CHAPTER 18

Trellis-Coded Modulation

All the coding schemes discussed so far have been designed for use with binary-input channels; that is, the encoded bits are represented by one-dimensional BPSK signals according to the mapping $0 \rightarrow -\sqrt{E_s}$ and $1 \rightarrow +\sqrt{E_s}$, or $0 \rightarrow -1$ and $1 \rightarrow +1$ for unit energy signals. (We note here that even nonbinary codes, such as RS codes, are usually transmitted using binary signaling by representing each symbol over $GF(2^m)$ as a binary m-tuple.) In this case the *spectral efficiency* η of the coded system is equal to the code rate R; that is, $\eta = R \leq 1$ bit/dimension or 1 bit/transmitted BPSK symbol, and at most one bit of information is transmitted each time a BPSK symbol is sent over the channel. Thus, since the bandwidth required to transmit a symbol without distortion is inversely proportional to the transmission rate, combining coding with binary modulation always requires *bandwidth expansion* by a factor of $1/R$. In other words, compared with uncoded modulation, the coding gains resulting from binary modulation are achieved at the expense of requiring a larger channel bandwidth.

For the first 25 or so years after the publication of Shannon's paper, research in coding theory concentrated almost exclusively on designing good codes and efficient decoding algorithms for binary-input channels. In fact, it was believed in the early 1970s that coding gain could be achieved only through bandwidth expansion and that coding could serve no useful purpose at spectral efficiencies $\eta \geq 1$ bit/dimension. Thus, in communication applications where bandwidth was limited and large modulation alphabets were needed to achieve high spectral efficiencies, such as data transmission over the dial-up telephone network, coding was not thought to be a viable solution. Indeed, the modulation system design emphasis was almost exclusively on constructing large signal sets in two-dimensional Euclidean space that had the highest possible minimum Euclidean distance between signal points, given certain constraints on average and/or peak signal energy.

In the next two chapters we introduce a combined coding and modulation technique, called *coded modulation*, that achieves significant coding gain without bandwidth expansion. Indeed, coding gain without bandwidth expansion can be achieved independently of the operating spectral efficiency of the modulation system. Thus, coded modulation is referred to as a *bandwidth-efficient* signaling scheme. In this chapter we discuss *trellis-coded modulation* (TCM) [1], a form of coded modulation based on convolutional codes, and in the next chapter we discuss *block-coded modulation* (BCM), based on block codes. Basically, TCM combines ordinary rate $R = k/(k+1)$ binary convolutional codes with an M-ary signal constellation ($M = 2^{k+1} > 2$) in such a way that coding gain is achieved without increasing the rate at which symbols are transmitted, that is, without increasing the required bandwidth, compared with uncoded modulation. For example, a rate $R = 2/3$ convolutional code can be combined with 8-PSK modulation by mapping the three encoder output bits in each T-second time interval into one 8-PSK symbol. This TCM scheme can then be compared with uncoded QPSK modulation, since they both have the same spectral efficiency of $\eta = 2$ bits/symbol[1] or 1 bit/dimension (both QPSK and 8-PSK are two-dimensional signal sets). The key to the technique is that the redundant bits introduced by coding are not used to send extra symbols, as in binary modulation, but instead they are used to expand the size of the signal constellation relative to an uncoded system. Thus coded modulation involves *signal set expansion* rather than bandwidth expansion.

To fairly compare a coded modulation system with an uncoded system, the expanded (coded) signal constellation must have the same average energy as the uncoded constellation, which implies that the signal points must be closer together in two-dimensional Euclidean space, thus reducing the minimum Euclidean distance between signal points. In a TCM system, if the convolutional codes are chosen according to the usual criterion of maximizing the *minimum free Hamming distance* between codewords, and encoder outputs are mapped to signal points in the expanded constellation independently of the code selection, coding gain is not achieved; however, if the code and signal mapping are designed jointly to maximize the *minimum free Euclidean distance* between signal sequences, coding gain can be achieved without expanding the bandwidth or increasing the average energy of the signal set. This joint design is accomplished using a technique known as *mapping by set partitioning* [1].

The original concept of TCM was introduced in a paper by Ungerboeck and Csajka [2] in June 1976, and the essential elements of the idea were later presented in three papers by Ungerboeck [1, 3, 4]. Other early contributions to the basic development of TCM include those by Massey [5], Anderson and Taylor [6], Forney, Gallager, Lang, Longstaff, and Qureshi [7], Calderbank and Mazo [8], Calderbank and Sloane [9], and Forney [10, 11].

A great deal of work has also been devoted to the design of TCM systems for fading (i.e., bursty) channels. As in the case of BPSK modulation, channel interleaving must be used to ensure that received symbols are affected independently by the fading. Unlike the case with binary modulation, however, new code design rules must also be employed for a TCM system to achieve the best possible performance in fading. These issues are developed in a series of papers by Divsalar and Simon [12, 13, 14] and in [15].

TCM systems also can be included in bandwidth-efficient versions of concatenated coding (see, e.g., [16, 17]) and turbo coding (see, e.g., [18, 19, 20]). For readers wishing to investigate the various aspects of TCM in more detail than is presented here, [21] contains a comprehensive view of the subject up to 1990. Also, [22] presents a good overview of the state of the art in 1998.

18.1 INTRODUCTION TO TRELLIS-CODED MODULATION

In our treatment of TCM we assume that the transmitted symbols are drawn from an M-ary signal constellation in either one- or two-dimensional Euclidean space. Several typical signal constellations appear in Figure 18.1. Some one-dimensional, or *amplitude modulation* (AM), signal constellations are shown in Figure 18.1(a). The simplest of these, 2-AM, is equivalent to BPSK. Figure 18.1(b) illustrates several

[1] Throughout this chapter we denote spectral efficiency in units of bits/symbol, where one signal (symbol) is transmitted in each T-second time interval. With this notation, the required bandwidth is proportional to $1/T$, and higher spectral efficiencies are thus more bandwidth efficient.

