The difference between the two SNR levels is 12 dB, which is a ratio of 16. Thus, the SNR for $L = 256$ is 16 times the SNR for $L = 64$. The former requires approximately 33% more bandwidth than the latter.

Comments on Logarithmic Units

Logarithmic units and logarithmic scales are very convenient when a variable has a large dynamic range. Such is the case with frequency variables or SNR. A logarithmic unit for the power ratio is the decibel (dB), defined as $10 \log_{10}$ (power ratio). Thus, an SNR is x dB, where

$$x = 10 \log_{10} \frac{S}{N}$$

We use the same unit to express power gain or loss over a certain transmission medium. For instance, if over a certain cable the signal power is attenuated by a factor of 15, the cable gain is

$$G = 10 \log_{10} \frac{1}{15} = -11.76 \text{ dB}$$

or the cable attenuation (loss) is 11.76 dB.

Although the decibel is a measure of power ratios, it is often used as a measure of power itself, as discussed in Chapter 2. For instance, "100 watt" may be considered to be a power ratio of 100 with respect to 1-watt power, and is expressed in units of dBW as

$$P_{\text{dBW}} = 10 \log_{10} 100 = 20 \text{ dBW}$$

Thus, 100-watt power is 20 dBW. Similarly, power measured with respect to 1 mW power is dBm. For instance, 100-watt power is

$$P_{\text{dBm}} = 10 \log \frac{100 \text{ W}}{1 \text{ mW}} = 50 \text{ dBm} = (\text{P}_{\text{dBW}} + 30) \text{ dBm}$$

5.3 DIGITAL TELEPHONY: PCM IN T1 SYSTEMS

A HISTORICAL NOTE

Lacking suitable switching devices, more than 20 years elapsed between the invention of PCM and its implementation. Vacuum tubes, used before the invention of the transistor, were not only bulky, but were also poor switches while dissipating a lot of heat. Systems using vacuum tube switches were large, rather unreliable, and tended to overheat. Everything changed with the invention of the transistor, which is a small and a nearly ideal switch that consumes little power.

Coincidentally, at about the time the transistor was invented, the demand for telephone service had become so high that the existing system was overloaded, particularly in large cities. It was not easy to install new underground cables by digging up streets and

causing many disruptions. An attempt was made on a limited scale to increase the capacity by frequency-division-multiplexing several voice channels through amplitude modulation. Unfortunately, the cables were primarily designed for the audio voice range (0–4 kHz) and suffered severely from noise. Furthermore, cross talk between pairs of channels bundled in the same cable was unacceptable at high frequencies. Ironically, PCM—requiring a bandwidth several times larger than that required for FDM signals—offered the solution. This is because digital systems with closely spaced regenerative repeaters can work satisfactorily on noisy lines despite poor high-frequency performance.[9] The repeaters, spaced approximately 6000 feet apart, clean up the signal and regenerate new pulses before the pulses get too distorted and noisy. This is the history of the Bell System's T1 carrier system.[3, 10] A wired link that used to transmit one audio signal of bandwidth 4 kHz was successfully upgraded to transmit 24 time-division-multiplexed PCM telephone signals with a total bandwidth of 1.544 MHz.

T1 Time Division Multiplexing

A schematic of a T1 carrier system is shown in Fig. 5.21a. All 24 channels are sampled in a sequence. The sampler output represents a time-division-multiplexed PAM signal. The multiplexed PAM signal is now applied to the input of an encoder that quantizes and encodes each sample into eight binary pulses*— a binary codeword (see Fig. 5.21b). The signal, now converted to digital form, is sent over the transmission medium. Regenerative repeaters detect the pulses and regenerate new pulses. At the receiver, the decoder converts the binary pulses back to samples (by decoding). The samples are then demultiplexed (i.e., distributed to each of the 24 channels). The desired audio signal is reconstructed in each channel.

The circular commutators in Fig. 5.21 are not mechanical but are high-speed electronic switching circuits. Several schemes are available for this purpose.[11] Sampling is done by electronic gates (such as a bridge diode circuit, as shown in Fig. 4.5a) opened periodically by narrow pulses of 2 μs duration. The 1.544 Mbit/s signal of the T1 system, called **digital signal level 1 (DS1)**, can be used further to multiplex into progressively higher level signals DS2, DS3, and DS4, as described next, in Sec. 5.4.

After the Bell System introduced the T1 carrier system in the United States, dozens of variations were proposed or adopted elsewhere before the ITU-T standardized its 30-channel PCM interface known as E1 carrier with a rate of 2.048 Mbit/s (in contrast to T1, with 24 channels and 1.544 Mbit/s). Because of the widespread adoption of the T1 carrier system in the United States and parts of Asia, both standards continue to be used in different parts of the world, with appropriate interfaces in international connections.

Synchronizing and Signaling

Binary codewords corresponding to samples of each of the 24 channels are multiplexed in a sequence, as shown in Fig. 5.22. A segment containing one codeword (corresponding to one sample) from each of the 24 channels is called a **frame**. Each frame has $24 \times 8 = 192$ information bits. Because the sampling rate is 8000 samples per second, each frame occupies 125 μs. To parse the information bits correctly at the receiver, it is necessary to be sure where each frame begins. Therefore, a **framing bit** is added at the beginning of each frame. This makes a total of 193 bits per frame. Framing bits are chosen so that a sequence of framing bits, one at the beginning of each frame, forms a special pattern that is unlikely to be formed in the underlying speech signal.

To synchronize at the receiver, the sequence formed by the first bit from each frame is examined by the logic of the receiving terminal. If this sequence does not follow the given

* In an earlier version, each sample was encoded by seven bits. An additional bit was later added for signaling.

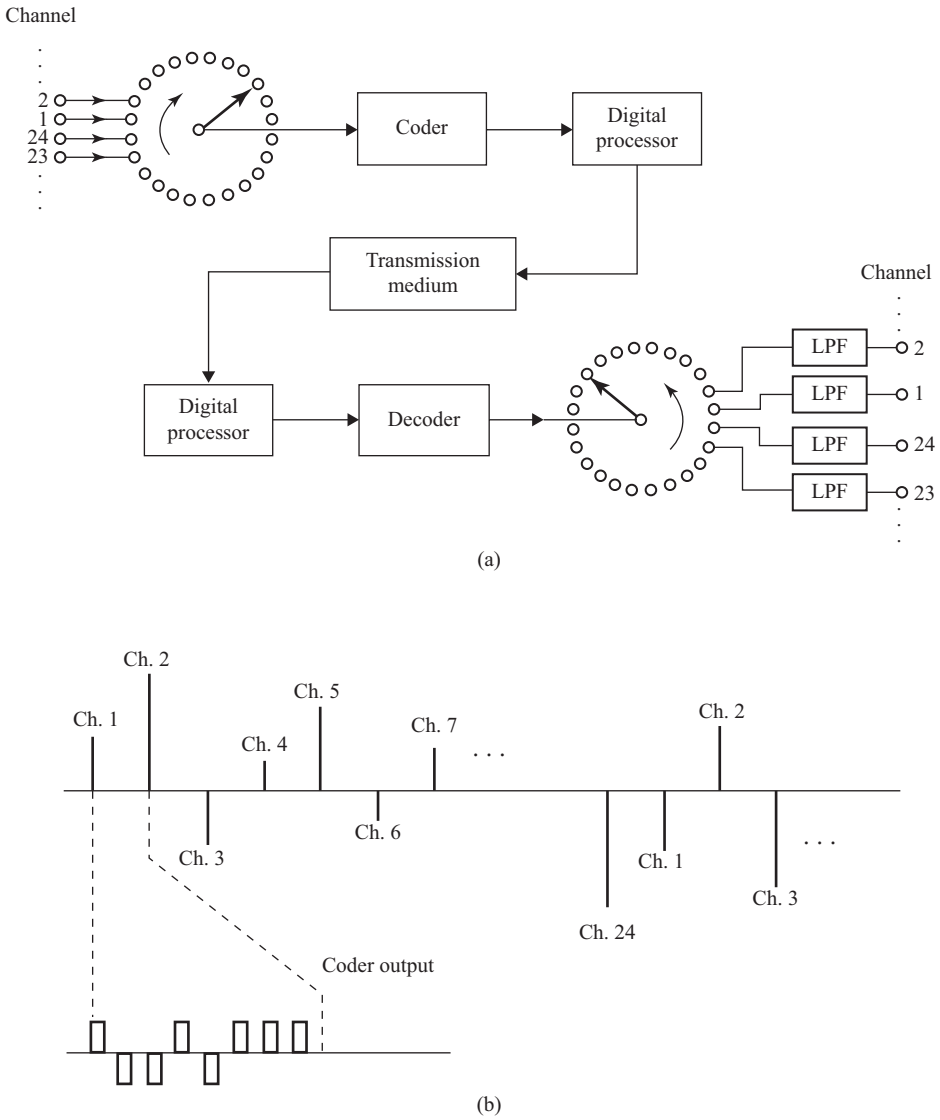

Figure 5.21 T1 carrier system.

code pattern (framing bit pattern), a synchronization loss is detected, and the next position is examined to determine whether it is actually the framing bit.

In addition to information and framing bits, we need to transmit signaling bits corresponding to dialing pulses, as well as telephone on-hook/off-hook signals. When channels developed by this system are used to transmit signals between telephone switching systems, the switches must be able to communicate with each other to use the channels effectively. Since all eight bits are now used for transmission instead of the seven bits used in the earlier version,* the signaling channel provided by the eighth bit is no longer available. Since only a rather low-speed signaling channel is required, rather than create extra time

* In the earlier version of T1, quantizing levels $L = 128$ required only seven information bits. The eighth bit was used for signaling.

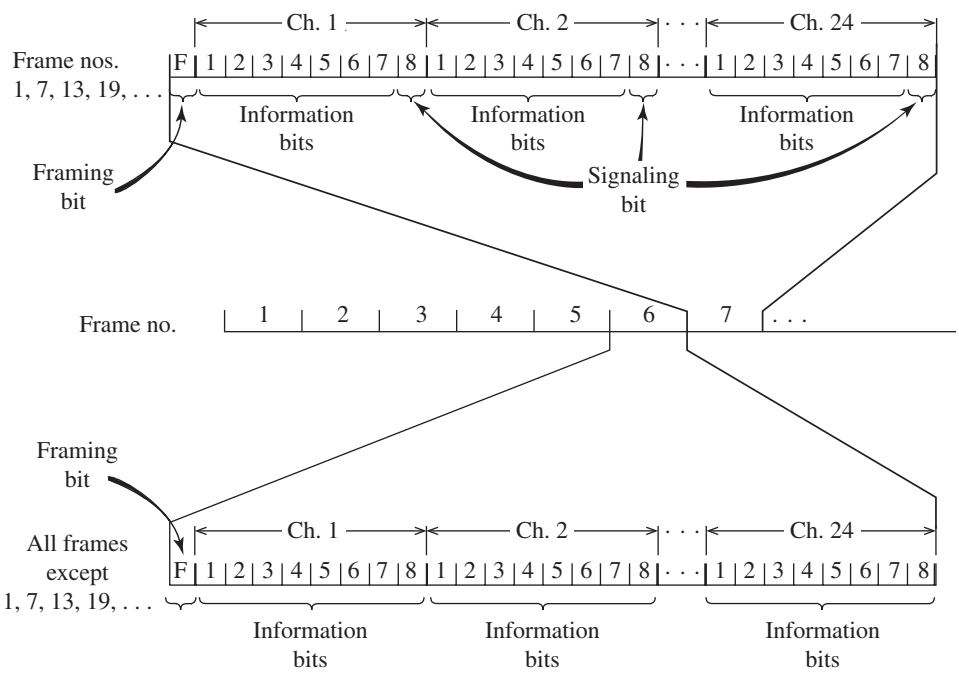

Figure 5.22
T1 system signaling format.

slots for this information, we rob one information bit (the least significant bit) of every sixth sample of a signal to transmit this information. This means that every sixth sample of each voice signal will have a potential error corresponding to the least significant digit. Every sixth frame, therefore, has $7 \times 24 = 168$ information bits, 24 signaling bits, and 1 framing bit. In all the remaining frames, there are 192 information bits and 1 framing bit. This technique is called $\mathbf{7\frac{5}{6}}$ bit encoding, and the signaling channel so derived is called **robbed-bit signaling**. The slight SNR degradation suffered by impairing one out of six frames is considered to be an acceptable penalty. The signaling bits for each signal occur at a rate of $8000/6 = 1333$ bit/s. The frame format is shown in Fig. 5.22.

The older seven-bit framing format required only that frame boundaries be identified so that the channels could be located in the bit stream. When signaling is superimposed on the channels in every sixth frame, it is necessary to identify, at the receiver, which frames are the signaling frames. A new framing structure, called the **superframe**, was developed to take care of this. The framing bits are transmitted at 8 kbit/s as before and occupy the first bit of each frame. The framing bits form a special pattern, which repeats in 12 frames: **100011011100**. The pattern thus allows the identification of frame boundaries as before, but also allows the determination of the locations of the sixth and twelfth frames within the superframe. Note that the superframe described here is 12 frames in length. Since two bits per superframe are available for signaling for each channel, it is possible to provide four-state signaling for a channel by using the four possible patterns of the two signaling bits: **00, 01, 10,** and **11**. Although most switch-to-switch applications in the telephone network require only two-state signaling, three- and four-state signaling techniques are used in certain special applications.

Advances in digital electronics and in coding theory have made it unnecessary to use the full 8 kbit/s of the framing channel in a DS1 signal to perform the framing task. A new

superframe structure, called the **extended superframe (ESF)** format, was introduced during the 1970s to take advantage of the reduced framing bandwidth requirement. An ESF is 24 frames in length and carries signaling bits in the eighth bit of each channel in frames 6, 12, 18, and 24. Sixteen-state signaling is thus possible and is sometimes used although, as with the superframe format, most applications require only two-state signaling.

The 8 kbit/s overhead (framing) capacity of the ESF signal is divided into three channels: 2 kbit/s for framing, 2 kbit/s for a cyclic redundancy check (CRC-6) error detection channel, and 4 kbit/s for a data channel. The highly reliable error checking provided by the CRC-6 pattern and the use of the data channel to transport information on signal performance as received by the distant terminal make ESF much more attractive to service providers than the older superframe format. More discussions on CRC error detection can be found in Chapter 13.

5.4 DIGITAL MULTIPLEXING HIERARCHY

Several low-bit-rate signals can be multiplexed, or combined, to form one high-bit-rate signal, to be transmitted over a high-frequency medium. Because the medium is time-shared by various incoming signals, this is a case of TDM. The signals from various incoming channels, or tributaries, may be as diverse as a digitized voice signal (PCM), a computer output, telemetry data, and a digital facsimile. The bit rates of various tributaries need not be identical.

To begin with, consider the case of all tributaries with identical bit rates. Multiplexing can be done on a bit-by-bit basis (known as bit or **digit interleaving**) as shown in Fig. 5.23a, or on a word-by-word basis (known as byte or **word interleaving**). Figure 5.23b shows the interleaving of words, formed by four bits. The North American digital hierarchy uses bit interleaving (except at the lowest level), where bits are taken one at a time from the various signals to be multiplexed. Byte interleaving, used in building the DS1 signal and SONET-formatted signals, involves inserting bytes in succession from the channels to be multiplexed.

The T1 carrier, discussed in Sec. 5.3, uses eight-bit word interleaving. When the bit rates of incoming channels are not identical, the high-bit-rate channel is allocated proportionately more slots. Four-channel multiplexing consists of three channels, B, C, and D of identical bit rate R, and one channel (channel A) with a bit rate of $3R$ (Figs. 5.23c and d). Similar results can be attained by combining words of different lengths. It is evident that the minimum length of the multiplex frame must be a multiple of the lowest common multiple of the incoming channel bit rates, and, hence, this type of scheme is practical only when some fairly simple relationship exists among these rates. The case of completely asynchronous channels is discussed later in Sec 5.4.2.

At the receiving terminal, the incoming digit stream must be divided and distributed to the appropriate output channel. For this purpose, the receiving terminal must be able to correctly identify each bit. This requires the receiving system to uniquely synchronize in time with the beginning of each frame, with each slot in a frame, and with each bit within a slot. This is accomplished by adding framing and synchronization bits to the data bits. These bits are part of the so-called **overhead bits.**

5.4.1 Signal Format

Figure 5.24 illustrates a typical format, that of the DM1/2 multiplexer. We have here bit-by-bit interleaving of four channels each at a rate of 1.544 Mbit/s. The main frame (multiframe)

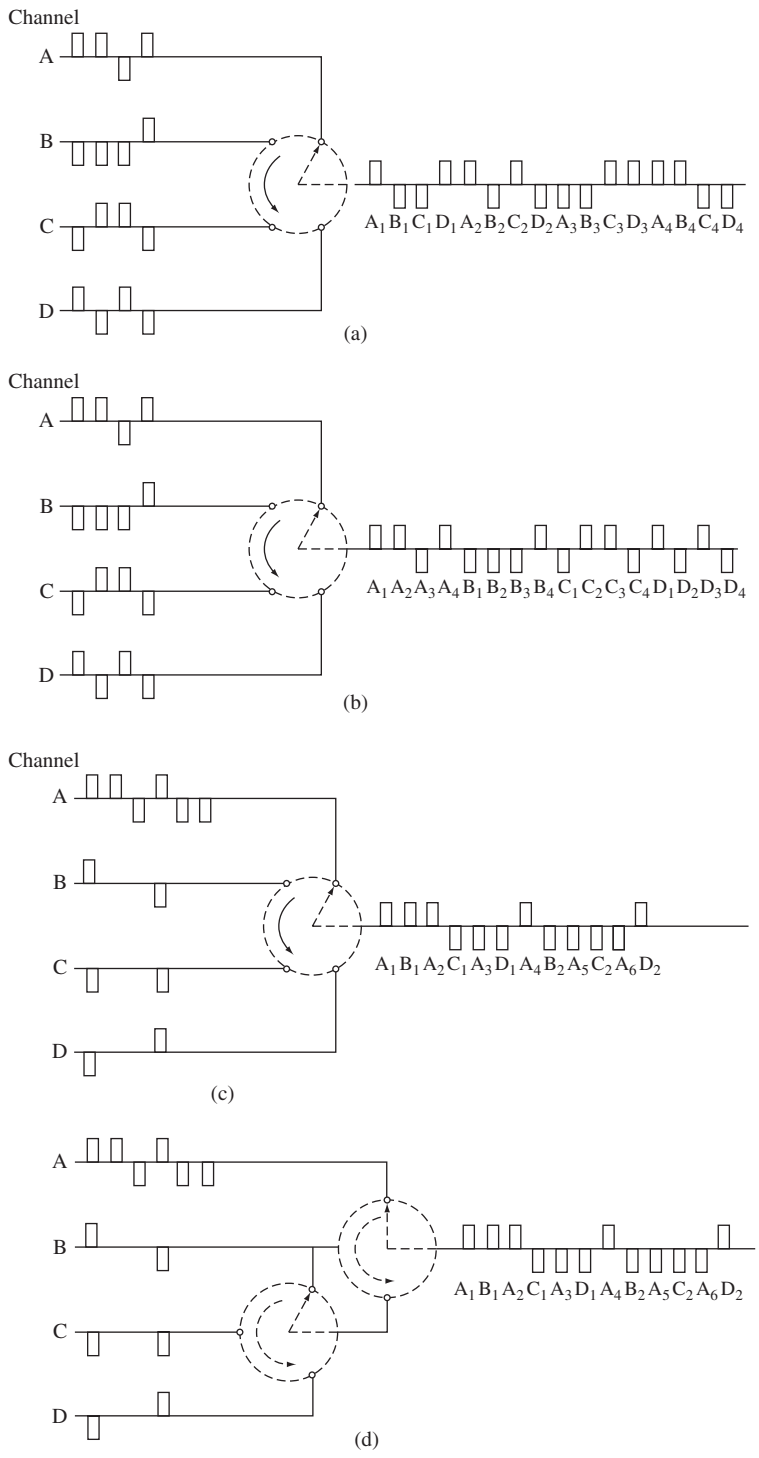

Figure 5.23
Time division multiplexing of digital signals: (a) digit interleaving; (b) word (or byte) interleaving; (c) interleaving channel having different bit rate; (d) alternate scheme for (c).

Figure 5.24 DM1/2 multiplexer format.

M_0	[48]	C_A	[48]	F_0	[48]	C_A	[48]	C_A	[48]	F_1	[48]
M_1	[48]	C_B	[48]	F_0	[48]	C_B	[48]	C_B	[48]	F_1	[48]
M_1	[48]	C_C	[48]	F_0	[48]	C_C	[48]	C_C	[48]	F_1	[48]
M_1	[48]	C_D	[48]	F_0	[48]	C_D	[48]	C_D	[48]	F_1	[48]

consists of four subframes. Each subframe has six overhead bits: for example the subframe 1 (first line in Fig. 5.24) has overhead bits M_0, C_A, F_0, C_A, C_A, and F_1. In between these overhead bits are 48 interleaved data bits from the four channels (12 data bits from each channel). We begin with overhead bit M_0, followed by 48 multiplexed data bits, then add a second overhead bit C_A followed by the next 48 multiplexed bits, and so on. Thus, there are a total of $48 \times 6 \times 4 = 1152$ data bits and $6 \times 4 = 24$ overhead bits making a total 1176 bits/frame. The efficiency is $1152/1176 \simeq 98\%$. The overhead bits with subscript 0 are always **0** and those with subscript 1 are always **1**. Thus, M_0, F_0 are all **0**s and M_1 and F_1 are all **1**s. The F digits are periodic **010101** ... and provide the main framing pattern, which the multiplexer uses to synchronize on the frame. After locking onto this pattern, the demultiplexer searches for the **0111** pattern formed by overhead bits $M_0M_1M_1M_1$. This further identifies the four subframes, each corresponding to a line in Fig. 5.24. It is possible, although unlikely, that message bits will also have a natural pattern **101010. . . .** The receiver could lock onto this wrong sequence. The presence of $M_0M_1M_1M_1$ provides verification of the genuine $F_0F_1F_0F_1$ sequence. The C bits are used to transmit additional information about bit stuffing, as discussed later in Sec 5.4.2.

