CHAPTER

TWELVE

CODED PULSE MODULATION

Having studied the basic concepts of digital transmission in Chap. 11, we return once more to analog communication. But now we consider *digital* transmission of *analog* messages via *code pulse modulation.* Coded pulse modulation systems employ sampling, quantizing, and coding to convert analog waveforms into digital signals.

Digital coding of analog information produces a *rugged* signal with a high degree of immunity to transmission distortion, interference, and noise. Digital coding also allows the use of regenerative repeaters for long-distance analog communication. However, the quantizing process essential for digital coding results in *quantization noise* which becomes the fundamental limitation on waveform reconstruction. To keep the quantization noise small enough for suitable fidelity, a coded pulse modulation system generally requires a much larger bandwidth than a comparable analog transmission system.

We'll develop these properties first in conjunction with *pulse-code modulation* (PCM). Then we'll describe *delta modulation* (DM) and other schemes that involve *predictive coding.* The chapter closes with an introduction to *digital multiplexing,* a valuable technique that makes it possible to combine analog and digital information for transmission in the form of a multiplexed digital signal.

12.1 PULSE-CODE MODULATION

This section describes the functional operation of pulse-code modulation (PCM). We'll view PCM as a digital transmission system with an *analog-to-digital converter* (ADC) at the input and a *digital-to-analog converter* (DAC) at the output.

When the digital error probability is sufficiently small, PCM performance as an *analog* communication system depends primarily on the quantization noise introduced by the ADC. Here we'll analyze analog message reconstruction with quantization noise, temporarily deferring to the next section the effects of random noise and digital errors.

PCM Generation and Reconstruction

Figure 12.1-1*a* diagrams the functional blocks of a PCM generation system. The analog input waveform $x(t)$ is lowpass filtered and sampled to obtain $x(kT_s)$. A *quantizer* rounds off the sample values to the nearest discrete value in a set of q *quantum levels.* The resulting quantized samples $x_q(kT_s)$ are discrete in time (by virtue of sampling) and discrete in amplitude (by virtue of quantizing).

To display the relationship between $x(kT_s)$ and $x_q(kT_s)$, let the analog message be a voltage waveform normalized such that $|x(t)| \le 1$ V. Uniform quantization subdivides the 2-V peak-to-peak range into q equal steps of height $2/q$ V, as shown in Fig. 12.1-1*b*. The quantum levels are then taken to be at $\pm 1/q$, $\pm 3/q, \ldots, \pm(q-1)/q$ in the usual case when q is an even integer. A quantized value such as $x_q(kT_s) = 5/q$ corresponds to any sample value in the range $4/q < x(kT_s) < 6/q$.

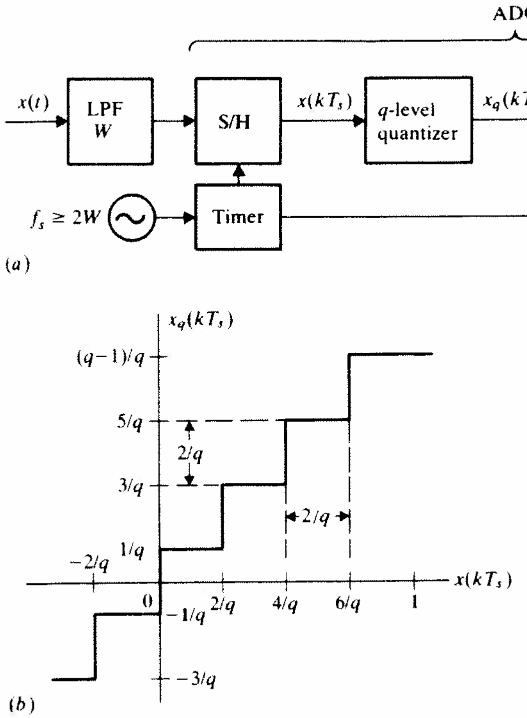


Figure 12.1-1 (*a*) PCM generation system; (*b*) quantization characteristic.

Next, an *encoder* translates the quantized samples into *digital code words*. The encoder works with M-ary digits and produces for each sample a codeword consisting of ν digits in parallel. Since there are M^ν possible M-ary codewords with ν digits per word, unique encoding of the q different quantum levels requires that $M^\nu \geq q$. The parameters M, ν, and q should be chosen to satisfy the equality, so that

$$q = M^\nu \qquad \nu = \log_M q \tag{1}$$

Thus, the number of quantum levels for *binary* PCM equals some power of 2, namely $q = 2^\nu$.

Finally, successive codewords are read out serially to constitute the PCM waveform, an M-ary digital signal. The PCM generator thereby acts as an ADC, performing analog-to-digital conversions at the sampling rate $f_s = 1/T_s$. A timing circuit coordinates the sampling and parallel-to-serial readout.

Each encoded sample is represented by a ν-digit output word, so the signaling rate becomes $r = \nu f_s$ with $f_s \geq 2W$. Therefore, the bandwidth needed for PCM baseband transmission is

$$B_T \geq \tfrac{1}{2}r = \tfrac{1}{2}\nu f_s \geq \nu W \tag{2}$$

Fine-grain quantization for accurate reconstruction of the message waveform requires $q \gg 1$, which increases the transmission bandwidth by the factor $\nu = \log_M q$ times the message bandwidth W.

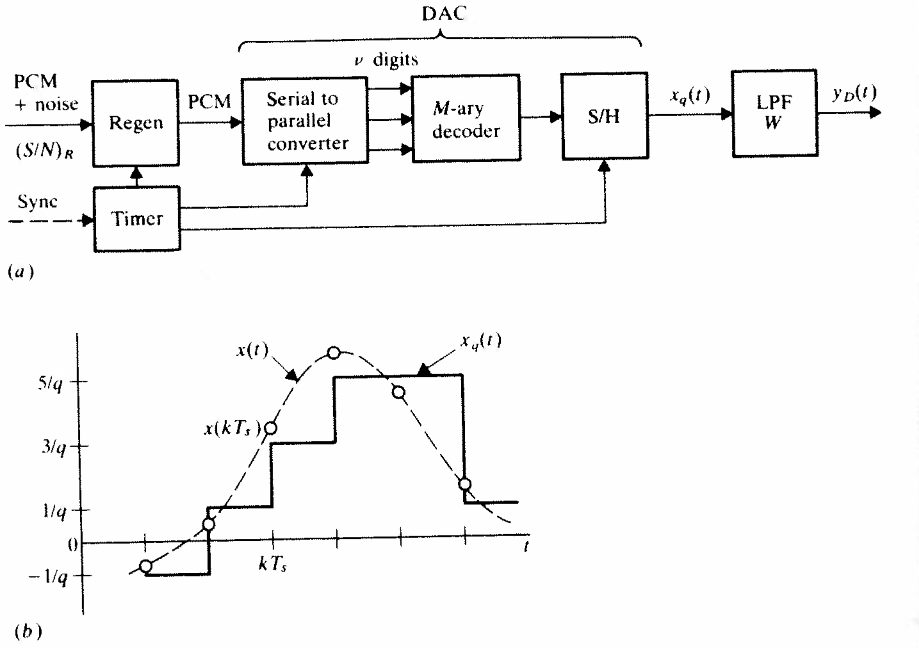
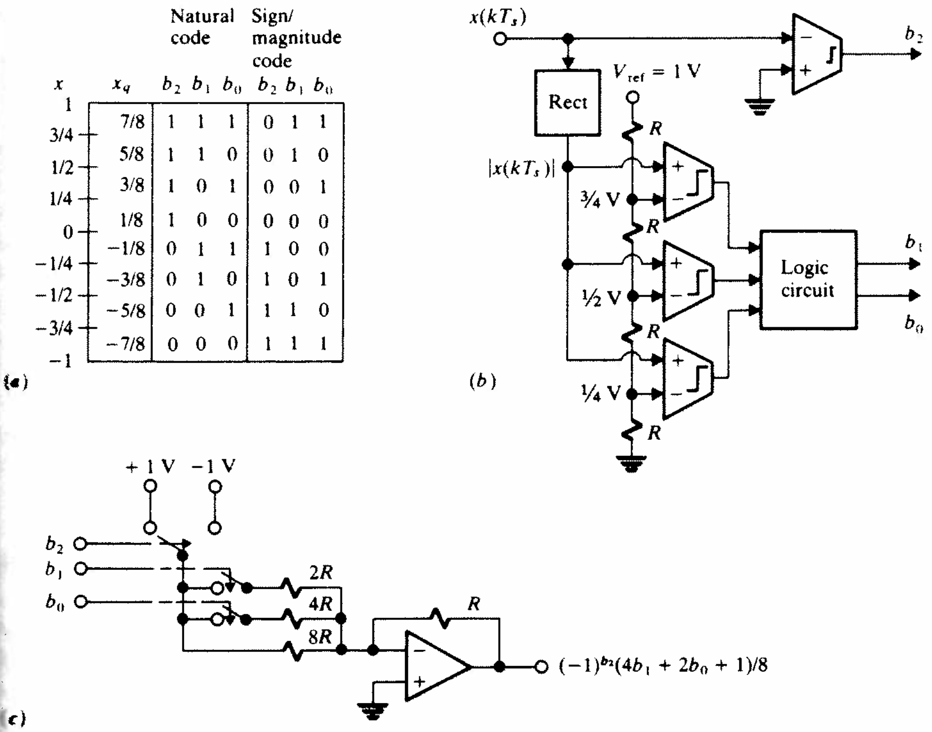


Figure 12.1-2 (*a*) PCM receiver; (*b*) reconstructed waveform.

Now consider a PCM receiver with the reconstruction system in Fig. 12.1-2*a*. The received signal may be contaminated by noise, but regeneration yields a clean and nearly errorless waveform if $(S/N)_R$ is sufficiently large. The DAC operations of serial-to-parallel conversion, M-ary decoding, and sample-and-hold generate the analog waveform $x_q(t)$ drawn in Fig. 12.1-2*b*. This waveform is a "staircase" approximation of $x(t)$, similar to flat-top sampling except that the sample values have been quantized. Lowpass filtering then produces the smoothed output signal $y_D(t)$, which differs from the message $x(t)$ to the extent that the quantized samples differ from the exact sample values $x(kT_s)$.

Perfect message reconstruction is therefore impossible in PCM, even when random noise has no effect. The ADC operation at the transmitter introduces permanent errors that appear at the receiver as quantization noise in the reconstructed signal. We'll study this quantization noise after an example of PCM hardware implementation.

Example 12.1-1 Suppose you want to build a binary PCM system with $q = 8$ so $\nu = \log_2 8 = 3$ bits per codeword. Figure 12.1-3*a* lists the 8 quantum levels and two types of binary codes. The "natural" code assigns the word 000 to



	Natural code			Sign/magnitude code		
x_q	b_2	b_1	b_0	b_2	b_1	b_0
7/8	1	1	1	0	1	1
5/8	1	1	0	0	1	0
3/8	1	0	1	0	0	1
1/8	1	0	0	0	0	0
−1/8	0	1	1	1	0	0
−3/8	0	1	0	1	0	1
−5/8	0	0	1	1	1	0
−7/8	0	0	0	1	1	1

Figure 12.1-3 (*a*) Binary PCM codes for $q = 8$; (*b*) direct-conversion ADC circuit for sign/magnitude code; (*c*) weighted-resistor decoder circuit.

the lowest level and progresses upward to 111 in the natural order of binary counting. The sign/magnitude code uses the leading bit b_2 for the algebraic sign of x_q while the remaining bits $b_1 b_0$ represent the magnitude. Other encoding algorithms are possible, and may include additional bits for error protection — the topic of Chap. 13.

A direct-conversion ADC circuit for the sign/magnitude code is shown in Fig. 12.1-3*b*. This circuit consists of one comparator for the sign bit and three parallel comparators plus combinational logic to generate the magnitude bits. Direct-conversion ADCs have the advantage of high operating speed and are called *flash encoders*, but they require a total of $q/2$ comparators. At lower speeds you can get by with one comparator and a feedback loop, a configuration found in the dual-slope, counter-comparison, and successive-approximation encoders.

Figure 12.1-3*c* shows the circuit for a *weighted-resistor decoder* that goes with a 3-bit sign/magnitude code. The sign bit operates a polarity selector switch, while the magnitude bits control the resistors to be connected to the reference voltage. The overall circuit acts as an inverting op-amp summer with output voltage $(-1)^{b_2}(4b_1 + 2b_0 + 1)/8$. Further information about DAC and ADC circuit modules is given by Mitra (1980, chap. 8).

