

MFSK there are two mechanisms at work. The numerator shows the same increase in R/W with larger M , as in the case of MPSK. But the denominator indicates a decrease in R/W with larger M . As M grows larger, the denominator grows faster than the numerator, and thus R/W decreases. The MFSK points plotted in Figure 9.6 confirm the relationship shown in Equation (9.21), that orthogonal signaling such as MFSK is a bandwidth-expansive scheme. From Figure 9.6, it can be seen that MFSK modulation can be used for realizing a reduction in required E_b/N_0 , at the cost of increased bandwidth.

It is important to emphasize that in Equations (9.18) and (9.19) for MPSK, and for all the MPSK points plotted in Figure 9.6, Nyquist (ideal rectangular) filtering has been assumed. Such filters are not realizable. For *realistic* channels and waveforms, the required transmission bandwidth must be *increased* in order to account for *realizable* filters.

In each of the examples that follow, we consider radio channels, disturbed *only* by additive white Gaussian noise (AWGN) and having no other impairments. For simplicity, the modulation choice is limited to *constant-envelope types*—either MPSK or noncoherent orthogonal MFSK. Thus, for an *uncoded* system, if the channel is bandwidth limited, MPSK is selected, and if the channel is power limited, MFSK is selected. Note that, *when error-correction coding is considered*, modulation selection is not so simple, because there exist coding techniques [9] that can provide power-bandwidth trade-offs more effectively than would be possible through the use of any M -ary modulation scheme.

Note that in the most general sense, M -ary signaling can be regarded as a *waveform-coding* procedure. That is, whenever we select an M -ary modulation technique instead of a binary one, we *in effect* have replaced the binary waveforms with *better* waveforms—either better for bandwidth performance (MPSK), or better for power performance (MFSK). Even though orthogonal MFSK signaling can be thought of as being a coded system (it can be described as a first-order Reed-Muller code [10]), we shall here restrict our use of the term *coded system* to refer only to those traditional error-correction codes using redundancies, such as block codes or convolutional codes.

9.7.4 Requirements for MPSK and MFSK Signaling

The basic relationship between the symbol (or waveform) transmission rate R_s and the data rate R was shown in Equation (9.16) to be

$$R_s = \frac{R}{\log_2 M}$$

Using this relationship together with Equations (9.18) through (9.21), and a given data rate of $R = 9600$ bit/s, Table 9.1 has been compiled [11]. The table is a summary of symbol rate, minimum bandwidth, and bandwidth efficiency for MPSK and noncoherent orthogonal MFSK, for the values of $M = 2, 4, 8, 16$, and 32 . Also included in Table 9.1 are the required values of E_b/N_0 to achieve a bit-error probability of 10^{-5} for MPSK and MFSK for each value of M shown. These E_b/N_0

TABLE 9.1 Symbol Rate, Minimum Bandwidth, Bandwidth Efficiency, and Required E_b/N_0 for MPSK and Noncoherent Orthogonal MFSK Signaling at 9600 bit/s

M	k	R (bit/s)	R_s (symb/s)	MPSK Minimum Bandwidth (Hz)	MPSK R/W	MPSK E_b/N_0 (dB) $P_B = 10^{-5}$	Noncoherent Orthog MFSK Min Bandwidth (Hz)	MFSK R/W	MFSK E_b/N_0 (dB) $P_B = 10^{-5}$
2	1	9600	9600	9600	1	9.6	19,200	1/2	13.4
4	2	9600	4800	4800	2	9.6	19,200	1/2	10.6
8	3	9600	3200	3200	3	13.0	25,600	1/3	9.1
16	4	9600	2400	2400	4	17.5	38,400	1/4	8.1
32	5	9600	1920	1920	5	22.4	61,440	5/32	7.4

entries were computed using relationships that are presented later. The E_b/N_0 entries corroborate the trade-offs shown in Figure 9.6. As M increases, MPSK signaling provides more bandwidth efficiency at the cost of increased E_b/N_0 , while MFSK signaling allows for a reduction in E_b/N_0 at the cost of increased bandwidth. The next three sections are presented in the context of examples taken from Table 9.1.

9.7.5 Bandwidth-Limited Uncoded System Example

Suppose we are given a bandwidth-limited AWGN radio channel with an available bandwidth of $W = 4000$ Hz. Also, consider that the link constraints (transmitter power, antenna gains, path loss, etc.) result in the ratio of received signal power to noise-power spectral density (P_r/N_0) being equal to 53 dB-Hz. Let the required data rate R be equal to 9600 bits/s, and let the required bit-error performance P_B be at most 10^{-5} . The goal is to choose a modulation scheme that meets the required performance. In general, an error-correction coding scheme may be needed if none of the allowable modulation schemes can meet the requirements. However, in this example, we will see that the use of error-correction coding is not necessary.

For any digital communication system, the relationship between received power to noise-power spectral density (P_r/N_0) and received bit-energy to noise-power spectral density (E_b/N_0) was shown in Equation (5.20c) to be

$$\frac{P_r}{N_0} = \frac{E_b}{N_0} R \quad (9.22)$$

Solving for E_b/N_0 , in decibels, we obtain

$$\begin{aligned} \frac{E_b}{N_0} \text{ (dB)} &= \frac{P_r}{N_0} \text{ (dB-Hz)} - R \text{ (dB-bit/s)} \\ &= 53 \text{ dB-Hz} - (10 \times \log_{10} 9600) \text{ dB-bit/s} = 13.2 \text{ dB (or 20.89)} \end{aligned} \quad (9.23)$$

Since the required data rate of 9600 bits/s is much larger than the available bandwidth of 4000 Hz, the channel can be described as *bandwidth limited*. We therefore select MPSK as our modulation scheme. Recall that we have confined the possible

modulation choices to be constant-envelope types; without such a restriction, it would be possible to select a modulation type with greater bandwidth efficiency. In an effort to conserve power, we next compute the *smallest possible* value of M , such that the symbol rate is *at most* equal to the available bandwidth of 4000 Hz. From Table 9.1, it is clear that the smallest value of M meeting this requirement is $M = 8$. Our next task is to determine whether the required bit-error performance of $P_B \leq 10^{-5}$ can be met by using 8-PSK modulation alone, or whether it is necessary to additionally use an error-correction coding scheme. It can be seen from Table 9.1, that 8-PSK *alone* will meet the requirements, since the required E_b/N_0 listed for 8-PSK is less than the received E_b/N_0 that was derived in Equation (9.23). However, imagine that we do not have Table 9.1. Let us demonstrate how to evaluate whether or not error-correction coding is necessary.

Figure 9.8 shows the basic modulator/demodulator (MODEM) block diagram summarizing the functional details of this design. At the modulator, the transformation from data bits to symbols yields an output symbol rate R_s that is a factor $(\log_2 M)$ smaller than the input data-bit rate R , as can be seen in Equation (9.16). Similarly, at the input to the demodulator, the symbol-energy to noise-power spectral density E_s/N_0 is a factor $(\log_2 M)$ larger than E_b/N_0 , since each symbol is made up of $(\log_2 M)$ bits. Because E_s/N_0 is larger than E_b/N_0 by the same factor that R_s is smaller than R , we can expand Equation (9.22), as follows:

$$\frac{P_r}{N_0} = \frac{E_b}{N_0} R = \frac{E_s}{N_0} R_s \quad (9.24)$$

The demodulator receives a waveform (in this example, one of $M = 8$ possible phase shifts) during each time interval T_s . The probability that the demodulator makes a symbol error $P_E(M)$ is well approximated by [12], and we write

$$P_E(M) \approx 2Q \left[\sqrt{\frac{2E_s}{N_0}} \sin\left(\frac{\pi}{M}\right) \right] \quad \text{for } M > 2 \quad (9.25)$$

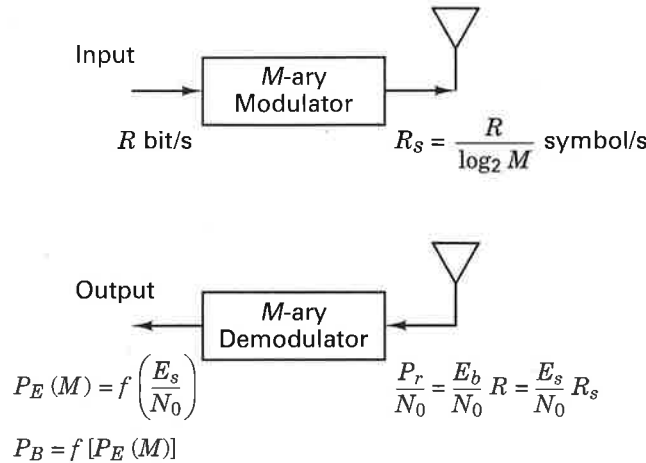


Figure 9.8: Basic modulator/demodulator (MODEM) without channel coding.

where $Q(x)$, the *complementary error function*, was defined in Equation (3.43) as

$$Q(x) = \frac{1}{\sqrt{2\pi}} \int_x^\infty \exp\left(-\frac{u^2}{2}\right) du$$

In Figure 9.8 and all the figures that follow, rather than show explicit probability relationships, the generalized notation $f(x)$ has been used to indicate some functional dependence on x .

A traditional way of characterizing communication (power) efficiency or error performance in digital systems is in terms of the received E_b/N_0 in decibels. This E_b/N_0 description has become standard practice. However, recall that at the input to the demodulator/detector, there are no bits; there are only waveforms that have been assigned bit meanings. Thus, the received E_b/N_0 value represents a bit-apportionment of the arriving waveform energy. A more precise (but unwieldy) name would be the energy per *effective bit* versus N_0 . To solve for $P_E(M)$ in Equation (9.25), we first need to compute the ratio of received symbol-energy to noise-power spectral density, E_s/N_0 . Since, from Equation (9.23), $E_b/N_0 = 13.2$ dB (or 20.89), and because each symbol is made up of $(\log_2 M)$ bits, we compute, with $M = 8$,

$$(9.24) \quad \frac{E_s}{N_0} = (\log_2 M) \frac{E_b}{N_0} = 3 \times 20.89 = 62.67 \quad (9.26)$$

Using the results of Equation (9.26) in Equation (9.25), yields the symbol-error probability, $P_E = 2.2 \times 10^{-5}$. To transform this to bit-error probability, we need to use the relationship between bit-error probability P_B and symbol-error probability P_E for multiple-phase signaling [10]. We write

$$(9.25) \quad P_B \approx \frac{P_E}{\log_2 M} \quad (\text{for } P_E \ll 1) \quad (9.27)$$

which is a good approximation, when Gray coding [12] is used for the bit-to-symbol assignment. This last computation yields $P_B = 7.3 \times 10^{-6}$, which meets the required bit-error performance. Thus, in this example, no error-correction coding is necessary and 8-PSK modulation represents the design choice to meet the requirements of the bandwidth-limited channel (which we had predicted by examining the required E_b/N_0 values in Table 9.1).

9.7.6 Power-Limited Uncoded System Example

Now, suppose that we have exactly the same data rate and bit-error probability requirements as in the example of Section 9.7.5. However, in this example, let the available bandwidth W be equal to 45 kHz, and let the available P_r/N_0 be equal to 48 dB-Hz. As before, the goal is to choose a modulation or modulation/coding scheme that yields the required performance. In this example, we shall again find that error-correction coding is not required.

The channel in this example is clearly not bandwidth limited since the available bandwidth of 45 kHz is more than adequate for supporting the required data rate of 9600 bits/s. The received E_b/N_0 is found from Equation (9.23), as follows:

$$\frac{E_b}{N_0}(\text{dB}) = 48 \text{ dB-Hz} - (10 \times \log_{10} 9600) \text{ dB-bit/s} = 8.2 \text{ dB (or 6.61)} \quad (9.28)$$

Since there is abundant bandwidth but a relatively small amount of E_b/N_0 for the required bit-error probability, this channel may be referred to as *power limited*. We therefore choose MFSK as the modulation scheme. In an effort to conserve power, we next search for the *largest possible* M such that the MFSK minimum bandwidth is not expanded beyond our available bandwidth of 45 kHz. From Table 9.1, we see that such a search results in the choice of $M = 16$. Our next task is to determine whether the required error performance of $P_B \leq 10^{-5}$ can be met by using 16-FSK alone, without the use of any error-correction coding. Similar to the previous example, it can be seen from Table 9.1, that 16-FSK *alone* will meet the requirements, since the required E_b/N_0 listed for 16-FSK is less than the received E_b/N_0 that was derived in Equation (9.28). However, imagine again that we do not have Table 9.1. Let us demonstrate how to evaluate whether or not error-correction coding is necessary.

As before, the block diagram in Figure 9.8 summarizes the relationship between symbol rate R_s and bit rate R , and between E_s/N_0 and E_b/N_0 , which is identical to each of the respective relationships in the previous bandwidth-limited example. In this example, the 16-FSK demodulator receives a waveform (one of 16 possible frequencies) during each symbol time interval T_s . For noncoherent MFSK, the probability that the demodulator makes a symbol error is approximated by [13]

$$P_E(M) \leq \frac{M-1}{2} \exp\left(-\frac{E_s}{2N_0}\right) \quad (9.29)$$

To solve for $P_E(M)$ in Equation (9.29), we need to compute E_s/N_0 , as we did in Example 1. Using the results of Equation (9.28) in Equation (9.26), with $M = 16$, we get

$$\frac{E_s}{N_0} = (\log_2 M) \frac{E_b}{N_0} = 4 \times 6.61 = 26.44 \quad (9.30)$$

Next, we combine the results of Equation (9.30) in Equation (9.29) to yield the symbol-error probability $P_E = 1.4 \times 10^{-5}$. To transform this to bit-error probability P_B , we need to use the relationship between P_B and P_E for orthogonal signaling [13], given by

$$P_B = \frac{2^{k-1}}{2^k - 1} P_E \quad (9.31)$$

This last computation yields $P_B = 7.3 \times 10^{-6}$, which meets the required bit-error performance. Thus, we can meet the given specifications for this power-limited channel by using 16-FSK modulation, without any need for error-correction coding (which we had predicted by examining the required E_b/N_0 values in Table 9.1).

9.7.7 Bandwidth-Limited and Power-Limited Coded System Example

In this example, we start with the same channel parameters as in the bandwidth-limited example of Section 9.7.5, namely, $W = 4000$ Hz, $P_r/N_0 = 53$ dB-Hz, and $R = 9600$ bits/s, with one exception. In the present example, we specify that the bit-error probability must be *at most* 10^{-9} . Since the available bandwidth is 4000 Hz, and from Equation (9.23) the available E_b/N_0 is 13.2 dB, it should be clear from Table 9.1, that the system is both bandwidth limited *and* power limited (8-PSK is the only possible choice to meet the bandwidth constraint; however, the available E_b/N_0 of 13.2 dB is certainly insufficient to meet the required bit-error probability of 10^{-9}). For such a small value of P_B , the system shown in Figure 9.8 will obviously be inadequate, and we need to consider the performance improvement that error-correction coding (within the available bandwidth) can provide. In general, one can use convolutional codes or block codes. To simplify the explanation, we shall choose a block code. The Bose, Chaudhuri, and Hocquenghem (BCH) codes form a large class of powerful error-correcting cyclic (block) codes [14]. For this example, let us select one of the codes from this family of codes. Table 9.2

TABLE 9.2 BCH Codes (Partial Catalog)

n	k	t
7	4	1
15	11	1
	7	2
	5	3
31	26	1
	21	2
	16	3
	11	5
63	57	1
	51	2
	45	3
	39	4
	36	5
	30	6
127	120	1
	113	2
	106	3
	99	4
	92	5
	85	6
	78	7
	71	9
	64	10
	57	11
	50	13
	43	14
	36	15
	29	21
	22	23
	15	27
	8	31

presents a partial catalog of the available BCH codes in terms of n , k , and t , where k represents the number of information or data bits that the codes transforms into a longer block or n code bits (also called *channel bits* or *channel symbols*), and t represents the largest number of incorrect channel bits that the code can correct within each n -sized block. The *rate* of a code is defined as the ratio k/n ; its inverse represents a measure of the code's redundancy.

Since this example is represented by the same bandwidth-limited parameters that were given in Section 9.7.5, we start with the same 8-PSK modulation as before in order to meet the stated bandwidth constraint. However, we now additionally need to employ error-correction coding so that the bit-error probability can be lowered to $P_B \leq 10^{-9}$. To make the optimum code selection from Table 9.2, we are guided by the following goals:

1. The output bit-error probability of the combined modulation/coding system must meet the system error requirement.
2. The rate of the code must not expand the required transmission bandwidth beyond the available channel bandwidth.
3. The code should be as simple as possible. Generally, the shorter the code, the simpler will be its implementation.

The uncoded 8-PSK minimum bandwidth requirement is 3200 Hz (see Table 9.1), and the allowable channel bandwidth is specified as 4000 Hz. Therefore, the uncoded signal bandwidth may be increased by *no more than* a factor of 1.25 (or an expansion of 25%). Thus, the very first step in this (simplified) code selection example is to eliminate the candidates from Table 9.2 that would expand the bandwidth by more than 25%. The remaining entries in Table 9.2 form a much reduced set of "bandwidth-compatible" codes, which have been listed in Table 9.3. In Table 9.3, two columns designated Coding Gain, G , have been added, where coding gain in decibels is defined as

$$G(\text{dB}) = \left(\frac{E_b}{N_0} \right)_{\text{uncoded}} (\text{dB}) - \left(\frac{E_b}{N_0} \right)_{\text{coded}} (\text{dB}) \quad (9.32)$$

From Equation (9.32), coding gain can be described as a measure of the *reduction* in the required E_b/N_0 (in decibels) that needs to be provided, due to the error-

TABLE 9.3 Bandwidth-Compatible BCH Codes

n	k	t	Coding Gain, G (dB), with MPSK	
			$P_B = 10^{-5}$	$P_B = 10^{-9}$
31	26	1	1.8	2.0
63	57	1	1.8	2.2
	51	2	2.6	3.2
127	120	1	1.7	2.2
	113	2	2.6	3.4
	106	3	3.1	4.0

performance properties of the channel coding. Coding gain is a function of the modulation type and bit-error probability. In Table 9.3, the coding gain G has been computed for MPSK at $P_B = 10^{-5}$ and 10^{-9} . For MPSK modulation, G is relatively independent of the value of M . Thus, for a particular bit-error probability, a given code will provide approximately the same coding gain when used with any of the MPSK modulation schemes. The coding gains in Table 9.3 were calculated using a procedure outlined under the section below, entitled "Calculating Coding Gain."

Figure 9.9 illustrates a block diagram that summarizes the details of this system containing a modulator/demodulator (MODEM) and coding. When comparing Figure 9.9 with Figure 9.8, we see that the introduction of the encoder/decoder blocks has brought about additional transformations. In Figure 9.9, at the encoder/modulator are shown the relationships that exist when transforming from R bit/s to R_c channel-bit/s to R_s symbol/s.

We assume that our communication system is a real-time system and thus cannot tolerate any message delay. Therefore, the channel-bit rate R_c must exceed the data-bit rate R by the factor n/k . Further, each transmission symbol is made up of $(\log_2 M)$ channel bits, so the symbol rate R_s is less than R_c by the factor $(\log_2 M)$. For a system containing both modulation and coding, we summarize the rate transformations, as follows:

$$R_c = \left(\frac{n}{k}\right) R \quad (9.33)$$

$$R_s = \frac{R_c}{\log_2 M} \quad (9.34)$$

At the demodulator/decoder shown in Figure 9.9, the transformations amongst data-bit energy, channel-bit energy, and symbol energy are related (in a reciprocal

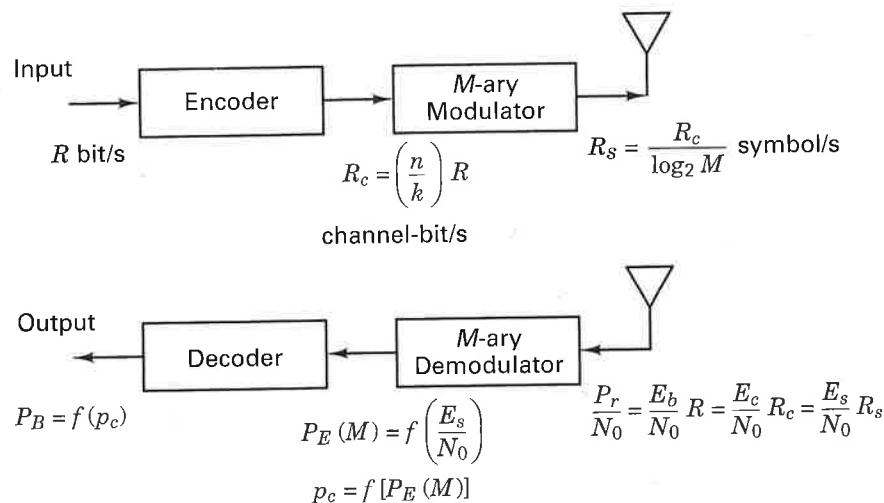


Figure 9.9: MODEM with channel coding.

fashion) by the same factors as shown amongst the rate transformations in Equations (9.33) and (9.34). Since the encoding transformation has replaced k data bits with n channel bits, then the ratio of channel-bit energy to noise-power spectral density, E_c/N_0 , is computed by decrementing the value of E_b/N_0 by the factor k/n . Also, since each transmission symbol is made up of $(\log_2 M)$ channel bits, then E_s/N_0 , which is needed in Equation (9.25) to solve for P_E , is computed by incrementing E_c/N_0 by the factor $(\log_2 M)$. For a system containing both modulation and coding, we summarize the energy-to-noise-power spectral density transformations, as follows:

$$\frac{E_c}{N_0} = \left(\frac{k}{n}\right) \frac{E_b}{N_0} \quad (9.35)$$

$$\frac{E_s}{N_0} = (\log_2 M) \frac{E_c}{N_0} \quad (9.36)$$

Therefore, using Equations (9.33) through (9.36), we can now expand the expression for P_r/N_0 in Equation (9.24), as follows:

$$\frac{P_r}{N_0} = \frac{E_b}{N_0} R = \frac{E_c}{N_0} R_c = \frac{E_s}{N_0} R_s \quad (9.37)$$

As before, a standard way of describing the link is in terms of the received E_b/N_0 in decibels. However, *there are no data bits* at the input to the demodulator/detector; *neither are there any channel bits*. There are *only* waveforms (transmission symbols) that have bit meanings, and thus the waveforms can be described in terms of bit-energy apportionments. Equation (9.37) illustrates that the predetection point in the receiver is a useful reference point at which we can relate the *effective* energy and the *effective* speed of various parameters of interest. We use the word “effective” because the only type of signals that actually appear at the predetection point are waveforms (transformed to baseband pulses) that we call symbols. Of course, these symbols are related to channel bits, which in turn are related to data bits. To emphasize the point that Equation (9.37) represents a useful kind of “bookkeeping,” consider a system wherein a stream of some number of bits, say 273 bits, appears so repeatedly as a module, that we give this group 273 bits a name; we all it a “chunk.” Engineers do that all the time—e.g., eight bits are referred to as a byte. The moment we identify this new entity, the chunk, it can immediately be related to the parameters in Equation (9.37), since P_r/N_0 will now also equal the energy in a chunk over N_0 times the chunk rate. In Chapter 12, we will in fact do something similar when we extend Equation (9.37) to include spread-spectrum *chips*.

Since P_r/N_0 and R were given as 53 dB-Hz and 9600 bits/s, respectively, we find as before from Equation (9.23) that the received $E_b/N_0 = 13.2$ dB. Note that the received E_b/N_0 is fixed and independent of the code parameters n and k , and the modulation parameter M . As we search, in Table 9.3, for the ideal code that will meet the specifications, we can iteratively repeat the computations that are summarized in Figure 9.9. It might be useful to program on a PC (or calculator) the following four steps as a function of n , k , and t . Step 1 starts by combining Equations (9.35) and (9.36), as follows:

$$\text{Step 1:} \quad \frac{E_s}{N_0} = (\log_2 M) \frac{E_c}{N_0} = (\log_2 M) \left(\frac{k}{n} \right) \frac{E_b}{N_0} \quad (9.38)$$

$$\text{Step 2:} \quad P_E(M) \approx 2Q \left[\sqrt{\frac{2 E_s}{N_0}} \sin \left(\frac{\pi}{M} \right) \right] \quad (9.39)$$

The expression in step 2 is the approximation (for M -ary PSK) for symbol-error probability P_E , rewritten from Equation (9.25). At each symbol-time interval, the demodulator makes a symbol decision, but it delivers to the decoder a channel-bit sequence representing that symbol. When the channel-bit output of the demodulator is quantized to two levels, denoted by 1 and 0, the demodulator is said to make *hard decisions*. When the output is quantized to more than two levels, the demodulator is said to make *soft decisions*. Throughout this section, hard-decision demodulation is assumed.

Now that a decoder block is present in the system, we designate the channel-bit-error probability out of the demodulator and into the decoder as p_c , and reserve the notation P_B for the bit-error probability *out of the decoder* (the decoded bit-error probability). Equation (9.27) is rewritten in terms of p_c as follows:

$$\text{Step 3:} \quad p_c \approx \frac{P_E}{\log_2 M} \quad (\text{for } P_E \ll 1) \quad (9.40)$$

Step 3 relates the channel-bit-error probability to the symbol-error probability out of the demodulator, assuming Gray coding, as referenced in Equation (9.27).

For a real-time communication system, using traditional channel-coding schemes, and a given value of received P_r/N_0 , the value of E_s/N_0 with coding will *always be less* than the value of E_s/N_0 without coding. Since the demodulator, with coding, receives less E_s/N_0 it makes more errors! However, when coding is used, the system error-performance doesn't only depend on the performance of the demodulator, it also depends on the performance of the decoder. Thus, for error-performance improvement due to coding, we require that the decoder provides enough error correction to *more than compensate* for the poor performance of the demodulator. The final output decoded bit-error probability P_B depends on the particular code, the decoder, and the channel-bit-error probability p_c . It can be expressed [15] by the following approximation:

$$\text{Step 4:} \quad P_B \approx \frac{1}{n} \sum_{j=t+1}^n j \binom{n}{j} p_c^j (1 - p_c)^{n-j} \quad (9.41)$$

In Step 4, t is the largest number of channel bits that the code can correct within each block of n bits. Using Equations (9.38) through (9.41) in the above four steps, the decoded bit-error probability P_B can be computed as a function of n , k , and t for each of the codes listed in Table 9.3. The entry that meets the stated error requirement with the *largest possible* code rate and the *smallest* value of n is the double-error correcting (63, 51) code. The computations are as follows:

$$\text{Step 1:} \quad \frac{E_s}{N_0} = 3 \left(\frac{51}{63} \right) 20.89 = 50.73$$

where $M = 8$, and the received $E_b/N_0 = 13.2$ dB (or 20.89)

$$\text{Step 2: } P_E \approx 2Q \left[\sqrt{101.5} \times \sin \left(\frac{\pi}{8} \right) \right] = 2Q(3.86) = 1.2 \times 10^{-4}$$

$$\text{Step 3: } p_c \approx \frac{1.2 \times 10^{-4}}{3} = 4 \times 10^{-5}$$

$$\begin{aligned} \text{Step 4: } P_B &\approx \frac{3}{63} \binom{63}{3} (4 \times 10^{-5})^3 (1 - 4 \times 10^{-5})^{60} \\ &\quad + \frac{4}{63} \binom{63}{4} (4 \times 10^{-5})^4 (1 - 4 \times 10^{-5})^{59} + \dots \\ &= 1.2 \times 10^{-10} \end{aligned}$$

In Step 4, the bit-error-correcting capability of the code is $t = 2$. For the computation of P_B in Step 4, only the first two terms in the summation of Equation (9.41) have been used, since the other terms have a vanishingly small effect on the result whenever p_c is small or E_b/N_0 reasonably large. It is important to note that when performing this computation with a computer, it is advised (for being safe) to *always* include all of the summation terms in Equation (9.41), since a truncated solution can be very erroneous whenever E_b/N_0 is small. Now that we have selected the (63, 51) code, the values of channel-bit rate R_c and symbol rate R_s are computed using Equations (9.33) and (9.34), with $M = 8$:

$$R_c = \left(\frac{n}{k} \right) R = \left(\frac{63}{51} \right) 9600 \approx 11,859 \text{ channel-bits/s}$$

$$R_s = \frac{R_c}{\log_2 M} = \frac{11,859}{3} = 3953 \text{ symbols/s}$$

9.7.7.1 Calculating Coding Gain

A more direct way to find the simplest code that meets the specified error performance for the example in Section 9.7.7 is to first compute, for the *uncoded* 8-PSK, how much more E_b/N_0 beyond the available 13.2 dB would be required to yield $P_B = 10^{-9}$. This additional E_b/N_0 is the required coding gain. Next, we simply choose, from Table 9.3, the code that provides this coding gain. The *uncoded* E_s/N_0 that will yield an error probability of $P_B = 10^{-9}$ is found by writing, from Equations (9.27) and (9.39),

$$P_B \approx \frac{P_E}{\log_2 M} \approx \frac{2Q \left[\sqrt{\frac{2E_s}{N_0}} \sin \left(\frac{\pi}{M} \right) \right]}{\log_2 M} = 10^{-9} \quad (9.42)$$

At this low value of bit-error probability, it is valid to use Equation (3.44) to approximate $Q(x)$ in Equation (9.42). By trial-and-error (on a programmable calculator), we find that the *uncoded* $E_s/N_0 = 120.67 = 20.8$ dB, and since each symbol is made up of $(\log_2 8) = 3$ bits, the required $(E_b/N_0)_{\text{uncoded}} = 120.67/3 = 40.22 = 16$ dB.

We know from the given parameters in this example and Equation (9.23), that the received $(E_b/N_0)_{\text{coded}} = 13.2$ dB. Therefore, using Equation (9.32), we see that the required coding gain to meet the bit-error performance of $P_B = 10^{-9}$ is

$$G(\text{dB}) = \left(\frac{E_b}{N_0} \right)_{\text{uncoded}} (\text{dB}) - \left(\frac{E_b}{N_0} \right)_{\text{coded}} (\text{dB}) = 16 \text{ dB} - 13.2 \text{ dB} = 2.8 \text{ dB}$$

To be precise in this computation, each of the E_b/N_0 values in the above computation must correspond to exactly the same value of bit-error probability (which they do not). They correspond to $P_B = 10^{-9}$ and $P_B = 1.2 \times 10^{-10}$, respectively. However, at these low probability values, even with such a discrepancy, this computation still provides a good approximation of the required coding gain. In searching Table 9.3 for the simplest code that will yield a coding gain of *at least* 2.8 dB, we see that the choice is the (63, 51) code, which corresponds to the same code choice that was made earlier. Note that coding gain must always be specified for a particular error probability and modulation type, as it is in Table 9.3.

9.7.7.2 Code Selection

Consider a real-time communication system, where the specifications cause it to be power-limited, but there is ample available bandwidth, and the users require a very small bit-error probability. The situation calls for error-correction coding. Suppose that we were asked to select one of the BCH codes listed in Table 9.2. Since the system is not bandwidth-limited, and it requires very good error performance, one might be tempted to simply choose the most powerful code in Table 9.2, that is the (127, 8) code, capable of correcting any combination of up to 31 flawed bits within a block of 127 code bits. Would anyone use such a code in a real-time communication system? No, they would not. Let us explain why such a choice would be *unwise*.

Whenever error-correction coding is used and E_b/N_0 is fixed, there are two mechanisms at work that influence error performance. One mechanism works to improve the performance, and the other works to degrade it. The improving mechanism is the coding; the greater redundancy, the greater will be the error-correcting capability of the code. The degrading mechanism is the energy reduction per channel symbol or code bit (compared with the data bit). This reduced energy stems from the increased redundancy (giving rise to faster signaling in a real-time communication system). The reduced symbol energy causes the demodulator to make more errors. Eventually, the second mechanism wins out, and thus at very low code rates we see degradation. This is demonstrated in Example 9.4 below. Note that the degrading mechanism applies for coding in a real-time system (where messages cannot be delayed). For systems with fixed power and extended transmission time (i.e., delay), there is no degradation with reduced code rate since there is no reduction in channel-symbol energy.

Example 9.4 Choosing a Code to Meet Performance Requirements

A system is specified with the following parameters: $P_r/N_0 = 67$ dB-Hz, data rate $R = 10^6$ bits/s, available bandwidth $W = 20$ MHz, decoded bit-error probability $P_B \leq 10^{-7}$,

and the modulation is BPSK. Choose a code from Table 9.2 that will fulfill these requirements. Start by considering the (127, 8) code. It appears attractive because it has the greatest bit-error correcting capability on the list.

Solution

The (127, 8) code expands the transmission bandwidth by a factor of $127/8 = 15.875$. Hence, the signaling rate of 1 Mbit/s (giving rise to a nominal bandwidth of 1 MHz) will be expanded by using this code to 15.875 MHz. The transmission signal is within the available bandwidth of 20 MHz, even after allowing another 25% bandwidth expansion for filtering. After choosing this code, we next evaluate the error performance, by following the steps outlined in Section 9.7.7, which yields

$$\frac{E_b}{N_0} = \frac{P_r}{N_0} \left(\frac{1}{R} \right) = 67 \text{ dB} - 60 \text{ dB} = 7 \text{ dB (or 5)}$$

$$\frac{E_s}{N_0} = \frac{E_c}{N_0} = \left(\frac{k}{n} \right) \frac{E_b}{N_0} = \left(\frac{8}{127} \right) 5 = 0.314$$

Since the modulation is binary, then $p_c = P_E$, so that

$$p_c = P_E \approx Q\left(\sqrt{\frac{2E_s}{N_0}}\right) = Q(\sqrt{0.628}) = Q(0.7936) = 0.2156$$

Since the (127, 8) code is a $t = 31$ error-correcting code, we next use Equation (9.41) to find the decoded bit-error probability, as follows:

$$P_B \approx \frac{1}{n} \sum_{j=t+1}^n j \binom{n}{j} p_c^j (1 - p_c)^{n-j} = \frac{1}{127} \sum_{j=32}^{127} j \binom{127}{j} (0.2156)^j (1 - 0.2156)^{127-j}$$

Whenever p_c is very small, it suffices to only use the first term, or the first few terms in the summation. But when p_c is large, as it is here, computer assistance is helpful. Solving the above with $p_c = 0.2156$ yields a decoded bit-error probability of $P_B = 0.05$, which is a far cry from the system requirement of 10^{-7} . Let us next select a code whose code rate is close to the popular rate $\frac{1}{2}$ —that is, the (127, 64) code. It is not as capable as the first choice because it only corrects 10 flawed bits in a block of 127 code bits. However, watch what happens. Using the same steps as before yields

$$\frac{E_s}{N_0} = \frac{E_c}{N_0} = \left(\frac{k}{n} \right) \frac{E_b}{N_0} = \left(\frac{64}{127} \right) 5 = 2.519$$

Notice how much larger the E_s/N_0 is here, compared with the case using (127, 8) coding:

$$p_c = Q(\sqrt{2 \times 2.519}) = Q(2.245) = 0.0124$$

$$P_B \approx \frac{1}{127} \sum_{j=11}^{127} j \binom{127}{j} (0.0124)^j (1 - 0.0124)^{127-j}$$

And the result yields $P_B = 5.6 \times 10^{-8}$, which meets the system requirements. From this example, one should see that the selection of a code needs to be made in concert with the modulation choice and the available E_b/N_0 . One can be guided by the fact that very high rates and very low rates generally perform poorly in a real-time communication system, as evidenced by the Figure 8.6 curves presented in Chapter 8.

9.8 BANDWIDTH-EFFICIENT MODULATION

The primary objective of spectrally efficient modulation techniques is to maximize bandwidth efficiency. The increasing demand for digital transmission channels has led to the investigation of spectrally efficient modulation techniques [8, 16] to maximize bandwidth efficiency and thus help ameliorate the spectral congestion problem.

Some systems have additional modulation requirements besides spectral efficiency. For example, satellite systems with highly nonlinear transponders require a constant envelope modulation. This is because the nonlinear transponder produces extraneous sidebands when passing a signal with amplitude fluctuations (due to a mechanism called AM-to-PM conversion). These sidebands deprive the information signals of some of their portion of transponder power, and also can interfere with nearby channels (adjacent channel interference) or with other communication systems (co-channel interference). *Offset QPSK* (OQPSK) and *Minimum shift keying* (MSK) are two examples of constant envelope modulation schemes that are attractive for systems using nonlinear transponders.

