TABLE 9.2.1 Optimum Step Sizes for Uniform Quantization of a Gaussian PDF

Number of Levels (L)	Step Size (Δ_{opt})	Minimum Mean-Squared Error (D)	SQNR (dB)
4	0.9957	0.1188	9.25
8	0.5860	0.03744	14.27
16	0.3352	0.01154	19.38

Source: J. Max, "Quantizing for Minimum Distortion," *IRE Trans. Inf. Theory*, © 1960 IEEE.

9.3 Nonuniform Quantization

The most important nonuniform quantization method to date has been the logarithmic quantization used in the telephone network for speech digitization for over 20 years. The general idea behind this type of quantization is that for a fixed, uniform quantizer, an input signal with an amplitude less than full load will have a lower SQNR than a signal whose amplitude occupies the full dynamic range of the quantizer (but without overload). This fact is illustrated by Example 9.2.2. Such a variation in performance (SQNR) as a function of quantizer input signal amplitude is particularly detrimental for speech, since low-amplitude signals can be very important perceptually. There is the additional consideration for speech signals that low amplitudes are more probable than larger amplitudes, since speech is generally stated to have a gamma or Laplacian probability density, which is highly peaked about zero.

Therefore, for speech signals a type of nonlinear quantization was invented called *logarithmic companding*. Initially, this scheme was implemented by passing the analog speech signal through a characteristic of the form

$$F_\mu(s) = \frac{\ln[1 + \mu|s|]}{\ln[1 + \mu]} \operatorname{sgn}(s), \qquad -1 \le s \le 1, \tag{9.3.1}$$

where s is the normalized speech signal and μ is a parameter, usually selected to be $\mu = 100$, or more recently, $\mu = 255$. The function $F_\mu(s)$ is shown in Fig. 9.3.1. Notice that $F_\mu(s)$ tends to amplify small amplitudes more than larger amplitudes whenever $\mu > 0$. The output of $F_\mu(s)$ then served as input to a uniform, n-bit quantizer. To resynthesize the speech signal, the quantizer output $\hat{s}$ was passed through the inverse function of Eq. (9.3.1) given by

$$F_\mu^{-1}(\hat{s}) = \frac{1}{\mu}[(1 + \mu)^{|\hat{s}|} - 1] \operatorname{sgn}(\hat{s}), \tag{9.3.2}$$

where, of course, $-1 \le \hat{s} \le 1$.

The performance in SQNR of this system for $\mu = 100$ and $n = 7$ bits is shown in Fig. 9.3.2. It is evident from this figure that the SQNR is relatively flat over

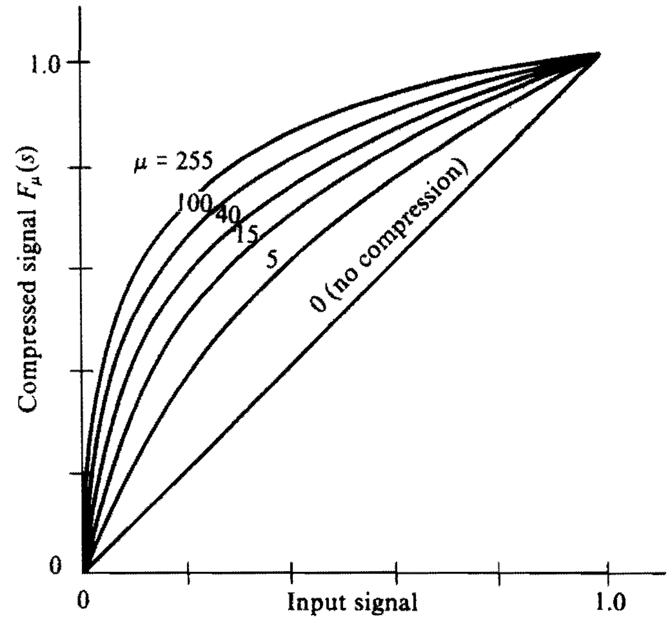

FIGURE 9.3.1 Logarithmic compression characteristics.

a wide dynamic range of input signal power (amplitudes), and hence low-amplitude signals are reproduced almost as well as higher-amplitude signals. In particular, we see from Eqs. (9.2.8) and (9.2.9) that for linear quantization, if we decrease the input signal power by 12 dB (from a peak value of $V/2$ to $V/8$), the output SQNR decreases by 12 dB. However, as we move along the "sine wave" curve in Fig. 9.3.2 from 0 to -12 dB on the input power axis, the SQNR decreases only by about 2 dB. The companding clearly improves the SQNR for low-amplitude signals.

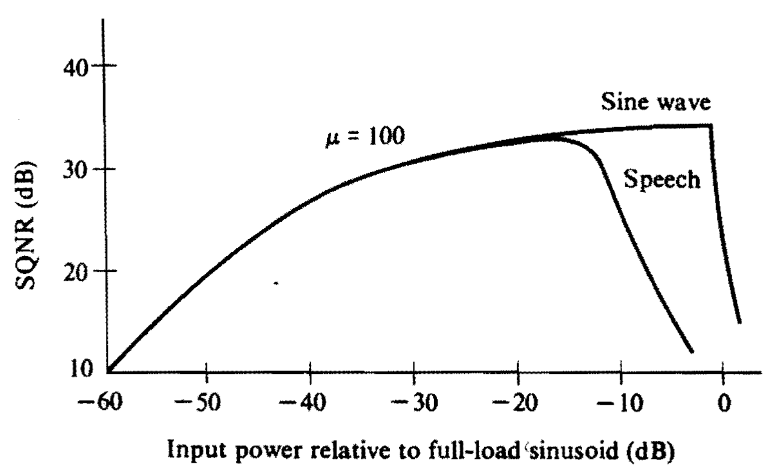

FIGURE 9.3.2 Performance of $\mu = 100$, $n = 7$ bit logarithmic companding.

EXAMPLE 9.3.1

To demonstrate further the utility of the logarithmic companding in Eq. (9.3.1), we consider the uniform 16-level quantizer designed for an input signal dynamic range of ± 10 V in Example 9.2.1. The resulting step size for this quantizer is thus 1.25 V. If we apply an input signal of amplitude 1.2 V to the quantizer, the quantization error is -0.575 V. However, this error should be reduced if we first pass the 1.2-V signal through $F_\mu(s)$ in Eq. (9.3.1), quantize this value, and then reconstruct the nonlinearly quantized output using Eq. (9.3.2).

Therefore, using Eq. (9.3.1) with $\mu = 255$ we have

$$F_\mu\left(\frac{1.2}{10}\right) = \frac{\ln[1 + 255|1.2/10|]}{\ln 256} \operatorname{sgn}\left(\frac{1.2}{10}\right) = 0.623.$$

The input to the uniform quantizer is thus $10(0.623) = 6.23$, so that the quantizer output is $9\Delta/2 = 5.625$. This value is then substituted as $\hat{s}$ into Eq. (9.3.2) so that the nonlinearly quantized output is 0.848 V for 1.2 V in. The quantization error in this case is $0.848 - 1.2 = -0.352$ V. Note that for the same low-level input voltage of 1.2 V, the quantization error is considerably less for the nonuniform quantizer (the one using companding).