Exercise 12.1-1 A binary channel with $r_b = 36{,}000$ bits/sec is available for PCM voice transmission. Find appropriate values of v, q, and f_s assuming $W \approx 3.2$ kHz.

Quantization Noise

Although PCM reconstruction most often takes the form of staircase filtering, as in Fig. 12.1-2, we'll find the *impulse reconstruction* model in Fig. 12.1-4 more convenient for the analysis of quantization noise. Here, a pulse converter in place of the sample-and-hold circuit generates the weighted impulse train

$$y_\delta(t) = \sum_k \left[x(kT_s) + \epsilon_k\right] \delta(t - kT_s) \tag{3a}$$

where ϵ_k represents the *quantization error*, namely

$$\epsilon_k = x_q(kT_s) - x(kT_s) \tag{3b}$$

Lowpass filtering with $B = f_s/2$ yields the final output

$$y_D(t) = x(t) + \sum_k \epsilon_k \operatorname{sinc}(f_s t - k) \tag{4}$$

This expression has the same form as reconstruction of analog pulse modulation with noisy samples; see Eq. (6), Sect. 10.4. Furthermore, when q is large enough for reasonable signal approximation, the ϵ_k will be uncorrelated and independent of $x(t)$. Accordingly, we identify $\overline{\epsilon_k^2}$ as the mean-square *quantization noise*.

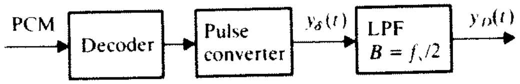


Figure 12.1-4 Impulse reconstruction model.

Round-off quantization with equispaced levels ensures that $|\epsilon_k| \le 1/q$. Lacking additional information to the contrary, we assume that the quantization error has zero mean value and a uniform probability density function over $-1/q \le \epsilon_k \le 1/q$. Thus, the quantization noise power is

$$\sigma_q^2 = \overline{\epsilon_k^2} = \frac{1}{(2/q)} \int_{-1/q}^{1/q} \epsilon^2 \, d\epsilon = \frac{1}{3q^2} \tag{5}$$

which reflects the intuitive observation that the quantization noise decreases when the number of quantum levels increases.

Now we measure PCM performance in terms of the destination signal power $S_D = \overline{x^2} = S_x \le 1$ and the quantization noise power σ_q^2. The destination signal-to-noise ratio then becomes

$$\left(\frac{S}{N}\right)_D = \frac{S_x}{\sigma_q^2} = 3q^2 S_x \tag{6}$$

A more informative relation for binary PCM is obtained by setting $q = 2^v$ and expressing $(S/N)_D$ in decibels. Thus,

$$\begin{aligned}\left(\frac{S}{N}\right)_D &= 10 \log_{10}(3 \times 2^{2v} S_x) \\ &\le 4.8 + 6.0v \qquad \text{dB}\end{aligned} \tag{7}$$

where the upper bound holds when $S_x = 1$. Voice telephone PCM systems typically have $v = 8$ so $(S/N)_D \le 52.8$ dB.

But many analog signals — especially voice and music — are characterized by a large *crest factor*, defined as the ratio of peak to rms value, $|x(t)|_{\max}/\sigma_x$. Our signal normalization establishes $|x(t)|_{\max} \le 1$, and a large crest factor then implies that $S_x = \sigma_x^2 \ll 1$. Consequently, the actual signal-to-noise ratio will be significantly less than the theoretical upper bound. For instance, some digital audio recording systems take $v = 14$ to get high-fidelity quality with $(S/N)_D \approx 60$ dB, compared to $(S/N)_D \le 88.8$ dB predicted by Eq. (7). As a bandwidth-conserving alternative to increasing v, PCM performance may be improved through the use of *companding*, equivalent to *nonuniform quantization*.

Exercise 12.1-2 Consider binary PCM transmission of a video signal with $f_s = 10$ MHz. (*a*) Calculate the signaling rate needed to get $(S/N)_D \ge 50$ dB when $S_x = 1$. (*b*) Repeat (*a*) with $S_x = 0.1$.

Nonuniform Quantizing and Companding★

A normalized signal $x(t)$ with a large crest factor can be represented by the typical probability density function $p_X(x)$ sketched in Fig. 12.1-5. The even sym-

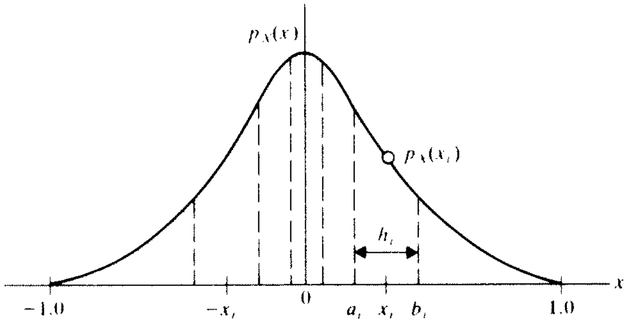


Figure 12.1-5 Message PDF with nonuniform quantization bands

metry and absence of an impulse at $x = 0$ correspond to $\bar{x} = 0$, so $\sigma_x^2 = \overline{x^2} = S_x$ and

$$S_x = \int_{-1}^{1} x^2 p_X(x)\,dx = 2\int_0^1 x^2 p_X(x)\,dx \tag{8}$$

This integration yields $S_x \ll 1$ because the PDF has a dominant peak at $x = 0$.

The shape of $p_X(x)$ also means that $|x(t)| \ll 1$ most of the time. It would therefore make good sense to use *nonuniform* quantization as indicated by the dashed lines. The quantum levels $\pm x_1, \ldots, \pm x_{q/2}$ are closely spaced near $x = 0$, but more widely spaced for the large values of $|x(t)|$ which occur infrequently. We calculate the resulting quantization noise as follows.

Consider a sample value $x = x(kT_s)$ in the band $a_i < x < b_i$ around the quantum level x_i. The quantizing error $\epsilon_i = x_i - x$ then has the mean square value

$$\overline{\epsilon_i^2} = \int_{a_i}^{b_i} (x_i - x)^2 p_X(x)\,dx \tag{9a}$$

Summing $\overline{\epsilon_i^2}$ over all q levels gives the quantization noise

$$\sigma_q^2 = 2\sum_{i=1}^{q/2} \overline{\epsilon_i^2} \tag{9b}$$

where we've taken advantage of the even symmetry. In the usual case of $q \gg 1$, the step height $h_i = b_i - a_i$ will be small enough that $p_X(x) \approx p_X(x_i)$ over each integration band and x_i will fall roughly in the middle of the step. Under these conditions Eq. (9*a*) simplifies to

$$\overline{\epsilon_i^2} \approx p_X(x_i)\int_{x_i - h_i/2}^{x_i + h_i/2} (x_i - x)^2\,dx = p_X(x_i)\,\frac{h_i^3}{12}$$

and thus

$$\sigma_q^2 \approx \frac{1}{6}\sum_{i=1}^{q/2} p_X(x_i) h_i^3 \tag{10}$$

As a check on this expression, we note that if the signal has the uniform PDF $p_X(x) = 1/2$ and if the steps have equal height $h_i = 2/q$, then $\sigma_q^2 = (1/6)(q/2)(1/2)(2/q)^3 = 1/3q^2$ which agrees with our earlier result in Eq. (5).

Theoretically, you could *optimize* PCM performance by finding the values of x_i, a_i, and b_i that result in *minimum quantization noise.* Such optimization is a difficult procedure that requires knowledge of the signal's PDF. Additionally, the custom-tailored hardware needed for nonlinear quantizing costs far more than standard uniform quantizers. Therefore, the approach taken in practice is to use *uniform* quantizing after *nonlinear signal compression*, the compression characteristics being determined from experimental studies with representative signals.

Figure 12.1-6 plots an illustrative compressor curve $z(x)$ versus x for $0 \le x \le 1$; the complete curve must have odd symmetry such that $z(x) = -z(|x|)$ for $-1 \le x \le 0$. Uniform quantization of $z(x)$ then corresponds to nonuniform quantization of x, as shown in the figure. The nonlinear distortion introduced by the compressor is corrected after reconstruction by a complementary expander, identical to the *companding* strategy discussed in Sect. 3.2. Hence, the postdetection signal-to-noise ratio for companded PCM is $(S/N)_D = S_x/\sigma_q^2$, with σ_q^2 given by Eq. (10).

Our next task is to obtain σ_q^2 in terms of the compressor curve. For that purpose let

$$z'(x) \triangleq \frac{dz(x)}{dx}$$

so $z'(x_i)$ equals the slope of $z(x)$ at $x = x_i$. The conditions $q \gg 1$ and $h_i \ll 1$ justify the approximation $z'(x_i) \approx (2/q)/h_i$, so

$$h_i^2 \approx \left[\frac{2/q}{z'(x_i)}\right]^2 = \frac{4}{q^2[z'(x_i)]^2}$$

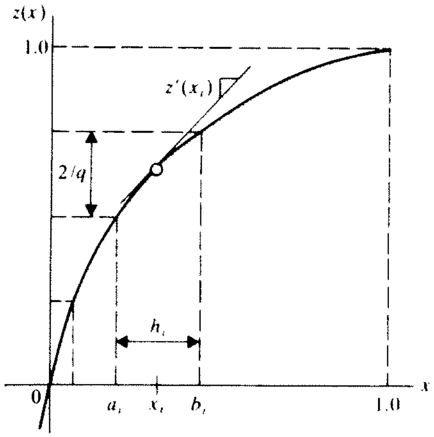


Figure 12.1-6 Compressor characteristic curve.

Equation (10) then becomes

$$\sigma_q^2 \approx \frac{4}{6q^2} \sum_{i=1}^{q/2} \frac{p_X(x_i)}{[z'(x_i)]^2} h_i \approx \frac{2}{3q^2} \int_0^1 \frac{p_X(x)}{[z'(x)]^2}\, dx$$

where we've passed from summation to integration via $h_i \rightarrow dx$. Therefore, our final result is

$$\left(\frac{S}{N}\right)_D = \frac{S_x}{\sigma_q^2} \approx \frac{3q^2 S_x}{K_z} \qquad (11a)$$

with

$$K_z \triangleq 2 \int_0^1 \frac{p_X(x)}{[z'(x)]^2}\, dx \qquad (11b)$$

which takes account of the compression. If $K_z < 1$ then $(S/N)_D > 3q^2 S_x$ and companding improves PCM performance by reducing the quantization noise.

Example 12.1-2 μ-law companding for voice PCM The popular μ-law companding for voice telephone PCM employs the compressor characteristic

$$z(x) = (\operatorname{sgn} x) \frac{\ln(1 + \mu|x|)}{\ln(1 + \mu)} \qquad |x| \le 1$$

and hence

$$z'(x) = \frac{\mu}{\ln(1 + \mu)} \frac{1}{1 + \mu|x|}$$

The parameter μ is a large number so $z'(x) \gg 1$ for $|x| \ll 1$ whereas $z'(x) \ll 1$ for $|x| \approx 1$. Substituting $z'(x)$ into Eq. (11*b*) and performing the integration, we obtain

$$K_z = \frac{\ln^2(1 + \mu)}{\mu^2} (1 + 2\mu\overline{|x|} + \mu^2 S_x) \qquad (12)$$

Now we need values for $\overline{|x|}$ and S_x to test the efficacy of μ-law companding.

Laboratory investigations have shown that the PDF of a voice signal can be modeled by a *Laplace* distribution in the form

$$p_X(x) = \frac{\alpha}{2} e^{-\alpha|x|} \qquad (13a)$$

with

$$S_x = \sigma_x^2 = \frac{2}{\alpha^2} \qquad \overline{|x|} = \frac{1}{\alpha} = \sqrt{\frac{S_x}{2}} \qquad (13b)$$

This distribution cannot be normalized for $|x(t)|_{\max} \le 1$, but the probability of $|x(t)| > 1$ will be less than 1% if $S_x < 0.1$. Other voice PDF models yield about the same relationship between $\overline{|x|}$ and S_x, which is the critical factor for evaluating K_z.

