the coding procedure as high as possible, and in the second case, move it as low as possible. Hence, for each of the two codes, find the average code-word length and the variance of the average code-word length over the ensemble of letters.

- 9.12 A discrete memoryless source has an alphabet of seven symbols whose probabilities of occurrence are as described here:

Symbol	s_0	s_1	s_2	s_3	s_4	s_5	s_6
Probability	0.25	0.25	0.125	0.125	0.125	0.0625	0.0625

Compute the Huffman code for this source, moving a “combined” symbol as high as possible. Explain why the computed source code has an efficiency of 100 percent.

- 9.13 Consider a discrete memoryless source with alphabet $\{s_0, s_1, s_2\}$ and statistics $\{0.7, 0.15, 0.15\}$ for its output.
- Apply the Huffman algorithm to this source. Hence, show that the average code-word length of the Huffman code equals 1.3 bits/symbol.

- (b) Let the source be extended to order two. Apply the Huffman algorithm to the resulting extended source, and show that the average code-word length of the new code equals 1.1975 bits/symbol.
- (c) Compare the average code-word length calculated in part (b) with the entropy of the original source.
- 9.14 Figure P9.14 shows a Huffman tree. What is the code word for each of the symbols A, B, C, D, E, F, and G represented by this Huffman tree? What are their individual code-word lengths?

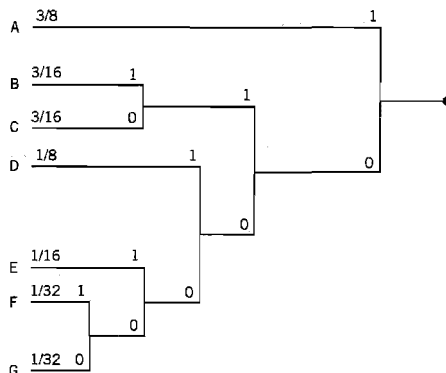


FIGURE P9.14

- 9.15 A computer executes four instructions that are designated by the code words (00, 01, 10, 11). Assuming that the instructions are used independently with probabilities (1/2, 1/8, 1/8, 1/4), calculate the percentage by which the number of bits used for the instructions may be reduced by the use of an optimum source code. Construct a Huffman code to realize the reduction.
- 9.16 Consider the following binary sequence

$$11101001100010110100 \dots$$

Use the Lempel–Ziv algorithm to encode this sequence. Assume that the binary symbols 0 and 1 are already in the codebook.

Binary Symmetric Channel

- 9.17 Consider the transition probability diagram of a binary symmetric channel shown in Figure 9.8. The input binary symbols 0 and 1 occur with equal probability. Find the probabilities of the binary symbols 0 and 1 appearing at the channel output.
- 9.18 Repeat the calculation in Problem 9.17, assuming that the input binary symbols 0 and 1 occur with probabilities 1/4 and 3/4, respectively.

Mutual Information and Channel Capacity

- 9.19 Consider a binary symmetric channel characterized by the transition probability p . Plot the mutual information of the channel as a function of p_1 , the *a priori* probability of symbol 1 at the channel input; do your calculations for the transition probability $p = 0, 0.1, 0.2, 0.3, 0.5$.

- 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.

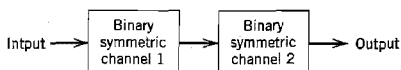


FIGURE P9.22

- 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.

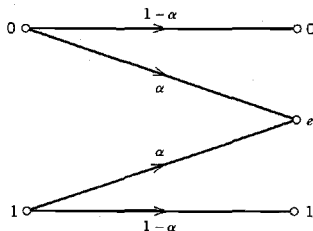


FIGURE P9.23

- 9.24 Consider a digital communication system that uses a *repetition code* for the channel encoding/decoding. In particular, each transmission is repeated n times, where $n = 2m + 1$ is an odd integer. The decoder operates as follows. If in a block of n received bits, the number of 0s exceeds the number of 1s, the decoder decides in favor of a 0. Otherwise, it decides in favor of a 1. An error occurs when $m + 1$ or more transmissions out of $n = 2m + 1$ are incorrect. Assume a binary symmetric channel.

(a) For $n = 3$, show that the average probability of error is given by

$$P_e = 3p^2(1 - p) + p^3$$

where p is the transition probability of the channel.

(b) For $n = 5$, show that the average probability of error is given by

$$P_e = 10p^3(1 - p)^2 + 5p^4(1 - p) + p^5$$

(c) Hence, for the general case, deduce that the average probability of error is given by

$$P_e = \sum_{i=m+1}^n \binom{n}{i} p^i (1 - p)^{n-i}$$

Differential Entropy

- 9.25 Let X_1, X_2, \dots, X_n denote the elements of a Gaussian vector \mathbf{X} . The X_i are independent with mean μ_i and variance σ_i^2 , $i = 1, 2, \dots, n$. Show that the differential entropy of the vector \mathbf{X} equals

$$h(\mathbf{X}) = \frac{n}{2} \log_2 [2\pi e (\sigma_1^2 \sigma_2^2 \dots \sigma_n^2)^{1/n}]$$

What does $h(\mathbf{X})$ reduce to if the variances are equal?

- 9.26 A continuous random variable X is constrained to a peak magnitude M ; that is, $-M < X < M$.

(a) Show that the differential entropy of X is maximum when it is uniformly distributed, as shown by

$$f_X(x) = \begin{cases} 1/2M, & -M < x \leq M \\ 0, & \text{otherwise} \end{cases}$$

(b) Show that the maximum differential entropy of X is $\log_2 2M$.

- 9.27 Prove the properties given in Equations (9.79) to (9.81) for the mutual information $I(X; Y)$.

- 9.28 Consider the continuous random variable Y defined by

$$Y = X + N$$

where X and N are statistically independent. Show that the conditional differential entropy of Y , given X , equals

$$h(Y|X) = h(N)$$

where $h(N)$ is the differential entropy of N .

Information Capacity

- 9.29 A voice-grade channel of the telephone network has a bandwidth of 3.4 kHz.

(a) Calculate the information capacity of the telephone channel for a signal-to-noise ratio of 30 dB.

(b) Calculate the minimum signal-to-noise ratio required to support information transmission through the telephone channel at the rate of 9,600 b/s.

- 9.30 Alphanumeric data are entered into a computer from a remote terminal through a voice-grade telephone channel. The channel has a bandwidth of 3.4 kHz and output signal-to-noise ratio of 20 dB. The terminal has a total of 128 symbols. Assume that the symbols are equiprobable and the successive transmissions are statistically independent.
- Calculate the information capacity of the channel.
 - Calculate the maximum symbol rate for which error-free transmission over the channel is possible.
- 9.31 A black-and-white television picture may be viewed as consisting of approximately 3×10^5 elements, each of which may occupy one of 10 distinct brightness levels with equal probability. Assume that (1) the rate of transmission is 30 picture frames per second, and (2) the signal-to-noise ratio is 30 dB.

Using the information capacity theorem, calculate the minimum bandwidth required to support the transmission of the resulting video signal.

Note: As a matter of interest, commercial television transmissions actually employ a bandwidth of 4.2 MHz, which fits into an allocated bandwidth of 6 MHz.

- 9.32 In this problem, we continue with Example 9.9. Suppose that the tightly packed constellation of Figure 9.15*b* is scaled upward so that the transmitted signal energy per symbol is maintained at the same average value as that consumed by the 64-QAM square constellation of Figure 9.15*a*. Construct the new constellation that results from this scaling. How does the bit error rate of this new constellation compare with that of Figure 9.15*a*? Justify your answer.
- 9.33 The squared magnitude response of a twisted-pair channel can be modeled as

$$|H(f)|^2 = \exp(-\alpha\sqrt{f})$$

The constant α is defined by

$$\alpha = \frac{kl}{l_0}$$

where k is a constant depending on wire gauge, l_0 is a reference line length, and l is the actual length of the twisted pair under study. The squared magnitude response of the coupling responsible for NEXT has the form

$$|H_{\text{NEXT}}(f)|^2 = \beta f^{3/2}$$

where β is a constant that depends on the type of cable used.

Formulate the expression for the information capacity of the NEXT-dominated channel described here.

Data Compression

- 9.34 Equation (9.138) for the signal-to-noise ratio (SNR) of a vector quantizer includes the SNR formula of Equation (3.33) for standard pulse-code modulation as a special case for which $k = 1$. Justify the validity of this inclusion.
- 9.35 All practical data compression and data transmission schemes lie between two limits set by the rate distortion function and the channel capacity theorem. Both of these theorems involve the notion of mutual information, but in different ways. Elaborate on the issues raised by these two statements.

Computer Experiment

- 9.36 In this problem, we revisit Example 9.12, which deals with coded binary antipodal signaling over an additive white Gaussian noise (AWGN) channel. Starting with Equation (9.112) and the underlying theory, develop a software package for computing the minimum E_b/N_0 required for a given bit error rate, where E_b is the signal energy per bit, and

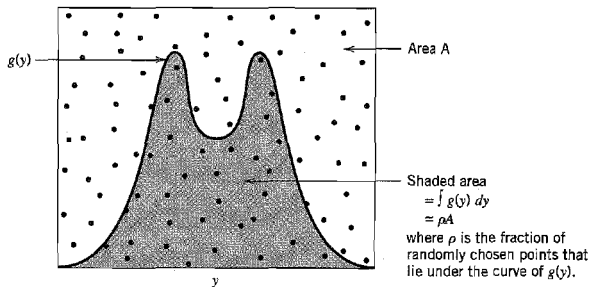


FIGURE P9.36

$N_0/2$ is the noise spectral density. Hence, compute the results plotted in parts *a* and *b* of Figure 9.18.

As mentioned in Example 9.12, the computation of the mutual information between the channel input and channel output is well approximated using Monte Carlo integration. To explain how this method works, consider a function $g(y)$ that is difficult to sample randomly, which is indeed the case for the problem at hand. (For our problem, the function $g(y)$ represents the complicated integrand in the formula for the differential entropy of the channel output.) For the computation, proceed as follows:

- Find an area A that includes the region of interest and that is easily sampled.
- Choose N points, uniformly randomly inside the area A .

Then the *Monte Carlo integration theorem* states that the integral of the function $g(y)$ with respect to y is approximately equal to the area A multiplied by the fraction of points that reside below the curve of g , as illustrated in Figure P9.36. The accuracy of the approximation improves with increasing N .

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 10.1a 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*

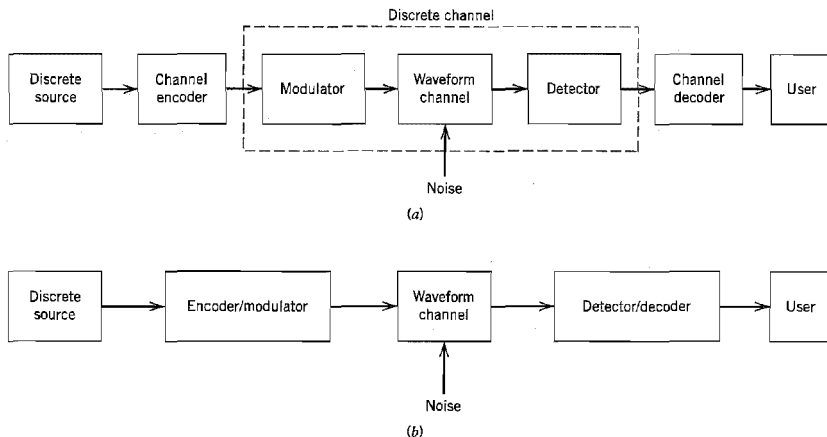


FIGURE 10.1 Simplified models of digital communication system. (a) Coding and modulation performed separately. (b) Coding and modulation combined.

decoder in the receiver exploits the redundancy to decide which message bits were actually transmitted. The combined goal of the channel encoder and decoder is to minimize the effect of channel noise. That is, the number of errors between the channel encoder input (derived from the source) and the channel decoder output (delivered to the user) is minimized.

For a fixed modulation scheme, the addition of redundancy in the coded messages implies the need for increased transmission bandwidth. Moreover, the use of error-control coding adds *complexity* to the system, especially for the implementation of decoding operations in the receiver. Thus, the design trade-offs in the use of error-control coding to achieve acceptable error performance include considerations of bandwidth and system complexity.

There are many different error-correcting codes (with roots in diverse mathematical disciplines) that we can use. Historically, these codes have been classified into *block codes* and *convolutional codes*. The distinguishing feature for this particular classification is the presence or absence of *memory* in the encoders for the two codes.

To generate an (n, k) block code, the channel encoder accepts information in successive k -bit *blocks*; for each block, it adds $n - k$ redundant bits that are algebraically related to the k message bits, thereby producing an overall encoded block of n bits, where $n > k$. The n -bit block is called a *code word*, and n is called the *block length* of the code. The channel encoder produces bits at the rate $R_0 = (n/k)R_s$, where R_s is the bit rate of the information source. The dimensionless ratio $r = k/n$ is called the *code rate*, where $0 < r < 1$. The bit rate R_0 , coming out of the encoder, is called the *channel data rate*. Thus, the code rate is a dimensionless ratio, whereas the data rate produced by the source and the channel data rate are both measured in bits per second.

In a convolutional code, the encoding operation may be viewed as the *discrete-time convolution* of the input sequence with the impulse response of the encoder. The duration of the impulse response equals the memory of the encoder. Accordingly, the encoder for a convolutional code operates on the incoming message sequence, using

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 10.1a, 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 10.1b. 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 use 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 throughput 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.

The three types of ARQ described here offer trade-offs of their own between the need for a half-duplex or full-duplex link and the requirement for efficient use of communication resources. In any event, they all rely on two premises:

- Error detection, which makes the design of the decoder relatively simple.
- Noiseless feedback channel, which is not a severe restriction because the rate of information flow over the feedback channel is typically quite low.

For these reasons, ARQ is widely used in computer-communication systems.

Nevertheless, the fact that FEC requires only one-way links for its operation makes the FEC much wider in application than ARQ. Moreover, the increased decoding complexity of FEC due to the combined need for error detection and correction is no longer a pressing practical issue because the decoder usually lends itself to microprocessor or VLSI implementation in a cost-effective manner.

10.2 Discrete-Memoryless Channels

Returning to the model of Figure 10.1a, the waveform channel is said to be memoryless if the detector output in a given interval depends only on the signal transmitted in that interval, and not on any previous transmission. Under this condition, we may model the combination of the modulator, the waveform channel, and the detector as a *discrete memoryless channel*. Such a channel is completely described by the set of transition probabilities $p(j|i)$, where i denotes a modulator input symbol, j denotes a demodulator output symbol, and $p(j|i)$ denotes the probability of receiving symbol j , given that symbol i was sent. (Discrete memoryless channels were described previously at some length in Section 9.5.)

The simplest discrete memoryless channel results from the use of binary input and binary output symbols. When binary coding is used, the modulator has only the binary symbols 0 and 1 as inputs. Likewise, the decoder has only binary inputs if binary quantization of the demodulator output is used, that is, a *hard decision* is made on the demodulator output as to which symbol was actually transmitted. In this situation, we have a *binary symmetric channel* (BSC) with a *transition probability diagram* as shown in Figure 10.2. The binary symmetric channel, assuming a channel noise modeled as additive white Gaussian noise (AWGN) channel, is completely described by the *transition probability* p . The majority of coded digital communication systems employ binary coding with hard-decision decoding, due to the simplicity of implementation offered by such an approach. *Hard-decision decoders*, or *algebraic decoders*, take advantage of the special algebraic

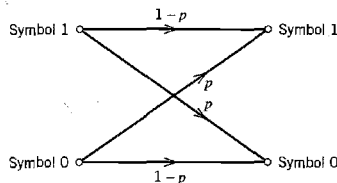


FIGURE 10.2 Transition probability diagram of binary symmetric channel.

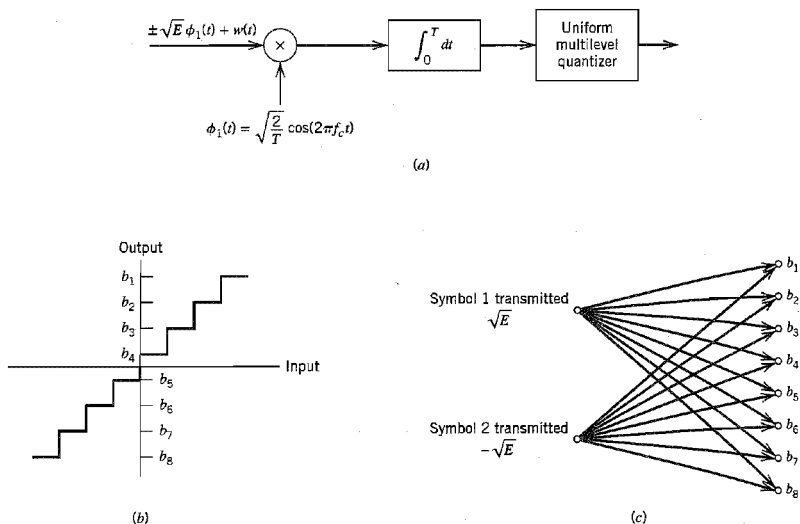


FIGURE 10.3 Binary input Q -ary output discrete memoryless channel. (a) Receiver for binary phase-shift keying. (b) Transfer characteristic of multilevel quantizer. (c) Channel transition probability diagram. Parts (b) and (c) are illustrated for eight levels of quantization.

structure that is built into the design of channel codes to make the decoding relatively easy to perform.

The use of hard decisions prior to decoding causes an irreversible loss of information in the receiver. To reduce this loss, *soft-decision* coding is used. This is achieved by including a multilevel quantizer at the demodulator output, as illustrated in Figure 10.3a for the case of binary PSK signals. The input–output characteristic of the quantizer is shown in Figure 10.3b. The modulator has only the binary symbols 0 and 1 as inputs, but the demodulator output now has an alphabet with Q symbols. Assuming the use of the quantizer as described in Figure 10.3b, we have $Q = 8$. Such a channel is called a *binary input Q -ary output discrete memoryless channel*. The corresponding channel transition probability diagram is shown in Figure 10.3c. The form of this distribution, and consequently the decoder performance, depends on the location of the representation levels of the quantizer, which, in turn, depends on the signal level and noise variance. Accordingly, the demodulator must incorporate automatic gain control if an effective multilevel quantizer is to be realized. Moreover, the use of soft decisions complicates the implementation of the decoder. Nevertheless, soft-decision decoding offers significant improvement in performance over hard-decision decoding by taking a probabilistic rather than an algebraic approach. It is for this reason that soft-decision decoders are also referred to as *probabilistic decoders*.

■ CHANNEL CODING THEOREM REVISITED

In Chapter 9, we established the concept of *channel capacity*, which, for a discrete memoryless channel, represents the maximum amount of information transmitted per

channel use. The *channel coding theorem* states that if a discrete memoryless channel has capacity C and a source generates information at a rate less than C , then there exists a coding technique such that the output of the source may be transmitted over the channel with an arbitrarily low probability of symbol error. For the special case of a binary symmetric channel, the theorem tells us that if the code rate r is less than the channel capacity C , then it is possible to find a code that achieves error-free transmission over the channel. Conversely, it is not possible to find such a code if the code rate r is greater than the channel capacity C .

The channel coding theorem thus specifies the channel capacity C as a *fundamental limit* on the rate at which the transmission of reliable (error-free) messages can take place over a discrete memoryless channel. The issue that matters is not the signal-to-noise ratio, so long as it is large enough, but how the channel input is encoded.

The most unsatisfactory feature of the channel coding theorem, however, is its non-constructive nature. The theorem asserts the existence of good codes but does not tell us how to find them. By *good codes* we mean families of channel codes that are capable of providing reliable transmission of information (i.e., at arbitrarily small probability of symbol error) over a noisy channel of interest at bit rates up to a maximum value less than the capacity of that channel. The error-control coding techniques described in this chapter provide different methods of designing good codes.

■ NOTATION

The codes described in this chapter are *binary codes*, for which the alphabet consists only of symbols 0 and 1. In such a code, the encoding and decoding functions involve the binary arithmetic operations of *modulo-2 addition and multiplication* performed on code words in the code.

Throughout this chapter, we use an ordinary plus sign (+) to denote modulo-2 addition. The use of this terminology will not lead to confusion because the whole chapter relies on binary arithmetic. In so doing, we avoid the use of a special symbol \oplus , as we did in preceding chapters. Thus, according to the notation used in this chapter, the rules for modulo-2 addition are as follows:

$$0 + 0 = 0$$

$$1 + 0 = 1$$

$$0 + 1 = 1$$

$$1 + 1 = 0$$

Because $1 + 1 = 0$, it follows that $1 = -1$. Hence, in binary arithmetic, subtraction is the same as addition. The rules for modulo-2 multiplication are as follows:

$$0 \times 0 = 0$$

$$1 \times 0 = 0$$

$$0 \times 1 = 0$$

$$1 \times 1 = 1$$

Division is trivial in that we have

$$1 \div 1 = 1$$

$$0 \div 1 = 0$$

and division by 0 is not permitted. Modulo-2 addition is the EXCLUSIVE-OR operation in logic, and modulo-2 multiplication is the AND operation.

10.3 Linear Block Codes

A code is said to be *linear* if any two code words in the code can be added in modulo-2 arithmetic to produce a third code word in the code. Consider then an (n, k) linear block code, in which k bits of the n code bits are always identical to the message sequence to be transmitted. The $n - k$ bits in the remaining portion are computed from the message bits in accordance with a prescribed encoding rule that determines the mathematical structure of the code. Accordingly, these $n - k$ bits are referred to as *generalized parity check bits* or simply *parity bits*. Block codes in which the message bits are transmitted in unaltered form are called *systematic codes*. For applications requiring both error detection and error correction, the use of systematic block codes simplifies implementation of the decoder.

Let m_0, m_1, \dots, m_{k-1} constitute a block of k arbitrary message bits. Thus we have 2^k distinct message blocks. Let this sequence of message bits be applied to a linear block encoder, producing an n -bit code word whose elements are denoted by c_0, c_1, \dots, c_{n-1} . Let $b_0, b_1, \dots, b_{n-k-1}$ denote the $(n - k)$ parity bits in the code word. For the code to possess a systematic structure, a code word is divided into two parts, one of which is occupied by the message bits and the other by the parity bits. Clearly, we have the option of sending the message bits of a code word before the parity bits, or vice versa. The former option is illustrated in Figure 10.4, and its use is assumed in the sequel.

According to the representation of Figure 10.4, the $(n - k)$ left-most bits of a code word are identical to the corresponding parity bits, and the k right-most bits of the code word are identical to the corresponding message bits. We may therefore write

$$c_i = \begin{cases} b_i, & i = 0, 1, \dots, n - k - 1 \\ m_{i+k-n}, & i = n - k, n - k + 1, \dots, n - 1 \end{cases} \quad (10.1)$$

The $(n - k)$ parity bits are *linear sums* of the k message bits, as shown by the generalized relation

$$b_i = p_{0i}m_0 + p_{1i}m_1 + \dots + p_{k-1,i}m_{k-1} \quad (10.2)$$

where the coefficients are defined as follows:

$$p_{ij} = \begin{cases} 1 & \text{if } b_i \text{ depends on } m_j \\ 0 & \text{otherwise} \end{cases} \quad (10.3)$$

The coefficients p_{ij} are chosen in such a way that the rows of the generator matrix are linearly independent and the parity equations are *unique*.

The system of Equations (10.1) and (10.2) defines the mathematical structure of the (n, k) linear block code. This system of equations may be rewritten in a compact form

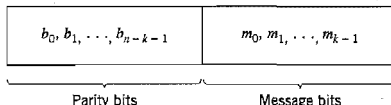


FIGURE 10.4 Structure of systematic code word.

using matrix notation. To proceed with this reformulation, we define the 1-by- k message vector, or information vector, \mathbf{m} , the 1-by- $(n - k)$ parity vector \mathbf{b} , and the 1-by- n code vector \mathbf{c} as follows:

$$\mathbf{m} = [m_0, m_1, \dots, m_{k-1}] \quad (10.4)$$

$$\mathbf{b} = [b_0, b_1, \dots, b_{n-k-1}] \quad (10.5)$$

$$\mathbf{c} = [c_0, c_1, \dots, c_{n-1}] \quad (10.6)$$

Note that all three vectors are *row vectors*. The use of row vectors is adopted in this chapter for the sake of being consistent with the notation commonly used in the coding literature. We may thus rewrite the set of simultaneous equations defining the parity bits in the compact matrix form:

$$\mathbf{b} = \mathbf{mP} \quad (10.7)$$

where \mathbf{P} is the k -by- $(n - k)$ coefficient matrix defined by

$$\mathbf{P} = \begin{bmatrix} p_{00} & p_{01} & \cdots & p_{0,n-k-1} \\ p_{10} & p_{11} & \cdots & p_{1,n-k-1} \\ \vdots & \vdots & & \vdots \\ p_{k-1,0} & p_{k-1,1} & \cdots & p_{k-1,n-k-1} \end{bmatrix} \quad (10.8)$$

where p_{ij} is 0 or 1.

From the definitions given in Equations (10.4)–(10.6), we see that \mathbf{c} may be expressed as a partitioned row vector in terms of the vectors \mathbf{m} and \mathbf{b} as follows:

$$\mathbf{c} = [\mathbf{b} : \mathbf{m}] \quad (10.9)$$

Hence, substituting Equation (10.7) into Equation (10.9) and factoring out the common message vector \mathbf{m} , we get

$$\mathbf{c} = \mathbf{m}[\mathbf{P} : \mathbf{I}_k] \quad (10.10)$$

where \mathbf{I}_k is the k -by- k identity matrix:

$$\mathbf{I}_k = \begin{bmatrix} 1 & 0 & \cdots & 0 \\ 0 & 1 & \cdots & 0 \\ \vdots & \vdots & & \vdots \\ 0 & 0 & \cdots & 1 \end{bmatrix} \quad (10.11)$$

Define the k -by- n generator matrix

$$\mathbf{G} = [\mathbf{P} : \mathbf{I}_k] \quad (10.12)$$

The generator matrix \mathbf{G} of Equation (10.12) is said to be in the *canonical form* in that its k rows are linearly independent; that is, it is not possible to express any row of the matrix \mathbf{G} as a linear combination of the remaining rows. Using the definition of the generator matrix \mathbf{G} , we may simplify Equation (10.10) as

$$\mathbf{c} = \mathbf{mG} \quad (10.13)$$

The full set of code words, referred to simply as *the code*, is generated in accordance with Equation (10.13) by letting the message vector \mathbf{m} range through the set of all 2^k binary k -tuples (1-by- k vectors). Moreover, the sum of any two code words is another

code word. This basic property of linear block codes is called *closure*. To prove its validity, consider a pair of code vectors c_i and c_j corresponding to a pair of message vectors m_i and m_j , respectively. Using Equation (10.13) we may express the sum of c_i and c_j as

$$\begin{aligned} c_i + c_j &= m_i G + m_j G \\ &= (m_i + m_j) G \end{aligned}$$

The modulo-2 sum of m_i and m_j represents a new message vector. Correspondingly, the modulo-2 sum of c_i and c_j represents a new code vector.

There is another way of expressing the relationship between the message bits and parity-check bits of a linear block code. Let H denote an $(n - k)$ -by- n matrix, defined as

$$H = [I_{n-k} : P^T] \quad (10.14)$$

where P^T is an $(n - k)$ -by- k matrix, representing the transpose of the coefficient matrix P , and I_{n-k} is the $(n - k)$ -by- $(n - k)$ identity matrix. Accordingly, we may perform the following multiplication of partitioned matrices:

$$\begin{aligned} HG^T &= [I_{n-k} : P^T] \begin{bmatrix} P^T \\ \vdots \\ I_k \end{bmatrix} \\ &= P^T + P^T \end{aligned}$$

where we have used the fact that multiplication of a rectangular matrix by an identity matrix of compatible dimensions leaves the matrix unchanged. In modulo-2 arithmetic, we have $P^T + P^T = 0$, where 0 denotes an $(n - k)$ -by- k null matrix (i.e., a matrix that has zeros for all of its elements). Hence,

$$HG^T = 0 \quad (10.15)$$

Equivalently, we have $GH^T = 0$, where 0 is a new null matrix. Postmultiplying both sides of Equation (10.13) by H^T , the transpose of H , and then using Equation (10.15), we get

$$\begin{aligned} cH^T &= mGH^T \\ &= 0 \end{aligned} \quad (10.16)$$

The matrix H is called the *parity-check matrix* of the code, and the set of equations specified by Equation (10.16) are called *parity-check equations*.

The generator equation (10.13) and the parity-check detector equation (10.16) are basic to the description and operation of a linear block code. These two equations are depicted in the form of block diagrams in Figure 10.5a and 10.5b, respectively.

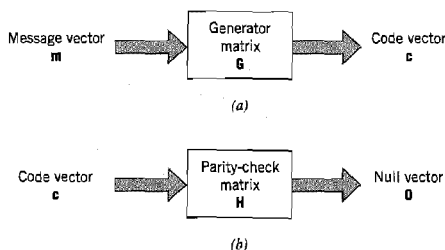


FIGURE 10.5 Block diagram representations of the generator equation (10.13) and the parity-check equation (10.16).

► 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 $I_k = 1$, 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

$$G = [1 \ 1 \ 1 \ 1 \ 1]$$

The transpose of the coefficient matrix P , namely, matrix P^T , consists of a 4-by-1 vector that has 1 for all of its elements. The identity matrix I_{n-k} consists of a 4-by-4 matrix. Hence, the parity-check matrix equals

$$H = \begin{bmatrix} 1 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & 1 \end{bmatrix}$$

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 $HG^T = 0$, 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

$$r = c + e \quad (10.17)$$

The vector e is called the *error vector* or *error pattern*. The i th element of e equals 0 if the corresponding element of r is the same as that of c . On the other hand, the i th 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 i th location. That is, for $i = 1, 2, \dots, n$, we have

$$e_i = \begin{cases} 1 & \text{if an error has occurred in the } i\text{th location} \\ 0 & \text{otherwise} \end{cases} \quad (10.18)$$

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

$$s = rH^T \quad (10.19)$$

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

$$\begin{aligned} s &= (c + e)H^T \\ &= cH^T + eH^T \\ &= eH^T \end{aligned} \quad (10.20)$$

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 c_i , $i = 0, 1, \dots, 2^k - 1$. Correspondingly, for any error pattern e , we define the 2^k distinct vectors e_i as

$$e_i = e + c_i, \quad i = 0, 1, \dots, 2^k - 1 \quad (10.21)$$

The set of vectors $\{e_i, i = 0, 1, \dots, 2^k - 1\}$ 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 H^T , we get

$$\begin{aligned} e_i H^T &= e H^T + c_i H^T \\ &= e H^T \end{aligned} \quad (10.22)$$

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

$$\begin{aligned} s_0 &= e_0 + e_{n-k}p_{00} + e_{n-k+1}p_{10} + \dots + e_{n-1}p_{k-1,0} \\ s_1 &= e_1 + e_{n-k}p_{01} + e_{n-k+1}p_{11} + \dots + e_{n-1}p_{k-1,1} \\ &\vdots \\ s_{n-k-1} &= e_{n-k-1} + e_{n-k}p_{0,n-k-1} + \dots + e_{n-1}p_{k-1,n-k-1} \end{aligned} \quad (10.23)$$

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

pattern e from 2^n to 2^{n-k} possibilities. Given these possibilities, the decoder has the task of making the best selection from the cosets corresponding to s .

■ MINIMUM DISTANCE CONSIDERATIONS

Consider a pair of code vectors c_1 and c_2 that have the same number of elements. The *Hamming distance* $d(c_1, c_2)$ between such a pair of code vectors is defined as the number of locations in which their respective elements differ.

The *Hamming weight* $w(c)$ of a code vector c is defined as the number of nonzero elements in the code vector. Equivalently, we may state that the Hamming weight of a code vector is the distance between the code vector and the all-zero code vector.

The *minimum distance* d_{\min} of a linear block code is defined as the smallest Hamming distance between any pair of code vectors in the code. That is, the minimum distance is the same as the smallest Hamming weight of the difference between any pair of code vectors. From the closure property of linear block codes, the sum (or difference) of two code vectors is another code vector. Accordingly, we may state that *the minimum distance of a linear block code is the smallest Hamming weight of the nonzero code vectors in the code*.

The minimum distance d_{\min} is related to the structure of the parity-check matrix H of the code in a fundamental way. From Equation (10.16) we know that a linear block code is defined by the set of all code vectors for which $cH^T = 0$, where H^T is the transpose of the parity-check matrix H . Let the matrix H be expressed in terms of its columns as follows:

$$H = [h_1, h_2, \dots, h_n] \quad (10.24)$$

Then, for a code vector c to satisfy the condition $cH^T = 0$, the vector c must have 1s in such positions that the corresponding rows of H^T sum to the zero vector 0. However, by definition, the number of 1s in a code vector is the Hamming weight of the code vector. Moreover, the smallest Hamming weight of the nonzero code vectors in a linear block code equals the minimum distance of the code. Hence, *the minimum distance of a linear block code is defined by the minimum number of rows of the matrix H^T whose sum is equal to the zero vector*.

The minimum distance of a linear block code, d_{\min} , is an important parameter of the code. Specifically, it determines the error-correcting capability of the code. Suppose an (n, k) linear block code is required to detect and correct all error patterns (over a binary symmetric channel), and whose Hamming weight is less than or equal to t . That is, if a code vector c_i in the code is transmitted and the received vector is $r = c_i + e$, we require that the decoder output $\hat{c} = c_i$, whenever the error pattern e has a Hamming weight $w(e) \leq t$. We assume that the 2^k code vectors in the code are transmitted with equal probability. The best strategy for the decoder then is to pick the code vector closest to the received vector r , that is, the one for which the Hamming distance $d(c_i, r)$ is the smallest. With such a strategy, the decoder will be able to detect and correct all error patterns of Hamming weight $w(e) \leq t$, provided that the minimum distance of the code is equal to or greater than $2t + 1$. We may demonstrate the validity of this requirement by adopting a geometric interpretation of the problem. In particular, the 1-by- n code vectors and the 1-by- n received vector are represented as points in an n -dimensional space. Suppose that we construct two spheres, each of radius t , around the points that represent code vectors c_i and c_j . Let these two spheres be disjoint, as depicted in Figure 10.6a. For this condition to be satisfied, we require that $d(c_i, c_j) \geq 2t + 1$. If then the code vector c_i is transmitted and the Hamming distance $d(c_i, r) \leq t$, it is clear that the decoder will pick c_i as it is the

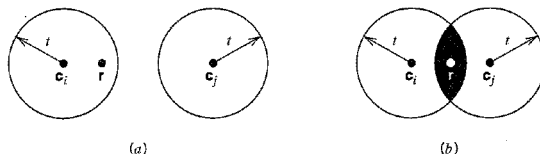


FIGURE 10.6 (a) Hamming distance $d(c_i, c_j) \geq 2t + 1$. (b) Hamming distance $d(c_i, c_j) < 2t$. The received vector is denoted by r .

code vector closest to the received vector r . If, on the other hand, the Hamming distance $d(c_i, c_j) \leq 2t$, the two spheres around c_i and c_j intersect, as depicted in Figure 10.6b. Here we see that if c_i is transmitted, there exists a received vector r such that the Hamming distance $d(c_i, r) \leq t$, and yet r is as close to c_j as it is to c_i . Clearly, there is now the possibility of the decoder picking the vector c_j , which is wrong. We thus conclude that *an (n, k) linear block code has the power to correct all error patterns of weight t or less if, and only if,*

$$d(c_i, c_j) \geq 2t + 1 \quad \text{for all } c_i \text{ and } c_j$$

By definition, however, the smallest distance between any pair of code vectors in a code is the minimum distance of the code, d_{\min} . We may therefore state that *an (n, k) linear block code of minimum distance d_{\min} can correct up to t errors if, and only if,*

$$t \leq \left\lfloor \frac{1}{2}(d_{\min} - 1) \right\rfloor \quad (10.25)$$

where $\lfloor \cdot \rfloor$ denotes the largest integer less than or equal to the enclosed quantity. Equation (10.25) gives the error-correcting capability of a linear block code a quantitative meaning.

■ SYNDROME DECODING

We are now ready to describe a syndrome-based decoding scheme for linear block codes. Let c_1, c_2, \dots, c_{2^k} denote the 2^k code vectors of an (n, k) linear block code. Let r denote the received vector, which may have one of 2^n possible values. The receiver has the task of partitioning the 2^n possible received vectors into 2^k disjoint subsets $\mathcal{D}_1, \mathcal{D}_2, \dots, \mathcal{D}_{2^k}$ in such a way that the i th subset \mathcal{D}_i corresponds to code vector c_i for $1 \leq i \leq 2^k$. The received vector r is decoded into c_i if it is in the i th subset. For the decoding to be correct, r must be in the subset that belongs to the code vector c_i that was actually sent.

The 2^k subsets described herein constitute a *standard array* of the linear block code. To construct it, we may exploit the linear structure of the code by proceeding as follows:

1. The 2^k code vectors are placed in a row with the all-zero code vector c_1 as the left-most element.
2. An error pattern e_2 is picked and placed under c_1 , and a second row is formed by adding e_2 to each of the remaining code vectors in the first row; it is important that the error pattern chosen as the first element in a row not have previously appeared in the standard array.
3. Step 2 is repeated until all the possible error patterns have been accounted for.

Figure 10.7 illustrates the structure of the standard array so constructed. The 2^k columns of this array represent the disjoint subsets $\mathcal{D}_1, \mathcal{D}_2, \dots, \mathcal{D}_{2^k}$. The 2^{n-k} rows of the array

$c_1 = 0$	c_2	c_3	\dots	c_i	\dots	c_{2^k}
e_2	$c_2 + e_2$	$c_3 + e_2$	\dots	$c_i + e_2$	\dots	$c_{2^k} + e_2$
e_3	$c_2 + e_3$	$c_3 + e_3$	\dots	$c_i + e_3$	\dots	$c_{2^k} + e_3$
\vdots	\vdots	\vdots	\vdots	\vdots	\vdots	\vdots
e_j	$c_2 + e_j$	$c_3 + e_j$	\dots	$c_i + e_j$	\dots	$c_{2^k} + e_j$
\vdots	\vdots	\vdots	\vdots	\vdots	\vdots	\vdots
e_{2^n-k}	$c_2 + e_{2^n-k}$	$c_3 + e_{2^n-k}$	\dots	$c_i + e_{2^n-k}$	\dots	$c_{2^k} + e_{2^n-k}$

FIGURE 10.7 Standard array for an (n, k) block code.

represent the cosets of the code, and their first elements e_2, \dots, 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 with 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 $\mathbf{s} = \mathbf{rH}^T$.
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

$$\mathbf{c} = \mathbf{r} + \mathbf{e}_0 \quad (10.26)$$

as the decoded version of the received vector \mathbf{r} .

This procedure is called *syndrome decoding*.

► EXAMPLE 10.2 Hamming Codes⁴

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:

$$\mathbf{G} = \left[\begin{array}{ccc|ccc} 1 & 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 & 1 \end{array} \right]$$

$\underbrace{\hspace{3cm}}_{\mathbf{P}} \quad \underbrace{\hspace{3cm}}_{\mathbf{I}_k}$

TABLE 10.1 Code words of a (7, 4) Hamming code

Message Word	Code Word	Weight of Code Word	Message Word	Code Word	Weight of Code Word
0000	0000000	0	1000	1101000	3
0001	1010001	3	1001	0111001	4
0010	1110010	4	1010	0011010	3
0011	0100011	3	1011	1001011	4
0100	0110100	3	1100	1011100	4
0101	1100101	4	1101	0001101	3
0110	1000110	3	1110	0101110	4
0111	0010111	4	1111	1111111	7

The corresponding parity-check matrix is given by

$$H = \left[\begin{array}{ccc|ccc} 1 & 0 & 0 & 1 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 & 1 & 1 \end{array} \right]$$

I_{n-k} P^T

With $k = 4$, there are $2^k = 16$ distinct message words, which are listed in Table 10.1. For a given message word, the corresponding code word is obtained by using Equation (10.13). Thus, the application of this equation results in the 16 code words listed in Table 10.1.

In Table 10.1, we have also listed the Hamming weights of the individual code words in the (7, 4) Hamming code. Since the smallest of the Hamming weights for the nonzero code words is 3, it follows that the minimum distance of the code is 3. Indeed, Hamming codes have the property that the minimum distance $d_{\min} = 3$, independent of the value assigned to the number of parity bits m .

To illustrate the relation between the minimum distance d_{\min} and the structure of the parity-check matrix H , consider the code word 0110100. In the matrix multiplication defined by Equation (10.16), the nonzero elements of this code word “sift” out the second, third, and fifth columns of the matrix H yielding

$$\begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \\ 1 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$$

We may perform similar calculations for the remaining 14 nonzero code words. We thus find that the smallest number of columns in H that sums to zero is 3, confirming the earlier statement that $d_{\min} = 3$.

An important property of Hamming codes is that they satisfy the condition of Equation (10.25) with the equality sign, assuming that $t = 1$. This means that Hamming codes are *single-error correcting binary perfect codes*.

Assuming single-error patterns, we may formulate the seven coset leaders listed in the right-hand column of Table 10.2. The corresponding 2^3 syndromes, listed in the left-hand column, are calculated in accordance with Equation (10.20). The zero syndrome signifies no transmission errors.

Suppose, for example, the code vector [1110010] is sent, and the received vector is

TABLE 10.2 *Decoding table for the (7, 4) Hamming code defined in Table 10.1*

Syndrome	Error Pattern
000	0000000
100	1000000
010	0100000
001	0010000
110	0001000
011	0000100
111	0000010
101	0000001

[1100010] with an error in the third bit. Using Equation (10.19), the syndrome is calculated to be

$$\begin{aligned}
 \mathbf{s} &= [1100010] \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 1 & 0 \\ 0 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 0 & 1 \end{bmatrix} \\
 &= [0 \ 0 \ 1]
 \end{aligned}$$

From Table 10.2 the corresponding coset leader (i.e., error pattern with the highest probability of occurrence) is found to be [0010000], indicating correctly that the third bit of the received vector is erroneous. Thus, adding this error pattern to the received vector, in accordance with Equation (10.26), yields the correct code vector actually sent. ◀

■ DUAL CODE

Given a linear block code, we may define its *dual* as follows. Taking the transpose of both sides of Equation (10.15), we have

$$\mathbf{GH}^T = \mathbf{0}$$

where \mathbf{H}^T is the transpose of the parity-check matrix of the code, and $\mathbf{0}$ is a new zero matrix. This equation suggests that every (n, k) linear block code with generator matrix \mathbf{G} and parity-check matrix \mathbf{H} has a *dual code* with parameters $(n, n - k)$, generator matrix \mathbf{H} and parity-check matrix \mathbf{G} .

10.4 Cyclic Codes

Cyclic codes form a subclass of linear block codes. Indeed, many of the important linear block codes discovered to date are either cyclic codes or closely related to cyclic codes. An

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 $(c_0, c_1, \dots, c_{n-1})$ denote a code word of an (n, k) linear block code. The code is a cyclic code if the n -tuples

$$\begin{aligned} &(c_{n-1}, c_0, \dots, c_{n-2}), \\ &(c_{n-2}, c_{n-1}, \dots, c_{n-3}), \\ &\vdots \\ &(c_1, c_2, \dots, c_{n-1}, c_0) \end{aligned}$$

are all code words in the code.

To develop the algebraic properties of cyclic codes, we use the elements c_0, c_1, \dots, c_{n-1} of a code word to define the *code polynomial*

$$c(X) = c_0 + c_1X + c_2X^2 + \dots + c_{n-1}X^{n-1} \quad (10.27)$$

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

$$\begin{aligned} X^i c(X) &= X^i(c_0 + c_1X + \dots + c_{n-i-1}X^{n-i-1} + c_{n-i}X^{n-i} \\ &\quad + \dots + c_{n-1}X^{n-1}) \\ &= c_0X^i + c_1X^{i+1} + \dots + c_{n-i-1}X^{n-1} + c_{n-i}X^n \\ &\quad + \dots + c_{n-1}X^{n+i-1} \\ &= c_{n-i}X^n + \dots + c_{n-1}X^{n+i-1} + c_0X^i + c_1X^{i+1} \\ &\quad + \dots + c_{n-i-1}X^{n-1} \end{aligned} \quad (10.28)$$

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:

$$\begin{aligned} X^i c(X) &= c_{n-i} + \dots + c_{n-1}X^{i-1} + c_0X^i + c_1X^{i+1} + \dots + c_{n-i-1}X^{n-1} \\ &\quad + c_{n-i}(X^n + 1) + \dots + c_{n-1}X^{i-1}(X^n + 1) \end{aligned} \quad (10.29)$$

Next, we introduce the following definitions:

$$\begin{aligned} c^{(i)}(X) &= c_{n-i} + \dots + c_{n-1}X^{i-1} + c_0X^i + c_1X^{i+1} \\ &\quad + \dots + c_{n-i-1}X^{n-1} \end{aligned} \quad (10.30)$$

$$q(X) = c_{n-i} + c_{n-i+1}X + \dots + c_{n-1}X^{i-1} \quad (10.31)$$

Accordingly, Equation (10.29) is reformulated in the compact form

$$X^i c(X) = q(X)(X^n + 1) + c^{(i)}(X) \quad (10.32)$$

The polynomial $c^{(i)}(X)$ is recognized as the code polynomial of the code word $(c_{n-i}, \dots, c_{n-1}, c_0, c_1, \dots, c_{n-i-1})$ obtained by applying i cyclic shifts to the code word $(c_0, c_1, \dots, c_{n-i-1}, c_{n-i}, \dots, c_{n-1})$. Moreover, from Equation (10.32) we readily see that $c^{(i)}(X)$ is the remainder that results from dividing $X^i c(X)$ by $(X^n + 1)$. We may thus formally state the cyclic property in polynomial notation as follows: *If $c(X)$ is a code polynomial, then the polynomial*

$$c^{(i)}(X) = X^i c(X) \bmod (X^n + 1) \quad (10.33)$$

*is also a code polynomial for any cyclic shift i ; the term *mod* is the abbreviation for *modulo*. The special form of polynomial multiplication described in Equation (10.33) is referred to as *multiplication modulo $X^n + 1$* . In effect, the multiplication is subject to the constraint $X^n = 1$, the application of which restores the polynomial $X^i c(X)$ to order $n - 1$ for all $i < n$. (Note that in modulo-2 arithmetic, $X^n + 1$ has the same value as $X^n - 1$.)*

■ GENERATOR POLYNOMIAL

The polynomial $X^n + 1$ and its factors play a major role in the generation of cyclic codes. Let $g(X)$ be a polynomial of degree $n - k$ that is a factor of $X^n + 1$; as such, $g(X)$ is *the polynomial of least degree in the code*. In general, $g(X)$ may be expanded as follows:

$$g(X) = 1 + \sum_{i=1}^{n-k-1} g_i X^i + X^{n-k} \quad (10.34)$$

where the coefficient g_i is equal to 0 or 1. According to this expansion, the polynomial $g(X)$ has two terms with coefficient 1 separated by $n - k - 1$ terms. The polynomial $g(X)$ is called the *generator polynomial* of a cyclic code. A cyclic code is uniquely determined by the generator polynomial $g(X)$ in that each code polynomial in the code can be expressed in the form of a polynomial product as follows:

$$c(X) = a(X)g(X) \quad (10.35)$$

where $a(X)$ is a polynomial in X with degree $k - 1$. The $c(X)$ so formed satisfies the condition of Equation (10.33) since $g(X)$ is a factor of $X^n + 1$.

Suppose we are given the generator polynomial $g(X)$ and the requirement is to encode the message sequence $(m_0, m_1, \dots, m_{k-1})$ into an (n, k) *systematic* cyclic code. That is, the message bits are transmitted in unaltered form, as shown by the following structure for a code word (see Figure 10.4):

$$\underbrace{(b_0, b_1, \dots, b_{n-k-1})}_{n-k \text{ parity bits}}, \underbrace{(m_0, m_1, \dots, m_{k-1})}_{k \text{ message bits}}$$

Let the *message polynomial* be defined by

$$m(X) = m_0 + m_1 X + \dots + m_{k-1} X^{k-1} \quad (10.36)$$

and let

$$b(X) = b_0 + b_1 X + \dots + b_{n-k-1} X^{n-k-1} \quad (10.37)$$

According to Equation (10.1), we want the code polynomial to be in the form

$$c(X) = b(X) + X^{n-k}m(X) \quad (10.38)$$

Hence, the use of Equations (10.35) and (10.38) yields

$$a(X)g(X) = b(X) + X^{n-k}m(X)$$

Equivalently, in light of modulo-2 addition, we may write

$$\frac{X^{n-k}m(X)}{g(X)} = a(X) + \frac{b(X)}{g(X)} \quad (10.39)$$

Equation (10.39) states that the polynomial $b(X)$ is the *remainder* left over after dividing $X^{n-k}m(X)$ by $g(X)$.

We may now summarize the steps involved in the encoding procedure for an (n, k) cyclic code assured of a systematic structure. Specifically, we proceed as follows:

1. Multiply the message polynomial $m(X)$ by X^{n-k} .
2. Divide $X^{n-k}m(X)$ by the generator polynomial $g(X)$, obtaining the remainder $b(X)$.
3. Add $b(X)$ to $X^{n-k}m(X)$, obtaining the code polynomial $c(X)$.

■ PARITY-CHECK POLYNOMIAL

An (n, k) cyclic code is uniquely specified by its generator polynomial $g(X)$ of order $(n - k)$. Such a code is also uniquely specified by another polynomial of degree k , which is called the *parity-check polynomial*, defined by

$$h(X) = 1 + \sum_{i=1}^{k-1} h_i X^i + X^k \quad (10.40)$$

where the coefficients h_i are 0 or 1. The parity-check polynomial $h(X)$ has a form similar to the generator polynomial in that there are two terms with coefficient 1, but separated by $k - 1$ terms.

The generator polynomial $g(X)$ is equivalent to the generator matrix G as a description of the code. Correspondingly, the parity-check polynomial, denoted by $h(X)$, is an equivalent representation of the parity-check matrix H . We thus find that the matrix relation $HG^T = 0$ presented in Equation (10.15) for linear block codes corresponds to the relationship

$$g(X)h(X) \bmod (X^n + 1) = 0 \quad (10.41)$$

Accordingly, we may state that *the generator polynomial $g(X)$ and the parity-check polynomial $h(X)$ are factors of the polynomial $X^n + 1$, as shown by*

$$g(X)h(X) = X^n + 1 \quad (10.42)$$

This property provides the basis for selecting the generator or parity-check polynomial of a cyclic code. In particular, we may state that if $g(X)$ is a polynomial of degree $(n - k)$ and it is also a factor of $X^n + 1$, then $g(X)$ is the generator polynomial of an (n, k) cyclic code. Equivalently, we may state that if $h(X)$ is a polynomial of degree k and it is also a factor of $X^n + 1$, then $h(X)$ is the parity-check polynomial of an (n, k) cyclic code.

A final comment is in order. Any factor of $X^n + 1$ with degree $(n - k)$, the number of parity bits, can be used as a generator polynomial. For large values of n , the polynomial $X^n + 1$ may have many factors of degree $n - k$. Some of these polynomial factors generate

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 G of the code by noting that the k polynomials $g(X), Xg(X), \dots, X^{k-1}g(X)$ span the code. Hence, the n -tuples corresponding to these polynomials may be used as rows of the k -by- n generator matrix G .

However, the construction of the parity-check matrix H of the cyclic code from the parity-check polynomial $b(X)$ requires special attention, as described here. Multiplying Equation (10.42) by $a(x)$ and then using Equation (10.35), we obtain

$$c(X)b(X) = a(X) + X^n a(X) \quad (10.43)$$

The polynomials $c(X)$ and $b(X)$ 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 $X^k, X^{k+1}, \dots, X^{n-1}$ do not appear in the polynomial on the right-hand side of Equation (10.43). Thus, setting the coefficients of $X^k, X^{k+1}, \dots, X^{n-1}$ in the expansion of the product polynomial $c(X)b(X)$ equal to zero, we obtain the following set of $n - k$ equations:

$$\sum_{i=j}^{j+k} c_i b_{k+j-i} = 0 \quad \text{for } 0 \leq j \leq n - k - 1 \quad (10.44)$$

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 $b(X)$ 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 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:

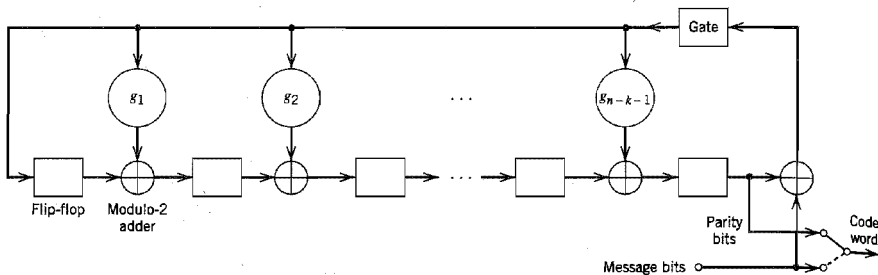
$$\begin{aligned} X^k b(X^{-1}) &= X^k \left(1 + \sum_{i=1}^{k-1} b_i X^{-i} + X^{-k} \right) \\ &= 1 + \sum_{i=1}^{k-1} b_{k-i} X^i + X^k \end{aligned} \quad (10.45)$$

which is also a factor of $X^n + 1$. The n -tuples pertaining to the $(n - k)$ polynomials $X^k b(X^{-1}), X^{k+1} b(X^{-1}), \dots, X^{n-1} b(X^{-1})$ may now be used in rows of the $(n - k)$ -by- n parity-check matrix H .

In general, the generator matrix G and the parity-check matrix 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-

FIGURE 10.8 Encoder for an (n, k) cyclic code.

vision of $X^{n-k}m(X)$ by the generator polynomial $g(X)$ to obtain the remainder $b(X)$, and (3) addition of $b(X)$ to $X^{n-k}m(X)$ to form the desired code polynomial. These three steps can be implemented by means of the encoder shown in Figure 10.8, consisting of a *linear feedback shift register* with $(n - k)$ stages.

The boxes in Figure 10.8 represent *flip-flops*, or *unit-delay elements*. The flip-flop is a device that resides in one of two possible states denoted by 0 and 1. An *external clock* (not shown in Figure 10.8) controls the operation of all the flip-flops. Every time the clock ticks, the contents of the flip-flops (initially set to the state 0) are shifted out in the direction of the arrows. In addition to the flip-flops, the encoder of Figure 10.8 includes a second set of logic elements, namely, *adders*, which compute the modulo-2 sums of their respective inputs. Finally, the *multipliers* multiply their respective inputs by the associated coefficients. In particular, if the coefficient $g_i = 1$, the multiplier is just a direct “connection.” If, on the other hand, the coefficient $g_i = 0$, the multiplier is “no connection.”

The operation of the encoder shown in Figure 10.8 proceeds as follows:

1. The gate is switched on. Hence, the k message bits are shifted into the channel. As soon as the k message bits have entered the shift register, the resulting $(n - k)$ bits in the register form the parity bits [recall that the parity bits are the same as the coefficients of the remainder $b(X)$].
2. The gate is switched off, thereby breaking the feedback connections.
3. The contents of the shift register are read out into the channel.

■ CALCULATION OF THE SYNDROME

Suppose the code word $(c_0, c_1, \dots, c_{n-1})$ is transmitted over a noisy channel, resulting in the received word $(r_0, r_1, \dots, r_{n-1})$. From Section 10.3, we recall that the first step in the decoding of a linear block code is to calculate the syndrome for the received word. If the syndrome is zero, there are no transmission errors in the received word. If, on the other hand, the syndrome is nonzero, the received word contains transmission errors that require correction.

In the case of a cyclic code in systematic form, the syndrome can be calculated easily. Let the received word be represented by a polynomial of degree $n - 1$ or less, as shown by

$$r(X) = r_0 + r_1X + \dots + r_{n-1}X^{n-1} \quad (10.46)$$

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:

$$r(X) = q(X)g(X) + s(X) \quad (10.47)$$

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

$$r(X) = c(X) + e(X) \quad (10.48)$$

where $e(X)$ is the *error polynomial*. Equivalently, we may write

$$e(X) = r(X) + c(X) \quad (10.49)$$

Hence, substituting Equations (10.35) and (10.47) into (10.49), we get

$$e(X) = u(X)g(X) + s(X) \quad (10.50)$$

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

$$Xr(X) = Xq(X)g(X) + Xs(X) \quad (10.51)$$

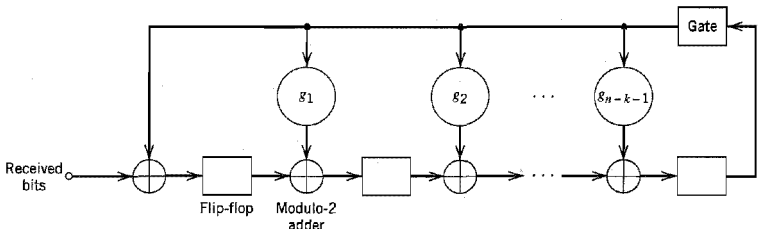


FIGURE 10.9 Syndrome calculator for (n, k) cyclic code.

from which we readily see that $Xs(X)$ is the remainder of the division of $Xr(X)$ by $g(X)$. Hence, the syndrome of $Xr(X)$ is $Xs(X)$ as stated. We may generalize this result by stating that if $s(X)$ is the syndrome of $r(X)$, then $Xs(X)$ is the syndrome of $Xr(X)$.

3. The syndrome polynomial $s(X)$ is identical to the error polynomial $e(X)$, assuming that the errors are confined to the $(n - k)$ parity-check bits of the received word polynomial $r(X)$.

The assumption made here is another way of saying that the degree of the error polynomial $e(X)$ is less than or equal to $(n - k - 1)$. Since the generator polynomial $g(X)$ is of degree $(n - k)$, by definition, it follows that Equation (10.50) can only be satisfied if the quotient $u(X)$ is zero. In other words, the error polynomial $e(X)$ and the syndrome polynomial $s(X)$ are one and the same. The implication of Property 3 is that, under the aforementioned conditions, error correction can be accomplished simply by adding the syndrome polynomial $s(X)$ to the received word polynomial $r(X)$.

► EXAMPLE 10.3 Hamming Codes Revisited

To illustrate the issues relating to the polynomial representation of cyclic codes, we consider the generation of a (7, 4) cyclic code. With the block length $n = 7$, we start by factorizing $X^7 + 1$ into three irreducible polynomials:

$$X^7 + 1 = (1 + X)(1 + X^2 + X^3)(1 + X + X^3)$$

By an “irreducible polynomial” we mean a polynomial that cannot be factored using only polynomials with coefficients from the binary field. An irreducible polynomial of degree m is said to be *primitive* if the smallest positive integer n for which the polynomial divides $X^n + 1$ is $n = 2^m - 1$. For the example at hand, the two polynomials $(1 + X^2 + X^3)$ and $(1 + X + X^3)$ are primitive. Let us take

$$g(X) = 1 + X + X^3$$

as the generator polynomial, whose degree equals the number of parity bits. This means that the parity-check polynomial is given by

$$\begin{aligned} b(X) &= (1 + X)(1 + X^2 + X^3) \\ &= 1 + X + X^2 + X^4 \end{aligned}$$

whose degree equals the number of message bits $k = 4$.

Next, we illustrate the procedure for the construction of a code word by using this generator polynomial to encode the message sequence 1001. The corresponding message polynomial is given by

$$m(X) = 1 + X^3$$

Hence, multiplying $m(X)$ by $X^{n-k} = X^3$, we get

$$X^{n-k}m(X) = X^3 + X^6$$

The second step is to divide $X^{n-k}m(X)$ by $g(X)$, the details of which (for the example at hand) are given below:

$$\begin{array}{r} \overline{X^3 + X} \\ X^3 + X + 1 \overline{)X^6} \\ \underline{X^6} \\ X^4 \\ \underline{X^4} \\ X^2 + X \\ \underline{X^2 + X} \\ 0 \end{array}$$

Note that in this long division we have treated subtraction the same as addition, since we are operating in modulo-2 arithmetic. We may thus write

$$\frac{X^3 + X^6}{1 + X + X^3} = X + X^3 + \frac{X + X^2}{1 + X + X^3}$$

That is, the quotient $a(X)$ and remainder $b(X)$ are as follows, respectively:

$$a(X) = X + X^3$$

$$b(X) = X + X^2$$

Hence, from Equation (10.38) we find that the desired code polynomial is

$$\begin{aligned} c(X) &= b(X) + X^{n-k}m(X) \\ &= X + X^2 + X^3 + X^6 \end{aligned}$$

The code word is therefore 0111001. The four right-most bits, 1001, are the specified message bits. The three left-most bits, 011, are the parity-check bits. The code word thus generated is exactly the same as the corresponding one shown in Table 10.1 for a (7, 4) Hamming code.

We may generalize this result by stating that *any cyclic code generated by a primitive polynomial is a Hamming code of minimum distance 3*.

We next show that the generator polynomial $g(X)$ and the parity-check polynomial $h(X)$ uniquely specify the generator matrix G and the parity-check matrix H , respectively.

To construct the 4-by-7 generator matrix G , we start with four polynomials represented by $g(X)$ and three cyclic-shifted versions of it, as shown by

$$g(X) = 1 + X + X^3$$

$$Xg(X) = X + X^2 + X^4$$

$$X^2g(X) = X^2 + X^3 + X^5$$

$$X^3g(X) = X^3 + X^4 + X^6$$

The polynomials $g(X)$, $Xg(X)$, $X^2g(X)$, and $X^3g(X)$ represent code polynomials in the (7, 4) Hamming code. If the coefficients of these polynomials are used as the elements of the rows of a 4-by-7 matrix, we get the following generator matrix:

$$G' = \begin{bmatrix} 1 & 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 & 1 \end{bmatrix}$$

Clearly, the generator matrix G' so constructed is not in systematic form. We can put it into a systematic form by adding the first row to the third row, and adding the sum of the first two rows to the fourth row. These manipulations result in the desired generator matrix:

$$G = \begin{bmatrix} 1 & 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 & 1 \end{bmatrix}$$

which is exactly the same as that in Example 10.2.

We next show how to construct the 3-by-7 parity-check matrix H from the parity-check polynomial $h(X)$. To do this, we first take the *reciprocal* of $h(X)$, namely, $X^4h(X^{-1})$. For the problem at hand, we form three polynomials represented by $X^4h(X^{-1})$ and two shifted versions of it, as shown by

$$X^4h(X^{-1}) = 1 + X^2 + X^3 + X^4$$

$$X^5h(X^{-1}) = X + X^3 + X^4 + X^5$$

$$X^6h(X^{-1}) = X^2 + X^4 + X^5 + X^6$$

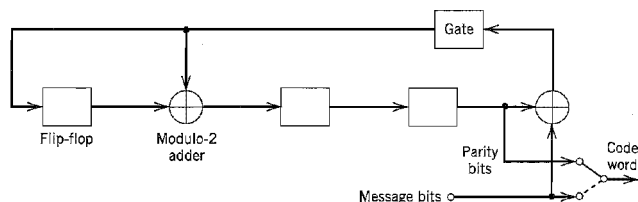


FIGURE 10.10 Encoder for the (7, 4) cyclic code generated by $g(X) = 1 + X + X^3$.