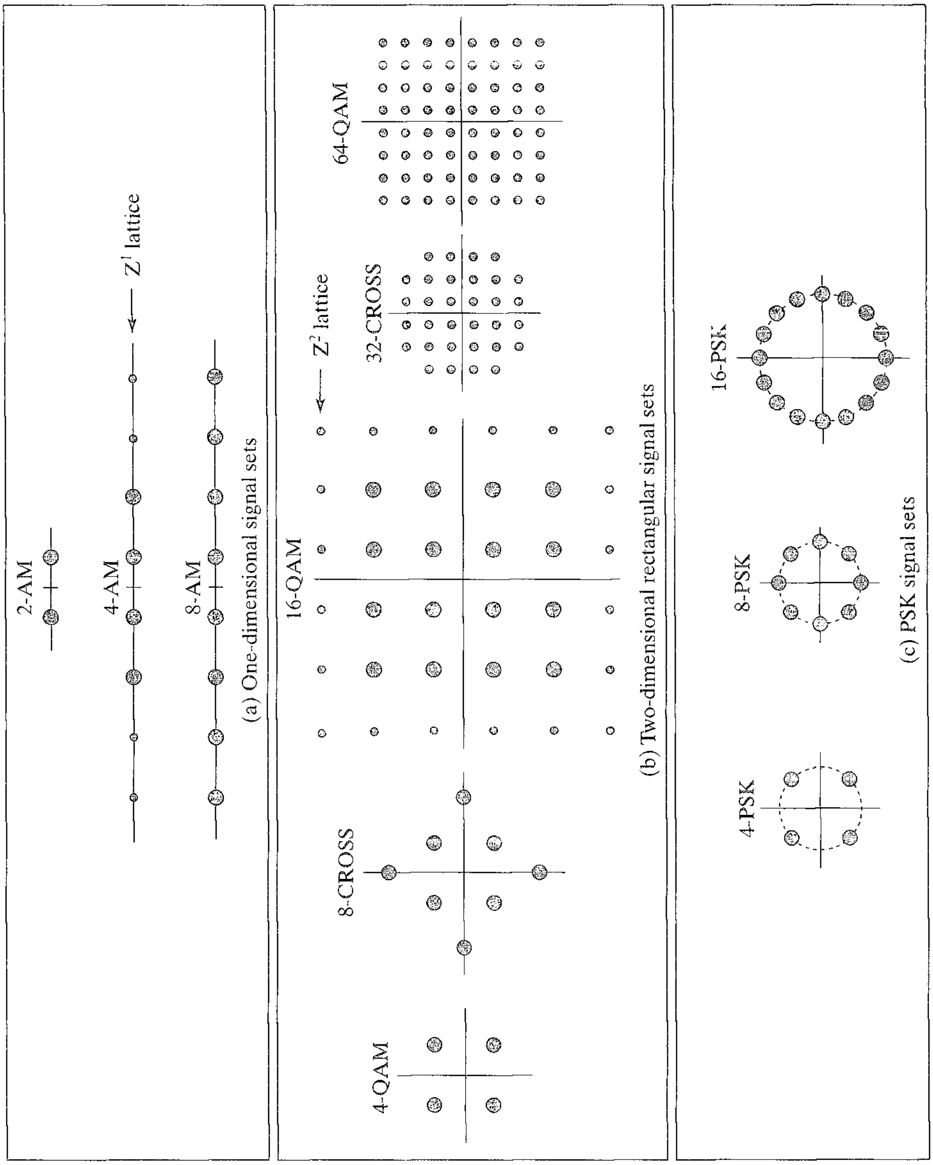


FIGURE 18.1: Typical signal constellations.

two-dimensional signal constellations that exhibit a combination of amplitude modulation and *phase modulation* (AM/PM). Rectangular constellations with $M = (2^p)^2 = 4^p$ signal points, $p = 1, 2, \cdots$, are also referred to as *quadrature amplitude modulation* (QAM) signal sets, since they can be generated by separately applying amplitude modulation to two quadrature carriers (a sine wave and a cosine wave) using a discrete set of 2^p possible amplitudes and then combining the two modulated signals. All practical one-dimensional AM and two-dimensional AM/PM signal constellations can be viewed as subsets of a *lattice*, an infinite array of regularly spaced points translated to its minimum average energy configuration. For example, 4-AM is a (translated) subset of the one-dimensional integer lattice $\mathbb{Z}^1$, whose points consist of all integers in one dimension, and 16-QAM is a (translated) subset of the two-dimensional integer lattice $\mathbb{Z}^2$, whose points consist of all pairs of integers in two dimensions. Finally, some two-dimensional *M-ary phase-shift-keying* (MPSK) signal sets are shown in Figure 18.1(c). MPSK signals all have the same amplitude, and thus they are a form of phase modulation. The simplest of these, 4-PSK (also denoted QPSK), is equivalent to 4-QAM.

Because TCM schemes use signal set expansion rather than additional transmitted symbols to accommodate the redundant bits introduced by coding, performance comparisons must be made with uncoded modulation systems that use smaller signal sets but have the same spectral efficiency, that is, the same number of information bits per transmitted symbol. Thus, care must be exercised to ensure that the different schemes being compared have the same average energy per transmitted symbol. As an illustration, we can compute the average signal energy of the three one-dimensional AM signal sets shown in Figure 18.1(a), where we have assumed that the minimum Euclidean distance between signal points is $d_{min} = 2$, as follows:

$$E_s = 2\frac{[(+1)^2]}{2} = 1 \qquad \text{(2-AM)} \tag{18.1a}$$

$$E_s = 2\frac{[(+1)^2 + (+3)^2]}{4} = 5 \qquad \text{(4-AM)} \tag{18.1b}$$

$$E_s = 2\frac{[(+1)^2 + (+3)^2 + (+5)^2 + (+7)^2]}{8} = 21. \qquad \text{(8-AM)} \tag{18.1c}$$

Thus, in comparing a TCM scheme with one information bit and one redundant bit that uses 4-AM modulation with an uncoded scheme using 2-AM modulation, we must reduce the energy of each signal point in the coded system by a factor of 5, or almost 7 dB, to maintain the same average energy per transmitted symbol; that is, we must scale the amplitude of each signal by the factor $1/\sqrt{5}$. If a TCM scheme with one information bit and two redundant bits using 8-AM is compared with uncoded 2-AM, the energy in the coded system must be reduced by a factor of 21, or more than 13 dB. This reduced signal energy results in a reduced minimum distance between signal points that must be overcome by coding for TCM to achieve a positive coding gain compared with an uncoded system with the same average energy. To minimize the reduction in signal energy of coded systems, practical TCM schemes employ codes with just one redundant bit, that is, rate $R = k/(k+1)$ codes. Thus, TCM system design involves the use of high-rate binary convolutional codes. In Table 18.1 we list the average energies of each signal set shown in Figure 18.1, where the minimum distance between signal points $d_{min} = 2$, and each constellation is in its minimum average energy configuration. The energy requirements of one signal set compared

TABLE 18.1: Average energies for the signal sets in Figure 18.1.

(a) One-dimensional signal sets		
Signal set	E_s	(dB)
2-AM	1	0.0
4-AM	5	7.0
8-AM	21	13.2
(b) Two-dimensional rectangular signal sets		
Signal set	E_s	(dB)
4-QAM	2	3.0
8-CROSS	5.5	7.4
16-QAM	10	10.0
32-CROSS	20	13.0
64-QAM	42	16.2
(c) PSK signal sets		
Signal set	E_s	(dB)
4-PSK	2	3.0
8-PSK	6.8	8.3
16-PSK	26.3	14.2

with another can be determined by simply taking the difference (in decibels) of the values listed in Table 18.1. For example, if uncoded 8-PSK is compared with coded 16-PSK with one redundant bit, we say that the *constellation expansion factor* γ_c of the coded system relative to the uncoded system is $\gamma_c = 14.2$ dB $- 8.3$ dB $= 5.9$ dB.