In the majority of cases, not all incoming channels are active all the time: some transmit data, and some are idle. This means the system is underutilized. We can, therefore, admit more input channels to take advantage of the inactivity, at any given time, of at least one channel. This obviously involves much more complicated switching operations, and also rather careful system planning. In any random traffic situation we cannot guarantee that the number of transmission channels demanded will not exceed the number available; but by taking account of the statistics of the signal sources, it is possible to ensure an acceptably low probability of this occurring. Multiplex structures of this type have been developed for satellite systems and are known as **time division multiple-access (TDMA) systems**.

In TDMA systems employed for telephony, the design parameters are chosen so that any overload condition lasts only a fraction of a second, which leads to acceptable performance for speech communication. For other types of data and telegraphy, modest transmission delays are unimportant. Hence, in overload condition, the incoming data can be stored and transmitted later.

5.4.2 Asynchronous Channels and Bit Stuffing

In the preceding discussion, we assumed synchronization between all the incoming channels and the multiplexer. This is difficult even when all the channels are nominally at the same rate. For example, consider a 1000 km coaxial cable carrying 2×10^8 pulses per second. Assuming the nominal propagation speed in the cable to be 2×10^8 m/s, it takes 1/200 second of transit time and 1 million pulses will be in transit. If the cable temperature increases by 1°F, the propagation velocity will increase by about 0.01%. This will cause the pulses in transit to

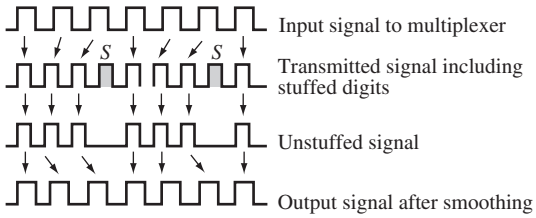

Figure 5.25
Pulse stuffing.

arrive sooner, thus producing a temporary increase in the rate of pulses received. Because the extra pulses cannot be accommodated in the multiplexer, they must be temporarily stored at the receiver. If the cable temperature drops, the rate of received pulses will drop, and the multiplexer will have vacant slots with no data. These slots need to be stuffed with dummy digits (**pulse stuffing**).

DS1 signals in the North American network are often generated by crystal oscillators in individual digital terminal equipment. Although the oscillators are quite stable, they will not oscillate at exactly the same frequency, leading to another cause of asynchronicity in the network.

This shows that even in synchronously multiplexed systems, the data are rarely received at a synchronous rate. We always need a storage (known as an **elastic store**) and pulse stuffing (also known as **justification**) to accommodate such a situation. Obviously, elastic store and pulse stuffing will work even when the channels are asynchronous.

Three variants of the pulse stuffing scheme exist: (1) positive pulse stuffing, (2) negative pulse stuffing, and (3) positive/negative pulse stuffing. In positive pulse stuffing, the multiplexer rate is higher than that required to accommodate all incoming tributaries at their maximum rate. Hence, the time slots in the multiplexed signal will become available at a rate exceeding that of the incoming data so that the tributary data will tend to lag (Fig. 5.25). At some stage, the system will decide that this lag has become great enough to require pulse stuffing. The information about the stuffed-pulse positions is transmitted through overhead bits. From the overhead bits, the receiver knows the stuffed-pulse position and eliminates that pulse.

Negative pulse stuffing is a complement of positive pulse stuffing. The time slots in the multiplexed signal now appear at a slightly slower rate than those of the tributaries, and thus the multiplexed signal cannot accommodate all the tributary pulses. Information about any dropped pulse and its position is transmitted through overhead bits. The positive/negative pulse stuffing is a combination of the first two schemes. The nominal rate of the multiplexer is equal to the nominal rate required to accommodate all incoming channels. Hence, we may need positive pulse stuffing at some times and negative stuffing at others. All this information is sent through overhead bits.

The C digits in Fig. 5.24 are used to convey stuffing information. Only one stuffed bit per input channel is allowed per frame. This is sufficient to accommodate expected variations in the input signal rate. The bits C_A convey information about stuffing in channel A, bits C_B convey information about stuffing in channel B, and so on. The insertion of any stuffed pulse in any one subframe is denoted by setting all the three Cs in that line to **1**. No stuffing is indicated by using **0**s for all the three Cs. If a bit has been stuffed, the stuffed bit is the first information bit associated with the immediate channel following the F_1 bit, that is, the first such bit in the last 48-bit sequence in that subframe. For the first subframe, the stuffed bit will immediately follow the F_1 bit. For the second subframe, the stuffed bit will be the second bit following the F_1 bit, and so on.

5.4.3 Plesiochronous (almost Synchronous) Digital Hierarchy

We now present the digital hierarchy developed by the Bell System and currently included in the ANSI standards for telecommunications (Fig. 5.26). The North American hierarchy is implemented in North America and Japan.

Two major classes of multiplexers are used in practice. The first category is used for combining low-data-rate channels. It multiplexes channels of rates of up to 9600 bit/s into a signal of data rate of up to 64 kbit/s. The multiplexed signal, called **"digital signal level 0" (DS0)** in the North American hierarchy, is eventually transmitted over a voice-grade channel. The second class of multiplexers is at a much higher bit rate.

Fig. 5.26 shows four orders, or levels, of multiplexing. The first level is the **T1 multiplexer** or **channel bank**, consisting of 24 channels of 64 kbit/s each. The output of this multiplexer is a **DS1 (digital level 1)** signal at a rate of 1.544 Mbit/s. Four DS1 signals are multiplexed by a DM1/2 multiplexer to yield a DS2 signal at a rate 6.312 Mbit/s. Seven DS2 signals are multiplexed by a DM2/3 multiplexer to yield a DS3 signal at a rate of 44.736 Mbit/s. Finally, three DS3 signals are multiplexed by a DM3/4NA multiplexer to yield a DS4NA signal at a rate 139.264 Mbit/s.

The inputs to a T1 multiplexer need not be restricted only to digitized voice channels alone. Any digital signal of 64 kbit/s of appropriate format can be transmitted. The case of the higher levels is similar. For example, all the incoming channels of the DM1/2 multiplexer need not be DS1 signals obtained by multiplexing 24 channels of 64 kbit/s each. Some of them may be 1.544 Mbit/s digital source signals of appropriate format, and so on.

In Europe and many other parts of the world, another hierarchy, recommended by the ITU as a standard, has been adopted. This hierarchy, based on multiplexing 30 telephone channels of 64 kbit/s (E-0 channels) into an E-1 carrier at 2.048 Mbit/s (30 channels) is shown in

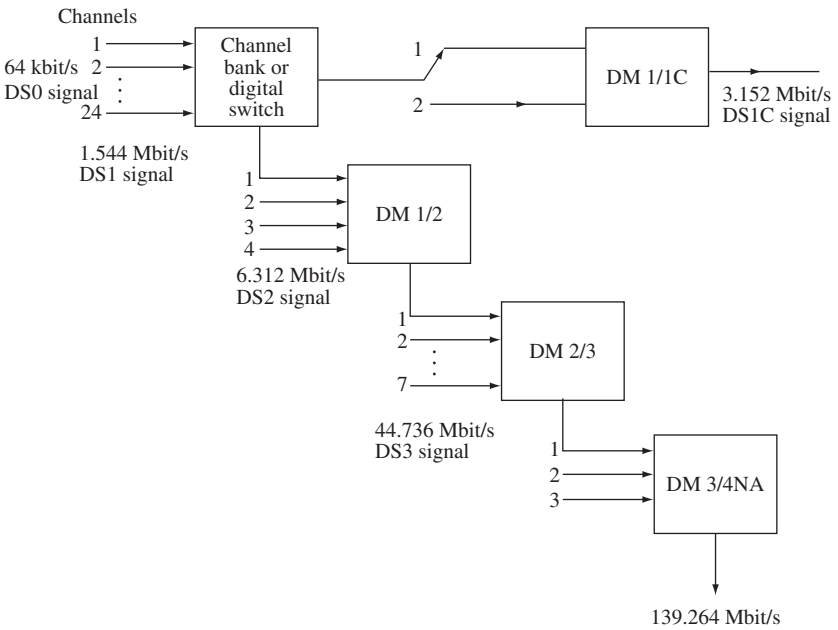

Figure 5.26 North American digital hierarchy (AT&T system).

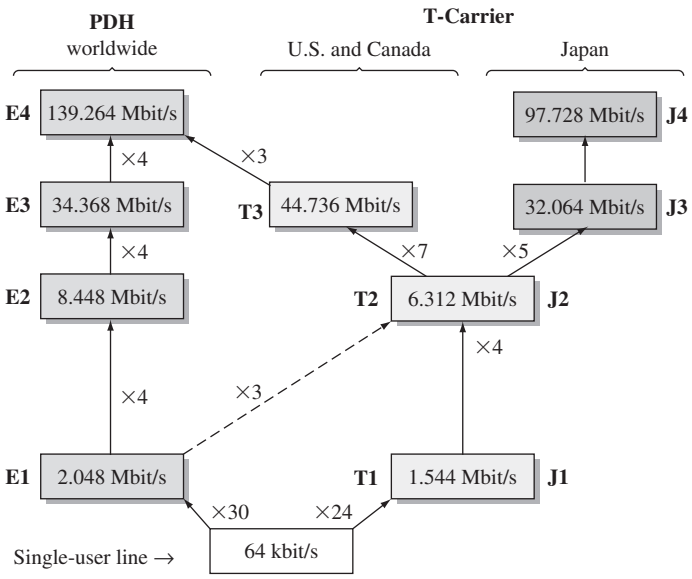

Figure 5.27 Plesiochronous digital hierarchy (PDH) according to ITU-T Recommendation G.704.

Fig. 5.27. Starting from the base level of E-1, four lower level lines form one higher level line progressively, generating an E-2 line with data throughput of 8.448 Mbit/s, an E-3 line with data throughput of 34.368 Mbit/s, an E-4 line with data throughput of 139.264 Mbit/s, and an E-5 line with data throughput of 565.148 Mbit/s. Because different networks must be able to interface with one another across the three different systems (North American, Japanese, and other) in the world, Fig. 5.27 demonstrates the relative relationship and the points of their common interface.

5.5 DIFFERENTIAL PULSE CODE MODULATION (DPCM)

PCM is not very efficient because it generates many bits that require too much bandwidth to transmit. Many different ideas have been proposed to improve the encoding efficiency of A/D conversion. In general, these ideas exploit the characteristics of the source signals. DPCM is one such scheme.

Often in analog messages we can make a good guess about a sample value from knowledge of the signal's past sample values. In other words, the sample values are not independent, and generally there is a great deal of redundancy in the signal samples. Proper exploitation of this redundancy allows signal encoding with fewer bits. Consider a simple scheme; instead of transmitting the sample values, we transmit the difference between the successive sample values. Thus, if $m[k]$ is the kth sample, instead of transmitting $m[k]$, we transmit the difference $d[k] = m[k] - m[k-1]$. At the receiver, knowing $d[k]$ and the previous sample value $m[k-1]$, we can reconstruct $m[k]$. Thus, from knowledge of the difference $d[k]$, we can reconstruct $m[k]$ iteratively at the receiver. Now, the difference between successive samples is generally much smaller than the sample values. Thus, the peak amplitude m_p of the transmitted values is reduced considerably. Because the quantization interval $\Delta v = m_p/L$, for a given L (or n), this reduces the quantization interval Δv used for $d[k]$, thereby reducing the quantization

6 PRINCIPLES OF DIGITAL DATA TRANSMISSION

Throughout much of the twentieth century, most communication systems were in analog form. However, by the end of the 1990s, digital transmission began to dominate most applications. One does not need to search hard to witness the wave of upgrade from analog to digital communications: from audio-cassette tape to MP3 and CD, from NTSC analog TV to digital HDTV, from traditional telephone to VoIP, and from videotape to DVD and Blu-ray. In fact, even the last holdout of broadcast radio is facing a strong digital competitor in the form of satellite radio, podcast, and HD radio. Given the dominating presence of digital communication systems in our lives today, it is never too early to study the basic principles and various aspects of digital data transmission, as we will do in this chapter.

This chapter deals with the problems of transmitting digital data over a channel. Hence, the starting messages are assumed to be digital. We shall begin by considering the binary case, where the data consist of only two symbols: **1** and **0**. We assign a distinct waveform (pulse) to each of these two symbols. The resulting sequence of data-bearing pulses is transmitted over a channel. At the receiver, these pulses are detected and converted back to binary data (**1**s and **0**s).

6.1 DIGITAL COMMUNICATION SYSTEMS

A digital communication system consists of several components, as shown in Fig. 6.1. In this section, we conceptually outline their functionalities in the communication systems. The details of their analysis and design will be given in dedicated sections later in this chapter.

6.1.1 Source

The input to a digital system is a sequence of digits. The input could be the output from a data set, a computer, or a digitized audio signal (PCM, DM, or LPC), digital facsimile or HDTV, or telemetry data, and so on. Although most of the discussion in this chapter is confined to the binary case (communication schemes using only two symbols), the more general case of M-ary communication, which uses M symbols, will also be discussed in Sec. 6.7 and Sec. 6.9.

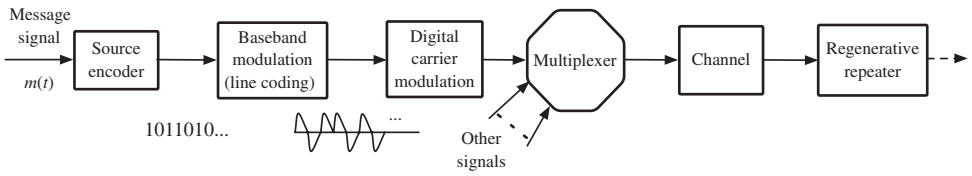

Figure 6.1 Fundamental building blocks of digital communication systems.

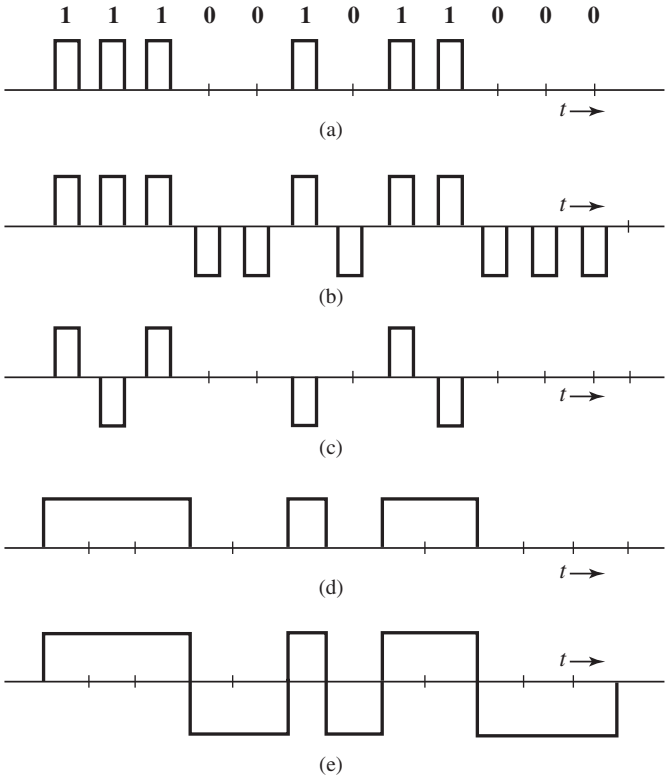

Figure 6.2 Line code examples: (a) on-off (RZ); (b) polar (RZ); (c) bipolar (RZ); (d) on-off (NRZ); (e) polar (NRZ).

6.1.2 Line Codes

The digital output of a source encoder is converted (or coded) into electric pulses (waveforms) for the purpose of transmission over the channel. This process is called **line coding** or **transmission coding**. There are many possible ways of assigning waveforms (pulses) to represent digital data. In the binary case (2 symbols), for example, conceptually the simplest line code is **on-off**, where a **1** is transmitted by a pulse $p(t)$ and a **0** is transmitted by no pulse (zero signal) as shown in Fig. 6.2a. Another commonly used code is **polar**, where **1** is transmitted by a pulse $p(t)$ and **0** is transmitted by a pulse $-p(t)$ (Fig. 6.2b). The polar line code is the most power-efficient code because it requires the least power for a given noise immunity (error probability). Another popular line code in PCM is **bipolar**, also known as **pseudoternary** or **alternate mark inversion (AMI)**, where **0** is encoded by no pulse and **1** is encoded alternately by $p(t)$ or $-p(t)$ depending on whether the previous **1** is encoded by $-p(t)$ or $p(t)$. In short, pulses representing consecutive **1**s alternate in sign, as shown in Fig. 6.2c.

This code has the advantage that if **one single** pulse error is made, the received pulse sequence will violate the bipolar rule and such error can be detected (although not corrected).*

Another line code that appeared promising earlier is the duobinary (and modified duobinary) proposed by Lender.[1, 2] This code is better than the bipolar in terms of bandwidth efficiency. Its more prominent variant, the **modified duobinary** line code, has seen applications in hard disk drive channels, in optical 10 Gbit/s transmission for metro-networks, and in the first-generation modems for integrated services digital networks (ISDN). Details of duobinary line codes will be discussed later in Sec. 6.3.

In our discussion so far, we have used half-width pulses just for the sake of illustration. We can also select other widths. Full-width pulses are often used in some applications. Whenever full-width pulses are used, the pulse amplitude is held to a constant value throughout the pulse interval (i.e., it does not have a chance to go to zero before the next pulse begins). For this reason, these schemes are called **non-return-to-zero** or **NRZ** schemes, in contrast to **return-to-zero** or **RZ** schemes (Fig. 6.2a–c). Figure 6.2d shows an on-off NRZ signal, whereas Fig. 6.2e shows a polar NRZ signal.

6.1.3 Multiplexer

Generally speaking, the capacity of a physical channel (e.g., coaxial cable, optic fiber) for transmitting data is much larger than the data rate of individual sources. To utilize this capacity effectively, a digital multiplexer can combine several sources into one signal of higher rate. The digital multiplexing can be achieved through frequency division or time division, as we have already discussed. Alternatively, code division is also a practical and effective approach (to be discussed in Chapter 10). In general, a true physical channel is often shared by several messages simultaneously.

6.1.4 Regenerative Repeater

Regenerative repeaters are used at regularly spaced intervals along a digital transmission path to detect the incoming digital signal and regenerate new "clean" pulses for further transmission down the line. This process periodically eliminates, and thereby combats, accumulation of noise and signal distortion along the transmission path. The ability of such regenerative repeaters to effectively eliminate noise and signal distortion effects is one of the biggest advantages of digital communication systems over their analog counterparts.

If the pulses are transmitted at a rate of R_b pulses per second, we require the periodic timing information—the clock signal at R_b Hz—to sample and detect the incoming pulses at a repeater. This timing information can be extracted from the received signal itself if the line code is chosen properly. When the *RZ* polar signal in Fig. 6.2b is rectified, for example, it results in a periodic signal of clock frequency R_b Hz, which contains the desired periodic timing signal of frequency R_b Hz. When this signal is applied to a resonant circuit tuned to frequency R_b, the circuit output is a sinusoid of frequency R_b Hz and can be used for timing. The on-off signal can be expressed as a sum of a periodic signal (of clock frequency) and a polar, or random, signal as shown in Fig. 6.3. Because of the presence of the periodic component, we can extract the timing information from this signal by using a resonant circuit

* This assumes no more than one error in sequence. Multiple errors in sequence could negate their respective effects and remain undetected. However, the probability of multiple errors is much smaller than that of single errors. Even for single errors, we cannot tell exactly where the error is located. Therefore, this code can detect the presence of single errors, but it cannot correct them.

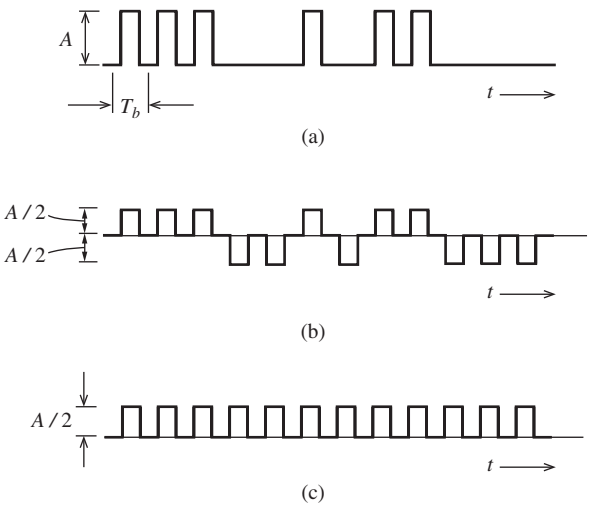

Figure 6.3 An on-off signal (a) is a sum of a random polar signal (b) and a clock frequency periodic signal (c).

tuned to the clock frequency. A bipolar signal, when rectified, becomes an on-off signal. Hence, its timing information can be extracted using the same way as that for an on-off signal.

The timing signal (the resonant circuit output) can be sensitive to the incoming bit pattern. In the on-off or bipolar case, a **0** is transmitted by "no pulse." Hence, if there are too many **0**s in a sequence (no pulses), there is no signal at the input of the resonant circuit and the sinusoidal output of the resonant circuit starts decaying, thus causing error in the timing information. A line code in which the transmitted bit pattern does not affect the accuracy of the timing information is said to be a **transparent** line code. The *RZ* polar scheme (where each bit is transmitted by some pulse) is transparent, whereas the on-off and bipolar are nontransparent because long strings of 0's would provide no timing information. We shall discuss later ways (e.g., scrambling) of overcoming this problem.

6.2 BASEBAND LINE CODING

Digital data can be transmitted by various **transmission** or **line codes**. We have given examples of on-off, polar, and bipolar. Each line code has its advantages and disadvantages. Among other desirable properties, a line code should have the following properties.

- *Low bandwidth.* Transmission bandwidth should be as small as possible.
- *Power efficiency*. For a given bandwidth and a specified detection error rate, the transmission power should be as low as possible.
- *Error detection and correction capability*. It is desirable to detect and preferably correct the detected errors. In a bipolar case, for example, a single error will cause bipolar violation and can easily be detected. Error-correcting codes will be covered later in Chapter 13.
- *Favorable power spectral density*. It is desirable to have zero power spectral density (PSD) at $f = 0$ (dc or direct current) because alternating current (ac) coupling and transformers are

often used at the regenerative repeaters.* Significant power in low-frequency components should also be avoided because it causes dc wander in the pulse stream when ac coupling is used.