A nonuniform quantizer can also be designed to match a given input probability density function. A general nonuniform midriser quantizer characteristic is shown in Fig. 9.3.3. The step points $x_0 = -\infty, x_1, x_2, \ldots, x_{L-1}, x_L = +\infty$, and the output levels $y_1, y_2, \ldots, y_L$, are constants that can be selected to minimize some function of the quantization error (noise) for a known or assumed input probability distribution. The parameter Δ is simply used for scaling, but it is still often called the step size, even though it cannot be strictly interpreted as such.

Upon letting the input step points be denoted by x_i, $i = 0, 1, 2, \ldots, L$, where $x_0 = -\infty$, $x_L = +\infty$, and letting the output values be denoted by y_i, $i = 1, 2, \ldots, L$, where y_1 is the most negative value, the distortion in Eq. (9.2.10) can be written as

$$D = \sum_{i=1}^{L} \int_{x_{i-1}}^{x_i} g[y_i - s] f_S(s)\, ds. \tag{9.3.3}$$

In Eq. (9.3.3), $\hat{s} = y_i$ if $x_{i-1} \le s < x_i$. For fixed L, we minimize D in Eq. (9.3.3) with respect to both the x_i and y_i. The necessary conditions to be satisfied are

$$g[y_j - x_j] = g[y_{j+1} - x_j], \tag{9.3.4}$$

$j = 1, 2, \ldots, L - 1$, and

$$\int_{x_{j-1}}^{x_j} g'[y_j - s] f_S(s)\, ds = 0, \tag{9.3.5}$$

for $j = 1, 2, \ldots, L$.

normalized and must be multiplied by the input standard deviation (Δ in Fig. 9.3.3) to get the correct step points and output levels. Similarly, the MSE is normalized and must be multiplied by the input variance to obtain the actual mean-squared error.

By comparing the minimum mean-squared errors and SQNRs between the uniform quantizers in Table 9.2.1 and the nonuniform quantizers in Tables 9.3.1 to 9.3.3, it is seen that the nonuniform quantizers generally provide better performance (smaller D, larger SQNR). The nonuniform quantizers may be slightly more complex to implement, however.

9.4 Quantization-Level Coding

Once we have quantized a continuous-amplitude signal sample to one of a finite number of output levels as described in Sections 9.2 and 9.3, we could proceed and transmit a pulse with this amplitude level directly over a communications channel. In practice, however, it is usual to represent each quantizer output level by a binary word, which tends to simplify and standardize interfaces for both communications and storage applications. The assignment of binary words to output quantization levels does not have a generally accepted name in the literature, and here we call it *quantization-level coding.*

Three common binary word assignments for an eight-level quantizer are given in Table 9.4.1 with reference to the quantizer characteristic in Fig. 9.4.1. The *natural binary code* (NBC) simply consists of the binary representations of the base-10 numbers 0, 1, . . . , $2^N - 1 = L - 1$, where L is the number of quantization levels. It is usual to start with 000 at the most negative level, but this is not mandatory.

TABLE 9.4.1 Binary Representations for an Eight-Level Quantizer

Level Number	Natural Binary Code (NBC)	Folded Binary Code (FBC)	Gray Code (GC)
1	000	011	010
2	001	010	011
3	010	001	001
4	011	000	000
5	100	100	100
6	101	101	101
7	110	110	111
8	111	111	110

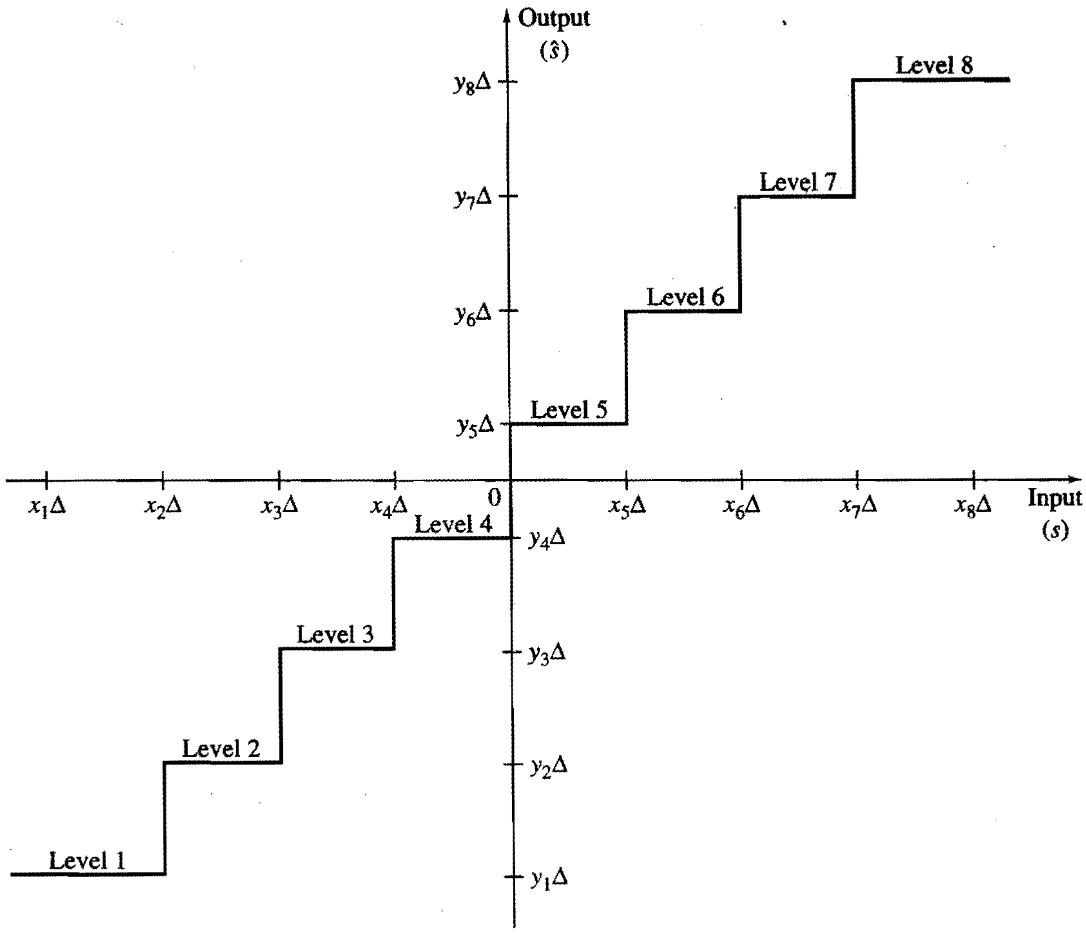

FIGURE 9.4.1 Eight-level symmetric quantizer.

The *folded binary code* (FBC) uses the most significant bit (MSB) or leftmost bit as a sign bit, and then counts out from the origin in both the positive and negative directions with the remaining bits. Since, exclusive of the sign bit, the codewords equidistant from the origin are identical, the FBC is also sometimes called the *sign–magnitude code.*

A third representation shown in Table 9.4.1 is the *Gray code* (GC). In this code, adjacent levels differ by only a single bit, and hence for equally likely bit errors in each position, a single channel bit error is more likely to produce an adjacent output level than in other codes. Of course, an error in the sign bit does not usually result in an adjacent output level (only at the origin).