Now, consider the transmission of a signal sequence (coded or uncoded) from an M-ary signal set $S = \{s_0, s_1, \cdots, s_{M-1}\}$ over an AWGN channel. Let $\mathbf{y}(D) = y_0 + y_1 D + y_2 D^2 + \cdots$ be the transmitted sequence, where $y_l \in S$ for all l, and let $\mathbf{r}(D) = r_0 + r_1 D + r_2 D^2 + \cdots = \mathbf{y}(D) + \mathbf{n}(D)$ be the received sequence, where $\mathbf{n}(D) = n_0 + n_1 D + n_2 D^2 + \cdots$ is the noise sequence, n_l is an independent Gaussian noise sample with zero mean and variance $N_0/2$ per dimension for all l, and r_l and n_l belong to either one- or two-dimensional Euclidean space, depending on whether S is one- or two-dimensional. We can also represent the transmitted, noise, and received sequences by the vectors $\mathbf{y} = (y_0, y_1, y_2, \cdots)$, $\mathbf{n} = (n_0, n_1, n_2, \cdots)$, and $\mathbf{r} = (r_0, r_1, r_2, \cdots)$, and for two-dimensional signal sets, we denote transmitted, noise, and received signal points by $y_l = (y_{il}, y_{jl})$, $n_l = (n_{il}, n_{jl})$, and $r_l = (r_{il}, r_{jl})$, respectively. Throughout the chapter we assume that $\mathbf{r}$ is unquantized; that is, soft demodulator outputs are available at the receiver.

To compute the *symbol-error probability* P_s of an uncoded system, we can consider the transmission of only a single symbol. For example, for the QPSK signal set shown in Figure 18.1(c), if each signal point has energy E_s, that is, its distance from the origin is $\sqrt{E_s}$, we can approximate its symbol error probability on an AWGN channel with one-sided noise power spectral density N_0 with the familiar union upper bound as follows:

$$\begin{aligned} P_s &\leq 2Q\left(\sqrt{\frac{E_s}{N_0}}\right) + Q\left(\sqrt{\frac{2E_s}{N_0}}\right) \\ &\approx 2Q\left(\sqrt{\frac{E_s}{N_0}}\right) \\ &= A_{min}Q\left(\sqrt{\frac{d^2_{min}}{2N_0}}\right) \\ &\approx \frac{A_{min}}{2}e^{-d^2_{min}/4N_0}, \quad \text{(uncoded)} \end{aligned} \tag{18.2}$$

where $d^2_{min} = 2E_s$ is the *minimum squared Euclidean* (MSE) *distance* between signal points, and $A_{min} = 2$ is the *number of nearest neighbors* in the QPSK constellation. (It is not difficult to compute the exact value for P_s in this case, but the approximate expression given in (18.2) allows for a direct comparison with the performance bounds of coded systems.)

For coded transmission, we assume that r is decoded using maximum-likelihood soft-decision Viterbi decoding, as presented in Chapter 12. (For two-dimensional signal constellations, the Viterbi algorithm metric is simply the distance in the two-dimensional Euclidean space.) In this case, given the transmission of a particular coded sequence y, the general form of the union upper bound on the *event-error probability* $P_e(\mathbf{y})$ becomes

$$P_e(\mathbf{y}) \leq \sum_{\mathbf{y}' \neq \mathbf{y}} Q\left(\sqrt{\frac{d^2_E\left(\mathbf{y}, \mathbf{y}'\right)}{2N_0}}\right), \tag{18.3}$$

where

$$d^2_E\left(\mathbf{y}, \mathbf{y}'\right) = \sum_l d^2_E\left(\mathbf{y}_l, \mathbf{y}'_l\right) = \sum_l \|\mathbf{y}_l - \mathbf{y}'_l\|^2 = \sum_l \left[\left(y_{il} - y'_{il}\right)^2 + \left(y_{jl} - y'_{jl}\right)^2\right] \tag{18.4}$$

is the squared Euclidean distance between the coded sequences y and y′. Now, defining d^2_{free} as the *minimum free squared Euclidean* (MFSE) *distance* between y and any other coded sequence y′, and $A_{d_{free}}$ as the *number of nearest neighbors*, we can approximate the bound on $P_e(\mathbf{y})$ as

$$\begin{aligned} P_e(\mathbf{y}) &\leq A_{d_{free}}Q\left(\sqrt{\frac{d^2_{free}}{2N_0}}\right) \\ &\approx \frac{A_{d_{free}}}{2}e^{-d^2_{free}/4N_0}. \quad \text{(coded)} \end{aligned} \tag{18.5}$$

The expressions for event-error probability given in (18.3) and (18.5) are conditioned on the transmission of a particular sequence y, because, in general, TCM systems are nonlinear; however, most known schemes have many of the symmetry properties of linear codes. Typically, d^2_{free} is independent of the transmitted sequence, but $A_{d_{free}}$ can vary depending on the transmitted sequence. The error analysis of TCM schemes is investigated more thoroughly in Section 18.3.

Because the exponential behavior of (18.2) and (18.5) depends on the MSE distances of the uncoded and coded systems, respectively, the *asymptotic coding gain* γ of coded TCM relative to uncoded modulation can be formulated as follows:

$$\gamma = \frac{d^2_{free/coded}/E_{coded}}{d^2_{min/uncoded}/E_{uncoded}}, \tag{18.6}$$

where E_{coded} and $E_{uncoded}$ are the average energies of the coded and uncoded signal sets, respectively. We can rewrite (18.6) as

$$\gamma = \frac{E_{uncoded}}{E_{coded}} \frac{d^2_{free/coded}}{d^2_{min/uncoded}} = \gamma_c^{-1}\gamma_d, \tag{18.7}$$

where γ_c is the constellation expansion factor, and γ_d is the *distance gain factor.*

We proceed by letting d_{min} and Δ_{min} represent the minimum distance between signal points in the uncoded and coded constellations, respectively, and by assuming that the minimum distance between points in the coded (expanded) constellation is reduced, so that the average energies of the coded and uncoded signal sets are equal; that is, $\Delta_{min} < d_{min}$, and $\gamma_c = 1$. Then, the TCM system must have a free distance between coded sequences that is greater than the minimum distance between signal points in the uncoded system to achieve coding gain. In other words, even though $\Delta_{min} < d_{min}$, a TCM system must achieve $d_{free} > d_{min}$.