9.8.1 QPSK and Offset QPSK Signaling

Figure 9.10 illustrates the partitioning of a typical pulse stream for QPSK modulation. Figure 9.10a shows the original data stream $d_k(t) = d_0, d_1, d_2, \dots$ consisting of bipolar pulses; that is, the values of $d_k(t)$ are +1 or -1, representing binary one and zero, respectively. This pulse stream is divided into an in-phase stream, $d_I(t)$, and a quadrature stream, $d_Q(t)$, illustrated in Figure 9.10b, as follows:

$$d_I(t) = d_0, d_2, d_4, \dots \text{ (even bits)} \quad (9.43)$$

$$d_Q(t) = d_1, d_3, d_5, \dots \text{ (odd bits)}$$

Note that $d_I(t)$ and $d_Q(t)$ each have half the bit rate of $d_k(t)$. A convenient orthogonal realization of a QPSK waveform, $s(t)$, is achieved by amplitude modulating the in-phase and quadrature data streams onto the cosine and sine functions of a carrier wave, as follows:

$$s(t) = \frac{1}{\sqrt{2}} d_I(t) \cos \left(2\pi f_0 t + \frac{\pi}{4} \right) + \frac{1}{\sqrt{2}} d_Q(t) \sin \left(2\pi f_0 t + \frac{\pi}{4} \right) \quad (9.44)$$

Using the trigonometric identities shown in Equations (D.5) and (D.6), Equation (9.44) can also be written as

$$s(t) = \cos [2\pi f_0 t + \theta(t)] \quad (9.45)$$

The QPSK modulator shown in Figure 9.10c uses the sum of cosine and sine terms, while a similar device, described in Section 4.6, uses the difference of such terms. The treatment in this section corresponds to that of Pasupathy [17]. Because a coherent receiver needs to resolve any phase ambiguities, then the use of a different phase format at the transmitter can be handled as part of that ambiguity. The pulse stream $d_I(t)$ amplitude-modulates the cosine function with an amplitude of +1 or

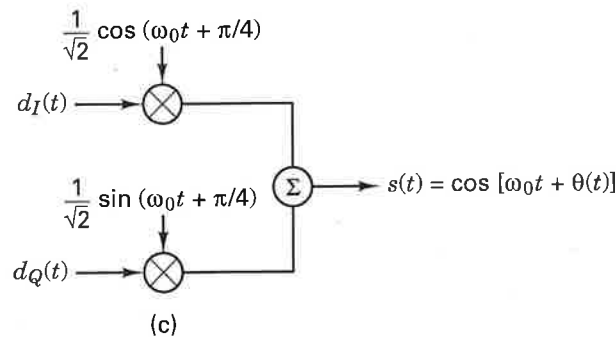
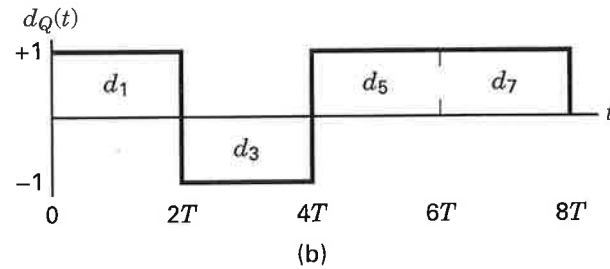
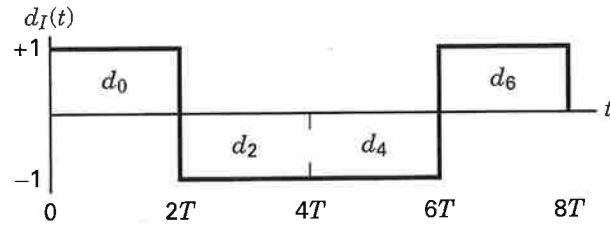
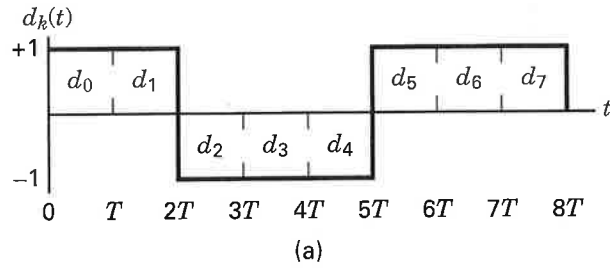


Figure 9.10 QPSK modulation.

-1. This is equivalent to shifting the phase of the cosine function by 0 or π ; consequently, this produces a BPSK waveform. Similarly, the pulse stream $d_Q(t)$ modulates the sine function, yielding a BPSK waveform orthogonal to the cosine function. The summation of these two orthogonal components of the carrier yields the QPSK waveform. The value of $\theta(t)$ will correspond to one of the four possible combinations of $d_I(t)$ and $d_Q(t)$ in Equation 9.44: $\theta(t) = 0^\circ, \pm 90^\circ$, or 180° ; the resulting signal vectors are seen in the signal space illustrated in Figure 9.11. Because

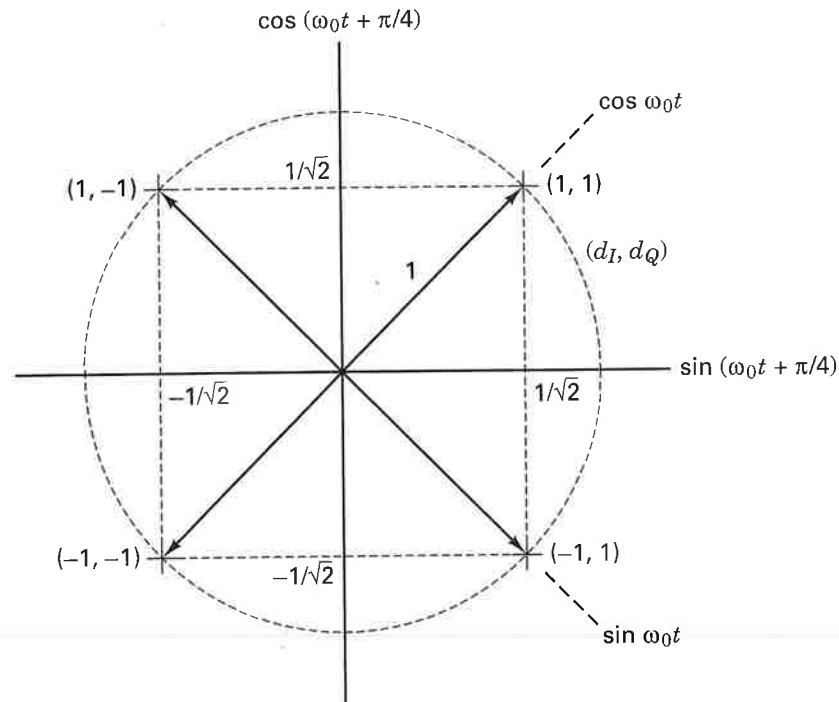


Figure 9.11 Signal space for QPSK and OQPSK.

$\cos(2\pi f_0 t + \pi/4)$ and $\sin(2\pi f_0 t + \pi/4)$ are orthogonal, the two BPSK signals can be detected separately.

Offset QPSK (OQPSK) signaling can also be represented by Equations (9.44) and (9.45); the difference between the two modulation schemes, QPSK and OQPSK, is only in the *alignment* of the two baseband waveforms. As shown in Figure 9.10, the duration of each original pulse is T (Figure 9.10a), and hence in the partitioned streams of Figure 9.10b, the duration of each pulse is $2T$. In standard QPSK, the odd and even pulse streams are both transmitted at the rate of $1/2T$ bit/s and are synchronously aligned, such that their transitions coincide, as shown in Figure 9.10b. In OQPSK, sometimes called *staggered QPSK* (SQPSK), there is the same data stream partitioning and orthogonal transmission; the difference is that the timing of the pulse stream $d_I(t)$ and $d_Q(t)$ is shifted such that the alignment of the two streams is offset by T . Figure 9.12 illustrates this offset.

In standard QPSK, due to the coincident alignment of $d_I(t)$ and $d_Q(t)$, the carrier phase can change only once every $2T$. The carrier phase during any $2T$ interval can be any one of the four phases shown in Figure 9.11, depending on the values of $d_I(t)$ and $d_Q(t)$ during that interval. During the next $2T$ interval, if neither pulse stream changes sign, the carrier phase remains the same. If only one of the pulse streams change sign, a phase shift of $\pm 90^\circ$ occurs. A change in both streams results in a carrier phase shift of 180° . Figure 9.13a shows a typical QPSK waveform for the sample sequence $d_I(t)$ and $d_Q(t)$ shown in Figure 9.10.

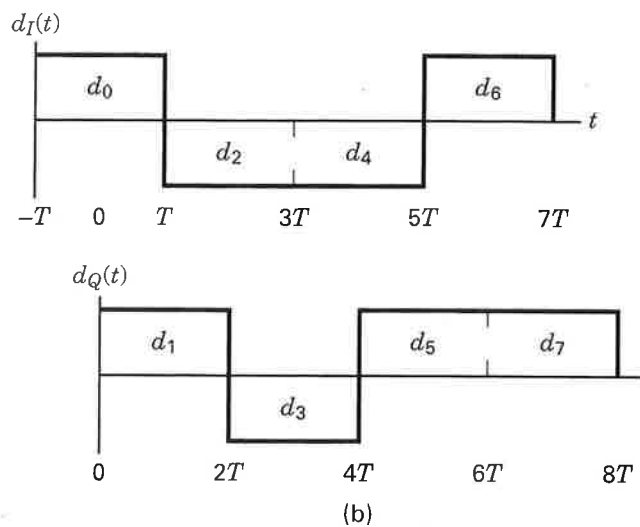


Figure 9.12 Offset QPSK (OQPSK) data streams.

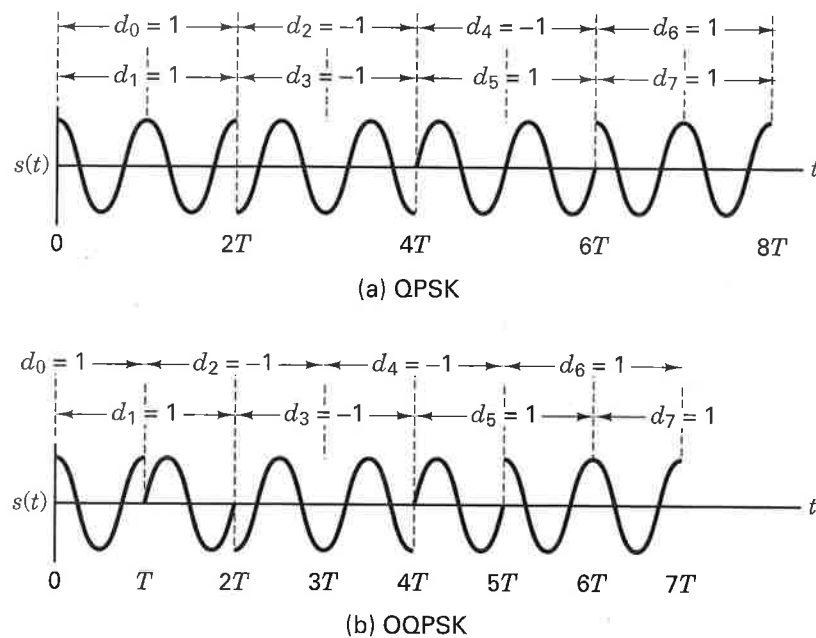


Figure 9.13 (a) QPSK and (b) OQPSK waveforms. (Reprinted with permission from S. Pasupathy, "Minimum Shift Keying: A Spectrally Efficient Modulation," *IEEE Commun. Mag.*, July 1979, Fig. 4, p. 17. © 1979 IEEE.)

If a QPSK modulated signal undergoes filtering to reduce the spectral side-lobes, the resulting waveform will no longer have a constant envelope and in fact, the occasional 180° phase shifts will cause the envelope to go to zero momentarily (see Figure 9.13a). When these signals are used in satellite channels employing highly nonlinear amplifiers, the constant envelope will tend to be restored. However, at the same time, all of the *undesirable* frequency side-lobes, which can interfere with nearby channels and other communication systems, are also restored.

In OQPSK, the pulse streams $d_I(t)$ and $d_Q(t)$ are staggered and thus do not change states simultaneously. The possibility of the carrier changing phase by 180° is eliminated, since only one component can make a transition at one time. Changes are limited to 0° and $\pm 90^\circ$ every T seconds. Figure 9.13b shows a typical OQPSK waveform for the sample sequence in Figure 9.12. When an OQPSK signal undergoes bandlimiting, the resulting intersymbol interference causes the envelope to droop slightly in the region of $\pm 90^\circ$ phase transition, but since the phase transitions of 180° have been avoided in OQPSK, the envelope will not go to zero as it does with QPSK. When the bandlimited OQPSK goes through a nonlinear transponder, the envelope droop is removed; however, the high-frequency components associated with the collapse of the envelope are not reinforced. Thus out-of-band interference is avoided [17].

9.8.2 Minimum Shift Keying

The main advantage of OQPSK over QPSK, that of suppressing out-of-band interference, suggests that further improvement is possible if the OQPSK format is modified to avoid discontinuous phase transitions. This was the motivation for designing continuous phase modulation (CPM) schemes. *Minimum shift keying* (MSK) is one such scheme [17–20]. MSK can be viewed as either a special case of *continuous-phase frequency shift keying* (CPFSK), or a special case of OQPSK with sinusoidal symbol weighting. When viewed as CPFSK, the MSK waveform can be expressed as [18]

$$s(t) = \cos \left[2\pi \left(f_0 + \frac{d_k}{4T} \right) t + x_k \right] \quad kT < t < (k+1)T \quad (9.46)$$

where f_0 is the carrier frequency, $d_k = \pm 1$ represents the bipolar data being transmitted at a rate $R = 1/T$, and x_k is a phase constant which is valid over the k th binary data interval. Notice that for $d_k = 1$, the frequency transmitted is $f_0 + 1/4T$, and for $d_k = -1$, the frequency transmitted is $f_0 - 1/4T$. The tone spacing in MSK is thus one-half that employed for noncoherently demodulated orthogonal FSK, giving rise to the name *minimum* shift keying. During each T -second data interval, the value of x_k is a constant, that is, $x_k = 0$ or π , determined by the requirement that the phase of the waveform be continuous at $t = kT$. This requirement results in the following recursive phase constraint for x_k :

$$x_k = \left[x_{k-1} + \frac{\pi k}{2} (d_{k-1} - d_k) \right] \text{ modulo } 2\pi \quad (9.47)$$

Equation (9.46) can be expressed in a quadrature representation, using the identities in Equation (D.5) and (D.6), and we write

$$s(t) = a_k \cos \frac{\pi t}{2T} \cos 2\pi f_0 t - b_k \sin \frac{\pi t}{2T} \sin 2\pi f_0 t \quad kT < t < (k+1)T \quad (9.48)$$

where

$$a_k = \cos x_k = \pm 1 \quad (9.49)$$

$$b_k = d_k \cos x_k = \pm 1$$

The in-phase (I) component is identified as $a_k \cos (\pi t/2T) \cos 2\pi f_0 t$, where $\cos 2\pi f_0 t$ is the carrier, $\cos (\pi t/2T)$ can be regarded as a *sinusoidal symbol weighting*, and a_k is a data-dependent term. Similarly, the quadrature (Q) component is identified as $b_k \sin (\pi t/2T) \sin 2\pi f_0 t$, where $\sin 2\pi f_0 t$ is the quadrature carrier term, $\sin (\pi t/2T)$ can be regarded as a sinusoidal symbol weighting, and b_k is a data-dependent term. It might appear that the a_k and b_k terms can change every T seconds, since the source data d_k can change every T seconds. However, because of the continuous phase constraint, the a_k term can only change value at the zero crossings of $\cos (\pi t/2T)$ and the b_k term can only change value at the zero crossings of $\sin (\pi t/2T)$. Thus, the symbol weighting in either the I - or Q -channel is a half-cycle sinusoidal pulse of duration $2T$ seconds with alternating sign. As in the case of OQPSK, the I and Q components are offset T seconds with respect to one another.

Notice that x_k in Equation (9.46) is a function of the difference between the prior data bit and the present data bit (differential encoding). Hence the a_k and b_k terms in Equation (9.48) can be viewed as *differentially encoded* components of the d_k source data. However, for bit-to-bit independent data d_k , the signs of successive I - or Q -channel pulses are also random from one $2T$ -second pulse interval to the next. Thus when viewed as a special case of OQPSK, Equation (9.48) can be rewritten with more straightforward (nondifferential) data encoding [18] as follows:

$$s(t) = d_I(t) \cos \frac{\pi t}{2T} \cos 2\pi f_0 t + d_Q(t) \sin \frac{\pi t}{2T} \sin 2\pi f_0 t \quad (9.50)$$

where $d_I(t)$ and $d_Q(t)$ have the same in-phase and quadrature data stream interpretation as in Equation (9.43). This MSK format in Equation (9.50) is sometimes referred to as *precoded MSK*. Figure 9.14 illustrates Equation (9.50) pictorially. Figure 9.14a and c show the sinusoidal weighting of the I - and Q -channel pulses. These sequences represent the same data sequences as in Figure 9.12, but here, multiplication by a sinusoid results in more gradual phase transitions compared to those of the original data representation. Figure 9.14b and d illustrate the modulation of the orthogonal components $\cos 2\pi f_0 t$ and $\sin 2\pi f_0 t$, respectively, by the sinusoidally shaped data streams. Figure 9.14e illustrates the summation of the orthogonal components from Figure 9.14b and d. In summary, the following properties of MSK modulation can be deduced from Equation (9.50) and Figure 9.14:

on, using the

$$T \quad (9.48)$$

$$(9.49)$$

here $\cos 2\pi f_0 t$ ghting, and a_k s identified as m, $\sin(\pi t/2T)$ pendent term. nds, since the he continuous ossings of $\cos$ of $\sin(\pi t/2T)$. rcle sinusoidal f OQPSK, the

e between the the a_k and b_k ponents of the s of successive interval to the (9.48) can be oding [18] as

$$(9.50)$$

ream interpre-sometimes re-50) pictorially. hannel pulses. 9.12, but here, s compared to te the modula-ly, by the sinu-nation of the ollowing prop-Id Figure 9.14:

ffs Chap. 9

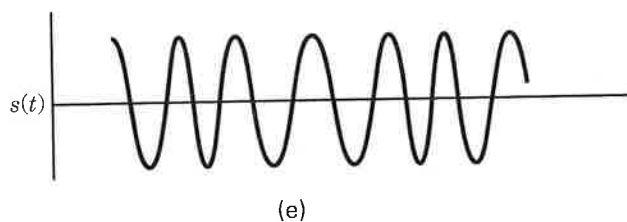
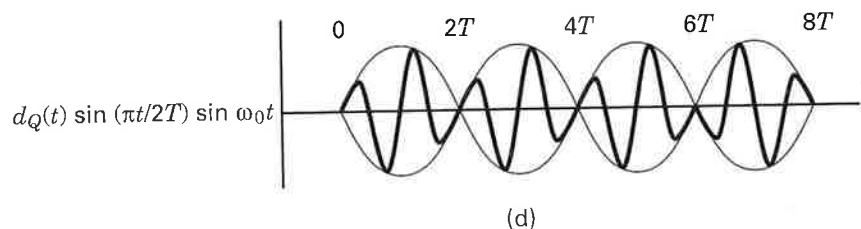
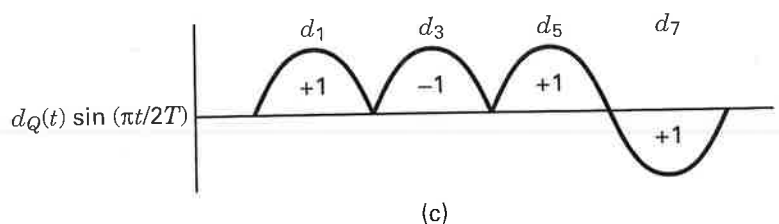
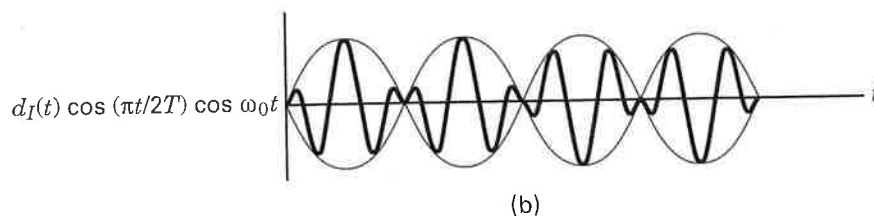
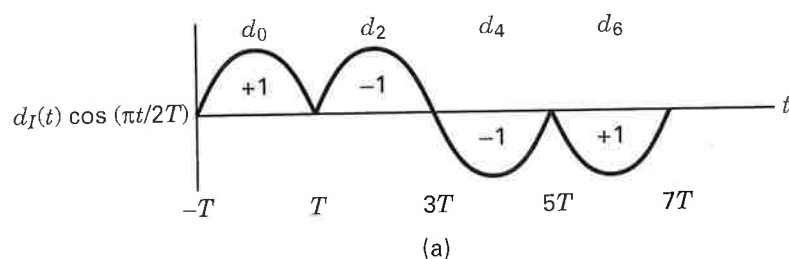


Figure 9.14 Minimum shift keying (MSK). (a) Modified I bit stream. (b) I bit stream times carrier. (c) Modified Q bit stream. (d) Q bit stream times carrier. (e) MSK waveform. (Reprinted with permission from S. Pasupathy, "Minimum Shift Keying: A Spectrally Efficient Modulation," *IEEE Commun. Mag.*, July 1979, Fig. 5, p. 18. © 1979 IEEE.)

(1) the waveform $s(t)$ has constant envelope; (2) there is phase continuity in the RF carrier at the bit transitions; and (3) the waveform $s(t)$ can be regarded as an FSK waveform with signaling frequencies $f_0 + 1/4T$ and $f_0 - 1/4T$. Therefore, the minimum tone separation required for MSK modulation is

$$\left(f_0 + \frac{1}{4T}\right) - \left(f_0 - \frac{1}{4T}\right) = \frac{1}{2T} \quad (9.51)$$

which is equal to half the bit rate. Notice that the required tone spacing for MSK is one-half the spacing, $1/T$, required for the noncoherent detection of FSK signals (see Section 4.5.4). This is because the carrier phase is known and continuous, enabling the signal to be coherently demodulated.

The power spectral density $G(f)$ for QPSK and OQPSK is given by [18]

$$G(f) = 2PT \left(\frac{\sin 2\pi fT}{2\pi fT} \right)^2 \quad (9.52)$$

where P is the average power in the modulated waveform. For MSK, $G(f)$ is given by [18]

$$G(f) = \frac{16PT}{\pi^2} \left(\frac{\cos 2\pi fT}{1 - 16f^2T^2} \right)^2 \quad (9.53)$$

The normalized power spectral density ($P = 1$ W) for QPSK, OQPSK, and MSK are sketched in Figure 9.15. A spectral plot of BPSK is included for comparison. The fact that BPSK requires more bandwidth than the others for a given level of spectral density should come as no surprise. In Section 9.5.1 and Figure 9.6 we saw that the theoretical bandwidth efficiency of BPSK is half that of QPSK. It is seen from Figure 9.15 that MSK has lower sidelobes than QPSK or OQPSK. This is a consequence of multiplying the data stream with a sinusoid, yielding more *gradual phase transitions*. The more gradual the transition, the faster the spectral tails drop to zero. MSK is *spectrally more efficient* than QPSK or OQPSK; however, as can be seen from Figure 9.15, the MSK spectrum has a wider mainlobe than QPSK and OQPSK. Therefore, MSK may not be the preferred method for narrowband links. However, MSK might be the preferred choice for multiple-carrier systems, because its relatively low spectral sidelobes help to avoid excessive adjacent channel interference (ACI). The reason for the QPSK spectrum having a narrower mainlobe than MSK is that, for a given bit rate, the QPSK symbol rate is half the MSK symbol rate.

9.8.2.1 Error Performance of OQPSK and MSK

We have seen that BPSK and QPSK have the same bit-error probability because QPSK is configured as two BPSK signals modulating orthogonal components of the carrier. Since staggering the bit streams does not change the orthogonality of the carriers, OQPSK has the same theoretical bit error performance as BPSK and QPSK.

Minimum shift keying uses antipodal symbol shapes, $\pm \cos(\pi t/2T)$ and $\pm \sin(\pi t/2T)$, over $2T$ to modulate the two quadrature components of the carrier. Thus

ity in the RF
ed as an FSK
re, the mini-

(9.51)

ig for MSK is
FSK signals
continuous,

by [18]

(9.52)

MSK, $G(f)$ is

(9.53)

SK, and MSK
r comparison.
given level of
re 9.6 we saw
SK. It is seen
PSK. This is a
more gradual
tral tails drop
ver, as can be
an QPSK and
owband links.
tems, because
channel inter-
wer mainlobe
half the MSK

probability be-
al components
thogonality of
as BPSK and

$/2T)$ and $\pm \sin$
carrier. Thus

ffs Chap. 9

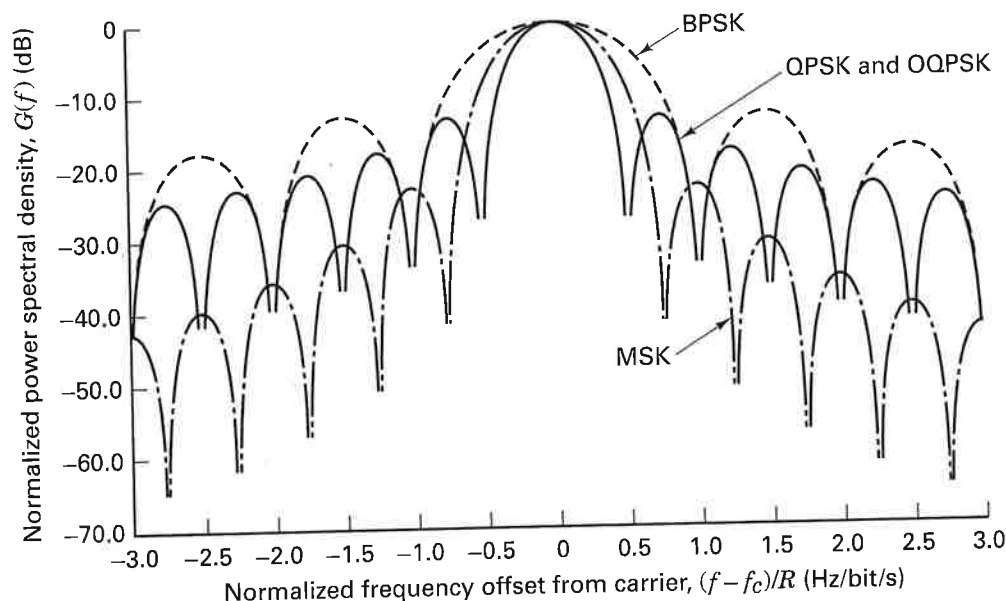


Figure 9.15 Normalized power spectral density for BPSK, QPSK, OQPSK, and MSK. (Reprinted with permission from F. Amoroso, "The Bandwidth of Digital Data Signals," *IEEE Commun. Mag.*, vol. 18, no. 6, Nov. 1980, Fig. 2A, p. 16. © 1980 IEEE.)

when a matched filter is used to recover the data from each of the quadrature components independently, MSK, as defined in Equation (9.50), has the same error performance properties as BPSK, QPSK, and OQPSK [17]. However, if MSK is coherently detected as an FSK signal over an observation interval of T seconds, it would be poorer than BPSK by 3 dB [17]. MSK, with differentially encoded data, as defined in Equation (9.46), has the same error probability performance as the coherent detection of differentially encoded PSK. MSK can also be noncoherently detected [19]. This permits inexpensive demodulation when the value of received E_b/N_0 permits.

9.8.3 Quadrature Amplitude Modulation

Coherent M -ary phase shift keying (MPSK) modulation is a well-known technique for achieving bandwidth reduction. Instead of using a binary alphabet with 1 bit of information per channel symbol period, an alphabet with M symbols is used, permitting the transmission of $k = \log_2 M$ bits during each symbol period. Since the use of M -ary symbols allows a k -fold increase in the data rate within the same bandwidth, then for a fixed data rate, use of M -ary PSK reduces the required bandwidth by a factor k . (See Section 4.8.3.)

From Equation (9.44) it can be seen that QPSK modulation consists of two independent streams. One stream amplitude-modulates the cosine function of a

carrier wave with levels $+1$ and -1 , and the other stream similarly amplitude-modulates the sine function. The resultant waveform is termed a double-sideband suppressed-carrier (DSB-SC) wave, since the RF bandwidth is twice the baseband bandwidth (see Section 1.7.1) and there is no isolated carrier term. *Quadrature amplitude modulation* (QAM) can be considered a logical extension of QPSK, since QAM also consists of two independently amplitude-modulated carriers in quadrature. Each block of k bits (k assumed even) can be split into two $(k/2)$ -bit blocks which use $(k/2)$ -bit digital-to-analog (D/A) converters to provide the required modulating voltages for the carriers. At the receiver, each of the two signals is independently detected using matched filters. QAM signaling can also be viewed as a combination of amplitude shift keying (ASK) and phase shift keying (PSK), giving rise to the alternative name, *amplitude phase keying* (APK). Finally, it can also be viewed as amplitude shift keying in two dimensions, giving rise to the name *quadrature amplitude shift keying* (QASK).

Figure 9.16a illustrates a two-dimensional signal space and a set of 16-ary QAM signal vectors or points arranged in a rectangular constellation. A canonical QAM modulator is shown in Figure 9.16b. Assuming that Gaussian noise is the only channel disturbance, the simple channel model of Figure 9.16c applies. Signals are sent in pairs (x, y) . The model indicates that the signal point coordinates (x, y) are transmitted over separate channels and independently perturbed by Gaussian

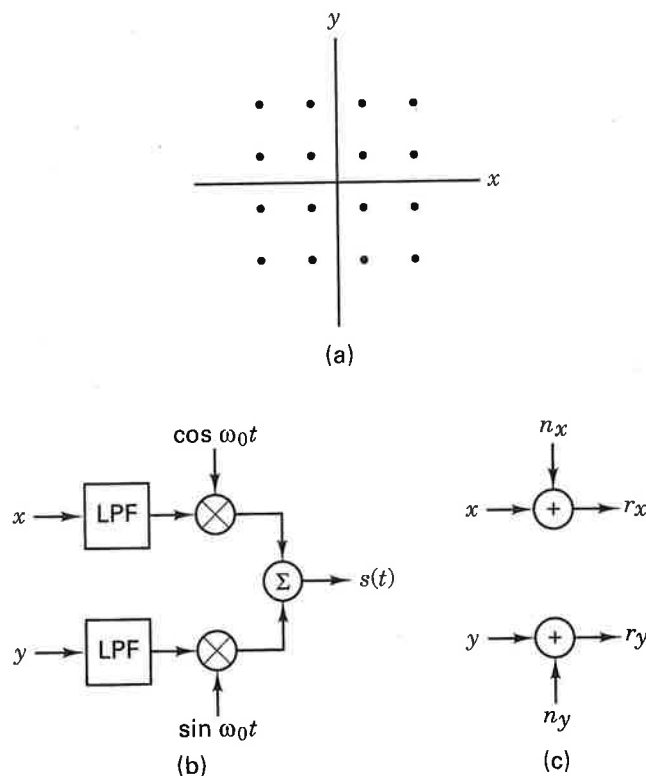


Figure 9.16 QAM modulation. (a) 16-ary signal space. (b) Canonical QAM modulator. (c) QAM channel model.

ly amplitude-
 sideband
 the baseband
 quadrature am-
 PSK, since
 ers in quadra-
 (2)-bit blocks
 the required
 signals is inde-
 e viewed as a
 (PSK), giving
 ly, it can also
 to the name

set of 16-ary
 i. A canonical
 n noise is the
 applies. Signals
 rdinates (x, y)
 l by Gaussian

modulation. (a)
 (b) Canonical
 QAM channel

noise variables (n_x, n_y) , each with zero mean and variance N . Or we can say that the two-dimensional signal point is perturbed by a two-dimensional Gaussian noise variable. If the average signal energy (mean-square value of the signal coordinates) is S , then the signal-to-noise ratio is S/N . The simplest method of digital signaling through such a system is to use one-dimensional pulse amplitude modulation (PAM) independently for each signal coordinate. In PAM, to send k bits/dimension over a Gaussian channel, each signal point coordinate takes on one of 2^k equally likely equispaced amplitudes. By convention, the signal points are grouped about the center of the space at amplitudes $\pm 1, \pm 3, \dots, \pm (2^k - 1)$.

9.8.3.1 QAM Probability of Bit Error

For a rectangular constellation, a Gaussian channel, and matched filter reception, the bit-error probability for M -QAM, where $M = 2^k$ and k is even, is [12]

$$P_B \approx \frac{2(1 - L^{-1})}{\log_2 L} Q \left[\sqrt{\left(\frac{3 \log_2 L}{L^2 - 1} \right) \frac{2E_b}{N_0}} \right] \quad (9.54)$$

where $Q(x)$ is as defined in Equation (3.43) and $L = \sqrt{M}$ represents the number of amplitude levels in one dimension. In the context of L -PAM, a sequence of $k/2 = \log_2 L$ bits are assigned to an L -ary symbol using a Gray code (defined in Section 4.9.4).

9.8.3.2 Bandwidth-Power Trade-Off

The bandwidth-power trade-off of M -ary QAM at a bit error probability of 10^{-5} is displayed on the bandwidth-efficiency plane in Figure 9.6, with the abscissa measured in average E_b/N_0 . We assume Nyquist filtering of the baseband pulses so that the DSB transmission bandwidth at IF is $W_{IF} = 1/T$, where T is the symbol duration. Thus the bandwidth efficiency is $R/W = \log_2 M$, where M is the symbol set size. For realistic channels and waveforms, the performance must be reduced to account for the increased bandwidth necessary to implement realizable filters. From Figure 9.6 it can be seen that QAM represents a method of reducing the bandwidth required for the transmission of digital data. As with M -ary PSK, bandwidth efficiency can be exchanged for power or E_b/N_0 ; however, in the case of QAM, a *much more efficient exchange* is possible than in the case of M -ary PSK.

Example 9.5 Waveform Design

Assume that a data stream with data rate $R = 144$ Mbits/s is to be transmitted on an RF channel using a DSB modulation scheme. Assume Nyquist filtering and an allowable DSB bandwidth of 36 MHz. Which modulation technique would you choose for this requirement? If the available E_b/N_0 is 20, what would be the resulting probability of bit error?

Solution

The required spectral efficiency is

$$\frac{R}{W} = \frac{144 \text{ Mbits/s}}{36 \text{ MHz}} = 4 \text{ bits/s/Hz}$$

From Figure 9.6 we note that 16-ary QAM, with a theoretical spectral efficiency of 4 bits/s/Hz, requires a lower E_b/N_0 than that of 16-ary PSK for the same P_B . Based on these considerations we choose a 16-ary QAM modem.

With the available E_b/N_0 given as 20, we use Equation (9.54) to calculate the expected bit error probability as

$$P_B \approx \frac{3}{4} Q\left(\sqrt{\frac{4}{5} \frac{E_b}{N_0}}\right) = 2.5 \times 10^{-5}$$

Example 9.6 Spectral Efficiency

- Explain the computation of the QAM spectral efficiency in Example 9.5, considering that QAM is transmitted on orthogonal components of a carrier wave.
- Since the DSB bandwidth is 36 MHz in Example 9.5, consider using half that amount at baseband to transmit the 144-Mbits/s data stream, using multilevel PAM. What is the spectral efficiency needed to accomplish this, and how many levels of PAM would be required? Assume Nyquist filtering.

Solution

- Bandpass channel using QAM:* The 144-Mbits/s data stream is partitioned into a 72-Mbits/s in-phase and a 72-Mbits/s quadrature stream; one stream amplitude-modulates the cosine component of a carrier over a bandwidth of 36 MHz, and the other stream amplitude-modulates the sine component of the carrier wave over the same 36-MHz bandwidth. Since each 72-Mbits/s stream modulates an orthogonal component of the carrier, the 36 MHz suffices for both streams, or for the full 144 Mbits/s. Thus the spectral efficiency is (144 Mbits/s)/36 MHz = 4 bits/s/Hz.
- Required spectral efficiency at baseband*

$$\frac{R}{W} = \frac{144 \text{ Mbits/s}}{18 \text{ MHz}} = 8 \text{ bits/s/Hz}$$

Assuming Nyquist filtering, a bandwidth of 18 MHz can support a maximum symbol rate of $R_s = 2W = 36$ megasymbols/s [see Equation (3.80)]. Each PAM pulse must therefore have an ℓ -bit meaning, such that

$$R = \ell R_s$$

Hence,

$$\ell = \frac{144 \text{ Mbits/s}}{36 \text{ megapulses/s}} = 4 \text{ bits/pulse}$$

where $\ell = \log_2 L$, and $L = 16$ levels.