The discussion thus far has emphasized quantizers with a number of output levels expressible as an integer power of 2. If we have a midtread quantizer (a zero level), there may be an odd number of levels, and of course, there is no rule which says that the number of output levels $L = 2^N$, where N is an integer. In these situations, one or more of the binary codewords are simply not used or they are dedicated to sending synchronization or some other type of signaling information. Which codewords are unused is not important in our present context; however, once we begin to assign symbols to the 1's and 0's, as we do in the next section, the use of one codeword over another may prove beneficial. We defer further discussion until then.

As a final note in this section, we mention that although the NBC and FBC are widely employed, they may not be the most efficient in terms of the average bit rate required. In fact, if the quantization levels are not equally likely (equally probable to occur), variable-length codes may provide a shorter average codeword length, and hence a lower transmitted bit rate. Such techniques are often called *entropy coding*, and the theory and applications of such codes are discussed in Chapter 11.

9.5 Transmission Line Coding

Once we have a sequence of 0's and 1's, whether the 0's and 1's are obtained as in Section 9.4 or are provided to us in some other manner, we must prepare these data for transmission to the receiver. If the data are to be sent over a single bandpass telephone channel, we would proceed as discussed in Chapter 8 (using a modem). However, if the binary information is to be transmitted in baseband or digital form directly, without analog modulation, we must assign a symbol or pulse to each 0 or 1 to be transmitted. This assignment is typically called *line coding*, or as we sometimes call it here, *transmission line coding*. The latter designation is an attempt to be slightly more specific, since many kinds of coding are discussed in this book. When assigning pulses to a sequence of 1's and 0's, we are often willing to work with rectangular pulses, but the key questions are: "How wide are the pulses in relation to the allocated pulse interval?" and "What are the polarities of the pulses?"

An intuitive form of line coding is to assign a pulse of width T seconds and amplitude $+V$ volts to each 1 and an amplitude of 0 V for T seconds to a 0. If the interval allocated to each bit is T seconds, this method is called *unipolar, nonreturn to zero* (NRZ) signaling. The "unipolar" refers to the fact that pulses of only a single polarity are used, and NRZ indicates that the assigned voltage level is maintained throughout the allocated pulse interval. On the other hand, a *polar*, NRZ line code might assign a pulse of amplitude $+V$ volts and width T seconds to a 1 and a pulse of $-V$ volts and width T seconds to a zero. If we continue to enumerate the possibilities, then, a polar, return to zero (RZ) line code might assign a $+V$ volts amplitude pulse of width $T/2$ to a 1 and a $-V$ volts pulse of width $T/2$ to a 0. In this case, the RZ case, the pulse is present only during half of the interval allocated to that particular bit. The voltage level is zero for the other half of the interval. Hence whatever voltage level or polarity a pulse is, it "returns to zero" during the pulse interval of T seconds. The RZ technique described uses pulses one-half the bit interval in width (50% duty cycle), which is the usual case. It is possible to have RZ pulses that occupy any portion of the T-second interval, as long as the pulse width is less than T.

The polar signaling scheme just described is designated as *bipolar* by some writers. However, we reserve the term *bipolar* coding for the alternate mark inversion method used in the digital T-carrier links in the telephone network. Specifically, this bipolar coding scheme uses 0 V for a zero, and alternating polarity, 50% duty cycle RZ pulses for 1's. There are particular advantages to this coding method, which we shall discuss shortly.

EXAMPLE 9.5.1

A binary sequence given by 1 0 1 1 0 0 0 1 is to be represented using (a) unipolar NRZ line coding, (b) polar RZ coding, and (c) bipolar coding. The results are shown in Fig. 9.5.1. In Fig. 9.5.1(a), we used an amplitude of $+V$ for a 1, but we could have also used a $+V$ for a 0 and 0 V for a 1, or we could have used negative polarities. For (a), however, the important things to note are that only one polarity is present and the pulses fill the bit interval.

Polar RZ coding is illustrated in Fig. 9.5.1(b), where $+V$ denotes 1 and $-V$ denotes a 0; of course, the polarities could be reversed. Since RZ pulses are employed, the pulses do not stay at $\pm V$ for the entire bit interval, but instead, drop back to 0 V. Although the pulses are shown as being in the first half of the bit interval, they could be located in the centers of the intervals, and often are.

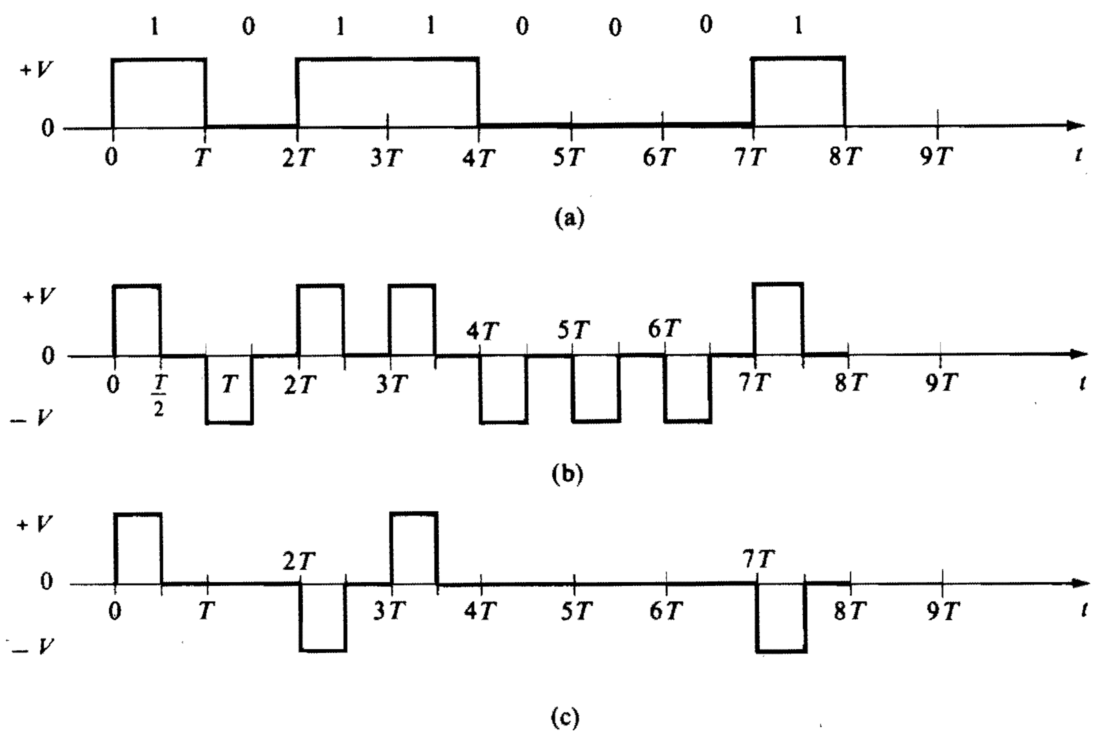

FIGURE 9.5.1 Line codes for Example 9.5.1: (a) Unipolar NRZ coding; (b) polar RZ coding; (c) bipolar (RZ) coding.