Using the coefficients of these three polynomials as the elements of the rows of the 3-by-7 parity-check matrix, we get

$$H' = \begin{bmatrix} 1 & 0 & 1 & 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 & 1 & 1 \end{bmatrix}$$

Here again we see that the matrix H' is not in systematic form. To put it into a systematic form, we add the third row to the first row to obtain

$$H = \begin{bmatrix} 1 & 0 & 0 & 1 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 & 1 & 1 \end{bmatrix}$$

which is exactly the same as that of Example 10.2.

Figure 10.10 shows the encoder for the (7, 4) cyclic Hamming code generated by the polynomial $g(X) = 1 + X + X^3$. To illustrate the operation of this encoder, consider the message sequence (1001). The contents of the shift register are modified by the incoming message bits as in Table 10.3. After four shifts, the contents of the shift register, and therefore the parity bits, are (011). Accordingly, appending these parity bits to the message bits (1001), we get the code word (0111001); this result is exactly the same as that determined earlier in the example.

Figure 10.11 shows the corresponding syndrome calculator for the (7, 4) Hamming code. Let the transmitted code word be (0111001) and the received word be (0110001); that is, the middle bit is in error. As the received bits are fed into the shift register, initially set to zero, its contents are modified as in Table 10.4. At the end of the seventh shift, the syndrome is identified from the contents of the shift register as 110. Since the syndrome is nonzero, the received word is in error. Moreover, from Table 10.2, we see that the error pattern corresponding to this syndrome is 0001000. This indicates that the error is in the middle bit of the received word, which is indeed the case. ◀

TABLE 10.3 Contents of the shift register in the encoder of Figure 10.10 for message sequence (1001)

Shift	Input	Register Contents
		0 0 0 (initial state)
1	1	1 1 0
2	0	0 1 1
3	0	1 1 1
4	1	0 1 1

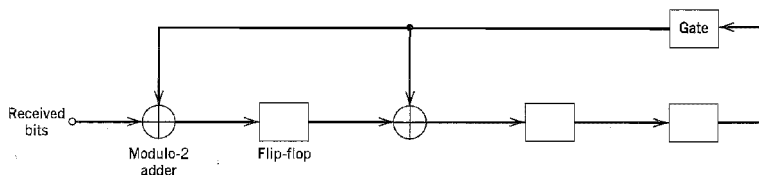


FIGURE 10.11 Syndrome calculator for the (7, 4) cyclic code generated by the polynomial $g(X) = 1 + X + X^3$.

EXAMPLE 10.4 Maximal-Length Codes

For any positive integer $m \geq 3$, there exists a *maximal-length code* with the following parameters:

Block length:	$n = 2^m - 1$
Number of message bits:	$k = m$
Minimum distance:	$d_{\min} = 2^{m-1}$

Maximal-length codes are generated by polynomials of the form

$$g(X) = \frac{1 + X^m}{h(X)} \quad (10.52)$$

where $h(X)$ is any primitive polynomial of degree m . Earlier we stated that any cyclic code generated by a primitive polynomial is a Hamming code of minimum distance 3 (see Example 10.3). It follows therefore that maximal-length codes are the *dual* of Hamming codes.

The polynomial $h(X)$ defines the feedback connections of the encoder. The generator polynomial $g(X)$ defines one period of the maximal-length code, assuming that the encoder is in the initial state 00...01. To illustrate these points, consider the example of a (7, 3) maximal-length code, which is the dual of the (7, 4) Hamming code described in Example 10.3. Thus, choosing

$$h(X) = 1 + X + X^3$$

we find that the generator polynomial of the (7, 3) maximal-length code is

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

TABLE 10.4 Contents of the syndrome calculator in Figure 10.11 for the received word 0110001

Shift	Input Bit	Contents of Shift Register
		0 0 0 (initial state)
1	1	1 0 0
2	0	0 1 0
3	0	0 0 1
4	0	1 1 0
5	1	1 1 1
6	1	0 0 1
7	0	1 1 0

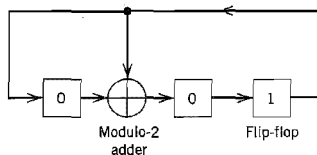


FIGURE 10.12 Encoder for the (7, 3) maximal-length code; the initial state of the encoder is shown in the figure.

Figure 10.12 shows the encoder for the (7, 3) maximal-length code, the feedback connections of which are exactly the same as those shown in Figure 8.2 in Chapter 8. The period of the code is $n = 7$. Thus, assuming that the encoder is in the initial state 001, as indicated in Figure 10.12, we find the output sequence is described by

$$\underbrace{1 \ 0 \ 0}_{\text{initial state}} \quad \underbrace{1 \ 1 \ 1 \ 0 \ 1 \ 0 \ 0}_{g(X) = 1 + X + X^2 + X^4}$$

This result may be readily validated by cycling through the encoder of Figure 10.12.

Note that if we were to choose the other primitive polynomial

$$h(X) = 1 + X^2 + X^3$$

for the (7, 3) maximal-length code, we would simply get the “image” of the code described above, and the output sequence would be “reversed” in time. ◀

■ OTHER CYCLIC CODES

We conclude the discussion of cyclic codes by presenting the characteristics of three other important classes of cyclic codes.

Cyclic Redundancy Check Codes

Cyclic codes are extremely well-suited for *error detection*. We make this statement for two reasons. First, they can be designed to detect many combinations of likely errors. Second, the implementation of both encoding and error-detecting circuits is practical. It is for these reasons that many of the error-detecting codes used in practice are of the cyclic-code type. A cyclic code used for error-detection is referred to as *cyclic redundancy check (CRC) code*.

We define an *error burst* of length B in an n -bit received word as a contiguous sequence of B bits in which the first and last bits or any number of intermediate bits are received in error. Binary (n, k) CRC codes are capable of detecting the following error patterns:

1. All error bursts of length $n - k$ or less.
2. A fraction of error bursts of length equal to $n - k + 1$; the fraction equals $1 - 2^{-(n-k-1)}$.
3. A fraction of error bursts of length greater than $n - k + 1$; the fraction equals $1 - 2^{-(n-k-1)}$.
4. All combinations of $d_{\min} - 1$ (or fewer) errors.
5. All error patterns with an odd number of errors if the generator polynomial $g(X)$ for the code has an even number of nonzero coefficients.

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) Codes⁵

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
31	6	7	11	001	011	011	110	101	000	100
									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.

BCH code. From Table 10.6 we have (111 010 001) for the coefficients of the generator polynomial; hence, we write

$$g(X) = X^8 + X^7 + X^6 + X^4 + 1$$

Reed–Solomon Codes⁶

The Reed–Solomon codes are an important subclass of *nonbinary* BCH codes; they are often abbreviated as RS codes. The encoder for an RS code differs from a binary encoder in that it operates on multiple bits rather than individual bits. Specifically, an RS (n, k) code is used to encode m -bit symbols into blocks consisting of $n = 2^m - 1$ symbols, that is, $m(2^m - 1)$ bits, where $m \geq 1$. Thus, the encoding algorithm expands a block of k symbols to n symbols by adding $n - k$ redundant symbols. When m is an integer power of two, the m -bit symbols are called *bytes*. A popular value of m is 8; indeed, 8-bit RS codes are extremely powerful.

A t -error-correcting RS code has the following parameters:

Block length:	$n = 2^m - 1$ symbols
Message size:	k symbols
Parity-check size:	$n - k = 2t$ symbols
Minimum distance:	$d_{\min} = 2t + 1$ symbols

The block length of the RS code is one less than the size of a code symbol, and the minimum distance is one greater than the number of parity-check symbols. The RS codes make highly efficient use of redundancy, and block lengths and symbol sizes can be adjusted readily to accommodate a wide range of message sizes. Moreover, the RS codes provide a wide range of code rates that can be chosen to optimize performance. Finally, efficient decoding techniques are available for use with RS codes, which is one more reason for their wide application (e.g., compact disc digital audio systems).

10.5 Convolutional Codes⁷

In block coding, the encoder accepts a k -bit message block and generates an n -bit code word. Thus, code words are produced on a block-by-block basis. Clearly, provision must be made in the encoder to buffer an entire message block before generating the associated code word. There are applications, however, where the message bits come in *serially* rather than in large blocks, in which case the use of a buffer may be undesirable. In such situations, the use of *convolutional coding* may be the preferred method. A convolutional coder generates redundant bits by using *modulo-2 convolutions*, hence the name.

The encoder of a binary convolutional code with rate $1/n$, measured in bits per symbol, may be viewed as a *finite-state machine* that consists of an M -stage shift register with prescribed connections to n modulo-2 adders, and a multiplexer that serializes the outputs of the adders. An L -bit message sequence produces a coded output sequence of length $n(L + M)$ bits. The *code rate* is therefore given by

$$r = \frac{L}{n(L + M)} \quad \text{bits/symbol} \quad (10.53)$$

Typically, we have $L \gg M$. Hence, the code rate simplifies to

$$r \approx \frac{1}{n} \quad \text{bits/symbol} \quad (10.54)$$

The *constraint length* of a convolutional code, expressed in terms of message bits, is defined as the number of shifts over which a single message bit can influence the encoder output. In an encoder with an M -stage shift register, the *memory* of the encoder equals M message bits, and $K = M + 1$ shifts are required for a message bit to enter the shift register and finally come out. Hence, the constraint length of the encoder is K .

Figure 10.13a shows a convolutional encoder with $n = 2$ and $K = 3$. Hence, the code rate of this encoder is $1/2$. The encoder of Figure 10.13a operates on the incoming message sequence, one bit at a time.

We may generate a binary convolutional code with rate k/n by using k separate shift registers with prescribed connections to n modulo-2 adders, an input multiplexer and

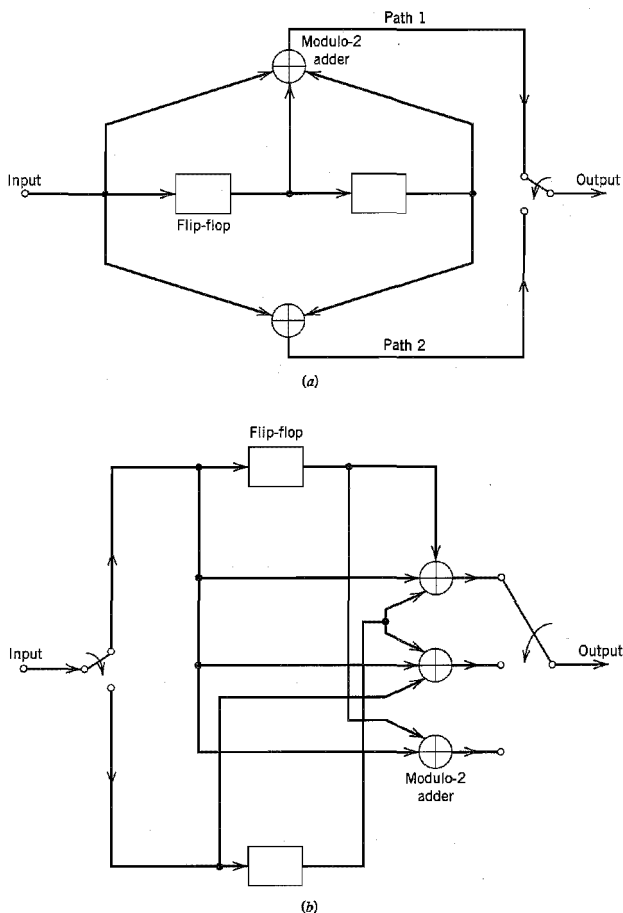


FIGURE 10.13 (a) Constraint length-3, rate- $1/2$ convolutional encoder. (b) Constraint length-2, rate- $2/3$ convolutional encoder.

an output multiplexer. An example of such an encoder is shown in Figure 10.13b, where $k = 2$, $n = 3$, and the two shift registers have $K = 2$ each. The code rate is $2/3$. In this second example, the encoder processes the incoming message sequence two bits at a time.

The convolutional codes generated by the encoders of Figure 10.13 are *nonsystematic* codes. Unlike block coding, the use of nonsystematic codes is ordinarily preferred over systematic codes in convolutional coding.

Each path connecting the output to the input of a convolutional encoder may be characterized in terms of its *impulse response*, defined as the response of that path to a symbol 1 applied to its input, with each flip-flop in the encoder set initially in the zero state. Equivalently, we may characterize each path in terms of a *generator polynomial*, defined as the *unit-delay transform* of the impulse response. To be specific, let the *generator sequence* $(g_0^{(i)}, g_1^{(i)}, g_2^{(i)}, \dots, g_M^{(i)})$ denote the impulse response of the i th path, where the coefficients $g_0^{(i)}, g_1^{(i)}, g_2^{(i)}, \dots, g_M^{(i)}$ equal 0 or 1. Correspondingly, the *generator polynomial* of the i th path is defined by

$$g^{(i)}(D) = g_0^{(i)} + g_1^{(i)}D + g_2^{(i)}D^2 + \dots + g_M^{(i)}D^M \quad (10.55)$$

where D denotes the unit-delay variable. The complete convolutional encoder is described by the set of generator polynomials $\{g^{(1)}(D), g^{(2)}(D), \dots, g^{(n)}(D)\}$. Traditionally, different variables are used for the description of convolutional and cyclic codes, with D being commonly used for convolutional codes and X for cyclic codes.

EXAMPLE 10.5

Consider the convolutional encoder of Figure 10.13a, which has two paths numbered 1 and 2 for convenience of reference. The impulse response of path 1 (i.e., upper path) is (1, 1, 1). Hence, the corresponding generator polynomial is given by

$$g^{(1)}(D) = 1 + D + D^2$$

The impulse response of path 2 (i.e., lower path) is (1, 0, 1). Hence, the corresponding generator polynomial is given by

$$g^{(2)}(D) = 1 + D^2$$

For the message sequence (10011), say, we have the polynomial representation

$$m(D) = 1 + D^3 + D^4$$

As with Fourier transformation, convolution in the time domain is transformed into multiplication in the D -domain. Hence, the output polynomial of path 1 is given by

$$\begin{aligned} c^{(1)}(D) &= g^{(1)}(D)m(D) \\ &= (1 + D + D^2)(1 + D^3 + D^4) \\ &= 1 + D + D^2 + D^3 + D^6 \end{aligned}$$

From this we immediately deduce that the output sequence of path 1 is (111001). Similarly, the output polynomial of path 2 is given by

$$\begin{aligned} c^{(2)}(D) &= g^{(2)}(D)m(D) \\ &= (1 + D^2)(1 + D^3 + D^4) \\ &= 1 + D^2 + D^3 + D^4 + D^5 + D^6 \end{aligned}$$

The output sequence of path 2 is therefore (1011111). Finally, multiplexing the two output sequences of paths 1 and 2, we get the encoded sequence

$$c = (11, 10, 11, 11, 01, 01, 11)$$

Note that the message sequence of length $L = 5$ bits produces an encoded sequence of length $n(L + K - 1) = 14$ bits. Note also that for the shift register to be restored to its zero initial state, a terminating sequence of $K - 1 = 2$ zeros is appended to the last input bit of the message sequence. The terminating sequence of $K - 1$ zeros is called the *tail of the message*. \blacktriangleleft

■ CODE TREE, TRELLIS, AND STATE DIAGRAM

Traditionally, the structural properties of a convolutional encoder are portrayed in graphical form by using any one of three equivalent diagrams: code tree, trellis, and state diagram. We will use the convolutional encoder of Figure 10.13a as a running example to illustrate the insights that each one of these three diagrams can provide.

We begin the discussion with the *code tree* of Figure 10.14. Each branch of the tree represents an input symbol, with the corresponding pair of output binary symbols indicated on the branch. The convention used to distinguish the input binary symbols 0 and 1 is as follows. An input 0 specifies the upper branch of a bifurcation, whereas input 1 specifies the lower branch. A specific *path* in the tree is traced from left to right in accordance with the input (message) sequence. The corresponding coded symbols on the branches of that path constitute the input (message) sequence. Consider, for example, the message sequence (10011) applied to the input of the encoder of Figure 10.13a. Following the procedure just described, we find that the corresponding encoded sequence is (11, 10, 11, 11, 01), which agrees with the first 5 pairs of bits in the encoded sequence $\{c_i\}$ derived in Example 10.5.

From the diagram of Figure 10.14, we observe that the tree becomes *repetitive* after the first three branches. Indeed, beyond the third branch, the two nodes labeled *a* are identical, and so are all the other node pairs that are identically labeled. We may establish this repetitive property of the tree by examining the associated encoder of Figure 10.13a. The encoder has memory $M = K - 1 = 2$ message bits. Hence, when the third message bit enters the encoder, the first message bit is shifted out of the register. Consequently, after the third branch, the message sequences (100 $m_3 m_4 \dots$) and (000 $m_3 m_4 \dots$) generate the same code symbols, and the pair of nodes labeled *a* may be joined together. The same reasoning applies to other nodes. Accordingly, we may collapse the code tree of Figure 10.14 into the new form shown in Figure 10.15, which is called a *trellis*.⁸ It is so called since a trellis is a treelike structure with remerging branches. The convention used in Figure 10.15 to distinguish between input symbols 0 and 1 is as follows. A code branch produced by an input 0 is drawn as a solid line, whereas a code branch produced by an input 1 is drawn as a dashed line. As before, each input (message) sequence corresponds to a specific path through the trellis. For example, we readily see from Figure 10.15 that the message sequence (10011) produces the encoded output sequence (11, 10, 11, 11, 01), which agrees with our previous result.

A trellis is more instructive than a tree in that it brings out explicitly the fact that the associated convolutional encoder is a *finite-state machine*. We define the *state* of a convolutional encoder of rate $1/n$ as the $(K - 1)$ message bits stored in the encoder's shift register. At time j , the portion of the message sequence containing the most recent K bits is written as $(m_{j-K+1}, \dots, m_{j-1}, m_j)$, where m_j is the *current* bit. The $(K - 1)$ -bit state of the encoder at time j is therefore written simply as $(m_{j-1}, \dots, m_{j-K+2}, m_{j-K+1})$. In the

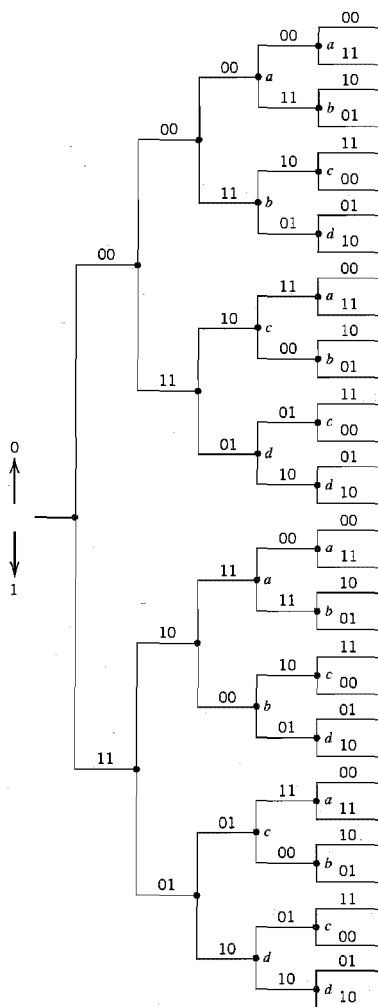


FIGURE 10.14 Code tree for the convolutional encoder of Figure 10.13a.

case of the simple convolutional encoder of Figure 10.13a we have $(K - 1) = 2$. Hence, the state of this encoder can assume any one of four possible values, as described in Table 10.7. The trellis contains $(L + K)$ levels, where L is the length of the incoming message sequence, and K is the constraint length of the code. The levels of the trellis are labeled as $j = 0, 1, \dots, L + K - 1$ in Figure 10.15 for $K = 3$. Level j is also referred to as *depth* j ; both terms are used interchangeably. The first $(K - 1)$ levels correspond to the encoder's departure from the initial state a , and the last $(K - 1)$ levels correspond to the encoder's

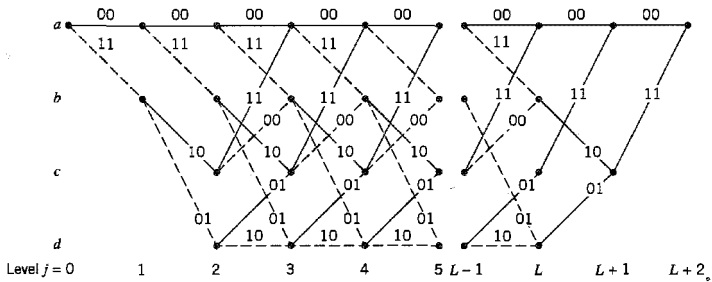


FIGURE 10.15 Trellis for the convolutional encoder of Figure 10.13a.

return to the state a . Clearly, not all the states can be reached in these two portions of the trellis. However, in the central portion of the trellis, for which the level j lies in the range $K - 1 \leq j \leq L$, all the states of the encoder are reachable. Note also that the central portion of the trellis exhibits a fixed periodic structure.

Consider next a portion of the trellis corresponding to times j and $j + 1$. We assume that $j \geq 2$ for the example at hand, so that it is possible for the current state of the encoder to be a , b , c , or d . For convenience of presentation, we have reproduced this portion of the trellis in Figure 10.16a. The left nodes represent the four possible current states of the encoder, whereas the right nodes represent the next states. Clearly, we may coalesce the left and right nodes. By so doing, we obtain the *state diagram* of the encoder, shown in Figure 10.16b. The nodes of the figure represent the four possible states of the encoder, with each node having two incoming branches and two outgoing branches. A transition from one state to another in response to input 0 is represented by a solid branch, whereas a transition in response to input 1 is represented by a dashed branch. The binary label on each branch represents the encoder's output as it moves from one state to another. Suppose, for example, the current state of the encoder is (01) , which is represented by node c . The application of input 1 to the encoder of Figure 10.13a results in the state (10) and the encoded output (00) . Accordingly, with the help of this state diagram, we may readily determine the output of the encoder of Figure 10.13a for any incoming message sequence. We simply start at state a , the all-zero initial state, and walk through the state diagram in accordance with the message sequence. We follow a solid branch if the input is a 0 and a dashed branch if it is a 1. As each branch is traversed, we output the corresponding binary label on the branch. Consider, for example, the message sequence (10011) . For this input we follow the path $abcad$, and therefore output the sequence $(11, 10, 11, 11, 01)$, which

TABLE 10.7 State table for the convolutional encoder of Figure 10.13a

State	Binary Description
a	00
b	10
c	01
d	11

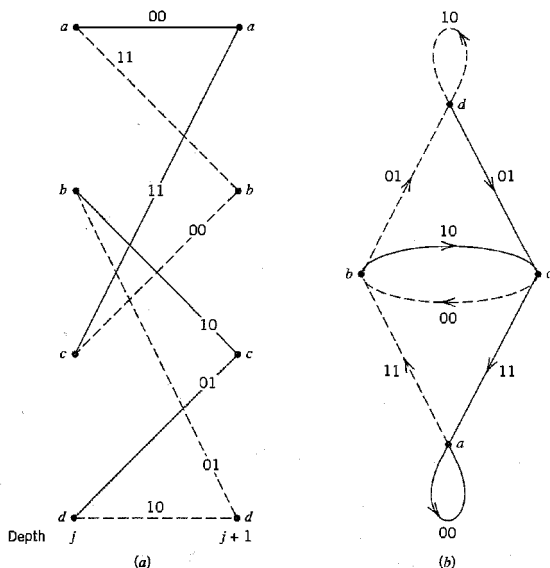


FIGURE 10.16 (a) A portion of the central part of the trellis for the encoder of Figure 10.13a
(b) State diagram of the convolutional encoder of Figure 10.13a.

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:

$$\text{Choose the estimate } \hat{\mathbf{c}} \text{ for which the} \quad (10.56)$$

$$\text{log-likelihood function } \log p(\mathbf{r}|\mathbf{c}) \text{ is maximum.}$$

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 i th elements of \mathbf{c} and \mathbf{r} , respectively. We then have

$$p(\mathbf{r}|\mathbf{c}) = \prod_{i=1}^N p(r_i|c_i) \quad (10.57)$$

Correspondingly, the log-likelihood is

$$\log p(\mathbf{r}|\mathbf{c}) = \sum_{i=1}^N \log p(r_i|c_i) \quad (10.58)$$

Let the transition probability $p(r_i|c_i)$ be defined as

$$p(r_i|c_i) = \begin{cases} p, & \text{if } r_i \neq c_i \\ 1 - p, & \text{if } r_i = c_i \end{cases} \quad (10.59)$$

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

$$\begin{aligned} \log p(\mathbf{r}|\mathbf{c}) &= d \log p + (N - d) \log(1 - p) \\ &= d \log\left(\frac{p}{1 - p}\right) + N \log(1 - p) \end{aligned} \quad (10.60)$$

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:

$$\begin{aligned} &\text{Choose the estimate } \hat{\mathbf{c}} \text{ that minimizes the Hamming distance} \\ &\text{between the received vector } \mathbf{r} \text{ and the transmitted vector } \mathbf{c}. \end{aligned} \quad (10.61)$$

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 ALGORITHM⁹

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

does not continue to grow as the number of incoming message bits increases; rather, it remains constant at 2^{K-1} , where K is the constraint length of the code.

Consider, for example, the trellis diagram of Figure 10.15 for a convolutional code with rate $r = 1/2$ and constraint length $K = 3$. We observe that at level $j = 3$, there are two paths entering any of the four nodes in the trellis. Moreover, these two paths will be identical onward from that point. Clearly, a minimum distance decoder may make a decision at that point as to which of those two paths to retain, without any loss of performance. A similar decision may be made at level $j = 4$, and so on. This sequence of decisions is exactly what the *Viterbi algorithm* does as it walks through the trellis. The algorithm operates by computing a *metric* or discrepancy for every possible path in the trellis. The metric for a particular path is defined as the Hamming distance between the coded sequence represented by that path and the received sequence. Thus, for each node (state) in the trellis of Figure 10.15 the algorithm compares the two paths entering the node. The path with the lower metric is retained, and the other path is discarded. This computation is repeated for every level j of the trellis in the range $M \leq j \leq L$, where $M = K - 1$ is the encoder's memory and L is the length of the incoming message sequence. The paths that are retained by the algorithm are called *survivor* or *active paths*. For a convolutional code of constraint length $K = 3$, for example, no more than $2^{K-1} = 4$ survivor paths and their metrics will ever be stored. This list of 2^{K-1} paths is always guaranteed to contain the maximum-likelihood choice.

A difficulty that may arise in the application of the Viterbi algorithm is the possibility that when the paths entering a state are compared, their metrics are found to be identical. In such a situation, we make the choice by flipping a fair coin (i.e., simply make a guess).

In summary, the Viterbi algorithm is a maximum-likelihood decoder, which is optimum for an AWGN channel. It proceeds in a step-by-step fashion as follows:

Initialization

Label the left-most state of the trellis (i.e., the all-zero state at level 0) as 0, since there is no discrepancy at this point in the computation.

Computation step $j + 1$

Let $j = 0, 1, 2, \dots$, and suppose that at the previous step j we have done two things:

- ▶ All survivor paths are identified.
- ▶ The survivor path and its metric for each state of the trellis are stored.

Then, at level (clock time) $j + 1$, compute the metric for all the paths entering each state of the trellis by adding the metric of the incoming branches to the metric of the connecting survivor path from level j . Hence, for each state, identify the path with the lowest metric as the survivor of step $j + 1$, thereby updating the computation.

Final Step

Continue the computation until the algorithm completes its forward search through the trellis and therefore reaches the termination node (i.e., all-zero state), at which time it makes a decision on the maximum likelihood path. Then, like a block decoder, the sequence of symbols associated with that path is released to the destination as the decoded version of the received sequence. In this sense, it is therefore more correct to refer to the Viterbi algorithm as a *maximum likelihood sequence estimator*.

However, when the received sequence is very long (near infinite), the storage requirement of the Viterbi algorithm becomes too high, and some compromises must be made.

The approach usually taken is to “truncate” the path memory of the decoder as described here. A *decoding window* of length ℓ is specified, and the algorithm operates on a corresponding frame of the received sequence, always stopping after ℓ steps. A decision is then made on the “best” path and the symbol associated with the first branch on that path is released to the user. The symbol associated with the last branch of the path is dropped. Next, the decoding window is moved forward one time interval, and a decision on the next code frame is made, and so on. The decoding decisions made in this way are no longer truly maximum likelihood, but they can be made almost as good provided that the decoding window is long enough. Experience and analysis have shown that satisfactory results are obtained if the decoding window length ℓ is on the order of 5 times the constraint length K of the convolutional code or more.

► EXAMPLE 10.6 Correct Decoding of Received All-Zero Sequence

Suppose that the encoder of Figure 10.13a generates an all-zero sequence that is sent over a binary symmetric channel, and that the received sequence is (0100010000 . . .). There are two errors in the received sequence due to noise in the channel: one in the second bit and the other in the sixth bit. We wish to show that this double-error pattern is correctable through the application of the Viterbi decoding algorithm.

In Figure 10.17, we show the results of applying the algorithm for level $j = 1, 2, 3, 4, 5$. We see that for $j = 2$ there are (for the first time) four paths, one for each of the four states of the encoder. The figure also includes the metric of each path for each level in the computation.

In the left side of Figure 10.17, for $j = 3$ we show the paths entering each of the states, together with their individual metrics. In the right side of the figure, we show the four survivors that result from application of the algorithm for level $j = 3, 4, 5$.

Examining the four survivors in Figure 10.17 for $j = 5$, we see that the all-zero path has the smallest metric and will remain the path of smallest metric from this point forward. This clearly shows that the all-zero sequence is the maximum likelihood choice of the Viterbi decoding algorithm, which agrees exactly with the transmitted sequence. ◀

► EXAMPLE 10.7 Incorrect Decoding of Received All-Zero Sequence

Suppose next that the received sequence is (1100010000 . . .), which contains three errors compared to the transmitted all-zero sequence.

In Figure 10.18, we show the results of applying the Viterbi decoding algorithm for $j = 1, 2, 3, 4$. We see that in this example the correct path has been eliminated by level $j = 3$. Clearly, a triple-error pattern is uncorrectable by the Viterbi algorithm when applied to a convolutional code of rate $1/2$ and constraint length $K = 3$. The exception to this rule is a triple-error pattern spread over a time span longer than one constraint length, in which case it is very likely to be correctable. ◀

■ FREE DISTANCE OF A CONVOLUTIONAL CODE

The performance of a convolutional code depends not only on the decoding algorithm used but also on the distance properties of the code. In this context, the most important single measure of a convolutional code's ability to combat channel noise is the free distance, denoted by d_{free} . The *free distance* of a convolutional code is defined as the *minimum Hamming distance between any two code words in the code*. A convolutional code with free distance d_{free} can correct t errors if and only if d_{free} is greater than $2t$.

The free distance can be obtained quite simply from the state diagram of the convolutional encoder. Consider, for example, Figure 10.16b, which shows the state diagram

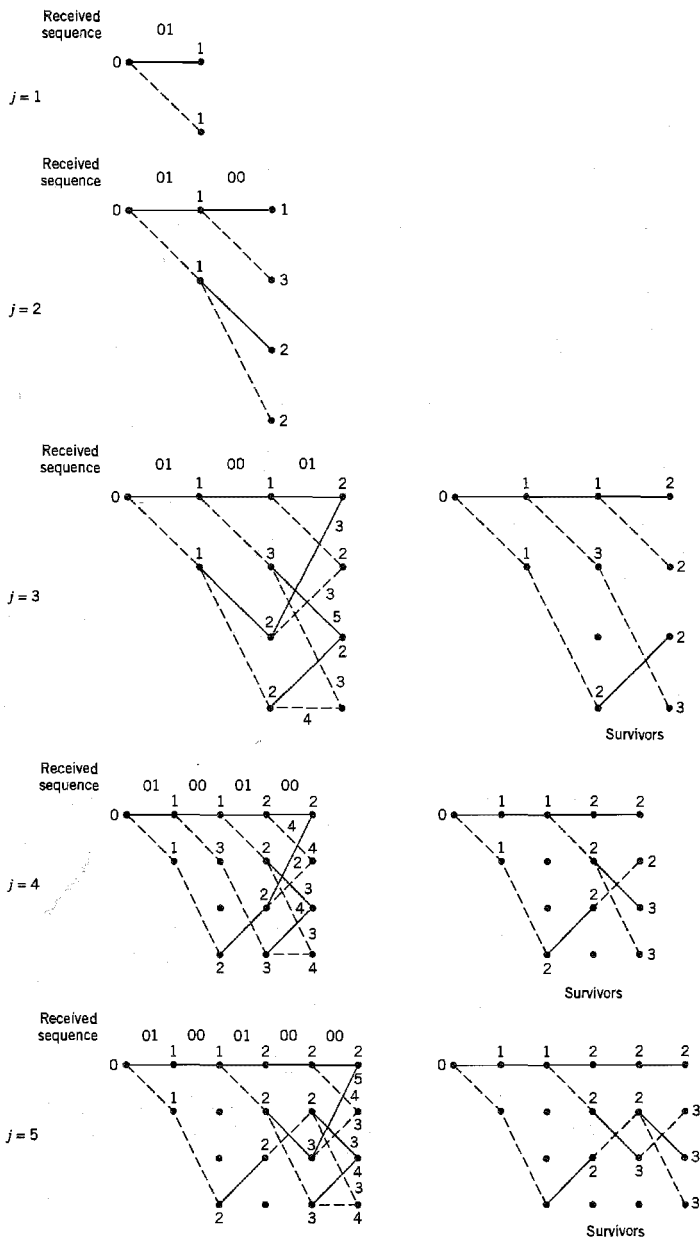


FIGURE 10.17 Illustrating steps in the Viterbi algorithm for Example 10.6.

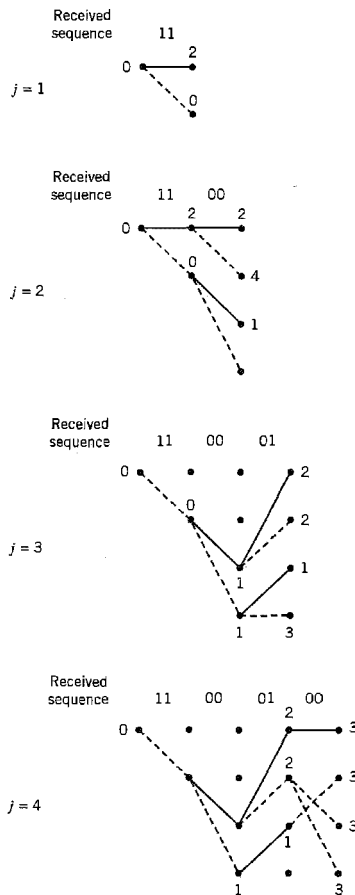


FIGURE 10.18 Illustrating breakdown of the Viterbi algorithm in Example 10.7.

of the encoder of Figure 10.13a. Any nonzero code sequence corresponds to a complete path beginning and ending at the 00 state (i.e., node a). We thus find it useful to split this node in the manner shown in the modified state diagram of Figure 10.19, which may be viewed as a *signal-flow graph* with a single input and a single output. A signal-flow graph consists of *nodes* and directed *branches*; it operates by the following rules:

1. A branch multiplies the signal at its input node by the *transmittance* characterizing that branch.
2. A node with incoming branches *sums* the signals produced by all of those branches.
3. The signal at a node is applied equally to all the branches outgoing from that node.
4. The *transfer function* of the graph is the ratio of the output signal to the input signal.

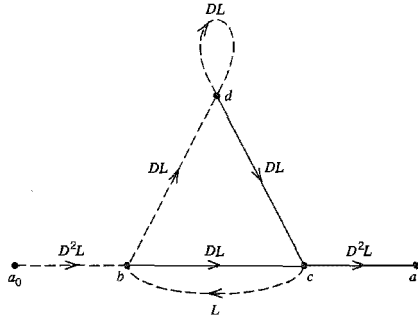


FIGURE 10.19 Modified state diagram of convolutional encoder.

Returning to the signal-flow graph of Figure 10.19, we note that the exponent of D on a branch in this graph describes the Hamming weight of the encoder output corresponding to that branch. The exponent of L is always equal to one, since the length of each branch is one. Let $T(D, L)$ denote the transfer function of the signal-flow graph, with D and L playing the role of dummy variables. For the example of Figure 10.19, we may readily use rules 1, 2, and 3 to obtain the following input-output relations:

$$\left. \begin{aligned} b &= D^2La_0 + Lc \\ c &= DLb + DLd \\ d &= DLb + DLd \\ a_1 &= D^2Lc \end{aligned} \right\} \quad (10.62)$$

where a_0 , b , c , d , and a_1 denote the node signals of the graph. Solving the set of Equations (10.62) for the ratio a_1/a_0 , we find that the transfer function of the graph in Figure 10.19 is given by

$$T(D, L) = \frac{D^5L^3}{1 - DL(1 + L)} \quad (10.63)$$

Using the binomial expansion, we may equivalently write

$$T(D, L) = D^5L^3 \sum_{i=0}^{\infty} (DL(1 + L))^i \quad (10.64)$$

Setting $L = 1$ in Equation (10.64), we thus get the *distance transfer function* expressed in the form of a power series:

$$T(D, 1) = D^5 + 2D^6 + 4D^7 + \dots \quad (10.65)$$

Since the free distance is the minimum Hamming distance between any two code words in the code and the distance transfer function $T(D, 1)$ enumerates the number of code words that are a given distance apart, it follows that the exponent of the first term in the expansion of $T(D, 1)$ defines the free distance. Thus, on the basis of Equation (10.65), the convolutional code of Figure 10.13a has a free distance $d_{\text{free}} = 5$.

This result indicates that up to two errors in the received sequence are correctable, for two or fewer transmission errors will cause the received sequence to be at most at a Hamming distance of 2 from the transmitted sequence but at least at a Hamming distance of 3 from any other code sequence in the code. In other words, in spite of the presence of

TABLE 10.8 Maximum free distances attainable with systematic and nonsystematic convolutional codes of rate 1/2

Constraint Length K	Systematic	Nonsystematic
2	3	3
3	4	5
4	4	6
5	5	7
6	6	8
7	6	10
8	7	10

any pair of transmission errors, the received sequence remains closer to the transmitted sequence than any other possible code sequence. However, this statement is no longer true if there are three or more *closely spaced* transmission errors in the received sequence. These observations confirm the results reported earlier in Examples 10.6 and 10.7.

In using the distance transfer function $T(D, 1)$ to calculate the free distance of a convolutional code, it is assumed that the power series in the unit-delay variable D representing $T(D, 1)$ is *convergent* (i.e., its sum has a “finite” value). This assumption is required to justify the expansion given in Equation (10.65) for the convolutional code of Figure 10.13a. However, there is no guarantee that $T(D, 1)$ is always convergent. When $T(D, 1)$ is nonconvergent, an infinite number of decoding errors are caused by a finite number of transmission errors; the convolutional code is then subject to catastrophic error propagation, and the code is called a *catastrophic code*.¹⁰ In this context it is noteworthy that a *systematic* convolutional code cannot be catastrophic. Unfortunately, for a prescribed constraint length K , the free distances that can be attained with systematic convolutional codes using schemes such as those shown in Figure 10.13 are usually smaller than for the case of nonsystematic convolutional codes, as indicated in Table 10.8.

■ ASYMPTOTIC CODING GAIN¹¹

The transfer function of the encoder state diagram, modified in a manner similar to that illustrated in Figure 10.19, may be used to evaluate a *bound on the bit error rate* for a given decoding scheme; details of this evaluation are, however, beyond the scope of our present discussion. Here we simply summarize the results for two special channels, namely, the binary symmetric channel and the binary-input additive white Gaussian noise (AWGN) channel, assuming the use of binary phase-shift keying (PSK) with coherent detection.

1. Binary symmetric channel. The binary symmetric channel may be modeled as an additive white Gaussian noise channel with binary phase-shift keying (PSK) as the modulation and with hard-decision demodulation. The transition probability p of the binary symmetric channel is then equal to the bit error rate (BER) for the uncoded binary PSK system. From Chapter 6 we recall that for large values of E_b/N_0 , the ratio of signal energy per bit-to-noise power spectral density, the bit error rate for binary PSK without coding is dominated by the exponential factor $\exp(-E_b/N_0)$. On the other hand, the bit error rate for the same modulation scheme with convolutional coding is dominated by the exponential

factor $\exp(-d_{\text{free}} r E_b / 2N_0)$, where r is the code rate and d_{free} is the free distance of the convolutional code. Therefore, as a figure of merit for measuring the improvement in error performance made by the use of coding with hard-decision decoding, we may use the exponents to define the *asymptotic coding gain* (in decibels) as follows:

$$G_a = 10 \log_{10} \left(\frac{d_{\text{free}} r}{2} \right) \text{ dB} \quad (10.66)$$

2. Binary-input AWGN channel. Consider next the case of a memoryless binary-input AWGN channel with no output quantization [i.e., the output amplitude lies in the interval $(-\infty, \infty)$]. For this channel, theory shows that for large values of E_b/N_0 the bit error rate for binary PSK with convolutional coding is dominated by the exponential factor $\exp(-d_{\text{free}} r E_b / N_0)$, where the parameters are as previously defined. Accordingly, in this case, we find that the asymptotic coding gain is defined by

$$G_a = 10 \log_{10}(d_{\text{free}} r) \text{ dB} \quad (10.67)$$

From Equations (10.66) and (10.67) we see that the asymptotic coding gain for the binary-input AWGN channel is greater than that for the binary symmetric channel by 3 dB. In other words, for large E_b/N_0 , the transmitter for a binary symmetric channel must generate an additional 3 dB of signal energy (or power) over that for a binary-input AWGN channel if we are to achieve the same error performance. Clearly, there is an advantage to be gained by permitting an unquantized demodulator output instead of making hard decisions. This improvement in performance, however, is attained at the cost of increased decoder complexity due to the requirement for accepting analog inputs.

The asymptotic coding gain for a binary-input AWGN channel is approximated to within about 0.25 dB by a binary input Q -ary output discrete memoryless channel with the number of representation levels $Q = 8$. This means that we may avoid the need for an analog decoder by using a soft-decision decoder that performs finite output quantization (typically, $Q = 8$), and yet realize a performance close to the optimum.

10.7 Trellis-Coded Modulation¹²

In the traditional approach to channel coding described in the preceding sections of the chapter, encoding is performed separately from modulation in the transmitter; likewise for decoding and detection in the receiver. Moreover, error control is provided by transmitting additional redundant bits in the code, which has the effect of lowering the information bit rate per channel bandwidth. That is, bandwidth efficiency is traded for increased power efficiency.

To attain a more effective utilization of the available bandwidth and power, coding and modulation have to be treated as a single entity. We may deal with this new situation by redefining coding as *the process of imposing certain patterns on the transmitted signal*. Indeed, this definition includes the traditional idea of parity coding.

Trellis codes for band-limited channels result from the treatment of modulation and coding as a *combined* entity rather than as two separate operations. The combination itself is referred to as *trellis-coded modulation* (TCM). This form of signaling has three basic features:

1. The number of signal points in the constellation used is larger than what is required for the modulation format of interest with the same data rate; the additional points allow redundancy for forward error-control coding without sacrificing bandwidth.

2. Convolutional coding is used to introduce a certain dependency between successive signal points, such that only certain *patterns* or *sequences of signal points* are permitted.
3. Soft-decision decoding is performed in the receiver, in which the permissible sequence of signals is modeled as a trellis structure; hence, the name “trellis codes.”

This latter requirement is the result of using an enlarged signal constellation. By increasing the size of the constellation, the probability of symbol error increases for a fixed signal-to-noise ratio. Hence, with hard-decision demodulation we would face a performance loss before we begin. Performing soft-decision decoding on the combined code and modulation trellis ameliorates this problem.

In the presence of AWGN, maximum likelihood decoding of trellis codes consists of finding that particular path through the trellis with *minimum squared Euclidean distance* to the received sequence. Thus, in the design of trellis codes, the emphasis is on maximizing the Euclidean distance between code vectors (or, equivalently, code words) rather than maximizing the Hamming distance of an error-correcting code. The reason for this approach is that, except for conventional coding with binary PSK and QPSK, maximizing the Hamming distance is not the same as maximizing the squared Euclidean distance. Accordingly, in what follows, the Euclidean distance is adopted as the distance measure of interest. Moreover, while a more general treatment is possible, the discussion is (by choice) confined to the case of *two-dimensional constellations of signal points*. The implication of such a choice is to restrict the development of trellis codes to multilevel amplitude and/or phase modulation schemes such as M -ary PSK and M -ary QAM.

The approach used to design this type of trellis codes involves partitioning an M -ary constellation of interest successively into 2, 4, 8, . . . subsets with size $M/2$, $M/4$, $M/8$, . . . , and having progressively larger increasing minimum Euclidean distance between their respective signal points. Such a design approach by *set partitioning* represents the “key idea” in the construction of efficient coded modulation techniques for band-limited channels.

In Figure 10.20, we illustrate the partitioning procedure by considering a circular constellation that corresponds to 8-PSK. The figure depicts the constellation itself and the 2 and 4 subsets resulting from two levels of partitioning. These subsets share the common

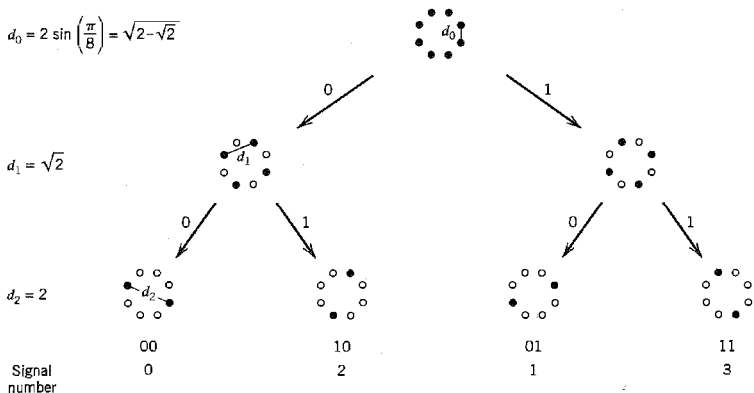


FIGURE 10.20 Partitioning of 8-PSK constellation, which shows that $d_0 < d_1 < d_2$.

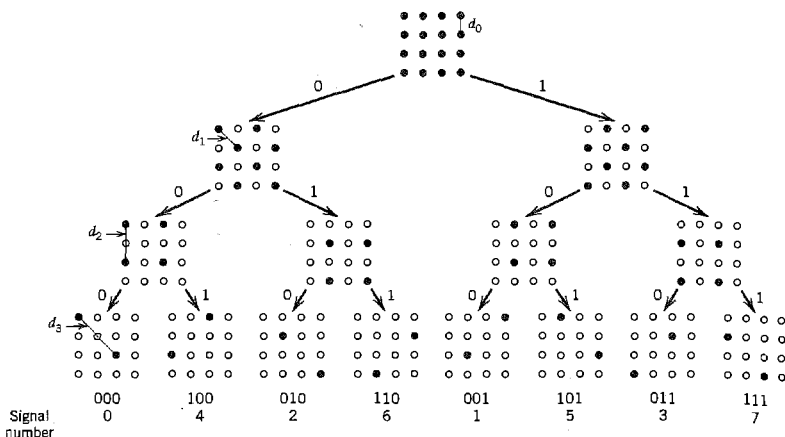


FIGURE 10.21 Partitioning of 16-QAM constellation, which shows that $d_0 < d_1 < d_2 < d_3$.

property that the minimum Euclidean distances between their individual points follow an increasing pattern: $d_0 < d_1 < d_2$.

Figure 10.21 illustrates the partitioning of a rectangular constellation corresponding to 16-QAM. Here again we see that the subsets have increasing within-subset Euclidean distances: $d_0 < d_1 < d_2 < d_3$.

Based on the subsets resulting from successive partitioning of a two-dimensional constellation, we may devise relatively simple and yet highly effective coding schemes. Specifically, to send n bits/symbol with *quadrature modulation* (i.e., one that has in-phase and quadrature components), we start with a two-dimensional constellation of 2^{n+1} signal points appropriate for the modulation format of interest; a circular grid is used for M -ary PSK, and a rectangular one for M -ary QAM. In any event, the constellation is partitioned into 4 or 8 subsets. One or two incoming bits per symbol enter a rate-1/2 or rate-2/3 binary convolutional encoder, respectively; the resulting two or three coded bits per symbol determine the selection of a particular subset. The remaining uncoded data bits determine which particular point from the selected subset is to be signaled. This class of trellis codes is known as *Ungerboeck codes*.

Since the modulator has memory, we may use the Viterbi algorithm to perform maximum likelihood sequence estimation at the receiver. Each branch in the trellis of the Ungerboeck code corresponds to a subset rather than an individual signal point. The first step in the detection is to determine the signal point within each subset that is closest to the received signal point in the Euclidean sense. The signal point so determined and its metric (i.e., the squared Euclidean distance between it and the received point) may be used thereafter for the branch in question, and the Viterbi algorithm may then proceed in the usual manner.

■ UNGERBOECK CODES FOR 8-PSK

The scheme of Figure 10.22a depicts the simplest Ungerboeck 8-PSK code for the transmission of 2 bits/symbol. The scheme uses a rate-1/2 convolutional encoder; the corre-

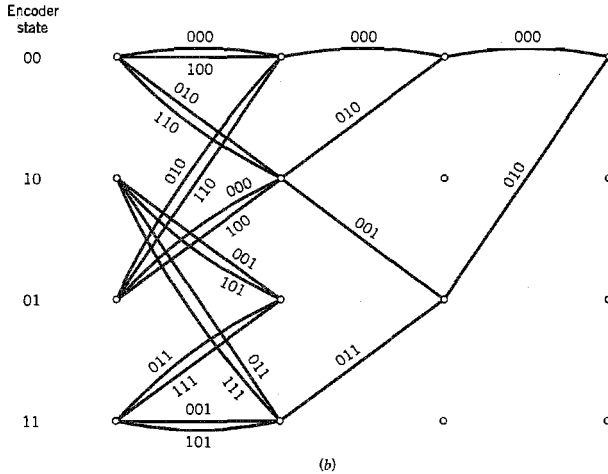
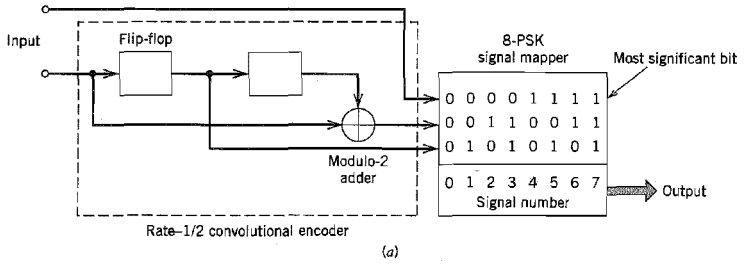
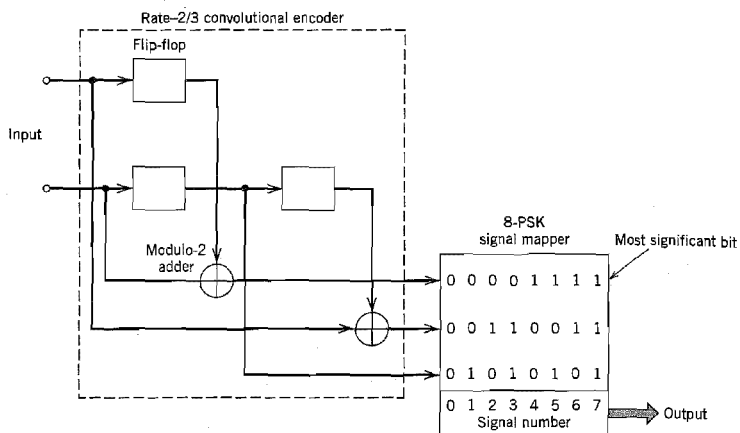


FIGURE 10.22 (a) Four-state Ungerboeck code for 8-PSK; the mapper follows Figure 10.20. (b) Trellis of the code.

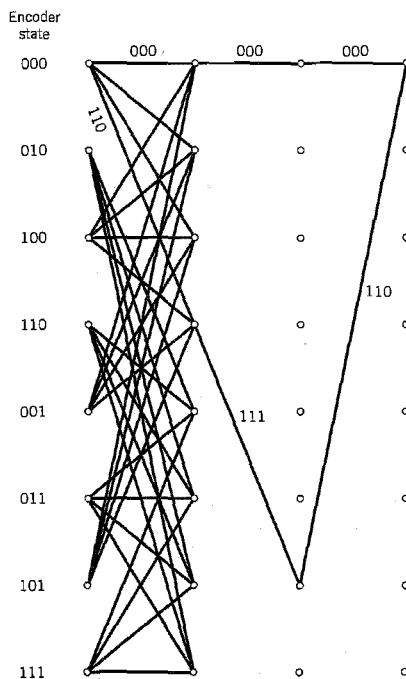
sponding trellis of the code is shown in Figure 10.22b, which has four states. Note that the most significant bit of the incoming binary word is left uncoded. Therefore, each branch of the trellis may correspond to two different output values of the 8-PSK modulator or, equivalently, to one of the four 2-point subsets shown in Figure 10.20. The trellis of Figure 10.22b also includes the minimum distance path.

The scheme of Figure 10.23a 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 10.23b. 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 10.23b also includes the minimum distance path.

Figures 10.22b and 10.23b 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.



(a)



(b)

FIGURE 10.23 (a) Eight-state Ungerboeck code for 8-PSK; the mapper follows Figure 10.20. (b) Trellis of the code with only some of the branches shown.

■ ASYMPTOTIC CODING GAIN

Following the discussion in Section 10.6, we define the *asymptotic coding gain* of Ungerboeck codes as

$$G_a = 10 \log_{10} \left(\frac{d_{\text{free}}^2}{d_{\text{ref}}^2} \right) \quad (10.68)$$

where d_{free} is the *free Euclidean distance* of the code and d_{ref} is the minimum Euclidean distance of an uncoded modulation scheme operating with the same signal energy per bit. For example, by using the Ungerboeck 8-PSK code of Figure 10.22a, the signal constellation has 8 message points, and we send 2 message bits per point. Hence, uncoded transmission requires a signal constellation with 4 message points. We may therefore regard uncoded 4-PSK as the reference for the Ungerboeck 8-PSK code of Figure 10.22a.

The Ungerboeck 8-PSK code of Figure 10.22a achieves an asymptotic coding gain of 3 dB, calculated as follows:

1. Each branch of the trellis in Figure 10.22b corresponds to a subset of two antipodal signal points. Hence, the free Euclidean distance d_{free} of the code can be no larger than the Euclidean distance d_2 between the antipodal signal points of such a subset. We may therefore write

$$d_{\text{free}} = d_2 = 2$$

where the distance d_2 is defined in Figure 10.24a; see also Figure 10.20.

2. The minimum Euclidean distance of an uncoded QPSK, viewed as a reference operating with the same signal energy per bit, equals (see Figure 10.24b)

$$d_{\text{ref}} = \sqrt{2}$$

Hence, as previously stated, the use of Equation (10.68) yields an asymptotic coding gain of $10 \log_{10} 2 = 3$ dB.

The asymptotic coding gain achievable with Ungerboeck codes increases with the number of states in the convolutional encoder. Table 10.9 presents the asymptotic coding gain (in dB) for Ungerboeck 8-PSK codes for increasing number of states, expressed with

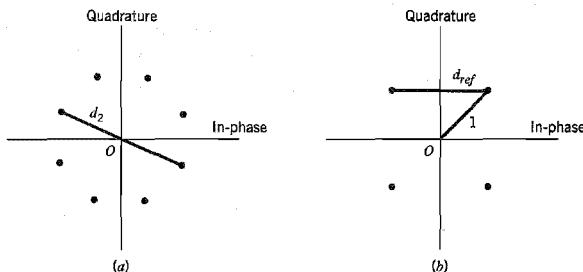


FIGURE 10.24 Signal-space diagrams for calculation of asymptotic coding gain of Ungerboeck 8-PSK code. (a) Definition of distance d_2 . (b) Definition of reference distance d_{ref} .

TABLE 10.9 Asymptotic coding gain of Ungerboeck 8-PSK codes, with respect to uncoded 4-PSK

Number of states	4	8	16	32	64	128	256	512
Coding gain (dB)	3	3.6	4.1	4.6	4.8	5	5.4	5.7

respect to uncoded 4-PSK. Note that improvements on the order of 6 dB require codes with a very large number of states.

10.8 Turbo Codes¹³

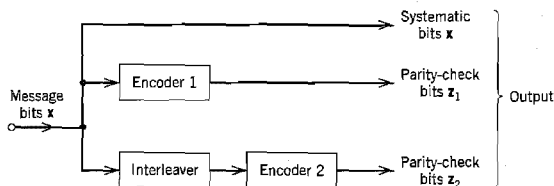
Traditionally, the design of good codes has been tackled by constructing codes with a great deal of algebraic structure, for which there are feasible decoding schemes. Such an approach is exemplified by the linear block codes and convolutional codes discussed in preceding sections. The difficulty with these traditional codes is that, in an effort to approach the theoretical limit for Shannon's channel capacity, we need to increase the code-word length of a linear block code or the constraint length of a convolutional code, which, in turn, causes the computational complexity of a maximum likelihood decoder to increase exponentially. Ultimately, we reach a point where complexity of the decoder is so high that it becomes physically unrealizable.

Various approaches have been proposed for the construction of powerful codes with large "equivalent" block lengths structured in such a way that the decoding can be split into a number of manageable steps. Building on these previous approaches, the development of *turbo codes* and *low-density parity-check codes* has been by far most successful. Indeed, this development has opened a brand new and exciting way of constructing good codes and decoding them with feasible complexity. Turbo codes are discussed in this section and low-density parity-check codes are discussed in Section 10.10.

■ TURBO CODING

In its most basic form, the encoder of a turbo code consists of two *constituent* systematic encoders joined together by means of an interleaver, as illustrated in Figure 10.25.

An *interleaver* is an input-output mapping device that *permutes* the ordering of a sequence of symbols from a fixed alphabet in a completely deterministic manner; that is, it takes the symbols at the input and produces identical symbols at the output but in a different temporal order. The interleaver can be of many types, of which the periodic and pseudo-random are two. Turbo codes use a pseudo-random interleaver, which operates

**FIGURE 10.25** Block diagram of turbo encoder.

only on the systematic bits. There are two reasons for the use of an interleaver in a turbo code:

- ▶ To tie together errors that are easily made in one half of the turbo code to errors that are exceptionally unlikely to occur in the other half. This is indeed the main reason why the turbo code performs better than a traditional code.
- ▶ To provide robust performance with respect to mismatched decoding, which is a problem that arises when the channel statistics are not known or have been incorrectly specified.

Typically, but not necessarily, the same code is used for both constituent encoders in Figure 10.25. The constituent codes recommended for turbo codes are short constraint-length *recursive systematic convolutional (RSC) codes*. The reason for making the convolutional codes recursive (i.e., feeding one or more of the tap outputs in the shift register back to the input) is to make the internal state of the shift register depend on past outputs. This affects the behavior of the error patterns (a single error in the systematic bits produces an infinite number of parity errors), with the result that a better performance of the overall coding strategy is attained.

▶ EXAMPLE 10.8 Eight-state RSC Encoder

Figure 10.26 shows an example eight-state RSC encoder. The generator matrix for this recursive convolutional code is

$$g(D) = \left[1, \frac{1 + D + D^2 + D^3}{1 + D + D^3} \right] \quad (10.69)$$

where D is the delay variable. The second entry of the matrix $g(D)$ is the transfer function of the feedback shift register, defined as the transform of the output divided by the transform of the input. Let $M(D)$ denote the transform of the message sequence $\{m_i\}_{i=1}^k$ and $B(D)$ denote the transform of the parity sequence $\{b_i\}_{i=1}^k$. By definition, we have

$$\frac{B(D)}{M(D)} = \frac{1 + D + D^2 + D^3}{1 + D + D^3}$$

Cross-multiplying, we get:

$$(1 + D + D^2 + D^3)M(D) = (1 + D + D^3)B(D)$$

which, on inversion into the time domain, yields

$$m_i + m_{i-1} + m_{i-2} + m_{i-3} + b_i + b_{i-1} + b_{i-3} = 0 \quad (10.70)$$

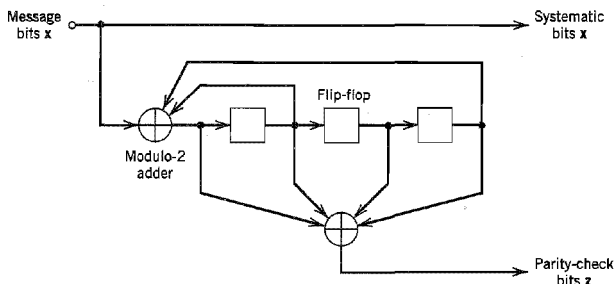


FIGURE 10.26 Example eight-state recursive systematic convolutional (RSC) encoder.

where the addition is modulo-2. Equation (10.70) is the parity-check equation, which the convolutional encoder of Figure 10.26 satisfies at each time step i . ◀

In Figure 10.25 the input data stream is applied directly to encoder 1, and the pseudo-randomly reordered version of the same data stream is applied to encoder 2. The systematic bits (i.e., original message bits) and the two sets of parity-check bits generated by the two encoders constitute the output of the turbo encoder. Although the constituent codes are convolutional, in reality turbo codes are block codes with the block size being determined by the size of the interleaver. Moreover, since both RSC encoders in Figure 10.25 are linear, we may describe turbo codes as *linear block codes*.

The block nature of the turbo code raises a practical issue: How do we know the beginning and the end of a code word? The common practice is to initialize the encoder to the *all-zero state* and then encode the data. After encoding a certain number of data bits a number of tail bits are added so as to make the encoder return to the all-zero state at the end of each block; thereafter the cycle is repeated. The termination approaches of turbo codes include the following:

- ▶ A simple approach is to terminate the first RSC code in the encoder and leave the second one unterminated. A drawback of this approach is that the bits at the end of the block due to the second RSC code are more vulnerable to noise than the other bits. Experimental work has shown that turbo codes exhibit a leveling off in performance as the SNR increases. This behavior is not like an error floor, but it has the appearance of an error floor compared to the steep drop in error performance at low SNR. This *error floor* is affected by a number of factors, the dominant one of which is the choice of interleaver.
- ▶ A more refined approach¹⁴ is to terminate both constituent codes in the encoder in a symmetric manner. Through the combined use of a good interleaver and dual termination, the error floor can be reduced by an order of magnitude compared to the simple termination approach.

In the original version of the turbo encoder, the parity-check bits generated by the two encoders in Figure 10.25 were punctured prior to data transmission over the channel to maintain the rate at 1/2. A *punctured code* is constructed by deleting certain parity check bits, thereby increasing the data rate. Puncturing is the inverse of extending a code. It should, however, be emphasized that the use of a puncture map is not a necessary requirement for the generation of turbo codes.

The novelty of the parallel encoding scheme of Figure 10.25 is in the use of recursive systematic convolutional (RSC) codes and the introduction of a pseudo-random interleaver between the two encoders. Thus a turbo code appears essentially *random* to the channel by virtue of the pseudo-random interleaver, yet it possesses sufficient structure for the decoding to be physically realizable. Coding theory asserts that a code chosen at random is capable of approaching Shannon's channel capacity, provided that the block size is sufficiently large.¹⁵ This is indeed the reason behind the impressive performance of turbo codes, as discussed next.