9.9 MODULATION AND CODING FOR BANDLIMITED CHANNELS

The channel coding techniques of Chapters 6–8 have generally *not* been associated with voice-grade telephone channels (although the first field test of sequential decoding of convolutional codes was on a telephone line). Recently, however, there has been considerable interest in techniques that can provide coding gain for bandlimited channels. The motivation is to enable the reliable transmission of

higher data rates over voice-grade channels. The potential gain is about 3 bits/symbol (for a given signal-to-noise ratio) [21] or, alternatively, a given error performance could be achieved with a power savings of 9 dB [21].

The greatest interest is in the following three separate coding research areas:

1. Optimum signal constellation boundaries (choosing a closely packed signal subset from any regular array or lattice of candidate points)
2. Higher-density lattice structures (adding improvement to the signal subset choice by starting with the densest possible lattice for the space)
3. Trellis-coded modulation (combined modulation and coding techniques for obtaining coding gain for bandlimited channels)

The first two areas are not "true" error control coding schemes. By "true error control coding" we refer to those techniques that employ some structured redundancy to improve the error performance. Only the third technique, trellis-coded modulation, involves redundancy. Each of these coding research areas and their expected performance improvements are discussed below.

9.9.1 Commercial Telephone Modems

The use of efficient modulation techniques has traditionally been spearheaded by the telecommunications industry, since the telephone company's foremost resource consists of sharply bandlimited voice-grade channels. The typical telephone channel is characterized by a high signal-to-noise ratio (SNR) of approximately 30 dB and a bandwidth of approximately 3 kHz. Table 9.4 lists the evolution of leased-line telephone modems, and Table 9.5 lists the evolution of dial-line telephone modem standards.

TABLE 9.4 Evolution of Leased-Line Telephone Modems

Year	Name	Maximum Bit Rate (bits/s)	Signaling Rate (symbols/s)	Modulation Technique	Signaling Efficiency (bits/symbol)
1962	Bell 201	2400	1200	4-PSK	2
1967	Milgo 4400/48	4800	1600	8-PSK	3
1971	Codex 9600C	9600	2400	16-QAM	4
1980	Paradyne MP14400	14,400	2400	64-QAM	6
1981	Codex SP14.4	14,400	2400	64-QAM	6
1984	Codex 2660	16,800	2400	Trellis-coded 256-QAM	7
1985	Codex 2680	19,200	2743	8-D Trellis-coded 160-QAM	7

TABLE 9.5 Evolution of Dial-Line Telephone Modem Standards

Year	Name	Maximum Bit Rate (bits/s)	Signaling Rate (symbols/s)	Modulation Technique	Signaling Efficiency (bits/symbol)
1984	V.32	9600	2400	2-D Trellis Coded 32-QAM	4
1991	V.32bis	14,400	2400	2-D Trellis Coded 128-QAM	6
1994	V.34	28,800	2400, 2743, 2800, 3000, 3200, 3429	4-D Trellis Coded 960-QAM	≈ 9
1996	V.34	33,600	2400, 2743, 2800, 3000, 3200, 3429	4-D Trellis Coded 1664-QAM	≈ 10
1998	4.90	downstream: 56,000 upstream: 33,600	8000 as in V.34	PCM* (<i>M</i> -PAM) as in V.34	7 ≈ 10
2000	V.92	downstream: 56,000 upstream: 48,000	8000 8000	PCM* (<i>M</i> -PAM) Trellis Coded PCM*	7 6

*In the G.711 ITU-T Recommendation, PCM is the term used for *M*-ary PAM signaling.

9.9.2 Signal Constellation Boundaries

Several researchers [22–26] have examined large numbers of possible QAM signal constellations in a search for designs that result in the best error performance for a given average signal-to-noise ratio. Figure 9.17 illustrates some examples of symbol constellations for $M = 4, 8$, and 16 that have been considered [22]. The circular sets are designated by the notation $(a, b, \dots)$, where there are a quantity of a signals on the inner circle, b signals on the next circle, and so on. In general, the constellation rule, known as the Campopiano-Glazer construction rule [24], that yields optimum signal set performance can be summarized as follows: From an infinite array of points closely packed in a *regular array or lattice*, select a closely packed subset of 2^k points as a signal constellation. In this case “optimum” means minimum average or peak power for a given error probability. In a two-dimensional signal space the optimum boundary surrounding an array of points tends toward a circle. Figure 9.18 illustrates examples of 64-ary ($k = 6$) and 128-ary ($k = 7$) signal sets from a rectangular array. The cross-shaped boundaries are a compromise to the optimum circle. The $k = 6$ constellation was used in the Paradyne 14.4-kbits/s modem. Compared with a square, the performance improvement resulting from a circular boundary is only a modest 0.2 dB [21].

9.9.3 Higher-Dimensional Signal Constellations

For any particular information rate and channel-noise process that is independent and identically distributed in two dimensions, signaling in a two-dimensional space can provide the same error performance with less average (or peak) power than signaling in a one-dimensional pulse-amplitude (PAM) space. This is accomplished

Signaling Efficiency (bits/symbol)
4
6
≈ 9
≈ 10
7
≈ 10
7
6

QAM signal
ormance for a
les of symbol
e circular sets
of a signals on
constellation
elds optimum
nite array of
ked subset of
mum average
nal space the
circle. Figure
ts from a rec-
the optimum
odem. Com-
m a circular

independent
nsional space
) power than
accomplished

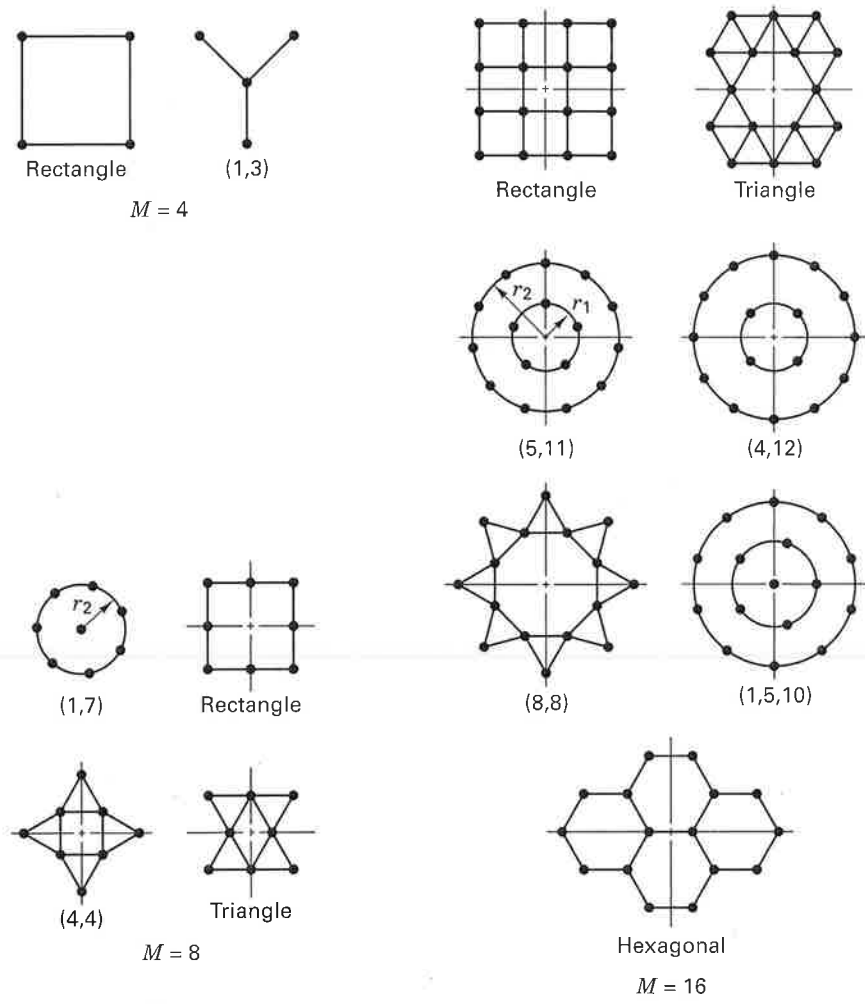


Figure 9.17 M -ary symbol constellations. (Reprinted with permission from C. M. Thomas, M. Y. Weidner, and S. H. Durrani, "Digital Amplitude-Phase Keying with M -ary Alphabets," *IEEE Trans. Commun.*, vol. COM22, no. 2, Feb. 1974, Figs. 2 and 3, p. 170. © 1974 IEEE.)

by choosing signaling points on a two-dimensional lattice from within a circular rather than a rectangular boundary. In the same way, by going to a higher number, N dimensions, and choosing points on an n -dimensional lattice from within an N -sphere rather than an N -cube, further energy savings are possible [27–30]. The goal of this constellation shaping is to make the required average energy of signal points from the N -sphere less than that from the N -cube; such reduction in required energy for a given error performance is referred to as a *shaping gain* [16]. Table 9.6 gives the energy savings possible in N dimensions. As N goes to infinity, the gain goes to 1.53 dB; it is not difficult to achieve shaping gain on the order of 1 dB [16, 21].

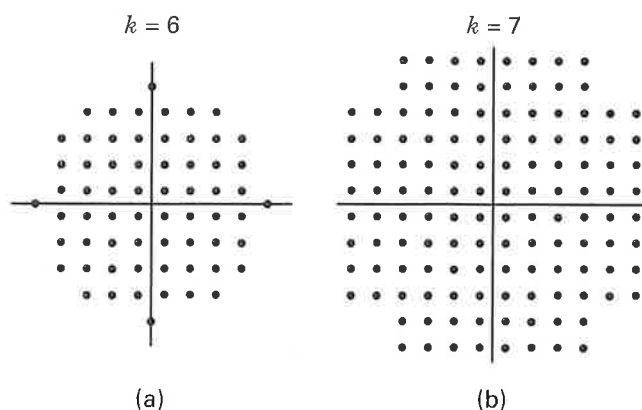


Figure 9.18 Examples of M -ary constellations using a rectangular array.

The channel is essentially two-dimensional, since symbols represented as points on a two-dimensional plane are transmitted in quadrature fashion. Multidimensional signaling is generally taken to mean signaling with two or more such planes. For transmitting n bits/symbol with N -dimensional (N even and greater than 2) signaling, the incoming bits are grouped into blocks of $nN/2$. A mapping must then be made that assigns data bits to $2^{nN/2}$ N -dimensional vectors that have the least energy among all such vectors. A corresponding inverse mapping must be made at the receiver.

Consider an example of mapping signals from a two-dimensional to a four-dimensional space. We start with a two-dimensional M -ary constellation, such as M -QAM with $M = 16$. Here the transmitted symbol, viewed as a point on a plane, is represented by $n = 4$ bits (two 4-ary amplitudes, and two bits per amplitude). Each symbol transmission consists of sending a vector from a space of 16 possible vectors. With four-dimensional signaling, the transmitted symbol, viewed as two points, one from each of two planes, is represented by 8 bits. Then, each (two-point) transmission consists of sending a vector from a space of $16 \times 16 = 256$ vec-

TABLE 9.6 Energy Savings from N -Sphere Mapping versus N -Cube Mapping (Shaping Gain)

Dimensions (N)	N -Sphere Mapping Gain (dB)
2	0.20
4	0.45
8	0.73
16	0.98
24	1.10
32	1.17
48	1.26
64	1.31

Source: G. D. Forney, Jr., et al., "Efficient Modulation for Bandlimited Channels," *IEEE J. Sel. Areas Commun.*, vol. SAC2, no. 5, September 1984, pp. 632-647.

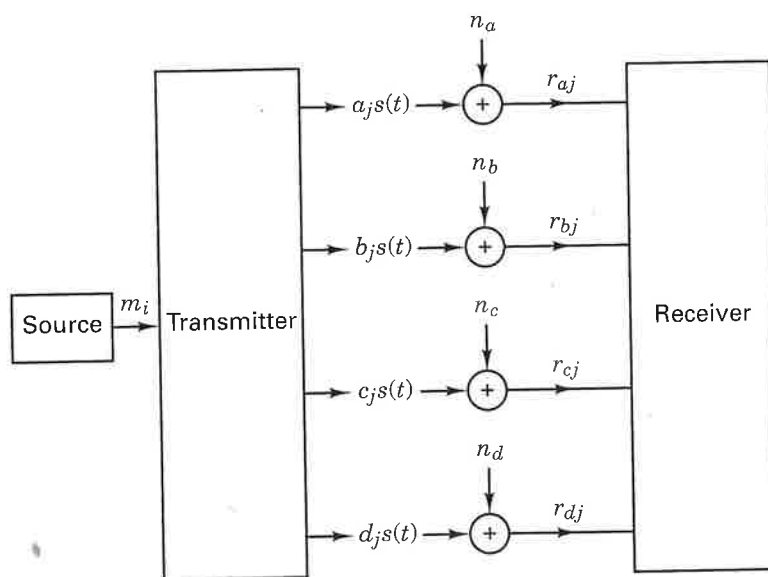


Figure 9.19 Four-dimensional system configuration.

tors. In general, the source-data bits are grouped into blocks of $nN/2$. In this example for four-dimensional signaling, we group the source-data bits into blocks of 8 bits (2 planes $\times n = 4$ bits/plane). Such an 8-bit transmission can be viewed as a mapping from a space of 2^n two-dimensional vectors to a space of $2^{nN/2}$ four-dimensional vectors. For the four-dimensional system depicted in Figure 9.19, a given source produces one of 256 four-dimensional vectors m_i ($i = 1, 2, \dots, 256$) by grouping two 16-ary symbols (two planes) at a time and transmitting waveforms, $a_j s(t)$, $b_j s(t)$, $c_j s(t)$, $d_j s(t)$, where $j = 1, \dots, 4$ represents one of 4-ary amplitude values. These baseband or bandpass waveforms are transmitted on separate noninterfering channels. In each channel, the waveforms are distorted by independent AWGN, and at the receiver they are demodulated with matched filters. We may choose to transmit the N -dimensional signal in a number of ways:

1. Using four separate wires representing four baseband channels.
2. Using two bandpass channels, each with separately modulated inphase and quadrature components.
3. Using time- or frequency-division multiplexing to carry the baseband or bandpass channels on a common transmission line.
4. Using orthogonal electromagnetic wave polarization.

Thus, if the Figure 9.19 example represents a radio system, we could follow method 2 above and modulate waveforms $a_j s(t)$, and $b_j s(t)$ in quadrature fashion onto a particular carrier wave, while modulating waveforms $c_j s(t)$, and $d_j s(t)$ onto a second carrier wave. Thus, during each $2T$ second interval, one would transmit four 4-ary numbers

representing 8 bits or a vector from a 256-ary space. Further shaping gain can be similarly achieved for the delivery of 16-ary symbols per plane with six-dimensional signaling, where every $3T$ seconds, a 16-ary symbol from each of three planes is transmitted. Thus, each six-dimensional signal constitutes three 16-ary values representing 12 bits or a point on a 4096-ary signal space. It is important to emphasize that it is not the mere grouping of the 16-ary symbols that brings about the shaping gain. The gain comes about because detection performed over a larger signal space can achieve a given error performance with a smaller E_b/N_0 . In the case of sending 16-ary symbols with six-dimensional signaling, a 12-bit sequence is detected every $3T$ seconds (not a 4-bit sequence every T seconds). Detection in a higher dimensional space entails greater (signal-mapping) complexity. Compromises are generally used to simplify the mapping complexity at the cost of some suboptimality in energy efficiency.

9.9.4 Higher-Density Lattice Structures

In Section 9.9.3, we discussed the selection of a closely packed subset of points from any regular array or lattice. Here we consider the added improvement by starting with the *densest possible lattice* in the space. In a two-dimensional signal space, the densest lattice is the hexagonal lattice (try penny packing). The result of employing a hexagonal lattice instead of a rectangular one, such as those shown in Figure 9.18, can be a 0.6-dB savings in average energy. Figure 9.20 illustrates some examples of hexagonal packing. The strange-looking $k = 4$ constellation in Figure 9.20a was discovered by Foschini et al. [26] and is still the best 16-ary constellation known. The $k = 6$ constellation in Figure 9.20b was used in the Codex SP14.4 modem.

The hexagonal lattice is optimum for two dimensions. For higher dimensions there are other lattice structures that provide the densest packing. Table 9.7 gives the gain over the rectangular lattice, in decibels, due to the densest packings currently known for various dimensions.

9.9.5 Combined Gain: N -Sphere Mapping and Dense Lattice

It is possible to combine the benefits of the Campopiano–Glazer boundary construction in N dimensions with the gain from the densest lattice in N -space. The

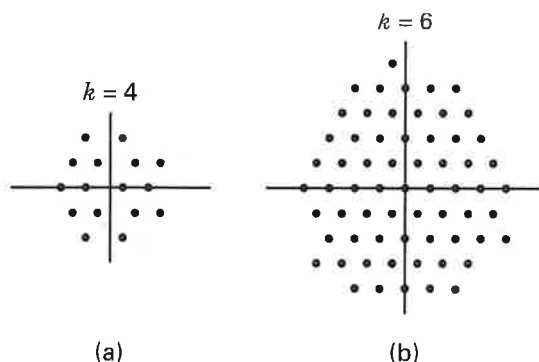


Figure 9.20 Examples of M -ary constellations using a hexagonal array.

TABLE 9.7 Energy Savings from Dense Lattices versus the Rectangular Lattice

Dimensions (N)	Dense Lattice Gain (dB)
2	0.62
4	1.51
8	3.01
16	4.52
24	6.02
32	6.02
48	7.78
64	8.09

Source: G. D. Forney, Jr., et al., "Efficient Modulation for Bandlimited Channels," *IEEE J. Sel. Areas Commun.*, vol. SAC2, no. 5, September 1984, pp. 632-647.

resulting gain is a combination of N -sphere versus N -cube boundary gain of Table 9.6 and the lattice packing density gain of Table 9.7. The combined energy savings are shown in Table 9.8.

TABLE 9.8 Combined Energy Savings from N -Sphere Mapping and Dense Lattices

Dimensions (N)	Combined Savings Gain (dB)
2	0.82
4	1.96
8	3.74
16	5.50
24	7.12
32	7.19
48	9.04
64	9.40

Source: G. D. Forney, Jr., et al., "Efficient Modulation for Bandlimited Channels," *IEEE J. Sel. Areas Commun.*, vol. SAC2, no. 5, September 1984, pp. 632-647.

9.10 TRELLIS-CODED MODULATION

The error-correction codes described in Chapters 6-8, when used in real-time communication systems, provide improvements in error performance at the cost of bandwidth expansion. For both block codes and convolutional codes, transforming each input data k -tuple into a larger output codeword n -tuple, requires additional transmission bandwidth. Therefore, in the past, coding generally was not popular for bandlimited channels such as telephone channels, where signal bandwidth expansion is not practical. Since about 1984, however, there has been active interest

in combined modulation and coding schemes, called *trellis-coded modulation* (TCM), that achieve error-performance improvements without expansion of signal bandwidth. TCM schemes use redundant nonbinary modulation in combination with a *finite-state machine* (the encoder). What is a finite-state machine, and what is meant by its state? Finite-state machine is the general name given to a device that has a memory of past signals; the adjective *finite* refers to the fact that there are only a finite number of unique states that the machine can encounter. What is meant by the *state* of a finite-state machine? In the most general sense, the state consists of the smallest amount of information that, together with a current input to the machine, can predict the output of the machine. The state provides some knowledge of the past signaling events and the restricted set of possible outputs in the future. A future state is restricted by the past state.

For each symbol interval, a TCM finite-state encoder selects one of a set of waveforms, thereby generating a sequence of coded waveforms to be transmitted. The noisy received signals are detected and decoded by a soft-decision maximum-likelihood detector/decoder. In conventional systems involving modulation and coding, it is common to separately describe and implement the detector and the decoder. With TCM systems, however, these functions must be treated jointly. Coding gain can be achieved without sacrificing data rate or without increasing either bandwidth or power [6, 31]. At first, it may seem that this statement violates some basic principle of power-bandwidth, error-probability trade-off. However, there is still a trade-off involved, since TCM achieves coding gain at the expense of decoder complexity.

Trellis-coded modulation combines a multilevel/phase modulation signaling set with a *trellis-coding scheme*. The term “trellis-coding scheme” refers to any code system that has memory (a finite state machine), such as a convolutional code. Multilevel/phase signals have constellations involving multiple amplitudes, multiple phases, or combinations of multiple amplitudes and multiple phases. In other words, a TCM signal set is best represented by any signal set (greater than binary) whose vector representations can be depicted on a plane, such as that shown in Figure 9.16a for QAM signals. A trellis-coding scheme is one that can be characterized with a state-transition (trellis) diagram, similar to the trellis diagrams describing convolutional codes. The convolutional codes presented in Chapter 7 are linear, although trellis codes are not constrained to be linear. Coding gains can be realized with block codes or trellis codes, but only trellis codes will be considered because the availability of the *Viterbi decoding algorithm* makes trellis decoding simple and efficient. Ungerboeck showed that in the presence of AWGN, TCM schemes can yield net coding gains of about 3-dB relative to uncoded systems with relative ease, while gains of about 6-dB can be achieved with greater complexity.

9.10.1 The Idea Behind Trellis-Coded Modulation

In TCM, channel coding and modulation are performed together; where one begins and the other ends cannot be easily established. Can you speculate what ideas might have prompted the development of TCM? Perhaps it started with the notion

that “not all signal subsets (in a constellation) have equal distance properties.” That is, for a nonorthogonal signal set, such as MPSK, antipodal signals have the best distance properties for easily discriminating one signal from the other, while nearest neighbor signals have relatively poor distance properties. It may be that the initial idea of coded modulation came about by trying to exploit these differences.

A simple analogy might be helpful in understanding the overall goals in TCM. Imagine that there is an all-knowing wizard at the transmitter. As the message bits enter the system, the wizard recognizes that some of the bits are most vulnerable to the degradation effects of channel impairments; hence, they are assigned modulation waveforms associated with the best distance properties. Similarly, other bits are judged to be very robust, and hence, they are assigned waveforms with poorer distance properties. Modulation and coding take place together. The wizard is assigning waveforms to bits (modulation), but, the assignment is being performed according to the criterion of better or worse distance properties (channel coding).

9.10.1.1 Increasing Signal Redundancy

TCM may be implemented with a convolutional encoder, wherein k current bits and $K - 1$ prior bits are used to produce $n = k + p$ code bits, where K is the encoder constraint length (see Chapter 7) and p is the number of parity bits. Notice that encoding increases the signal set size from 2^k to 2^{k+p} . Ungerboeck [31] investigated the increase in channel capacity achievable by signal set expansion, and concluded that most of the achievable coding gain over conventional uncoded multilevel modulation could be realized by expanding the uncoded signaling set by the factor of two ($p = 1$). This can be accomplished by encoding with a rate $k/(k + 1)$ code and subsequently mapping groups of $(k + 1)$ bits into the set of 2^{k+1} waveforms. Figure 9.21a illustrates an uncoded 4-ary PAM signal set, before and after being rate 2/3 encoded into an 8-ary PAM signal set. Similarly, Figure 9.21b illustrates an uncoded 4-ary PSK (QPSK) signal set, before and after being rate 2/3 encoded into an 8-ary PSK signal set. Similarly, Figure 9.21c illustrates an uncoded 16-ary QAM signal set, before and after being rate 4/5 encoded into a 32-ary QAM signal set. In each of the cases shown in Figure 9.21, the system is configured to use the same average signal power before and after coding. Also, to provide the needed redundancy for coding, the signaling set is increased from $M = 2^k$ to $M' = 2^{k+1}$. Thus, $M' = 2M$; however, the increase in the alphabet size does not result in an increase in required bandwidth. Recall from Section 9.7.2 that the transmission bandwidth of nonorthogonal signaling does not depend on the density of signaling points in the constellation; it depends only on the rate of signaling. The expanded signal set *does* result in a reduced distance between adjacent symbol points (for signal sets with a constant average power), as seen in Figure 9.21. In an *uncoded* system, such a reduced distance degrades the error performance. However, because of the redundancy introduced by the code, this reduced distance no longer plays a critical role in determining the error performance. Instead, the *free distance*, which is the minimum distance between members of the set of *allowed* code sequences, determines the error performance. The free distance characterizes the “easiest

modulation
ion of signal
combination
, and what is
device that
at there are
ter. What is
se, the state
ent input to
vides some
e outputs in

e of a set of
transmitted.
a maximum-
lulation and
and the de-
ointly. Cod-
easing either
iolates some
ver, there is
e of decoder

on signaling
efers to any
itional code.
udes, multi-
ses. In other
than binary)
hown in Fig-
characterized
is describing
re linear, al-
be realized
ered because
g simple and
schemes can
relative ease,

here one be-
e what ideas
th the notion

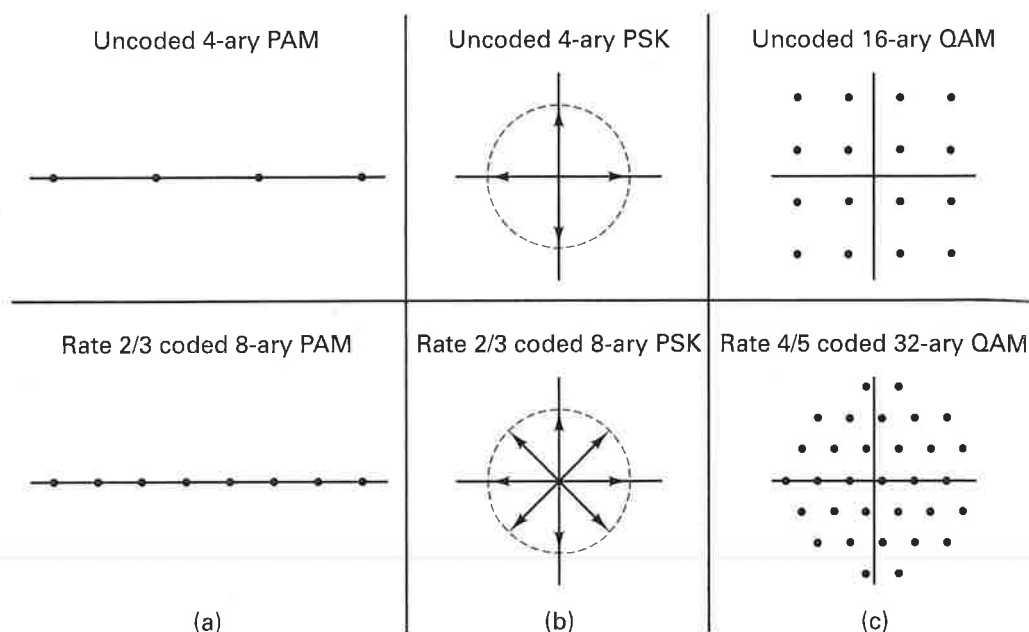


Figure 9.21 Increase of signal set size for trellis-coded modulation.

way” for the decoder to make an error. (See Section 9.10.3.1.) Whenever there is a code at work, the signaling space is not the proper place for examining the error-performance advantage that coding can achieve. This is because the code is defined by rules and constraints that are not visible in the signaling space. When two signals are in close proximity in the signaling space of a coded system, their closeness may not have much performance significance because the rules of the code may not allow for the transitioning between two such vulnerable signal points. Where is the proper place for evaluating allowed code sequences and distance properties? It is the trellis diagram. Using this diagram, the objective of TCM is to assign waveforms to trellis transitions so as to increase the free distance between the waveforms that are the most likely to be confused.

9.10.2 TCM Encoding

9.10.2.1 Ungerboeck Partitioning

Assume that the receiver uses soft decision, so that the appropriate distance metric is Euclidean. In order to maximize the free Euclidean distance (ED), the code-to-signal mapping pioneered by Ungerboeck [31] follows from the successive partitioning of a modulation-signal constellation into subsets having increasing minimum distances $d_0 < d_1 < d_2 \dots$ between the elements of the subsets. The concept is illustrated in Figure 9.22 for the 8-PSK signal set. In Figure 9.22, the original constellation of signals is labeled A_0 , and the individual signals are numbered sequentially from 0 to 7. If the average signal power (amplitude squared) is chosen

y QAM



2-ary QAM



ver there is a
ng the error-
de is defined
n two signals
lossenness may
ode may not
Where is the
perties? It is
assign wave-
en the wave-

iate distance
ce (ED), the
he successive
ig increasing
ets. The con-
, the original
numbered se-
ed) is chosen

s Chap. 9

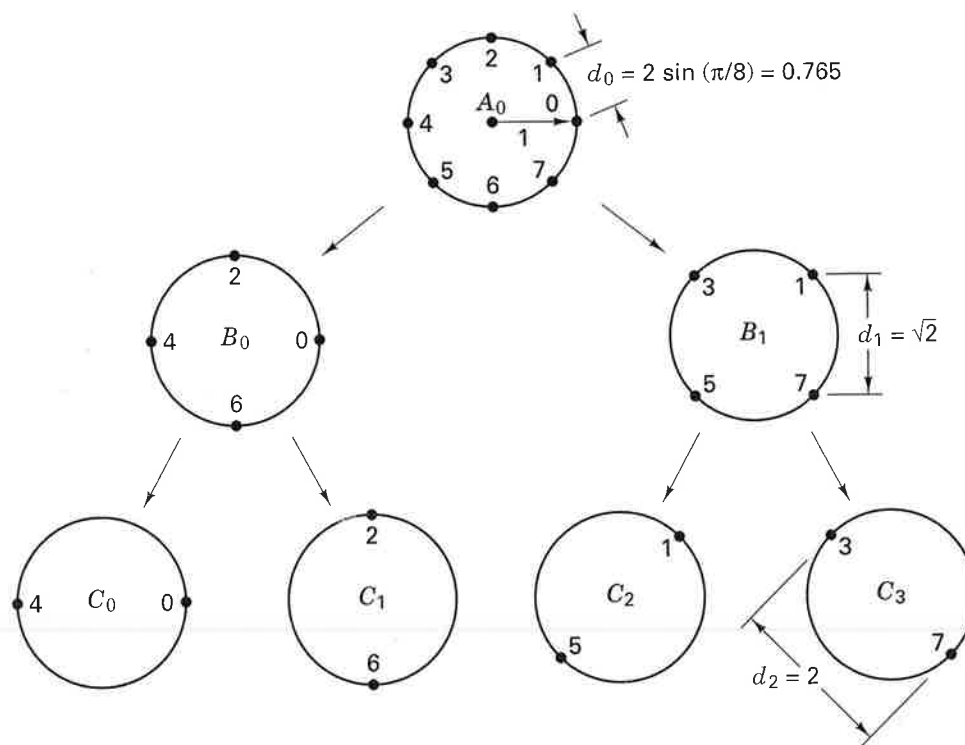


Figure 9.22 Ungerboeck partitioning of an 8-PSK signal set.

equal to unity, then the distance d_0 between any two adjacent signals is seen to be $2 \sin(\pi/8) = 0.765$. The first level of partitioning results in subsets B_0 and B_1 , where the distance between adjacent signals is $d_1 = \sqrt{2}$. The next level of partitioning results in subsets C_0 to C_3 , where the distance between adjacent signals is $d_2 = 2$. The construction of simple codes (up to eight states) can be determined heuristically. First a suitable trellis structure is selected, which can be done without any particular encoder in mind. TCM is classified as a waveform-coding technique because in order to describe the concept, only a suitable trellis and a set of modulation waveforms are required; it is not yet necessary to introduce bits. Waveforms from the extended set of $M' = 2^{k+1}$ waveforms are assigned to the trellis transitions so as to maximize the free ED. In the treatment of convolutional coding in Chapter 7, the encoder-trellis transitions (reflecting the behavior of an encoding circuit) were labeled with code bits. For TCM, the trellis transitions are labeled with modulation waveforms. The uncoded 4-PSK waveform set will serve as a reference for the 8-PSK coded set. This reference set, as seen in Figure 9.23, has a trivial one-state trellis diagram with four parallel transitions. This is a trivial trellis because a one-state trellis means that the system has no memory. There are no constraints or restrictions as to which of the 4-PSK signals can be transmitted during any time interval; hence, for this uncoded case, the optimum detector simply makes independent, nearest-neighbor decisions for each noisy 4-PSK signal received.

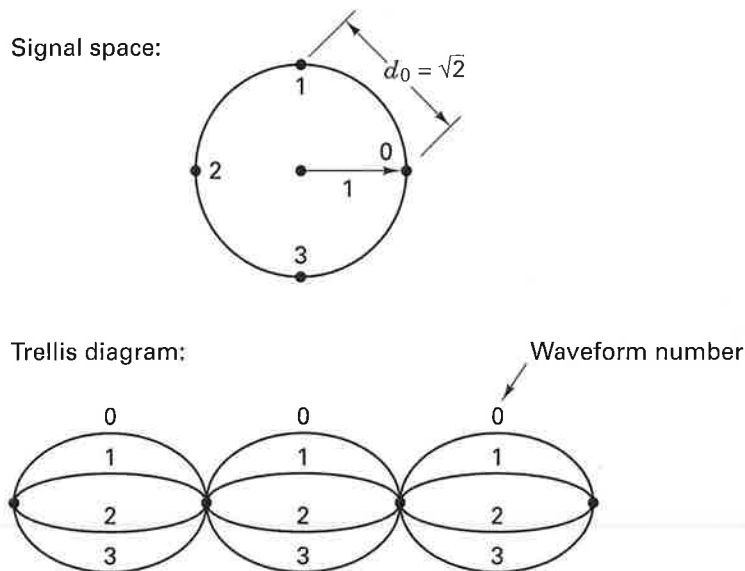


Figure 9.23 Uncoded 4-PSK and its one-state trellis diagram.

9.10.2.2 Mapping of Waveforms to Trellis Transitions

Ungerboeck devised a heuristic set of rules [31] for assigning trellis-transition branches to waveforms so that coding gain is assured, given an adequate choice of trellis states. The rules for forming the trellis structure and for partitioning the waveform set (in the context of coded 8-ary PSK modulation) can be summarized as follows:

1. If k bits are to be encoded per modulation interval, the trellis must allow for 2^k possible transitions from each state to a successor state.
2. More than one transition may occur between pairs of states.
3. All waveforms should occur with equal frequency and with a fair amount of regularity and symmetry.
4. Transitions originating from the same state are assigned waveforms either from subset B_0 or B_1 —never a mixture between them.
5. Transitions joining into the same state are assigned waveforms either from subset B_0 or B_1 —never a mixture between them.
6. Parallel transitions are assigned waveforms either from subset C_0 or C_1 or C_2 or C_3 —never a mixture between them.

The rules guarantee that codes constructed in this way will have a regular structure and a free ED that will always exceed the minimum distance between signal points of the uncoded reference modulation. Figure 9.24 illustrates a possible code-to-signal mapping, using a four-state trellis with parallel paths. The code-to-signal assignments are made by examining the partitioned signal space in Figure 9.22 in con-

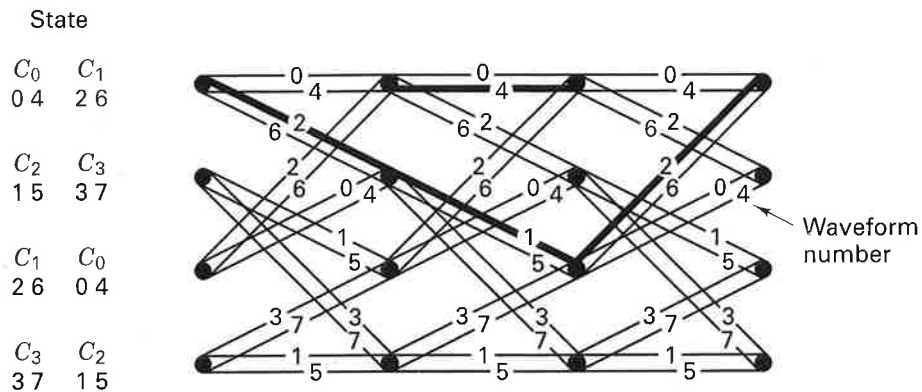


Figure 9.24 Four-state trellis with parallel paths.

cert with the trellis diagram in Figure 9.24 and the rules listed above. Written on the trellis transitions are the waveform numbers that have been assigned to those transitions by following the partitioning rules. Notice that for 8-PSK, the waveform assignments comply with rule 1: There are $k + 1 = 3$ code bits and thus $k = 2$ information bits, and there are $2^2 = 4$ transitions into and out of each state. The waveform assignments comply with rule 6 because each parallel pair of transitions have been assigned waveforms from subset C_0 or C_1 or C_2 or C_3 . Also, the assignments comply with rules 4 and 5, because the four branches leaving (or entering) a state have been assigned waveforms from subset B_0 or B_1 . In Figure 9.24, the states of the trellis have been designated according to the waveform types that may appear on the transitions leaving that state. Thus, we can refer to the states in terms of the signal subsets as the $C_0 C_1$ state or the $C_2 C_3$ state, and so forth. Or, another possible designation in terms of waveform numbers would be the 0426 state or the 1537 state, and so forth. In Figure 9.24, both designations are shown. This assignment of modulation waveforms to trellis transitions by observing the set-partitioning rules concludes the specification of the trellis encoder. Note that the final assignment of the code bits to the waveform signals (codeword-to-transition mapping) can now be made in an arbitrary fashion. Although you might think it surprising that bits can now be assigned to trellis transitions and waveforms with impunity, recall that a circuit does not yet exist. Hence, there are no bits, and the trellis transitions can mean anything that we choose for them to mean. What are the consequences of such an *arbitrary* assignment? Different codeword-to-transition choices will reflect upon the encoder design. Hence, if we are lucky, designing the encoder circuit whose output bits correspond to the way that we assigned them to state transitions will be easily accomplished. If however, we are not lucky, such a circuit design will be difficult to accomplish. Some choices of codeword assignments will give rise to simple encoder designs, while other choices may dictate unwieldy designs.

