A typical binary data sequence is shown at the top of Figure 12.12. Since the modulation is 8-ary FSK, the bits are grouped three at a time to form symbols. In a *conventional* 8-ary FSK scheme, a single-sideband tone (offset from f_0, the *fixed* center frequency of the data band), would be transmitted according to an assignment like the one shown in the legend. The only difference in this FH/MFSK example is that the center frequency of the data band f_0 is *not fixed.* For each new symbol, f_0 hops to a new position in the hop bandwidth, and the entire data-band structure moves with it. In the example of Figure 12.12, the first symbol in the data sequence, 0 1 1, yields a tone 25 Hz above f_0. The diagram depicts f_0 with a dashed line and the symbol tone with a solid line. During the second symbol interval, f_0 has hopped to a new spectral location, as indicated by the dashed line. The second symbol, 1 1 0, dictates that a tone indicated by the solid line, 125 Hz below f_0, shall be transmitted. Similarly, the final symbol in this example, 0 0 1, calls for a tone 125 Hz above f_0. Again, the center frequency has moved, but the relative positions of the symbol tones remain fixed.

12.4.2 Robustness

A common dictionary definition describes the term *robustness* as the state of being strong and healthy; full of vigor; hardy. In the context of communications, the usage is not too different. Robustness characterizes a signal's ability to withstand impairments from the channel, such as noise, jamming, fading, and so on. A signal configured with multiple replicate copies, each transmitted on a different frequency, has a greater likelihood of survival than does a single such signal with equal total power. The greater the diversity (multiple transmissions, at different frequencies, spread in time), the more robust the signal against random interference.

The following example should clarify the concept. Consider a message consisting of four symbols: s_1, s_2, s_3, s_4. The introduction of diversity starts by repeating the message N times. Let us choose $N = 8$. Then, the repeated symbols, called *chips,* can be written.

$$s_1 s_1 s_1 s_1 s_1 s_1 s_1 s_1 s_2 s_2 s_2 s_2 s_2 s_2 s_2 s_2 s_3 s_3 s_3 s_3 s_3 s_3 s_3 s_3 s_4 s_4 s_4 s_4 s_4 s_4 s_4 s_4$$

Each chip is transmitted at a different hopping frequency (the center of the data bandwidth is changed for each chip). The resulting transmissions at frequencies f_i, $f_j, f_k, \ldots$ yield a more robust signal than without such diversity. A target-shooting analogy is that a pellet from a barrage of shotgun pellets has a better chance of hitting a target, compared with the action of a single bullet.

12.4.3 Frequency Hopping with Diversity

In Figure 12.13 we extend the example illustrated in Figure 12.12, with the additional feature of a chip repeat factor of $N = 4$. During each 20-ms symbol interval, there are now four columns, corresponding to the four separate chips to be transmitted for each symbol. At the top of the figure we see the same data sequence,

INTEL-1022, Part 7
9,584,262

with $R = 150$ bps, as in the earlier example; and we see the same 3-bit partitioning to form the 8-ary symbols. Each symbol is transmitted four times, and for each transmission the center frequency of the data band is hopped to a new region of the hopping band, under the control of a PN code generator. Therefore, for this example, each chip interval, T_c, is equal to $T/N = 20$ ms/4 = 5 ms in duration, and the hopping rate is now

$$\frac{NR}{\log_2 8} = 200 \text{ hops/s}$$

Notice that the spacing between frequency tones must change to meet the changed requirement for orthogonality. Since the duration of each FSK tone is now equal to the chip duration, that is, $T_c = T/N$, the minimum separation between tones is $1/T_c = N/T = 200$ Hz. As in the earlier example, Figure 12.13 illustrates that the center of the data band (plus the modulation structure) is shifted at each new chip time. The position of the solid line (transmission frequency) has the same relationship to the dashed line (center of the data band) for each of the chips associated with a given symbol.

12.4.4 Fast Hopping versus Slow Hopping

In the case of direct-sequence spread-spectrum systems, the term "chip" refers to the PN code symbol (the symbol of shortest duration in a DS system). In a similar sense for frequency hopping systems, the term "chip" is used to characterize the

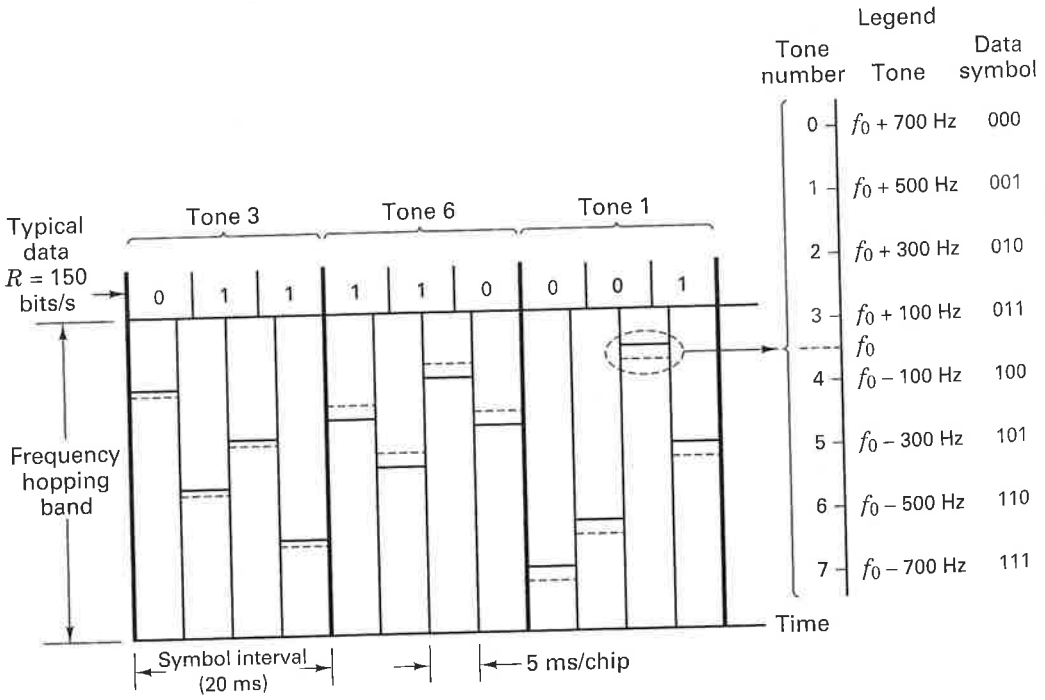
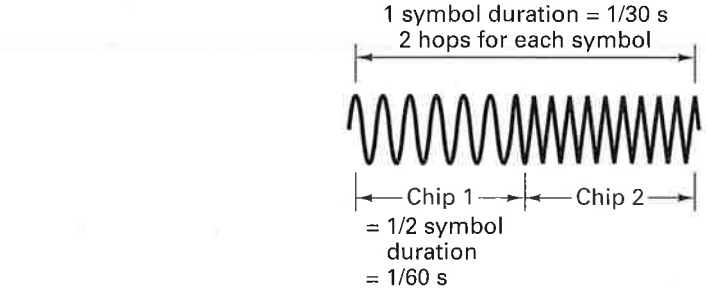
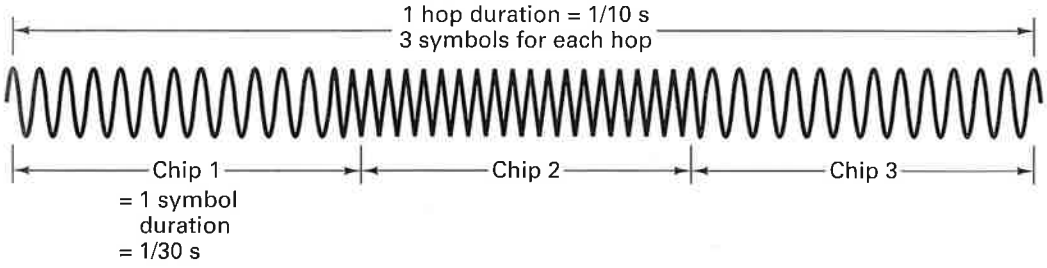


Figure 12.13 Frequency hopping example with diversity ($N = 4$).

shortest uninterrupted waveform in the system. Frequency hopping systems are classified as *slow-frequency hopping* (SFH), which means there are several modulation symbols per hop, or as *fast-frequency hopping* (FFH), which means that there are several frequency hops per modulation symbol. For SFH, the shortest uninterrupted waveform in the system is that of the data symbol; however, for FFH, the shortest uninterrupted waveform is that of the hop. Figure 12.14a illustrates an example of FFH; the data symbol rate is 30 symbols/s and the frequency hopping rate is 60 hops/s. The figure illustrates the waveform $s(t)$ over one symbol duration ($\frac{1}{30}$ s). The waveform change in (the middle of) $s(t)$ is due to a new frequency hop. In this example, a chip corresponds to a hop since the hop duration is shorter than the symbol duration. Each chip corresponds to half a symbol. Figure 14.14b illustrates an example of SFH; the data symbol rate is still 30 symbols/s, but the frequency hopping rate has been reduced to 10 hops/s. The waveform $s(t)$ is shown over a duration of three symbols ($\frac{1}{10}$ s). In this example, the hopping boundaries appear only at the beginning and end of the three-symbol duration. Here, the changes in the waveform are due to the modulation state changes; therefore, in this



(a)



(b)

Figure 12.14 Chip—in the context of an FH/MFSK system. (a) Example 1: Frequency hopping MFSK system with symbol rate = 30 symbols/s and hopping rate = 60 hops/s. 1 chip = 1 hop. (b) Example 2: Same as part (a) except hopping rate = 10 hops/s. 1 chip = 1 symbol.

example a chip corresponds to a data symbol, since the data symbol is shorter than the hop duration.

Figure 12.15a illustrates an FFH example of a binary FSK system. The diversity is $N = 4$. There are 4 chips transmitted per bit. As in Figure 12.13, the dashed line in each column corresponds to the center of the data band and the solid line corresponds to the symbol frequency. Here, for FFH, the chip duration is the hop duration. Figure 12.15b illustrates an example of an SFH binary FSK system. In this case, there are 3 bits transmitted during the time duration of a single hop. Here, for SFH, the chip duration is the bit duration. If this SFH example were changed from a binary system to an 8-ary system, what would the chip duration then correspond to? If the system were implemented as an 8-ary scheme, each 3 bits would be transmitted as a single data symbol. The symbol boundaries and the hop boundaries would then be the same, and the chip duration, the hop duration, and the symbol duration would all be the same.

12.4.5 FFH/MFSK Demodulator

Figure 12.16 illustrates the schematic for a typical fast frequency hopping MFSK (FFH/MFSK) demodulator. First, the signal is dehopped using a PN generator identical to the one used for hopping. Then, after filtering with a low-pass filter that

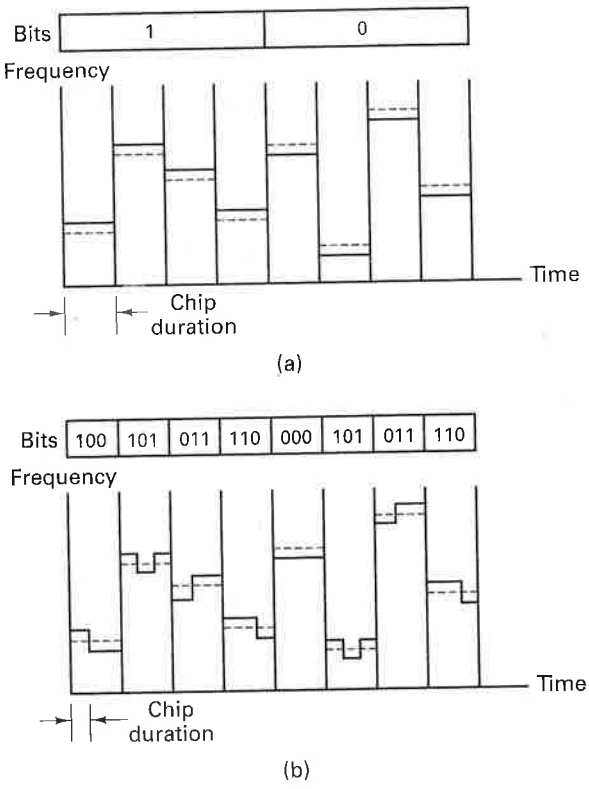


Figure 12.15 Fast hopping versus slow hopping in a binary system. (a) Fast-hopping example: 4 hops/bit. (b) Slow-hopping example: 3 bits/hop.

has a bandwidth equal to the data bandwidth, the signal is demodulated using a bank of M envelope or energy detectors. Each envelope detector is followed by a clipping circuit and an accumulator. The clipping circuit serves an important function in the presence of an intentional jammer or other strong unpredictable interference; it is treated in a later section. The demodulator does *not* make symbol decisions on a chip-by-chip basis. Instead, the energy from the N chips are accumulated, and after the energy from the Nth chip is added to the $N-1$ earlier ones, the demodulator makes a symbol decision by choosing the symbol that corresponds to the accumulator, z_i $(i = 1, 2, \ldots, M)$, with maximum energy.

12.4.6 Processing Gain

Equation (12.27) shows the general expression for processing gain as $G_p = W_{ss}/R$. In the case of direct-sequence spread-spectrum, W_{ss} was set equal to the chip rate R_{ch}. In the case of frequency hopping, Equation (12.27) still expresses the processing gain, but here we set W_{ss} equal to the frequency band over which the system may hop. We designate this band as the *hopping band* W_{hopping}, and thus the processing gain for frequency hopping systems is written as

$$G_p = \frac{W_{\text{hopping}}}{R} \tag{12.29}$$

12.5 SYNCHRONIZATION

For both DS and FH spread-spectrum systems, a receiver must employ a *synchronized* replica of the spreading or code signal to demodulate the received signal successfully. The process of synchronizing the locally generated spreading signal with

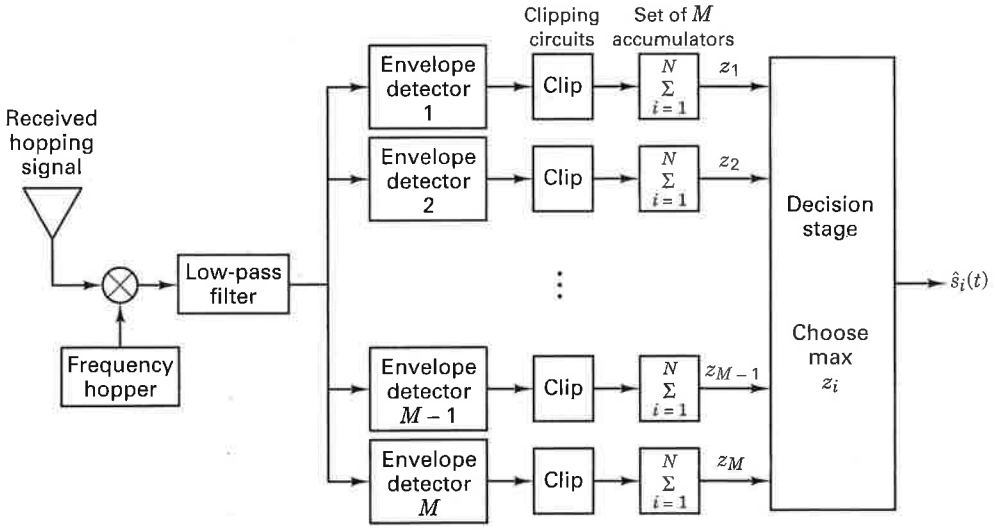


Figure 12.16 FFH/MFSK demodulator.

the received spread-spectrum signal is usually accomplished in two steps. The first step, called *acquisition,* consists of bringing the two spreading signals into *coarse* alignment with one another. Once the received spread-spectrum signal has been acquired, the second step, called *tracking,* takes over and continuously maintains the best possible waveform *fine* alignment by means of a feedback loop.

12.5.1 Acquisition

The acquisition problem is one of searching throughout a region of time and frequency uncertainty in order to synchronize the received spread-spectrum signal with the locally generated spreading signal. Acquisition schemes can be classified as coherent or noncoherent. Since the despreading process typically takes place before carrier synchronization, and therefore the carrier phase is unknown at this point, most acquisition schemes utilize noncoherent detection. When determining the limits of the uncertainty in time and frequency, the following items must be considered:

1. Uncertainty in the distance between the transmitter and the receiver translates into uncertainty in the amount of propagation delay.
2. Relative clock instabilities between the transmitter and the receiver result in phase differences between the transmitter and receiver spreading signals that will tend to grow as a function of elapsed time between synchronization.
3. Uncertainty of the receiver's relative velocity with respect to the transmitter translates into uncertainty in the value of Doppler frequency offset of the incoming signal.
4. Relative oscillator instabilities between the transmitter and the receiver result in frequency offsets between the two signals.

12.5.1.1 Correlator Structures

A common feature of all acquisition methods is that the received signal and the locally generated signal are first correlated to produce a measure of similarity between the two. This measure is then compared with a threshold to decide if the two signals are in synchronism. If they are, the tracking loop takes over.* If they are not, the acquisition procedure provides for a phase or frequency change in the locally generated code as a part of a systematic search through the receiver's phase and frequency uncertainty region, and another correlation is attempted.

Consider the direct-sequence *parallel-search* acquisition system shown in Figure 12.17. The locally generated code $g(t)$ is available with delays that are spaced one-half chip ($T_c/2$) apart. If the time uncertainty between the local code and the received code is N_c chips, and a complete parallel search of the entire time uncertainty region is to be accomplished in a single search time, $2N_c$ correlators are used. Each correlator simultaneously examines a sequence of λ chips, after which the $2N_c$

*Quite often to maintain a small false alarm probability, the threshold crossing must be further verified by a suitable verification algorithm before the tracking loop takes over [4].

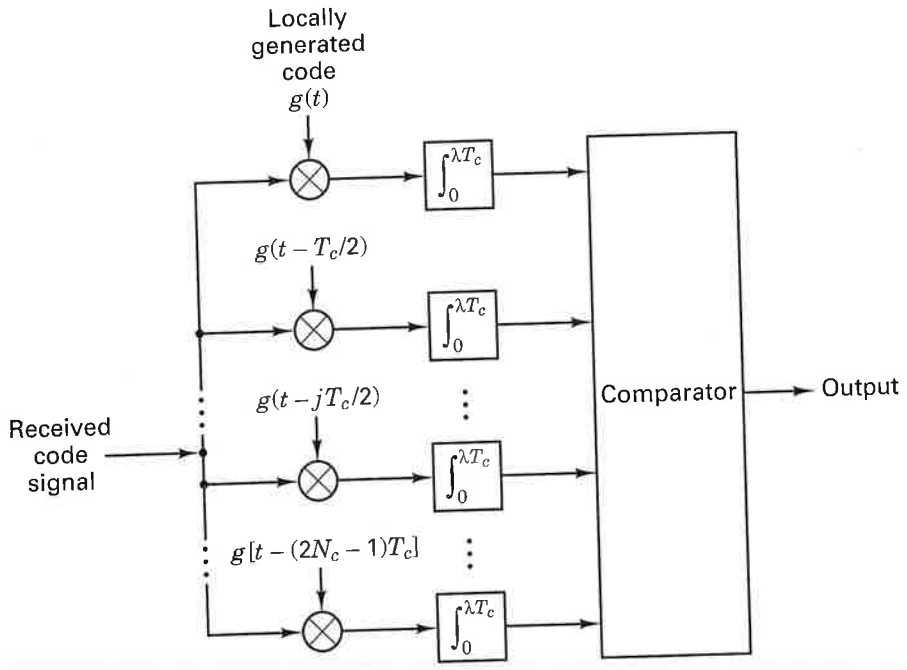


Figure 12.17 Direct-sequence parallel search acquisition.

correlator outputs are compared. The locally generated code, corresponding to the correlator with the largest output is chosen. Conceptually, this is the simplest of the search techniques; it considers all possible code positions (or fractional code positions) in parallel and uses a maximum likelihood algorithm for acquiring the code. Each detector output pertains to the identical observation of received signal plus noise. As λ increases, the synchronization error probability (i.e., the probability of choosing the incorrect code alignment) decreases. Thus, λ is chosen as a compromise between minimizing the probability of a synchronization error and minimizing the time to acquire.

Figure 12.18 illustrates a simple acquisition scheme for a frequency hopping system. Assume that a sequence of N consecutive frequencies from the hop sequence is chosen as a synchronization pattern (without data modulation). The N noncoherent matched filters each consists of a mixer followed by a bandpass filter (BPF) and a square-law envelope detector (an envelope detector followed by a square-law device). If the frequency hopping sequence is $f_1, f_2, \ldots, f_N$, delays are inserted into the matched filters so that when the correct frequency hopping sequence appears, the system produces a large output, indicating detection of the synchronization sequence. Acquisition can be accomplished rapidly because all possible code offsets are examined simultaneously. Note that the presence of bandpass filters (BPF) in Figure 12.18 indicates that the local oscillator frequencies $f_1, f_2, \ldots, f_N$ are chosen to have offsets by some intermediate frequency (IF) from

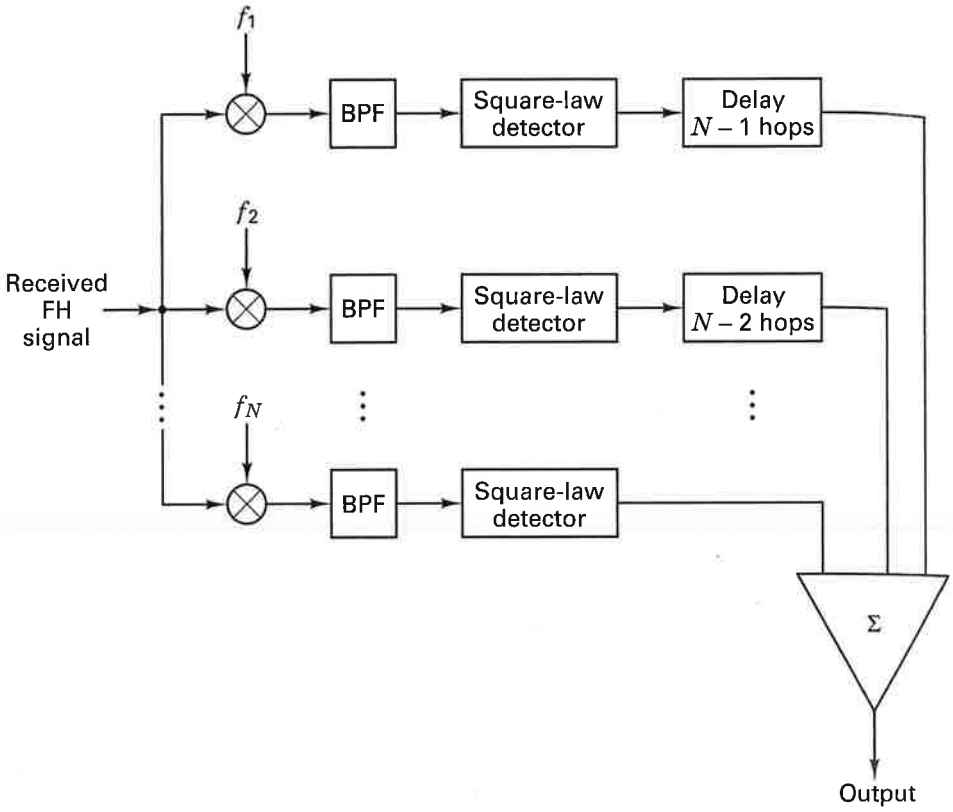


Figure 12.18 Frequency hopping acquisition scheme.

the expected received hop sequence. The same system can be implemented with local oscillator frequencies chosen (without offsets) so that the mixers yield baseband signals, and thus the filters would need to be low-pass filters (LPF). The mixers are typically complex, yielding in-phase and quadrature terms.

If, during each correlation, λ chips (each chip having a duration of T_c) are examined, the maximum time required for a fully parallel search is

$$(T_{\text{acq}})_{\max} = \lambda T_c \tag{12.30}$$

The mean acquisition time of a parallel search system can be approximated by noting that after integrating over λ chips, a correct decision will be made with probability P_D, called the *probability of detection.* If an incorrect output is chosen, an additional λ chips are again examined to make a determination of the correct output. Therefore, on the average, the acquisition time is [4]

$$\begin{aligned}\overline{T}_{\text{acq}} &= \lambda T_c P_D + 2\lambda T_c P_D(1 - P_D) + 3\lambda T_c P_D(1 - P_D)^2 + \cdots \\ &= \frac{\lambda T_c}{P_D}\end{aligned} \tag{12.31}$$

Since the required number of correlators or matched filters can be prohibitively large, fully parallel acquisition techniques are not usually used. In place of Figures 12.17 and 12.18, a single correlator or matched filter can be implemented that will *serially search* until synchronization is achieved. Naturally, trade-offs between fully parallel, fully serial, and combinations of the two involve hardware complexity versus time to acquire for the same uncertainty and chip rate.

12.5.1.2 Serial Search

A popular strategy for the acquisition of spread-spectrum signals is to use a single correlator or matched filter to serially search for the correct phase of the DS code signal or the correct hopping pattern of the FH signal. A considerable reduction in complexity, size, and cost can be achieved by a serial implementation that repeats the correlation procedure for each possible sequence shift. Figures 12.19 and 12.20 illustrate the basic configuration for DS and FH spread-spectrum schemes, respectively. In a stepped serial acquisition scheme for a DS system, the timing epoch of the local PN code is set, and the locally generated PN signal is correlated with the incoming PN signal. At fixed examination intervals of λT_c (search dwell time), where $\lambda >> 1$, the output signal is compared to a preset threshold. If the output is below the threshold, the phase of the locally generated code signal is incremented by a fraction (usually one-half) of a chip and the correlation is reexamined. When the threshold is exceeded, the PN code is assumed to have been acquired, the phase-incrementing process of the local code is inhibited, and the code tracking procedure will be initiated. In a similar scheme for FH systems, shown in Figure 12.20, the PN code generator controls the frequency hopper. Acquisition is accomplished when the local hopping is aligned with that of the received signal.

The maximum time required for a fully serial DS search, assuming that the search proceeds in half-chip increments, is

$$(T_{\text{acq}})_{\max} = 2N_c \lambda T_c \tag{12.32}$$

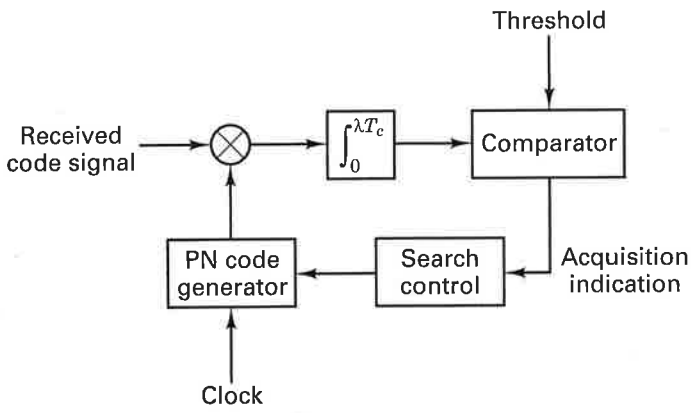


Figure 12.19 Direct-sequence serial search acquisition.

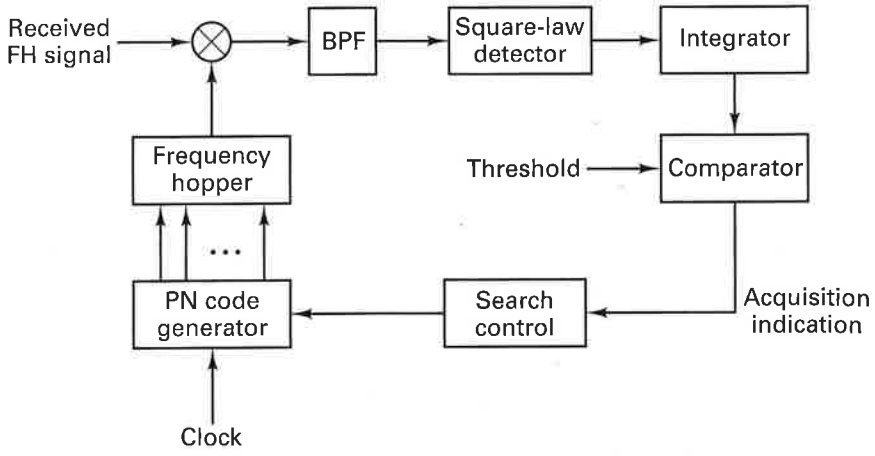


Figure 12.20 Frequency hopping serial search acquisition.

where the uncertainty region to be searched is N_c chips long. The mean acquisition time of a serial DS search system can be shown, for $N_c >> \frac{1}{2}$ chip, to be [10]

$$\overline{T}_{\text{acq}} = \frac{(2 - P_D)(1 + KP_{\text{FA}})}{P_D}(N_c \lambda T_c) \tag{12.33}$$

where λT_c is the search dwell time, P_D is the probability of correct detection, and P_{FA} is the probability of false alarm. We can regard the time interval $K\lambda T_c$, where $K >> 1$, as the time needed to verify a detection. Therefore, in the event of a false alarm, $K\lambda T_c$ seconds is the time penalty incurred. For $N_c >> \frac{1}{2}$ chip and $K << 2N_c$, the variance of the acquisition time is

$$(\text{var})_{\text{acq}} = (2N_c\lambda T_c)^2(1 + KP_{\text{FA}})\left(\frac{1}{12} + \frac{1}{P_D^2} - \frac{1}{P_D}\right) \tag{12.34}$$

12.5.1.3 Sequential Estimation

Another search technique, called *rapid acquisition by sequential estimation* (RASE), proposed by Ward [11], is illustrated in Figure 12.21. The switch is initially in position 1. The RASE system enters its best estimate of the first n received code chips into the n stages of its local PN generator. The fully loaded register defines a starting state from which the generator begins its operation. A PN sequence has the property that the next combination of register states depends only on the present combination of states. Therefore, if the first n received chips are correctly estimated, all the following chips from the local PN generator will be correctly generated. The switch is next thrown to position 2. If the starting state had been correctly estimated, the local generator generates the same sequences as the incoming waveform, in the absence of noise. If the correlator output after λT_c exceeds a preset threshold level, we assume that synchronization has occurred. If the output is less than the threshold, the switch is returned to position 1, the register is reloaded

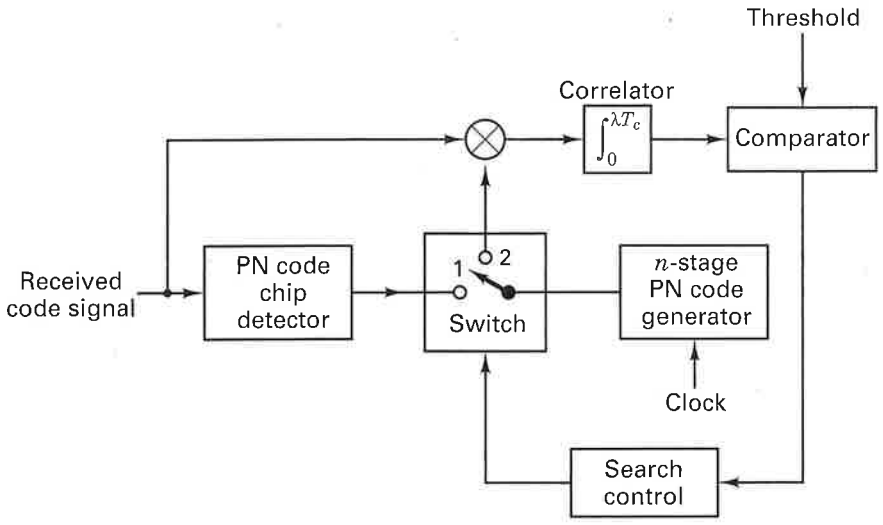


Figure 12.21 Rapid acquisition by sequential estimation.

with estimates of the next n received chips, and the procedure is repeated. Once synchronization has occurred, the system no longer needs estimates of the input code chips. We can calculate the *minimum* acquisition time for the case when no noise is present. The first n chips will be correctly loaded into the register, and therefore, the acquisition time is

$$T_{\text{acq}} = nT_c \tag{12.35}$$

While the RASE system has a rapid acquisition capability it has the drawback of being highly vulnerable to noise and interference signals. The reason for this is that the estimation process consists of a simple chip-by-chip hard-decision demodulation, without using the interference rejection benefits of the PN code.

For an extensive treatment of sequential estimation, see Reference [4].

12.5.2 Tracking

Once acquisition or coarse synchronization is completed, tracking or fine synchronization takes place. Tracking code loops can be classified as coherent or noncoherent. A coherent loop is one in which the carrier frequency and phase are known exactly so that the loop can operate on a baseband signal. A noncoherent loop is one in which the carrier frequency is not known exactly (due to Doppler effects, for example), nor is the phase. In most instances, since the carrier frequency and phase are not known exactly, a priori, a noncoherent code loop is used to track the received PN code. Tracking loops are further classified as a *full-time* early-late tracking loop, often referred to as a *delay-locked loop* (DLL), or as a *time-shared* early-late tracking loop, frequently referred to as a *tau-dither loop* (TDL). A basic

noncoherent DLL loop for a direct-sequence spread-spectrum system using binary phase shift keying (BPSK) is shown in Figure 12.22. The data $x(t)$ and the code $g(t)$ each modulate the carrier wave using BPSK, and as before in the absence of noise and interference, the received waveform can be expressed as

$$r(t) = A\sqrt{2P}\, x(t)\, g(t) \cos(\omega_0 t + \phi) \tag{12.36}$$

where the constant A is a system gain parameter and ϕ is a random phase angle in the range $(0, 2\pi)$. The locally generated code of the tracking loop is offset in phase from the incoming $g(t)$ by a time τ, where $\tau < T_c/2$. The loop provides *fine* synchronization by first generating two PN sequences $g(t + T_c/2 + \tau)$ and $g(t - T_c/2 + \tau)$ delayed from each other by one chip. The two bandpass filters are designed to pass the data and to average the product of $g(t)$ and the two PN sequences $g(t \pm T_c/2 + \tau)$. (See Reference [4] for the optimum filter bandwidth for a given filter type.) The square-law envelope detector eliminates the data since $|\, x(t)\,| = 1$. The output of each envelope detector is given approximately by

$$E_D \approx \mathbf{E}\left\{ \left| g(t) g\left(t \pm \frac{T_c}{2} + \tau\right) \right| \right\} = \left| R_g\left(\tau \pm \frac{T_c}{2}\right) \right| \tag{12.37}$$

where the operator $\mathbf{E}\{\cdot\}$ means *expected value* and $R_g(x)$ is the autocorrelation function of the PN waveform as shown in Figure 12.8. The feedback signal $Y(\tau)$ is shown in Figure 12.23. When τ is positive, the feedback signal $Y(\tau)$ instructs the voltage-controlled oscillator (VCO) to increase its frequency, thereby forcing τ to decrease, and when τ is negative, $Y(\tau)$ instructs the VCO to decrease, thereby forcing τ to increase. When τ is a suitably small number, $g(t)g(t + \tau) \approx 1$, yielding the

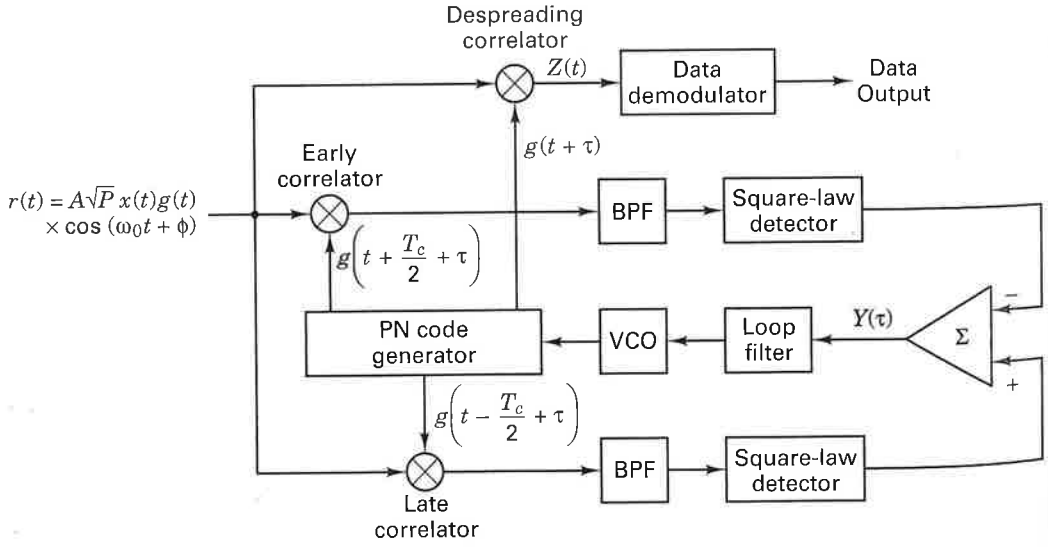
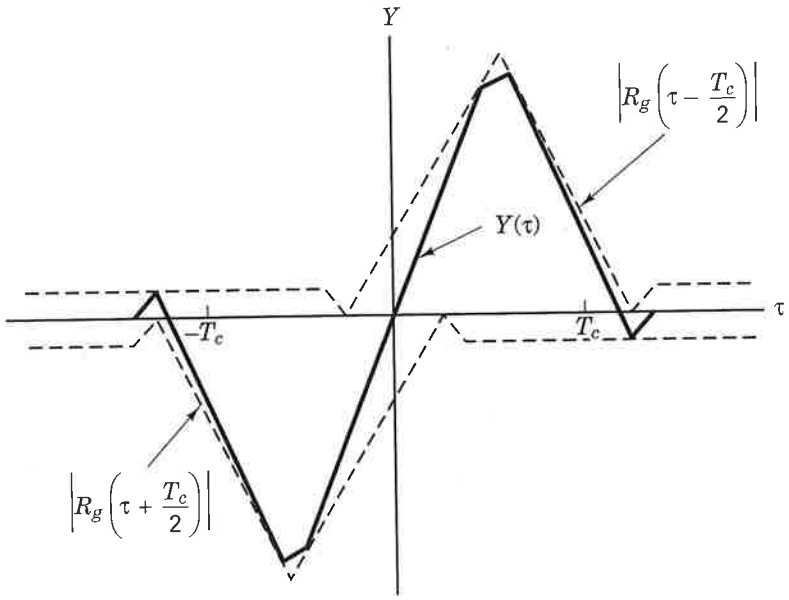


Figure 12.22 Delay-locked loop for tracking direct-sequence signals.



12.23 DLL feedback signal $Y(\tau)$.

despread signal $Z(t)$, which is then applied to the input of a conventional data demodulator. Detailed analysis of the DLL can be found in References [4, 12–14].