■ PERFORMANCE OF TURBO CODES

Figure 10.27 shows the error performance of a 1/2 rate, turbo code with a large block size for binary data transmission over an AWGN channel.¹⁶ The code uses an interleaver of

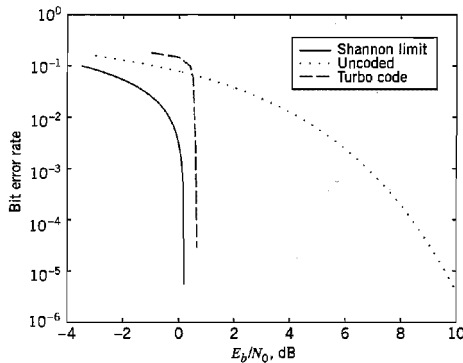


FIGURE 10.27 Noise performances of 1/2 rate, turbo code and uncoded transmission for AWGN channel; the figure also includes Shannon's theoretical limit on channel capacity for code rate $r = 1/2$.

size 65,536 and a BCJR-based decoder; details of this decoder are presented later in the section. Eighteen iterations of turbo decoding were used in the computation.

For the purpose of comparison, Figure 10.27 also includes two other curves for the same AWGN channel:

- Uncoded transmission (i.e., code rate $r = 1$).
- Shannon's theoretical limit for code rate 1/2, which follows from Figure 9.18b.

From Figure 10.27, we may draw two important conclusions:

1. Although the bit error rate for the turbo-coded transmission is significantly higher than that for uncoded transmission at low E_b/N_0 , the bit error rate for the turbo-coded transmission drops very rapidly once a critical value of E_b/N_0 has been reached.
2. At a bit error rate of 10^{-5} , the turbo code is less than 0.5 dB from Shannon's theoretical limit.

Note, however, attaining this highly impressive performance requires that the size of the interleaver, or, equivalently, the block length of the turbo code, be large. Also, the large number of iterations needed to improve performance increases the decoder latency. This drawback is due to the fact that the digital processing of information does not lend itself readily to the application of feedback, which is a distinctive feature of the turbo decoder.

Now that we have an appreciation for the impressive performance of turbo codes, the stage is set for a discussion of how turbo decoding is actually performed.

■ TURBO DECODING

Turbo codes derive their distinctive name from analogy of the decoding algorithm to the "turbo engine" principle. Figure 10.28a shows the basic structure of the turbo decoder. It operates on noisy versions of the systematic bits and the two sets of parity-check bits in two decoding stages to produce an estimate of the original message bits.

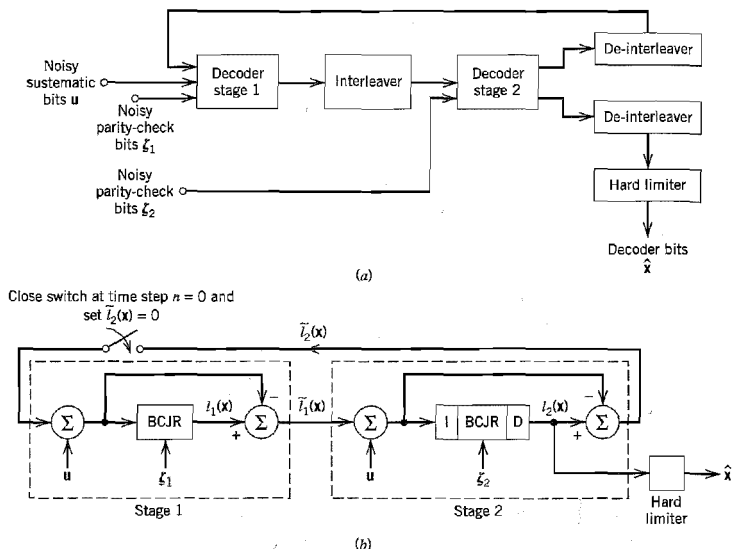


FIGURE 10.28 (a) Block diagram of turbo decoder. (b) Extrinsic form of turbo decoder, where I stands for interleaver, D for de-interleaver, and BCJR for BCJR algorithm for log-MAP decoding.

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-

sented as a trellis, then the present state of the trellis depends only on the past state and the input bit. (A mathematical treatment of the BCJR algorithm is given later in this section.)

Before proceeding to describe the operation of the two-stage turbo decoder in Figure 10.28a, we find it desirable to introduce the notion of extrinsic information. The most convenient representation for this concept is as a log-likelihood ratio, in which case extrinsic information is computed as the difference between two log-likelihood ratios as depicted in Figure 10.29. Formally, *extrinsic information*, generated by a decoding stage for a set of systematic (message) bits, is defined as the difference between the log-likelihood ratio computed at the output of that decoding stage and the *intrinsic information* represented by a log-likelihood ratio fed back to the input of the decoding stage. In effect, extrinsic information is the incremental information gained by exploiting the dependencies that exist between a message bit of interest and incoming raw data bits processed by the decoder.

On this basis, we may depict the flow of information in the two-stage turbo decoder of Figure 10.28a in a *symmetric extrinsic* manner as shown in Figure 10.28b. The first decoding stage uses the BCJR algorithm to produce a soft estimate of systematic bit x_j , expressed as the log-likelihood ratio

$$l_1(x_j) = \log \left(\frac{P(x_j = 1 | \mathbf{u}, \xi_1, \tilde{l}_2(\mathbf{x}))}{P(x_j = 0 | \mathbf{u}, \xi_1, \tilde{l}_2(\mathbf{x}))} \right), \quad j = 1, 2, \dots, k \quad (10.71)$$

where \mathbf{u} is the set of noisy systematic bits, ξ_1 is the set of noisy parity-check bits generated by encoder 1, and $\tilde{l}_2(\mathbf{x})$ is the extrinsic information about the set of message bits \mathbf{x} derived from the second decoding stage and fed back to the first stage. Assuming that the k message bits are statistically independent, the total log-likelihood ratio at the output of the first decoding stage is therefore

$$l_1(\mathbf{x}) = \sum_{j=1}^k l_1(x_j) \quad (10.72)$$

Hence, the extrinsic information about the message bits derived from the first decoding stage is

$$\tilde{l}_1(\mathbf{x}) = l_1(\mathbf{x}) - \tilde{l}_2(\mathbf{x}) \quad (10.73)$$

where $\tilde{l}_2(\mathbf{x})$ is to be defined.

Before application to the second decoding stage, the extrinsic information $\tilde{l}_1(\mathbf{x})$ is reordered to compensate for the pseudo-random interleaving introduced in the turbo encoder. In addition, the noisy parity-check bits ξ_2 generated by encoder 2 are used as input. Thus by using the BCJR algorithm, the second decoding stage produces a more refined

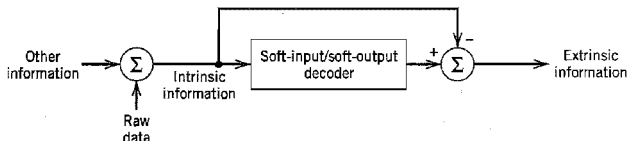


FIGURE 10.29 Illustrating the concept of extrinsic information.

soft estimate of the message bits \mathbf{x} . This estimate is re-interleaved to produce the total log-likelihood ratio $l_2(\mathbf{x})$. The extrinsic information $\tilde{l}_2(\mathbf{x})$ fed back to the first decoding stage is therefore

$$\tilde{l}_2(\mathbf{x}) = l_2(\mathbf{x}) - \tilde{l}_1(\mathbf{x}) \quad (10.74)$$

where $\tilde{l}_1(\mathbf{x})$ is itself defined by Equation (10.73), and $l_2(\mathbf{x})$ is the log-likelihood ratio computed by the second stage. Specifically, for the j th element of the vector \mathbf{x} , we have

$$l_2(x_j) = \log_2 \left(\frac{P(x_j = 1 | \mathbf{u}, \zeta_2, \tilde{l}_1(\mathbf{x}))}{P(x_j = 0 | \mathbf{u}, \zeta_2, \tilde{l}_1(\mathbf{x}))} \right), \quad j = 1, 2, \dots, k \quad (10.75)$$

Through the application of $\tilde{l}_2(\mathbf{x})$ to the first stage, the feedback loop around the pair of decoding stages is thereby closed. Note that although in actual fact the set of noisy systematic bits \mathbf{u} is only applied to the first decoding stage as in Figure 10.28a, by formulating the information flow in the symmetric extrinsic manner depicted in Figure 10.28b we find that \mathbf{u} is, in effect, also applied to the second decoding stage.

An estimate of the message bits \mathbf{x} is computed by hard-limiting the log-likelihood ratio $l_2(\mathbf{x})$ at the output of the second stage, as shown by

$$\hat{\mathbf{x}} = \text{sgn}(l_2(\mathbf{x})) \quad (10.76)$$

where the signum function operates on each element of $l_2(\mathbf{x})$ individually.

To initiate the turbo decoding algorithm, we simply set $l_2(\mathbf{x}) = 0$ on the first iteration of the algorithm; see Figure 10.28b.

The motivation for feeding only extrinsic information from one stage to the next in the turbo decoder of Figure 10.28 is to maintain as much statistical independence between the bits as possible from one iteration to the next. The feedback decoding strategy described herein implicitly relies on this assumption. If this assumption of statistical independence is strictly true, it can be shown that the estimate $\hat{\mathbf{x}}$ defined in Equation (10.76) approaches the MAP solution as the number of iterations approaches infinity.¹⁸ The assumption of statistical independence appears to be close to the truth in the vast majority of cases encountered in practice.

■ THE BCJR ALGORITHM

For a discussion of turbo decoding to be complete, a mathematical exposition of the BCJR algorithm for MAP estimation is in order.

Let $x(t)$ be the input to a trellis encoder at time t . Let $y(t)$ be the corresponding output observed at the receiver. Note that $y(t)$ may include more than one observation; for example, a rate $1/n$ code produces n bits for each input bit, in which case we have an n -dimensional observation vector. Let the observation vector be denoted by

$$\mathbf{y}_{(1,t)} = [y(1), y(2), \dots, y(t)]$$

Let $\lambda_m(t)$ denote the probability that a state $s(t)$ of the trellis encoder equals m , where $m = 1, 2, \dots, M$. We may then write

$$\lambda(t) = P[s(t) | \mathbf{y}] \quad (10.77)$$

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

$$P(x(t) = 1 | \mathbf{y}) = \sum_{s \in \mathcal{F}_A} \lambda_s(t) \quad (10.78)$$

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

$$\boldsymbol{\alpha}(t) = P(\mathbf{s}(t) | \mathbf{y}_{(1,t)}) \quad (10.79)$$

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

$$\boldsymbol{\beta}(t) = P(\mathbf{s}(t) | \mathbf{y}_{(t,k)}) \quad (10.80)$$

where

$$\mathbf{y}_{(t,k)} = [y(t), y(t+1), \dots, y(k)]$$

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

$$\boldsymbol{\lambda}(t) = \frac{\boldsymbol{\alpha}(t) \cdot \boldsymbol{\beta}(t)}{\|\boldsymbol{\alpha}(t) \cdot \boldsymbol{\beta}(t)\|_1} \quad (10.81)$$

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

$$\boldsymbol{\alpha}(t) \cdot \boldsymbol{\beta}(t) = \begin{bmatrix} \alpha_1(t)\beta_1(t) \\ \alpha_2(t)\beta_2(t) \\ \vdots \\ \alpha_M(t)\beta_M(t) \end{bmatrix} \quad (10.82)$$

and the L_1 norm of $\boldsymbol{\alpha}(t) \cdot \boldsymbol{\beta}(t)$ is defined by

$$\|\boldsymbol{\alpha}(t) \cdot \boldsymbol{\beta}(t)\|_1 = \sum_{m=1}^M \alpha_m(t)\beta_m(t) \quad (10.83)$$

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

$$\gamma_{m',m}(t) = P(\mathbf{s}(t) = m, \mathbf{y}(t) | \mathbf{s}(t-1) = m') \quad (10.84)$$

and denote the M -by- M matrix of transition probabilities as

$$\Gamma(t) = \{\gamma_{m',m}(t)\} \quad (10.85)$$

We may then formulate the *recursion theorem* as follows:

The forward estimate $\alpha(t)$ and backward estimate $\beta(t)$ are computed recursively as

$$\alpha^T(t) = \frac{\alpha^T(t-1)\Gamma(t)}{\|\alpha^T(t-1)\Gamma(t)\|_1} \quad (10.86)$$

and

$$\beta(t) = \frac{\Gamma(t+1)\beta(t+1)}{\|\Gamma(t+1)\beta(t+1)\|_1} \quad (10.87)$$

where the superscript T denotes matrix transposition.

The separability and recursion theorems together define the BCJR algorithm for the computation of *a posteriori* probabilities of the states and transitions of a code trellis, given the observation vector. Using these estimates, the likelihood ratios needed for turbo decoding may then be computed by performing summations over selected subsets of states as required.

10.9 Computer Experiment: Turbo Decoding

Two properties constitute the hallmark of turbo codes:

Property 1:

The error performance of the turbo decoder improves with the number of iterations of the decoding algorithm. This is achieved by feeding extrinsic information from the output of the first decoding stage to the input of the second decoding stage in the forward path and feeding extrinsic information from the output of the second stage to the input of the first stage in the backward path, and then permitting the iterative decoding process to take its natural course in response to the received noisy message and parity bits.

Property 2

The turbo decoder is capable of approaching the Shannon theoretical limit of channel capacity in a computationally feasible manner; this property has been demonstrated experimentally but not yet proven theoretically.

Property 2 requires that the block length of the turbo code be large. Unfortunately, a demonstration of this property requires the use of sophisticated implementations of the turbo decoding algorithm that are beyond the scope of this book. Accordingly, we focus our attention on a demonstration of Property 1 in this computer experiment.

So, as the primary objective of this computer experiment, we wish to use the log-MAP implementation of the BCJR algorithm to demonstrate Property 1 of turbo decoding.

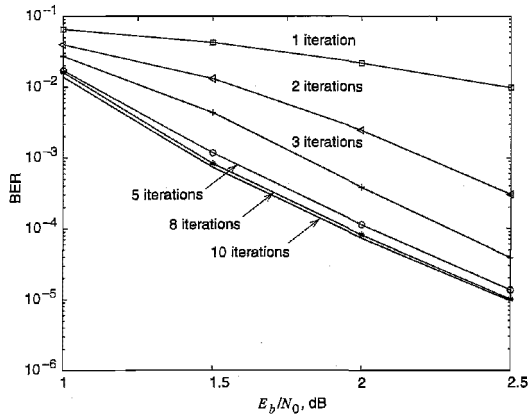


FIGURE 10.30 Results of the computer experiment on turbo decoding, for increasing number of iterations.

The only channel impairment assumed in the experiment is additive white Gaussian noise. Details of the turbo encoder and decoder are as follows:

Turbo Encoder (described in Figure 10.25):

Encoder 1: convolutional encoder [1, 1, 1]

Encoder 2: convolutional encoder [1, 0, 1]

Block (i.e., interleaver) length: 1,200 bits

Turbo Decoder (described in Figure 10.28):

The BCJR algorithm for log-MAP decoding.

The experiment was carried out for $E_b/N_0 = 1, 1.5, 2$, and 2.5 dB, with varying number of iterations at each E_b/N_0 . For each trial of the experiment, the number of bit errors was calculated after accumulating a total of 20 blocks of data (each 1,200 bits long) that were noise-corrupted. The probability of error was then evaluated as the ratio of bit errors to the total number of encoded bits. Note that in this calculation, many of the blocks of encoded bits were correctly decoded.

The results of the experiment are plotted in Figure 10.30. The following observations can be made from this figure:

1. For a given E_b/N_0 , the probability of error decreases with increasing number of iterations, confirming Property 1 of turbo decoding.
2. After eight iterations, there is no significant improvement in decoding performance.
3. For a fixed number of iterations, the probability of error decreases with increasing E_b/N_0 , which is to be expected.

10.10 Low-Density Parity-Check Codes¹⁹

Turbo codes, discussed in Section 10.8, and low-density parity-check (LDPC) codes, discussed in this section, belong to a broad family of error-control coding techniques called

compound codes. The two most important advantages of LDPC codes over turbo codes are:

- ▶ Absence of low-weight code words.
- ▶ Iterative decoding of lower complexity.

With regard to the issue of low-weight code words, we usually find that a small number of code words in a turbo code are undesirably close to the given code word. Due to this closeness in weights, once in a while the channel noise causes the transmitted code word to be mistaken for a nearby code word. Indeed, it is this behavior that is responsible for the error floor (typically around a bit error rate of 10^{-5} to 10^{-6}) that was mentioned earlier. In contrast, LDPC codes can be easily constructed so that they do not have such low-weight code words, and they can therefore achieve *vanishingly small* bit error rates. The error-floor problem in turbo codes can be alleviated by careful design of the interleaver.

Turning next to the issue of decoding complexity, we note that the computational complexity of a turbo decoder is dominated by the BCJR algorithm, which operates on the trellis for the convolutional code used in the encoder. The number of computations in each recursion of the BCJR algorithm scales linearly with the number of states in the trellis. Commonly used turbo codes employ trellises with 16 states or more. In contrast, LDPC codes use a simple parity-check trellis that has just two states. Consequently, the decoders for LDPC codes are significantly simpler than those for turbo decoders. Moreover, being parallelizable, LDPC decoding may be performed at greater speeds than turbo decoding.

However, a practical objection to the use of LDPC codes is that for large block lengths, their encoding complexity is high compared to turbo codes.

■ CONSTRUCTION OF LDPC CODES

LDPC codes are specified by a parity-check matrix denoted by \mathbf{A} , which is *sparse*; that is, it consists mainly of 0s and a small number of 1s. In particular, we speak of (n, t_c, t_r) LDPC codes, where n denotes the block length, t_c denotes the weight (i.e., number of 1s) in each column of the matrix \mathbf{A} , and t_r denotes the weight of each row with $t_r > t_c$. The rate of such a LDPC code is

$$r = 1 - \frac{t_c}{t_r} \quad (10.88)$$

whose validity may be justified as follows. Let ρ denote the *density* of 1s in the parity-check matrix \mathbf{A} . Then, following the terminology introduced in Section 10.3, we may set

$$t_c = \rho(n - k)$$

and

$$t_r = \rho n$$

where $(n - k)$ is the number of rows in \mathbf{A} and n is the number of columns (i.e., the block length). Therefore, dividing t_c by t_r , we get

$$\frac{t_c}{t_r} = 1 - \frac{k}{n}$$

By definition, the code rate of a block code is k/n , hence the result of Equation (10.88) follows. For this result to hold, however, the rows of \mathbf{A} must be *linearly independent*.

The structure of LDPC codes is well portrayed by *bipartite graphs*. Figure 10.31 shows such a graph for the example code of $n = 10$, $t_c = 3$, and $t_r = 5$. The left-hand nodes in the graph of Figure 10.31 are *variable nodes*, which correspond to elements of the code word. The right-hand nodes of the graph are *check nodes*, which correspond to the set of parity-check constraints satisfied by code words in the code. LDPC codes of the type exemplified by the graph of Figure 10.31 are said to be *regular* in that all the nodes of a similar kind have exactly the same degree. In the example graph of Figure 10.31, the degree of the variable nodes is $t_c = 3$, and the degree of the check nodes is $t_r = 5$. As the block length n approaches infinity, each check node is connected to a vanishingly small fraction of variable nodes, hence the term *low-density*.

The matrix \mathbf{A} is constructed by putting 1s in \mathbf{A} at *random*, subject to the *regularity constraints*:

- Each column contains a small fixed number, t_c , of 1s.
- Each row contains a small fixed number, t_r , of 1s.

In practice, these regularity constraints are often violated slightly in order to avoid having linearly dependent rows in the parity-check matrix \mathbf{A} .

Unlike the linear block codes discussed in Section 10.3, the parity-check matrix \mathbf{A} of LDPC codes is *not* systematic (i.e., it does not have the parity-check bits appearing in diagonal form), hence the use of a symbol different from that used in Section 10.3. Nevertheless, for coding purposes, we may derive a generator matrix \mathbf{G} for LDPC codes by means of *Gaussian elimination* performed in modulo-2 arithmetic; this procedure is illustrated later in Example 10.9. Following the terminology introduced in Section 10.3, the 1-by- n code vector \mathbf{c} is first partitioned as

$$\mathbf{c} = [\mathbf{b} : \mathbf{m}]$$

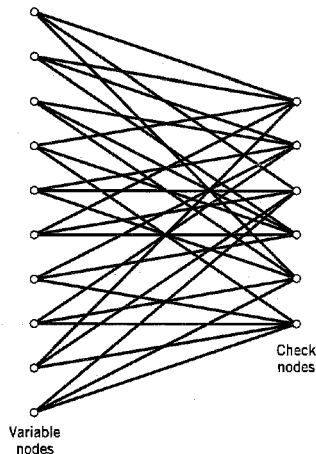


FIGURE 10.31 Bipartite graph of the (10, 3, 5) LDPC code.

where \mathbf{m} is the k -by-1 message vector, and \mathbf{b} is the $(n-k)$ -by-1 parity vector; see Equation (10.9). Correspondingly, the parity-check matrix \mathbf{A} is partitioned as

$$\mathbf{A}^T = \begin{bmatrix} \mathbf{A}_1 \\ \dots \\ \mathbf{A}_2 \end{bmatrix} \quad (10.89)$$

where \mathbf{A}_1 is a square matrix of dimensions $(n-k) \times (n-k)$, and \mathbf{A}_2 is a rectangular matrix of dimensions $k \times (n-k)$; transposition symbolized by the superscript T is used in the partitioning of matrix \mathbf{A} for convenience of presentation. Imposing the constraint of Equation (10.16) on the LDPC code, we may write

$$[\mathbf{b} : \mathbf{m}] \begin{bmatrix} \mathbf{A}_1 \\ \dots \\ \mathbf{A}_2 \end{bmatrix} = 0$$

or, equivalently,

$$\mathbf{b}\mathbf{A}_1 + \mathbf{m}\mathbf{A}_2 = 0 \quad (10.90)$$

Recall from Equation (10.7) that the vectors \mathbf{m} and \mathbf{b} are related by

$$\mathbf{b} = \mathbf{m}\mathbf{P}$$

where \mathbf{P} is the coefficient matrix. Hence, substituting this relation into Equation (10.90), we readily find that, for any nonzero message vector \mathbf{m} , the coefficient matrix of LDPC codes satisfies the condition

$$\mathbf{P}\mathbf{A}_1 + \mathbf{A}_2 = 0$$

which holds for *all* nonzero message vectors and, in particular, for \mathbf{m} in the form $[0 \cdots 0 \ 1 \ 0 \cdots 0]$ that will isolate individual rows of the generator matrix.

Solving this equation for matrix \mathbf{P} , we get

$$\mathbf{P} = \mathbf{A}_2\mathbf{A}_1^{-1} \quad (10.91)$$

where \mathbf{A}_1^{-1} is the *inverse* of matrix \mathbf{A}_1 , which is naturally defined in modulo-2 arithmetic. Finally, the generator matrix of LDPC codes is defined by

$$\begin{aligned} \mathbf{G} &= [\mathbf{P} : \mathbf{I}_k] \\ &= [\mathbf{A}_2\mathbf{A}_1^{-1} : \mathbf{I}_k] \end{aligned} \quad (10.92)$$

where \mathbf{I}_k is the k -by- k identity matrix; see Equation (10.12).

It is important to note that if we take the parity-check matrix \mathbf{A} for some arbitrary LDPC code and just pick $(n-k)$ columns of \mathbf{A} at random to form a square matrix \mathbf{A}_1 , there is *no* guarantee that \mathbf{A}_1 will be *nonsingular* (i.e., the inverse \mathbf{A}_1^{-1} will exist), even if the rows of \mathbf{A} are linearly independent. In fact, for a typical LDPC code with large block length n , such a randomly selected \mathbf{A}_1 is highly unlikely to be nonsingular, because it is very likely that at least one row of \mathbf{A}_1 will be all 0s. Of course, when the rows of \mathbf{A} are linearly independent, there will be *some* set of $(n-k)$ columns of \mathbf{A} that will make a nonsingular \mathbf{A}_1 , as illustrated in Example 10.9. For some construction methods for LDPC codes the first $(n-k)$ columns of \mathbf{A} may be guaranteed to produce a nonsingular \mathbf{A}_1 , or at least do so with a high probability, but that is *not* true in general.

► **EXAMPLE 10.9 (10, 3, 5) LDPC Code**

Consider the bipartite graph of Figure 10.31 pertaining to a (10, 3, 5) LDPC code. The parity-check matrix of the code is defined by

$$\mathbf{A} = \left[\begin{array}{cccccc|ccccc} 1 & 1 & 0 & 1 & 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 1 & 1 & 0 & 1 & 0 & 1 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 0 & 1 & 1 & 1 & 0 & 1 & 1 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 & 0 & 1 & 1 & 1 & 1 \end{array} \right]$$

$\underbrace{\hspace{10em}}_{\mathbf{A}_1^T} \quad \underbrace{\hspace{10em}}_{\mathbf{A}_2^T}$

which appears to be random, while maintaining the regularity constraints: $t_c = 3$ and $t_r = 5$. Partitioning the matrix \mathbf{A} in the manner described in Equation (10.89):

$$\mathbf{A}_1 = \begin{bmatrix} 1 & 0 & 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 1 & 0 \\ 1 & 0 & 0 & 1 & 0 & 1 \\ 0 & 1 & 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 1 & 0 & 0 \end{bmatrix}$$

$$\mathbf{A}_2 = \begin{bmatrix} 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 \end{bmatrix}$$

To derive the inverse of matrix \mathbf{A}_1 , we first use Equation (10.90) to write

$$\underbrace{[b_0, b_1, b_2, b_3, b_4, b_5]}_{\mathbf{b}} \underbrace{\begin{bmatrix} 1 & 0 & 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 1 & 0 \\ 1 & 0 & 0 & 1 & 0 & 1 \\ 0 & 1 & 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 1 & 0 & 0 \end{bmatrix}}_{\mathbf{A}_1} = \underbrace{[u_0, u_1, u_2, u_3, u_4, u_5]}_{\mathbf{u}} = \mathbf{mA}_2$$

where we have introduced the vector \mathbf{u} to denote the matrix product \mathbf{mA}_2 . By using Gaussian elimination, the matrix \mathbf{A}_1 is transformed into *lower diagonal form* (i.e., all the elements above the main diagonal are zero), as shown by

$$\mathbf{A}_1 \rightarrow \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 1 & 0 \\ 1 & 0 & 0 & 1 & 0 & 1 \end{bmatrix}$$

This transformation is achieved by the following modulo-2 additions performed on the columns of square matrix A_1 :

- Columns 1 and 2 are added to column 3.
- Column 2 is added to column 4.
- Columns 1 and 4 are added to column 5.
- Columns 1, 2 and 5 are added to column 6.

Correspondingly, the vector u is transformed as

$$u \rightarrow [u_0, u_1, u_0 + u_1 + u_2, u_1 + u_3, u_0 + u_3 + u_4, u_0 + u_1 + u_4 + u_5]$$

Accordingly, premultiplying the transformed matrix A_1 by the parity vector b , using successive eliminations in modulo-2 arithmetic working backwards, and putting the solutions for the elements of the parity vector b in terms of the elements of the vector u in matrix form, we get

$$\underbrace{[u_0, u_1, u_2, u_3, u_4, u_5]}_u \underbrace{\begin{bmatrix} 0 & 0 & 1 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 & 0 & 1 \\ 1 & 1 & 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 & 1 \\ 1 & 1 & 1 & 1 & 0 & 1 \end{bmatrix}}_{A_1^{-1}} = \underbrace{[b_0, b_1, b_2, b_3, b_4, b_5]}_b$$

The inverse of matrix A_1 is therefore

$$A_1^{-1} = \begin{bmatrix} 0 & 0 & 1 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 & 0 & 1 \\ 1 & 1 & 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 & 1 \\ 1 & 1 & 1 & 1 & 0 & 1 \end{bmatrix}$$

The matrix product $A_2 A_1^{-1}$ is (using the given value of A_2 and the value of A_1^{-1} just found)

$$A_2 A_1^{-1} = \begin{bmatrix} 1 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 & 1 & 0 \\ 0 & 1 & 0 & 1 & 1 & 0 \end{bmatrix}$$

Finally, using Equation (10.92), the generator of the (10, 3, 5) LDPC code is

$$G = \begin{bmatrix} 1 & 0 & 0 & 1 & 1 & 0 & : & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & : & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 0 & : & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 & 1 & 0 & : & 0 & 0 & 0 & 1 \end{bmatrix}$$

$\underbrace{\hspace{10em}}_{A_2 A_1^{-1}} \quad \underbrace{\hspace{10em}}_{I_k}$

It is important to recognize that the LDPC code described in this example is intended only for the purpose of illustrating the procedure involved in the generation of such a code. In practice, the block length n is orders of magnitude larger than that considered in this example. Moreover, in constructing the matrix A , we may constrain all pairs of columns to

have a *matrix overlap* (i.e., inner product of any two columns in matrix \mathbf{A}) not to exceed 1; such a constraint, over and above the regularity constraints, is expected to improve the performance of LDPC codes. Unfortunately, with a small block length as that considered in this example, it is difficult to satisfy this additional requirement. \triangleleft

■ MINIMUM DISTANCE OF LDPC CODES

In practice, the block length of a LDPC code is large, ranging from 10^3 to 10^6 , which means that the number of code words in a particular code is correspondingly large. Consequently, the algebraic analysis of LDPC codes is rather difficult. It is much more productive to perform a *statistical analysis* on an ensemble of LDPC codes. Such an analysis permits us to make statistical statements about certain properties of member codes in the ensemble. Moreover, an LDPC code with these properties can be found with high probability by a random selection from the ensemble.

Among these properties, the minimum distance of the member codes is of particular interest. From Section 10.3 we recall that the minimum distance of a linear block code is, by definition, the smallest Hamming distance between any pair of code vectors in the code. Over an ensemble of LDPC codes, the minimum distance of a member code is naturally a random variable. Elsewhere²⁰ it is shown that as the block length n increases, for fixed $t_c \geq 3$ and $t_r > t_c$ the probability distribution of the minimum distance can be overbounded by a function that approaches a unit step function at a fixed fraction Δ_{t_c, t_r} of the block length n . Thus, for large n , practically all the LDPC codes in the ensemble have a minimum distance of at least $n \Delta_{t_c, t_r}$. Table 10.10 presents the rate r and Δ_{t_c, t_r} of LDPC codes for different values of the weight-pair (t_c, t_r) . From this table we see that for $t_c = 3$ and $t_r = 6$ the code rate r attains its highest value of $1/2$ and the fraction Δ_{t_c, t_r} attains its smallest value, hence the preferred choice of $t_c = 3$ and $t_r = 6$ in the design of LDPC codes.

■ PROBABILISTIC DECODING OF LDPC CODES

At the transmitter, a message vector \mathbf{m} is encoded into a code vector $\mathbf{c} = \mathbf{mG}$, where \mathbf{G} is the generator matrix for a specified weight-pair (t_c, t_r) and therefore minimum distance d_{\min} . The vector \mathbf{c} is transmitted over a noisy channel to produce the received vector

$$\mathbf{r} = \mathbf{c} + \mathbf{e}$$

where \mathbf{e} is the error vector due to channel noise; see Equation (10.17). By construction, the matrix \mathbf{A} is a parity matrix of the LDPC code; that is, $\mathbf{AG}^T = \mathbf{0}$. Given the received

TABLE 10.10^a The rate r and fractional term Δ_{t_c, t_r} of LDPC codes for varying weights t_c and t_r

t_c	t_r	Rate r	Δ_{t_c, t_r}
5	6	0.167	0.255
4	5	0.2	0.210
3	4	0.25	0.122
4	6	0.333	0.129
3	5	0.4	0.044
3	6	0.5	0.023

^aAdapted from Gallager (1962) with permission of the IEEE.

vector \mathbf{r} , the bit-by-bit decoding problem is to find the most probable vector $\hat{\mathbf{c}}$ that satisfies the condition $\hat{\mathbf{c}}\mathbf{A}^T = 0$.

In what follows, a bit refers to an element of the received vector \mathbf{r} , and a check refers to a row of matrix \mathbf{A} . Let $\mathcal{J}(i)$ denote the set of bits that participate in check i . Let $\mathcal{S}(j)$ denote the set of checks in which bit j participates. A set $\mathcal{J}(i)$ that excludes bit j is denoted by $\mathcal{J}(i) \setminus j$. Likewise, a set $\mathcal{S}(j)$ that excludes check i is denoted by $\mathcal{S}(j) \setminus i$.

The decoding algorithm has two alternating steps: horizontal step and vertical step, which run along the rows and columns of matrix \mathbf{A} , respectively. In the course of these steps, two probabilistic quantities associated with nonzero elements of matrix \mathbf{A} are alternately updated. One quantity, denoted by P_{ij}^x , defines the probability that bit j is symbol x (i.e., symbol 0 or 1), given the information derived via checks performed in the horizontal step, except for check i . The second quantity, denoted by Q_{ij}^x , defines the probability that check i is satisfied, given that bit j is fixed at the value x and the other bits have the probabilities $P_{ij'} : j' \in \mathcal{S}(i) \setminus j$.

The LDPC decoding algorithm then proceeds as follows:²¹

Initialization

The variables P_{ij}^0 and P_{ij}^1 are set equal to the *a priori* probabilities p_j^0 and p_j^1 of symbols 0 and 1, respectively, with $p_j^0 + p_j^1 = 1$.

Horizontal Step

In the horizontal step of the algorithm, we run through the checks i . Define

$$\Delta P_{ij} = P_{ij}^0 - P_{ij}^1$$

For each weight-pair (i, j) , compute

$$\Delta Q_{ij} = \prod_{i' \in \mathcal{S}(i) \setminus j} \Delta P_{ij'}$$

Hence, set

$$Q_{ij}^0 = \frac{1}{2} (1 + \Delta Q_{ij})$$

$$Q_{ij}^1 = \frac{1}{2} (1 - \Delta Q_{ij})$$

Vertical Step

In the vertical step of the algorithm, the values of the probabilities P_{ij}^0 and P_{ij}^1 are updated using the quantities computed in the horizontal step. In particular, for each bit j , compute

$$P_{ij}^0 = \alpha_{ij} p_j^0 \prod_{i' \in \mathcal{S}(j) \setminus i} Q_{i'j}^0$$

$$P_{ij}^1 = \alpha_{ij} p_j^1 \prod_{i' \in \mathcal{S}(j) \setminus i} Q_{i'j}^1$$

where the scaling factor α_{ij} is chosen to make

$$P_{ij}^0 + P_{ij}^1 = 1$$

In the vertical step, we may also update the *pseudo-posterior probabilities*:

$$P_j^0 = \alpha_j p_j^0 \prod_{i \in \mathcal{S}(j)} Q_{ij}^0$$

$$P_j^1 = \alpha_j p_j^1 \prod_{i \in \mathcal{S}(j)} Q_{ij}^1$$

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{c} . If the condition $\hat{c}A^T = 0$ is satisfied, the decoding algorithm is terminated. Otherwise, the algorithm goes back to the horizontal step. If after some 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

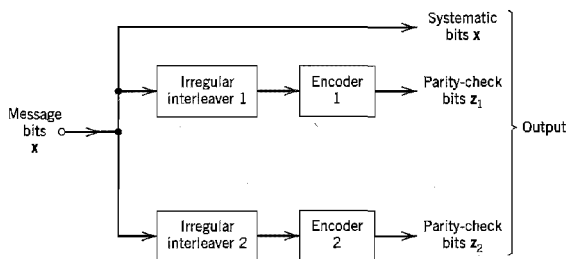


FIGURE 10.32 Block diagram of irregular turbo encoder.

a single one. As shown in Figure 10.32, similar *irregular interleavers* are used in both convolutional encoding paths to generate the parity-check bits z_1 and z_2 in response to the message bits x . Irregular turbo codes are decoded in a similar fashion to regular turbo codes.

To construct an *irregular LDPC code*,²³ the degrees of the variable and check nodes in the bipartite graph are chosen according to some distribution. For example, we may have an irregular LDPC code with the following graphical representation:

- ▶ One half of the variable nodes have degree 5 and the other half of the variable nodes have degree 3.
- ▶ One half of the check nodes have degree 6 and the other half of the check nodes have degree 8.

For a given block length and a given degree sequence, we define an ensemble of codes by choosing the edges (i.e., the connections between the variable and check nodes) in a random fashion. Specifically, the edges emanating from the variable nodes are enumerated in some arbitrary order, and likewise for the edges emanating from the check nodes.

Figure 10.33 plots the error performances of the following codes:²⁴

- ▶ Irregular LDPC code: $k = 50,000$, $n = 100,000$, rate = 1/2
- ▶ Turbo code (regular): $k = 65,536$, $n = 131,072$, and rate = 1/2
- ▶ Irregular turbo code: $k = 65,536$, $n = 131,072$, and rate = 1/2

where k is the number of message bits and n is the block length. The generator polynomials for the two convolutional encoders in the regular/irregular turbo codes are as follows:

$$\text{Encoder 1: } g(D) = 1 + D^4$$

$$\text{Encoder 2: } g(D) = 1 + D + D^2 + D^3 + D^4$$

Figure 10.33 also includes the corresponding theoretical limit on channel capacity for code rate $r = 1/2$.

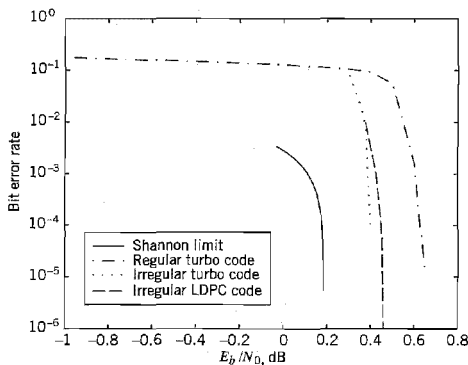


FIGURE 10.33 Noise performances of regular turbo code, irregular turbo code and irregular low-density parity-check (LDPC) code, compared to the Shannon limit for code rate $r = 1/2$.

Based on the results presented in Figure 10.33, we may make the following observations:

- ▶ The irregular LDPC code outperforms the regular turbo code in that it comes closer to Shannon's theoretical limit by 0.175 dB.
- ▶ Among the three codes displayed therein, the irregular turbo code is the best in that it is just 0.213 dB away from Shannon's theoretical limit.

10.12 Summary and Discussion

In this chapter, we studied error-control coding techniques that have established themselves as indispensable tools for reliable digital communication over noisy channels. The effect of errors occurring during transmission is reduced by adding redundancy to the data prior to transmission in a controlled manner. The redundancy is used to enable a decoder in the receiver to detect and correct errors.

Error-control coding techniques may be divided into two broadly defined families:

1. *Algebraic codes*, which rely on abstract algebraic structure built into the design of the codes for decoding at the receiver. Algebraic codes include Hamming codes, maximal-length codes, BCH codes, and Reed-Solomon codes. These particular codes share two properties:

Linearity property, the sum of any two code words in the code is also a code word.

Cyclic property, any cyclic shift of a code word is also a code word in the code.

Reed-Solomon codes are very powerful codes, capable of combatting both random and burst errors; they find applications in difficult environments such as deep-space communications and compact discs.

2. *Probabilistic codes*, which rely on probabilistic methods for their decoding at the receiver. Probabilistic codes include trellis codes, turbo codes, and low-density parity-check codes. In particular, the decoding is based on one or the other of two basic methods, as summarized here:

Soft input-hard output, which is exemplified by the Viterbi algorithm that performs maximum likelihood sequence estimation in the decoding of trellis codes.

Soft input-soft output, which is exemplified by the BCJR algorithm that performs maximum *a posteriori* estimation on a bit-by-bit basis in the decoding of turbo codes, or a special form of the sum-product algorithm in the decoding of low-density parity-check codes.

Trellis codes combine linear convolutional encoding and modulation to permit significant coding gains over conventional uncoded multilevel modulation without sacrificing bandwidth efficiency. Turbo codes and low-density parity-check codes share the following properties:

- ▶ Random encoding of a linear block kind.
- ▶ Error performance within a hair's breadth of Shannon's theoretical limit on channel capacity in a physically realizable fashion.

In practical terms, turbo codes and low-density parity-check codes have made it possible to achieve coding gains on the order of 10 dB, which is unmatched previously. These coding gains may be exploited to dramatically extend the range of digital communication receivers, substantially increase the bit rates of digital communication systems, or signifi-

cantly decrease the transmitted signal energy per symbol. These benefits have significant implications for the design of wireless communications and deep-space communications, just to mention two important applications of digital communications. Indeed, turbo codes have already been standardized for use on deep-space communication links and wireless communication systems.

NOTES AND REFERENCES

1. For an introductory discussion of error correction by coding, see Chapter 2 of Lucky (1989); see also the book by Adámek (1991), and the paper by Bhargava (1983). The classic book on error-control coding is Peterson and Weldon (1972). Error-control coding is also discussed in the classic book of Gallager (1968). The books of Lin and Costello (1983), Micleleson and Levesque (1985), MacWilliams and Sloane (1977), and Wilson (1998) are also devoted to error-control coding. For a collection of key papers on the development of coding theory, see the book edited by Berlekamp (1974).
2. For a survey of various ARQ schemes, see Lin, Costello, and Miller (1984).
3. In medicine, the term *syndrome* is used to describe a pattern of symptoms that aids in the diagnosis of a disease. In coding, the error pattern plays the role of the disease and parity-check failure that of a symptom. This use of *syndrome* was coined by Hagelbarger (1959).
4. The first error-correcting codes (known as Hamming codes) were invented by Hamming at about the same time as the conception of information theory by Shannon; for details, see the classic paper by Hamming (1950).
5. For a description of BCH codes and their decoding algorithms, see Lin and Costello (1983, pp. 141–183) and MacWilliams and Sloane (1977, pp. 257–293). Table 10.6 on binary BCH codes is adapted from Lin and Costello (1983).
6. The Reed-Solomon codes are named in honor of their inventors: see their classic 1960 paper. For details of Reed-Solomon codes, see MacWilliams and Sloane (1977, pp. 294–306). The book edited by Wicker and Bhargava (1994) contains an introduction to Reed-Solomon codes, a historical overview of these codes written by their inventors, Irving S. Reed and Gustave Solomon, and the applications of Reed-Solomon codes to the exploration of the solar system, the compact disc, automatic repeat-request protocols, and spread-spectrum multiple-access communications, and chapters on other related issues.
7. Convolutional codes were first introduced, as an alternative to block codes, by P. Elias (1955).
8. The term *trellis* was introduced by Forney (1973).
9. In a classic paper, Viterbi (1967) proposed a decoding algorithm for convolutional codes that has become known as the *Viterbi algorithm*. The algorithm was recognized by Forney (1972, 1973) to be a maximum likelihood decoder. Readable accounts of the Viterbi algorithm are presented in Lin and Costello (1983), Blahut (1990), and Adámek (1991).
10. Catastrophic convolutional codes are discussed in Benedetto, Biglieri, and Castellani (1987). Table 10.8 is adapted from their book.
11. For details of the evaluation of asymptotic coding gain for binary symmetric and binary-input AWGN channels, see Viterbi and Omura (1979, pp. 242–252) and Lin and Costello (1983, pp. 322–329).
12. Trellis-coded modulation was invented by G. Ungerboeck; its historical evolution is described in Ungerboeck (1982). Table 10.9 is adapted from this latter paper.

Trellis-coded modulation may be viewed as a form of *signal-space coding*—a viewpoint discussed at an introductory level in Chapter 14 of the book by Lee and Messer-

schmitt (1994). For an extensive treatment of trellis-coded modulation, see the books by Biglieri, Divsalar, McLane, and Simon (1991), and Schlegel (1997).

13. Turbo codes were originated by C. Berrou and A. Glavieux. Work on these codes was motivated by two papers on error-correcting codes: Battail (1987), and Hagenauer and Hoecher (1989). The first description of turbo codes using heuristic arguments was presented at a conference paper by Berrou, Glavieux, and Thitimajshima (1993); see also Berrou and Glavieux (1996). For reflections on the early work on turbo codes and subsequent developments, see Berrou and Glavieux (1998).

For a book on the basics of turbo codes, see Heegard and Wicker (1999). Using a procedure reminiscent of random coding (see Note 15), Benedetto and Montorosi (1996) have provided partial explanations for the impressive performance of turbo codes.

In two independent studies reported in the papers by McEliece, MacKay, and Cheng (1998), and Kschischang and Frey (1998), it is shown that turbo decoding duplicates an algorithm in artificial intelligence due to Pearl (1982), which involves the propagation of belief. The term *belief* is another way of referring to a *a posteriori* probability. These two papers have opened a new avenue of research, which links turbo decoding and learning machines. For an insightful discussion of turbo codes, see the book by Frey (1998).

A pseudo-random interleaver is basic to the operation of turbo codes. Deneshgaran and Mondin (1999) present a systematic procedure for designing interleavers (i.e., permuters) for turbo codes.

14. The dual termination of turbo codes is discussed in Guinand and Lodge (1996).
15. Random coding is discussed in Cover and Thomas (1991), Section 8.7.
16. The plots presented in Fig. 10.27 follow those in Fig. 6.8 of the book by Frey (1998).
17. In the early 1960s, Baum and Welch derived an iterative procedure for solving the parameter estimation problem, hence the name *Baum-Welch algorithm* (Baum and Petrie (1966); Baum et al. (1970)). In the *BCJR algorithm*, named after Bahl, Cocke, Jelinek, and Raviv (1974), the Baum-Welch algorithm is applied to the problem of soft output, maximum likelihood decoding of convolutional codes.
18. The proof that the estimate $\hat{\mathbf{x}}$ in Eq. (10.76) approaches the MAP solution as the number of iterations approaches infinity is discussed in the paper by Moher and Gulliver (1998).
19. Low-density parity-check (LDPC) codes were originally discovered by Gallager (1962, 1963). They were rediscovered independently by MacKay and Neal (1995); see also MacKay (1999).
In the 1960s and for a good while thereafter, the computers available at that time were not powerful enough to process the long block lengths that are needed to achieve excellent performance with LDPC codes, hence the lack of interest in their use for over twenty years.
20. For a detailed treatment of the statement that the probability distribution of the minimum distance of an LDPC code approaches a unit step function of the block length for certain values of weight-pair (t_o, t_e) , see Gallager (1962, 1963).
21. The decoding algorithm of LDPC codes described herein follows MacKay and Neal (1996, 1997).
22. Irregular turbo codes were invented by Frey and MacKay (1999).
23. Irregular LDPC codes were invented independently by MacKay et al. (1999) and Richardson et al. (1999).
24. The codes, whose performances are plotted in Fig. 10.34, are due to the following originators:
 - ▶ Regular turbo codes: Berrou and Glavieux (1996); Berrou et al. (1995).
 - ▶ Irregular turbo codes: Frey and MacKay (1999).
 - ▶ Irregular LDPC codes: Richardson et al. (1999).

PROBLEMS

Soft-Decision Coding

- 10.1 Consider a binary input Q -ary output discrete memoryless channel. The channel is said to be symmetric if the channel transition probability $p(j|i)$ satisfies the condition:

$$p(j|0) = p(Q-1-j|1), \quad j = 0, 1, \dots, Q-1$$

Suppose that the channel input symbols 0 and 1 are equally likely. Show that the channel output symbols are also equally likely; that is,

$$p(j) = \frac{1}{Q}, \quad j = 0, 1, \dots, Q-1$$

- 10.2 Consider the quantized demodulator for binary PSK signals shown in Fig. 10.3a. The quantizer is a four-level quantizer, normalized as in Fig. P10.2. Evaluate the transition probabilities of the binary input-quaternary output discrete memoryless channel so characterized. Hence, show that it is a symmetric channel. Assume that the transmitted signal energy per bit is E_b , and the additive white Gaussian noise has zero mean and power spectral density $N_0/2$.

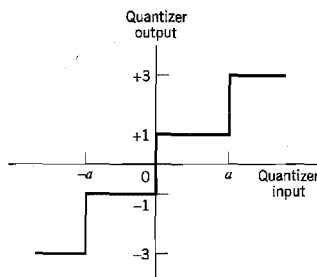


FIGURE P10.2

- 10.3 Consider a binary input AWGN channel, in which the binary symbols 1 and 0 are equally likely. The binary symbols are transmitted over the channel by means of phase-shift keying. The code symbol energy is E , and the AWGN has zero mean and power spectral density $N_0/2$. Show that the channel transition probability is given by

$$p(y|0) = \frac{1}{\sqrt{2\pi}} \exp \left[-\frac{1}{2} \left(y + \sqrt{\frac{2E}{N_0}} \right)^2 \right], \quad -\infty < y < \infty$$

Linear Block and Cyclic Codes

- 10.4 In a *single-parity-check code*, a single parity bit is appended to a block of k message bits (m_1, m_2, \dots, m_k) . The single parity bit b_1 is chosen so that the code word satisfies the *even parity rule*:

$$m_1 + m_2 + \dots + m_k + b_1 = 0, \quad \text{mod } 2$$

For $k = 3$, set up the 2^k possible code words in the code defined by this rule.

- 10.5 Compare the parity-check matrix of the (7, 4) Hamming code considered in Example 10.2 with that of a (4, 1) repetition code.

- 10.6 Consider the (7, 4) Hamming code of Example 10.2. The generator matrix G and the parity-check matrix H of the code are described in that example. Show that these two matrices satisfy the condition

$$HG^T = 0$$

- 10.7 (a) For the (7, 4) Hamming code described in Example 10.2, construct the eight code words in the dual code.
 (b) Find the minimum distance of the dual code determined in part (a).
- 10.8 Consider the (5, 1) repetition code of Example 10.1. Evaluate the syndrome s for the following error patterns:
 (a) All five possible single-error patterns
 (b) All 10 possible double-error patterns
- 10.9 For an application that requires error detection *only*, we may use a *nonsystematic* code. In this problem, we explore the generation of such a cyclic code. Let $g(X)$ denote the generator polynomial, and $m(X)$ denote the message polynomial. We define the code polynomial $c(X)$ simply as

$$c(X) = m(X)g(X)$$

Hence, for a given generator polynomial, we may readily determine the code words in the code. To illustrate this procedure, consider the generator polynomial for a (7, 4) Hamming code:

$$g(X) = 1 + X + X^3$$

Determine the 16 code words in the code, and confirm the nonsystematic nature of the code.

- 10.10 The polynomial $1 + X^7$ has $1 + X + X^3$ and $1 + X^2 + X^3$ as primitive factors. In Example 10.3, we used $1 + X + X^3$ as the generator polynomial for a (7, 4) Hamming code. In this problem, we consider the adoption of $1 + X^2 + X^3$ as the generator polynomial. This should lead to a (7, 4) Hamming code that is different from the code analyzed in Example 10.3. Develop the encoder and syndrome calculator for the generator polynomial:

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

Compare your results with those in Example 10.3.

- 10.11 Consider the (7, 4) Hamming code defined by the generator polynomial

$$g(X) = 1 + X + X^3$$

The code word 0111001 is sent over a noisy channel, producing the received word 0101001 that has a single error. Determine the syndrome polynomial $s(X)$ for this received word, and show that it is identical to the error polynomial $e(X)$.

- 10.12 The generator polynomial of a (15, 11) Hamming code is defined by

$$g(X) = 1 + X + X^4$$

Develop the encoder and syndrome calculator for this code, using a systematic form for the code.

- 10.13 Consider the (15, 4) maximal-length code that is the dual of the (15, 11) Hamming code of Problem 10.12. Do the following:
 (a) Find the feedback connections of the encoder, and compare your results with those of Table 7.1 on maximal-length codes presented in Chapter 7.
 (b) Find the generator polynomial $g(X)$; hence, determine the output sequence assuming the initial state 0001. Confirm the validity of your result by cycling the initial state through the encoder.

10.14 Consider the (31, 15) Reed-Solomon code.

- How many bits are there in a symbol of the code?
- What is the block length in bits?
- What is the minimum distance of the code?
- How many symbols in error can the code correct?

Convolutional Codes

10.15 A convolutional encoder has a single-shift register with two stages, (i.e., constraint length $K = 3$), three modulo-2 adders, and an output multiplexer. The generator sequences of the encoder are as follows:

$$g^{(1)} = (1, 0, 1)$$

$$g^{(2)} = (1, 1, 0)$$

$$g^{(3)} = (1, 1, 1)$$

Draw the block diagram of the encoder.

Note: For Problems 10.16–10.23, the same message sequence 10111... is used so that we may compare the outputs of different encoders for the same input.

10.16 Consider the rate $r = 1/2$, constraint length $K = 2$ convolutional encoder of Fig. P10.16. The code is systematic. Find the encoder output produced by the message sequence 10111....

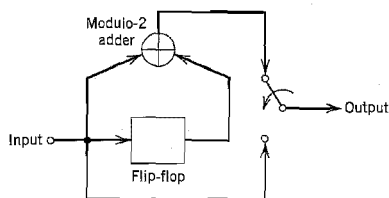


FIGURE P10.16

10.17 Figure P10.17 shows the encoder for a rate $r = 1/2$, constraint length $K = 4$ convolutional code. Determine the encoder output produced by the message sequence 10111....

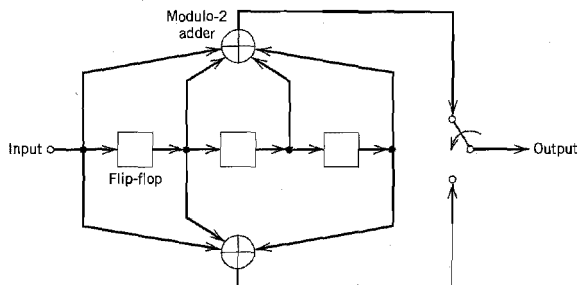


FIGURE P10.17

- 10.18 Consider the encoder of Fig. 10.13b for a rate $r = 2/3$, constraint length $K = 2$ convolutional code. Determine the code sequence produced by the message sequence 10111...
- 10.19 Construct the code tree for the convolutional encoder of Fig. P10.16. Trace the path through the tree that corresponds to the message sequence 10111..., and compare the encoder output with that determined in Problem 10.16.
- 10.20 Construct the code tree for the encoder of Fig. P10.17. Trace the path through the tree that corresponds to the message sequence 10111.... Compare the resulting encoder output with that found in Problem 10.17.
- 10.21 Construct the trellis diagram for the encoder of Fig. P10.17, assuming a message sequence of length 5. Trace the path through the trellis corresponding to the message sequence 10111.... Compare the resulting encoder output with that found in Problem 10.17.
- 10.22 Construct the state diagram for the encoder of Fig. P10.17. Starting with the all-zero state, trace the path that corresponds to the message sequence 10111..., and compare the resulting code sequence with that determined in Problem 10.17.
- 10.23 Consider the encoder of Fig. 10.13b.
- Construct the state diagram for this encoder.
 - Starting from the all-zero state, trace the path that corresponds to the message sequence 10111.... Compare the resulting sequence with that determined in Problem 10.18.
- 10.24 By viewing the minimum shift keying (MSK) scheme as a finite-state machine, construct the trellis diagram for MSK. (A description of MSK is presented in Chapter 6.)
- 10.25 The trellis diagram of a rate-1/2, constraint length-3 convolutional code is shown in Figure P10.25. The all-zero sequence is transmitted, and the received sequence is 100010000.... Using the Viterbi algorithm, compute the decoded sequence.

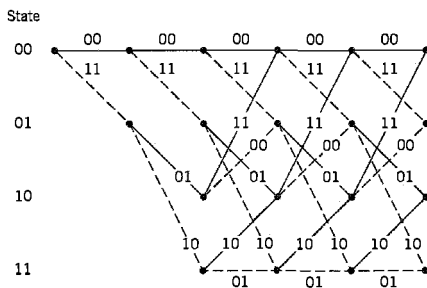


FIGURE P10.25

- 10.26 Consider a rate-1/2, constraint length-7 convolutional code with free distance $d_{\text{free}} = 10$. Calculate the asymptotic coding gain for the following two channels:
- Binary symmetric channel
 - Binary input AWGN channel
- 10.27 In Section 10.6 we described the Viterbi algorithm for maximum likelihood decoding of a convolutional code. Another application of the Viterbi algorithm is for maximum likelihood demodulation of a received sequence corrupted by intersymbol interference due to a dispersive channel. Figure P10.27 shows the trellis diagram for intersymbol interference, assuming a binary data sequence. The channel is discrete, described by the

finite impulse response (1, 0.1). The received sequence is (1.0, -0.3, -0.7, 0, ...). Use the Viterbi algorithm to determine the maximum likelihood decoded version of this sequence.

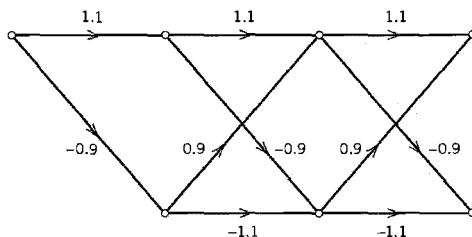


FIGURE P10.27

- 10.28 Figure P10.28 depicts 32-QAM cross constellation. Partition this constellation into eight subsets. At each stage of the partitioning, indicate the within-subset (shortest) Euclidean distance.

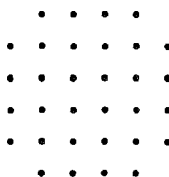


FIGURE P10.28

- 10.29 As explained in the Introduction to this chapter, channel coding can be used to reduce the E_b/N_0 required for a prescribed error performance or reduce the size of the receiving antenna for a prescribed E_b/N_0 . In this problem we explore these two practical benefits of coding by revisiting Example 8.2 in Chapter 8 on the downlink power calculations for a domestic satellite communication system. In particular, we now assume that the design of the downlink includes the use of a coding scheme consisting of a rate-1/2 convolutional encoder with length $K = 7$ and Viterbi decoding. The coding gain of this scheme is 5.1 dB, assuming the use of soft quantization. Hence do the following:
- Recalculate the required E_b/N_0 ratio of the system.
 - Assuming that the required E_b/N_0 ratio remains unchanged, calculate the reduction in the size of the receiving dish antenna that is made possible by the use of this coding scheme in the downlink.
- 10.30 Unlike the convolutional codes considered in this chapter, we recall from Chapter 6 that the convolutional code used in the voiceband modem V.32 modem is *nonlinear*. Figure P10.30 shows the circuit diagram of the convolutional encoder used in this modem; it uses modulo-2 multiplication and gates in addition to modulo-2 additions and delays. Explain the reason for nonlinearity of the encoder in Fig. P10.30, and use an example to illustrate your explanation.

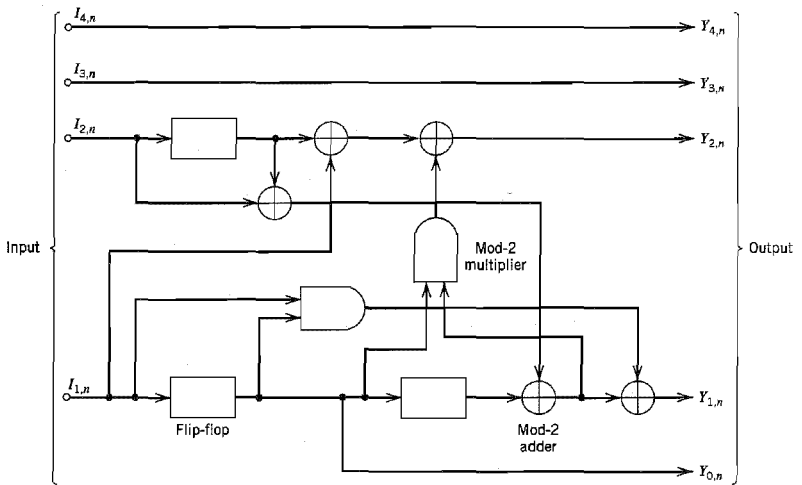


FIGURE P10.30

Turbo Codes

- 10.31 Let $r_c^{(1)} = p/q_1$ and $r_c^{(2)} = p/q_2$ be the code rates of RSC encoders 1 and 2 in the turbo encoder of Fig. 10.25. Find the code rate of the turbo code.
- 10.32 The feedback nature of the constituent codes in the turbo encoder of Fig. 10.25 has the following implication: A single bit error corresponds to an infinite sequence of channel errors. Illustrate this phenomenon by using a message sequence consisting of symbol 1 followed by an infinite number of symbols 0.
- 10.33 Consider the following generator matrices for rate 1/2 turbo codes:

$$\begin{aligned} \text{4-state encoder:} \quad g(D) &= \left[1, \frac{1 + D + D^2}{1 + D^2} \right] \\ \text{8-state encoder:} \quad g(D) &= \left[1, \frac{1 + D^2 + D^3}{1 + D + D^2 + D^3} \right] \\ \text{16-state encoder:} \quad g(D) &= \left[1, \frac{1 + D^4}{1 + D + D^2 + D^3 + D^4} \right] \end{aligned}$$

- (a) Construct the block diagram for each one of these RSC encoders.
- (b) Setup the parity-check equation associated with each encoder.
- 10.34 The turbo encoder of Fig. 10.25 involves the use of two RSC encoders.
- (a) Generalize this encoder to encompass a total of M interleavers.
- (b) Construct the block diagram of the turbo decoder that exploits the M sets of parity-check bits generated by such a generalization.
- 10.35 Turbo decoding relies on the feedback of extrinsic information. The fundamental principle adhered to in the turbo decoder is to avoid feeding a decoding stage information that stems from the stage itself. Explain the justification for this principle in conceptual terms.
- 10.36 Suppose a communication receiver consists of two components, a demodulator and a decoder. The demodulator is based on a Markov model of the combined modulator and

channel, and the decoder is based on a Markov model of a forward error correction code. Discuss how the turbo principle may be applied to construct a joint demodulator/decoder for this system.

Computer Experiment

- 10.37 In this experiment we continue the investigation into turbo codes presented in Section 10.9 by evaluating the effect of block size on the noise performance of the decoder.

As before, the two convolutional encoders of the turbo encoder are as follows;

Encoder 1: [1, 1, 1]

Encoder 2: [1, 0, 1]

The transmitted E_b/N_0 is 1 dB. The block errors to termination are prescribed not to exceed 15.

With this background information, plot the bit error rate of the turbo decoder versus the number of iterations for two different block (i.e., interleaver) sizes: 200 and 400.

PROBABILITY THEORY

A1.1 Probabilistic Concepts

Probability theory is rooted in phenomena that, explicitly or implicitly, can be modeled by an experiment with an outcome that is subject to *chance*. Moreover, if the experiment is repeated, the outcome can differ because of the influence of an underlying random phenomenon or chance mechanism. Such an experiment is referred to as a *random experiment*. For example, the experiment may be the observation of the result of tossing a fair coin. In this experiment, the possible outcomes of a trial are “heads” or “tails.”

To be more precise in the description of a random experiment, we ask for three features:

1. The experiment is repeatable under identical conditions.
2. On any trial of the experiment, the outcome is unpredictable.
3. For a large number of trials of the experiment, the outcomes exhibit *statistical regularity*; that is, a definite *average* pattern of outcomes is observed if the experiment is repeated a large number of times.