The same trellis structure of Figure 9.24 will soon be examined in the context of detection and decoding to verify that a coding gain is assured by the fact that Ungerboeck's rules have been followed during the encoding process.

9.10.3 TCM Decoding

9.10.3.1 An Error Event and Free Distance

The task of a convolutional decoder is to estimate the path that the message had traversed through the encoding trellis. If all input message sequences are equally likely, a decoder that will achieve the minimum probability of error is one that compares the conditional probabilities $P(\mathbf{Z}|\mathbf{U}^{(m)})$ —where $\mathbf{Z}$ is the received sequence of waveforms, and $\mathbf{U}^{(m)}$ is one of the possible transmitted sequences of waveforms—and chooses the maximum. This decision-making criterion, known as *maximum likelihood*, is described in Section 7.3.1. Finding the sequence $\mathbf{U}^{(m')}$ which maximizes $P(\mathbf{Z}|\mathbf{U}^{(m)})$ is equivalent to finding the sequence $\mathbf{U}^{(m')}$ that has the greatest similarity to $\mathbf{Z}$. Since a maximum-likelihood decoder will choose the trellis path whose corresponding sequence $\mathbf{U}^{(m')}$ is at the minimum distance to the received sequence $\mathbf{Z}$, the maximum-likelihood problem is identical to the problem of finding the shortest distance through the trellis diagram.

Because a convolutional code is a group (or linear) code, the set of distances that must be examined is independent of which sequence is selected as a test sequence. Therefore, there is no loss in generality in selecting an all-zeros test sequence, which is shown as a dashed line in Figure 9.25. Assuming that the all-zeros sequence is transmitted, an error event is identified as a divergence from the all-zeros path followed by a remergence with the all-zeros path. Error events both start and end in state a , and they do not return to state a anywhere in between.

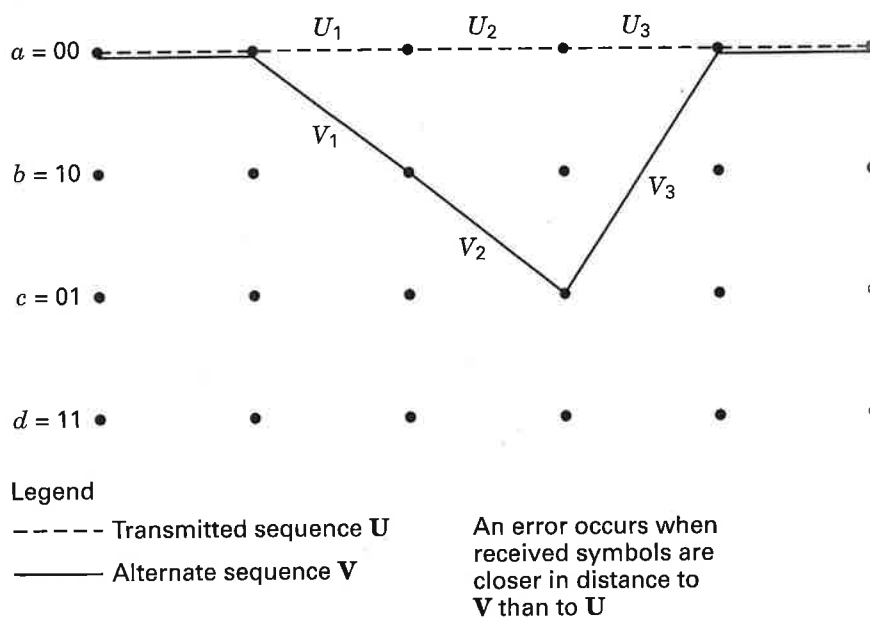


Figure 9.25 Illustration of an error event.

Figure 9.25 illustrates an error event in a trellis code; that is, the figure illustrates a transmitted all-zeros sequence marked $\mathbf{U} = \dots, U_1, U_2, U_3, \dots$ and an alternate sequence marked $\mathbf{V} = \dots, V_1, V_2, V_3, \dots$. The alternate sequence is seen to diverge and then remerge with the transmitted sequence. Assuming soft-decision decoding, an error event occurs whenever the received symbols are closer in Euclidean distance to some alternate sequence $\mathbf{V}$ than to the actually transmitted sequence $\mathbf{U}$. This implies that codes for multilevel/phase signals should be designed to achieve maximum free Euclidean distance (ED); the larger the Euclidean distance, the lower the probability of error. Therefore, assigning waveforms to trellis transitions at the encoder in a way that maximizes this free ED (see Section 9.10.2) is the key to optimizing the trellis codes.

9.10.3.2 Coding Gain

Using soft-decision, maximum-likelihood decoding, and assuming unit average signal power and Gaussian noise with variance σ^2 per dimension, a lower bound on the error-event probability can be expressed in terms of the free distance d_f [32] as

$$P_e \geq Q\left(\frac{d_f}{2\sigma}\right) \quad (9.55)$$

where $Q(\cdot)$ is the complementary error function defined in Equation (3.43). The term "error event" is used, rather than "bit-error," because the error might entail more than one flawed bit. At high signal-to-noise ratios (SNR), the bound in Equation (9.55) is asymptotically exact. The *asymptotic coding gain* G , in dB, compared with some uncoded reference system with the same average signal power and noise variance, is therefore expressed as a ratio of distances or distances squared, and we write

$$G(\text{dB}) = 20 \times \log_{10}\left(\frac{d_f}{d_{\text{ref}}}\right) \quad \text{or} \quad G(\text{dB}) = 10 \times \log_{10}\left(\frac{d_f^2}{d_{\text{ref}}^2}\right) \quad (9.56)$$

where d_f and d_{ref} represent the free ED of the coded system and the free ED of an uncoded reference system, respectively. Note that for high SNR and a given error probability, Equation (9.56) yields the same result as the coding gain expressed in Equation (6.19), which is rewritten as

$$G(\text{dB}) = \left(\frac{E_b}{N_0}\right)_u (\text{dB}) - \left(\frac{E_b}{N_0}\right)_c (\text{dB}) \quad (9.57)$$

where $(E_b/N_0)_u$ and $(E_b/N_0)_c$ represent the required E_b/N_0 (given in dB) for the uncoded system and the coded system, respectively. Keep in mind that the coding gain expressed in Equation (9.56) provides the same information (at high SNR) as the more familiar error-performance improvement expression in Equation (9.57). Equation (9.56), in effect, summarizes the primary goal of TCM code construction. It is to achieve a free distance that exceeds the minimum distance between uncoded modulation signals (at the same information rate, bandwidth, and power).

9.10.3.3 Coding Gain for 8-PSK with a 4-State Trellis

We now calculate the coding gain for 8-PSK with the 4-state trellis that was designed by following the encoding rules in Section 9.10.2.2. The trellis in Figure 9.24 will now be examined in the context of a decoding procedure. First the all-zeros sequence is chosen as the test vehicle. In other words, assume that the transmitter sent a sequence containing only copies of the waveform number 0. To demonstrate the benefits of this TCM system (using the Viterbi decoding algorithm), it must be shown that the easiest way for the coded system to make an error is still more difficult than the easiest way for the uncoded system to make an error. It is necessary to examine every possible divergence-from and remergence-to the correct path (all-zeros sequence), and find the one with the minimum ED from the correct path. First, look at the candidate error-event path in Figure 9.24 that is shown darkened and labeled with waveform numbers 2, 1, 2. We compute the squared distance from the all zeros path as the sum of the individual squared distances of waveform 2 from waveform 0, waveform 1 from waveform 0, and waveform 2 from waveform 0. The individual distances are taken from the partitioning diagram of Figure 9.22, and the computation results in

$$d^2 = d_1^2 + d_0^2 + d_1^2 = 2 + 0.585 + 2 = 4.585$$

or,

(9.58)

$$d = \sqrt{4.585} = 2.2$$

In Equation (9.58), Euclidean distance d is formed in the same way that a resultant vector in a Euclidean space is constructed—that is, as the square root of the sum of the squares of the individual components (distances). In Figure 9.24, there is a divergence/remergence path that actually has a smaller ED than $d = 2.2$. It is the darkened error event (labeled waveform 4) that results if the parallel path survives (using Viterbi decoding), rather than the correct path associated with waveform 0. One might ask: If the decoder chooses the parallel path,—that is, the successor state is the same state in either case—is that really a serious error? If the parallel path is incorrectly chosen (it is still a divergence/remergence path, even though it occupies only one interval of time), then when we later introduce circuits and bits, a surviving waveform 4 will yield an incorrect bit meaning. The distance of the waveform 4 parallel path from waveform 0 is seen in Figure 9.22 to be $d = 2$. This distance is smaller than that of any other potential error event (try them all to be convinced); hence, the free ED for this coded system is $d_f = 2$. The minimum ED for the uncoded reference signal set in Figure 9.23 is $d_{\text{ref}} = \sqrt{2}$. Equation (9.56) is now used to compute the asymptotic gain as follows:

$$G(\text{dB}) = 10 \log_{10} \left(\frac{d_f^2}{d_{\text{ref}}^2} \right) = 10 \log_{10} \left(\frac{4}{2} \right) = 3 \text{ dB} \quad (9.59)$$

9.10.4 Other Trellis Codes

9.10.4.1 Parallel Paths

The trellis diagram requires parallel paths whenever the number of states is less than the size of the coded waveform set M' . Hence the 4-state trellis for 8-PSK requires parallel paths. To understand why this is the case, we first state again rule 1 from Ungerboeck: If k bits are to be encoded per modulation interval, the trellis must allow for 2^k possible transitions from each state to a successor state. Each waveform in the case of coded 8-PSK represents $k + 1 = 3$ code bits or $k = 2$ data bits. Hence, this first rule calls for $2^k = 2^2 = 4$ transitions to each successor state. At first glance, the 4-state trellis without parallel paths can meet this provision, since a fully connected trellis (each state connected to each of the other succeeding states) complies. However, try drawing a 4-state, fully connected trellis without parallel paths, and try meeting the Ungerboeck rules 4 and 5 in the context of a coded 8-PSK system. It is not possible. Violating the rules will yield suboptimum results. In the next section, an 8-state trellis for 8-PSK (the number of states is no longer less than M') is shown, where all of the partitioning rules can be observed without the need for parallel paths.

9.10.4.2 8-State Trellis

After experimentation with various trellis structures and channel-signal assignments, the 8-PSK code depicted in Figure 9.26 was chosen as optimum for eight states [31]. The error-event path with minimum distance from the all-zeros path is labeled with waveform numbers 6, 7, 6. Here, since there are no parallel paths to limit the free ED, the squared free ED is $d_f^2 = d_1^2 + d_0^2 + d_1^2 = 4.585$, where the distances d_0 and d_1 are obtained from Figure 9.22. The asymptotic coding gain of the 8-state TCM system compared with the reference 4-PSK system is

$$G(\text{dB}) = 10 \times \log_{10} \frac{(d_1^2 + d_0^2 + d_1^2)_{\text{coded 8-PSK}}}{(d_{\text{ref}}^2)_{\text{uncoded 4-PSK}}} = 10 \times \log_{10} \left(\frac{4.585}{2} \right) = 3.6 \text{ dB} \quad (9.60)$$

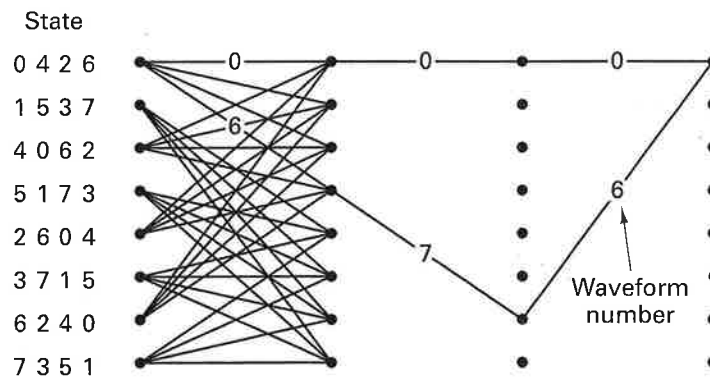


Figure 9.26 Eight-state trellis diagram for coded 8-PSK.

trellis that was
shown in Figure

First the all-
that the trans-
number 0. To
encoding algo-
make an error
take an error.
dependence to the
ED from the
in 9.24 that is
compute the
squared dis-
0, and wave-
partitioning

(9.58)

at a resultant
of the sum of
there is a di-
2.2. It is the
path survives
waveform 0.
the successor
of the parallel
even though it
units and bits,
distance of the
is $d = 2$. This
them all to be
minimum ED
equation (9.56) is

(9.59)

Similarly, a 16-state trellis structure for the coded 8-PSK constellation can be shown to yield a coding gain of 4.1 dB over uncoded 4-PSK [31]. For less than eight states, further coding gain may be realized by introducing asymmetry into the constellation of modulation signals [33].

9.10.4.3 Trellis Coding for QAM

The set-partitioning technique can be applied to other types of modulation waveforms. Consider the use of coded 16-QAM, with three information bits per modulation interval, where uncoded 8-PSK is designated as the reference system. For normalizing the 16-QAM space, we choose the average value of the signal-set squared-amplitude equal to unity, resulting in $d_0 = 2/\sqrt{10}$. Figure 9.27 illustrates a partitioning of 16-QAM waveforms into subsets with increasing subset distances ($d_0 < d_1 < d_2 < d_3$). An 8-state 16-QAM code system, obtained by using set-partitioning similar to the procedure outlined earlier, is shown in Figure 9.28 [31]. The minimum-distance error-event path is labeled D_6, D_5, D_2 . Despite the coding gain that can be achieved with TCM schemes, potential phase ambiguities exist in decoding the expanded signal space that can seriously degrade performance. Wei [34] has applied the concepts of differential encoding to TCM techniques, so that the resultant codes are transparent to signal element rotations of 90-, 180-, and 270-degrees.

In summary, trellis coding for bandlimited channels employs larger signal alphabets (i.e., M-ary PAM, PSK, or QAM) to accommodate the redundancy introduced by coding, so that channel bandwidth is not increased. Even though an increase in the modulation signal-set size reduces the minimum distance between signals, the free ED between *valid* code sequences more than compensates for this reduction. The result is a net coding gain of 3 to 6 dB, without any bandwidth

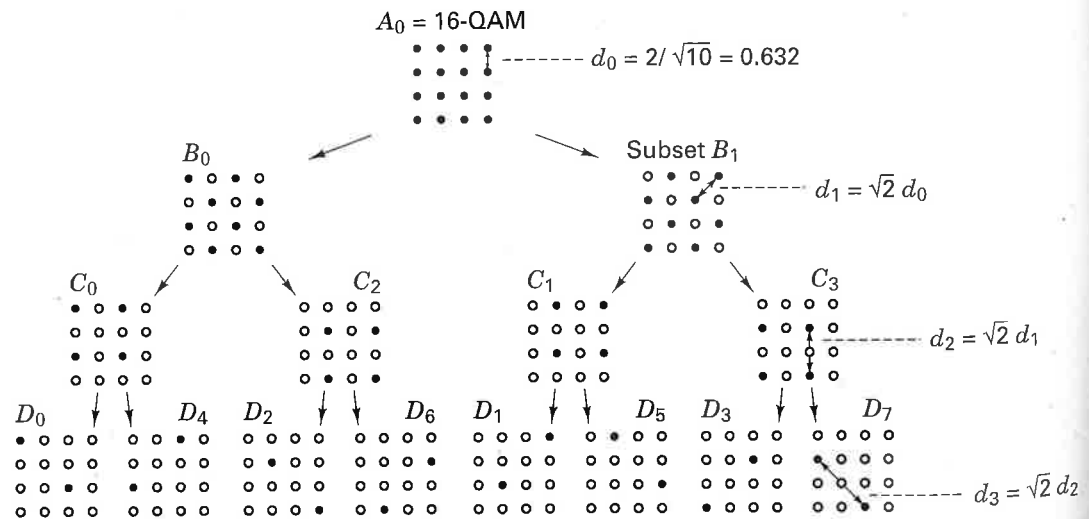


Figure 9.27 Ungerboeck partitioning of 16-QAM signals.

ion can be
s than eight
ry into the

modulation
on bits per
nce system.
e signal-set
illustrates a
subset dis-
y using set-
ce 9.28 [31].
the coding
ties exist in
nance. Wei
ues, so that
-, 180-, and

er signal al-
lancy intro-
though an
ce between
ates for this
bandwidth

$$t_2 = \sqrt{2} d_1$$

$$d_3 = \sqrt{2} d_2$$

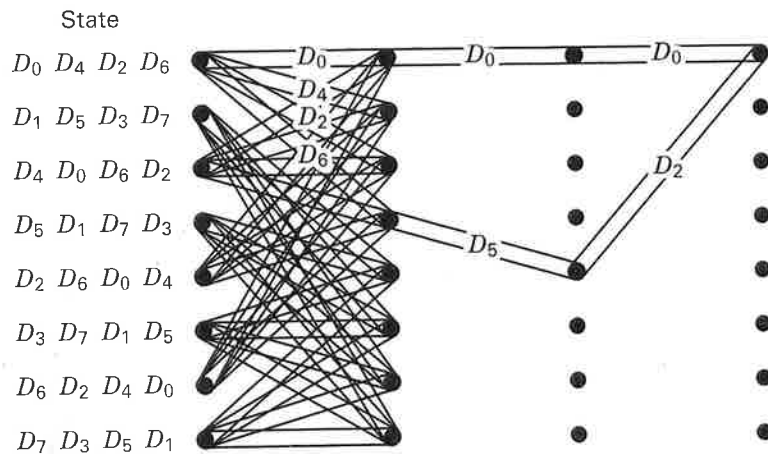


Figure 9.28 Eight-state trellis diagram for 16-QAM signaling.

expansion [6, 31]. In the following section, these ideas are further illustrated with an example.

9.10.5 Trellis-Coded Modulation Example

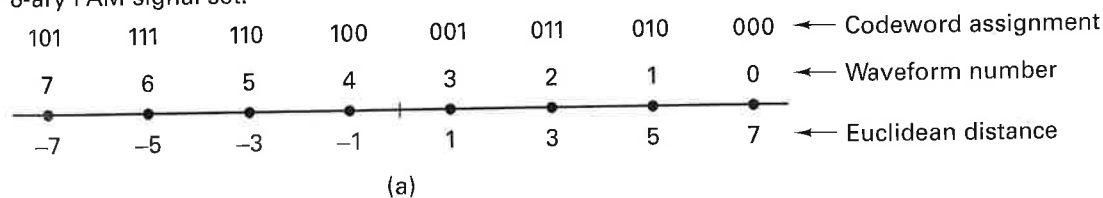
The previous sections dealt with mapping waveforms to trellis transitions, without regard to the final mapping of channel symbols (code bits or codewords) to the trellis transitions. In this section, a TCM example is approached by first specifying an encoder design. Once the encoder circuit is specified, a trellis diagram and the assignment of codewords to trellis transitions are also specified (defined by the circuitry). Thus, in this example, when waveforms are assigned to trellis transitions (and hence, to the underlying codewords), it will no longer be possible to arbitrarily assign codewords to waveforms, as was done earlier when there was no encoder circuit present.

Consider an encoder that uses a rate $2/3$ convolutional code to transmit two information bits per modulation interval. A candidate encoder is shown in Figure 9.29. The rate $2/3$ encoding is accomplished by transmitting one bit from each pair of bits in the input sequence unmodified, and encoding the other bit into two code bits using a rate $1/2$, constraint length $K = 3$ encoder. As the figure shows, only every other bit ($m_2, m_4, \dots$) from the input sequence enters the shift register. One might ask: How good can such a system be, since only 50% of the input bits are receiving the benefits of redundancy? Recall the analogy of the wizard who recognizes that some bits are quite vulnerable and thus they are assigned modulation waveforms with the best distance properties, while other bits are judged to be robust and are assigned waveforms with poorer distance properties. Modulation and coding take place together; the seemingly "uncoded" bits will not be forgotten, but will be receiving the benefits of the best waveform assignments. We emphasize that the encoding and decoding of TCM take place primarily at the waveform level (our

rightmost $K - 1 = 2$ stages in the shift register. The parallel transitions on the trellis of Figure 9.30 are due to the uncoded bits; an uncoded bit is represented as the left-most bit on each trellis transition. Four transitions emerge from each state. For each state, the two upper transitions stem from a pair of input information bits (such as $m_1 m_2$ being 00 and 10, respectively); the two lower transitions stem from the pair being 01 and 11, respectively. Figure 9.30 illustrates the same trellis structure as the trellis in Figure 9.24, except that each transition in Figure 9.30 is labeled with its codeword designation. It is worth repeating that the encoder circuit determines which codewords appear on the trellis transitions; the system designer only assigns waveforms to the transitions. Therefore, once there is a circuit (whose behavior is described by the trellis), any waveform assigned to a trellis transition automatically carries the codeword meaning corresponding to that transition.

Assume that the coded modulation is 8-ary pulse amplitude modulation (8-PAM), as shown in Figure 9.31. Part (a) of the figure illustrates the coded signal set, where the ED of each signal from the center of the signal space is shown in arbitrary units that are equally spaced and symmetrical about zero. Figure 9.31b illustrates the 4-ary PAM reference set, with signal points and distances similarly labeled. The important step in the encoder design is to assign the 8-ary PAM waveforms to the trellis transitions according to the Ungerboeck partitioning rules shown in Figure 9.32. Observing these rules can lead to the same assignment of waveform numbers to trellis transitions as seen in the trellis of Figure 9.24. These waveform assignments, as well as the codewords assigned by the circuit, are shown in Figure 9.30. The most disparate waveform pairs (with distance $d_2 = 8$) have been assigned to the most vulnerable error events—the parallel transitions. Also, as directed by the Ungerboeck rules, the next most distant ($d_1 = 4$) waveforms have been assigned to transitions originating from or joining into the same state. For easy reference, the codeword-to-waveform assignment (the result of waveforms mapped to trellis transitions) is also shown in Figure 9.31a.

8-ary PAM signal set:



4-ary PAM signal set:

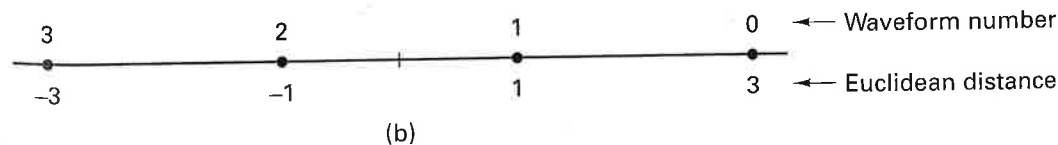


Figure 9.31 Coded 8-ary and uncoded 4-ary PAM signal sets.

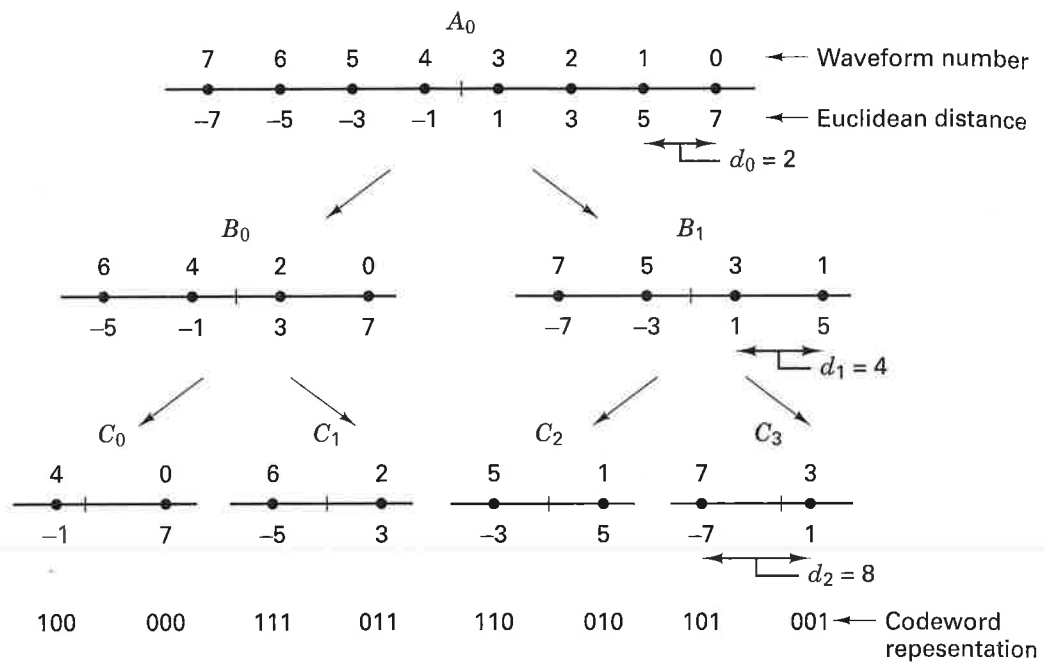


Figure 9.32 Ungerboeck partitioning of 8-PAM signals.

In Figure 9.24, the error event path labeled with waveform numbers 2, 1, 2, is the minimum distance path for this 8-PAM example. The distance from the all-zeros path is computed in the same way as was done in Equation (9.58). In this example, taking the individual distances from Figure 9.32, the computation for d_f yields

$$d_f^2 = d_1^2 + d_0^2 + d_1^2 = 16 + 4 + 16 = 36$$

or,

$$d_f = 6$$

(9.61)

For this modulation type, one can easily verify that the parallel path (with $d = 8$) is not the minimum distance error path (as it was for the 8-PSK case). Next, for finding the 4-PAM reference distance, observe from the arbitrary metric used in Figure 9.31b that $d_{\text{ref}} = 2$. We can now compute the asymptotic coding gain for this example by comparing the squared free ED of the coded system to that of the reference system. But here it is necessary to insure that the distances in the coded and uncoded systems are on an equal footing, meaning that the average power of the signals in each set is the same. In the earlier 8-PSK example, the choice of a unit circle for both the coded and uncoded systems meant that the average signal power was the same for both sets. But that is not the case in this example. Therefore, a modified version of Equation (9.56) for computing the asymptotic gain is

necessary to normalize the effect of unequal signal-set average power [35]. Thus, we write

$$G(\text{dB}) = 10 \times \log_{10} \left(\frac{d_f^2 / S_{av}}{d_{\text{ref}}^2 / S'_{av}} \right) \quad (9.62)$$

where S_{av} and S'_{av} represent average signal power in the coded set and in the reference set, respectively. Distance corresponds to a signal amplitude or voltage; thus, distance squared corresponds to voltage squared, or power. The average signal power of the signal constellation is then computed as

$$S_{av} = \frac{d_1^2 + d_2^2 + \cdots + d_M^2}{M} \quad (9.63)$$

where d_i is the Euclidean distance of the i th signal from the center of the space, and M is the number of codeword symbols in that set. For the 8-PAM coded signal set, illustrated in Figure 9.31a, use of Equation (9.63) yields $S_{av} = 21$. For the 4-ary PAM reference signal set, illustrated in Figure 9.31b, use of Equation (9.63) yields $S'_{av} = 5$.

Using Equation (9.62), the asymptotic coding gain for an 8-PAM coded system with a 4-state trellis, is given by

$$G(\text{dB}) = 10 \times \log_{10} \left(\frac{36/21}{4/5} \right) = 3.3 \text{ dB} \quad (9.64)$$

Larger coding gains can be achieved with an increased number of trellis states (larger constraint length) at the expense of increased decoding complexity. For the rate 2/3 coded 8-ary PAM example discussed here, a 256-state trellis yields a coding gain of 5.83 dB, relative to an uncoded 4-ary PAM signal set [9]. There is only a slight increase in complexity at the transmitter, due to the trellis coding. The decoding problem at the receiver is made more complex, but the use of large scale integrated (LSI) circuits and very high-speed integrated circuits (VHSIC) makes these coding techniques extremely attractive for achieving significant coding gain without bandwidth expansion.

9.10.6 Multi-Dimensional Trellis-Coded Modulation

It was pointed out in Section 9.9.3 that for a given data rate, signaling in a two-dimensional space can provide the same error performance with less average power than signaling in a one-dimensional PAM space. This is accomplished by choosing signal points on a two dimensional lattice from within a circular-shaped boundary rather than a rectangular-shaped one. By performing similar shaping at higher dimensions, the potential energy savings approaches 1.53 dB as N approaches infinity. For practical systems, such multidimensional signaling can achieve an energy savings, called the *shaping gain*, of approximately 1-dB over one-dimensional signaling [21, 36, 37]. In the V.34 High-Speed Modem Standard, 16-dimensional QAM is specified, where the technique for mapping bits to higher-dimensional signal-constellation points is called *shell mapping*, and can provide a

shaping gain of 0.8 dB [16]. Using four-, eight-, or 16-dimensional signal constellations can have some advantages over the usual two-dimensional schemes. These advantages include smaller, two-dimensional constellation building blocks, better tolerance-to-phase ambiguities, and a better trade-off between coding gain and complexity. Wei [36] has presented and evaluated a number of such schemes. (For readers who may be interested in reading further in the area of coded modulation, particularly trellis-coded modulation, references [38–46] are recommended.)

9.11 CONCLUSION

In this chapter we have integrated some of the ideas in earlier Chapters dealing with modulation and coding. We have reviewed the basic system design goals: to maximize data rate while simultaneously minimizing error probability, bandwidth, E_b/N_0 , and complexity. We examined the trade-offs heuristically on two performance planes: the error probability plane and the bandwidth efficiency plane. The former explicitly illustrates the P_B versus E_b/N_0 trade-offs while only implicitly displaying the bandwidth expenditure. The latter explicitly illustrates the R/W versus E_b/N_0 trade-offs while only implicitly displaying the P_B performance. We outlined typical steps that need to be considered in meeting bandwidth power, and error-performance requirements of a digital communication system, and we discussed some of the basic constraints to improvement without limit. The Nyquist criterion establishes that we cannot continue to reduce system bandwidth indefinitely. There is a theoretical limitation; in order to transmit R_s symbols/second without intersymbol interference, we must utilize a minimum of $R_s/2$ hertz of bandwidth. The Shannon–Hartley theorem relates to the power–bandwidth trade-off and results in another important limitation, the Shannon limit. The Shannon limit of -1.6 dB is the theoretical minimum amount of E_b/N_0 that is necessary (in concert with channel coding) to achieve an arbitrarily low error probability over an AWGN channel. The more general limitation is the channel capacity, above which there cannot be error-free signaling. We have also examined some of the bandwidth-efficient modulation schemes, such as minimum shift keying (MSK), quadrature amplitude modulation (QAM), and trellis-coded modulation. The latter technique offers an attractive way to obtain coding gain without paying the price of additional bandwidth.

REFERENCES

1. *IEEE Personal Communications*, Special Issue on Software Radio, vol. 6, no. 4, August 1999.
2. Nyquist, H., "Certain Topics on Telegraph Transmission Theory," *Trans. AIEE*, vol. 47, April 1928, pp. 617–644.

ial constella-
emes. These
locks, better
ing gain and
chemes. (For
modulation,
aded.)

pters dealing
sign goals: to
, bandwidth,
two perfor-
y plane. The
mplicitly dis-
e R/W versus
We outlined
r, and error-
we discussed
uist criterion
nitely. There
out intersym-
dwidth. The
and results in
of -1.6 dB is
rt with chan-
GN channel.
re cannot be
idth-efficient
re amplitude
que offers an
of additional

, no. 4, August

AIEE, vol. 47,

's Chap. 9

3. Shanon, C. E., "A Mathematical Theory of Communication," *BSTJ*, vol. 27, 1948, pp. 379-423, 623-657.
4. Shannon, C. E., "Communication in the presence of Noise," *Proc. IRE*, vol. 37, no. 1, January 1949, pp. 10-21.
5. Bedrosian, E., "Spectrum Conservation by Efficient Channel Utilization," *Rand Corp., Report WN-9275-ARPA*, Contract DAHC-15-73-C-0181, Santa Monica, California, October, 1975.
6. Ungerboeck, G., "Trellis-Coded Modulation with Redundant Signal Sets," Part I and Part II, *IEEE Communications Magazine*, vol. 25, February 1987, p. 5-21.
7. Hodges, M.R.L., "The GSM Radio Interface," *British Telecom Tech. J.*, vol. 8, no. 1, January 1990, pp. 31-43.
8. Anderson, J. B., and Sundberg, C-E. W., "Advances in Constant Envelope Coded Modulation," *IEEE Commun., Mag.*, vol. 29, no. 12, Dec. 1991, pp. 36-45.
9. Clark, G. C. Jr., and Cain, J. B., *Error Correction Coding for Digital Communications*, Plenum Press, New York, 1981.
10. Lindsey, W. C., and Simon, M. K., *Telecommunication Systems Engineering*, Prentice-Hall, Englewood Cliffs, NJ, 1973.
11. Sklar, B., "Defining, Designing, and Evaluating Digital Communication Systems," *IEEE Commun. Mag.*, Vol. 31, no. 11, Nov. 1993, pp. 92-101
12. Korn, I., *Digital Communications*, Van Nostrand Reinhold Co., New York, 1985.
13. Viterbi, A. J., *Principles of Coherent Communications*, McGraw-Hill Book Co., New York, 1966.
14. Lin, S., and Costello, D. J., Jr., *Error Control Coding: Fundamentals and Applications*, Prentice-Hall, Englewood Cliffs, NJ, 1983.
15. Odenwalder, J. P., *Error Control Coding Handbook*, Linkabit Corporation, San Diego, California, July 15, 1976.
16. Forney, G. D., Jr., et. al., "The V.34 High-Speed Modem Standard," *IEEE Communications Magazine*, December 1996.
17. Pasupathy S., "Minimum Shift Keying: A Spectrally Efficient Modulation," *IEEE Commun. Mag.*, July 1979, pp. 14-22.
18. Gronemeyer, S. A., and McBride, A. L., "MSK and Offset QPSK Modulation," *IEEE Trans. Commun.*, vol. COM-24, August 1976, pp. 809-820.
19. Simon, M. K. "A Generalization of Minimum Shift Keying (MSK) Type Signaling Based Upon Input Data Symbol Pulse Shaping," *IEEE Trans. Commun.*, vol. COM-24, August 1976, pp. 845-857.
20. Leib, H., and Pasupathy, S., "Inherent Error Control Properties of Minimum Shift Keying," *IEEE Communications Mag.*, vol. 31, no. 1, January 1993, pp. 52-61.
21. Forney, G. D. Jr. et. al., "Efficient Modulation for Bandlimited Channels," *IEEE J. Selected Areas in Commun.*, vol. SAC-2, no. 5, September 1984, pp. 632-647.
22. Thomas, C. M., Weidner, M. Y., and Durrani, S. H., "Digital Amplitude-Phase Keying with M-ary Alphabets," *IEEE Trans. Commun.*, vol. COM-22, no. 2, February 1974, pp. 168-180.
23. Lucky, R. W., and Hancock, J. C., "On the Optimum Performance of N-ary Systems Having Two Degrees of Freedom," *IRE Trans. on Commun. Sys.*, vol. CS-10, June 1962, pp. 185-192.