Bipolar (RZ) coding of the binary data is demonstrated by Fig. 9.5.1(c). Since there is no preceding 1, the first 1 is arbitrarily shown as a pulse of amplitude $+V$. A 0 is represented by 0 V or no pulse. The next 1 is indicated by a pulse opposite in polarity to the last preceding pulse—no matter how many 0's have occurred in between. Thus the second 1 is sent by a pulse of amplitude $-V$ volts. This process is continued to complete the example.

Clearly, the three line coding methods illustrated in Fig. 9.5.1 yield quite different transmitted signals for the given binary sequence, so the question arises as to how to select an appropriate line coding technique. Three principal considerations in making this selection are the spectrum of the line code, its synchronization properties, and error detection characteristics. Thus the unipolar NRZ code, although conceptually simple, has the disadvantages that there are no pulse transitions for long sequences of 0's or 1's, which are necessary if one wishes to extract timing or synchronization information, and that there is no way to detect when and if an error has occurred from the received pulse sequence. The polar RZ code guarantees the availability of timing information, but there is no error detection capability. The bipolar code has an error detection property, since if two pulses in a row (ignoring intervening 0's) are detected with the same polarity, it is evident that an error has occurred. However, to guarantee that timing data are available for the bipolar code, it is necessary to restrict the allowable number of consecutive 0's.

As we have mentioned, the spectrum of the transmitted line code can also be extremely important. The spectral density of a unipolar sequence, where a 1 is represented by a pulse and a 0 is represented by no pulse, is given by [see Eq. (A.10.13)]

$$S(\omega) = |P(\omega)|^2 \left\{ \frac{p(1-p)}{T} + 2\pi \frac{p^2}{T^2} \sum_{n=-\infty}^{\infty} \delta\left(\omega - \frac{2n\pi}{T}\right) \right\}, \tag{9.5.1}$$

where $P(\omega)$ is the Fourier transform of the pulse shape, p the probability of a pulse (a 1), and $1/T$ the symbol rate. Equation (9.5.1) holds for any $P(\omega)$. A sketch of $S(\omega)$ for rectangular RZ pulses of width $T/2$ and with $p = \frac{1}{2}$ is shown in Fig. 9.5.2. We see from this figure that there is a significant dc component and that most of the energy in the sequence lies at frequencies below twice the symbol rate.

The spectral density of the bipolar code is quite different and can be shown to be (see Problem 9.11)

$$S(\omega) = \frac{2p(1-p)}{T} |P(\omega)|^2 \left\{ \frac{1 - \cos \omega T}{1 + 2(2p-1)\cos \omega T + (2p-1)^2} \right\}, \tag{9.5.2}$$

where $P(\omega)$, p, and T are as given for Eq. (9.5.1). The spectral density in Eq. (9.5.2) is shown in Fig. 9.5.3 for rectangular pulses of width $T/2$ and several values of p. There are significant differences between Figs. 9.5.2 and 9.5.3. First,

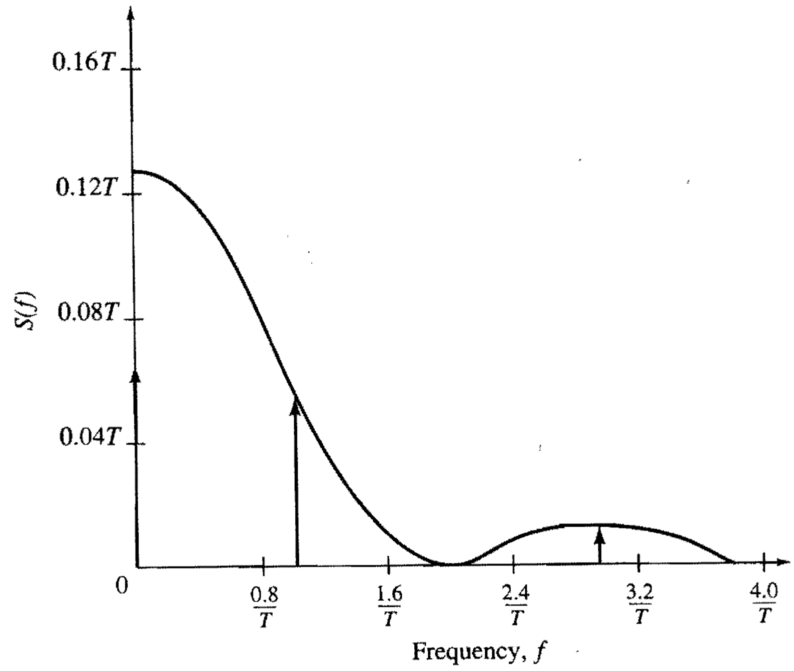

FIGURE 9.5.2 Spectral content of a unipolar RZ Sequence (pulse width = $T/2$ and $p = 1/2$).

the bipolar code has zero frequency content at dc. Second, the bipolar code has its energy concentrated at frequencies below the symbol rate. (Energy above the symbol rate is not shown in Fig. 9.5.3.) Third, there are no discrete components in Fig. 9.5.3. It is thus evident that the selection of a line code can have a considerable impact on the characteristics of the transmitted signal.

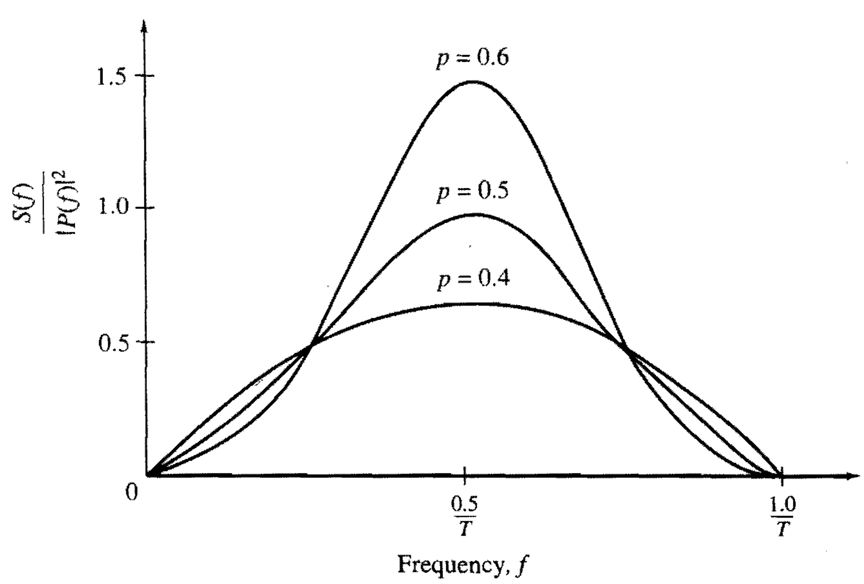

FIGURE 9.5.3 Spectral content of the bipolar code.

In addition to those already discussed, there are several other commonly used line codes. A class of codes currently used in the telephone network is the class of binary with N zero substitution (BNZS) codes. As we noted previously, if the bipolar code is used, care must be taken to prevent long strings of zeros from occurring, since timing information would be unavailable. The BNZS codes are bipolar codes that are modified to replace strings of N zeros with special sequences that contain pulses which cause bipolar violations. The bipolar violations allow the special sequences to be identified at the receiver, so that the N zeros can be reinserted; the presence of the special code during transmission increases the number of pulses, so that timing extraction is easier.