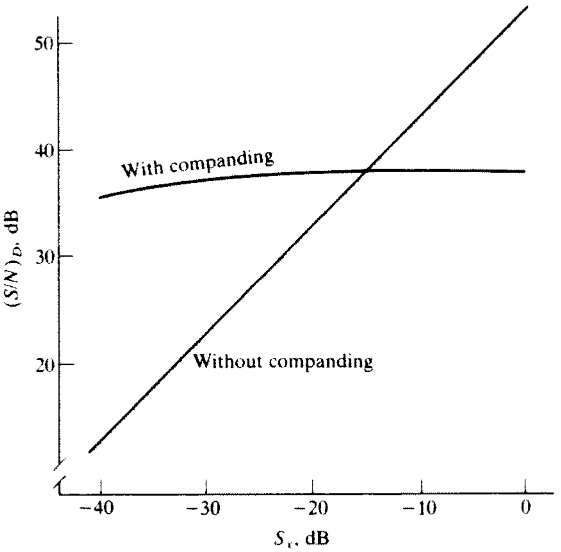


Figure 12.1-7 PCM performance with μ-law companding.

Taking the standard value $\mu = 255$ and putting $\overline{|x|} = \sqrt{S_x/2}$ in Eq. (12), we obtain

$$K_z = 4.73 \times 10^{-4}(1 + 361\sqrt{S_x} + 65{,}025 S_x)$$

Numerical calculations then show that $K_z < 1$ for $S_x < 0.03$. But more significant is the fact that S_x/K_z stays nearly constant over a wide range of S_x. Consequently, μ-law companding for voice PCM provides an essentially *fixed* value of $(S/N)_D$, despite wide variations of S_x among individual talkers. Figure 12.1-7 brings out this desirable feature by plotting $(S/N)_D$ in dB versus S_x in dB, with and without companding, when $q = 2^8$. Notice the companding improvement for $S_x < -20$ dB.

12.2 PCM WITH NOISE

In this section we account for the effects of random noise in PCM transmission. The resulting *digital errors* produce *decoding noise.* After defining the error threshold level, we'll be in a position to make a meaningful comparison of PCM with analog modulation methods.

Decoding Noise

Random noise added to the PCM signal at the receiver causes regeneration errors that appear as erroneous digits in the codewords. The decoder then puts out a different quantum level than the one intended for a particular sample.

Hence, the reconstructed message waveform becomes contaminated with decoding noise as well as quantization noise.

The analysis of decoding noise is not too difficult if we restrict our attention to binary PCM with uniform quantization and a relatively small bit error probability P_e. The number of bit errors in a ν-digit codeword is a random variable governed by the binomial distribution. However, when $P_e \ll 1$, the probability of one error in a given word approximately equals νP_e and the probability of two or more errors is small enough for us to ignore that event. Of course, the effect of a single error depends on where it falls in the word, because the different bit positions have different decoding interpretations.

Consider a "natural" binary codeword of the form $b_{\nu-1}b_{\nu-2}\cdots b_1 b_0$ in which the mth bit distinguishes between quantum levels spaced by 2^m times the step height $2/q$. (The sign/magnitude code illustrated back in Fig. 12.1-3 follows this pattern for the magnitude bits, but the sign bit has a variable level meaning that makes analysis more complicated.) An error in the mth bit then shifts the decoded level by the amount $\epsilon_m = \pm(2/q)2^m$, and the average of ϵ_m^2 over the ν bit positions equals the mean-square decoding error for a random bit-error location. Thus,

$$\begin{aligned}\overline{\epsilon_m^2} &= \frac{1}{\nu}\sum_{m=0}^{\nu-1}\left(\frac{2}{q}2^m\right)^2 = \frac{4}{\nu q^2}\sum_{m=0}^{\nu-1}4^m \\ &= \frac{4}{\nu q^2}\,\frac{4^\nu-1}{3} = \frac{4}{3\nu}\,\frac{q^2-1}{q^2} \approx \frac{4}{3\nu} \qquad (1)\end{aligned}$$

where we've used the formula for summing a geometric progression and substituted $4^\nu = 2^{2\nu} = q^2 \gg 1$. The *decoding noise power* is therefore

$$\sigma_d^2 = \nu P_e \overline{\epsilon_m^2} \approx {}^4\!/_3 P_e \qquad (2)$$

since erroneous words occur with probability νP_e.

The *total* destination noise power consists of decoding noise σ_d^2 and quantization noise $\sigma_q^2 = 1/3q^2$, which come from essentially independent processes. Therefore,

$$N_D = \sigma_q^2 + \sigma_d^2 = \frac{1+4q^2P_e}{3q^2}$$

and

$$\left(\frac{S}{N}\right)_D = \frac{3q^2}{1+4q^2P_e}S_x \qquad (3)$$

so the effect of decoding noise depends upon the relative value of the quantity $4q^2P_e$. Indeed, we see in Eq. (3) the two extreme conditions

$$\left(\frac{S}{N}\right)_D \approx \begin{cases} 3q^2S_x & P_e \ll 1/4q^2 \\ \dfrac{3}{4P_e}S_x & P_e \gg 1/4q^2 \end{cases} \qquad (4)$$

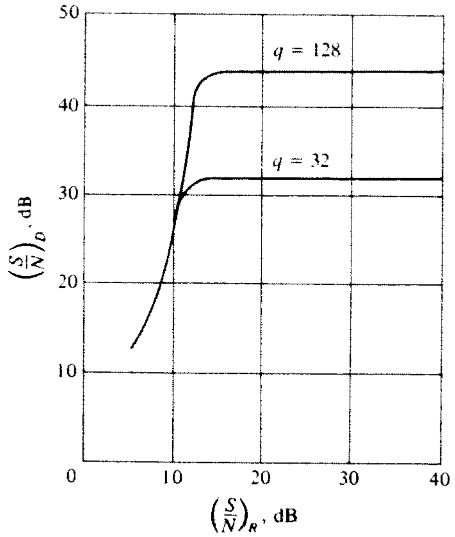


Figure 12.2-1 Noise performance of PCM.

Quantization noise dominates when P_e is small, but decoding noise dominates and reduces $(S/N)_D$ when P_e is large compared to $1/4q^2$.

Now recall that the value of P_e is determined by the received signal-to-noise ratio $(S/N)_R$ at the input to the digital regenerator. Specifically, for polar binary signaling in gaussian white noise we know that $P_e = Q[\sqrt{(S/N)_R}]$. Figure 12.2-1 plots $(S/N)_D$ versus $(S/N)_R$ for this case, with $S_x = 1/2$ and two values of q. The precipitous decline of $(S/N)_D$ as $(S/N)_R$ decreases constitutes a *threshold effect* caused by increasing errors. Below the error threshold, when $P_e \gg 1/4q^2$, the errors occur so often that the reconstructed waveform bears little resemblance to the original signal and the message has been mutilated beyond recognition.

Error Threshold

The PCM error threshold level is usually defined at the point where decoding noise reduces $(S/N)_D$ by 1 dB. Unfortunately, this definition does not lend itself to analytical investigations. As a more convenient alternative, we'll say that decoding errors have negligible effect if $P_e \le 10^{-5}$. Then we obtain the corresponding condition on $(S/N)_R$ for polar M-ary signaling using Eq. (20), Sect. 11.2, namely

$$P_e = 2\left(1-\frac{1}{M}\right)Q\left[\sqrt{\frac{3}{M^2-1}\left(\frac{S}{N}\right)_R}\,\right] \le 10^{-5}$$

Solving for the minimum value of $(S/N)_R$ yields the threshold level

$$\left(\frac{S}{N}\right)_{R_{th}} \approx 6(M^2-1) \qquad (5)$$

If $(S/N)_R < 6(M^2-1)$, the PCM output will be hopelessly mutilated by decoding noise.

A subtle but important implication of Eq. (5) relates to the digital signaling rate and transmission bandwidth. We'll bring out that relationship with the help of the analog transmission parameter $\gamma = S_R/\eta W = (B_T/W)(S/N)_R$. The PCM transmission bandwidth is $B_T \geq r/2 \geq \nu W$, so

$$\gamma_{\text{th}} = (B_T/W)(S/N)_{R_{\text{th}}}$$

$$\approx 6\,\frac{B_T}{W}\,(M^2 - 1) \geq 6\nu(M^2 - 1) \qquad (6)$$

Given ν and M, this equation tells you the minimum value of γ needed for PCM operation above threshold. It also facilitates the comparison of PCM with other transmission schemes.

Example 12.2-1 Example 12.1-2 showed that a voice PCM system with $M = 2$, $\nu = 8$, and μ-law companding has$(S/N)_D \approx 37$ dB. Equation (6) gives the corresponding threshold level $\gamma_{\text{th}} \geq 144 \approx 22$ dB. Hence, the PCM system has a potential 15-dB advantage over direct analog baseband transmission in which $(S/N)_D = \gamma$. The full advantage would not be realized in practice where allowance must be made for $B_T > \nu W$ and $\gamma > \gamma_{\text{th}}$.

Exercise 12.2-1 The 1-dB definition for error threshold is equivalent to $10 \log_{10}(1 + 4q^2 P_e) = 1$. Calculate P_e from this definition when $q = 2^8$. Then find the corresponding value of $(S/N)_R$ for polar binary signaling, and compare your result with Eq. (5).

PCM versus Analog Modulation

The PCM threshold effect reminds us of analog modulation methods such as FM and PPM that have the property of *wideband noise reduction* above their threshold levels. As our initial point of comparison, let's demonstrate that PCM also provides wideband noise reduction when operated above threshold so $(S/N)_D = 3q^2 S_x$. For this purpose, we'll assume that the sampling frequency is close to the Nyquist rate and $B_T \approx \nu W$. Then $q = M^\nu \approx M^b$ where $b = B_T/W$ is the bandwidth ratio. Hence,

$$\left(\frac{S}{N}\right)_D \approx 3M^{2b} S_x \qquad (7)$$

which exhibits noise reduction as an *exponential* exchange of bandwidth for signal-to-noise ratio. Granted that the noise being reduced is quantization noise, but random noise has no effect on PCM above threshold. The exponential factor in Eq. (7) is far more dramatic than that of wideband analog modulation, where $(S/N)_D$ increases proportionally to b or b^2.

For further comparison including threshold limitations, Fig. 12.2-2 illustrates the performance of several modulation types as a function of γ. All curves are calculated with $S_x = 1/2$, and the heavy dots indicate the threshold points. The PCM curves are based on Eqs. (6) and (7) with $M = 2$ and $\nu = b$.

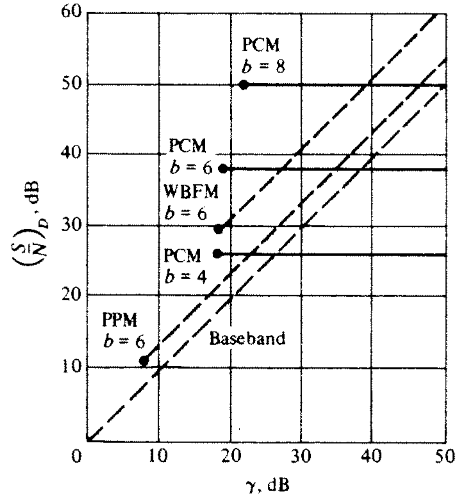


Figure 12.2-2 Performance comparison of PCM and analog modulation.

Clearly, in the name of power efficiency, PCM should be operated just above threshold, since any power increase beyond $\gamma \approx \gamma_{\text{th}}$ yields no improvement of $(S/N)_D$. Near threshold, PCM does offer some advantage over FM or PPM with the same value of b and $(S/N)_D$. And even a 3-dB power advantage, being a factor of 2, may spell the difference between success and failure for some applications. But that advantage is gained at the price of more complicated and costly hardware. In fact, PCM was deemed totally impractical prior to the development of high-speed digital electronics in the late 1950s—two decades after the invention of PCM.

Two other PCM benefits don't appear in Fig. 12.2-2. One is the advantage of *regenerative repeaters* when the transmission link requires many repeater stations. Another benefit comes from the fact that PCM allows analog message transmission as a digital signal. *Digital multiplexing* then makes it possible to combine PCM and digital data signals for flexible and efficient utilization of a communication channel. Taken together, these two benefits account for PCM's preeminence in the design of new systems for long-distance telephony.

But PCM is not suited to all applications. In radio broadcasting, for instance, we want a relatively large signal-to-noise ratio, say $(S/N)_D \approx 60$ dB. Figure 12.2-2 reveals that this would require binary PCM with $b > 8$, or FM with a smaller bandwidth ratio $b = 6$ and much simpler hardware at the transmitter and receivers. Likewise, bandwidth and hardware considerations would reject PCM for most single-channel systems.