■ RELATIVE-FREQUENCY APPROACH

Let *event A* denote one of the possible outcomes of a random experiment. For example, in the coin-tossing experiment, event *A* may represent “heads.” Suppose that in n trials of the experiment, event *A* occurs $N_n(A)$ times. We may then assign the ratio $N_n(A)/n$ to the event *A*. This ratio is called the *relative frequency* of the event *A*. Clearly, the relative frequency is a *nonnegative real number less than or equal to one*. That is to say,

$$0 \leq \frac{N_n(A)}{n} \leq 1 \quad (\text{A1.1})$$

If event *A* occurs in none of the trials, $N_n(A)/n = 0$. If, on the other hand, event *A* occurs in all the n trials, $N_n(A)/n = 1$.

We say that the experiment exhibits *statistical regularity* if for any sequence of n trials the relative frequency $N_n(A)/n$ converges to the same limit as n becomes large. It thus seems natural for us to define the *probability of event A* as

$$P(A) = \lim_{n \rightarrow \infty} \left(\frac{N_n(A)}{n} \right) \quad (\text{A1.2})$$

The limit shown in Equation (A1.2) should not be viewed in a mathematical sense. Rather, we think of Equation (A1.2) as a statement that the probability of an event is the long-term proportion of times that a particular event *A* occurs in a long sequence of trials. For example, in the coin-tossing experiment, we may expect that out of a million tosses of a fair coin, about one half of them will show up heads.

The probability of an event is intended to represent the *likelihood* that a trial of the experiment will result in the occurrence of that event. For many engineering applications and games of chance, the use of Equation (A1.2) to define the probability of an event is acceptable. However, for many other applications, this definition is inadequate. Consider,

for example, the statistical analysis of the stock market: How are we to achieve repeatability of such an experiment? A more satisfying approach is to state the properties that any measure of probability is expected to have, postulate them as *axioms*, and then use relative-frequency interpretations to justify them.

■ AXIOMS OF PROBABILITY

When we perform a random experiment, it is natural for us to be aware of the various outcomes that are likely to arise. In this context, it is convenient to think of an experiment and its possible outcomes as defining a space and its points. With the k th outcome of the experiment, say, we associate a point called the *sample point*, which we denote by s_k . The totality of sample points corresponding to the aggregate of all possible outcomes of the experiment is called the *sample space*, which we denote by S . An event corresponds to either a single sample point or a set of sample points. In particular, the entire sample space S is called the *sure event*; the null set \emptyset is called the *null or impossible event*; and a single sample point is called an *elementary event*.

Consider, for example, an experiment that involves the throw of a die. In this experiment there are six possible outcomes: the showing of one, two, three, four, five, and six dots on the upper face of the die. By assigning a sample point to each of these possible outcomes, we have a one-dimensional sample space that consists of six sample points, as shown in Figure A1.1. The elementary event describing the statement “a six shows” corresponds to the sample point {6}. On the other hand, the event describing the statement “an even number of dots shows” corresponds to the subset {2, 4, 6} of the sample space. Note that the term *event* is used interchangeably to describe the subset or the statement.

We are now ready to make a formal definition of probability. A *probability system* consists of the triple:

1. A sample space S of elementary events (outcomes).
2. A class \mathcal{E} of events that are subsets of S .
3. A probability measure $P(\cdot)$ assigned to each event A in the class \mathcal{E} , which has the following properties:

(i)

$$P(S) = 1 \quad (\text{A1.3})$$

(ii)

$$0 \leq P(A) \leq 1 \quad (\text{A1.4})$$

(iii) If $A + B$ is the union of two mutually exclusive events in the class \mathcal{E} , then

$$P(A + B) = P(A) + P(B) \quad (\text{A1.5})$$

Properties (i), (ii), and (iii) are known as the *axioms of probability*. Axiom (i) states that the probability of the sure event is unity. Axiom (ii) states that the probability of an event

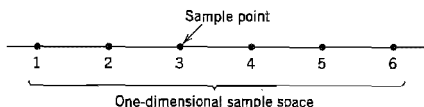


FIGURE A1.1 Sample space for the experiment of throwing a die.

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

$$P(\bar{A}) = 1 - P(A) \quad (\text{A1.6})$$

where \bar{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 \bar{A} :

$$S = A + \bar{A}$$

Then, the use of axioms (i) and (iii) yields

$$1 = P(A) + P(\bar{A})$$

from which Equation (A1.6) follows directly.

Property 2

If M mutually exclusive events A_1, A_2, \dots, A_M have the exhaustive property

$$A_1 + A_2 + \dots + A_M = S \quad (\text{A1.7})$$

then

$$P(A_1) + P(A_2) + \dots + P(A_M) = 1 \quad (\text{A1.8})$$

To prove this property, we first use axiom (i) in Equation (A1.7), and so write

$$P(A_1 + A_2 + \dots + A_M) = 1$$

Next, we generalize axiom (iii) by writing

$$P(A_1 + A_2 + \dots + A_M) = P(A_1) + P(A_2) + \dots + P(A_M)$$

Hence, the result of Equation (A1.8) follows. When the M events are *equally likely* (i.e., they have equal probabilities of occurrence), then Equation (A1.8) simplifies as

$$P(A_i) = \frac{1}{M}, \quad i = 1, 2, \dots, M$$

Property 3

When events A and B are not mutually exclusive, then the probability of the union event "A or B" equals

$$P(A + B) = P(A) + P(B) - P(AB) \quad (\text{A1.9})$$

where $P(AB)$ is the probability of the joint event "A and B."

The probability $P(AB)$ is called a *joint probability*. It has the following relative-frequency interpretation:

$$P(AB) = \lim_{n \rightarrow \infty} \left(\frac{N_n(AB)}{n} \right) \quad (\text{A1.10})$$

where $N_n(AB)$ denotes the number of times the events A and B occur simultaneously in n trials of the experiment. Axiom (iii) is a special case of Equation (A1.9); when A and B are mutually exclusive, $P(AB)$ is zero, and Equation (A1.9) reduces to the same form as Equation (A1.5).

■ CONDITIONAL PROBABILITY

Suppose we perform an experiment that involves a pair of events A and B. Let $P(B|A)$ denote the probability of event B, given that event A has occurred. The probability $P(B|A)$ is called the *conditional probability of B given A*. Assuming that A has nonzero probability, the conditional probability $P(B|A)$ is defined by

$$P(B|A) = \frac{P(AB)}{P(A)} \quad (\text{A1.11})$$

where $P(AB)$ is the joint probability of A and B.

We justify the definition of conditional probability given in Equation (A1.11) by presenting a relative-frequency interpretation of it. Suppose that we perform an experiment and examine the occurrence of a pair of events A and B. Let $N_n(AB)$ denote the number of times the joint event AB occurs in n trials. Suppose that in the same n trials, the event A occurs $N_n(A)$ times. Since the joint event AB corresponds to both A and B occurring, it follows that $N_n(A)$ must include $N_n(AB)$. In other words, we have

$$\frac{N_n(AB)}{N_n(A)} \leq 1$$

The ratio $N_n(AB)/N_n(A)$ represents the relative frequency of B given that A has occurred. For large n , this ratio equals the conditional probability $P(B|A)$; that is,

$$P(B|A) = \lim_{n \rightarrow \infty} \left(\frac{N_n(AB)}{N_n(A)} \right)$$

or, equivalently,

$$P(B|A) = \lim_{n \rightarrow \infty} \left(\frac{N_n(AB)/n}{N_n(A)/n} \right)$$

Recognizing that

$$P(AB) = \lim_{n \rightarrow \infty} \left(\frac{N_n(AB)}{n} \right)$$

and

$$P(A) = \lim_{n \rightarrow \infty} \left(\frac{N_n(A)}{n} \right)$$

the result of Equation (A1.11) follows.

We may rewrite Equation (A1.11) as

$$P(AB) = P(B|A)P(A) \quad (\text{A1.12})$$

It is apparent that we may also write

$$P(AB) = P(A|B)P(B) \quad (\text{A1.13})$$

Accordingly, we may state that *the joint probability of two events may be expressed as the product of the conditional probability of one event given the other, and the elementary probability of the other*. Note that the conditional probabilities $P(B|A)$ and $P(A|B)$ have essentially the same properties as the various probabilities previously defined.

Situations may exist where the conditional probability $P(A|B)$ and the probabilities $P(A)$ and $P(B)$ are easily determined directly, but the conditional probability $P(B|A)$ is desired. From Equations (A1.12) and (A1.13), it follows that, provided $P(A) \neq 0$, we may determine $P(B|A)$ by using the relation

$$P(B|A) = \frac{P(A|B)P(B)}{P(A)} \quad (\text{A1.14})$$

This relation is a special form of *Bayes' rule*.

Suppose that the conditional probability $P(B|A)$ is simply equal to the elementary probability of occurrence of event B , that is,

$$P(B|A) = P(B)$$

Under this condition, the probability of occurrence of the joint event AB is equal to the product of the elementary probabilities of the events A and B :

$$P(AB) = P(A)P(B)$$

so that

$$P(A|B) = P(A)$$

That is, the conditional probability of event A , assuming the occurrence of event B , is simply equal to the elementary probability of event A . We thus see that in this case a knowledge of the occurrence of one event tells us no more about the probability of occurrence of the other event than we knew without that knowledge. Events A and B that satisfy this condition are said to be *statistically independent*.

A1.2 Random Variables

It is customary, particularly when using the language of sample space, to think of the outcome of an experiment as a variable that can wander over the set of sample points and whose value is determined by the experiment. *A function whose domain is a sample space and whose range is some set of real numbers is called a random variable of the experiment.* However, the term *random variable* is somewhat confusing. First, the word *random* is not used in the sense of equal probability of occurrence, for which it should be reserved. Second, the word *variable* does not imply dependence (on the experimental outcome), which is an essential part of the meaning. Nevertheless, the term is so deeply imbedded in the literature of probability that its usage has persisted.

When the outcome of an experiment is s , the random variable is denoted as $X(s)$ or simply X . For example, the sample space representing the outcomes of the throw of a die is a set of six sample points that may be taken to be the integers 1, 2, ..., 6. Then if we identify the sample point k with the event that k dots show when the die is thrown, the function $X(k) = k$ is a random variable such that $X(k)$ equals the number of dots that show when the die is thrown. In this example, the random variable takes only a discrete set of values. In such a case, we say that we are dealing with a *discrete random variable*. More precisely, the random variable X can take only a finite number of values in any finite observation interval. If, however, the random variable X can take any value in a whole observation interval, X is called a *continuous random variable*. For example, the random variable that represents the amplitude of a noise voltage at a particular instant of time is a continuous random variable because it may take any value between plus and minus infinity.

To proceed further, we need a probabilistic description of random variables that works equally well for discrete as well as continuous random variables. Let us consider the random variable X and the probability of the event $X \leq x$. We denote this probability by $P(X \leq x)$. It is apparent that this probability is a function of the *dummy variable* x . To simplify the notation, we write

$$F_X(x) = P(X \leq x) \quad (\text{A1.15})$$

The function $F_X(x)$ is called the *cumulative distribution function* (cdf) or simply the *distribution function* of the random variable X . Note that $F_X(x)$ is a function of x , not of the random variable X . However, it depends on the assignment of the random variable X , which accounts for the use of X as subscript. For any point x , the distribution function $F_X(x)$ expresses a probability.

The distribution function $F_X(x)$ has the following properties, which follow directly from Equation (A1.15):

1. The distribution function $F_X(x)$ is bounded between zero and one.
2. The distribution function $F_X(x)$ is a nondecreasing function of x ; that is,

$$F_X(x_1) \leq F_X(x_2) \quad \text{if } x_1 < x_2 \quad (\text{A1.16})$$

An alternative description of the probability of the random variable X is often useful. This is the derivative of the distribution function, as shown by

$$f_X(x) = \frac{d}{dx} F_X(x) \quad (\text{A1.17})$$

which is called the *probability density function* (pdf) of the random variable X . Note that the differentiation in Equation (A1.17) is with respect to the dummy variable x . The name, density function, arises from the fact that the probability of the event $x_1 < X \leq x_2$ equals

$$\begin{aligned} P(x_1 < X \leq x_2) &= P(X \leq x_2) - P(X \leq x_1) \\ &= F_X(x_2) - F_X(x_1) \\ &= \int_{x_1}^{x_2} f_X(x) dx \end{aligned} \quad (\text{A1.18})$$

The probability of an interval is therefore the area under the probability density function in that interval. Putting $x_1 = -\infty$ in Equation (A1.18), and changing the notation somewhat, we readily see that the distribution function is defined in terms of the probability density function as follows:

$$F_X(x) = \int_{-\infty}^x f_X(\xi) d\xi \quad (\text{A1.19})$$

Since $F_X(\infty) = 1$, corresponding to the probability of a certain event, and $F_X(-\infty) = 0$, corresponding to the probability of an impossible event, we readily find from Equation (A1.18) that

$$\int_{-\infty}^{\infty} f_X(x) dx = 1 \quad (\text{A1.20})$$

Earlier we mentioned that a distribution function must always be nondecreasing. This means that its derivative or the probability density function must always be nonnegative. Accordingly, we may state that *a probability density function must always be a nonnegative function, and with a total area of one.*

Thus far we have focused attention on situations involving a single random variable. However, we find frequently that the outcome of an experiment requires several random variables for its description. We now consider situations involving two random variables. The probabilistic description developed in this way may be readily extended to any number of random variables.

Consider two random variables X and Y . We define the *joint distribution function* $F_{X,Y}(x, y)$ as the probability that the random variable X is less than or equal to a specified value x and that the random variable Y is less than or equal to a specified value y . The variables X and Y may be two separate one-dimensional random variables or the components of a single two-dimensional random variable. In either case, the joint sample space is the xy -plane. The joint distribution function $F_{X,Y}(x, y)$ is the probability that the outcome of an experiment will result in a sample point lying inside the quadrant $(-\infty < X \leq x, -\infty < Y \leq y)$ of the joint sample space. That is,

$$F_{X,Y}(x, y) = P(X \leq x, Y \leq y) \quad (\text{A1.21})$$

Suppose that the joint distribution function $F_{X,Y}(x, y)$ is continuous everywhere, and that the partial derivative

$$f_{X,Y}(x, y) = \frac{\partial^2 F_{X,Y}(x, y)}{\partial x \partial y} \quad (\text{A1.22})$$

exists and is continuous everywhere. We call the function $f_{X,Y}(x, y)$ the *joint probability density function* of the random variables X and Y . The joint distribution function

$F_{X,Y}(x, y)$ is a nondecreasing function of both x and y . Therefore, from Equation (A1.22) it follows that the joint probability density function $f_{X,Y}(x, y)$ is always nonnegative. Also the total volume under the graph of a joint probability density function must be unity, as shown by

$$\int_{-\infty}^{\infty} \int_{-\infty}^{\infty} f_{X,Y}(\xi, \eta) d\xi d\eta = 1 \quad (\text{A1.23})$$

The probability density function for a single random variable (X , say) can be obtained from its joint probability density function with a second random variable (Y , say) in the following way. We first note that

$$F_X(x) = \int_{-\infty}^{\infty} \int_{-\infty}^x f_{X,Y}(\xi, \eta) d\xi d\eta \quad (\text{A1.24})$$

Therefore, differentiating both sides of Equation (A1.24) with respect to x , we get the desired relation:

$$f_X(x) = \int_{-\infty}^{\infty} f_{X,Y}(x, \eta) d\eta \quad (\text{A1.25})$$

Thus the probability density function $f_X(x)$ is obtained from the joint probability density function $f_{X,Y}(x, y)$ by simply integrating it over all possible values of the undesired random variable Y . The use of similar arguments in the other dimension yields $f_Y(y)$. The probability density functions $f_X(x)$ and $f_Y(y)$ are called *marginal densities*. Hence, the joint probability density function $f_{X,Y}(x, y)$ contains all the possible information about the joint random variables X and Y .

Suppose that X and Y are two continuous random variables with joint probability density function $f_{X,Y}(x, y)$. The *conditional probability density function* of Y given that $X = x$ is defined by

$$f_Y(y|x) = \frac{f_{X,Y}(x, y)}{f_X(x)} \quad (\text{A1.26})$$

provided that $f_X(x) > 0$, where $f_X(x)$ is the marginal density of X . The function $f_Y(y|x)$ may be thought of as a function of the variable y , with the variable x arbitrary, but *fixed*. Accordingly, it satisfies all the requirements of an ordinary probability density function, as shown by

$$f_Y(y|x) \geq 0$$

and

$$\int_{-\infty}^{\infty} f_Y(y|x) dy = 1$$

If the random variables X and Y are *statistically independent*, then knowledge of the outcome of X can in no way affect the distribution of Y . The result is that the conditional probability density function $f_Y(y|x)$ reduces to the marginal density $f_Y(y)$, as shown by

$$f_Y(y|x) = f_Y(y)$$

In such a case, we may express the joint probability density function of the random variables X and Y as the product of their respective marginal densities, as shown by

$$f_{X,Y}(x, y) = f_X(x)f_Y(y)$$

In words, we may state that if the joint probability density function of the random variables X and Y equals the product of their marginal densities, then X and Y are statistically independent.

A1.3 Statistical Averages

Having discussed probability and some of its ramifications, we now seek ways for determining the *average* behavior of the outcomes arising in random experiments.

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

$$\mu_X = E[X] = \int_{-\infty}^{\infty} x f_X(x) dx \quad (\text{A1.27})$$

where E denotes the *statistical expectation operator*. That is, the mean μ_X locates the center of gravity of the area under the probability density curve of the random variable X . To interpret the expected value μ_X , we write the integral in the defining Equation (A1.27) as the limit of an approximating sum formulated as follows. Let $\{x_k | k = 0, \pm 1, \pm 2, \dots\}$ denote a set of uniformly spaced points on the real line:

$$x_k = \left(k + \frac{1}{2}\right) \Delta, \quad k = 0, \pm 1, \pm 2, \dots \quad (\text{A1.28})$$

where Δ is the spacing between adjacent points. We may then rewrite Equation (A1.27) as the limiting form of a sum:

$$\begin{aligned} E[X] &= \lim_{\Delta \rightarrow 0} \sum_{k=-\infty}^{\infty} \int_{k\Delta}^{(k+1)\Delta} x_k f_X(x) dx \\ &= \lim_{\Delta \rightarrow 0} \sum_{k=-\infty}^{\infty} x_k P\left(x_k - \frac{\Delta}{2} < X \leq x_k + \frac{\Delta}{2}\right) \end{aligned} \quad (\text{A1.29})$$

For a physical interpretation of the sum on the right-hand side of Equation (A1.29), suppose that we make n independent observations of the random variable X . Let $N_n(k)$ denote the number of times that the random variable X falls inside the k th bin:

$$x_k - \frac{\Delta}{2} < X \leq x_k + \frac{\Delta}{2}, \quad k = 0, \pm 1, \pm 2, \dots$$

Then, as the number of observations, n , is made large, the ratio $N_n(k)/n$ approaches the probability $P(x_k - \Delta/2 < X \leq x_k + \Delta/2)$. Accordingly, we may approximate the expected value of the random variable X as

$$\begin{aligned} E[X] &\approx \sum_{k=-\infty}^{\infty} x_k \left(\frac{N_n(k)}{n} \right) \\ &= \frac{1}{n} \sum_{k=-\infty}^{\infty} x_k N_n(k), \quad n \text{ large} \end{aligned} \quad (\text{A1.30})$$

We now recognize the quantity on the right-hand side of Equation (A1.30) simply as the *sample average*. The sum is taken over all the values x_k , each of which is weighted by the number of times it occurs; the sum is then divided by the total number of observations to give the sample average. Indeed, Equation (A1.30) provides the basis for computing the expected value $E[X]$.

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

$$Y = g(X) \quad (\text{A1.31})$$

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

$$E[g(X)] = \int_{-\infty}^{\infty} g(x) f_X(x) dx \quad (\text{A1.32})$$

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 n th *moment* of the probability distribution of the random variable X ; that is,

$$E[X^n] = \int_{-\infty}^{\infty} x^n f_X(x) dx \quad (\text{A1.33})$$

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 :

$$E[X^2] = \int_{-\infty}^{\infty} x^2 f_X(x) dx \quad (\text{A1.34})$$

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 n th central moment is

$$E[(X - \mu_X)^n] = \int_{-\infty}^{\infty} (x - \mu_X)^n f_X(x) dx \quad (\text{A1.35})$$

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

$$\text{var}[X] = E[(X - \mu_X)^2] = \int_{-\infty}^{\infty} (x - \mu_X)^2 f_X(x) dx \quad (\text{A1.36})$$

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 .

A precise statement of this constraint is due to Chebyshev. The *Chebyshev inequality* states that for any positive number ϵ , we have

$$P(|X - \mu_X| \geq \epsilon) \leq \frac{\sigma_X^2}{\epsilon^2} \quad (\text{A1.37})$$

From this inequality we see that the mean and variance of a random variable give a *partial description* of its probability distribution, hence their common use in practice.

We note from Equations (A1.34) and (A1.36) that the variance σ_X^2 and mean-square value $E[X^2]$ are related by

$$\begin{aligned} \sigma_X^2 &= E[X^2 - 2\mu_X X + \mu_X^2] \\ &= E[X^2] - 2\mu_X E[X] + \mu_X^2 \\ &= E[X^2] - \mu_X^2 \end{aligned} \quad (\text{A1.38})$$

where, in the second line, we have used the *linearity* property of the statistical expectation operator E . Equation (A1.38) shows that if the mean μ_X is zero, then the variance σ_X^2 and the mean-square value $E[X^2]$ of the random variable X are equal.

■ CHARACTERISTIC FUNCTION

Another important statistical average is the *characteristic function* $\phi_X(\nu)$ of the probability distribution of the random variable X , which is defined as the expectation of the complex exponential function $\exp(j\nu X)$, as shown by

$$\begin{aligned} \phi_X(\nu) &= E[\exp(j\nu X)] \\ &= \int_{-\infty}^{\infty} f_X(x) \exp(j\nu x) dx \end{aligned} \quad (\text{A1.39})$$

where ν is real and $j = \sqrt{-1}$. In other words, the characteristic function $\phi_X(\nu)$ is (except for a sign change in the exponent) the Fourier transform of the probability density function $f_X(x)$; the Fourier transform is reviewed in Appendix 2. In this relation we have used $\exp(j\nu x)$ rather than $\exp(-j\nu x)$, so as to conform with the convention adopted in probability theory. Recognizing that ν and x play analogous roles to the variables $2\pi f$ and t of Fourier transforms, respectively, we deduce the following inverse relation from analogy with the inverse Fourier transform:

$$f_X(x) = \frac{1}{2\pi} \int_{-\infty}^{\infty} \phi_X(\nu) \exp(-j\nu x) d\nu \quad (\text{A1.40})$$

This relation may be used to evaluate the probability density function $f_X(x)$ of the random variable X from its characteristic function $\phi_X(\nu)$.

■ JOINT MOMENTS

Consider next a pair of random variables X and Y . A set of statistical averages of importance in this case is the *joint moments*, namely, the expected value of $X^i Y^k$, where i and k may assume any positive integer values. We may thus write

$$E[X^i Y^k] = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} x^i y^k f_{X,Y}(x, y) dx dy \quad (\text{A1.41})$$

A joint moment of particular importance is the *correlation* defined by $E[XY]$, which corresponds to $i = k = 1$ in Equation (A1.41).

The correlation of the centered random variables $X - E[X]$ and $Y - E[Y]$, that is, the joint moment

$$\text{cov}[XY] = E[(X - E[X])(Y - E[Y])] \quad (\text{A1.42})$$

is called the *covariance* of X and Y . Letting $\mu_X = E[X]$ and $\mu_Y = E[Y]$, we may expand Equation (A1.42) to obtain the result

$$\text{cov}[XY] = E[XY] - \mu_X \mu_Y \quad (\text{A1.43})$$

Let σ_X^2 and σ_Y^2 denote the variances of X and Y , respectively. Then the covariance of X and Y , normalized with respect to $\sigma_X \sigma_Y$, is called the *correlation coefficient* of X and Y :

$$\rho = \frac{\text{cov}[XY]}{\sigma_X \sigma_Y} \quad (\text{A1.44})$$

We say that the two random variables X and Y are *uncorrelated* if and only if their covariance is zero, that is, if and only if

$$\text{cov}[XY] = 0$$

We say that they are *orthogonal* if and only if their correlation is zero, that is, if and only if

$$E[XY] = 0$$

From Equation (A1.43) we observe that if one of the random variables X and Y or both have zero means, and if they are orthogonal, then they are uncorrelated, and vice versa. Note also that if X and Y are statistically independent, then they are uncorrelated; however, the converse of this statement is not necessarily true.

REPRESENTATION OF SIGNALS AND SYSTEMS

A2.1 Fourier Analysis

Let $g(t)$ denote a *nonperiodic deterministic signal*, expressed as some function of time t . By definition, the *Fourier transform* of the signal $g(t)$ is given by the integral

$$G(f) = \int_{-\infty}^{\infty} g(t) \exp(-j2\pi ft) dt \quad (\text{A2.1})$$

where $j = \sqrt{-1}$, and the variable f denotes *frequency*. Given the Fourier transform $G(f)$, the original signal $g(t)$ is recovered exactly using the formula for the *inverse Fourier transform*:

$$g(t) = \int_{-\infty}^{\infty} G(f) \exp(j2\pi ft) df \quad (\text{A2.2})$$

Note that in Equations (A2.1) and (A2.2) we have used a lowercase letter to denote the time function and an uppercase letter to denote the corresponding frequency function. The functions $g(f)$ and $G(f)$ are said to constitute a Fourier-transform pair.

For the Fourier transform of a signal $g(t)$ to exist, it is sufficient but not necessary that $g(t)$ satisfies three conditions known collectively as *Dirichlet's conditions*:

1. The function $g(t)$ is single-valued, with a finite number of maxima and minima in any finite time interval.
2. The function $g(t)$ has a finite number of discontinuities in any finite time interval.
3. The function $g(t)$ is absolutely integrable, that is,

$$\int_{-\infty}^{\infty} |g(t)| dt < \infty$$

We may safely ignore the question of the existence of the Fourier transform of a time function $g(t)$ when it is an accurately specified description of a physically realizable signal. In other words, physical realizability is a sufficient condition for the existence of a Fourier transform. Indeed, we may go one step further and state that all energy signals, that is, signals $g(t)$ for which

$$\int_{-\infty}^{\infty} |g(t)|^2 dt < \infty$$

are Fourier transformable.

The Fourier transform provides the mathematical tool for measuring the frequency content, or spectrum, of a signal. For this reason, the terms *Fourier transform* and *spectrum* are often used interchangeably. Thus, given a signal $g(t)$ with Fourier transform $G(f)$, we may refer to $G(f)$ as the spectrum of the signal $g(t)$. By the same token, we refer to $|G(f)|$ as the *magnitude spectrum* of the signal $g(t)$, and refer to $\arg \{G(f)\}$ as its *phase spectrum*.

■ PROPERTIES OF THE FOURIER TRANSFORM

It is useful to have insight into the relationship between a time function $g(t)$ and its Fourier transform $G(f)$, and also into the effects that various operations on the function $g(t)$ have on the transform $G(f)$. This may be achieved by examining certain properties of the Fourier transform, which are summarized in Table A6.2.

■ DIRAC DELTA FUNCTION

Strictly speaking, the theory of the Fourier transform is applicable only to time functions that satisfy the Dirichlet conditions. Such functions include energy signals. However, it would be highly desirable to extend this theory in two ways:

1. To combine the Fourier series and Fourier transform into a unified theory, so that the Fourier series may be treated as a special case of the Fourier transform.
2. To include power signals (i.e., signals for which the average power is finite) in the list of signals to which we may apply the Fourier transform.

It turns out that both of these objectives can be met through the “proper use” of the *Dirac delta function*, or *unit impulse*.

The Dirac delta function or just delta function, denoted by $\delta(t)$, is defined as having zero amplitude everywhere except at $t = 0$, where it is infinitely large in such a way that it contains unit area under its curve; that is,

$$\delta(t) = 0, \quad t \neq 0 \quad (\text{A2.3})$$

and

$$\int_{-\infty}^{\infty} \delta(t) dt = 1 \quad (\text{A2.4})$$

An implication of this pair of relations is that the delta function $\delta(t)$ must be an even function of time t , which is centered at $t = 0$.

For the delta function to have meaning, however, it has to appear as a factor in the integrand of an integral with respect to time and then, strictly speaking, only when the other factor in the integrand is a continuous function of time. Let $g(t)$ be such a function, and consider the product of $g(t)$ and the time-shifted delta function $\delta(t - t_0)$. In light of the two defining equations, Equations (A2.3) and (A2.4), we may express the integral of this product as follows:

$$\int_{-\infty}^{\infty} g(t) \delta(t - t_0) dt = g(t_0) \quad (\text{A2.5})$$

The operation indicated on the left-hand side of this equation sifts out the value $g(t_0)$ of the function $g(t)$ at time $t = t_0$, where $-\infty < t < \infty$. Accordingly, Equation (A2.5) is referred to as the *sifting property* of the delta function. This property is sometimes used as the defining equation of a delta function; in effect, it incorporates Equations (A2.3) and (A2.4) into a single relation.

Noting that the delta function $\delta(t)$ is an even function of t , we may rewrite Equation (A2.5) so as to emphasize its resemblance to the convolution integral, as shown by

$$\int_{-\infty}^{\infty} g(\tau) \delta(t - \tau) d\tau = g(t) \quad (\text{A2.6})$$

In words, the convolution of any function with the delta function leaves that function unchanged. We refer to this statement as the *replication property* of the delta function.

It is important to realize that no function in the ordinary sense has the two properties of Equations (A2.3) and (A2.4) or the equivalent sifting property of Equation (A2.5). However, we can imagine a sequence of functions that have progressively taller and thinner peaks at $t = 0$, with the area under the curve remaining equal to unity, whereas the value of the function tends to zero at every point except $t = 0$, where it tends to infinity. That is, we may view the delta function as *the limiting form of a pulse of unit area as the duration of the pulse approaches zero*. It is immaterial what sort of pulse shape is used.

■ FOURIER TRANSFORMS OF PERIODIC SIGNALS

It is well known that by using the Fourier series, a periodic signal can be represented as a sum of complex exponentials. Also, in a limiting sense, Fourier transforms can be defined by complex exponentials. Therefore, it seems reasonable to represent a periodic signal in terms of a Fourier transform, provided that this transform is permitted to include delta functions.

Consider then a periodic signal $g_{T_0}(t)$ of period T_0 . We can represent $g_{T_0}(t)$ in terms of the *complex exponential Fourier series*:

$$g_{T_0}(t) = \sum_{n=-\infty}^{\infty} c_n \exp(j2\pi n f_0 t) \quad (\text{A2.7})$$

where c_n is the *complex Fourier coefficient* defined by

$$c_n = \frac{1}{T_0} \int_{-T_0/2}^{T_0/2} g_{T_0}(t) \exp(-j2\pi n f_0 t) dt \quad (\text{A2.8})$$

and f_0 is the *fundamental frequency* defined as the reciprocal of the period T_0 ; that is,

$$f_0 = \frac{1}{T_0} \quad (\text{A2.9})$$

Let $g(t)$ be a pulselike function, which equals $g_{T_0}(t)$ over one period and is zero elsewhere; that is

$$g(t) = \begin{cases} g_{T_0}(t), & -\frac{T_0}{2} < t \leq \frac{T_0}{2} \\ 0 & \text{elsewhere} \end{cases} \quad (\text{A2.10})$$

The periodic signal $g_{T_0}(t)$ may now be expressed in terms of the function $g(t)$ as an infinite summation, as shown by

$$g_{T_0}(t) = \sum_{m=-\infty}^{\infty} g(t - mT_0) \quad (\text{A2.11})$$

Based on this representation, we may view $g(t)$ as a *generating function*, which generates the periodic signal $g_{T_0}(t)$.

The function $g(t)$ is Fourier transformable. Accordingly, we may rewrite the formula for the complex Fourier coefficient as follows:

$$\begin{aligned} c_n &= f_0 \int_{-\infty}^{\infty} g(t) \exp(-j2\pi n f_0 t) dt \\ &= f_0 G(nf_0) \end{aligned} \quad (\text{A2.12})$$

where $G(nf_0)$ is the Fourier transform of $g(t)$ evaluated at the frequency nf_0 . We may thus rewrite the formula for the reconstruction of the periodic signal $g_{T_0}(t)$ as

$$g_{T_0}(t) = f_0 \sum_{n=-\infty}^{\infty} G(nf_0) \exp(j2\pi n f_0 t) \quad (\text{A2.13})$$

or, equivalently, in light of Equation (A2.11)

$$\sum_{m=-\infty}^{\infty} g(t - mT_0) = f_0 \sum_{n=-\infty}^{\infty} G(nf_0) \exp(j2\pi n f_0 t) \quad (\text{A2.14})$$

Equation (A2.14) is one form of *Poisson's sum formula*.

It is of interest to observe that the function $g(t)$, which constitutes one period of the periodic signal $g_{T_0}(t)$, has a continuous spectrum defined by $G(f)$. On the other hand, the period signal $g_{T_0}(t)$ itself has a discrete spectrum. We conclude, therefore, that *periodicity in the time domain has the effect of changing the frequency-domain description or spectrum of the signal into a discrete form defined at integer multiples of the fundamental frequency*.

■ FOURIER-TRANSFORM PAIRS

Table A6.3 presents a listing of some commonly used Fourier-transform pairs, the derivations of which follow from the material just presented.

■ TRANSMISSION OF SIGNALS THROUGH LINEAR SYSTEMS

A *system* refers to any physical device that produces an output signal in response to an input signal. It is customary to refer to the input signal as the *excitation* and to the output signal as the *response*. In a *linear* system, the *principle of superposition* holds; that is, the response of a linear system to a number of excitations applied simultaneously is equal to the sum of the responses of the system when each excitation is applied individually.

In the time domain, a linear system is described in terms of its *impulse response*, which is defined as *the response of the system (with zero initial conditions) to a unit impulse or delta function $\delta(t)$ applied to the input of the system*. If the system is *time invariant*, then the shape of the impulse response is the same no matter when the unit impulse is applied to the system. Thus, assuming that the unit impulse or delta function is applied at time $t = 0$, we may denote the impulse response of a linear time-invariant system by $h(t)$. Let this system be subjected to an arbitrary excitation $x(t)$. The response, $y(t)$, of the system is defined in terms of the impulse response $h(t)$ by

$$y(t) = \int_{-\infty}^{\infty} x(\tau) h(t - \tau) d\tau \quad (\text{A2.15})$$

which is called the *convolution integral*. Equivalently, we may write

$$y(t) = \int_{-\infty}^{\infty} h(\tau) x(t - \tau) d\tau \quad (\text{A2.16})$$

Hence, convolution is *commutative*.

In the convolution integral, three different time scales are involved: *excitation time* τ , *response time* t , and *system-memory time* $t - \tau$. This relation is the basis of time-domain analysis of linear time-invariant systems. According to Equation (A2.15), the present value of the response of a linear time-invariant system is a weighted integral over the past history

of the input signal, weighted according to the impulse response of the system. Thus the impulse response acts as a *memory function* for the system.

■ FREQUENCY RESPONSE OF LINEAR TIME-INVARIANT SYSTEMS

Consider a linear time-invariant system of impulse response $h(t)$ driven by a complex exponential input of unit amplitude and frequency f , that is,

$$x(t) = \exp(j2\pi ft)$$

Using this excitation in Equation (A2.16), the response of the system is obtained as

$$\begin{aligned} y(t) &= \int_{-\infty}^{\infty} h(\tau) \exp[j2\pi f(t - \tau)] d\tau \\ &= \exp(j2\pi ft) \int_{-\infty}^{\infty} h(\tau) \exp(-j2\pi f\tau) d\tau \end{aligned} \quad (\text{A2.17})$$

Define the *frequency response* of the system as the Fourier transform of its impulse response, as shown by

$$H(f) = \int_{-\infty}^{\infty} h(t) \exp(-j2\pi ft) dt \quad (\text{A2.18})$$

The integral in the last line of Equation (A2.17) is the same as that of Equation (A2.18), except that τ is used in place of t . Hence, we may rewrite Equation (A2.17) in the form

$$y(t) = H(f) \exp(j2\pi ft) \quad (\text{A2.19})$$

The response of a linear time-invariant system to a complex exponential function of frequency f is, therefore, the same complex exponential function multiplied by a constant coefficient $H(f)$.

The frequency response $H(f)$ is, in general, a complex quantity, so we may express it in the form

$$H(f) = |H(f)| \exp[j\beta(f)] \quad (\text{A2.20})$$

where $|H(f)|$ is called the *magnitude response*, and $\beta(f)$ is the *phase*, or *phase response*. In the special case of a linear system with a real-valued impulse response $h(t)$, the frequency response $H(f)$ exhibits conjugate symmetry, which means that

$$|H(f)| = |H(-f)|$$

and

$$\beta(f) = -\beta(-f)$$

That is, the magnitude response $|H(f)|$ of a linear system with real-valued impulse response is an even function of frequency, whereas the phase $\beta(f)$ is an odd function of frequency.

In some applications, it is preferable to work with the logarithm of $H(f)$ expressed in polar form rather than with $H(f)$ itself. Define the natural logarithm

$$\log H(f) = \alpha(f) + j\beta(f) \quad (\text{A2.21})$$

where

$$\alpha(f) = \log |H(f)| \quad (\text{A2.22})$$

The function $\alpha(f)$ is called the *gain* of the system. It is measured in *nepers*, whereas $\beta(f)$ is measured in *radians*. Equation (A2.21) indicates that the gain $\alpha(f)$ and phase $\beta(f)$ are the real and imaginary parts of the (natural) logarithm of the frequency response $H(f)$, respectively. The gain may also be expressed in *decibels* (dB) by using the definition

$$\alpha'(f) = 20 \log_{10} |H(f)| \quad (\text{A2.23})$$

The two gain functions $\alpha(f)$ and $\alpha'(f)$ are related by

$$\alpha'(f) = 8.69\alpha(f) \quad (\text{A2.24})$$

That is, 1 neper is equal to 8.69 dB.

A2.2 Bandwidth

The time-domain and frequency-domain descriptions of a signal are *inversely* related. In particular, we may make the following important statements:

1. If the time-domain description of a signal is changed, the frequency-domain description of the signal is changed in an *inverse* manner, and vice versa. This inverse relationship prevents arbitrary specifications of a signal in both domains. In other words, *we may specify an arbitrary function of time or an arbitrary spectrum, but we cannot specify both of them together.*
2. If a signal is strictly limited in frequency, the time-domain description of the signal will trail on indefinitely, even though its amplitude may assume a progressively smaller value. We say a signal is *strictly limited in frequency* or *strictly band limited* if its Fourier transform is exactly zero outside a finite band of frequencies. The sinc pulse

$$\text{sinc}(t) = \frac{\sin(\pi t)}{\pi t}$$

is an example of a strictly band-limited signal. It is also *asymptotically limited in time*, which confirms the opening statement we made for a strictly band-limited signal. In an inverse manner, if a signal is *strictly limited in time* (i.e., the signal is exactly zero outside a finite time interval), then the spectrum of the signal is infinite in extent, even though the amplitude spectrum may assume a progressively smaller value. This behavior is exemplified by a rectangular pulse. Accordingly, we may state that a *signal cannot be strictly limited in both time and frequency.*

The *bandwidth* of a signal provides a measure of the *extent of significant spectral content of the signal for positive frequencies*. When the signal is strictly band limited, the bandwidth is well defined. For example, the sinc pulse $\text{sinc}(2Wt)$ has a bandwidth equal to W . However, when the signal is not strictly band limited, as is generally the case, we encounter difficulty in defining the bandwidth of the signal. The difficulty arises because the meaning of “significant” attached to the spectral content of the signal is mathematically imprecise. Consequently, there is no universally accepted definition of bandwidth. Nevertheless, there are some commonly used definitions for bandwidth, as discussed next.

When the spectrum of a signal is symmetric with a *main lobe* bounded by well-defined *nulls* (i.e., frequencies at which the spectrum is zero), we may use the main lobe as the basis for defining the bandwidth of the signal. Specifically, if the signal is *low-pass* (i.e., its spectral content is centered around the origin), the bandwidth is defined as one half the total width of the main spectral lobe since only one half of this lobe lies inside the positive frequency region. For example, a rectangular pulse of duration T seconds has a main

spectral lobe of total width $2/T$ hertz centered at the origin. Accordingly, we may define the bandwidth of this rectangular pulse as $1/T$ hertz. If, on the other hand, the signal is *band-pass* with main spectral lobes centered around $\pm f_c$, where f_c is large enough, the bandwidth is defined as the width of the main lobe for positive frequencies. This definition of bandwidth is called the *null-to-null bandwidth*. For example, an RF pulse of duration T seconds and frequency f_c has main spectral lobes of width $2/T$ hertz centered around $\pm f_c$, where it is assumed that f_c is large compared to $1/T$. Hence, we may define the null-to-null bandwidth of this RF pulse as $2/T$ hertz. On the basis of the definitions presented here, we may state that shifting the spectral content of a low-pass signal by a sufficiently large frequency has the effect of doubling the bandwidth of the signal; such a frequency translation is attained by using modulation.

Another popular definition of bandwidth is the *3-dB bandwidth*. Specifically, if the signal is low-pass, the 3-dB bandwidth is defined as the separation between zero frequency, where the magnitude spectrum attains its peak value, and the *positive frequency*, at which the amplitude spectrum drops to $1/\sqrt{2}$ of its peak value. For example, the decaying exponential $\exp(-at)$ has a 3-dB bandwidth of $a/2\pi$ hertz. If, on the other hand, the signal is band-pass, centered at $\pm f_c$, the 3-dB bandwidth is defined as the separation (along the positive frequency axis) between the two frequencies at which the magnitude spectrum of the signal drops to $1/\sqrt{2}$ of the peak value of f_c . The 3-dB bandwidth has the advantage in that it can be read directly from a plot of the magnitude spectrum. However, it has the disadvantage in that it may be misleading if the magnitude spectrum has slowly decreasing tails.

Yet another measure for the bandwidth of a signal is the *root mean square (rms) bandwidth*, which is defined as the square root of the second moment of a properly normalized form of the squared magnitude spectrum of the signal about a suitably chosen point. We assume that the signal is low-pass, so that the second moment may be taken about the origin. As for the normalized form of the squared magnitude spectrum, we use the nonnegative function

$$\frac{|G(f)|^2}{\int_{-\infty}^{\infty} |G(f)|^2 df}$$

in which the denominator applies the correct normalization in the sense that the integrated value of this ratio over the entire frequency axis is unity. We may thus formally define the rms bandwidth of a low-pass signal $g(t)$ with Fourier transform $G(f)$ as follows:

$$W_{\text{rms}} = \left(\frac{\int_{-\infty}^{\infty} f^2 |G(f)|^2 df}{\int_{-\infty}^{\infty} |G(f)|^2 df} \right)^{1/2} \quad (\text{A2.25})$$

An attractive feature of the rms bandwidth W_{rms} is that it lends itself more readily to mathematical evaluation than the other two definitions of bandwidth, but it is not as easily measurable in the laboratory.

■ TIME-BANDWIDTH PRODUCT

For any family of pulse signals that differ by a time-scaling factor, the product of the signal's duration and its bandwidth is always a constant, as shown by

$$(\text{duration} \times \text{bandwidth}) = \text{constant}$$

The product is called the *time-bandwidth product* or *bandwidth-duration product*. The constancy of the time-bandwidth product is another manifestation of the inverse relationship that exists between the time-domain and frequency-domain descriptions of a signal. In particular, if the duration of a pulse signal is decreased by reducing the time scale by a factor a , the frequency scale of the signal's spectrum, and therefore the bandwidth of the signal, is increased by the same factor a , by virtue of the *time-scaling property* of the Fourier transform, and the time-bandwidth product of the signal is thereby maintained constant; see item 2 of Table A6.2. For example, a rectangular pulse of duration T seconds has a bandwidth (defined on the basis of the positive-frequency part of the main lobe) equal to $1/T$ hertz, making the time-bandwidth product of the pulse equal unity. Whatever definition we use for the bandwidth of a signal, the time-bandwidth product remains constant over certain classes of pulse signals. The choice of a particular definition for bandwidth merely changes the value of the constant.

To be more specific, consider the rms bandwidth defined in Equation (A2.25). The corresponding definition for the *rms duration* of the signal $g(t)$ is

$$T_{\text{rms}} = \left(\frac{\int_{-\infty}^{\infty} t^2 |g(t)|^2 dt}{\int_{-\infty}^{\infty} |g(t)|^2 dt} \right)^{1/2} \quad (\text{A2.26})$$

where it is assumed that the signal $g(t)$ is centered around the origin. It may be shown that, using the rms definitions of Equations (A2.25) and (A2.26), the time-bandwidth product has the following form:

$$T_{\text{rms}} W_{\text{rms}} \geq \frac{1}{4\pi} \quad (\text{A2.27})$$

where the constant is $1/4\pi$. The Gaussian pulse $\exp(-\pi^2 t^2)$ satisfies this condition with the equality sign.

■ NOISE EQUIVALENT BANDWIDTH

The definitions of bandwidth just presented (i.e., 3-dB bandwidth, null-to-null bandwidth, and rms bandwidth) are all formulated in terms of deterministic signals. Another definition of bandwidth that presents itself in the study of random signals and systems is the noise equivalent bandwidth. Suppose that a white noise source of power spectral density $N_0/2$ is connected to the input of the simple RC low-pass filter of Figure A2.1; the corresponding value of the average output noise power is equal to $N_0/(4RC)$. For this filter, the half-power or 3-dB bandwidth is equal to $1/(2\pi RC)$. Here again we find that the average output noise power of the filter is proportional to the bandwidth.

We may generalize this statement to include all kinds of low-pass filters by defining a noise equivalent bandwidth as follows. Suppose that we have a source of white noise of

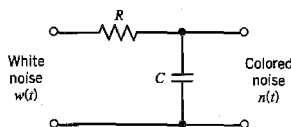


FIGURE A2.1 RC low-pass filter.

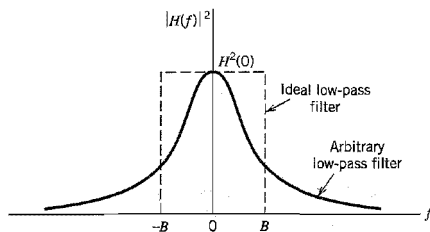


FIGURE A2.2 Illustrating the definition of noise-equivalent bandwidth for a low-pass filter.

zero mean and power spectral density $N_0/2$ connected to the input of an arbitrary low-pass filter of transfer function $H(f)$. The resulting average output noise power is therefore

$$\begin{aligned} N_{\text{out}} &= \frac{N_0}{2} \int_{-\infty}^{\infty} |H(f)|^2 df \\ &= N_0 \int_0^{\infty} |H(f)|^2 df \end{aligned} \quad (\text{A2.28})$$

where, in the last line, we have made use of the fact that the magnitude response $|H(f)|$ is an even function of frequency.

Consider next the same source of white noise connected to the input of an *ideal* low-pass filter of zero-frequency response $H(0)$ and bandwidth B . In this case, the average output noise power is

$$N_{\text{out}} = N_0 B H^2(0) \quad (\text{A2.29})$$

Therefore, equating this average output noise power to that in Equation (A2.28), we may formally define the *noise equivalent bandwidth* as

$$B = \frac{\int_0^{\infty} |H(f)|^2 df}{H^2(0)} \quad (\text{A2.29})$$

Thus the procedure for calculating the noise equivalent bandwidth consists of replacing the arbitrary low-pass filter of transfer function $H(f)$ by an equivalent ideal low-pass filter of zero frequency response $H(0)$ and bandwidth B , as illustrated in Figure A2.2. In a similar way, we may define a noise equivalent bandwidth for bandpass filters.

A2.3 Hilbert Transform

The Fourier transform is particularly useful for evaluating the frequency content of an energy signal or, in a limiting sense, that of a power signal. As such, it provides the mathematical basis for analyzing and designing frequency-selective filters for the separation of signals on the basis of their frequency content. Another method of separating signals is based on *phase selectivity*, which uses phase shifts between the pertinent signals to achieve the desired separation. The simplest phase shift is that of 180 degrees, which is merely a polarity reversal in the case of a sinusoidal signal. Shifting the phase angles of all components of a given signal by 180 degrees requires the use of an *ideal transformer*. Another phase shift of interest is that of ± 90 degrees. In particular, when the phase angles of all

components of a given signal are shifted by ± 90 degrees, the resulting function of time is known as the Hilbert transform of the signal.

To be specific, consider a signal $g(t)$ with Fourier transform $G(f)$. The Hilbert transform of $g(t)$, which we shall denote by $\hat{g}(t)$, is defined by

$$\hat{g}(t) = \frac{1}{\pi} \int_{-\infty}^{\infty} \frac{g(\tau)}{t - \tau} d\tau \quad (\text{A2.31})$$

Clearly, the Hilbert transformation of $g(t)$ is a linear operation. The inverse Hilbert transform, by means of which the original signal $g(t)$ is recovered from $\hat{g}(t)$, is defined by

$$g(t) = -\frac{1}{\pi} \int_{-\infty}^{\infty} \frac{\hat{g}(\tau)}{t - \tau} d\tau \quad (\text{A2.32})$$

The functions $g(t)$ and $\hat{g}(t)$ are said to constitute a Hilbert-transform pair. A short table of Hilbert-transform pairs is given in Table A6.4.

We note from the definition of the Hilbert transform that $\hat{g}(t)$ may be interpreted as the convolution of $g(t)$ with the time function $1/\pi t$. We also know from the convolution theorem that the convolution of two functions in the time domain is transformed into the multiplication of their Fourier transforms in the frequency domain; see item 12 of Table A6.2. For the time function $1/\pi t$, we have (see Table A6.3)

$$\frac{1}{\pi t} \Rightarrow -j \operatorname{sgn}(f) \quad (\text{A2.33})$$

where $\operatorname{sgn}(f)$ is the *signum function* defined in the frequency domain as

$$\operatorname{sgn}(f) = \begin{cases} 1, & f > 0 \\ 0, & f = 0 \\ -1, & f < 0 \end{cases} \quad (\text{A2.34})$$

It follows therefore that the Fourier transform $\hat{G}(f)$ of $\hat{g}(t)$ is given by

$$\hat{G}(f) = -j \operatorname{sgn}(f) G(f) \quad (\text{A2.35})$$

Equation (A2.35) states that given a signal $g(t)$, we may obtain its Hilbert transform $\hat{g}(t)$ by passing $g(t)$ through a linear two-port device whose frequency response is equal to $-j \operatorname{sgn}(f)$. This device may be considered as one that produces a phase shift of -90 degrees for all positive frequencies of the input signal and $+90$ degrees for all negative frequencies, as in Figure A2.3. The amplitudes of all frequency components in the signal, however, are

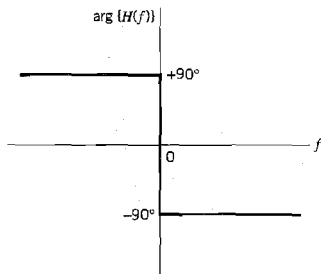


FIGURE A2.3 Phase characteristic of linear two-port device for obtaining the Hilbert transform of a real-valued signal.

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

$$g_+(t) = g(t) + j\hat{g}(t) \quad (\text{A2.36})$$

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

$$G_+(f) = \begin{cases} 2G(f), & f > 0 \\ G(0), & f = 0 \\ 0, & f < 0 \end{cases} \quad (\text{A2.37})$$

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.

From the foregoing analysis it is apparent that for a given signal $g(t)$ we may determine its pre-envelope $g_+(t)$ in one of two equivalent ways:

1. We determine the Hilbert transform $\hat{g}(t)$ of the signal $g(t)$, and then use Equation (A2.36) to compute the pre-envelope $g_+(t)$.
2. We determine the Fourier transform $G(f)$ of the signal $g(t)$, use Equation (A2.37) to determine $G_+(f)$, and then evaluate the inverse Fourier transform of $G_+(f)$ to obtain

$$g_+(t) = 2 \int_0^{\infty} G(f) \exp(j2\pi ft) df \quad (\text{A2.38})$$

For a particular signal $g(t)$ of Fourier transform $G(f)$, one of these two ways may be better than the other.

Equation (A2.36) defines the pre-envelope $g_+(t)$ for positive frequencies. Symmetrically, we may define the pre-envelope for *negative frequencies* as

$$g_-(t) = g(t) - j\hat{g}(t) \quad (\text{A2.39})$$

The two pre-envelopes $g_+(t)$ and $g_-(t)$ are simply the complex conjugate of each other, as shown by

$$g_-(t) = g_+^*(t) \quad (\text{A2.40})$$

where the asterisk denotes complex conjugation. The spectrum of the pre-envelope $g_+(t)$ is nonzero only for *positive* frequencies, as emphasized in Equation (A2.37); hence, the use of a plus sign in the subscript. In contrast, the spectrum of the other pre-envelope $g_-(t)$ is nonzero only for *negative* frequencies, as shown by the Fourier transform

$$G_-(f) = \begin{cases} 0, & f > 0 \\ G(f), & f = 0 \\ 2G(f), & f < 0 \end{cases} \quad (\text{A2.41})$$

Thus the pre-envelopes $g_+(t)$ and $g_-(t)$ constitute a complementary pair of complex-valued signals. Note also that the sum of $g_+(t)$ and $g_-(t)$ is exactly twice the original signal $g(t)$.

■ CANONICAL REPRESENTATIONS OF BAND-PASS SIGNALS

Consider a band-pass signal $g(t)$ whose Fourier transform $G(f)$ is nonnegligible only in a band of frequencies of total extent $2W$, say, centered about some frequency $\pm f_c$. This is illustrated in Figure A2.4a. We refer to f_c as the *carrier frequency*. In the majority of communication signals, we find that the bandwidth $2W$ is small compared with f_c , and so we refer to such a signal as a *narrowband signal*. However, a precise statement about how small the bandwidth must be for the signal to be considered narrowband is not necessary for our present discussion.

Let the pre-envelope of a narrowband signal $g(t)$, with its Fourier transform $G(f)$ centered about some frequency $\pm f_c$, be expressed in the form

$$g_+(t) = \tilde{g}(t) \exp(j2\pi f_c t) \quad (\text{A2.42})$$

We refer to $\tilde{g}(t)$ as the *complex envelope* of the signal. Equation (A2.42) may be viewed as the basis of a definition for the complex envelope $\tilde{g}(t)$ in terms of the pre-envelope $g_+(t)$. We note that the spectrum of $g_+(t)$ is limited to the frequency band $f_c - W \leq f \leq f_c + W$, as illustrated in Figure A2.4b. Therefore, applying the frequency-shifting property of the

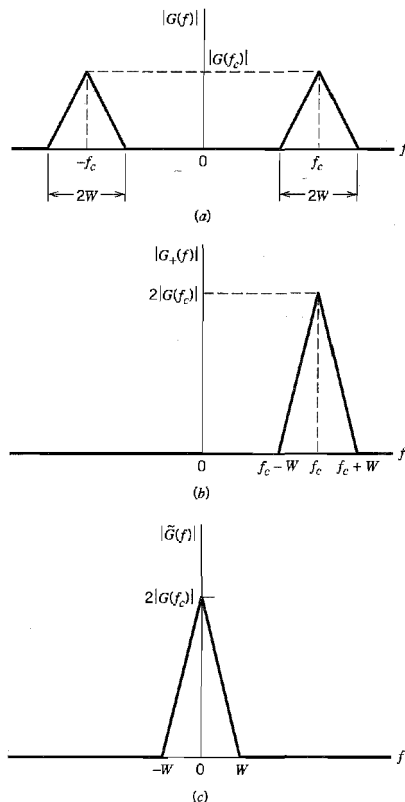


FIGURE A2.4 (a) Magnitude spectrum of band-pass signal $g(t)$. (b) Magnitude spectrum of pre-envelope $g_+(t)$. (c) Magnitude spectrum of complex envelope $\tilde{g}(t)$.

Fourier transform to Equation (A2.42), which is described as item 5 in Table A6.2, we find that the spectrum of the complex envelope $\tilde{g}(t)$ is limited to the band $-W \leq f \leq W$ and centered at the origin as illustrated in Figure A2.4c. That is, the complex envelope $\tilde{g}(t)$ of a band-pass signal $g(t)$ is a *low-pass signal*, which is an important result.

By definition, the given signal $g(t)$ is the real part of the pre-envelope $g_+(t)$. We may thus express the original band-pass signal $g(t)$ in terms of the complex envelope $\tilde{g}(t)$ as follows:

$$g(t) = \text{Re}[\tilde{g}(t) \exp(j2\pi f_c t)] \quad (\text{A2.43})$$

In general, $\tilde{g}(t)$ is a complex-valued quantity; to emphasize this property, we may express it in the form

$$\tilde{g}(t) = g_I(t) + jg_Q(t) \quad (\text{A2.44})$$

where $g_I(t)$ and $g_Q(t)$ are both real-valued low-pass functions; their low-pass property is inherited from the complex envelope $\tilde{g}(t)$. We may therefore use Equations (A2.43) and (A2.44) to express the original band-pass signal $g(t)$ in the *canonical*, or *standard*, form:

$$g(t) = g_I(t) \cos(2\pi f_c t) - g_Q(t) \sin(2\pi f_c t) \quad (\text{A2.45})$$

We refer to $g_I(t)$ as the *in-phase component* of the band-pass signal $g(t)$ and to $g_Q(t)$ as the *quadrature component* of the signal; this nomenclature recognizes that $\sin(2\pi f_c t)$ [i.e., the multiplying factor of $g_Q(t)$] is in phase-quadrature with respect to $\cos(2\pi f_c t)$ [i.e., the multiplying factor of $g_I(t)$] and $\cos(2\pi f_c t)$ is viewed as the reference.

According to Equation (A2.44), the complex envelope $\tilde{g}(t)$ may be pictured as a *time-varying phasor* positioned at the origin of the (g_I, g_Q) -plane, as indicated in Figure A2.5a. With time t varying, the end of the phasor moves about in the plane. Figure A2.5b shows the phasor representation of the complex exponential $\exp(j2\pi f_c t)$. In the definition given in Equation (A2.43), the complex envelope $\tilde{g}(t)$ is multiplied by the complex exponential $\exp(j2\pi f_c t)$. The angles of these two phasors therefore add and their lengths multiply, as shown in Figure A2.5c. Moreover, in this latter figure, we show the (g_I, g_Q) -plane rotating with an angular velocity equal to $2\pi f_c$ radians per second. Thus, in the picture portrayed here, the phasor representing the complex envelope $\tilde{g}(t)$ moves in the (g_I, g_Q) -plane and at the same time the plane itself rotates about the origin. The original band-pass signal $g(t)$ is the projection of this time-varying phasor on a *fixed line* representing the real axis, as indicated in Figure A2.5c.

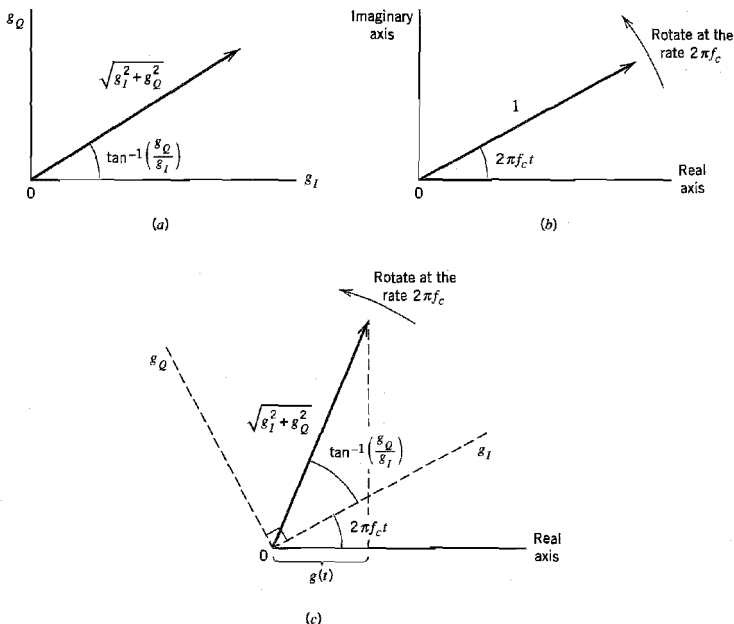


FIGURE A2.5 Illustrating an interpretation of the complex envelope $\tilde{g}(t)$ and its multiplication by $\exp(j2\pi f_c t)$.

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 A2.6a. 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 A2.6b.

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

$$\tilde{g}(t) = a(t) \exp[j\phi(t)] \quad (\text{A2.46})$$

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

$$g(t) = a(t) \cos[2\pi f_c t + \phi(t)] \quad (\text{A2.47})$$

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 $\tilde{g}(t)$.

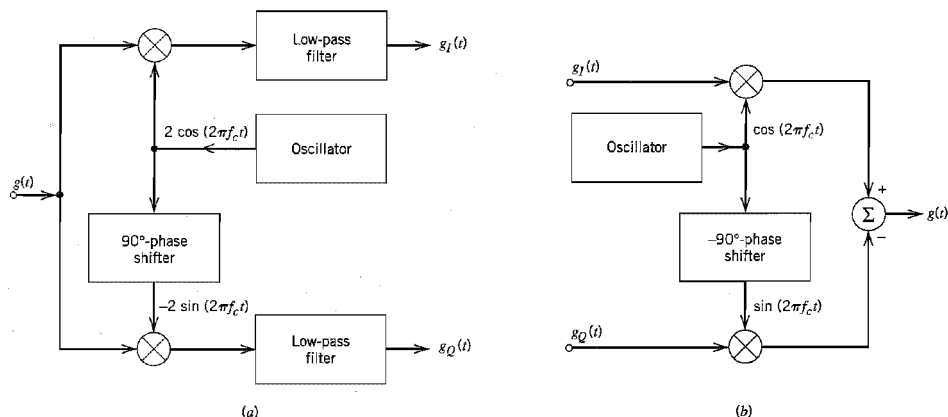


FIGURE A2.6 (a) Scheme for deriving the in-phase and quadrature components of a band-pass signal. (b) Scheme for reconstructing the band-pass signal from its in-phase and quadrature components.

■ TERMINOLOGY

The distinctions among the three different envelopes that we have introduced to describe a band-pass signal $g(t)$ should be carefully noted. We summarize their definitions here:

1. The pre-envelope $g_+(t)$ for positive frequencies is defined by

$$g_+(t) = g(t) + j\hat{g}(t)$$

where $\hat{g}(t)$ is the Hilbert transform of the signal $g(t)$. According to this representation, $\hat{g}(t)$ may be viewed as the quadrature function of $g(t)$. Correspondingly, in the frequency domain we have

$$G_+(f) = \begin{cases} 2G(f), & f > 0 \\ G(0), & f = 0 \\ 0, & f < 0 \end{cases}$$

2. The complex envelope $\tilde{g}(t)$ equals a frequency-shifted version of the pre-envelope $g_+(t)$, as shown by

$$\tilde{g}(t) = g_+(t) \exp(-j2\pi f_c t)$$

where f_c is the carrier frequency of the band-pass signal $g(t)$.