24. Campopiano, C. N., and Glazer, B. G., "A Coherent Digital Amplitude and Phase Modulation Scheme," *IRE Trans. on Commun. Sys.*, vol. CS-10, June 1962, pp. 90-95.
25. Cahn, C. R., "Combined Digital Phase and Amplitude Modulation Communication Systems," *IRE Trans. on Commun. Tech.*, Sept. 1960.
26. Foschini, G. J., and Gitlin, R. D., "Optimization of Two Dimensional Signal Constellations in the Presence of Gaussian Noise," *IEEE Trans. Commun.*, vol. COM-22, no. 1, January 1974, pp. 23-38.
27. Welty, G. R., and Jhong, S. L., "Digital Transmission with Coherent Four-Dimensional Modulation," *IEEE Trans. Inform. Theory*, vol. IT-20, no. 4, July 1974, pp. 497-502.
28. Gersho, A., and Lawrence, V. B., "Multidimensional Signal Constellations for Voiceband Data Transmission," *IEEE J. Selected Areas in Commun.*, vol. SAC-2, no. 5, September 1984, pp. 687-702.
29. Zetterberg, L. H., and Brandstrom, H., "Codes for Combined Phase and Amplitude Modulated Signals in a Four-Dimensional Space," *IEEE Trans. Commun.*, vol. COM-25, no. 9, September 1977, pp. 943-950.
30. Wilson, S. G., Sleeper, H. A., and Srinath, N. K., "Four-Dimensional Modulation and Coding: An Alternative to Frequency Reuse," *IEEE 1984 Int'l. Commun. Conf.*, pp. 919-923.
31. Ungerboeck, G., "Channel Coding with Multilevel/Phase Signals," *IEEE Trans. Inform. Theory*, vol. IT-28, January 1982, pp. 55-67.
32. Forney, G. D., "The Viterbi Algorithm," *Proceedings of the IEEE*, vol. 61, no. 3, March 1978, pp. 268-278.
33. Divsalar, D., Simon, M. K., and Yuen, J. H., "Trellis Coding with Asymmetric Modulations," *IEEE Trans. Commun.* vol. COM-35, no. 2, February 1987.
34. Wei, J.-F., "Rotationally Invariant Convolutional Channel Coding with Expanded Signal Space—Parts I and II," *IEEE J. Sel. Areas Commun.*, vol. SAC-2, no. 5, Sept. 1984, pp. 659-686.
35. Thapar, H. K., "Real-Time Application of Trellis Coding to Highspeed Voiceband Data Transmission," *IEEE J. Sel. Areas Commun.*, vol. SAC-2, no. 5, Sept. 1984, pp. 648-658.
36. Wei, L.-F., "Trellis-Coded Modulation with Multidimensional Constellations," *IEEE Trans. Information Theory*, vol. IT-33, no. 4, July 1987, pp. 483-501.
37. Tretter, S. A., "An Eight-Dimensional 64-State Trellis Code for Transmitting 4 Bits Per 2-D Symbol," *IEEE J. on Sel. Areas of Commun.*, vol. 7, no. 9, December 1989, pp. 1392-1395.
38. Kato, S., Morikura, M., and Kubota, S., "Implementation of Coded Modems," *IEEE Communications Magazine*, vol. 29, no. 12, December, 1991, pp. 88-97.
39. Special Issue on Coded Modulation, *IEEE Communications Magazine*, vol. 29, no. 12, December 1991.
40. Biglieri, E., et. al., *Introduction to Trellis-Coded Modulation with Application*, MacMillan, New York, NY, 1991.
41. Edbauer, F., "Performance of Interleaved Trellis-Code Differential 8-PSK Modulation over Fading Channels," *IEEE J. on Selected Areas in Commun.*, vol. 7, no. 9, December 1989, pp. 1340-1346.
42. Rimoldi, B., "Design of Coded CPFSK Modulation Systems for Bandwidth and Energy Efficiency," *IEEE Transactions on Communications*, vol. 37, no. 9, September 1989, pp. 897-905.

43. Viterbi, A. J., et. al., "A Pragmatic Approach to Trellis-Coded Modulation," *IEEE Communications Magazine*, vol. 27, no. 7, July 1989, pp 11–19.
44. Divsalar, D., and Simon, M. K., "The Design of Trellis Coded MPSK for Fading Channels: Performance Criteria," *IEEE Trans. on Comm.*, vol. 36, no. 9, September 1988, pp. 1004–1012.
45. Divsalar, D., and Simon, M. K., "The Design of Trellis Coded MPSK for Fading Channels: Set Partitioning for Optimum Code Design," *IEEE Trans. on Comm.*, vol. 36, no. 9, September 1988, pp. 1013–1021.
46. Divsalar, D., and Simon, M. K., "Multiple Trellis Coded Modulation (MTCM)," *IEEE Trans. on Commun.*, vol. 36, no. 4, April 1988, pp. 410–419.

PROBLEMS

- 9.1. Consider a voice-grade telephone circuit with a bandwidth of 3 kHz. Assume that the circuit can be modeled as an AWGN channel.
 - (a) What is the capacity of such a circuit if the SNR is 30 dB?
 - (b) What is the minimum SNR required for a data rate of 4800 bits/s on such a voice-grade circuit?
 - (c) Repeat part (b) for a data rate of 19,200 bits/s.
- 9.2. Consider that a 100-kbits/s data stream is to be transmitted on a voice-grade telephone circuit (with a bandwidth of 3 kHz). Is it possible to approach error-free transmission with a SNR of 10 dB? Justify your answer. If it is not possible, suggest system modifications that might be made.
- 9.3. Consider a source that produces six messages with probabilities $\frac{1}{2}$, $\frac{1}{4}$, $\frac{1}{8}$, $\frac{1}{16}$, $\frac{1}{32}$, and $\frac{1}{32}$. Determine the average information content in bits, of a message.
- 9.4. A given source alphabet consists of 300 words, of which 15 occur with probability 0.06 each and the remaining 285 words occur with probability 0.00035 each. If 1000 words are transmitted each second, what is the average rate of information transmission?
- 9.5. (a) Find the average capacity in bits per second that would be required to transmit a high-resolution black-and-white TV signal at the rate of 32 pictures per second if each picture is made up of 2×10^6 picture elements and 16 different brightness levels. All picture elements are assumed to be independent and all levels have equal likelihood occurrence.
 - (b) For color TV, this system additionally provides for 64 different shades of color. How much more system capacity is required for a color system compared to the black and white system?
 - (c) Find the required capacity if 100 of the possible brightness-color combinations occur with a probability of 0.003 each, 300 of the combinations occur with a probability of 0.001, and 624 of the combinations occur with a probability of 0.00064.
- 9.6. Prove that entropy is maximized when all source outputs have equal probability.
- 9.7. Compute the equivocation or message uncertainty in bits per character for a textual transmission using 7-bit ASCII coding. Assume that each character is equally likely and that the noise on the channel results in a bit error probability of 0.01.

- 9.8. Suppose a binary noncoherent FSK link has a maximum data rate of 2.4 kbits/s without ISI over a channel whose nominal bandwidth is 2.4 kHz. Suggest ways of increasing the data rate under the following system constraints.
- The system is power limited.
 - The system is bandwidth limited.
 - The system is both power and bandwidth limited.
- 9.9. Table P9.1 characterizes four different satellite-to-earth-terminal links. For each link assume that the space loss is 196 dB, the margin is 0 dB, and there are no other incidental losses. For each link, plot an operating point on the bandwidth efficiency plane, R/W versus E_b/N_0 , and characterize the link according to one of the following descriptions: bandwidth limited, severely bandwidth limited, power limited, and severely power limited. Justify your answers.

TABLE P9.1 Downlink Capacity for Four Satellite Links

Satellite	Receive Terminal	Maximum Data Rate
INTELSAT IV EIRP = 22.5 dBW Bandwidth = 36 MHz	Large fixed antenna diameter = 30 m $G/T = 40.7$ dB/K	165 Mbits/s
DSCS II EIRP = 28 dBW Bandwidth = 50 MHz	Shipboard antenna diameter = 4 ft $G/T = 10$ dB/K	100 kbits/s
DSCS II EIRP = 28 dBW Bandwidth = 50 MHz	Large fixed antenna diameter = 60 ft $G/T = 39$ dB/K	72 Mbits/s
GAPSAT/MARISAT EIRP = 28 dBW Bandwidth = 500 kHz	Aircraft antenna gain = 0 dB $G/T = -30$ dB/K	500 bits/s

- 9.10. You are required to make modulation and error-correction code choices for a real-time communication system operating over an AWGN channel with an available bandwidth of 2400 Hz. The available E_b/N_0 is 14 dB. The required data rate and bit-error probability are 9600 bits/s and 10^{-5} , respectively. You are allowed to choose one of two modulation types—either noncoherent orthogonal 8-FSK or 16-QAM with matched filter detection. You are allowed to choose one of two codes—either the (127, 92) BCH code or a rate $\frac{1}{2}$ convolutional code that provides 5 dB of coding gain at a bit-error probability of 10^{-5} . Assuming ideal filtering, verify that your choices achieve the desired bandwidth- and error-performance requirements.
- 9.11. You are required to make the same kind of design choices of modulation and coding as in Problem 9.10. The requirements and assumptions in this problem are the same as in Problem 9.10, except that the available bandwidth is now increased to 40 kHz, and the available E_b/N_0 is now reduced to 7.3 dB. Verify that your choices achieve the desired bandwidth- and error-performance requirements.
- 9.12. You are required to make the same kind of design choices as in Problem 9.10. The requirements and assumptions in this problem are the same as in Problem 9.10, except that the AWGN channel now manifests deep fades lasting up to 100 ms in duration. The available bandwidth is now 3400 Hz, and the available E_b/N_0 is now 10 dB. In addition to making modulation and coding choices you are asked to select an interleaver design (see Section 8.2) to combat the worst-case fading problem. You are

f 2.4 kbits/s with-
ways of increas-

ks. For each link
re no other inci-
width efficiency
of the following
ver limited, and

Rate
s/s
/s
/s

choices for a real-
with an available
data rate and bit-
allowed to choose
PSK or 16-QAM
vo codes—either
s 5 dB of coding
verify that your
irements.

ation and coding
em are the same
eased to 40 kHz,
choices achieve

problem 9.10. The
problem 9.10, ex-
o 100 ms in dura-
/N₀ is now 10 dB.
d to select an in-
problem. You are

allowed to choose one of two types—either a 16×32 block interleaver or a 150×300 convolutional interleaver. Verify that your choices achieve the desired bandwidth- and error-performance requirements, as well as providing the needed mitigation against the longest fade duration.

- 9.13. (a)** Consider a real-time communication system operating over an AWGN channel that uses 8-PSK modulation with Gray coding. Select an error-correcting code that provides a decoded bit-error probability of at most 10^{-7} when the received P_r/N_0 is 70 dB-Hz, and the data rate is 1 Mbit/s. You are permitted to choose one of the following codes: (24, 12) Extended Golay, or (127, 64) BCH, or (127, 36) BCH. The input-output transfer functions for these codes appear in Figure 6.21. To guide you in making a selection, consider that at $P_B = 10^{-7}$, the transfer functions intercept the abscissa at the following points: the (24, 12) code intercepts at 3×10^{-3} , the (127, 64) code intercepts at 1.3×10^{-2} , and the (127, 36) BCH code intercepts at 3×10^{-2} .
- (b)** The general appearance of the code transfer functions may have caused you to form an initial judgement as to which code is the best choice for the stated design requirements. Does your final choice agree with your initial guess? Do the results of part a) surprise you? Explain your results in the context of the two mechanisms at work when using error-correction coding in a real-time communication system.
- (c)** How much coding gain in decibels does your code choice in part a) provide?

- 9.14.** Consider a real-time communication satellite system, operating over an AWGN channel (disturbed by periodic fades). The overall link is described by the following specifications from a mobile transmitter to a low-earth-orbit satellite receiver:

Data rate $R = 9600$ bits/s
Available bandwidth $W = 3000$ Hz
Link margin $M = 0$ dB (see Section 5.6)
Carrier frequency $f_c = 1.5$ GHz
EIRP = 6 dBW
Distance between transmitter and receiver $d = 1000$ km
Satellite receiver figure of merit $G/T = 30$ dBI
Receiver antenna temperature $T_A = 290$ K
Line loss from the receiver antenna to the receiver, $L = 3$ dB
Receiver noise figure $F = 10$ dB
Losses due to fading $L_f = 20$ dB
Other losses $L_o = 6$ dB

You are allowed to choose one of two modulation schemes—MPSK with Gray coding, or noncoherent orthogonal MFSK—such that the available bandwidth is not exceeded and power is conserved. For error-correction coding, you are to choose one of the (127, k) BCH codes from Table 9.2 that provides the most redundancy, but still meets the bandwidth constraints. Calculate the output decoded bit-error probability. How much coding gain, if any, characterizes your choices. Hint: Proceed by calculating parameters in the following order, E_b/N_0 , E_s/N_0 , $P_E(M)$, p_c , P_B . When using Equation (9.41) for computing decoded bit-error probability, a small E_b/N_0 necessitates using many terms in the summation. Hence, computer assistance is helpful here.

- 9.15.** You are required to provide a real-time communication system to support 9600 bits/s with a required bit-error probability of at most 10^{-5} within an available bandwidth of

2700 Hz. The predetection P_r/N_0 is 54.8 dB-Hz. Choose one of two modulation schemes—either MPSK with Gray coding or noncoherent orthogonal MFSK, such that the available bandwidth is not exceeded and power is conserved. If error-correction coding is needed, choose the simplest (shortest) code in Table 9.3 that provides the needed error performance, but does not exceed the available bandwidth. Verify that your design choices result in meeting the requirements.

- 9.16. (a) For a fixed error probability, show that the relationship between alphabet size M and required average power for MPSK versus QAM can be expressed as

$$\frac{\text{average power for MPSK}}{\text{average power for QAM}} \approx \frac{3M^2}{2(M-1)\pi^2}$$

- (b) Discuss the advantage of one type of signaling over the other.

- 9.17. Consider a telephone modem operating at 28.8 kbits/s that uses trellis-coded QAM modulation.

- (a) Calculate the bandwidth efficiency of such a modem, assuming that the usable channel bandwidth is 3429 Hz.
 (b) Assuming AWGN and an available $E_b/N_0 = 10$ dB, calculate the theoretically available capacity in the 3429-Hz bandwidth.
 (c) What is the required E_b/N_0 that will enable a 3429-Hz bandwidth to have a capacity of 28.8 kbits/s?

- 9.18. Figure 9.17 shows several 16-ary symbol constellations.

- (a) For the (5, 11) circular constellations, compute the minimum radial distances r_1 and r_2 if the minimum distance between each symbol must be 1 unit.
 (b) Compute the average signal power for the (5, 11) circular constellation, and compare it to the average signal power for the 4×4 ($M = 16$) square constellation (with the same minimum distance between symbols.)
 (c) Why might the square constellation be more practical?

- 9.19. Consider that the rate $\frac{2}{3}$ trellis-coded system of Section 9.10.5 is used over a binary symmetric channel (BSC). Assume that the initial encoder state is the 00 state. At the output of the BSC, the sequence $\mathbf{Z} = (1\ 1\ 1\ 0\ 0\ 1\ 1\ 0\ 1\ 0\ 1\ 1\ \text{rest all "0"})$ is received.

- (a) Find the maximum likelihood path through the trellis diagram and determine the first 6 decoded information bits. If a tie occurs between any two merged paths, choose the upper branch entering the particular state.
 (b) Determine if any channel bits in $\mathbf{Z}$ had been inverted by the channel during transmission, and if so, identify them.
 (c) Explain how you would proceed with the problem if the channel were specified as a Gaussian channel instead of a BSC.

- 9.20. Find the asymptotic coding gain for a 4-state trellis-coded modulation (TCM) scheme. A code rate of $\frac{2}{3}$ is achieved by using the encoder configuration shown in Figure 9.29, where 50% of the data bits are inputted to a rate $\frac{1}{2}$ convolutional encoder, and the remaining 50% are passed directly to the output. The coded modulation is 8-PAM as seen in the upper part of Figure 9.31. Use a 4-PAM signal set with amplitudes of $-16, -1, +1, +16$ to serve as the reference set. Does your answer appear to violate Shannon's theorem that predicts a coding-gain limit of about 11–12 dB? Would anyone ever use a reference set, such as the one proposed here? Perhaps the notion of coding gain for combined modulation/coding schemes is slightly different than that for coding alone. In this context explain your results.

- 9.21. Find the asymptotic coding gain for an 8-state trellis-coded modulation scheme having the following features. The coded modulation is 8-PSK, and the uncoded reference is 4-PSK. The trellis structure is formed from time t_k to time t_{k+1} in the following way: Number the states (from top to bottom) with the arbitrary designations 1–8. Then, connect each of the states 1, 3, 5, and 7 at time t_k to states 1–4 at time t_{k+1} . Similarly, connect each of the states 2, 4, 6, and 8, at time t_k , to states 5–8, at time t_{k+1} . Next, draw three sections (time intervals) of the trellis structure. Then, make waveform assignments to branches, and search for the shortest error path.

QUESTIONS

- 9.1. Why do binary and 4-ary orthogonal frequency shift keying (4-FSK) manifest the same *bandwidth-efficiency* relationship? (See Section 9.5.1.)
- 9.2. For MPSK modulation, *bandwidth efficiency* increases with higher-dimensional signaling, but for MFSK, it decreases. Explain why this is the case. (See Sections 9.7.2 and 9.7.3.)
- 9.3. Consider the assortment of signaling elements that flow through a typical system, and describe the subtle energy and rate transformations among them: from data-bits to channel-bits to symbols to chips. (See Section 9.7.7.)
- 9.4. The steep decrease in MSK *spectral sidelobes* in Figure 9.15 illustrates why MSK is considered to be more spectrally efficient than QPSK. How do you explain then that the QPSK spectrum has a *narrower mainlobe* than the MSK spectrum? (See Section 9.8.2.)
- 9.5. In Chapter 4, we presented binary phase shift keying (BPSK) and quaternary phase shift keying (QPSK) as manifesting the same bit-error-probability relationship. (See Section 4.8.4.) Does the same hold true for M -ary pulse amplitude modulation (M -PAM) and M^2 -ary quadrature amplitude modulation (M^2 -QAM); that is, do these schemes also manifest the same *bit-error probability*? (See Section 9.8.3.1.)
- 9.6. Although trellis-coded modulation schemes do not require additional bandwidth or power, their use still involves a *trade-off*. What is the cost of achieving coding gain with TCM? (See Section 9.10.)
- 9.7. What is meant by the *state* of a finite-state machine? (See Section 9.10.)
- 9.8. When using TCM, how much *waveform redundancy* is sufficient to realize the benefits of coding (improved error performance or increased capacity)? (See Section 9.10.1.1.)
- 9.9. For TCM schemes, define what is meant by *asymptotic coding gain*, and from that definition, explain what goals one should strive for in the construction of TCM codes. (See Section 9.10.3.2.)
- 9.10. In a TCM trellis diagram, when are *parallel paths* required in order to comply with the Ungerboeck partitioning rules? What is the result of violating the rules? (See Section 9.10.4.1.)

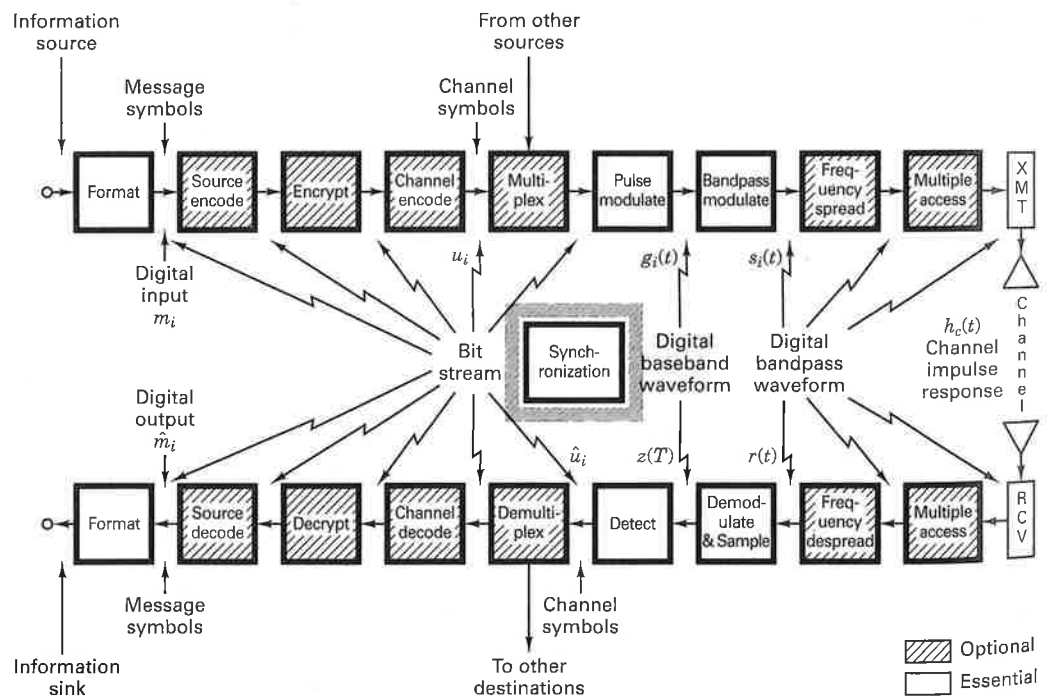
EXERCISES

Using the Companion CD, run the exercises associated with Chapter 9.

CHAPTER 10

Synchronization

Maurice A. King, Jr.
The Aerospace Corporation
El Segundo, California

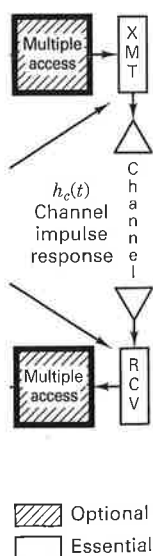


10.1 INTRODUCTION

10.1.1 Synchronization Defined

In almost every discussion of receiver or demodulator performance, some level of signal synchronization is assumed, although this assumption is often not explicitly stated. For example, in the case of coherent phase demodulation (PSK), the receiver is assumed to be able to generate reference signals whose phases are identical (except perhaps for a constant offset) to those of the signaling alphabet at the transmitter. These reference signals are compared with the incoming signals in the process of making maximum-likelihood symbol decisions.

In order to be able to generate these reference signals, the receiver has to be in synchronization with the received carrier. This means that there has to be phase concurrence between the incoming carrier and a replica of it in the receiver. In other words, if there were no information modulated on the incoming carrier, the incoming carrier and the replica in the receiver would pass through zero simultaneously. This is what is known as being in *phase lock* and is a condition that must be closely approximated if coherently modulated signals are going to be accurately demodulated at the receiver. Being in phase lock means that the receiver's local oscillator is synchronized in both frequency and phase with the received signal. If the information-bearing signal is not modulated directly on the carrier but indirectly through the use of a subcarrier, both the phase of the carrier and that of the subcarrier must be determined. If the carrier and subcarrier are not kept in phase synchronism by the transmitter (they typically are not), this will require the generation



of a replica of the subcarrier by the receiver, where the phase of the subcarrier replica is controlled separately from that of the carrier replica. This will enable the receiver to achieve phase lock on both the carrier and subcarrier.

It is also assumed that the receiver has accurate knowledge of when an incoming symbol started and when it is over. This knowledge is required in order to know the proper symbol integration interval—the interval over which energy is integrated prior to making symbol decisions. Clearly if the receiver integrates over an interval of an inappropriate length, or over an interval that spans two symbols, the ability to make accurate symbol decisions will be degraded.

It can be seen that symbol synchronization and phase synchronization are similar in that they both involve producing in the receiver a replica of a portion of the transmitted signal. For phase synchronization, it was an accurate replica of the carrier. For symbol synchronization, it is a square wave at the symbol transition rate. The receiver must, in effect, be able to produce a square wave that will transition through zero simultaneously with the incoming signal's transitions between symbols. A receiver that is able to do this can be said to have symbol synchronization, or to be in *symbol lock*. Since there are typically a very large number of carrier cycles per symbol period, this second level of synchronization is much coarser than phase synchronization and is usually done with different circuitry than that used for phase synchronization.

In many communication systems an even higher level of synchronization is required. This is usually called *frame synchronization*. Frame synchronization is required when the information is organized in blocks, or messages of some uniform number of symbols. This will occur, for example, if a block code is used for forward error control, or if the communications channel is being time-shared, on a regular basis, by several users (TDMA). In the case of block coding, the decoder needs to know the location of boundaries between code words in order to decode the message correctly. In the case of a time-shared channel, it is necessary to know where the location of boundaries between channel users are, in order to route the information appropriately. Similar to symbol synchronization, frame synchronization is equivalent to being able to generate a square wave at the frame rate, with the zero crossings coincident with the transitions from one frame to the next.

Most digital communications systems using coherent modulation require all three levels of synchronization: phase, symbol, and frame. Systems using noncoherent modulation techniques will typically require symbol and frame synchronization, but since the modulation is not coherent, accurate phase lock is not required. Instead, noncoherent systems require *frequency synchronization*. Frequency synchronization differs from phase synchronization in that the replica of the carrier that is generated by the receiver is allowed to have an arbitrary constant phase offset from the received carrier. Receiver designs can be simplified by removing the requirement to determine the exact value of the incoming carrier phase. Unfortunately, as is shown in the discussion of modulation techniques, this simplification carries a penalty in terms of degraded performance versus signal-to-noise ratio. The relative trade-offs of synchronization levels versus performance and system versatility are discussed further in the next section.

All of the discussion thus far had been oriented toward the receiving end of a communication link. There are instances, however, when the transmitter assumes the more active role in synchronization—by varying the timing and frequency of its transmissions to correspond to the expectations of the receiver. An example of this situation is a satellite communication network, where many terrestrial terminals are beaming signals toward a single satellite receiver. In most of these cases the transmitter relies on a return path from the receiver to determine the accuracy of its synchronization. Thus, transmitter synchronization often implies two-way communications or a network in order to be successful. Thus, transmitter synchronization is often called *network synchronization*. Transmitter or network synchronization is discussed later in this chapter.

10.1.2 Costs versus Benefits

There is a cost associated with the need for receiver synchronization. Each additional level of synchronization implies more cost. The most obvious cost is in the need for additional hardware or software in the receiver for acquisition and tracking. Possibly less obvious costs lie in the extra time required to achieve synchronization before commencing communications, or in the energy expended by the transmitter on signals to be used at the receiver as acquisition or tracking aids. In the face of these costs to the system, one might question why a communications system designer would consider a system design requiring a high degree of synchronization. The answer: improved performance and versatility.

Consider a standard commercial analog AM radio. This radio may be considered part of a broadcast communication system involving a central transmitter and many receivers. This communication system involves no synchronization. However, the receiver passband must be wide enough to accommodate not only the information-bearing signal, but also any fluctuations in the carrier, due perhaps to Doppler shift* or drift in the transmitter's frequency reference. This requirement in the receiver passband means that additional noise energy is passed to the detector, over and above the amount theoretically required by the bandwidth of the information. A somewhat more complicated receiver that employs a carrier frequency tracking loop would be able to keep a narrow passband filter centered about the carrier, thereby substantially reducing the detected noise energy and improving the received signal-to-noise ratio. Thus, although a standard radio may be perfectly adequate for reception of signals from large transmitters a few tens of kilometers distant, it may prove totally inadequate under less benign conditions.

For digital communications, examples of the trade-off between performance and receiver complexity are often seen in the choice of modulation. Among the

*An offset in frequency as perceived by a receiver, from the nominally transmitted frequency caused by relative motion of the transmitter and receiver. Ignoring second- and higher-order effects the value of the frequency offset, Δf , is given by Vf_0/c , where V is the relative velocity (positive when the relative distance between transmitter and receiver is being reduced), f_0 the nominal frequency and c the speed of light.

simplest digital receivers are those designed to be used with noncoherently detected binary FSK. The only synchronization requirements are bit timing and frequency tracking. However, the same bit error probability could be achieved with approximately 4 dB less signal-to-noise ratio if the modulation is coherent BPSK. The disadvantage of BPSK is that the receiver requires accurate phase tracking, which can present a complex design problem if the signals experience high Doppler rates** or fading. (See Chapter 15.)

A third cost-versus-performance trade-off involves the use of error-control coding. As was established in earlier chapters, there are substantial performance advantages in the use of appropriate error-control coding techniques. The cost, however, measured in receiver complexity, can be high. For a block decoder to operate properly requires the receiver to achieve block, message, or frame synchronization. This is a procedure over and above the usual decoding procedure, although some error-correcting codes have been designed with block synchronization aids built in [1]. Convolutional codes also require some degree of additional synchronization in order to provide optimum performance. Although the performance analysis of convolutional codes often makes the assumption that the input data sequence is infinitely long, in practice it is not. In order to provide the minimum error probability, the decoder must know the beginning state (usually all zeros) when the data sequence will begin, the eventual ending state, and when the ending state is to be reached. Knowing when the beginning state was left and when the ending state is to be reached, however, is equivalent to having frame synchronization. In addition, the decoder will have to know how to group the channel symbols in order to make branch decisions. This is also a synchronization requirement.

The trade-offs discussed thus far have been in terms of the performance versus complexity of individual links and receivers. The ability to synchronize has a large potential consequence in terms of system efficiency and versatility as well. Frame synchronization allows the use of advanced, versatile, multiple-access techniques, such as the variety of demand-assignment-multiple-access (DAMA) schemes, which have become increasingly popular as communication channel resources become increasingly scarce. In addition, the use of spread-spectrum techniques—both as multiple access schemes and for interference rejection—requires a high level of system synchronization. (Spread-spectrum techniques are treated in Chapter 12.) It will be seen that these techniques provide the potential for a great deal of system versatility, which is a very valuable feature if the system encounters changing or unstable conditions, such as the effects of intentional and unintentional interference from external sources.

10.1.3 Approach and Assumptions

There have been at least two substantial developments in the general area of synchronization since the first edition of this text. One has been the emergence, and then near total dominance, of signal processing (including synchronizers) by sam-

**The rate of change of the Doppler shift. This rate sets requirements on the tracking ability of the phase tracking loop.

pled data techniques. The other has been the publication of several book-length treatments of synchronization [2–4]. This single chapter will not attempt the treatment of a full-length text. The goal here is to provide a broad intuitive understanding of the issues, rather than attempt to describe a catalog of synchronizer-design methods. Thus, we will generally follow a traditional analog development knowing that these principles apply equally well to sampled-data systems, even if the implementation of the synchronizers differ. Phase-locked-loops are commercially available as relatively small gate-count chips, or as one part of a larger signal-processing device. It is assumed that the reader interested in modern design implementations will be able to make the reasonable straightforward transition from the principles presented here to the basic sampled data representations.

10.2 RECEIVER SYNCHRONIZATION

All digital communication systems require some degree of synchronization to incoming signals by the receivers. In this section the fundamentals of the various levels of receiver synchronization are discussed. The discussion begins with the basic levels of synchronization required for coherent reception—frequency and phase synchronization—and a brief discussion of the principles of phase-locked-loop (PLL) operation and design. The discussion then broadens into the topic of symbol synchronization. Some degree of symbol synchronization is required for all digital communications reception, either coherent or noncoherent. The final topics in the section are receiver frame synchronization and techniques for achieving and maintaining it.

10.2.1 Frequency and Phase Synchronization

At the heart of nearly all synchronization circuits is some version of a phase-locked-loop (PLL). In modern digital receivers this loop may be difficult to recognize, but the functional equivalent is essentially always present. A schematic diagram of the basic PLL is given in Figure 10.1. Phase-locked loops are servo-control loops, whose controlled parameter is the phase of a locally generated replica of the incoming carrier signal. Phase-locked loops have three basic components: a phase detector, a loop filter, and a voltage-controlled oscillator (VCO). The phase detector is a device that produces a measure of the difference in phase between an incoming signal and the local replica. As the incoming signal and the local replica change with respect to each other, the phase difference (or phase error) becomes a time-varying signal into the loop filter. The loop filter governs the PLL's response to these variations in the error signal. A well-designed loop should be able to track changes in the incoming signal's phase but not be overly responsive to receiver noise. The VCO is the device that produces the carrier replica. The VCO, as the name implies, is a sinusoidal oscillator whose frequency is controlled by a voltage level at the device input. In Figure 10.1, the phase detector is shown as a multiplier, the loop filter is described by its impulse response function $f(t)$, with Fourier transform $F(\omega)$, and the VCO is so indicated.

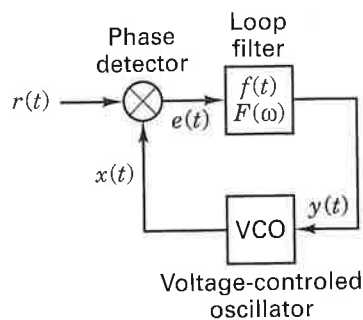


Figure 10.1 Schematic of the basic phase-locked loop.

A VCO is an oscillator whose output frequency is a linear function of its input voltage over some range of input and output. A positive input voltage will cause the VCO output frequency to be greater than its uncontrolled value, ω_0 , while a negative voltage will cause it to be less. Phase lock is achieved by feeding a filtered version of the phase difference (i.e., the phase error) between the incoming signal $r(t)$ and the output of the VCO, $x(t)$, back to the input of the VCO, $y(t)$.

In the case of modern digital receivers, the error detector may be mathematically much more complicated than the simple multiplier shown in Figure 10.1. For example, the error detector might be a set of matched-filter correlators, each matched to a slightly different phase offset feeding a weighting or decision function. The output of the weighting function would be the phase error estimate. Such a function might be mathematically very complex, but it would be easily approximated using modern digital technology. The VCO may not appear to be a sinusoidal oscillator, but it may be implemented as a read-only memory whose pointers are controlled by a combination of a clock and the output of the error estimator. The feedback path may not be continuous (as shown in Figure 10.1), but phase corrections may only be applied once per frame, or once per packet, depending on the signal structure. A special header or known sequence of symbols may be inserted into the information stream for the expressed purpose of aiding the estimation process. These obvious differences notwithstanding, the basic principles are still illuminated with the simple model of Figure 10.1.

Consider a normalized input signal of the form

$$r(t) = \cos [\omega_0 t + \theta(t)] \quad (10.1)$$

where ω_0 is the nominal carrier frequency and $\theta(t)$ is a slowly varying phase. Similarly, consider a normalized VCO output of the form

$$x(t) = -2 \sin [\omega_0 t + \hat{\theta}(t)] \quad (10.2)$$

These signals will produce an output error signal at the phase detector output of the form

$$\begin{aligned} e(t) &= x(t)r(t) = 2 \sin [\omega_0 t + \hat{\theta}(t)] \cos [\omega_0 t + \theta(t)] \\ &= \sin [\theta(t) - \hat{\theta}(t)] + \sin [2\omega_0 t + \theta(t) + \hat{\theta}(t)] \end{aligned} \quad (10.3)$$

Assuming that the loop filter is low pass, the second term on the right hand side of Equation (10.3) will be filtered out and can be ignored. This low-pass assumption is a reasonable loop design decision. A low-pass filter provides an error signal that is solely a function of the difference in phases between the input [Equation (10.1)] and the VCO output [Equation (10.2)]. This is exactly the error signal that is needed. The VCO output frequency is the time derivative of the argument of the sine function in Equation (10.2). If we make the assumption that ω_0 is the uncontrolled frequency of the VCO (the output frequency when the input voltage is zero), we can express the difference in the VCO output frequency from ω_0 as the time differential of the phase term $\hat{\theta}(t)$. The output frequency of the VCO is a linear function of the input voltage. Therefore, since an input voltage of zero produces an output frequency of ω_0 , the difference in the output frequency from ω_0 will be proportional to the value of the input voltage $y(t)$, or

$$\begin{aligned}\Delta\omega(t) &= \frac{d}{dt} [\hat{\theta}(t)] = K_0 y(t) \\ &= K_0 e(t) * f(t) \\ &\approx K_0 [\theta(t) - \hat{\theta}(t)] * f(t)\end{aligned}\tag{10.4}$$

where $\Delta\omega(t)$ denotes the frequency difference, the notation $*$ indicates the convolution operation (see Appendix A), and the small-angle approximation [i.e., $e(t) = \sin [\theta(t) - \hat{\theta}(t)] \approx \theta(t) - \hat{\theta}(t)$] has been used in the last line of Equation (10.4). The small-angle approximation will be accurate when the output phase error is small (the loop is close to phase lock). This will be the situation when the loop is operating normally. The factor K_0 is the gain of the VCO, and $f(t)$ is the loop-filter impulse response. This linear differential equation in $\hat{\theta}(t)$ (utilizing the small-angle approximation) is known as the linearized loop equation. It is the single most useful relationship in determining loop behavior during normal operation (where the phase error is small).

Example 10.1 Linearized Loop Equation

Show that for appropriately chosen K_0 and $f(t)$ the linearized loop equation [Equation (10.4)] demonstrates a tendency toward phase lock—that is, the phase difference between the incoming signal and the VCO output tends to decrease.