Exercise 12.2-2 Starting with Eqs. (6) and (7), show that a PCM system operated at the threshold point has

$$\left(\frac{S}{N}\right)_{D_{\text{th}}} = 3\left(1 + \frac{\gamma_{\text{th}}}{6b}\right)^b S_x \qquad (8)$$

Compare this expression with that of WBFM by setting $D = b/2 \gg 1$ in Eq. (20*b*), Sect. 9.4.

12.3 DELTA MODULATION AND PREDICTIVE CODING

Sample values of analog waveforms derived from physical processes often exhibit *predictability* in the sense that the average change from sample to sample is small. Hence, you can make a reasonable guess of the next sample value based on previous values. The predicted value has some error, of course, but the range of the error should be much less than the peak-to-peak signal range. Predictive coded modulation schemes exploit this property by transmitting just the *prediction errors*. An identical prediction circuit at the destination combines the incoming errors with its own predicted values to reconstruct the waveform.

Predictive methods work especially well with audio and video signals, and much effort has been devoted to prediction strategies for efficient voice and image transmission. *Delta modulation* (DM) employs prediction to simplify hardware in exchange for increased signaling rate compared to PCM. *Differential pulse-code modulation* (DPCM) reduces signaling rate but involves more elaborate hardware. We'll discuss both DM and DPCM in this section, along with the related and fascinating topic of *speech synthesis* using prediction.

Delta Modulation

Let an analog message waveform $x(t)$ be lowpass filtered and sampled every T_s seconds. We'll find it convenient here to use *discrete-time* notation, with the integer independent variable k representing the sampling instant $t = kT_s$. We thus write $x(k)$ as a shorthand for $x(kT_s)$, etc.

When the sampling frequency is greater than the Nyquist rate, we expect that $x(k)$ roughly equals the previous sample value $x(k-1)$. Therefore, given the quantized sample value $x_q(k-1)$, a reasonable guess for the next value would be

$$\tilde{x}_q(k) = x_q(k-1) \tag{1}$$

where $\tilde{x}_q(k)$ denotes our *prediction* of $x_q(k)$. A delay line with time delay T_s then serves as the prediction circuit. The difference between the predicted and actual value can be expressed as

$$x_q(k) = \tilde{x}_q(k) + \epsilon_q(k) \tag{2}$$

in which $\epsilon_q(k)$ is the *prediction error*.

If we transmit $\epsilon_q(k)$, we can use the system in Fig. 12.3-1 to generate $x_q(k)$ by delaying the current output and adding it to the input. This system implements Eqs. (1) and (2), thereby acting as an *accumulator*. The accumulation effect is brought out by writing $x_q(k) = \epsilon_q(k) + x_q(k-1)$ with $x_q(k-1) = \epsilon_q(k-1) + x_q(k-2)$, and so forth; hence

$$\begin{aligned} x_q(k) &= \epsilon_q(k) + \epsilon_q(k-1) + x_q(k-2) \\ &= \epsilon_q(k) + \epsilon_q(k-1) + \epsilon_q(k-2) + \cdots \end{aligned}$$

An *integrator* accomplishes the same accumulation when $\epsilon_q(k)$ takes the form of brief rectangular pulses.

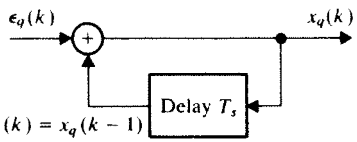


Figure 12.3-1 Accumulator for delta modulation.

At the transmitting end, prediction errors are generated by the simple delta modulation system diagrammed in Fig. 12.3-2. The comparator serves as a *binary quantizer* with output values $\pm\Delta$, depending on the difference between the predicted value $\tilde{x}_q(k)$ and the unquantized sample $x(k)$. Thus, the resulting DM signal is

$$\epsilon_q(k) = [\operatorname{sgn} \epsilon(k)]\, \Delta \tag{3a}$$

where

$$\epsilon(k) = x(k) - \tilde{x}_q(k) \tag{3b}$$

which represents the unquantized error. An accumulator (or integrator) in a feedback loop produces $\tilde{x}_q(k)$ from $\epsilon_q(k)$, similar to Fig. 12.3-1 except that the feedback signal comes from the delayed output.

Observe that this DM transmitter requires no analog-to-digital conversion other than the comparator. Also observe that an accumulator like Fig. 12.3-1 performs the digital-to-analog conversion at the receiver, reconstructing $x_q(k)$ from $\epsilon_q(k)$. A DM system thereby achieves digital transmission of analog signals with very simple hardware compared to a PCM system.

The name delta modulation reflects the fact that each input sample $x(k)$ has been encoded as a single pulse of height $+\Delta$ or $-\Delta$. But we can also view $\epsilon_q(k)$ as a *binary waveform* with signaling rate $r_b = f_s$, or *one bit per sample*. For this reason DM is sometimes called "one-bit PCM." The corresponding transmission bandwidth requirement is

$$B_T \geq r_b/2 = f_s/2 \tag{4}$$

We get by with just one bit per sample because we're transmitting prediction errors, not sample values. Nonetheless, successful operation requires rather high sampling rates, as we'll soon see.

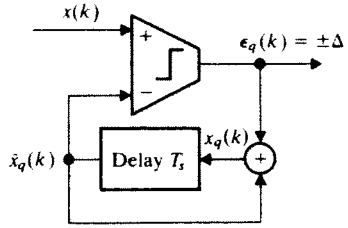


Figure 12.3-2 DM transmitter.

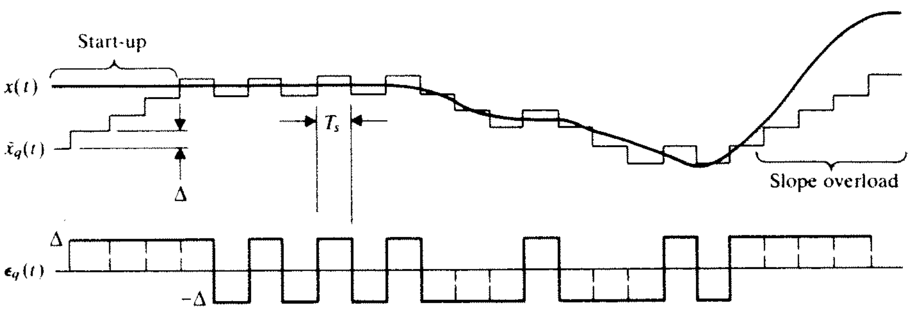


Figure 12.3-3 DM waveforms.

Figure 12.3-3 depicts illustrative continuous-time waveforms $x(t)$, $\tilde{x}_q(t)$, and $\epsilon_q(t)$ involved in DM. The staircase waveform $x_q(t)$ at the receiver differs from $\tilde{x}_q(t)$ only by a time shift of T_s seconds. The transmitter starts with an arbitrary initial prediction such as $\tilde{x}_q(0) < x(0)$ so $\epsilon_q(0) = +\Delta$. Then $\epsilon_q(0)$ is fed back through the accumulator to form the updated prediction $\tilde{x}_q(T_s) = x_q(0) + \epsilon_q(0)$. Continual updating at each sampling instant causes $\tilde{x}_q(t)$ to increase by steps of Δ until the start-up interval ends when $\tilde{x}_q(kT_s) > x(kT_s)$ and $\epsilon_q(kT_s) = -\Delta$. If $x(t)$ remains constant, $\tilde{x}_q(t)$ takes on a hunting behavior. When $x(t)$ varies with time, $\tilde{x}_q(t)$ follows it in stepwise fashion as long as the rate of change does not exceed the DM tracking capability. The difference between $\tilde{x}_q(t)$ and $x(t)$ is called *granular noise*, analogous to quantization noise in PCM. The reconstructed and smoothed waveform at the receiver will be a reasonable approximation for $x(t)$ if Δ and T_s are sufficiently small.

But when $x(t)$ increases or decreases too rapidly, $\tilde{x}_q(t)$ lags behind and we have the phenomenon known as *slope overload*, a fundamental limitation of DM. Since $\tilde{x}_q(t)$ changes by $\pm\Delta$ every $T_s = 1/f_s$ seconds, the maximum DM slope is $\pm f_s\,\Delta$ and a sufficient condition for slope tracking is

$$f_s\,\Delta \geq |\dot{x}(t)|_{\max} \tag{5}$$

where $\dot{x}(t) = dx/dt$. Consider, for instance, the modulating tone $x(t) = A_m \cos 2\pi f_m t$ so $\dot{x}(t) = -2\pi f_m A_m \sin 2\pi f_m t$ and $|\dot{x}(t)|_{\max} = 2\pi f_m A_m \leq 2\pi W$, where the upper bound incorporates our message conventions $A_m \leq 1$ and $f_m \leq W$. Equation (5) therefore calls for a high sampling frequency $f_s \geq 2\pi W/\Delta \gg \pi W$, since we want $\Delta \ll 2$ to make the steps of $\tilde{x}_q(t)$ small compared to the peak-to-peak signal range $-1 \leq x(t) \leq 1$.

DM performance quality depends on the granular noise, slope-overload noise, and regeneration errors. However, only granular noise has a significant effect under normal operating conditions, which we assume hereafter. Even so, the analysis of granular noise is a difficult problem best tackled by computer simulation for accurate results or by approximations for rough results.

We'll estimate DM performance using the receiver modeled by Fig. 12.3-4*a* with $\epsilon_q(t - T_s) = \epsilon_q(k-1)$ at the input to the accumulator. Equations (1) and (3*b*) then give the accumulator output as $x_q(k-1) = \tilde{x}_q(k) = x(k) - \epsilon(k)$ or

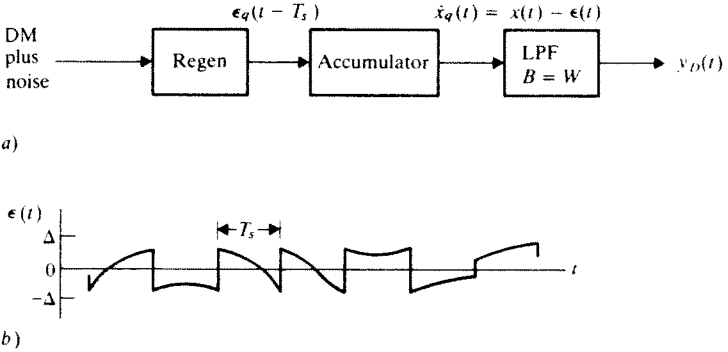


Figure 12.3-4 (*a*) DM receiver; (*b*) granular noise waveform.

$$\tilde{x}_q(t) = x(t) - \epsilon(t) \tag{6}$$

where $\epsilon(t)$ is the granular noise waveform sketched in Fig. 12.3-4*b*. The shape of $\epsilon(t)$ and the fact that $|\epsilon(t)| \leq \Delta$ suggest a uniform amplitude distribution with

$$\overline{\epsilon^2} = \Delta^2/3$$

Furthermore, experimental studies confirm that the power spectrum of $\epsilon(t)$ is essentially flat over $|f| \leq 1/T_s = f_s$. Thus,

$$G_\epsilon(f) \approx \overline{\epsilon^2}/2f_s \qquad |f| \leq f_s$$

and lowpass filtering yields

$$N_g = \int_{-W}^{W} G_\epsilon(f)\,df = \frac{W}{f_s}\,\overline{\epsilon^2} = \frac{W}{f_s}\,\frac{\Delta^2}{3} \tag{7}$$

which is the average power of the granular noise component of $x_q(t)$.

When granular noise is the only contamination in the filtered output, we obtain the signal-to-noise ratio

$$\left(\frac{S}{N}\right)_D = \frac{S_x}{N_g} = \frac{3f_s}{\Delta^2 W}\,S_x \tag{8}$$

This result is almost identical to the PCM expression $(S/N)_D = 3q^2 S_x$ if $f_s = 2W$ and $\Delta = 1/q$. But f_s and Δ must satisfy Eq. (5). We therefore need a more general relationship for the slope-tracking condition.

Recall from Eq. (22), Sect. 5.2, that if $G_x(f)$ is the power spectrum of $x(t)$, then the power spectrum of the derivative $\dot{x}(t)$ is $(2\pi f)^2 G_x(f)$. Hence, the *mean square signal slope* can be put in the form

$$\overline{|\dot{x}(t)|^2} = \int_{-\infty}^{\infty} (2\pi f)^2 G_x(f)\,df = (2\pi\sigma W_{\text{rms}})^2 \tag{9}$$

where $\sigma = \sqrt{S_x}$ is the signal's rms value and W_{rms} is its *rms bandwidth*, defined by

$$W_{\text{rms}} \triangleq \frac{1}{\sigma}\left[\int_{-\infty}^{\infty} f^2 G_x(f)\, df\right]^{1/2} \tag{10}$$

Now we introduce the so-called *slope loading factor*

$$s \triangleq \frac{f_s \Delta}{2\pi\sigma W_{\text{rms}}} \tag{11}$$

which is the ratio of the maximum DM slope to the rms signal slope.