3. The envelope $a(t)$ equals the magnitude of the complex envelope $\tilde{g}(t)$ and also that of the pre-envelope $g_+(t)$, as shown by

$$a(t) = |\tilde{g}(t)| = |g_+(t)|$$

Note that for a band-pass signal $g(t)$, the pre-envelope $g_+(t)$ is a complex band-pass signal whose value depends on the carrier frequency f_c . On the other hand, the envelope $a(t)$ is always a real low-pass signal and, in general, the complex envelope $\tilde{g}(t)$ is a complex low-pass signal; the values of the latter two envelopes are independent of the choice of the carrier frequency f_c . This property gives the complex envelope $\tilde{g}(t)$ an analytic advantage over the original signal $g(t)$.

The envelope $a(t)$ and phase $\phi(t)$ of $g(t)$ are related to the quadrature components $g_I(t)$ and $g_Q(t)$ as follows (see the time-varying phasor representation of Figure A2.5a):

$$\begin{aligned} a(t) &= \sqrt{g_I^2(t) + g_Q^2(t)} \\ \phi(t) &= \tan^{-1} \left(\frac{g_Q(t)}{g_I(t)} \right) \end{aligned}$$

Conversely, we may write

$$\begin{aligned} g_I(t) &= a(t) \cos[\phi(t)] \\ g_Q(t) &= a(t) \sin[\phi(t)] \end{aligned}$$

Thus, each of the quadrature components of a band-pass signal contains both amplitude and phase information. Both components are required for a unique definition of the phase $\phi(t)$, modulo 2π .

■ BAND-PASS SYSTEMS

Now that we know how to handle the complex low-pass representation of band-pass signals, it is logical that we develop a corresponding procedure for handling the analysis of band-pass systems. Specifically, we wish to show that the analysis of band-pass systems can be greatly simplified by establishing an analogy (or, more precisely, an isomorphism)

between low-pass and band-pass systems. This analogy is based on the use of the Hilbert transform for the representation of band-pass signals.

Consider a narrowband signal $x(t)$, with its Fourier transform denoted by $X(f)$. We assume that the spectrum of the signal $x(t)$ is limited to frequencies within $\pm W$ Hz of the carrier frequency f_c . Also, we assume that $W < f_c$. Let this signal be represented in terms of its in-phase and quadrature components as follows:

$$x(t) = x_I(t) \cos(2\pi f_c t) - x_Q(t) \sin(2\pi f_c t) \quad (\text{A2.48})$$

where $x_I(t)$ is the in-phase component and $x_Q(t)$ is the quadrature component. Then, using $\tilde{x}(t)$ to denote the complex envelope of $x(t)$, we may write

$$\tilde{x}(t) = x_I(t) + jx_Q(t) \quad (\text{A2.49})$$

Let the signal $x(t)$ be applied to a linear time-invariant band-pass system with impulse response $h(t)$ and frequency response $H(f)$. We assume that the frequency response of the system is limited to frequencies within $\pm B$ of the carrier frequency f_c . The system bandwidth $2B$ is usually narrower than or equal to the input signal bandwidth $2W$. We wish to represent the band-pass impulse response $h(t)$ in terms of two quadrature components, denoted by $h_I(t)$ and $h_Q(t)$. Thus, by analogy to the representation of band-pass signals, we may express $h(t)$ in the form

$$h(t) = h_I(t) \cos(2\pi f_c t) - h_Q(t) \sin(2\pi f_c t) \quad (\text{A2.50})$$

Define the *complex impulse response* of the band-pass system as

$$\tilde{h}(t) = h_I(t) + jh_Q(t) \quad (\text{A2.51})$$

Hence, we have the complex representation

$$h(t) = \text{Re}[\tilde{h}(t) \exp(j2\pi f_c t)] \quad (\text{A2.52})$$

Note that $h_I(t)$, $h_Q(t)$, and $\tilde{h}(t)$ are all low-pass functions limited to the frequency band $-B \leq f \leq B$.

We may determine the complex impulse response $\tilde{h}(t)$ in terms of the quadrature components $h_I(t)$ and $h_Q(t)$ of the band-pass impulse response $h(t)$ by using Equation (A2.51). Alternatively, we may determine it from the band-pass frequency response $H(f)$ in the following way. We first note from Equation (A2.52) that

$$2h(t) = \tilde{h}(t) \exp(j2\pi f_c t) + \tilde{h}^*(t) \exp(-j2\pi f_c t) \quad (\text{A2.53})$$

where $\tilde{h}^*(t)$ is the complex conjugate of $\tilde{h}(t)$. Therefore, applying the Fourier transform to Equation (A2.53), and using the complex-conjugation property of the Fourier transform, which is described in item 10 in Table A6.2, we get

$$2H(f) = \tilde{H}(f - f_c) + \tilde{H}^*(-f - f_c) \quad (\text{A2.54})$$

where $H(f)$ is the Fourier transform of $h(t)$, and $\tilde{H}(f)$ is the Fourier transform of $\tilde{h}(t)$. Equation (A2.54) satisfies the requirement that $H^*(f) = H(-f)$ for a real impulse response $h(t)$. Since $\tilde{H}(f)$ represents a low-pass frequency response limited to $|f| \leq B$ with $B < f_c$, we deduce from Equation (A2.54) that

$$\tilde{H}(f - f_c) = 2H(f), \quad f > 0 \quad (\text{A2.55})$$

Equation (A2.55) indicates that for a specified band-pass frequency response $H(f)$, we may determine $\tilde{H}(f)$ by taking the part of $H(f)$ corresponding to positive frequencies,

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

$$\tilde{h}(t) = \int_{-\infty}^{\infty} \tilde{H}(f) \exp(j2\pi ft) df \quad (\text{A2.56})$$

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 f_c . 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 , say, simply corresponds to absorbing (or removing) the factor $\exp(\pm j2\pi \Delta f_c t)$ in 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:

$$y(t) = \text{Re}[\tilde{y}(t) \exp(j2\pi f_c t)] \quad (\text{A2.57})$$

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

$$y(t) = \int_{-\infty}^{\infty} h(\tau) x(t - \tau) d\tau \quad (\text{A2.58})$$

In terms of pre-envelopes, we have $h(t) = \text{Re}[h_+(t)]$ and $x(t) = \text{Re}[x_+(t)]$. We may therefore rewrite Equation (A2.58) in terms of the pre-envelopes $x_+(t)$ and $h_+(t)$ as follows:

$$y(t) = \int_{-\infty}^{\infty} \text{Re}[h_+(\tau)] \text{Re}[x_+(t - \tau)] d\tau \quad (\text{A2.59})$$

To proceed further, we make use of a basic property of pre-envelopes that is described by the following relation (presented here without proof):

$$\int_{-\infty}^{\infty} \text{Re}[h_+(\tau)] \text{Re}[x_+(t - \tau)] d\tau = \frac{1}{2} \text{Re} \left[\int_{-\infty}^{\infty} h_+(\tau) x_+^*(\tau) d\tau \right] \quad (\text{A2.60})$$

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

$$\begin{aligned} y(t) &= \frac{1}{2} \text{Re} \left[\int_{-\infty}^{\infty} h_+(\tau) x_+(t - \tau) d\tau \right] \\ &= \frac{1}{2} \text{Re} \left[\int_{-\infty}^{\infty} \tilde{h}(\tau) \exp(j2\pi f_c \tau) \tilde{x}(t - \tau) \exp(j2\pi f_c (t - \tau)) d\tau \right] \\ &= \frac{1}{2} \text{Re} \left[\exp(j2\pi f_c t) \int_{-\infty}^{\infty} \tilde{h}(\tau) \tilde{x}(t - \tau) d\tau \right] \end{aligned} \quad (\text{A2.61})$$

Thus comparing the right-hand sides of Equations (A2.57) and (A2.61), we readily deduce that for a large enough carrier frequency f_c , the complex envelope $\tilde{y}(t)$ of the output signal is related to the complex envelope $\tilde{x}(t)$ of the input signal and the complex impulse response $\tilde{h}(t)$ of the band-pass system as follows:

$$2\tilde{y}(t) = \int_{-\infty}^{\infty} \tilde{h}(\tau) \tilde{x}(t - \tau) d\tau \quad (\text{A2.62})$$

or, using the shorthand notation for convolution,

$$2\tilde{y}(t) = \tilde{h}(t) \star \tilde{x}(t) \quad (\text{A2.63})$$

where \star denotes convolution. In other words, *except for the scaling factor 2, the complex envelope $\tilde{y}(t)$ of the output signal of a band-pass system is obtained by convolving the complex impulse response $\tilde{h}(t)$ of the system with the complex envelope $\tilde{x}(t)$ of the input band-pass signal.* Equation (A2.63) is the result of the isomorphism, for convolution, between a band-pass function and the corresponding low-pass function.

The significance of this result is that in dealing with band-pass signals and systems, we need only concern ourselves with the low-pass functions $\tilde{x}(t)$, $\tilde{y}(t)$, and $\tilde{h}(t)$, representing the excitation, the response, and the system, respectively. That is, the analysis of a band-pass system, which is complicated by the presence of the multiplying factor $\exp(j2\pi f_c t)$, is replaced by an equivalent but much simpler low-pass analysis that completely retains the essence of the filtering process. This procedure is illustrated schematically in Figure A2.7.

The complex envelope $\tilde{x}(t)$ of the input band-pass signal and the complex impulse response $\tilde{h}(t)$ of the band-pass system are defined in terms of their respective in-phase and quadrature components by Equations (A2.49) and (A2.51), respectively. Substituting these relations in Equation (A2.63), we get

$$2\tilde{y}(t) = [h_I(t) + jh_Q(t)] \star [x_I(t) + jx_Q(t)] \quad (\text{A2.64})$$

Because convolution is *distributive*, we may rewrite Equation (A2.64) in the equivalent form

$$2\tilde{y}(t) = [h_I(t) \star x_I(t) - h_Q(t) \star x_Q(t)] + j[h_Q(t) \star x_I(t) + h_I(t) \star x_Q(t)] \quad (\text{A2.65})$$

Let the complex envelope $\tilde{y}(t)$ of the response be defined in terms of its in-phase and quadrature components as

$$\tilde{y}(t) = y_I(t) + jy_Q(t) \quad (\text{A2.66})$$

Comparing the real and imaginary parts in Equations (A2.65) and (A2.66), we have for the in-phase component $y_I(t)$ the relation

$$2y_I(t) = h_I(t) \star x_I(t) - h_Q(t) \star x_Q(t) \quad (\text{A2.67})$$

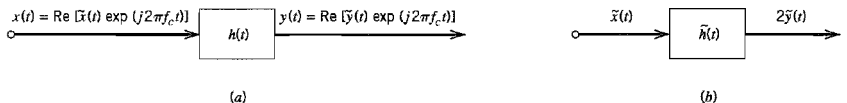


FIGURE A2.7 (a) Narrowband filter of impulse response $h(t)$ with narrowband input signal $x(t)$. (b) Equivalent low-pass filter of complex impulse response $\tilde{h}(t)$ with complex low-pass input $\tilde{x}(t)$.

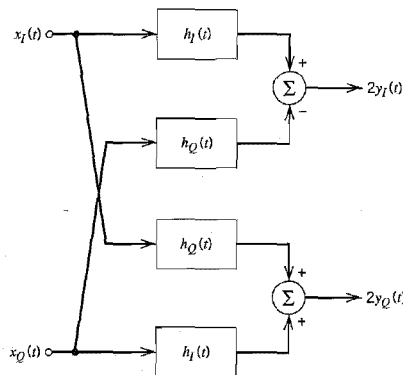


FIGURE A2.8 Block diagram illustrating the relationships between the in-phase and quadrature components of the response of a band-pass filter and those of the input signal.

and for the quadrature component $y_Q(t)$ the relation

$$2y_Q(t) = h_Q(t) \star x_I(t) + h_I(t) \star x_Q(t) \quad (\text{A2.68})$$

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 $\tilde{x}(t)$, which is related to $x(t)$ by

$$x(t) = \text{Re}[\tilde{x}(t) \exp(j2\pi f_c t)]$$

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

$$2\tilde{y}(t) = \tilde{h}(t) \star \tilde{x}(t)$$

4. The desired output $y(t)$ is finally derived from the complex envelope $\tilde{y}(t)$ by using the relation

$$y(t) = \text{Re}[\tilde{y}(t) \exp(j2\pi f_c t)]$$

BESSEL FUNCTIONS

A3.1 Series Solution of Bessel's Equation

In its most basic form, *Bessel's equation of order n* is written as

$$x^2 \frac{d^2 y}{dx^2} + x \frac{dy}{dx} + (x^2 - n^2)y = 0 \quad (\text{A3.1})$$

which is one of the most important of all variable-coefficient differential equations.¹ For each n , a solution of this equation is defined by the power series

$$J_n(x) = \sum_{m=0}^{\infty} \frac{(-1)^m \left(\frac{1}{2}x\right)^{n+2m}}{m!(n+m)!} \quad (\text{A3.2})$$

The function $J_n(x)$ is called a *Bessel function of the first kind of order n* . Equation (A3.1) has two coefficient functions, namely, $1/x$ and $(1 - n^2/x^2)$. Hence, it has no finite singular points except the origin. It follows therefore that the series expansion of Equation (A3.2) converges for all $x > 0$. Equation (A3.2) may thus be used to numerically calculate $J_n(x)$ for $n = 0, 1, 2, \dots$. Table A6.5 presents values of $J_n(x)$ for different orders n and varying x . It is of interest to note that the graphs of $J_0(x)$ and $J_1(x)$ resemble the graphs of $\cos x$ and $\sin x$, respectively; see the graphs of Figure 2.23 in Chapter 2.

The function $J_n(x)$ may also be expressed in the form of an integral as

$$J_n(x) = \frac{1}{\pi} \int_0^{\pi} \cos(x \sin \theta - n\theta) d\theta \quad (\text{A3.3})$$

or, equivalently,

$$J_n(x) = \frac{1}{2\pi} \int_{-\pi}^{\pi} \exp(jx \sin \theta - jn\theta) d\theta \quad (\text{A3.4})$$

A3.2 Properties of the Bessel Function

The Bessel function $J_n(x)$ has the following properties:

$$1. \quad J_n(x) = (-1)^n J_{-n}(x) \quad (\text{A3.5})$$

To prove this relation, we replace θ by $(\pi - \theta)$ in Equation (A3.3). Then, noting that $\sin(\pi - \theta) = \sin \theta$, we get

$$\begin{aligned} J_n(x) &= \frac{1}{\pi} \int_0^{\pi} \cos(x \sin \theta + n\theta - n\pi) d\theta \\ &= \frac{1}{\pi} \int_0^{\pi} [\cos(n\pi) \cos(x \sin \theta + n\theta) + \sin(n\pi) \sin(x \sin \theta + n\theta)] d\theta \end{aligned}$$

For integer values of n , we have

$$\begin{aligned}\cos(n\pi) &= (-1)^n \\ \sin(n\pi) &= 0\end{aligned}$$

Therefore,

$$J_n(x) = \frac{(-1)^n}{\pi} \int_0^\pi \cos(x \sin \theta + n\theta) d\theta \quad (\text{A3.6})$$

From Equation (A3.3), we also find that by replacing n with $-n$:

$$J_{-n}(x) = \frac{1}{\pi} \int_0^\pi \cos(x \sin \theta + n\theta) d\theta \quad (\text{A3.7})$$

The desired result follows immediately from Equations (A3.6) and (A3.7).

$$2. \quad J_n(x) = (-1)^n J_n(-x) \quad (\text{A3.8})$$

This relation is obtained by replacing x with $-x$ in Equation (A3.3), and then using Equation (A3.6).

$$3. \quad J_{n-1}(x) + J_{n+1}(x) = \frac{2n}{x} J_n(x) \quad (\text{A3.9})$$

This *recurrence formula* is useful in constructing tables of Bessel coefficients; its derivation follows from the power series of Equation (A3.2).

4. For small values of x , we have

$$J_n(x) \approx \frac{x^n}{2^n n!} \quad (\text{A3.10})$$

This relation is obtained simply by retaining the first term in the power series of Equation (A3.2) and ignoring the higher-order terms. Thus, when x is small, we have

$$\begin{aligned}J_0(x) &\approx 1 \\ J_1(x) &\approx \frac{x}{2}\end{aligned} \quad (\text{A3.11})$$

$$J_n(x) \approx 0 \quad \text{for } n > 1$$

5. For large values of x , we have

$$J_n(x) \approx \sqrt{\frac{2}{\pi x}} \cos\left(x - \frac{\pi}{4} - \frac{n\pi}{2}\right) \quad (\text{A3.12})$$

This shows that for large values of x , the Bessel function $J_n(x)$ behaves like a sine wave with progressively decreasing amplitude.

6. With x real and fixed, $J_n(x)$ approaches zero as the order n goes to infinity.

$$7. \quad \sum_{n=-\infty}^{\infty} J_n(x) \exp(jn\phi) = \exp(jx \sin \phi) \quad (\text{A3.13})$$

To prove this property, consider the sum $\sum_{n=-\infty}^{\infty} J_n(x) \exp(jn\phi)$ and use the formula of Equation (A3.4) for $J_n(x)$ to obtain

$$\sum_{n=-\infty}^{\infty} J_n(x) \exp(jn\phi) = \frac{1}{2\pi} \sum_{n=-\infty}^{\infty} \exp(jn\phi) \int_{-\pi}^{\pi} \exp(jx \sin \theta - jn\theta) d\theta$$

Interchanging the order of integration and summation:

$$\sum_{n=-\infty}^{\infty} J_n(x) \exp(jn\phi) = \frac{1}{2\pi} \int_{-\pi}^{\pi} d\theta \exp(jx \sin \theta) \sum_{n=-\infty}^{\infty} \exp[jn(\phi - \theta)] \quad (\text{A3.14})$$

We now invoke the following relation from Fourier transform theory:

$$\delta(\phi) = \frac{1}{2\pi} \sum_{n=-\infty}^{\infty} \exp[jn(\phi)], \quad -\pi \leq \phi \leq \pi \quad (\text{A3.15})$$

where $\delta(\phi)$ is a delta function. Therefore, using Equation (A3.15) in (A3.14) and then applying the sifting property of the delta function, we get

$$\begin{aligned} \sum_{n=-\infty}^{\infty} J_n(x) \exp(jn\phi) &= \int_{-\pi}^{\pi} \exp(jx \sin \theta) \delta(\phi - \theta) d\theta \\ &= \exp(jx \sin \phi) \end{aligned}$$

which is the desired result.

$$8. \quad \sum_{n=-\infty}^{\infty} J_n^2(x) = 1 \quad \text{for all } x \quad (\text{A3.16})$$

To prove this property, we may proceed as follows. We observe that $J_n(x)$ is real. Hence, multiplying Equation (A3.4) by its own complex conjugate and summing over all possible values of n , we get

$$\sum_{n=-\infty}^{\infty} J_n^2(x) = \frac{1}{(2\pi)^2} \sum_{n=-\infty}^{\infty} \int_{-\pi}^{\pi} \int_{-\pi}^{\pi} \exp(jx \sin \theta - jn\theta - jx \sin \phi + jn\phi) d\theta d\phi$$

Interchanging the order of double integration and summation:

$$\begin{aligned} \sum_{n=-\infty}^{\infty} J_n^2(x) &= \\ \frac{1}{(2\pi)^2} \int_{-\pi}^{\pi} \int_{-\pi}^{\pi} d\theta d\phi \exp[jx(\sin \theta - \sin \phi)] \sum_{n=-\infty}^{\infty} \exp[jn(\phi - \theta)] \end{aligned} \quad (\text{A3.17})$$

Using Equation (A3.15) in (A3.17) and then applying the sifting property of the delta function, we finally get

$$\sum_{n=-\infty}^{\infty} J_n^2(x) = \frac{1}{2\pi} \int_{-\pi}^{\pi} d\theta = 1$$

which is the desired result.

Many of these properties of the Bessel function $J_n(x)$ may also be illustrated in numerical terms by referring to Table A6.5.

A3.3 Modified Bessel Function

The modified Bessel equation of order n is written as

$$x^2 \frac{d^2 y}{dx^2} + x \frac{dy}{dx} - (x^2 + n^2)y = 0 \quad (\text{A3.18})$$

With $j^2 = -1$, where j is the square root of -1 , we may rewrite this equation as

$$x^2 \frac{d^2 y}{dx^2} + x \frac{dy}{dx} + (j^2 x^2 - n^2)y = 0$$

From this rewrite it is evident that Equation (A3.18) is nothing but Bessel's equation, namely, Equation (A3.1), with x replaced by jx . Thus replacing x by jx in Equation (A3.2), we get

$$\begin{aligned} J_n(jx) &= \sum_{m=0}^{\infty} \frac{(-1)^m \left(\frac{jx}{2}\right)^{n+2m}}{m!(n+m)!} \\ &= j^n \sum_{m=0}^{\infty} \frac{\left(\frac{x}{2}\right)^{n+2m}}{m!(n+m)!} \end{aligned}$$

Next we note that $J_n(jx)$ multiplied by a constant will still be a solution of Bessel's equation. Accordingly, we multiply $J_n(jx)$ by the constant j^{-n} , obtaining

$$j^{-n} J_n(jx) = \sum_{m=0}^{\infty} \frac{\left(\frac{1}{2}x\right)^{n+2m}}{m!(n+m)!}$$

This new function is called the *modified Bessel function of the first kind of order n* , denoted by $I_n(x)$. We may thus formally express a solution of the modified Bessel equation, Equation (A3.18), as

$$\begin{aligned} I_n(x) &= j^{-n} J_n(jx) \\ &= \sum_{m=0}^{\infty} \frac{\left(\frac{1}{2}x\right)^{n+2m}}{m!(n+m)!} \end{aligned} \quad (\text{A3.19})$$

The modified Bessel function $I_n(x)$ is a monotonically increasing real function of the argument x for all n , as shown in Figure A3.1 for $n = 0, 1, 2$.

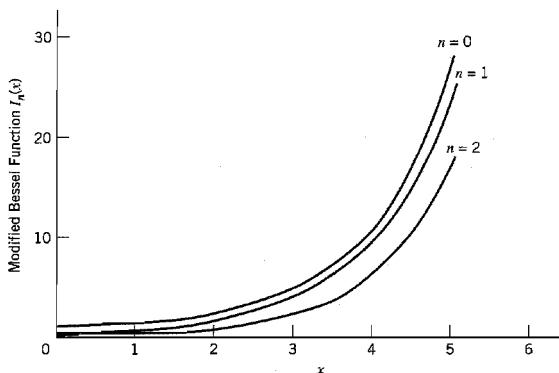


FIGURE A3.1 Modified Bessel function $I_n(x)$ of varying order n .

The modified Bessel function $I_n(x)$ is identical to the original Bessel function $J_n(x)$ except for an important difference: The terms in the series expansion of Equation (A3.19) are all positive, whereas they alternate in sign in the series expansion of Equation (A3.2). The relationship between $J_n(x)$ and $I_n(x)$ is analogous to the way in which the trigonometric functions $\cos x$ and $\sin x$ are related to the hyperbolic functions $\cosh x$ and $\sinh x$.

An interesting property of the modified Bessel function $I_n(x)$ is derived from Equation (A3.13). Specifically, replacing x by jx and the angle ϕ by $\theta - \pi/2$ in this equation, and then invoking the definition of $I_n(x)$ in the first line of Equation (A3.19), we obtain

$$\sum_{n=-\infty}^{\infty} I_n(x) \exp(jn\theta) = \exp(x \cos \theta) \quad (\text{A3.20})$$

From this relation it follows that

$$I_n(x) = \frac{1}{2\pi} \int_{-\pi}^{\pi} \exp(x \cos \theta) \cos(n\theta) d\theta \quad (\text{A3.21})$$

This integral formula for $I_n(x)$ may, of course, also be derived from Equation (A3.4) by making the appropriate changes.

When the argument x is small, we obtain the following asymptotic estimates directly from the series representation of Equation (A3.19):

$$I_0(x) \rightarrow 1 \quad \text{for } x \rightarrow 0 \quad (\text{A3.22})$$

and

$$I_n(x) \rightarrow 0 \quad \text{for } n \geq 1 \text{ and } x \rightarrow 0 \quad (\text{A3.23})$$

For large values of x we have the following asymptotic estimate for $I_n(x)$, which is valid for all integers $n \geq 0$:

$$I_n(x) \approx \frac{\exp(x)}{\sqrt{2\pi x}} \quad \text{for } x \rightarrow \infty \quad (\text{A3.24})$$

Note that this asymptotic behavior of $I_n(x)$ is independent of the order n for large values of x .

NOTES AND REFERENCES

1. Equation (A3.1) is named for the German mathematician and astronomer Friedrich Wilhelm Bessel (1784–1846). For detailed treatments of the solution to this equation and related issues, see Wylie and Barrett (1982) and Watson (1966).

CONFLUENT HYPERGEOMETRIC FUNCTIONS

A4.1 Kummer's Equation

The confluent hypergeometric function¹ is a solution of Kummer's differential equation:

$$x \frac{d^2 y}{dx^2} + (b - x) \frac{dy}{dx} - ay = 0 \quad (\text{A4.1})$$

where, in general, the parameters a and b are complex numbers. For the case when $b \neq 0, -1, -2, \dots$, the solution of Kummer's equation is defined by the series

$${}_1F_1(a; b; x) = 1 + \frac{a}{b} \frac{x}{1!} + \frac{a(a+1)}{b(b+1)} \frac{x^2}{2!} + \dots \quad (\text{A4.2})$$

where ${}_1F_1(a; b; x)$ denotes a confluent hypergeometric function parameterized by a and b . In this notation, the first subscript denotes the number of factorials in the numerator of the general term in Equation (A4.2), the second subscript denotes the number of factorials, apart from $n!$, in the denominator. In Equation (A4.2), both subscripts are clearly 1.

A4.2 Properties of the Confluent Hypergeometric Function

Property 1

For small values of x , the confluent hypergeometric function approximates as

$${}_1F_1(a; b; x) \approx 1 + \frac{a}{b} x \quad \text{for } x \rightarrow 0 \quad (\text{A4.3})$$

This property follows directly from the series expansion of Equation (A4.2).

Property 2

For $a = -1$ and $b = 1$ we have the exact identity:

$${}_1F_1(-1; 1; x) = 1 - x \quad \text{for all } x \quad (\text{A4.4})$$

This property also follows directly from the series expansion of Equation (A4.2).

Property 3

The confluent hypergeometric function for $a = -1/2$ and $b = 1$ is related exactly to the modified Bessel function for all x as follows:

$${}_1F_1\left(-\frac{1}{2}; 1; -x\right) = \exp\left(-\frac{x}{2}\right) \left((1+x)I_0\left(\frac{x}{2}\right) + xI_2\left(\frac{x}{2}\right) \right) \quad (\text{A4.5})$$

where $I_n(x)$ is the modified Bessel function of order n .

A special case of Equation (A4.5) occurs when x is large. From the definition of the modified Bessel function given in Appendix 3, we have the following asymptotic formula for large x :

$$I_n(x) \simeq \frac{\exp(x)}{\sqrt{2\pi x}} \quad \text{for } x \rightarrow \infty \quad (\text{A4.6})$$

Hence, combining Equations (A4.5) and (A4.6), we obtain the simple result

$${}_1F_1\left(-\frac{1}{2}; 1; -x\right) \simeq 2\sqrt{\frac{x}{\pi}} \quad \text{for } x \rightarrow \infty \quad (\text{A4.7})$$

NOTES AND REFERENCES

1. For a discussion of confluent hypergeometric functions, see Jeffreys and Jeffreys (1956). Tabulated values of these functions are presented in Abramowitz and Stegun (1965).

CRYPTOGRAPHY

Secrecy is certainly important to the security or integrity of information transmission. Indeed, the need for secure communications is more profound than ever, recognizing that the conduct of much of our commerce, business, and personal matters is being carried out today through the medium of computers, which has replaced the traditional medium of papers.

Cryptology is the umbrella term used to describe the science of secret communications; it is derived from the Greek *kryptos* and *logos* which mean “hidden” and “word,” respectively.¹ The subject matter of cryptology may be partitioned neatly into *cryptography* and *cryptanalysis*. Cryptography deals with the transformations of a message into coded form by *encryption* and the recovery of the original message by *decryption*. The original message to be encrypted (enciphered) is called the *plaintext*, and the result produced by encryption is called a *cryptogram* or *ciphertext*; the latter two terms are used interchangeably. The set of data transformations used to do the encryption is called a *cipher*; normally, the transformations are parameterized by one or more *keys*. *Cryptanalysis*, on the other hand, deals with how to undo cryptographic communications by breaking a cipher or forging coded signals that may be accepted as genuine.

Cryptographic systems offer three important services:

1. *Secrecy*, which refers to the denial of access to information by unauthorized users.
2. *Authenticity*, which refers to the validation of the source of a message.
3. *Integrity*, which refers to the assurance that a message was not modified by accidental or deliberate means in transit.

A conventional cryptographic system relies on the use of a single piece of private and necessarily secret information known as the *key*; hence, conventional cryptography is referred to as *single-key cryptography* or *secret-key cryptography*.² This form of cryptography operates on the premise that the key is known to the encrypter (sender) and by the decrypter (receiver) but to no others; the assumption is that once the message is encrypted, it is (probably) impossible to do the decryption without knowledge of the key.

Public-key cryptography,³ also called *two-key cryptography*, differs from conventional cryptography in that there is no longer a single secret key shared by two users. Rather, each user is provided with key material of one's own, and the key material is divided into two portions: a public component and a private component. The public component generates a public transformation, and the private component generates a private transformation. But, of course, the private transformation must be kept secret for secure communication between the two users.

A5.1 Secret-Key Cryptography

Basically, the flow of information in a secret-key cryptographic system is as shown in Figure A5.1. The message source generates a plaintext message, which is encrypted into a cryptogram at the transmitting end of the system. The cryptogram is sent to an *authorized user* at the receiving end over an “insecure” channel; a channel is considered insecure if

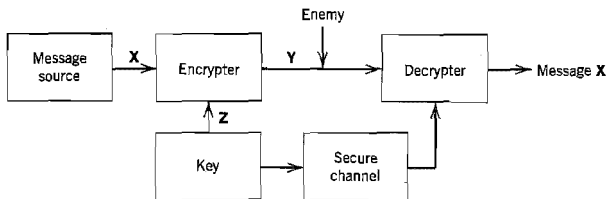


FIGURE A5.1 Block diagram of secret-key cryptographic system.

its security is inadequate for the needs of its users. It is assumed that in the course of transmission the cryptogram may be intercepted by an *enemy cryptanalyst*⁴ (i.e., would-be intruder into a cryptographic system). The requirement is to do the encryption in such a way that the enemy is prevented from learning the contents of the plaintext message.

In abstract terms, a *cryptographic system* or *cipher* (for short) is defined as a set of invertible transformations of the plaintext space (i.e., the set of possible plaintext messages) into the cryptogram space (i.e., the set of all possible cryptograms). Each particular transformation corresponds to encryption (enciphering) of a plaintext with a particular key. The invertibility of the transformation means that unique decryption (deciphering) of the cryptogram is possible when the key is known. Let X denote the plaintext message, Y denote the cryptogram, and Z denote the key. Let F denote the invertible transformation producing the cryptogram Y , as follows:

$$Y = F(X, Z) = F_z(X) \quad (\text{A5.1})$$

The transformation is intended to make the cryptogram Y useless to the enemy. At the receiving end of the system, the cryptogram Y is decrypted with the inverse transformation F^{-1} to recover the original plaintext message X , as shown by

$$F^{-1}(Y, Z) = F_z^{-1}(Y) = F_z^{-1}(F_z(X)) = X \quad (\text{A5.2})$$

In physical terms, the cryptographic system consists of a set of instructions, a piece of physical hardware, or a computer program. In any event, the system is designed to have the capability of encrypting the plaintext (and, of course, decrypting the resulting cryptogram) in a variety of ways; the particular way chosen to do the actual encryption is determined by the specific key.

The security of the system resides in the secret nature of the key, which requires that the key must be delivered to the receiver over a *secure channel* (e.g., registered mail, courier service) as implied in Figure A5.1. The cryptographic system depicted in this figure provides a solution to the *secrecy problem*, preventing an enemy from extracting information from messages transmitted over an insecure communication channel. Cryptography also provides a solution to the *authentication problem*, preventing an enemy cryptanalyst from impersonating the message sender. In this second situation, the enemy cryptanalyst is the one who originates a “fraudulent” cryptogram Y' that is delivered to the receiver (decrypter), as shown in Figure A5.2. The authentic cryptogram Y is shown as a dashed input to the enemy cryptanalyst, indicating that the enemy produces the fraudulent cryptogram Y' without ever seeing the authentic one. The receiver may be able to recognize Y' as fraudulent by decrypting it with the correct key Z ; hence, the line from the receiver output to the destination is shown dashed to suggest rejection of the fraudulent cryptogram Y' by the receiving user.

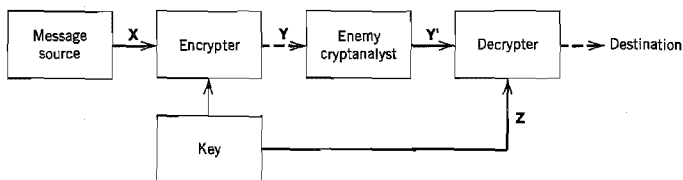


FIGURE A5.2 Illustrating the intrusion of an enemy cryptanalyst.

A5.2 Block and Stream Ciphers

Much as error-correcting codes are classified into block codes and convolutional codes, cryptographic systems (ciphers) may be classified into two broad classes: *block ciphers* and *stream ciphers*. Block ciphers operate in a purely combinatorial fashion on large blocks of plaintext, whereas stream ciphers process the plaintext in small pieces (i.e., characters or bits).

Figure A5.3 shows the generic form of a block cipher. The plaintext (consisting of serial data) is divided into large blocks, each of which is usually made up of a fixed number of bits. Successive blocks of the plaintext are enciphered (encrypted) using the same secret key, otherwise independently; the resulting enciphered blocks are finally converted into serial form. Thus, a particular plaintext block identical to a previous such block gives rise to an identical ciphertext block. Specifically, each bit of a particular ciphered block is chosen to be a function of all the bits of the associated plaintext block and the key; the goal of a block cipher is to have no specific bit of the plaintext ever appear in the ciphertext directly.

Block ciphers operate with a fixed transformation applied to large blocks of plaintext data, on a block-by-block basis. In contrast, a stream cipher operates on the basis of a time-varying transformation applied to individual bits of the plaintext. The most popular stream ciphers are the so-called *binary additive stream ciphers*, the generic form of which is shown in Figure A5.4. In such a cipher, the secret key is used to control a *keystream generator* that emits a binary sequence called the *keystream*, whose length is much larger than that of the key. Let x_n , y_n , and z_n denote the plaintext bit, ciphertext bit, and keystream bit at time n , respectively. The ciphertext bits are then determined by simple modulo-2 addition of the plaintext bits and the keystream bits, as shown by

$$y_n = x_n \oplus z_n, \quad n = 1, 2, \dots, N \quad (\text{A5.3})$$

where N is the length of the keystream. Because addition and subtraction in modulo-2 arithmetic are exactly the same, Equation (A5.3) also implies the following relation

$$x_n = y_n \oplus z_n, \quad n = 1, 2, \dots, N \quad (\text{A5.4})$$

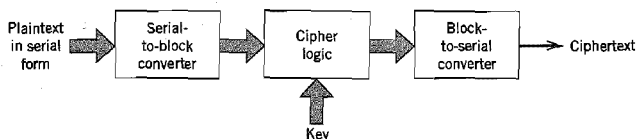


FIGURE A5.3 Block diagram of a block cipher.

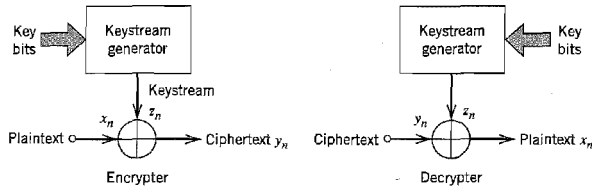


FIGURE A5.4 Binary additive stream cipher.

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-

graphic system can be subjected, and any system that succumbs to it is therefore considered totally insecure. Thus, for a cryptographic system to provide secrecy, at the minimum it should be immune to ciphertext-only attacks; ideally, it should also be immune to known-plaintext attacks.

A5.3 Information-Theoretic Approach

In the *Shannon model of cryptography*, named in recognition of Shannon's 1949 landmark paper on the information-theoretic approach to secrecy systems, the enemy cryptanalyst is assumed to have unlimited time and computing power. But the enemy is presumably restricted to a ciphertext-only attack. Cryptanalysis in the Shannon model is defined as the process of finding the secret key, given the cryptogram (ciphertext) and the *a priori* probabilities of the various plaintexts and keys. The secrecy of the system is considered *broken* when the enemy cryptanalyst performs decryption successfully, obtaining a *unique* solution to the cryptogram.⁶

Let $\mathbf{X} = (X_1, X_2, \dots, X_N)$ denote an N -bit plaintext message, and $\mathbf{Y} = (Y_1, Y_2, \dots, Y_N)$ denote the corresponding N -bit cryptogram; that is, both the plaintext and the cryptogram have the same number of bits. It is assumed that the secret key \mathbf{Z} used to construct the cryptogram is drawn according to some probability distribution. The uncertainty about \mathbf{X} is expressed by the entropy $H(\mathbf{X})$, and the uncertainty about \mathbf{X} given knowledge of \mathbf{Y} is expressed by the conditional entropy $H(\mathbf{X}|\mathbf{Y})$. The *mutual information* between \mathbf{X} and \mathbf{Y} is defined by

$$I(\mathbf{X}; \mathbf{Y}) = H(\mathbf{X}) - H(\mathbf{X}|\mathbf{Y}) \quad (\text{A5.5})$$

The mutual information $I(\mathbf{X}; \mathbf{Y})$ represents a basic measure of security (secrecy) in the Shannon model.

■ PERFECT SECURITY

Assuming that an enemy cryptanalyst can observe only the cryptogram \mathbf{Y} , it seems appropriate that we define the *perfect security* of a cryptographic system to mean that the plaintext \mathbf{X} and the cryptogram \mathbf{Y} are statistically independent. In other words, we have

$$I(\mathbf{X}; \mathbf{Y}) = 0 \quad (\text{A5.6})$$

Then, using Equation (A5.5), we find that the condition for perfect security may be rewritten as

$$H(\mathbf{X}|\mathbf{Y}) = H(\mathbf{X}) \quad (\text{A5.7})$$

Equation (A5.7) states that the best an enemy cryptanalyst can do, given the cryptogram \mathbf{Y} , is to guess the plaintext message \mathbf{X} according to the probability distribution of all possible messages.

Given the secret key \mathbf{Z} , we recognize that

$$\begin{aligned} H(\mathbf{X}|\mathbf{Y}) &\leq H(\mathbf{X}, \mathbf{Z}|\mathbf{Y}) \\ &= H(\mathbf{Z}|\mathbf{Y}) + H(\mathbf{X}|\mathbf{Y}, \mathbf{Z}) \end{aligned} \quad (\text{A5.8})$$

The conditional entropy $H(\mathbf{X}|\mathbf{Y}, \mathbf{Z})$ is zero if, and only if, \mathbf{Y} and \mathbf{Z} together uniquely determine \mathbf{X} ; this is indeed a valid assumption when the decryption process is performed with knowledge of the secret key \mathbf{Z} . Hence, we may simplify Equation (A5.8) as follows:

$$\begin{aligned} H(\mathbf{X}|\mathbf{Y}) &\leq H(\mathbf{Z}|\mathbf{Y}) \\ &\leq H(\mathbf{Z}) \end{aligned} \quad (\text{A5.9})$$

Thus, substituting Equation (A5.9) into (A5.7), we find that for a cryptographic system to provide perfect security, the following condition must be satisfied:

$$H(Z) \geq H(X) \quad (\text{A5.10})$$

The inequality of Equation (A5.10) is *Shannon's fundamental bound for perfect security*; it states that for perfect security, the uncertainty of a secret key Z must be at least as large as the uncertainty of the plaintext X that is concealed by the key.

For the case when the plaintext and key alphabets are of the same size, the use of Shannon's bound for perfect security yields the following result: *The key must be at least as long as the plaintext.* The conclusion to be drawn from this result is that the length of the *secret key* needed to build a perfectly secure cryptographic system may be impractically large for most applications. Nevertheless, perfect security has a place in the practical picture: It may be used when the number of possible messages is small or in cases where the greatest importance is attached to perfect security.

A well-known, perfectly secure cipher is the *one-time pad*⁷ (sometimes called the *Vernam cipher*), which is used for unconventional applications such as two users communicating on a hotline with high confidentiality requirements. The one-time pad is a stream cipher for which the key is the same as the keystream, as shown in Figure A5.5. For encryption the input consists of two components: a message represented by a sequence of message bits $\{x_n | n = 1, 2, \dots\}$, and a key represented by a sequence of statistically independent and uniformly distributed bits $\{z_n | n = 1, 2, \dots\}$. The resultant cipher $\{y_n | n = 1, 2, \dots\}$ is obtained by the modulo-2 addition of the two input sequences, as shown by

$$y_n = x_n \oplus z_n, \quad n = 1, 2, \dots$$

Consider, for example, the binary message sequence 00011010 and the binary key sequence 01101001. The modulo-2 addition of these two sequences is written as follows:

Message:	00011010
Key:	<u>01101001</u>
Cipher:	01110011

In the encryption rule described here, key bit 1 interchanges 0s and 1s in the message sequence, and key bit 0 leaves the message bits unchanged. The message sequence is recovered simply by modulo-2 addition of the binary cipher and key sequences, as shown by

Cipher:	01110011
Key:	<u>01101001</u>
Message:	00011010

The one-time pad is perfectly secure, because the mutual information between the message and the cipher is zero; it is therefore completely undecipherable.

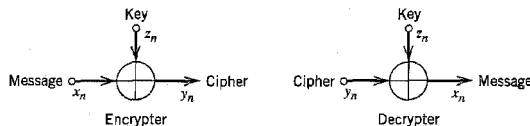


FIGURE A5.5 One-time pad (Vernam cipher).

■ UNICITY DISTANCE

Consider now the practical case of an imperfect cipher and ask the question: When can an enemy cryptanalyst break the cipher? As the amount of intercepted text increases, intuitively we expect that a point may be reached at which it becomes possible for an enemy cryptanalyst with unlimited time and computing power to find the key and thus break the cipher. This critical point in the Shannon model is called the *unicity distance*, which is formally defined as the smallest N such that the conditional entropy $H(Z|Y_1, Y_2, \dots, Y_N)$ is approximately zero. For a particular kind of "random cipher," the unicity distance is approximately given by⁸

$$N_0 \approx \frac{H(Z)}{r \log L_y} \quad (\text{A5.11})$$

where $H(Z)$ is the entropy of the key Z , and L_y is the size of the ciphertext alphabet. The parameter r is the *percentage redundancy* of the message information contained in the N -bit ciphertext; it is itself defined by

$$r = 1 - \frac{H(X)}{N \log L_y} \quad (\text{A5.12})$$

where $H(X)$ is the entropy of the plaintext X . In most cryptographic systems, the size L_y of the ciphertext alphabet is the same as the size L_x of the plaintext alphabet; in such a case, r is just the percentage redundancy of the plaintext itself. Although the derivation of Equation (A5.11) assumes a certain well-defined "random cipher," it can be used to estimate the unicity distance for ordinary types of ciphers, which is the routine practice today.

Let K be the number of digits in the key Z that are chosen from an alphabet of size L_z ; then we may express the entropy of the key Z as follows:

$$H(Z) \leq \log(L_z^K) = K \log L_z \quad (\text{A5.13})$$

with equality if and only if the key is completely random. Let the size L_z of the key alphabet be the same as the size L_y of the ciphertext alphabet, and let the key be chosen completely at random to maximize the unicity distance. Then, substituting Equation (A5.13) with equality into Equation (A5.11), we get the simple result

$$N_0 \approx \frac{K}{r} \quad (\text{A5.14})$$

To illustrate the application of Equation (A5.14), consider a cryptographic system with $L_x = L_y = L_z$, which is used for the encryption of English text. The percentage redundancy r for typical English text is about 75 percent. Hence, according to Equation (A5.14), an enemy cryptanalyst can break the cipher after intercepting only about $1.333K$ bits of ciphertext data, where K is the key size.

However, it is important to note that an imperfect cipher that is potentially breakable can still be of practical value. When the intercepted ciphertext contains sufficient information to satisfy Equation (A5.11), there is no guarantee that an enemy cryptanalyst with limited computational resources can actually break the cipher. Specifically, it is possible for the cipher to be designed in such a way that the task of the cryptanalysis, though known to be attainable with a finite amount of computation, is so overwhelming that it will literally exhaust the physical computing resources of the universe. In such a case, the imperfect cipher is said to be *computationally secure*.

■ 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 view 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

$$\mathbf{X} = (x_1, x_2, x_3, x_4, \dots)$$

where x_1, x_2, x_3, \dots are the successive letters, is transformed into the ciphertext

$$\begin{aligned} \mathbf{Y} &= (y_1, y_2, y_3, y_4, \dots) \\ &= (f(x_1), f(x_2), f(x_3), f(x_4), \dots) \end{aligned} \quad (\text{A5.15})$$

Plaintext letters	ABCDEFGHIJKLMNOPQRSTUVWXYZ
Ciphertext letters	YDUBHNACSVXELPFMKQJRWGOZIT

FIGURE A5.6 Substitution cipher.

where $f(\cdot)$ is a function with an inverse. When the substitutes are letters, the key is a permutation of the alphabet. Consider, for example, the ciphertext alphabet of Figure A5.6, where we see that the first letter Y is the substitute for A, the second letter D is the substitute for B, and so on. The use of a substitution cipher results in confusion.

2. Transposition cipher. In a transposition cipher, the plaintext is divided into groups of fixed period d and the same permutation is applied to each group, with the particular permutation rule being determined by the secret key. For example, consider the permutation rule described in Figure A5.7, for which the period is $d = 4$. According to this cipher, letter x_1 is moved from position 1 in the plaintext to position 4 in the ciphertext. Thus, the plaintext

$$X = (x_1, x_2, x_3, x_4, x_5, x_6, x_7, x_8, \dots)$$

is transformed into the ciphertext

$$Y = (x_3, x_4, x_2, x_1, x_7, x_8, x_6, x_5, \dots)$$

Although the single-letter statistics of the ciphertext Y are the same as those of the plaintext X, the higher-order statistics are changed. The use of a transposition cipher results in diffusion.

By interleaving the simple substitutions and transpositions and repeating the interleaving process many times, it is possible to build a strong cipher equipped with good diffusion and confusion.

► EXAMPLE A5.1

Consider the plaintext message

THE KING IS DEAD LONG LIVE THE KING

Using the permuted alphabet described in Figure A5.6 for the substitution cipher, this plaintext is transformed into the ciphertext

RCHXSPASJBHYBEFPAESGHRCHXSPA

Suppose next we apply the permutation rule described in Figure A5.7 for the transposition cipher; accordingly, the ciphertext resulting from the substitution cipher is further transformed into

HXCRASPSIHYBJFBEBSGEACHRHPASX

which has no resemblance to the original plaintext. ▲

Plaintext letters	x_1	x_2	x_3	x_4
Ciphertext letters	x_3	x_4	x_2	x_1

FIGURE A5.7 Transposition cipher.

A5.4 Data Encryption Standard

The *data encryption standard* (DES)⁹ is certainly the best known, and arguably the most widely used, secret-key cryptoalgorithm; the term *algorithm* is used to describe a sequence of computations. The basic DES algorithm can be used for both data encryption and data authentication. It is the standard cryptoalgorithm for data storage and mail systems, electronic funds transfers (retail and wholesale), and electronic business data interchange.

The DES algorithm is a strong block cipher that operates on 64-bit blocks of plaintext data and uses a 56-bit key; it is designed in accordance with Shannon's methods of diffusion and confusion. Essentially the same algorithm is used for encryption and decryption. The overall transformations employed in the DES algorithm may be written as $P^{-1}\{F[P(X)]\}$, where X is the plaintext, P is a certain permutation, and the function F combines substitutions and transpositions. The function F is itself obtained by cascading a certain function f , with each stage of the cascade referred to as a *round*.

The flow-chart of Figure A5.8 shows the details of the DES algorithm for encryption. After a certain initial permutation, a plaintext of 64 bits is divided into a left-half L_0 and a right-half R_0 , each of which is 32 bits long. The algorithm then performs 16 rounds of a key-dependent computation, with the i th round of the computation described as follows:

$$L_i = R_{i-1} \quad i = 1, 2, \dots, 16 \quad (\text{A5.16})$$

$$R_i = L_{i-1} \oplus f(R_{i-1}, Z_i) \quad i = 1, 2, \dots, 16 \quad (\text{A5.17})$$

On the right-hand side of Equation (A5.17), the addition is modulo-2 and each Z_i is a different 48-bit block of the key used in round i . The function $f(\cdot, \cdot)$ is a function with a 32-bit output. The result of the 16th round is reversed, obtaining the sequence $R_{16}L_{16}$. This 32-bit sequence is input into a final permutation P^{-1} to produce the 64-bit ciphertext. The aim is that after 16 rounds of key-dependent computations, the patterns in the original plaintext are undetectable in the ciphertext. From Equations (A5.16) and (A5.17), we note that for decryption the function $f(\cdot, \cdot)$ need not be invertible, because (L_{i-1}, R_{i-1}) can be recovered from (L_i, R_i) simply as follows:

$$R_{i-1} = L_i \quad i = 1, 2, \dots, 16 \quad (\text{A5.18})$$

$$L_{i-1} = R_i \oplus f(L_i, Z_i) \quad i = 1, 2, \dots, 16 \quad (\text{A5.19})$$

Equation (A5.19) holds even if the function $f(\cdot, \cdot)$ is a many-to-one function (i.e., it does not have a unique inverse).

Figure A5.9 shows the flowchart for computing the function $f(\cdot, \cdot)$. The 32-bit block R is first expanded into a new 48-bit block R' by repeating the edge bits of each successive 4-bit word (i.e., the bits numbered 1, 4, 5, 8, 9, 12, 13, 16, ..., 28, 29, 32). Thus, given the 32-bit block R written as

$$R = \underbrace{r_1 r_2 r_3 r_4}_{\text{first 4-bit word}} \quad \underbrace{r_5 r_6 r_7 r_8}_{\text{second 4-bit word}} \quad \cdots \quad \underbrace{r_{29} r_{30} r_{31} r_{32}}_{\text{eighth 4-bit word}}$$

we construct the expanded 48-bit block R' as follows:

$$R' = \underbrace{r_{32} r_1 r_2 r_3 r_4 r_5}_{\text{first 6-bit word}} \quad \underbrace{r_4 r_5 r_6 r_7 r_8 r_9}_{\text{second 6-bit word}} \quad \cdots \quad \underbrace{r_{28} r_{29} r_{30} r_{31} r_{32} r_1}_{\text{eighth 6-bit word}}$$

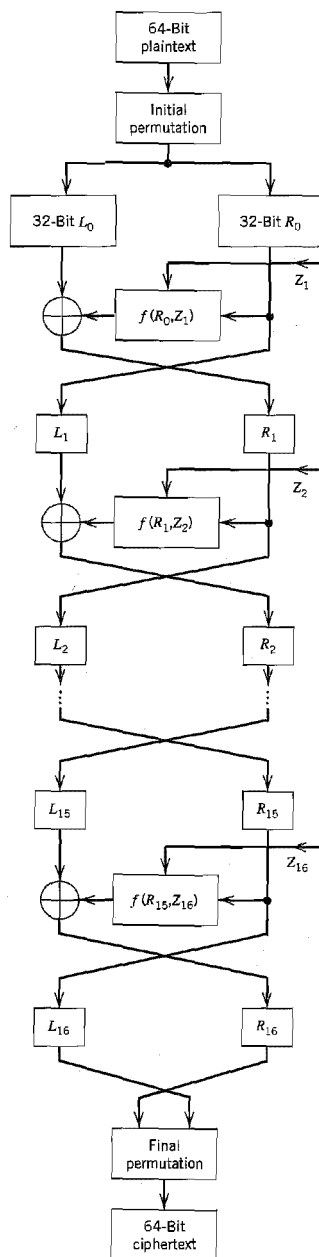


FIGURE A5.8 Data encryption standard. (From Diffie and Hellman, 1979, with permission of the IEEE.)

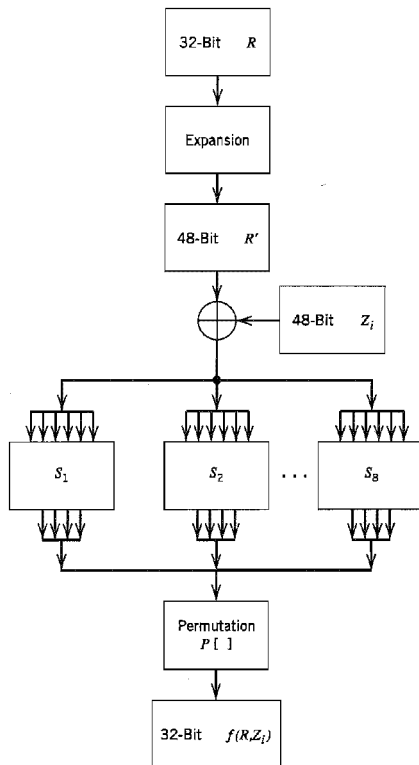


FIGURE A5.9 $f(R, K)$ flowchart. (From Diffie and Hellman, 1979, with permission of the IEEE.)

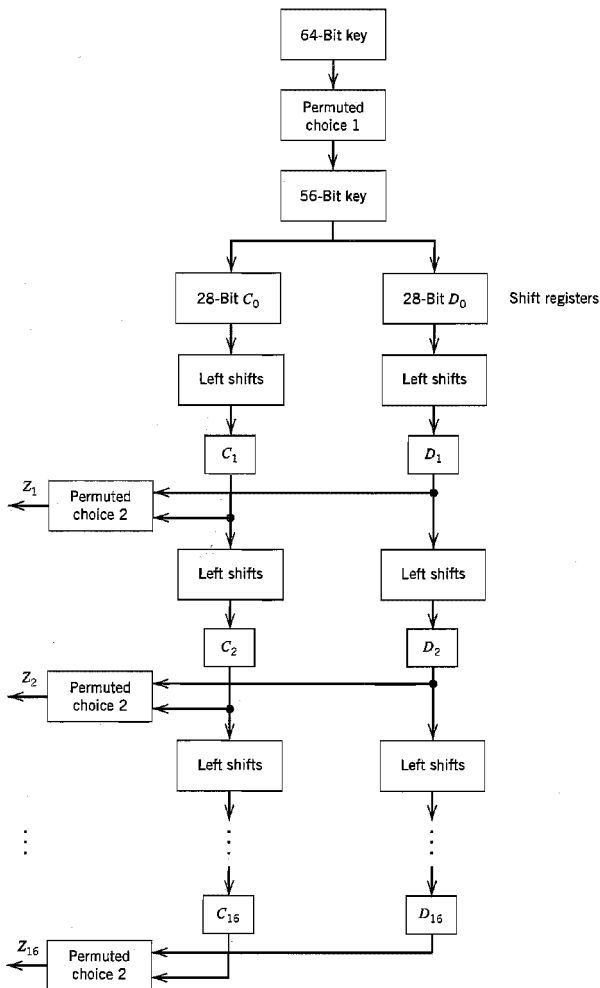
The 48-bit blocks R' and Z_i are added modulo-2, and the resultant is divided into eight 6-bit words. Let these words be denoted by B_1, B_2, \dots, B_8 . We thus write

$$B_1 B_2 \dots B_8 = R' \oplus Z_i \quad (\text{A5.20})$$

Each 6-bit word B_i is input to a substitution box S_i in the form of a look-up table, producing a 4-bit output $S_i(B_i)$. Each output bit of the substitution box $S_i(B_i)$ is a Boolean function of the 6-bit word B_i . The eight outputs $S_1(B_1), S_2(B_2), \dots, S_8(B_8)$ are arranged into a single 32-bit block that is input to the permutation box denoted by $P[\cdot]$. The permuted output so produced is the desired 32-bit function $f(R, Z_i)$, as shown by

$$f(R, Z_i) = P[S_1(B_1)S_2(B_2) \dots S_8(B_8)] \quad (\text{A5.21})$$

The 48-bit block Z_i for the i th iteration uses a different subset of the 64-bit key Z_0 . The procedure used to determine each Z_i is called the *key-schedule calculation*, the flowchart of which is shown in Figure A5.10. The key Z_0 has eight parity bits in positions 8, 16, \dots , 64, which are used for error detection in their respective 8-bit bytes; the errors



The outputs resulting from these 16 iterations provide the different 48-bit blocks Z_1, Z_2, \dots, Z_{16} of the key used in iteration 1, 2, \dots , 16, respectively.

Despite all the claims to the contrary, it appears that no one has yet demonstrated a fundamental weakness of the DES algorithm. Notwithstanding all the controversy surrounding its use, perhaps the most significant contribution of the DES algorithm is the fact that it has been instrumental in raising the level of interest in using cryptography as a mechanism for secure computer networks.

A5.5 Public-Key Cryptography¹⁰

For a pair of users to engage in cryptographic communication over an insecure channel, it is necessary for the users to exchange key information prior to communication. The requirement for a secure distribution of keys among authorized users applies to all cryptographic systems, regardless of their type. In conventional cryptography, the users employ a physically secure channel (e.g., courier service or registered mail) for key distribution. However, the use of such a supplementary channel points to a major limitation of conventional cryptography. Needless to say, the use of courier service or registered mail for key distribution is costly, inconvenient, low-bandwidth, and slow; also, it is not always secure.

The problem of key distribution is particularly accentuated in large communication networks, where the number of possible connections grows as $(n^2 - n)/2$ for n users. For large n , the cost of key distribution becomes prohibitive. Thus, in the development of large, secure communication networks, we are compelled to rely on the use of insecure channels for both exchange of key information and subsequent secure communication. This constraint raises a fundamental question: How can key information be exchanged securely over an insecure channel? In *public-key cryptography*, this seemingly difficult issue is resolved by making some key material "public" and thereby considerably simplifying the task of key management. This is in direct contrast to conventional cryptography, where the key is kept completely secret from an enemy cryptanalyst.

A public key cryptographic system is described by two sets of algorithms that compute invertible functions (transformations). Let these two sets of algorithms be denoted by $\{E_z\}$ and $\{D_z\}$ that are indexed by z . The invertible transformations computed by these algorithms may be written as follows

$$E_z: f_z(x) = y \quad (\text{A5.22})$$

$$D_z: f_z^{-1}(y) = x \quad (\text{A5.23})$$

where x is a certain input message in the domain of some function f_z indexed by z , and y is the corresponding cryptogram in the range of f_z . A fundamental requirement of the system is that the function f_z must be a *trapdoor one-way function*. The term "one-way" refers to the fact that for x in the domain of f_z , it must be easy to compute $f_z(x)$ from knowledge of the algorithm E_z , but for a certain cryptogram y in the range of f_z , an enemy cryptanalyst must find it extremely difficult to compute the inverse $f_z^{-1}(y)$. On the other hand, an authorized user in possession of the associated algorithm D_z would find it easy to compute the inverse $f_z^{-1}(y)$. Thus the *private key* (algorithm) D_z provides a "trapdoor" that makes the problem of inverting the function f_z appear extremely difficult from the viewpoint of the cryptanalyst, but easy for the (sole authorized) possessor of D_z . Since knowledge of the key (algorithm) E_z does not by itself make it possible to compute the inverse of f_z , it may be made *public*; hence, the name "public-key cryptography."

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 names and addresses. 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*

$$Y = \alpha^X \bmod p \quad \text{for } 1 \leq X \leq p - 1 \quad (\text{A5.24})$$

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

$$X = \log_{\alpha} Y \bmod p \quad \text{for } 1 \leq Y \leq p - 1 \quad (\text{A5.25})$$

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 $\{1, 2, \dots, p\}$ that is kept as a *private secret*. But the discrete exponential

$$Y_i = \alpha^{X_i} \bmod p \quad (\text{A5.26})$$

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

privately. To proceed, user i fetches Y_i from the public directory and uses the private secret X_i to compute

$$\begin{aligned} K_{ji} &= (Y_j)^{X_i} \bmod p \\ &= (\alpha^{X_j})^{X_i} \bmod p \\ &= \alpha^{X_j X_i} \bmod p \end{aligned} \quad (\text{A5.27})$$

In a similar way, user j computes K_{ji} . But we have

$$K_{ji} = K_{ij} \quad (\text{A5.28})$$

Accordingly, users i and j arrive at K_{ji} as the *secret key* in a conventional cryptosystem. Another user must compute K_{ji} using the information Y_i and Y_j obtained from the public directory, applying the alternative formula

$$K_{ji} = (Y_j)^{\log_{Y_i} Y_i} \bmod p \quad (\text{A5.29})$$

Apparently, there is no other method for an enemy to find the secret key K_{ji} ; however, there is no proof for it. In light of what we said earlier, Equation (A5.29) is difficult to calculate as it involves a discrete logarithm, whereas Equation (A5.27) is easy to calculate as it involves a discrete exponential. Thus, security of the system depends on the difficulty encountered in computing a discrete logarithm.

The Diffie-Hellman public key-distribution system is the oldest system in its class; nevertheless, it is still generally considered to be one of the most secure and practical public key-distribution systems.

A5.6 Rivest-Shamir-Adleman System

To develop a public-key cryptographic system is no easy task. Indeed, numerous such systems have been proposed in the literature, but unfortunately most of them have proven to be insecure. To date, the most successful implementation of public-key cryptography is the *Rivest-Shamir-Adleman (RSA) system*,¹¹ which uses ideas from classical number theory. It is considered to be one of the most secure cryptographic systems in that it has withstood many attempts by experts in the field to break it.

The RSA algorithm is a block cipher based on the fact that finding a random prime number of large size (e.g., 100 digit) is computationally easy, but factoring the product of two such numbers is currently considered computationally infeasible. Specifically, the computation of parameters specific to the RSA algorithm proceeds as follows:

1. Choose two *very large prime numbers*, p and q , at random; the prime numbers have to be fairly carefully chosen as some prime numbers lead to a very weak system.
2. Multiply the numbers p and q , obtaining the product

$$pq = n \quad (\text{A5.30})$$

Find the *Euler totient function* of n , using the formula

$$\phi(n) = (p-1)(q-1) \quad (\text{A5.31})$$

Equation (A5.31) follows from the definition of the Euler totient function $\phi(n)$ as the number of positive integers i less than n , such that the *greatest common divisor* of i and n is equal to one.

3. Let e be a positive integer less than $\phi(n)$, such that the greatest common divisor of e and $\phi(n)$ is equal to one. Hence, find a positive integer d less than $\phi(n)$, such that

$$de = 1 \bmod \phi(n) \quad (\text{A5.32})$$

The RSA trapdoor one-way function is then defined simply by computing the *discrete exponentiation*

$$f_z(x) = x^e = y \bmod n \quad (\text{A5.32})$$

The values of n and e constitute the public key; hence, publishing the easy-to-find algorithm E_z to compute the function f_z amounts just to publishing the numbers n and e .