In the design of codes for binary modulation, the MFSE distance between two signal sequences $\mathbf{y}$ and $\mathbf{y}'$ is given by (see Problem 18.1)

$$d^2_{free} = 4E_s d_{H,free}, \quad \text{(binary modulation)} \tag{18.8}$$

where $d_{H,free}$ is the *minimum free Hamming distance* of the convolutional code. Thus, for binary modulation, the best system design is achieved by choosing the code that maximizes $d_{H,free}$. We will shortly see that, in general, this is not true for TCM system design.

In the TCM case, consider a rate $R = k/(k+1)$ convolutional code with minimum free Hamming distance $d_{H,free}$ in which we denote the $k+1$ encoder output bits at any time unit l by the vector $\mathbf{v}_l = (v_l^{(k)}, v_l^{(k-1)}, \cdots, v_l^{(0)})$. (Throughout the remainder of this chapter, when it is not necessary to specifically denote the time unit l of a vector, the subscript l will be deleted; that is, vectors such as $\mathbf{v}_l$ will be denoted simply by $\mathbf{v}$.) Then, assume that the 2^{k+1} binary vectors $\mathbf{v}$ are mapped into elements s_i of an M-ary signal set S using a one-to-one *mapping function* $f(\mathbf{v}) \rightarrow s_i$, where $M = 2^{k+1}$.

We see from (18.3) that the performance of a TCM system depends on the *squared Euclidean* (SE) distances between signal sequences. Thus, the set of SE distances between all possible pairs of signal points must be determined. Denoting the *binary labels* of two signal points by the vectors $\mathbf{v}$ and $\mathbf{v}'$, we define the *error* *vector* $\mathbf{e} = \mathbf{v} \oplus \mathbf{v}'$ as their modulo-2 sum. It follows that $w_H(\mathbf{e}) = d_H(\mathbf{v}, \mathbf{v}')$; that is, the Hamming weight of an error vector equals the Hamming distance between its two corresponding signal labels. For each error vector $\mathbf{e}$, there are M pairs of signal vectors $\mathbf{v}$ and $\mathbf{v}'$ such that $\mathbf{v}' = \mathbf{v} \oplus \mathbf{e}$, and thus M possible SE distances $\Delta^2_{\mathbf{v}}(\mathbf{e}) = \|f(\mathbf{v}) - f(\mathbf{v} \oplus \mathbf{e})\|^2$. To denote this set of distances, we introduce the *average Euclidean weight enumerator* (AEWE), defined as follows:

$$\Delta^2_{\mathbf{e}}(X) = (1/M) \sum_{\mathbf{v}} X^{\Delta^2_{\mathbf{v}}(\mathbf{e})} = (1/M) \sum_{\mathbf{v}} X^{\|f(\mathbf{v}) - f(\mathbf{v} \oplus \mathbf{e})\|^2}. \qquad (18.9a)$$

Clearly, $\Delta^2_{\mathbf{e}}(X)$ depends both on the signal constellation S and the mapping function $f(\cdot)$. When we are interested only in the MFSE distance d^2_{free} between signal sequences, it is sufficient to define the *minimum Euclidean weight enumerator* (MEWE) of an error vector $\mathbf{e}$ as

$$\delta^2_{\mathbf{e}}(X) = X^{\Delta^2(\mathbf{e})} = X^{\min_{\mathbf{v}} \Delta^2_{\mathbf{v}}(\mathbf{e})}, \qquad (18.9b)$$

where

$$\Delta^2(\mathbf{e}) \overset{\Delta}{=} \min_{\mathbf{v}} \Delta^2_{\mathbf{v}}(\mathbf{e}) \qquad (18.9c)$$

is called the *Euclidean weight* (EW) of $\mathbf{e}$.

The AEWEs and the MEWEs can be used to compute the *average weight enumerating function* $A_{av}(X)$ and the MFSE distance d^2_{free} of a TCM system, respectively. The basic technique is as follows. We label each branch of a conventional trellis for a rate $R = k/(k+1)$ linear binary convolutional code with a vector $\mathbf{v}$ representing the $k+1$ current encoder output bits. Alternatively, we can label each trellis branch with the error vector $\mathbf{e} = \mathbf{v} \oplus \mathbf{v}'$, representing the difference (mod-2 sum) between $\mathbf{v}$ and the corresponding label $\mathbf{v}'$ of an arbitrary reference codeword. Because of linearity, it is easy to see that this *error trellis* is identical to the conventional trellis. If we then modify the error trellis by replacing its binary labels with the Hamming weight enumerators $X^{w_H(\mathbf{e})}$ of each branch, we can compute the weight enumerating function $A(X)$ and the minimum free Hamming distance $d_{H,free}$ of the code by converting the error trellis to an error state diagram and using the standard transfer function approach discussed in Chapter 11. If certain symmetry conditions on the signal constellation S and the mapping function $f(\cdot)$ are satisfied, exactly the same method applies to TCM systems, even though they are in general nonlinear. In this case, we replace the binary labels on the error trellis of the convolutional code with the AEWEs or MEWEs from (18.9), depending on whether we wish to compute $A_{av}(X)$ or d^2_{free}. The AEWEs are in general polynomials in X, indicating that more than one distance can correspond to a given error vector $\mathbf{e}$, whereas the MEWEs, since they represent only the minimum distance corresponding to a given $\mathbf{e}$, are monomials, as in the case of binary convolutional codes. To explain when the transfer function method can be applied to TCM systems, we now introduce the notion of a *uniform mapping*.

First, we partition the signal set S into two subsets, $Q(0)$ and $Q(1)$, such that $Q(0)$ contains the 2^k signal points labeled by a vector $\mathbf{v}$ with $v^{(0)} = 0$, and $Q(1)$ contains the 2^k signal points corresponding to labels with $v^{(0)} = 1$. Next, we let $\Delta^2_{\mathbf{e},0}(X)$ be the AEWE for the subset $Q(0)$, and $\Delta^2_{\mathbf{e},1}(X)$ be the AEWE for the subset $Q(1)$. The subset MEWEs, $\delta^2_{\mathbf{e},0}(X)$ and $\delta^2_{\mathbf{e},0}(X)$, are defined in an analogous way.

Definition 18.1 A one-to-one mapping function $f(\mathbf{v}) \rightarrow s_i$ from a rate $R = k/(k+1)$ convolutional encoder output vector $\mathbf{v} = (v^{(k)}, v^{(k-1)}, \cdots, v^{(0)})$ to a signal point s_i belonging to a 2^{k+1}-ary signal set S is *uniform* if and only if $\Delta^2_{\mathbf{e},0}(X) = \Delta^2_{\mathbf{e},1}(X)$ for all error vectors $\mathbf{e}$.