Solution

Consider the case where the phase of the input signal, $\theta(t)$, is slowly varying with time. It can be seen that if the phase difference on the right-hand side of Equation (10.4) is positive [i.e., $\theta(t) > \hat{\theta}(t)$], then by appropriate choice of K_0 and $f(t)$, the time derivative of $\hat{\theta}(t)$ will be positive, so that $\hat{\theta}(t)$ will increase with time, which will tend to reduce the magnitude of the difference $|\theta(t) - \hat{\theta}(t)|$. On the other hand, if the phase difference is negative, $\hat{\theta}(t)$ will decrease with time, which will also reduce the magnitude of the phase difference. Finally, if $\theta(t) = \hat{\theta}(t)$, then Equation (10.4) indicates that $\hat{\theta}(t)$ will not change with time, and the equality will be maintained.

Consider the Fourier transform of Equation (10.4),

$$j\omega\hat{\Theta}(\omega) = K_0[\Theta(\omega) - \hat{\Theta}(\omega)]F(\omega)\tag{10.5}$$

of the

ion of its
ltage will
value, ω_0 ,
feeding a
incoming
 $y(t)$.
athemati-
10.1. For
ors, each
ion func-
ate. Such
approxi-
a sinus-
pointers
stimulator.
hase cor-
ng on the
inserted
stimulation
are still

(10.1)

use. Simi-

(10.2)

output of

(10.3)

where the capitalized functions of ω are the Fourier transforms of the lowercase functions of t in Equation (10.4). That is, $\hat{\Theta}(\omega) \leftrightarrow \hat{\theta}(t)$, $\Theta(\omega) \leftrightarrow \theta(t)$, and $F(\omega) \leftrightarrow f(t)$. Reorganizing Equation (10.5) provides

$$\frac{\hat{\Theta}(\omega)}{\Theta(\omega)} = \frac{K_0 F(\omega)}{j\omega + K_0 F(\omega)} = H(\omega) \quad (10.6)$$

The term $H(\omega)$ is known as the closed-loop transfer function of the PLL. This term is very useful in characterizing the transient response of a PLL. The order of a PLL is defined to be the order of the highest-order term in $j\omega$ in the denominator of $H(\omega)$. Equation (10.6) indicates that this is always one more than the order of the loop filter $F(\omega)$. This is because when $F(\omega)$ is expressed analytically as $F(\omega) = N(\omega)/D(\omega)$, the denominator of $H(\omega)$ when expressed as a polynomial in $j\omega$ will have the term $j\omega D(\omega)$, which must have a term in $j\omega$ that is one order higher than the highest-order term in $D(\omega)$ alone. The order of a PLL is critical for determining the loop's steady-state response to a steady-state input. This is discussed in the next section.

10.2.1.1 Steady-State Tracking Characteristics

By reorganizing Equation (10.6), we can obtain an expression for the Fourier transform of the phase error:

$$\begin{aligned} E(\omega) &= \mathcal{F}\{e(t)\} \\ &= \Theta(\omega) - \hat{\Theta}(\omega) \\ &= [1 - H(\omega)]\Theta(\omega) \\ &= \frac{j\omega\Theta(\omega)}{j\omega + K_0 F(\omega)} \end{aligned} \quad (10.7)$$

Equation (10.7) can be used in conjunction with the final value theorem of Fourier transforms to determine the steady-state error response of a loop to a variety of possible input characteristics. The steady-state error is the residual error after all transients have died away, and thus provides a measure of a loop's ability to cope with various types of changes in the input. The final value theorem states that

$$\lim_{t \rightarrow \infty} e(t) = \lim_{j\omega \rightarrow 0} j\omega E(\omega) \quad (10.8)$$

Combining Equations (10.7) and (10.8) yields

$$\lim_{t \rightarrow \infty} e(t) = \lim_{j\omega \rightarrow 0} \frac{(j\omega)^2 \Theta(\omega)}{j\omega + K_0 F(\omega)} \quad (10.9)$$

Example 10.2 Response to a Phase Step

Consider a loop's steady-state response to a phase step at the loop input.

Solution

Assuming that the PLL was originally in phase lock, a phase step will throw the loop out of lock. Having abruptly changed, however, the input phase again becomes stable.

This should be the easiest type of phase disturbance for a PLL to deal with. The Fourier transform of a phase step will be taken to be

$$\begin{aligned}\Theta(\omega) &= \mathcal{F}\{\Delta\phi u(t)\} \\ &= \frac{\Delta\phi}{j\omega}\end{aligned}\quad (10.10)$$

where $\Delta\phi$ is the magnitude of the step and $u(t)$ is the unit step function

$$\begin{aligned}u(t) &= \begin{cases} 1 & \text{for } t > 0 \\ 0 & \text{for } t < 0 \end{cases} \\ &= \int_{-\infty}^t \delta(\tau) d\tau\end{aligned}$$

in which $\delta(\tau)$ is the Dirac delta function. From Equation (10.9) and (10.10),

$$\lim_{t \rightarrow \infty} e(t) = \lim_{j\omega \rightarrow 0} \frac{j\omega \Delta\phi}{j\omega + K_0 F(\omega)} = 0$$

assuming that $F(0) \neq 0$. Thus the loop will eventually track out any phase step that appears at the input if the loop filter has a nonzero dc response. This means that for any loop filter with the property that $F(\omega) = N(\omega)/D(\omega)$ and $N(0) \neq 0$, the PLL will automatically tend to recover phase lock if the input is displaced by a constant phase. This is clearly a very desirable loop characteristic.

Example 10.3 Response to a Frequency Step

Next, consider a loop's steady-state response to a frequency step at the input.

Solution

A frequency step can approximate the effect of a Doppler shift in the incoming signal frequency due to relative motion between the transmitter and the receiver. Thus, this is an important example for systems with mobile terminals. Since phase is the integral of frequency, the input phase will change linearly as a function of time for a constant input-frequency offset. The Fourier transform of the phase characteristic will be the transform of the integral of the frequency characteristic. Since the frequency characteristic is a step, and the transform of an integral is the transform of the integrand divided by the parameter $j\omega$, it follows that

$$\Theta(\omega) = \frac{\Delta\omega}{(j\omega)^2} \quad (10.11)$$

where $\Delta\omega$ is the magnitude of the frequency step. Substituting Equation (10.11) into Equation (10.9) yields

$$\lim_{t \rightarrow \infty} e(t) = \lim_{j\omega \rightarrow 0} \frac{\Delta\omega}{j\omega + K_0 F(\omega)} = \frac{\Delta\omega}{K_0 F(0)} \quad (10.12)$$

The steady-state result in this case depends on more properties of the loop filter than merely a nonzero dc response. If the filter is "all-pass," then

$$F_{\text{ap}}(\omega) = 1 \quad (10.13)$$

If it is low-pass, then

$$F_{ep}(\omega) = \frac{\omega_1}{j\omega + \omega_1} \quad (10.14)$$

or if it is a lead-lag, then

$$F_{el}(\omega) = \left(\frac{\omega_1}{\omega_2}\right) \frac{j\omega + \omega_2}{j\omega + \omega_1} \quad (10.15)$$

Equation (10.12) indicates that the loop will track the input phase ramp with a constant steady-state error whose value will depend on the gain term K_0 and the magnitude of the frequency step. Using any of $F_{ap}(\omega)$, $F_{ep}(\omega)$, or $F_{el}(\omega)$ for $F(\omega)$ in Equation (10.12) yields

$$\lim_{t \rightarrow \infty} e(t) = \frac{\Delta\omega}{K_0}$$

Notice that a product of several filters with filter characteristics of the form of Equation (10.13), (10.14), or (10.15) would still produce this result. This steady-state error which is called the *velocity error*, will exist regardless of the order of the filter, unless the denominator of $F(\omega)$, contains $j\omega$ as a factor [$\omega_1 = 0$ in the denominator of Equation (10.14) or (10.15) with the appropriate renormalization in the numerators]. Having $j\omega$ as a factor of $D(\omega)$ is equivalent to having a perfect integrator in the loop filter. It is not possible to build a perfect integrator, but one may be closely approximated either digitally or by using active integrated circuits [5]. Thus if the system design requires the tracking of Doppler shifts with zero steady-state error the loop filter design must contain an approximation to a perfect integrator. It should be noted that even with a nonzero velocity error, the frequency is still being tracked: there are important applications where tracking to zero phase error is not important. Noncoherent signaling, such as the standard use of FSK modulation, is an example. For noncoherent signaling it is actually frequency tracking that is required, and the absolute value of phase is unimportant.

Example 10.4 Response to a Frequency Ramp

Consider a loop's steady state response when the input frequency is changing linearly with time (a frequency ramp function).

Solution

This example corresponds to the effect of a step change in the time derivative of the input frequency. This would approximate a change in the Doppler rate, which could model acceleration in the motion between a satellite or an aircraft and a ground receiver. In this case, the Fourier transform of the phase characteristic is given by

$$\Theta(\omega) = \frac{\Delta\dot{\omega}}{(j\omega)^3} \quad (10.16)$$

where $\dot{\omega}$ is the magnitude of the rate of frequency change. In this case, Equation (10.9) yields

$$\lim_{t \rightarrow \infty} e(t) = \lim_{j\omega \rightarrow 0} \frac{\Delta\dot{\omega}/j\omega}{j\omega + K_0 F(\omega)} = \lim_{j\omega \rightarrow 0} \frac{\Delta\dot{\omega}}{j\omega K_0 F(\omega)} \quad (10.17)$$

If the loop has a nonzero velocity error—that is, if the right-hand side of Equation (10.12) is not equal to zero—Equation (10.17) shows the steady-state phase error to be

(10.14)

(10.15)

amp with a con-
and the magni-
 $\gamma(\omega)$ in Equation

e form of Equa-
eady-state error
the filter, unless
inator of Equa-
merators]. Hav-
n the loop filter.
pproximated ei-
stem design re-
oop filter design
noted that even
re are important
coherent signal-
noncoherent sig-
e value of phase

hanging linearly

derivative of the
ate, which could
it and a ground
is given by

(10.16)

Equation (10.9)

(10.17)

ide of Equation
hase error to be

on Chap. 10

unbounded due to a frequency ramp. This says that a PLL with loop filters given by any of Equations (10.13) to (10.15) will not be able to track a frequency ramp. In order to track a frequency ramp, the denominator of the loop filter transform $D(\omega)$ must have $j\omega$ as a factor. From Equation (10.17) it can be seen that a loop filter with a transfer function of the type $F(\omega) = N(\omega)/j\omega D_1(\omega)$ will allow the PLL to track a frequency ramp with a constant phase error. This implies that in order to track a signal with a linearly changing Doppler shift (constant relative acceleration), the receiver must have a PLL that is second order or higher. To track a frequency ramp with zero phase error, the loop filter would be required to have a transfer function with $(j\omega)^2$ as a factor of the denominator, $F(\omega) = N(\omega)/(j\omega)^2 D_2(\omega)$. This implies a PLL that is third order or higher. Thus high-performance aircraft that need to track phase accurately through violent maneuvers may require third- or higher-order PLLs. In all cases, frequency lock is available with a loop of one order less than that required for phase lock. Steady-state error analysis is therefore a useful indicator of the required complexity of the loop filters.

In practice, the vast majority of PLL designs are second order. This is because a second-order loop can be made to be unconditionally stable [5]. Unconditionally, stable loops will always try to track the input. No set of input conditions—regardless of how extreme—will cause the loop to respond in the inappropriate direction to changes in the input. Second-order loops will track out the effect of a frequency step (Doppler shift), and they are relatively easy to analyze, since the closed-form results obtained for first-order loops are good approximations for second-order loop performance. Third-order loops are used for some special applications [e.g., some Global Positioning System (GPS) navigation receivers and some airborne receivers have third-order PLLs], but loop performance for third-order loops is relatively difficult to determine, and third- and higher-order loops are only conditionally stable. Often if the signal dynamics are expected to be such that high-order loops would be required for coherent demodulation, noncoherent demodulation is used instead.

10.2.1.2 Performance in Noise

The steady-state analysis of the preceding section tacitly assumed that the input signal was noise free. In some situations this may be approximately correct, but as in other parts of communication analysis, the more general case would include the effects of noise.

Reconsider the normalized loop input signal of Equation (10.1) and Figure 10.1. With the inclusion of normalized narrowband additive Gaussian noise $n(t)$, the expression for the input becomes

$$r(t) = \cos(\omega_0 t + \theta) + n(t) \quad (10.18)$$

where, for the moment, we consider the input phase offset, θ to be a constant. The noise process $n(t)$, assumed to be a zero-mean narrowband Gaussian process, can be expanded into quadrature components about the carrier frequency as [6]

$$n(t) = n_c(t) \cos \omega_0 t + n_s(t) \sin \omega_0 t \quad (10.19)$$

where both $n_c(t)$ and $n_s(t)$ are zero-mean Gaussian random processes and are statistically independent. Now the output of the phase detector can be written as [see Equation (10.3)]

$$e(t) = x(t)r(t) \quad (10.20)$$

$$= \sin(\theta - \hat{\theta}) + n_c(t) \cos \hat{\theta} + n_s(t) \sin \hat{\theta} + (\text{terms at twice the carrier frequency})$$

As before, the loop filter eliminates the twice-carrier-frequency terms. Denoting the second and third terms of Equation (10.20) as

$$n'(t) = n_c(t) \cos \hat{\theta} + n_s(t) \sin \hat{\theta} \quad (10.21)$$

we see that it is easy to verify that the variance of $n'(t)$ is identical to the variance of $n(t)$. This variance will be denoted by σ_n^2 .

Consider the autocorrelation function of $n'(t)$,

$$\begin{aligned} R(t_1, t_2) &= \mathbf{E}\{n'(t_1)n'(t_2)\} \\ &= \mathbf{E}\{n_c(t_1)n_c(t_2)\} \cos^2 \hat{\theta} + \mathbf{E}\{n_s(t_1)n_s(t_2)\} \sin^2 \hat{\theta} \\ &\quad + [\mathbf{E}\{n_c(t_1)n_s(t_2)\} + \mathbf{E}\{n_s(t_1)n_c(t_2)\}] \sin \hat{\theta} \cos \hat{\theta} \end{aligned} \quad (10.22)$$

where $\mathbf{E}\{\cdot\}$ denotes the expected value. The cross-terms on the right-hand side of Equation (10.22) are equal to zero because n_c and n_s are mutually independent and have zero means [6]. With the assumption of wide-sense stationarity [7], we have

$$R(\tau) = R_c(\tau) \cos^2 \hat{\theta} + R_s(\tau) \sin^2 \hat{\theta} \quad (10.23)$$

where $\tau = t_1 - t_2$. Taking Fourier transforms, the power spectral density of $n'(t)$ is seen to be

$$\begin{aligned} G(\omega) &= \mathcal{F}[R(t)] \\ &= G_c(\omega) \cos^2 \hat{\theta} + G_s(\omega) \sin^2 \hat{\theta} \end{aligned} \quad (10.24)$$

where G_c and G_s are the Fourier transforms of R_c and R_s , respectively. But from Equation (10.19), it can be seen that the spectra G_c and G_s are made of shifted versions of the spectra of the original noise process $n(t)$. Therefore, because of our construction [8],

$$G_s(\omega) = G_c(\omega) = G_n(\omega_0 - \omega) + G_n(\omega_0 + \omega)$$

where $G_n(\omega)$ is the spectral density of the original bandpass noise process $n(t)$. Equation (10.24) can be rewritten as

$$G(\omega) = G_n(\omega_0 - \omega) + G_n(\omega_0 + \omega) \quad (10.25)$$

For the special case of white noise, we have $G_n(\omega) = N_0/2$ Watts/Hertz, where N_0 is the single-sided spectral density of the white noise. Thus, from Equation (10.25), for this important special case,

$$G(\omega) = N_0 \quad (10.26)$$

The value in this development is that for the same small-angle approximations that were made in the preceding section, the spectral density of the VCO phase, G_{θ} , is related to the spectral density of the noise process through the loop transfer function [Equation (10.6)]. That is,

$$G_{\theta}(\omega) = G(\omega) |H(\omega)|^2 \quad (10.27)$$

where $G(\omega)$ is as given in Equation (10.25) and $H(\omega)$ as defined in Equation (10.6). The variance of the output phase is then

$$\sigma_{\theta}^2 = \frac{1}{2\pi} \int_{-\infty}^{\infty} G(\omega) |H(\omega)|^2 d\omega \quad (10.28)$$

For the special case of white noise,

$$\sigma_{\theta}^2 = \frac{N_0}{2\pi} \int_{-\infty}^{\infty} |H(\omega)|^2 d\omega \quad (10.29)$$

The integral in Equation (10.29) (renormalized to natural frequency) is called the *two-sided loop bandwidth* W_L . The *single-sided loop bandwidth* is termed B_L . The definitions of these terms are

$$W_L = 2B_L = \frac{1}{2\pi} \int_{-\infty}^{\infty} |H(\omega)|^2 d\omega \text{ Hertz} \quad (10.30)$$

Thus, if the noise process is white and the small-angle approximation holds (in other words, the loop is successfully tracking the input phase), the phase variance is given by

$$\sigma_{\theta}^2 = 2N_0 B_L \quad (10.31)$$

The phase variance is a measure of the amount of jitter or wobble in the VCO output due to noise at the input. Equations (10.31) and (10.7) highlight one of the many trade-offs in communication theory. Clearly, one would wish σ_{θ}^2 to be small, which for a given noise level implies a small loop bandwidth, B_L , which from Equation (10.30) implies a narrow $H(\omega)$. However, it can be inferred from Equation (10.7) that the narrower the effective bandwidth of $H(\omega)$, the poorer will be the loop's ability to track incoming signal phase changes $\Theta(\omega)$. Thus a loop design must balance noise response with desired input phase response. The designer's dilemma is to design a loop that responds appropriately to the changes in the input signal, while not being overly responsive to the apparent changes, which are actually only artifacts of the noise process.

10.2.1.3 Nonlinear Loop Analysis

All of the PLL discussion in the previous sections has utilized what is called the linearized PLL model. This model is shown schematically in Figure 10.2. The model makes use of the small-angle approximation

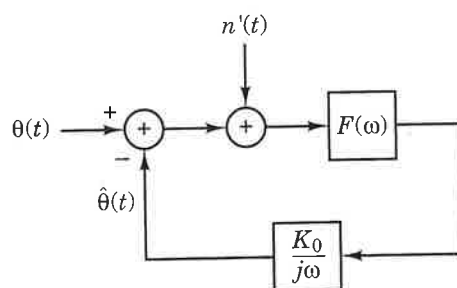


Figure 10.2 Schematic of linearized PLL model.

$$\sin(\theta - \hat{\theta}) \approx \theta - \hat{\theta} \quad (10.32)$$

which is accurate when the loop is "in lock" and performing as desired (i.e., with small phase errors). Clearly, these conditions form only part of the picture. A complete analysis of PLL performance must allow for the times when Equation (10.32) is not accurate. When the small-angle approximation is inaccurate, an appropriate model is the one shown schematically in Figure 10.3. From Equations (10.4), (10.20), and (10.21) and Figure 10.3, the model can be described by the differential equation

$$\frac{d}{dt} [\hat{\theta}(t)] = K_0 f(t) * \sin[\theta(t) - \hat{\theta}(t)] + K_0 f(t) * n'(t) \quad (10.33)$$

where, as before, * denotes the convolution operation. In spite of the best efforts of many researchers, this differential equation has resisted general solution for many years. However, Viterbi [8] derived a closed-form solution for an important special case.

Consider the case where $\theta(t)$, the input phase as a function of time, is a constant θ . We can now define a new phase variable

$$\phi(t) = [\theta - \hat{\theta}(t)] \text{ modulo } 2\pi \quad (10.34)$$

Because θ is constant, Equation (10.33) can be rewritten as

$$\frac{d}{dt} [\phi(t)] = K_0 f(t) * \sin \phi(t) + K_0 f(t) * n'(t) \quad (10.35)$$

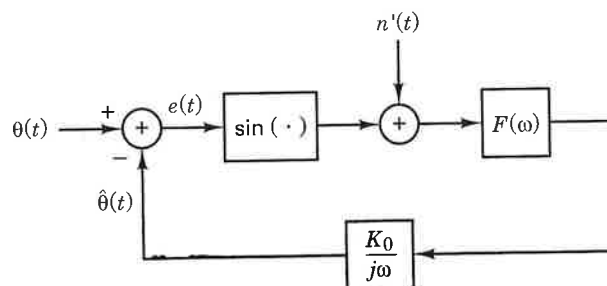


Figure 10.3 Schematic of nonlinearized PLL model.

10.2.1.4 Suppressed Carrier Loops

The discussion of PLLs to this point has presumed that the carrier input is a fairly stable sinusoid with some known positive average energy. In the case of a phase modulated communication system, if the carrier phase variation due to the modulation is less than $\pi/2$ radians, there will be positive energy at the carrier frequency. This is called a system design that has a residual carrier component, and all of the discussion of PLL development to this point would apply directly to this residual component. A diagram of the signal space for a binary phase modulated system with a residual carrier component is given in Figure 10.4, for a modulating angle of $\gamma \leq \pi/2$. At one time, most phase modulated systems were designed in this way. However, the residual carrier component is, in a sense, wasted energy—in the sense that the energy in the residual carrier is not being used to transmit the information, only to transmit the carrier. Thus most modern phase modulated systems are suppressed carrier systems. This means that there is no average energy transmitted at the carrier frequency. All of the transmitted energy goes into the modulation. Unfortunately, this means that there is no longer any signal for the basic PLL of Figure 10.1 to track.

Consider, as an example, a BPSK signal

$$r(t) = m(t) \sin(\omega_0 t + \theta) + n(t) \quad (10.40)$$

where $m(t) = \pm 1$ with equal probability. This is a suppressed carrier transmission—the average energy at radian frequency ω_0 is zero. This situation is represented graphically in Figure 10.4, when $\gamma = \pi/2$. The figure indicates that for this case the vertical carrier component will vanish. To acquire and track the phase of the carrier, the effects of the modulation must be eliminated. One way to eliminate the modulation is to square the signal:

$$\begin{aligned} r^2(t) &= m^2(t) \sin^2(\omega_0 t + \theta) + n^2(t) + 2n(t)m(t) \sin(\omega_0 t + \theta) \\ &= \frac{1}{2} - \frac{1}{2} \cos(2\omega_0 t + 2\theta) + n^2(t) + 2n(t)m(t) \sin(\omega_0 t + \theta) \end{aligned} \quad (10.41)$$

Here, we have made use of the fact that $m^2(t) = 1$. The second term on the right-hand side of Equation (10.41) is a carrier-related term (at twice the original carrier

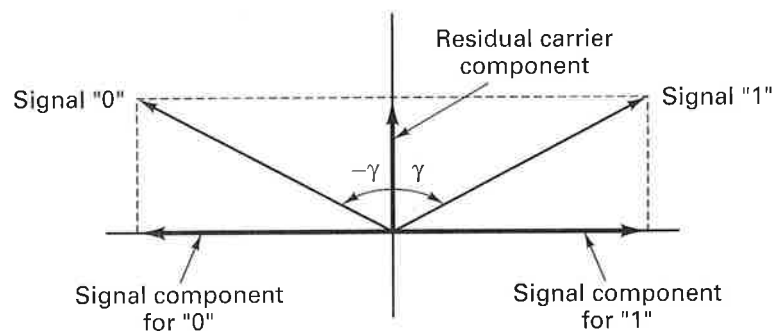


Figure 10.4 Residual carrier binary phase modulation.

Since, from Equation (10.35), $\phi(t)$ is a function of the random process $n'(t)$, $\phi(t)$ itself is a random process. Because $\phi(t)$ is defined modulo 2π , it can be shown [5] that $\phi(t)$ is stationary in the limit when all transient effects have died down (i.e., θ is a constant). Viterbi [8] determined that for a first-order PLL (i.e., the loop filter is a short circuit, or equivalently $f(t) = \delta(t)$), the probability density function of ϕ is of the form

$$p(\phi) = \frac{\exp(\rho \cos \phi)}{2\pi I_0(\rho)} \quad \text{for } |\phi| \leq \pi \quad (10.36)$$

where $\rho = 1/\sigma_\theta^2$ [see Equation (10.31)] is the normalized (to unit signal energy) loop signal-to-noise ratio, and $I_0(\rho)$ is the zeroth-order modified Bessel function of the first kind, evaluated at ρ . The phase variance, modulo 2π , can now be computed using Equation (10.36). The resulting value of the phase variance will be exact for first-order loops, and is an extremely useful approximation for the behavior of many second-order loops [5]. It has also been shown to be an exact form for higher-order loops under a modified definition of ρ [9].

The change of variable from a phase that can take any real value to a phase that is modulo 2π results in the concept of loop cycle slips. A cycle slip occurs when the magnitude of the original phase error, $|\theta - \hat{\theta}(t)|$, exceeds 2π radians. This will cause the value of ϕ [Equation (10.34)] to abruptly change from about 2π to about 0. This event can be thought of as a momentary loss of lock with an almost immediate reacquisition. The statistics of cycle slips can be as important an indicator of PLL performance as phase variance—especially at low-loop signal-to-noise ratios, when cycle slips may occur frequently.

By manipulating his phase-distribution results, Viterbi [8] derived an expression for the mean time to the first cycle slip, T_m , beginning at some arbitrary reference time:

$$T_m = \frac{\pi^2 \rho I_0^2(\rho)}{2B_L} \quad (10.37)$$

For large ρ , this expression can be approximated by

$$T_m \approx \frac{\pi \exp(2\rho)}{4B_L} \quad (10.38)$$

As was true with the probability density function of Equation (10.36), these results were derived for first-order loops, but they are useful approximations for the behavior of second-order loops, and they provide an upper bound to second-order loop performance at medium and large loop signal-to-noise ratios. In addition, computer simulations and laboratory measurements [5] indicate that the time T between cycle slips is exponentially distributed:

$$P(T) = 1 - \exp\left(-\frac{T}{T_m}\right) \quad (10.39)$$

This is to say that the probability that a loop will cycle-slip within time T , starting from zero phase error, is given by Equation (10.39).

frequency) that can be acquired and tracked with a basic PLL of the type illustrated in Figure 10.1. Such an arrangement is illustrated in Figure 10.5. When the incoming suppressed-carrier waveform is squared, the resulting twice-carrier component can be acquired and tracked by a PLL of standard design. Some of the problems with this procedure can be inferred from Equation (10.41). The first problem is simply that all phase angles have been doubled. Thus, the phase noise and phase jitter has been doubled, and the phase error variance (related to the phase noise squared) is larger by a factor of 4 than that of the original signal. This angle doubling is offset by the divide-by-2 circuit at the VCO output, and, therefore, does not directly affect the accuracy of the loop's output signal that is used by the data demodulator. However, this larger internal variation will cause the PLL to require a 6-dB-larger carrier signal-to-noise ratio than a residual carrier system in order to maintain phase lock. In addition, now there are two effective noise terms interfering with loop operation, because of the cross-correlation term between noise and signal in Equation (10.41). For cases of medium or low loop signal-to-noise ratio, these two noise terms will reduce the available signal-to-noise ratio even further relative to the original unmodulated carrier. This additional loss due to signal-times-noise and noise-times-noise terms is called the *loop squaring loss* S_L . Gardner [5] shows that if the input noise process $n(t)$ is a narrowband Gaussian noise of bandwidth B_i , the squaring loss is upper bounded by

$$S_L \leq 1 + N_0 B_i \quad (10.42)$$

where, as before, N_0 is the single-sided power spectral density of the prefiltered, normalized white Gaussian noise process. Equation (10.42) is an upper bound because the filter bandwidth B_i is tacitly assumed to be wide enough to pass the signal undistorted. In an actual design, signal distortion can be traded for squaring loss, as is shown in [10].

Since the normalization in Equation (10.42) is with respect to the signal powers, the second term is proportional to a signal-to-noise ratio

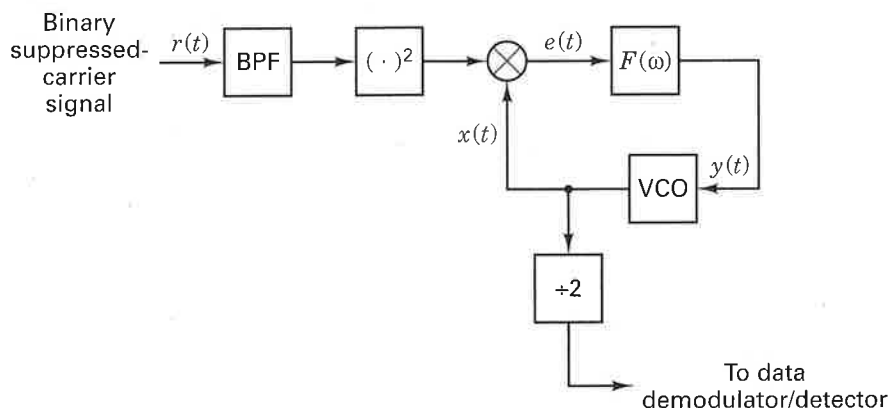


Figure 10.5 Basic squaring loop schematic.

$$\rho_i = \frac{1}{2N_0B_i} \quad (10.43)$$

where ρ_i is the signal-to-noise ratio in the input filter bandwidth. For large loop signal-to-noise ratios, the output phase variance can now be expressed as

$$\sigma_{\theta}^2 = 2N_0B_LS_L = 2N_0B_L \left(1 + \frac{1}{2\rho_i} \right) \quad (10.44)$$

The leading term on the right-hand side of Equation (10.44) can be seen to be identical to that of Equation (10.31), the phase variance of the standard PLL. It can also be seen that for large input signal-to-noise ratios, the second term in the squaring loss will vanish, and we are left with the phase variance of the standard PLL.

Another potentially serious problem, associated mainly with suppressed carrier loops, is that of *false lock* [5, 11–13]. This can be a problem especially during acquisition or reacquisition of carrier phase. The interaction of the data stream with the loop nonlinearities (especially the squaring circuit) and loop filters will produce sidebands in the spectrum that is input to the phase detector. These sidebands can contain stable frequency components. Care must be taken that these stable components are not allowed to capture the tracking loop. If the loop is captured, it will appear to be operating correctly; the VCO control signal $y(t)$ will be small but the VCO output will be offset in frequency from the correct carrier component. This is false lock. The loop is tracking a sideband frequency component, and the loop filter is filtering out the real carrier. False lock is a hardware-implementation problem that typically sets an effective lower limit on the bandwidth of the loop filters. Because they have fewer nonlinear elements, false locking is not usually a problem with residual carrier loops.

10.2.1.5 Costas Loops

An important form of a suppressed carrier loop is the Costas loop, shown schematically in Figure 10.6. This loop design is important because it eliminates the square-law device, which can be difficult to implement at carrier frequencies, and replaces it with a multiplier and relatively simple low-pass filters. Although the appearance of the circuits in Figures 10.5 and 10.6 is quite different, their theoretical performance can be shown to be the same [5]. The main remaining implementation problem with Costas loops is that to achieve the theoretically optimum performance, the two low-pass arm filters must be perfectly matched. This can only be approximated in any analog hardware implementation. If the arm filters are implemented digitally, there will be no problem keeping them matched, but the designer will confront the usual sampled data design issues. Thus the decision as to whether to implement a Costas loop or the classical design of Figure 10.5 amounts to a design decision between the difficulty of implementing the squaring device and the difficulty of implementing closely matched arm filters. This design decision will depend on the parameters and requirements of the particular receiving system, and cannot be generalized here.

(10.43)

large loop
s

(10.44)

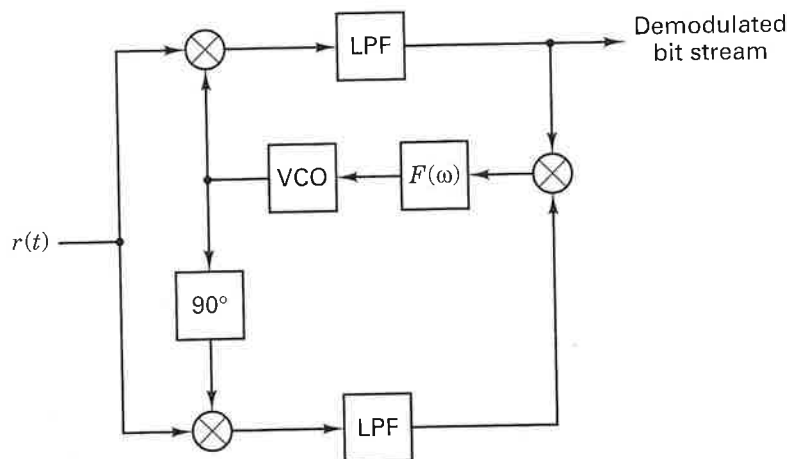
to be iden-
. It can also
he squaring
PLL.pressed car-
ially during
data stream
filters will
These side-
at these sta-
the loop is
nal $y(t)$ will
rect carrier
ncy compo-
a hardware-
mit on the
ments, falseloop, shown
minates the
uencies, and
ough the ap-
r theoretical
lementation
num perfor-
can only be
rs are imple-
the designer
s to whether
mounts to a
vice and the
decision will
; system, and

Figure 10.6 Costas loop.

10.2.1.6 High-Order Suppressed-Carrier Loops

Binary phase-shift keying is not the only type of suppressed-carrier modulation. In fact, assuming that all signals are equally likely *a priori*, any modulation scheme whose average amplitude, averaged over the signal set, is zero will have no average energy in the transmitted carrier. Perhaps the most common nonbinary suppressed-carrier modulation is quadrature phase-shift keying, or QPSK (4-ary PSK). If a QPSK signal is squared, the result “looks like” a BPSK signal. Thus, for equally likely QPSK signals, the carrier is still suppressed. However, squaring the signal a second time—equivalent to taking the original signal to the fourth power—can be seen to produce a term with a carrier component at four times the transmitted carrier’s frequency. As in the binary case, operating on the incoming signal with a power-law device produces cross products among the noise terms and signal terms, and introduces the equivalent of a “squaring loss.” Under the assumption that the noise bandwidth will pass the signal undistorted, the loss for fourth-power loops is upper bounded by [5]

$$S_L \leq 1 + \frac{9}{\rho_i} + \frac{6}{\rho_i^2} + \frac{3}{2\rho_i^3} \quad (10.45)$$

As was the case with the squaring loop, for sufficiently high input signal-to-noise ratios, ρ_i , Equation (10.45) indicates that the additional loss terms vanish, and the loop performance approaches that of the basic loop. As was also the case for the squaring loop, there are Costas loop designs equivalent to fourth-order loops [5, 14, 15] that may exhibit hardware implementation advantages. Their theoretical performance, however, is the same as that of the straightforward fourth-power design.

Example 10.5 Squaring Loss Bounds

Compare the upper bounds on squaring loss S_L given by Equations (10.42) and (10.45) for second- and fourth-power loops, respectively, for an input loop signal-to-noise ratio ρ_i of 10 dB.

Solution

A 10dB signal-to-noise ratio is also 10 in terms of its power ratio. Therefore, from Equations (10.42) to (10.44), for the squaring loop,

$$S_L = 1 + \frac{1}{2\rho_i} = 1.05 = 0.2 \text{ dB}$$

From Equation (10.45), for the fourth-power loop,

$$S_L = 1 + 0.9 + 0.06 + 0.0015 = 1.9615 = 2.9 \text{ dB}$$

Thus, while an input signal-to-noise ratio of 10 dB is adequate to keep losses small for the squaring loop, the same signal-to-noise ratio may allow significant losses for the fourth-power loop.

10.2.1.7 Acquisition

In most of the discussion thus far, the assumption has been that the PLL is in lock. This was the justification for assuming that the phase error $|\theta - \hat{\theta}|$ was small. At one time or another, however, every loop must acquire lock—that is, it must be brought **into** lock. Acquisition can be accomplished with the aid of external circuits or signals (aided acquisition) or in **some** cases by an unaided PLL (self-acquisition) [5].

Acquisition is an inherently nonlinear operation and therefore is difficult to analyze in general. However, some intuition may be obtained by considering a noise-free, first-order loop. Such a loop is shown schematically in Figure 10.3, where $n'(t) = 0$ (noise-free) and $F(\omega) = 1$ (first-order). Denote the input phase as

$$\theta(t) = \omega_i t$$

and the output phase as

$$\hat{\theta}(t) = \omega_0 t + \int_0^t K_0 \sin e(t) dt + \hat{\theta}(0) \quad (10.46)$$

where ω_i and ω_0 are the radian frequencies of the input and output signals, respectively. Thus the phase error is given by

$$e(t) = \theta(t) - \hat{\theta}(t) \quad (10.47)$$

$$= (\omega_i - \omega_0)t - \int_0^t K_0 \sin e(t) dt - \hat{\theta}(0)$$

Differentiating both sides and letting $\Delta\omega = \omega_i - \omega_0$ provides

$$\frac{de}{dt} = \Delta\omega - K_0 \sin e \quad (10.48)$$

where the time dependence of the function $e(t)$ has been suppressed to ease notation. This differential equation describes the behavior of the first-order noise-free PLL. The loop being in lock requires that

$$\frac{de}{dt} = 0 \quad (10.49)$$

Equation (10.49) is a necessary, but not a sufficient, condition for phase lock. This can be verified by observing the phase plane diagram of Figure 10.7. This figure is obtained by dividing both sides of Equation (10.48) by the gain term K_0 , and plotting the results. First observe point a . If the phase error is displaced a little to the left or right of point a , the sign of the derivative term is such that the phase error e will be driven back toward a . Thus, point a is a stable point of the system, a point where phase lock can be obtained and will be maintained. Now consider the case of point b . If the phase error is exactly at b , Equation (10.49) will be satisfied. However, if there is any slight offset from b , the sign of the derivative term will be such that the error will be driven away from b . Thus b is a point of marginal stability for the loop, a point where Equation (10.49) is satisfied, but not a stable lock point.