A problem with the DLL is that the early and late arms must be precisely gain balanced or else the feedback signal $Y(\tau)$ will be offset and will not produce a zero signal when the error is zero. This problem is solved by using a time-shared tracking loop in place of the full-time delay-locked loop. The time-shared loop time shares the use of the early-late correlators. The main advantages are that only one correlator need be used in the design of the loop, and further, that dc offset problems are reduced.

A problem with some control loops is that if things are going well and the loop is tracking accurately, the control signal is essentially zero. When the control signal is zero, the loop can get "confused" and do erratic things. This is especially the case in more sophisticated tracking loops that modify their own loop gain in response to the perceived environment. An offshoot of the time-shared tracking loop, called the *tau-dither loop* (TDL), shown in Figure 12.24, tends to deal with this potential problem by intentionally injecting a small error in the tracking correction, so that the loop kind of vibrates around the correct answer. This vibration is typically small, so that the loss in performance is minimal. This design has the advantage that only one correlator is needed to provide the code *tracking* function *and* the *despreading* function. Just as in the case of a DLL, the received signal is correlated with an early and a late version of the locally generated PN code. As shown in Figure 12.24, the PN code generator is driven by a clock signal whose phase is *dithered* back and forth with a square-wave switching function; this eliminates the necessity of ensuring identical transfer functions of the early and late

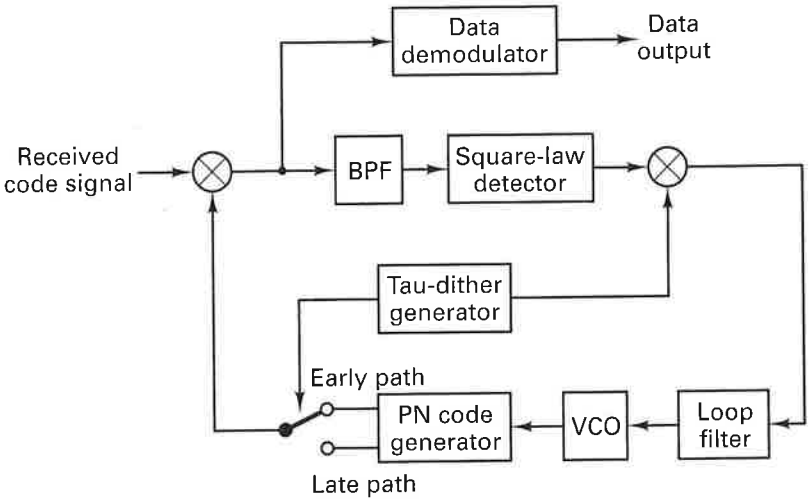


Figure 12.24 Tau-dither tracking loop.

paths. The signal-to-noise performance of the TDL is only about 1.1 dB worse than that of the DLL if the arm filters are designed properly [4]. (For a comprehensive treatment of synchronization of PN codes, see References [4, 15, 16].)

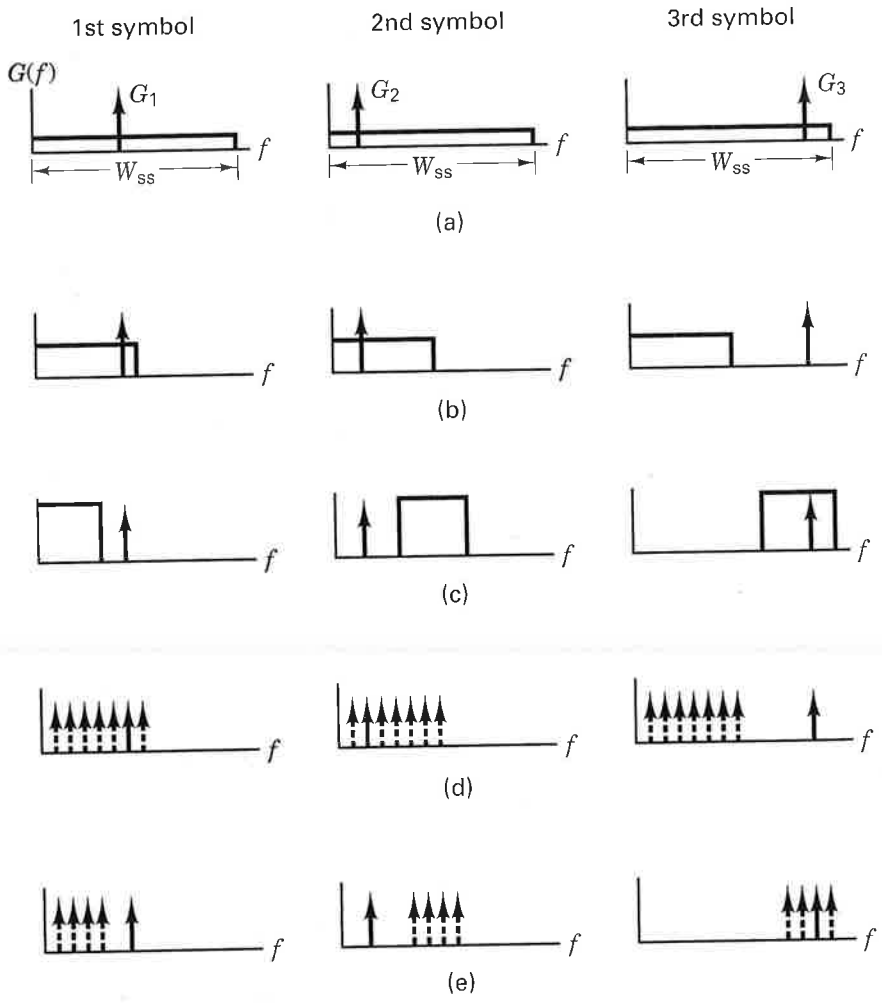
12.6 JAMMING CONSIDERATIONS

12.6.1 The Jamming Game

The goals of a jammer are to deny reliable communications to his adversary and to accomplish this at minimum cost. The goals of the communicator are to develop a jam-resistant communication system under the following assumptions: (1) complete invulnerability is not possible; (2) the jammer has a priori knowledge of most system parameters, such as frequency bands, timing, traffic, and so on; (3) the jammer has *no* a priori knowledge of the PN spreading or hopping codes. The signaling waveform should be designed so that the jammer cannot gain any appreciable jamming advantage by choosing a jammer waveform and strategy other than wideband Gaussian noise (i.e., being clever should gain nothing for the jammer). The fundamental design rule in specifying a jam-resistant system is to make it as costly as possible for the jammer to succeed in jamming the system.

12.6.1.1 Jammer Waveforms

There are many different waveforms that can be used for jamming communication systems. The most appropriate choice depends on the targeted system. Figure 12.25 shows power spectral density plots of examples of jammer waveforms versus a communicator's frequency hopped M-ary FSK (FH/MFSK) tone. The



12.25 Jammer waveforms. (a) Full-band noise. (b) Partial-band noise. (c) Stepped noise. (d) Partial-band tones. (e) Stepped tones.

range of the abscissa represents the spread-spectrum bandwidth W_{ss}. The three columns in the figure represent three instances in time (three hop times) when symbols having spectra G_1, G_2, and G_3, respectively, are being transmitted. Figure 12.25a illustrates a relatively low-level noise jammer occupying the full spread-spectrum bandwidth. In Figure 12.25b the jammer strategy is to trade bandwidth occupancy for greater power spectral density (the total power, or area under the curve, remains the same). The figure indicates that in this case, the jammer noise does not always share the same bandwidth region as the signal, but when it does, the effect can be destructive. In Figure 12.25c the noise jammer strategy is again to jam only part of the band, so that the jammer power spectral density can be increased, but in this case the jammer steps through different regions of the band at

random times, thus preventing the communicator from using adaptive techniques to avoid the jamming. In Figure 12.25d and e the jammer uses a group of tones, instead of a continuous frequency band, in partial-band (Figure 12.25d) and stepped fashion (Figure 12.25e). This is a technique most often used against FH systems. Another jamming technique, not shown in Figure 12.25, is a pulse jammer, consisting of pulse-modulated bandlimited noise. Unless otherwise stated, we shall assume that the jammer waveform is wideband noise and that the jammer strategy is to jam the entire bandwidth W_{ss} continuously. The effects of partial band jamming and pulse jamming are considered later.

12.6.1.2 Tools of the Communicator

The usual design goal for an anti-jam (AJ) communication system is to force a jammer to expend its resources over (1) a wide-frequency band, (2) for a maximum time, and (3) from a diversity of sites. The most prevalent design options are (1) frequency diversity, by the use of direct-sequence and frequency-hopping spread-spectrum techniques; (2) time diversity, by the use of time hopping; (3) spatial discrimination, by the use of a narrow-beam antenna, which forces a jammer to enter the receiver via an antenna sidelobe and hence suffer, typically, a 20- to 25-dB disadvantage, and (4) combinations of the previous three options.

12.6.1.3 *J/S* Ratio

In Chapter 5 we were concerned primarily with link error performance as a function of thermal noise interference. Emphasis was placed on the signal-to-noise ratio parameters—required E_b/N_0 and available E_b/N_0 for meeting a specified error performance. In this section we are similarly concerned with link error performance as a function of interference. However, here the source of interference is assumed to be wideband Gaussian noise power from a jammer in addition to thermal noise. Therefore, the SNR of interest is $E_b/(N_0 + J_0)$, where J_0 is the noise power spectral density due to the jammer. Unless otherwise specified, J_0 is assumed equal to J/W_{ss}, where J is the average received jammer power (jammer power referred to the receiver front end) and W_{ss} is the spread-spectrum bandwidth. Since the jammer power is generally much greater than the thermal noise power, the SNR of interest in a jammed environment is usually taken to be E_b/J_0. Therefore, similar to the thermal noise case, we define $(E_b/J_0)_{reqd}$ as the bit energy per jammer noise power spectral density *required* for maintaining the link at a specified error probability. The parameter E_b can be written as

$$E_b = ST_b = \frac{S}{R}$$

where S is the received signal power, T_b the bit duration, and R the data rate in bits/s. Then we can express $(E_b/J_0)_{reqd}$ as

$$\left(\frac{E_b}{J_0}\right)_{reqd} = \left(\frac{S/R}{J/W_{ss}}\right)_{reqd} = \frac{W_{ss}/R}{(J/S)_{reqd}} = \frac{G_p}{(J/S)_{reqd}} \tag{12.38}$$

where $G_p = W_{ss}/R$ represents the *processing gain*, and $(J/S)_{reqd}$ can be written

$$\left(\frac{J}{S}\right)_{reqd} = \frac{G_p}{(E_b/J_0)_{reqd}} \tag{12.39}$$

The ratio $(J/S)_{reqd}$ is a figure of merit that provides a measure of how *invulnerable* a system is to interference. Which system has better jammer-rejection capability: one with a larger $(J/S)_{reqd}$ or a smaller $(J/S)_{reqd}$? The *larger* the $(J/S)_{reqd}$, the *greater* is the system's noise rejection capability, since this figure of merit describes how much noise power relative to signal power is *required* in order to degrade the system's specified error performance. Of course, the communicator would like the communication system *not* to degrade at all.

Another way of describing the relationship in Equation (12.39) is as follows. An adversary would like to employ a jamming strategy that forces the effective $(E_b/J_0)_{reqd}$ to be as large as possible. The adversary may employ pulse, tone, or partial-band jamming rather than wideband noise jamming. A large $(E_b/J_0)_{reqd}$ implies a small $(J/S)_{reqd}$ ratio for a fixed processing again. This may force the communicator to employ a larger processing gain to increase the $(J/S)_{reqd}$. The system designer strives to choose a signaling waveform such that the jammer can gain no special advantage by using a jamming strategy other than wideband Gaussian noise.

12.6.1.4 Anti-Jam Margin

Sometimes the $(J/S)_{reqd}$ ratio is referred to as the *anti-jam* (AJ) *margin*, since it characterizes the system jammer-rejection capability. But this is not really a good use of the phrase since AJ margin usually means the safety margin against a *particular threat.* Using the same approach as in Chapter 5 (for calculating the margin against thermal noise), we can define the AJ margin as

$$M_{AJ}(\text{dB}) = \left(\frac{E_b}{J_0}\right)_r(\text{dB}) - \left(\frac{E_b}{J_0}\right)_{reqd}(\text{dB}) \tag{12.40}$$

where $(E_b/J_0)_r$ is the E_b/J_0 *actually received.* Following the same format as Equation (12.38), we can express $(E_b/J_0)_r$ as

$$\left(\frac{E_b}{J_0}\right)_r = \frac{G_p}{(J/S)_r} \tag{12.41}$$

where $(J/S)_r$, or simply J/S, is the ratio of the actually received jammer power to signal power. Later, we develop an expression for received E_b/I_0, similar to Equation (12.41), where I_0 is the interference power spectral density due to other users in a CDMA cellular system. The concept of computing such a bit-energy to interference ratio is the same, whether the interference stems from a jammer, an accidental interferer, or other users who are authorized to share the same spectral region.

We now combine Equation (12.41) with Equations (12.38) and (12.40), as follows:

$$M_{AJ}(\text{dB}) = \frac{G_p}{(J/S)_r}(\text{dB}) - \frac{G_p}{(J/S)_{reqd}}(\text{dB}) \tag{12.42}$$

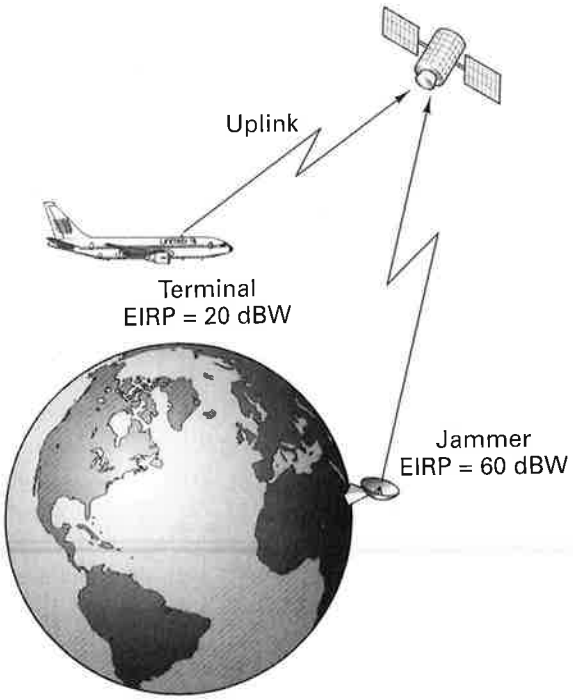


Figure 12.26 Satellite jamming scenario.

$$= \left(\frac{J}{S}\right)_{\text{reqd}} (\text{dB}) - \left(\frac{J}{S}\right)_{r} (\text{dB}) \tag{12.43}$$

Example 12.2 Satellite Jamming

Figure 12.26 illustrates a satellite jamming scenario. The airplane terminal is equipped with a frequency hopping (FH) spread-spectrum system transmitting with an EIRP_T = 20 dBW. The data rate is R = 100 bits/s. The jammer is transmitting wideband Gaussian noise, continually, with an EIRP_J = 60 dBW. Assume that $(E_b/J_0)_{\text{reqd}}$ = 10 dB and that the path loss is identical for both the airplane terminal and the jammer.

(a) Should the communicators be concerned more with the jamming of the uplink or with that of the downlink?

(b) If it is desired to have an AJ margin of 20 dB, what should be the value of the hopping bandwidth W_{ss}?

Solution

(a) Jamming the uplink is of much greater concern, since such single-point interference could degrade the communications of a multitude of terminals that are simultaneously using the satellite transponder. To achieve an equivalent degradation by jamming the downlink, the jammer would have to jam each of the receiving terminals. Downlink jamming is of some concern for critical military missions, but of less concern than uplink jamming.

(b) With the assumption that the path loss is the same for both the communicator and the jammer, we can replace $(J/S)_r$ in Equation (12.43) with the ratio of *transmitted* jammer-to-signal power, $\text{EIRP}_J/\text{EIRP}_T$. Therefore, we can write

$$M_{\text{AJ}}\,(\text{dB}) = (J/S)_{\text{reqd}}\,(\text{dB}) + \text{EIRP}_T\,(\text{dBW}) - \text{EIRP}_J\,(\text{dBW})$$

$$= G_p\,(\text{dB}) - \left(\frac{E_b}{J_0}\right)_{\text{reqd}}(\text{dB}) + \text{EIRP}_T\,(\text{dBW}) - \text{EIRP}_J\,(\text{dBW})$$

$$G_p = 20\text{ dB} + 10\text{ dB} - 20\text{ dBW} + 60\text{ dBW} = 70\text{ dB}$$

$$W_{\text{ss}} = G_p\,(\text{dB}) + R\,(\text{dB-Hz}) = 70\text{ dB} + 20\text{ dB-Hz}$$

$$= 90\text{ dB-Hz} = 1\text{ GHz}$$

Example 12.3 Satellite Downlink Jamming

In Example 12.2 the distance from the transmitting airplane to the receiving satellite and the distance from the jammer to the satellite were assumed identical. Certainly, the closer the jammer gets to the receiver, the greater will be the jamming interference. Consider a downlink jamming scenario where the satellite $\text{EIRP}_s = 35$ dBW, the jammer $\text{EIRP}_J = 60$ dBW, the space loss from the satellite to the receiving terminal is $L_s = 200$ dB, and the space loss from the jammer to the receiving terminal is $L_s' = 160$ dB. How much processing gain is needed to close the link with an AJ margin of 0 dB? Assume that $(E_b/J_0)_{\text{reqd}} = 10$ dB.

Solution

For the downlink jamming scenario the proximity of the jammer to the receiving airplane is much closer than that of the satellite to the airplane. These distances show up as the space losses in the $(J/S)_r$ term of Equation (12.43), as follows:

$$M_{\text{AJ}}\,(\text{dB}) = \left(\frac{J}{S}\right)_{\text{reqd}}(\text{dB}) - \left(\frac{J}{S}\right)_r(\text{dB})$$

where

$$\left(\frac{J}{S}\right)_r(\text{dB}) = \text{EIRP}_J\,(\text{dBW}) - L_s'\,(\text{dB}) - \text{EIRP}_s\,(\text{dBW}) + L_s\,(\text{dB})$$

and

$$\left(\frac{J}{S}\right)_{\text{reqd}}(\text{dB}) = \frac{W_{\text{ss}}}{R}(\text{dB}) - \left(\frac{E_b}{J_0}\right)_{\text{reqd}}(\text{dB})$$

Combining the above equations, and solving for processing gain, $G_p = W_{\text{ss}}/R$, yields

$$G_p\,(\text{dB}) = 75\text{ dB}$$

12.6.2 Broadband Noise Jamming

If the jamming signal is modeled as a zero-mean wide-sense stationary Gaussian noise process with a flat power spectral density over the frequency range of interest, then for a fixed jammer received power, J, the jammer power spectral density J_0' is equal to J/W, where W is the bandwidth that the jammer chooses to occupy. If the jammer strategy is to jam the entire spread-spectrum bandwidth, W_{ss}, with its

fixed power, the jammer is referred to as a wideband or *broadband jammer,* and the jammer power spectral density is

$$J_0 = \frac{J}{W_{ss}} \tag{12.44}$$

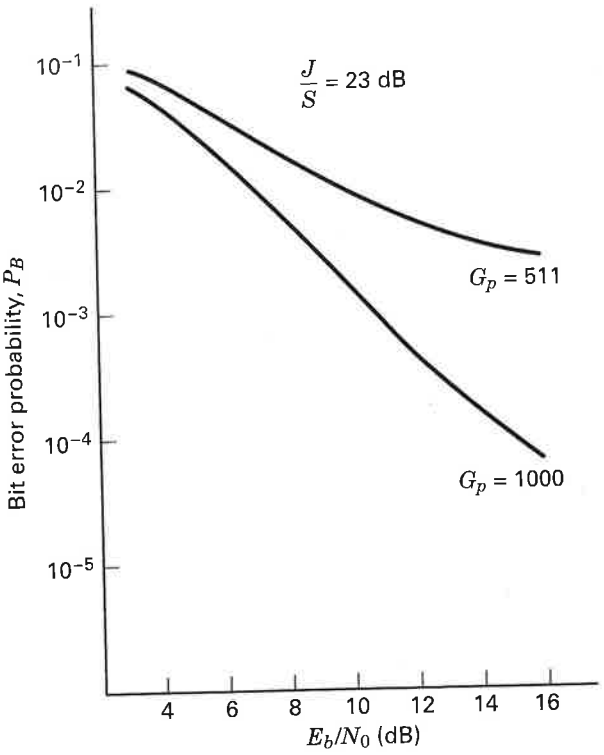
In Chapter 4 it was shown that the bit error probability P_B for a coherently demodulated BPSK system (without channel coding) is

$$P_B = Q\left(\sqrt{\frac{2E_b}{N_0}}\right) \tag{12.45}$$

where $Q(x)$ is defined in Equations (3.43) and (3.44) and tabulated in Table B.1. The single-sided noise power spectral density N_0 represents thermal noise at the front end of the receiver. The presence of the jammer increases this noise power spectral density from N_0 to $(N_0 + J_0)$. Thus the average bit error probability for a coherent BPSK system in the presence of broadband jamming is

$$P_B = Q\left(\sqrt{\frac{2E_b}{N_0 + J_0}}\right) = Q\left[\sqrt{\frac{2E_b/N_0}{1 + (E_b/N_0)(J/S)/G_p}}\right] \tag{12.46}$$

When P_B is plotted versus E_b/N_0 for a given J/S ratio, the resulting curves are such as those in Figure 12.27 [6, 21]. The curves in Figure 12.27, shown for two different



12.27 Bit-error probability versus E_b/N_0 for a given J/S ratio. (Reprinted with permission from R. L. Pickholtz, D. L. Schilling, and L. B. Milstein, "Theory of Spread-Spectrum Communications—A Tutorial," *IEEE Trans. Commun.*, vol. COM30, no. 5, May 1982, Fig. 11, p. 866 © 1982 IEEE.)

values of processing gain, *tend to flatten out* as E_b/N_0 increases, indicating that for a given ratio of jammer power to signal power, the jammer will cause some irreducible error probability. The only way to reduce this error probability is to increase the processing again.

12.6.3 Partial-Band Noise Jamming

A jammer can often increase the degradation to a FH system by employing *partial-band* jamming. Assuming that the frequency hopped modulation format is noncoherently detected binary FSK, the probability of a bit error, from Equation (4.96), is

$$P_B = \frac{1}{2}\exp\left(-\frac{E_b}{2N_0}\right) \tag{12.47}$$

Let us define a parameter, ρ, where $0 < \rho \leq 1$, representing the fraction of the band being jammed. The jammer can trade bandwidth jammed for in-band jammer power, such that by jamming a band $W = \rho W_{ss}$, the jammer noise power spectral density can be concentrated to a level J_0/ρ, thus maintaining a constant average jamming received power J where $J = J_0 W_{ss}$.

In the case of partial-band jamming, a specific transmitted symbol will be received unjammed, with probability $(1 - \rho)$, and will be perturbed by jammer power with spectral density J_0/ρ, with probability ρ. Therefore, the average bit error probability can be written from Equation (12.47), as follows:

$$P_B = \frac{1-\rho}{2}\exp\left(-\frac{E_b}{2N_0}\right) + \frac{\rho}{2}\exp\left[-\frac{E_b}{2(N_0 + J_0/\rho)}\right] \tag{12.48}$$

Since, in a jamming environment, it is often the case that $J_0 >> N_0$, we can simplify Equation (12.48) to the form

$$P_B \approx \frac{\rho}{2}\exp\left(-\frac{\rho E_b}{2J_0}\right) \tag{12.49}$$

Figure 12.28 illustrates the probability of bit error versus E_b/J_0 for various values of the fraction, ρ. Clearly, the jammer would choose the fraction $\rho = \rho_0$ that maximizes P_B. Notice that ρ_0 decreases with increasing values of E_b/J_0 (see the ρ_0 locus in Figure 12.28). An expression for ρ_0 is easily found by differentiation (setting $dP_B/d\rho = 0$ and solving for ρ). This yields

$$\rho_0 = \begin{cases} \dfrac{2}{E_b/J_0} & \text{for } \dfrac{E_b}{J_0} > 2 \\ 1 & \text{for } \dfrac{E_b}{J_0} \leq 2 \end{cases} \tag{12.50}$$

In this case, $(P_B)_{max}$ is given by

$$(P_B)_{\max} = \begin{cases} \dfrac{e^{-1}}{E_b/J_0} & \text{for } \dfrac{E_b}{J_0} > 2 \\ \dfrac{1}{2}\exp\left(-\dfrac{E_b}{2J_0}\right) & \text{for } \dfrac{E_b}{J_0} \le 2 \end{cases} \tag{12.51}$$

where e is the base of the natural logarithm ($e = 2.7183$). This result is dramatic; the effect of a worst-case partial-band jammer on a system with spread spectrum *but without coding* changes the exponential relationship of Equation (12.49) into the inverse linear one of Equation (12.51). The ρ_0 locus in Figure 12.28 illustrates the

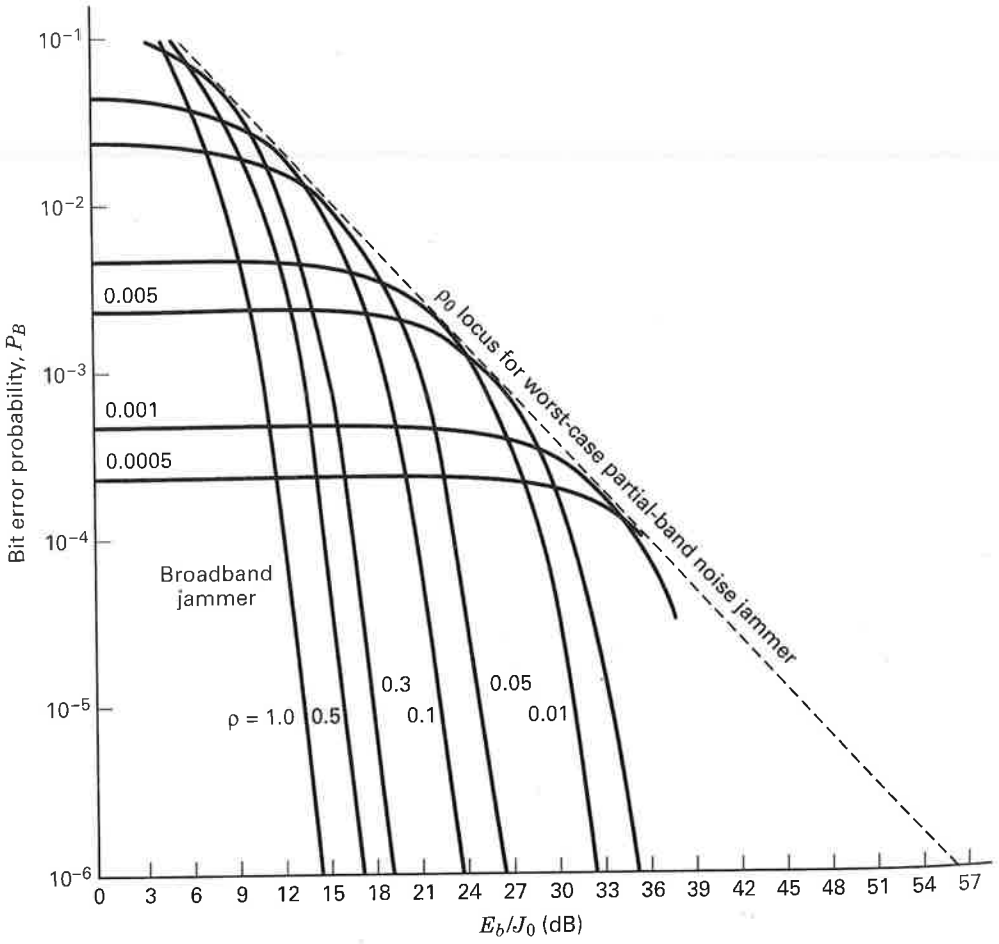


Figure 12.28 Partial-band noise jammer (FH/BFSK signaling). (Reprinted from M. K. Simon, J. K. Omura, R. A. Scholtz, and B. K. Levitt, *Spread Spectrum Communications*, Vol. 1, Fig. 3.24, p. 173. © 1985, with permission of the publisher, Computer Science Press, Inc., 1803 Research Blvd., Rockville, MD., 20850, USA.)

P_B versus E_b/J_0 performance for the worst-case partial-band jammer. Here at 10^{-6} bit-error probability there is over 40-dB difference between broadband noise jamming and the worst-case partial-band jamming for the same jamming power [4, 22]. Hence, an intelligent jammer, with fixed finite power, can produce significantly greater degradation with partial-band jamming than is possible with broadband jamming. Forward error correction (FEC) coding with appropriate interleaving can mitigate this degradation [9]. In fact, for codes with low enough rates, FEC can *force* a partial-band jammer to be a worst-case jammer only when operating as a broadband jammer [23, 24].

12.6.4 Multiple-Tone Jamming

In the case of *multiple-tone jamming,* the jammer divides its total received power, J, into distinct, equal-power, random-phase CW tones. These are distributed over the spread-spectrum bandwidth, W_{ss}, according to some strategy [9]. The analysis of the effects of tone jamming is more complicated than that of noise jamming, especially for DS systems. Therefore, the effect of a despread tone is often approximated as Gaussian noise. Reference [25] provides analysis of the performance of DS systems in the presence of multiple-tone interference. For a noncoherent FH/FSK system operating in the presence of partial-band tone jamming, the performance is often assumed the same as that of partial-band noise jamming [26]. However, multiple-CW-tone jamming can be more effective than partial-band noise against FH/MFSK signals because CW tones are the most efficient way for a jammer to inject energy into noncoherent detectors [8]. References, [8, 9, 26, 27] provide extensive treatment and analysis of the performance of various communications systems in the presence of various types of jammers.

In the FFH/MFSK demodulator of Figure 12.16, a chip-clipping circuit is shown between each envelope detector and accumulator. The function of such a circuit in a tone-jamming environment can best be understood with the aid of the example shown in Figure 12.29. An 8-ary FSK frequency-hopping system with no diversity, indicated in Figure 12.29a, is compared with a *fast* frequency-hopping system that combines chip repeating ($N = 4$ in this example) with the clipping of each chip, indicated in Figure 12.29b. Each row in the figures represents one of the $M = 8$ accumulators shown in Figure 12.16. The presence of a signal in the accumulator is indicated by a vector. In Figure 12.29a we see that, for a particular frequency hop, the data band is occupied by a received message symbol with received signal power S. If, by chance, a jamming tone with received power J, where $J \geq S$, falls on a different tone within this data band during the same hop, the detector would not be able to decide reliably on the correct symbol.

In Figure 12.29b, the communicator's four chips (the length of each vector is a measure of the clipped signal power, S') sum to the maximum capacity of the accumulator. If the jammer tones, by chance, fall in the same spectral region as that of the signal, they will not confuse the detector, since the jamming tones are also clipped to the same level, $J' = S'$, as the signal chips. In Figure 12.29b, two of the jamming tones fall in the data band, but because they are clipped, there is no confusion about the correct symbol decision.

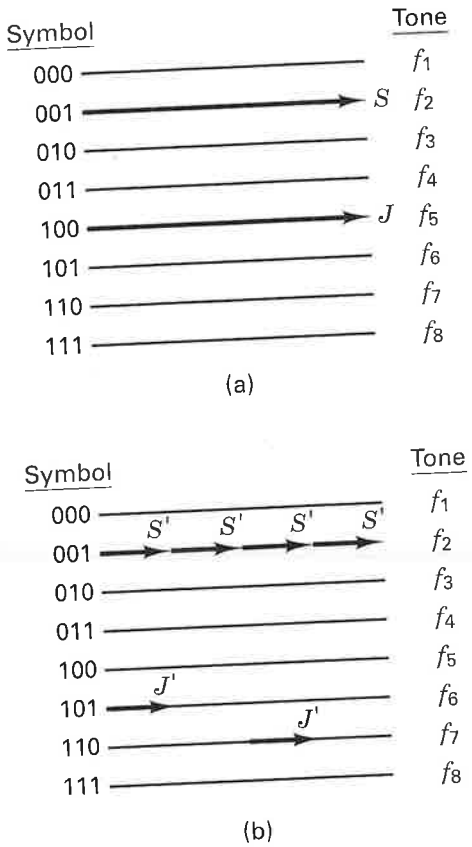


Figure 12.29 Fast-hopping symbol repeat versus tone jamming. (a) One frequency hop. (b) Four frequency hops.

12.6.5 Pulse Jamming

Consider a spread-spectrum DS/BPSK communication system in the presence of a pulse-noise jammer. A pulse-noise jammer transmits pulses of bandlimited white Gaussian noise having a time-averaged received power J, although the actual power during a jamming pulse duration is larger. Assume that the jammer can choose the center frequency and bandwidth of the noise to be the same as the receiver's center frequency and bandwidth. Assume also that the jammer can trade duty cycle for increased (concentrated) jammer power, such that if the jamming is present for a fraction $0 < \rho < 1$ of the time, then during this time, the jammer power spectral density is increased to a level J_0/ρ, thus maintaining a constant time-averaged power J (where $J = J_0 W_{ss}$ and W_{ss} is the system spread-spectrum bandwidth).

The bit error probability P_B for a coherently demodulated BPSK system (without channel coding) was given in Equation (12.45):

$$P_B = Q\left(\sqrt{\frac{2E_b}{N_0}}\right)$$

The single-sided noise power spectral density N_0 represents thermal noise at the front end of the receiver. The presence of the jammer increases this noise power spectral density from N_0 to $(N_0 + J_0/\rho)$. Since the jammer transmits with duty cycle ρ, the average bit-error probability is

$$P_B = (1 - \rho)Q\left(\sqrt{\frac{2E_b}{N_0}}\right) + \rho Q\left(\sqrt{\frac{2E_b}{N_0 + J_0/\rho}}\right) \tag{12.52}$$

We can generally assume that in a jamming environment, N_0 can be neglected. Therefore, we can write

$$P_B \approx \rho Q\left(\sqrt{\frac{2E_b\rho}{J_0}}\right) \tag{12.53}$$

The jammer will, of course, attempt to choose the duty cycle ρ that maximizes P_B. Figure 12.30 illustrates P_B for various values of ρ. The value of $\rho = \rho_0$ that maximizes P_B decreases with increasing values of E_b/J_0, as was the case with partial-band jamming. This is seen by differentiating Equation (12.53) to obtain [4]

$$\rho_0 = \begin{cases} \dfrac{0.709}{E_b/J_0} & \text{for } \dfrac{E_b}{J_0} > 0.709 \\ 1 & \text{for } \dfrac{E_b}{J_0} \leq 0.709 \end{cases} \tag{12.54}$$

which results in the maximum bit error probability

$$(P_B)_{\max} = \begin{cases} \dfrac{0.083}{E_b/J_0} & \text{for } \dfrac{E_b}{J_0} > 0.709 \\ Q\left(\sqrt{\dfrac{2E_b}{J_0}}\right) & \text{for } \dfrac{E_b}{J_0} \leq 0.709 \end{cases} \tag{12.55}$$

The effect of a worst-case pulse jammer upon a system with spread spectrum *but without coding* changes the complementary error function relationship of Equation (12.53) into the inverse linear one of Equation (12.55). As a result, at an error probability of 10^{-6}, there is about a 40-dB difference in E_b/J_0 between the broadband jammer and the worst-case pulse jammer (see Figure 12.30). For the same jammer power, the jammer can do considerably more harm to an uncoded DS/BPSK system with pulse jamming than with constant power jamming. The effect of a pulse-noise jammer on uncoded DS/BPSK is similar to the effect of a partial-band noise jammer on uncoded FH/BFSK, treated in Section 12.6.3. In both cases considerable degradation is brought about by concentrating more jammer power on a fraction of the transmitted uncoded symbols. Forward error correction coding with appropriate interleaving can almost fully restore this degraded performance [8, 23–25, 28].

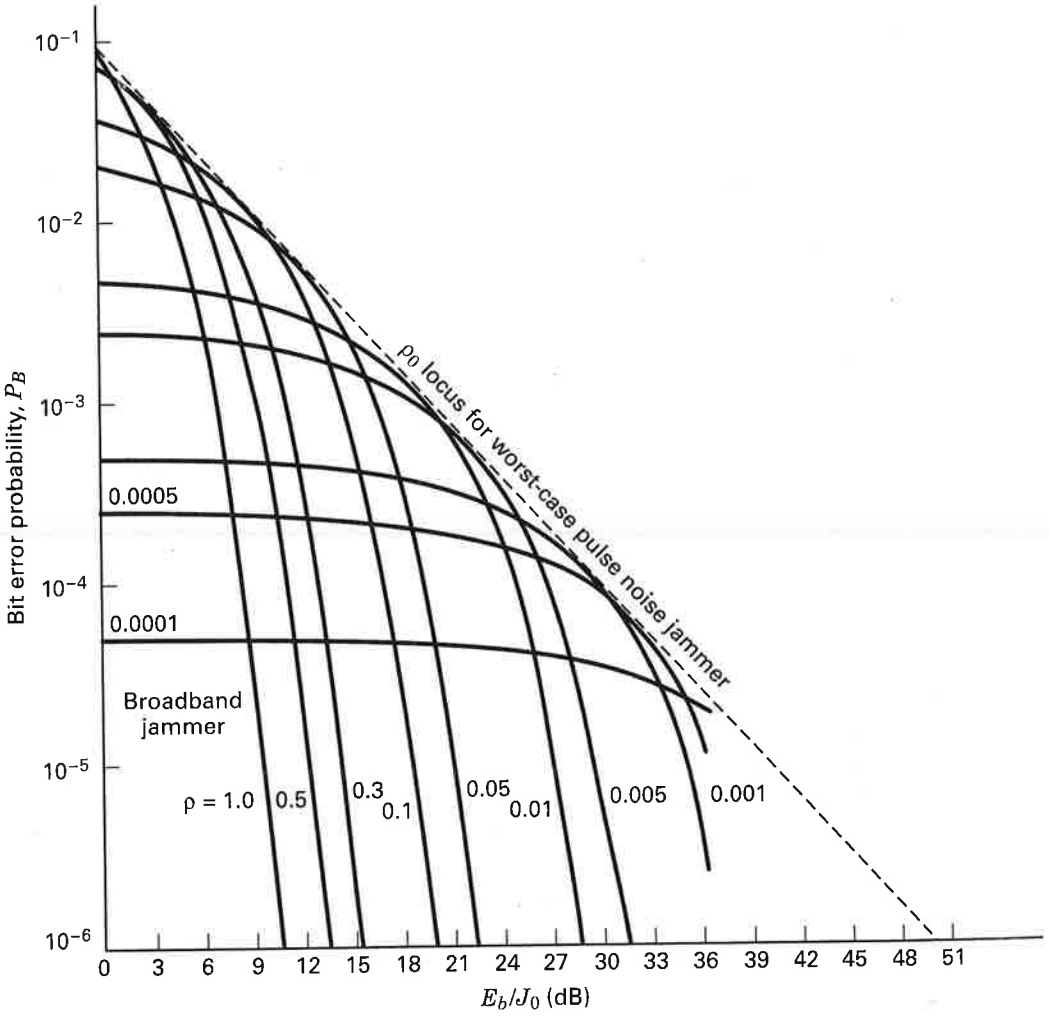


Figure 12.30 Pulse noise jammer (DS/BPSK signaling). (Reprinted from M. K. Simon, J. K. Omura, R. A. Scholtz, and B. K. Levitt, *Spread Spectrum Communications*, Vol. 1, Fig. 3.7, p. 150 © 1985, with permission of the publisher, Computer Science Press, Inc., 1803 Research Blvd., Rockville, Md. 20850 USA.)