EXAMPLE 18.1 Uniform Mapping

Consider the three 8-PSK signal sets shown in Figure 18.2 along with their associated labels. Using the signal point labeled by $\mathbf{v} = (000)$ as a reference and assuming unit energy signals, we see that there are four distinct Euclidean distances between 8-PSK signal points:

$$a^2 = \left[1/\sqrt{2}\right]^2 + \left[1 - \left(1/\sqrt{2}\right)\right]^2 = 1/2 + \left(3/2 - \sqrt{2}\right) = 0.586, \quad (18.10a)$$

$$b^2 = 1^2 + 1^2 = 2, \quad (18.10b)$$

$$c^2 = \left[1/\sqrt{2}\right]^2 + \left[1 + \left(1/\sqrt{2}\right)\right]^2 = 1/2 + \left(3/2 + \sqrt{2}\right) = 3.414, \quad (18.10c)$$

$$d^2 = 2^2 = 4. \quad (18.10d)$$

We now examine the eight possible SE distances corresponding to the error vector $\mathbf{e} = (001)$ for the labeling of Figure 18.2(a). We see that there are a total of four code vectors $\mathbf{v}$ for which $\Delta^2_{\mathbf{v}}(\mathbf{e}) = \|f(\mathbf{v}) - f(\mathbf{v} \oplus \mathbf{e})\|^2 = 2$: the vectors $\mathbf{v} = (000), (001), (110)$, and (111), and four code vectors $\mathbf{v}$ for which $\Delta^2_{\mathbf{v}}(\mathbf{e}) = 3.414$, that is, the vectors $\mathbf{v} = (010), (011), (100)$, and (101). Thus, the AEWE for the error vector $\mathbf{e} = (001)$ is $\Delta^2_{\mathbf{e}}(X) = (1/2)X^2 + (1/2)X^{3.414}$. If we now partition the 8-PSK signal set into two subsets $Q(0)$ and $Q(1)$, depending on the value of the bit $v^{(0)}$ in the label vector, we see that for the error vector $\mathbf{e} = (001)$, each subset contains exactly two signal points for which $\Delta^2_{\mathbf{v}}(\mathbf{e}) = 2$, and two signal points for which $\Delta^2_{\mathbf{v}}(\mathbf{e}) = 3.414$; that is, $\Delta^2_{\mathbf{e},0}(X) = \Delta^2_{\mathbf{e},1}(X) = (\frac{1}{2})X^2 + (\frac{1}{2})X^{3.414}$. Repeating this calculation for each possible error vector $\mathbf{e}$ gives us the AEWEs listed, along with the corresponding MEWEs, in Table 18.2(a) for each subset $Q(0)$ and $Q(1)$. Because $\Delta^2_{\mathbf{e},0}(X) = \Delta^2_{\mathbf{e},1}(X)$ for all $\mathbf{e}$, the mapping is uniform.

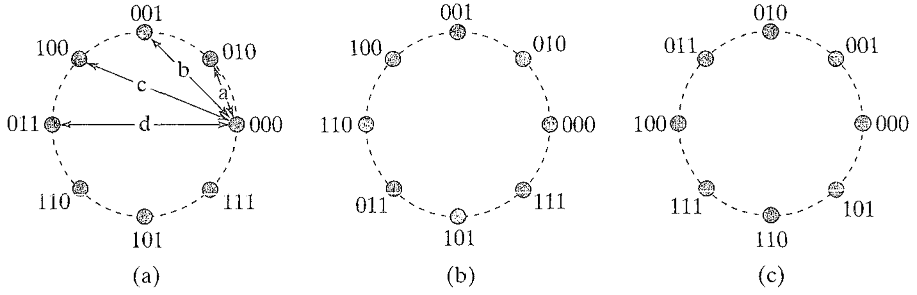


FIGURE 18.2: Three 8-PSK signal sets with different labelings.

TABLE 18.2: The AEWEs and MEWEs corresponding to the three 8-PSK signal sets in Figure 18.2.

(a)

$\mathbf{e}$	$\Delta^2_{\mathbf{e},0}(X)$	$\Delta^2_{\mathbf{e},1}(X)$	$\Delta^2_{\mathbf{e}}(X)$	$\delta^2_{\mathbf{e},0}(X)$	$\delta^2_{\mathbf{e},1}(X)$	$\delta^2_{\mathbf{e}}(X)$
000	X^0	X^0	X^0	X^0	X^0	X^0
001	$\frac{1}{2}X^2 + \frac{1}{2}X^{3.414}$	$\frac{1}{2}X^2 + \frac{1}{2}X^{3.414}$	$\frac{1}{2}X^2 + \frac{1}{2}X^{3.414}$	X^2	X^2	X^2
010	$\frac{1}{2}X^{0.586} + \frac{1}{2}X^2$	$\frac{1}{2}X^{0.586} + \frac{1}{2}X^2$	$\frac{1}{2}X^{0.586} + \frac{1}{2}X^2$	$X^{0.586}$	$X^{0.586}$	$X^{0.586}$
011	$\frac{1}{2}X^{0.586} + \frac{1}{2}X^4$	$\frac{1}{2}X^{0.586} + \frac{1}{2}X^4$	$\frac{1}{2}X^{0.586} + \frac{1}{2}X^4$	$X^{0.586}$	$X^{0.586}$	$X^{0.586}$
100	$\frac{1}{2}X^{3.414} + \frac{1}{2}X^4$	$\frac{1}{2}X^{3.414} + \frac{1}{2}X^4$	$\frac{1}{2}X^{3.414} + \frac{1}{2}X^4$	$X^{3.414}$	$X^{3.414}$	$X^{3.414}$
101	$\frac{1}{2}X^{0.586} + \frac{1}{2}X^2$	$\frac{1}{2}X^{0.586} + \frac{1}{2}X^2$	$\frac{1}{2}X^{0.586} + \frac{1}{2}X^2$	$X^{0.586}$	$X^{0.586}$	$X^{0.586}$
110	$\frac{1}{2}X^2 + \frac{1}{2}X^{3.414}$	$\frac{1}{2}X^2 + \frac{1}{2}X^{3.414}$	$\frac{1}{2}X^2 + \frac{1}{2}X^{3.414}$	X^2	X^2	X^2
111	$\frac{1}{2}X^{0.586} + \frac{1}{2}X^{3.414}$	$\frac{1}{2}X^{0.586} + \frac{1}{2}X^{3.414}$	$\frac{1}{2}X^{0.586} + \frac{1}{2}X^{3.414}$	$X^{0.586}$	$X^{0.586}$	$X^{0.586}$

(b)