The B6ZS line code is an important specific example of the BNZS line coding procedure. If there are no sequences of 0's longer than five, the B6ZS line code is just the bipolar code described previously. However, when a sequence of six 0's occurs, these six 0's are removed and replaced with one of two possible six bit sequences. If the last pulse before the six 0's is positive, the code substituted for the six 0's is $0 + -0 - +$, where $+$ denotes a positive 50% duty cycle RZ pulse, $-$ denotes a similar negative pulse, and 0 indicates "no pulse." On the other hand, if the last pulse before the six 0's is negative, the substituted code is $0 - +0 + -$. Note that these substitutions guarantee (if there are no channel errors) that bipolar violations occur in the second and fifth bit positions, thus allowing the substituted code to be detected at the receiver and replaced with six 0's.

Another important line code is the B3ZS format. In the B3ZS format, the sequence of binary 1's and 0's are encoded by the straight bipolar code except when a string of three consecutive 0's occurs. The three 0's are represented by one of two codes, B0V or 00V, where B denotes a pulse satisfying the bipolar rule, 0 indicates no pulse, and V indicates a pulse that violates the bipolar rule. The choice between these two codes is made such that the number of pulses satisfying the bipolar rule between violations is odd. The bipolar violation after an odd number of pulses satisfying the bipolar rule allows the receiver to detect the substituted code and decode it as three 0's.

EXAMPLE 9.5.2

In this example we wish to illustrate the use of B6ZS and B3ZS line codes and to contrast them with the standard bipolar code introduced previously. Rather than sketch pulses explicitly here, we use a "+" to denote a positive, 50% duty cycle RZ pulse, a "−" to indicate a negative, 50% duty cycle RZ pulse, and a 0 to represent no pulse. We assume that we are given the binary data sequence 10100000000011000110, and we wish to represent this sequence using the bipolar, B6ZS, and B3ZS line codes. The results are shown in Table 9.5.1.

Notice that for all three codes we must specify the polarity of the last pulse representing a 1. Further, for the B3ZS code we must know whether the number of pulses satisfying the bipolar rule since the last violation is odd or even. In Table 9.5.1 we assumed that an even number has occurred.

TABLE 9.5.1 Bipolar, B6ZS, and B3ZS Line Codes for Example 9.5.2

Binary input data	1	0	1	0	0	0	0	0	0	0	1	1	0	0	0	1	1	0
Bipolar code (last 1 a −)	+	0	−	0	0	0	0	0	0	0	+	−	0	0	0	+	−	0
B6ZS (last 1 a −)	+	0	−	0	−	+	0	+	−	0	+	−	0	0	0	+	−	0
B3ZS (last 1 a − and even number of bipolar pulses since last violation)				B	0	V	B	0	V				B	0	V			
	+	0	−	+	0	+	−	0	−	0	+	−	+	0	+	−	+	0

If we had started with the assumption of an odd number, the B3ZS code in Table 9.5.1 would be different. As a further note, it should be recognized that for the B6ZS format, the nonzero pulses are the same as in a straight bipolar coding of the sequence; however, for B3ZS this may not be true. As an example, compare the final three pulses of the bipolar, B6ZS, and B3ZS codes in Table 9.5.1.

A class of line codes closely related to the binary with N zero substitution codes is the high-density bipolar codes. The most important of these codes is the HDB3 format, which has been adopted as an international standard. The HDB3 code uses bipolar coding whenever possible, but when strings of four 0's occur, they are encoded as either B00V or 000V, where the choice between these two is made (as in B3ZS) such that the number of pulses between bipolar violations is odd.

Another line coding format of some importance, due to its application in some local area networks called Ethernet, is *digital biphase coding* or *Manchester coding*. The primary attractions of a digital biphase code are that it contains a strong timing component and zero dc spectral content. In a digital biphase code, one cycle of a square wave with a particular phase is used to designate a 1 and one cycle of the square wave with the opposite phase is used to indicate a 0. Since there is a transition at the center of every symbol interval, a strong timing signal is available for synchronization. Additionally, since both 1's and 0's are represented by one cycle of a square wave, there is no dc component. Two disadvantages of the digital biphase code are that it has no redundancy for performance monitoring and it requires a wider bandwidth than other line codes, such as bipolar coding. Its principal applications are thus where bandwidth is not at a premium.

There are numerous other line coding formats that have various advantages and have practical applications. Of these, the pair selected ternary (PST) and the 4B3T format come to mind. The PST code maps pairs of binary digits (bits) into two ternary digits, while the 4B3T code maps 4 bits into three ternary digits. Development and discussion of these codes are left to the problems.

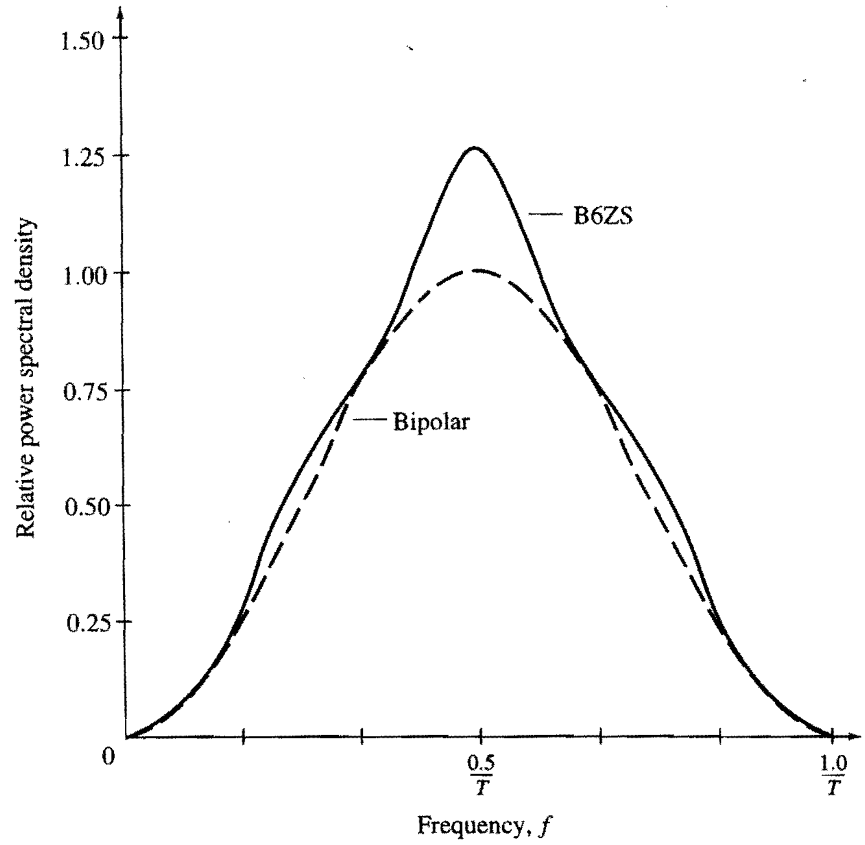

FIGURE 9.5.4 Spectra of bipolar and B6ZS line codes.