12.6.6 Repeat-Back Jamming

In Examples 12.2 and 12.3 we considered an FH spread-spectrum system performance against a broadband Gaussian noise jammer. Notice that the frequency hopping rate did not enter into the margin computations. Isn't this disturbing? Intuitively, it would seem that the faster the frequency hops, the easier it is to "hide" the signal from the jammer. If the hopping rate truly does not enter into the computations, why not hop only once a day or once a week? The answer is that the

measure of jammer-rejection capability, namely processing gain, G_p, is based on the assumption that the jammer is a "dumb" jammer; that is, the jammer knows the extent of the spread-spectrum bandwidth, W_{ss}, but does *not* know the exact spectral location of the signal at any moment in time. We assume that the hopping rate is *fast enough* to preclude the jammer from monitoring the transmitted signal so as to usefully change this jamming strategy. Under what condition is this assumption questionable? There are "smart" jammers that are known as *repeat-back jammers* or *frequency-follower (FF) jammers.* These jammers monitor a communicator's signal (usually via a sidelobe beam from the transmitting antenna). They possess wideband receivers and high-speed signal processing capability that enable them to rapidly concentrate their jamming signal power in the spectral vicinity of a communicator's FH/FSK signal. By so doing, the smart jammer can increase the jamming power in the communicator's instantaneous bandwidth, thereby gaining an advantage over a wideband jammer. Notice that this strategy is useful only against frequency-hopping spread-spectrum signals. In direct-sequence systems, there is no instantaneous narrowband signal for the jammer to detect.

What can be done to defeat the repeat-back jammer? One method is to simply hop so fast that by the time the jammer receives, detects, and transmits the jamming signal, the communicator is already transmitting at a *new* hop (which of course will be unaffected by jamming at the frequency of the prior hop). The following example should make this point clear.

Example 12.4 Fast Hopping to Evade the Repeat-Back Jammer

Assume that a repeat-back jammer is located $d = 30$ km away from the communicator. Assume further that the jammer can monitor any uplink transmission from the com-

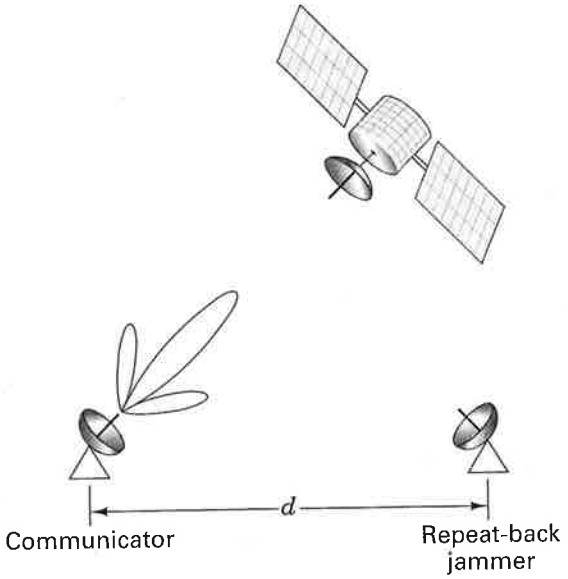


Figure 12.31 Example of fast hopping to evade the repeat-back jammer.

municator to a nearby satellite, as shown in Figure 12.31. How fast must the communicator hop his frequency to evade the repeat-back jammer? Assume that the jammer can change its jamming frequency in zero time, and that the only differential delay between the communicator's uplink signal and the jamming uplink signal is the propagation delay from the communicator to the jammer.

Solution

To ensure that the communicator's tone transmission and the jammer's attempt to disrupt that tone do not overlap in time, it is necessary that the duration of each hop have the value

$$T_{\text{hop}} \leq \frac{d}{c} = \frac{3 \times 10^4 \text{ m}}{3 \times 10^8 \text{ m/s}} = 10^{-4} \text{ s}$$

where c is the speed of light. Then $R_{\text{hop}} \geq 10{,}000$ hops/s.

12.6.7 BLADES System

Another technique capable of defeating the repeat-back jammer dates back to the mid-1950s, when Sylvania engineers developed a system named the *Buffalo Laboratories Application of Digitally Exact Spectra,* or BLADES. The system used its code generator to independently select two new frequencies for each bit; the *final choice* of the frequency tone actually transmitted was dictated by the data bit about to be transmitted. Figure 12.32 illustrates a typical data stream of binary ones and zeros, called *marks* and *spaces,* respectively, and a sequence of frequency pairs f_1 and f_1', f_2 and f_2', ... The appearance of a mark dictates the choice of frequency f_i, while the appearance of a space dictates the choice of frequency f_i'. As shown in the figure, the data stream in this example gives rise to the sequence of transmitted tones, f_1', f_2, f_3', f_4', f_5, and so on. How can such a system defeat a repeat-back jammer? The jammer monitors the transmissions and sends up energy in the neighborhood of the frequencies it perceives. The modulation of the BLADES system has no structure in the usual sense; either there *is* energy present or there is *no* energy

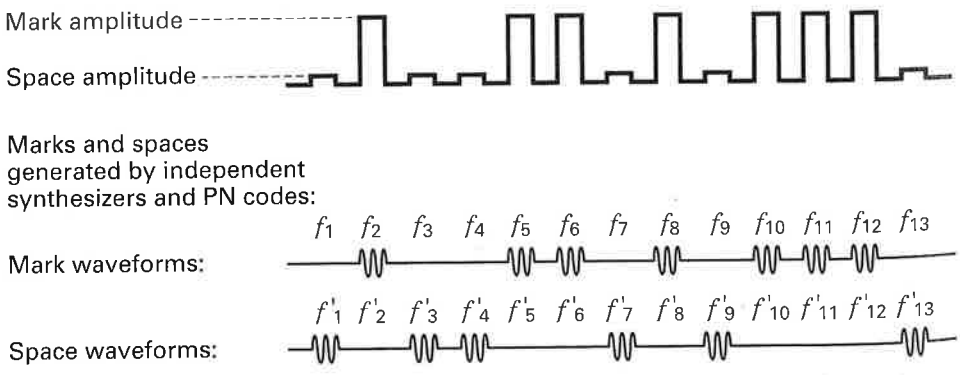


Figure 12.32 BLADES system.

present at a given frequency. The jammer sending narrowband energy in the same spectral neighborhood as the signal, is not destroying any modulation structure. For a noncoherent system, the jammer is only enhancing the communicator's signal. The only recourse for the repeat-back jammer is to change strategy by becoming a broadband jammer, and to jam the entire spread-spectrum bandwidth.

Notice that it is not really necessary to have a *pair* of frequencies for each bit. A *single* frequency will do. The communicator then transmits the pseudo-random frequency for a binary one and sends nothing for a binary zero. The receiver has the same code generator and therefore monitors the same pseudo-random frequencies. A binary one is detected by virtue of energy at the monitored frequency, and a binary zero is known by a lack of energy at the monitored frequency. Of course, the system is not as robust as when the marks and spaces are each transmitted on independently selected frequencies.

12.7 COMMERCIAL APPLICATIONS

12.7.1 Code-Division Multiple Access

Spread-spectrum multiple access techniques allow multiple signals occupying the same RF bandwidth to be transmitted simultaneously without interfering with one another. The application of spread-spectrum techniques to the problem of multiple access was discussed in Chapter 11 for a frequency hopped code-division multiple access (FH/CDMA) scheme. Here we consider CDMA using direct sequence (DS/CDMA). In these schemes, each of N user groups is given its own code, $g_i(t)$, where $i = 1, 2, \ldots, N$. The user codes are approximately orthogonal, so that the cross-correlation of two different codes is near zero. The main advantage of a CDMA system is that all the participants can share the full spectrum of the resource asynchronously; that is, the transition times of the different users' symbols do not have to coincide.

A typical DS/CDMA block diagram is shown in Figure 12.33. The first block illustrates the data modulation of a carrier, $A \cos \omega_0 t$. The output of the data modulator belonging to a user from group 1 is

$$s_1(t) = A_1(t) \cos [\omega_0 t + \phi_1(t)] \tag{12.56}$$

The waveform is very general in form; no restriction has been placed on the type of modulation that can be used.

Next, the data-modulated signal is multiplied by the spreading signal $g_1(t)$ belonging to user group 1, and the resulting signal $g_1(t)s_1(t)$ is transmitted over the channel. Simultaneously, users from group 2 through N multiply their signals by their own code functions. Frequently, each code function is kept secret, and its use is restricted to the community of authorized users. The signal present at the receiver is the linear combination of the emanations from each of the users. Neglecting signal delays, we show this linear combination as

$$g_1(t)s_1(t) + g_2(t)s_2(t) + \cdots + g_N(t)s_N(t) \tag{12.57}$$

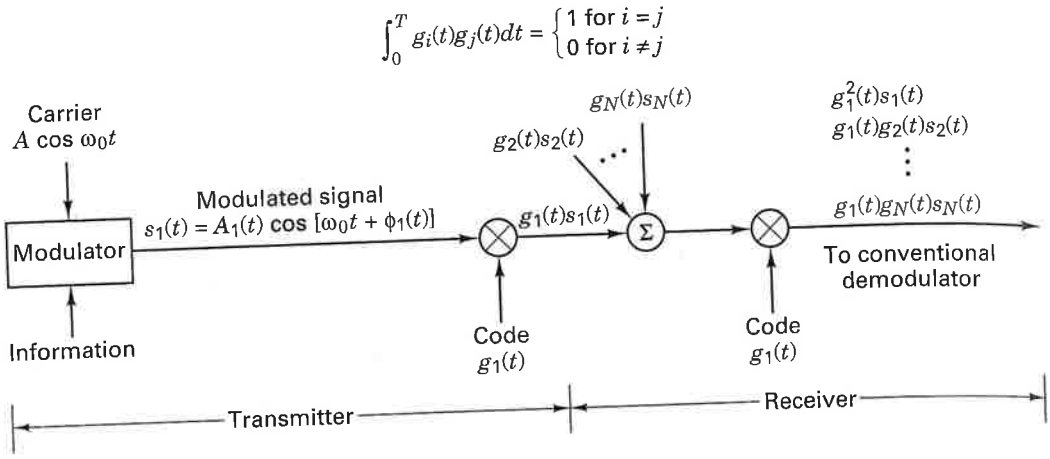


Figure 12.33 Code-division multiple access.

As mentioned earlier, multiplication of $s_1(t)$ by $g_1(t)$ produces a signal whose spectrum is the convolution of the spectrum of $s_1(t)$ with the spectrum of $g_1(t)$. Thus, assuming that the signal $s_1(t)$ is relatively narrowband compared with the code or spreading signal $g_1(t)$, the product signal $g_1(t)s_1(t)$ will have approximately the bandwidth of $g_1(t)$. Assume that the receiver is configured to receive messages from user group 1. Assume, too, that the $g_1(t)$ code, generated at the receiver, is perfectly synchronized with the received signal from a group 1 user. The first stage of the receiver multiplies the incoming signal of Equation (12.57) by $g_1(t)$. The output of the multiplier will yield the desired signal,

$$g_1^2(t)s_1(t)$$

plus a composite of undesired signals,

$$g_1(t)g_2(t)s_2(t) + g_1(t)g_3(t)s_3(t) + \cdots + g_1(t)g_N(t)s_N(t) \tag{12.58}$$

If the code functions $\{g_i(t)\}$ are chosen with orthogonal properties, similar to Equation (12.14), the desired signal can be extracted perfectly in the absence of noise since $\int_0^T g_i^2(t) = 1$, and the undesired signals are easily rejected, since $\int_0^T g_i(t)g_j(t)\,dt = 0$ for $i \neq j$. In practice, the codes are not perfectly orthogonal; hence, the cross-correlation between user codes introduces performance degradation, which limits the maximum number of simultaneous users.

Consider the frequency-domain view of the DS/CDMA receiver. Figure 12.34a illustrates the wideband input to the receiver; it consists of wanted and unwanted signals, each spread by its own code with code rate R_{ch}, and each having a power spectral density of the form $\text{sinc}^2(f/R_{ch})$. Receiver thermal noise is also shown as having a flat spectrum across the band. The combined waveform of Equa-

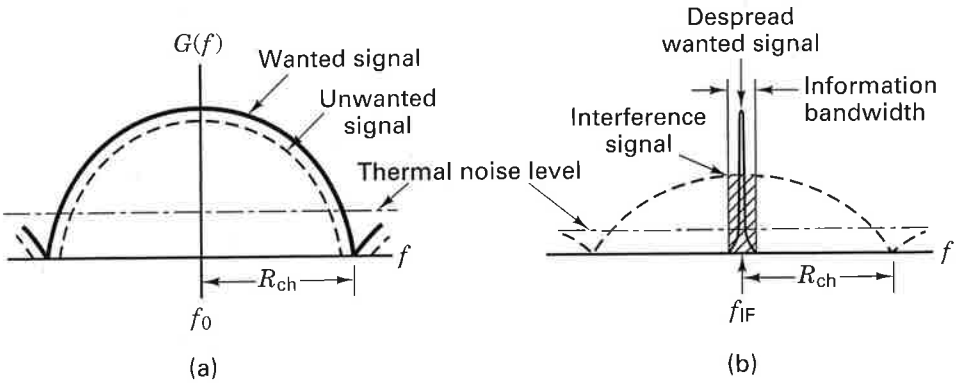


Figure 12.34 Spread-spectrum signal detection. (a) Spectrum at the input to receiver. (b) Spectrum after correlation with the correct and synchronized PN code.

tion (12.57) (desired plus undesired signals) is applied to the input of the receiver correlator driven by a synchronous replica of $g_1(t)$. Figure 12.34b illustrates the spectrum after correlation with the code $g_1(t)$ (despreading). The desired signal, occupying the information bandwidth centered at an intermediate frequency (IF), is then applied to a conventional demodulator, with bandwidth just wide enough to accommodate the despread signal. The undesired signals of Equation (12.58) remain effectively spread by $g_1(t)g_i(t)$. Only that portion of the spectrum of the unwanted signals falling in the information bandwidth of the receiver will cause interference with the desired signal.

Pursley [17] presents an excellent treatment on the performance of SSMA using DS, taking correlation properties of the code sequences into account. Also, Geraniotis [18] and Geraniotis and Pursley [19, 20] evaluate the performance of FH and DS multiple access systems subject to interference.

12.7.2 Multipath Channels

Consider a DS binary PSK communication system operating over a multipath channel that has more than one path from the transmitter to the receiver. Such multiple paths may be due to atmospheric reflection or refraction, or reflections from buildings or other objects, and may result in fluctuations in the received signal level. The different paths may consist of several discrete paths each with a different attenuation and time delay, or they might consist of a continuum of paths. Figure 12.35 illustrates a communication link with two discrete paths. The multipath wave is delayed by some time τ, compared with the direct wave. In television receivers, signals such as these cause "ghosts," or under extreme conditions, complete loss of picture synchronization.

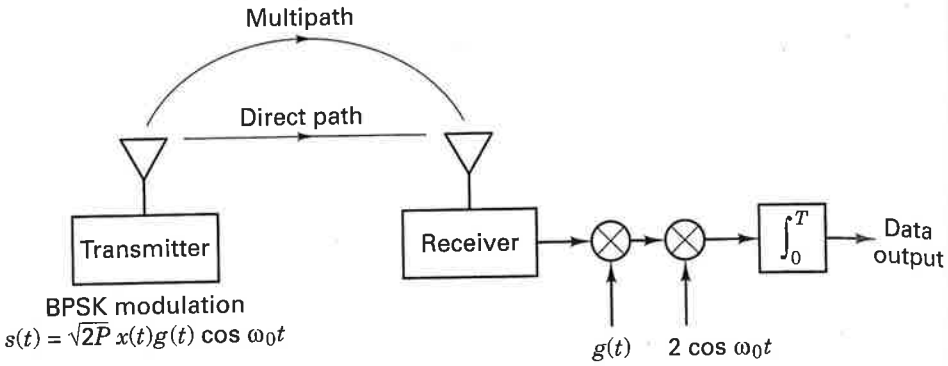


Figure 12.35 Direct-sequence BPSK system operating over a multipath channel.

In a direct-sequence spread-spectrum system, if we assume that the receiver is synchronized to the time delay and RF phase of the direct path, the received signal can be expressed as

$$r(t) = Ax(t)\,g(t)\,\cos \omega_0 t + \alpha Ax(t - \tau)g(t - \tau)\cos(\omega_0 t + \theta) + n(t) \quad (12.59)$$

where $x(t)$ is the data signal, $g(t)$ is the code signal, $n(t)$ is a zero-mean Gaussian noise process, and τ is the differential time delay between the two paths, assumed to be in the interval $0 < \tau < T$. The angle θ is a random phase, assumed to be uniformly distributed in the range $(0, 2\pi)$, and α is the attenuation of the multipath signal relative to the direct path signal. For the receiver, synchronized to the direct path signal, the output of the correlator can be written as

$$z(t = T) = \int_0^T [Ax(t)g^2(t)\cos \omega_0 t + \alpha Ax(t - \tau)g(t)g(t - \tau)\cos(\omega_0 t + \theta) + n(t)g(t)]2\cos \omega_0 t\, dt \quad (12.60)$$

where $g^2(t) = 1$. Also, for $\tau > T_c$, $g(t)g(t - \tau) \approx 0$ (for codes with long periods), where T_c is the chip duration. Therefore, if T_c is less than the differential time delay between the multipath and direct path signals, we can write

$$z(t = T) = \int_0^T 2Ax(t)\cos^2 \omega_0 t + 2n(t)g(t)\cos \omega_0 t\, dt = Ax(T) + n_0(T) \quad (12.61)$$

where $n_0(T)$ is a zero-mean Gaussian random variable. We see that the spread-spectrum system, similar to the case of CDMA, effectively eliminates the multipath interference by virtue of its code-correlation receiver.

If frequency hopping (FH) is used against the multipath problem, improvement in system performance is also possible but through a different mechanism. FH receivers avoid multipath losses by rapid changes in the transmitter frequency

band, thus avoiding the interference by changing the receiver band position before the arrival of the multipath signal.

12.7.3 The FCC Part 15 Rules for Spread-Spectrum Systems

In the United States, the Federal Communications Commission (FCC) allows the general unlicensed operation of very lower power (less than 1 mW) radio equipment freely, except in certain restricted frequency bands. In 1985, Dr. Michael Marcus of the FCC was responsible for allowing higher power (up to 1 W) spread-spectrum radios in some of the bands, referred to as *Industrial, Scientific,* and *Medical* (ISM). The rules of allowable electromagnetic radiation for unlicensed devices appear in the *Code of Federal Regulations* (CFR) Title 47, Part 15; they are known simply as the Part-15 rules (wherein Section 15.247 covers spread spectrum).

The ISM frequency bands are used for instruments (e.g., medical diathermy equipment) as well as for critical government systems (e.g., radio location equipment) that radiate strong electromagnetic fields which can cause interference to other users. The ISM bands are particularly noisy bands. An unlicensed radio can be thought of as an "unwelcome guest" in a licensee's band. An unlicensed radio must be able to suffer interference but is not permitted to cause any interference to a licensed user.

For frequency-hopping systems, the Part-15 rules require that the average time of occupancy on any frequency shall not be greater than 0.4 second (or a minimum hopping rate of 2.5 hops/s). For direct sequence systems, the minimum required processing gain is 10 dB. For hybrid systems employing both direct sequence and frequency hopping, the minimum required processing gain is 17 dB. Three ISM spectral regions were designated for the operation of unlicensed spread-spectrum radios. Some of the details regarding the operation in these bands are shown in Table 12.1.

As a result of allowing higher power limits and no FCC licensing as described previously, commercial companies have introduced a wide range of innovative spread-spectrum radios capable of communications over greater distances than earlier low-power narrowband unlicensed radios. Some of these products include radios that link office equipment (e.g., shared printer, wireless local area net-

TABLE 12.1 Spread-Spectrum Operation Under Part-15 Rules

ISM Band	Total Bandwidth	Max. Bandwidth per Channel for FH*	Min. Number of Hopping Frequencies per Channel	Min. Bandwidth per Channel for DS*
902–928 MHz	26 MHz	500 kHz	25–50**	500 kHz
2.4000–2.4835 Ghz	83.5 MHz	1 MHz	75	500 kHz
5.7250–5.8500 GHz	125 MHz	1 MHz	75	500 kHz

*Maximum bandwidth per channel for frequency hopping is defined as the 20 dB bandwidth; minimum bandwidth per channel for direct sequence is defined as the 6 dB bandwidth.

**FH channels with bandwidth less than 250 kHz require at least 50 hopping frequencies per channel; FH channels with bandwidth greater than 250 kHz require at least 25 hopping frequencies per channel.

works), cordless telephones, wireless point-of-sales equipment (e.g., cash registers, bar-code readers).

12.7.4 Direct Sequence versus Frequency Hopping

Without interference from other radios and in free space, both direct-sequence (DS) and frequency-hopping (FH) spread-spectrum radios can, in theory, give the same performance. For mobile applications with large multipath delays, DS represents a reliable mitigation method, because such signaling renders all multipath signal copies that are delayed by more than one chip time from the direct signal as "invisible" to the receiver. (See Section 12.7.2.) FH systems can provide the same mitigation, only if the hopping rate is faster than the symbol rate, and if the hopping bandwidth is large. (See Chapter 15.)

Implementing a fast frequency-hopping (FFH) radio can be costly due to the need for high-speed frequency synthesizers. Consequently, hopping rates of commercial FH radios are generally slow compared with the data rate, and hence such systems behave like narrowband radios. Slow frequency hopping (SFH) and DS signaling each experience somewhat different interference. SFH radios typically suffer occasional strong bursty errors, while DS radios encounter more randomly distributed errors that are continuous and lower level. For high data rates, the impact of multipath tends to degrade such SFH radios more than DS radios. To mitigate the effects of bursty errors in FH radios, interleaving would have to be performed over long time durations. (See Chapter 15.) SFH is used for providing diversity in fixed wireless access applications or slowly moving systems, or merely to meet the Part-15 Rules. For commercial applications, implementation of DS radios with large processing gain can also be costly due to the need for high-speed circuits; thus, the processing gain for such radios is usually limited to less than 20 dB to avoid having to use high-speed circuits [29].

Example 12.5 Detection of Signals Buried in the Noise

In Section 12.1.1.1 it was shown that spread-spectrum techniques offer no error-performance advantage against thermal noise. In this example, we show that whatever value of received E_b/N_0 might be available to a narrowband system remains the same after spreading. Although there is no error-performance advantage against thermal noise, there is *no disadvantage* either, which makes it an attractive option for building systems to meet the FCC Part-15 Rules, as described in Section 12.7.3, and for multiple access systems (e.g., CDMA systems meeting Interim Standard IS-95).

Direct-sequence spread-spectrum techniques allow for the detection of signals that have power-spectral density (psd) levels well below the noise. Consider Figure 12.36a illustrating the received psd of a communication signal with an ideal rectangular shape, having an intensity of $S_0(f) = 10^{-5}$ W/Hz over a bandwidth of 1 MHz. Assume that the transmitted data rate is $R = 10^6$ bits/s. Also, assume AWGN noise and a received psd level of $N_0(f) = 10^{-6}$ W/Hz (not drawn to scale) at all frequencies. Find the received E_b/N_0 for this narrowband case. Next, consider that the communication signal is spread over a spread-spectrum bandwidth of $W_{ss} = 10^8$ Hz, as shown in Figure 12.36b, keeping the total average signal power the same as in the narrowband case

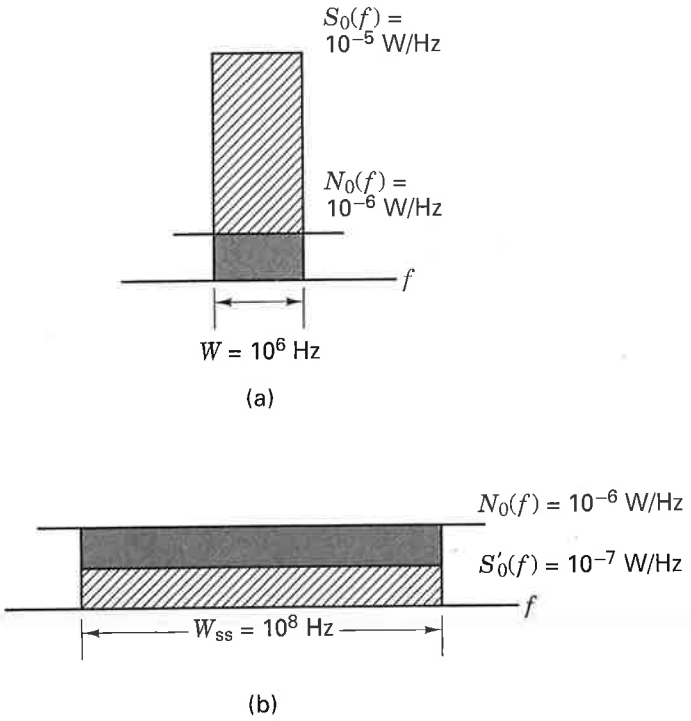


Figure 12.36 Signal and noise power spectral densities (a) Before spreading (b) After spreading.

(not drawn to scale). Show that with a spread-spectrum receiver, the received E_b/N_0 is the same as in the narrowband case, and hence the error performance is unchanged.

Solution

Before spreading, the total average power of the communication signal is $S = 10^{-5}$ W/Hz $\times$ 10^6 Hz = 10 W, and the total average noise power is $N = 10^{-6}$ W/Hz $\times$ 10^6 Hz = 1 Watt. The received E_b/N_0 is

$$\frac{E_b}{N_0} = \frac{S/R}{N_0} = \frac{10 \text{ W}/10^6 \text{ bits/s}}{10^{-6} \text{ W/Hz}} = 10 \text{ or } 10 \text{ dB}$$

After spreading, the psd, $S_0'(f)$, of the communication signal is reduced by the same factor (two orders of magnitude) as the increase in bandwidth; hence, the total average power of the communication signal is still 10 W. However, since the noise was assumed to be AWGN, its psd is not reduced; hence, the total average noise power is now equal to $N' = 10^{-6}$ W/Hz $\times$ 10^8 Hz = 100 W. Therefore, after spreading, the received E_b/N_0 can be expressed as

$$\frac{E_b}{N_0} = \frac{S/R}{N'/W_{ss}} = \frac{S}{N'}\left(\frac{W_{ss}}{R}\right) = \frac{S}{N'}G_p = \frac{10 \text{ W}}{100 \text{ W}} \times 100 = 10 \text{ or } 10 \text{ dB}$$

where $G_p = W_{ss}/R = 100$ is the processing gain. The process of detecting direct-sequence spread-spectrum signals "buried in noise" does not lend itself to an intuitive

illustration, as can be verified in Figure 12.36b. Similarly, in the expression for received E_b/N_0 after spreading, the communication signal power is only 10 W and the noise power is 100 W, leading to the same lack of intuition regarding the detectability of the signal. It is the processing gain (which is not very visual) that allows us to receive the same value of E_b/N_0 as in the narrowband case.

12.8 CELLULAR SYSTEMS

Wireless personal communication systems, particularly cellular systems are relatively young applications of the communications technology. The following list includes some of the events that illustrate the evolution of this ever-growing business:

- 1921 Radio dispatch service initiated for police cars in Detroit, Michigan.
- 1934 Amplitude modulation (AM) mobile communication systems used by hundreds of state and municipal police forces in the U.S.
- 1946 Radiotelephone connections made to the public-switched telephone network (PSTN).
- 1968 Development of the cellular telephony concept at Bell Laboratories.
- 1981 Ericsson Corporation's Nordic Mobile Telephone (NMT) in Scandinavian countries becomes the first cellular system fielded.
- 1983 Cellular service in the United States—called the Advanced Mobile Phone System (AMPS) and using frequency modulation (FM)—placed in service in Chicago by Ameritech Corporation.
- 1990s Second generation digital cellular deployed throughout the world. The Global system for Mobile (GSM) Communications becomes the pan-European standard. (Prior to GSM, many different cellular systems operating in Europe became operationally impractical.)
- 1990s Second generation digital systems known as IS-54 and its successor IS-136 (TDMA), and IS-95 (CDMA) become operational in the United States.
- 2000s Third-generation digital systems standardized at the network level to allow world-wide roaming start becoming operational. They offer enhanced services, such as connection to various PSTN systems with a single phone, and connecting to high data rate packet systems such as Internet Protocol (IP) networks.

12.8.1 Direct Sequence CDMA

Figures 11.3 and 11.7 depict sharing a communications resource via FDMA or TDMA. In the case of FDMA, different frequency bands are orthogonal to one another (assuming ideal filtering), and in the case of TDMA, different time slots are orthogonal to one another (assuming perfect timing). One can visualize a similar orthogonality among different channels in the case of frequency-hopping CDMA (as shown in Figure 11.14) if the codes that control the frequency hopping operate in

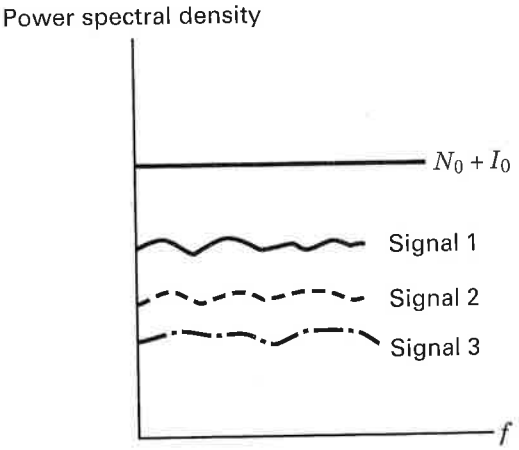


Figure 12.37 Three DS/SS signals occupying the same spectral region.

such a way that no two users are collocated in time and frequency. It is easy to visualize the user transmissions hopping in frequency and time without any contention. However, in the case of direct-sequence spread-spectrum (DS/SS), visualization of the necessary orthogonality conditions (with multiple users simultaneously occupying the same spectrum) is not as easy. Figure 12.37 shows three different DS/SS signals that are spread over a broad range of frequencies below the level of noise-and-interference power spectral density, $N_0 + I_0$ (assumed to be wideband and Gaussian). An often asked question regarding Figure 12.37 is "How can one of these signals be detected when it is spectrally "buried" below the noise and interference, and it is collocated with other similar signals?" The answer is that the DS/SS receiver correlates the received signal to a particular user's PN code. If the PN codes are orthogonal to each other, then the other users' signals will average to zero during a long observation time. If the codes are not purely orthogonal, they will contribute some interference to the detection process.

In a mobile telephone system using CDMA, each of the users *do* interfere with one another for the following reasons: (1) Two different spreading codes from a family of perfectly orthogonal *long codes* may not yield zero correlation over a short interval of time, such as a symbol time. (2) Serving a large population of users, typically dictates the use of long codes. Such codes can be designed to have low cross-correlation properties but are not orthogonal. (3) Multipath propagation and imperfect synchronization cause interchip interference among the users.

Consider a reverse channel (mobile to base station) in a heavily loaded cell, where the interference caused by the many simultaneous CDMA signals typically outweighs the degradation caused by thermal noise. The assumption is generally made that the thermal noise can be neglected compared with the interference from other users. Thus, for $N_0 << I_0$, the following relationship for the received E_b/I_0, designated as $(E_b/I_0)_r$, can be obtained:

$$\left(\frac{E_b}{N_0 + I_0}\right)_r \approx \left(\frac{E_b}{I_0}\right)_r = \frac{S/R}{I/W_{ss}} = \frac{W_{ss}/R}{I/S} = \frac{G_p S}{I} \tag{12.62}$$

where $G_p = W_{ss}/R$ is the processing gain, W_{ss} is the spread-spectrum bandwidth, S is one user's received power, and I is the interference power from all other users. Equation (12.62) shows that even when the received interference greatly exceeds a user's received power, it is the processing gain (via the mechanism of correlating to a code) that can yield an acceptable value of E_b/I_0. When the base station exercises power control so that each user's received power is balanced, then $I = S \times (M - 1)$, where M is the total number of users contributing to interference at the receiver. It is now possible to express $(E_b/I_0)_r$ in terms of the processing gain and the number of active users in the cell, as follows:

$$\left(\frac{E_b}{I_0}\right)_r \approx \frac{G_p S}{I} = \frac{G_p S}{S \times (M-1)} = \frac{G_p}{M-1} \tag{12.63}$$

Note that the received E_b/I_0 in Equation (12.63) is analogous to the E_b/J_0 for a jammed receiver in Equation (12.41), with J_0 and J replaced by I_0 and I, respectively. CDMA systems are affected by such interference (assumed wideband and Gaussian) in the same way, whether caused by jammers, accidental interferers, or authorized participants. In Equation (12.63), knowing G_p and the *required* E_b/I_0, designated $(E_b/I_0)_{\text{reqd}}$, for a given error performance, the *maximum* number of allowable users (interferers) per cell is

$$M_{\max} \approx \frac{G_p}{(E_b/I_0)_{\text{reqd}}} \tag{12.64}$$

Note that Equation (12.63) indicates that for a heavily loaded cell, a CDMA system is interference limited. For example, if the number of active users occupying a cell were to suddenly double, then the *received* E_b/I_0 would essentially be halved. Also, by examining Equation (12.64), it can be seen that any reduction in $(E_b/I_0)_{\text{reqd}}$ has the effect of increasing the maximum allowable number of users in the cell. The following is a list of other factors that influence the final calculation for the maximum number of allowable users per cell:

- **Sectorizing or Antenna Gain (G_A).** Dividing the cell into three 120° sectors by using a separate directional antenna for each sector, provides a gain G_A of about 2.5 (or 4 dB) in the number of users that can be accommodated.
- **Voice Activity Factor (G_V).** The average speaker pauses about 60% of the time between words and sentences and for listening. Thus, for a CDMA voice circuit, transmission need take place only 40% of the time, whenever there is speech activity. For voice channels, this contributes an improvement factor G_V of about 2.5 (or 4 dB) in the number of users that can be accommodated.
- **Outer-Cell Interference Factor (H_0).** As described in Section 12.8.2, 100% frequency reuse can be employed for CDMA; all neighboring cells can use the same spectrum. Therefore, for a given level of interference I_x originating within a cell, there is additional interference originating outside of the cell. For signal-propagation loss that follows an $n = 4$th power exponent law (see Section 15.2.1), this additional interference is estimated at about 55% of the within-cell interference [30, 31]. The total interference is therefore approxi-

mately 1.55 I_x, resulting in a user capacity degradation factor H_0 of about 1.55 (or 1.9 dB).

- **Nonsynchronous Interference Factor (γ).** For estimating interference from other (within-cell and outer-cell) users, we assume an identical set of channels (e.g., all voice users requiring the same performance). We further assume that their despread interference can be approximated as a Gaussian random variable, that the users are spatially distributed in a uniform manner, and that power control within each cell is perfect. The worst-case interference comes about if all the interferers are chip and phase synchronized with the desired signal. For a nonsynchronous link, interference will not always be worst case. This lesser interference can be described by a factor γ that modifies Equation (12.64), thereby yielding more users per cell than that of the worst case. Assuming ideal rectangular-shaped chips, γ is equal to 1.5 [31–34]; this value will change for different chip shapes [31].

Using the factors G_A, G_V, H_0, and γ (and their typical values shown above) to determine the maximum possible number of simultaneous users per cell, M', yields

$$M' = \frac{\gamma\, G_A\, G_V}{H_0} \times M_{\max} = \frac{\gamma\, G_p\, G_A G_V}{(E_b/I_0)_{\text{reqd}}\, H_0} \approx 6 \times M_{\max} \tag{12.65}$$

An accurate computation of capacity for a CDMA system is much more involved than Equation (12.65) suggests. The treatment leading to Equation (12.65) assumed perfect power control and a uniform distribution of users' location within cells. Thermal noise was neglected and no provision was made for traffic loading within cells. Terrain variations, which impact the accuracy of assuming an $n = 4$th power exponent law, were not considered. For lower values of n, there is potentially greater interference. The subject of CDMA capacity has been investigated in many publications, particularly in the context of systems designed to meet IS-95. The issue of capacity of CDMA systems is further addressed in [30–32, 35–38]. A very simplified analysis of three multiple access techniques that allow us to illustrate the capacity advantage of CDMA follows.

12.8.2 Analog FM versus TDMA versus CDMA

In 1976, prior to the implementation of cellular communication systems, New York City (with a population exceeding 10 million) could only support 543 simultaneous mobile users—3700 customers were on a waiting list. The cellular concept is illustrated in Figure 12.38 with a 7-cell configuration (one of several used). The idea of dividing a geographical region into cells and allowing the frequency allocation of one cell to be reused at other spatially separated cells represents one of the most important bandwidth-efficiency improvements in radio telephone systems.

In the United States, the frequency allocation for AMPS and other cellular systems is in the range of 869–894 MHz for base station transmit (mobile receive) channels, called *forward* or *downlink* channels, and 824–849 MHz for mobile transmit (base station receive) channels, called *reverse* or *uplink* channels. A single

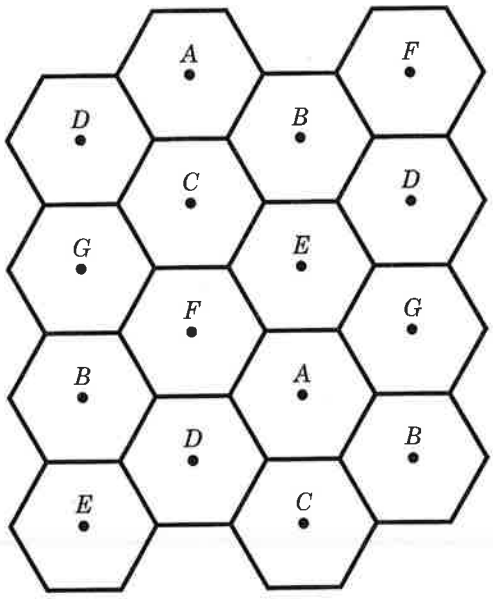


Figure 12.38 Seven-cell structure.

channel occupies a bandwidth of 30 kHz, sometimes called a *subband;* thus, a duplex pair (forward and reverse) occupy 60 kHz. The forward and reverse channels in each duplex pair are separated by 45 MHz. For mobile cellular service, the FCC has allocated each large metropolitan area (there about 750 such areas in the U.S.) 25 MHz for transmit and 25 MHz for receive. To foster competition, there are generally two service-provider companies allocated to each metropolitan area. Thus, each company has a total of 12.5 MHz for transmit and 12.5 MHz for receive.