$\mathbf{e}$	$\Delta^2_{\mathbf{e},0}(X)$	$\Delta^2_{\mathbf{e},1}(X)$	$\Delta^2_{\mathbf{e}}(X)$	$\delta^2_{\mathbf{e},0}(X)$	$\delta^2_{\mathbf{e},1}(X)$	$\delta^2_{\mathbf{e}}(X)$
000	X^0	X^0	X^0	X^0	X^0	X^0
001	$\frac{1}{4}X^2 + \frac{1}{2}X^{3.414} + \frac{1}{4}X^4$	$\frac{1}{4}X^2 + \frac{1}{2}X^{3.414} + \frac{1}{4}X^4$	$\frac{1}{4}X^2 + \frac{1}{2}X^{3.414} + \frac{1}{4}X^4$	X^2	X^2	X^2
010	$X^{0.586}$	$\frac{1}{2}X^{0.586} + \frac{1}{2}X^{3.414}$	$\frac{3}{4}X^{0.586} + \frac{1}{4}X^{3.414}$	$X^{0.586}$	$X^{0.586}$	$X^{0.586}$
011	$\frac{1}{4}(X^{0.586} + X^2 + X^{3.414} + X^4)$	$\frac{1}{4}(X^{0.586} + X^2 + X^{3.414} + X^4)$	$\frac{1}{4}(X^{0.586} + X^2 + X^{3.414} + X^4)$	$X^{0.586}$	$X^{0.586}$	$X^{0.586}$
100	$X^{3.414}$	$\frac{1}{2}X^2 + \frac{1}{2}X^4$	$\frac{1}{4}X^2 + \frac{1}{2}X^{3.414} + \frac{1}{4}X^4$	$X^{3.414}$	X^2	X^2
101	$\frac{1}{2}X^{0.586} + \frac{1}{2}X^2$	$\frac{1}{2}X^{0.586} + \frac{1}{2}X^2$	$\frac{1}{2}X^{0.586} + \frac{1}{2}X^2$	$X^{0.586}$	$X^{0.586}$	$X^{0.586}$
110	$\frac{1}{2}X^2 + \frac{1}{2}X^4$	$\frac{1}{2}X^{0.586} + \frac{1}{2}X^{3.414}$	$\frac{1}{4}(X^{0.586} + X^2 + X^{3.414} + X^4)$	X^2	$X^{0.586}$	$X^{0.586}$
111	$\frac{1}{4}X^{0.586} + \frac{1}{2}X^2 + \frac{1}{4}X^{3.414}$	$\frac{1}{4}X^{0.586} + \frac{1}{2}X^2 + \frac{1}{4}X^{3.414}$	$\frac{1}{4}X^{0.586} + \frac{1}{2}X^2 + \frac{1}{4}X^{3.414}$	$X^{0.586}$	$X^{0.586}$	$X^{0.586}$

(continued overleaf)

TABLE 18.2: (*continued*)

(c)						
$\mathbf{e}$	$\Delta^2_{\mathbf{e},0}(X)$	$\Delta^2_{\mathbf{e},1}(X)$	$\Delta^2_{\mathbf{e}}(X)$	$\delta^2_{\mathbf{e},0}(X)$	$\delta^2_{\mathbf{e},1}(X)$	$\delta^2_{\mathbf{e}}(X)$
000	X^0	X^0	X^0	X^0	X^0	X^0
001	$\frac{3}{4}X^{0.586}+\frac{1}{4}X^{3.414}$	$\frac{3}{4}X^{0.586}+\frac{1}{4}X^{3.414}$	$\frac{3}{4}X^{0.586}+\frac{1}{4}X^{3.414}$	$X^{0.586}$	$X^{0.586}$	$X^{0.586}$
010	X^2	X^2	X^2	X^2	X^2	X^2
011	$\frac{3}{4}X^{0.586}+\frac{1}{4}X^{3.414}$	$\frac{3}{4}X^{0.586}+\frac{1}{4}X^{3.414}$	$\frac{3}{4}X^{0.586}+\frac{1}{4}X^{3.414}$	$X^{0.586}$	$X^{0.586}$	$X^{0.586}$
100	X^4	X^2	$\frac{1}{2}X^2+\frac{1}{2}X^4$	X^4	X^2	X^2
101	$\frac{1}{4}X^{0.586}+\frac{3}{4}X^{3.414}$	$\frac{1}{4}X^{0.586}+\frac{3}{4}X^{3.414}$	$\frac{1}{4}X^{0.586}+\frac{3}{4}X^{3.414}$	$X^{0.586}$	$X^{0.586}$	$X^{0.586}$
110	X^2	X^4	$\frac{1}{2}X^2+\frac{1}{2}X^4$	X^2	X^4	X^2
111	$\frac{1}{4}X^{0.586}+\frac{3}{4}X^{3.414}$	$\frac{1}{4}X^{0.586}+\frac{3}{4}X^{3.414}$	$\frac{1}{4}X^{0.586}+\frac{3}{4}X^{3.414}$	$X^{0.586}$	$X^{0.586}$	$X^{0.586}$

The AEWEs and MEWEs corresponding to each error vector $\mathbb{e}$ for the mappings in Figures 18.2(b) and 18.2(c) are likewise listed in Tables 18.2(b) and 18.2(c), respectively. In the case of the 8-PSK mapping shown in Figure 18.2(b), we see that three error vectors, namely, $\mathbb{e} = (010)$, (100), and (110), result in $\Delta^2_{\mathbb{e},0}(X) \neq \Delta^2_{\mathbb{e},1}(X)$, and for $\mathbb{e} = (100)$ and (110) the Euclidean weights in the two subsets are different. Thus, this is a *nonuniform* mapping. Finally, for the 8-PSK mapping shown in Figure 18.2(c), we see that the error vectors $\mathbb{e} = (100)$ and (110) result in $\Delta^2_{\mathbb{e},0}(X) \neq \Delta^2_{\mathbb{e},1}(X)$ and in different Euclidean weights for $Q\,(0)$ and $Q\,(1)$. Thus, the mapping in Figure 18.2(c) is also nonuniform.

The following remarks relate to Example 18.1:

- For the uniform mapping in Figure 18.2(a), $\Delta^2_{\mathbb{e}}(X) = \Delta^2_{\mathbb{e},0}(X) = \Delta^2_{\mathbb{e},1}(X)$, and the MEWEs and the EWs are also equal in the two subsets $Q\,(0)$ and $Q\,(1)$ for all error vectors $\mathbb{e}$. This is true for any uniform mapping.
- In Figures 18.2(a) and (c), the subsets $Q\,(0)$ and $Q\,(1)$ are *isomorphic*; that is, one subset can be obtained from the other by some combination of rotation, translation, and reflection about an axis of the signal points. A one-to-one mapping that takes one signal set into another isomorphic signal set, thus preserving the set of distances between signal points, is called an *isometry*.
- For a mapping to be uniform, it is necessary that there be an isometry between the subsets $Q\,(0)$ and $Q\,(1)$; however, the existence of an isometry is not a sufficient condition for uniformity. Thus, even when an isometry exists, a mapping may be nonuniform, as is the case in Figure 18.2(c) (also see Problem 18.3).
- In Figure 18.2(b) there is no isometry between the subsets $Q\,(0)$ and $Q\,(1)$. Thus, this mapping cannot be uniform.