- *Adequate timing content*. It should be possible to extract timing or clock information from the signal.
- *Transparency*. It should be possible to correctly receive a digital signal regardless of the pattern of **1**s and **0**s. We saw earlier that a long string of **0**s could cause problems in timing extraction for the on-off and bipolar cases. A code is transparent if the data are so coded that for every possible sequence of data, the coded signal is received faithfully.

6.2.1 PSD of Various Baseband Line Codes

In Example 3.23, we discussed a procedure for finding the PSD of a polar pulse train. We shall use a similar procedure to find a general expression for PSD of the baseband modulation (line coding) output signals as shown in Fig. 6.2. In particular, we directly apply the relationship between the PSD and the autocorrelation function of the baseband modulation signal given in Section 3.8 [Eq. (3.85)].

In the following discussion, we consider a generic pulse $p(t)$ whose corresponding Fourier transform is $P(f)$. We can denote the line code symbol at time k as a_k. When the transmission rate is $R_b = 1/T_b$ pulses per second, the line code generates a pulse train constructed from the basic pulse $p(t)$ with amplitude a_k starting at time $t = kT_b$; in other words, the kth symbol is transmitted as $a_k p(t - kT_b)$. Figure 6.4a provides an illustration of a special pulse $p(t)$, whereas Fig. 6.4b shows the corresponding pulse train generated by the line coder at baseband. As shown in Fig. 6.4b, counting a succession of symbol transmissions T_b seconds apart, the baseband signal is a pulse train of the form

$$y(t) = \sum a_k p(t - kT_b) \tag{6.1}$$

Note that the line coder determines the symbol $\{a_k\}$ as the amplitude of the pulse $p(t - kT_b)$.

The values a_k are random and depend on the line coder input and the line code itself; $y(t)$ is a PAM signal. The on-off, polar, and bipolar line codes are all special cases of this pulse train $y(t)$, where a_k takes on values 0, 1, or -1 randomly, subject to some constraints. We can, therefore, analyze many line codes according to the PSD of $y(t)$. Unfortunately, the PSD of $y(t)$ depends on both a_k and $p(t)$. If the pulse shape $p(t)$ changes, we do not want to derive the PSD all over again. This difficulty can be overcome by the simple artifice of selecting an ideal PAM signal $x(t)$ that uses a unit impulse for the basic pulse $p(t)$ (Fig. 6.4c). The impulses occur at the intervals of T_b and the strength (area) of the kth impulse is a_k. If $x(t)$ is applied to the input of a filter that has a unit impulse response $h(t) = p(t)$ (Fig. 6.4d), the output will be the pulse train $y(t)$ in Fig. 6.4b. Also, applying Eq. (3.91), the PSD of $y(t)$ is

$$S_y(f) = |P(f)|^2 S_x(f)$$

This relationship allows us to determine $S_y(f)$, the PSD of a line code corresponding to any pulse shape $p(t)$, once we know $S_x(f)$ which only depends on the line code $\{a_k\}$. This approach is attractive because of its generality.

* The ac coupling is required because the dc paths provided by the cable pairs between the repeater sites are used to transmit the power needed to operate the repeaters.

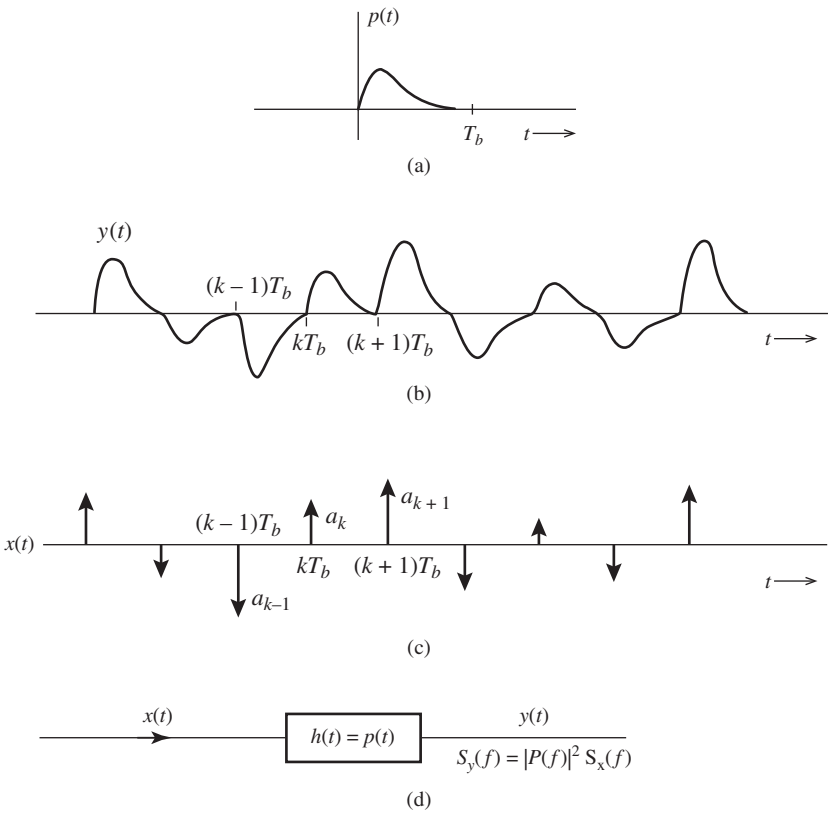

Figure 6.4 Random pulse-amplitude-modulated signal and its generation from a PAM impulse.

We now need to derive $\mathcal{R}_x(\tau)$ to determine $S_x(f)$, the time autocorrelation function of the impulse train $x(t)$. This can be conveniently done by considering the impulses as a limiting form of the rectangular pulses, as shown in Fig. 6.5a. Each pulse has a width $\epsilon \to 0$, and the kth pulse height

$$h_k = \frac{a_k}{\epsilon} \to \infty$$

This way, we guarantee that the strength of the kth impulse is a_k, that is, $\epsilon h_k = a_k$. If we designate the corresponding rectangular pulse train as $\hat{x}(t)$, then by definition [Eq. (3.82) in Section 3.8.2]

$$\mathcal{R}_{\hat{x}}(\tau) = \lim_{T\to\infty} \frac{1}{T} \int_{-T/2}^{T/2} \hat{x}(t)\hat{x}(t-\tau)\, dt \tag{6.2}$$

Because $\mathcal{R}_{\hat{x}}(\tau)$ is an even function of τ [Eq. (3.83)], we need to consider only positive τ. To begin with, consider the case of $\tau < \epsilon$. In this case, the integral in Eq. (6.2) is the area under the signal $\hat{x}(t)$ multiplied by $\hat{x}(t)$ delayed by $\tau (\tau < \epsilon)$. As seen from Fig. 6.5b, the area

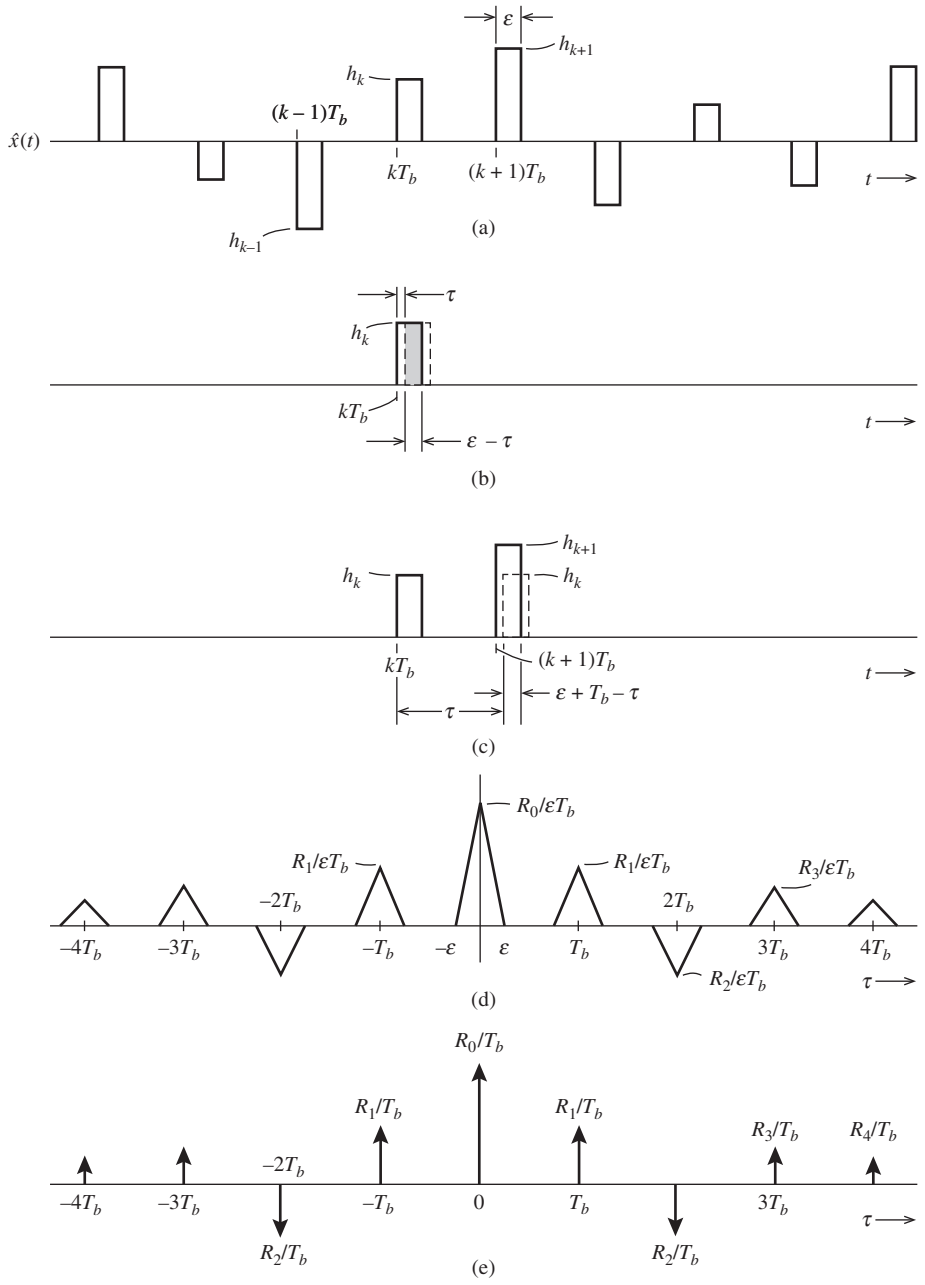

Figure 6.5 Derivation of PSD of a random PAM signal with a very narrow pulse of width ϵ and height $h_k = a_k/\epsilon$.

associated with the kth pulse is $h_k^2 \cdot (\epsilon - \tau)$, and

$$\begin{aligned}\mathcal{R}_{\hat{x}} &= \lim_{T\to\infty} \frac{1}{T} \sum_k h_k^2 \cdot (\epsilon - \tau) \\ &= \lim_{T\to\infty} \frac{1}{T} \sum_k a_k^2 \left(\frac{\epsilon - \tau}{\epsilon^2}\right) \\ &= \frac{R_0}{\epsilon T_b}\left(1 - \frac{\tau}{\epsilon}\right) \end{aligned} \tag{6.3a}$$

where

$$R_0 = \lim_{T\to\infty} \frac{T_b}{T} \sum_k a_k^2 \tag{6.3b}$$

During the averaging interval T ($T \to \infty$), there are N pulses ($N \to \infty$), where

$$N = \frac{T}{T_b} \tag{6.4}$$

and from Eq. (6.3b)

$$R_0 = \lim_{N\to\infty} \frac{1}{N} \sum_k a_k^2 \tag{6.5}$$

Observe that the summation is over N pulses. Hence, R_0 is the time average of the square of the pulse amplitudes a_k. Using our time average notation, we can express R_0 as

$$R_0 = \lim_{N\to\infty} \frac{1}{N} \sum_k a_k^2 = \overbrace{a_k^2} \tag{6.6}$$

We also know that $\mathcal{R}_{\hat{x}}(\tau)$ is an even function of τ [see Eq. (3.83)]. Hence, Eq. (6.3) can be expressed as

$$\mathcal{R}_{\hat{x}}(\tau) = \frac{R_0}{\epsilon T_b}\left(1 - \frac{|\tau|}{\epsilon}\right) \qquad |\tau| < \epsilon \tag{6.7}$$

This is a triangular pulse of height $R_0/\epsilon T_b$ and width 2ϵ centered at $\tau = 0$ (Fig. 6.5d). As expected, if τ increases beyond ϵ, there is no overlap between the delayed signal $\hat{x}(t-\tau)$ and $\hat{x}(t)$; hence, $\mathcal{R}_{\hat{x}}(\tau) = 0$, as seen from Fig. 6.5d. However, when τ grows further, we find that the kth pulse of $\hat{x}(t-\tau)$ will start overlapping the $(k+1)$th pulse of $\hat{x}(t)$ as τ approaches T_b (Fig. 6.5c). Repeating the earlier argument, we see that $\mathcal{R}_{\hat{x}}(\tau)$ will have another triangular pulse of width 2ϵ centered at $\tau = T_b$ and of height $R_1/\epsilon T_b$ where

$$\begin{aligned} R_1 &= \lim_{T\to\infty} \frac{T_b}{T} \sum_k a_k a_{k+1} \\ &= \lim_{N\to\infty} \frac{1}{N} \sum_k a_k a_{k+1} \\ &= \overbrace{a_k a_{k+1}} \end{aligned}$$

Observe that R_1 is obtained by multiplying every pulse strength (a_k) by the strength of its immediate neighbor (a_{k+1}), adding all these products before dividing by the total number of pulses. This is clearly the time average (mean) of the product $a_k a_{k+1}$ and is, in our notation, $\overbrace{a_k a_{k+1}}$. A similar phenomenon happens around $\tau = 2T_b, 3T_b, \ldots$. Hence, $\mathcal{R}_{\hat{x}}(\tau)$ consists of a sequence of triangular pulses of width 2ϵ centered at $\tau = 0, \pm T_b, \pm 2T_b, \ldots$. The height of

the triangular pulses centered at $\pm nT_b$ is $R_n/\epsilon T_b$, where

$$\begin{aligned} R_n &= \lim_{T\to\infty} \frac{T_b}{T} \sum_k a_k a_{k+n} \\ &= \lim_{N\to\infty} \frac{1}{N} \sum_k a_k a_{k+n} \\ &= \overbrace{a_k a_{k+n}}, \qquad n = 0, \pm 1, \pm 2, \cdots \end{aligned}$$

R_n is essentially the discrete autocorrelation function of the line code symbols $\{a_k\}$.

To find $\mathcal{R}_x(\tau)$, we let $\epsilon \to 0$ in $\mathcal{R}_{\hat{x}}(\tau)$. As $\epsilon \to 0$, the width of each triangular pulse$\to 0$ and the height$\to \infty$ in such a way that the area is still finite. Thus, in the limit as $\epsilon \to 0$, the triangular pulses converge to impulses. For the nth pulse centered at nT_b, the height is $R_n/\epsilon T_b$ and the area is R_n/T_b. Hence, (Fig. 6.5e)

$$\mathcal{R}_x(\tau) = \frac{1}{T_b} \sum_{n=-\infty}^{\infty} R_n \delta(\tau - nT_b) \tag{6.8}$$

The PSD $S_x(f)$ is the Fourier transform of $\mathcal{R}_x(\tau)$. Therefore,

$$S_x(f) = \frac{1}{T_b} \sum_{n=-\infty}^{\infty} R_n e^{-jn2\pi fT_b} \tag{6.9}$$

Recognizing that $R_{-n} = R_n$ [because $\mathcal{R}(\tau)$ is an even function of τ], we have

$$S_x(f) = \frac{1}{T_b} \left[R_o + 2 \sum_{n=1}^{\infty} R_n \cos n2\pi fT_b \right] \tag{6.10}$$

The input $x(t)$ to the filter with impulse response $h(t) = p(t)$ results in the output $y(t)$, as shown in Fig. 6.4d. If $p(t) \Longleftrightarrow P(f)$, the transfer function of the filter is $H(f) = P(f)$, and according to Eq. (3.91)

$$S_y(f) = |P(f)|^2 S_x(f) \tag{6.11a}$$

$$= \frac{|P(f)|^2}{T_b} \left[\sum_{n=-\infty}^{\infty} R_n e^{-jn2\pi fT_b} \right] \tag{6.11b}$$

$$= \frac{|P(f)|^2}{T_b} \left[R_o + 2 \sum_{n=1}^{\infty} R_n \cos n2\pi fT_b \right] \tag{6.11c}$$

Thus, the PSD of a line code is fully characterized by its R_n and the pulse-shaping selection $P(f)$. We shall now use this general result to find the PSDs of various line codes at baseband by first determining the symbol autocorrelation R_n.

6.2.2 Polar Signaling

In polar signaling, **1** is transmitted by a pulse $p(t)$ and **0** is represented by $-p(t)$. In this case, a_k is equally likely to be 1 or -1, and a_k^2 is always 1. Hence,

$$R_0 = \lim_{N\to\infty} \frac{1}{N} \sum_k a_k^2$$

There are N pulses and $a_k^2 = 1$ for each one, and the summation on the right-hand side of R_0 in the preceding equation is N. Hence,

$$R_0 = \lim_{N\to\infty} \frac{1}{N}(N) = 1 \tag{6.12a}$$

Moreover, both a_k and a_{k+1} are either 1 or -1. Hence, $a_k a_{k+1}$ is either 1 or -1. Because the pulse amplitude a_k is equally likely to be 1 and -1 on the average, out of N terms the product $a_k a_{k+1}$ is equal to 1 for $N/2$ terms and is equal to -1 for the remaining $N/2$ terms. Therefore,

Possible Values of $a_k a_{k+1}$

a_{k+1} \ a_k	-1	$+1$
-1	1	-1
$+1$	-1	1

$$R_1 = \lim_{N\to\infty} \frac{1}{N}\left[\frac{N}{2}(1) + \frac{N}{2}(-1)\right] = 0 \tag{6.12b}$$

Arguing this way, we see that the product $a_k a_{k+n}$ is also equally likely to be 1 or -1. Hence,

$$R_n = 0 \qquad n \geq 1 \tag{6.12c}$$

Therefore from Eq. (6.11c)

$$\begin{aligned} S_y(f) &= \frac{|P(f)|^2}{T_b} R_0 \\ &= \frac{|P(f)|^2}{T_b} \end{aligned} \tag{6.13}$$

For the sake of comparison of various schemes, we shall consider a specific pulse shape. Let $p(t)$ be a rectangular pulse of width $T_b/2$ (half-width rectangular pulse), that is,

$$p(t) = \Pi\left(\frac{t}{T_b/2}\right) = \Pi\left(\frac{2t}{T_b}\right)$$

and

$$P(f) = \frac{T_b}{2}\ \text{sinc}\left(\frac{\pi f T_b}{2}\right) \tag{6.14}$$

Therefore

$$S_y(f) = \frac{T_b}{4}\ \text{sinc}^2\left(\frac{\pi f T_b}{2}\right) \tag{6.15}$$

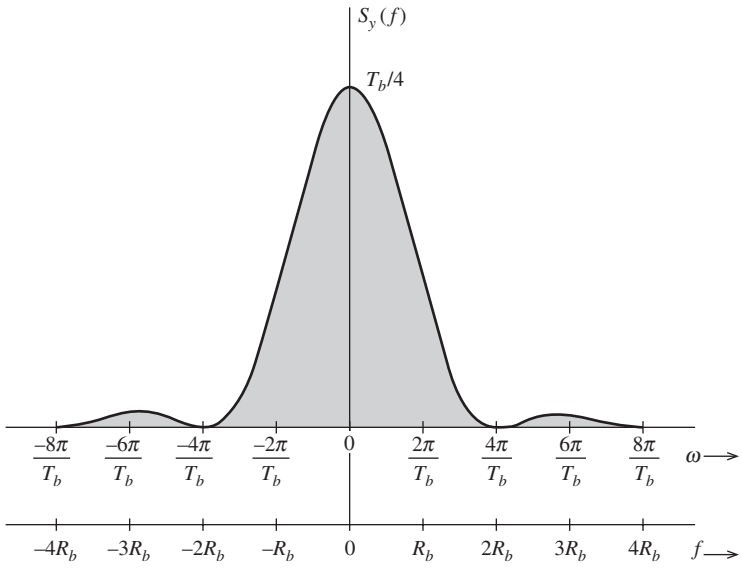

Figure 6.6 Power spectral density of a polar signal.

Figure 6.6 shows the spectrum $S_y(f)$. It is clear that the polar signal has most of its power concentrated in lower frequencies. Theoretically, the spectrum becomes very small as frequency increases but never becomes totally zero above a certain frequency. To define a meaningful measure of bandwidth, we consider its **first non-dc null frequency** to be its **effective bandwidth**.*

From the PSD, the effective bandwidth of this signal is seen to be $2R_b$ Hz (where R_b is the clock frequency). This is 4 times the theoretical minimum bandwidth required to transmit R_b pulses per second. Increasing the pulse width would reduce the bandwidth since expansion in the time domain results in compression in the frequency domain. For a full-width rectangular pulse† (maximum possible pulse width), the bandwidth marked by the first null is halved to R_b Hz, still twice the theoretical minimum. Thus, polar NRZ signaling is not the most bandwidth efficient.

Second, polar signaling has no capability for error detection or error correction. A third disadvantage of polar signaling is that it has nonzero PSD at dc ($f = 0$). This will pose a challenge to the use of ac coupling during transmission. The mode of ac coupling is very important in practice as it permits transformers and blocking capacitors to aid in impedance matching and bias removal. In Sec. 6.2.3, we shall show how a PSD of a line code may be forced to zero at dc by properly shaping $p(t)$.

On the positive side, polar signaling is the most efficient scheme from the power requirement viewpoint. For a given power, it can be shown that the error-detection probability for a polar scheme is the lowest among all signaling techniques (see Chapter 9). RZ polar signaling is also transparent because there is always some pulse (positive or negative)

* Strictly speaking, the location of the first null frequency above dc is not always a good measure of signal bandwidth. Whether the first non-dc null is a meaningful bandwidth depends on the amount of signal power contained in the main (or first) lobe of the PSD, as we will see later in the PSD comparison of several lines codes (Fig. 6.9). In most practical cases, this approximation is acceptable for commonly used line codes and pulse shapes.