When considering a wide geographical region made up of many cells, as seen in Figure 12.38, let's compare the capacity in units of channels per cell for three cellular systems: analog FM, TDMA, and CDMA. Computing capacity for the analog FM channels used in the AMPS system is quite straightforward. Consider the 12.5 MHz allocated to a service provider. In order to avoid interference between users operating in the same 12.5-MHz frequency band at comparable power levels, adjacent cells must operate at different frequencies. In the 7-cell configuration of Figure 12.38, communications within cell F may not operate in the same frequency band as communications in cells labeled A, B, C, D, E, and G. Although the service provider has been allocated 12.5 MHz, the frequency reuse pattern involved here dictates that only one-seventh of the allocation can be utilized within each cell. Thus, one-seventh of 12.5 MHz or equivalently 1.78 MHz can be used for transmit (and a similar amount for receive) within each cell. We say that such a 7-cell configuration has a *frequency-reuse factor* of $\frac{1}{7}$. Therefore, the number of 30-kHz subbands for analog FM channels is 1.78 MHz/30 kHz or approximately 57 channels per cell (not counting the control channels).

The U.S. standard describing the multiple access strategy for cellular TDMA is designated IS-54, which has been upgraded to IS-136. Systems designed to these standards must fit into the same frequency plan that was outlined for AMPS.

Therefore, each TDMA channel occupies 30 kHz. Fortunately, capacity improvements have come about only because the discipline of source coding has improved so dramatically since the 1950s. For *terrestrial* digital telephony, each voice signal is digitized to a bit rate of 64 kbits/s. Would a similar standard be used for *cellular* systems? Of course not, because cellular systems are so bandwidth limited. Source coding of speech can now produce telephone-quality fidelity at data rates of 8 kilobit/s, and it can even produce acceptable quality at lower data rates. For purposes of computation, if the often-chosen benchmark value of 10 kbit/s is used, then the capacity computation is again straightforward. Each of the 30 kHz channels can service 30 kHz/10 kbits/s = 3 users per 30 kHz subband. Thus, in TDMA, the number of simultaneous users per cell can be increased by a factor of 3 over the analog FM system. In other words, the number of TDMA channels is $57 \times 3 = 171$ channels per cell.

The main advantage of a CDMA cellular system over either analog FM or TDMA is that a frequency reuse factor of unity (100%) can be used. This means that the total FCC allocation of 12.5 MHz can be used for transmit (similarly for receive). In order to compare CDMA with the multiple access strategies in AMPS involving analog FM (which we can call FDMA) and IS-54-based TDMA, we start with Equation (12.65), but for a fair comparison, we eliminate the antenna gain factor G_A achieved through sectorizing the cell. The reason for this elimination is that G_A was not used in calculating the capacity for FDMA or TDMA, although both systems would also benefit from sectorization. Hence, the capacity of CDMA without sectorization becomes

$$M'' = \frac{\gamma\, G_p\, G_V}{(E_b/I_0)_{\text{reqd}}\, H_0} \tag{12.66}$$

Equation (12.28) then gives the processing gain

$$G_p = \frac{R_{\text{ch}}}{R} = \frac{12.5 \text{ Mchips/s}}{10 \text{ kbits/s}} = 1250 \tag{12.67}$$

Note that the chip rate of 12.5 Mchips/s is *not consistent* with IS-95 standards. It is used here to equitably compare CDMA across the entire allocation of the 12.5 MHz bandwidth, the same bandwidth used for analog FM and TDMA.

Selecting a nominal value of $(E_b/I_0)_{\text{reqd}}$ to be 7 dB (or the factor 5) [30], and for the factors G_V, γ, and H_0, using the values 2.5, 1.5 and 1.55, respectively, as described in Section 12.8.1, we then use Equation (12.66) to obtain

$$M'' = \frac{1.5 \times 1250 \times 2.5}{5 \times 1.55} \approx 605 \tag{12.68}$$

In summary, FDMA, using analog FM, TDMA, and CDMA, support 57, 171, and 605 channels per cell, respectively. Hence, it can be said that, in a given bandwidth, CDMA can exhibit about 10 times more user capacity than AMPS, and about 3.5 times the capacity of TDMA. It should be noted that the simple analysis leading to Equation (12.68) does not take into account other considerations, such as flat fading (see Chapter 15), which is sometimes encountered and may degrade the results

of Equation (12.68). It should also be emphasized that the analysis was based on a CDMA reverse link, where unsynchronized users with long codes were assumed. In the forward direction (base station to mobile) orthogonal channelization can be used, which would improve the results of Equation (12.68).

It is difficult to compare CDMA with TDMA/FDMA in a fair way. On a single-cell basis, TDMA/FDMA capacity is dimension limited, while CDMA capacity is interference limited (discussed in the following section). From a multi-cell system view, all the systems are eventually interference limited. They attempt to optimize capacity with the following trade-offs. TDMA/FDMA systems trade-off larger reuse factors at the expense of greater interference. CDMA systems trade-off increased loading at the expense of greater interference.

12.8.3 Interference-Limited versus Dimension-Limited Systems

The interference in a properly designed and operating CDMA system is not severe; hence, all users can occupy the same spectrum. Nevertheless, from Equations (12.63) and (12.64), a CDMA system must be classified as interference-limited. Any reduction in $(E_b/I_0)_{\text{reqd}}$ can be translated almost directly into a larger number of simultaneous users. One can therefore see how important the incorporation of error-correction coding is to CDMA systems. An increase in coding gain by only 1 dB, which of course would reduce the $(E_b/I_0)_{\text{reqd}}$ by 1 dB, would yield a 25% increase in the number of allowable active users per cell.

In the context of single-cell operation, an FDMA and a TDMA system can be termed *frequency-dimension* and *time-dimension* limited, respectively. Consider a TDMA system. As time slots are assigned to an increasing number of users, there is no interference at the base station receiver caused by other mobile radios to the reception of a given user (assuming perfect synchronization). The user population can be increased until the number of time slots are exhausted. It is not possible to increase the number of users beyond the time-slot limit without intolerable interference. In a similar way, FDMA is frequency-dimension limited. It is not possible to increase the number of users beyond the frequency band limit without intolerable interference.

CDMA is interference limited because the introduction of each additional user raises the overall level of interference at the base station receivers. Each mobile radio introduces interference as a function of power level, synchronization, and code cross-correlation with other CDMA signals. The number of CDMA channels allowed depends on the level of total interference that can be tolerated. Figure 12.39 illustrates the basic difference between interference-limited systems, such as CDMA, and dimension-limited systems, such as TDMA. Assume that a fixed-size bandwidth is available for both. With TDMA in the context of a single cell, as time slots are filled by an increasing number of TDMA users, there is no interference at the base station receiver (caused by other mobile radios) to the reception of a given user. The number of TDMA users can increase until the number of available time slots is exhausted. It is then not possible to assign another time slot without causing an intolerable amount of interference. With CDMA, the introduction of each addi-

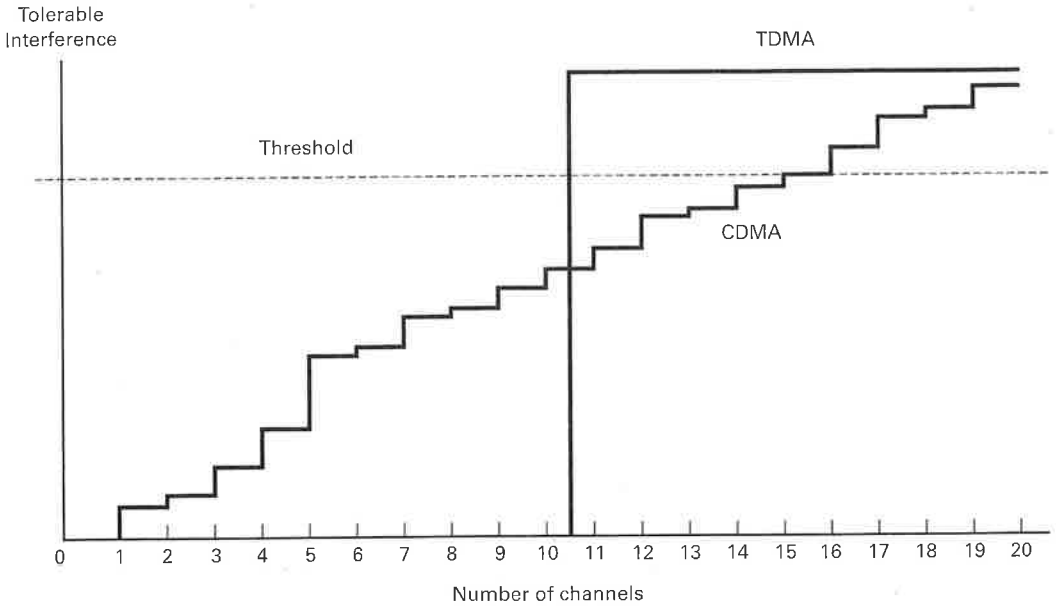


Figure 12.39 TDMA is time-dimension limited. CDMA is interference limited.

tional user raises the overall level of interference to the base station receivers. Each mobile radio might introduce a unique level of interference, owing to differences in power level, timing-synchronization, and cross-correlation with other code signals. Within a cell, channels are assigned to users until some predetermined interference threshold is reached [29]. Figure 12.39 shows that an interference-limited system is inherently more adaptive than a dimension-limited system. For example, on certain days of the year when it is well known that telephone traffic increases (such as Christmas Day and Mother's Day), a CDMA Operations Center can choose to tolerate a bit more interference in order to allow a larger number of users. With dimension-limited systems, no such dynamic trade-off can be made.

It is worth repeating that dimension-limited systems, such as FDMA and TDMA, are strictly dimension limited in the context of a single-cell operation. However, from a multi-cell perspective, one can trade-off frequency reuse factors versus the signal-to-interference (S/I) ratio to arrive at an interference-limited situation.

12.8.4 IS-95 CDMA Digital Cellular System

Interim Standard 95 (IS-95) specifies a wireless telephony system that uses direct-sequence spread-spectrum (DS/SS) as a multiple access technique. It was introduced by Qualcomm Corporation, and it was designed to operate in the same frequency band as the U.S. analog cellular system (AMPS), in which full duplex operation is achieved by using frequency division duplexing (FDD). The frequency allocation for AMPS provides 25 MHz in the range of 869–894 MHz for base sta-

tion to mobile transmission (forward channels), and 25 MHz in the range of 824–849 MHz for mobile-to-base-station transmission (reverse channels). The IS-95 implementation strategy has been to introduce this code-division multiple-access (CDMA) system 1.25 MHz at a time, using dual-mode (AMPS and CDMA) mobile units. Being interference limited, systems designed to meet IS-95 specifications utilize various signal processing techniques to help reduce the $(E_b/N_0)_{\text{reqd}}$. The basic waveform, coding, and interference suppression features of such systems are outlined as follows:

- Each channel is spread across a bandwidth of about 1.25 MHz and filtered for spectral containment.
- The chip rate R_{ch} of the PN code is 1.2288 Mchips/s. The nominal data rate, known as Rate Set 1 (RS1), is 9.6 kbits/s, making the processing gain $G_p = R_{\text{ch}}/R = 128$. An extension to the original IS-95 introduced Rate Set 2 (RS2) at 14.4 kbits/s.
- The data modulation is binary phase-shift keying (BPSK), with quadrature phase-shift keying (QPSK) spreading. (Each quadrature component of the carrier wave is a BPSK signal modulated with the same data.)
- Convolutional coding with Viterbi decoding is used.
- Interleavers with a 20-ms time span are used for time diversity.
- Path diversity is exploited with a Rake receiver, and spatial diversity is implemented with two receive antennas per cell sector.
- Orthogonal code multiplexing is used for channelization.
- Power control is used to minimize transmitted power and thereby reduce interference.

The forward link comprises four types of channels: pilot, synchronization (SYNC), paging, and traffic. The reverse link comprises two types of channels: access and traffic. The history of IS-95 involves several standard committees and versions, with numbers such as IS-95A, JSTD-008, IS-95B, and IS-2000. IS-95B is a merging of IS-95-based methods for the cellular frequency band and the personal communication services (PCS) frequency band, for both voice and data. It provides data rates up to 115.2 kbits/s by aggregating up to eight RS2 channels. IS-2000 is a specification used to denote third-generation CDMA wireless systems, known as multi-carrier systems and having an assortment of new features. The treatment of CDMA in this section focuses on the original IS-95; the original structure remains valid for all IS-95-based variations, because they all share the basic architecture of the original system.

12.8.4.1 Forward Channel

The base station transmits a multiplex of 64 channels containing one pilot channel, one SYNC channel and at least one paging channel. The remaining 61 (or fewer) channels transmit user traffic. The IS-95 standard supports simultaneous transmission of voice, data, and signaling; variable rates for speech signals of 9600,

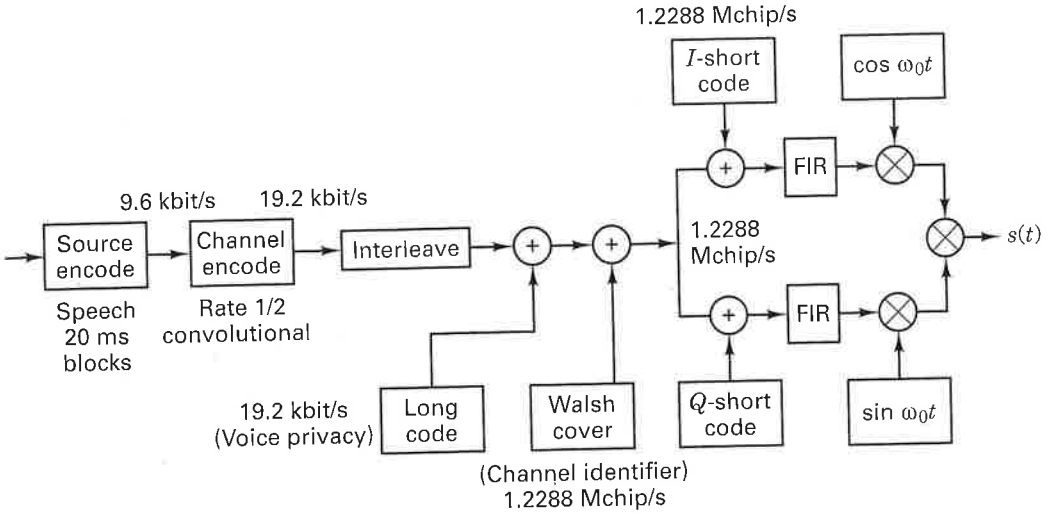


Figure 12.40 CDMA forward-traffic channel with full-rate speech.

4800, 2400, and 1200 bit/s are permitted. These rates are known as Rate Set 1. (Rate Set 2 supports up to 14.4 kbit/s.) Figure 12.40 is a simplified block diagram of the base station transmitter, implementing a typical 9.6 kilobits/s traffic channel. Using a linear predictive coding (LPC) algorithm (see Section 13.4.2), voice is first digitized to yield approximately 8 kilobits/s of raw digital speech. Error-detection bits are added, bringing the digital rate to 9.6 kilobits/s. The bit sequence is then processed in frame lengths of 20 ms. Hence, each 9.6 kilobits/s frame contains 192 bits. The next step shown in Figure 12.40 is convolutional coding (rate $\frac{1}{2}$, $K = 9$), where all information bits are equally protected. This brings the channel bit rate to 19.2 kilobits/s, which remains unchanged after interleaving by a block interleaver, with a span equal to one frame length of 20 ms. The next three steps involve the modulo-2 addition of binary digits representing different PN codes and orthogonal sequences for privacy, channelization, and base station identification. Each time a code is introduced, it can be thought of as a *barrier* or *door* that separates a specific message from others for a particular reason. Consider the privacy code. It is a *long* PN code implemented with a maximal-length, 42-stage shift register. at the system chip rate of 1.2288 Mchips/s, the code repeats approximately every 41 days. Systems designed to IS-95 specifications employ the same long-code hardware for all base stations and mobile units. To provide each mobile unit with its own unique code for privacy, each mobile is assigned a phase (time) offset of a privacy code. The parties carrying on a conversation do not need knowledge of each other's unique long-code offsets, since the base station demodulates and remodulates all traffic signals it processes. At the point that the privacy code is introduced in Figure 12.40, the channel-bit rate of 19.2 kilobits/s is not yet at the final chip rate. Hence, in the forward direction, the user's private code is applied in decimated fashion; that is, only every 64th bit of the sequence is used (which doesn't take away from the code's uniqueness).

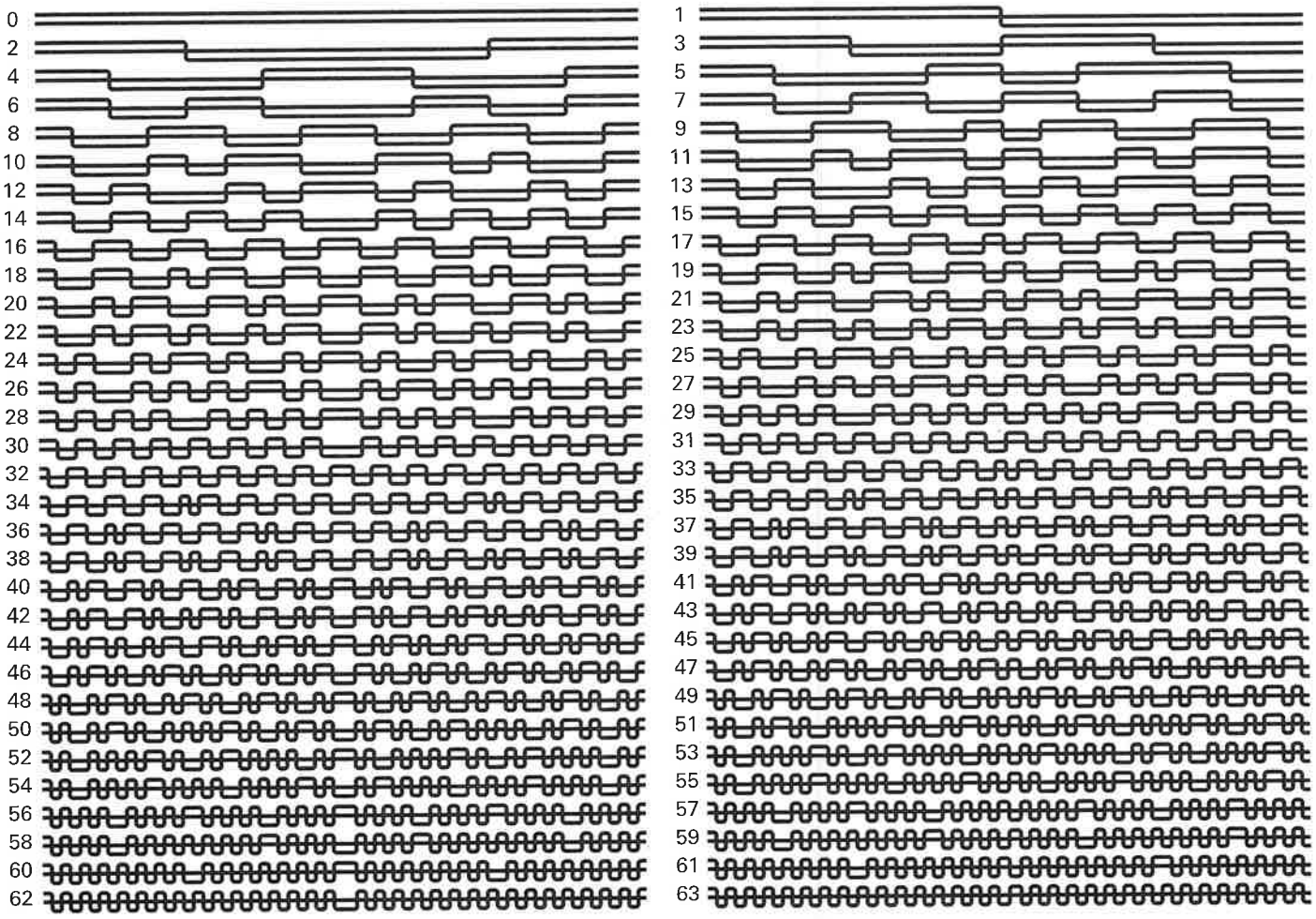


Figure 12.41 The set of 64 Walsh waveforms.

The next code, called a *Walsh cover,* is used for channelization plus spreading. It is an orthogonal code, which is mathematically constructed via the *Hadamard matrix.* (See Section 6.1.3.1 for the construction rules.) Using such a rule, one can form an orthogonal Walsh code of any desired dimension $2^k \times 2^k$, where k is a positive integer. The set of Walsh codes is described by a 64×64 array, where each row generates a different code. One of the 64 Walsh codes is modulo-2 added to the privacy-protected binary sequence, as shown in Figure 12.40. Because each of the 64 members of the Walsh code set are orthogonal to one another, their use in this manner channelizes the forward transmissions into 64 orthogonal signals. Channel number 0 is used as a pilot signal to assist coherent reception at the mobile unit, channel number 32 is used for synchronization, and at least one channel is reserved for paging. That leaves a maximum of 61 channels for traffic use. The Walsh cover is applied at the chip rate of 1.2288 Mchips/s. Thus, in the forward direction, each channel bit (at a rate of 19.2 kilobits/s) is transformed into 64 Walsh chips, producing a final chip rate of 1.2288 Mchips/s. Figure 12.41 illustrates the set of 64 Walsh waveforms. Figure 12.42 shows a simple channelization example using an orthogonal code such as a Walsh code. Unless the receiver applies the correct waveform for accessing a user's channel, the output is zero. Applying the correct waveform yields some nonzero value, A, that "unlocks the door" to that channel.

The next code in the forward direction (see Figure 12.40) is called the *short code* because it is configured with a 15-stage shift register, it repeats every $2^{15} - 1$ chips, and one period lasts 26.67 ms. This final "cloak" or "barrier," applied in quadrature at the chip rate of 1.2288 Mchips/s, provides scrambling of the signal. All base stations reuse the same Walsh channelization; without such scrambling,

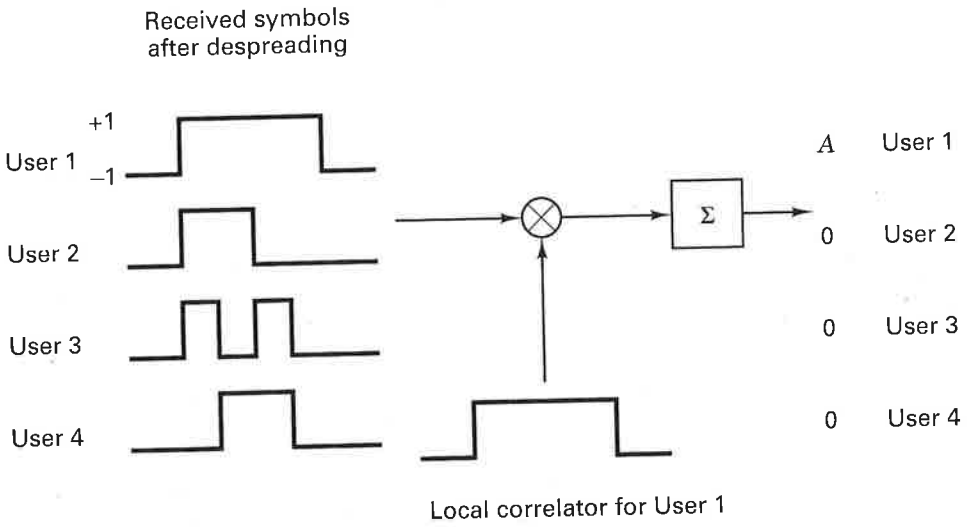


Figure 12.42 Example of channelizing transmissions with orthogonal functions.

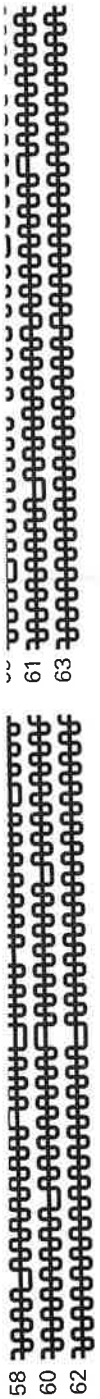


Figure 12.41 The set of 64 Walsh waveforms.

the signals from different base stations would be somewhat correlated (which is certainly not desired). The short code can also be thought of as the address of the base station. Its implementation requires two different 15-stage shift registers: one for the inphase (I) channel, and one for the quadrature (Q) channel. Each base station uses a different 64-chip offset of the I and Q codes to identify its location; thus allowing for 512 unique addresses. This is deemed to be a sufficiently large number because addresses can be reused at base stations that are sufficiently separated from one another.

To summarize the functions of the three codes: the Walsh code provides orthogonality (for channelization) among all users located in the same cell; the short PN code maintains mutual randomness among users of different cells (for base station addressing); and the long PN code provides mutual randomness among different users of the system (for privacy). For the Walsh code to provide perfect orthogonality among channels, all the users must be synchronized in time with an accuracy corresponding to a small fraction of one chip. This is theoretically possible for the forward link because transmissions to all mobiles have a common origin at the base station. However, due to multipath effects, it is more accurate to say that the Walsh codes provide partial orthogonality. To obtain similar benefits on the reverse link would require closed-loop timing control, which is not implemented in IS-95. The reduced complexity is realized at the cost of greater within-cell interference. For third-generation wideband CDMA (WCDMA) systems, this option is present [38].

The last blocks of Figure 12.40 show wideband filtering (1.25 MHz) with finite impulse response (FIR) filters and the heterodyning of a carrier wave with BPSK modulation and QPSK spreading. The same coded bits are simultaneously present on the I and Q channels, but due to the short-code scrambling, the I and Q signals are different.

12.8.4.2 Reverse Channel

Each base station can transmit a multiplex of 64 channels, where 61 or fewer channels are used for traffic. But in the reverse direction (mobile to base station), there is just a single channel (signal) being transmitted (access request or traffic). Figure 12.43 depicts a simplified block diagram of a reverse-traffic channel transmission. The general structure is similar to the forward-traffic channel shown in Figure 12.40; however, there are several important differences. In IS-95, the reverse link does not support a pilot channel, since one would be required for each mobile unit. Thus, the reverse-channel signal is demodulated noncoherently at the base station. (In IS-2000, a pilot signal is provided for each reverse channel.) Since the reverse channel is less robust than the forward channel, a more powerful rate $\frac{1}{3}$ convolutional code is used to improve performance. Also, following the interleaver, notice that the channel bits modulate a 64-ary Walsh waveform. This is the same type of waveform that was used for channelization in the forward direction. However, in the reverse direction, Walsh waveforms are used for a totally different purpose: They become the modulating waveforms. Assuming a data rate of $R = 9.6$ kilobits/s, two information bits (which after coding are transformed into six channel bits, sometimes called code

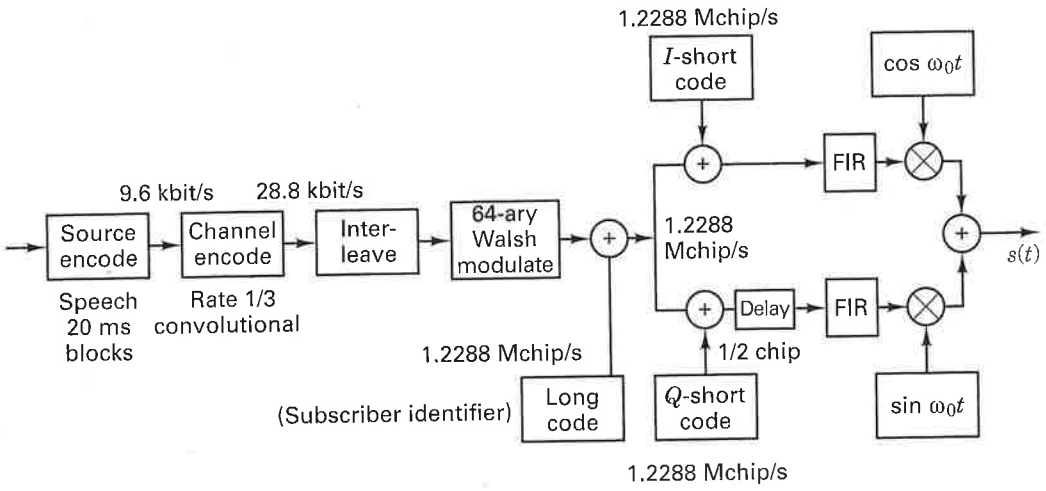


Figure 12.43 CDMA reverse-traffic channel with full-rate speech.

symbols), are mapped after interleaving into one of 64 orthogonal Walsh waveforms to be transmitted. Therefore, the Walsh waveform rate is

$$R_w = \frac{R_c}{\log_2 M} = \frac{R(n/k)}{\log_2 M} = \frac{9600 \times 3}{6} = 4{,}800 \text{ Walsh-symbols/s} \qquad (12.69)$$

where the channel-bit rate R_c is equal to the data rate times the inverse of the code rate, namely, $R\ (n/k)$. Each of the 64-ary Walsh waveforms is made up of 64 elements, termed *Walsh chips*. Then, from Equation (12.69), we see that the Walsh chip rate is $64 \times 4800 = 307{,}200$ Walsh chips/s. Thus, the modulation has resulted in some spreading (not to the full bandwidth). The Walsh chips are then repeated 4 times to arrive at the final spread-spectrum rate of 1.2288 Mchips/s.

One might ask, "Why were 64-ary Walsh functions chosen as the modulation waveforms?" Consider the trade-offs described in Section 9.7.3 for power-limited channels. A natural choice for conserving power at the expense of bandwidth is M-ary orthogonal signaling such as MFSK. The larger the value of M, the greater will be the bandwidth expansion—yet, the greater will be the reduction in required E_b/N_0 for a specified level of performance. Choosing such a signaling scheme in a narrowband system is a true trade-off—for the price of expanded bandwidth, a reduction in required power is obtained. However, in spread-spectrum systems such as those that meet IS-95 specifications, the selection of 64-ary Walsh waveforms for modulation can be seen as "getting something for nothing," because the spread-spectrum system already occupies an expanded bandwith of 1.25 MHz. The choice of 64-ary orthogonal waveforms does not expand the bandwidth any further. When you look at the Walsh waveforms in Figure 12.41, imagine the pulse shapes to be somewhat rounded. Doesn't this waveform set remind you of MFSK? Well, they are in fact similar, and at the base station, a 64-ary Walsh waveform is (generally)

detected noncoherently, much like the noncoherent detection of a 64-ary FSK tone. (Some base station receivers use coherent processing techniques, thereby providing 1-2 dB gain over noncoherent processing.)

One might ask, "Isn't channelization needed in the reverse direction?" Yes. It is always necessary to keep users separated; however, in the reverse direction, one user is distinguished from another via the long (privacy) code. In the forward direction, this code was used in a decimated fashion for privacy. In the reverse direction, as shown in Figure 12.43, the code is applied at the 1.2288 Mchips/s rate for channelization (addressing), and also for privacy, scrambling, and spreading. After spreading by this long code, the waveform is further spread by a pair of short PN codes to assure that the I and Q symbols are uncorrelated. The last blocks in Figure 12.43 show FIR filtering (1.25 MHz) and heterodyning a carrier with BPSK modulation in an offset QPSK (OQPSK) fashion. OQPSK is used in order to eliminate the possibility of the carrier wave changing phase by 180°. (See Section 9.8.1.) This feature reduces the peak-to-average power specification of the transmitter power amplifier, making its design easier. Notice that OQPSK is not used on the forward link since the transmitter sends a multiplex of 64 signals. Each forward transmission consists of a phasor representing the entire muliplex, whose resultant value can be one of a myriad of phase/amplitude possibilities. Hence, there would be no benefit from offsetting the I and Q channels, since carrier transitions through the origin could not be avoided. The final waveform is filtered to generate a spectrum with a 3-dB double-sided bandwidth of 1.25 MHz.

12.8.4.3 Receiver Structures

Mobile Receiver. The mobile receiver demodulates each of the forward-channel quadrature-BPSK waveforms coherently, using the pilot signal as a reference. The receiver structure implements a 3-finger Rake receiver to recover the three strongest multipath components (the minimum as defined in IS-95). The multipath components of the spread-spectrum signal are resolved and separated by the Rake receiver, provided that the differential path delays exceed one chip duration. FDMA waveforms cannot be so separated because they are inherently narrowband. Multipath components of TDMA waveforms can be better separated since each user transmits data in bursts. However, in a typical TDMA system, the bursts produce waveforms that are still too narrowband for multipath resolution at nominal delays. But for CDMA, if the spread-spectrum bandwidth exceeds 1 MHz, any multipath components that are separated by 1 μs delays or greater are separable. The Rake receiver tracks such paths rapidly and combines them constructively (even coherently at the mobile receiver). The operation of the Rake receiver is presented in Section 15.7.2. The soft-decision outputs of the demodulator are processed by a Viterbi decoder. The final step in recovering the information consists of determining which of the four possible data rates (9600, 4600, 2400, or 1200 bits/s) was actually utilized at the transmitter. This is accomplished without any overhead penalty by decoding the demodulated output four times, once for each of the four hypotheses. Several metrics are obtained from the decoding process and also from the pass/fail metrics of the error-detection bits. These are analyzed in order to select one final decoded sequence.

Base Station Receiver. The base station dedicates a separate channel in order to receive the transmissions of each active user in the cell. Each user's reverse-channel 64-ary Walsh-modulated signals are received noncoherently (much like the reception of noncoherent orthogonal MFSK). The receiver structure typically implements a 4-finger Rake receiver to demodulate the four strongest multipath components at the output of two antennas (see Section 15.7.2), which are spatially separated by several wavelengths for diversity reception. The soft-decision outputs of the demodulator are processed by a Viterbi decoder. The final step in recovering the information consists of decoding the demodulated data four times, using a procedure similar to that used in the forward direction, where metrics are compared in order to select one final decoded sequence.

12.8.4.4 Power Control

Power control is a necessity for a system in which many users simultaneously transmit to a base station using the same frequency. Without power control, users transmitting from locations near the base station would be received at power levels much higher than those of users transmitting from locations near the cell's edge. The main goal of power control is to adjust the users' transmitted power so as to provide at the base station an equal (and near constant) received power level from each mobile unit. In order to accomplish this, a key feature of the power-control algorithm is to command users to transmit at power levels that are inversely proportional to the received power level (from the base station). There are three power-control methods specified in IS-95: reverse-link open-loop control, forward/reverse-link closed-loop control, and forward link control.

Reverse-Link Open-Loop Control. The assumption is made that there are similar path losses on a forward and reverse channel, even though this is not completely true since they operate at frequencies that are separated by 45 MHz. The base station continually transmits a calibration constant (determined by its EIRP) on the SYNC channel. This information allows the mobile unit to use an estimated transmit power in order for the received power at the base station to be the same as that of other mobile units. Consider the following example of an open-loop control algorithm. The mobile transmission power is selected so that its transmission power plus the power received from the base station (reflecting path loss) should equal some value (for example, –73 dBm). This value is a function of the base station EIRP and appears on the SYNC channel. Before a mobile begins its transmission, it determines the power received on the forward link from the automatic gain control (AGC) circuit in its receiver. If the received power is, for example, –83 dBm, then the open-loop power-control algorithm dictates a transmit power of (–73 dBm) – (–83 dBm), or 10 dBm.

Forward/Reverse Link Closed-Loop Control. Power-control bits are sent on the forward link by "stealing" from the channel bits transmitting encoded traffic (resulting in a punctured code). Once every 6 Walsh waveforms, 2 channel bits are replaced with a power-control bit. Since the Walsh waveforms are transmitted at a

rate of 4800 waveforms/s, the rate at which the power-control bits are transmitted is 800 bits/s. Thus, there are 16 such control bits in each 20 ms frame. The goal of this closed-loop power control is to correct the open-loop power-control estimate every 1.25 ms in steps of 1 dB. Later versions added the option for step sizes of 0.5 dB and 0.25 dB. The most significant benefit of such fast and accurate closed-loop power control is a significant reduction in the average transmitter power on the reverse channel. Analog mobile radios transmit enough power to maintain a link even during fading. Thus, most of the time such analog radios transmit excessive amounts of power. CDMA mobile radios operate at power levels no greater than what is needed to close the reverse link. A mobile unit using CDMA designed to meet IS-95 specifications requires approximately 20 dB to 30 dB less power than a mobile unit operating in an analog AMPS system [30].

Forward Link Control. The base station periodically reduces the power transmitted to the mobile unit. Whenever the mobile senses an increased frame-error rate, it requests additional power from the base station. Adjustments can be made periodically based on reported frame-error rates (FER).

Example 12.6 Signaling Elements (Numerology) Used in IS-95

There is a rich set of signaling elements used in CDMA systems that are designed to IS-95 specifications: data bits, channel bits, Walsh waveforms, Walsh chips, spread-spectrum chips, and BPSK waveforms. Consider a reverse traffic channel that is carrying full-rate digitized speech at 9.6 kbits/s, with a received $E_b/(N_0 + I_0) \approx E_b/I_0 = 7$ dB (assuming that $N_0 << I_0$). Find the values of the following received power-to-noise spectral-density and energy-to-noise spectral-density parameters: P_r/I_0, E_c/I_0, E_w/I_0, E_{wch}/I_0, and E_{ch}/I_0. Also, find the values of the following rates: R_c, R_w, R_{wch}, and R_{ch}, where c, w, wch, and ch represent channel bit, Walsh waveform, Walsh chip, and spread-spectrum chip, respectively. How many spread-spectrum (SS) chips correspond to one Walsh chip?