A reasonably large value of s ensures negligible slope overload. Hence, we'll incorporate this factor explicitly in Eq. (8) by writing Δ in terms of s from Eq. (11). Thus,

$$\begin{aligned}\left(\frac{S}{N}\right)_D &= \frac{3}{4\pi^2}\frac{f_s^3}{s^2 W_{\text{rms}}^2 W} \\ &= \frac{6}{\pi^2}\left(\frac{W}{W_{\text{rms}}}\right)^2 \frac{b^3}{s^2}\end{aligned} \tag{12}$$

where

$$b \triangleq f_s/2W \tag{13}$$

The parameter b equals our usual bandwidth ratio B_T/W when B_T has the minimum value $f_s/2$. Equation (12) brings out the fact that DM performance falls between PCM and PPM since the wideband noise reduction goes as b^3 rather than b^2 or exponentially. DM generally requires a larger transmission bandwidth than PCM to achieve the same signal-to-noise ratio, so its applications are limited to those cases where ease of implementation takes precedence over bandwidth considerations.

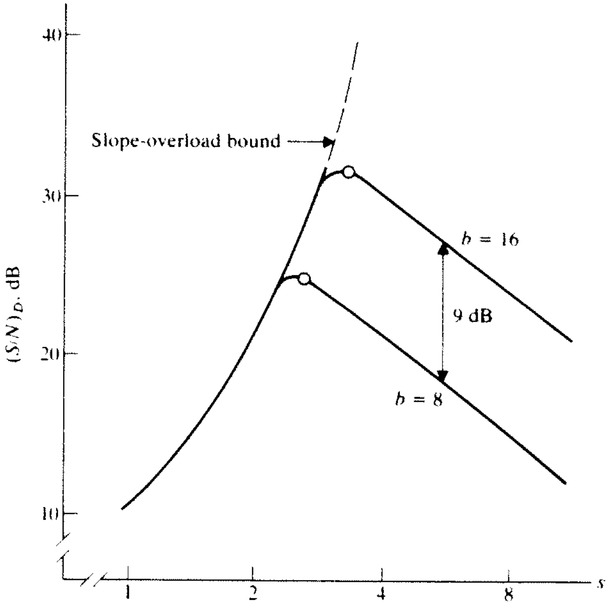


Figure 12.3-5 DM performance versus slope loading factor.

Computer simulations by Abate (1967) indicate that Eq. (12) holds for $\ln 2b \leq s < 8$. If $s < \ln 2b$, then *slope-overload noise* dominates and $(S/N)_D$ drops off quite rapidly. Figure 12.3-5 illustrates how $(S/N)_D$ varies with s. For a specified bandwidth ratio, DM performance is maximized by taking the empirically determined optimum slope loading factor

$$s_{\text{opt}} \approx \ln 2b \tag{14}$$

The maximum value of $(S/N)_D$ is then given by Eq. (12) with $s = s_{\text{opt}}$.

Example 12.3-1 DM voice transmission When a typical voice signal has been prefiltered so that $W \approx 4$ kHz, its rms bandwidth will be $W_{\text{rms}} \approx 1.3$ kHz. Substituting these values and $s = s_{\text{opt}} = \ln 2b$ in Eq. (12) yields the maximum DM signal-to-noise ratio $(S/N)_D \approx 5.8b^3/(\ln 2b)^2$. If $b = 16$, then $(S/N)_D \approx 33$ dB which is comparable to binary PCM with $\nu = 7$ and μ-law companding. But the DM signal requires $r_b = f_s = 2bW = 128$ kbps and $B_T \geq 64$ kHz, whereas the PCM signal would have $r_b = \nu f_s \geq 56$ kbps and $B_T \geq 28$ kHz.

Exercise 12.3-1 Consider a signal with a uniform power spectrum $G_x(f) = (S_x/2W)\Pi(f/2W)$. Show that $W_{\text{rms}} = W/\sqrt{3}$. Then calculate the optimum value of Δ in terms of S_x when $b = 16$.

Adaptive Delta Modulation

Adaptive delta modulation (ADM) involves additional hardware designed to provide *variable step size*, thereby reducing slope-overload effects without increasing the granular noise. A reexamination of Fig. 12.3-3 reveals that slope overload appears in $\epsilon_q(t)$ as a sequence of pulses having the same polarity, whereas the polarity tends to alternate when $x_q(t)$ tracks $x(t)$. This sequence information can be utilized to adapt the step size in accordance with the signal's characteristics.

Figure 12.3-6 portrays the action of an ADM transmitter in which the step size in the feedback loop is adjusted by a variable gain $g(k)$ such that

$$\tilde{x}_q(k) = \tilde{x}_q(k-1) + g(k-1)\epsilon_q(k-1)$$

The step-size controller carries out the adjustment algorithm

$$g(k) = \begin{cases} g(k-1) \times K & \epsilon_q(k) = \epsilon_q(k-1) \\ g(k-1)/K & \epsilon_q(k) \neq \epsilon_q(k-1) \end{cases}$$

where K is a constant taken to be in the range $1 < K < 2$. Thus, the effective step size increases by successive powers of K during slope-overload conditions, as signified by $\epsilon_q(k) = \epsilon_q(k-1)$, but decreases when $\epsilon_q(k) \neq \epsilon_q(k-1)$. Another adaptive

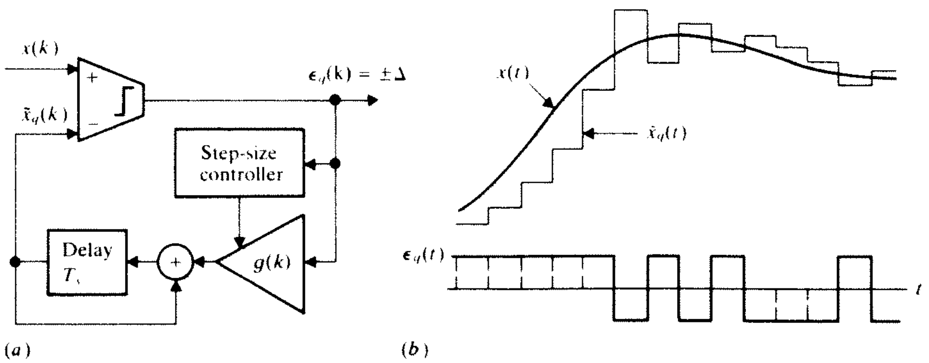


Figure 12.3-6 (a) Adaptive DM transmitter; (b) waveforms.

scheme called *continuously variable slope delta modulation* (CVSDM) provides a continuous range of step-size adjustment instead of a set of discrete values.

The signal-to-noise ratio of ADM is typically 8–14 dB better than ordinary DM. Furthermore, the variable step size yields a wider *dynamic range* for changing values of S_x, similar to the effect of μ-law companding in PCM. As a net result, ADM voice transmission gets by with a bandwidth ratio of $b = 6$–8 or $B_T = 24$–32 kHz.

Differential PCM

Differential pulse-code modulation (DPCM) combines prediction with multilevel quantizing and coding. The transmitter diagrammed in Fig. 12.3-7*a* has a q-level quantizer with quantum levels at $\pm\Delta, \pm 3\Delta, \ldots, \pm(q-1)\,\Delta$. The unquantized error $x(k) - \tilde{x}_q(k)$ is applied to the quantizer to produce the prediction error $\epsilon_q(k)$ which is then encoded as a binary word with $v = \log_2 q$ bits, just like binary PCM. DPCM transmission therefore requires

$$r_b = vf_s \qquad B_T \geq vf_s/2$$

However, since $\tilde{x}_q(k)$ changes as much as $\pm(q-1)\,\Delta$ from sample to sample, the slope tracking condition becomes

$$f_s(q-1)\,\Delta \geq |\dot{x}(t)|_{\max}$$

If $q \gg 1$ and $W_{\text{rms}} \ll W$, the sampling frequency can be nearly as low as the Nyquist rate.

Multilevel quantization of the prediction error obviously provides better information for message reconstruction at the receiver. To gain full advantage of this potential, the DPCM prediction circuit usually takes the form of a transversal filter shown in Fig. 12.3-7*b*, where

$$\tilde{x}_q(k) = \sum_{i=1}^{n} c_i x_q(k-i)$$

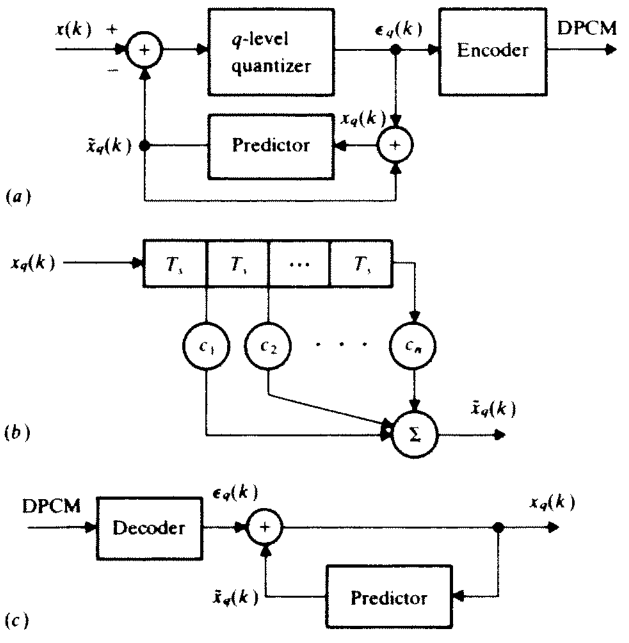


Figure 12.3-7 Differential PCM. (a) Transmitter; (b) prediction circuit; (c) receiver.

so the predictor draws upon the previous n samples. The tap gains c_i are chosen to minimize the mean square value of the error $x(k) - \tilde{x}_q(k)$. The receiver in Fig. 12.3-7*c* includes an identical prediction filter after the decoder.

Assuming $q \gg 1$ and no slope overload, DPCM performs essentially like PCM enhanced by a *prediction gain* G_p such that

$$\left(\frac{S}{N}\right)_D = G_p 3q^2 S_x \tag{15}$$

The gain of an optimum predictor is given by

$$G_p = \left[1 - \sum_{i=1}^{n} c_i \rho_i\right]^{-1} \tag{16a}$$

where $\rho_i = R_x(iT_s)/S_x$ is the normalized signal correlation and the tap gains satisfy the matrix relationship

$$\begin{bmatrix} \rho_0 & \rho_1 & \cdots & \rho_{n-1} \\ \rho_1 & \rho_0 & \cdots & \rho_{n-2} \\ \vdots & & & \vdots \\ \rho_{n-1} & \rho_{n-2} & \cdots & \rho_0 \end{bmatrix} \begin{bmatrix} c_1 \\ c_2 \\ \vdots \\ c_n \end{bmatrix} = \begin{bmatrix} \rho_1 \\ \rho_2 \\ \vdots \\ \rho_n \end{bmatrix} \tag{16b}$$

Jayant (1974) outlines the derivation of Eqs. (15) and (16) and presents experimental data showing that $G_p \approx 5$–10 dB for voice signals. The higher correlation of a TV video signal results in prediction gains of about 12 dB.

In contrast to delta modulation, DPCM employs more elaborate hardware than PCM for the purpose of improving performance quality or reducing the signaling rate and transmission bandwidth. *Adaptive* DPCM (ADPCM) achieves even greater improvement by adapting the quantizer or predictor or both to the signal characteristics.

Exercise 12.3-2 Suppose a DPCM predictor yields $G_p = 6$ dB. Show that the DPCM word needs one less bit than that of binary PCM, all other factors being equal. *Hint:* See Eq. (7), Sect. 12.1.

LPC Speech Synthesis

Linear predictive coding (LPC) is a new and radically different approach to digital representation of analog signals. The method uses a transversal filter (or its digital-circuit equivalent) plus some auxiliary components to *synthesize* the waveform in question. The parameters of the waveform synthesizer are then encoded for transmission, instead of the actual signal. Considerable efficiency results if the synthesizer accurately mimics the analog process. Since there already exists extensive knowledge about speech processes, LPC is particularly well suited to speech synthesis and transmission.