† Scheme using the full-width pulse $p(t) = \Pi(t/T_b)$ is an example of an NRZ scheme. The half-width pulse scheme, on the other hand, is an example of an RZ scheme.

regardless of the bit sequence. Rectification of this specific RZ polar signal yields a periodic signal of clock frequency and can readily be used to extract timing.

6.2.3 Constructing a DC Null in PSD by Pulse Shaping

Because $S_y(f)$, and the PSD of a line code contains a multiplicative factor $|P(f)|^2$, we can force the PSD to have a dc null by selecting a pulse $p(t)$ such that $P(f)$ is zero at dc ($f = 0$). From

$$P(f) = \int_{-\infty}^{\infty} p(t)e^{-j2\pi ft}\,dt$$

we have

$$P(0) = \int_{-\infty}^{\infty} p(t)\,dt$$

Hence, if the area under $p(t)$ is made zero, $P(0)$ is zero, and we have a dc null in the PSD. For a finite width pulse, one possible shape of $p(t)$ to accomplish this is shown in Fig. 6.7a. When we use this pulse with polar line coding, the resulting signal is known as **Manchester code**, or **split-phase** (also called **twinned-binary**), signal. The reader can follow Eq. (6.13) to show that for this pulse, the PSD of the Manchester line code has a dc null (see Prob. 6.2-1).

6.2.4 On-Off Signaling

In on-off signaling, a **1** is transmitted by a pulse $p(t)$ and a **0** is transmitted by no pulse. Hence, a pulse strength a_k is equally likely to be 1 or 0. Out of N pulses in the interval of T seconds, a_k is 1 for $N/2$ pulses and is 0 for the remaining $N/2$ pulses on the average. Hence,

$$R_0 = \lim_{N\to\infty} \frac{1}{N}\left[\frac{N}{2}(1)^2 + \frac{N}{2}(0)^2\right] = \frac{1}{2} \tag{6.16}$$

To compute R_n, we need to consider the product $a_k a_{k+n}$. Note that a_k and a_{k+n} are equally likely to be 1 or 0. Therefore, on the average, the product $a_k a_{k+n}$ is equal to 1 for $N/4$ terms

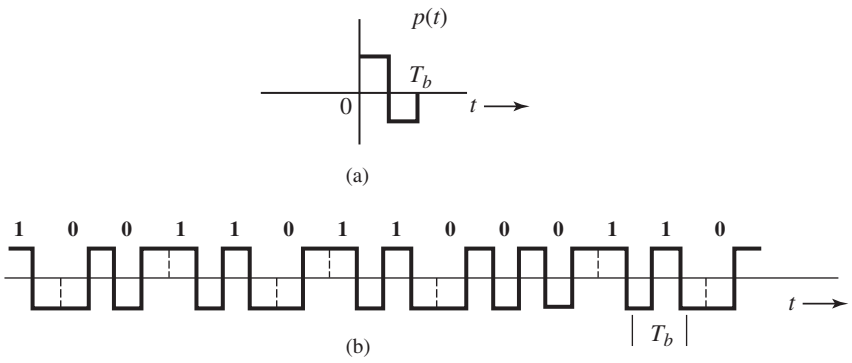

Figure 6.7 Split-phase (Manchester or twinned-binary) signal. (a) Basic pulse $p(t)$ for Manchester signaling. (b) Transmitted waveform for binary data sequence using Manchester signaling.

and 0 for $3N/4$ terms

Possible Values of $a_k a_{k+1}$

a_{k+n} \ a_k	0	1
0	0	0
1	0	1

$$R_n = \lim_{N\to\infty} \frac{1}{N}\left[\frac{N}{4}(1) + \frac{3N}{4}(0)\right] = \frac{1}{4} \qquad n \geq 1 \tag{6.17}$$

Consequently, from Eq. (6.9) we find

$$S_x(f) = \frac{1}{2T_b} + \frac{1}{4T_b}\sum_{\substack{n=-\infty \\ n\neq 0}}^{\infty} e^{-jn2\pi fT_b} \tag{6.18a}$$

$$= \frac{1}{4T_b} + \frac{1}{4T_b}\sum_{n=-\infty}^{\infty} e^{-jn2\pi fT_b} \tag{6.18b}$$

Equation (6.18b) is obtained from Eq. (6.18a) by splitting the term $1/2T_b$ corresponding to R_0 into two: $1/4T_b$ outside the summation and $1/4T_b$ inside the summation (corresponding to $n = 0$). We now use the formula (the proof is left as an exercise in Prob. 6.2-2b)

$$\sum_{n=-\infty}^{\infty} e^{-jn2\pi fT_b} = \frac{1}{T_b}\sum_{n=-\infty}^{\infty} \delta\left(f - \frac{n}{T_b}\right)$$

Substitution of this result in Eq. (6.18b) yields

$$S_x(f) = \frac{1}{4T_b} + \frac{1}{4T_b^2}\sum_{n=-\infty}^{\infty} \delta\left(f - \frac{n}{T_b}\right) \tag{6.19a}$$

and the desired PSD of the on-off waveform $y(t)$ is [from Eq. (6.11a)]

$$S_y(f) = \frac{|P(f)|^2}{4T_b}\left[1 + \frac{1}{T_b}\sum_{n=-\infty}^{\infty} \delta\left(f - \frac{n}{T_b}\right)\right] \tag{6.19b}$$

Note that unlike the continuous PSD spectrum of polar signaling, the on-off PSD of Eq. (6.19b) also has an additional discrete part

$$\frac{|P(f)|^2}{4T_b^2}\sum_{n=-\infty}^{\infty} \delta\left(f - \frac{n}{T_b}\right) = \frac{1}{4T_b^2}\sum_{n=-\infty}^{\infty} \left|P\left(\frac{n}{T_b}\right)\right|^2 \delta\left(f - \frac{n}{T_b}\right) \tag{6.19c}$$

This discrete part may be nullified if the pulse shape is chosen such that

$$P\left(\frac{n}{T_b}\right) = 0 \qquad n = 0, \pm 1, \ldots$$

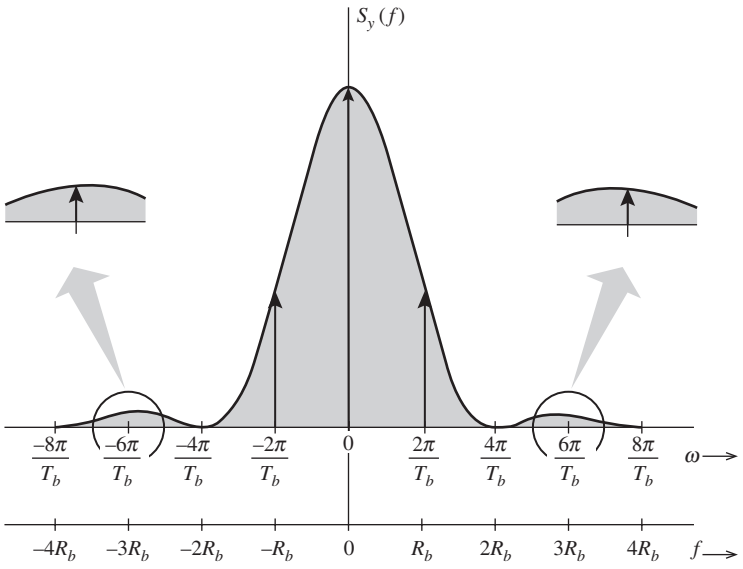

Figure 6.8
PSD of an on-off signal.

For the example case of a half-width rectangular pulse [see Eq. (6.14)],

$$S_y(f) = \frac{T_b}{16}\,\text{sinc}^2\left(\frac{\pi f T_b}{2}\right)\left[1 + \frac{1}{T_b}\sum_{n=-\infty}^{\infty}\delta\left(f - \frac{n}{T_b}\right)\right] \tag{6.20}$$

The resulting PSD is shown in Fig. 6.8. The continuous component of the spectrum is $(T_b/16)\ \text{sinc}^2\,(\pi f T_b/2)$. This is identical (except for a scaling factor) to the spectrum of the polar signal [Eq. (6.15)]. Each discrete component is an impulse in frequency domain at $f = n/T_b$ scaled by $(1/16)\ \text{sinc}^2\,(n\pi/2)$. Hence, the impulses repeat at the frequencies 0, $\pm 1/T_b$, $\pm 3/T_b$, $\cdots$ This is a logical result because as Fig. 6.3 shows, an on-off signal can be expressed as a sum of a polar and a periodic rectangular component in time. The polar component is exactly half the polar signal discussed earlier. Hence, the PSD of this component is one-fourth the PSD in Eq. (6.15). The periodic rectangular component is of clock frequency R_b; it consists of discrete components at the dc and the fundamental frequency R_b, plus its odd harmonics.

On-off signaling has very little to brag about. For a given transmitted power, it is less immune to noise and interference than the polar scheme, which uses a positive pulse for **1** and a negative pulse for **0**. This is because the noise immunity depends on the difference of amplitudes representing **1** and **0**. Hence, for the same immunity, if on-off signaling uses pulses of amplitudes 2 and 0, polar signaling only needs to use pulses of amplitudes 1 and -1. It is simple to show that on-off signaling requires twice as much power as polar signaling. If a pulse of amplitude 1 or -1 has energy E, then the pulse of amplitude 2 has energy $(2)^2E = 4E$. Because $1/T_b$ digits are transmitted per second, polar signal power is E/T_b. For the on-off case, on the other hand, each pulse energy is $4E$, though on the average such a pulse is transmitted over half of the time while nothing is transmitted over the other half. Hence, the average signal power of on-off is

$$\frac{1}{T_b}\left(4E\frac{1}{2} + 0\cdot\frac{1}{2}\right) = \frac{2E}{T_b}$$

which is twice of what is required for the polar signal. Moreover, unlike polar, on-off signaling is not transparent. A long string of **0**s (or offs) causes the absence of a signal and can lead to errors in timing extraction. In addition, all the disadvantages of polar signaling [e.g., excessive transmission bandwidth, nonzero power spectrum at dc, no error detection (or correction) capability] are also shared by on-off signaling.

6.2.5 Bipolar Signaling

The signaling scheme used in PCM for telephone networks is called bipolar (pseudoternary or alternate mark inverted). A **0** is transmitted by no pulse, and a **1** is transmitted by alternating between $p(t)$ and $-p(t)$, depending on whether the previous **1** uses a $-p(t)$ or $p(t)$. With consecutive pulses alternating, we can obtain a dc null in the PSD. Bipolar signaling actually uses three symbols [$p(t)$, 0, and $-p(t)$], and, hence, it is in reality ternary rather than binary signaling.

To calculate the PSD, recall that

$$R_o = \lim_{N\to\infty} \frac{1}{N} \sum_k a_k^2$$

On the average, half of the a_k's are 0, and the remaining half are either 1 or -1, with $a_k^2 = 1$. Therefore,

$$R_o = \lim_{N\to\infty} \frac{1}{N} \left[\frac{N}{2}(\pm 1)^2 + \frac{N}{2}(0)^2 \right] = \frac{1}{2}$$

To compute R_1, we consider the pulse strength product $a_k a_{k+1}$. There are four equally likely sequences of two bits: **11, 10, 01, 00**. Since bit **0** is encoded by no pulse ($a_k = 0$), the product $a_k a_{k+1}$ is zero for the last three of these sequences. This means, on the average, that $3N/4$ combinations have $a_k a_{k+1} = 0$ and only $N/4$ combinations have nonzero $a_k a_{k+1}$. Because of the bipolar rule, the bit sequence **11** can be encoded only by two consecutive pulses of opposite polarities. This means the product $a_k a_{k+1} = -1$ for the $N/4$ combinations. Therefore

Possible Values of $a_k a_{k+1}$

a_{k+1} \ a_k	0	1
0	0	0
1	0	-1

$$R_1 = \lim_{N\to\infty} \frac{1}{N} \left[\frac{N}{4}(-1) + \frac{3N}{4}(0) \right] = -\frac{1}{4}$$

To compute R_2 in a similar way, we need to observe the product $a_k a_{k+2}$. For this, we need to consider all possible combinations of three bits in sequence here. There are eight equally likely combinations: **111, 101, 110, 100, 011, 010, 001, 000**. The last six combinations have either the first and/or the last bit being **0**. Hence $a_k a_{k+2} = 0$ for these six combinations. From the bipolar rule, the first and the third pulses in the combination **111** are of the same polarity, yielding $a_k a_{k+2} = 1$. But for **101**, the first and the third pulse are of opposite polarity, yielding

$a_k a_{k+2} = -1$. Thus, on the average, $a_k a_{k+2} = 1$ for $N/8$ terms, -1 for $N/8$ terms, and 0 for $3N/4$ terms. Hence,

Possible Values of $a_k a_{k+1} a_{k+2}$

a_{k+2} \ a_k a_{k+1}	0 0	0 1	1 0	1 1
0	0	0	0	0
1	0	0	−1	1

$$R_2 = \lim_{N\to\infty} \frac{1}{N}\left[\frac{N}{8}(1) + \frac{N}{8}(-1) + \frac{3N}{8}(0)\right] = 0$$

In general, for $n > 2$, the product $a_k a_{k+n}$ can be 1, -1, or 0. Moreover, an equal number of combinations have values 1 and -1. This causes $R_n = 0$, that is,

$$R_n = \lim_{N\to\infty} \frac{1}{N} \sum_k a_k a_{k+n} = 0 \qquad n > 1$$

and [see Eq. (6.11c)]

$$S_y(f) = \frac{|P(f)|^2}{2T_b}\left[1 - \cos 2\pi f T_b\right] \tag{6.21a}$$

$$= \frac{|P(f)|^2}{T_b} \sin^2(\pi f T_b) \tag{6.21b}$$

Note that $S_y(f) = 0$ for $f = 0$ (dc), regardless of $P(f)$. Hence, the PSD has a dc null, which is desirable for ac coupling. Moreover, $\sin^2(\pi f T_b) = 0$ at $f = 1/T_b$, that is, at $f = 1/T_b = R_b$ Hz. Thus, regardless of $P(f)$, we are assured of the first non-dc null bandwidth R_b Hz. The bipolar PSD for the half-width pulse is

$$S_y(f) = \frac{T_b}{4} \operatorname{sinc}^2\left(\frac{\pi f T_b}{2}\right) \sin^2(\pi f T_b) \tag{6.22}$$

This is shown in Fig. 6.9. The effective bandwidth of the signal is R_b ($R_b = 1/T_b$), which is half that of polar using the same half-width pulse or on-off signaling and twice the theoretical minimum bandwidth. Observe that we were able to obtain the same bandwidth R_b for polar (or on-off) case using full-width pulse. For the bipolar case, however, the effective bandwidth defined by the first non-dc null frequency is R_b Hz regardless of whether the pulse is half-width or full-width.

Bipolar signaling has several advantages: (1) its spectrum is amenable to ac coupling; (2) its bandwidth is not excessive; (3) it has single-error-detection capability. This is because a single transmission error within a bit sequence will cause a violation of the alternating pulse rule, and this will be immediately detected. If a bipolar signal is rectified, we get an on-off signal that has a discrete component at the clock frequency. Among the disadvantages of a bipolar signal is the requirement for twice as much power (3 dB) as a polar signal needs. This is because bipolar detection is essentially equivalent to on-off signaling from the detection point of view. One distinguishes between $+p(t)$ or $-p(t)$ from 0 rather than between $\pm p(t)$.

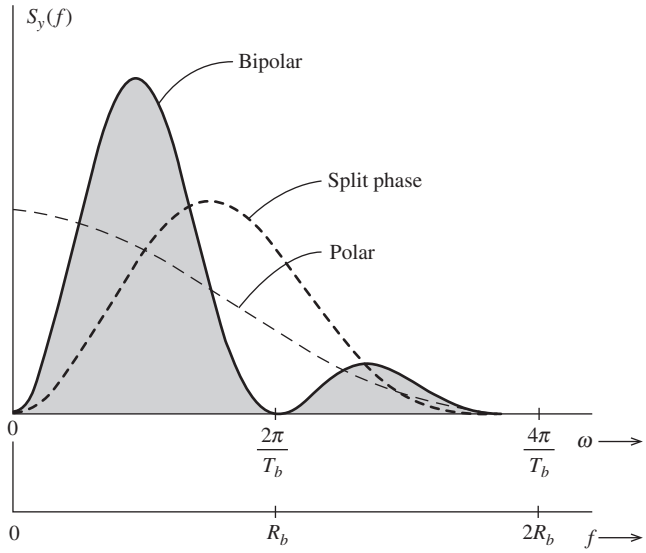

Figure 6.9 PSD of bipolar, polar, and split-phase signals normalized for equal powers. Half-width rectangular pulses are used.

Another disadvantage of bipolar signaling is that it is not transparent. In practice, various substitution schemes are used to prevent long strings of logic zeros from allowing the extracted clock signals to drift away. We shall now discuss two such schemes.

High-Density Bipolar (HDB) Signaling

The HDB scheme is an ITU (formerly CCITT) standard. In this scheme the problem of nontransparency in bipolar signaling is eliminated by adding pulses when the number of consecutive **0**s exceeds n. Such a modified coding is designated as **high-density bipolar** (HDBn) coding for a given positive integer n. The most important of the HDB codes is HDB3 format, which has been adopted as an international standard.

The basic idea of the HDBn code is that when a run of $n+1$ zeros occurs, this group of zeros is replaced by one of the special $n+1$ binary digit sequences. To strengthen the timing content of the signal, the replacement sequences are chosen to include some binary **1**s. The **1**s included would deliberately violate the bipolar rule for easy identification of the substituted sequence. In HDB3 coding, for example, the special sequences used are **000V** and **B00V** where **B = 1** that conforms to the bipolar rule and **V = 1** that violates the bipolar rule. The choice of sequence **000V** or **B00V** is made in such a way that consecutive **V** pulses alternate signs to avoid dc wander and to maintain the dc null in the PSD. This requires that the sequence **B00V** be used when there are an even number of **1**s following the last special sequence and the sequence **000V** be used when there are an odd number of **1**s following the last sequence. Figure 6.10a shows an example of this coding. Note that in the sequence **B00V**, both **B** and **V** are encoded by the same pulse. The decoder has to check two things—the bipolar violations and the number of **0**s preceding each violation to determine if the previous **1** is also a substitution.

Despite deliberate bipolar violations, HDB signaling retains error detecting capability. Any single error will insert a spurious bipolar violation (or will delete one of the deliberate violations). This will become apparent when, at the next violation, the alternation of violations does not appear. This also shows that deliberate violations can be detected despite single

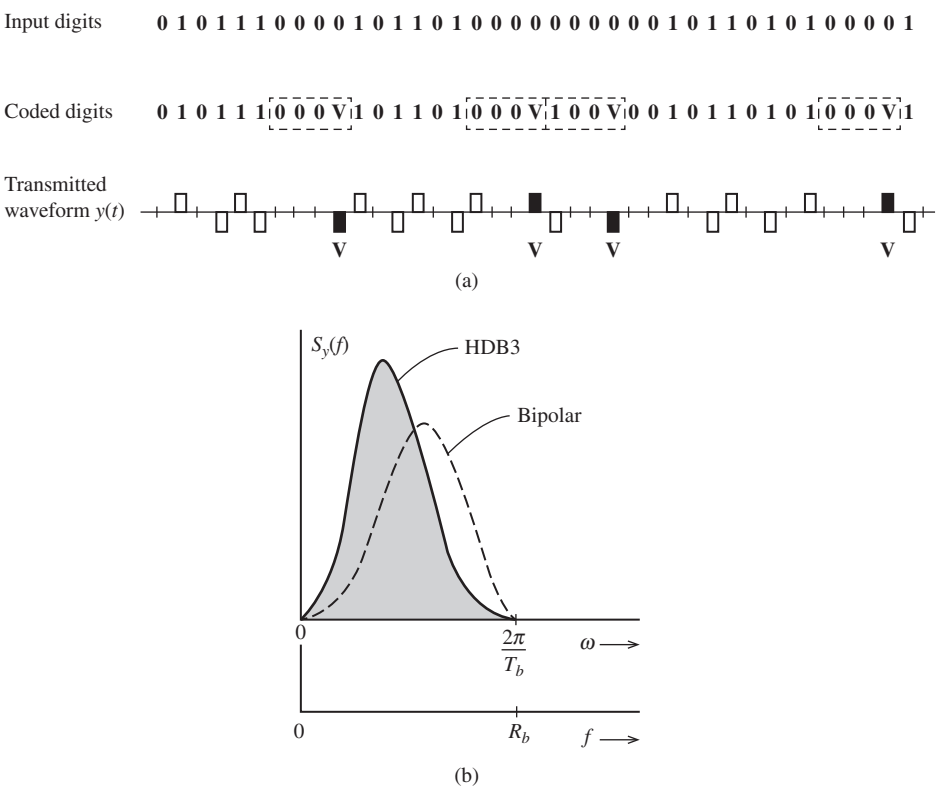

Figure 6.10 (a) HDB3 signal and (b) its PSD.

errors. Figure 6.10b shows the PSD of HDB3 as well as that of a bipolar signal to facilitate comparison.[3]

Binary with *N* Zero Substitution (BNZS) Signaling

A class of line codes similar to HDBN is the **binary with *N* zero substitution, or BNZS** code, where if N zeros occur in succession, they are replaced by one of the two special sequences containing some **1**s to increase timing content. There are deliberate bipolar violations just as in HDBN. Binary with eight-zero substitution (B8ZS) is used in DS1 signals of the digital telephone hierarchy in Sec. 5.4. It replaces any string of eight zeros in length with a sequence of ones and zeros containing two bipolar violations. Such a sequence is unlikely to be counterfeited by errors, and any such sequence received by a digital channel bank is replaced by a string of eight logic zeros prior to decoding. The sequence used as a replacement consists of the pattern **000VB0VB**. Similarly, in **B6ZS** code used in DS2 signals, a string of six zeros is replaced with **0VB0VB**, and DS3 signal features a three-zero B3ZS code. The B3ZS code is slightly more complex than the others in that either **B0V** or **00V** is used, the choice being made so that the number of **B** pulses between consecutive **V** pulses is odd. These BNZS codes with $N = 3, 6$, or 8 involve bipolar violations and must therefore be carefully replaced by their equivalent zero strings at the receiver.