The prime numbers p and q constitute the private key. Since d is related to p and q , possession of the easy-to-find (when one knows the trapdoor z) algorithm D_z to compute the inverse function f_z^{-1} amounts just to knowing p and q . In particular, the inverse function is defined by

$$f_z^{-1}(y) = y^d \bmod n \quad (\text{A5.34})$$

The decrypting exponent d is found using Equation (A5.32), which is equivalent to the statement (in ordinary integer arithmetic) that

$$de = \phi(n)Q + 1 \quad (\text{A5.35})$$

for some integer Q . Note that $\phi(n)$ is itself related to p and q by Equation (A5.31). Since $y = x^e$, we may use Equations (A5.32) and (A5.33) to write

$$\begin{aligned} y^d &= x^{de} \\ &= x^{\phi(n)Q+1} \\ &= ((x^{\phi(n)})^Q)x \end{aligned} \quad (\text{A5.36})$$

We now make use of a celebrated theorem of Euler, which states that for any positive integers x and n with $x < n$, we have

$$x^{\phi(n)} = 1 \bmod n \quad (\text{A5.37})$$

Hence, the use of Equation (A5.37) in (A5.36) yields the desired decryption:

$$y^d = x \quad (\text{A5.38})$$

We thus see that finding the inverse function f_z^{-1} is easy, given knowledge of the prime numbers p and q .

The *security* of the RSA cryptosystem rests on the premise that any method of inverting the function f_z is *equivalent to factoring* $n = pq$. This equivalence raises the question: Is an attack by factoring n computationally feasible? It appears that the answer is no, provided that the prime numbers p and q are on the order of 100 decimal digits each and that there is no revolutionary breakthrough in factoring algorithms.

■ DIGITAL SIGNATURES¹²

For an electronic mail system to replace the use of ordinary paper mail for business transactions, it must be possible for a user of the system to “sign” an electronic message. The

use of a *digital signature* provides proof that the message did originate from the sender. To satisfy this requirement, the digital signature must have the following properties:

- ▶ The receiver of an electronic message is able to verify the sender's signature.
- ▶ The signature is not forgeable.
- ▶ The sender of a signed electronic message is unable to disclaim it.

To implement digital signatures using the RSA algorithm, we may proceed as follows. A user in possession of the private key d may sign a given message block m by forming the signature

$$s = m^d \bmod n \quad (\text{A5.39})$$

It is difficult to compute s unless the private key d is known. Hence, a digital signature defined in accordance with Equation (A5.39) is difficult to forge. Moreover, the sender of message m cannot deny having sent it, since no one else could have created the signature s . The receiver proceeds by using the public key e to compute

$$\begin{aligned} s^e &= (m^d)^e \bmod n \\ &= m^{de} \bmod n \\ &= m \bmod n \end{aligned} \quad (\text{A5.40})$$

where, in the last line, use is made of Equation (A5.32). Hence, the receiver is able to validate the sender's signature by establishing that the computation of $s^e \bmod n$ produces the same result as the deciphered message m . Thus, the RSA algorithm satisfies all the three necessary properties of a digital signature.

A5.7 Summary and Discussion

Cryptography is a "hot" research area. This statement should not come as a surprise. Considering the fact that we are in an *information society*, the importance of cryptography as a security mechanism will continue to grow. In this appendix, we have presented an introductory treatment of this highly important subject.

We may classify cryptography into secret-key cryptography and public-key cryptography, depending on whether the key used for the encryption of a message and its decryption is completely secret or partly public. Alternatively, we may classify a cryptographic system into a block cipher or stream cipher, depending on the method of implementation. A block cipher exhibits error propagation, which can prove highly valuable in authentication.

Among the many cryptographic systems developed to date, the data encryption standard (DES) and the Rivest–Shamir–Adleman (RSA) algorithms stand out as the most successful ones. Both of these cryptoalgorithms are block ciphers. They differ from each other in that the DES algorithm involves the use of a secret key whereas the RSA algorithm involves the use of a public key. In a secret-key system, the same key is shared both by the sender and the receiver. On the other hand, in a public-key system, the key is split into two parts: a public key located in the transmitter and a private (secret) key located in the receiver; in the latter system, it is computationally infeasible to recover the plaintext message from its encrypted version without knowledge of the private key.

Although public-key cryptosystems such as RSA provide an effective method for key management, they are inefficient for the bulk encryption of data due to low bandwidths. In contrast, conventional cryptosystems such as DES provide better throughput, but they

require key management. This suggests the possible use of a hybrid approach exploiting the best elements of both cryptosystems as the basis for the practical design of a secure communication system. For example, the RSA algorithm may be used for authentication, and the DES algorithm for encryption.

NOTES AND REFERENCES

1. For an introductory treatment of cryptography, see Chapter 15 of the book by Adámek (1991). For a comprehensive treatment of the many facets of cryptology, see the book edited by Simmons (1992); this book is an expanded edition of a Special Issue of the *Proceedings of the IEEE* (1988) on cryptology. The chapter contributions of the book by Simmons are written by leading authorities on the subject of cryptology. A nice treatment of cryptology is also presented in the book by van Tilborg (1988).
2. The era of scientific secret-key cryptography was ushered in with the publication of a landmark paper by Shannon (1949), which established the connection between cryptography and information theory.
3. The era of public-key cryptography was established with the publication of another landmark paper by Diffie and Hellman (1976), which showed for the first time that it is possible to have secret communications without any transfer of a key between sender and receiver. It was the paper by Diffie and Hellman that sparked the explosion of research interest in cryptology, which has continued ever since.
4. The term *enemy cryptanalyst* is commonly used in cryptology to refer to a cryptogram interceptor (eavesdropper); its usage originates from military applications.
5. For a comprehensive treatment of stream ciphers, see Chapter 2 written by R. A. Rueppel in the book *Contemporary Cryptology*, edited by Simmons (1992).
6. For a highly readable account of the Shannon model of cryptography, see the opening chapter by J. L. Massey in the book edited by Simmons (1992).
7. The one-time pad derives its name from its use (shortly before, during, and after World War II) by spies of several governments, who were given a pad of paper with a randomly chosen key and told to use it only for a single encryption. The one-time pad is also known as Vernam's cipher, so named in recognition of its originator, G. S. Vernam.
8. For a derivation of Equation (A5.11), see the original paper by Shannon (1949).
9. The history of the DES algorithm is recounted by M. E. Smid and D. K. Branstad in Chapter 1 of the book edited by Simmons (1992). For a description of the DES algorithm, see Diffie and Hellman (1979). See also the books by Meyer and Matyas (1982) and Torrieri (1992, Chapter 6).
10. For a comprehensive treatment of public-key cryptography, see Chapter 4 by J. Nechvatal in the book edited by Simmons (1992). This book also includes a chapter contribution by W. Diffie that describes the several attempts to devise secure public-key cryptoalgorithms and the gradual evolution of a variety of protocols based on them.
11. The RSA system is patented; it is named in recognition of its originators R. L. Rivest, A. Shamir, and L. Adleman. The original reference for this cryptosystem is Rivest, Shamir, and Adleman (1978).
12. The idea of a digital signature was first discussed by Diffie and Hellman (1976). Its implementation using the RSA algorithm is described by Rivest, Shamir, and Adleman (1978). For a detailed treatment of digital signatures, see Chapter 6 by C. J. Mitchell, F. Piper, and R. Wild in the book edited by Simmons (1992).

TABLES

The twelve tables compiled in this final appendix cover the following:

- ▶ *ASCII code*
- ▶ *Fourier and Hilbert transforms*
- ▶ *Bessel functions*
- ▶ *Error function*
- ▶ *Selected modem standards*
- ▶ *Trigonometric identities, series expansions, and integrals*
- ▶ *Useful constants and recommended unit prefixes*

TABLE A6.1 *ASCII code*

Bit Position											
				7 0	0	0	0	1	1	1	1
				6 0	0	1	1	0	0	1	1
4	3	2	1	5 0	1	0	1	0	1	0	1
0	0	0	0	NUL	DLE	SP	0	@	P	\	p
0	0	0	1	SOH	DC1	!	1	A	Q	a	q
0	0	1	0	STX	DC2	"	2	B	R	b	r
0	0	1	1	ETX	DC3	#	3	C	S	c	s
0	1	0	0	EOT	DC4	\$	4	D	T	d	t
0	1	0	1	ENQ	NAK	%	5	E	U	e	u
0	1	1	0	ACK	SYN	&	6	F	V	f	v
0	1	1	1	BEL	ETB	'	7	G	W	g	w
1	0	0	0	BS	CAN	(8	H	X	h	x
1	0	0	1	HT	EM)	9	I	Y	i	y
1	0	1	0	LF	SUB	*	:	J	Z	j	z
1	0	1	1	VT	ESC	+	;	K	[k	{
1	1	0	0	FF	FS	'	<	L	\	l	:
1	1	0	1	CR	GS	-	=	M]	m	}
1	1	1	0	SO	RS	.	>	N	^	n	~
1	1	1	1	SI	US	/	?	O	—	o	DEL
ACK	Acknowledge			ENQ	Enquiry			NUL	Null or all zeros		
BEL	Bell or alarm			EOT	End of transmission			RS	Record separator		
BS	Backspace			ESC	Escape			SI	Shift in		
CAN	Cancel			ETB	End of transmission block			SO	Shift out		
CR	Carriage return			ETX	End of text			SOH	Start of heading		
DC1	Device control 1			FF	Form feed			SP	Space		
DC2	Device control 2			FS	File separator			STX	Start of text		
DC3	Device control 3			GS	Group separator			SUB	Substitute		
DC4	Device control 4			HT	Horizontal tab			SYN	Synchronous idle		
DEL	Delete			LF	Line feed			US	Unit separator		
DLE	Data link escape			NAK	Negative acknowledge			VT	Vertical tab		
EM	End of medium										

(From Couch, 1990, with permission of Macmillan.)

TABLE A6.2 Summary of properties of the Fourier transform

Property	Mathematical Description
1. Linearity	$ag_1(t) + bg_2(t) \Rightarrow aG_1(f) + bG_2(f)$ where a and b are constants
2. Time scaling	$g(at) \Rightarrow \frac{1}{ a } G\left(\frac{f}{a}\right)$ where a is a constant
3. Duality	If $g(t) \Rightarrow G(f)$, then $G(t) \Rightarrow g(-f)$
4. Time shifting	$g(t - t_0) \Rightarrow G(f) \exp(-j2\pi f t_0)$
5. Frequency shifting	$\exp(j2\pi f_c t)g(t) \Rightarrow G(f - f_c)$
6. Area under $g(t)$	$\int_{-\infty}^{\infty} g(t) dt = G(0)$
7. Area under $G(f)$	$g(0) = \int_{-\infty}^{\infty} G(f) df$
8. Differentiation in the time domain	$\frac{d}{dt} g(t) \Rightarrow j2\pi f G(f)$
9. Integration in the time domain	$\int_{-\infty}^t g(\tau) d\tau \Rightarrow \frac{1}{j2\pi f} G(f) + \frac{G(0)}{2} \delta(f)$
10. Conjugate functions	If $g(t) \Rightarrow G(f)$, then $g^*(t) \Rightarrow G^*(-f)$
11. Multiplication in the time domain	$g_1(t)g_2(t) \Rightarrow \int_{-\infty}^{\infty} G_1(\lambda)G_2(f - \lambda) d\lambda$
12. Convolution in the time domain	$\int_{-\infty}^{\infty} g_1(\tau)g_2(t - \tau) d\tau \Rightarrow G_1(f)G_2(f)$

TABLE A6.3 *Fourier-transform pairs*

Time Function	Fourier Transform
$\text{rect}\left(\frac{t}{T}\right)$	$T \text{sinc}(fT)$
$\text{sinc}(2Wt)$	$\frac{1}{2W} \text{rect}\left(\frac{f}{2W}\right)$
$\exp(-at)u(t), \quad a > 0$	$\frac{1}{a + j2\pi f}$
$\exp(-a t), \quad a > 0$	$\frac{2a}{a^2 + (2\pi f)^2}$
$\exp(-\pi t^2)$	$\exp(-\pi f^2)$
$\begin{cases} 1 - \frac{ t }{T}, & t < T \\ 0, & t \geq T \end{cases}$	$T \text{sinc}^2(fT)$
$\delta(t)$	1
1	$\delta(f)$
$\delta(t - t_0)$	$\exp(-j2\pi f t_0)$
$\exp(j2\pi f_c t)$	$\delta(f - f_c)$
$\cos(2\pi f_c t)$	$\frac{1}{2}[\delta(f - f_c) + \delta(f + f_c)]$
$\sin(2\pi f_c t)$	$\frac{1}{2j}[\delta(f - f_c) - \delta(f + f_c)]$
$\text{sgn}(t)$	$\frac{1}{j\pi f}$
$\frac{1}{\pi t}$	$-j \text{sgn}(f)$
$u(t)$	$\frac{1}{2} \delta(f) + \frac{1}{j2\pi f}$
$\sum_{i=-\infty}^{\infty} \delta(t - iT_0)$	$\frac{1}{T_0} \sum_{n=-\infty}^{\infty} \delta\left(f - \frac{n}{T_0}\right)$

Notes: $u(t)$ = unit step function $\delta(t)$ = delta function, or unit impulse $\text{rect}(t)$ = rectangular function of unit amplitude and unit duration centered on the origin $\text{sgn}(t)$ = signum function $\text{sinc}(t)$ = sinc function

TABLE A6.4 Hilbert transform pairs^a

Time Function	Hilbert Transform
$m(t) \cos(2\pi f_c t)$	$m(t) \sin(2\pi f_c t)$
$m(t) \sin(2\pi f_c t)$	$-m(t) \cos(2\pi f_c t)$
$\cos(2\pi f_c t)$	$\sin(2\pi f_c t)$
$\sin(2\pi f_c t)$	$-\cos(2\pi f_c t)$
$\frac{\sin t}{t}$	$\frac{1 - \cos t}{t}$
$\text{rect}(t)$	$-\frac{1}{\pi} \log \left \frac{t - \frac{1}{2}}{t + \frac{1}{2}} \right $
$\delta(t)$	$\frac{1}{\pi t}$
$\frac{1}{1 + t^2}$	$\frac{t}{1 + t^2}$
$\frac{1}{t}$	$-\pi \delta(t)$

^aIn the first two pairs, it is assumed that $m(t)$ is band-limited to the interval $-W \leq f \leq W$, where $W < f_c$.

Notes: $\delta(t)$: delta function

$\text{rect}(t)$: rectangular function of unit amplitude and unit duration centered on the origin

log: natural logarithm

TABLE A6.5 Table of Bessel functions^a

	$J_n(x)$								
$n \backslash x$	0.5	1	2	3	4	6	8	10	12
0	0.9385	0.7652	0.2239	-0.2601	-0.3971	0.1506	0.1717	-0.2459	0.0477
1	0.2423	0.4401	0.5767	0.3391	-0.0660	-0.2767	0.2346	0.0435	-0.2234
2	0.0306	0.1149	0.3528	0.4861	0.3641	-0.2429	-0.1130	0.2546	-0.0849
3	0.0026	0.0196	0.1289	0.3091	0.4302	0.1148	-0.2911	0.0584	0.1951
4	0.0002	0.0025	0.0340	0.1320	0.2811	0.3576	-0.1054	-0.2196	0.1825
5	—	0.0002	0.0070	0.0430	0.1321	0.3621	0.1858	-0.2341	-0.0735
6		—	0.0012	0.0114	0.0491	0.2458	0.3376	-0.0145	-0.2437
7			0.0002	0.0025	0.0152	0.1296	0.3206	0.2167	-0.1703
8			—	0.0005	0.0040	0.0565	0.2235	0.3179	0.0451
9				0.0001	0.0009	0.0212	0.1263	0.2919	0.2304
10				—	0.0002	0.0070	0.0608	0.2075	0.3005
11					—	0.0020	0.0256	0.1231	0.2704
12						0.0005	0.0096	0.0634	0.1953
13						0.0001	0.0033	0.0290	0.1201
14						—	0.0010	0.0120	0.0650

^aFor more extensive tables of Bessel functions, see Watson (1966, pp. 666–697), and Abramowitz and Stegun (1965, pp. 358–406).

TABLE A6.6 *The error function^a*

u	$\text{erf}(u)$	u	$\text{erf}(u)$
0.00	0.00000	1.10	0.88021
0.05	0.05637	1.15	0.89612
0.10	0.11246	1.20	0.91031
0.15	0.16800	1.25	0.92290
0.20	0.22270	1.30	0.93401
0.25	0.27633	1.35	0.94376
0.30	0.32863	1.40	0.95229
0.35	0.37938	1.45	0.95970
0.40	0.42839	1.50	0.96611
0.45	0.47548	1.55	0.97162
0.50	0.52050	1.60	0.97635
0.55	0.56332	1.65	0.98038
0.60	0.60386	1.70	0.98379
0.65	0.64203	1.75	0.98667
0.70	0.67780	1.80	0.98909
0.75	0.71116	1.85	0.99111
0.80	0.74210	1.90	0.99279
0.85	0.77067	1.95	0.99418
0.90	0.79691	2.00	0.99532
0.95	0.82089	2.50	0.99959
1.00	0.84270	3.00	0.99998
1.05	0.86244	3.30	0.999998

^aThe error function is tabulated extensively in several references; see for example, Abramowitz and Stegun (1965, pp. 297–316).

TABLE A6.7 Selection of ITU voiceband (telephone line) modem standards

	ITU Standard ^a	Type of modulation	Bit rate, b/s	Symbol rate, bauds
(a) Symmetric modems:	V.21	Binary FSK	300	300
	V.22 bis	QPSK	1,200	600
	V.26	QPSK	2,400	1,200
	V.27	8-PSK	4,800	2,400
	V.32	16-QAM	9,600	2,400
	V.34	1024-QAM	28,800	3,429
	V.34 High Speed	Nested-constellation of four 960-QAM constellations	33,600	
(b) Asymmetric modems:	V.90: Downstream	Digital	56,000	
	Upstream	V.34 High Speed	33,600	

^aThe suffix "bis" designates the second version of a particular standard.

TABLE A6.8 Trigonometric identities

$$\exp(\pm j\theta) = \cos \theta \pm j \sin \theta$$

$$\cos \theta = \frac{1}{2}[\exp(j\theta) + \exp(-j\theta)]$$

$$\sin \theta = \frac{1}{2j}[\exp(j\theta) - \exp(-j\theta)]$$

$$\sin^2 \theta + \cos^2 \theta = 1$$

$$\cos^2 \theta - \sin^2 \theta = \cos(2\theta)$$

$$\cos^2 \theta = \frac{1}{2}[1 + \cos(2\theta)]$$

$$\sin^2 \theta = \frac{1}{2}[1 - \cos(2\theta)]$$

$$2 \sin \theta \cos \theta = \sin(2\theta)$$

$$\sin(\alpha \pm \beta) = \sin \alpha \cos \beta \pm \cos \alpha \sin \beta$$

$$\cos(\alpha \pm \beta) = \cos \alpha \cos \beta \mp \sin \alpha \sin \beta$$

$$\tan(\alpha \pm \beta) = \frac{\tan \alpha \pm \tan \beta}{1 \mp \tan \alpha \tan \beta}$$

$$\sin \alpha \sin \beta = \frac{1}{2}[\cos(\alpha - \beta) - \cos(\alpha + \beta)]$$

$$\cos \alpha \cos \beta = \frac{1}{2}[\cos(\alpha - \beta) + \cos(\alpha + \beta)]$$

$$\sin \alpha \cos \beta = \frac{1}{2}[\sin(\alpha - \beta) + \sin(\alpha + \beta)]$$

TABLE A6.9 *Series expansions***Taylor series**

$$f(x) = f(a) + \frac{f'(a)}{1!} (x - a) + \frac{f''(a)}{2!} (x - a)^2 + \cdots + \frac{f^{(n)}(a)}{n!} (x - a)^n + \cdots$$

where

$$f^{(n)}(a) = \left. \frac{d^n f(x)}{dx^n} \right|_{x=a}$$

MacLaurin series

$$f(x) = f(0) + \frac{f'(0)}{1!} x + \frac{f''(0)}{2!} x^2 + \cdots + \frac{f^{(n)}(0)}{n!} x^n + \cdots$$

where

$$f^{(n)}(0) = \left. \frac{d^n f(x)}{dx^n} \right|_{x=0}$$

Binomial series

$$(1 + x)^n = 1 + nx + \frac{n(n-1)}{2!} x^2 + \cdots, \quad |nx| < 1$$

Exponential series

$$\exp x = 1 + x + \frac{1}{2!} x^2 + \cdots$$

Logarithmic series

$$\log(1 + x) = x - \frac{1}{2}x^2 + \frac{1}{3}x^3 - \cdots$$

Trigonometric series

$$\sin x = x - \frac{1}{3!} x^3 + \frac{1}{5!} x^5 - \cdots$$

$$\cos x = 1 - \frac{1}{2!} x^2 + \frac{1}{4!} x^4 - \cdots$$

$$\tan x = x + \frac{1}{3} x^3 + \frac{2}{15} x^5 + \cdots$$

$$\sin^{-1} x = x + \frac{1}{6} x^3 + \frac{3}{40} x^5 + \cdots$$

$$\tan^{-1} x = x - \frac{1}{3} x^3 + \frac{1}{5} x^5 - \cdots, \quad |x| < 1$$

$$\operatorname{sinc} x = 1 - \frac{1}{3!} (\pi x)^2 + \frac{1}{5!} (\pi x)^4 - \cdots$$

TABLE A6.10 Integrals

Indefinite integrals

$$\int x \sin(ax) dx = \frac{1}{a^2} [\sin(ax) - ax \cos(ax)]$$

$$\int x \cos(ax) dx = \frac{1}{a^2} [\cos(ax) + ax \sin(ax)]$$

$$\int x \exp(ax) dx = \frac{1}{a^2} \exp(ax)(ax - 1)$$

$$\int x \exp(ax^2) dx = \frac{1}{2a} \exp(ax^2)$$

$$\int \exp(ax) \sin(bx) dx = \frac{1}{a^2 + b^2} \exp(ax)[a \sin(bx) - b \cos(bx)]$$

$$\int \exp(ax) \cos(bx) dx = \frac{1}{a^2 + b^2} \exp(ax)[a \cos(bx) + b \sin(bx)]$$

$$\int \frac{dx}{a^2 + b^2 x^2} = \frac{1}{ab} \tan^{-1} \left(\frac{bx}{a} \right)$$

$$\int \frac{x^2 dx}{a^2 + b^2 x^2} = \frac{x}{b^2} - \frac{a}{b^3} \tan^{-1} \left(\frac{bx}{a} \right)$$

Definite integrals

$$\int_0^\infty \frac{x \sin(ax)}{b^2 + x^2} dx = \frac{\pi}{2} \exp(-ab), \quad a > 0, b > 0$$

$$\int_0^\infty \frac{\cos(ax)}{b^2 + x^2} dx = \frac{\pi}{2b} \exp(-ab), \quad a > 0, b > 0$$

$$\int_0^\infty \frac{\cos(ax)}{(b^2 - x^2)^2} dx = \frac{\pi}{4b^3} [\sin(ab) - ab \cos(ab)], \quad a > 0, b > 0$$

$$\int_0^\infty \operatorname{sinc} x dx = \int_0^\infty \operatorname{sinc}^2 x dx = \frac{1}{2}$$

$$\int_0^\infty \exp(-ax^2) dx = \frac{1}{2} \sqrt{\frac{\pi}{a}}, \quad a > 0$$

$$\int_0^\infty x^2 \exp(-ax^2) dx = \frac{1}{4a} \sqrt{\frac{\pi}{a}}, \quad a > 0$$

TABLE A6.11 *Useful constants***Physical Constants**

Boltzmann's constant	$k = 1.38 \times 10^{-23}$ joule/degree Kelvin
Planck's constant	$h = 6.626 \times 10^{-34}$ joule-second
Electron (fundamental) charge	$q = 1.602 \times 10^{-19}$ coulomb
Speed of light in vacuum	$c = 2.998 \times 10^8$ meters/second
Standard (absolute) temperature	$T_0 = 273$ degrees Kelvin
Thermal voltage	$V_T = 0.026$ volt at room temperature
Thermal energy kT at standard temperature	$kT_0 = 3.77 \times 10^{-21}$ joule
One hertz (hz) = 1 cycle/second; 1 cycle = 2π radians	
One watt (W) = 1 joule/second	

Mathematical Constants

Base of natural logarithm	$e = 2.7182818$
Logarithm of e to base 2	$\log_2 e = 1.442695$
Logarithm of 2 to base e	$\log 2 = 0.693147$
Logarithm of 2 to base 10	$\log_{10} 2 = 0.30103$
Pi	$\pi = 3.1415927$

TABLE A6.12 *Recommended unit prefixes*

<i>Multiples and Submultiples</i>	<i>Prefixes</i>	<i>Symbols</i>
10^{12}	tera	T
10^9	giga	G
10^6	mega	M
10^3	kilo	K (k)
10^{-3}	milli	m
10^{-6}	micro	μ
10^{-9}	nano	n
10^{-12}	pico	p

GLOSSARY

Conventions and Notations

1. The symbol $| \cdot |$ means the absolute value, or magnitude, of the complex quantity contained within.
2. The symbol $\arg(\cdot)$ means the phase angle of the complex quantity contained within.
3. The symbol $\text{Re}[\cdot]$ means the “real part of,” and $\text{Im}[\cdot]$ 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 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 $F[\cdot]$ indicates the Fourier-transform operation, e.g., $F[g(t)] = G(f)$, and the symbol $F^{-1}[\cdot]$ indicates the inverse Fourier-transform operation, e.g., $F^{-1}[G(f)] = g(t)$.
8. The symbol \star denotes convolution, e.g.,

$$x(t) \star h(t) = \int_{-\infty}^{\infty} x(\tau)h(t - \tau) d\tau$$

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 of mid-band 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 $\text{var}[\]$ means the variance of the random variable enclosed within.
20. The symbol $\text{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 k th elements of $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$, respectively. The L_1 norm of the vector product $\boldsymbol{\alpha} \cdot \boldsymbol{\beta}$ is defined by

$$\|\boldsymbol{\alpha} \cdot \boldsymbol{\beta}\|_1 = \sum_{m=1}^M \alpha_m \beta_m$$

Functions

1. Rectangular function:

$$\text{rect}(t) = \begin{cases} 1, & -\frac{1}{2} < t < \frac{1}{2} \\ 0, & |t| > \frac{1}{2} \end{cases}$$

2. Unit step function:

$$u(t) = \begin{cases} 1, & t > 0 \\ 0, & t < 0 \end{cases}$$

3. Signum function:

$$\text{sgn}(t) = \begin{cases} 1, & t > 0 \\ 0, & t = 0 \\ -1, & t < 0 \end{cases}$$

4. (Dirac) delta function:

$$\delta(t) = 0, \quad t \neq 0$$

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

or, equivalently,

$$\int_{-\infty}^{\infty} g(t) \delta(t - t_0) dt = g(t_0)$$

5. Sinc function:

$$\text{sinc}(x) = \frac{\sin(\pi x)}{\pi x}$$

6. Sine integral:

$$\text{Si}(u) = \int_0^u \frac{\sin x}{x} dx$$

7. Error function:

$$\text{erf}(u) = \frac{2}{\sqrt{\pi}} \int_0^u \exp(-x^2) dx$$

Complementary error function:

$$\text{erfc}(u) = 1 - \text{erf}(u).$$

8. Binomial coefficient

$$\binom{n}{k} = \frac{n!}{(n-k)!k!}$$

9. Bessel function of the first kind of order n :

$$J_n(x) = \frac{1}{2\pi} \int_{-\pi}^{\pi} \exp(jx \sin \theta - jn\theta) d\theta$$

10. Modified Bessel function of the first kind of zero order:

$$I_0(x) = \frac{1}{2\pi} \int_{-\pi}^{\pi} \exp(x \cos \theta) d\theta$$

11. Confluent hypergeometric function

$${}_1F_1(a; b; x) = 1 + \frac{a}{b} \frac{x}{1!} + \frac{a(a+1)}{b(b+1)} \frac{x^2}{2!} + \dots$$

Abbreviations

A:	ampere
AC:	alternating current
ADC:	analog-to-digital converter
ADM:	adaptive delta modulation
ADPCM:	adaptive differential pulse-code modulation
ADSL:	asymmetric digital subscriber line
AM:	amplitude modulation
ANSI:	American National Standards Institute
APB:	adaptive prediction with backward estimation
APF:	adaptive prediction with forward estimation
AQB:	adaptive quantization with backward estimation

AQF:	adaptive quantization with forward estimation
ARQ:	automatic-repeat request
ASCII:	American National Standard Code for Information Interchange
ASK:	amplitude-shift keying
ATM:	asynchronous transfer mode
AWGN:	additive white Gaussian noise
b/s:	bits/second
BER:	bit error rate
BISDN:	broadband ISDN
BPF:	band-pass filter
BSC:	binary symmetric channel
CAP:	carrierless amplitude/phase modulation
CCITT:	Consultative Committee for International Telephone and Telegraph (Now renamed the ITU)
CDM:	code-division multiplexing
CDMA:	code-division multiple access
CELP:	code excited linear predictive (model)
CO:	central office
codec:	coder/decoder
CPFSK:	continuous-phase frequency-shift keying
CRC:	cyclic redundancy check
CW:	continuous wave
DAC:	digital-to-analog converter
dB:	decibel
dBW:	decibel referenced to 1 watt
dBmW:	decibel reference to 1 milliwatt
DC:	direct current
DEM:	demodulator
DES:	data encryption standard
DFT:	discrete Fourier transform
DM:	delta modulation
DMT:	discrete multitone
DPCM:	differential pulse-code modulation
DPSK:	differential phase-shift keying
DSB-SC:	double sideband-suppressed carrier
DS/BPSK:	direct sequence/binary phase-shift keying
DSL:	digital subscriber line
exp:	exponential
FDM:	frequency-division multiplexing
FDMA:	frequency-division multiple access
FEXT:	far-end crosstalk
FFT:	fast Fourier transform
FH:	frequency hop

FH/MFSK:	frequency hop/M-ary frequency-shift keying
FMPB:	frequency modulator with feedback
FSK:	frequency-shift keying
GMSK:	Gaussian filtered MSK
GSM:	global system for mobile communication
HDTV:	high definition television
Hz:	Hertz
IDFT:	inverse discrete Fourier transform
IF:	intermediate frequency
I/O:	input/output
IP:	internet protocol
IS-95:	intermediate standard-95
ISDN:	integrated services digital network
ISI:	intersymbol interference
ISO:	International Organization for Standardization
ITU:	International Telecommunications Union
JPEG:	joint photographic experts group
LAN:	local-area network
LDM:	linear delta modulation
LMS:	least-mean-square
log:	natural logarithm
log₂:	logarithm to base 2
log₁₀:	logarithm to base 10
LPC:	linear predictive coding (model)
LPF:	low-pass filter
MAP:	maximum <i>a posteriori</i> probability
ML:	maximum likelihood
mmse:	minimum mean-square error
modem:	modulator-demodulator
MPEG:	motion photographic experts group
ms:	millisecond
μs:	microsecond
MSK:	minimum shift keying
NCO:	number-controlled oscillator
NEXT:	near-end crosstalk
nm:	nanometer
NRZ:	nonreturn-to-zero
NTSC:	National Television Systems Committee
OC:	optical carrier
OFDM:	orthogonal frequency-division multiplexing
OOK:	on-off keying
OSI:	open systems interconnection
PAM:	pulse-amplitude modulation

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:	quadrature phase-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

BIBLIOGRAPHY

■ BOOKS

- M. Abramowitz and I.A. Stegun, *Handbook of Mathematical Functions with Formulas, Graphs, and Mathematical Tables* (New York: Dover Publications, 1965).
- N. Abramson, *Information Theory and Coding* (New York: McGraw-Hill, 1963).
- J. Adamék, *Foundations of Coding* (New York: Wiley, 1991).
- Y. Akaiwa, *Introduction to Digital Mobile Communication* (New York: Wiley, 1997).
- J.B. Anderson, T. Aulin, and C.E. Sundberg, *Digital Phase Modulation* (New York: Plenum Publishers, 1986).
- J.B. Anderson and S. Mohan, *Source and Channel Coding: An Algorithmic Approach* (Boston, Mass: Kluwer Academic, 1991).
- J.B. Anderson, *Digital Transmission Engineering* (Piscataway, N.J.: IEEE Press, 1999).
- R.B. Ash, *Information Theory* (New York: Wiley, 1965).
- Bell Laboratories Technical Staff, *A History of Engineering Science in the Bell System: The Early Years (1875–1925)*, (Books on Demand, Ann Arbor, Michigan: 1975).
- J.C. Bellamy, *Digital Telephony*, Second Edition (New York: Wiley, 1991).
- S. Benedetto and E. Biglieri, *Principles of Digital Transmission with Wireless Applications* (New York: Kluwer Academic/Plenum Publishers, 1999).
- S. Benedetto, E. Biglieri, and V. Castellani, *Digital Transmission Theory* (Englewood Cliffs, N.J.: Prentice-Hall, 1987).
- W.R. Bennett, *Introduction to Signal Transmission* (New York: McGraw-Hill, 1970).
- K.B. Benson and J.C. Whitaker, *Television Engineering Handbook*, rev. ed. (New York: McGraw-Hill, 1992).
- T. Berger, *Rate Distortion Theory: A Mathematical Basis for Data Compression* (Englewood Cliffs, N.J.: Prentice-Hall, 1971).
- E.R. Berlekamp, *Algebraic Coding Theory* (New York: McGraw-Hill, 1968).
- E.R. Berlekamp (editor), *Key Papers in the Development of Coding Theory* (Piscataway, N.J.: IEEE Press, 1974).
- V.K. Bhargava, D. Haccoun, R. Matyas, and P. Nuspl, *Digital Communications by Satellite: Modulation, Multiple Access, and Coding* (New York: Wiley, 1981).
- E. Biglieri, D. Divsalar, P.J. McLane, and M.K. Simon, *Introduction to Trellis-Coded Modulation with Applications* (New York: Macmillan, 1991).
- J.A.C. Bingham, *The Theory and Practice of Modem Design* (New York: Wiley, 1988).
- R.B. Blachman and J.W. Tukey, *The Measurement of Power Spectra, from the Point of View of Communication Engineering* (New York: Dover, 1958).
- H.S. Black, *Modulation Theory* (Princeton, N.J.: Van Nostrand, 1953).
- R.E. Blahut, *Principles and Practice of Information Theory* (Reading, Mass.: Addison-Wesley, 1987).
- R.E. Blahut, *Digital Transmission of Information* (Reading, Mass.: Addison-Wesley, 1990).
- G.E.P. Box and G.M. Jenkins, *Time Series Analysis: Forecasting and Control* (San Francisco: Holden Day, 1976).
- R.N. Bracewell, *The Fourier Transform and Its Applications*, 2nd ed., rev. (New York: McGraw-Hill, 1986).
- L. Brillouin, *Science and Information Theory*, 2nd ed. (New York: Academic Press, 1962).
- K.W. Cattermole, *Principles of Pulse-code Modulation* (New York: American Elsevier, 1969).
- W.Y. Chen, *DSL: Simulation Techniques and Standards Development for Digital Subscriber Line Systems* (Indianapolis, Ind.: Macmillan Technical Publishing, 1998).
- J.M. Cioffi, *Digital Data Transmission*, EE379C Course Textbook, Stanford University, 1998.

- G.C. Clark, Jr., and J.B. Cain, *Error-correction Coding for Digital Communications* (New York: Plenum Publishers, 1981).
- L.W. Couch, *Digital and Analog Communication Systems*, 5th ed. (Englewood Cliffs, N.J.: Prentice-Hall, 1997).
- T.M. Cover and J.A. Thomas, *Elements of Information Theory* (New York: Wiley, 1991).
- H. Cramér and M.R. Leadbetter, *Stationary and Related Stochastic Processes: Sample Function Properties and Their Applications* (New York: Wiley, 1967).
- W.B. Davenport, Jr., and W.I. Root, *An Introduction to the Theory of Random Signals and Noise* (New York: McGraw-Hill, 1958).
- R.C. Dixon, *Spread Spectrum Systems*, 2nd ed. (New York: Wiley, 1984).
- R.C. Dixon (editor), *Spread Spectrum Techniques* (New York: IEEE Press, 1976).
- L.J. Doob, *Stochastic Processes* (New York: Wiley, 1953).
- J.J. Downing, *Modulation Systems and Noise* (Englewood Cliffs, N.J.: Prentice-Hall, 1964).
- W. Feller, *An Introduction to Probability Theory and its Application*, vol. 1, 3rd ed. (New York: Wiley, 1968).
- T.L. Fine, *Theories of Probability: An Examination of Foundations* (New York: Academic Press, 1973).
- L.E. Franks (editor), *Data Communication: Fundamentals of Baseband Transmission* (Dowden, Hurchison, and Ross, 1974).
- L.E. Franks, *Signal Theory* (Englewood Cliffs, N.J.: Prentice-Hall, 1969).
- R.L. Freeman, *Telecommunications Transmission Handbook*, 4th ed. (New York: Wiley, 1998).
- R.M. Gagliardi, *Introduction to Communications Engineering*, 2nd ed. (New York: Wiley, 1988).
- R.G. Gallager, *Information Theory and Reliable Communication* (New York: Wiley, 1968).
- R.G. Gallager, *Low-Density Parity-Check Codes* (Cambridge, Mass.: MIT Press, 1963).
- F.M. Gardner, *Phaselock Techniques*, 2nd ed. (New York: Wiley, 1979).
- V.K. Gary and J.E. Wilkes, *Principles & Applications of GSM* (Englewood Cliffs, N.J.: Prentice-Hall, 1999).
- A. Gersho and R.M. Gray, *Vector Quantization and Signal Compression* (Boston, Mass.: Kluwer Academic, 1992).
- J.D. Gibson (editor), *The Mobile Communications Handbook* (Piscataway, N.J.: IEEE Press, 1996).
- R.D. Gitlin, J.F. Hayes, and S.B. Weinstein, *Data Communications Principles* (New York: Plenum, 1992).
- B. Goldberg and H.S. Bennett (editors), *Communication Channels: Characterization and Behavior* (New York: IEEE Press, 1976).
- S.W. Golomb (editor), *Digital Communications with Space Applications* (Englewood Cliffs, N.J.: Prentice-Hall, 1964).
- S.W. Golomb, *Shift Register Sequences* (San Francisco: Holden-Day, 1967).
- R.M. Gray and L.D. Davisson, *Random Processes: A Mathematical Approach for Engineers* (Englewood Cliffs, N.J.: Prentice-Hall, 1986).
- P.E. Green, Jr., *Computer Network Architectures and Protocols* (New York: Plenum, 1982).
- P.E. Green, Jr., *Fiber Optic Networks* (Englewood Cliffs, N.J.: Prentice-Hall, 1993).
- M.S. Gupta (editor), *Electrical Noise: Fundamentals and Sources* (New York: IEEE Press, 1977).
- R.W. Hamming, *Coding and Information Theory* (Englewood Cliffs, N.J.: Prentice-Hall, 1980).
- S. Haykin, *Communication Systems*, 3rd ed. (New York: Wiley, 1994).
- S. Haykin, *Adaptive Filter Theory*, 3rd ed. (Englewood Cliffs, N.J.: Prentice-Hall, 1996).
- S. Haykin and B. Van Veen, *Signals and Systems* (New York: Wiley, 1999).
- C. Heegard and S.B. Wicker, *Turbo Coding* (Boston, Mass.: Kluwer Academic Publishers, 1999).
- G. Held, *The Complete Modem Reference*, 3rd ed. (New York: Wiley, 1997).
- C.W. Helstrom, *Statistical Theory of Signal Detection* (Elmsford, N.Y.: Pergamon Press, 1968).
- C.W. Helstrom, *Probability and Stochastic Processes for Engineers*, 2nd ed. (New York: Macmillan, 1990).
- K. Henney (editor), *Radio Engineering Handbook* (New York: McGraw-Hill, 1959).
- J.K. Holmes, *Coherent Spread Spectrum Systems* (New York: Wiley, 1982).
- W.C. Jakes, Jr. (editor), *Microwave Mobile Communications* (New York: Wiley, 1974).

- N.S. Jayant and P. Noll, *Digital Coding of Waveforms: Principles and Applications to Speech and Video* (Englewood Cliffs, N.J.: Prentice-Hall, 1984).
- N.S. Jayant (editor), *Waveform Quantization and Coding* (New York: IEEE Press, 1976).
- H. Jeffreys, Sir, *Theory of Probability*, 3rd ed. (Oxford: Clarendon Press, 1967).
- H. Jeffreys, Sir, and B.S. Jeffreys, *Methods of Mathematical Physics*, 3rd ed. (Cambridge University Press, 1956).
- M.C. Jeruchim, B. Balaban, and J.S. Shanmugan, *Simulation of Communication Systems* (New York: Plenum, 1992).
- E.C. Jordan and K.G. Balmain, *Electromagnetic Waves and Radiating Systems*, 2nd ed. (Englewood Cliffs, N.J.: Prentice Hall, 1968).
- A. Khintchin, *Mathematical Foundations of Information Theory* (New York: Dover, 1957).
- A.N. Kolmogorov, *Foundations of the Theory of Probability* (New York: Chelsea Publishing, 1956).
- V.A. Kotelnikov, *The Theory of Optimum Noise Immunity* (New York: McGraw-Hill, 1960).
- J.D. Kraus, *Antennas* (New York: McGraw-Hill, 1950).
- S. Kullback, *Information Theory and Statistics* (New York: Dover, 1968).
- P. Lafrance, *Fundamental Concepts in Communication* (Englewood Cliffs, N.J.: Prentice-Hall, 1990).
- B.P. Lathi, *Modern Digital and Analog Communication Systems*, 2nd ed. (Oxford University Press, 1995).
- I. Lebow, *Information Highways and Byways* (Piscataway, N.J.: IEEE Press, 1995).
- E.A. Lee and D.G. Messerschmitt, *Digital Communication*, 2nd ed. (Boston, Mass.: Kluwer Academic, 1994).
- J.S. Lee and L.E. Miller, *CDMA Systems Engineering Handbook* (Boston, Mass.: Artech House Publishers, 1998).
- Y.W. Lee, *Statistical Theory of Communication* (New York: Wiley, 1960).
- W.C.(Y.) Lee, *Mobile Communications Engineering* (New York: McGraw-Hill, 1982).
- A. Leon-Garcia, *Probability and Random Processes for Electrical Engineering*, 2nd ed. (Reading, Mass.: Addison-Wesley, 1994).
- C.R. Lewart, *The Ultimate Modem Handbook* (Englewood Cliffs, N.J.: Prentice-Hall, 1998).
- S. Lin and D.J. Costello, Jr., *Error Control Coding: Fundamentals and Applications* (Englewood Cliffs, N.J.: Prentice-Hall, 1983).
- W.C. Lindsey, *Synchronization Systems in Communication and Control* (Englewood Cliffs, N.J.: Prentice-Hall, 1972).
- W.C. Lindsey and M.K. Simon (editors), *Phase-locked Loops and Their Applications* (New York: IEEE Press, 1978).
- W.C. Lindsey and M.K. Simon, *Telecommunication Systems Engineering* (Englewood Cliffs, N.J.: Prentice-Hall, 1973).
- M. Loève, *Probability Theory* (Princeton, N.J.: Van Nostrand, 1963).
- R.W. Lucky, *Silicon Dreams: Information, Man, and Machine* (New York: St. Martin's Press, 1989).
- R.W. Lucky, J. Salz, and E.J. Weldon, Jr., *Principles of Data Communication* (New York: McGraw-Hill, 1968).
- F.J. MacWilliams and N.J.A. Sloane, *The Theory of Error-correcting Codes* (Amsterdam: North-Holland, 1977).
- V.K. Madisetti and D.B. Williams (editors), *The Digital Signal Processing Handbook* (Piscataway, N.J.: IEEE Press, 1998).
- R.J. Marks II, *Introduction to Shannon Sampling and Interpolation Theory* (New York/Berlin: Springer-Verlag, 1991).
- J.C. McDonald (editor), *Fundamentals of Digital Switching*, 2nd ed. (New York: Plenum, 1990).
- R. McDonough and A.D. Whalen, *Detection of Signals in Noise*, 2nd ed. (New York: Academic Press, 1995).
- R.J. McEliece, *The Theory of Information and Coding: A Mathematical Framework for Communication* (Reading, Mass.: Addison-Wesley, 1977).
- A. Mengali and N. D'Andrea, *Synchronization Techniques for Digital Receivers* (New York: Plenum, 1997).

- D.J.G. Mestdagh, *Fundamentals of Multiaccess Optical Fiber Networks* (Boston, Mass.: Artech House Publishers, 1995).
- C.H. Meyer and S.M. Matyas, *Cryptography: A New Dimension in Computer Data Security* (New York: Wiley, 1982).
- H. Meyr and G. Ascheid, *Synchronization in Digital Communications*, vol. 1 (New York: Wiley, 1990).
- H. Meyr, M. Moeneclaey, and S.A. Fechtel, *Digital Communication Receivers: Synchronization, Channel Estimation and Signal Processing* (New York: Wiley, 1998).
- A.M. Michelson and A.H. Levesque, *Error-control Techniques for Digital Communication* (New York: Wiley, 1985).
- D. Middleton, *An Introduction to Statistical Communication Theory* (New York: McGraw-Hill, 1960).
- J.G. Nellist, *Understanding Telecommunications and Lightwave Systems: An Entry Level Guide* (Piscataway, N.J.: IEEE Press, 1992).
- C.F.J. Overhage (editor), *The Age of Electronics* (New York: McGraw-Hill, 1962).
- P.F. Panter, *Modulation, Noise and Spectral Analysis, Applied to Information Transmission* (New York: McGraw-Hill, 1965).
- A. Papoulis, *Probability, Random Variables, and Stochastic Processes*, 2nd ed. (New York: McGraw-Hill, 1984).
- J.D. Parsons, *The Mobile Radio Propagation Channel* (New York: Wiley, 1992).
- K. Pahlavan and A.H. Levesque, *Wireless Information Networks* (New York: Wiley, 1996).
- J. Pearl, *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference* (San Mateo, Calif: Morgan Kaufman Publishers, 1988).
- W.W. Peterson and E.J. Weldon, Jr., *Error Correcting Codes*, 2nd ed. (Cambridge, Mass.: MIT Press, 1972).
- J.R. Pierce and A.M. Noll, *Signals: The Science of Telecommunications* (New York: Scientific American Library, 1990).
- J.R. Pierce, *Symbols, Signals and Noise: The Nature and Process of Communication* (New York: Harper, 1961).
- H.V. Poor, *An Introduction to Signal Detection and Estimation*, 2nd ed. (New York/Berlin: Springer-Verlag, 1994).
- T. Pratt and C.W. Bostian, *Satellite Communications* (New York: Wiley, 1986).
- J.G. Proakis, *Digital Communications*, 3rd ed. (New York: McGraw-Hill, 1995).
- L.R. Rabiner and R.W. Schafer, *Digital Processing of Speech Signals* (Englewood Cliffs, N.J.: Prentice-Hall, 1978).
- K.R. Rao and P. Yip, *Discrete Cosine Transform: Algorithms, Advantages, Applications* (New York: Academic Press, 1990).
- T.S. Rappaport, *Smart Antennas* (Piscataway, N.J.: IEEE Press, 1998).
- T.S. Rappaport, *Wireless Communications: Principles and Practice* (Piscataway, N.J.: IEEE Press, 1996).
- S.O. Rice, "Noise in FM receivers," in M. Rosenblatt (editor), *Proceedings of the Symposium on Time Series Analysis* (New York: Wiley, 1963), pp. 395-411.
- J.H. Roberts, *Angle Modulation: The Theory of Systems Assessment*, IEE Communication Series 5 (London: Institution of Electrical Engineers, 1977).
- H.E. Rowe, *Signals and Noise in Communication Systems* (Princeton, N.J.: Van Nostrand, 1965).
- D.J. Sakrison, *Communication Theory: Transmission of Waveforms and Digital Information* (New York: Wiley, 1968).
- C. Schlegel, *Trellis Coding* (Piscataway, N.J.: IEEE Press, 1997).
- M. Schwartz, W.R. Bennett, and S. Stein, *Communication Systems and Techniques* (New York: McGraw-Hill, 1966).
- M. Schwartz, *Information Transmission, Modulation and Noise: A Unified Approach*, 3rd ed. (New York: McGraw-Hill, 1980).
- M. Schwartz, *Telecommunication Networks: Protocols, Modeling, and Analysis* (Reading, Mass.: Addison Wesley, 1987).

- K.S. Shanmugan, *Digital and Analog Communication Systems* (New York: Wiley, 1979).
- C.E. Shannon and W. Weaver, *The Mathematical Theory of Communication* (Urbana: University of Illinois Press, 1949).
- G.J. Simmons (editor), *Contemporary Cryptology: The Science of Information Integrity* (Piscataway, N.J.: IEEE Press, 1992).
- M.K. Simon, J.K. Omura, R.A. Scholtz, and B.K. Levitt, *Spread Spectrum Communications*, vols. I, II, and III (New York: Computer Science Press, 1985).
- B. Sklar, *Digital Communications: Fundamentals and Applications* (Englewood Cliffs, N.J.: Prentice-Hall, 1988).
- D. Slepian (editor), *Key Papers in the Development of Information Theory* (New York: IEEE Press, 1974).
- N.J.A. Sloane and A.D. Wyner, *Claude Shannon: Collected Papers* (Piscataway, N.J.: IEEE Press, 1993).
- D.R. Smith, *Digital Transmission Systems* (Princeton, N.J.: Van Nostrand Reinhold, 1985).
- I.S. Sokolnikoff and R.M. Redheffer, *Mathematics of Physics and Modern Engineering* (New York: McGraw-Hill, 1966).
- J.J. Spilker, Jr., *Digital Communications by Satellite* (Englewood Cliffs, N.J.: Prentice-Hall, 1977).
- W. Stallings, *ISDN and Broadband ISDN*, 2nd ed. (New York: Macmillan, 1992).
- T. Starr, J.M. Cioffi, and P.J. Silverman, *Understanding Digital Subscriber Line Technology* (Englewood Cliffs, N.J.: Prentice-Hall, 1999).
- R. Steele, *Delta Modulation Systems* (New York: Wiley, 1975).
- R. Steele and L. Hanzo (editors), *Mobile Radio Communications*, 2nd ed. (New York: Wiley, 1999).
- J.J. Stiffler, *Theory of Synchronous Communications* (Englewood Cliffs, N.J.: Prentice-Hall, 1971).
- E.D. Sunde, *Communications Systems Engineering Theory* (New York: Wiley, 1969).
- A.S. Tanenbaum, *Computer Networks*, 2nd ed. (Englewood Cliffs, N.J.: Prentice-Hall, 1995).
- S. Tantarana and K.M. Ahmed, *Wireless Applications of Spread Spectrum Systems: Selected Readings* (Piscataway, N.J.: IEEE Press, 1998).
- T.M. Thompson, *From Error-correcting Codes Through Sphere Packings to Simple Groups* (The Mathematical Association of America, Washington D.C.: 1983).
- D.J. Torrieri, *Principles of Military Communication Systems*, 2nd ed. (Boston, Mass.: Artech House Publishers, 1992).
- G.L. Turin, *Notes on Digital Communications* (Princeton, N.J.: Van Nostrand-Reinhold, 1969).
- A. Van der Ziel, *Noise: Source, Characterization, Measurement* (Englewood Cliffs, N.J.: Prentice-Hall, 1970).
- H.F. Vanlandingham, *Introduction to Digital Control Systems* (New York: Macmillan, 1985).
- H.C.A. van Tilborg, *An Introduction to Cryptology* (Boston, Mass.: Kluwer, 1988).
- H.L. Van Trees, *Detection, Estimation, and Modulation Theory*, Part I (New York: Wiley, 1968).
- A.J. Viterbi, *Principles of Coherent Communication* (New York: McGraw-Hill, 1966).
- A.J. Viterbi and J.K. Omura, *Principles of Digital Communication and Coding* (New York: McGraw-Hill, 1979).
- G.N. Watson, *A Treatise in the Theory of Bessel Functions*, 2nd ed. (New York: Cambridge University Press, 1966).
- N. Wax (editor), *Selected Papers on Noise and Stochastic Processes* (New York: Dover Publications, 1954).
- E.T. Whittaker and G.N. Watson, *A Course in Modern Analysis*, 4th ed. (New York: Cambridge University Press, 1952).
- S.B. Wicker and V.K. Bhargava (editors), *Reed-Solomon Codes* (Piscataway, N.J.: IEEE Press, 1994).
- B. Widrow and S.D. Stearns, *Adaptive Signal Processing* (Englewood Cliffs, N.J.: Prentice-Hall, 1985).
- N. Wiener, *The Extrapolation, Interpolation, and Smoothing of Stationary Time Series, with Engineering Applications* (New York: Wiley, 1949).
- S.G. Wilson, *Digital Modulation and Coding* (Englewood Cliffs, N.J.: Prentice-Hall, 1996).
- E. Wong, *Stochastic Processes in Information and Dynamical Systems* (New York: McGraw-Hill, 1971).

- P.M. Woodward, *Probability and Information Theory, with Applications to Radar*, 2nd ed. (Elmsford, N.Y.: Pergamon Press, 1964).
- J.M. Wozencraft and I.M. Jacobs, *Principles of Communication Engineering* (New York: Wiley, 1965).
- W.W. Wu, *Elements of Digital Satellite Communication*, vol. I (New York: Computer Science Press, 1984).
- C.R. Wylie and L.C. Barrett, *Advanced Engineering Mathematics*, 5th ed. (New York: McGraw-Hill, 1982).
- R.D. Yates and D.J. Goodman, *Probability and Stochastic Processes: A Friendly Introduction for Electrical and Computer Engineers* (New York: Wiley, 1999).
- J.H. Yuen (editor), *Deep Space Telecommunications Systems Engineering* (New York: Plenum, 1983).
- R.E. Ziemer and R.L. Peterson, *Digital Communications and Spread Spectrum Systems* (New York: Macmillan, 1985).
- R.E. Ziemer and W.H. Tranter, *Principles of Communications*, 3rd ed. (Boston, Mass.: Houghton Mifflin, 1990).

■ PAPERS, REPORTS, PATENTS¹

- M.R. Aaron and D.W. Tufts, "Intersymbol interference and error probability," *IEEE Trans. on Information Theory*, vol. IT-12, pp. 26–34, 1966.
- J.E. Abate, "Linear and adaptive delta modulation," *Proceedings of the IEEE*, vol. 55, pp. 298–308, 1967.
- A.N. Akansu, P. Duhamel, X. Lin, and M. de Courville, "Orthogonal transmultiplexers in communications: A review," *IEEE Transactions on Signal Processing*, vol. 46, pp. 979–995, 1998.
- Y. Akaïwa and Y. Nagata, "Highly efficient digital mobile communications with a linear modulation method," *IEEE Journal on Selected Areas in Communications*, vol. SAC-5, pp. 890–895, 1987.
- O. Al-Shaykh, R. Neff, D. Taubman, and A. Zakhour, "Video sequence compression." In V.K. Madisetti and D.B. Williams (editors), *The Digital Signal Processing Handbook*, CRC Press, pp. 55-1–55-19, 1998.
- F. Amoroso, "The bandwidth of digital data signals," *IEEE Communications Magazine*, vol. 18, no. 6, pp. 13–24, 1980.
- J.B. Anderson and D.P. Taylor, "A bandwidth-efficient class of signal space codes," *IEEE Transactions on Information Theory*, vol. IT-24, pp. 703–712, 1978.
- R.R. Anderson and J. Salz, "Spectra of digital FM," *Bell System Tech. J.*, vol. 44, pp. 1165–1189, 1965.
- R. Arens, "Complex processes for envelopes of normal noise," *IRE Trans. on Information Theory*, vol. IT-3, pp. 204–207, 1957.
- E.H. Armstrong, "A method of reducing disturbances in radio signaling by a system of frequency modulation," *Proceedings of the IRE*, vol. 24, pp. 689–740, 1936.
- E. Arthurs and H. Dym, "On the optimum detection of digital signals in the presence of white Gaussian noise—A geometric interpretation and a study of three basic data transmission systems," *IRE Trans. on Communication Systems*, vol. CS-10, pp. 336–372, 1962.
- B.S. Atal and J.R. Remde, "A new model of LPC excitation for producing natural-sounding speech at low bit rates," *Proc. ICASSP '82*, pp. 614–17, 1982.
- B.S. Atal and M.R. Schroeder, "Stochastic coding of speech signals at very low bit rates," *IEEE International Conference on Communications*, May 1984.
- M. Austin, "Decision-feedback equalization for digital communication over dispersive channels," *MIT Research Laboratory of Electronics Technical Report 461*, 1967.
- E. Ayanoglu, N.R. Dagdeviren, J.E. Mazo, and R. Saltzberg, "High-speed modem synchronized to a remote codec," United States Patent 5,394,437, February 28, 1995.
- E. Ayanoglu, N.R. Dagdeviren, G.D. Golden, and J.E. Mazo, "An equalizer design technique for the PCM modem: a new model for the digital public switched network," *IEEE Transactions on Communications*, vol. 46, pp. 763–774, 1998.

- 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 symboles 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, 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. Thitimajshima, "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, 1966.
- 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-

- tions," tutorial notes, TU18, International Conference on Communications '99, Vancouver, British Columbia, Canada, June, 1999.
- A.C. Clarke, "Extraterrestrial relays," *Wireless World*, vol. 51, pp. 305-308, October 1945.
- C.E. Cook and H.S. Marsh, "An introduction to spread spectrum," *IEEE Communications Magazine*, vol. 21, no. 2, pp. 8-16, 1983.
- J.P. Costas, "Synchronous communications," *Proceedings of the IRE*, vol. 44, pp. 1713-1718, 1956.
- J.P. Costas, "Poisson, Shannon, and the radio amateur," *Proceedings of the IRE*, vol. 47, pp. 2058-2068, 1959.
- M.G. Crosby, "Frequency modulation noise characteristics," *Proceedings of the IRE*, vol. 25, pp. 472-514, April 1937.
- C.C. Cutler, "Differential quantization of communication signals," United States Patent 2-505-361, 1952.
- C.L. Dammann, L.D. McDaniel and C.L. Maddox, "D2 channel bank—Multiplexing and coding," *Bell System Tech. J.* vol. 51, pp. 1675-1699, 1972.
- F. Daneshgaran and M. Mondin, "Design of interleavers for turbo codes: Iterative interleaver growth algorithms of polynomial complexity," *IEEE Transactions on Information Theory*, vol. 45, pp. 1845-1859, 1999.
- R. deBuda, "Coherent demodulation of frequency-shift keying with low deviation ratio," *IEEE Trans. on Communications*, vol. COM-20, pp. 429-435, 1972.
- F.E. DeJager, "Deltamodulation, a method of PCM transmission using the 1-unit code," *Phillips Research Reports*, vol. 7, pp. 442-46, 1952.
- F.E. DeJager and C.B. Dekker, "Tamed frequency modulation: A novel method to achieve spectrum economy in digital transmission," *IEEE Transactions on Communications*, vol. COM-26, pp. 534-542, 1978.
- J.A. Develet, "A threshold criterion for phase-lock demodulation," *Proceedings of the IEEE*, vol. 51, pp. 349-356, 1963.
- W. Diffie and M.E. Hellman, "New directions in cryptography," *IEEE Transactions on Information Theory*, vol. IT-22, pp. 644-654, 1976.
- W. Diffie and M.E. Hellman, "Privacy and authentication: An introduction to cryptography," *Proceedings of the IEEE*, vol. 67, pp. 397-427, 1979.
- D. Divsalar, "Turbo codes," MILCOM 96 tutorial, San Diego, November 1996.
- M.I. Doelz and E.H. Heald, "Minimum shift data communication system," U.S. Patent 2977417, March 1961.
- R.M. Dolby, "An audio reduction system," *Journal of the Audio Engineering Society*, vol. 15, p. 383, 1967.
- J. Dungundji, "Envelopes and pre-envelopes of real wave-forms," *IRE Transactions on Information Theory*, vol. IT-4, pp. 53-57, 1958.
- P. Elias, "Coding for noisy channels," *IRE Convention Record*, Part 4, pp. 37-46, March 1955.
- L.H. Enloe, "Decreasing the threshold in FM by frequency feedback," *Proceedings of the IRE*, vol. 50, pp. 18-30, 1962.
- V.M. Eyuboglu, "Detection of coded modulation signals on linear, severely distorted channels using decision-feedback noise prediction with interleaving," *IEEE Transactions on Communications*, vol. COM-36, pp. 401-409, 1988.
- D.D. Falconer, "Carrierless AM/PM," *Bell Laboratories, Internal Memorandum*, July 3, 1975.
- K. Feher, "MODEMS for emerging digital cellular-mobile radio system," *IEEE Transactions on Vehicular Technology*, vol. 40, pp. 355-365, 1991.
- J.L. Flanagan, M.R. Schroeder, B.S. Atal, R.E. Crochiere, N.S. Jayant, and J.M. Tribolet, "Speech coding," *IEEE Transactions on Communications*, vol. COM-27, pp. 710-737, 1979.
- B. LeFloch, R. Halbert-Lassalle, and D. Castelain, "Digital sound broadcasting to mobile receivers," *IEEE Transactions on Broadcasting*, vol. 35, pp. 493-503, 1989.
- G.D. Forney, Jr., "Maximum likelihood sequence estimation of digital sequences in the presence of intersymbol interference," *IEEE Transactions on Information Theory*, vol. IT-18, pp. 363-378, 1972.
- G.D. Forney, Jr. "The Viterbi algorithm," *Proceedings of the IEEE*, vol. 61, pp. 268-278, 1973.