Figure 12.3-8 diagrams a speech synthesizer consisting of two input generators, a variable-gain amplifier, and a transversal filter in a feedback loop. The amplifier gain and filter tap gains are adjusted to model the acoustical properties of the vocal tract. Unvoiced speech (such as hissing sound) is produced by connecting the white-noise generator. Voiced speech is produced by connecting the impulse-train generator set at an appropriate pitch frequency.

If the filter has about 10 tap gains, and all parameter values are updated every 10 to 25 ms, the synthesized speech is quite intelligible although it may sound rather artificial. Some talking toys and recorded-message systems generate speech sounds by the synthesis method, drawing upon parameter values stored in a digital memory.

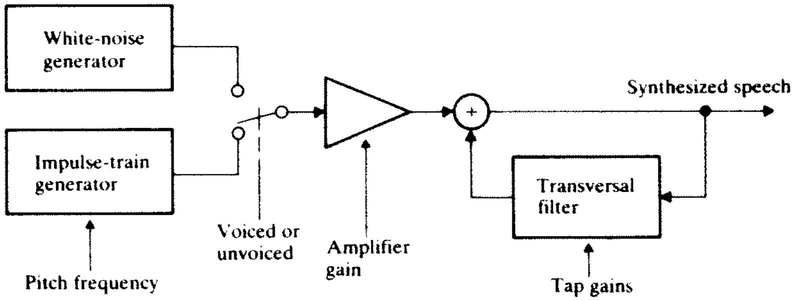


Figure 12.3-8 Speech synthesizer.

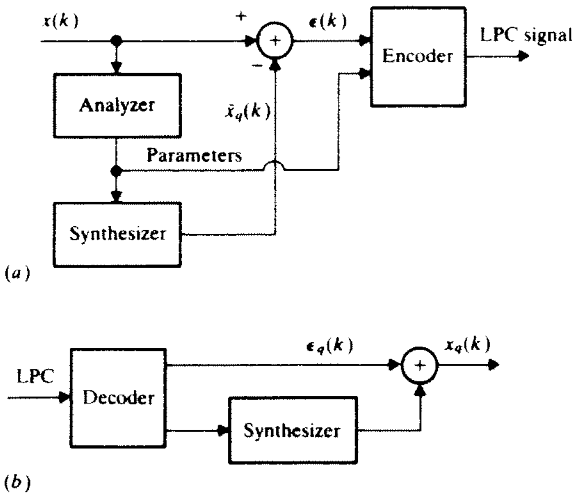


Figure 12.3-9 LPC transmission system. (*a*) Transmitter; (*b*) receiver.

Now consider the LPC transmitter in Fig. 12.3-9*a*. Sample values of the voice input are analyzed to determine the parameters for the synthesizer, whose output is compared with the input. The resulting error is encoded along with the parameter values to form the transmitted digital signal. The receiver in Fig. 12.3-9*b* uses the parameter values and quantized error to reconstruct the voice waveform.

A complete LPC code word consists of about 80 bits — 1 bit for the voiced/unvoiced switch, 6 bits for the pitch frequency, 5 for the amplifier gain, 6 for each of the 10 tap gains, and a few bits for the error. Updating the parameters every 10–25 ms is equivalent to sampling at 40–100 Hz, so LPC requires a very modest bit rate in the vicinity of 3000–8000 bps. Table 12.3-1 compares LPC with other voice encoding methods. The substantial bit-rate reduction made possible by LPC has stimulated efforts to improve the quality of speech synthesis for voice communication. See Rabiner and Schafer (1978) for a general introduction to digital processing applied to speech signals.

Table 12.3-1 Comparison of voice encoding methods

Encoding method	Sampling rate, kHz	Bits per sample	Bit rate, kbps
DM	64–128	1	64–128
PCM	8	7–8	56–64
ADM	48–64	1	48–64
DPCM	8	4–6	32–48
ADPCM	8	3–4	24–32
LPC	0.04–0.1	≈80	3–8

12.4 DIGITAL MULTIPLEXING

Analog signal multiplexing was previously discussed under the headings of frequency-division and time-division multiplexing. While those same techniques could be applied to waveforms representing digital signals, we gain greater efficiency and flexibility by taking advantage of the inherent nature of a digital signal as a sequence of symbols. Digital multiplexing is therefore based on the principle of *interleaving symbols* from two or more digital signals, similar to time-division multiplexing but free from the rigid constraints of periodic sampling and waveform preservation.

This section presents the general concepts and problems of digital multiplexing. The signals to be multiplexed may come from digital data sources or from analog sources that have been digitally encoded. We'll deal exclusively with binary multiplexers, recognizing that appropriate symbol conversion accommodates M-ary signals when necessary. Multiplexing techniques will be illustrated by specific cases, including telephone system hierarchies and an introduction to computer networks.

Multiplexers and Hierarchies

A binary multiplexer (MUX) merges input bits from different sources into one signal for transmission via a digital communication system. Or, in other words, a MUX divides the capacity of the system between several pairs of input and output terminals. The multiplexed signal consists of source digits interleaved bit-by-bit or in clusters of bits (words or characters).

Successful demultiplexing at the destination requires a carefully constructed multiplexed signal with a constant bit rate. Towards this end, a MUX usually must perform four functional operations:

1. Establish a frame as the smallest time interval containing at least one bit from every input.
2. Assign to each input a number of unique bit slots within the frame.
3. Insert control bits for frame identification and synchronization.
4. Make allowance for any variations of the input bit rates.

Bit-rate variation poses the most vexing design problem in practice, and leads to three broad categories of multiplexers.

Synchronous multiplexers are used when a master clock governs all sources, thereby eliminating bit-rate variations. Synchronous multiplexing systems attain the highest throughput efficiency, but they require elaborate provision for distributing the master-clock signal.

Asynchronous multiplexers are used for digital data sources that operate in a start/stop mode, producing bursts of characters with variable spacing between bursts. Buffering and character interleaving makes it possible to merge these sources into a synchronous multiplexed bit stream, as discussed later in conjunction with computer networks.

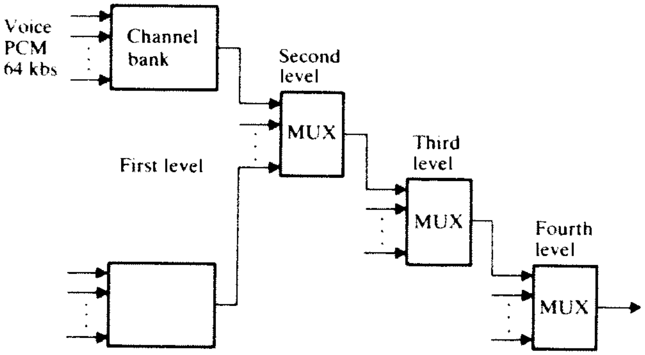


Figure 12.4-1 Multiplexing hierarchy for digital telecommunication.

Quasi-synchronous multiplexers are used when the input bit rates have the same nominal value but vary within specified bounds. These multiplexers, arranged in a hierarchy of increasing bit rates, constitute the building blocks of interconnected digital telecommunication systems.

Two slightly different multiplexing patterns have been adopted for digital telecommunication, the AT & T hierarchy in North America and Japan and the CCIT hierarchy in Europe. (CCIT stands for the International Telegraph and Telephone Consultive Committee of the International Telecommunications Union.) Both hierarchies are based on a 64-kbps voice PCM unit and have the same structural layout shown in Fig. 12.4-1. The third level is intended only for multiplexing purposes, whereas the other three levels are designed for point-to-point transmission as well as multiplexing. The parameters of the AT & T and CCIT hierarchies are listed in Table 12.4-1.

Observe in all cases that the output bit rate at a given level exceeds the sum of the input bit rates. This surplus allows for control bits and additional *stuff bits* needed to yield a steady output rate. Consequently, when we include the PCM bandwidth expansion, a digital network has very low *bandwidth efficiency* if devoted entirely to voice transmission. For instance, the fourth level of the

Table 12.4-1 Multiplexing hierarchies

	AT & T		CCIT	
	Number of inputs	Output rate, Mbps	Number of inputs	Output rate, Mbps
First level	24	1.544	30	2.048
Second level	4	6.312	4	8.448
Third level	7	44.736	4	34.368
Fourth level	6	274.176	4	139.264

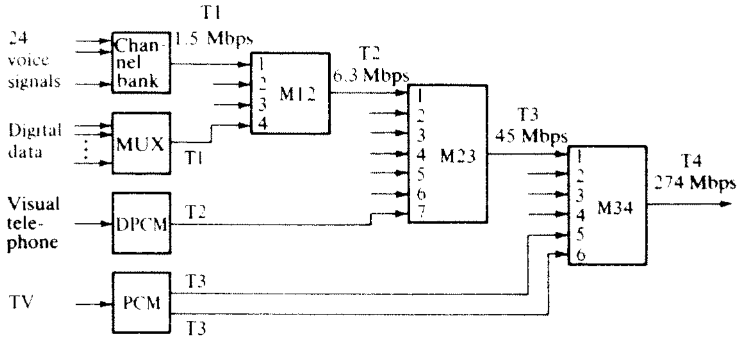


Figure 12.4-2 Illustrative configuration of the AT & T hierarchy.

AT & T multiplexing system requires $B_T \geq r_b/2 \approx 137$ MHz to transmit $24 \times 4 \times 7 \times 6 = 4032$ voice PCM signals, so the bandwidth efficiency is $(4032 \times 4 \text{ kHz})/137 \text{ MHz} \approx 12\%$. Going back to Table 8.2-1 we find that the Jumbogroup in the AT & T FDM hierarchy accommodates 3600 analog voice signals in $B_T = 17$ MHz, for a much higher bandwidth efficiency of $(3600 \times 4 \text{ kHz})/17 \text{ MHz} \approx 85\%$. Thus, digital multiplexing sacrifices analog bandwidth efficiency in exchange for the advantages of digital transmission.

Previously noted advantages of digital transmission include hardware cost reduction made possible by digital integrated circuits and power cost reduction made possible by regenerative repeaters. Now we can begin to appreciate the advantage of *flexibility* made possible by digital multiplexing, since the input bit streams at any level in Fig. 12.4-1 can be any desired mix of digital data and digitally encoded analog signals.

By way of example, Fig. 12.4-2 shows an illustrative configuration of the AT & T hierarchy with voice, digital data, visual telephone (Picturephone®), and color TV signals combined for transmission on the fourth-level T4 line. The first-level T1 signals include PCM voice and multiplexed digital data. The second-level T2 signals are multiplexed T1 signals along with visual telephone signals encoded as binary differential PCM (DPCM) with $f_s \approx 2$ MHz and $\nu = 3$ bits per word. PCM encoding of color TV requires a 90-Mbps bit rate ($f_s \approx 10$ MHz, $\nu = 9$), so two third-level T3 lines are allocated to this signal. The higher-level multiplexers labeled M12, M23, and M34 belong to the quasi-synchronous class whose properties we'll examine in some detail. But let's start at the first-level synchronous multiplexer called the channel bank.

Example 12.4-1 T1 voice PCM channel bank Synchronous multiplexing of voice PCM requires that the signals be delivered in *analog* form to the channel bank. Then, as diagrammed in Fig. 12.4-3*a*, sequential sampling under the control of a local clock generates an analog TDM PAM signal (see Figs. 10.3-1 and 10.3-2). This signal is transformed by the encoder into TDM PCM with interleaved words. Finally, the processor appends framing and signaling information to produce the output T1 signal.

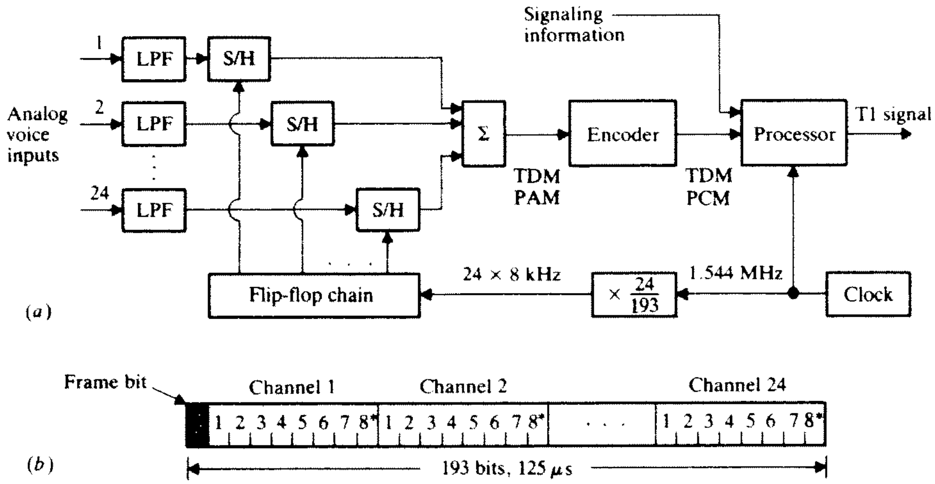


Figure 12.4-3 (*a*) T1 channel bank; (*b*) frame structure.