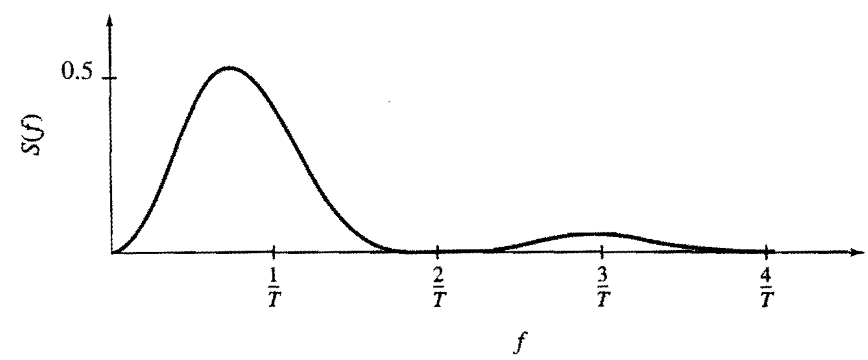

FIGURE 9.5.5 Spectrum of the digital biphase code.

Of course, depending on the transmission medium involved, the spectral content of the various line codes can be of the utmost importance. To close this section, then, we present (without derivation), the spectral densities of several line codes with equally likely 1's and 0's. Figure 9.5.4 shows the spectral content of bipolar, B6ZS, and the PST line codes, and Fig. 9.5.5 illustrates the spectrum of the digital biphase code.

9.6 Codecs and Channel Banks

At the present time, speech digitization is most often accomplished in the telephone network by devices called *codecs* (*co*ders/*dec*oders), which are contained in terminals designated as D-type *channel banks.* The codecs transform the analog speech into 8-bit PCM form, and the channel banks combine the PCM version of numerous voice channels into a single data stream using *time-division multiplexing* (TDM). In this section we briefly describe a few important details of the (U.S.) standard PCM codec and the various channel banks.

We begin by considering the codec used in the D2, D3, and D4 channel banks. This codec bandpass filters the analog voice signal to 200 to 3400 Hz and samples the filtered signal at a rate of 8000 samples/sec. Each of these samples is then quantized to 8-bit accuracy using a nonlinear quantizer based on the μ-law logarithmic characteristic discussed in Section 9.3. In particular, the quantizer is based on a 16-segment piecewise linear approximation to the $\mu = 255$ logarithmic companding characteristic in Eq. (9.3.1). There are eight positive and eight negative segments, but since the two segments around zero are collinear, it is often referred to as a 15-segment approximation. There are 16 equal quantization steps for each segment.

Table 9.6.1 specifies the nonlinear quantizer characteristic assuming that the maximum magnitude is scaled to 8159. The code for each quantization level is also shown in the table. For the 8-bit representation, the first bit is a polarity bit (1 denotes positive values, 0 represents negative values), the next 3 bits indicate the segment number, and the final 4 bits designate the particular step within a segment. Note from Table 9.6.1 that the segment codes and quantization steps for each segment are binary numbers that proceed from largest to smallest as the magnitude increases. Since lower amplitudes occur more often for speech than larger amplitudes, this tends to increase the density (number) of 1's. This is useful because the bipolar transmission line code described in Section 9.5 is employed with these codecs, and thus the number of pulses is increased. The higher density of pulses provides a strong timing component and aids synchronization.

In fact, another constraint is placed on the code assigned to quantizer output levels in Table 9.6.1. If an input sample falls within the most negative range of input amplitudes (quantization bin), the table indicates that the all 0's codeword would be transmitted. However, to guarantee a certain density of 1's, and hence bipolar pulses, the all 0's code is replaced by the codeword 00000010. Although it would cause less of an error to replace the all 0's codeword by 00000001, this is not done, for reasons that will be explained shortly.

Note now that we sampled the analog input voice signal (after filtering) at a rate of 8000 samples/sec, and thus if we pass each sample through the quantizer represented by Table 9.6.1, we get a bit rate of (8000 samples/sec) × (8 bits/sample) = 64,000 bits/sec or 64 kbits/sec for each voice channel. Referring to Fig. 9.5.4, we see that for the bipolar code and $1/T = 64{,}000$, the required bandwidth is 64 kHz. Thus, by using PCM (the codec), we have expanded

TABLE 9.6.1 Quantizer Characteristic and Code Assignment for D2, D3, and D4 Channel Bank Codecs[a]

Input Amplitude Range:	Step Size:	Polarity Bit:	Quantization Segment Code:	Quantizer Step Code:	Output Value:
0–1	1	1	111	1111	0
1–3				1110	2
3–5	2	1	111	1101	4
⋮				⋮	⋮
29–31				0000	30
31–35				1111	33
⋮	4	1	110	⋮	⋮
91–95				0000	93
95–103				1111	99
⋮	8	1	101	⋮	⋮
215–223				0000	219
223–239				1111	231
⋮	16	1	100	⋮	⋮
463–479				0000	471
479–511				1111	495
⋮	32	1	011	⋮	⋮
959–991				0000	975
991–1055				1111	1023
⋮	64	1	010	⋮	⋮
1951–2015				0000	1983
2015–2143				1111	2079
⋮	128	1	001	⋮	⋮
3935–4063				0000	3999
4063–4319				1111	4191
⋮	256	1	000	⋮	⋮
7903–8159				0000	8031

[a] Positive inputs only; assumed symmetric about zero.

the bandwidth required by a single voice channel from approximately 4 kHz (3.4 kHz at the 3-dB point) to 64 kHz. We must be getting something in return for this extra bandwidth, and in the development in the remainder of this section and in Section 9.7, the primary advantages of PCM transmission are pointed out.

Before proceeding further, it is necessary to define clearly what is meant by time-division multiplexing and how it is used in the digital transmission of speech. In frequency-division multiplexing (FDM), we are given a specified band of frequencies, and we allocate nonoverlapping portions of this band to several different messages or channels. In TDM, we are given a specified time interval, and we allot nonoverlapping subintervals of this larger time slot to binary codewords generated by different codecs.

As a specific example of TDM, we mention what is called the T1 carrier system, which combines 24 PCM voice channels into a single data stream. Since each channel requires 8 bits, we thus have 192 bits for each set of 24 voice channels, usually called a *frame*. One bit is added to this total for synchronization purposes, so that 193 bits are transmitted per frame. Since each voice channel produces a new (8-bit) binary word 8000 times/sec, our frame rate is 8000 frames/sec and the transmitted bit rate for T1 carrier systems is 1.544 Mbits/sec.

Other than synchronization in the usual sense, the added "framing" bit has another purpose. When a telephone call is placed, there is a requirement for what is called *control signaling* information, such as "on-hook" or "off-hook" signals. For T1 carrier systems (with D2, D3, or D4 channel banks) this information is carried in the least significant bit position of each PCM word in every sixth frame. Therefore, in every sixth frame, the samples representing the 24 voice channels have only 7 bits of accuracy, while the intervening five frames use the full 8 bits. As a result, the PCM systems are sometimes said to employ $7\frac{5}{6}$-bit encoding. Since the least significant bit is sometimes "stolen" for signaling, this is why the most negative quantizer output level is encoded as 00000010 rather than 00000001. We would have gained nothing (in terms of a higher density of 1's) if the latter code appeared in the sixth frame.