Solution

The key to this problem lies in the fundamental relationships between received power-to-noise spectral density and each of the signaling parameters, in a manner similar to that presented in Section 9.7.7. From that development, we can write

$$\frac{P_r}{I_0} = \frac{E_b}{I_0} R = \frac{E_c}{I_0} R_c = \frac{E_w}{I_0} R_w = \frac{E_{\text{wch}}}{I_0} R_{\text{wch}} = \frac{E_{\text{ch}}}{I_0} R_{\text{ch}} \tag{12.69}$$

Since E_b/N_0 is 7 dB (or 5), and the data rate R is 9600 bits/s, from Equation (12.69) we obtain

$$\frac{P_r}{I_0} = \frac{E_b}{I_0} R = 48{,}000 \text{ Hz} \text{ or } 46.8 \text{ dB-Hz}$$

For the reverse traffic channel, the code rate is $\frac{1}{3}$. Therefore,

$$\frac{E_c}{I_0} = \left(\frac{1}{3}\right)\frac{E_b}{I_0} = \frac{5}{3} \text{ or } 2.2 \text{ dB}$$

and

$$R_c = 3 \times R = 3 \times 9600 = 28{,}800 \text{ channel bits/s}$$

Each 64-ary Walsh waveform corresponds to 6 channel bits. Therefore,

$$\frac{E_w}{I_0} = 6 \times \frac{E_c}{I_0} = 6 \times \left(\frac{5}{3}\right) = 10 \text{ or } 10 \text{ dB}$$

and

$$R_w = \left(\frac{1}{6}\right) R_c = \left(\frac{1}{6}\right) 28{,}800 = 4800 \text{ Walsh waveforms/s}$$

A Walsh waveform is composed of 64 Walsh chips. Hence,

$$\frac{E_{\text{wch}}}{I_0} = \left(\frac{1}{64}\right) \frac{E_w}{I_0} = \left(\frac{1}{64}\right) \times 10 = \frac{10}{64} \text{ or } -8.1 \text{ dB}$$

and

$$R_{\text{wch}} = 64 \times R_w = 64 \times 4800 = 307{,}200 \text{ Walsh-chips/s}$$

In IS-95, the spread-spectrum chip rate is 1.2288 Mchips/s. Thus,

$$\frac{E_{\text{ch}}}{I_0} = \frac{P_r}{I_0} \times \left(\frac{1}{R_{\text{ch}}}\right) = \left(\frac{48{,}000}{1.2288 \times 10^6}\right) = 0.039 \text{ or } -14.1 \text{ dB}$$

SS-chips per Walsh chip: $$\frac{R_{\text{ch}}}{R_{\text{wch}}} = \frac{1.2288 \times 10^6}{307{,}200} = 4$$

12.8.4.5 Typical Telephone Call Scenario

Turn on and Synchronization. Once power is applied to the mobile unit, the receiver scans continuously in search of available pilot signals. Such signals will be received from different base stations with different time-offsets of the short PN code (described in Section 12.8.4.1). The time-offset used by a base station differs by a multiple of 64 chips from all other base stations. Since the short code is maximal length, its 15-stage shift register produces $2^{15} - 1 = 32{,}767$ bits. After "bit stuffing" the sequence with one bit, 32,768 bits are produced before the whole process repeats itself. Thus, there are 32,768/64, or 512 available unique addresses. The 512 short PN codes can be generated by a simple time shift of a single PN sequence, because the base stations are time synchronized within a few microseconds of each other. At the chip rate of 1.2288 Mchips/s, there are 75 frames of the short code corresponding to a 2-second interval. The zero-offset address of the short code occurs on even second time marks. Consider the case of a base station whose address is represented by offset number 18. Then its transmission cycle begins at (18×64 chips $\times$ ($1/1.2288 \times 10^6$) s/chip, or 937.5 μs after every even-second time mark.

Once the mobile unit completes its scan and is correlated to the strongest pilot signal, it is now synchronized with one of the 512 unique base station addresses. The mobile unit can now despread any of that base station's transmissions; however, it does not yet have system time, which is needed for access, paging, and traffic channels. Next, using the pilot signal as a reference, the mobile unit coher-

ently demodulates the SYNC channel signal (Walsh 32), which the base station transmits continuously. The SYNC channel transmissions provide several system parameters, the key one being the state of a long code 320 ms in the future, giving the mobile unit time to decode, load its registers, and become system-time synchronized. This long code is one of a specific group of long codes used for access and paging. The mobile selects a predefined paging channel based on its serial number, and it monitors this paging channel for incoming calls. The mobile can now register with the base station, which allows for location-based paging rather than system-based paging when there is an incoming telephone call.

Idle-State Handoff. The mobile unit continually scans for alternative pilot signals. If it finds a stronger pilot signal from a different base station, the mobile locks on to the base station with the stronger pilot. Since there is no call in progress, the process simply serves to update the location of the mobile. The mobile has obtained system time from the SYNC channel. If there were only one base station, system time could be defined by whatever reference the base station chooses. With several operating base stations, the *handoff* process is facilitated if time is coordinated throughout the system. In IS-95, system time is specified to be *Universally Coordinated Time* (UTC) $\pm$ 3 μs. A practical way to implement this is with the use of a Global Positioning System (GPS) receiver at each base station.

Call Initiation. A call is initiated by the user keying in a telephone number and pushing the *send* button. This initiates an access probe. The mobile uses open-loop power control, choosing an initial transmission power level estimated from the pilot signal, as described in Section 12.8.4.4. All access channels use different long-code offsets. At the beginning of an access probe, the mobile pseudorandomly chooses one of the access channels associated with its paging channel. The transmission of an access probe is timed to begin at the start of an access channel slot, which is determined pseudorandomly. A key element of the access procedure involves the identification of the caller's serial number. Identification is needed because the base station cannot discriminate accesses from different users, since the access channel is a common channel.

The mobile-terminal time reference for transmission is determined by the earliest multipath component being used for demodulation. The mobile does not make transmission adjustments to account for propagation delay. Instead, the base station continually searches and tests for the presence of reverse channel signals. The mobile "listens" on the paging channel for a response from the base station. If there is none (collisions can occur during transmission on the access channels), the mobile attempts access again after waiting a pseudorandom time. When the mobile's access probe is successful, the base station response is a traffic channel assignment (Walsh code number).

Traffic channels use different long-code offsets than paging channels. Therefore, the mobile unit changes its long-code offset to one based on its serial number. After receiving the Walsh code assignment, the mobile begins an all-zeros data transmission on the traffic channel, and waits for a positive acknowledgment on the

forward traffic channel. If the exchange is successful, the next step is ringing at the telephone that was called. Conversation can then commence.

Soft Handoff. During a call, the mobile may find an alternate strong pilot signal. It then transmits a control message to its base station, identifying the new base station with the stronger pilot signal and requesting a soft handoff. The original base station passes the request to a base station controller (BSC) that handles the radio resource control of the link; the BSC may or may not be collocated with a Mobile Switching Center (MSC) that handles the non-radio aspects of the link (e.g., switching). The BSC contacts the new base station and obtains a Walsh number assignment. This assignment is sent to the mobile via its original base station connection. During the transition, the mobile is supported by (connected to) both base stations, and a land link connection is maintained from the BSC to both base stations. The mobile combines the signals received from both base stations by using the two respective pilot signals as coherent phase references. Signal reception from two base stations simultaneously is facilitated by the Rake receiver, since the transmissions from both base stations appear as multipath components to the mobile receiver. At the BSC, where the signals are received noncoherently, the two received signals from the mobile are examined, and the better one is chosen in each 20-ms frame. The original base station drops the call when connection is firmly established in the new cell. Such dual connection, sometimes called "make before break," reduces the probability of a dropped call and of poor reception at a cell's edge.

12.9 CONCLUSION

Spread-spectrum (SS) technology has only emerged since the 1950s. Yet, this novel approach to applications, such as multiple access, ranging, and interference rejection, has rendered SS techniques extremely important to most current NASA and military communication systems. In this chapter we presented an overview enumerating the benefits and types of spread-spectrum techniques, as well as some historical background.

Since SS techniques were initially developed with military applications in mind, we started the treatment with discussions of anti-jam (AJ) systems. Pseudorandom sequences are at the heart of all present-day SS systems; we therefore treated PN generation and properties. Emphasis was placed on the two major spread-spectrum techniques; direct sequence and frequency hopping. Consideration was given to synchronization, a crucial aspect of spread-spectrum operation. Also, attention was devoted to the commercial use of spread-spectrum techniques for code-division multiple access (CDMA) systems, particularly direct-sequence CDMA, as it is specified in interim standard 95 (IS-95).

REFERENCES

1. Scholtz, R. A., "The Origins of Spread Spectrum Communications," *IEEE Trans. Commun.,* vol. COM30, no. 5, May 1982, pp. 822–854.

2. Shannon, C. E., "Communication in the Presence of Noise," *Proc. IRE,* Jan. 1949, pp. 10–21.
3. Dillard, R. A., "Detectability of Spread Spectrum Signals," *IEEE Trans. Aerosp. Electron. Syst.,* July 1979.
4. Simon, M. K., Omura, J. K., Scholtz, R. A., and Levitt, B. K., *Spread Spectrum Communications,* Computer Science Press, Inc., Rockville, Md., 1985.
5. de Rosa, L. A., and Rogoff, M., Sec. I (Communications) of *Application of Statistical Methods to Secrecy Communication Systems,* Proposal 946, Fed. Telecommun. Lab., Nutley, N.J., Aug. 28, 1950.
6. Pickholtz, R. L., Schilling, D. L., and Milstein, L. B., "Theory of Spread-Spectrum Communications—A Tutorial," *IEEE Trans. Commun.,* vol. COM30, no. 5, May 1982, pp. 855–884.
7. Pickholtz, R. L., Schilling, D. L., and Milstein, L. B., Revisions to "Theory of Spread-Spectrum Communications—A Tutorial," *IEEE Trans. Commun.,* vol. COM32, no. 2, Feb. 1984, pp. 211–212.
8. Simon, M. K., Omura, J. K., Scholtz, R. A., and Levitt, B. K., *Spread Spectrum Communications,* Vol. 2, Computer Science Press, Inc., Rockville, Md., 1985.
9. Simon, M. K., and Polydoros, A., "Coherent Detection of Frequency-Hopped Quadrature Modulations in the Presence of Jamming: Part I. QPSK and QASK; Part II. QPR class I Modulation," *IEEE Trans. Commun.,* vol. COM29, Nov. 1981, pp. 1644–1668.
10. Holmes, J. K., and Chen C. C., "Acquisition Time Performance of PN Spread-Spectrum Systems," *IEEE Trans. Commun.,* COM-25, August 1977, pp. 778–783.
11. Ward, R. B., "Acquisition of Pseudonoise Signals by Sequential Estimation," *IEEE Trans. Commun.,* COM13, Ded. 1965, pp. 475–483.
12. Spilker, J. J., and Magill, D. T., "The Delay-Lock Discriminator—An Optimum Tracking Device," *Proc. IRE,* Sept. 1961.
13. Spiler, J. J., "Delay-Lock Tracking of Binary Signals," *IEEE Trans. Space Electron. Telem.,* Mar. 1963.
14. Simon, M. K., "Noncoherent Pseudonoise Code Tracking Performance of Spread Spectrum Receivers," *Commun.,* vol. COM25, Mar. 1977.
15. Ziemer, R. E., and Peterson, R. L., *Digital Communications and Spread Spectrum Systems,* Macmillan Publishing Company, New York, 1985.
16. Holmes, J. K., *Coherent Spread Spectrum Systems,* John Wiley & Sons, Inc., New York, 1982.
17. Pursley, M. B., "Performance Evaluation for Phase-Coded Spread-Spectrum Multiple-Access Communication: Part I. System Analysis," *IEEE Trans. Commun.,* vol. COM25, no. 8, Aug. 1977, pp. 795–799.
18. Geraniotis, E., "Noncoherent Hybrid DS-SFH Spread-Spectrum Multiple-Access Communications," *IEEE Trans. Commun.,* vol. COM34, no. 9, Sept. 1986, pp. 862–872.
19. Greaniotis, E., and Pursley, M. B., "Error Probability for Direct-Sequence Spread-Spectrum Multiple-Access Communications: Part I. Upper and Lower Bounds," *IEEE Trans. Commun.,* vol. COM30, no. 5, May 1982, pp. 985–995.
20. Geraniotis, E., and Pursley, M. B., "Error Probabilities for Direct-Sequence Spread-Spectrum Multiple-Access Communications: Part II. Approximations," *IEEE Trans. Commun.,* vol. COM30, no. 5, May 1982, pp. 996–1009.
21. Schilling, D. L., Milstein, L. B., Pickholtz, R. L., and Brown, R. W., "Optimization of the Processing Gain of an *M*-ary Direct Sequence Spread Spectrum Communication System," *IEEE Trans. Commun.,* vol. COM28, no. 8, Aug. 1980, pp. 1389–1398.

22. Viterbi, A. J., and Jacobs, I. M., "Advances in Coding and Modulation for Noncoherent Channels Affected by Fading, Partial Band, and Multiple Access Interference," in A. S. Viterbi, ed., *Advances in Communication Systems,* Vol. 4, Academic Press, Inc., New York, 1975.
23. Stark, W. E., "Coding for Frequency-Hopped Spread-Spectrum Communication with Partial-Band Interference: Part I. Capacity and Cutoff Rate," *IEEE Trans. Commun.,* vol. COM33, no. 10, Oct. 1985, pp. 1036–1044.
24. Stark, W. E., "Coding for Frequency-Hopped Spread-Spectrum Communication with Partial-Band Interference: Part II. Coded Performance," *IEEE Trans. Commun.,* vol. COM33, no. 10, Oct. 1985, pp. 1045–1057.
25. Milstein, L. B., Davidovici, S., and Schilling, D. L., "The Effect of Multiple-Tone Interfering Signals on a Direct Sequence Spread Communication System," *IEEE Trans. Commun.,* vol. COM30, Mar. 1982, pp. 436–446.
26. Milstein, L. B., Pickholtz, R. L., and Schilling, D. L., "Optimization of the Processing Gain of an FSK-FH system," *IEEE Trans. Commun.,* vol. COM28, July 1980, pp. 1062–1079.
27. Huth, G. K., "Optimization of Coded Spread Spectrum Systems Performance," *IEEE Trans. Commun.,* vol. COM25, Aug. 1977, pp. 763–770.
28. Viterbi, A. J., "Spread Spectrum Communications—Myths and Realities," *IEEE Commun. Mag.,* May 1979, pp. 11–18.
29. Simon, M. K., Omura, J. K., Scholtz, R. A., and Levitt, B., *Spread Spectrum Communications Handbook,* Revised Edition, McGraw-Hill, Inc., New York, 1994.
30. Viterbi, A. J., "The Orthogonal-Random Waveform Dichotomy for Digital Mobile Personal Communication," *IEEE Personal Communications,* First Quarter 1994, pp. 18–24.
31. Kohno, R., Meidan, R., and Milstein, L. B., "Spread Spectrum Access Methods for Wireless Communications," *IEEE Communications Magazine,* January 1995, pp. 58–67.
32. Pickholtz, R. L., Milstein, L. B., and Schilling, D. L., "Spread Spectrum for Mobile Communications," *IEEE Trans. Vehicular Tech.,* vol. 40, no. 2, May 1991, pp. 313–321.
33. Morrow, R. K., Jr. and Lehnert, J. S., "Bit-to-Bit Error Dependence on Slotted DS/SSMA Packet Systems with Random Signature Sequences," *IEEE Trans. Commun.,* vol. 37, no. 10, October 1989, pp. 1052–1061.
34. Schilling, D. L., et al., "Spread Spectrum for Commercial Communications," *IEEE Communications Magazine,* April 1991, pp. 66–78.
35. Gilhousen, K. S., "On the Capacity of a Cellular CDMA System," *IEEE Trans. Vehicular Tech.,* vol. 40, no. 2, May 1991, pp. 303–312.
36. Viterbi, A. M., and Viterbi, A. J., "Erlang Capacity of a Power Controlled CDMA System," *IEEE JSAC, vol. 11, no. 6, pp. 892–899.*
37. Padovani, R., "Reverse Link Performance of IS-95 Based Cellular Systems, *IEEE Personal Communications,* Third Quarter 1994, pp. 28–34.
38. Wideband CDMA Special Issue, *IEEE Communications Magazine,* vol. 36, no. 9, September 1998.

PROBLEMS

12.1. Explain why a maximal-length n-stage linear feedback shift register can produce a sequence with a period no greater than $2^n - 1$.

12.2. Show that in a maximal-length n-stage linear feedback shift register the output stage must always be an input to the feedback network.

12.3. Consider the DS/BPSK spread-spectrum transmitter of Figure 12.9a or b. Let $x(t)$ be the sequence 1 0 0 1 1 0 0 0 1, arriving at a rate of 75 bits/s, where the leftmost bit is the earliest bit. Let $g(t)$ be generated by the shift register of Figure 12.7, with an initial state of 1 1 1 1 and a clock rate of 225 Hz.

(a) Sketch the final transmitted sequence $x(t)g(t)$.

(b) What is the bandwidth of the transmitted (spread) signal?

(c) What is the processing gain?

(d) Suppose that the estimated delay, $\hat{T}_d$, of Figure 12.9c is too large by one chip time. Sketch the despread chip sequence.

(e) Choose a decision rule for deciding on $\hat{x}(t)$ and identify the errors.

12.4. A total of 24 equal-power terminals are to share a frequency band through a code-division multiple access (CDMA) system. Each terminal transmits information at 9.6 kbits/s with a direct-sequence spread-spectrum BPSK modulated signal. Calculate the minimum chip rate of the PN code in order to maintain a bit error probability of 10^{-3}. Assume that the receiver noise is negligible with respect to the interference from the other users.

12.5. A feedback shift register PN generator produces a 31-bit PN sequence at a clock rate of 10 MHz. What are the equation and graphical form of the autocorrelation function and power spectral density of the sequence? Assume that the pulses have values of ± 1.

12.6. Consider an FH/MFSK system such as the one shown in Figure 12.11. Let the PN generator be defined by a 20-stage linear feedback shift register with a maximal length sequence. Each state of the register dictates a new center frequency within the hopping band. The minimum step size between center frequencies (hop to hop) is 200 Hz. The register clock rate is 2 kHz. Assume that 8-ary FSK modulation is used and that the data rate is 1.2 kbits/s.

(a) What is the hopping bandwidth?

(b) What is the chip rate?

(c) How many chips are there in each data symbol?

(d) What is the processing gain?

12.7. The block diagram of Figure 12.16 is described in Section 12.4.5 for a fast frequency hopping (FFH) demodulator. Draw a similar block diagram for a slow frequency hopping (SFH) demodulator, and explain how it would work.

12.8. Find the mean and the standard deviation of the time needed to acquire a 10-megachip/s BPSK modulated PN code sequence using a serial search where 100 chips are examined at a time. Assume that a correct detection results when all 100 received chips match the locally generated ones. The ratio of received chip energy to noise power spectral density is 9.6 dB, and the uncertainty time between the received and local code sequences is 1 ms. Assume that the probability of false lock (false alarm) is negligible.

12.9. There are 11 equal-power terminals in a CDMA communication system, transmitting signals toward a central node. Each terminal transmits information at 1 kbit/s on a 100-kchips/s direct-sequence spreading signal using BPSK modulation.

(a) If receiver noise is negligible with respect to the interference from other users, what is the ratio of bit energy to interference power spectral density (E_b/I_0) received by a receiving terminal?

(b) What is the effect on E_b/I_0 if all users double their output power?

(c) If the users wish to expand their service to 101 equal-power users, what must be done to the spreading codes to maintain the original E_b/I_0 ratio?

12.10. A CDMA system uses direct-sequence modulation with a data bandwidth of 10 kHz and a spread bandwidth of 10 MHz. With only one signal being transmitted, the received E_b/N_0 is 16 dB.

(a) If the required $(E_b/N_0 + I_0)$ is 10 dB, how many equal-power users can share the band? Do not neglect receiver noise.

(b) If each user's transmitted power is reduced by 3 dB, how many equal-power users can share the band?

(c) If the received $E_b/N_0 \to \infty$ for each receiver, what is the maximum number of users that can share the band?

12.11. A DS/SS system is used to combat multipath. If the path length of the multipath wave is 100 m longer than that of the direct wave, what is the minimum chip rate necessary to reject the multipath interference?

12.12. A ground-to-synchronous satellite link must be closed in a jamming environment. The data rate is 1 kbit/s and the ground station has a 60-ft antenna. Antijam protection is provided by a 10-Mbits/s direct-sequence spread-spectrum code. The jammer has a 150-ft antenna and a transmitter with 400 kW of power. Assume equal space and propagation losses. How much power is required of the earth station transmitter to achieve an E_b/J_0 of 16 dB at the satellite receiver? Assume that the receiver noise is negligible.

12.13. Input data at 75 bits/s are channel encoded using a rate $\frac{1}{2}$ encoder. The coded bits are then modulated using 8-ary FSK. The FSK symbols are then spread by frequency hopping at a rate of 2000 hops/s.

(a) What is the chip rate?

(b) What is the order of diversity?

(c) If there are two such signals, time-division multiplexed (TDM'd) on the channel at the same hopping rate, how would this affect the chip rate, symbol rate, and order of diversity?

(d) If there are 80 such signals TDM's on the channel, how would this effect the chip rate, symbol rate, and order of diversity?

12.14. A frequency hopping noncoherent binary FSK system operates at an E_b/N_0 of 30 dB with a hopping bandwidth of 2 GHz. Assume that no channel coding is used. A jammer operating over the same broadband bandwidth yields a received $J_0 = 100N_0$.

(a) What is the bit error probability, P_B?

(b) If the jammer becomes a partial-band jammer, what bandwidth should it occupy to be most effective?

(c) What is P_B as a result of such optimum partial-band jamming?

(d) What is the unjammed P_B?

12.15. A noncoherent frequency hopping 8-ary FSK system hops at 12,000 hops/s over a bandwidth of 1 MHz. The symbol rate is 3000 symbols/s. Assume that channel coding is not used. The signal power at the input of the receiver is 10^{-12} W. A partial-band noise jammer occupies 50 kHz (assumed to be entirely within the hopping bandwidth of the signal). The received jammer power is 10^{-11} W. Assume that the system temperature is 290 K. What is the probability of bit error?

12.16. A coherent DS/BPSK system is transmitting at a data rate of 10 kbits/s in the presence of a broadband jammer. Assume that the system does not use channel coding. Also assume that the propagation losses are the same for the system and the jammer.

(a) If the EIRP of the communicator is 20 kW and the EIRP of the jammer is 60 kW, calculate the required spread-spectrum bandwidth to achieve a bit error probability of $P_B = 10^{-5}$.

(b) If the jammer is a pulse jammer, calculate the pulse duty cycle that results in worst-case jamming. What is the value of P_B at this duty cycle?

12.17. A communicator intends to use frequency hopping at a hop rate of 10,000 hops/s to avoid a threat of repeat-back jamming.

(a) Ignoring the curvature of the earth, and assuming that the communicator is transmitting to a satellite of geosynchronous altitude (approximately 36,000 km) that is directly overhead, compute the *radius of vulnerability,* which is the radius outside of which the communicator is unconditionally safe from repeat-back jamming by a ground-based jammer.

(b) If the communicator knows that the jammer requires a minimum of 10 μs to identify the transmission frequency and tune the jammer output, compute the radius of vulnerability conditioned on this information.

12.18. Consider an airborne repeat-back jammer as shown in Figure P12.1. The communicator is using a FH/SS system. What is the minimum hop rate required in order that the repeat-back jamming does not degrade the message? What would be the minimum required hopping rate if the communicator and jammer switched positions (i.e., fixed land jammer and airborne communicator).

12.19. Spread-spectrum techniques can be used to meet government regulations regarding flux (power) density radiating the surface of the earth. If a satellite at synchronous altitude (36,000 km) transmit 4-kbits/s data using 100 W of EIRP, what spreading bandwidth is required to maintain a flux density on the earth's surface no greater than -151 dBW/m^2 in any 4-kHz band?

12.20. A communicator uses noncoherent BFSK modulation and frequency hopping to combat the effects of a jammer. The power of the communicator's signal at the re-

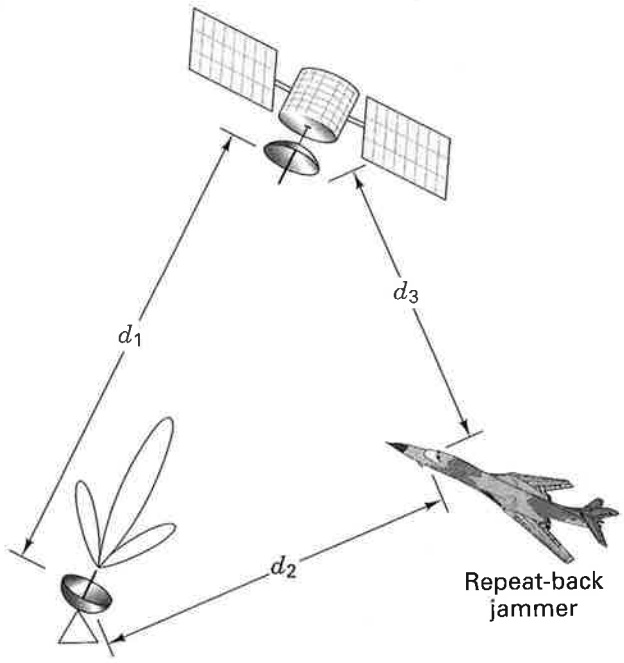


Figure P12.1

ceiver input is 10 μW. The SNR in the absence of jamming is assumed to be very large. The power of the jamming signal at the receiver input is 1 W.

(a) If the jammer jams the entire hopping bandwidth with equal amounts of Gaussian noise (the noise will be white within the band), what bandwidth expansion factor will allow the communicator to maintain a bit error probability of 10^{-4}?

(b) Assume that the jammer decides to "color" its jamming noise by reducing its energy by a fraction, α ($0 \leq \alpha \leq 1$), in half the hop bandwidth, and increasing it by a like amount in the other half (thereby keeping its transmitted energy constant). Assuming that the communicator does not modify his hopping pattern to avoid the jammer strategy, develop an expression for the bit error probability for this case of colored jamming.

(c) Determine the fraction, α, that is optimum from the jammer standpoint for each of the limiting cases (i) when the effective SNR is large and, (ii) when it is small.

12.21. Spread-spectrum (SS) techniques can provide impressive error-performance benefits against interfering signals. One might therefore think that such SS techniques might provide similar benefits against AWGN. Explain why this is not possible.

12.22. A hand-held direct-sequence spread-spectrum (DS/SS) radio is part of a cellular CDMA system. The system specifications are as follows: data and SS-code modulation is BPSK, data rate is 8,000 bits/s, carrier frequency is 1 GHz, chip rate is 25 Mchips/s, worst-case path loss is 138.6 dB, gain of transmitting antenna is 5 dBI, receiver figure-of-merit G/T is −18 dB/K, occasional deep small-scale fading loss is 30 dB, other losses are 4 dB, required E_b/N_0 is 4 dB. The factors G_A, G_V, H_0, and γ are 2.5, 2.5, 1.6, and 1, respectively. Hint: Refer to Chapter 5 for treatment of link parameters.

(a) Find the required transmitter power P_t during deep small-scale fading.

(b) To what level can P_t be powered down when there is no small-scale fading?

(c) What is the minimum required E_{ch}/N_0 to meet the specifications?

(d) What is the processing gain?

(e) What is the maximum number of users per cell?

12.23. A direct-sequence spread-spectrum system using BPSK modulation for both the data and the code is required to support a data rate of 9600 bits/s. The received predetection power-received versus N_0 (P_r/N_0) is 48-dB Hz, and the SS-processing gain is 1000. A BCH (63, 51) error-correcting code is used. Verify that these system specifications can provide a bit-error probability of 10^{-4}. Hint: Use Equation (6.46) in Chapter 6, for computing decoded bit-error probability.

12.24. **(a)** Consider a CDMA direct-sequence cellular telephone system, where each user requires an E_b/I_0 of 6 dB for acceptable voice quality. The chip rate is 3.68 Mchips/s, and the data rate is 14.4 kbits/s. Assume that the factors γ, G_V, and H_0, are 1.5, 2.5, and 1.5, respectively, and that transmission ceases during speech pauses. How many users per cell can be supported?

(b) If a powerful error-correcting code is used to lower the required E_b/I_0 by 1 dB, how many users per cell can be supported?

12.25. A direct-sequence spread-spectrum system uses QPSK modulation for transmitting data. It is required that the bit-error probability be 10^{-5} and that $E_{ch}/I_0 \leq -30.4$ dB. Assuming perfect synchronization, what is the minimum number of chips/bit required?

12.26. A direct-sequence spread-spectrum system with a processing gain of 20 dB uses QPSK modulation for transmitting data. A rate $\frac{1}{2}$ error-correcting code is used, and

the required bit-error probability is 10^{-5}. Assuming perfect synchronization, what is the minimum value of E_{ch}/I_0 and E_c/I_0 needed to support these requirements?

12.27. **(a)** A fast frequency-hopping spread-spectrum (FFH/SS) system uses 8-ary FSK modulation and a rate $\frac{1}{2}$ error-correcting code. Chips are transmitted with a repeat factor of $N = 4$. That is, each symbol is sent four times, each on a different hop. The required E_b/I_0 is 13 dB, the chip rate is 32 kchips/s, and the hopping bandwidth is 1.2 MHz. Find the data rate R, the processing gain G_p, the (P_r/I_0), the E_{ch}/I_0, the E_s/I_0, and the E_c/I_0.

(b) Will this system meet the FCC Part-15 requirements in the ISM band, for processing gain and bandwidth?

12.28. Consider a CDMA cellular telephone system designed to meet a standard similar to IS-95, with the following modifications. The spread-spectrum chip rate is 10.24 Mchips/s, and the data rate is 20 kbits/s. The reverse link uses a 256-ary Walsh waveform for modulating the rate $\frac{1}{2}$ coded data, requiring an E_b/I_0 of 6 dB. Find the values of the following received power-to-interference spectral density and energy-to-interference spectral density parameters: P_r/I_0, E_c/I_0, E_w/I_0, E_{wch}/I_0, and E_{ch}/I_0. Also, find the values of the following rates: R_c, R_w, and R_{wch}, where *c*, *w*, wch, and ch represent channel bit, Walsh waveform, Walsh chip, and spread-spectrum chip, respectively. What is the processing gain, and how many spread-spectrum chips correspond to one Walsh chip?

QUESTIONS

12.1. Frequency modulation (FM) and pulse code modulation (PCM) represent techniques that spread the spectrum of an information signal. Yet, such FM and PCM signals do not qualify as *spread-spectrum* signals. Why not? (See Section 12.1.)

12.2. List four *beneficial attributes* of spread-spectrum systems. (See Section 12.1.1.)

12.3. Describe three *randomness* properties that make pseudorandom signals appear to be random. (See Section 12.2.1.)

12.4. Define the term *chip,* in the context of a direct-sequence system, and in the context of a frequency-hopping system. (See Sections 12.3.2 and 12.4.4.)

12.5. What is meant by a *robust signal?* (See Section 12.4.2.)

12.6. What is the difference between *fast hopping* and *slow hopping?* (See Section 12.4.4.)

12.7. How does the *processing gain* parameter differ for direct-sequence systems compared with frequency-hopping systems? (See Sections 12.3.2 and 12.4.6.)

12.8. Explain how spread-spectrum systems can reliably receive signals that are "*buried in the noise.*" (See Example 12.5.)

12.9. For systems designed to meet IS-95, *Walsh codes* are used in the *forward* and *reverse* links for completely different functions. Explain the functions? (See Sections 12.8.4.1 and 12.8.4.2.)

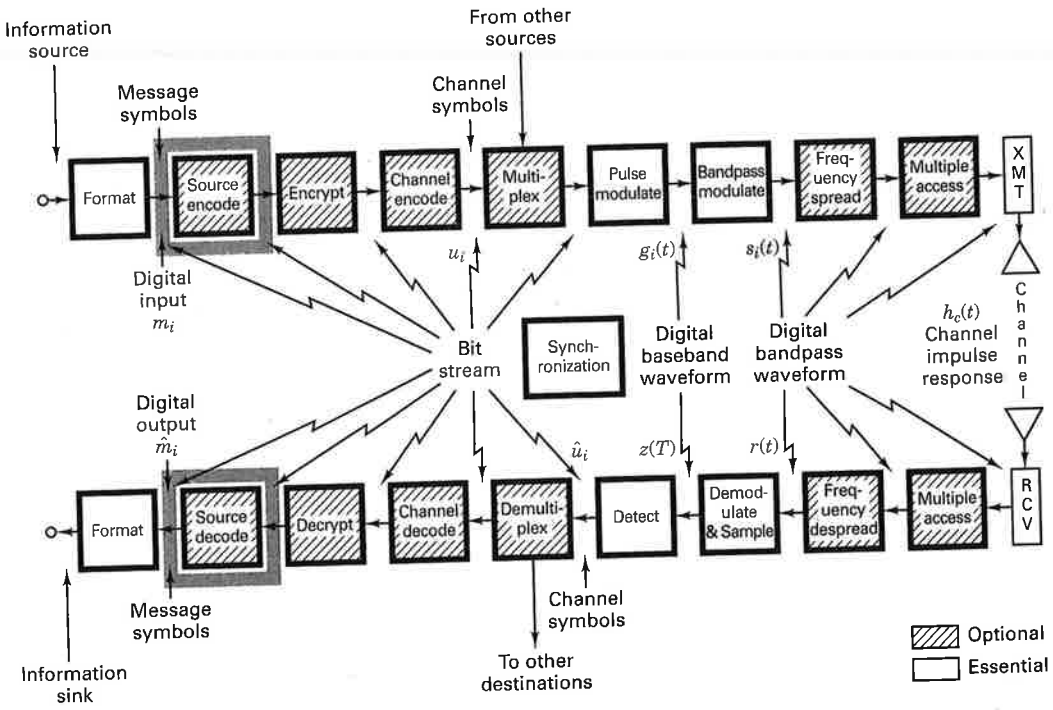
EXERCISES

Using the Companion CD, run the exercises associated with Chapter 12.

CHAPTER 13

Source Coding

Fredric J. Harris
San Diego University
San Diego, California



13.1 SOURCES

Source coding deals with the task of forming efficient descriptions of information sources. Efficient descriptions permit a reduction in the memory or bandwidth resources required to store or to transport sample realizations of the source data. For discrete sources, the ability to form reduced data-rate descriptions is related to the information content and the statistical correlation among the source symbols. For analog sources, the ability to form reduced data rate descriptions, subject to a stated fidelity criterion is related to the amplitude distribution and the temporal correlation of the source waveform. The goal of source coding is to form good fidelity description of the source for a given available bit rate, or to permit low bit rates to obtain a specified fidelity description of the source. To understand where the tools and techniques of source coding are effective, it is important to have common measures of source parameters. For this reason, in this section, we examine simple models of discrete and analog sources, and then we describe how source coding can be applied to these models.

13.1.1 Discrete Sources

A discrete source generates (or emits) a sequence of symbols $X(k)$, selected from a source alphabet at discrete time intervals kT, where $k = 1, 2, \ldots$ is a counting index. If the alphabet contains a finite number of symbols, say N symbols, the source is said to be a *finite discrete source.* An example of such a source is the

output of a 12-bit digital-to-analog converter (which outputs one of 4096 discrete levels), or the output of a 10-bit analog-to-digital converter (which outputs one of 1024 binary 10-tuples). Another example of a discrete source is the succession of 8-bit ASCII characters emitted by a computer keyboard.

A finite discrete source is defined by the list of source symbols (sometimes called the alphabet) and the probability assigned to these symbols or (letters). We will assume that the source is short-term stationary—that is, the probability assignment is fixed over the observation interval. An example in which the alphabet is fixed but the probability assignment changes is found in the sequence of symbols emitted by a keyboard when someone is typing English text, followed by typing Spanish text, and finally by typing French text.

If we know that the probability of each symbol X_j is $P(X_j)$, we can determine the *self-information* $I(X_j)$ for each symbol in the alphabet set:

$$I(X_j) = -\log_2(p_j) \tag{13.1}$$

The average self-information for the symbols in an alphabet, also called the *source entropy,* is

$$H(X) = \mathbf{E}\{I(X_j)\} = -\sum_{j=1}^{N} p_j \log_2(p_j) \tag{13.2}$$

where $\mathbf{E}\{X\}$ is the expected value of X. The source entropy is defined as the average amount of information per source output. The source entropy is the average amount of uncertainty that is resolved by the use of the alphabet. It is thus the average amount of information that must be moved through the communication channel to resolve that uncertainty. It can be shown that this amount of information in bits per symbol is bounded below by zero if there is no uncertainty, and it is bounded above by $\log_2(N)$ if there is maximum uncertainty.

$$0 \leq H(X) \leq \log_2(N) \tag{13.3}$$

Example 13.1 Entropy of a Binary Source

Consider the binary source that generates independent symbols 0 and 1, with probabilities equal to p and $(1 - p)$, respectively. We described this source in Section 9.4.2 and presented its entropy function in Figure 9.5. If $p = 0.1$ and $(1 - p) = 0.9$, the source entropy is

$$\begin{aligned} H(X) &= -[p \log_2(p) + (1 - p) \log_2(1 - p)] \\ &= 0.47 \text{ bit/symbol} \end{aligned} \tag{13.4}$$

Thus, this source can be described (with the use of appropriate coding) with less than half a bit per symbol, rather than with one-bit per symbol in its present form.

We note that the first reason that source coding works is because the information content of an N-symbol alphabet used in real communication systems is usually less than the upper bound of Equation (13.3). We know from experience, as we have noted in Example 9.2, that the symbols of English text are not equally likely. For instance, we use the high probability of certain letters in text as part of the

strategy to initialize the game of Hangman. (In this game, a player must guess the letters, but not the positions, of a hidden word of known length. Penalties accrue to false guesses, and the letters of the entire word must be determined prior to the occurrence of six false guesses.)

A discrete source is said to be *memoryless* if the symbols emitted by the source are statistically independent. In particular, this means that for the symbols taken two at a time, the joint probability of the two elements is simply the product of their respective probabilities:

$$P(X_j, X_k) = P(X_j | X_k)P(X_k) = P(X_j)P(X_k) \tag{13.5}$$

A result of statistical independence is that the information required to transmit a sequence of M symbols (called an M-tuple) from a given alphabet is precisely M times the average information required to transmit a single symbol. This happens because the probability of a statistically independent M-tuple is given by

$$P(X_1, X_2, \ldots\ldots, X_M) = \prod_{m=1}^{M} P(X_m) \tag{13.6}$$

so that the average entropy per symbol of a statistically independent M-tuple is

$$\begin{aligned} H_M(X) &= \frac{1}{M}\,\mathbf{E}\{-\log_2 P(X_1, X_2, \ldots, X_M)\} \\ &= \frac{1}{M}\sum_{X_m}[-P(X_m)\log_2 P(X_m)] \\ &= H(X) \end{aligned} \tag{13.7}$$

A discrete source is said to have memory if the source elements composing the sequence are not independent. The dependency between symbols means that in a sequence of M symbols, there is reduced uncertainty about the M-th symbol when we know the previous (M-1) symbols. For instance, is there much uncertainty about the next symbol for the 10-tuple CALIFORNI_? The M-tuple with dependent symbols contains less information, or resolves less uncertainty, than does one with independent symbols. The entropy of a source with memory is the limit

$$H(X) = \lim_{M\to\infty} H_M(X) \tag{13.8}$$

We observe that the entropy of an M-tuple from a source with memory is always less than the entropy of a source with the same alphabet and symbol probability, but without memory:

$$H_M(X)_{\text{memory}} < H_M(X)_{\text{no memory}} \tag{13.9}$$

For example, given a symbol (or letter) "q" in English text, we know that the next symbol will probably be a "u". Hence, in a communication task, being told that the letter "u" follows a letter 'q' adds little information to our knowledge of the word being transmitted. As another example, given the letters 'th' the most likely symbol to follow is one of the following: a, e, i, o, u, r, and, space. Thus, adding the next symbol to the given set resolves some uncertainty, but not much. A formal state-

ment of this awareness is that the average entropy per symbol of an M-tuple from a source with memory *decreases as the length M increases.* A consequence is that it is more efficient to encode symbols from a source with memory in groups of several symbols rather than to encode them one symbol at a time. For purposes of source encoding, encoder complexity, memory constraints, and delay considerations limit the size of symbol sequences treated as a group.