The amount of time required for a loop to come into lock can be a very important system design consideration. By observing Equation (10.48), we can see that the requirement of Equation (10.49) for phase lock cannot be met unless

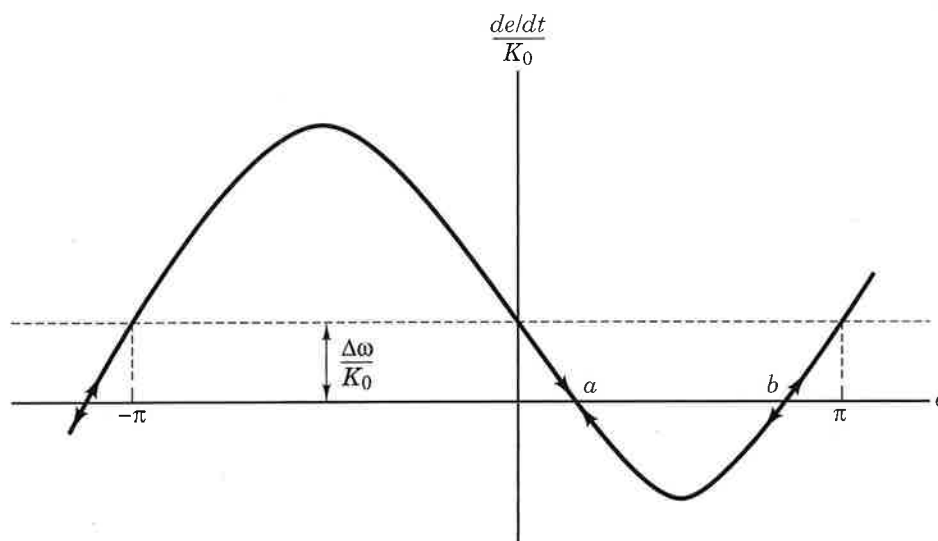


Figure 10.7 Phase-plane plot of first-order loop.

$$\frac{|\Delta\omega|}{|K_0|} \leq 1 \quad (10.50)$$

This is because sinusoidal functions have a maximum amplitude of unity. This range of frequency difference, $-K_0 < \Delta\omega < K_0$, is sometimes called the lock-in range of the loop. Assuming that Equation (10.50) holds, Gardner [5] gives a rule of thumb of $3/K_0$ seconds for the time required for loop acquisition. Actual values can be obtained from Equation (10.47) for well-defined sets of initial conditions, or by extensive computer simulation. It can be seen from the phase plot of Figure 10.7 that the time required will vary widely as a function of the initial phase error. For phase errors very close to point b , the driving force $(de/dt)/K_0$ will be very small. Thus, for this worst-case phase error, the error could "linger" in the vicinity of b for a long time. This phenomenon is called terminal loop hang-up [16] and can be a serious problem for system designs that depend on self-acquisition.

Perhaps the most important operational difference between first-order and higher-order loops is the higher-order loop's ability to "pull in" from frequency differences that are larger than the lock-in range. A first-order loop with a frequency error larger than the lock-in range will drift toward lock but never quite lock in. Why? Second- and higher-order loops can pull in and achieve phase lock because of their more complicated phase-plane characteristics. (Interested readers should consult Viterbi [8] and other texts on PLLs for more details [5, 9, 17-19].)

The study of self-acquisition for phase-locked loops is mostly of academic interest. Gardner [5] states that loops using self-acquisition can be guaranteed to acquire in reasonable time only under very benign circumstances. This, unfortunately, is rarely the case in practice.

Acquisition aiding drives the loop through the region of phase space expected to contain the lock-in region by means of some external driving signal. This is the most common means of achieving acquisition. Aiding can be implemented by simply applying a voltage ramp to the input of the VCO. This driving signal will cause the VCO output frequency to vary linearly with time. As was shown earlier [Equation (10.17)], loops with loop filters that do not contain $j\omega$ as a factor of their transfer function's denominator cannot track a frequency ramp with finite phase error. Therefore, if frequency sweeping is to be employed with a first-order loop or a second-order loop without this transfer function characteristic, the rate of frequency sweep must be slow enough so that when the loop achieves lock, the presence of phase lock can be detected and the sweeping signal removed before it drives the loop back out of lock. With loops that contain $j\omega$ as a factor of $D(\omega)$, it may not be necessary to remove the sweeping signal at all, because, at least in theory, the loop will be able to track out the frequency ramp. In any case, the sweep rate must not be too large, or the loop will be driven through the lock point so fast that it will fail to acquire. For a second-order loop with loop transfer function [see Equation (10.6)],

$$H(\omega) = \frac{1}{-(j\omega/\omega_n)^2 + 2\zeta(j\omega/\omega_n) + 1} \quad (10.51)$$

(10.50)

Gardner [5] indicates that the maximum sweep rate, $\Delta\dot{\omega}$, must be in the vicinity of

$$\Delta\dot{\omega} \approx \frac{1}{2} \omega_n^2 (1 - 2\sigma_{\hat{\theta}}) \quad (10.52)$$

where $\sigma_{\hat{\theta}}$ is as defined in Equation (10.31) and ω_n , implicitly defined in Equation (10.51), is called the *natural frequency* of a second-order PLL and is related to the loop bandwidth B_L and loop damping factor ζ [9] by

$$\omega_n = \frac{8\zeta}{4\zeta^2 + 1} B_L$$

Blanchard [17] gives more detailed results for aided phase acquisition.

10.2.1.8 Phase Tracking Errors and Link Performance

If a loop is unable to track out all phase errors, the received symbol-error probability will be degraded relative to what is theoretically achievable. The analysis required to determine the amount of the degradation is very involved, but for most of the standard coherent signaling systems, curves are available [14, 15, 20]. Figure 10.8 is an example of such a performance curve for a residual carrier-phase tracking loop operating on a signal with BPSK modulation in additive Gaussian noise. It can be seen that for signal-to-noise ratios of moderate value, small phase errors produce very little degradation. It is only when the standard deviation of phase error exceeds 0.3 that the degradations become significant. This means that the inherent degradation in performance caused by a well-designed loop operating in benign conditions can generally be ignored. The curve also indicates that if conditions are such that the phase variance is large, increasing the data signal-to-Gaussian noise ratio may not be effective in reducing the detected error probability. It should be noted that the presence of an irreducible error in these situations is a characteristic of residual carrier designs with constant loop signal-to-noise ratios ρ_i . Suppressed carrier tracking loops tend not to have irreducible errors, because an increase in the data signal-to-noise ratio will increase the signal-to-noise ratio of the suppressed carrier tracking loop, reducing the tracking error.

Example 10.6 PLL Signal-to-Noise Ratio

Develop an integral expression for the effect on link bit error probability of slowly varying phase tracking errors for a residual carrier BPSK link. Compare the effect of a normalized loop signal-to-noise ratio ($\rho = 1/\sigma_{\hat{\theta}}^2$) of 20 dB with one of 10 dB on error performance at a desired bit error probability of 10^{-5} using Figure 10.8.

Solution

From Chapter 4, the theoretically possible bit error probability for a BPSK link in additive white Gaussian noise of single-sided spectral density N_0 Watts/Hertz is given by

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

(10.51)

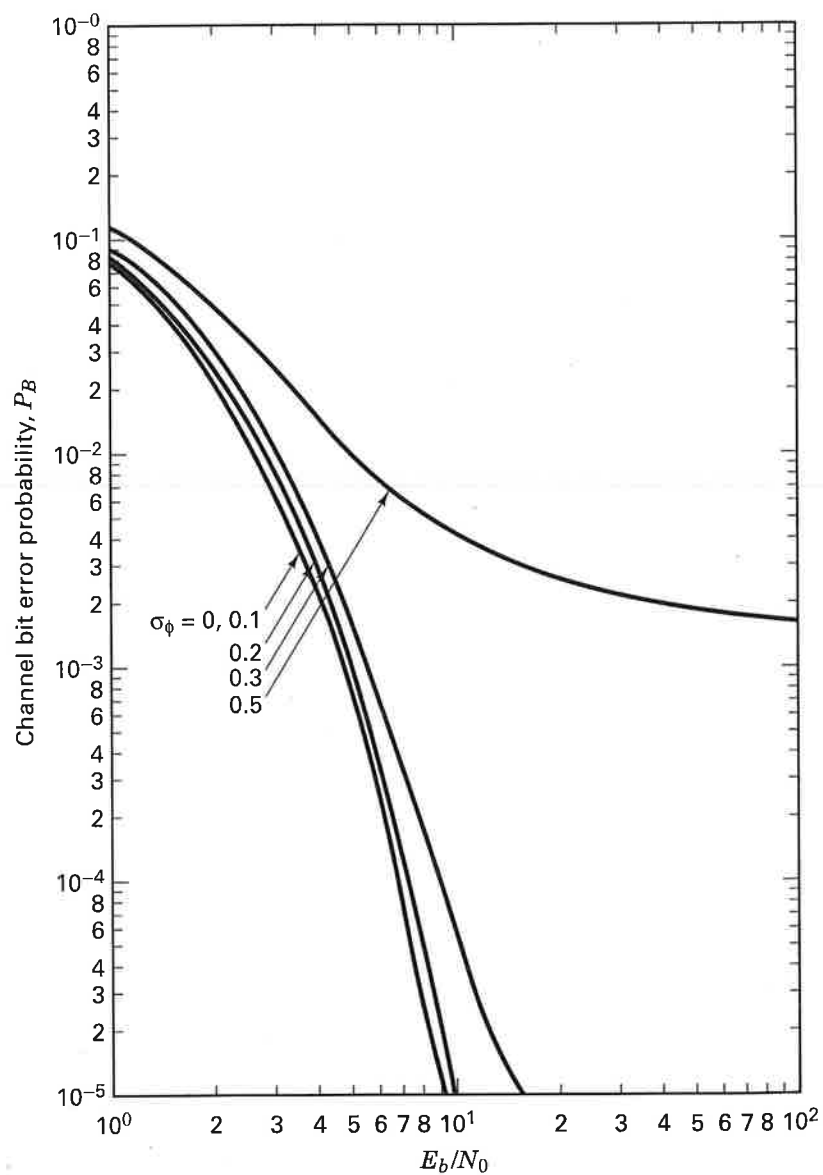


Figure 10.8 Channel bit error probability versus E_b/N_0 for BPSK with imperfect carrier synchronization. (Reprinted with permission from J. J. Stiffler, *Theory of Synchronous Communications*, Prentice-Hall, Inc., Englewood Cliffs, N.J., Fig. 9.1, p. 270.)

where E_b is the received energy per bit time. From the derivation of this expression for error probability, it can be shown that if there is a slowly varying (with respect to the data rate) phase-tracking error of β radians, the resulting probability of error will be given by

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

Now if the phase error β is the result of tracking errors caused by system noise, β will be described stochastically by some probability density function $p(\beta)$. Then the expected bit-error probability is given by

$$P_B = \int_0^{2\pi} P_B(\beta) p(\beta) d\beta$$

For the special case of a first-order loop, the probability density function is given by Equation (10.36). Then the final expression for the expected bit-error probability is given by

$$P_B = \int_0^{2\pi} Q\left(\sqrt{\frac{2E_b \cos \beta}{N_0}}\right) \frac{\exp(\rho \cos \beta)}{2\pi I_0(\rho)} d\beta$$

A loop signal-to-noise ratio (ρ_i) of 20 dB will correspond to a standard deviation of phase noise of $\sigma_\theta = 0.1$ rad. From Figure 10.8, this small amount of phase noise produces no appreciable degradation in the bit-error probability. A loop ρ_i of 10 dB, however, corresponds to a phase noise standard deviation of $\sigma_\theta = 0.32$ rad. It can be seen from Figure 10.8 that for a bit-error probability of 10^{-5} , this phase noise standard deviation will require a data SNR of somewhat more than 11 (10.4 dB), rather than a data SNR of 9.1 (9.6 dB) for perfect phase tracking. Thus, this loop signal-to-noise will cause an error-performance degradation of somewhat more than 0.8 dB, at an error probability of 10^{-5} . It should be noted that for loop SNRs less than about 10 dB, the degradation in performance increases very rapidly. Thus, 10 dB is something of a threshold for reasonable system performance for residual carrier designs. Suppressed carrier designs, having no problem with irreducible error, may do better.

10.2.1.9 Spectrum Analysis Techniques

The techniques we have considered thus far belong to a class of synchronizers sometimes called *spectral line techniques*. These techniques all either use an existing spectral line at the carrier frequency or produce such a line at the carrier frequency or a multiple thereof, as a crucial part of the error determination. There is another set of techniques that are especially useful in carrier-frequency estimation or tracking that utilize the shape of the signal's passband spectrum. These techniques have solid roots in maximum-likelihood estimation theory [4], but they also are intuitively appealing and will be approached here from the intuitive direction.

Possibly the most intuitive technique in this class is a simple bank of matched filters, with each filter matched to the expected signal with a different carrier frequency offset. Such a bank of filters could be implemented directly, or by performing a weighting and combining operation on the output of a Fast Fourier

Transform. In any case, the filter with the maximum output would be associated with the signal's frequency offset. Such a frequency detector is shown notionally in Figure 10.9. Depending on the signal design and its sensitivity to frequency errors, and on the density of the frequency offsets, either the largest output could be taken as the frequency estimate directly, or additional processing to refine the estimate could be performed. In either case, it is clear that a filter bank that spans the range of possible frequency offsets could be designed in concept, and that such a design would provide a quick and reliable estimate of the carrier frequency offset.

An advantage of the filter bank approach discussed previously is the width of the frequency uncertainty that can be easily accommodated. A disadvantage is the granularity of the initial estimate. A second spectral technique, sometimes called *band-edge filtering*, can provide a much more accurate estimate, at the cost of a reduction in the initial frequency uncertainty that can be accommodated. The idea can be easily seen through a graphical example.

In the upper graph shown in Figure 10.10, the signal-bandpass spectrum is shown as the wide shaded region, centered on a nominal carrier frequency ω_0 . Also shown in this graph are two narrower passband filters at the edges, or roll-off regions of the signal spectrum. If, as is shown in the second graph, the detected signal in both of the two band-edge filters is equal, the signal spectrum must be centered between them, and the nominal carrier frequency error is zero. However, if, as is shown in the third and fourth graphs, the input signal spectrum is shifted relative to the band-edge filters, one of the filters will have more detectable signal, and an

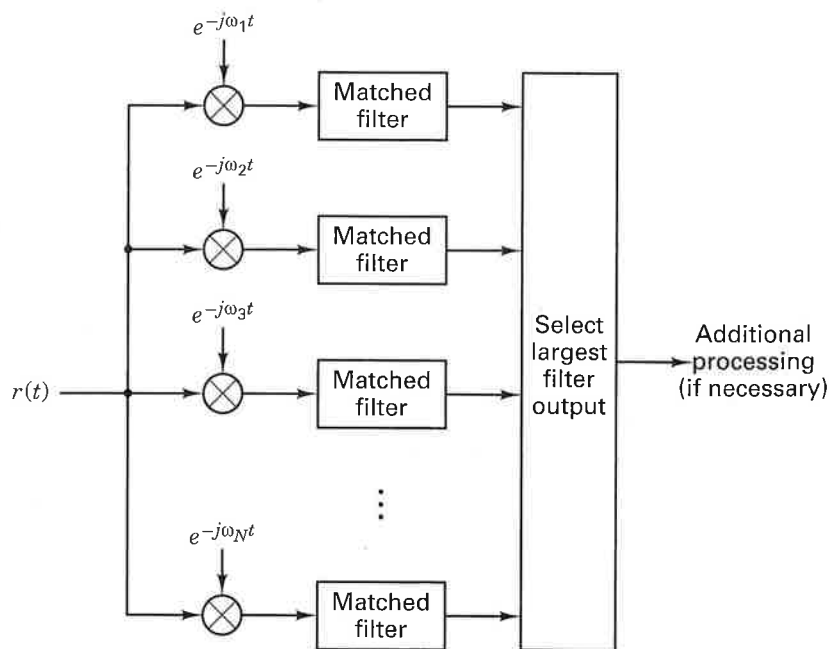


Figure 10.9 Matched filter-bank frequency estimator.

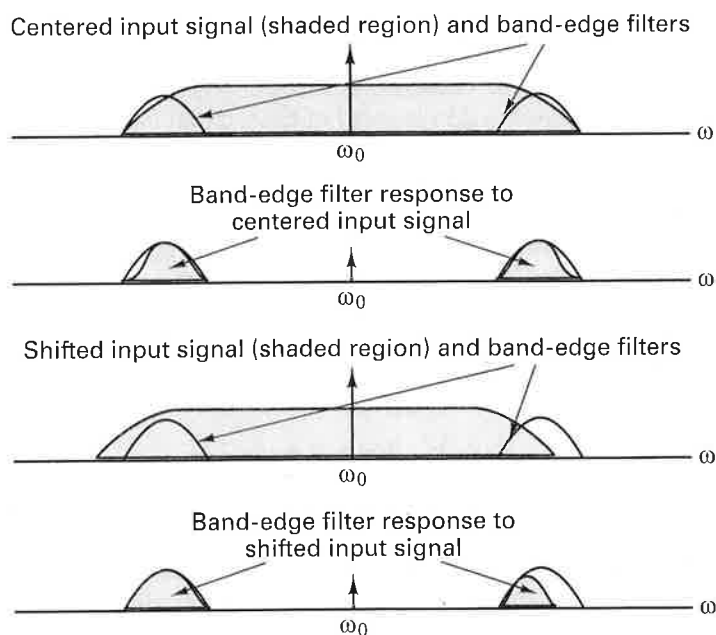


Figure 10.10 Band edge filter example.

error measure can be formed. This error measure could be used to drive a control loop, or it could be used to compute a frequency correction directly. The primary advantage of this kind of technique is that no noise-enhancing nonlinearity is required. The disadvantages are that it requires more knowledge of the signal spectrum, and the implementation of two narrow filters with well-matched passband characteristics. Narrow well-matched filters such as these could represent a design challenge if done with analog circuitry, but, in concept, it could be easily accomplished with digital techniques.

10.2.2 Symbol Synchronization—Discrete Symbol Modulations

All digital receivers need to be synchronized to the incoming digital symbol transitions in order to achieve optimum demodulation. In the discussion that follows, we will consider several of the basic types of designs of symbol or data synchronizers. The discussion will center on a random binary baseband signal, for ease of terminology and notation, but the extension to nonbinary baseband signals should be apparent.

The presentation in this section assumes that nothing is known about the actual data sequence. This class of synchronizers is called non-data-aided (NDA) synchronizers. There is another class of symbol synchronizers that use known information about the data stream. This knowledge may be obtained by feeding back decisions on received data, or because a known sequence has been injected

into the data stream. Data-aided (DA) techniques have become more important and prevalent with the increasing use of bandwidth-efficient modulation. This is especially true with the class of continuous phase modulations. Data-aided techniques will be considered in somewhat more detail in the succeeding section.

The symbol synchronizers that will be considered here can be classified into two basic groups. The first group consists of the open-loop synchronizers. These circuits recover a replica of the transmitter data clock output directly from operations on the incoming data stream. The second group comprises the closed-loop synchronizers. Closed-loop data synchronizers attempt to lock a local data clock to the incoming signal by use of comparative measurements on the local and incoming signals. Closed-loop methods tend to be more accurate, but they are much more costly and complex.

10.2.2.1 Open-Loop Symbol Synchronizers

Open-loop symbol synchronizers are also occasionally called nonlinear filter synchronizers [20], a very descriptive title. This class of synchronizers generates a frequency component at the symbol rate by operating on the incoming baseband sequence with a combination of filtering and a nonlinear device. The operation is analogous to carrier recovery in a suppressed carrier-tracking loop. In the present case, the desired frequency component, at the data symbol rate, is isolated with a bandpass filter, and “shaped” with a high-gain saturating amplifier. The shaping recovers the square-wave appearance of the data clock signal.

Three examples of open-loop bit synchronizers are shown in Figure 10.11. In the first example (Figure 10.11a), the incoming signal $s(t)$ is filtered with a matched filter. The output of this filter will be the autocorrelation function of the input signal shape. For square-wave signaling, for example, the output will be the familiar isosceles-triangular waveshape. The sequence of bit autocorrelation waveshapes is then “rectified” by some type of memoryless even-law nonlinearity—a square-law device, for example. The resulting waveform will have positive amplitude peaks that correspond, to within a time delay, with the input symbol transitions. This sequence of processes is illustrated in Figure 10.12. Thus the output waveform from the even-law device will contain a Fourier component at the fundamental frequency of the data clock. This frequency component is isolated from its harmonics with a bandpass filter (BPF) and shaped with an ideal saturating amplifier, with transfer function

$$\text{sgn } x = \begin{cases} 1 & \text{for } x > 0 \\ -1 & \text{otherwise} \end{cases} \quad (10.53)$$

The second example in Figure 10.11 produces a Fourier component at the data clock frequency by means of a delay and multiply. The delay shown in Figure 10.11b is half a bit period, which is the best value because it provides the strongest Fourier component [20]. The waveform $m(t)$ will always be positive in the second half of every bit period, but will have a negative first half if there has been a state change in the incoming bit stream, $s(t)$. This produces a square-wave signal with

are important on. This is es- a-aided tech- ection. classified into nizers. These y from opera- e closed-loop data clock to and incoming e much more

nonlinear filter rs generates a ring baseband e operation is In the present solated with a he shaping re-

figure 10.11. In with a matched f the input sig- ce the familiar waveshapes is —a square-law mplitude peaks itions. This se- vaveform from idamental fre- 1 its harmonics amplifier, with

(10.53)

ponent at the hown in Figure s the strongest e in the second as been a state ave signal with

on Chap. 10

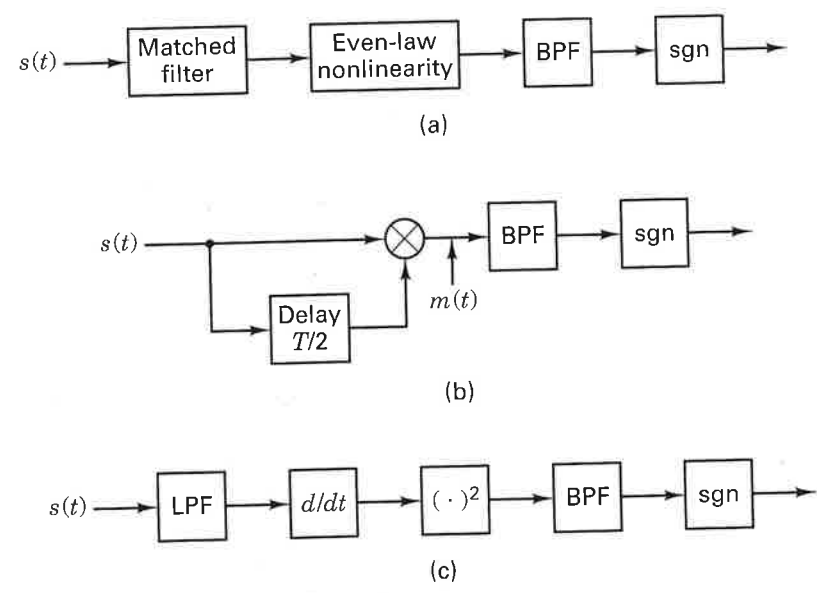


Figure 10.11 Three types of open-loop bit synchronizers.

spectral components at the data rate and all harmonics, as in Figure 10.11a. As before, the appropriate spectral component can be isolated with a BPF and shaped.

The final example (Figure 10.11c) amounts to an edge detector. The main operations are those of differentiation and rectification (by use of a square-law device). For a square-wave input, the differentiator will produce positive or negative spikes at all symbol transitions. When rectified, the resulting sequence of positive spikes will have a Fourier component at the data symbol rate. A potential problem with this particular scheme is that differentiators are typically very sensitive to wideband noise. This necessitates the low-pass filter (LPF) that precedes the differentiator in Figure 10.11c. The LPF, however, will also remove the high frequency components of the data symbols, causing them to lose their original rectangular wave shape. This will cause the resulting differential signal to have some finite rise and fall time, rather than being a set of impulses.

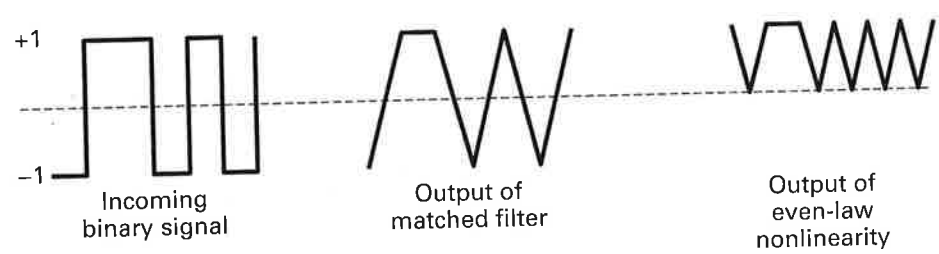


Figure 10.12 Open-loop bit synchronizer illustration.

Clearly, there will be some hardware delay associated with the signal processing steps illustrated in Figure 10.11. Wintz and Luecke [21] have shown that for a BPF that effectively averages K input symbols (bandwidth = $1/KT$), the magnitude of the fractional mean time error (delay) is approximated by

$$\frac{|\bar{\varepsilon}|}{T} \approx \frac{0.33}{\sqrt{KE_b/N_0}} \quad \text{for } \frac{E_b}{N_0} > 5, \quad K \geq 18 \quad (10.54)$$

where T is the bit period, E_b the detected energy per bit, and N_0 the single-sided received noise spectral density. Wintz and Luecke have also shown that at high signal-to-noise ratios the fractional standard deviation of the fractional timing error is given by

$$\frac{\sigma_\varepsilon}{T} \approx \frac{0.411}{\sqrt{KE_b/N_0}} \quad \text{for } \frac{E_b}{N_0} > 1 \quad (10.55)$$

Thus, for a given BPF, when the received signal-to-noise ratio is sufficiently large all of the techniques shown in Figure 10.11 will provide accurate bit timing.

10.2.2.2 Closed-Loop Symbol synchronizers

The primary disadvantage of open-loop symbol synchronization methods is that there is an unavoidable non-zero-mean tracking error. This error can be made small for large signal-to-noise ratios, but since the synchronization signal waveform depends directly on the incoming signal, the error will never vanish.

Closed-loop, symbol-data synchronizers use comparative measurements on the incoming signal and a locally generated data-clock signal to bring the locally generated signal into synchronism with the incoming data transitions. The procedure is essentially the same as that used for closed-loop carrier tracking.

Among the most popular of the closed-loop symbol synchronizers is the early/late-gate synchronizer. An example of such a synchronizer is shown schematically in Figure 10.13. The synchronizer operates by performing two separate integrations of the incoming signal energy over two different $(T - d)$ second portions of a symbol interval. The first integration (the early gate) begins integration at the loop's best estimate of the beginning of a symbol period (the nominal time zero) and integrates for the next $(T - d)$ seconds. The second integral (the late gate) delays the start of its integration for d seconds, and then integrates to the end of the symbol period (the nominal time T). The difference in the absolute values of the outputs of these two integrations, y_1 and y_2 , is a measure of the receiver's symbol-timing error, and it can be fed back to the loop's timing reference to correct loop timing.

The action of the early/late-gate synchronizer can be understood by referring to Figure 10.14. In the case of perfect synchronization, Figure 10.14a shows that both gates are entirely within a signal symbol interval. In this case, both integrators will accumulate the same amount of signal, and their difference (the error signal e , in Figure 10.13) is zero. Thus, when the device is synchronized, it is stable—there is no tendency to drive itself away from synchronization. The case shown in Figure 10.14b is for a receiver whose data clock is early relative to the incoming data. In

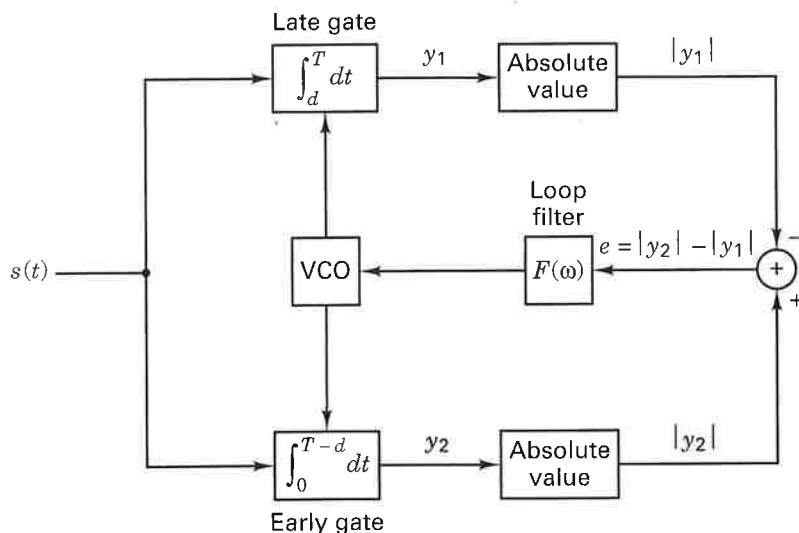


Figure 10.13 Early/late-gate data synchronizer.

in this case the first portion of the early gate falls in the previous bit interval, while the late gate is still entirely inside the current symbol. The late-gate integrator will accumulate signal over its entire $(T - d)$ integration interval, as in the case in Figure 10.14a; but the early-gate integrator will end up with energy accumulated only over $[(T - d) - 2\Delta]$, where Δ is the portion of the early-gate interval falling in the previous bit interval. Thus, for this case, the error signal will be $e = -2\Delta$, which will lower the input voltage to the VCO in Figure 10.13. This will reduce the VCO output frequency and retard the receiver's timing to bring it back toward the incoming signal's bit timing. Using Figure 10.14 as a guide, it can be seen that if the receiver's timing had been late, the amounts of energy integrated in the early gate and late gate would be reversed, as would the sign of the error signal. Thus, late receiver timing produces an increase in the VCO input voltage, increasing the output frequency and advancing the receiver's timing toward that of the incoming signal.

The example illustrated in Figure 10.14 tacitly assumes that there will be data state changes before and after the channel symbol of interest. If there are no transitions, it can be seen that the early gate and late gate will have the same integrated energy. Thus, there will be no error signal generated for cases where there is no data-state change. This is a practical implementation consideration in the use of all symbol synchronizers. Reconsider Figure 10.13. It is not possible to build two integrators that are exactly the same. Thus, the signals from the two arms of the early/late-gate loop will contain an offset with respect to each other, even when they should be identical. This offset will be small for well-designed integrators but will cause the loop to drift out of synchronism if there are long sequences of identical data symbols. There are two common responses to this problem. The first, and perhaps most obvious, is to format the data in a manner which ensures that there will be no transitionless intervals that are long enough to allow the loop to break lock. The sec-

ond response is to modify the loop design so that it contains a single integrator. An example of this type of modified design is the tau-dither loop, considered in conjunction with the synchronization of spread-spectrum systems in Chapter 12.

Another loop design issue is the integration interval of the two gates. The example illustrated in Figure 10.14 shows the gates to occupy about three-fourths of a symbol period. Actually, this interval can vary from half a symbol interval to nearly a whole symbol interval. Why not less than half? The trade-off is between the amount of integrated noise and interference in a gate versus the amount of signal. As was true with the nonlinear model of phase-locked loops, loops of this type are difficult to analyze; the determination of performance is usually via computer simulation. This will be especially true for overlapping gates, as in Figure 10.14, because the noise samples in the two gates will be correlated. Gardner [5] has shown that for a normalized incoming signal of one volt, additive white Gaussian noise, random data (the probability of a transition is $\frac{1}{2}$), and early and late gates that are half a bit interval in duration, for large-loop signal-to-noise ratios, the fractional timing jitter is approximated by

$$\frac{\sigma_e^2}{T^2} = 2 N_0 B_L \quad (10.56)$$

where N_0 is the (normalized) noise power spectral density, T is the symbol interval, and B_L is the loop bandwidth.

10.2.2.3 Symbol Synchronization Errors and Symbol Error Performance

The effect of symbol-synchronization error on bit-error probability for a BPSK signal in additive white Gaussian noise is shown in Figure 10.15. It can be seen from the figure that the degradation is less than about 1 dB in signal-to-noise

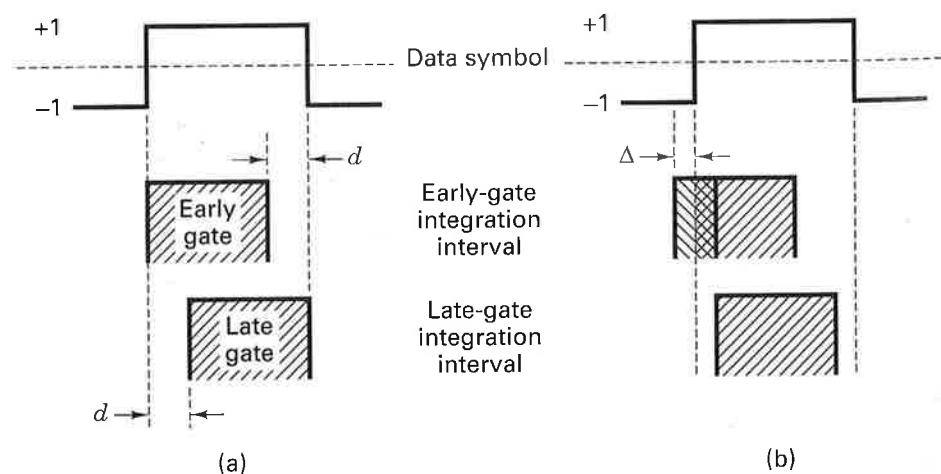


Figure 10.14 (a) Correct receiver timing. (b) Early receiver timing.

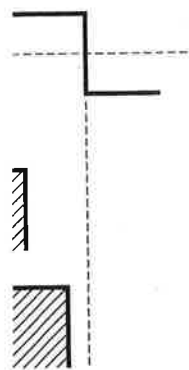
integrator. An
red in conjunc-
12.

gates. The ex-
ee-fourths of a
interval to nearly
is between the
ount of signal.
of this type are
computer simu-
10.14, because
has shown that
sian noise, ran-
es that are half
actional timing

(10.56)

symbol interval,

obability for a
10.15. It can be
signal-to-noise



)

iming.

tion Chap. 10

ratio for a fractional timing jitter of less than 5%. Comparing symbol timing error effects with the effect of phase noise (see Figure 10.8), it can be seen that the symbol synchronization error, taken as a fraction of the symbol interval, does not affect system performance as strongly as does phase noise taken as a fraction of a cycle. In both cases, however, the degradation increases with increases in error.

Example 10.7 Effect of Timing Jitter

Through the use of Figure 10.15, determine the effect of a 10% symbol-fractional timing jitter on a system required to maintain a 10^{-3} bit error probability.

Solution

It can be seen from Figure 10.15 that a 10^{-3} -bit-error probability will require a SNR of about 6.7 dB in the absence of all timing jitter. The same figure indicates that for a fractional timing jitter of 10% ($\sigma_e/T = 0.1$), a SNR of about 12.9 dB is required. Thus, the ability to accommodate this large timing jitter would require a 6.2-dB higher signal-to-noise ratio than that needed to maintain a 10^{-3} -bit-error probability without jitter. This illustrates a use to which Figure 10.15 can be put; however, this example is clearly extreme. No communication system would be designed with over four times the nominally required power level in order to accommodate a large symbol-synchronization error. Some other answer would be found, such as redesigning the system filtering to increase the value of K in Equation (10.55), which will reduce the symbol timing jitter.