The T1 systems constitute the first level (or lowest rate) in what is called the *digital hierarchy* in the telephone network. In recent years the various rates (or levels) in the digital hierarchy have been given DS designations, as shown in Table 9.6.2 and Fig. 9.6.1. From Fig. 9.6.1 we see that two DS1 links can be combined to form one DS1C link, four DS1 lines can be combined to form one DS2 link, or 28 DS1 links can be combined to form one DS3 link. Finally, six DS3 links can be combined to form one DS4 signal. Starting with the DS1 rate, which TDMs 24 voice channels in a frame, the method used to combine

TABLE 9.6.2 Data Rates and Line Codes in the Digital Hierarchy

Signal	Bit Rate (Mbits/sec)	Line Code
DS1	1.544	Bipolar
DS1C	3.152	Bipolar
DS2	6.312	B6ZS
DS3	44.736	B3ZS
DS4	274.176	Polar

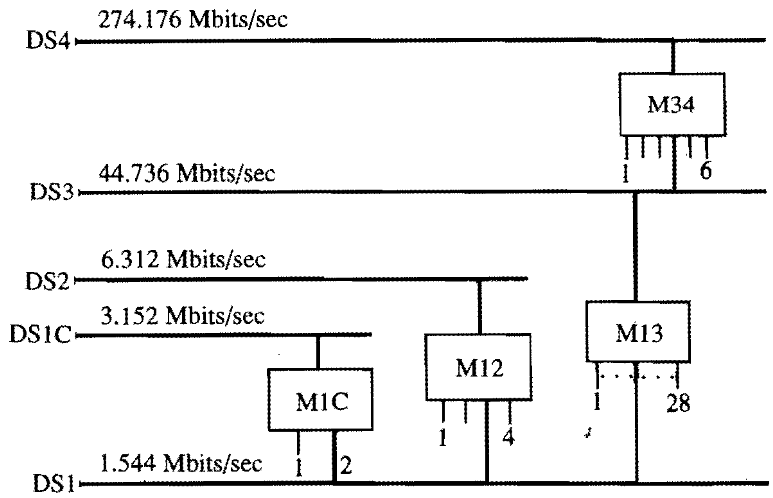

FIGURE 9.6.1 Digital hierarchy multiplexing plan.

all these signals is time-division multiplexing. Note also that in forming the next-highest level of the hierarchy, there seems to be some bits left over; that is, combining four DS1 signals should require 1.544 Mbits/sec $\times$ 4 = 6.176 Mbits/sec, but a DS2 line uses 6.312 Mbits/sec. The "extra" 0.136 Mbit/sec is used for synchronization and framing information, so that the multiplexing can be "undone."

It is also pointed out that as shown in Table 9.6.2, the line codes change as the data rate is increased, except in going from DS1 to DS1C signals. This is because the specifications on timing extraction, dc wander, and hardware change as the bit rate is increased. The line codes designated in Table 9.6.2 should be familiar to the reader from Section 9.5.

9.7 Repeaters

A principal advantage of employing time-division multiplexing of PCM signals to transmit analog waveforms such as speech is that PCM signals can be transmitted, in theory, over any distance without *any* degradation by noise simply by employing devices called *regenerative repeaters.* The concept that forms thebasis for repeaters can be explained as follows. Given a particular line code, only a finite number of fixed levels are allowable. For example, the bipolar line code has allowable levels of $+V$, $-V$, or 0 in each pulse interval, and no other voltage values are used to designate a 1 or 0. As these pulses proceed through a transmission medium (say, a pair of wires), they are smeared by deterministic distortion (amplitude and time delay) and they are subjected to additive noise as well as other random impairments. If the pair of wires is long enough, the pulses will eventually overlap so much and become so distorted that their original identity is irretrievably lost. If, however, we detect these pulses while the three amplitude levels are still distinguishable and they are not too smeared out into adjacent pulse intervals, we can regenerate the original pulse sequence exactly.

Therefore, regenerative repeaters are placed at specified intervals along a time-division multiplexed line to detect and retransmit the desired pulse sequence. The four main functions of these repeaters are equalization, clock recovery, pulse detection, and retransmission. The equalization first corrects for deterministic distortion that is on the line. Next, clock (timing) recovery is accomplished, so that the incoming pulses can be sampled at the center of the pulse intervals, and later, accurate timing is required to resynthesize the pulse sequence. The samples taken from the pulse intervals are used to detect the presence or absence of a pulse, and then the "undistorted" pulse sequence is regenerated and retransmitted. A block diagram of a regenerative repeater is shown in Fig. 9.7.1. The maximum spacing of the repeaters is determined such that the pulses can be reliably detected and the sequence accurately regenerated.

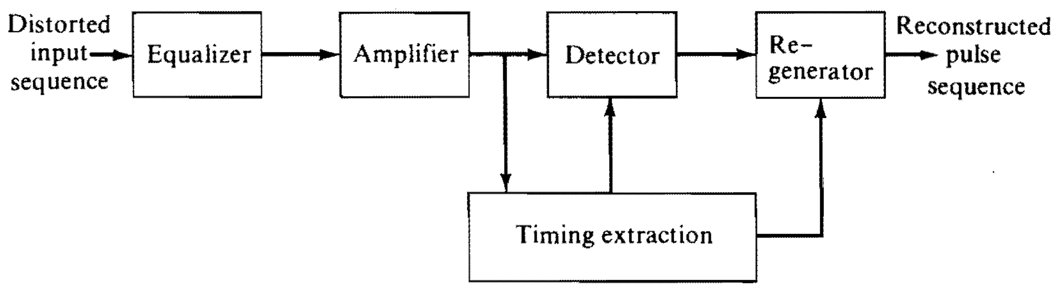

FIGURE 9.7.1 Regenerative repeater block diagram.

Theoretically, it is possible using PCM with TDM over repeatered lines to transmit the original PCM sequence undistorted over any distance. In practice, what occurs is that clock recovery is difficult, and timing inaccuracies from several repeaters begin to accumulate. The resulting distortion, called *timing jitter*, causes bit errors to occur eventually. However, the use of repeaters with PCM is much different from using amplifiers with analog signals. This is because the analog amplifiers not only increase the amplitude of the desired signal, but also amplify in-band noise. The repeaters with PCM are able to eliminate the noise, since only a few amplitude levels are allowable, and thus these levels can be detected and regenerated.

9.8 International Standards

The μ-law quantization characteristic given by Eq. (9.3.1) and the digital (PCM) TDM hierarchy described in Section 9.6 are used in the United States, Japan, and Canada. However, there are other systems implemented in other parts of the world that we mention briefly here. An alternative to the μ-law characteristic discussed in Section 9.3 is the A-law characteristic given by

$$F_A(s) = \begin{cases} \left[\dfrac{A|s|}{1 + \ln A}\right] \operatorname{sgn}(s), & 0 \le |s| \le \dfrac{1}{A} \\ \left[\dfrac{1 + \ln |As|}{1 + \ln A}\right] \operatorname{sgn}(s), & \dfrac{1}{A} \le |s| \le 1, \end{cases} \tag{9.8.1}$$

where $0 \le |s| \le 1$ and $A = 87.6$. The inverse to this characteristic is

$$F_A^{-1}(\hat{s}) = \begin{cases} \dfrac{|\hat{s}|[1 + \ln A]}{A} \operatorname{sgn}(\hat{s}), & 0 \le |\hat{s}| \le \dfrac{1}{1 + \ln A} \\ \dfrac{\exp\{|\hat{s}|[1 + \ln A] - 1\}}{A} \operatorname{sgn}(\hat{s}), & \dfrac{1}{1 + \ln A} \le |\hat{s}| \le 1, \end{cases} \tag{9.8.2}$$

where, if there is no quantization involved, $\hat{s} = F_A(s)$. The primary differences between the A-law and μ-law characteristics are that the A-law has a slightly wider dynamic range but the μ-law has a little better (less) idle channel noise.