To help us understand the gains to be had in coding sources with memory, we form simple models of these sources. One such model is called a *first-order Markov source* [1]. This model identifies a number of states (or symbols in the context of information theory) and the conditional probabilities of transitioning to each next state. In the first-order model, the transition probabilities depend only on the present state. This is, $P(X_{i+1}|X_i, X_{i-1}, \ldots) = P(X_{i+1}|X_i)$. The model's memory does not extend beyond the present state. In the context of a binary sequence, this expression gives the probability of the next bit conditioned on the value of the current bit.

Example 13.2 Entropy of a Binary Source With Memory

Consider the binary (i.e., two symbol) first-order Markov source described by the state transition diagram shown in Figure 13.1. The source is defined by the state transition probabilities P(0|1) and P(1|0) of 0.45 and 0.05, respectively. The entropy of the source X is the weighted sum of the conditional entropies that correspond to the transition probabilities of the model. That is,

$$H(X) = P(0)H(X|0) + P(1)H(X|1) \tag{13.10}$$

where

$$H(X|0) = -[P(0|0)\log_2 P(0|0) + P(1|0)\log_2 P(1|0)]$$

and

$$H(X|1) = -[P(0|1)\log_2 P(0|1) + P(1|1)\log_2 P(1|1)]$$

The a priori probability of each state is found by the total probability equations

$$P(0) = P(0|0)P(0) + P(0|1)P(1)$$
$$P(1) = P(1|0)P(0) + P(1|1)P(1)$$
$$P(0) + P(1) = 1$$

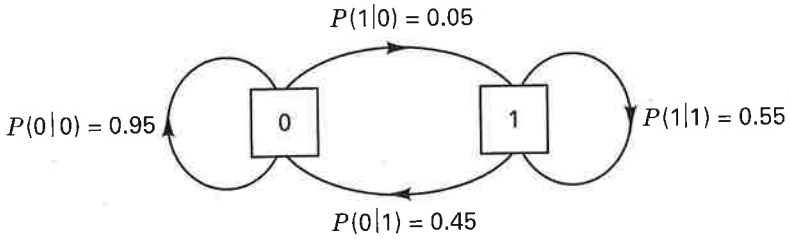


Figure 13.1 State transition diagram for first-order Markov model.

Solving for the a priori probabilities using the transition probabilities, we have

$$P(0) = 0.9 \quad \text{and} \quad P(1) = 0.1$$

Solving for the source entropy using Equation (13.10), we have

$$\begin{aligned} H(X) &= P(0)\,H(X|\,0) + P(1)\,H(X|\,1) \\ &= (0.9)(0.286) + (0.1)(0.993) \\ &= 0.357 \text{ bit/symbol} \end{aligned}$$

Comparing this result with the result of Example 13.1, we see that the source with memory has lower *entropy* than the source without memory, even though the a priori symbol probabilities are the same.

Example 13.3 Extension Codes

The alphabet for the binary Markov source of Example 13.2 consists of 0 and 1, occurring with probabilities 0.9 and 0.1, respectively. Successive symbols are not independent, and we can define a new set of code symbols as binary 2-tuples (an extension code) to take advantage of this dependency:

Binary 2-tuple	Extension Symbol	Extension Symbol Probability
00	a	$P(a) = P(0\|0)P(0) = (0.95)(0.9) = 0.855$
11	b	$P(b) = P(1\|1)P(1) = (0.55)(0.1) = 0.055$
01	c	$P(c) = P(0\|1)P(1) = (0.45)(0.1) = 0.045$
10	d	$P(d) = P(1\|0)P(0) = (0.05)(0.9) = 0.045$

where the rightmost digit of the 2-tuple is the earliest digit. The entropy for this code extension alphabet is found by using an extension of Equation (13.10) as follows:

$$H(\mathbf{X}_2) = P(a)\,H(\mathbf{X}_2|\,a) + P(b)\,H(\mathbf{X}_2|\,b) + P(c)\,H(\mathbf{X}_2|\,c) + P(d)\,H(\mathbf{X}_2|\,d)$$

$$H(\mathbf{X}_2) = 0.825 \text{ bit/output symbol}$$

$$H(\mathbf{X}_2) = 0.412 \text{ bit/input symbol}$$

where $\mathbf{X}_k$ is the kth-order extension of the source $\mathbf{X}$. A longer extension code, which takes advantage of the adjacent symbol dependency, is of the following form:

Binary 3-tuple	Extension Symbol	Extension Symbol Probability
000	a	$P(a) = P(0\|00)P(00) = (0.95)(0.855) = 0.8123$
100	b	$P(b) = P(1\|00)P(00) = (0.05)(0.855) = 0.0428$
001	c	$P(c) = P(0\|01)P(01) = (0.95)(0.045) = 0.0428$
111	d	$P(d) = P(1\|11)P(11) = (0.55)(0.055) = 0.0303$
110	e	$P(e) = P(1\|10)P(10) = (0.55)(0.045) = 0.0248$
011	f	$P(f) = P(0\|11)P(11) = (0.45)(0.055) = 0.0248$
010	g	$P(g) = P(0\|10)P(10) = (0.45)(0.045) = 0.0203$
101	h	$P(h) = P(1\|01)P(01) = (0.05)(0.045) = 0.0023$

Again, using an extension of Equation (13.10), the entropy for this extension code is found to be

$$H(\mathbf{X}_3) = 1.223 \text{ bit/output symbol}$$

$$H(\mathbf{X}_3) = 0.408 \text{ bit/input symbol}$$

We note that the entropy of the one-symbol, two-symbol, and three-symbol descriptions of the source (0.470-, 0.412-, and 0.408-bit, respectively) are decreasing asymptotically toward the source entropy of 0.357 bit/input symbol. Remember that the source entropy is the lower bound in bits per input symbol for this (infinite memory) alphabet, and this bound cannot be achieved with finite length coding.

13.1.2 Waveform Sources

A waveform source is a random process of some random variable. We classically consider this random variable to be time, so that the waveform of interest is a time-varying waveform. Important examples of time-varying waveforms are the outputs of transducers used in process control, such as temperature, pressure, velocity, and flow rates. Examples of particularly high interest include speech and music. The waveform can also be a function of one or more spatial variables (e.g., displacement in x and y). Important examples of spatial waveforms include single images, such as a photograph, or moving images, such as successive images (at 24-frames/sec) of moving picture film. Spatial waveforms are often converted to time-varying waveforms by a scanning operation. This is done, for example, for facsimile and Joint Photographic Expert Group (JPEG) transmission, as well as for standard broadcast television transmission.

13.1.2.1 Amplitude Density Functions

Discrete sources were described by their list of possible elements (called letters of an alphabet) and their multidimensional probability density functions (pdf) of all orders. By analogy, waveform sources are similarly described in terms of their probability density functions as well as by parameters and functions derived from these functions. We model many waveforms as random processes with classical probability density functions and with simple correlation properties. In the modeling process, we distinguish between short-term or local (time) characteristics and long-term or global characteristics. This partition is necessary because many waveforms are nonstationary.

The probability density function of the actual process may not be available to the system designer. Sample density functions can of course be rapidly formed in real time during a short preliminary interval and used as reasonable estimates over the subsequent interval. A less ambitious task is simply to form short-term waveform-related averages. These include the sample mean (or time average value), the sample variance (or mean-square value of the zero mean process), and the sample correlation coefficients formed over the prior sample interval. In many applications of waveform analysis, the input waveform is converted into a zero-mean process by subtracting an estimate of its mean value. For instance, this happens, in comparators used in analog-to-digital converters for which auxiliary circuitry measures the internal dc-offset voltages and subtracts them in a process known as *autozero*. Further, the variance estimate is often used to scale the input waveform to match the dynamic amplitude range of subsequent waveform conditioning circuitry. This process, performed in a data collection process, is called

autoranging or *automatic gain control* (AGC). The function of these signal-conditioning operations, mean removal, and variance control or gain adjustment (shown in Figure 13.2) is to normalize the probability density functions of the input waveforms. This normalization assures optimal utility of the limited dynamic range of subsequent recording, transmission, or processing subsystems.

Many waveform sources exhibit significant amplitude correlation in successive time intervals. This correlation means that signal levels in successive time intervals are not independent. If the time signal is independent over successive intervals, the autocorrelation function would be an impulse function. Many signals of engineering interest have finite width correlation functions. The effective width of the correlation function (in seconds) is called the correlation time of the process and is akin to the time constant of a low-pass filter. This time interval is an indication of how much shift along the time axis is required to decorrelate the data. If the correlation time is large we interpret this to mean that the waveform makes significant amplitude changes slowly. Conversely, if the correlation time is small, we infer that the waveform makes significant changes in amplitude very quickly.

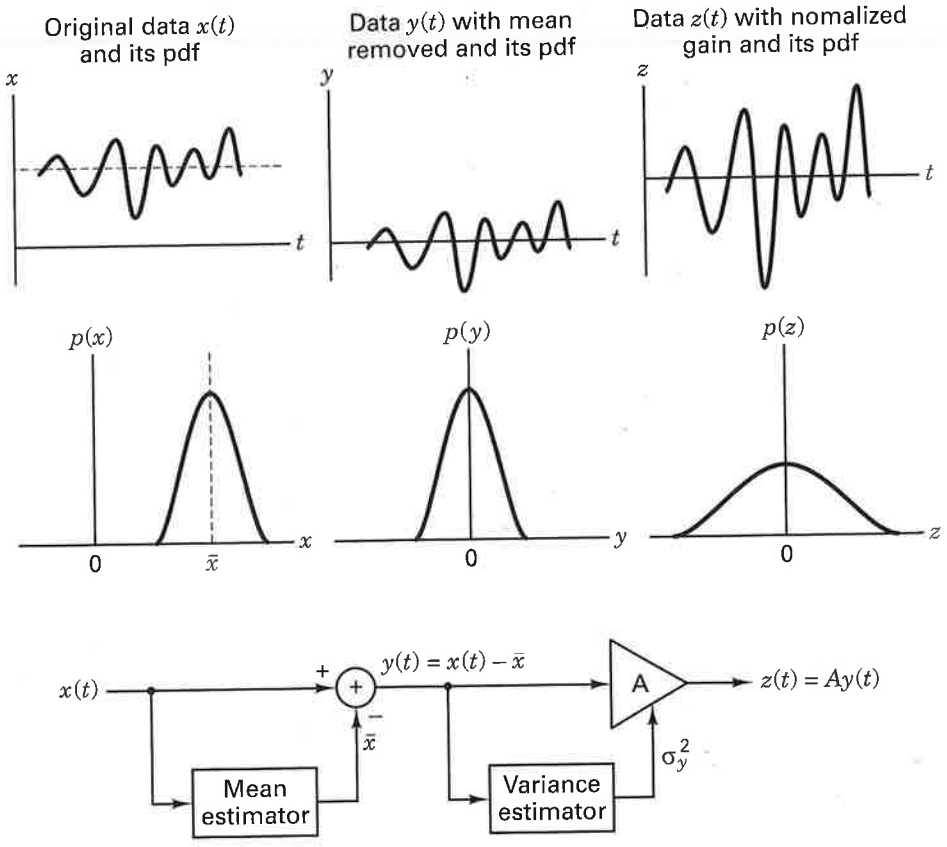


Figure 13.2 Mean removal and variance normalization (gain adjustment) for a data dependent signal conditioning system.

13.2 AMPLITUDE QUANTIZING

Amplitude quantizing is the task of mapping samples of a continuous amplitude waveform to a finite set of amplitudes. The hardware that performs the mapping is the analog-to-digital converter (ADC or A-to-D). The amplitude quantizing occurs after the sample-and-hold operation. The simplest quantizer to visualize performs an instantaneous mapping from each continuous input sample level to one of the preassigned equally spaced output levels. Quantizers that exhibit equally spaced increments between possible quantized output levels are called *uniform quantizers* or *linear quantizers.* Possible instantaneous input–output characteristics are easily visualized by a simple staircase graph consisting of risers and treads of the types shown in Figure 13.3. Figure 13.3a, b, and d show quantizers with uniform quantizing steps, while Figure 13.3 c is a quantizer with nonuniform quantizing steps. Figure 13.3a depicts a quantizer with midtread at the origin, while Figure 13.3b and d

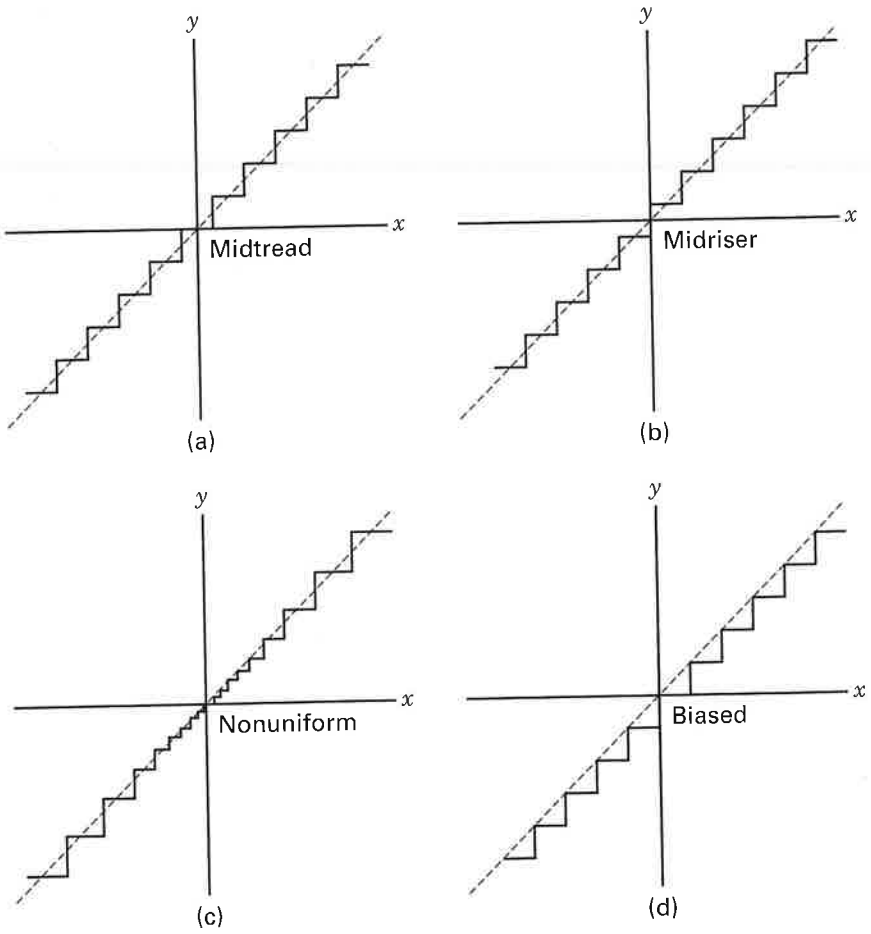


Figure 13.3 Various quantizer transfer functions.

present quantizers with midrisers at the origin. A distinguishing property of midriser and midtread converters is related to the presence or absence, respectively, of output level changes when the input to the converter is low-level idle noise. Further, Figure 13.3d presents a biased (i.e., truncation) quantizer, while the remaining quantizers in the figure are unbiased and are referred to as *rounding quantizers.* Such unbiased quantizers represent ideal models, but rounding is never implemented in A/D converters. Quantizers are typically implemented as truncation quantizers. The terms "midtread" and "midriser" are staircase terms used to describe whether the horizontal or vertical member of the staircase is at the origin. The unity-slope dashed line passing through the origin represents the ideal nonquantized input–output characteristic we are trying to approximate with the staircase. The difference between the staircase and the unity-slope line segment represents the approximation error made by the quantizer at each input level. Figure 13.4 illustrates the approximation error amplitude versus input amplitude function for each quantizer characteristic in Figure 13.3. Parts (a) through (d) of Figure 13.4 correspond to the same parts in Figure 13.3. This error is often modeled as quantizing noise because the error sequence obtained when quantizing a wideband random process is reminiscent of an additive noise sequence. Unlike true additive noise sources, however, the quantizing errors are signal dependent and are highly structured. It is desirable to break up this structure, which can be accomplished by introducing an independent noise perturbation, known as *dither,* prior to the quantization step. (This topic is discussed in Section 13.2.4.)

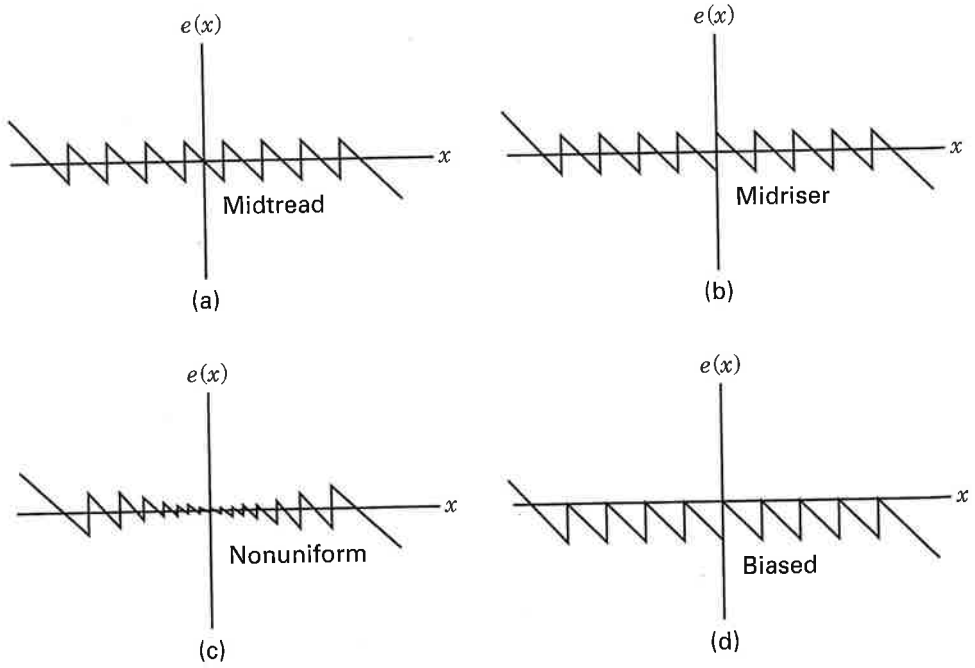


Figure 13.4 Instantaneous error for various quantizer transfer functions.

The linear quantizer is simple to implement and is particularly easy to understand. It is the universal form of the quantizer, in the sense that it makes no assumptions about the amplitude statistics and correlation properties of the input waveform, nor does it take advantage of user-related fidelity specifications. Quantizers that take advantage of these considerations are more efficient as source coders and are more task specific then the general linear quantizer; these quantizers are often more complex and more expensive, but they are justified in terms of improved system performance. There are applications for which the uniform quantizer is the most desirable amplitude quantizer. These include signal processing applications, graphics and display applications, and process control applications. There are other applications for which nonuniform adaptive quantizers are more desirable amplitude quantizers. These include waveform encoders for efficient storage and communication, contour encoders for images, vector encoders for speech, and analysis/synthesis encoders (such as the vocoder) for speech.

13.2.1 Quantizing Noise

The difference between the input and output of a quantizer is called the *quantizing error*. In Figure 13.5 we demonstrate the process of mapping the input sequence $x(t)$ to the quantized output sequence $\hat{x}(t)$. We can visualize forming $\hat{x}(t)$ by adding to each $x(t)$ an error sequence, $e(t)$:

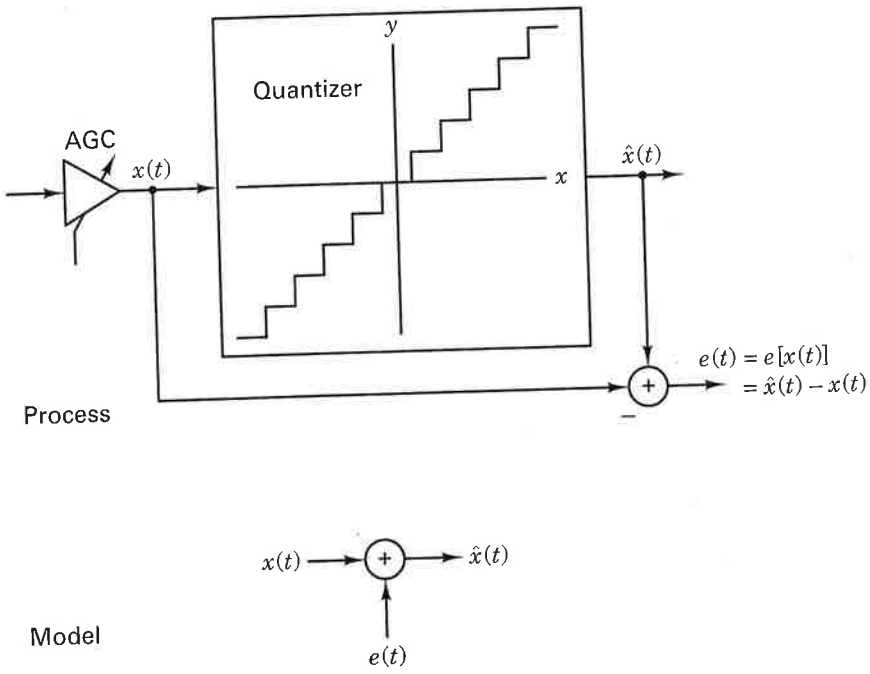


Figure 13.5 Process and model of quantizing noise corruption of input signal.

$$\hat{x}(t) = x(t) + e(t)$$

The error sequence $e(t)$ is deterministically defined by the input amplitude through the instantaneous error versus amplitude characteristic of the form in Figure 13.4. We note that the error sequence exhibits two distinct characteristics over different input operating regions.

The first operating interval is the granular error region corresponding to the input sawtooth-shaped error characteristic. Within this interval, the quantizer errors are confined by the size of the nearby staircase risers. The errors that occur in this region are called the *granular errors,* or sometimes the *quantizing errors.* The input interval for which the quantizing errors are granular defines the dynamic range of the quantizer. This interval is sometimes called the *region of linear operation.* Proper use of the quantizer requires that the input signal conditioning match the dynamic range of the input signal to the dynamic range of the quantizer. This is the function of the signal-dependent gain control system, called *automatic gain control* (AGC), indicated in the signal flow path of Figure 13.5.

The second operating interval is the nongranular error region corresponding to the linearly increasing (or decreasing) error characteristic. The errors that occur in this interval are called *saturation* or *overload* errors. When the quantizer operates in this region, we say that the quantizer is *saturated.* Saturation errors are larger than the granular errors and may have a more objectionable effect on reconstruction fidelity.

The quantization error corresponding to each value of input amplitude represents an error or noise term associated with that input amplitude. Under the assumptions that the quantization interval is small compared with the dynamic range of the input signal, and that the input signal has a smooth probability density function over the quantization interval, we can assume that the quantization errors are uniformly distributed over that interval, as illustrated in Figure 13.6. The pdf with zero mean corresponds to a rounding quantizer, while the pdf with a mean of $-q/2$ corresponds to a truncation quantizer.

A quantizer or analog-to-digital converter (ADC) is defined by the number, size, and location of its quantizing levels or step boundaries, and the corresponding step sizes. In a uniform quantizer, the step sizes are equal and are equally spaced. The number of levels N is typically a power of 2 of the form $N = 2^b$, where b is the number of bits used in the conversion process. This number of levels is equally distributed over the dynamic range of the possible input levels. Normally, this range is defined as $\pm E_{\max}$, such as ± 1.0 V or ± 5.0 V. Thus, accounting for the full range of $2E_{\max}$, the size of a quantization step is

$$q = \frac{2E_{\max}}{2^b} \tag{13.11}$$

As an example, using Equation (13.11), the quantizing step (hereafter called a *quantile.*) for a 10-bit converter operating over the ±1.0 V range is 1.953 mV. Occasionally, the operating range of a converter is altered so that the quantile is a "whole" number. For example, changing the operating range of the converter to ±1.024 V results in a quantizing step size of 2.0 mV. A useful figure of merit for the

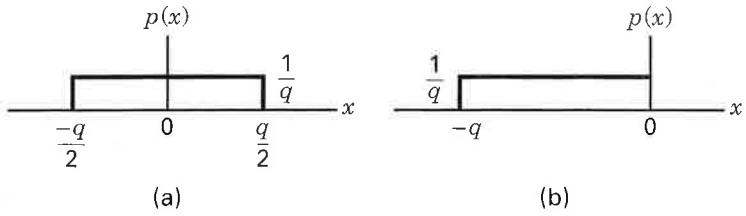


Figure 13.6 Probability density functions for quantizing error uniformly distributed over one quantile, q. (a) Probability density function for a rounding quantizer. (b) Probability density function for a truncating quantizer.

uniform quantizer is the quantizer output variance. If we assume that the quantization error is uniformly distributed over a single quantile interval q-wide, the quantizer variance (which represents the quantizer noise or error power) for the zero-mean error is found to be

$$\sigma^2 = \int_{-q/2}^{q/2} e^2 p(e)de = \int_{-q/2}^{q/2} e^2 \frac{1}{q} de = \frac{q^2}{12} \tag{13.12}$$

where $p(e) = 1/q$, over an interval of width q, is the probability density function (pdf) of the quantization error e. Thus, the rms quantizer noise in a quantile interval of width q is found to be $q\sqrt{12}$ or $0.29q$. Equation (13.12) determined the quantizing noise power over one quantile, assuming that the errors are equiprobable over the quantization interval. If we include operation in the saturation interval of a quantizer, or if we include nonuniform quantizers, we find that the quantization intervals are not of equal width over the range of the input variable, and that the amplitude density is not uniform over the quantization interval. We can account for this amplitude-dependent error power σ_q^2 by averaging the squared error over the amplitude variable weighting by the probability of that amplitude. This is expressed by

$$\sigma_q^2 = \mathbf{E}\{[x - q(x)]^2\} = \int_{-\infty}^{\infty} e^2(x)p(x)dx \tag{13.13}$$

where x is the input variable, $q(x)$ is its quantized version, $e(x) = x - q(x)$ is the error, and $p(x)$ is the pdf of the amplitude of x. We can partition the interval of integration in Equation (13.13) into two main intervals—one accounting for errors in the staircase or linear region of the quantizer, and the second accounting for errors in the saturation region. We define the saturation amplitude of the quantizer as $E_{\max}$. Also, we assume an odd symmetric transfer function for the quantizer, and a symmetric pdf for the input signal. The error power σ_q^2, defined in Equation (13.13), is the total error power, which can be partitioned as

$$\sigma_q^2 = 2\int_0^{\infty} e^2(x)p(x)dx \tag{13.14a}$$

$$= 2\int_0^{E_{\max}} e^2(x)p(x)dx + 2\int_{E_{\max}}^{\infty} e^2(x)p(x)dx \qquad (13.14b)$$
$$= \sigma^2_{\text{Lin}} + \sigma^2_{\text{Sat}}$$

where σ^2_{Lin} is the error power in the linear region and σ^2_{Sat} is the error power in the saturation region. The error power σ^2_{Lin} can be further divided into subintervals corresponding to the successive discrete quantizer output levels (i.e., quantiles). If we assume that there are N such quantile levels, the integral becomes

$$\sigma^2_{\text{Lin}} = 2\sum_{n=0}^{N/2-1}\int_{x_n}^{x_{n+1}} e^2(x)p(x)dx \qquad (13.15)$$

where x_n is a quantizer level and an interval or step size between two such levels is called a *quantile interval.* Recall that N is typically a power of 2. Thus, there are $N/2 - 1$ positive levels, $N/2 - 1$ negative levels, and a zero level, making a total of $N - 1$ levels and $N - 2$ intervals. If we now approximate the density function by a constant $q_n = (x_{n+1} - x_n)$, in each quantile interval, Equation (13.15) simplifies to

$$\sigma^2_{\text{Lin}} = 2\sum_{n=0}^{N/2-1}\frac{x^3}{3}\Bigg|_{x=-q_n/2}^{x=+q_n/2} p(x_n)$$
$$= 2\sum_{n=0}^{N/2-1}\frac{q_n^2}{12}\,p(x_n)q_n \qquad (13.16)$$

where $e(x)$ in Equation (13.15) has been replaced by x in Equation (13.16), since $e(x)$ is a linear function of x with unity slope and passes through zero at the midpoint of each interval. Also, the limits of integration in Equation (13.15) have been replaced by the change in x over a quantile interval. Since the change has been denoted as q_n, the lower and upper limits can be designated as $x = -q_n/2$ and $x = +q_n/2$, respectively. Equation (13.16) describes the error power in the linear region as a summation of error power $q_n^2/12$ in each quantile interval weighted by the probability $p(x_n)q_n$ of that error power.

13.2.2 Uniform Quantizing

If the quantizer has uniform quantiles equal to q and all intervals are equally likely, Equation (13.16) simplifies further to

$$\sigma^2_{\text{Lin}} = \frac{2}{12}\sum_{n=0}^{N/2-1} q_n^2\,p(x_n)q_n = \frac{2}{12}\sum_{n=0}^{N/2-1} q^2\frac{1}{q(N-2)}q = \frac{q^2}{12} \qquad (13.17)$$

If the quantizer does not operate in the saturation region (the quantization noise power), then $\sigma_q^2 = \sigma^2_{\text{Lin}}$; and these terms are often used interchangeably. Noise power alone will not fully describe the noise performance of the quantizer. A more meaningful measure of quality is the ratio of the second central moment (variance) of the quantizing noise and the input signal. Assuming that the input signal has zero mean, the signal variance is

$$\sigma_x^2 = \int_{-\infty}^{\infty} x^2\, p(x)dx \tag{13.18}$$

Further insight into the average quantizer noise requires that we examine a specific density function and a specific quantizer.

Example 13.4 Uniform Quantizer

Determine the quantizer variance and the quantizer noise-to-signal power ratio (NSR) for a signal that is uniformly distributed over the full dynamic range of a uniform quantizer with 2^b equally spaced quantile levels. In this case there is no saturation noise and only the linear noise term must be computed. Each quantile interval is

$$q = (2E_{\max})\,2^{-b} \tag{13.19}$$

where $2E_{\max}$ is the input interval between the positive and negative boundaries of the linear quantizing range.

Solution

Substituting Equation (13.19) into Equation (13.12) or (13.17), we have the following quantizing noise power (in the linear region):

$$\sigma_q^2 = \frac{1}{12}(2E_{\max}2^{-b})^2 = \frac{1}{12}(2E_{\max})^2\,2^{-2b} \tag{13.20}$$

The input signal power is found by performing the integration of Equation (13.18) for a uniform probability density function in the zero-mean interval spanning $2E_{\max}$, so that $p(x) = 1/(2E_{\max})$, and the signal variance is found to be

$$\sigma_x^2 = \int_{-E_{\max}}^{+E_{\max}} \frac{1}{2E_{\max}} x^2\, dx = \frac{1}{12}(2E_{\max})^2 \tag{13.21}$$

Taking the ratio of noise power to signal power (NSR), we have

$$\text{NSR} = \frac{\sigma_q^2}{\sigma_x^2} = 2^{-2b} \tag{13.22}$$

Now, converting the NSR to decibels, we have

$$\text{NSR}_{\text{dB}} = 10\log_{10}(\text{NSR}) = 10\log_{10}(2^{-2b}) \tag{13.23a}$$

$$= -20b\log_{10}(2) = -6.02b\ (\text{dB}) \tag{13.23b}$$

Equation (13.23b) suggests that each bit used in the conversion process is worth –6.02 dB in noise-to-signal ratio. In fact, the NSR for any uniform quantizer not operating in saturation is of the form

$$\text{NSR}_{\text{dB}} = -6.02b + C \tag{13.24}$$

where the term C depends on the signal probability density function (pdf); it is positive for density functions that are narrow with respect to the saturation level of the quantizer.

13.2.2.1 Signal and Quantization Noise in the Frequency Domain

Up until now we have discussed quantization noise in terms of its effect on samples of the time series representing the sampled signal. The quantization noise can also be described in the frequency domain, which offers insight into effects of operating conditions that we will examine shortly. These include saturation (Section 13.2.3), dithering (Section 13.2.4), and noise feedback quantizers (Section 13.2.6).

Figure 13.7 presents the discrete Fourier transform of two sinusoids that have been sampled by a linear 10-bit ADC. The two sinusoids have relative amplitudes of 1.0 and 0.01 (i.e., one is 40 dB below the other). In Figure 13.7a the lower-frequency signal (labeled 0-dB) is scaled to 1 dB below full dynamic range of the 10-bit converter, which for convenience we set to be unity. Note that in Figure 13.7a, the full-scale 0-dB signal dwells an additional 6-dB below the input attenuation level of 1-dB. This is due to the factor of one-half in the spectral decomposition of a real signal over all nonzero frequencies. The average signal-to-quantization noise ratio (SNR) for a 10-bit quantizer is $60 + C$ dB. For a full-scale sinusoid, the constant C is 1.76 dB, resulting in a SNR of approximately 62 dB. The discrete Fourier transforms (DFTs) performed for Figure 13.7 were of length 256, and since the SNR of a transform increases in proportion to transform length (or integration time), there is an improvement in SNR of 24 dB due to the transform [2],

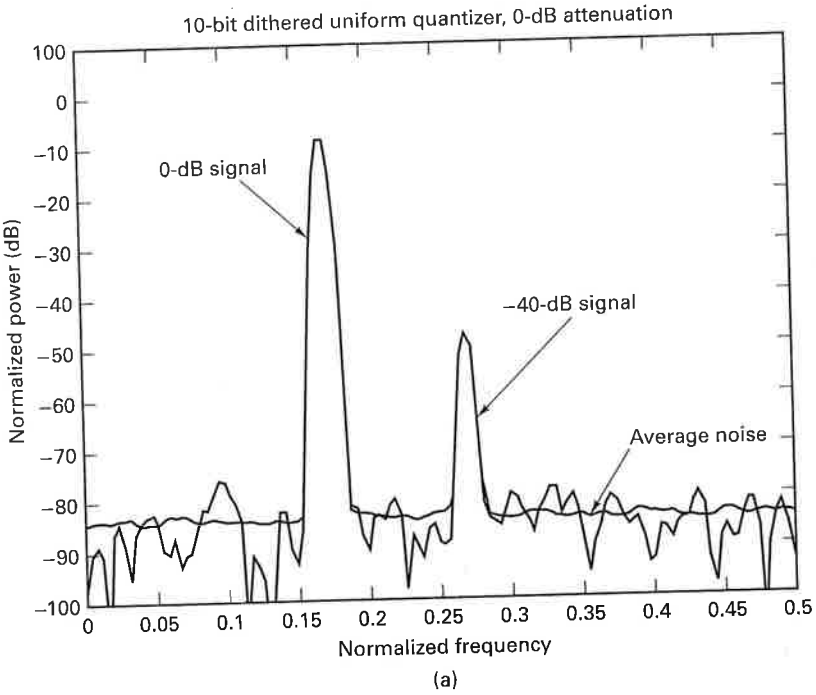
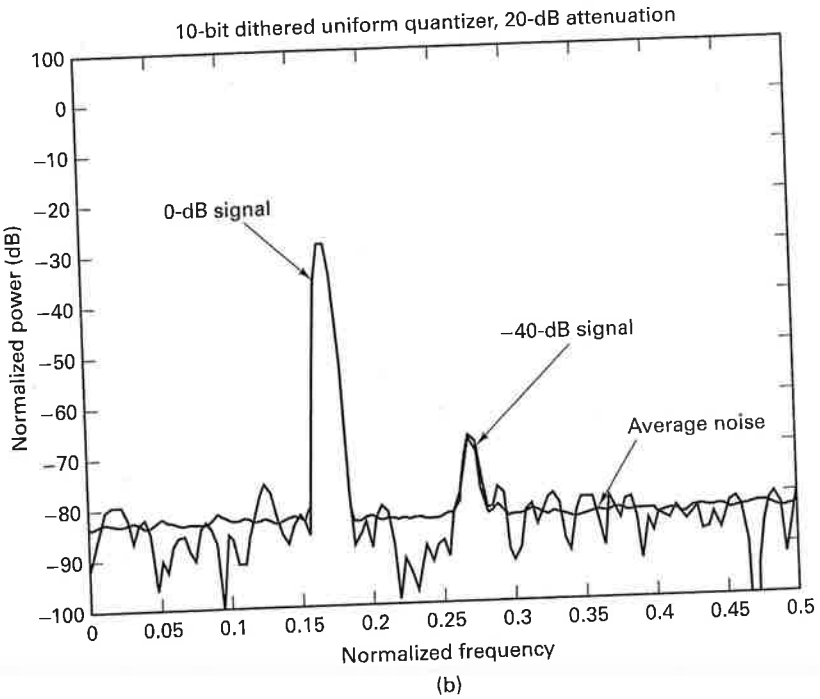
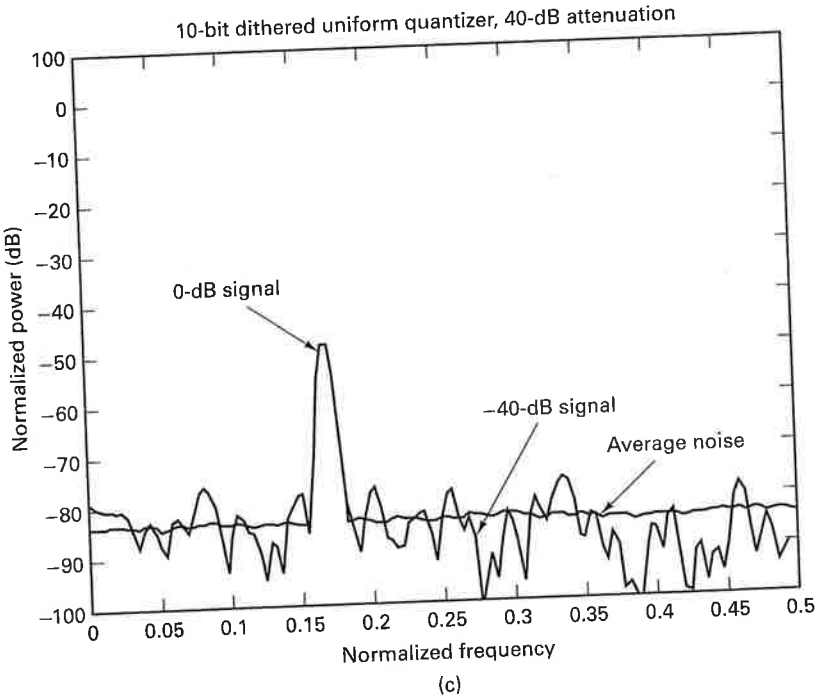


Figure 13.7 Power spectrum of signals quantized by uniform ADC.