There are many other transmission (line) codes, too numerous to list here. A list of codes and appropriate references can be found in Bylanski and Ingram.[3]

6.3 PULSE SHAPING

In the previous section, we have established that the PSD $S_y(f)$ of a digital signal $y(t)$ can be controlled by both line code and the pulse shape $P(f)$. The PSD $S_y(f)$ is strongly and directly influenced by the pulse shape $p(t)$ because $S_y(f)$ contains the multiplicative term $|P(f)|^2$. Thus, in comparison to the impact of the line code, the pulse shape is a more direct and potent factor in shaping the PSD $S_y(f)$. In this section, we examine how to select $p(t)$ or $P(f)$ to shape the PSD $S_y(f)$ to a desired form, and to mitigate self-interferences caused by limited channel bandwidth that would otherwise hamper the accurate detection of the digital baseband transmissions at the receiver.

6.3.1 Intersymbol Interferences (ISI) and Effect

For the sake of illustration, we used the simple half-width rectangular pulse $p(t)$ as an illustrate example. Strictly speaking, in this case the bandwidth of $S_y(f)$ is infinite, since $P(f)$ of rectangular pulse has infinite bandwidth. But we found that the effective bandwidth of $S_y(f)$ was finite. Specifically, most of the power of a bipolar signal is contained within the band from 0 to R_b Hz. Note, however, that the PSD is low but remains nonzero for $f > R_b$ Hz. Therefore, when such baseband pulse-modulated signals are transmitted over a lowpass channel of strict bandwidth R_b Hz, a significant portion of its spectrum goes through, but a small portion of the spectrum fails to reach the receiver. In Sec. 3.5 and Sec. 3.6, we saw how such a spectral distortion tends to spread the pulse (dispersion). Spreading of a pulse beyond its allotted time interval T_b will cause it to interfere with neighboring pulses. This is known as **intersymbol interference** or **ISI**.

First, ISI is **not** noise. ISI is caused by nonideal channels that are not distortionless over the entire input signal bandwidth. In the case of half-width rectangular pulse, the signal bandwidth is strictly infinity. ISI, as a manifestation of channel distortion, can cause errors in pulse detection if it is large enough.

To overcome the problem of ISI, let us review briefly our problem. We need to transmit a pulse every T_b interval, the kth pulse being $a_k p(t - kT_b)$. The channel has a finite bandwidth, and we are required to detect the pulse amplitude a_k correctly (i.e., without ISI). In our discussion so far, we have considered time-limited pulses. Since such pulses cannot be bandlimited, part of their spectra will always be blocked by a bandlimited channel. Thus, bandlimited channels cause pulse distortion (spreading out) and, consequently, ISI. We can try to resolve this difficulty by using pulses that are bandlimited to begin with so that they can be transmitted intact over a bandlimited channel. However, since bandlimited pulses cannot be time-limited, such pulses will obviously extend beyond their finite time slot of T_b to cause successive pulses to overlap and hence ISI. Thus, whether we begin with time-limited pulses or bandlimited pulses, it appears that ISI cannot be avoided. It is inherent in the finite transmission bandwidth. Fortunately, there is an escape from this dead end.

It is important to note that pulse amplitudes can be detected correctly despite pulse spreading (or overlapping), if there is no ISI at the **decision-making** instants. This can be accomplished by a properly shaped bandlimited pulse. To eliminate the effect of ISI, Nyquist proposed three different criteria for pulse shaping,[4] where the pulses are allowed to overlap in general. Yet, they are shaped to cause zero (or controlled) interference to all the other pulses at the critical decision-making instants. In summary, by limiting the noninterfering requirement only to the decision-making instants, we eliminate the unreasonable need for the bandlimited

pulse to be totally nonoverlapping. We shall consider only the first two Nyquist criteria. The third criterion is much less useful than the first two criteria, and hence, we refer our readers to the detailed discussions of Sunde.[5]

6.3.2 Nyquist's First Criterion for Zero ISI

Nyquist's first criterion achieves zero ISI by choosing a pulse shape that has a fixed nonzero amplitude at its center (say $t = 0$) and zero amplitudes at $t = \pm nT_b$ ($n = 1, 2, 3, \ldots$), where T_b is the separation between successive transmitted pulses (Fig. 6.11a). In other words, Nyquist's first criterion for zero ISI is

$$p(t) = \begin{cases} 1 & t = 0 \\ 0 & t = \pm nT_b \end{cases} \qquad \left(T_b = \frac{1}{R_b}\right) \tag{6.23}$$

A pulse satisfying this criterion causes zero ISI at all the remaining pulse centers, or signaling instants as shown in Fig. 6.11a, where we show several successive (dashed) pulses centered at $t = 0, T_b, 2T_b, 3T_b, \ldots$ ($T_b = 1/R_b$). For the sake of convenience, we have shown all pulses to be positive.* It is clear from this figure that the samples at $t = 0, T_b, 2T_b, 3T_b, \ldots$ consist of the amplitude of only one pulse (centered at the sampling instant) with no interference from the remaining pulses.

Recall from Chapter 5 that transmission of R_b bit/s requires a theoretical minimum bandwidth $R_b/2$ Hz. It would be nice if a pulse satisfying Nyquist's criterion had this minimum bandwidth $R_b/2$ Hz. Can we find such a pulse $p(t)$? We have already solved this problem (Example 5.1 with $B = R_b/2$), where we showed that there exists one (and only one) pulse that meets Nyquist's criterion of Eq. (6.23) and has a bandwidth $R_b/2$ Hz. This pulse,

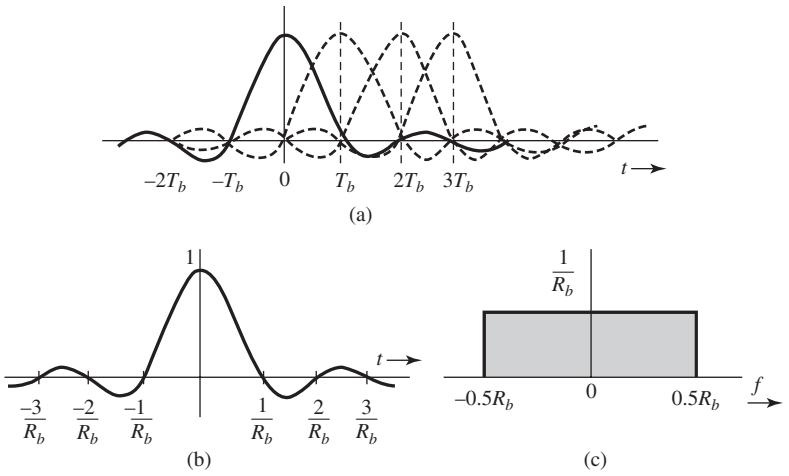

Figure 6.11 The minimum bandwidth pulse that satisfies Nyquist's first criterion and its spectrum.

* Actually, a pulse corresponding to **0** would be negative. But considering all positive pulses does not affect our reasoning. Showing negative pulses would make the figure needlessly confusing.

$p(t) = \text{sinc}\,(\pi R_b t)$ (Fig. 6.11b), has the desired property

$$\text{sinc}\,(\pi R_b t) = \begin{cases} 1 & t = 0 \\ 0 & t = \pm nT_b \end{cases} \qquad \left(T_b = \frac{1}{R_b}\right) \tag{6.24a}$$

Moreover, the Fourier transform of this pulse is

$$P(f) = \frac{1}{R_b}\Pi\left(\frac{f}{R_b}\right) \tag{6.24b}$$

which has a bandwidth $R_b/2$ Hz as seen from Fig. 6.11c. We can use this pulse (known as the **minimum bandwidth pulse for Nyquist's first criterion**) to transmit at a rate of R_b pulses per second without ISI, over the minimum bandwidth of only $R_b/2$.

This scheme shows that we can attain the theoretical limit of data rate for a given bandwidth by using the ideal sinc pulse without suffering from ISI. Unfortunately, this **minimum bandwidth** pulse is not feasible because it starts at $-\infty$. We will have to wait for eternity to accurately generate it. Any attempt to truncate it in time would increase its bandwidth beyond $R_b/2$ Hz. Furthermore, this pulse decays rather slowly at a rate $1/t$, causing some serious practical problems. For instance, if the nominal data rate of R_b bit/s required for this scheme deviates a little, the pulse amplitudes will not vanish at the other pulse centers kT_b. Because the pulses decay only as $1/t$, the cumulative interference at any pulse center from all the remaining pulses is of the order $\sum(1/n)$. It is well known that the infinite series of this form does not converge and can add up to a very large value. A similar result occurs if everything is perfect at the transmitter but the sampling rate at the receiver deviates from the rate of R_b Hz. Again, the same thing happens if the sampling instants deviate a little because of receiver timing jitter, which is inevitable even in the most sophisticated systems. And all this is because sinc $(\pi R_b t)$ decays too slowly (as $1/t$). The solution is to find a pulse $p(t)$ that satisfies Nyquist's first criterion in Eq. (6.23) but decays faster than $1/t$. Nyquist has shown that such a pulse requires a larger bandwidth $(1 + r)R_b/2$, with $r > 0$.

This can be proved by going into the frequency domain. Consider a pulse $p(t) \Longleftrightarrow P(f)$, where the bandwidth of $P(f)$ is in the range $(R_b/2,\ R_b)$ (Fig. 6.12a). Since the desired pulse $p(t)$ satisfies Eq. (6.23), if we sample $p(t)$ every T_b seconds by multiplying $p(t)$ by the impulse

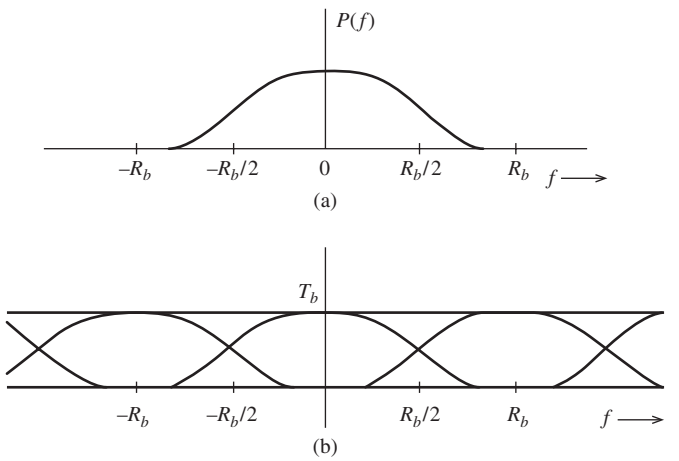

Figure 6.12 Derivation of the zero ISI Nyquist criterion pulse.

train $\delta_{T_b}(t)$, then all the samples would vanish except the one at the origin $t = 0$. Thus, the sampled signal $\overline{p}(t)$ is

$$\overline{p}(t) = p(t)\delta_{T_b}(t) = \delta(t) \tag{6.25}$$

Following the analysis of Eq. (5.4) in Chapter 5, we know that the spectrum of a sampled signal $\overline{p}(t)$ is equal to ($1/T_b$ times) the spectrum of $p(t)$ repeating periodically at intervals of the sampling frequency R_b. Therefore, the Fourier transform of Eq. (6.25) yields

$$\mathcal{F}\{\overline{p}(t)\} = \frac{1}{T_b}\sum_{n=-\infty}^{\infty} P(f - nR_b) = 1 \qquad \text{where} \qquad R_b = \frac{1}{T_b} \tag{6.26}$$

or

$$\sum_{n=-\infty}^{\infty} P(f - nR_b) = T_b \tag{6.27}$$

Thus, the sum of the spectra formed by repeating $P(f)$ spaced R_b apart is a constant T_b, as shown in Fig. 6.12b.*

Consider the spectrum in Fig. 6.12b over the range $0 < f < R_b$. Over this range, only two terms $P(f)$ and $P(f - R_b)$ in the summation in Eq. (6.27) are involved. Hence

$$P(f) + P(f - R_b) = T_b \qquad 0 < f < R_b$$

Letting $x = f - R_b/2$, we have

$$P\left(x + \frac{R_b}{2}\right) + P\left(x - \frac{R_b}{2}\right) = T_b \qquad |x| < 0.5R_b \tag{6.28}$$

Use of the conjugate symmetry property [Eq. (3.11)] on Eq. (6.28) yields

$$P\left(\frac{R_b}{2} + x\right) + P^*\left(\frac{R_b}{2} - x\right) = T_b \qquad |x| < 0.5R_b \tag{6.29}$$

If we choose $P(f)$ to be real-valued and positive, then only $|P(f)|$ needs to satisfy Eq. (6.29). Because $P(f)$ is real, Eq. (6.29) implies

$$\left|P\left(\frac{R_b}{2} + x\right)\right| + \left|P\left(\frac{R_b}{2} - x\right)\right| = T_b \qquad |x| < 0.5R_b \tag{6.30}$$

* Observe that if $R_b > 2B$, where B is the bandwidth (in hertz) of $P(f)$, the repetitions of $P(f)$ are nonoverlapping, and the condition in Eq. (6.27) cannot be satisfied. For $R_b = 2B$, the condition is satisfied only for the ideal lowpass $P(f)[p(t) = \text{ sinc } (\pi R_b t)]$, which has been discussed. Hence, we must have $B > R_b/2$.

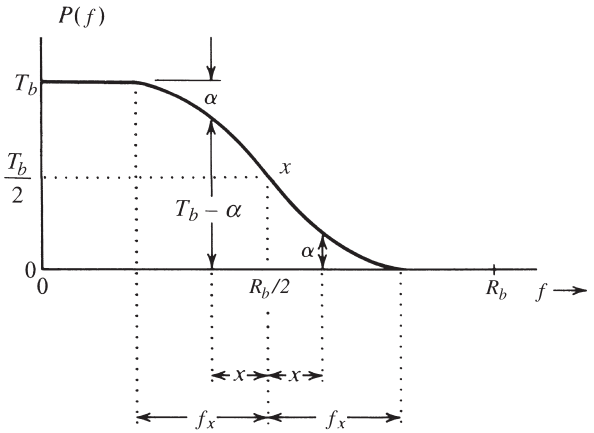

Figure 6.13 Vestigial (raised-cosine) spectrum.

Hence, $|P(f)|$ should be of the vestigial form shown in Fig. 6.13. This curve has an odd symmetry about the set of axes intersecting at point α [the point on $|P(f)|$ curve at $f = R_b/2$]. Note that this requires that

$$|P(0.5R_b)| = 0.5|P(0)|$$

The bandwidth of $P(f)$ is $0.5R_b + f_x$ in hertz, where f_x is the bandwidth in excess of the minimum bandwidth $R_b/2$. Let r be the ratio of the excess bandwidth f_x to the theoretical minimum bandwidth $R_b/2$:

$$\begin{aligned} r &= \frac{\text{excess bandwidth}}{\text{theoretical minimum bandwidth}} \\ &= \frac{f_x}{0.5R_b} \\ &= 2f_xT_b \end{aligned} \tag{6.31}$$

Observe that because f_x cannot be larger than $R_b/2$,

$$0 \le r \le 1 \tag{6.32}$$

In terms of frequency f, the theoretical minimum bandwidth is $R_b/2$ Hz, and the excess bandwidth is $f_x = rR_b/2$ Hz. Therefore, the bandwidth of $P(f)$ is

$$B_T = \frac{R_b}{2} + \frac{rR_b}{2} = \frac{(1+r)R_b}{2} \tag{6.33}$$

The constant r is called the **roll-off factor**. For example, if $P(f)$ is a Nyquist first criterion spectrum with a bandwidth that is 50% higher than the theoretical minimum, its roll-off factor $r = 0.5$ or 50%.

A filter having an amplitude response with the same characteristics is required in the vestigial sideband modulation discussed in Sec. 4.4.3 [Eq. (4.25)]. For this reason, we shall refer to the spectrum $P(f)$ in Eqs. (6.29) and (6.30) as a **vestigial spectrum**. The pulse $p(t)$ in Eq. (6.23) has zero ISI at the centers (decision instants) of all other pulses transmitted at a rate of R_b pulses per second, which satisfies the Nyquist's first criterion. Thus, we have shown that a pulse with a vestigial spectrum [Eq. (6.29) or Eq. (6.30)] satisfies the Nyquist's first criterion for zero ISI.

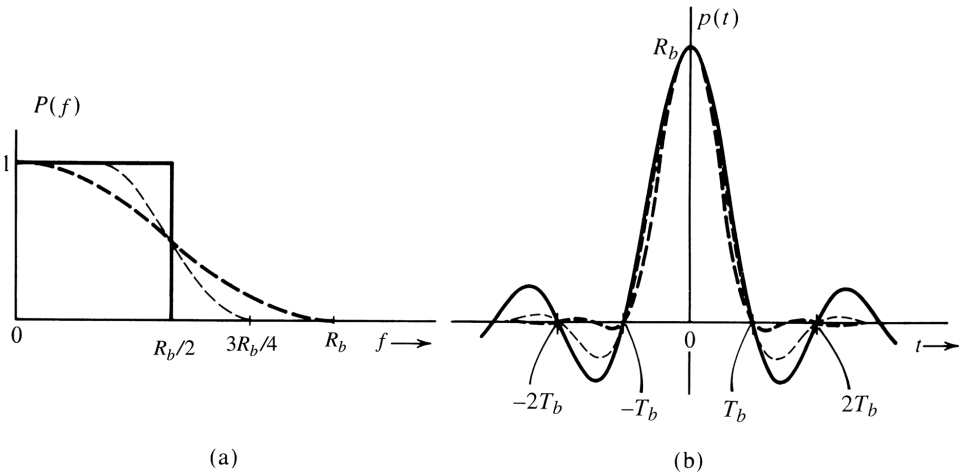

Figure 6.14 Pulses satisfying Nyquist's first criterion: solid curve, ideal $f_x = 0$ ($r = 0$); light dashed curve, $f_x = R_b/4$ ($r = 0.5$); heavy dashed curve, $f_x = R_b/2$ ($r = 1$).

Because it is typical that $0 \leq r < 1$, the bandwidth of $P(f)$ is restricted to the range $[R_b/2, R_b]$ in Hz. The pulse $p(t)$ can be generated as a unit impulse response of a filter with transfer function $P(f)$. But because $P(f) = 0$ over the frequency band $|f| \geq B_T$, it violates the Paley-Wiener criterion and is therefore unrealizable. However, the vestigial roll-off characteristic is smooth and gradual, making it easier to approximate by using a practical filter. One family of spectra that satisfies Nyquist's first criterion is the **raised cosine**

$$P(f) = \begin{cases} 1, & |f| < \dfrac{R_b}{2} - f_x \\ \dfrac{1}{2}\left[1 - \sin \pi \left(\dfrac{f - R_b/2}{2f_x}\right)\right] & \left|f - \dfrac{R_b}{2}\right| < f_x \\ 0, & |f| > \dfrac{R_b}{2} + f_x \end{cases} \tag{6.34}$$

Figure 6.14a shows three curves from this family, corresponding to $f_x = 0$ ($r = 0$), $f_x = R_b/4$ ($r = 0.5$) and $f_x = R_b/2$ ($r = 1$). The respective impulse responses are shown in Fig. 6.14b. It can be seen that increasing f_x (or r) simplifies the implementation of $p(t)$; that is, more gradual cutoff reduces the oscillatory nature of $p(t)$ and causes it to decay more rapidly in time domain. For the case of the maximum value of $f_x = R_b/2$ ($r = 1$), Eq. (6.34) reduces to

$$P(f) = \frac{1}{2}(1 + \cos \pi f T_b)\, \Pi\left(\frac{f}{2R_b}\right) \tag{6.35a}$$

$$= \cos^2\left(\frac{\pi f T_b}{2}\right) \Pi\left(\frac{f T_b}{2}\right) \tag{6.35b}$$

This characteristic of Eq. (6.34) is known in the literature as the **raised-cosine** characteristic, because it represents a cosine raised by its peak amplitude. Eq. (6.35) is also known as the

full-cosine roll-off characteristic. The inverse Fourier transform of this spectrum is readily found as (see Prob 6.3-8)

$$p(t) = R_b \frac{\cos \pi R_b t}{1 - 4R_b^2 t^2} \operatorname{sinc}(\pi R_b t) \tag{6.36}$$

This pulse is shown in Fig. 6.14b ($r = 1$). We can make several important observations about the full raised-cosine pulse of Eq. (6.36). First, the bandwidth of this pulse is R_b Hz and equals R_b at $t = 0$. It is zero not only at all the remaining signaling instants but also at points midway between all the signaling instants. Second, it decays rapidly, as $1/t^3$. As a result, the full raised-cosine pulse is relatively insensitive to deviations of R_b, sampling rate, timing jitter, and so on. Furthermore, the pulse-generating filter with transfer function $P(f)$ [Eq. (6.35b)] is approximately realizable. The phase characteristic that goes along with this filter is very close to linear, so that no additional phase equalization is needed.

It should be remembered that those pulses received at the detector input should have the form for zero ISI. In practice, because the channel is not distortionless, the transmitted pulses should be shaped so that after passing through the channel with transfer function $H_c(f)$, they have the proper shape (such as raised-cosine pulses) at the receiver. Hence, the transmitted pulse $p_i(t)$ should satisfy

$$P_i(f)H_c(f) = P(f)$$

where $P(f)$ has the vestigial spectrum in Eq. (6.30). For convenience, the transfer function $H_c(f)$ as a channel may **further** include a receiver filter designed to reject interference and other out-of-band noises.

Example 6.1 Determine the pulse transmission rate in terms of the transmission bandwidth B_T and the roll-off factor r. Assume a scheme using Nyquist's first criterion.

From Eq. (6.33)

$$R_b = \frac{2}{1+r} B_T$$

Because $0 \le r \le 1$, the pulse transmission rate varies from $2B_T$ to B_T, depending on the choice of r. A smaller r gives a higher signaling rate. But the corresponding pulse $p(t)$ decays more slowly, creating the same problems as those discussed for the sinc pulse. For the full raised-cosine pulse $r = 1$ and $R_b = B_T$, we achieve half the theoretical maximum rate. But the pulse decays faster as $1/t^3$ and is less vulnerable to ISI.