We now let $\mathbb{v}(D)$ and $\mathbb{v}'(D) = \mathbb{v}(D) \oplus \mathbb{e}(D)$ be any two sequences in the binary code trellis, where

$$\mathbb{v}(D) = \mathbb{v}_0 + \mathbb{v}_1 D + \mathbb{v}_2 D^2 + \cdots, \tag{18.11a}$$

$$\mathbb{v}'(D) = \mathbb{v}'_0 + \mathbb{v}'_1 D + \mathbb{v}'_2 D^2 + \cdots, \tag{18.11b}$$

and

$$\mathbb{e}(D) = \mathbb{e}_h D^h + \mathbb{e}_{h+1} D^{h+1} + \cdots + \mathbb{e}_{h+L} D^{h+L}, \quad \mathbb{e}_h, \mathbb{e}_{h+L} \neq \mathbb{0}, \; L \geq 0, \; h \geq 0, \tag{18.11c}$$

is a nonzero path through the error trellis of length $L+1$ branches; that is, $\mathbb{v}(D)$ and $\mathbb{v}'(D)$ differ in at most $L+1$ branches. The term $\mathbb{e}(D)$ then represents an error event of length $L+1$. If $\mathbb{y}(D)$ and $\mathbb{y}'(D)$ are the two channel signal sequences corresponding to $\mathbb{v}(D)$ and $\mathbb{v}'(D)$, respectively, that is, $\mathbb{y}(D) = f(\mathbb{v}_0) + f(\mathbb{v}_1)D + f(\mathbb{v}_2)D^2 + \cdots$ and $\mathbb{y}'(D) = f(\mathbb{v}'_0) + f(\mathbb{v}'_1)D + f(\mathbb{v}'_2)D^2 + \cdots$, then the SE distance between $\mathbb{y}(D)$ and $\mathbb{y}'(D)$ is given by

$$\begin{aligned} d_E^2[\mathbf{y}(D), \mathbf{y}'(D)] &= \sum_{(h \le l \le h+L)} d_E^2\left[f(\mathbf{v}_l), f(\mathbf{v}_l')\right] \\ &= \sum_{(h \le l \le h+L)} \| f(\mathbf{v}_l) - f(\mathbf{v}_l \oplus \mathbf{e}_l) \|^2 \\ &= \sum_{(h \le l \le h+L)} \Delta_{\mathbf{v}_l}^2(\mathbf{e}_l) \\ &\ge \sum_{(h \le l \le h+L)} \Delta^2(\mathbf{e}_l) \triangleq \Delta^2[\mathbf{e}(D)]. \end{aligned} \tag{18.12}$$

where the inequality follows from the definition of Euclidean weight given in (18.9c), and $\Delta^2[\mathbf{e}(D)]$ is called the Euclidean weight of the error sequence $\mathbf{e}(D)$. We now prove a lemma that establishes the conditions under which the Euclidean weights can be used to compute the MFSE distance d_{free}^2 of a TCM system.

> **Lemma 18.1 (Rate $R = k/(k+1)$ Code Lemma)** [1] Assume the mapping from the output vector $\mathbf{v}$ of a rate $R = k/(k+1)$ binary convolutional encoder to the elements of a 2^{k+1}-ary signal set S is uniform. Then, for each binary error sequence $\mathbf{e}(D)$ in the error trellis, there exists a pair of signal sequences $\mathbf{y}(D)$ and $\mathbf{y}'(D)$ such that (18.12) is satisfied with equality.
>
> ***Proof.*** From the definition of Euclidean weight, $\Delta^2(\mathbf{e}_l) = \min_{\mathbf{v}_l = [v_l^{(k)}, \cdots, v_l^{(1)}, v_l^{(0)}]} \Delta_{\mathbf{v}_l}^2(\mathbf{e}_l)$ for all time units l. Because the mapping is uniform, minimizing over the k-bit vector $[v_l^{(k)}, \cdots, v_l^{(1)}]$ yields the same result in the subset $Q(0)$ with $v_l^{(0)} = 0$ as in the subset $Q(1)$ with $v_l^{(0)} = 1$; that is, the Euclidean weight is independent of the value of $v_l^{(0)}$, and $\Delta^2(\mathbf{e}_l) = \min_{[v_l^{(k)}, \cdots, v_l^{(1)}]} \Delta_{\mathbf{v}_l}^2(\mathbf{e}_l)$. Further, a rate $R = k/(k+1)$ encoder can produce any sequence of k-bit vectors $[v_l^{(k)}, \cdots, v_l^{(1)}]$ (only one bit is constrained); that is, every such sequence of k-bit vectors corresponds to a path through the trellis. Thus, for each binary error sequence $\mathbf{e}(D)$, there exists an encoder output sequence $\mathbf{v}(D)$ such that (18.12) is satisfied with equality. **Q.E.D.**

Lemma 18.1 implies that the MFSE distance d_{free}^2 between signal sequences can be computed by replacing the binary labels on the error trellis with the MEWEs and finding the minimum-weight path through the trellis; that is,

$$d_{free}^2 = \min_{\mathbf{e}(D) \neq \mathbf{0}(D)} \Delta^2[\mathbf{e}(D)]. \tag{18.13}$$

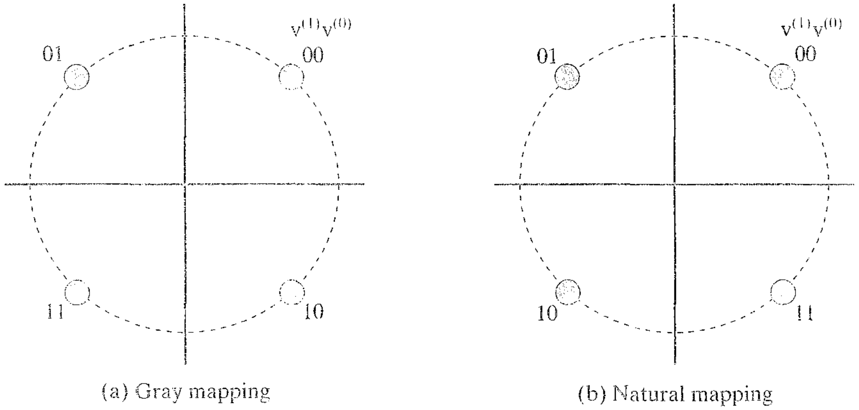
A similar argument can be used to show that the average weight enumerating function $A_{av}(X)$ can be computed by labeling the error trellis with the AEWEs and finding the transfer function of the modified state diagram (see Problem 18.4). If the mapping is nonuniform, the rate $R = k/(k+1)$ code lemma does not hold, and the computation of $A_{av}(X)$ and d_{free}^2 becomes much more complex. An example illustrating this point is given later in this section.