(b)



(c)

Figure 13.7 Continued

with a 3.0 dB loss due to windowing. Thus, at the output of the transform, the peak SNR due to quantizing is 62 + 24 − 3 = 83 dB. The noise signal, at each frequency of a DFT, can be represented as the square root of a sum of squared Gaussian random variables, which is characterized as a chi-square two-degrees-of-freedom process. The variance (noise power) is equal to the squared mean. Thus, there is significant variation about the expected noise power level. An ensemble average is required to obtain a stable estimate of the noise floor. The noise floor shown in figure 13.7, obtained from 400 averages, is seen to be –83 dB. A noise dither signal (described in Section 13.2.4), has been added to the signal prior to the quantization in order to randomize the quantization errors. In Figure 13.7b and c, the input signals are attenuated relative to the full-scale input, by 20 and 40 dB, respectively. This attenuation increases the constant C in Equation (13.24) by 20 and 40 dB, which we observe as reductions in the spectral levels of the input sinusoids by these same values. Note that the higher frequency input signal in Figure 13.7c, now attenuated 80 dB relative to full scale, is now 3-dB below the average noise level of the converter. The lower frequency sinusoid in Figure 13.7c is now attenuated 40 dB relative to full scale, and therefore exhibits an SNR that is 40 dB lower than the same signal of Figure 13.7a.

Given the task of minimizing the average quantizing noise-to-signal ratio in a quantizer, we are faced with a conflict of requirements. On one hand, we wish to keep the signals large with respect to the quantizing interval q, in order to achieve a large SNR. But we also find it necessary to keep the signal small to avoid saturating the quantizer. We resolve the opposing requirements by scaling the input signal so that its rms value is a specified fraction of the full-scale quantizer range. The specified fraction is chosen to balance the saturation errors (weighted by their probability of occurrence) against the quantizing errors (which are similarly weighted) and thus achieve a minimum noise-to-signal ratio. We discuss the location of this desired operating point for the converter in the next section.

13.2.3 Saturation

Figure 13.8 presents the average NSR of a uniform quantizer as a function of the ratio of quantizer saturation level to the rms signal value. The figure shows the NSR of signals with three different pdfs: arc-sine (sine wave density), uniform, and Gaussian.

In Figure 13.8, the abscissa is the ratio of the quantizer saturation level to the rms level of the input signal. For each of the three density functions shown, there is a value on the abscissa corresponding to a minimum NSR for a fixed number of bits; in other words, for a given input density function, we can determine the input signal level (relative to saturation) to achieve the minimum NSR. Reduced input signal levels correspond to larger NSR values on the abscissa, and represent movement to the right. Similarly, increased input signal levels also correspond to larger NSR values on the abscissa, and represent movement to the left. This increase is due to operating in the saturation region of the quantizer. We note that the rate of change of NSR as we move to the left of the optimum operating point is higher

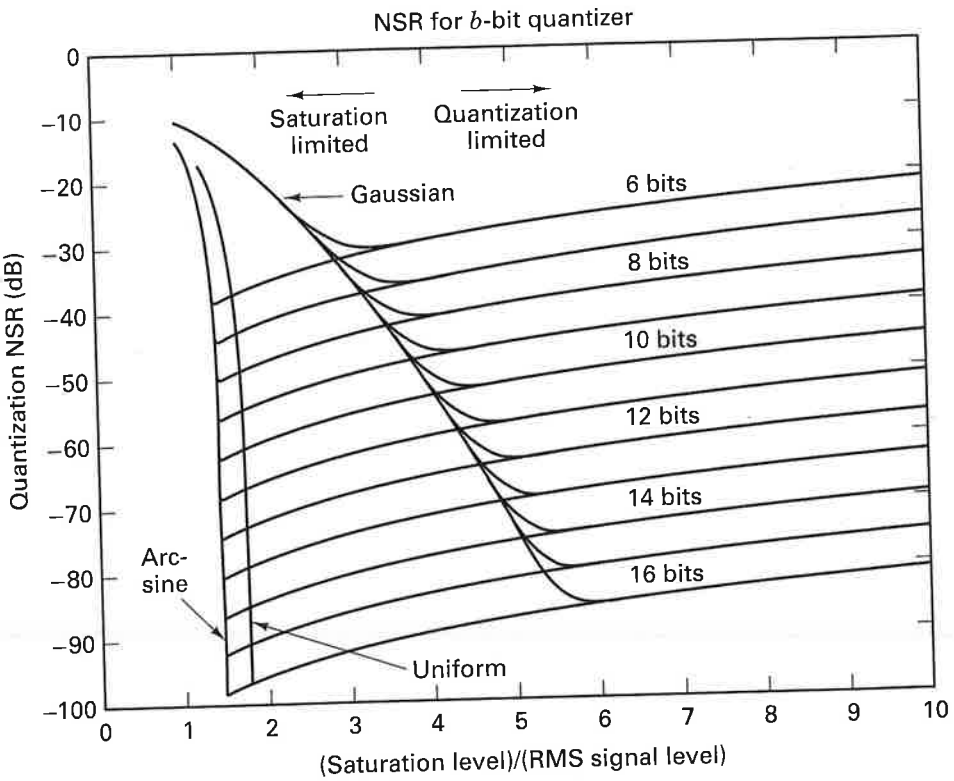


Figure 13.8 ADC NSR versus ratio of ADC saturation level to RMS signal level.

than that if we move to the right. This is particularly true, for instance, in the case of the arc-sine and uniform densities. This suggests that saturation noise is more objectionable that linear quantization noise. A consequence of this is that if we err in setting the operating point, called the *attack point* of a quantizer, it would be better to have the error on the side of excess attenuation rather than insufficient attenuation of the input signal. The onset of saturation occurs at different values of the abscissa. For sine waves (arc-sine density) it occurs at approximately $\sqrt{2}$. For triangle waves, (uniform density) it occurs at approximately $\sqrt{3}$. For noise-like signals, (Gaussian density) it occurs continuously, with decreasing probability as the signal level is reduced relative to saturation. As an example, a 10-bit converter has a minimum NSR of –60 dB for a uniform density when operating at the edge of saturation, and an NSR of – 62 dB for an arc-sine density, when similarly operating at the edge of saturation. On the other hand, the same 10-bit converter has a minimum NSR of approximately – 52 dB for all densities, when the rms level is set at $\frac{1}{4}$ of saturation (value of 4 on the abscissa). The figure dramatically demonstrates that saturation noise is more severe than is quantizing noise. This can be explained fairly simply by examining the instantaneous error characteristic, (as shown in Figure

13.4,) and noting that saturation errors are very large relative to the quantizing errors. Thus, a small amount of saturation, even if it occurs infrequently, will make a large contribution to the average noise level of the quantizer.

Saturation noise and quantization noise differ in another important way. Quantization noise tends to be white noise. Dither signals may be intentionally added to the analog signal prior to the quantizer to assure this property. Saturation noise, on the other hand, tends to be white only when the input signal has a broad bandwidth and tends to be harmonically related to the input signal if it has a narrow bandwidth, Thus, the effects of quantizing noise can be filtered or averaged because it has the characteristics of white noise. Saturation noise, on the other hand, is indistinguishable from signal content and generally cannot be reduced by subsequent averaging or filtering techniques.

Figure 13.9 presents the discrete Fourier transforms of the same signal set presented in Figure 13.7 quantized with a 10-bit ADC, except that in Figure 13.9 the peak signal amplitude was adjusted to be 10% (0.83 dB) above the ADC saturation level. Note the very many spectral artifacts caused by the saturation. These artifacts (the saturation noise) will grow larger as the signal excursions go deeper

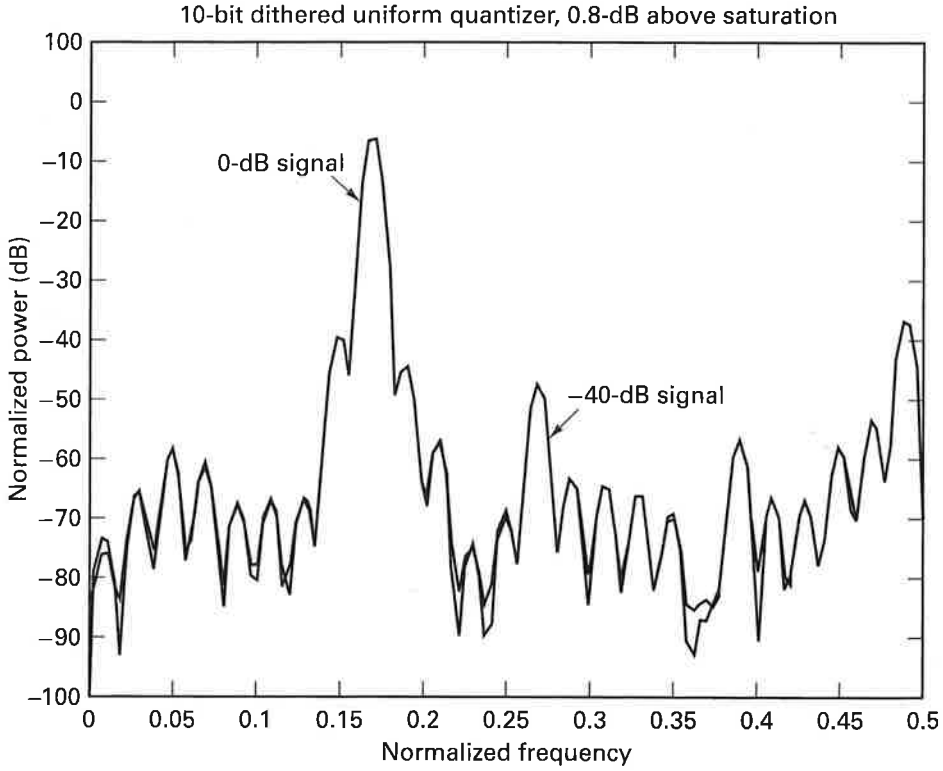


Figure 13.9 Power spectrum of uniformly quantized signals, with the quantizer saturating on signal peaks 0.8 dB beyond full-scale input level.

into saturation. Also note that some of these artifacts are larger than the –40 dB signal. Compare this figure with Figure 13.7 to see the dramatic difference that too little signal attenuation (hence, saturation) makes in the noise output of an ADC.

13.2.4 Dithering

Dithering is one of the cleverest applications of noise as a useful engineering tool. A dither signal is a small perturbation or disturbance added to a measurement process to reduce the effect of small local nonlinearities. The most familiar form of dither is the slight tapping we apply to the side of a d'Arsonval meter movement prior to taking the reading (before the days of digital meters). The tapping is a sequence of little impulses for displacing the needle movement beyond the local region, which exhibits a nonlinear coefficient of friction at low velocities. A more sophisticated example of this same effect is the mechanical dither applied to the counter-rotating laser beams of a laser beam gyro to break up low-level frequency entrapment, known as a *dead band* [3].

In the analog-to-digital converter application, the effect of the dither is to reduce or eliminate the local discontinuities (i.e., the risers and treads) of the instantaneous input–output transfer function. We can best visualize the effect of these discontinuities by listing the desired properties of the error sequence formed by the quantizer process and then examining the actual properties of the same sequence. The quantizer error sequence is modeled as additive noise. The desired properties of such a noise sequence $e(n)$ are as follows:

1. Zero mean: $\mathbf{E}\{e(n)\} = 0$
2. White: $\mathbf{E}\{e(n)e(n+m)\} = \sigma^2\,\delta(m)$
3. Uncorrelated with data $x(n)$: $\mathbf{E}\{e(n)\,x(n+m)\} = 0$

where n and m are sample indices, and $\delta(m)$ is a Dirac delta function. In Figure 13.10, we examine a sequence of samples formed by a truncating ADC and make the following observations:

1. The error sequence is all of the same polarity; therefore, it is not zero mean.
2. The error sequence is not independent, sample-to-sample; therefore, it is not white.
3. The error sequence is correlated with the input; therefore, it is not independent.

Repeated measurements of the same signal would result in the same noise, and thus no amount of averaging could reduce the deviation from the true input signal. Paradoxically, we would like this noise to be "noisier." If the noise were independent on successive measurements, averaging would reduce the deviation from the true values. Thus, faced with the problem that the noise we get is not the noise we want, we choose to alter that noise by adding our own. We add a perturbation to the measurement to override the undesired low-level structure of the quantizer noise. The added perturbation, in a sense, converts *bad noise to good noise* [4].

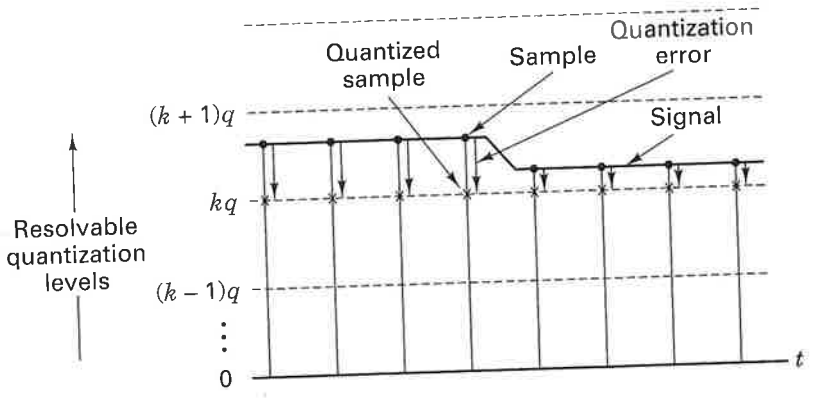


Figure 13.10 Sampled data sequence quantized to next-lowest quantile level by associated error sequence.

Example 13.5 Dither Linearization

We hypothesize a quantizer that can only measure integers and converts input data into the next lowest integer—a process called *truncation*. We make 10 measurements of a signal of, say, amplitude 3.7. In the absence of a dither we have readings all equal to 3.0. Now add a uniformly distributed (over 0 to 1) random number sequence to the input prior to performing the reading. The sequence of data has the following form:

Reading	Raw Signal	Quantized Raw Signal	Dither	Dithered Signal	Dith + Quant Signal
1	3.7	3.0	0.3485	4.0485	4.0
2	3.7	3.0	0.8685	4.5685	4.0
3	3.7	3.0	0.2789	3.9789	3.0
4	3.7	3.0	0.3615	4.0615	4.0
5	3.7	3.0	0.1074	3.8074	3.0
6	3.7	3.0	0.2629	3.9629	3.0
7	3.7	3.0	0.9252	4.6252	4.0
8	3.7	3.0	0.5599	4.2599	4.0
9	3.7	3.0	0.3408	4.0408	4.0
10	3.7	3.0	0.5228	4.2228	4.0
Means =		3.0	0.4576	4.1576	3.7
Dither mean =				0.4576	
Dithered Signal Mean – Dither mean =				3.7	

In this example, a biased dither was used to remove the quantizer bias. The average of the dithered and quantized readings if (for this example) a correct reading and, in general, will be closer to the true signal than will the nondithered and quantized measurements [5, 6].

To illustrate the effect that dithering has on the quantization process of a time-varying signal, consider the following experiment. Let us apply 60 dB of attenuation to a sinusoidal signal of amplitude 1.0. The attenuated signal, then, has full-scale amplitude of 0.001, which is approximately one half of the quantization in-

terval of 0.001957 for a ten-bit uniform quantizer (obtained by dividing the peak-to-peak signal amplitude of 2 by $2^{10} - 2$). When the attenuated sinusoid is applied to a rounding quantizer, the output will be essentially all zeros except for an occasional count of ±1 quantile, which occurs when the input crosses the $\pm q/2$ level of 0.000979 (corresponding to the least significant bit of the ADC). If the input signal were attenuated another 0.23 dB, the threshold levels of the least significant bit would never be crossed and the output sequence would be all zeros. Now, let us add a dither signal with rms amplitude of 0.001 to the attenuated sinusoid of 0.001, so that the signal plus dither regularly crosses the $\pm q/2$ levels of the ADC. Figure 13.11 shows the power spectra obtained by transforming and averaging 400 realizations of this dithered signal. Lo and behold, the 60-dB attenuated signal, at the edge of the ADC's resolvability, is indeed present and has been accurately measured at –63 dB (–3 dB is due to the windowing). The dither signal has had the effect of extending the dynamic range of the ADC (typically by 9 to 12 dB or 1.5 to 2.0 bits) and has improved the effective linearity of the low-level ADC staircase approximation.

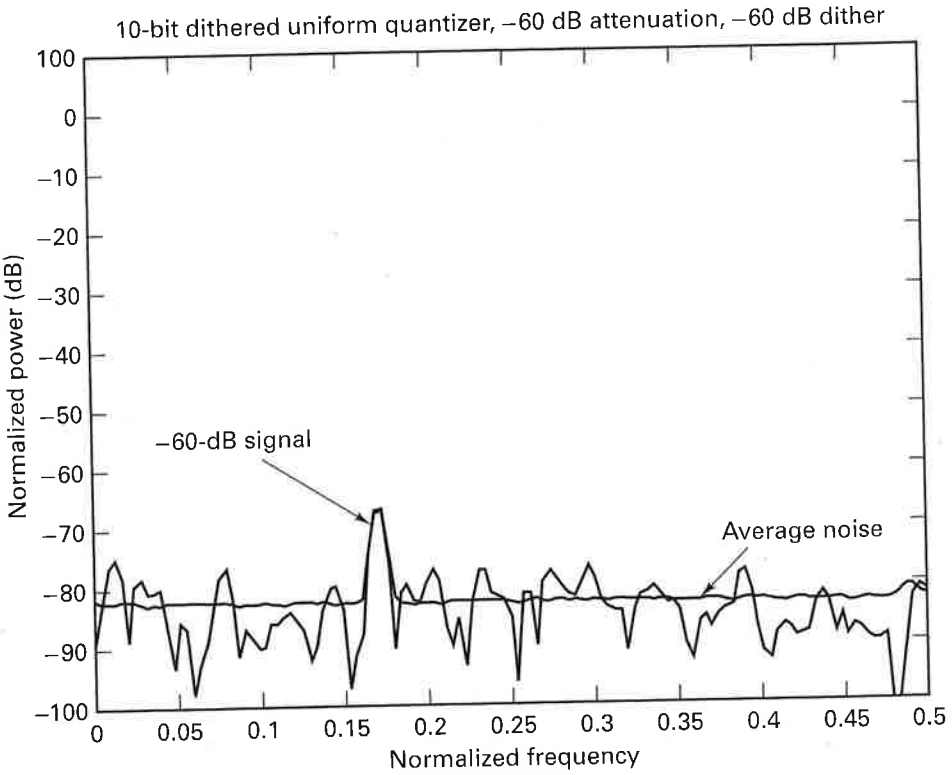


Figure13.11 Power spectrum of uniformly quantized dithered low level signal.

13.2.5 Nonuniform Quantizing

Uniform quantizers are the most common type of analog-to-digital converters because they are the most robust. By "robust" we mean that they are relatively insensitive to small changes in the input statistics. They achieve this robustness by not being finely tuned to one specific set of input parameters. This allows them to perform well even in the face of uncertain input parameters, and it means that small changes in input statistics will result in only small changes in output statistics.

When there is small uncertainty in the input signal statistics, it is possible to design a nonuniform quantizer that exhibits a smaller quantizer NSR than a uniform quantizer using the same number of bits. This is accomplished by partitioning the input dynamic range into nonuniform intervals such that the noise power, weighted by the probability of occurrence in each interval, is the same. Iterative solutions for the decision boundaries and step sizes for an optimal quantizer can be found for specific density functions and for a small number of bits. This task is simplified by modeling the nonuniform quantizer as a sequence of operators, as depicted in Figure 13.12. The input signal is first mapped, via a nonlinear function called a *compressor,* to an alternative range of levels. These levels are uniformly quantized and the quantized signal levels are then mapped, via a complementary nonlinear function called an *expander,* to the output range of levels. Borrowing part of the name from each of the operations COMpress and exPAND, we form the acronym by which this process is commonly identified: *companding.*

13.2.5.1 (Near) Optimal Nonuniform Quantizing

Examining the compressor characteristics $y = C(x)$ of Figure 13.13, we note that the quantizing step sizes for the output variable y are related to the quantizing step sizes for the input variable x through the slope $\dot{C}(x)$ [e.g., $\Delta y = \Delta x\, \dot{C}(x)$]. For an arbitrary pdf and arbitrary compressor characteristics, we can arrive at the output quantizing noise variance [7]

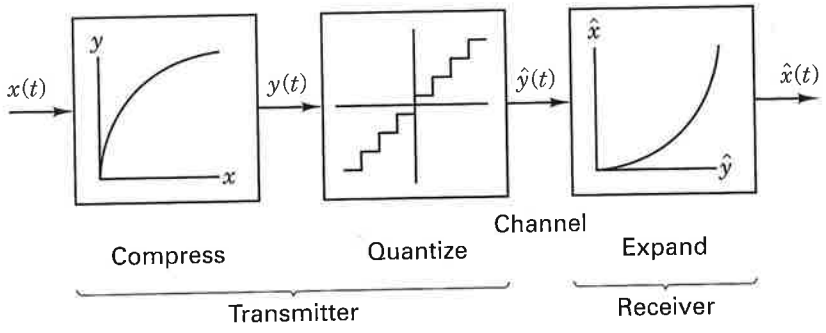


Figure 13.12 Nonuniform quantizer as a sequence of operators: compression, uniform quantization, and expansion.

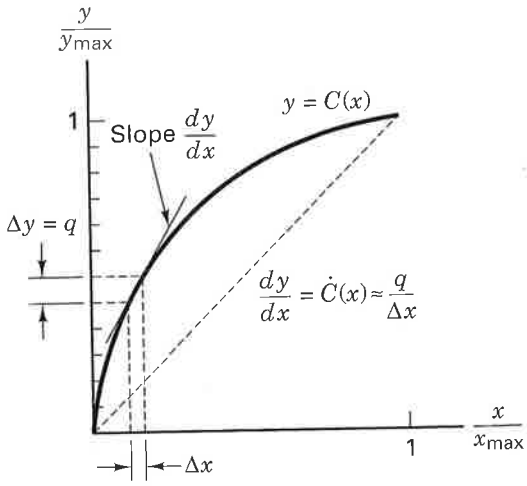


Figure 13.13 Compressor characteristics $C(x)$ and estimate to local slope $\dot{C}(x)$.

$$\sigma_q^2 = \frac{q^2}{12} \int_{-x_{max}}^{x_{max}} \frac{p(x)}{|\dot{C}(x)|^2} dx \tag{13.25}$$

for a specific pdf, the compression characteristics $C(x)$ can be found which minimize σ_q^2. The optimal compressor law for a given pdf is [8]

$$C(x) = \int_0^x \sqrt[3]{Kp(z)}\, dz \tag{13.26}$$

We find that the optimal compressor characteristic is proportional to the integral of the cube root of the input probability density function. This is called *fine tuning*. If the compressor is designed to operate with one density function and it is used with some other density function (including scaled versions), the quantizer is said to be mismatched and there may be severe performance degradation due to the mismatch [6].

13.2.5.2 Logarithmic Compression

In the preceding section, we presented the compression law for the case in which the input signal's pdf is well defined. We now address the case for which little is known about the signal's pdf. This case occurs, for instance, when the average power of the input signal is a random variable. As an example, the voice level of a randomly chosen telephone user may vary from one extreme of a barely audible whisper to the other extreme of a bellowing shout.

For the case of an unknown pdf, the compressor characteristics of the nonuniform quantizer must be selected such that the resultant noise performance is independent of the specific density function. Although this is a worthy undertaking, it may not be possible to achieve this independence. We are willing to compromise, however, and we will settle for virtual independence over a large range of input variance and input density functions. An example of a quantizer that exhibits a

SNR independent of the input signal's pdf can be visualized with the aid of Figure 2.18. There we can see a very large difference in NSR ratio for different amplitude input signals when quantized with a uniform quantizer. By comparison, we can see that the nonuniform quantizer only permits large errors for large signals. This makes intuitive sense. If the SNR is to be independent of the amplitude distribution, the quantizing noise must be proportional to the input level. Equation (13.25) presented the quantizer noise variance for an arbitrary pdf and arbitrary compressor characteristics. The signal variance for any pdf is

$$\sigma_x^2 = \int_{-\infty}^{\infty} x^2 p(x) dx \tag{13.27}$$

In the absence of saturation, the quantizer SNR is of the form

$$\frac{\sigma_x^2}{\sigma_q^2} = \frac{\displaystyle\int_{-x_{\max}}^{x_{\max}} x^2 p(x)\, dx}{(q^2/12)\displaystyle\int_{-x_{\max}}^{x_{\max}} [p(x)/\dot{C}^2(x)]\, dx} \tag{13.28}$$

To have the SNR be independent of the specific density function, we require that the numerator be a scaled version of the denominator. This happens if the following is true:

$$[\dot{C}(x)]^2 = \left(\frac{K}{x}\right)^2 \tag{13.29}$$

or

$$\dot{C}(x) = \frac{K}{x} \tag{13.30}$$

from which we obtain by integration,

$$C(x) = \int_0^x \frac{K}{z} dz \tag{13.31}$$

or

$$C(x) = \log_e(x) + \text{constant} \tag{13.32}$$

This result is intuitively appealing. A *logarithmic compressor* allows for a *constant* SNR output, because with a logarithmic scale equal distances (or errors) are, in-fact, equal ratios, which is what we require in order for the SNR to remain fixed over the input signal range. In Equation (13.32), the constant is present to match the boundary conditions between $x_{\max}$ and $y_{\max}$. Accounting for this boundary condition, we have the logarithmic converter of the form

$$\frac{y}{y_{\max}} = \frac{C(x)}{y_{\max}} = \log_e\left(\frac{x}{x_{\max}}\right) \tag{13.33}$$

The form of the compression suggested by the logarithm function is shown in Figure 13.14a. The first difficulty with this function is that it does not map the negative input signals. We account for the negative signals by adding a reflected version of the log to the negative axis. This modification results in Figure 13.14 yielding

$$\frac{y}{y_{\max}} = \log_e\left(\frac{|x|}{x_{\max}}\right)\text{sgn}(x) \tag{13.34}$$

where

$$\text{sgn } x = \begin{cases} +1 \text{ for } x \geq 0 \\ -1 \text{ for } x < 0 \end{cases}$$

The remaining difficulty we face is that the compression described by Equation (13.34) is not continuous through the origin; in fact, if completely misses the origin. We need to make a smooth transition between the logarithmic function and a linear segment passing through the origin. There are two standard compression functions that perform this transition—the μ-law and A-law companders.

μ-Law Compander. The μ-law compander, introduced by the Bell System for use in North America, is of the form

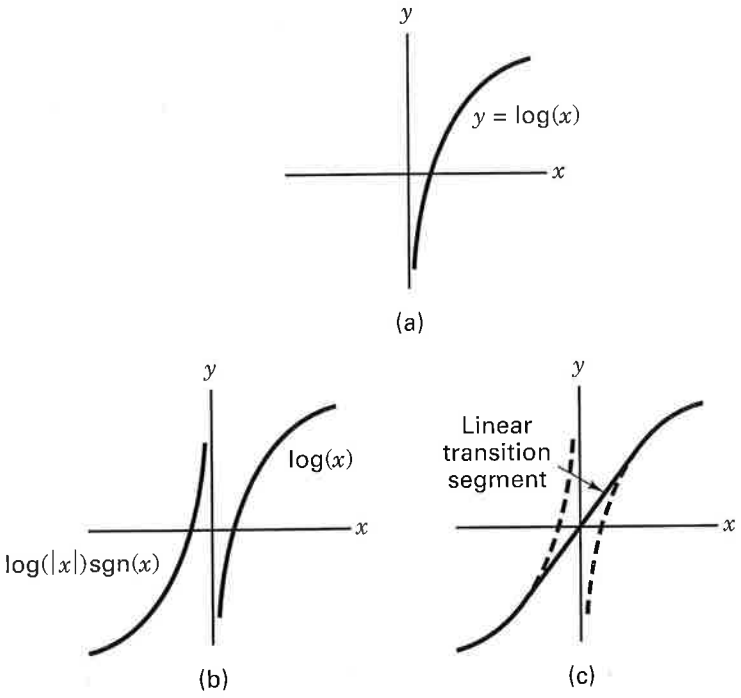


Figure 13.14 (a) Log function prototype for compression law. (b) Log | x | sgn x function prototype for compression law. (c) Log | x | sgn x function with a smooth transition between segments.

$$y = C(x) = y_{\max} \frac{\log_e[1 + \mu(|x|/x_{\max})]}{\log_e(1 + \mu)} \operatorname{sgn} x \tag{13.35}$$

The approximate behavior of this compressor in the regions corresponding to small and large values of the argument are

$$y = C(x) = \begin{cases} y_{\max} \dfrac{\mu(|x|/x_{\max})}{\log_e(\mu)} & \text{for } \mu\dfrac{|x|}{x_{\max}} \ll 1 \\ y_{\max} \dfrac{\log_e[\mu(|x|/x_{\max})]}{\log_e(\mu)} & \text{for } \mu\dfrac{|x|}{x_{\max}} \gg 1 \end{cases} \tag{13.36}$$

The parameter μ in the μ-law compander had originally been set to 100 for use with a 7-bit converter. It was later changed to 255 for use with an 8-bit converter. The 8-bit $\mu = 255$ μ-law converter has become the standard North American conversion law.

Example 13.6 Average SNR for μ-Law Compressor

The SNR for the μ-law compressor can be estimated by substituting the μ-law expression into Equation (13.28). For positive values of the input variable x, the compression law is

$$y = C(x) = y_{\max} \frac{\log_e[1 + \mu(|x|/x_{\max})]}{\log_e(1 + \mu)} \tag{13.37}$$

Then the derivative is

$$\dot{y} = \dot{C}(x) = y_{\max} \frac{1}{\log_e(1 + \mu)} \frac{\mu(1/x_{\max})}{1 + \mu(|x|/x_{\max})} \tag{13.38}$$

For values of the input variable for which $\mu(x/x_{\max})$ is large compared with unity, the derivative becomes

$$\dot{y} = \dot{C}(x) \approx \frac{1}{x} \frac{y_{\max}}{\log_e(\mu)} \tag{13.39}$$

Substituting for $1/\dot{C}(x)$ in Equation (13.28), we find that

$$\text{SNR} = \frac{\sigma_s^2}{\sigma_q^2} = \frac{1}{(q^2/12)[\log_e(\mu)/y_{\max}]^2} \tag{13.40}$$

$$= 3\left(\frac{2y_{\max}}{q}\right)^2 \left(\frac{1}{\log_e(\mu)}\right)^2 \tag{13.41}$$

The ratio $2y_{\max}/q$ is approximately equal to the number of quantizing levels (2^b) of the b-bit compressed quantizer. For the 8-bit converter with $\mu = 255$,

$$\text{SNR} = 3\left[\frac{2^8}{\log_e(255)}\right]^2 = 3(46.166)^2 = 38.1 \text{ dB} \tag{13.42}$$

For a comparison, the SNR of a μ-law quantizer is presented in Figure 13.15. Here, the SNR is plotted for input sinusoids of different amplitudes. Shown on the same fig-

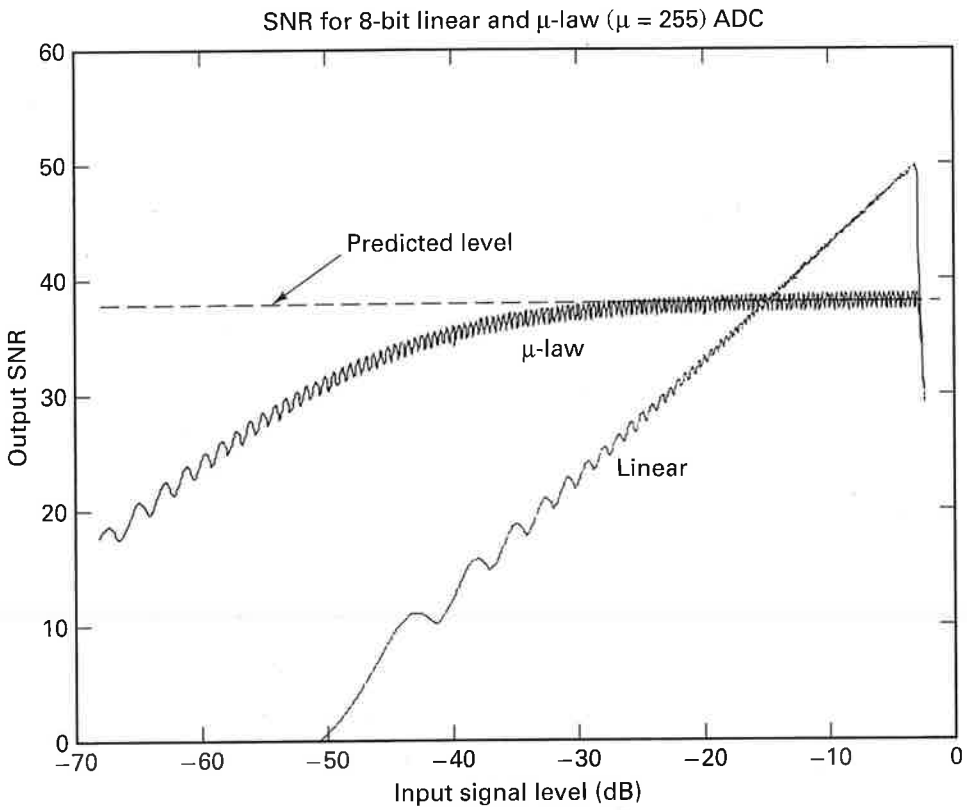
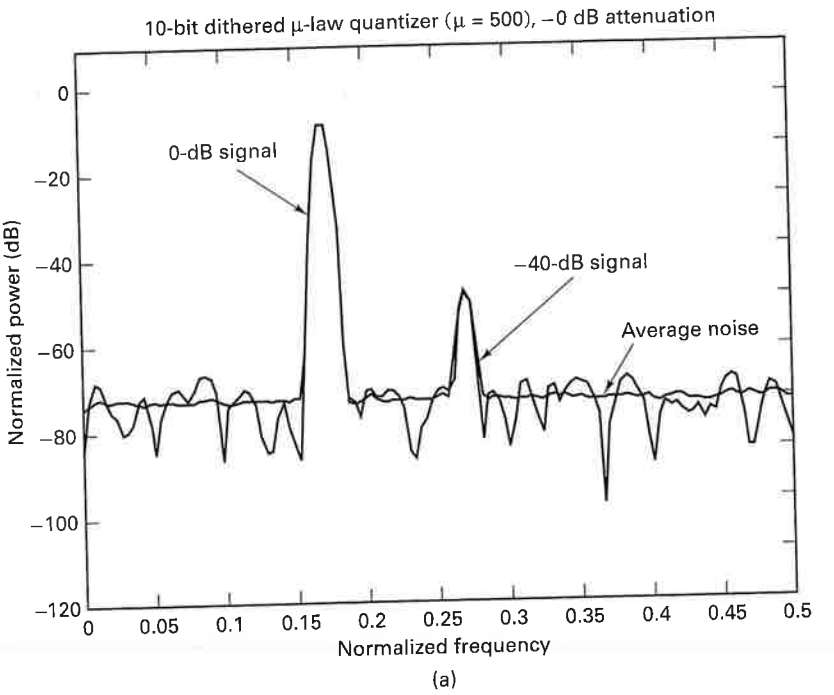


Figure 13.15 Predicted and measured SNR for a μ-law ADC.

ure is the 38.1 dB level computed in equation 13.42, and the SNR for a linear quantizer over the same range of input amplitudes. As predicted, the μ-law quantizer maintains a constant SNR over a significant range of input levels. The serration of the performance curve (granularity of the quantizer) is due to the log compression function. The actual converters exhibit additional serration due to piecewise linear approximation to the continuous μ-law curve. This is described shortly.

Figure 13.16 presents the discrete Fourier transform of the pair of input sinusoids of relative amplitude 1.0 (0-dB) and 0.01 (−40-dB). The input signal is quantized with a 10 bit μ-law (μ = 500) converter and in Figures 13.16a, b, and c, the signal levels are attenuated by 0, 20, and 40 dB, respectively, relative to full-scale input. Note that the quantizing noise levels for the full-scale signal in Figure 13.16a are higher than that of the uniform quantizer ADC, −72 dB as opposed to −83 dB seen in Figure 13.7. The improved SNR performance of the log-compressed ADC compared with the uniform ADC is to be seen for the attenuated signals. We see that as the input signal levels are attenuated, the quantizing noise is also reduced and that for 40 dB attenuation, the noise level drops to −108 dB. Thus, the log compressed ADC has no problem "seeing" the low-level input signal even with 40-dB attenuation in Figure 13.16c, while the same signal is lost in the noise of the uniform converter as seen in Figure 13.7c.



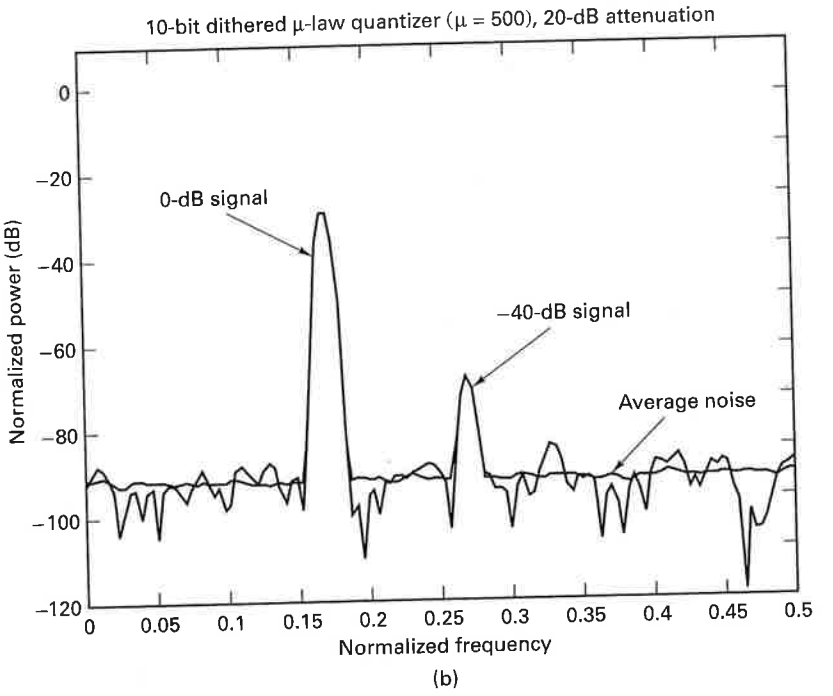


Figure 13.16 Power spectrum of signals quantized by μ-law ADC.