Example 6.2 A pulse $p(t)$ whose spectrum $P(f)$ is shown in Fig. 6.14a satisfies the Nyquist criterion. If $f_x = 0.8$ MHz and $T_b = 0.5\mu s$, determine the rate at which binary data can be transmitted by this pulse via the Nyquist criterion. What is the roll-off factor?

For this transmission, $R_b = 1/T_b = 2$ MHz. Moreover, the roll-off factor equals

$$r = \frac{f_x}{0.5R_b} = \frac{0.8 \times 10^6}{2 \times 10^6} = 0.8$$

6.3.3 Controlled ISI or Partial Response Signaling

The Nyquist's first criterion pulse requires a bandwidth somewhat larger than the theoretical minimum. If we wish to use the minimum bandwidth, we must find a way to widen the pulse $p(t)$ (the wider the pulse, the narrower the bandwidth) without detrimental ISI effects. Widening the pulse may result in interference (ISI) with the neighboring pulses. However, in the binary transmission with just two possible symbols, it may be easier to remove or compensate a known and controlled amount of ISI since there are only a few possible interference patterns to consider.

Consider a pulse specified by (see Fig. 6.15)

$$p_{\text{II}}(nT_b) = \begin{cases} 1 & n = 0, 1 \\ 0 & \text{for all other } n \end{cases} \tag{6.37}$$

This leads to a known and controlled ISI from the kth pulse to the next $(k+1)$th transmitted pulse. If we use polar signaling with this pulse, **1** is transmitted as $p_{\text{II}}(t)$ and **0** is transmitted as $-p_{\text{II}}(t)$. The received signal is sampled at $t = nT_b$, and the pulse p_{II} has zero value at all n except for $p_{\text{II}}(0) = p_{\text{II}}(T_b) = 1$ (Fig. 6.15). Clearly, such a pulse causes zero ISI to all other pulses except the very next pulse. Therefore, we only need to overcome the ISI on the succeeding pulse. Consider two such successive pulses located at 0 and T_b, respectively. If both pulses were positive, the sample value of the resulting signal at $t = T_b$ would be 2. If both pulses were negative, the sample value would be -2. But if the two pulses were of opposite polarity, the sample value would be 0. With only these three possible values, the signal sample clearly allows us to make the correct decision at the sampling instants by applying a decision rule as follows. If the sample value is positive, the present bit is **1** and the previous bit is also **1**. If the sample value is negative, the present bit is **0** and the previous bit is also **0**. If the sample value is zero, the present bit is the opposite

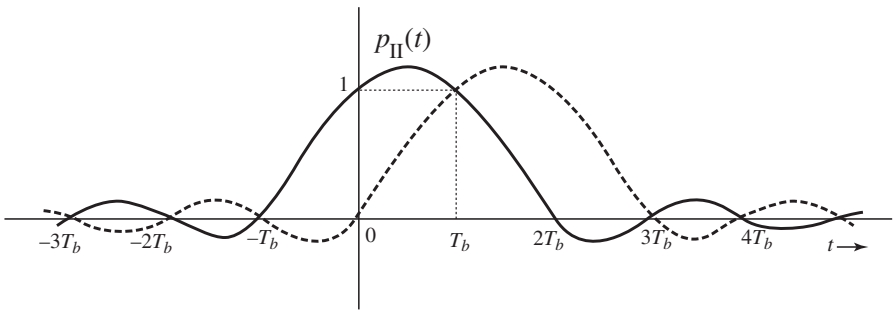

Figure 6.15 Communication using controlled ISI or Nyquist second criterion pulses.

TABLE 6.1
Transmitted Bits and the Received Samples in Controlled ISI Signaling

Information sequence	1	1	0	1	1	0	0	0	1	0	1	1	1
Samples $y(kT_b)$	1	2	0	0	2	0	−2	−2	0	0	0	2	2
Detected sequence	1	1	0	1	1	0	0	0	1	0	1	1	1

of the previous bit. Knowledge of the previous bit then allows the determination of the present bit.

Table 6.1 shows a transmitted bit sequence, the sample values of the received signal $y(t)$ (assuming no errors caused by channel noise), and the detector decision. This example also indicates the error-detecting property of this scheme. Examination of samples of the waveform $y(t)$ in Table 6.1 shows that there are always an even number of zero-valued samples between two full-valued samples of the same polarity and an odd number of zero-valued samples between two full-valued samples of opposite polarity. If one of the sample values is detected wrong, this rule is violated, and the error can be detected.

The pulse $p_{\Pi}(t)$ goes to zero at $t = -T_b$ and $2T_b$, resulting in the pulse width (of the primary lobe) 50% higher than that of the Nyquist's first criterion pulse. This pulse broadening in the time domain leads to bandwidth reduction. Such is the Nyquist's second criterion. This scheme of controlled ISI is also known as **correlative** or **partial-response** scheme. A pulse satisfying the second criterion in Eq. (6.37) is also known as the **duobinary pulse**.

6.3.4 Example of a Duobinary Pulse

If we restrict the pulse bandwidth to $R_b/2$, then following the procedure of Example 6.1, we can show that (see Prob 6.3-9) only the following pulse $p_{\Pi}(t)$ meets the requirement in Eq. (6.37) for the duobinary pulse:

$$p_{\Pi}(t) = \frac{\sin(\pi R_b t)}{\pi R_b t(1 - R_b t)} \tag{6.38}$$

The Fourier transform $P_{\Pi}(f)$ of the pulse $p_{\Pi}(t)$ is given by (see Prob 6.3-9)

$$P_{\Pi}(f) = \frac{2}{R_b} \cos\left(\frac{\pi f}{R_b}\right) \Pi\left(\frac{f}{R_b}\right) e^{-j\pi f/R_b} \tag{6.39}$$

The pulse $p_{\Pi}(t)$ and its amplitude spectrum $|P(f)|$ are shown in Fig. 6.16.* This pulse transmits binary data at a rate of R_b bit/s and has the theoretical minimum bandwidth $R_b/2$ Hz. This pulse is not ideally realizable because $p_{\Pi}(t)$ is noncausal and has infinite duration [because $P_{\Pi}(f)$ is bandlimited]. However, Eq. (6.38) shows that this pulse decays rapidly with time as $1/t^2$, and therefore can be closely approximated.

* The phase spectrum is linear with $\theta_p(f) = -\pi f T_b$.

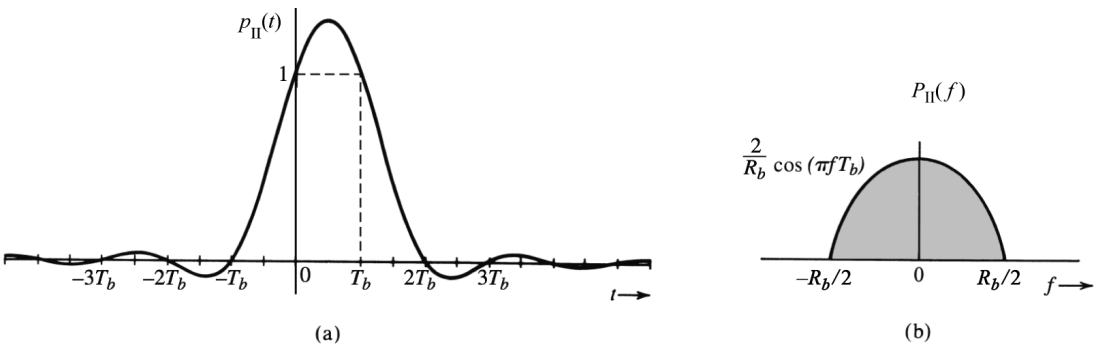

Figure 6.16 (a) The minimum bandwidth pulse that satisfies the duobinary pulse criterion and (b) its spectrum.

6.3.5 Pulse Relationship between Zero-ISI, Duobinary, and Modified Duobinary

Now we can establish the simple relationship between a pulse $p(t)$ satisfying the Nyquist's first criterion (zero ISI) and a duobinary pulse $p_{II}(t)$ (with controlled ISI). From Eqs. (6.23) and (6.37), it is clear that $p(kT_b)$ and $p_{II}(kT_b)$ only differ for $k = 1$. They have identical sample values for all other integers k. Therefore, one can easily construct a pulse $p_{II}(t)$ from $p(t)$ via

$$p_{II}(t) = p(t) + p(t - T_b)$$

This addition is the "controlled" ISI or partial-response signaling that we deliberately introduced to reduce the bandwidth requirement. To see what effect "duobinary" signaling has on the spectral bandwidth, consider the relationship of the two pulses in the frequency domain:

$$P_{II}(f) = P(f)[1 + e^{-j2\pi f T_b}] \tag{6.40a}$$

$$|P_{II}(f)| = |P(f)|\sqrt{2(1 + \cos(2\pi f T_b)} = 2|P(f)|\,|\cos(\pi f T_b)| \tag{6.40b}$$

We can see that partial-response signaling is actually forcing a frequency null at $f = 0.5/T_b$. Therefore, conceptually we can see how partial-response signaling provides an additional opportunity to reshape the PSD or the transmission bandwidth. Indeed, duobinary signaling, by forcing a frequency null at $0.5/T_b$, forces its effective bandwidth to be the minimum transmission bandwidth needed for a data rate of $1/T_b$ (as discussed in Sec. 5.1.3).

In fact, many physical channels such as magnetic recording have a zero gain at dc. Therefore, it makes no sense for the baseband signal to have any dc component in its PSD. Modified partial-response signaling is often adopted to force a null at dc. One notable example is the so-called **modified duobinary** signaling that requires

$$p_{MD}(nT_b) = \begin{cases} 1 & n = -1 \\ -1 & n = 1 \\ 0 & \text{for all other integers } n \end{cases} \tag{6.41}$$

A similar argument indicates that $p_{\mathrm{MD}}(t)$ can be generated from any pulse $p(t)$ satisfying the first Nyquist criterion via

$$p_{\mathrm{MD}}(t) = p(t + T_b) - p(t - T_b)$$

Equivalently, in the frequency domain, the duobinary pulse is

$$P_{\mathrm{MD}}(f) = 2jP(f)\sin(2\pi f T_b)$$

which uses $\sin(2\pi f T_b)$ to force a null at dc to comply with the physical channel constraint.

6.3.6 Detection of Duobinary Signaling and Differential Encoding

For the controlled ISI method of duobinary signaling, Fig. 6.17 provides a basic transmitter diagram. We now take a closer look at the relationship of all the data symbols at the baseband and the detection procedure. For binary message bit $I_k = 0$, or 1, the polar symbols are simply

$$a_k = 2I_k - 1$$

Under the controlled ISI, the samples of the transmission signal $y(t)$ are

$$y(kT_b) = b_k = a_k + a_{k-1} \tag{6.42}$$

The question for the receiver is how to **detect** I_k from $y(kT_b)$ or b_k. This question can be answered by first considering all the possible values of b_k or $y(kT_b)$. Because $a_k = \pm 1$, then $b_k = 0, \pm 2$. From Eq. (6.42), it is evident that

$$\begin{array}{lcll} b_k = 2 & \Rightarrow & a_k = 1 & \text{or } I_k = 1 \\ b_k = -2 & \Rightarrow & a_k = -1 & \text{or } I_k = 0 \\ b_k = 0 & \Rightarrow & a_k = -a_{k-1} & \text{or } I_k = 1 - I_{k-1} \end{array} \tag{6.43}$$

Therefore, a simple detector of duobinary signaling is to first detect all the bits I_k corresponding to $b_k = \pm 2$. The remaining $\{b_k\}$ are zero-valued samples that imply transition: that is, the current digit is **1** and the previous digit is **0**, or vice versa. This means the digit detection must be based on the previous digit. An example of such a digit-by-digit detection was shown in Table 6.1. The disadvantage of the detection method in Eq. (6.43) is that when $y(kT_b) = 0$, the current bit decision depends on the previous bit decision. If the previous digit were detected incorrectly, then the error would tend to propagate, until a sample value of ± 2 appeared. To mitigate this error propagation problem, we apply an effective mechanism known as **differential coding**.

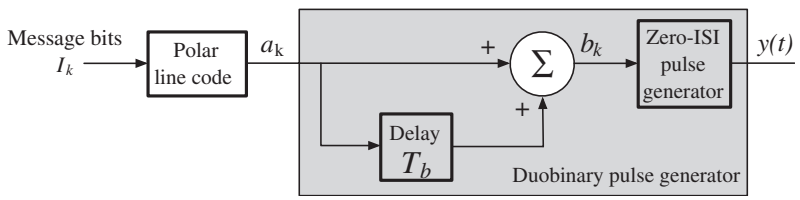

Figure 6.17 Equivalent duobinary signaling.

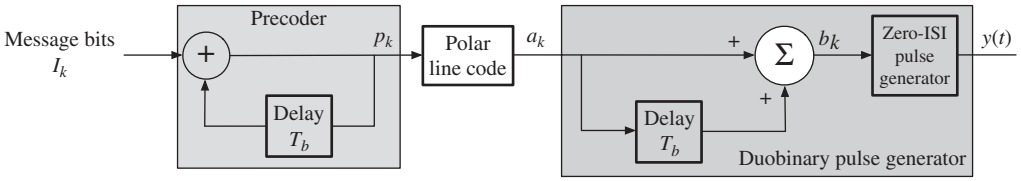

Figure 6.18 Differential encoded duobinary signaling.

Figure 6.18 illustrates a duobinary signal generator by introducing an additional differential encoder prior to partial-response pulse generation. As shown in Fig. 6.18, differential encoding is a very simple step that changes the relationship between line code and the message bits. Differential encoding generates a new binary sequence

$$p_k = I_k \oplus p_{k-1} \qquad \text{modulo 2}$$

with the assumption that the precoder initial state is either $p_0 = 0$ or $p_0 = 1$. Now, the precoder output enters a polar line coder and generates

$$a_k = 2p_k - 1$$

Because of the duobinary signaling $b_k = a_k + a_{k-1}$ and the zero-ISI pulse, the samples of the received signal $y(t)$ without noise become

$$\begin{aligned} y(kT_b) = b_k &= a_k + a_{k-1} \\ &= 2(p_k + p_{k-1}) - 2 \\ &= 2\,(p_{k-1} \oplus I_k + p_{k-1} - 1) \\ &= \begin{cases} 2(1 - I_k) & p_{k-1} = 1 \\ 2(I_k - 1) & p_{k-1} = 0 \end{cases} \end{aligned} \tag{6.44}$$

Based on Eq. (6.44), we can summarize the direct relationship between the message bits and the sample values as

$$y(kT_b) = \begin{cases} 0 & I_k = 1 \\ \pm 2 & I_k = 0 \end{cases} \tag{6.45}$$

This relationship serves as our basis for a symbol-by-symbol detection algorithm. In short, the decision algorithm is based on the current sample $y(kT_b)$. When there is no noise, $y(kT_b) = b_k$, and the receiver decision is

$$I_k = \frac{2 - |y(kT_b)|}{2} \tag{6.46}$$

Therefore, the incorporation of differential encoding with duobinary signaling not only simplifies the decision rule but also makes the decision independent of the previous digit and eliminates error propagation. In Table 6.2, the example of Table 6.1 is recalculated under differential encoding. The decoding relationship of Eq. (6.45) is clearly shown in this example.

The differential encoding defined for binary information symbols can be conveniently generalized to nonbinary symbols. When the information symbols I_k are M-ary, the only change to the differential encoding block is to replace "modulo 2" with "modulo M."

TABLE 6.2
Binary Duobinary Signaling with Differential Encoding

Time k	0	1	2	3	4	5	6	7	8	9	10	11	12	13
I_k		1	1	0	1	1	0	0	0	1	0	1	1	1
p_k	0	1	0	0	1	0	0	0	0	1	1	0	1	0
a_k	−1	1	−1	−1	1	−1	−1	−1	−1	1	1	−1	1	−1
b_k		0	0	−2	0	0	−2	−2	−2	0	2	0	0	0
Detected bits		1	1	0	1	1	0	0	0	1	0	1	1	1

Similarly, other generalized partial-response signaling such as the modified duobinary must also face the error propagation problem at its detection. A suitable type of differential encoding can be similarly adopted to prevent error propagation.

6.3.7 Pulse Generation

A pulse $p(t)$ satisfying a Nyquist criterion can be generated as the unit impulse response of a filter with transfer function $P(f)$. A simpler alternative is to generate the waveform directly, using a transversal filter (tapped delay line) discussed here. The pulse $p(t)$ to be generated is sampled with a sufficiently small sampling interval T_s (Fig. 6.19a), and the filter tap gains are set in proportion to these sample values in sequence, as shown in Fig. 6.19b. When a narrow rectangular pulse with the width T_s, the sampling interval, is applied at the input of the transversal filter, the output will be a staircase approximation of $p(t)$. This output, when passed through a lowpass filter, is smoothed out. The approximation can be improved by reducing the pulse sampling interval T_s.

It should be stressed once again that the pulses arriving at the detector input of the receiver need to meet the desired Nyquist criterion. Hence, the transmitted pulses should be shaped such that after passing through the channel, they are received in the desired (zero ISI) form. In practice, however, pulses need not be shaped fully at the transmitter. The final shaping can be carried out by an equalizer at the receiver, as discussed later (Sec. 6.5).

6.4 SCRAMBLING

In general, a scrambler tends to make the data more random by removing long strings of **1**s or **0**s. Scrambling can be helpful in timing extraction by removing long strings of **0**s in binary data. Scramblers, however, are primarily used for preventing unauthorized access to the data. The digital network may also cope with these long zero strings by adopting the zero replacement techniques discussed in Sec. 6.2.5.

Figure 6.20 shows a typical scrambler and descrambler. The scrambler consists of a feedback shift register, and the matching descrambler has a feedforward shift register, as shown in Fig. 6.20. Each stage in the shift register delays a bit by one unit. To analyze the scrambler and the matched descrambler, consider the output sequence T of the scrambler (Fig. 6.20a). If S is the input sequence to the scrambler, then

$$S \oplus D^3T \oplus D^5T = T \tag{6.47}$$

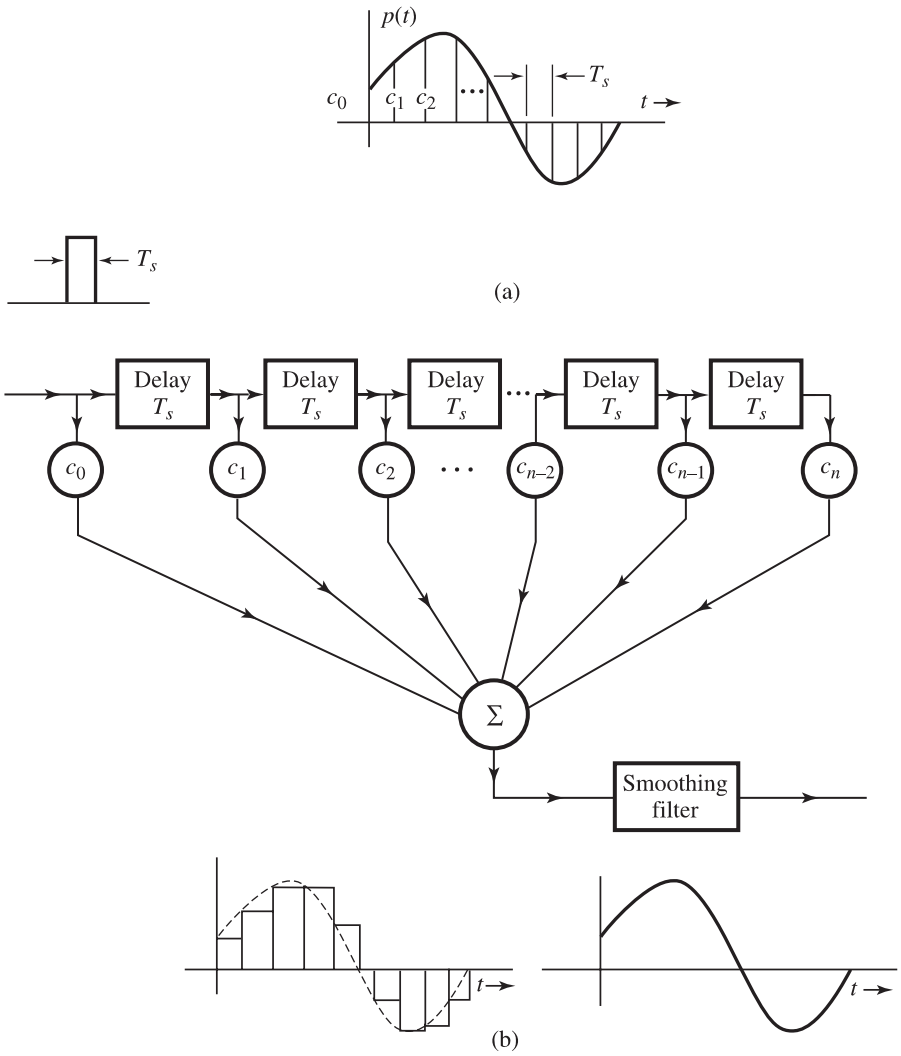

Figure 6.19
Pulse generation by transversal filter.

where D represents a unit delay; that is, D^nT is the sequence T delayed by n units. Now, recall that the modulo 2 sum of any sequence with itself gives a sequence of all **0**s. Adding $(D^3 \oplus D^5)T$ to both sides of Eq. (6.47), we get

$$\begin{aligned} S &= T \oplus (D^3 \oplus D^5)T \\ &= [1 \oplus (D^3 \oplus D^5)]T \\ &= (1 \oplus F)T \end{aligned} \tag{6.48}$$

where $F = D^3 \oplus D^5$.