10.2.3 Synchronization with Continuous-Phase Modulations (CPM)

10.2.3.1 Background

Continuous-Phase Modulations (CPM) have grown from a research topic to an increasingly important signaling technique because of their bandwidth efficiency. As bandwidth becomes increasingly dear, their importance will continue to increase. These modulations raise new issues in synchronization, especially symbol synchronization. The bandwidth efficiency of CPM is obtained by increasing the smoothness of the waveforms in the time domain. If done properly, this smoothness will concentrate the signal's energy in a narrower bandwidth, reducing the amount of bandwidth required to pass the signal, and allowing adjacent signals to be packed closer together. However, this smoothness in the time domain also tends to eliminate the symbol transition features upon which many symbol synchronization schemes depend. A related problem is that, in general, it is difficult with CPM to separate the effects of carrier-phase error from symbol-timing error, making the phase and timing tasks interrelated. A mitigating feature of the smoothed waveforms is that in most cases of practical interest, the performance of the receivers is relatively insensitive to moderate timing errors [3].

A normalized CPM signal can be represented in complex notation as

$$s(t) = \exp \{ j[\omega_0 t + \theta + \psi(t - \tau, \alpha)] \} \quad (10.57)$$

where ω_0 is the carrier frequency, θ is the carrier phase (measured relative to the phase of the receiver), and $\psi(t, \alpha)$ is called the *excess phase* of $s(t)$. It is $\psi(t, \alpha)$ that carries the information in the signal. It is also $\psi(t, \alpha)$ that determines the amount

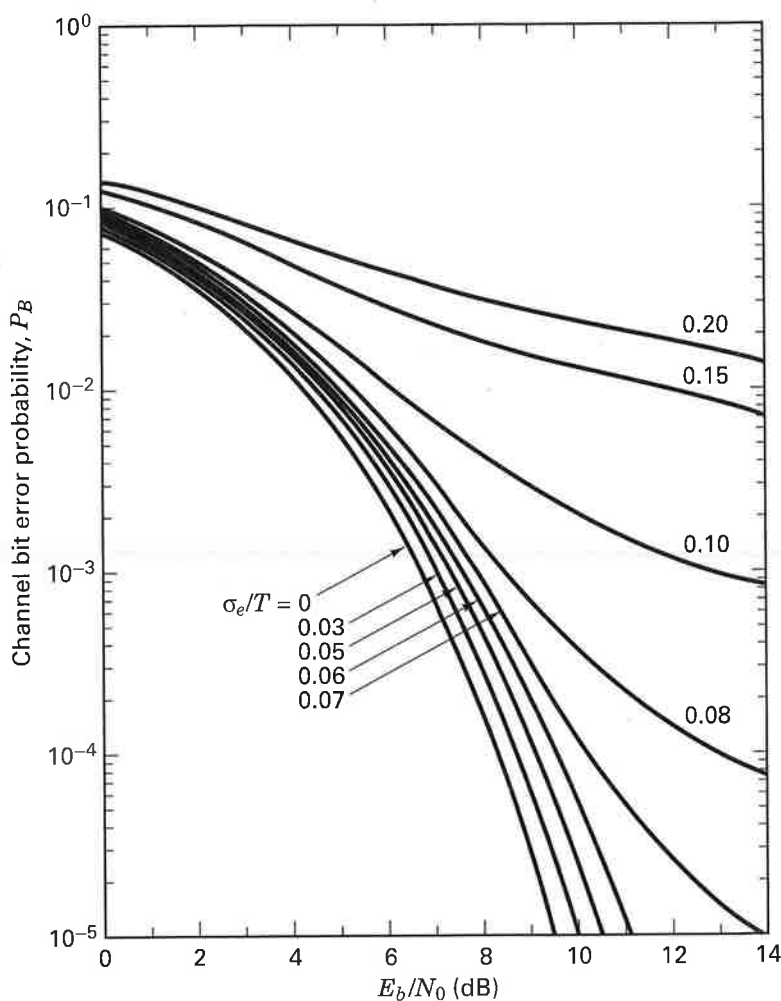


Figure 10.15 Channel bit error probability versus E_b/N_0 with the standard deviation of the symbol sync error σ_e as a parameter. (Reprinted from W. C. Lindsey and M. K. Simon, *Telecommunication Systems Engineering*, Prentice-Hall, Inc., Englewood Cliffs, N.J., 1973, courtesy of W. C. Lindsey and Marvin K. Simon.)

of bandwidth the signal will need. The required bandwidth is sometimes referred to as the *bandwidth occupancy* of the signal. When considering the goal of reducing or minimizing the required bandwidth from the standpoint of Fourier Theory, it can be seen that relatively high frequency components are associated with relatively abrupt changes in the time domain signal [22]. Thus, in order to reduce or eliminate the high frequency components, one must smooth out all the rough edges or abrupt changes in the time domain signal. In CPM signaling, this is accomplished through a combination of three techniques:

1. Use signal pulses that have several orders of continuous derivatives.
2. Allow individual signal pulses to occupy multiple signal time intervals (i.e., intentionally inject some intersymbol interference between symbols).
3. Reduce the maximum allowed phase change per symbol interval.

Not all CPM schemes use all of these techniques, but they all use some. For CPM schemes, it should be noted that at the beginning of each symbol interval, the excess phase, $\psi(t, \alpha)$, is a Markov Process [4], in that it depends only on the phase at the beginning of the symbol and the current symbol value. The phase value at the beginning of the symbol is a consequence of some number of previous symbols. Therefore, for the practical case where there are a finite number of possible phase states, the result is a finite state channel. Thus, the excess phase can be defined as

$$\psi(t, \alpha) = \eta(t, \mathbf{C}_k, \alpha_k) + \Phi_k \quad kT \leq t \leq (k+1)T \quad (10.58)$$

where

$$\eta(t, \mathbf{C}_k, \alpha_k) = 2\pi h \sum_{i=k-L+1}^k \alpha_i q(t - iT) \quad (10.59)$$

$\mathbf{C}_k$ is called the *correlative state*, k is a time index, and α_k is the k th information symbol drawn from the alphabet $\{\alpha_k\} = \{\pm 1, \pm 3, \dots, \pm(M-1)\}$. This alphabet allows for the general case of M -ary (rather than just binary) signaling. The parameter h is the modulation index, and $q(t)$ is called the *modulation phase response*, defined outside of the region $0 < t < LT$, as

$$q(t) = \begin{cases} 0 & \text{for } t \leq 0 \\ 1/2 & \text{for } t \geq LT \end{cases} \quad (10.60)$$

where L is called the correlation length. The correlation length is the number of information-symbol periods of length T seconds that are affected by a single information symbol. This is a measure of the amount of intentional intersymbol interference. When $L = 1$, we refer to the signaling as *full response*. This is the condition that was assumed in earlier chapters dealing with modulation. In full-response signaling, each pulse is confined to its own time boundaries. However, when $L > 1$, the signaling is called *partial response*, which means that each pulse is not restricted to its own symbol interval, but rather it is "smeared" into $L - 1$ neighboring symbol intervals. This is the vehicle used in many CPM schemes for purposely injecting controlled intersymbol interference between symbols and thereby increasing bandwidth efficiency. Classical minimum-shift-keying (MSK), one of the earliest examples of CPM (see Chapter 9), does not use multiple symbol intervals per pulse. Thus, classical MSK is an example of full-response signaling. Observing Equations (10.60), it can be seen that the consequence of $q(LT) = \frac{1}{2}$ is that the maximum possible phase change over an LT interval is $(M-1)\pi h$, as can be seen from Equation (10.58) and (10.59).

The vector $\mathbf{C}_k$, which is called the *correlative state*, is a sequence of information symbols $\{\alpha_k\}$, starting from the earliest time that can affect the signal's phase at the current time k , as follows:



$$\mathbf{C}_k = (\alpha_{k-L+1}, \dots, \alpha_{k-2}, \alpha_{k-1})$$

The term Φ_k in Equation (10.58), expressed as

$$\Phi_k = \pi h \sum_{i=0}^{k-L} \alpha_i \mod 2\pi \quad (10.61)$$

is called the *phase state*. The phase state is one of a set of discrete phases that the signal can take as a consequence of the values of past symbols. Starting the phase transition for the next symbol from this phase state is a necessary condition for continuous phase. In the context of a trellis diagram, Φ_k can be viewed as an initial state or node, and $\mathbf{C}_k$ as defining a path to one of the other nodes. The definition of $q(t)$ in the interval $(0 < t < LT)$ is what gives the modulation its particular characteristics. MSK has the parameters $h = \frac{1}{2}$, $L = 1$, $M = 2$ and $q(t) = t/(2T)$ in the interval $(0 < t < T)$. The frequency response, defined as $g(t) \triangleq dq(t)/dt$, is clearly rectangular for MSK:

$$g(t) = \begin{cases} 1/(2T) & 0 \leq t \leq T \\ 0 & t < 0, \quad t > T \end{cases} \quad (10.62)$$

Gaussian MSK (GMSK), another example of CPM, is defined as having a frequency response that is the convolution of this rectangle with a Gaussian shaped pulse.

Many of the synchronization techniques we have described in previous sections are based on *ad hoc* methods. People who had gained a fair amount of intuition in synchronization invented them by trying things that seemed to make sense. With a few exceptions, intuition has been less successful with CPM. Most techniques are based on classical estimation theory principles, the most popular being maximum likelihood estimation. The principles involved are the same as those developed for maximum likelihood signal detection.

Maximum likelihood estimation involves the maximization of conditional probabilities, and it is based on Bayesian Theory [7]. Let $s(t, \gamma)$ represent a signal with some set of unknown parameters, γ . The parameters could be carrier phase, or symbol-timing offset, or the values of the transmitted information symbols, or possibly other parameters. Let

$$r(t) = s(t, \gamma) + n(t) \quad (10.63)$$

represent a received signal, where $n(t)$ is some additive receiver noise process. Let $R(t)$ represent a realization of the process $r(t)$. Then the maximum-likelihood estimate for the unknown parameter set γ is the value of γ that maximizes the likelihood $p[r(t) = R(t) | \gamma]$ over all γ . As has been developed in Chapter 3, a realization of a maximum-likelihood detector for a known signal is a filter matched to that signal. For the case of CPM, this leads to a receiver structure as shown in Figure 10.16.

In the primary signal detection process, the carrier frequency ω_0 , carrier phase θ , and symbol timing offset τ are assumed known. The receiver structure is effectively a bank of matched filters, each filter matched to an L -symbol signal realization, which feeds a Viterbi Algorithm. The number of filters is M^L , and the number of nodes in the branch metric computation is PM^{L-1} , where P is the num-

(10.61)

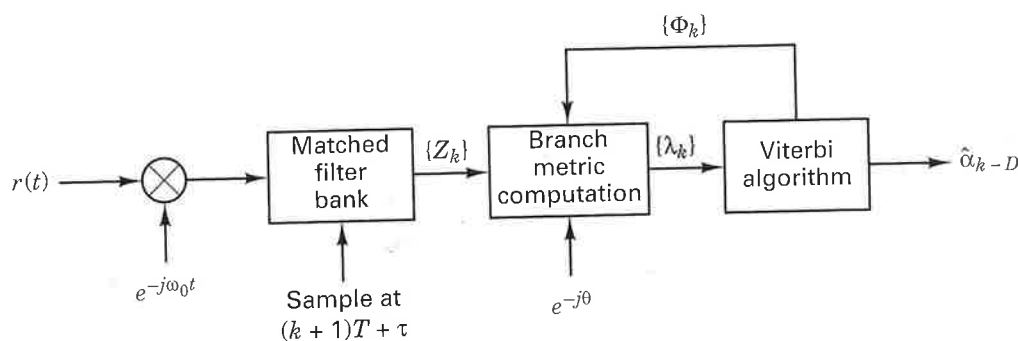


Figure 10.16 CPM receiver structure.
(Note: $\hat{\alpha}_{k-D}$ is the k -th output symbol with processing delay, D)

(10.62)

ber of phase states $\{\Phi_k\}$. Both of these numbers can be awkwardly large, and so simpler receiver structures are often used in practice [3, 4, 22]; however, the structure is still useful as a basis for synchronization.

From the preceding CPM description, the individual filters in the filter bank will have impulse responses given by

$$h^{(\ell)}(t) \triangleq \begin{cases} e^{-j\eta_\ell(T-t, \mathbf{C}_0^{(\ell)}, \alpha_0^{(\ell)})} & 0 \leq t \leq T \\ 0 & \text{elsewhere} \end{cases} \quad (10.64)$$

with $(\ell = 1, 2, \dots, M^L)$ denoting the generic L -symbol string $(\mathbf{C}_0^{(\ell)}, \alpha_0^{(\ell)}) = (\alpha_{-L+1}^{(\ell)}, \dots, \alpha_{-1}^{(\ell)}, \alpha_0^{(\ell)})$, where each $\alpha_k^{(\ell)}$ is selected from the signal alphabet and ℓ denotes a particular path (symbol sequence) in the set of M^L possible paths. Similar to the earlier notation,

$$\eta_\ell(t, \mathbf{C}_0^{(\ell)}, \alpha_0^{(\ell)}) = 2\pi h \sum_{i=-L+1}^0 \alpha_i^{(\ell)} q(t - iT) \quad (10.65)$$

From Figure 10.16, it can be seen that the individual filter outputs are given by

$$Z_k^{(\ell)}(\mathbf{C}_k, \alpha_k, \tau) \triangleq \int_{\tau+kT}^{\tau+(k+1)T} r(t) h^{(\ell)}(t - \tau - kT) e^{-j\omega_0 t} dt \quad (10.66)$$

This set of outputs $\{Z_k\}$, along with the carrier phase estimate $\hat{\theta}$ and the phase state $\{\Phi_k\}$, are used to compute the path metrics and, ultimately, to determine the Viterbi Algorithm's decisions.

10.2.3.2 Data-Aided Synchronization

Techniques for synchronizing CPM receivers can be divided into those that rely on knowledge of the information symbols, and those that do not. Those that rely on such knowledge are called data-aided techniques. Those that do not are termed non-data-aided (NDA). Clearly this is a distinction that could be applied to receiver synchronization for all modulation techniques, but data aided techniques

appear especially useful and popular with CPM. There are two ways in which the data symbols could be known: either the symbols under consideration are a part of a known header or training sequence inserted into the data stream, or decisions from the Viterbi Algorithm are being fed back into the synchronization process. If decision feedback is to be employed, clearly the decisions must be fairly reliable, which implies that the receiver must be close to lock. Initial acquisition based on decision feedback is not likely to be a practical approach.

Given that the transmitted symbol string is known over some observation interval, L_0 , the index (ℓ) in Equation (10.66) can be dropped. With the usual assumptions of Gaussian noise processes and equal energy signals, the likelihood function, $\Lambda(R|\hat{\theta}, \hat{\tau})$, associated with θ and τ , the unknown phase and time offsets, respectively, is given by [3]

$$\Lambda(R|\hat{\theta}, \hat{\tau}) = \exp \left\{ \sum_{k=0}^{L_0-1} \text{Re} [Z_k(\mathbf{C}_k, \alpha_k, \hat{\tau}) e^{-j(\hat{\theta} + \Phi_k)}] \right\} \quad (10.67)$$

where unimportant constant factors have been dropped and $\text{Re}\{\cdot\}$ refers to the real part of the complex argument. It is clear that the right-hand side of Equation (10.67) will be maximized when the summation is maximized. Therefore, taking the partial derivatives of the summation with respect to $\hat{\theta}$ and $\hat{\tau}$ and then setting the results to zero yields

$$\sum_{k=0}^{L_0-1} \text{Im} [Z_k(\mathbf{C}_k, \alpha_k, \hat{\tau}) e^{-j(\hat{\theta} + \Phi_k)}] = 0 \quad (10.68)$$

and

$$\sum_{k=0}^{L_0-1} \text{Re} [Y_k(\mathbf{C}_k, \alpha_k, \hat{\tau}) e^{-j(\hat{\theta} + \Phi_k)}] = 0 \quad (10.69)$$

where $Y_k = \partial Z_k / \partial \hat{\tau}$, and $\text{Im}\{\cdot\}$ refers to the imaginary part of the complex argument. Mengali [3] points out that the left-hand side of Equation (10.69) can be obtained in two ways: either by taking the derivative in the straightforward manner, or by implementing a set of "derivative filters." Which is the preferred implementation would depend on the exact case at hand.

Unfortunately, Equations (10.68) and (10.69) do not provide much in the way of intuition or insight, and there are no known closed-form solutions. The equations must be solved numerically through some jointly iterative procedure for $\hat{\theta}$ and $\hat{\tau}$. Mengali suggests an iterative procedure, where the successive terms in each summation are used to form error terms for a successive approximation. That is,

$$\hat{\theta}_{k+1} = \hat{\theta}_k + \gamma_P e_P(k-1) \quad (10.70)$$

$$\hat{\tau}_{k+1} = \hat{\tau}_k + \gamma_T e_T(k-1) \quad (10.71)$$

where e_P and e_T are higher order terms from the left-hand sides of Equations (10.68) and (10.69), respectively, and γ_P and γ_T are "gains," selected to assure that

the process converges. Clearly, this iterative procedure is more appropriate in a decision-feedback operation than with a fixed-length training sequence.

10.2.3.3 Non-data-Aided Synchronization

One of the first tenets of Information Theory is that having more information is better than having less. In the current context this means that knowing the symbol sequence will allow better estimates of carrier phase and symbol timing than not knowing. There may be cases, however, where training sequences are impractical or inconvenient, and the decision process is not sufficiently reliable for decision feedback. In these cases, non-data-aided (NDA) synchronization processes are called for. Two techniques of general applicability will be discussed, and one power-law technique that can be applied in a large number of cases.

The first technique is a direct extension of the development in the previous section. Clearly, if the symbol sequence (C_k, α_k) is not known, a new likelihood function similar to Equation (10.67) can be written to accommodate that fact:

$$\Lambda(R|\hat{C}_k, \hat{\alpha}_k, \hat{\theta}, \hat{\tau}) = \exp \left\{ \sum_{k=0}^{L_0-1} \text{Re} [Z_k(\hat{C}_k, \hat{\alpha}_k, \hat{\tau}) e^{-j(\hat{\theta} + \Phi_k)}] \right\} \quad (10.72)$$

Since the likelihood function is proportional to a conditional probability, the chain rule of conditional probabilities can be applied to get back to a likelihood function dependent on $\hat{\theta}$ and $\hat{\tau}$ alone. The chain rule states that [7]

$$p(r(t) = R(t)|\gamma) = \int_{\text{all } \beta} p[r(t) = R(t)|\gamma, \beta] p(\beta) d\beta \quad (10.73)$$

which implies that the desired likelihood function is given by

$$\Lambda'(R|\hat{\theta}, \hat{\tau}) = \frac{1}{M^L} \sum_{\text{all } (\hat{C}_k, \hat{\alpha}_k)} \Lambda(R|\hat{C}_k, \hat{\alpha}_k, \hat{\theta}, \hat{\tau}) \quad (10.74)$$

where the assumption has been made that all symbol sequences are equally likely. The likelihood function on the right-hand side of Equation (10.74) can now be differentiated to produce two equations analogous to Equations (10.68) and (10.69). Clearly, the result is considerably more computationally complicated than the results in Equations (10.68) and (10.69), and simpler if sub-optimal techniques may be desired. Mengali [3] discusses some approximations which lead to a somewhat simplified estimator for $\hat{\tau}$.

Another approach is indicated by the form of a sub-optimal receiver structure using Laurent filters [23, 24]. This approach approximates the CPM signal by a set of superimposed pulse amplitude modulated (PAM) waveforms. Considering only the first in the series, the expression is

$$e^{j\psi(t, \alpha)} \approx \sum_i a_{0,i} h_0(t - iT) \quad (10.75)$$

where $\psi(t, \alpha)$ is defined in Equation (10.58) and the coefficients $a_{0,i}$ are called *pseudo-symbols*. The pseudo-symbols, whose values depend on the past and present data symbols, are defined by

$$a_{0,i} = \exp \left(j \pi h \sum_{\ell=0}^i \alpha_{\ell} \right) \quad (10.76)$$

where the modulation index h can take on any noninteger value. For the important special case of MSK, where $h = \frac{1}{2}$, the expression in Equation (10.75) is exact for a filter function of the form

$$h_0(t) = \begin{cases} \sin\left(\frac{\pi t}{2T}\right) & 0 \leq t \leq 2T \\ 0 & \text{elsewhere} \end{cases} \quad (10.77)$$

For other modulations, the approximation will be more or less accurate, and the form of $h_0(t)$ will vary [23]. In any case, ignoring the noise process momentarily, the normalized signal can now be rewritten in the form

$$s(t) \approx e^{j(\omega_0 t + \theta)} \sum_i a_{0,i} h_0(t - iT - \tau) \quad (10.78)$$

From this expression, it is clear that the standard techniques for phase and symbol timing that were developed in the previous sections for linear modulations could be applied to this approximation. Mengali [3] points out that care must be taken in following this approach, however, because a filter actually matched to $h_0(t)$ might produce a very poorly shaped pulse. The issue is discussed by Kaleh [25].

Finally, in the special cases where the modulation index is rational, $h = k_1/k_2$, with (k_1, k_2) integers, power-law techniques can be applied [22]. For this case, Equation (10.57) can be rewritten as

$$s(t) = \exp \left\{ j \left[\omega_0 t + \theta + 2\pi \frac{k_1}{k_2} \sum_{i=k-L+1}^k \alpha_i q(t - iT) \right] \right\} \quad (10.79)$$

where, for simplicity, Φ_k from Equation (10.58) has been absorbed into θ . Taking the k_2^{th} power of $s(t)$ gives

$$[s(t)]^{k_2} = \exp \left\{ j \left[k_2(\omega_0 t + \theta) + 2\pi k_1 \sum_{i=k-L+1}^k \alpha_i q(t - iT) \right] \right\} \quad (10.80)$$

The $\omega_0 t + \theta$ term on the right-hand side is clearly a high-frequency term that can be filtered out. The right-most term is the k_1^{th} power of the information portion of the signal. From the development in Equations (10.57) to (10.60), this last term is repetitive with a period of at most LT . Depending on the exact nature of the phase response $q(t)$, Fourier series components at multiples of $2\pi k_1/(LT)$ radians may be produced. At least in theory, these components could be isolated and tracked. Even if spectral lines are not available, if the multiple of the signaling spectrum can be isolated, band-edge filtering techniques (described in Section 10.2.1.9) could be used to estimate a multiple of the symbol rate. The phase term θk_2 can also be isolated. There are several practical problems with this procedure. The symbol period will have a (k_1/L) -way ambiguity, and the phase estimate will have a k_2 -way ambiguity that must be resolved in some fashion. Depending on the nature of $q(t)$, the

5) is exact for a

curate, and the
momentarily, the

ase and symbol
ations could be
ist be taken in
! to $h_0(t)$ might
[25].
onal, $h = k_1/k_2$,
For this case,

into θ . Taking

portion of the
is last term is
e of the phase
adians may be
and tracked.
spectrum can
(1.9) could be
an also be iso-
symbol period
 k_2 -way ambi-
re of $q(t)$, the

10.2.4 Frame Synchronization

Almost all digital data streams have some sort of frame structure. This is to say that the data stream is organized into uniformly sized groups of bits. If the data stream is digitized TV, each pixel is represented by a word having several bits, which is further organized into horizontal raster scans, which is further organized in terms of vertical raster scans. Computer data are typically organized into words of some number of 8-bit bytes, and these, in turn, are organized into card images, packets, frames, or files. Any system that uses block-error control coding must be organized around the codeword length. Digital speech is typically transmitted in packets or frames which are indistinguishable from other digital data.

For a receiver to make sense of the incoming data stream, the receiver needs to be synchronized with the data stream's frame structure. Frame synchronization is usually accomplished with the aid of some special signaling procedure from the transmitter. This procedure may be very simple, or fairly involved, depending on the environment in which the system is required to operate.

Probably the simplest frame synchronization aid is the frame marker, illustrated in Figure 10.17. The frame marker is a single bit, or a short pattern of bits that the transmitter injects periodically into the data stream. The receiver must know the pattern and the injection interval. The receiver, having achieved data synchronization, correlates the known pattern with the incoming data stream at the known injection interval. If the receiver is not in synchronization with the framing pattern, the accumulated correlation will be low. When the receiver comes into frame synch, however, the correlation should be nearly perfect, blemished only by an occasional detection error.

The advantage of the frame marker is its simplicity. Even a single bit can suffice as a frame marker if a sufficient number of correlations are accumulated before deciding whether or not the system has achieved synchronization. The major drawback is that the sufficient number may be very large, and thus the expected time required to acquire synchronization would be long. Therefore, frame markers are most useful in systems that transmit data continuously, like many telephony and computer links, and would be inappropriate for systems that transmit in iso-

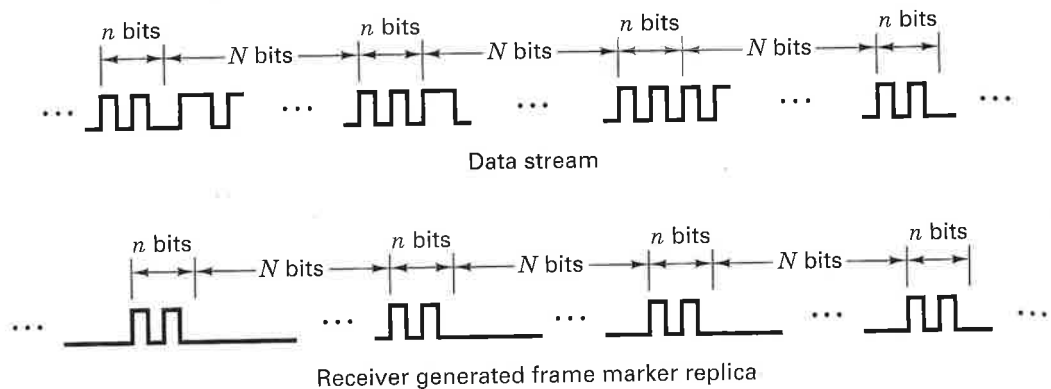


Figure 10.17 Frame marker illustration.

lated bursts or systems that require rapid frame acquisition. A secondary drawback is that the inserted bit(s) may make the organization of the data stream awkward.

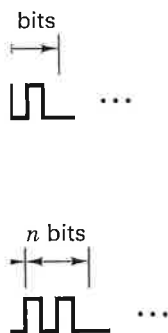
An example is the T1 carrier stream developed by Bell Labs and in general use in North American telephony systems. The T1 carrier structure includes a single bit frame marker after each set of 24 8-bit bytes, each byte representing one of 24 possible voice data streams. This yields a data structure that is an integer multiple of 193 bits—an unhandy number from the standpoint of most integrated circuits.

An approach for systems with inconsistent or bursty transmissions, or systems with rapid acquisition requirements, is a synchronization codeword. A synchronization codeword would typically be sent as part of a message header. The receiver must know the codeword and be constantly searching for it in the data stream, possibly with a matched filter correlator. Detection of the codeword would indicate a known position (typically the beginning) in the data frame. The advantage of this system is that frame acquisition can be essentially immediate. The only delay would be that required to process the incoming codeword. The disadvantage is that the codeword must be long relative to the frame marker, to keep the probability of false detections low. The complexity of the correlation operation is proportional to the length of the sequence, so the correlator may be relatively complicated.

A good synchronization codeword is one that has the property that the absolute value of its “correlation sidelobes” is small. A correlation sidelobe is the value of the correlation of a codeword with a time-shifted version of itself. Thus, the correlation sidelobe value for a k -symbol shift of an N -bit code sequence $\{X_i\}$ is given by

$$C_k = \sum_{j=1}^{N-k} X_j X_{j+k} \quad (10.81)$$

where X_i ($1 \leq i \leq N$) is an individual code symbol taking values ± 1 and the adjacent data symbols (associated with index value $i > N$) are assumed to be zero. An example of correlation sidelobe computation is shown in Figure 10.18. The 5-bit



dary drawback
um awkward.
and in general
includes a sin-
-sents one of
n integer mul-
most integrated

ons, or systems
A synchroniza-
-. The receiver
ta stream, pos-
ould indicate a
vantage of this
ily delay would
age is that the
probability of
proportional to
licated.
ty that the ab-
sidelobe is the
of itself. Thus,
sequence $\{X_i\}$ is

(10.81)

nd the adjacent
be zero. An ex-
0.18. The 5-bit

on Chap. 10

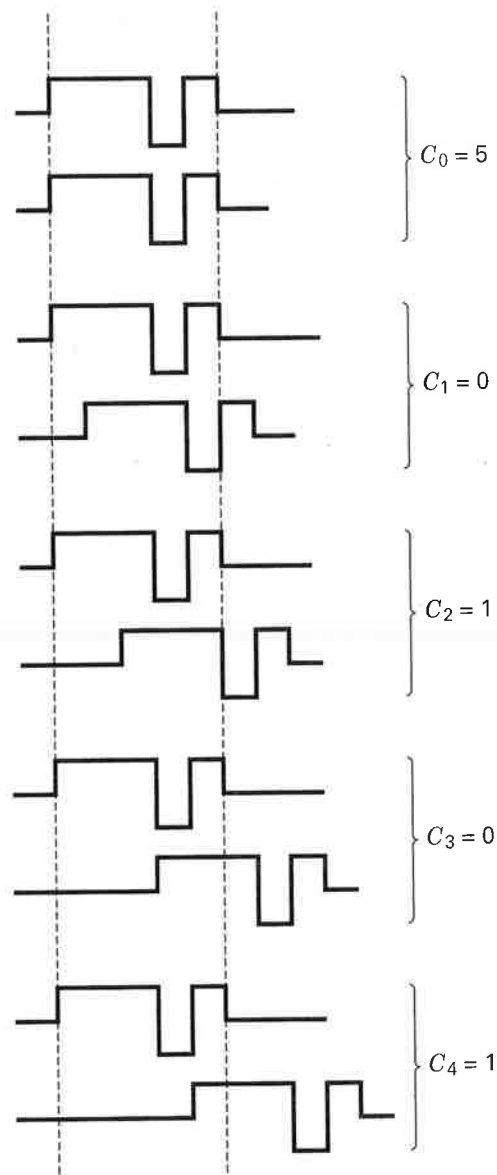


Figure 10.18 Correlation sidelobe example.

sequence in the example is seen to have good correlation properties, in that the largest sidelobe is one-fifth of the main lobe, C_0 . Sequences like the example in Figure 10.18, with the property that their largest sidelobe has a magnitude of unity, are known as Barker sequences or Barker words [26]. There is no known constructive method for finding Barker words, and only 10 unique words are known, the longest of which has 13 symbols. The known unique Barker words are given in Table 10.1. Some thought should make it clear that a completely exhaustive list of known

Barker sequences would include those sequences produced by inverting the sign of the symbols and those produced by reversing the time ordering of the symbols in the sequences of Table 10.1.

The sidelobe correlation properties of Barker codes are based on the assumption that the adjacent symbols have zero value. This is an approximation to the effect of equally likely random binary data adjacent to the Barker word, taking the values ± 1 . Unfortunately, the Barker sequences are too short for this approximation to provide the best codeword in random binary data in all cases. Willard [27] found the best sequences in terms of the minimum probability of false synchronization for random adjacent symbols for the Barker word lengths by the use of a computer simulation. The Willard sequences are shown in Table 10.2.

Two probabilities characterize the performance of a system using a synchronization word. These are the probability of a missed detection and the probability of false alarm. Clearly, the system designer would wish both probabilities to be as small as possible. These are conflicting desires. In order to decrease the probability of a miss, the system design may allow less-than-perfect correlation of an incoming synchronization word. That is, a word may be accepted even if it contains a small number of errors. This, however, enlarges the number of symbol patterns that will be accepted and thereby increases the probability of a false alarm. The probability of a miss for an N -bit word where k or fewer errors are accepted is given by

$$P_m = \sum_{j=k+1}^N \binom{N}{j} p^j (1-p)^{N-j} \quad (10.82)$$

where p is the probability of a detector bit error. The probability of a false alarm generated by N bits of random data is given by

$$P_{FA} = \sum_{j=0}^k \frac{\binom{N}{j}}{2^N} \quad (10.83)$$

It can be seen that for small p , P_m will decrease roughly exponentially with increasing k . Unfortunately, P_{FA} increases roughly exponentially with increasing k . To obtain acceptable values of both P_m and P_{FA} for a given value of p , the system designer often needs values of N larger than those provided by the Barker and

TABLE 10.1 Barker Synchronization Codewords

N	Barker Sequence
1	+
2	++ or +-
3	++-
4	+++ or +-+
5	++++
7	++++-
11	++++-+-
13	++++-++-

TABLE 10.2 Willard Synchronization Codewords

N	Willard Sequences
1	+
2	+ -
3	++ -
4	+++ -
5	++++ -
7	+++++ -
11	++++++ -
13	+++++++ -

Willard sequences. fortunately, there is a fairly large body of literature dealing with longer sequences. Most of these sequences were discovered through exhaustive computer searches. Spilker [20] lists sequences of up to $N = 24$ found by Newman and Hofman [28] and mentions that their original paper had sequences to $N = 100$. Wu [29] provides a list of Maury-Styles sequences to length $N = 30$, a list of Linder sequences to length 40. He also provides a fairly complete discussion of the topic of synchronization sequences, including constructive techniques for reasonable but non-optimum sequences, and insight into the frame synchronization procedures of some operational satellite digital communication systems.

10.3 NETWORK SYNCHRONIZATION

For systems using coherent modulation techniques, one-direction communications such as broadcast channels, or single-link communications, such as most microwave links, land-line links, or fiber-optics links, the synchronization architecture that makes the most sense is to make synchronization totally a receiver function. For communications systems using noncoherent modulation techniques, or that involve many users accessing a central communication node, such as many satellite communication systems, it often makes sense for synchronization to be mostly or entirely a terminal function. This means that the terminal transmitter parameters are modified to achieve synchronization, rather than modifying the central node's receiver parameters. This must be the approach if the system uses time-division multiple access (TDMA). In TDMA, each user is allotted a segment of time in which to transmit its information. The terminal transmitter must be synchronized with the system in order for its transmitted burst of data to arrive at the central node at the time when the node is prepared to receive the data. Synchronization of the terminal transmitter also makes sense with systems that combine signal processing at the central node with frequency-division multiple access (FDMA). If the terminals pre-correct their transmission to be synchronized with the central node, the node can use a fixed set of channel filters and a single timing reference for the processing of all channels. Otherwise, the node would require a separate time and frequency acquisition and tracking capability for each incoming channel, and it would need to deal with the possibility of varying amounts of adjacent channel interference. It

seems clear that terminal transmitter synchronization is often the cleaner, more reasonable system approach to synchronizing a network.

Transmitter synchronization procedures may be classified as being either open loop or closed loop. Open-loop techniques do not rely on any measurement of the arriving signal parameters at the central node. The terminal pre-corrects its transmission based on stored knowledge of link parameters that have been provided by some external authority but may possibly be modified by observations of a return signal from the central node. Open-loop techniques rely on link parameters being accurately known and predictable. They work best when link geometry is nominally fixed, and the links themselves operate continually for relatively long periods, once established. They tend to be difficult to use efficiently when the link geometry is not static or when the terminals access the system sporadically.

The main advantages of the open-loop methods are that acquisition is fast—the procedure can work without a return link, and the amount of real-time computation that is required is small. The disadvantages of open-loop methods are that they require the existence of the external authority that provides knowledge of the required link parameters, and that they are relatively inflexible. The lack of any direct real-time measure of system characteristics means that the system cannot adjust quickly to any unplanned change in conditions.

Closed-loop techniques, on the other hand, require little in the way of *a priori* knowledge of link parameters. Knowledge would be useful in reducing the time required for acquisition, but need not be precise as is required by open-loop methods. Closed-loop methods involve measurements of the synchronization accuracy of the incoming transmissions from the terminal upon their arrival at the central node, and the return of the results of these measurements to the terminal via a return path. Thus, closed-loop methods require a return path that provides a response to the terminal's transmission, the ability in the terminal to recognize the response for what it is, and the ability in the terminal to modify the transmitter characteristics appropriately, based on the response. This amounts to a requirement for a relatively large amount of real-time processing in the terminal, and two-way links between every terminal and the central node. The disadvantages of closed-loop methods are that they require a relatively large amount of real-time processing, require two-way links to every terminal, and that acquisition can take a relatively long time. The advantages are that no external source of knowledge is required for the system to work, and the responses on the return link allow the system to adapt easily and quickly to changing geometries and link conditions.

10.3.1 Open-Loop Transmitter Synchronization

Open-loop systems can be further subdivided into systems that employ information gained by observing a return link, and those that do not. Those that do not are the simplest of all, in terms of real-time processing requirements, but communication performance for these simple terminals is clearly very dependent on stable link characteristics.

All transmitter synchronization schemes attempt to precorrect the timing and transmission frequency of the signal in such a manner that the signal will arrive at a