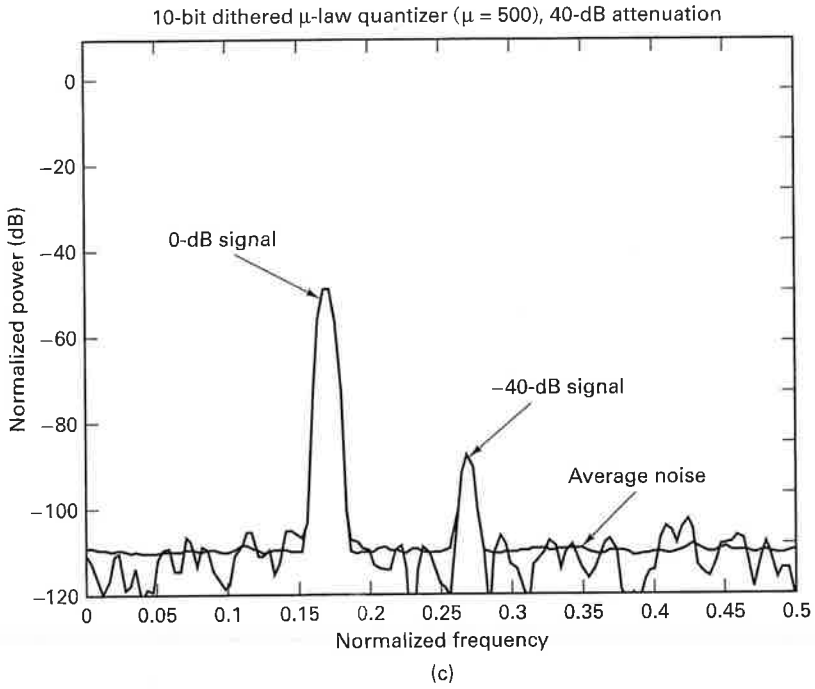


Figure 13.16 Continued

The μ-law compressor realization differs from the expression presented in Equation (13.35) in a minor way. As shown in Figure 13.17, 16 linear chord segments approximate the functional expression over the possible 256 output levels. Eight of these segments are in the first quadrant, eight are in the third quadrant, and the "0" segment has the same slope in both quadrants. Over each chord segment, the quantization is uniform in the four lower-order conversion bits. Thus the 8-bit compressed conversion format is of the form

$$\underbrace{b_7}_{\text{sign bit}} \quad \underbrace{b_6\, b_5\, b_4}_{\text{segment}} \quad \underbrace{b_3\, b_2\, b_1\, b_0}_{\text{position on segment}}$$

It is the piecewise chord approximation to the smooth function and a staircase approximation of each chord that accounts for additional serrations in the SNR curve presented in Figure 13.15.

A-Law Compander. The *A*-law compander is the CCITT (hence the European) standard approximation to the logarithmic compression. The form of the compressor is

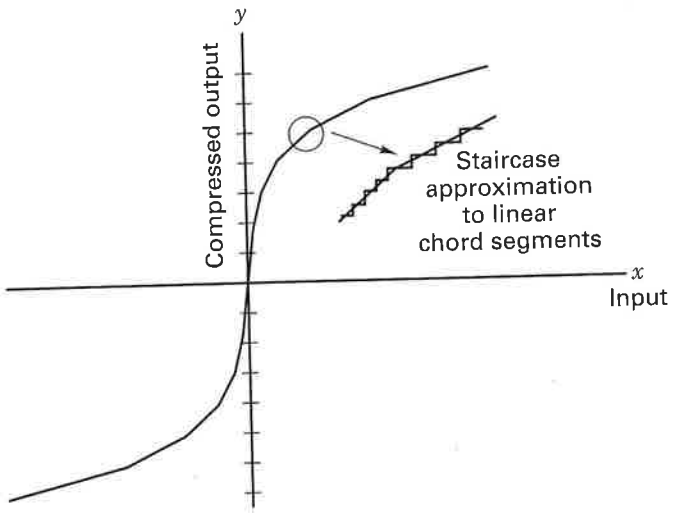


Figure 13.17 Seven-bit compressed quantization with 16-segment approximation to μ-law.

$$y = C(x) = \begin{cases} y_{\max} \dfrac{A(|x|/x_{\max})}{1 + \log_e(A)} \operatorname{sgn} x & \text{for } 0 < \dfrac{|x|}{x_{\max}} < \dfrac{1}{A} \\ y_{\max} \dfrac{1 + \log_e[A(|x|/x_{\max})]}{1 + \log_e A} \operatorname{sgn} x & \text{for } \dfrac{1}{A} < \dfrac{|x|}{x_{\max}} < 1 \end{cases} \tag{13.43}$$

The standard value of the parameter A is 87.56, and for this value, using an 8-bit converter, the SNR is 38.0 dB. The A-law compression characteristic is approximated, in a manner similar to the μ-law compressor, by a sequence of 16 linear chords spanning the output range. The lower two chords in each quandrant are in fact a signal chord corresponding to the linear segment of the A-law compressor. One important difference between the A-law and the μ-law compression characteristics is that the A-law standard has a midriser at the origin, while the μ-law standard has a midtread at the origin. Thus, the A-law compressor has no zero value, and hence it exhibits no interval for which data are not being transmitted for zero input.

There are direct mappings from the A-law 8-bit compressed ADC format to a 12-bit linear binary code, and from the μ-law 8-bit compressed format to a 13-bit linear code [8]. This operation permits the A/D conversion to be performed with a uniform quantizer and then to be mapped to the smaller number of bits in a code converter. This also permits the inverse mapping at the receiver (i.e., the expansion) to be performed on the digital sample.

Pulse-Code Modulation. One of the tasks performed by a pulse-code modulation (PCM) process is the conversion of a waveform source to a binary sequence

discrete source. This task is performed in a three-step process—sampling, quantizing, and encoding. We have addressed the sampling process in Chapter 2, and we have addressed the quantizing process in this chapter, as well as in Chapter 2. We note that the encoding process, following quantization (see Figure 2.2), is often embedded in the hardware that performs the quantization. It can be described as follows: successive approximation (SA) analog-to-digital (A/D) converters form the successive bits of the encoded data by a feedback, comparison, and decision process. In the feedback process, a binary search is conducted over the range of possible input levels by repeatedly asking, Is the input signal above or below the midpoint of the remaining uncertainty interval? By this technique, the uncertainty interval is reduced by one-half for each comparison and decision step until the uncertainty range matches the allowable quantizing interval.

In the SA conversion, the results of each previous decision reduce the uncertainty to be resolved during the next decision. In a similar manner, the results of the previous A/D conversions can be used to reduce the uncertainty to be resolved during the next conversion. This reduction in uncertainty is achieved by carrying forward to the next sample auxiliary information from earlier samples. This information is called the redundant part of the signal, and by carrying it forward we reduce the interval of uncertainty over which the quantizer and encoder must search for the next signal sample. Carrying data forward is one method of achieving *redundancy reduction.*

13.3 DIFFERENTIAL PULSE-CODE MODULATION

By the use of past data to assist in measuring (i.e., quantizing) new data, we leave ordinary PCM and enter the realm of differential PCM (DPCM). In DPCM, a prediction of the next sample value is formed from past values. This prediction can be thought of as instructions for the quantizer to conduct its search for the next sample value in a particular interval. By using the redundancy in the signal to form a prediction, the region of uncertainty is reduced and the quantization can be performed with a reduced number of decisions (or bits) for a given quantization level or with reduced quantization levels for a given number of decisions (or bits). The reduction in redundancy is realized by subtracting the prediction from the next sample value. This difference is called the *prediction error.*

The quantizing methods described in Section 13.2 are called *instantaneous* or *memoryless* quantizers because the digital conversion is based on the single (current) input sample. In Section 13.1 we identified the properties of sources that permitted source rate reductions. These properties were nonequiprobable source levels and nonindependent sample values. Instantaneous quantizers achieve source-coding gains by taking into account the probability density assignment for each sample. The quantizing methods that take account of sample-to-sample correlation are noninstantaneous quantizers. These quantizers reduce source redundancy by first converting the correlated input sequence into a related sequence with reduced correlation, reduced variance, or reduced bandwidth. This new sequence is then quantized with fewer bits.

The correlation characteristics of a source can be visualized in the time domain by samples of its autocorrelation function and in the frequency domain by its power spectrum. If we examine a power spectrum $G_x(f)$ of a short-term speech signal, as shown in Figure 13.18, we find that the spectrum has a global maxima in the neighborhood of 300 to 800 Hz and falls off at a rate of 6 to 12 dB/octave. By interpreting this power spectrum, we can infer certain properties of the time function from which it was derived. We observe that large changes in the signal occur slowly (low frequency) and that rapid changes in the signal (high frequency) must be of low amplitude. An equivalent interpretation can be found in the autocorrelation function $R_x(T)$ of the signal, as shown in Figure 13.19. Here a broad, slowly changing autocorrelation function suggests that there will be only slight change on a sample-to-sample basis, and that a time interval exceeding the correlation distance is required for a full amplitude change. The correlation distance seen in Figure 13.19 is the time difference between the peak correlation and the first zero correlation. In particular, correlation values for typical single-sample delay is on the order of 0.79 to 0.87, and the correlation distance is on the order of 4 to 6 sample intervals of T seconds per interval.

Since the difference between adjacent time samples for speech is small, coding techniques have evolved based on transmitting sample-to-sample differences rather than actual sample values. Successive differences are in fact a special case of a class of non-instantaneous converters called N-tap linear predictive coders. These coders, sometimes called predictor-corrector coders, predict the next input sample value based on the previous input sample values. This structure is shown in Figure 13.20. In this type of converter, the transmitter and the receiver have the same prediction model, which is derived from the signal's correlation characteristics. The encoder forms the prediction error (or the residue) as the difference between the next measured sample value and the predicted sample value. The equation for the prediction loop is

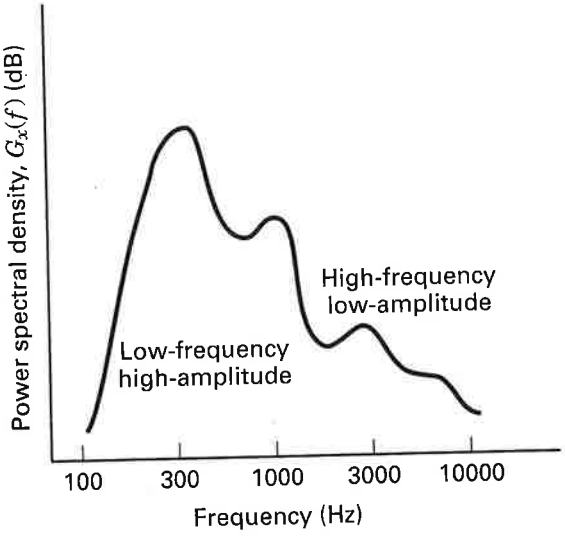


Figure 13.18 Typical power spectrum for speech signals.

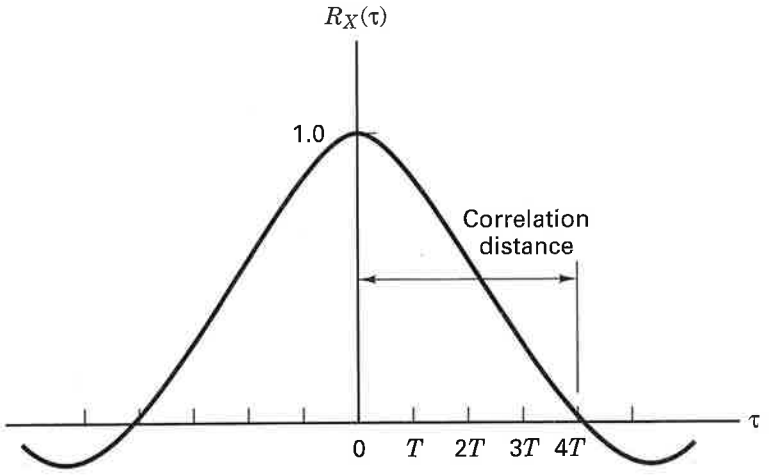


Figure 13.19 Autocorrelation function for typical speech signals.

$$d(n) = x(n) - \hat{x}(n)$$

where $x(n)$ is the nth input sample, $\hat{x}(n)$ is the predicted value of that sample, and $d(n)$ is the associated prediction error. This is performed in the predict-and-compare loop, the upper loop of the encoder shown in Figure 13.20. The encoder corrects its prediction by forming the sum of its prediction and the prediction error. The equations for the correction loop are

$$\tilde{d}(n) = \text{quant}\,[d(n)]$$

$$\tilde{x}(n) = \hat{x}(n) + \tilde{d}(n)$$

where quant $(\cdot)$ represents the quantization operation, $\tilde{d}(n)$ is the quantized version of the prediction error, and $\tilde{x}(n)$ is the corrected and quantized version of the input sample. This is performed in the predict-and-correct loop, the lower loop of the encoder, and the only loop of the decoder in Figure 13.20. The decoder must also be informed of the prediction error so that it can use its correction loop to correct its prediction. The decoder "mimics" the feedback loop of the encoder. The communication task is that of transmitting the difference (the error signal) between the predicted and the actual data sample. For this reason, this class of coder is often called a differential pulse code modulator (DPCM). If the prediction model forms predictions that are close to the actual sample values, the residues will exhibit reduced variance (relative to the original signal). From Section 13.2, we know that the number of bits required to move data through the channel with a given fidelity is related to the signal variance. Hence, the reduced variance sequence of residues can be moved through the channel with a reduced data rate.

The predictive converters must have a short-term memory that supports the real-time operations required for the prediction algorithm. In addition, they will

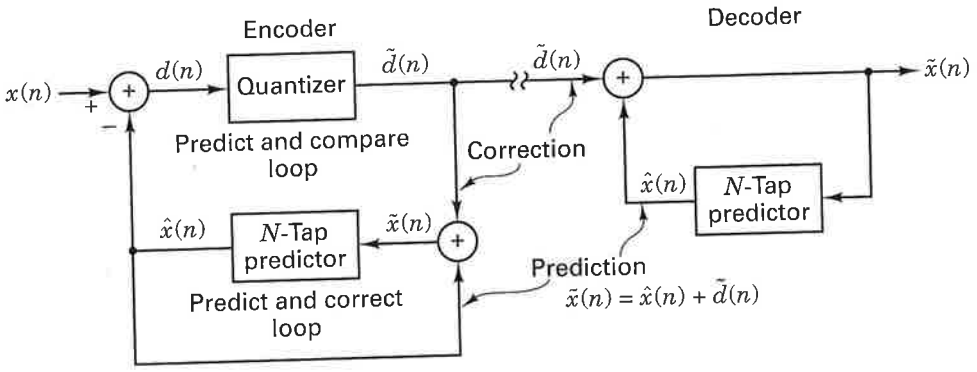


Figure 13.20 *N*-tap predictive differential pulse code modulator (DPCM).

often have a long-term memory that supports the slow time, often data-dependent operations, such as automatic gain control and filter coefficient adjustments. Predictors that incorporate the slower, data-dependent adjustment algorithms are called *adaptive predictors.*

13.3.1 One-Tap Prediction

The one-tap linear prediction coding (LPC) filter in the DPCM process predicts the next input sample value based on the previous input sample value. The prediction equation is of the form

$$x(n|n-1) = a\,x(n-1)|n-1) \tag{13.44}$$

where $x(n|m)$ is the estimate of x at time n given all the samples collected up through time m, and where "a" is a parameter used to minimize the prediction error. The prediction error available after the measurement is of the form

$$d(n) = [x(n) - x(n|n-1)] \tag{13.45a}$$

$$= [x(n) - a\,x(n-1|n-1)] \tag{13.45b}$$

The mean-squared error is of the form

$$\mathbf{E}\{d^2(n)\} = \mathbf{E}\{x(n)\,x(n) - 2\,a\,x(n)\,x(n-1|n-1) + a^2\,x(n-1|n-1)\,x(n-1|n-1)\} \tag{13.46}$$

If $x(n-1|n-1)$ is an unbiased estimate of $x(n-1)$, Equation (13.46) can be written as

$$R_d(0) = R_x(0) - 2\,a\,R_x(1) + a^2\,R_x(0) \tag{13.47a}$$

$$= R_x(0)\,[1 + a^2 - 2\,a\,C_x(1)] \tag{13.47b}$$

where $R_d(n)$ and $R_x(n)$ are the autocorrelation functions of the prediction error and the input signal, respectively, $R_d(0)$ is the power in the error, $R_x(0)$ is the power in the signal, and $C_x(n) = R_x(n)/R_x(0)$ is the normalized autocorrelation func-

tion. We can select the parameter a to minimize the prediction error power of Equation (13.47) by setting a zero the partial derivative of $R_d(0)$ with respect to a:

$$\frac{\partial R_d(0)}{\partial a} = R_x(0)\,[2a - 2C_x(1)] \tag{13.48}$$

Setting to zero and solving for a^{opt}, the optimal solution, we have

$$a^{\text{opt}} = C_x(1) \tag{13.49}$$

Substituting a^{opt} back into Equation (13.47), we obtain

$$R_d^{\text{opt}}(0) = R_x(0)[1 + a^{\text{opt}}C_x(1) - 2a^{\text{opt}}C_x(1)] \tag{13.50a}$$

$$= R_x(0)\,[1 - a^{\text{opt}}C_x(1)] \tag{13.50b}$$

$$= R_x(0)\,[1 - C_x^2(1)] \tag{13.50c}$$

We can define the *prediction gain* of the encoder as the ratio of input to output variances, $R_x(0)/R_d(0)$. For a fixed bit rate this gain represents an increase in output SNR, while for a fixed output SNR this gain represents a reduced bit-rate description. We note that the prediction gain for the optimal predictor is always greater than one for any value of signal correlation $R_x(0)$ as used in Equation (13.50b). On the other hand, the prediction gain is greater than one for the nonoptimum unity-gain, one-tap predictor, only if the signal correlation exceeds 0.5 as used in Equation (13.47b).

Example 13.7 Prediction Gain of a One-Tap LPC Filter

A signal with correlation coefficient $C_x(1)$ equal to 0.8 is to be quantized with a one-tap LPC filter. Determine the prediction gain when the prediction coefficient is (a) optimized with respect to the minimum prediction error, or (b) set to unity.

Solution

(a) From Equation (13.50c),

$$R_d^{\text{opt}}(0) = R_x(0)\,(1 - 0.64) = 0.36\,R_x(0) \tag{13.51a}$$

$$\text{Prediction gain} = 1/(0.36) = 2.78 \text{ or } 4.44 \text{ dB} \tag{13.51b}$$

(b) From Equation (13.47b),

$$R_d(0) = 2R_x(0)\,(1 - 0.8) = 0.40\,R_x(0) \tag{13.51c}$$

$$\text{Prediction gain} = 1/(0.40) = 2.50 \text{ or } 3.98 \text{ dB} \tag{13.51d}$$

13.3.2 *N*-Tap Prediction

The N-tap LPC filter predicts the next sample value based on a linear combination of the previous N sample values. We will assume that the quantized estimates used by the prediction filters are unbiased and error free. With this assumption, we can drop the double indices (used in Section 13.3.1) from the data in the filter but still use them for the predictions. Then the N-tap prediction equation takes the form

$$x(n \mid n-1) = a_1\,x(n-1) + a_2\,x(n-2) + \cdots + a_N\,x(n-N) \tag{13.52}$$

The prediction error takes the form

$$d(n) = x(n) - x(n \mid n-1) \tag{13.53a}$$

$$= x(n) - a_1 x(n-1) - a_2 x(n-2) - \cdots - a_N x(n-N) \tag{13.53b}$$

The mean-square prediction error is of the form

$$\mathbf{E}\{d(n)d(n)\} = \mathbf{E}\{[x(n) - x(n \mid n-1)]^2\} \tag{13.54}$$

Clearly, the mean-square prediction error is quadratic in the filter coefficients a_j. As we did in Section 13.3.1, we can form the partial derivative of the mean-squared error with respect to each coefficient and solve for the coefficients that set the partials to zero. Formally, taking the partial derivative with respect to the jth coefficient prior to expanding $x(n|n-1)$, we have

$$\frac{\partial R_d(0)}{\partial a_j} = \mathbf{E}\left\{2[x(n) - x(n|n-1)]\frac{\partial x(n|n-1)}{\partial a_j}x(n|n-1)\right\} \tag{13.55a}$$

$$= \mathbf{E}\{2[x(n) - x(n|n-1)][-x(n-j)]\} \tag{13.55b}$$

$$= 2\mathbf{E}\{[x(n) - a_1 x(n-1) - a_2 x(n-2) - \cdots - a_N x(n-N)][-x(n-j)]\} \tag{13.55c}$$

$$= 2[R_x(j) - a_1 R_x(j-1) - a_2 R_x(j-2) - \cdots - a_N R_x(j-N)] \tag{13.55d}$$

This collection of equations (one for each j) can be arranged in matrix form known as the normal equations. This form is

$$\begin{bmatrix} R_x(1) \\ R_x(2) \\ R_x(3) \\ \vdots \\ R_x(N) \end{bmatrix} = \begin{bmatrix} R_x(0) & R_x(-1) & R_x(-2) & \cdots & R_x(-N+1) \\ R_x(1) & R_x(0) & R_x(-1) & \cdots & R_x(-N+2) \\ R_x(2) & R_x(1) & R_x(0) & \cdots & R_x(-N+3) \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ R_x(N-1) & R_x(N-2) & R_x(N-3) & \cdots & R_x(0) \end{bmatrix} \begin{bmatrix} a_1 \\ a_2 \\ a_3 \\ \vdots \\ a_N \end{bmatrix}^{\text{opt}} \tag{13.56a}$$

The normal equations can be written more compactly as

$$\mathbf{r}_x(1, N) = \mathbf{R}_{xx}\,\mathbf{a}^{\text{opt}} \tag{13.56b}$$

where $\mathbf{r}_x(1, N)$ is the correlation vector of delays from 1 through N, $\mathbf{R}_{xx}$ is the correlation matrix (assuming a zero-mean process), and $\mathbf{a}^{\text{opt}}$ is the optimum filter weight vector.

To gain insight into the solution of the normal equations, we now recast the mean-square-error equation (13.54) in matrix form. We have

$$R_d(0) = \mathbf{E}\{[x(n) - \mathbf{a}^T\mathbf{x}(n-1)][x(n) - \mathbf{x}^T(n-1)\mathbf{a}]\} \tag{13.57a}$$

$$= R_x(0) - \mathbf{r}_x^T(1, N)\,\mathbf{a} - \mathbf{a}^T\mathbf{r}_x(-1,-N) + \mathbf{a}^T\mathbf{R}_{xx}\mathbf{a} \tag{13.57b}$$

where $\mathbf{r}^T$ is the transpose of $\mathbf{r}$. Substituting $\mathbf{a}^{\text{opt}}$ for $\mathbf{a}$ in Equation (13.57b), and then substituting $\mathbf{r}_x(1, N)$ for $\mathbf{R}_{xx}\mathbf{a}^{\text{opt}}$ in the resulting equation, yields

$$R_d(0) = R_x(0) - \mathbf{r}_x^T(1, N)\,\mathbf{a}^{\text{opt}} - \mathbf{a}^{\text{opt}\,T}\mathbf{r}_x(-1,-N) + \mathbf{a}^{\text{opt}\,T}\mathbf{r}_x(1, N) \tag{13.58a}$$

$$= R_x(0) - \mathbf{r}_x^T(-1,-N)\,\mathbf{a}^{\text{opt}} \tag{13.58b}$$

We can now bring the right-hand side of Equation (13.56) over to the left-hand side, and use Equation (13.58b) to augment the top row of the matrix to obtain the whitening form of the optimal predictor:

$$\begin{bmatrix} R_x(0) & R_x(-1) & R_x(-2) & R_x(-3) & \cdots & R_x(-N) \\ R_x(1) & R_x(0) & R_x(-1) & R_x(-2) & \cdots & R_x(-N+1) \\ R_x(2) & R_x(1) & R_x(0) & R_x(-1) & \cdots & R_x(-N+2) \\ R_x(3) & R_x(2) & R_x(1) & R_x(0) & \cdots & R_x(-N+3) \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ R_x(N) & R_x(N-1) & R_x(N-2) & R_x(N-3) & \cdots & R_x(0) \end{bmatrix} \begin{bmatrix} 1 \\ -a_1 \\ -a_2 \\ -a_3 \\ \vdots \\ -a_N \end{bmatrix}^{\text{opt}} = \begin{bmatrix} R_d(0) \\ 0 \\ 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix} \tag{13.59}$$

In this form, the only nonzero output of the matrix product occurs at time zero, which is akin to an output impulse.

The top row of Equation (13.59) states that the power in the prediction error is of the form

$$R_d(0) = R_x(0)[1 - a_1\,C_x(1) - a_2\,C_x(2) - \cdots - a_N\,C_x(N)] \tag{13.60}$$

Compare this form with that of Equation (13.50b). An interesting property of the optimal N-tap predictor filter is that the coefficient set that obtains the minimum mean-square prediction error also predicts, with zero error, the next $N-1$ correlation samples from the previous $N-1$ correlation samples. For fixed filter coefficients, the DPCM coder can achieve a prediction gain, relative to linear quantizing, of 6 to 8 dB [9, 10]. This prediction gain is essentially independent of filter length once the length exceeds three or four taps. Additional gain is available if the coder has slow adaptive capabilities. Adaptive coders are introduced in Section 13.3.3 and discussed in some detail in Section 13.3.4.

13.3.3 Delta Modulation

The delta modulator, often denoted Δ modulator, is a process that embeds a low resolution A-to-D converter in a sampled data feedback loop that operates at rates far in excess of the signal's required Nyquist rate. The motivation for this technique is our awareness that in the conversion process, speed is less expensive than precision, and that by being clever one can use faster signal processing to obtain higher precision.

Equation (13.50c) demonstrated that the prediction gain for a one-tap predictor could be large if the normalized correlation coefficient, $C_x(1)$, is close to unity. Working toward the goal of high sample-to-sample correlation, the predictive filter is generally operated at a rate that far exceeds the Nyquist rate. For example, the sample rate might be chosen to be 64 times the Nyquist rate. Then for a 20 kHz bandwidth with a nominal sample rate of 48 kHz, the high correlation prediction

filter would operate at a 3.072 MHz sample rate. The justification for the high sample rate is to insure that the sampled data is highly correlated so that a simple one-tap predictor will exhibit a small prediction error, which in turn permits the quantizer operating in the error loop to operate with a very small number of bits. The simplest form of the quantizer is a one-bit quantizer, which is, in fact, only a comparator that detects and reports the sign of the difference signal. As a result, the prediction error signal is a 1-bit word that has the interesting advantage of not requiring word framing in subsequent processing.

The block diagram of the one-tap linear predictor, illustrated in Figure 13.20, is shown in Figure 13.21, with slight modifications. Note that the one-tap predict and correct loop is now a simple integrator and that a low-pass reconstruction filter follows the predict and correct loop at the decoder. This filter removes the out-of-band quantizing noise that is generated by the two-level coding and that extends beyond the information bandwidth of this coding process. The coder is completely characterized by the sampling frequency, the quantizing step size to resolve the prediction error or *delta* of the loop, and the reconstruction filter. The equations for prediction and for the residual error of the modulator are of the form

$$x(n \mid n-1) = x(n-1 \mid n-1) \tag{13.61a}$$

$$d(n) = x(n) - x(n \mid n-1) \tag{13.61b}$$

where n is a sample index. This structure, sometimes called the *deltamodulator*, is a DPCM process for which the predict-and-correct loop consists of a digital accumulator.

13.3.4 Sigma-Delta Modulation

The structure of the Σ-Δ modulator can be examined from a number of perspectives, the most appealing being that of a modified one-tap DPCM converter, and that of an error-feedback converter. Let us start with the modified one-tap DPCM

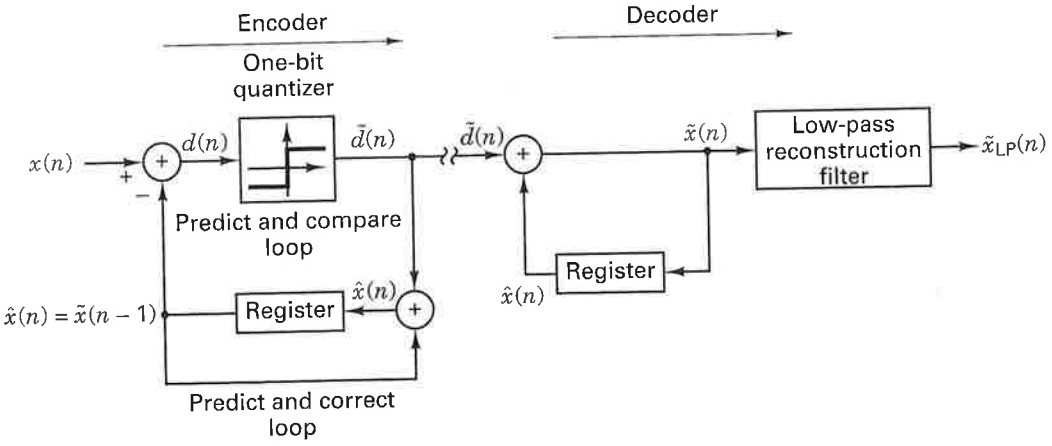


Figure 13.21 One-tap, one-bit DPCM coder (delta-modulator).

converter. As indicated earlier, the loop relies on high correlation of successive samples, a condition we assure by significant oversampling. We can enhance the correlation of the sampled data presented to the modulator by prefiltering the data with an integrator and then compensating for the prefilter with a postfiltering differentiator. This structure is shown in Figure 13.22, where the integrators, differentiator, and delay functions are expressed in terms of the z transform. (See Appendix E.) We can then rearrange the signal flow blocks to realize an economy of implementation. At the input to the encoder, there are the outputs of two digital integrators, which are summed and presented to the loop quantizer. Our first modification is that we can share a single digital integrator by sliding the two integrators through the summing junction in the encoder. Our second modification to the encoder is that the post filter differentiator can be moved to the decoder, which then cancels the digital integrator at the input to the decoder. All that remains of the decoder is the low-pass reconstruction filter. This simplified form of the modified DPCM system is shown in Figure 13.23. This form, called a *sigma-delta* modulator, contains an integrator (the *sigma*) and a DPCM modulator (the *delta*) [11].

The second perspective useful for understanding the Σ-Δ modulator is that of noise feedback loop. We understand that a quantizer adds an error to its input to form its output. When the signal is highly oversampled, not only are the samples highly correlated, the errors are as well. When errors are highly correlated, they are predictable, and thus they can be subtracted from the signal presented to the quantizer prior to the quantization process. When the signal and error are highly oversampled, the previous quantization error can be used as a good estimate of the current error. The previous error, formed as the difference between the input and output of the quantizer, is stored in a delay register for use as the estimate of the next quantization error. This structure is shown in Figure 13.24. The signal flow graph of Figure 13.24 can be redrawn to emphasize the two inputs, signal and quantization noise, as well as the two loops, one including the quantizer and one not including it. This form is shown in Figure 13.25 and except for explicitly showing the feedback leg of the digital integrator, this has the same structure as presented in Figure 13.23. From Figure 13.25, if follows that the output of the Σ-Δ modulator and its z-transform (see Appendix E) can be written as

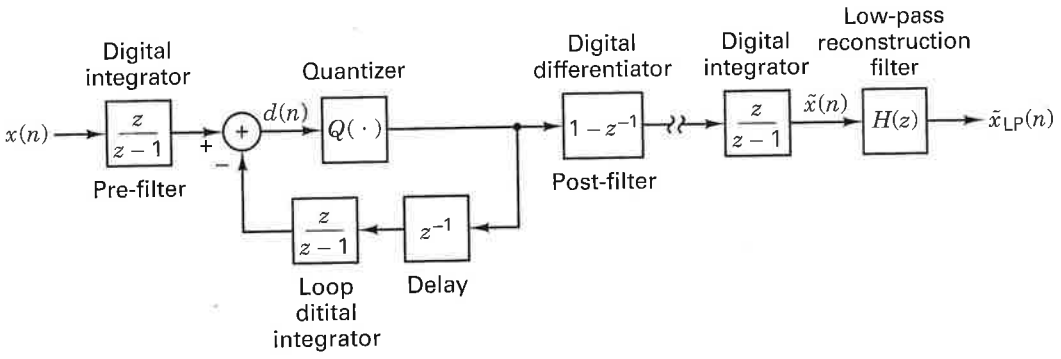


Figure 13.22 One-bit delta modulator, pre and post filtered form.

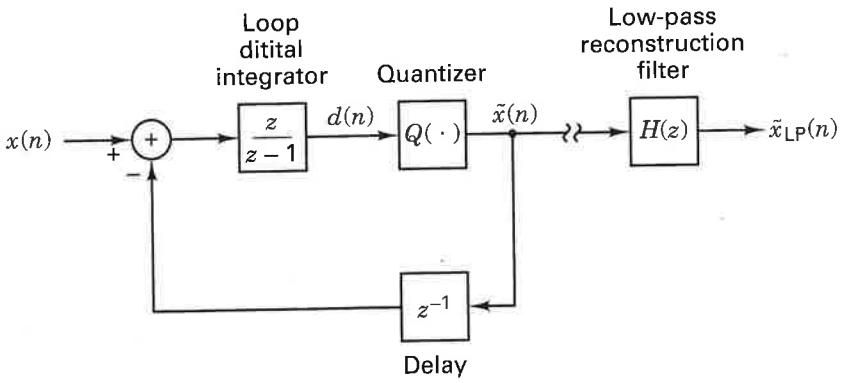


Figure 13.23 Σ-Δ modulator as a rearranged pre and post filtered Δ-modulator.

$$\begin{aligned} y(n) = \tilde{x}(n) &= x(n) - q(n-1) + q(n) \\ &= x(n) + [q(n) - q(n-1)] \end{aligned} \tag{13.62}$$

$$\begin{aligned} Y(Z) &= X(Z) - Z^{-1}Q(Z) + Q(Z) \\ &= X(Z) + Q(Z)[1 - Z^{-1}] \\ &= X(Z) + Q(Z)\frac{Z-1}{Z} \end{aligned} \tag{13.63}$$

Intuitively, Equation (13.63) is pleasing, since it shows that the loop does not affect the input signal since only the noise is being circulated by the loop, and only the noise experiences the effect of the loop. The integrator in the feedback path of the

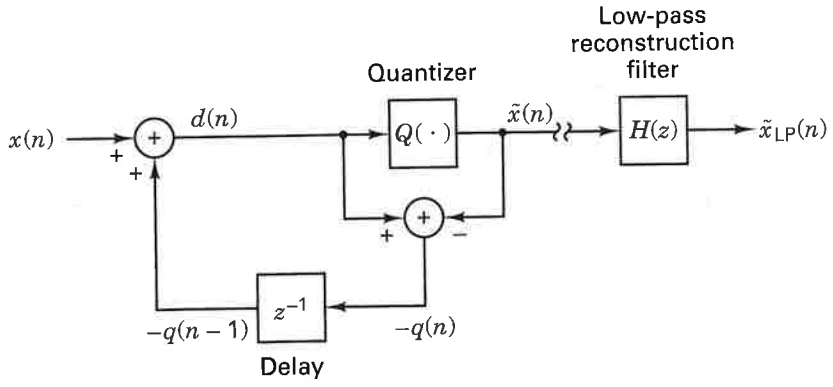


Figure 13.24 Σ-Δ modulator as a noise feedback process.

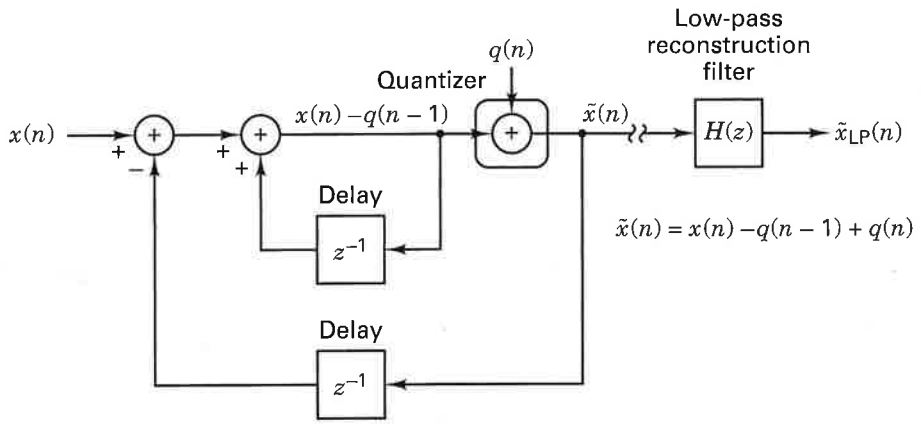


Figure 13.25 Noise feedback quantizer redrawn as Σ-Δ modulator.

noise signal is inverted by the unity-gain feedback loop to become a differentiator to the noise source.

The z-plane (like its analog counterpart, the s-plane) provides a convenient vehicle for mapping a frequency transfer function. (See Appendix E.) Such a transfer function is typically characterized in terms of a numerator and denominator polynomial whose roots are referred to as the *zeros* and *poles*, respectively, of the transfer function. These zeros and poles can be viewed as a surface emanating from the plane, representing the magnitude of the transfer function. The surface is easily visualized as being a stretched rubber sheet pulled off the ground by tent poles located at the pole positions and being held to the ground by tent stakes at the zero positions. The magnitude of the frequency response is the level of this surface observed as we traverse the unit circle in the z-plane (or $j\omega$ axis in the s-plane). Note that the *noise transfer function* (NTF), which is the frequency transfer function of the loop applied to the noise, has a pole at the origin and a transmission zero at dc ($z = e^{j\theta}$, $\theta = 0$, so $z = 1$). The pole zero plot of the NTF and the spectral response of the NTF along with a typical spectrum of the input signal are shown in Figure 13.26. Note that the zero of the noise transfer function is located at dc, and that in the neighborhood of dc, the quantization noise is suppressed by the NTF. Thus, there is no significant noise near dc due to the NTF, while concurrently, the signal spectrum is restricted by the significant oversampling to reside in a small neighborhood around dc with a typical width of approximately 1.5% of the sample rate. The function of the reconstruction filter is to suppress the quantization noise outside the signal bandwidth. The sample rate at the output of the filter is reduced to match the reduced bandwidth of the now nearly noise-free signal. Additional improvement in noise suppression can be obtained by raising the order of the zero in the NTF of the loop. Many Σ-Δ modulators are designed with NTFs displaying dual or triple zeros. Since the NTF zeros set the output noise power to zero, it hardly matters what level of noise power is presented to the feedback loop. Consequently, most Σ-Δ modulators are designed to operate with 1-bit converters with a number of very high performance modulators operating with 4-bit converters.