The T1 frame structure is represented by Fig. 12.4-3*b*. Each frame contains one 8-bit word from each of the 24 input channels plus one bit for framing, giving a total of 193 bits. The 8-kHz sampling frequency corresponds to 125-μs frame duration, so the T1 bit rate is $r_b = 193 \text{ bits} \div 125\ \mu\text{s} = 1.544$ Mbps. Signaling information (dial pulses, "busy" signals, etc.) is incorporated by a method aptly known as *bit robbing*. Every sixth frame, a signaling bit replaces the least-significant bit of each channel word—denoted by the starred bit locations in the figure. Bit-robbing reduces the effective voice-PCM word length to $\nu = 7\,^5\!/_6$ and has inconsequential effect on reproduction quality. Yet it allows 24 signaling bits every $6 \times 125\ \mu$s, or an equivalent signaling rate of 32 kbps.

T1 signals may be either combined at an M12 multiplexer or transmitted directly over short-haul links for local service up to 80 km. The T1 transmission line is a twisted-pair cable with regenerative repeaters every 2 km. A bipolar signal format eliminates the problems of absolute polarity and DC transmission.

Exercise 12.4-1 Assume that the first-level multiplexer in the CCIT hierarchy is a synchronous voice-PCM channel bank with 30 input signals, output bit rate $r_b = 2.048$ Mbps, and no bit-robbing. Find the number of framing plus signaling bits per frame.

Quasi-Synchronous Multiplexing

Quasi-synchronous multiplexing becomes necessary when the input bit streams have small variations around some nominal rate. This condition prevails at an M12 multiplexer, for instance, because the arriving T1 signals were generated by channel banks operating from stable but unsychronized local clocks. In general, the problems of quasi-synchronism arise in any digital multiplexing system that lacks overall master-clock control.

Quasi-synchronous multiplexers require three essential features: a sufficiently high output bit rate to accommodate the maximum expected input rates, special buffers called *elastic stores* for temporary storage of input bits, and *bit-stuffing* to pad the output stream. We'll introduce these features by considering the simplified rate-regulation system in Fig. 12.4-4*a*. Here a single input bit stream with variable rate r is to be regulated such that the output has fixed rate $r_0 \geq r_{\max}$. Whenever $r < r_{\max}$, the regulator stuffs extra bits into the output stream to maintain a constant rate. This system operates in the following manner.

Input bits are written sequentially into an elastic store consisting of a bank of storage cells and two switching circuits that act like commutators rotating at the input and output clock rates. Both clock signals are applied to a set-reset (SR) flip-flop whose output waveform drawn in Fig. 12.4-4*b* has a DC component that progressively decreases if $r_0 > r$. The control unit issues a stuff command pulse when this DC component gets too low, indicating that the read circuit is overtaking the write circuit. The stuff command inhibits the read circuit for one clock cycle and inserts a stuff bit into the output stream. Normal operation then resumes until the next time for a stuff bit. Unwanted stuff bits can be "destuffed" at the destination by another elastic store whose read circuit skips over storage cells identified by the transmitted stuff commands.

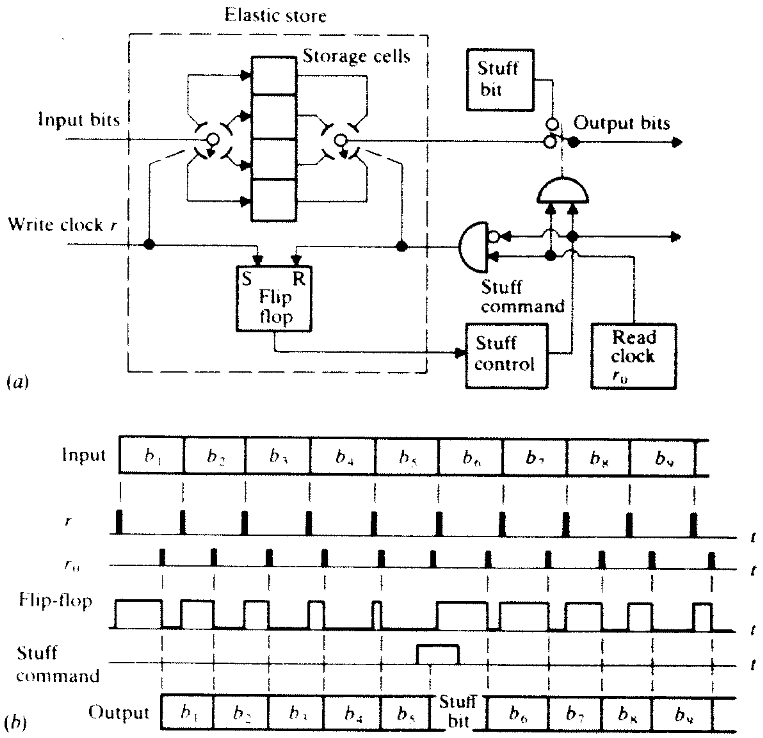


Figure 12.4-4 (*a*) Bit rate regulation system; (*b*) timing diagram.

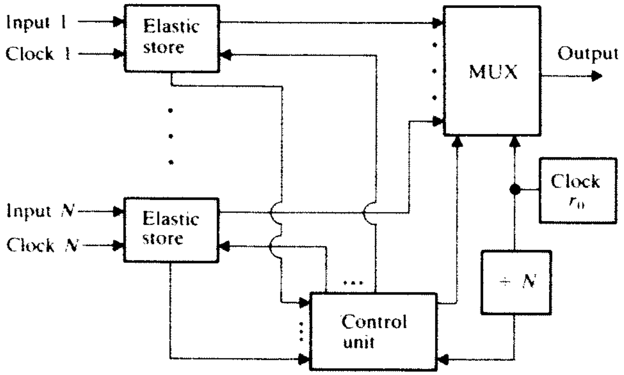


Figure 12.4-5 Quasi-synchronous multiplexer.

Now we put bit stuffing together with synchronous multiplexing to build the quasi-synchronous multiplexer in Fig. 12.4-5. There are N input signals whose rates vary from $r_{\min}$ to $r_{\max}$ around the nominal value $\bar{r}$. The MUX has a constant output rate $r_0 > Nr_{\max}$. The control unit receives flip–flop waveforms and returns read signals to each input elastic store. This unit also supplies the control and stuff bits to the MUX.

For an analytic description of quasi-synchronous multiplexing, let T_f be the frame duration. The total number of output bits per frame is

$$T_f r_0 = D + X \tag{1}$$

where D represents the number of message and stuff bits and X represents the number of control bits. Although D is a fixed parameter, the proportion of message and stuff bits depends on the input rates. If all inputs have the same rate $\bar{r}$, then

$$D = T_f N\bar{r} + Ns \tag{2}$$

in which s stands for the average number of stuff bits per channel per frame. Thus, for instance, $s = 1/2$ means that each channel gets one stuff bit every other frame under nominal operating conditions. Using this notation, we write the output rate as

$$\begin{aligned} r_0 &= \frac{D+X}{T_f} = \frac{D+X}{T_f D}\, T_f N\bar{r}\, \frac{D}{D-Ns} \\ &= N\bar{r}\left(\frac{D+X}{D}\right)\frac{D}{D-Ns} \end{aligned} \tag{3}$$

This expression relates r_0 and s when the other parameters have been specified.

Given r_0 and s, we determine the allowable input rate variations by worst-case analysis as follows. The maximum rate must satisfy $T_f N r_{\max} \le D$, where the upper limit corresponds to a frame with no stuff bits. At the other extreme, we usually want at most one stuff bit per channel per frame in order to simplify destuffing, so the minimum rate must satisfy $T_f N r_{\min} \ge D - N$. Equations (1) and (2) thus yield

$$r_{\max} \le \bar{r} + \frac{r_0}{D+X} s \qquad r_{\min} \ge \bar{r} - \frac{r_0}{D+X}(1-s) \tag{4}$$

In principle, Eq. (4) suggests taking $s = 1/2$ to permit equal rate variations around $\bar{r}$. In practice, lower average stuffing rates are employed to reduce timing jitter in the destuffed signals.

Example 12.4-2 M12 multiplexer The AT & T M12 multiplexer combines $N = 4$ T1 signals with $\bar{r} = 1.544$ Mbps. The framing pattern is divided into four subframes depicted by Fig. 12.4-6*a*. The letters *M*, *C*, and *F* denote control bits, and [48] denotes a sequence of 12 interleaved message and stuff bits from each of the four inputs. A complete frame consists of $X = 4 \times 6 = 24$ control bits and $D = 4 \times 6 \times 48 = 1152$ message and stuff bits. After canceling common factors, Eq. (3) becomes

$$r_0 = 4 \times 1.544 \text{ Mbps} \times \frac{49}{48} \times \frac{288}{288 - s}$$

The value $r_0 = 6.312$ Mbps was selected as a multiple of 8 kHz, and corresponds to $s \approx 1/3$. Each frame contains $Ns \approx 4/3$ stuff bits on the average, for a throughput efficiency of $(D - Ns)/(D + X) = N\bar{r}/r_0 \approx 98\%$.

The specific arrangement of control bits within the frame involves several considerations, some of which are made clearer with the help of Fig. 12.4-6*b*. The *F* control bits form the main framing sequence $F_0 F_1 F_0 F_1 \cdots = 0101 \cdots$ with 146 other bits between *F* bits. The demultiplexer synchronizes to this pattern and then identifies the subframes from the secondary sequence $M_0 M_1 M_1 M_1 = 0111$.

The *C* control bits in a given subframe serve as a stuff code. If $C_I C_I C_I =$ 111, then channel I has a stuff bit in its first allocated slot after the F_1 bit; otherwise, $C_I C_I C_I = 000$ indicates no stuff bit. Stuff bits for channel II are likewise handled in the second subframe, and so forth. The triple bit pattern provides protection against regeneration errors, since the probability of two or more errors in three bits is quite small.

Exercise 12.4-2 Suppose 24 voice signals arrive at a channel bank already encoded as PCM with $\bar{r} = 64$ kbps. Further suppose that the channel bank is a quasi-synchronous multiplexer whose output frame is divided into 24 subframes, each subframe containing 3 control bits and 8 message and stuff bits. Calculate r_0 taking $s = 1/3$, and determine the throughput efficiency $N\bar{r}/r_0$.

(*a*)

Subframe I:	M_0	[48]	C_I	[48]	F_0	[48]	C_I	[48]	C_I	[48]	F_1	[48]
Subframe II:	M_1	[48]	C_{II}	[48]	F_0	[48]	C_{II}	[48]	C_{II}	[48]	F_1	[48]
Subframe III:	M_1	[48]	C_{III}	[48]	F_0	[48]	C_{III}	[48]	C_{III}	[48]	F_1	[48]
Subframe IV:	M_1	[48]	C_{IV}	[48]	F_0	[48]	C_{IV}	[48]	C_{IV}	[48]	F_1	[48]

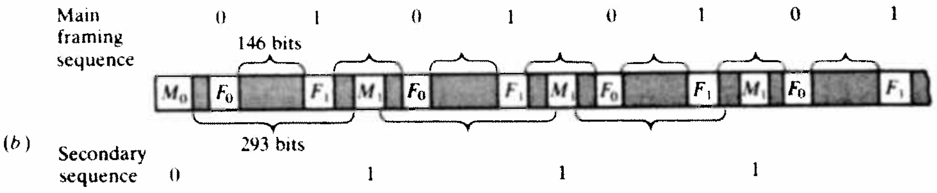


Figure 12.4-6 M12 multiplexer. (*a*) Framing pattern; (*b*) frame synchronization sequence.