To design the descrambler at the receiver, we start with T, the sequence received at the descrambler. From Eq. (6.48), it follows that

$$T \oplus FT = T \oplus (D^3 \oplus D^5)T = S$$

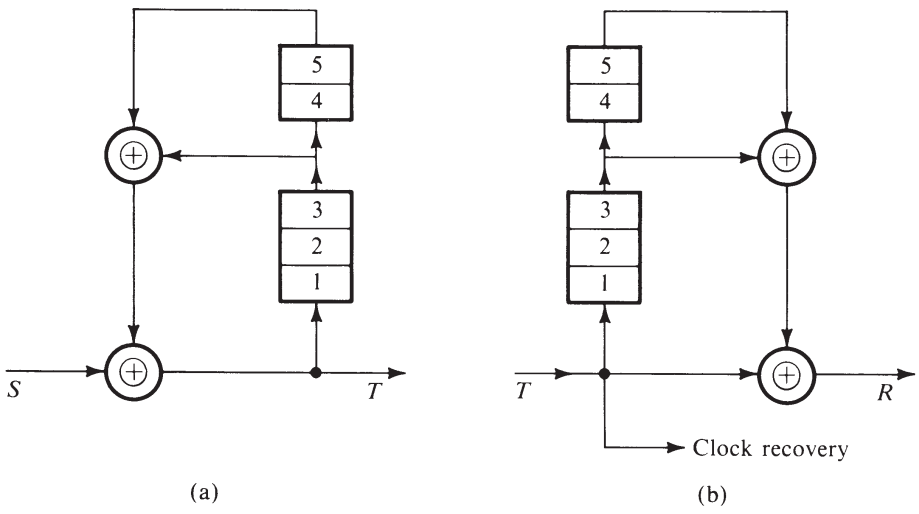

Figure 6.20
(a) Scrambler.
(b) Descrambler.

This equation, through which we regenerate the input sequence S from the received sequence T, is readily implemented by the descrambler shown in Fig. 6.20b.

Note that a single detection error in the received sequence T will affect three output bits in R. Hence, scrambling has the disadvantage of causing multiple errors because of feedback error propagation from a single received bit error at the descrambler input.

Example 6.3 The data stream **101010100000111** is fed to the scrambler in Fig. 6.20a. Find the scrambler output T, assuming the initial content of the registers to be zero.

From Fig. 6.20a we observe that initially $T = S$, and the sequence S enters the register and is returned as $(D^3 \oplus D^5)S = FS$ through the feedback path. This new sequence FS again enters the register and is returned as F^2S, and so on. Hence

$$\begin{aligned} T &= S \oplus FS \oplus F^2S \oplus F^3S \oplus \cdots \\ &= (1 \oplus F \oplus F^2 \oplus F^3 \oplus \cdots)S \end{aligned} \tag{6.49}$$

Recognizing that

$$F = D^3 \oplus D^5$$

we have

$$F^2 = (D^3 \oplus D^5)(D^3 \oplus D^5) = D^6 \oplus D^{10} \oplus D^8 \oplus D^8$$

Because modulo-2 addition of any sequence with itself is zero, $D^8 \oplus D^8 = 0$, and

$$F^2 = D^6 \oplus D^{10}$$

Similarly

$$F^3 = (D^6 \oplus D^{10})(D^3 \oplus D^5) = D^9 \oplus D^{11} \oplus D^{13} \oplus D^{15}$$

and so on. Hence [see Eq. (6.49)],

$$T = (1 \oplus D^3 \oplus D^5 \oplus D^6 \oplus D^9 \oplus D^{10} \oplus D^{11} \oplus D^{12} \oplus D^{13} \oplus D^{15} \cdots)S$$

Because $D^n S$ is simply the sequence S delayed by n bits, various terms in the preceding equation correspond to the following sequences:

$$\begin{aligned}
S &= \mathbf{101010100000111} \\
D^3S &= \mathbf{000101010100000111} \\
D^5S &= \mathbf{0000101010100000111} \\
D^6S &= \mathbf{0000010101010000111} \\
D^9S &= \mathbf{000000000101010100000111} \\
D^{10}S &= \mathbf{000000000010101010000111} \\
D^{11}S &= \mathbf{000000000001010101000001111} \\
D^{12}S &= \mathbf{000000000000101010100000111} \\
D^{13}S &= \mathbf{000000000000010101010000111} \\
\underline{D^{15}S} &= \underline{\mathbf{000000000000000}}\mathbf{101010100000111} \\
T &= \mathbf{101110001101001}
\end{aligned}$$

Note that the input sequence contains the periodic sequence **10101010** $\cdots$, as well as a long string of **0**s. The scrambler output effectively removes the periodic component, as well as the long string of **0**s. The input sequence has 15 digits. The scrambler output up to the 15th digit only is shown, because all the output digits beyond 15 depend on input digits beyond 15, which are not given.

We can verify that the descrambler output is indeed S when the foregoing sequence T is applied at its input (Prob. 6.4-1).

6.5 DIGITAL RECEIVERS AND REGENERATIVE REPEATERS

Basically, a receiver or a regenerative repeater must perform three functions: (1) reshaping incoming pulses by means of an equalizer, (2) extracting the timing information required to sample incoming pulses at optimum instants, and (3) making symbol detection decisions based on the pulse samples. The repeater shown in Fig. 6.21 consists of a receiver plus a "regenerator," which must further re-modulate and re-transmit the recovered data from its receiver output. A complete repeater may also include provision for separation of dc power from ac signals. This is normally accomplished by transformer-coupling the signals and bypassing the dc around the transformers to the power supply circuitry.

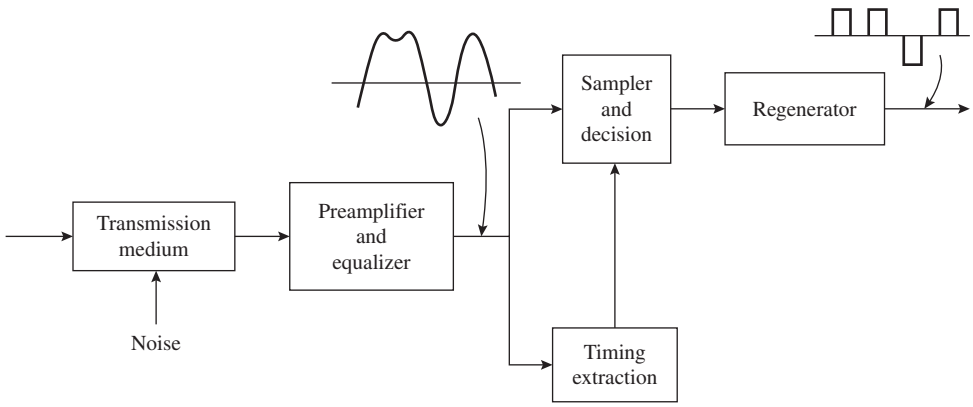

Figure 6.21 Regenerative repeater.

6.5.1 Equalizers

A data modulated baseband pulse train is often attenuated and distorted by the transmission medium. The attenuation can be compensated by the preamplifier, whereas the distortion can be compensated by an equalizer. Channel distortion is in the form of dispersion, which is caused by an attenuation of certain **critical frequency components** of the baseband data pulse train. Theoretically, an equalizer should have a frequency characteristic that is the inverse of that of the distortive channel medium. This apparatus will restore the critical frequency components and eliminate pulse dispersion. Unfortunately, the equalizer could also enhance the received channel noise by boosting its components at these critical frequencies. This undesirable phenomenon is known as **noise enhancement** or **noise amplification**.

For digital signals, however, complete equalization is in fact unnecessary because a detector only needs to make relatively simple decisions—such as whether the pulse is positive or negative in polar signaling (or whether the pulse is present or absent in on-off signaling). Therefore, considerable residual pulse dispersion can be tolerated. Pulse dispersion results in ISI and the consequent increase in detection errors. Noise enhancement resulting from the equalizer (which boosts the high frequencies) can also increase the detection error probability. For this reason, designing an optimum equalizer involves an inevitable compromise between mitigating the ISI and suppressing the channel noise. A judicious choice of the equalization characteristics is a central feature in all well-designed digital communication systems.[6] We now describe two common and well-known equalizer designs for combating ISI: (a) zero-forcing (ZF) equalization; (b) minimum MSE equalization.

Zero-Forcing Equalizer Design

It is really not necessary to eliminate or minimize ISI (interference) with neighboring pulses for all t. All that is needed is to eliminate or minimize interference among neighboring pulses at their respective **sampling instants** only. This is because the receiver decision is based on signal sample values only. This kind of (relaxed) equalization can be accomplished by equalizers using the transversal filter structure as shown in Fig. 6.22a (also encountered earlier). Unlike traditional filters, transversal filter equalizers are easily adjustable to compensate against different channels or even slowly time-varying channels. The design goal

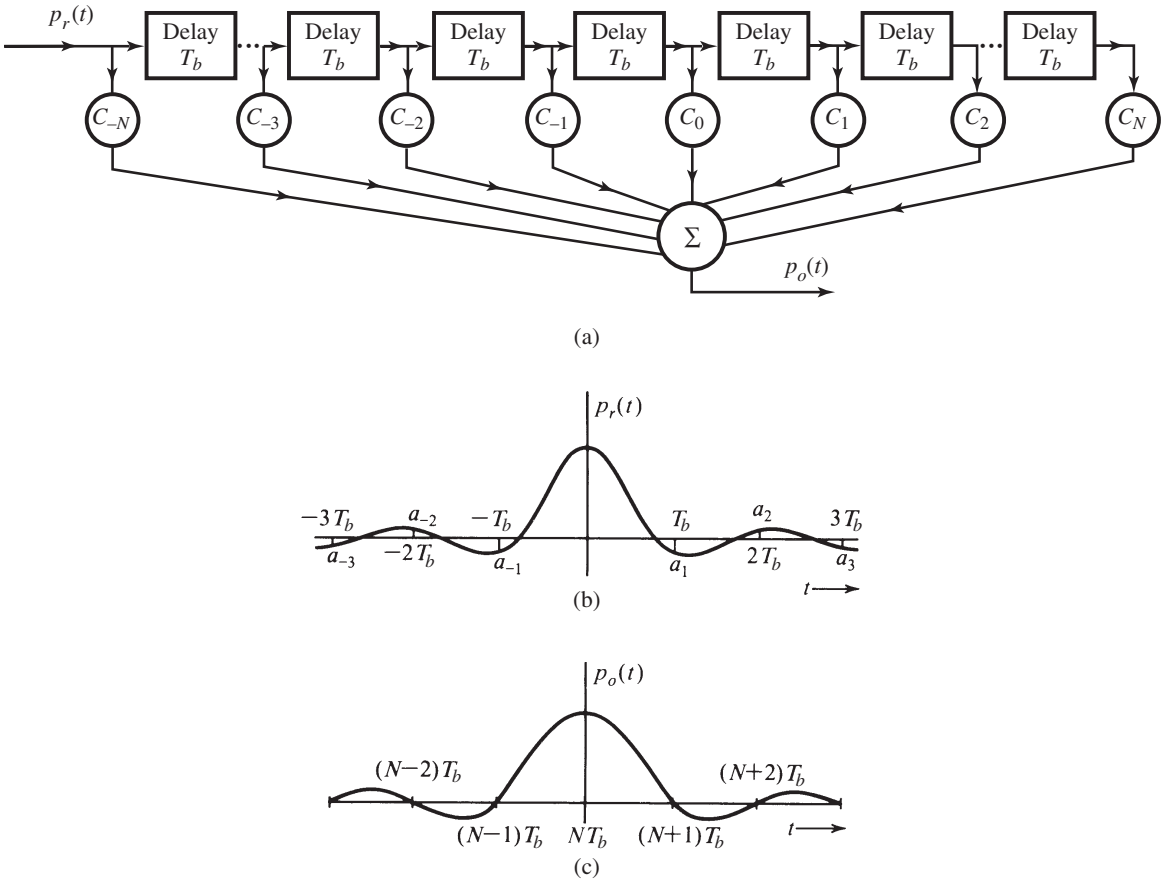

Figure 6.22 Zero-forcing equalizer analysis.

is to force the equalizer output pulse to have zero ISI values at the sampling (decision-making) instants. In other words, the goal is for the equalizer output pulses to satisfy the Nyquist's first criterion of Eq. (6.23). The time delay between successive taps is chosen to be T_b, the same interval for each data symbol in baseband modulation.

To begin, set the tap gains $c_0 = 1$ and $c_k = 0$ for all other values of k in the transversal filter in Fig. 6.22a. Thus the output of the filter will be the same as the input delayed by NT_b. For a single pulse $p_r(t)$ (Fig. 6.22b) at the input of the transversal filter with the tap setting just given, the filter output $p_o(t)$ will be exactly $p_r(t - NT_b)$, that is, $p_r(t)$ delayed by NT_b. This means that $p_r(t)$ in Fig. 6.22b also represents the filter output $p_o(t)$ for this tap setting ($c_0 = 1$ and $c_k = 0$, $k \neq 0$). We require that the output pulse $p_o(t)$ satisfy the Nyquist's criterion or the controlled ISI criterion, as the case may be. For the Nyquist criterion, the output pulse $p_o(t)$ must have zero values at $t = kT_b$ except for $k = N$. From Fig. 6.22b, we see that the pulse amplitudes a_1, a_{-1}, and a_2 at T_b, $-T_b$, and $2T_b$, respectively, are not negligible. By adjusting the tap gains (c_k), we generate additional shifted pulses of proper amplitudes that will force the resulting output pulse to have desired 0 value at $t = 0, \pm T_b, \pm 2T_b, \ldots$ except at the desired decision instant $p_o(NT_b) = 1$.

The output $p_o(t)$ (Fig 6.22c) is the sum of pulses of the form $c_k p_r(t - kT_b)$. Thus

$$p_o(t) = \sum_{n=0}^{2N} c_{n-N} p_r(t - nT_b) \tag{6.50}$$

The samples of $p_o(t)$ at $t = kT_b$ are

$$p_o(kT_b) = \sum_{n=0}^{2N} c_{n-N} p_r(kT_b - nT_b) \qquad k = 0, \pm 1, \pm 2, \pm 3, \ldots \tag{6.51a}$$

By using a more convenient notation $p_r[k]$ to denote $p_r(kT_b)$ and $p_o[k]$ to denote $p_o(kT_b)$, Eq. (6.51a) can be expressed as

$$p_o[k] = \sum_{n=0}^{2N} c_{n-N} p_r[k - n] \qquad k = 0, \pm 1, \pm 2, \pm 3, \ldots \tag{6.51b}$$

Considering the delay in the transversal filter, we can rewrite Nyquist's first criterion to require that the samples $p_o[k] = 0$ for $k \neq N$, and $p_o[N] = 1$. Upon substituting these values in Eq. (6.51b), we obtain a set of infinite simultaneous equations in terms of $2N + 1$ variables. Clearly, it is not possible to solve all the equations. However, if we specify the values of $p_o[k]$ only at $2N + 1$ points as

$$p_o[k] = \begin{cases} 1 & k = N \\ 0 & k = 0, \cdots, N-1, N+1, \cdots, 2N \end{cases} \tag{6.52}$$

then a unique solution may exist. This assures that a pulse will have zero interference at sampling instants of N preceding and N succeeding pulses. Because the pulse amplitude decays rapidly, interference beyond the Nth pulse in general is not significant, for large enough N. Substitution of the condition in Eq. (6.52) into Eq. (6.51b) yields a set of $2N + 1$ simultaneous equations for $2N + 1$ variables. These $2N + 1$ equations can be rewritten in the matrix form of

$$\underbrace{\begin{bmatrix} p_o[0] \\ \vdots \\ p_o[N] \\ \vdots \\ p_o[2N] \end{bmatrix}}_{\mathbf{p}_o} = \begin{bmatrix} 0 \\ \vdots \\ 0 \\ 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix} = \underbrace{\begin{bmatrix} p_r[0] & p_r[-1] & \cdots & p_r[-2N+1] & p_r[-2N] \\ p_r[1] & p_r[0] & \cdots & p_r[-2N+2] & p_r[-2N+1] \\ \vdots & \ddots & \ddots & \vdots & \vdots \\ p_r[2N-1] & p_r[2N-2] & \ddots & p_r[0] & p_r[-1] \\ p_r[2N] & p_r[2N-1] & \cdots & p_r[1] & p_r[0] \end{bmatrix}}_{\mathbf{P}_r} \underbrace{\begin{bmatrix} c_{-N} \\ c_{-N+1} \\ \vdots \\ c_{-1} \\ c_0 \\ c_1 \\ \vdots \\ c_{N-1} \\ c_N \end{bmatrix}}_{\mathbf{c}} \tag{6.53}$$

In this compact expression, the $(2N+1)\times(2N+1)$ matrix $\mathbf{P}_r$ has identical entries along all the diagonal lines. Such a matrix is known as the Toeplitz matrix and is commonly encountered in describing a convolutive relationship. A Toeplitz matrix is fully determined by its first row and first column. It has some nice properties and admits simpler algorithms for computing its inverse (see, e.g., the method by Trench[7]). The tap gain c_k can be obtained by solving this set of equations by taking the inverse of the matrix $\mathbf{P}_r$

$$\mathbf{c} = \mathbf{P}_r^{-1}\mathbf{p}_o$$

Example 6.4 For the received pulse $p_r(t)$ in Fig. 6.22b, let

$$\begin{aligned} p_r[0] &= 1 & p_r[k] &= 0 & k &\neq 0, \pm 1, \pm 2 \\ p_r[1] &= -0.3 & p_r[2] &= 0.18 & & \\ p_r[-1] &= -0.2 & p_r[-2] &= 0.24 & & \end{aligned}$$

Design a three-tap ($N = 1$) equalizer and also determine the residual ISI for this ZF equalizer.

Substituting the foregoing values in Eq. (6.53), we obtain

$$\begin{bmatrix} p_o[0] \\ p_o[1] \\ p_o[2] \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix} = \begin{bmatrix} 1 & -0.2 & 0.24 \\ -0.3 & 1 & -0.2 \\ 0.18 & -0.3 & 1 \end{bmatrix} \begin{bmatrix} c_{-1} \\ c_0 \\ c_1 \end{bmatrix} \tag{6.54}$$

Solution of this set yields $c_{-1} = 0.1479$, $c_0 = 1.1054$, and $c_1 = 0.3050$. This tap setting assures us that $p_0[1] = 1$ and $p_0[0] = p_0[2] = 0$. The ideal output $p_o(t)$ is sketched in Fig. 6.22c.

Note that the equalizer determined from Eq. (6.53) can guarantee only the zero ISI condition of Eq. (6.52). In other words, ISI is zero only for $k = 0, 1, \ldots, 2N$. In fact, for k outside this range, it is quite typical that the samples $p_o(kT_b) \neq 0$, indicating some residual ISI. For this example, the samples of the equalized pulse have zero ISI for $k = 0, 1, 2$. However, from

$$p_o[k] = \sum_{n=0}^{2N} c_{n-N} p_r[k-n]$$

we can see that the three-tap, ZF equalizer will lead to

$$p_o[k] = 0, \quad k = \cdots, -4, -3, 5, 6, \cdots$$

However, we also have the residual ISI as

$$\begin{aligned} &p_o[-2] = 0.0355 \quad p_o[-1] = 0.2357 \quad p_o[0] = 0 \quad p_o[1] = 1 \quad p_o[2] = 0 \\ &p_o[3] = 0.1075 \quad p_o[4] = 0.0549 \end{aligned}$$

It is therefore clear that not all the ISI has been removed by this particular ZF equalizer because of the four nonzero samples of the equalizer output pulse at $k = -2, -1, 3, 4$.

In fact, because we only have $2N+1$ ($N=1$ in Example 6.4) parameters in the equalizer, it is impossible to force $p_o[k]=0$, $k\neq N$ unless $N=\infty$. This means that we will not be able to design a practical finite tap equalizer to achieve perfect zero ISI. Still, when N is sufficiently large, typically the residual nonzero sample values will be small, indicating that most of the ISI has been suppressed by well designed ZF equalizers.

Minimum Mean Square Error (MMSE) Equalizer Design

In practice, we also apply another design approach aimed at minimizing the mean square error between the equalizer output response $p_o[k]$ and the desired zero ISI response. This is known as the minimum MSE (MMSE) method for designing transversal filter equalizers. The MMSE design does not try to force the pulse samples to zero at $2N$ points. Instead, we minimize the squared errors averaged over a set of output samples. This method involves more simultaneous equations. Thus we must find the equalizer tap values to minimize the average (mean) square error over a larger window of length $2K+1$, that is, we aim to minimize the mean square error (MSE):

$$\text{MSE} \triangleq \frac{1}{2K+1}\sum_{k=N-K}^{N+K}(p_o[k]-\delta[k-N])^2 \tag{6.55}$$

where we use a function known as the Kronecker delta

$$\delta[k]=\begin{cases}1 & k=0\\ 0 & k\neq 0\end{cases}$$

Applying Eq. (6.51b), the equalizer output sample values are

$$p_o[k+N]=\sum_{n=0}^{2N}c_{n-N}p_r[k+N-n] \qquad k=0,\ \pm1,\ \pm2,\ldots,\pm K.$$

The solution to this minimization problem can be better represented in matrix form as

$$\mathbf{c}=\mathbf{P}_r^{\dagger}\mathbf{p}_o$$

where $\mathbf{P}_r^{\dagger}$ represents the Moore-Penrose pseudo-inverse[8] of the nonsquare matrix $\mathbf{P}_r$ of size $(2K+1)\times(2N+1)$.

$$\mathbf{P}_r=\begin{bmatrix} p_r[N-K] & p_r[N-K-1] & \cdots & p_r[-N-K+1] & p_r[-N-K]\\ p_r[N-K+1] & p_r[N-K] & \cdots & p_r[-N-K+2] & p_r[-N-K+1]\\ \vdots & \ddots & \ddots & \vdots & \vdots\\ p_r[N+K-1] & p_r[N+K-2] & \ddots & p_r[-N+K] & p_r[-N+K-1]\\ p_r[N+K] & p_r[N+K-1] & \cdots & p_r[-N+K+1] & p_r[-N+K]\end{bmatrix} \tag{6.56}$$

The MMSE design often leads to a more robust equalizer for the reduction of ISI.

Example 6.5 For the received pulse $p_r(t)$ in Fig. 6.22b, let

$$\begin{aligned} p_r[0] &= 1 & p_r[k] &= 0 \qquad k \neq 0, \pm 1, \pm 2 \\ p_r[1] &= -0.3 & p_r[2] &= 0.1 \\ p_r[-1] &= -0.2 & p_r[-2] &= 0.05 \end{aligned}$$

Design a three-tap ($N = 1$) MMSE equalizer for $K = 3$ (window size of 7). Also determine the achieved MSE for this equalizer and compare against the achieved MSE by the ZF equalizer in Example 6.4.