- G.D. Forney, Jr., and M.V. Eyuboglu, "Combined equalization and coding using precoding," *IEEE Communications Magazine*, vol. 29, no. 12, pp. 25-34, 1991.
- G.D. Forney, Jr., L. Brown, M.V. Eyuboglu, and J.L. Moran III, "The V.34 high-speed modem standard," *IEEE Communications Magazine*, pp. 28-93, December 1996.
- L.E. Franks, "Carrier and bit synchronization in data communications—A tutorial review," *IEEE Transactions on Communications*, vol. COM-28, pp. 1107-1121, 1980.
- B.J. Frey and D.J.C. MacKay, "Irregular turbocodes," *Proceedings of the 37th Annual Allerton Conference on Communication, Control, and Computing*, Allerton House, Illinois, September 1999.
- H.T. Friis, "Noise figures in radio receivers," *Proceedings of the IRE*, vol. 32, pp. 419-422, 1944.
- K.E. Fulz and D.B. Penick, "T1 carrier system," *Bell System Tech. J.*, vol. 44, pp. 1405-1451, 1965.
- D. Gabor, "Theory of communications," *Journal of IEE (London)*, vol. 93, Part III, pp. 429-457, 1946.
- D.L. Gall, "MPEG: a video compression standard for multimedia applications," *Communications of the ACM*, vol. 34, pp. 47-58, 1991.
- R.G. Gallager, "Low-density parity-check codes," *IRE Transactions on Information Theory*, vol. 8, pp. 21-28, 1962.
- W.A. Gardner, "Introduction to Einstein's contribution to time-series analysis," *IEEE ASSP Magazine*, vol. 4, pp. 4-5, October 1987.
- W.A. Gardner and L.E. Franks, "Characterization of cyclostationary random signal processes," *IEEE Transactions on Information Theory*, vol. IT-21, pp. 4-14, 1975.
- D.A. George, "Matched filters for interfering signals," *IEEE Transactions on Information Theory*, vol. IT-11, pp. 153-154, 1965.
- A. Gersho, "Adaptive equalization of highly dispersive channels for data transmission," *Bell System Tech. J.*, vol. 48, pp. 55-70, 1969.
- R.A. Gibby and J.W. Smith, "Some extensions of Nyquist's telegraph transmission theory," *Bell Systems Tech. J.*, vol. 44, pp. 1487-1510, 1965.
- R.D. Gitlin and E.Y. Ho, "The performance of staggered quadrature amplitude modulation in the presence of phase jitter," *IEEE Transactions on Communications*, vol. COM-23, pp. 348-352, 1975.
- M.J.E. Golay, "Note on digital coding," *Proceedings of the IRE*, vol. 37, p. 657, 1949.
- M.J.E. Golay, "Binary coding," *IRE Transactions on Information Theory*, vol. PGIT-4, pp. 23-28, 1954.
- R. Gold, "Optimal binary sequences for spread spectrum multiplexing," *IEEE Transactions on Information Theory*, vol. IT-13, pp. 619-621, 1967.
- R. Gold, "Maximal recursive sequences with 3-valued recursive cross correlation functions," *IEEE Transactions on Information Theory*, vol. IT-14, pp. 154-156, 1968.
- B. Goode, "Scanning the issue: Special issue on global information infrastructure," *Proceedings of the IEEE*, vol. 85, pp. 1883-1886, 1997.
- R.M. Gray, "Vector quantization," *IEEE ASSP Magazine*, vol. 1, no. 2, pp. 4-29, 1984.
- W.J. Gruen, "Theory of AFC synchronization," *Proceedings of the IRE*, vol. 41, pp. 1043-1048, 1953.
- P. Guinand and J. Lodge, "Trellis termination for turbo encoders," *Proceedings of 18th Biennial Symposium on Communications, Queen's University, Kingston, Canada, June 1996*.
- D.W. Hagelbarger, "Recurrent codes: Easily mechanized, burst-correcting binary codes," *Bell System Tech. J.*, vol. 38, pp. 969-984, 1959.
- J. Hagenauer, E. Offer, and L. Papke, "Iterative decoding of binary block and convolutional codes," *IEEE Transactions on Information Theory*, vol. 42, pp. 429-445, 1996.
- J. Hagenauer and P. Hoehner, "A Viterbi algorithm with soft-decision outputs and its applications," *IEEE Globecom 89*, pp. 47.11-47.17, November 1989, Dallas, Texas.
- R.W. Hamming, "Error detecting and error correcting codes," *Bell System Tech. J.*, vol. 29, pp. 147-160, 1950.
- J.C. Hancock and R.W. Lucky, "Performance of combined amplitude and phase-modulated communication systems," *IRE Transactions on Communication Systems*, vol. CS-8, pp. 232-237, 1960.

- H.H. Hanning and J.W. Pan, "D2 channel bank system aspects," *Bell System Tech. J.*, vol. 51, pp. 1641-1657, 1972.
- H. Harashima and H. Miyakawa, "Matched-transmission technique for channels with intersymbol interference," *IEEE Transactions on Communications*, vol. COM-20, pp. 774-779, 1972.
- T.V.L. Hartley, "Transmission of information," *Bell System Tech. J.*, vol. 7, pp. 535-563, 1928.
- F.S. Hill, Jr., "On time-domain representations for vestigial sideband signals," *Proceedings of the IEEE*, vol. 62, pp. 1032-1033, 1974.
- D.A. Huffman, "A method for the construction of minimum redundancy codes," *Proceedings of the IRE*, vol. 40, pp. 1098-1101, 1952.
- P.A. Humblet and M.G. Troulis, "The information driveway," *IEEE Communications Magazine*, pp. 64-68, December, 1996.
- G.-H. Im and J.-J. Werner, "Bandwidth-efficient digital transmission over unshielded twisted-pair wiring," *IEEE Journal on Selected Areas in Communications*, vol. 13, pp. 1643-1655, 1995.
- H. Insoe, Y. Yasuda, and J. Murakami, "A telemetering system by code modulation: $\Delta\Sigma$ modulation," *IRE Transactions on Space Electronics and Telemetry*, vol. SET-8, pp. 204-209, 1962.
- M. Ishizuka and K. Hirade, "Optimum Gaussian filter and deviated-frequency locking scheme for coherent detection of MSK," *IEEE Transactions on Communications*, vol. COM-28, pp. 850-857, 1980.
- I.M. Jacobs, "Practical applications of coding," *IEEE Transactions on Information Theory*, vol. IT-20, pp. 305-310, 1974.
- N.S. Jayant, "Adaptive delta modulation with a one-bit memory," *Bell System Tech. J.*, vol. 49, pp. 321-342, 1970.
- N.S. Jayant, "Digital coding of speech waveforms, PCM, DPCM and DM quantizers," *Proceedings of the IEEE*, vol. 62, pp. 611-632, 1974.
- N.S. Jayant, "Coding speech at low bit rates," *IEEE Spectrum*, vol. 23, no. 8, pp. 58-63, 1986.
- A.J. Jerri, "The Shannon sampling theorem—its various extensions and applications: A tutorial review," *Proceedings of the IEEE*, vol. 65, no. 11, pp. 1565-1596, 1977.
- J.B. Johnson, "Thermal agitation of electricity in conductors," *Physical Review*, second series, vol. 32, pp. 97-109, 1928.
- P. Kabal and S. Pasupathy, "Partial-response signaling," *IEEE Transactions on Communications*, vol. COM-23, pp. 921-934, 1975.
- I. Kalat, "The multitone channel," *IEEE Transactions on Communications*, vol. 37, pp. 119-124, 1989.
- I. Kalat, J.E. Mazo, and B.R. Saltzberg, "The capacity of PCM voiceband channels," *IEEE International Conference on Communications*, pp. 507-511, Geneva, Switzerland, 1993.
- H. Kaneko, "A unified formulation of segment companding laws and synthesis of codes and digital companders," *Bell System Tech. J.*, vol. 49, pp. 1555-1588, 1970.
- A.I. Khintchine, "Korrelationstheorie der stationären stochastischen Prozesse," *Mathematische Annalen*, vol. 1, 109, pp. 415-458, 1934.
- H. Kobayashi, "Correlative level coding and maximum-likelihood decoding," *IEEE Transactions on Information Theory*, vol. IT-17, pp. 586-594, 1971.
- R. Kohno, "Spatial and temporal communication theory using adaptive antenna array," *IEEE Personal Communications*, pp. 28-35, February, 1998.
- E.T. Kretzmer, "Generalization of a technique for binary data communications," *IEEE Transactions on Communication Technology*, vol. COM-14, pp. 67-68, Feb. 1966.
- F.R. Kschischang and B.J. Frey, "Interactive decoding of compound codes by probability propagation in graphical models," *IEEE Journal on Selected Areas in Communication*, vol. 16, pp. 219-230, 1998.
- J.W. Lechleider, "Line codes for digital subscriber lines," *IEEE Communications Magazine*, vol. 27, pp. 25-32, September 1989.
- B.M. Leiner, V.G. Cerf, D.D. Clark, R.E. Kohn, L. Kleinrock, D.C. Lynch, J. Postel, L.G. Roberts, and S. Wolff, "A brief history of the Internet," *Commun. ACM*, vol. 40, pp. 102-108, February 1997.
- A. Lender, "The duobinary technique for high-speed data transmission," *IEEE Transactions on Communications and Electronics*, vol. 82, pp. 214-218, May 1963.

- A. Lender, "Correlative digital communication techniques," *IEEE Transactions on Communication Technology*, vol. COM-12, pp. 128-135, 1964.
- A. Lender, "Correlative level coding for binary-data transmission," *IEEE Spectrum*, vol. 3, no. 2, pp. 104-115, 1966.
- N.-S. Lin and C.-P.J. Tzeng, "Full-duplex data over local loops," *IEEE Communications Magazine*, vol. 26, pp. 31-42, February 1988.
- S. Lin, D.J. Costello, and M.J. Miller, "Automatic-repeat-request error control schemes," *IEEE Communications Magazine*, vol. 22, no. 12, pp. 5-16, 1984.
- Y. Linde, A. Buzo and R.M. Gray, "An algorithm for vector quantizer design," *IEEE Trans. on Communications*, vol. COM-28, pp. 84-95, 1980.
- D. Linden, "A discussion of sampling theorems," *Proceedings of the IRE*, vol. 47, pp. 1219-1226, 1959.
- C.L. Liu and K. Feher, "Noncoherent detection of $\pi/4$ -shifted systems in a CCI-AWGN combined interference environment," *Proceedings of the IEEE 40th Vehicular Technology Conference*, San Francisco, 1989.
- S.P. Lloyd, "Least squares quantization in PCM," unpublished Bell Laboratories Technical Note, 1957. This report was reprinted in *IEEE Transactions on Information Theory*, vol. IT-28, pp. 129-137, 1982.
- J. Lodge, R. Young, P. Hoeher, and J. Hagenauer, "Separable MAP 'filters' for the decoding of product and concatenated codes," *Proceedings of the IEEE International Conference on Communications*, pp. 1740-1745, Geneva, Switzerland, May 1993.
- R. W. Lucky, "Automatic equalization for digital communication," *Bell System Tech. J.*, vol. 44, pp. 547-588, 1965.
- R. W. Lucky, "Techniques for adaptive equalization of digital communication systems," *Bell System Tech. J.*, vol. 45, pp. 255-286, 1966.
- R. Lugannani, "Intersymbol interference and probability of error in digital systems," *IEEE Transactions on Information Theory*, vol. IT-15, pp. 682-688, 1969.
- V.H. MacDonald, "Advanced mobile phone service: the cellular concept," *Bell System Tech. J.*, vol. 58, pp. 15-41, 1979.
- D.J.C. MacKay, "Good error-correcting codes based on very sparse matrices," *IEEE Transactions on Information Theory*, vol. 45, pp. 399-431, 1999.
- D.J.C. MacKay and R.M. Neal, "Near Shannon limit performance of low density parity check codes," *Electronics Letters*, vol. 33, No. 6, pp. 457-458, 1997; and vol. 32, no. 18, pp. 1645-1646, 1996.
- D.J.C. MacKay, S.T. Wilson, and M.C. Davey, "Comparison of constructions of irregular Gallager codes," *IEEE Transactions on Communications*, vol. 47, pp. 1449-1454, 1999.
- J. Max, "Quantizing for minimum distortion," *IRE Transactions on Information Theory*, vol. IT-6, pp. 7-12, 1960.
- K. Maxwell, "Asymmetric digital subscriber line: Interim technology for the next forty years," *IEEE Communications Magazine*, vol. 34, pp. 100-106, October 1996.
- R.J. McEliece, D.J.C. MacKay, and J.-F. Cheng, "Turbo coding as an instance of Pearl's belief propagation algorithm," *IEEE Journal on Selected Areas of Communication*, vol. 16, pp. 140-152, 1998.
- D. Mennie, "AM stereo: Five competing options," *IEEE Spectrum*, vol. 15, no. 6, pp. 24-31, 1978.
- M.L. Moher, "Cross-entropy and iterative detection," Ph.D. thesis, Department of Systems and Computer Engineering, Carleton University, Ottawa, Canada, May 1997.
- M.L. Moher and T.A. Gulliver, "Cross-entropy and iterative decoding," *IEEE Transactions on Information Theory*, vol. 44, pp. 3097-3104, 1998.
- P. Monsen, "Feedback equalization for fading dispersive channels," *IEEE Transactions on Information Theory*, vol. IT-17, pp. 56-64, 1971.
- K.H. Mueller and J.J. Werner, "A hardware efficient passband equalizer structure for data transmissions," *IEEE Transactions on Communications*, vol. COM-30, pp. 538-541, 1982.
- K. Murota and K. Hirade, "GMSK modulation for digital mobile radio telephone," *IEEE Transactions on Communications*, vol. COM-29, pp. 1044-1050, 1981.
- E. Murphy, "Whatever happened to AM stereo?" *IEEE Spectrum*, vol. 25, p. 17, 1988.

- P. Noll, "MPEG digital audio coding standards." In V.K. Madiseti and D.B. Williams (editors), *The Digital Signal Processing Handbook*, Piscataway, N.J.: IEEE Press, pp. 40-1-40-28, 1998.
- D.O. North, "An analysis of the factors which determine signal/noise discrimination in pulsed carrier systems," *Proceedings of the IEEE*, vol. 51, pp. 1016-1027, 1963; this paper is a reprint of a classified RCA Report published in 1943.
- H. Nyquist, "Certain factors affecting telegraph speed," *Bell System Tech. J.*, vol. 3, pp. 324-346, 1924.
- H. Nyquist, "Thermal agitation of electric charge in conductors," *Physical Review*, second series, vol. 32, pp. 110-113, 1928.
- H. Nyquist, "Certain topics in telegraph transmission theory," *Transactions of the AIEE*, vol. 47, pp. 617-644, Feb. 1928.
- M.W. Oliphant, "The mobile phone meets the Internet," *IEEE Spectrum*, vol. 36, pp. 20-28, August, 1999.
- B.M. Oliver, J.R. Pierce, and C.E. Shannon, "The philosophy of PCM," *Proceedings of the IRE*, vol. 36, pp. 1324-1331, 1948.
- D.Y. Pan, "Digital audio compression," *Digital Technical Journal*, vol. 5, pp. 1-14, 1993.
- S. Pasupathy, "Nyquist's third criterion," *Proceedings of the IEEE*, vol. 62, pp. 860-861, 1974.
- S. Pasupathy, "Correlative coding—A bandwidth-efficient signaling scheme," *IEEE Communications Magazine*, vol. 15, no. 4, pp. 4-11, 1977.
- S. Pasupathy, "Minimum shift keying—A spectrally efficient modulation," *IEEE Communications Magazine*, vol. 17, no. 4, pp. 14-22, 1979.
- A.J. Paulraj and B.C. Ng, "Space-time models for wireless personal communications," *IEEE Personal Communications*, pp. 36-48, February, 1998.
- A.J. Paulraj and C.B. Papadakis, "Space-time processing for wireless communications," *IEEE Signal Processing Magazine*, pp. 49-83, November, 1997.
- R.L. Pickholtz, D.L. Schilling, and L.B. Milstein, "Theory of spread-spectrum communications—A tutorial," *IEEE Transactions on Communications*, vol. COM-30, pp. 855-884, 1982.
- R. Price, "Nonlinearly feedback-equalized PAM vs. capacity for noisy filter channels," International Conference on Communications, ICC '72, pp. 22.12-22.17, June 1972, Philadelphia.
- R. Price and P.E. Green, Jr., "A communication technique for multipath channels," *Proceedings of the IRE*, vol. 46, pp. 555-570, 1958.
- J.G. Proakis, "Advances in equalization for intersymbol interference," *Advances in Communications Systems*, edited by A.J. Viterbi, vol. 4, pp. 123-198, Academic Press, 1975.
- S. Qureshi, "Adaptive equalization," *IEEE Communications Magazine*, vol. 20, no. 2, pp. 9-16, March 1982.
- S. Qureshi, "Adaptive equalization," *Proceedings of the IEEE*, vol. 73, pp. 1349-1387, 1985.
- T.A. Ramstad, "Still image compression." In V.K. Madiseti and D.B. Williams (editors), *The Digital Signal Processing Handbook*, Piscataway, N.J.: IEEE Press, pp. 52-1-52-27, 1998.
- I.S. Reed and G. Solomon, "Polynomial codes over certain finite fields," *Journal of SIAM*, vol. 8, pp. 300-304, 1960.
- A.H. Reeves, "The past, present and future of PCM," *IEEE Spectrum*, vol. 12, no. 5, pp. 58-63, 1975.
- S.A. Rhodes, "Effect of noisy phase reference on coherent detection of offset-QPSK signals," *IEEE Transactions on Communications*, vol. COM-22, pp. 1046-1055, 1974.
- S.O. Rice, "Mathematical analysis of random noise," *Bell System Tech. J.*, vol. 23, pp. 282-332, 1944; vol. 24, pp. 46-156, 1945.
- S.O. Rice, "Statistical properties of a sine-wave plus random noise," *Bell System Tech. J.*, vol. 27, pp. 109-157, 1948.
- S.O. Rice, "Envelopes of narrow-band signals," *Proceedings of the IEEE*, vol. 70, pp. 692-699, 1982.
- T. Richardson, A. Shokrollahi, and R. Urbanke, "Design of provably good low-density parity check codes," submitted in 1999 to *IEEE Transactions on Information Theory*.
- R.L. Rivest, A. Shamir, and L. Adleman, "A method for obtaining digital signatures and public key cryptosystems," *Communications of the ACM*, vol. 21, pp. 120-126, 1978.
- W.L. Root, "Remarks, mostly historical, on signal detection and signal parameter estimation," *Proceedings of the IEEE*, vol. 75, pp. 1446-1457, 1987.

- A. Ruiz, J.M. Cioffi, and S. Kasturia, "Discrete multiple tone modulation with coset coding for the spectrally shaped channel," *IEEE Transactions on Communications*, vol. 40, pp. 1012-1029, 1992.
- W.D. Rummier, "A new selective fading model—Application to propagation data," *Bell System Tech. J.*, vol. 58, pp. 1037-1071, 1979.
- B.R. Saltzberg, "Comparison of single-carrier and multitone digital modulation for ADSL applications," *IEEE Communications Magazine*, vol. 36, pp. 114-121, November, 1998.
- B.R. Saltzberg, "Performance of an efficient parallel data transmission system," *IEEE Transactions on Communication Technology*, vol. COM-15, pp. 805-811, 1967.
- S.D. Sandberg and M.A. Tzannes, "Overlapped discrete multitone modulation for high speed copper wire communications," *IEEE Journal on Selected Areas in Communications*, vol. 13, pp. 1571-1585, 1995.
- D.V. Sarwate and M.B. Pursley, "Crosscorrelation properties of pseudorandom and related sequences," *Proceedings of the IEEE*, vol. 68, pp. 593-619, 1980.
- B. Sayar and S. Pasupathy, "Nyquist 3 pulse shaping in continuous phase modulation," *IEEE Transactions on Communications*, vol. COM-35, pp. 57-67, 1987.
- H.R. Schindler, "Delta modulation," *IEEE Spectrum*, vol. 7, no. 10, pp. 69-78, 1970.
- R.A. Scholz, "The origins of spread-spectrum communications," *IEEE Transactions on Communications*, vol. COM-30, pp. 822-854, May 1982.
- R.A. Scholz, "Notes on spread-spectrum history," *IEEE Transactions on Communications*, vol. COM-31, pp. 82-84, 1983.
- J.S. Schouten, F. DeJager, and J.A. Greefkes, "Delta modulation, a new modulation system for telecommunication," *Phillips Technical Review*, vol. 13, pp. 237-245, 1952.
- C.E. Shannon, "A mathematical theory of communication," *Bell System Tech. J.*, vol. 27, pp. 379-423, 623-656, 1948.
- C.E. Shannon, "Communication theory of secrecy systems," *Bell System Tech. J.*, vol. 28, pp. 656-715, 1949.
- C.E. Shannon, "Communication in the presence of noise," *Proceedings of the IRE*, vol. 37, pp. 10-21, 1949.
- M.K. Simon and D. Divsalar, "On the implementation and performance of single and double differential detection schemes," *IEEE Trans. on Communications*, vol. 40, pp. 278-291, 1992.
- B. Sklar, "A primer on turbo code concepts," *IEEE Communications Magazine*, vol. 35, pp. 94-102, December 1997.
- B. Sklar, "A structural overview of digital communications—A tutorial review," Part I, *IEEE Communications Magazine*, vol. 21, no. 5, pp. 4-17, 1983; Part II, vol. 21, no. 7, pp. 6-21, 1983.
- D. Slepian, "On bandwidth," *Proceedings of the IEEE*, vol. 64, pp. 292-300, 1976.
- B. Smith, "Instantaneous companding of quantized signals," *Bell System Tech. J.*, vol. 36, pp. 653-709, 1957.
- E.S. Sousa and S. Pasupathy, "Pulse shape design for teletext data transmission," *IEEE Trans. on Communications*, vol. COM-31, pp. 871-878, 1983.
- S. Stein, "Unified analysis of certain coherent and noncoherent binary communication systems," *IEEE Transactions on Information Theory*, vol. IT-10, pp. 43-51, 1964.
- C.E. Sundberg, "Continuous phase modulation," *IEEE Communications Magazine*, vol. 24, no. 4, pp. 25-38, 1986.
- M. Tomlinson, "New automatic equaliser employing modulo arithmetic," *Electronics Letters*, vol. 7, pp. 138-139, March 1971.
- D.W. Tufts, "Nyquist's problem—The joint optimization of transmitter and receiver in pulse amplitude modulation," *Proceedings of the IEEE*, vol. 53, pp. 248-259, 1965.
- G.L. Turin, "An introduction to matched filters," *IRE Transactions on Information Theory*, vol. IT-6, pp. 311-329, 1960.
- G.L. Turin, "An introduction to digital matched filters," *Proceedings of the IEEE*, vol. 64, pp. 1092-1112, 1976.
- G. Ungerboeck, "Channel coding with multilevel/phase signals," *IEEE Transactions on Information Theory*, IT-28, pp. 55-67, 1982.

- G. Ungerboeck, "Trellis-coded modulation with redundant signal sets," Parts 1 and 2, *IEEE Communications Magazine*, vol. 25, no. 2, pp. 5-21, 1987.
- M.C. Valenti, "An introduction to turbo codes," EE Department, Virginia Polytechnic Institute & State University, Blacksburg, Virginia, unpublished, 1998.
- B. van der Pol, "The fundamental principles of frequency modulation," *Journal of IEE (London)*, vol. 93, part III, pp. 253-258, 1946.
- J.H. Van Vleck and D. Middleton, "A theoretical comparison of visual, aural, and meter reception of pulsed signals in the presence of noise," *Journal of Applied Physics*, vol. 17, pp. 940-971, 1946.
- A.J. Viterbi, "Error bounds for convolutional codes and an asymptotically optimum decoding algorithm," *IEEE Trans. on Information Theory*, vol. IT-13, pp. 260-269, 1967.
- A.J. Viterbi, "Spread-spectrum communications—Myths and realities," *IEEE Communications Magazine*, vol. 17, no. 3, pp. 11-18, May 1979.
- A.J. Viterbi, "When not to spread spectrum—A sequel," *IEEE Communications Magazine*, vol. 23, no. 4, pp. 12-17, 1985.
- A.J. Viterbi, "Wireless digital communication: A view based on three lessons learned," *IEEE Communications Magazine*, vol. 29, no. 9, pp. 33-36, 1991.
- G.K. Wallace, "The JPEG still picture compression standard," *Communications of the ACM*, vol. 34, pp. 31-44, 1991.
- D.K. Weaver, Jr., "A third method of generation and detection of single-sideband signals," *Proceedings of the IRE*, vol. 44, pp. 1703-1705, 1956.
- J. Weiss and D. Schremp, "Putting data on a diet," *IEEE Spectrum*, vol. 30, pp. 36-39, August 1993.
- T.A. Welch, "A technique for high performance data compression," *Computer*, vol. 17, no. 6, pp. 8-19, 1984.
- L.-F. Wei, "Rotationally invariant convolutional channel coding with expanded signal space—part I: 180°," *IEEE Journal on Selected Areas in Communications*, vol. SAC-2, pp. 659-671, 1984.
- L.-F. Wei, "Rotationally invariant convolutional channel coding with expanded signal space—part II: nonlinear codes," *IEEE Journal on Selected Areas in Communications*, vol. SAC-2, pp. 672-686, 1984.
- L.-F. Wei, "Trellis-coded modulation with multidimensional constellations," *IEEE Transactions on Information Theory*, vol. IT-33, pp. 483-501, 1987.
- S.B. Weinstein, "Echo cancellation in the telephone network," *IEEE Communications Magazine*, vol. 15, no. 1, pp. 8-15, 1977.
- S.B. Weinstein and P.M. Ebert, "Data transmission by frequency-division multiplexing using the discrete Fourier transform," *IEEE Transactions on Communications*, vol. COM-19, pp. 628-634, 1971.
- J.J. Werner, "Tutorial on carrierless AM/PM—Part I: Fundamentals and digital CAP transmitter," *AT&T Bell Laboratories Report*, Minneapolis, June 23, 1992.
- J.J. Werner, "Tutorial on carrierless AM/PM—Part II: Performance of bandwidth-efficient line codes," *AT&T Bell Laboratories Report*, Middletown, February 6, 1993.
- B. Widrow and M.E. Hoff, Jr., "Adaptive switching circuits," *WESCON Convention Record*, Pt. 4, pp. 96-104, 1960.
- J.H. Winters, "Smart antennas for wireless systems," *IEEE Personal Communications*, pp. 23-27, February, 1998.
- J.H. Winters, "Adaptive antennas for wireless communications," International Conference on Communications '99, Tutorial Notes TU5, Vancouver, June 6, 1999.
- A.D. Wyner, "Fundamental limits in information theory," *Proceedings of the IEEE*, vol. 69, pp. 239-251, 1981.
- J.L. Yen, "On the non-uniform sampling of bandwidth-limited signals," *IRE Transactions on Circuit Theory*, vol. CT-3, pp. 251-257, 1956.
- O.C. Yue, R. Luganani, and S.O. Rice, "Series approximations for the amplitude distribution and density of shot processes," *IEEE Transactions on Communications*, vol. COM-26, pp. 45-54, 1978.

- N. Zervos and I. Kalet, "Optimized decision feedback equalization versus optimized orthogonal frequency division multiplexing for high-speed data transmission over the local cable network," International Conference on Communications, ICC '89, pp. 35.2.1-35.2.6, June 1989.
- J. Ziv and A. Lempel, "A universal algorithm for sequential data compression," *IEEE Transactions on Information Theory*, vol. IT-23, pp. 337-343, 1977.
- J. Ziv and A. Lempel, "Compression of individual sequences via variable-rate coding," *IEEE Transactions on Information Theory*, vol. IT-24, pp. 530-536, 1978.
- W.Y. Zou and Y. Wu, "COFDM: An overview," *IEEE Transactions on Broadcasting*, vol. 41, pp. 1-5, 1995.

NOTES

1. The following abbreviations are used for some of the journal papers:

ACM: Association for Computing Machinery
 AIEE: American Institute of Electrical Engineers
 IEEE: Institute of Electrical and Electronics Engineers
 IEE: Institution of Electrical Engineers (London)
 IRE: Institute of Radio Engineers
 SIAM: Society for Industrial and Applied Mathematics

INDEX

A

- absolute entropy, 594
- accumulative error, 221–222
- acquisition, 493
- adaptive, 230
- adaptive antenna array, 557
 - optimizing performance of, 558
 - as specialized technique, 559
 - structure of, 557
 - use of, 559
- adaptive delta modulation (ADM), 232–235
- adaptive delta modulation algorithm, 232
- adaptive delta modulation system, 234–235
- adaptive differential pulse-code modulation (ADPCM), 229–232
 - description of, 230–231
 - use of, 231
 - for voice signals, 232
- adaptive equalization
 - algorithm for, 287–288
 - as the method of choice, 379
- adaptive equalizer
 - modes of operation for, 290
 - preferred approach of, 297
 - tracking capability of, 291
- adaptive filtering, 297
- adaptive prediction
 - with backward estimation, 231
 - disadvantages of, 231
 - with forward estimation, 231
 - preferred method of, 231
 - schemes for, 230
- adaptive predictor, 225
- adaptive quantization, 230
 - with backward estimation, 230
 - with forward estimation, 230
 - problems of, 230
- adaptive quantizer, 230
- adaptive receiver, 287
- adaptive spatial processing, 557
- adaptive synchronous equalizer, 287
- adders, 646
- additive code-modulated interference, 493
- additive white Gaussian noise (AWGN), 378
 - as channel impairment, 379
 - and receiver design, 337
 - signal detection in, 349
- additive white Gaussian noise (AWGN) channel, 559
 - capacity, 431–432
 - characterization of, 322
 - and error probability, 332
 - received signal for, 403
 - and signal detection, 329
 - and signal transmission, 309
 - techniques in, 464
- additive white Gaussian noise (AWGN) model, 516
- adjustment signal, 453
- ADM. *See* adaptive delta modulation
- ADSL. *See* asymmetric digital subscriber lines
- Advanced Research Project Agency Network (ARPANET)
 - and impact on computer communications, 28
 - and pioneering work, 28
- A-law
 - capabilities of, 203
 - defined as, 202, 203
- algebraic code
 - properties of, 641–642
 - types of, 693
- algebraic decoder, 629–630
- aliased spectrum, 187
- aliasing, 187
- allowed frequency band, 154–155
- alternate mark inversion (AMI) signaling, 207
- AM. *See* amplitude modulation
- American Standard Code for Information Interchange (ASCII), 6, 762
- amplification, 128
- amplitude distortion, 191
- amplitude limiter, 129, 142, 143
- amplitude-modulated signal, 89–90
- amplitude modulation (AM), 20, 89–90, 422, 729
 - definition of, 90
 - limitations of, 92–93
 - merits of, 162–163
 - types of, 162–163
 - virtues of, 92–93
- amplitude modulation system noise analysis of, 135
 - process of, 90–91
- amplitude quantization, 194
- amplitude sensitivity, 90
- amplitude-shift keying (ASK)
 - basic signaling scheme, 344–345
 - signals, 345
- amplitude-shift keying (ASK) modulation, 345
- AM receiver
 - comparison of, 136–137
 - envelope detection, 135–137
 - model of, 135
 - performance of, 136–137
- AM signal
 - demodulation of, 162–163
 - and FM signal comparison, 164
- AM-to-PM conversion, 127–128
- analog communication system
 - design of, 22
 - reasons for study of, 23
 - use of, 21

- analog information-bearing signal, 184
 - analog modem
 - design philosophy of, 429–430
 - limited operation of, 429
 - noise performance of, 429
 - analog pulse modulation, 20
 - feature of, 236
 - optimum form of, 193
 - transmission of, 183
 - variations of, 236
 - analog television, 5
 - analog-to-digital (A/D) converter, 201, 379–380, 445
 - analysis-by-synthesis codec, 552
 - analysis equation, 442–443
 - analytic signal. *See* pre-envelope
 - angle, 107, 312
 - angle-modulated signal, 387
 - interpretation of, 108
 - waveform of, 89–90
 - angle modulation, 20, 89–90
 - classification of, 163
 - forms of, 108–109
 - important feature of, 107
 - provisions of, 107
 - angular velocity, 108
 - antenna
 - beamwidth, 521
 - designing of, 17–18
 - multibeam use, 514
 - receiving end of, 18
 - antenna, multibeam, 514
 - antenna arrays, 553
 - anti-aliasing filter, 187
 - antijam characteristics, 498–499
 - antipodal signal, 349
 - aperture effect, 191
 - a priori* probabilities, 323–324, 583
 - arc tangent computer, 364
 - argument function, 54
 - Armstrong, Edwin H., 27
 - ARQ. *See* automatic repeat request
 - array output signal, 557
 - array signal processor, 553, 556
 - ASCII. *See* American Standard Code for Information Interchange
 - asymmetric digital subscriber lines (ADSL)
 - advantages of, 446
 - motivation for, 282
 - services supported by, 281
 - use of, 446
 - asymmetric modem
 - configuration of, 425–426
 - design of, 426
 - asymmetry ratio, 282
 - asymptotic coding gain, 668, 673
 - asynchronous transfer mode (ATM), 14–15
 - asynchronous transmission, 7
 - ATM. *See* asynchronous transfer mode
 - auditory masking, 9
 - auditory masking phenomenon, 234
 - auditory system, 235
 - autocorrelation function, 36, 43, 482
 - definition of, 35
 - evaluation of, 51
 - graphical summary of, 75–76
 - properties of, 36–37
 - significance of, 37
 - autocovariance function, 36
 - automatic-repeat request (ARQ), 628–629
 - for error detection, 628
 - philosophy of, 628
 - types of, 628–629
 - AWGN. *See* additive white Gaussian noise
- B**
- band-limited channel, 3
 - band-limited signal
 - defined as, 427
 - sampling theorem for, 186–187
 - band-limited white Gaussian noise, 608
 - bandpass communication channel, 348–349
 - band-pass filter, 98–99, 515
 - band-pass signal
 - components of, 728, 730
 - Hilbert transform and, 731
 - representation of, 113, 726–729
 - band-pass system
 - analysis of, 730–734
 - impulse response, 731
 - bandwidth, 720–723
 - definitions of, 720–721
 - efficiency of, 347, 348
 - bandwidth-duration product, 721–722
 - bandwidth efficiency, 347, 348
 - defined as, 347
 - diagram of, 601–602
 - product of, 348
 - bandwidth-limited channel, 16
 - bandwidth-noise trade-off, 193
 - Bardeen, John, 28
 - barrage noise jammer, 508
 - baseband, 88
 - baseband binary data transmission system, 259
 - baseband binary PAM system, 259
 - baseband channel
 - channel requirements, 247
 - and digital data transmission, 247
 - baseband M-ary PAM transmission, 275–277
 - baseband power spectral density of a binary PSK signal, 353
 - to evaluate, 347
 - baseband pulse, 374
 - baseband-pulse transmission system
 - and fixed characteristics, 297
 - performance of, 296
 - and signal-to-noise ratio, 297
 - source of bit errors, 259
 - baseband signal, 88, 95
 - baseband signal-to-noise ratio, 154
 - baseband space-time processor, 556
 - baseband spread-spectrum system, 488
 - base station, 530
 - basis function, 451
 - Baudot, Emile, 26
 - bauds, 276
 - Bayes' rule, 585–587, 707
 - BCH codes, 653–654
 - BCJR algorithm
 - formulation of, 678
 - mathematical exposition of, 680
 - purpose of, 678
 - versus the Viterbi algorithm, 678
 - Bell, Alexander Graham, 27
 - Berners-Lee, Tim, 28–29
 - Bessel's equation, 735
 - Bessel equation, modified, 738
 - Bessel function, 735–739
 - behavior of, 114–115
 - versus the modulation index, 114
 - properties of, 735–737
 - Bessel function, modified, 737–739
 - "best effort service", 14
 - binary additive stream ciphers, 744
 - binary BCH code
 - common types of, 653

- binary BCH code (*Continued*)
 - versus nonbinary, 654
- binary code
 - efficiency of, 197
 - symbols of, 204
- binary CRC codes
 - capabilities of, 652
 - and error detection, 652–653
- binary data sequence, 359
- binary data transmission system, 285
- binary detection problem, 403
- binary differential phase-shift keying (DPSK), 415
- binary digit, 204, 569
- binary digital communication system, 403–405
- binary frequency modulation, 397
- binary frequency-shift keying (FSK)
 - bit error rate for, 384
 - error probability of, 382–384
- binary FSK signal
 - baseband power spectral density of, 386
 - with continuous phase, 385–386
 - detection of, 387
 - to generate, 384
 - power spectra of, 353, 385–386
- binary FSK system, 381–386
- binary FSK transmitter, 384–385
- binary hypothesis test, 405
- binary-input additive white Gaussian noise (AWGN) channel, 667, 668
- binary phase-shift keying (PSK), 349–353
 - error probability of, 350–352
 - as a linear operation, 492
 - modulator, 490
 - signals, 353
 - and spread-spectrum modulation, 550
 - transmitter, 352
 - use of, 550
- binary pulse, coded, 193
- binary pulse code modulation (PCM) wave, 193
- binary signaling
 - bit error rate, 543–544
 - scheme, 407
- binary symmetric channel (BSC), 258, 629, 667
- bipartite graphs, 685
- bipolar code, 281
- bipolar return-to-zero (BRZ) signaling, 207
- B-ISDN. *See* broadband integrated services digital network
- bit, 204, 569
- bit-by-bit interleaving procedure, 214
- bit duration, 253
- bit energy-to-noise density ratio, 384
- bit error rate (BER)
 - assumptions of, 209
 - for coherent binary FSK, 384
 - of digital modulation schemes, 417
 - probability of, 23
 - in signal regeneration, 208
 - and symbol error probability, 335–336
- bit-rate reduction, 218
- bit stuffing, 215
- blanking pulse, 5
- block cipher, 744, 745, 746
- block code, 590, 632
 - distinguishing feature of, 627
 - rate of, 685
- blocks, 572
- Bose-Chaudhuri-Hocquenghem (BCH) codes, 653–654
- bounds
 - for prediction, 332
 - use of, 332
- Brattain, Walter H., 28
- British Broadcasting Corporation (BBC), 27
- broadband integrated services digital network (B-ISDN), 14–15
 - cells in, 15
 - quality of service, 14
 - and the telephone network, 14
- broadband networks, 14–15
- broadcasting mode, 2–3
- broadcasting system, 128
- burstiness, 7
- byte, 7
- C**
- cable-television systems, 17
- Campbell's theorem, 60
- CAP. *See* carrierless amplitude/phase
- capacity, 568
- capacity boundry, 602
- capture effect, 148–149
- carrier and timing synchronization systems, 449
- carrier component, 135
- carrier frequency, 92
- carrier-frequency tuning, 128
- carrierless amplitude/phase (CAP), 369, 380
- carrierless amplitude/phase (CAP) modulation, 373, 431
 - bandwidth for, 375
 - idea behind, 369
- carrierless amplitude/phase (CAP) receiver
 - digital implementation of, 379–380
 - improved performance of, 379
 - in an unknown environment, 379
- carrierless amplitude/phase (CAP) system
 - application of, 380
 - basic structure of, 378–379
 - modulation, 369, 373, 375
 - receiver, 379–380
 - structure of, 378–379
 - transmitter, 378
- carrierless amplitude/phase (CAP) transmitter, 378
- carrier phase, 403
- carrier phase recovery, 448, 458, 459–463
- carrier synchronization, 448
- carrier-to-noise ratio, 137–139, 144
 - defined as, 138, 150
 - large versus small, 141–142
 - level of operation, 138
 - low versus high, 149
 - versus signal-to-noise ratio, 151
- carrier wave, 19–20
- Carson's rule
 - for approximate evaluation, 163
 - and the universal curve, 119
- Cartesian product, 369–370
- cascade connection
 - noise figure of, 526
 - of two-port networks, 524–525
- catastrophic code, 667
- CDMA. *See* code division multiple access

- CDM systems, 505
- cell delay
 - defined as, 14
 - variation of, 14
- cell loss ratio, 14
- cells, 14–15
- cell splitting, 531
- cell-switching technology, 14
- cellular concept, 530
- cellular radio
 - idealized model of, 530
 - propagation problems of, 532
 - wireless communications in the context of, 530
- CELP
 - codec implementation, 553
 - distinguishing feature of, 552
 - encoder for, 552
 - modeling of, 616
 - See also* code-excited LPC
- central limit theorem, 60, 498
 - definition of, 56
 - and the Gaussian process, 55
- central moments, 712
- channel, 2
 - characteristics of, 309–310
 - frequency response of, 261
 - and random noise, 259
- channel bandwidth
 - definition of, 3
 - occupancy of, 347
 - primary communication resource, 92
 - used in North America, 102
- channel capacity, 587–589
 - concept of, 630–631
 - of a discrete memoryless channel, 588
- channel capacity theorem, 599
- channel code word, 21
- channel coding
 - design goal of, 589
 - and mapping, 589
 - techniques for, 628
- channel coding theorem, 589–593, 616, 630–631
 - application of, 591–592
 - operations of, 628
 - Shannon's second theorem, 590–591
 - significance of, 592
 - unsatisfactory feature of, 631
- channel data rate, 627
- channel decoder
 - under designer's control, 590
 - goal of, 627
 - inverse mapping operation, 590
- channel encoder, 626
 - under designer's control, 590
 - goal of, 627
 - introduction of redundancy, 590
 - mapping operation, 590
 - Markovian assumption for, 681
- channel imperfections, 2
- channel impulse response, 291
- channel input, 597, 610
- channel matrix, 582
- channel model, 130
- channel noise, 31–32
 - absence of, 228
 - and bit error, 248
 - condition of acting alone, 282–283
 - effects of, 209, 296
 - in PCM systems, 209, 253
 - reducing the effect of, 210
 - source of, 32
 - uncertainty due to, 403
- channel output, 72
- channel parameters, 587
- channel signal-to-noise ratio, 134
 - for AM, 135
 - definition of, 132
 - formula for, 147
- characteristic function, 713
- Chebyshev inequality, 713
- check node, 685
- chip, 488, 501
- chip duration, 494
- chip rate, 501
- chrominance signal, 6
- cipher, 742
- ciphertext, 742
- circuit, 10–11
- circuit-switched network, 11
 - controlled by, 11
 - establishing a connection, 11
- circuit switching, 10–11
- circulant matrix, 442, 443
- circular constellation, 368
- Clark, Arthur C., 29
- clock recovery, 448
- closed-loop optimization
 - procedure, 551
- coaxial cable
 - application of, 17
 - consists of, 17
 - versus twisted-pairs, 17
- co-channel cells
 - determination of, 531
 - finding of, 531
 - as interference, 553
- code, catastrophic, 667
- code book, 580
- codec, 552
- coded binary pulses, 193
- code division multiple access (CDMA)
 - advantage of, 514
 - codes for, 505
 - systems, 548
- code-division multiplexing (CDM)
 - as an alternative method, 505
 - bandwidth requirements of, 505
 - defined as, 21
- coded pulse
 - in analog modulation, 217
 - in digital pulse modulation, 183
 - use of, 184
- code elements, 203–204, 212
- code-excited LPC, 552
- code rate, 590, 627, 654
- code tree, 657, 657–658
- code vector, 660
- code word, 627
 - average length, 574
 - in binary form, 574
 - duration of, 212
- code-word length, average, 574
- coding efficiency, 574
- coding gain, 425
- coding theory, 28, 676
- coherence bandwidth, 540
- coherent binary frequency-shift keying (FSK), 380
- coherent binary FSK system, 384
 - characterized by, 381
 - generation and detection of, 384
 - receiver, 384–385
- coherent binary phase-shift keying (PSK)
 - bit error rate, 352, 417
 - characteristics of, 350
- coherent binary PSK system, 384
 - bit error of, 357
 - characteristics of, 350
 - receiver, 352
 - signals, 352
- coherent detection, 98, 131, 133

- coherent detection (*Continued*)
 - and demodulation, 95
 - effect of, 97–98
 - use of, 132
- coherent *M*-ary PSK, 367
- coherent *M*-ary quadrature amplitude modulation (QAM), 464
- coherent MSK
 - bit error rate for, 394
 - expressions for, 417
- coherent phase-shift keying (PSK), 349
- coherent QPSK
 - bit error rate for, 417
 - symbol error probability, 358
- coherent QPSK system
 - signals of, 359
 - specifications of, 458
- coherent quadriphase-shift keying (QPSK), 354
- colored noise channel, 607–611
- color receptors
 - in the human eye, 6
 - types of, 6
- communication, 1
 - applications of, 1
 - frequencies for, 4
 - fundamentals of, 2–3
 - types of, 21–23
- communication, error-free, 568
- communication channel, 7, 15–19, 277
 - classification of, 3, 15, 19
 - description of, 15–19
 - simultaneous use of, 512
 - use of, 88
- communication link
 - analysis of, 517
 - in circuit switching, 11
- communication networks, 10–15
- communication process, 1–3
- communication resources, 3
- communications satellites
 - in geostationary orbit, 19
 - historical notes, 26–29
 - role of, 19
 - “second generation”, 515
- communication system
 - common feature of, 3
 - design of, 21
 - elements of, 2
 - noise analysis of, 3, 64, 523
 - primary resources of, 3
 - purpose of, 19, 88
 - source of limitations, 248–252
 - transition from analog to digital, 183
- communication system designer, 23
- community-antenna television (CATV) system, 17
- commutator, 211
- compander, 203
- companding circuitry, 203
- companding law, fifteen-segment, 426
- complementary error function, 255, 256, 334
- complex envelope, 347, 727, 728, 729, 730
- complex exponential Fourier series, 717
- complex Fourier coefficient, 717
- complex least-mean-square (LMS) algorithm
 - advantages of, 558
 - limitations of, 558
 - See also* least-mean-square algorithm
- composite signal, 105, 149
- compound codes, 683–684
- compression algorithms, standard, 8
- compression laws, 202–203
- compressor, 203
- computer communications, 2
- computer-generated data, 7
- conditional likelihood function, 403–404
- conditional mean requirement, 200
- conditional probability, 706–707
- conditional probability density function, 320–321, 383, 451–452, 710
- conditional probability of error, 255, 256
- conditional probability of symbol error, 333
- confluent hypergeometric function, 740–741
- confusion, 749
- conservation of time, 211–212
- constant angular velocity, 108
- constant envelope, 111
- constellation encoder, 444
- constrained optimization problem
 - definition of, 437
 - solving of, 609
- continuous AWGN channel, 318–319
- continuous-phase frequency-shift keying (CPFSK), 381, 387
- continuous-phase frequency-shift keying (CPFSK) signal
 - components of, 389
 - deviation ratio of, 388
 - phase of, 388
 - representation of, 387
- continuous random variable, 594, 708
- continuous source, 615
- continuous-time channel
 - partitioning, 432–436
- continuous-wave (CW) modulation, 20
 - effects on reception, 130
 - families of, 88–90
 - principles of, 162
 - techniques for, 130
- continuous-wave (CW) modulation system, 89–90
 - comparison of, 132
 - components of, 88–89
 - noise in, 130
 - performance of, 164
- control symbols
 - in ASCII, 6
 - for communication purposes, 6
 - for printing of characters, 6
- conventional coherent binary FSK
 - bit error rate expressions for, 417
 - with one-bit decoding, 417
- convolutional code, 654–656
 - constraint length of, 655
 - distance properties of, 663
 - distinguishing feature of, 627
 - maximum likelihood decoding of, 660–663
 - performance of, 663
 - use of, 654–656
- convolutional coding, 669
- convolutional encoder
 - code tree for, 657–658
 - input-output relation of, 660
 - state of, 657–658
 - trellis for, 658–659
 - use of, 425
- convolution integral, 42–44, 718
- correlation coefficient, 342, 471
- correlation functions, 35–41
- correlation matrix, 40

- correlation receiver, 326–328, 329
- correlative-level coding, 266–271
 - basis of, 268
 - generalized form of, 274–275
 - idea illustrated, 267
 - premise of, 267
- correlator, 24–25
 - inputs of, 549–550
 - outputs of, 319–322
- coset leaders, 638–639
- Costas loop
 - generalization of, 454
 - for phase recovery, 454
- Costas receiver
 - consists of, 96–97
 - phase control in, 97
 - use of, 96–97
- cost function, 558
- covariance function, 35–36
- CPFSK. *See* continuous-phase frequency-shift keying
- Cramér-Rao bound
 - defined as, 462
 - modification of, 462
- CRC code. *See* cyclic redundancy check (CRC) code
- critical band, 234
- cross constellations, 369–370, 372
- cross-correlation functions, 40, 52
- cross-spectral densities, 52
- crosstalk, 21
 - cause of, 279
 - defined as, 501
 - as impairment, 279
 - types of, 279–280
- cryptanalysis, 742
 - and authorized user, 742–743
 - definition of, 746
 - description of, 742
- cryptogram, 742–743
- cryptographic system
 - classes of, 744
 - classifications of, 759
 - consists of, 743
 - definition of, 743
 - services of, 742
- cryptography, 617, 742
 - and authentication problem, 743
 - classifications of, 759
 - data compression in, 749
 - fundamental assumption, 745
 - importance of, 759
 - and secrecy problem, 743
- cryptology, 742
- crystal-controlled oscillator, 120
- cumulative distribution function, 708
- cyclic code, 641–643
 - advantage of, 641–642
 - characteristics of, 652–654
 - classes of, 652–654
 - encoder for, 645–646
 - generation of, 643
 - properties of, 642
 - in systematic form, 645, 646
- cyclic prefix, 441
- cyclic property, 642
- cyclic redundancy check (CRC)
 - code
 - for error detection, 652
 - generator polynomials of, 653
- D**
- damping factor, 160
- data-aided synchronization, 449
- data bits
 - binary pattern of, 6
 - for error detection, 7
- data communication, 7
- data compaction, 8, 575
 - achieved by, 575
 - assessing, 616
 - schemes for, 575
- data compression, 7, 614–616
 - in cryptography, 749
 - forms of, 7–8
 - idea of, 614
 - as a lossy operation, 614–615
 - reason for using, 615
 - system components, 749
 - techniques for, 218
- data compressor, 614
- data encryption, 617
- data encryption standard (DES), 751–755, 759
- data encryption standard (DES)
 - algorithm, 754
- data-modulated carrier, 500
- data multiplexers, 7
- data network, 11
- data signaling rate, 426
- data transmission system
 - asynchronous versus synchronous, 7
 - capabilities of, 446
 - performance of, 293
- decision device, 259
- decision-directed mode, 290–291
- decision-directed recursive
 - algorithm, 450
- decision errors, 210
- decision feedback, 270
- decision feedback equalization, 291–293, 379, 430
- decision feedback equalizer (DFE)
 - consists of, 292
 - error propagation in, 292–293
 - feedback section of, 292
 - feedforward section of, 292
- decision-making criterion, 323
- decision-making device
 - designs of, 277
 - operation of, 25
- decision rule, 350–351, 357
 - applying, 382
 - defined as, 661
 - as the MAP rule, 323–324
 - as the maximum likelihood rule, 324
 - used by the coherent detector, 497
- decision threshold, 194
- decision tree, 575–576
- decoder, 446
 - condition for optimality, 200–201
 - consists of, 552
 - function of, 552
 - units of, 235
- decoding algorithms, 678
- decoding complexity, 684
- decoding decisions, 663
- decoding error, 660
- decoding process
 - methods of, 693
 - and pulse generation, 208
 - requirements of, 261
- decoding rule, 660
- decoding spheres
 - maximum number of, 600
 - packing of, 600
- decoding window, 663
- decommutator, 211
- decorrelation time, 37
- decryption, 742
- de Forest, Lee, 27
- delay, average, 540
- delay power spectrum, 539
- delay spread, 539
 - as a channel impairment, 553
 - defined as, 540
 - effect of, 556

- delta function, 62
 - property of, 716–717
 - sifting property of, 190, 320
- delta modulation (DM), 218–221
 - advantage of, 219, 237
 - and digital pulse modulation, 237
 - principles of, 218–219
 - quantization error of, 220
 - simplicity of, 223
 - and transmitters, 228
- delta-sigma modulation, 221–223
- demodulation, 20, 88
 - method of, 99
 - stages of, 491
- demodulation scheme, 132
- demodulator output, 92
- demultiplexer
 - in receiver, 359, 445–446
 - in transmitter, 444
- demultiplexing system, 125–126
- DES. *See* data encryption
 - standard detection and error correction, 628
 - of a pulse signal, 248
- detector, 326, 349
- deviation ratio, 387
 - definition of, 119
 - versus modulation index, 119
- DFT. *See* discrete Fourier transform
- diagonal matrix, 443
- dibits, 276
- difference-frequency term, 158
- differential detector
 - components of, 364
 - tangent type, 364–365
- differential encoder
 - of the binary wave, 414
 - consists of, 421–422
 - method used, 207
 - requirement of, 207
- differential entropy, 593–597
- differential phase encoder, 415
- differential phase modulation, 422
- differential phase-shift keying (DPSK), 407, 414–417
 - bit error rate of, 417
 - generation and detection of, 415
 - receiver, 416
 - transmitter, 415–416
- differential pulse-code modulation (DPCM), 227–229
 - basic idea of, 227
 - and digital pulse modulation, 237
 - system comparison, 228
 - and transmitters, 228
- differential quantization scheme, 227, 228
- differentiator, 143
- Diffie-Hellman public key-distribution system, 756
- diffraction, 17–18
- diffusion, 749
- digital audio broadcasting, 448
- digital circuit technology, 189
- digital communication
 - basic form of, 309–310
 - and bit error rate, 24
 - and design goals, 354
 - elements of, 24–25
 - receiver, 337
 - reliability of, 23, 24–26
 - requirements of, 23
 - and system design, 22
 - task of designer, 626
 - use of, 21
- digital data transmission, 247
- digital filter, second-order, 454–455
- digital hierarchy, 214
- digital modem
 - bidirectional, 428
 - capabilities of, 428
 - and data rates, 429
 - design constraints of, 426
 - fundamental design philosophy of, 426
 - one realization of, 426–427
 - signaling scheme for, 427
 - solution to design problems, 428
 - theoretical basis for the design of, 429
- digital modulation schemes
 - comparison of, 417–420
 - probability of error, 417
 - types of, 346
 - using a single carrier, 417–420
 - virtues of, 347
- digital modulation techniques
 - operation of, 448
 - types of, 345–346
- digital multiplexers, 214–215
 - design problems, 215
 - major groups of, 214
- digital multiplexing-demultiplexing operation, 214
- digital passband transmission system, 344
 - assessing performance of, 335
 - performance degradation of, 544
- digital PSTN, 426
- digital pulse modulation
 - basic form of, 193
 - feature of, 236
 - transmission of, 183
- digital satellite communication, 419
- digital signals, 214
- digital signal zero (DS0), 214
- digital signature
 - for electronic mail systems, 758–759
 - properties of, 759
 - use of, 758–759
- digital subscriber line (DSL)
 - as a growing application, 277
 - line codes for, 280–281
 - operational environment of, 277, 447
 - and twisted pairs, 277, 297
 - versus voiceband modems, 446
- digital switch, 215, 446
- digital-to-analog converter (DAC), 445
- digital transmission facility, 215
- digital wireless communication systems, 550–551
- Dirac delta function. *See* delta function
- direct broadcast satellite (DBS)
 - simplicity and affordability of, 517
- direct broadcast satellites (DBS)
 - use of, 517
- direct frequency modulation, 120–121, 396
- directive gain, 520
- directivity, 520
- direct matrix inversion (DMI), 558, 559
- direct-sequence M-ary phase shift keying (DS/MPSK), 508
- direct-sequence spread binary phase-shift-keyed (DS/BPSK) signal, 490, 492
- direct-sequence spread spectrum with coherent BPSK, 490–493
 - principles of, 480
 - systems, 498
- Dirichlet's conditions, 715
- discrete cosine transform (DCT), 8
- discrete cosine transform coefficients, 8
- discrete Fourier transform (DFT), 445

- defined as, 442
 - and digital signal processing, 443
 - discrete memoryless channel, 581–584, 629–631
 - channel capacity of, 588
 - defined as, 581–582
 - discrete memoryless source, 570
 - extension of, 572
 - properties of, 568
 - discrete multitone (DMT), 431, 440–443, 444–446
 - applications of, 446
 - basic idea of, 441
 - and multichannel modulation, 447–448
 - use of, 441
 - discrete pulse-amplitude modulation (PAM), 259
 - discrete pulse modulation, 259
 - discrete random variable, 708
 - discrete source, 615
 - discrete-time, memoryless Gaussian channel, 597
 - discrete-time channel, 291
 - discrete-time convolution, 627
 - discrete-time Fourier transform, 185
 - discriminator, 144
 - discriminator output, 145, 155
 - dispersive channel, doubly, 542
 - distance transfer function, 666
 - distortion, 2
 - acceptable, 614
 - methods of reduction, 103
 - produced by, 102–103
 - unavoidable, 611–612
 - distortion, amplitude, 191
 - distortionless baseband binary transmission, 261–262
 - distortion measure, 199
 - distribution function
 - properties of, 708
 - of a stationary random process, 34
 - diversity techniques, 544–547
 - performance with, 546–547
 - specialized techniques, 559
 - “divide and conquer”, 431
 - DMI. *See* direct matrix inversion
 - DMT. *See* discrete multitone
 - Donald Duck voice effect, 99–100
 - Doppler shift, 535
 - Doppler spectrum, 540–541
 - Doppler spread, 539, 541
 - double-frequency term, 158
 - double sideband-suppressed carrier (DSB-SC) modulated signal (wave), 95, 96
 - double sideband-suppressed carrier (DSB-SC) modulation, 133, 134
 - definition of, 93
 - generated by, 94
 - transmission of sidebands, 163
 - double sideband-suppressed carrier (DSB-SC) receiver
 - compared to an AM receiver, 136–137
 - model of, 132–133
 - doubly dispersive channel, 542
 - down conversion, 105
 - downconverter, 448
 - downlink, 19, 514–515
 - downstream data transmission, 281–282
 - DPSK. *See* differential phase-shift keying
 - DS/BPSK waveform, 490, 492
 - DSL environment, 447
 - DS/MPSK system, 508
 - dual code, 641
 - duobinary code, modified, 281
 - duobinary coding, 270
 - duobinary conversion filter, 268–269
 - duobinary encoder, 267–268
 - duobinary signaling scheme, 267–271
 - frequency response of, 268
 - technique, 271–272
 - duobinary technique, 274–275
 - “dynamic” multipath environment, 532–533
- E**
- echo cancellation, 277–278
 - comparison of schemes, 278–279
 - mode of operation for, 277–278
 - echo canceller
 - in transceiver, 278–279
 - use of, 516
 - effective aperture, 521
 - effective radiated power referenced to an isotropic source (EIRP), 521
 - Einstein-Wiener-Khintchine relations, 46
 - EIRP. *See* effective radiated power referenced to an isotropic source/elastic store, 215
 - electromagnetic interference (EMI), 17
 - electron beam, 4
 - elementary event, 704
 - encoded text, 6
 - encoder
 - condition for optimality of, 199–200
 - functional units of, 235
 - main parts of, 551
 - operation of, 646
 - states of, 659, 667
 - encoding process
 - operations of, 9
 - steps of, 551
 - use of, 203–204
 - encryption, 742
 - enemy cryptanalyst
 - forms of attack by, 745
 - intrusion of, 743, 744
 - energy gap, 99
 - energy signals, 312
 - energy spectral density, 48, 353
 - ENIAC, 28
 - ensemble average
 - autocorrelation function, 284
 - estimation of, 41
 - parameter, 75
 - substituting time averages for, 41
 - entropic coding redundancy, 9
 - entropy
 - conditional, 584
 - definition of, 569
 - formula for, 568
 - properties of, 570–571
 - entropy, conditional, 584
 - envelope
 - defined as, 730
 - and phase components, 67–69
 - types of, 730
 - envelope delay, 16
 - envelope detection, 102–103, 131
 - envelope detector, 123, 143
 - consists of, 92
 - found in, 92
 - loss of message in, 138
 - need for, 406
 - performance of, 137
 - signal comparison, 141–142
 - envelope distortion, 90–91
 - equalizer, 191
 - equiprobable symbols, 571
 - equivalent noise temperature, 61, 524–525
 - ergodic process, 41–42, 51

- error
 - minimization of, 551
 - possible kinds, 254
 - probability of, 497-499
- error burst, 652
- error control, 626
- error-control code
 - theory of, 485
 - types of, 627
- error-control coding
 - classes of, 626
 - for reliable communication, 567
 - techniques, 626
 - techniques for, 683, 693
 - use of, 626, 627
- error-detection bit, 7
- error-free communication, 568
- error function, complementary, 255
- error minimization, 551
- error pattern, 635
- error probabilities, conditional, 352
- error propagation
 - elimination of possibility, 273
 - phenomenon, 270
 - property, 745
- error rate, 253
- error signal
 - calculation of, 457
 - definition of, 288, 453
 - for timing recovery, 457
 - use of, 557
- error-syndrome vector, 635-636
- error threshold, 209-210
- error vector, 635
- estimation procedure, 517
- Euler's formula, 453
- excess bandwidth factor, 441
- excess mean-square error, 291
- excitation generator, 551, 552
- excitation time, 718
- expander, 203
- expansion laws, 203
- exponential law, 193
- extended code
 - average code-word length, 578
 - use of, 577
- extended prefix code, 578
- extended source, 572
- extended-threshold demodulators, 152, 153
- extraction, 261
- extrinsic information, 679
- eye opening, 293
- eye pattern
 - definition of, 293
 - as an experimental tool, 293
 - interpretation of, 293
 - and performance information, 293
- F**
- facsimile (fax) machine
 - basic principle of, 6
 - purpose of, 6
 - in a receiving mode of operation, 420
- fade rate, 541
- fading channel
 - characteristics of, 541
 - effects of, 545
- fading multipath channel, 536-539
- far-end crosstalk (FEXT), 279-280
- Farnsworth, Philo T., 27
- fast Fourier transform (FFT)
 - algorithm, 443-444
- fast-frequency hopping, 502-503, 504
- FDMA. *See* frequency division multiplexing
- FDMA. *See* frequency division multiple access
- FDMA system, 516
- FDM system
 - block diagram of, 105-106
 - modulation steps in, 107
- FEC. *See* feed-forward error correction
- feedback shift register, 480, 481
- feedback system, second-order, 160
- feed-forward error correction (FEC), 628, 629
- Fessenden, Reginald, 27
- FH/MFSK system
 - fast versus slow, 502-503
 - jamming effect on receiver, 502
 - symbol error in, 502
- field-power pattern, 521
- figure of merit, 134
 - for amplitude modulation, 136
 - definition of, 132, 193
 - for frequency modulation, 147
- filtering, 128, 208-209
- filtering scheme, 100
- fine synchronization, 493
- finite-duration impulse response (FIR) filter, 379
- finite-state machine, 654
- fixed channel input, 582
- fixed channel output, 582
- fixed modulation scheme, 627
- fixed point-to-point links, 18-19
- fixed scatterers, 536
- flat-fading channel, 71-72
- flat-flat channel, 542
- flat Rayleigh fading channel, 554
- flat-top samples, 191
- Fleming, John Ambrose, 27
- flip-flops, 646
- flyback. *See* horizontal retrace
- FM demodulator
 - with negative feedback, 153
 - and oscillator types, 152
- FMFB demodulator, 152, 153, 154
- FMFB receiver, 154
- FM receiver
 - breaking point of, 149
 - interference suppression in, 148-149
 - model of, 142-143
 - noise analysis of, 146
 - noise in, 142
 - threshold effects in, 152
- FM signal
 - average power of, 115
 - complex envelope of, 114
 - demodulation of, 121-124
 - desirable properties, 397
 - detection of, 397
 - distinguishing from AM signal, 109
 - effective bandwidth for, 117-119
 - fundamental characteristic of, 110
 - generation of, 120-121
 - side frequencies of, 117
 - spectral analysis of, 110
 - spectrum of, 115
 - in theory and practice, 117
- FM signal, single-tone, 112-113
- FM stereo
 - multiplexing, 124-126
 - specification of standards, 124
 - transmission, 124
- FM system
 - emphasis in, 154-156
 - nonlinear effects in, 126-128
 - See also* frequency modulation (FM) system
- FM threshold effect, 149-152
- FM threshold reduction, 152-154
- FM wave
 - bandwidth requirement, 118