Data Multiplexers and Computer Networks

Multiplexing for computer communication departs in two significant respects from general-purpose telecommunication service. On the one hand, complications arise from the fact that each computer and data terminal has its own independent clock and operates in an asynchronous start/stop mode. On the other hand, simplifications come from the fact that computers and data terminals don't require the nearly instantaneous response needed for two-way voice communication. Consequently, extensive buffering is essential and the associated time delay is tolerable.

A third characteristic of computer communication is relatively low bit rates compared to the megabits-per-second needed for multiplexed voice PCM. Although internal data transfers take place at very high speeds within a central processor, the limitations of electromechanical input/output devices result in much slower external transmission rates. Even high-speed printers can handle only 50 kbps, while multiplexed signals from remote data terminals are typically less than 10 kbps. We'll first look at asynchronous data multiplexers, and then describe how they are interconnected with central processors to form a computer network.

Teletypewriters and other keyboard data terminals work with alphanumeric symbols encoded as characters of 7–11 bits. These characters are transmitted asynchronously, one at a time, but the bit interval stays fixed throughout all characters. Figure 12.4-7 displays the 10-bit character format of the American

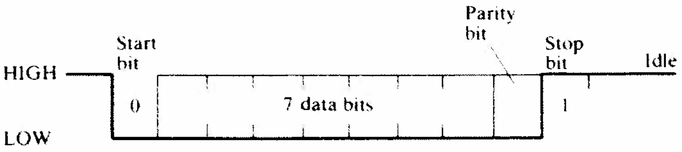


Figure 12.4-7 Format of ASCII 10-bit character code

Table 12.4-2 Selected AT & T data modems

Model	Bit rate	Modulation	Transmission channel
103A	300 bps	FSK	
202E	600 bps	FSK	Dial-up
212A	1200 bps	QAM	voice-grade line
201A	2000 bps	PSK	
203B	2400 bps	VSB	Leased
203C	3600 bps	VSB	voice-grade line
208A	7200 bps	PSK	Leased
209	9600 bps	QAM	conditioned line
303B	19.2 kbps	VSB	Half-group channel
303C	50.0 kbps	VSB	Group channel
303D	230.4 kbps	VSB	Supergroup channel

Standard Code for Information Interchange (ASCII). The start and stop bits provide character framing, the parity bit serves as error control, and the remaining 7 data bits identify the particular character. Incidentally, the signal voltage remains HIGH during idle times to keep the connection "alive."

Bit rates within the characters of low-speed asynchronous terminals range from 75 to 1200 bps. Higher-speed synchronous terminals have rates up to 7200 bps. Commercial data multiplexers are designed to combine characters from a mix of asynchronous and synchronous terminals. The multiplexed signal is then applied to a *modem* (modulator/demodulator) for transmission over a telephone channel. For purposes of illustration, Table 12.4-2 lists the maximum bit rates of selected modems available from AT & T. The carrier modulation methods will be covered in Chap. 14.

A data multiplexer operates in the same basic manner as a quasi-synchronous multiplexer, with interleaved characters and character-stuffing in place of single bits. However, the frame structure must account for the potential variety of input rates. Borrowing a strategy from TDM telemetry, frames are comprised of several characters from high-speed terminals but just one character or alternate characters from low-speed terminals. Figure 12.4-8 provides a simplified example of this strategy with input rates of 110, 110, 300, and 600 bps multiplexed into a 1200-bps output, the 80-bps surplus allowing for control characters. A subframe would then contain two characters from the 600-bps input, one from the 300-bps input, and alternate characters from the 110-bps inputs. Notice the similarity to the telemetry multiplexer back in Fig. 10.3-5.

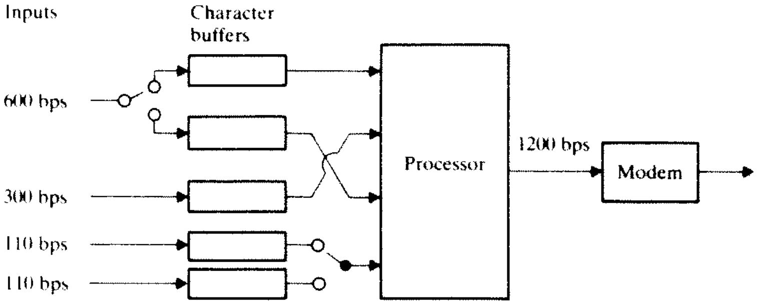


Figure 12.4-8 Illustrative data multiplexer.

Conventional or "nonintelligent" data multiplexers have *fixed input assignments* and continue to transmit dummy characters whenever a particular terminal happens to be idle. *Demand-assignment* multiplexers achieve better efficiency by skipping over idle inputs, so the output bit rate can be reduced in proportion to the expected demand factor. Of course, the input buffers must hold several characters in order to smooth out demand variations. The more sophisticated data *concentrator* goes even further in this direction, with the help of a microprocessor and extensive memory capacity. A concentrator assembles and temporarily stores blocks of characters from several inputs, performs data-compression coding, and carries out other operations designed to minimize the output transmission bit rate.

As implied by the foregoing discussion, efficient data multiplexing involves significant processing time delays—delays that would be unacceptable in two-way voice communication. But such delays don't bother machines, and human users of data terminals have become accustomed to brief turn-around lags. Most computer networks operate in a *store-and-forward* mode that also takes advantage of time-delay tolerance.

Figure 12.4-9 diagrams a hypothetical computer network modeled after the famous ARPANET. The ARPANET was established in 1969 by the Advanced Research Projects Agency of the U.S. Department of Defense; it now interconnects more than 100 host computers via telephone lines and satellite links, mostly at 50-kbps rates. Our hypothetical network consists of five host computers, symbolized by circles, and four additional terminal interfaces, symbolized by squares. Only host computers do computational work, but every circle or square represents a communication *node* equipped with a concentrator having the ability to transmit, receive, and store data.

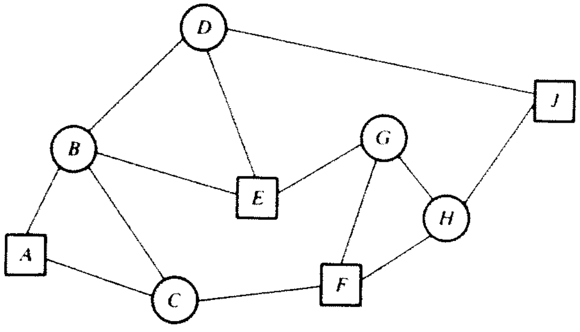


Figure 12.4-9 Computer network.

When a message is to be sent from one node to another, the source node subdivides the data into *packets* of about 1000 bits. Control bits attached to each packet include identification of the source and destination, The packets are transmitted, one by one, either directly to the destination node if possible or to the first available intermediate node. Packets arriving at an intermediate node are temporarily *stored* and then *forwarded* to the destination or to another intermediate node, depending upon the available lines. Suppose, for instance, that a two-packet message is to go from the terminal interface at node *A* to the host computer at node *G*; one packet might traverse the route *A-C-F-G* while the other follows the more circuitous path *A-B-E-D-J-H-G*.

This network illustrates the principles of store-and-forward operation with *packet-switching*. Both packet-switching and message-switching make better use of a transmission network than line-switching, which requires a complete route to be established before transmission begins. Packet-switching has the added advantage of eliminating the excessive time delay incurred when an entire message must be stored at an intermediate node.

But packet-switching has some interesting problems since, at any given time, the network contains numerous packets attempting to reach their respective destinations. Packets from one message can take different routes and may arrive at the destination in scrambled order. Furthermore, a *deadlock* situation results if too many packets have entered the network and compete for the available lines and buffer space—something like traffic gridlock in New York City on a Friday afternoon.

The packetizing technique is also used for *time-division multiple-access* (TDMA) in which a satellite or ground-based transponder links several locations. Each source location has periodic time slots assigned for packet transmission, and the transponder rebroadcasts a composite signal consisting of interleaved packets. As an example, recall the FDMA satellite system in Fig. 8.2-4 with ground stations in the United States, France, and Brazil. Figure 12.4-10*a* shows how the U.S. transmitter might be arranged for TDMA operation. A low-speed *background processor* stores the incoming data while a high-speed *foreground processor* compresses packets into short bursts at the assigned times, adding a preamble for identification and synchronization purposes. All ground stations receive the transponder signal represented by Fig. 12.4-10*b* and extract the appropriate packets. Other TDMA systems have demand-assigned or random packet transmission times, rather than fixed time slots.

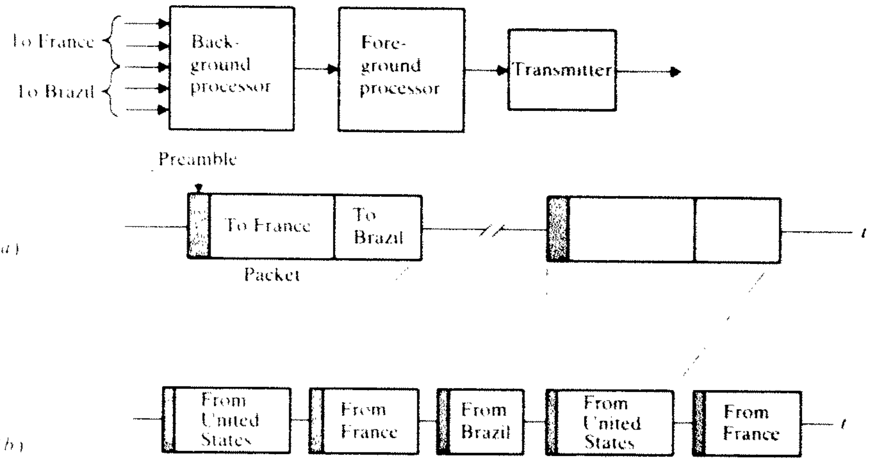


Figure 12.4-10 TDMA system. (*a*) Ground station transmitter; (*b*) transponder signal.

Analysis and design methods for packet-switching and computer networks are covered in textbooks such as Tanenbaum (1981). Spilker (1977) is devoted entirely to digital communication by satellite.

12.5 PROBLEMS

12.1-1 An analog waveform with $W = 15$ kHz is to be quantized to $q \geq 200$ levels and transmitted via an M-ary PCM signal having $M = 2^n$. Find the maximum allowed values of ν and f_s and the corresponding value of n when the available transmission bandwidth is $B_T = 50$ kHz.

12.1-2 Do Prob. 12.1-1 with $B_T = 80$ kHz.

12.1-3 *Hyperquantization* is the process whereby N successive quantized sample values are represented by a single pulse with q^N possible values. Describe how PCM with hyperquantization can achieve *bandwidth compression*, so $B_T < W$.

12.1-4 Suppose the PCM quantization error ϵ_k is specified to be no greater than $\pm P\%$ of the peak-to-peak signal range. Obtain the corresponding condition on ν in terms of M and P.

12.1-5 A voice signal having $W = 3$ kHz and $S_x = 1/4$ is to be transmitted via M-ary PCM. Determine values for M, ν, and f_s such that $(S/N)_D \geq 40$ dB if $B_T = 16$ kHz.

12.1-6 Do Prob. 12.1-5 with $(S/N)_D \geq 36$ dB and $B_T = 20$ kHz.

12.1-7 An audio signal with $S_x = 0.3$ is to be transmitted via a PCM system whose parameters must satisfy the standards for broadcast-quality audio transmission listed in Table 5.4-1. (*a*) If $M = 2$, then what are the required values of ν and B_T? (*b*) If $B_T = 4W$, then what's the minimum value of M?

12.1-8 Do Prob. 12.1-7 for high-fidelity audio transmission standards.

12.1-9‡ Consider the signal PDF in Fig. P12.1-9. Use Eq. (9) to show that uniform quantization with $q = 2^\nu$ levels and $\nu \geq 3$ yields $\sigma_q^2 = 1/3q^2$. Then calculate S_x and find $(S/N)_D$ in terms of q.

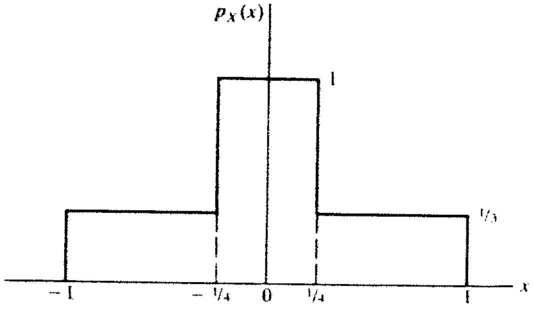


Figure P12.1-9