The A-law quantizing characteristic is implemented in Europe, Africa, Australia, and South America, and in these countries, a digital hierarchy different from that in Fig. 9.6.1 is used. In particular, thirty 64-kbits/sec voice channels are combined with two 64-kbits/sec channels used for synchronization and signaling to obtain a first-level TDM data rate of 2.048 Mbits/sec. Successive levels of the hierarchy then proceed as shown in Fig. 9.8.1. Just as for the U.S.

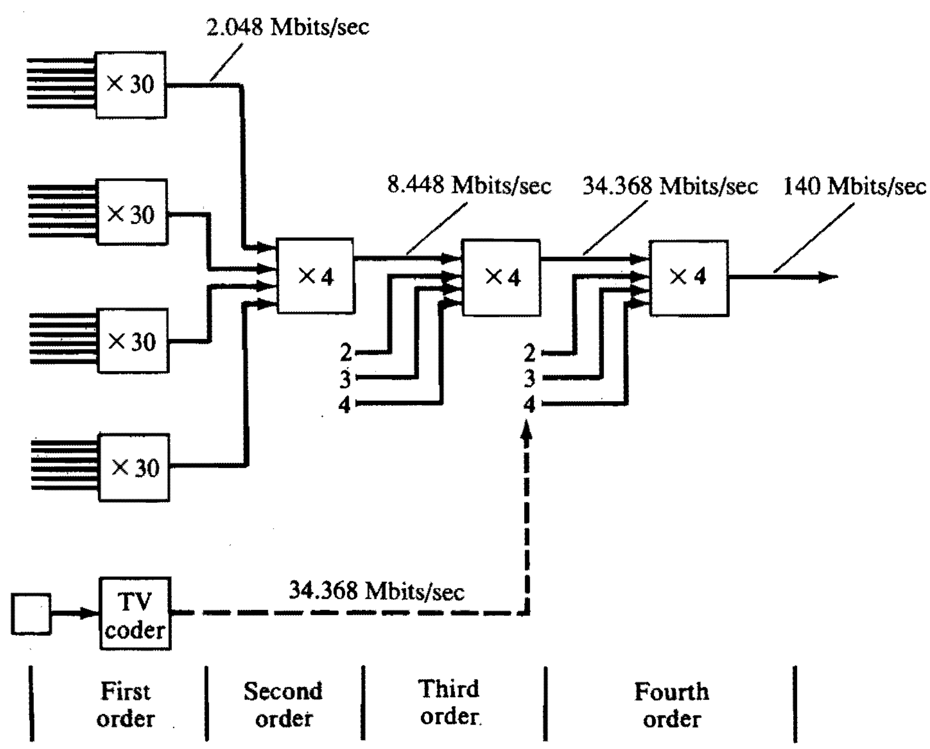

FIGURE 9.8.1 Digital hierarchy for Europe, Africa, Australia, and South America.

digital hierarchy, there are many other interesting details to the functioning of the system, but due to lack of space, they are not covered here. Some elements of A-law codecs and the hierarchy in Fig. 9.8.1 are covered in the problems at the end of the chapter.

SUMMARY

In this chapter we have briefly surveyed the various theoretical, functional, and applied aspects of pulse code modulation. PCM encoding consists of filtering, sampling, quantizing, assigning a level code, assigning a line code, and time-division multiplexing several channels into a chosen frame or block size for transmission. Nonlinear quantization of signals can produce substantial reductions in the required data rate with no loss in voice quality and intelligibility, as is demonstrated by the success of 8 bits/sample μ-law and A-law quantizer characteristics. The assignment of binary codewords to quantization levels and the choice of a line code are affected by several practical considerations, such as synchronization requirements, the spectral content of the line code, and the desired error monitoring capabilities. PCM signals offer the advantages of transmission over long distances with nominal distortion, digital switching, and compatibility with ever-increasing digital traffic. As a result, digital transmission networks based upon time-division multiplexed PCM signals are now an integral part of telecommunications systems throughout the world.

PROBLEMS

9.1 Design a 15-level uniform midtread quantizer for an input signal with a dynamic range of ± 10 V. Find the quantizer output value and the quantization error for an input signal amplitude of 1.2 V.

9.2 Plot the signal-to-quantization noise ratio in Eq. (9.2.7) for $L = 2, 3, 4, 5, 6, 7, 8, 12, 16, 24$, and 32.

9.3 The signal-to-quantization noise ratio expression in Eq. (9.2.7) is exact only for full-load sinusoidal inputs. Derive an equivalent expression if the input signals have the form of $s(t)$ in Fig. P9.3. Assume that the quantization noise is uniformly distributed in $[-\Delta/2, \Delta/2]$. Repeat Problem 9.2 for this new SQNR expression.

9.4 Given a uniform n-bit quantizer designed for a full-load sine wave with a peak value of $V/2$, plot the SQNR of this quantizer for sine-wave inputs with less than a full-load amplitude. What happens to the SQNR expression if the sine-wave input is greater than full load?

9.26 Repeat Example 9.3.1 using A-law companding with $A = 87.6$.

9.27 Table P9.27 shows the quantizer characteristic and code assignments for an $A = 87.6$ law codec. Repeat Problems 9.18, 9.19, and 9.20 for this characteristic.

TABLE P9.27 Quantizer Characteristic and Code Assignment for the Segmented A-Law Codec

Input Amplitude Range	Step Size	Quantization Segment Code	Quantizer Step Code	Output Value
0–2			1111	1
2–4		111	1110	3
⋮			⋮	⋮
30–32	2		0000	31
32–34			1111	33
⋮		110	⋮	⋮
62–64			0000	63
64–68			1111	66
⋮	4	101	⋮	⋮
124–128			0000	126
128–136			1111	132
⋮	8	100	⋮	⋮
248–256			0000	252
256–272			1111	264
⋮	16	011	⋮	⋮
496–512			0000	504
512–544			1111	528
⋮	32	010	⋮	⋮
992–1024			0000	1008
1024–1088			1111	1056
⋮	64	001	⋮	⋮
1984–2048			0000	2016
2048–2176			1111	2112
⋮	128	000	⋮	⋮
3968–4096			0000	4032

9.28 Repeat Problem 9.16 for the A-law characteristic in Table P9.27.

9.29 Repeat Problem 9.17 for the A-law characteristic in Table P9.27.

9.30 With reference to Fig. 9.8.1, what are the excess bit rates allocated to frame signaling and synchronization at each stage in the European hierarchy?