- with reduced modulation index, 152–153
 - forward error-control coding, 628, 668
 - forward error correction (FEC), 626
 - forward estimation, 681
 - forward link, 547
 - Fourier analysis, 715–720
 - Fourier series expansion, 317
 - Fourier series representation, 114
 - Fourier transform
 - definition of, 715
 - inverse, 715
 - of periodic signals, 717–718
 - properties of, 716
 - theory of, 716
 - fractionally spaced equalizer (FSE), 287
 - frame, 552
 - make-up of, 5
 - method of synchronization, 215–216
 - structure, 547
 - frame packing, 9
 - frame-packing unit, 236
 - free distance, 663
 - free propagation
 - channels based on, 15
 - types of, 15
 - free-space loss, 522
 - free-space propagation model, 518–523
 - frequency demodulation
 - defined as, 121
 - methods of, 121
 - frequency deviation, 110, 152
 - frequency-discrimination method
 - stages of, 98–99
 - use of, 100
 - frequency discriminator, 121, 124
 - consists of, 121–122
 - input, 149
 - requirements of, 99
 - frequency diversity, 544–545
 - frequency division duplexing (FDD), 547
 - frequency division multiple access (FDMA), 513, 516
 - frequency-division multiplexing (FDM)
 - defined as, 20–21, 105
 - method of modulation in, 106
 - frequency-domain description, 444, 720
 - frequency down converter, 105, 516
 - frequency flat, 541
 - frequency-flat channel, 542
 - frequency-hop *M*-ary frequency shift-keying (FH/MFSK), 508
 - frequency hopping, 500
 - frequency-hop spread spectrum, 499, 500
 - communication systems, 500
 - principles of, 480
 - frequency-modulated wave, 413
 - frequency modulation (FM), 20, 27, 729
 - capability of, 165
 - cases of, 111
 - characteristic of, 149
 - definition of, 108–109
 - direct, 120–121
 - and mixing, 500
 - as a nonlinear process, 109
 - theory of, 126
 - frequency modulation (FM) system
 - noise analysis of, 142–147
 - similarities to PPM system, 193
 - frequency multiplication ratio, 121
 - frequency multiplier
 - consists of, 120
 - diagram of, 120–121
 - frequency parameters, 128
 - frequency response
 - choice of, 155
 - to denote, 44–45
 - frequency reuse, 530
 - frequency-shift keying (FSK)
 - basic signaling scheme, 344–345
 - and design of modems, 421
 - and frequency modulation, 345
 - represented by, 464
 - frequency-shift keying (FSK) schemes, 418
 - frequency-shift keying (FSK) signal, 386
 - frequency translation, 103, 103–105
 - frequency up converter, 105
 - Friis formula, 526
 - Friis free-space equation
 - defined as, 522
 - used for, 522
 - FSK. *See* frequency-shift keying
 - full amplitude modulation, 162–163
 - full-cosine rolloff characteristic, 266
 - full-duplex link, 628
 - functional
 - defined as, 54
 - versus function, 54
 - fundamental frequency, 717
 - fundamental inequality, 571
- G**
- gain, 720
 - gap, 432
 - Gaussian assumption, 498
 - Gaussian channel, 597
 - Gaussian-distributed random variable, 54
 - Gaussian distribution, 54, 72–73
 - Gaussian filter, 397
 - Gaussian-filtered minimum shift keying (GMSK)
 - as a special kind of binary frequency modulation, 398
 - undesirable feature of, 398
 - Gaussian-filtered minimum shift keying (GMSK) modulator
 - frequency shaping pulse of, 397
 - and intersymbol interference, 398
 - Gaussian-filtered minimum shift keying (GMSK) signal, 397
 - power spectrum of, 400
 - spectral compactness of, 398–400
 - Gaussian-filtered MSK, 396–400
 - Gaussian function, 397
 - Gaussianity, 75
 - Gaussian model, 55
 - Gaussian process, 54–58
 - definition of, 57
 - mathematical justification, 55–56
 - in the study of communications, 55
 - useful properties of, 56–58
 - virtues of, 55
 - Gaussian random variable, 54, 58
 - generator equation, 634
 - generator polynomial, 645
 - of a cyclic code, 643
 - definition of, 656
 - geometric mean, 436
 - geometric representation of signals, 311

- geometric signal-to-noise ratio, 436
 - geostationary satellite
 - communications system, 514–515
 - global coverage, 512
 - Global System for Mobile Communications (GSM), 548
 - frame efficiency of, 548
 - wireless communication system, 548
 - glottis, 4
 - GMSK. *See* Gaussian-filtered minimum shift keying
 - Gold's theorem, 505
 - Gold sequences (codes)
 - class of, 505
 - correlation properties of, 507
 - good codes, 631
 - Gram-Schmidt orthogonalization procedure, 315–317
 - granular noise, 220
 - and distortion, 221
 - versus quantization noise, 221
 - Gray coding scheme, 422–423
 - Gray encoder, 276
 - Gray-encoded dibits, 363
 - GSM. *See* Global System for Mobile Communications
 - guard bands, 513
 - guard interval, 441
 - guard time, 547
 - guided propagation
 - channels based on, 15
 - types of, 15
- H**
- half-cycle cosine pulse, 389
 - half-cycle sine pulse, 389–390
 - half-duplex link, 628
 - Hamming distance, 637, 661, 666
 - Hamming single-error correcting code, 653
 - Hamming weight, 637, 666
 - handover, 530
 - hard-decision coding, 630
 - hard-decision decoders, 629–630
 - hard-decision demodulation, 669
 - hard decisions, 630
 - harmonic distortion, 112
 - harmonic structure, 4
 - head end, 17
 - hearing mechanism, 4
 - Hermitian transposition, 443
 - Hertz, Heinrich, 26–27
 - heterodyning function, 128
 - hexagonal cellular geometry, 531
 - high-performing CAP system, 375
 - Hilbert transform, 374, 408, 723–725
 - properties of, 725
 - of a signal, 724
 - Hilbert-transform pair, 376, 724
 - Hockham, G. A., 29
 - hop rate, 501
 - horizontal retrace, 5
 - host, 13
 - Huffman code
 - algorithm used to synthesize, 578
 - as a class of prefix codes, 578
 - drawback of, 580
 - nonuniqueness of, 579
 - Huffman coding, 578–580, 616
 - basic idea of, 578
 - compared to Lempel-Ziv algorithm, 581
 - and data compression, 8
 - as entropic coding, 8
 - Huffman decoding, 8–9
 - Huffman encoding process, 578, 579
 - human auditory system, 234
 - human communication, 4
 - hybrid-modulated signal, 123
 - hybrid modulation process, 374
 - hybrid transformer
 - definition of, 278
 - simplified circuit of, 278–279
- I**
- I-channel, 97
 - ideal baseband pulse transmission, 262
 - ideal delay element, 268
 - ideal envelope detector, 135
 - ideal frequency discriminator, 124
 - ideal narrowband filter, 45
 - ideal Nyquist channel, 262–264, 265–266
 - difficulties of, 263–264
 - use of, 263–264
 - ideal sampled signal, 184
 - ideal slope circuit
 - characterized by, 121–122
 - frequency response of, 122
 - ideal system, 601
 - identity matrix, 329
 - image interference, 129
 - impossible event, 704
 - impulse function, 62
 - impulse noise, 446
 - impulse response, 656, 718
 - index of performance, 224
 - indirect frequency modulation, 120–121
 - individual demodulators, 106
 - infinite bandwidth, 602
 - information, 2
 - information-bearing signal, 31–32, 88
 - in the digital domain, 277
 - multiplying by the PN signal, 488
 - information capacity, 598, 616
 - of a channel, 597–598
 - defined as, 23, 598
 - evaluation of, 598
 - increasing of, 599
 - information capacity theorem, 616
 - application of, 607
 - argument for, 599–600
 - as a colored noise channel, 607
 - and Gaussian channels, 597
 - implications of, 601–603
 - system parameters of, 599
 - water-filling interpretation of, 610
 - information-theoretic concepts, 572
 - information theory
 - fundamental limits in, 567
 - important result of, 591
 - as a mathematical discipline, 567
 - Shannon's landmark paper, 567
 - information transmission, 581
 - information vector, 633
 - inner conductor, 17
 - inner product, 313, 314
 - innovation symbol, 581
 - in-phase channel, 408
 - in-phase coherent detector, 97
 - in-phase component, 93
 - power spectral density of, 395–396
 - properties of, 65–66
 - representation of, 67–69
 - in-phase noise component, 131
 - input alphabet, 582
 - input signal-to-noise ratio, 134
 - definition of, 131
 - equation for, 497
 - insertion loss, 16
 - instantaneous codes, 577

- instantaneous frequency, 110
 - definition of, 163
 - equation for, 108
 - instantaneous sampling, 184
 - integration
 - beneficial effects of, 221–222
 - as a linear operation, 223
 - interface, 11
 - interference
 - average power of, 495
 - effect of, 490
 - and fading, 71–72
 - strength of, 148–149
 - as unintentional or intentional, 479
 - interference suppression, 148–149
 - interframe redundancy, 9
 - interlaced fields, 5
 - interlaced raster scan, 5
 - interleaver
 - definition of, 674
 - types of, 674
 - use of, 675
 - intermediate frequency (IF), 128
 - intermediate frequency (IF) band, 18
 - Internet, 13–14
 - architecture of, 13–14
 - evolution of, 28–29
 - growth of, 29
 - protocols for, 13–14
 - Internet architecture
 - functional blocks of, 13–14
 - Internet protocol (IP), 13–14
 - Internet Service Provider (ISP), 420
 - and communication between PSTN, 425
 - and public switched telephone network (PSTN), 420–425
 - and voice modems, 420–422
 - interpixel redundancy, 9
 - interpolation formula, 186
 - interpolation function, 186, 427
 - intersymbol interference (ISI), 259–261, 398
 - and bit errors, 247
 - as channel impairment, 379
 - condition of, 282–283
 - under designer's control, 268
 - as a dominant impairment, 279
 - effects of, 294, 296
 - as a form of interference, 296
 - minimizing effects of, 260
 - and noise presence, 294
 - overcoming effects of, 441
 - in peak distortion, 288
 - and timing error, 266
 - as an undesirable effect, 267
 - intrinsic information, 679
 - invariance, 331
 - inverse discrete Fourier transform (IDFT), 442, 445
 - inverse Fourier transform, 186, 715
 - inverse mapping, 589
 - inverse-square law, 519
 - irreducible polynomial, 505
 - irregular codes, 691
 - irregular interleavers, 691–692
 - irregular LDPC code, 692
 - irregular turbo code, 691, 692
- J**
- jammer, 493
 - strategy of, 495
 - types of, 508
 - waveforms of, 508
 - jammer, barrage noise, 508
 - jammer, multitone, 508
 - jammer, pulse noise, 508
 - jammer, single-tone, 508
 - jamming margin, 499
 - jamming signal, 488
 - jamming waveforms, 488
 - jitter, 208
 - joint distribution function, 33, 709
 - joint moments, 713–714
 - Joint Photographic Experts Group (JPEG), 8
 - joint probability, 706, 707
 - joint probability density function, 594, 709–710
 - joint probability distribution, 583
 - JPEG image coding standard, 8
- K**
- Kao, K. C., 29
 - keys, 756
 - key-schedule calculation, 753, 754
 - keystream, 744, 745
 - Kotel'nikov, V. A., 27
 - Kraft-McMillan inequality, 576–577
 - Kummer's differential equation, 740
- L**
- Lagrange multipliers
 - method of, 437
 - use of, 609
 - laser, 29
 - layer, 11
 - layered architecture, 11
 - layer-to-layer interface, 13
 - least-mean-square (LMS)
 - algorithm, 288–290, 557
 - for adaptive equalization, 288–289
 - and combined use, 297
 - equations for, 289
 - for linear adaptive prediction, 226
 - popularity of, 226, 227
 - similarities, 290
 - simplification of, 289
 - summary of, 289
 - uses of, 292
 - using matrix notation, 289
 - Leibniz's rule, 257
 - Lempel-Ziv algorithm, 8, 616
 - compared to Huffman coding, 581
 - definition of, 580
 - encoding process performed by, 580
 - standard for file compression, 581
 - Lempel-Ziv coding, 580–581
 - light, 6
 - likelihood functions, 322
 - linear adaptive prediction, 225–227
 - linear array signal processor
 - to design, 554
 - for the receiver, 554
 - requirements of, 554
 - linear block code
 - basic property of, 634
 - classes of, 653–654
 - decoding procedure for, 639
 - definition of, 632
 - mathematical structure of, 632–633
 - minimum distance of, 637
 - standard array of, 638
 - linear combiner, 547
 - linear delta modulator, 221, 232–233
 - linear diversity combining
 - structure, 545
 - linear equalization, 379, 556
 - linear function, 54
 - linearity property, 642
 - linear modulation
 - definition of, 93
 - examples of, 163

- linear modulation (*Continued*)
 - forms of, 93–94
 - types of, 93
 - linear modulation systems, 729
 - linear prediction, 223–227
 - linear predictive coding (LPC), 551
 - linear pre-emphasis and de-emphasis filters
 - applications in, 157
 - use of, 157
 - linear receiver, 248
 - design of, 283
 - performance of, 132
 - using coherent detection, 132
 - linear system, 718
 - linear time-invariant filter
 - defined as, 250–251
 - impulse response of, 44
 - as a matched filter, 250–251
 - use of, 248
 - linear time-invariant system, 719–720
 - line codes
 - candidates for, 281
 - comparison of, 281
 - for electrical representation, 204–207
 - power spectra of, 206
 - selection of, 280–281
 - types of, 205–207
 - use of, 204–207
 - line-scanning frequency, 5–6
 - link budget, 518
 - link budget analysis, 517
 - link budget balance sheet, 517
 - Lloyd-Max quantizer, 198, 200–201
 - loading problem, 438
 - loading process, 438
 - local loop, 420
 - local oscillator, 97
 - Lodge, Oliver, 27
 - logarithmic function, 322
 - log-likelihood function, 452
 - defined as, 322
 - defined for AWGN channel, 325
 - relationship of, 322
 - loop filter, 157, 159–160
 - loop-gain parameter, 159
 - lossless compression
 - definition of, 7–8
 - for digital text, 8
 - versus lossy compression, 8
 - lossless data compression, 575
 - lossy compression
 - definition of, 8
 - the preferred approach, 8
 - low-density parity-check (LDPC) codes, 683–686
 - advantages of, 684
 - block length of, 689
 - construction of, 684–685
 - decoding algorithm, 690
 - decoding of, 689–690
 - initialization of, 690
 - minimum distance of, 689
 - shared properties of, 693
 - statistical analysis of, 689
 - steps of, 690–691
 - use of, 684
 - low-noise amplifier, 515
 - low-weight code words, 684
 - LPC. *See* linear predictive coding
 - Lucky, Robert, 28
 - luminance signal, 6
- M**
- magnitude response, 45, 608, 719
 - magnitude spectrum, 715
 - main lobe, 368, 720
 - Manchester code, 207, 281
 - many-to-one mapping, 8
 - MAP decoder, 678
 - Marconi, Guglielmo, 27
 - marginal densities, 710
 - marginal probability distribution, 583
 - Markov process, 678
 - M-ary digital modulation techniques, 419–420
 - M-ary frequency-shift keying (MFSK), 398–400
 - consists of, 401
 - for frequency hopping systems, 500
 - property of, 400
 - M-ary FSK signal
 - bandwidth efficiency of, 401–402
 - bandwidth requirements, 401
 - orthogonal signals of, 401–402
 - power spectra of, 401
 - spectral analysis of, 401, 402
 - M-ary PAM system
 - in a channel bandwidth, 276
 - consideration of, 276
 - design complexity of, 277
 - power requirements of, 276–277
 - M-ary PSK
 - comparison of, 419
 - likelihood function for, 452
 - power-bandwidth requirements for, 419–420
 - signal constellations of, 365, 420
 - similar spectral and bandwidth characteristics, 419
 - special case, 365
 - symbol duration of, 367
 - symbol error equation for, 365–366
 - M-ary PSK signal
 - bandwidth efficiency of, 368
 - baseband power spectral density of, 367
 - power spectra of, 367
 - as spectrally efficient, 402
 - M-ary PSK systems, 449–450
 - M-ary QAM
 - detection for, 371
 - functions in, 369
 - performance of, 420
 - symbol error probability for, 371
 - transmitted energy in, 371
 - M-ary QAM signal, 372
 - M-ary quadrature amplitude modulation (QAM), 369–373
 - M-ary signal, 402
 - M-ary signaling scheme, 345–346
 - M-ary system, 276
 - masking threshold, 9, 234, 234–235
 - matched filter, 248–252, 286
 - in the frequency domain, 251
 - output, 406
 - properties of, 251–252
 - matched filter receiver
 - correlation and, 326–327
 - detector part of, 328
 - mathematical models
 - classes of, 31
 - in probabilistic terms, 31
 - matrixer
 - difference signal generation, 125
 - sum signal generation, 125
 - maximal-length sequence
 - autocorrelation function of, 482
 - balance property of, 482
 - choosing a, 484
 - defined as, 482
 - properties of, 482–484
 - maximal-ratio combiner, 547

- maximal ratio combining principle, 550
- maximum *a posteriori* probability (MAP) detection, 678
- maximum *a posteriori* probability (MAP) rule, 323–324
- maximum likelihood decision rule for an AWGN channel, 325
- purpose of, 325
- maximum likelihood decoder, 322–326
 - for computation, 324
 - defined as, 661
 - as an implementation device, 324
 - theory of, 660–661
- maximum likelihood decoding rule, 661
- maximum likelihood detection, 330, 337
- maximum likelihood detectors, 435
- maximum likelihood estimation of the carrier phase, 453–458
 - for problem solving, 449
- maximum likelihood rule, 324
- maximum likelihood signal detection, 346
- maximum-power transfer theorem
 - applying, 61
 - use of, 61
- Maxwell, James Clerk, 26
- mean, 35
- mean Doppler shift, 541
- mean functions, 35–39
- mean output noise power
 - definition of, 139
 - equation for, 141
- mean output signal, 139
- mean-square distortion, 199
- mean-square error, 285
 - as the cost function, 558
 - definition of, 284
- mean-square error criterion
 - for receiver design, 283
 - uses of, 288
- median signal strength, 530
- melodic structure, 4
- memoryless channel, 321
- memoryless Gaussian channel, discrete-time, 597
- memoryless quantizer, 194
- Mersenne prime length sequences, 485
- message, 155
- message bandwidth, 90
- message point, 322–323
- message polynomial, 643
- message signal, 132–133
 - description of, 2
 - generation of, 2
- message source, 348
- message spectrum
 - for negative frequencies, 103
 - requirement of, 99
- message vector, 660
- method of steepest descent, 225–226
- microcells, 531
- microphone, 15–16
- Middleton, D., 27
- minimum average energy, 332
- minimum distance
 - considerations of, 637–638
 - definition of, 637
- minimum distance decoder, 661
- minimum energy signals, 331–332
- minimum energy translate, 332
- minimum mean square error (MMSE), 558
 - criterion for, 558
 - equalizer, 285
 - receiver, 286–287
- minimum shift keying (MSK), 387
 - as a form of binary FSK, 361
 - signal-space diagram of, 389–392
- minimum shift keying (MSK) signal
 - power spectra of, 360
- mixer
 - consists of, 103–104
 - function of, 129
 - operation of, 105
- mobile radio, 18, 529
- mobile radio channel
 - capability of, 18
 - as a linear time-varying channel, 18
 - propagation effects of, 18
- mobile switching center, 530
- mobility, 18
- modem, 7
 - configuration of, 421
 - as a conversion device, 420
 - design of, 421
 - and the Internet, 420
 - portions of, 420
- modem, facsimile, 420
- modem configuration, 421
- modified Bessel equation, 738
- modified Bessel function, 737–739
- modified duobinary code, 281
- modified duobinary coder
 - responses of, 272, 273
 - useful feature of, 272–273
- modified duobinary conversion filter, 273
- modulated signal (wave), 95
- modulating signal (wave), 88
- modulation, 19
 - operations of, 628
 - stages of, 490
- modulation format, 101–102
- modulation index
 - defined as, 110
 - restriction of, 112
 - small values of, 119
 - values of, 117–118
- modulation process, 19–21
 - classification of, 20
 - definition of, 88
- modulation scheme, bandwidth-conserving, 354
- modulation system, binary-coded, 197
- modulator-demodulator, 420
- modulo- 2π correction logic, 364
- moments, 712
- Morse, Samuel, 26
- Morse code, 26, 574
- Motion Picture Experts Group (MPEG), 9, 234
- moving-coil receiver, 15–16
- MPEG-1 audio coding standard, 9
 - capabilities of, 234
 - operation of, 235
 - performance of, 234
 - suitable for, 10
- MPEG-1 video coding standard, 9
- MPEG audio coding standard, 234, 237
- MSK receiver, 394, 395
- MSK signal, 391
 - characteristics of, 396
 - demodulation of, 394
 - detection of, 394
 - error probability of, 392
 - generation of, 394, 396
 - possible forms of, 390
 - power spectra of, 394–396, 398
 - properties of, 396
- MSK system, 391
- MSK transmitter, 394, 395
- μ -law, 202–203

- multibeam antennas, 514
- multichannel data transmission system, 433, 434
- multichannel modulation, 440–441
 - basic idea of, 431
 - form of, 431, 465
- multichannel transmission system, 437–438
- multilevel encoding, 348, 378
- multiloop feedback circuit, 480
- multipath
 - physical phenomenon of, 513
 - presence of, 513
- multipath autocorrelation profile, 537
- multipath channel
 - as a channel impairment, 553
 - classification of, 541–542
 - frequency selection, 541–542
 - model of, 71–72
 - statistical characterization of, 535–542, 554
 - type of fading exhibited, 72
- multipath component, 549
- multipath intensity profile, 539
- multipath phenomenon
 - forms of, 532–533
 - in a mobile radio environment, 18
 - nature of, 532
- multiple access
 - basic types of, 513–514
 - versus multiplexing, 513
- multiple-access interference (MAI), 548
- multiple-access system
 - goal of, 549
 - interference, 548
- multiple-access techniques
 - common feature of, 514
 - defined as, 513–514
 - ideas behind, 513, 514
 - and shared communication resources, 513
- multiple-receiver combining techniques, 544
- multiplexed signal, 125–126
- multiplexed systems, 347
- multiplexer, 446
- multiplexing, 105
 - definition of, 20
 - of digital signals, 214
 - types of, 20–21
- multiplier, 157, 646
- multi-pulse excited LPC, 551–552
- multitone jammer, 508
- multiuser communications, 512
 - environment, 396
 - types of, 559
- music
 - as a source of information, 4
 - structures of, 4
- musical signals
 - and channel bandwidth, 4
 - versus speech signals, 4
- mutual information, 584–585
 - for continuous ensemble, 593–597
 - defined as, 596
 - properties of, 585–587, 596
 - in the Shannon model, 746
- N**
- narrowband AM signal (wave), 112–113
- narrowband FM signal (wave), 111–112, 113
- narrowband FM waves, 112–113
- narrowband frequency modulation, 111–113
- narrowband noise, 64
 - characterization of, 64
 - components of, 64, 65–66
 - effects of, 64
 - representation and coordinate system for, 67–68
 - representation of, 64–66, 67–69
- narrowband noise analyzer, 64–66
- narrowband noise synthesizer, 64–66
- narrowband phase modulator, 120
- narrowband process, 65–66
- National Television System Committee (NTSC), 6
- natural frequency, 160
- N*-dimensional Euclidean space
 - angles in, 312
 - lengths of vectors, 312
 - vectors in, 311, 312
- N*-dimensional vector, 311
- near-end crosstalk (NEXT), 279–280, 380
- nearest neighbor condition, 200
- near-far problem, 548, 549
- negative-going click, 150
- network, 10
- network, interconnected, 13
- network resources, 11, 14
- 90 degrees rotational invariance, 422–423
- noise
 - absence of, 261
 - calculations, 61
 - in communications systems, 58
 - definition of, 3
 - effect of, 3
 - minimizing the effects of, 157, 260
 - presence of, 589
 - signals in, 322–326
 - sources of, 3, 58
 - as unwanted signals, 58
- noise analysis
 - comparison of, 163–164
 - at the receiver, 523
- noise calculations, 61
- noise channel, colored, 607–611
- noise enhancement, 283
- noise equivalent bandwidth, 722–723
 - for bandpass filters, 723
 - defined as, 723
- noise figure, 523–526
- noise-free estimates, 545
- noise jammer, barrage, 508
- noise jammer, partial-band, 508
- noise margin, 281
- noise masking, 234
- noise power, 3, 61
- noise power, average
 - output calculation of, 151
 - determining, 145–147
- noise-quieting effect, 147
- noise-to-mask ratio (NMR), 234
- noise vector
 - covariance matrix of, 330
 - particular realization of, 323
 - statistical characteristics of, 330
- noisy receiver model, 130
- noisy resistor, 60
- noncoherent binary DPSK, 407
- noncoherent binary frequency-shift keying (FSK), 407, 413–414
 - bit error rate for, 414
 - consideration of, 413
 - expressions for, 417
- noncoherent *M*-ary FSK detector, 500
- noncoherent matched filter, 406
- noncoherent orthogonal modulation, 407–409
- noise performance of, 407

- receiver for, 408
 - special case of, 414
 - noncoherent receiver, 405–406, 409
 - nondata-aided early-late delay (NDA-ELD) synchronizer, 458
 - nondata-aided recursive algorithm, 455
 - nondata-aided synchronization, 449
 - nonflat channel, 542
 - nonlinearity
 - basic forms of, 126
 - effects of, 126
 - the presence of, 126
 - nonlinear modulation process, 163
 - nonlinear pre-emphasis and de-emphasis techniques, 157
 - nonredundant coding, 421
 - nonreturn-to-zero level encoder, 359
 - nonsystematic code, 656
 - nonuniform quantizer, 202–203
 - normalized Gaussian distribution, 54–55, 56
 - normalized transmission bandwidth, 164
 - North, D. O., 27
 - North American digital TDM hierarchy, 214–215
 - Norton equivalent circuit, 60
 - Noyce, Robert, 28
 - NRZ binary data, 398
 - NTSC system, 6
 - null event, 704
 - null-to-null bandwidth, 368, 721
 - number-controlled oscillator (NCO), 458
 - Nyquist, Harry, 27
 - Nyquist bandwidth, 262
 - Nyquist criterion, 262
 - Nyquist interval, 186–187
 - Nyquist rate, 186–187, 262
- O**
- observable element, 321
 - observation space, 324, 325–326
 - observation vector, 321, 331
 - octaphase-shift-keying, 365–366
 - octet, 7
 - offset QPSK, 362
 - onboard switching, 516
 - one-time pad, 747
 - on-off level encoder, 384
 - on-off signaling, 205
 - open systems interconnection (OSI) reference model, 11
 - optical communication, 29
 - optical fiber
 - advantages of, 17
 - consists of, 17
 - properties of, 17
 - as a transmission medium, 17
 - optical transmission system, 15
 - optimization problem, 438–440
 - optimum CAP receiver, 379
 - optimum decision rule, 323–324
 - optimum equalizer, 288
 - optimum filter
 - consists of, 378–379
 - impulse response of, 250–251
 - optimum information capacity, 610
 - optimum in-phase filter, 378–379
 - optimum linear receiver, 282–287
 - interpretation of, 286
 - and transmitter, 286
 - optimum quadratic receiver, 403–405
 - optimum quadrature filter, 378–379
 - optimum quantization problem, 199–201
 - optimum receiver
 - as a correlation receiver, 326–327
 - design of, 310
 - detector part of, 328
 - subsystems of, 326–327
 - optimum receiver subsystems, 326–327
 - optimum threshold, 257
 - orthogonal frequency-division multiplexing (OFDM), 447–448
 - applications of, 448
 - techniques for broadcasting, 344
 - use of, 447–448
 - orthonormal basis functions, 311, 315, 494
 - as a desirable property, 440–441
 - for MSK, 390
 - shortcomings of, 441
 - orthonormal matrix, 443
 - oscillator, crystal-controlled, 120
 - OSI model, 11–13
 - outer conductor, 17
 - output alphabet, 582
 - output current, 4–5
 - output noise power, 523
 - output signal-to-noise ratio
 - of AM receiver, 136
 - calculation of, 151
 - versus carrier-to-noise ratio, 151
 - definition of, 131, 145
 - determining, 133
 - equation for, 139, 497
 - to evaluate, 135
 - improvement factor in, 155–156
 - increasing of, 155
 - of a uniform quantizer, 197
 - overmodulated, 90
- P**
- packet switching
 - network, 11
 - principle of, 11
 - pairwise error probability, 334
 - PAM. *See* pulse amplitude modulation (PAM)
 - PAM signal
 - generation of, 188–189
 - performance of, 191
 - sampling of, 189
 - transmission of, 191
 - waveform of, 188–189
 - PAM system, 191
 - parallel encoding scheme, 676
 - parallel-to-serial converter, 445
 - parity bit, 7, 632
 - parity-check equations, 634
 - parity-check matrix, 634
 - parity-check polynomial
 - defined as, 644
 - reciprocal of, 645
 - partial-band noise jammer, 508
 - partial-response signaling, 269, 274–275
 - partial-response signaling scheme, 267
 - achieved by, 274–275
 - classes of, 275
 - useful characteristics of, 275
 - partitioning, 670
 - passband basis functions
 - properties of, 434–436
 - time variations of, 373
 - passband data transmission, 344
 - alternative techniques for, 465
 - applications of, 344
 - communication channel used for, 344
 - over nonlinear channels, 345

- passband data transmission systems
 - determining the bandwidth efficiency of, 348
 - goal of, 346
- passband in-phase filter, 375, 378
- passband in-phase pulse, 375
- passband line code, 344
- passband modulation, 729
- passband pulse, 375
- passband quadrature pulse, 374
- passband signaling waveform, 729
- passband transmission model, 348–349
- path loss, 522
- pattern matching operation, 615
- PCM. *See* pulse-code modulation
- peak distortion, 288
- peak pulse signal-to-noise ratio
 - defined as, 248–249, 250
 - of a matched filter, 251
- peer process, 13
- percentage modulation, 90
- perception, 4
- perceptual coding, 9
- perfect security, 746–747
- periodicity, 718
- periodic signals, 717–718
- periodogram, 51
- persistence of vision, 5
- personal computers (PCs), 6, 6–7
- “phase and gain adjustors”, 550
- phase continuity, 388
- phase correction, 365
- phase decisions, 394
- phase demodulation, 492
- phase-difference computer, 364
- phase discriminator, 97
- phase distortion
 - and the human ear, 99–100
 - presence of, 99–100
- phase error
 - defined as, 158
 - effect of, 99
- phase-error generator, 459
- phase-locked loop, 121
 - complexity of, 159–160
 - components of, 157–160
 - limitation of, 159–160
 - loop filter in, 158
 - model of, 158–160
 - simplest form of, 159–160
 - understanding, 157, 158
 - use of, 157–160
- phase-locked loop demodulator, 152
 - and threshold extension capacity, 154
 - as a tracking filter, 154
- phase-locked loop theory, 157
- phase modulation (PM), 20, 108, 729
- phase modulation schemes, 368
- phase nonlinearity, 127
- phase recovery, 450
- phase-recovery circuit, 345
- phase response, 719
- phase selectivity, 723–725
- phase sensitivity, 108
- phase-shift keying (PSK), 24
 - and coherent systems, 490
 - of phase modulation, 345
 - represented by, 464
 - signaling scheme, 344–345
- phase-shift keying (PSK) schemes, 418
- phase spectrum, 715
- phase tree, 388
- phase trellis, 388–389
- phasors, 532
- photocathode, 4
- photodetector circuit, 58–59
- physical layer, 13
- $\pi/4$ -shifted DQPSK signals, 364
- $\pi/4$ -shifted DQPSK symbols, 363
- $\pi/4$ -shifted QPSK scheme, 363
- $\pi/4$ -shifted QPSK signal
 - demodulation of, 365
 - residing in one of eight possible phase states, 362
- pictures
 - and the human visual system, 4
 - perception of, 4
 - as a source of information, 4
- piecewise linear approximation, 203
- Pierce, John R., 29
- pilot carrier, 99
- plain old telephone service (POTS), 281–282
- plaintext, 742
- PM signal, 109
- PN sequence
 - correlation properties of, 506
 - as an independent and identically distributed (iid) binary sequence, 496
 - as a reference signal, 550
- pointer, 581
- point-to-point communication, 2–3
- Poisson’s sum formula, 718
- Poisson distribution, 59
- polar nonreturn-to-zero (NRZ) level encoder, 352
- polar nonreturn-to-zero (NRZ) signaling
 - binary PCM system based on, 253
 - disadvantages of, 205–206
- polyvinylchloride (PVC) sheath, 16
- positive-going click, 150
- postdetection filter, 143
- power, available, 61
- power control
 - in CDMA systems, 549
 - use of, 549
- power gain, 523
 - of an antenna, 520
 - concept of, 521
 - definition of, 520
- power-limited channel, 3
- power spectra, 347
- power spectral density, 44–46, 347
 - and amplitude spectrum, 50–52
 - frequency portions of, 155
 - graphical summary of, 75–76
 - properties of, 46–47
 - of random process, 50, 52
 - significance of, 45
- power spectrum, 4, 45
- power theorem, 520
- Poynting vector, 519
- PPM. *See* pulse-position modulation
- precoded duobinary scheme, 270–271
- prediction, 9
- prediction filter, 228
- pre-envelope
 - basic property of, 732
 - defined as, 725, 730
 - determining, 726
 - quadrature components of, 374
- prefix code
 - definition of, 575–576
 - distinguished by, 577
 - property of, 576
- prefix coding, 575
- prefix condition, 575
- premodulation low-pass filter, 396–397

- preset threshold values, 277
 - primary colors, 6
 - represented by video signals, 6
 - transmission of, 6
 - primary rate, 214–215
 - primitive BCH codes, 653
 - primitive polynomial, 505
 - principle of analysis by synthesis, 551–552
 - principle of rotational invariance
 - illustration of, 330–331
 - stated as, 330
 - principle of superposition, 718
 - principle of translational invariance
 - application of, 331–332
 - stated as, 331
 - probabilistic code, 693
 - probabilistic concepts, 703–707
 - probabilistic decoder, 630
 - probability
 - axioms of, 704–706
 - basic properties of, 705–706
 - of bit error, 384
 - of a correct decision, 357
 - of error, 254, 409
 - of symbol error, 258, 328–329, 352
 - probability, conditional, 706–707
 - probability density function, 67–68, 255, 594, 708–709, 710
 - probability distribution, 583
 - probability of error, 328–329, 497–499
 - invariance of, 329–331
 - for a noisy channel, 589
 - union bound on the, 332–335
 - probability of occurrence, 568
 - probability of symbol error, 334
 - determination of, 373
 - evaluation of, 346–347
 - formula for, 401
 - for signal constellation, 337
 - probability system, 704
 - probability theory, 703, 705
 - processing gain (PG)
 - defined as, 229, 497
 - produced by, 229
 - product cipher, 749
 - product modulator, 94, 98–99, 111, 490
 - propagation, 4
 - propagation effects, 532–535
 - propagation time delay, 516
 - protocol
 - of the Internet, 13–14
 - types of, 13–15
 - pseudo-noise (PN) sequence, 480, 488
 - consists of, 288
 - generation and properties of, 480
 - as a training sequence, 288
 - pseudo-random-ordered sequence, 500
 - PSK. *See* phase-shift keying
 - psychoacoustic modeling, 9
 - psychovisual redundancy, 9
 - public-key cryptographic system, 755
 - public-key cryptography, 742, 755–757
 - public-key system, 759
 - public switched telephone network (PSTN), 237, 420
 - as an analog network, 420, 421
 - distortion on, 286–287
 - efficient use of, 425
 - pulse, 5
 - pulse-amplitude modulated signal, 429
 - pulse-amplitude modulation (PAM), 20, 188–191, 236
 - definition of, 188
 - and modulator design, 277
 - and natural sampling, 188
 - pulse-code modulation (PCM), 193, 615
 - advantages of, 217, 237
 - bandwidth requirement of, 218
 - basic condition of, 194
 - cost of advantages, 217–218
 - definition of, 201
 - as a form of digital pulse modulation, 237
 - performance of, 227
 - as the preferred method, 20
 - for speech coding, 229–230
 - use of, 217–218, 560
 - pulse-code modulation (PCM) link, 210
 - pulse-code modulation (PCM) receiver, 258
 - pulse-code modulation (PCM) signal, 208
 - pulse-code modulation (PCM) system, 218
 - basic operations of, 201
 - characteristic of, 210
 - description of, 201–209
 - and interference, 210
 - noise considerations in, 209–210
 - operation of, 212
 - performance influenced by noise, 209
- pulse demodulator, 211
 - pulse-duration modulation (PDM), 20, 191–192, 236
 - pulse-modulated signal, 237
 - pulse modulation
 - families of, 183
 - forms of, 191–193, 237
 - lossy nature of, 237–238
 - method used to transmit, 211
 - as source coding techniques, 237
 - standard digital form of, 20
 - types of, 20
 - pulse-modulation process
 - incurred information loss of, 238
 - loss of information and designer control, 238
 - pulse modulation systems, 236
 - pulse modulator, 211
 - pulse noise jammer, 508
 - pulse-position modulation (PPM), 20, 192, 236–237
 - pulse-position modulation (PPM) system
 - versus frequency modulation system, 193
 - noise analysis of, 193
 - performance of, 193
 - pulse shaping, 247
 - pulse-shaping filter
 - desirable properties of, 396–397
 - Gaussian impulse response of, 398
 - pulse-shaping function, 373
 - pulse-width modulation, 191
 - punctured code, 676
 - puncturing, 676
- Q**
- QAM. *See* quadrature-amplitude modulation
 - Q-channel, 97
 - QPSK. *See* quadrature-phase-shift keying
 - quadrants, 421
 - quadratic receiver
 - equation for, 405
 - forms of, 405

- quadrature-amplitude modulation (QAM), 97–98
 - versus CAP, 369
 - cross constellation, 371
 - quadrature-amplitude modulation (QAM) constellations, 369–370
 - quadrature-amplitude modulators, 433–434
 - quadrature-carrier multiplexing, 97–98, 354
 - quadrature-carrier multiplexing system, 98
 - quadrature channel, 408–409
 - quadrature component, 93
 - power spectral density of, 386
 - properties of, 65–66
 - role of, 93, 101
 - quadrature modulation, 670
 - quadrature noise component, 131
 - quadrature null effect, 97
 - quadrature-phase coherent detector, 97
 - quadrature receiver
 - channels of, 408
 - using correlators, 405–406
 - using matched filters, 405–406
 - quadrature-phase-shift keying (QPSK), 354, 354–361
 - characterization of, 354–355
 - error probability of, 356–358
 - mode of operation, 425
 - motivation for using, 508
 - signal-space diagram, 354–355
 - quadrature-phase-shift keying (QPSK) receiver, 360
 - quadrature-phase-shift keying (QPSK) signal
 - amplitude fluctuations, 362
 - and binary PSK signal, 360
 - commonly used constellations for, 362
 - filtered, 361
 - interference production, 396
 - observations of, 360
 - phase transitions of, 361
 - power spectra of, 360–361
 - quadrature-phase-shift keying (QPSK) transmitter, 359
 - quality of service (QoS), 14
 - quantization
 - application of, 202–203
 - and coding, 9
 - purpose of, 8
 - types of, 194–195
 - use of, 195
 - function of, 196
 - types of, 220
 - quantization noise, 195–197, 228
 - designer's control of, 209
 - in delta modulation, 221
 - as a function of time, 195
 - and human ear perception, 9
 - in PCM systems, 209
 - quantization process, 193–195, 236
 - in the generation of a binary PCM wave, 193
 - illustration of, 195
 - nonlinear nature of, 198
 - results of, 20
 - quantization table, 8
 - quantized excitation, 552
 - quantized filter parameters, 552
 - quantizer
 - characteristics of, 194
 - classes of, 615
 - components of, 199
 - as a signal compressor, 615
 - types of, 194
 - quantizer, nonuniform, 202–203
 - quantizer input, 221
 - quantum, 194
 - quaternary system
 - eye diagram for, 294–295
 - output of, 276
- R**
- radiation efficiency factor, 520
 - radiation intensity, 519
 - radiation-intensity pattern, 520
 - radio communication link, 522
 - radio communication system, 31
 - radio link analysis, 517–523
 - radio propagation
 - in free space, 512
 - in urban areas, 532–533
 - radix, 570, 576–577
 - raised cosine spectrum
 - flat portion of, 264–265
 - rolloff portion of, 264–265
 - RAKE receiver, 549–550
 - basic idea of, 549
 - consists of, 549
 - as a diversity receiver, 549
 - techniques of, 559
 - random binary sequence, 482
 - random experiment
 - description of, 703
 - features of, 703
 - random hopping, 500
 - random interference, 31–32
 - random process
 - average power of, 610
 - classes of, 75
 - definition of, 33
 - ensemble averages of, 41
 - expectations of, 41
 - in linear systems, 42–44
 - mathematical definition of, 32–33
 - parameter of, 75
 - properties of, 32
 - through a linear time-invariant filter, 42–44
 - random variable, 33, 594, 708–710
 - definition of, 708
 - description of, 708
 - distribution of, 55–56
 - expected value of, 711
 - mean of, 711
 - standard deviation of, 712
 - variance of, 712
 - random vectors, 594
 - raster scanning, 4–5
 - rate distortion function, 612–613, 616
 - application of, 616
 - definition of, 613
 - rate distortion theory, 611–613
 - application of, 612
 - main parameters of, 613
 - and Shannon's coding theorems, 612
 - Rayleigh's energy theorem
 - definition of, 251
 - use of, 251–252
 - Rayleigh distribution, 68–69, 70, 74–75
 - Rayleigh fading channel, 536, 541
 - binary signaling over, 542–547
 - performance of, 545
 - received signal
 - components of, 31–32
 - mean value of energy, 543
 - received signal point, 323
 - received vector, 660
 - receive filter, 259
 - receiver
 - of an analog communication system, 88–89
 - assumptions of, 403
 - de-emphasis in, 154–155
 - model, 130
 - moving-coil, 15–16

- noise performance of, 387
 - as an optimum maximum likelihood detector, 436
 - and preprocessing the received signal, 64
 - receiver model, 130
 - receiving antenna, 518
 - reciprocity principle in antennas, 521
 - reconstruction filter, 187–188
 - reconstruction levels, 194
 - rectangular function, 262
 - recursion theorem, 682
 - recursive algorithm
 - for phase recovery, 454
 - for timing recovery, 457–458
 - recursive symmetric convolutional (RSC) code, 675
 - recursive Costas loop, 454
 - convergence behavior of, 461, 462
 - operations of, 458
 - phase-acquisition behavior of, 459
 - for phase synchronization, 454
 - recursive early-late-delay synchronizer, 463–464
 - redundancy
 - addition of, 626
 - basic forms of, 9
 - controlled use of, 628
 - redundant information, 227, 575
 - Reed-Solomon codes, 654, 693
 - Reeves, Alec, 27
 - reference antenna
 - definition of, 519
 - as an isotropic source, 519
 - reference signal, 557
 - reflector antenna, 522
 - regeneration, 208
 - regenerative repeater, 208
 - region of integration, 332
 - regular-pulse excitation, 552
 - regular turbo code, 692
 - relative-frequency approach, 703–704
 - relative phase difference, 414
 - relative phase shift, 532
 - replication of delta function property, 717
 - representation levels, 194
 - reproduction quality, 5–6
 - residual amplitude modulation, 112
 - resolution, 6
 - resolution of uncertainty, 568
 - response time, 718
 - reverse link
 - versus forward link, 559
 - subbands for, 547
 - Rician distribution
 - graphical presentation of, 70–71
 - normalized form of, 71
 - Rician fading channel, 536
 - Rivest-Shamir-Adleman (RSA) algorithm, 759
 - Rivest-Shamir-Adleman (RSA) system, 757
 - rms duration, 722
 - robust system, 22
 - rolloff factor, 265
 - frequency response for, 265–266
 - time response for, 265–266
 - root mean square (rms) bandwidth, 721
 - rotated noise vector, 330
 - router
 - defined as, 13
 - and host devices, 10
 - primary purpose of, 10
 - row vectors, 633
 - RS-232 standard, 6–7
 - RSA algorithm, 757–758
 - RSA cryptoalgorithm, 758
 - RSA trapdoor one-way function, 758
- S**
- sample functions, 32–33
 - sample point, 32, 704
 - sample space, 32, 704
 - sampling, 201–202
 - sampling period, 184
 - sampling process, 236
 - and digital signal processing and digital communications, 184
 - in the generation of a binary PCM wave, 193
 - and pulse modulation systems, 184, 236
 - use of, 184
 - sampling rate, 184, 201
 - sampling theorem, 201, 236
 - for band-limited signals, 186–187
 - derivation of, 186–187
 - essence of, 184
 - of a pulse-modulation system, 186–187
 - recurrent nonuniform equivalent form of, 427
 - satellite
 - for communication, 29
 - in geostationary orbit, 18
 - services of, 18–19
 - satellite channel
 - capabilities of, 516
 - coverage of, 18
 - remote area access, 18
 - satellite communications, 514–517
 - frequency band for, 19
 - global coverage, 512
 - most popular frequency band for, 515
 - as a type of multiuser communications, 512
 - satellite communication system, 18
 - design of, 517
 - global coverage, 559
 - rely on, 512
 - unique system capabilities of, 18–19
 - scalar quantization
 - form of, 194
 - use of, 194
 - scalar quantizer
 - conditions for optimality of, 198–201
 - designing of, 198
 - as a simple signal compressor, 615
 - scanning, 4–5, 4–6
 - scanning spot, 5
 - scattered beams, 71
 - scatterers, 71–72
 - scattering function, 539
 - Schwarz's inequality
 - as a mathematical result, 249–250
 - proving, 313, 314
 - SDMA, 516
 - second flyback. *See* vertical retracesecnd-order digital filter, 454–455
 - second-order feedback system, 160
 - secrecy, 745
 - secret key
 - versus public key, 759
 - selection of, 745
 - secret-key cryptoalgorithm, 751
 - secret-key cryptography, 742–743
 - secret-key system, 759
 - secure channel, 743
 - secure communications
 - in a hostile environment, 479
 - need for, 742

- security of transmission, 490
- segments, 11
 - See also* packets
- separability theorem, 681
- sequential scanning
 - of pictures, 4
 - process of, 4–6
- serial-to-parallel converter, 445
- Shannon, Claude
 - capacity theorem, 611
 - and “The Mathematical Theory of Communication”, 27–28
 - and the theoretical foundations of digital communications, 27–28
 - Shannon’s capacity theorem, 611
 - Shannon’s fundamental bound for perfect security, 747
 - Shannon’s information capacity theorem, 23–24, 433
 - Shannon’s information theory, 617
 - Shannon’s second theorem, 616
 - Shannon’s third remarkable theorem, 616
 - Shannon’s third theorem, 599
 - Shannon limit, 602
 - Shannon model of cryptography
 - method of confusion, 749
 - method of diffusion, 749
 - methods of designing, 749
 - shift parameters, 531
 - shift register, 481
- Shockley, William, 28
- shot noise, 58–60
- sideband, upper and lower, 91
- side information, 230
- sigma-delta modulation, 222
- signal
 - definition of, 3–4
 - detection in noise, 322–326
 - dimensions of, 3–4
 - received versus transmitted, 2
- signal bandwidth, 3
- signal constellation, 322–323, 337
 - as circularly symmetric, 335
 - constructed from one-dimensional PCM symbols, 429
 - defining minimum distance of, 335
- signal detection problem, 322
 - likelihood function for, 405
 - stated as, 323
- signal energy-to-noise spectral density ratio, 252
- signal fading, 532–533
- signal-flow graph, 665–666
- signaling binary information, 345
- signaling interval, 568
- signaling rate, 276
- signal parameters, 403
- signal power average, 3
- signal regeneration, 208
- signal-space analysis, 337
- signal-space dimensionality, 312, 493
- signal-space representations
 - of the interfering signal (jammer), 493
 - of the transmitted signal, 493
- signals with unknown phase, 403–406
- signal-to-mask ratio (SMR), 234
- signal-to-noise ratio
 - basic definitions of, 3, 130–132
 - at the device output, 524
 - of an FMFB receiver, 153
 - limitation of, 261
 - of the source, 524
- signal-to-noise ratio gap, 432
- signal-to-quantization noise ratio, 229
- signal transitions, 207
- signal transmission decoder, 326, 349
- signal transmission encoder, 348, 352
- signal variability, 530
- signal vector, 311
- simplex signals, 342
- signum function, 724
- sinc function, 262
- sine wave plus narrowband noise, 69, 69–71
- single-key cryptography, 742
- single keyed oscillator, 384
- single-letter distortion measure, 612
- single sideband (SSB) modulation, 98–100, 163
 - basic operation in, 103
 - definition of, 93
 - in frequency-division multiplexing, 106
- single-sideband modulated signal, 98–99
- single-tone FM signal, 112–113
- single-tone jammer, 508
- single-tone modulation
 - and a narrowband FM signal, 110
 - and a wideband FM signal, 110
- sinusoidal carrier wave
 - defined as, 90
 - waveform of, 490, 492
- sinusoidal modulating signal (wave), 110
- sinusoidal modulation, 112–113
- sinusoidal wave, 88
- slicing levels, 277
- slope circuit, 121–122
- slope network, 143
- slope overload distortion, 220, 221
- slow FH/MFSK signal, 501
- slow FH/MFSK system, 502
- slow-frequency hopping, 500–502
- smoothness, 223
- SNR ratio. *See* signal-to-noise ratio
- soft-decision coding, 630
- soft-decision decoding, 669
- soft decisions, 630
- soft input-hard output, 693
- soft input-soft output, 693
- SONET, 15
- source code, 574
 - type of, 575
 - variability in lengths of, 579
- source code word, 21
- source coding, 574
 - dissection of, 616–617
 - for efficient communication, 567
 - with a fidelity criterion, 611–612
- source-coding theorem, 574–575, 612, 616
 - average code-word length of, 611
 - in Shannon’s first theorem, 574–575
- source decoder, 575
- source encoder, 21, 574
 - functional requirements of, 574
 - purpose of, 21
- spaced-frequency spaced-time correlation function, 538
- space diversity, 544–545
- space diversity technique, 546
- space-division multiple access (SDMA), 514
- space-time processor, 557
- spatial phenomenon, 534
- spatial sampling, 4–5
- spectral analysis, 110
- spectral content, 492, 493
- spectral decomposition, 443

- spectrally efficient modulation, 347
- spectrally efficient schemes, 347
- spectral nulls, 368
- spectral shaping, 348
- spectrum, 3, 4, 715
- spectrum despread
 - in demodulation, 491
 - as a linear operation, 492
- spectrum spreading
 - as a linear operation, 492
 - and phase modulation, 491
- speech coding
 - applications of, 229–230
 - design philosophy of, 230
 - at low bit rates, 229–230
 - techniques for, 551
- speech communication process, 4
- speech-production process, 4
- speech signal, 4
 - as bipolar, 6
 - limits of, 16
- sphere packing, 599–600
- split-phase signaling, 207
- splitter, 282
- spontaneous fluctuations, 58–61
- spreading code
 - with pseudo-random properties, 490
 - use of, 490
- spread spectrum
 - communications, 508
 - important attribute of, 488
 - notion of, 488–490
- spread-spectrum communication system
 - advantage of, 479
 - rejection of interference, 479
 - requirements of, 493
- spread-spectrum modulation
 - definition of, 479–480
 - for military applications, 480
 - principles of, 480
 - to provide multipath rejection, 480
- secure communications of, 479, 480
- signaling techniques known as, 479
- spread-spectrum techniques
 - as direct-sequence spread spectrum, 490
 - in passband transmission, 490
 - versus standard modulation techniques, 480
- square constellations, 369–370
- square law, 193
- SSB modulated signal, 99
- SSB modulation, 134
- standard modulation techniques, 480
- state diagram, 657–660, 659
- state probabilities, 681
- static picture, 4
- stationary process, 33–34
 - versus strictly stationary, 33
 - various names for, 36
- statistical average, 711–714
- statistical expectation operator, 711
- statistical regularity, 703
- step-size, 194
- step-size parameter, 225–226
- stereo multiplexing
 - in FM radio broadcasting, 124
 - as a form of frequency-division multiplexing, 124
- stochastic process, 32
- stop-and-wait automatic repeat request, 628
- stop-and-wait strategy, 628
- stream ciphers, 744–746
 - operation of, 744
 - used in, 745
- strictly stationary process, 35
- Strowger, A. B., 27
- Strowger switch, 27
- subframes, 552
- subnets, 13
- substitution cipher
 - description of, 749–750
 - use of, 749–750
- successive errors, 232
- sufficient statistics, 321
- sum-product algorithm, 691
- Sunde's FSK, 381, 385–386, 388
- Sunde's FSK signal, 386
- superhet, 128
- superheterodyne receiver, 27, 128–129
 - consists of, 128
 - differences between AM and FM, 129
- survivor paths, 662
- switching center, mobile, 530
- symbol, 2
- symbol energy-to-noise spectral density ratio, 502
- symbol error
 - average probability of, 276
 - conditional probability of, 333
- symbol error probability
 - versus bit error rate (BER), 335–336
 - to calculate, 357
 - definition of, 209, 310
 - evaluation of, 543
 - formula for, 256–257
 - to minimize, 310, 346
 - as a ratio, 358
 - simplification of, 335
- symbol rate, 501
- symbol shaping function
 - defined as, 353
 - energy spectral density of, 386, 395
- symbol synchronization, 448
- symbol timing, 455, 458, 463–464
- symbol timing recovery, 463–464, 465
- symmetric modem configurations, 421–425
- synchronization, 448–450, 493
 - algorithmic (modern) approach, 449
 - basic modes of, 448
 - classical approach to, 449
 - implementation of, 449
 - process of, 448
 - as a statistical parameter estimation problem, 449
 - in a TDM system, 212
 - of transmitter and receiver clocks, 212
- synchronization problem
 - approaches for solving, 449
 - solution to, 493
- synchronizing pulses, 5, 212
- synchronous demodulation
 - quadrature null effect, 96
- synchronous optical network (SONET), 15
- syndrome
 - calculation of, 646–648
 - importance of, 635
 - properties of, 635–636
- syndrome calculator, 647
- syndrome decoding, 635–636, 638, 639
- syndrome polynomial, 647
- synthesis equation, 442–443
- synthesis filter, 552
 - consists of, 552
 - as part of the encoder, 551
- systematic block code, 632

system capacity
 definition of, 549
 description of, 718
 to maximize, 549
 system-dependent scaling factor,
 132–133

system design objectives, 3

system-memory time, 718

T

TDMA. *See* time-division multiple access

telecommunications environment,
 3–7

telegraph, 26

telegraphic code, Baudot's, 26

telephone channel, 15–17, 287

telephone circuit frequencies, 3

telephone network, 27

as a communication network,
 10

first commercial service, 28

primary purpose of, 15

television network, 14

television picture, 5–6

television signals, 101–103

modulation format of, 101–102
 as wideband signals, 7

Telstar satellite, 29

temporal autocorrelation function,
 284

ternary code, 204

t-error-correcting RS code, 654

theorem of irrelevance, 321,
 321–322

theory of error-control codes, 485

theory of spectral analysis of
 random processes, 46

thermal noise, 32, 60

Thévenin equivalent circuit, 60

3-dB bandwidth, 721

three-level output, 267–268

threshold effect, 137–138

AM and FM, 164

clicks heard in, 149–150

definition of, 138, 149

in an envelope detector, 138

threshold extension, 154

threshold reduction, 152

time-averaged autocorrelation
 function, 42, 51

time-bandwidth product, 721–722

choice of, 398

as a design parameter, 397

time compression (TC)

multiplexing

scheme, 278–279

use of, 277–278

time diversity, 544–545

time-division multiple access

(TDMA), 211, 211–212, 513

efficient system, 516

as wireless communications

systems, 547–550

time-division multiplexing (TDM)

concept of, 211

defined as, 21, 105

use of, 211–212

time-domain description, 720

time-flat channel, 542

time-frequency mapping, 9

time response, 266

time-scaling property, 722

time slot, 513

time-to-frequency mapping
 network, 235

time-varying phasor, 728

time-varying transfer function, 536

timing error, 264

timing synchronization, 455

Toeplitz property, 225

toll connection, 16

Tomlinson-Harashima precoding,
 430

tone-modulation analysis, 119

tracking, 493

tracking filter, 154

trade-offs, 602

training mode, 290

transceiver, 278

transistor, 28

transition matrix, 582

transition probability, 582, 616

transmission bandwidth

defined as, 118

instantaneous spreading, 499

transmission delay, 260

transmission path, 212

transmission security, 490

transmit filter, 259

transmitted code vector, 660

transmitted FH/MFSK signal, 500

transmitted power

definition of, 3

primary communication

resource, 92

transmitted pulse amplitude, 253

transmitted pulse shape, 261

transmitted signal, 280–281

transmitted signal energy per bit,
 258

transmitted signal point, 322–323

transmitted TV signal, 102

transmitter

of an analog communication
 system, 88–89

by combining operations, 414

location of, 2

power limited, 597

purpose of, 2

use of pre-emphasis in, 154–155

transmitting antenna

function of, 518

mounting of, 17–18

as a point source, 519

and power density, 521

transorthogonal signals. *See*

simplex signals

transponder, 19, 514–515

transponder channel, 515

transposition cipher

description of, 750

use of, 750

transversal equalizer, 286

trapdoor one-way function, 755

traveling-wave tube amplifier,
 516

trellis, 657–660, 661–662

trellis-coded modulation, 668–669

trellis codes

for band-limited channels,
 668–669

design of, 669

trellis coding

as an error-control coding
 technique, 430

as a forward-error correction
 scheme, 424

trellis encoder, 425

turbo codes

consist of, 674

development of, 674

performance of, 676–677

properties of, 682, 693

termination approaches of, 676

turbo coding, 674

turbo decoder

basic structure of, 677–678

complexity of, 684

details of, 683

turbo decoding, 677–680, 682–683

turbo encoder, 683

- twisted-pair cable
 - versus coaxial cables, 17
 - consists of, 16
 - susceptible to, 16–17
 - uses of, 277
 - 2B1Q code, 281
 - amplitude levels of, 317
 - compared to other line codes, 281
 - desirable properties of, 281
 - as the North American standard, 281
 - two-dimensional matched filter, 378–379
 - two-dimensional optimum receiver, 378
 - two-key cryptography, 742
 - two-stage spectral analysis, 110
 - two-step subspace procedure, 556
 - two-way transmission, 106
- U**
- Ungerboeck, G., 28
 - Ungerboeck codes, 670
 - for 8-PSK, 670–672
 - asymptotic coding gain of, 673
 - unicity distance, 748
 - uniform quantizer, 196
 - union bound, 337
 - illustration of, 333
 - simplification of, 335
 - as a useful upper bound, 332–333
 - use of, 401
 - union of events, 333
 - unipolar nonreturn-to-zero (NRZ) signaling, 205
 - unipolar return-to-zero (RZ) signaling
 - disadvantages of, 207
 - feature of, 207
 - uniquely decodable, 574
 - unitary matrix, 443
 - unit delay, 219
 - unit-delay elements, 646
 - universal curve, 118–119
 - unmodulated carrier, 108
 - up conversion, 104–105
 - upconverter, 448
 - uplink, 19, 514–515
 - upper bound, 401
 - upstream data transmission, 281–282
- V**
- V.32 modemand nonredundant coding, 423–424
 - phase changes in, 422–423
 - switching to QPSK mode, 424
 - and trellis coding, 423–424, 425
 - V.32 modem standardand
 - alternative modulation schemes, 421
 - characteristics of, 421
 - V.90 modem, 431
 - van Duuren, H. C. A., 28
 - Van Vleck, J. H., 27
 - variable-length code, 574
 - variable nodes, 685
 - variance, 579
 - VDSL. *See* very-high-rate digital subscriber lines
 - vector product, 681
 - vector quantizer
 - advantage of, 615
 - encoding process in, 615
 - versus scalar quantizer, 615
 - signal-to-quantization noise ratio for, 615
 - vectors, 633
 - vector space, 554–555
 - Vernam cipher, 747
 - very-high-rate digital subscriber lines (VDSL)
 - advantages of, 446
 - use of, 446
 - very-large-scale integrated (VLSI) circuits
 - development of, 28
 - vestigial sideband (VSB) filter
 - frequency response of, 102
 - magnitude response of, 100–101
 - vestigial sideband (VSB) modulated wave
 - methods of generating, 100
 - quadrature component of, 102–103
 - vestigial sideband (VSB) modulation, 100–101, 163
 - vestigial sideband (VSB) shaping filter, 102
 - vestigial sideband modulation
 - definition of, 93
 - and its role in commercial TV broadcasting, 101–102
 - video bandwidth, 6
 - video-on-demand, 9, 282
 - video signal, 4–5, 6
- virtual communication, 13**
- Viterbi algorithm, 661–663, 670**
- difficulty in the application of, 662
 - as a maximum likelihood decoder, 662
 - as a maximum likelihood sequence estimator, 662
- VLSI. *See* very-large-scale integrated (VLSI) circuits**
- vocal tract, 4**
- voiceband modem, 420**
- versus digital subscriber lines, 446
 - operational environment of, 447
- voice communication, 22**
- voice effect, Donald Duck, 99–100**
- voice signals, 99**
- voice spectrum, 3**
- voltage-controlled oscillator (VCO), 152–153, 157–158**
- von Neumann, John, 28**
- VSB. *See* vestigial sideband**
- W**
- water-filling interpretation, 610–611**
- water-filling solution, 438**
- waveform, 21, 276**
- of important line codes, 204–207
 - in modulation, 490, 492
- waveform distortion, 102–103**
- wavelength-division multiplexing (WDM), 21**
- wave motion, 18**
- weak signal suppression, 142**
- weight vector, 554**
- white Gaussian noise, 62**
- identically distributed, 334
 - process, 392
- white noise, 61–63**
- autocorrelation function of, 61–62
 - characteristics of, 61–62
 - mathematical properties of, 62
 - power spectral density of, 61–63
- white noise process, 62**
- wideband communication channels, 218**
- wideband FM signal, 118**
- wideband frequency modulation, 113–115**
- wideband signal, 7**
- wideband transmitted signal, 490**

Wiener-Hopf equations, 224
wired communications, 559–560
wireless broadcast channels, 17–18
wireless communications, 529–535
 adaptive antenna arrays for,
 553
 features of, 512
 goal of, 553
 major channel impairments of,
 553
 and mobility, 512

 and OFDM, 448
 source coding for, 550–553
 as a type of multiuser
 communication, 512
 as a type of multiuser radio
 communication system,
 529–530
 versus wired communications,
 559–560
wireless communication system
 mobility of, 559

 practical requirements of, 396
 problems using MSK, 396
World Wide Web, 28–29

Z

zero-forcing equalizer, 283
zero-forcing kind, 556
zero-mean white Gaussian noise
 process, 310
zero state, 481
Zworykin, Vladimir K., 27

A new edition that takes students to the cutting edge and back!

Extensively revised and updated, this new fourth edition of **COMMUNICATION SYSTEMS** is the most complete undergraduate textbook on the theories and principles behind today's most advanced communications systems.

New features include:

- MATLAB computer experiments that demonstrate important aspects of communication theory
- Expanded coverage of emerging digital technologies, such as digital subscriber lines (DSL), carrierless amplitude modulation/phase modulation (CAP), and discrete multi-tone (DMT)
- Dozens of examples that relate theory to real-world communication systems

Superbly organized, the text skillfully guides students through topics ranging from pulse modulation to passband digital transmission, and from random processes to error-control coding. Throughout, Haykin presents difficult concepts in language that students can easily understand.

Wiley & Sons, Inc.
New York / Chichester / Weinheim
Munich / Singapore / Toronto
wiley.com/college

ISBN 0-471-17869-1