Since $N = 1$ and $K = 3$, hence the MSE window specified in Eq. (6.55) is from $N - K = -2$ to $N + K = 4$. Construct the MMSE equation

$$\begin{bmatrix} p_o[-2] \\ p_o[-1] \\ p_o[0] \\ p_o[1] \\ p_o[2] \\ p_o[3] \\ p_o[4] \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \\ 0 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} 0.241 & 0 & 0 \\ -0.2 & 0.24 & 0 \\ 1 & -0.2 & 0.24 \\ -0.3 & 1 & -0.2 \\ 0.18 & -0.3 & 1 \\ 0 & 0.18 & -0.3 \\ 0 & 0 & 0.18 \end{bmatrix} \begin{bmatrix} c_{-1} \\ c_0 \\ c_1 \end{bmatrix} \tag{6.57}$$

Solution of this set yields $c_{-1} = 0.1526$, $c_0 = 1.0369$, and $c_1 = 0.2877$. According to Eq. (6.51b), This equalizer setting yields $p_o[k] = 0$ for $k \leq (N - K - 1) = -3$ and $k \geq (N + K + 1) = 5$. Moreover, for $N - K \leq k \leq N + K$, the output pulse samples are

$$\begin{aligned} &p_o[-2] = 0.0366 \quad p_o[-1] = 0.2183 \quad p_o[0] = 0.0142 \quad p_o[1] = 0.9336 \\ &p_o[2] = 0.0041 \quad p_o[3] = 0.1003 \quad p_o[4] = 0.0518 \end{aligned}$$

From Eq. (6.55), we find the minimized MSE as

$$\begin{aligned} \text{MSE} &= \frac{1}{7} \sum_{k=-2}^{4} (p_o[k] - \delta[k-1])^2 \\ &= 0.095 \end{aligned}$$

To compute the MSE obtained by using the ZF equalizer determined from Eq. (6.53), recall from Example 6.4 that

$$\begin{aligned} &p_o[-2] = 0.0355 \quad p_o[-1] = 0.2357 \quad p_o[0] = 0 \quad p_o[1] = 1 \quad p_o[2] = 0 \quad p_o[3] = 0.1075 \\ &p_o[4] = 0.0549 \quad p_o[k] = 0 \quad k = \cdots, -4, -3, 5, 6, \cdots \end{aligned}$$

Similarly using Eq. (6.55), we find the MSE of the ZF equalizer is in fact

$$\begin{aligned} \text{MSE} &= \frac{1}{7} \sum_{k=-2}^{4} (p_o[k] - \delta[k-1])^2 \\ &= 0.102 \end{aligned}$$

As expected, the MMSE design generates smaller MSE than the ZF design.

Adaptive Equalization and Other More General Equalizers

The equalizer filter structure that is described here has the simplest form. Practical digital communication systems often apply much more sophisticated equalizer structures and more advanced equalization algorithms.[6] Because of some additional probabilistic and statistical tools needed for clearer discussion, we will defer more detailed coverage on the specialized topics of equalization to Chapter 11.

6.5.2 Timing Extraction

The received digital signal needs to be sampled at decision instants for symbol detection. This requires a precise clock signal at the receiver in synchronism with the clock signal at the transmitter (**symbol** or **bit synchronization**), delayed by the channel response. There exist three general methods of synchronization:

1. Derivation from a primary or a secondary clock (e.g., transmitter and receiver slaved to a master timing source).
2. Transmission of a separate auxiliary synchronizing (pilot) clock for the receiver.
3. Self-synchronization, where the receiver extracts timing information from the modulated signal itself.

Because of its high cost, the first method is suitable for large volumes of data and high-speed communication systems. The second method, which uses part of the channel capacity and transmitter power to transmit the timing clock, is suitable when there are excess channel capacity and additional transmission power. The third method is the most efficient and commonly used method of timing extraction or clock recovery, which derives timing information from the modulated message signal itself. An example of the self-synchronization method will be discussed here.

We have already shown that a digital signal, such as an on-off signal (Fig. 6.2a), contains a discrete component of the clock frequency itself (Fig. 6.3c). Hence, when the on-off binary signal is applied to a resonant circuit tuned to the clock frequency, the output signal is the desired clock signal. However, not all baseband signals contain a discrete frequency component of the clock rate. For example, a bipolar signal has no discrete component of any frequency [see Eq. (6.21) or Fig. 6.9]. In such cases, it may be possible to extract timing by pre-processing the received signal with a **nonlinear device** to generate a frequency tone that is tied to the timing clock. In the bipolar case, for instance, a simple rectifier (that is nonlinear) can convert a bipolar signal to an on-off signal, which can readily be used to extract timing.

Small random deviations of the incoming pulses from their ideal location (known as **timing jitter**) are always present, even in the most sophisticated systems. Although the source emits pulses at the constant rate, channel distortions during transmission (e.g., Doppler shift) tend to cause pulses to deviate from their original positions. The Q-value of the tuned circuit used for timing extraction must be large enough to provide an adequate suppression of timing jitter, yet small enough to be sensitive to timing changes in the incoming signals. During those intervals in which there are no pulses in the input, the oscillation continues because of the flywheel effect of the high-Q circuit. But still the oscillator output is sensitive to the pulse pattern; for example, during a long string of **1**s, the output amplitude will increase, whereas during a long string of **0**s, it will decrease. This causes additional jitter in the timing signal extracted.

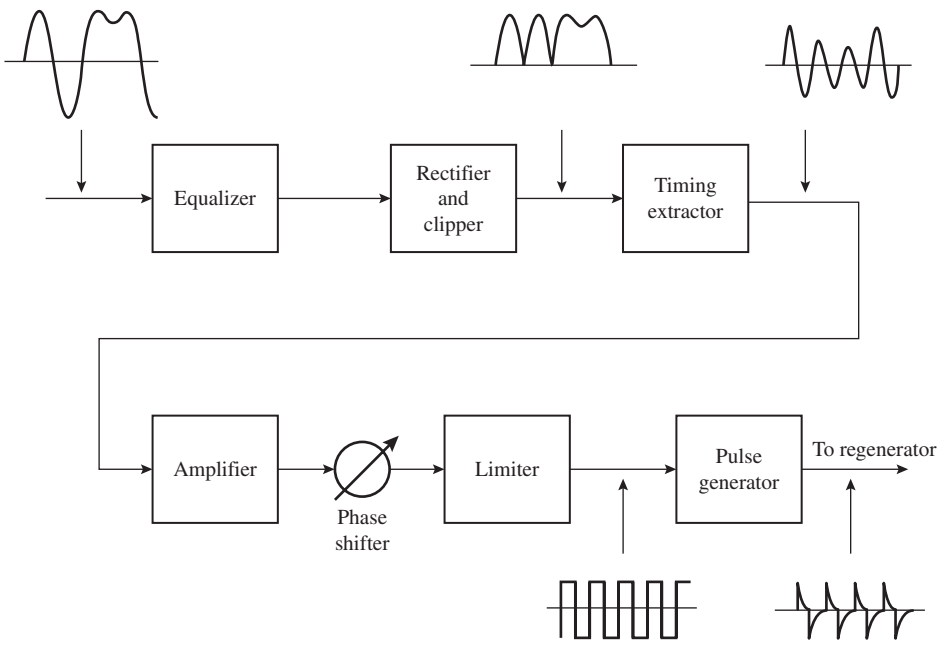

Figure 6.23 Timing extraction.

A complete timing extractor and time pulse generator circuit for a polar baseband transmission in shown in Fig. 6.23. In this circuit, the sinusoidal output of the oscillator (timing extractor) is passed through a phase shifter that adjusts the phase of the timing signal such that the timing pulses occur at the maximum sampling points of the baseband input signal for detection. This method is used to recover the clock at each of the regenerators in a PCM system. The jitter introduced by successive regenerators may accumulate, and after a certain number of regenerators, it is often necessary to use a regenerator with a more sophisticated clock recovery system such as a phase-locked loop.

6.5.3 Detection Error

Once the transmission has passed through the equalizer, detection can take place at the receiver that samples the received signal based on the clock provided by the timing extractor. The signal received at the detector consists of the equalized pulse train plus additive random channel noise. The noise can cause errors in pulse detection. Consider, for example, the case of polar transmission using a basic pulse $p(t)$ (Fig. 6.24a). This pulse has a peak amplitude A_p. A typical received pulse train is shown in Fig. 6.24b. Pulses are sampled at their peak values. If ISI and noise were absent, the sample of the positive pulse (corresponding to **1**) would be A_p and that of the negative pulse (corresponding to **0**) would be $-A_p$. By considering additive noise, these samples would be $\pm A_p + n$ where n is the random noise amplitude (see Fig. 6.24b). From the symmetry of the situation, the detection threshold is zero; that is, if the pulse sample value is positive, the digit is detected as **1**; if the sample value is negative, the digit is detected as **0**.

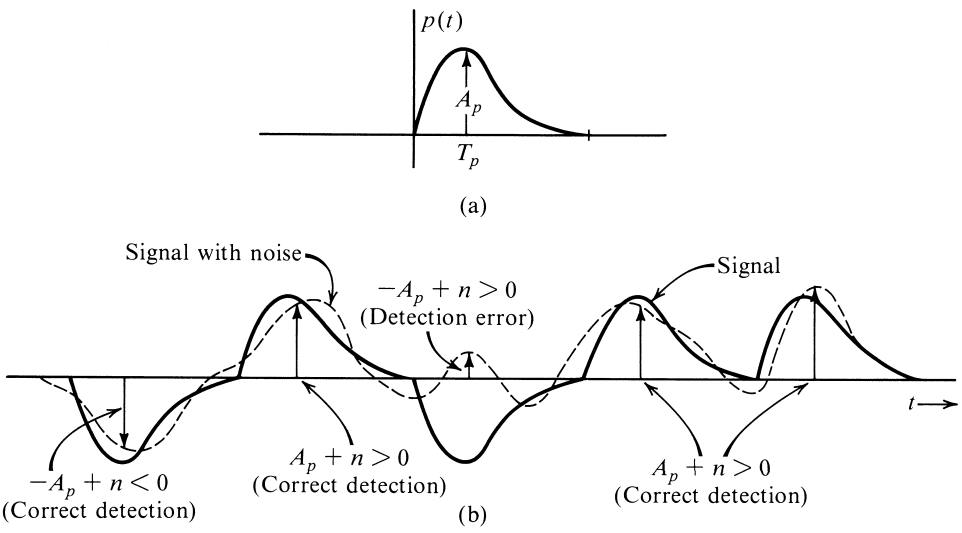

Figure 6.24
Error probability in threshold detection.

The detector's decision of whether to declare **1** or **0** could be made readily from the pulse sample, except that the noise value n is random, meaning that its exact value is unpredictable. The random noise n may have a large or a small value, and it can be negative or positive. It is possible that **1** is transmitted but n at the sampling instant has a large negative value. This may reverse the polarity of the sample value $A_p + n$, leading to the erroneous detection output of **0** instead. Conversely, if **0** is transmitted and n has a large positive value at the sampling instant, the sample value $-A_p + n$ can be positive and the digit will be detected wrongly as **1**. This is clear from Fig. 6.24b.

The performance of digital communication systems is typically specified by the average number of detection errors. For example, if two cellphones (receivers) in the same spot are attempting to receive the same transmission from a cell tower, the cellphone with the lower number of detection errors is the better receiver. It is more likely to have fewer dropped calls and less trouble receiving clear speech. However, because noise is random, sometimes one cellphone may have few errors while other times it may have many errors. The real measure of receiver performance is therefore the average ratio of the number of errors to the total number of transmitted data. Thus, the meaningful performance comparison is the likelihood of detection error, or the **detection error probability**.

Precise analysis and evaluation of this error likelihood require the knowledge and tools from probability theory. Thus, we will postpone error analysis until after the introduction of probability in Chapter 7 and Chapter 8. Later, in Chapter 9, we will discuss fully the error probability analysis of different digital communication systems for different noise models as well as system designs. For example, Gaussian noise can generally characterize the random channel noises from thermal effects and inter-system crosstalk. Optimum detectors can be designed to minimize the error probability against Gaussian noise. However, switching transients, sparks, power line load-switching, and other singular events cause very high level noise pulses of short duration against digital signals. These effects, collectively called **impulse noise**, cannot conveniently be engineered away, and they may lead to error bursts of up to several hundred bits at a time. To correct error burst, we use special **burst error correcting codes** described in Chapter 13.

6.6 EYE DIAGRAMS: AN IMPORTANT DIAGNOSTIC TOOL

In previous sections, we discussed the effect of noise and channel ISI on the detection of digital transmissions. We also described ways to design equalizers to combat channel distortion and explained the timing-extraction process. We now present a practical engineering tool known as the **eye diagram**. The eye diagram is easy to generate and is often applied by engineers on received signals. As a useful diagnostic tool, eye diagram makes it possible for visual inspection of the received signals to determine the severity of ISI, the accuracy of timing extraction, the noise immunity, and other important factors.

We need only a basic oscilloscope to generate the eye diagram. Given a baseband signal at the channel output

$$y(t) = \sum a_k p(t - kT_b)$$

it can be applied to the vertical input of the oscilloscope. The time base of the oscilloscope is triggered at the same rate $1/T_b$ as that of the incoming pulses, and it generates a sweep lasting exactly T_b, the interval of one transmitted data symbol a_k. The oscilloscope shows the superposition of many traces of length T_b from the channel output $y(t)$. What appears on the oscilloscope is simply the input signal (vertical input) that is cut up into individual pieces of T_b in duration and then superimposed on top of one another. The resulting pattern on the oscilloscope looks like a human eye, hence the name eye diagram. More generally, we can also apply a time sweep that lasts m symbol intervals, or mT_b. The oscilloscope pattern is simply the input signal (vertical input) that is cut up every mT_b in time before superposition. The oscilloscope will then display an eye diagram that is mT_b wide and has the shape of m eyes in a horizontal row.

We now present an example. Consider the transmission of a binary signal by polar NRZ pulses (Fig. 6.25a). Its eye diagrams are shown in Fig. 6.25b for the time base of T_b and $2T_b$, respectively. In this example, the channel has infinite bandwidth to pass the NRZ pulse and there is no channel distortion. Hence, we obtain eye diagrams with totally **open** eye(s). We can also consider a channel output using the same polar line code and a different (RZ) pulse shape, as shown in Fig. 6.25c. The resulting eye diagrams are shown in Fig. 6.25d. In this case, the eye is wide open only at the midpoint of the pulse duration. With proper timing extraction, the receiver should sample the received signal right at the midpoint where the eye is totally open, achieving the best noise immunity at the decision point (Sec. 6.5.3). This is because the midpoint of the eye represents the best sampling instant of each pulse, where the pulse amplitude is maximum without interference from any other neighboring pulse (zero ISI).

We now consider a channel that is distortive or has finite bandwidth, or both. After passing through this nonideal channel, the NRZ polar signal of Fig. 6.25a becomes the waveform of Fig. 6.25e. The received signal pulses are no longer rectangular but are rounded, distorted, and spread out. The eye diagrams are not fully open anymore, as shown in Fig. 6.25f. In this case, the ISI is not zero. Hence, pulse values at their respective sampling instants will deviate from the full-scale values by a varying amount in each trace, causing blurs. This leads to a partially closed eye pattern.

In the presence of additive channel noise, the eye will tend to close partially in all cases. Weaker noise will cause smaller amount of closing, whereas stronger noise can cause the eyes to be completely closed. The decision threshold with respect to which symbol (**1** or **0**) was

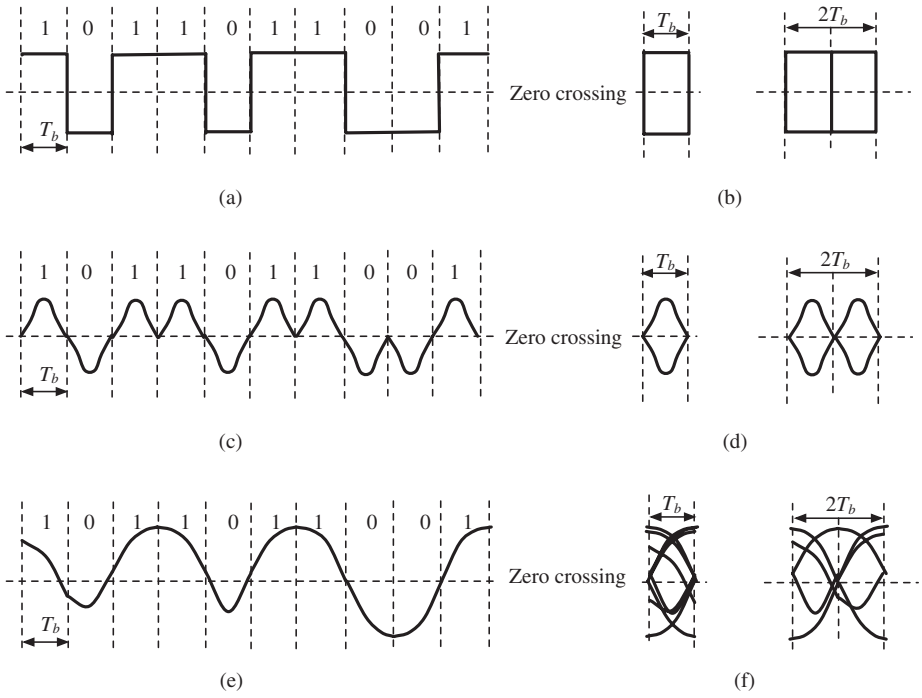

Figure 6.25 The eye diagram.

transmitted is the midpoint of the eye.* Observe that for zero ISI, the system can tolerate noise of up to half the vertical opening of the eye. Any noise value larger than this amount would cause a decision error if its sign is opposite to the sign of the data symbol. Because ISI reduces the eye opening, it clearly reduces noise tolerance. The eye diagram is also used to diagnostically determine optimum tap settings of the equalizer. Equalizer taps should be adjusted to obtain the maximum vertical and horizontal eye opening.

The eye diagram is a very effective tool for baseband signal diagnosis during real-time experiments. It not only is simple to generate, it also provides very rich and important information about the quality and vulnerability of the received digital signal. From the typical eye diagram given in Fig. 6.26, we can extract several key measures regarding the signal quality.

- *Maximum opening point.* The eye opening amount at the sampling and decision instant indicates what amount of noise the detector can tolerate without making an error. The quantity is known as the *noise margin*. The instant of maximum eye opening indicates the optimum sampling or decision-making instant.
- *Sensitivity to timing jitter.* The width of the eye indicates the time interval over which a correct decision can still be made, and it is desirable to have an eye with the maximum horizontal opening. If the decision-making instant deviates from the instant when the eye has a maximum vertical opening, the margin of noise tolerance is reduced. This causes higher error probability in pulse detection. The slope of the eye shows how fast the noise

* This is true for a two-level decision [e.g., when $p(t)$ and $-p(t)$ are used for **1** and **0**, respectively]. For a three-level decision (e.g., bipolar signaling), there will be two thresholds.

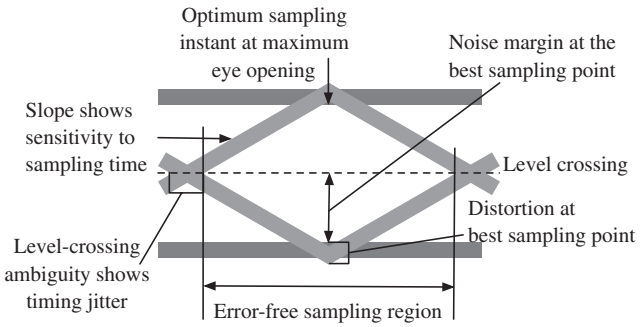

Figure 6.26 Reading an eye diagram.

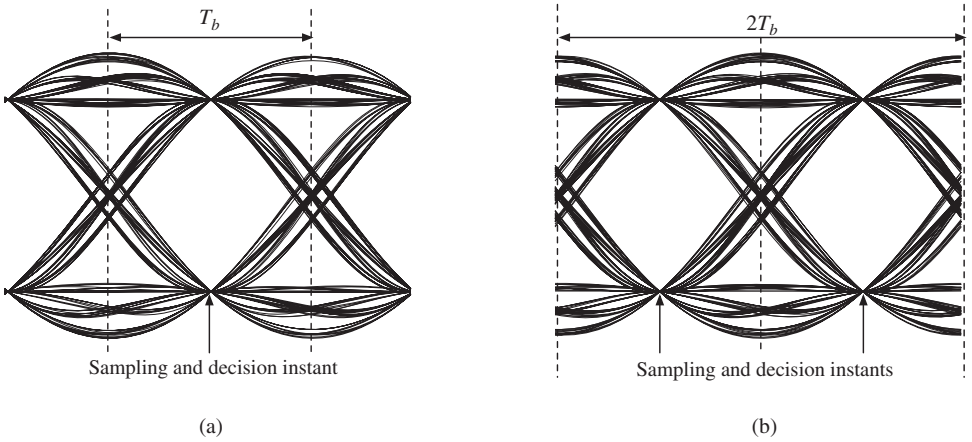

Figure 6.27 Eye diagrams of a polar signaling system using a raised cosine pulse with roll-off factor 0.5: (a) over 2 symbol periods $2T_b$ with a time shift $T_b/2$; (b) without time shift.

tolerance is reduced and, hence, so is the sensitivity of the decision noise tolerance to variation of the sampling instant. It demonstrates the sensitivity to timing jitter.

- *Level-crossing (timing) jitter.* Typically, practical receivers extract timing information about the pulse rate and the sampling clock from the (zero) level crossing of the received signal waveform. The variation of level crossing can be seen from the width of the eye corners. This measure provides information about the timing jitter such a receiver is expected to experience from its timing extractor.

Finally, we provide a practical eye diagram example for a polar signaling waveform. In this case, we select a raised cosine roll-off pulse that satisfies Nyquist's first criterion of zero ISI. The roll-off factor is chosen to be $r = 0.5$. The eye diagram is shown in Fig. 6.27 for a time base of $2T_b$. In fact, even for the same signal, the eye diagrams may be somewhat different for different time offset (or initial point) values. Figure 6.27a illustrates the eye diagram of this polar signaling waveform for a display time offset of $T_b/2$, whereas Fig. 6.27b shows the normal eye diagram when the display time offset value is zero. It is clear from comparison that these two diagrams have a simple horizontal circular shift relationship. By observing the maximum eye opening, we can see that this baseband signal has zero ISI, confirming the key advantage of the raised-cosine pulses. On the other hand, because Nyquist's first criterion places no requirement on the zero crossing of the pulse, the eye diagram indicates that timing jitter would be likely.