The technique for using the AEWEs to compute $A_{av}(X)$ will be presented in Section 18.3. In the remainder of this section we present a series of examples illustrating the basic principles of designing a TCM system to maximize d_{free}^2.

EXAMPLE 18.2 Rate $R = 1/2$ Trellis-Coded QPSK

Consider a rate $R = 1/2$ convolutional code with minimum free Hamming distance $d_{H,free}$ in which we denote the two encoder output bits by the vector $\mathbf{v} = (v^{(1)}v^{(0)})$. In Figure 18.3 we show two rules for mapping the vector $\mathbf{v}$ into the QPSK signal set: *Gray mapping* and *natural mapping*. Gray mapping, which assigns labels to adjacent signals that differ in only one bit position, is commonly used in uncoded modulation, so the most likely error symbols, the nearest neighbors, will differ in only one bit position from the correct symbol. Natural mapping, which assigns labels in the order of their integer equivalents, is often used in TCM system applications that require insensitivity to carrier phase offset. This topic will be discussed in more detail in Section 18.4. Each signal is assumed to have unit energy, so the MSE distance between signal points is $\Delta^2_{min} = 2$.

For Gray-mapped QPSK, we see that the SE distance between all four pairs of signal points that differ by the error vector $\mathbf{e} = (01)$ is equal to 2, and thus $\Delta^2_{\mathbf{v}}(01) = 2$ for all $\mathbf{v}$, the AEWE is given by $\Delta^2_{(01)}(X) = X^2$, the MEWE is $\delta^2_{(01)}(X) = X^2$, and the EW is $\Delta^2(01) = \min_{\mathbf{v}} \Delta^2_{\mathbf{v}}(01) = 2$. In Table 18.3 we list, for both Gray and natural mapping of QPSK, the four possible error vectors $\mathbf{e}$, their Hamming weights $w_H(\mathbf{e})$, and the four corresponding AEWEs $\Delta^2_{\mathbf{e}}(X)$ and MEWEs $\delta^2_{\mathbf{e}}(X)$.



(a) Gray mapping

(b) Natural mapping

FIGURE 18.3: Two mapping rules for the QPSK signal set.

TABLE 18.3: Euclidean distance structure for Gray- and naturally mapped QPSK.

		(a) Gray mapping		(b) Natural mapping	
$\mathbf{e}$	$w_H(\mathbf{e})$	$\Delta^2_{\mathbf{e}}(X)$	$\delta^2_{\mathbf{e}}(X)$	$\Delta^2_{\mathbf{e}}(X)$	$\delta^2_{\mathbf{e}}(X)$
00	0	X^0	X^0	X^0	X^0
01	1	X^2	X^2	X^2	X^2
10	1	X^2	X^2	X^4	X^4
11	2	X^4	X^4	X^2	X^2

From Table 18.3 we see that all the AEWEs are monomials and that $\Delta_{\mathbb{e}}^2(X) = \delta_{\mathbb{e}}^2(X)$ for all $\mathbb{e}$. Such mappings are called *regular* mappings, which implies that they are also *uniform* mappings. Further, in the case of Gray-mapped QPSK, the two error vectors for which $w_H(\mathbb{e}) = 1$ result in $\Delta_{\mathbb{v}}^2(\mathbb{e}) = 2$ for all $\mathbb{v}$, and the error vector for which $w_H(\mathbb{e}) = 2$ results in $\Delta_{\mathbb{v}}^2(\mathbb{e}) = 4$ for all $\mathbb{v}$; that is, $\Delta_{\mathbb{v}}^2(\mathbb{e}) = 2w_H(\mathbb{e})$ for all $\mathbb{e}$ and $\mathbb{v}$. In other words, there is a linear relationship between SE distance and Hamming distance. Thus, the rate $R = 1/2$ convolutional codes with the best minimum free Hamming distance $d_{H,free}$ will also have the best MFSE distance d_{free}^2 when combined with Gray-mapped QPSK. For example, the optimum free distance (2, 1, 2) code with $d_{H,free} = 5$, when used with Gray-mapped QPSK, results in an MFSE distance of $d_{free}^2 = 10$. Compared with uncoded BPSK with unit energy signals and $d_{min}^2 = 4$, this (2, 1, 2) code results in an asymptotic coding gain of $\gamma = 10\log_{10}(d_{free}^2/d_{min}^2) = 10\log_{10}(10/4) = 3.98$ dB, exactly the same as when this code is used with BPSK modulation. Thus, designing optimum TCM schemes for Gray-mapped QPSK is identical to finding optimum binary convolutional codes for BPSK modulation.

For naturally mapped QPSK, however, the situation is different. The two error vectors for which $w_H(\mathbb{e}) = 1$ give, for all $\mathbb{v}$, $\Delta_{\mathbb{v}}^2(\mathbb{e}) = 2$ in one case and $\Delta_{\mathbb{v}}^2(\mathbb{e}) = 4$ in the other case, and the error vector for which $w_H(\mathbb{e}) = 2$ gives $\Delta_{\mathbb{v}}^2(\mathbb{e}) = 2$ for all $\mathbb{v}$. In other words, there is no linear relationship between SE distance and Hamming distance when natural mapping is used. Thus, traditional code design techniques will not give the best codes for use with naturally mapped QPSK.

Continuing with the naturally mapped case, let us now consider two different (2, 1, 2) code designs:

$$\text{Code 1}: \mathbb{G}_1(D) = \begin{bmatrix} 1 + D^2 & 1 + D + D^2 \end{bmatrix} \tag{18.14a}$$

$$\text{Code 2}: \mathbb{G}_2(D) = \begin{bmatrix} 1 + D^2 & D \end{bmatrix}. \tag{18.14b}$$

Code 1 is the optimum free distance (2, 1, 2) code with $d_{H,free} = 5$, whereas code 2 is suboptimum and has $d_{H,free} = 3$. The encoder diagrams for these two codes are shown in Figure 18.4(a), and their error trellises with binary labels are shown in Figure 18.4(b). Now, replacing the binary labels with the MEWEs of naturally mapped QPSK from Table 18.3(b), we obtain the modified error trellises of Figure 18.4(c). Examining the modified error trellises for the minimum-weight error events, we see that $d_{free}^2 = 6$ for code 1, resulting in a coding gain of $\gamma = 1.76$ dB compared with uncoded BPSK, whereas code 2 achieves $d_{free}^2 = 10$ and $\gamma = 3.98$ dB. Thus code 2, clearly inferior to code 1 for binary modulation or for Gray-mapped QPSK, is the better choice for naturally mapped QPSK.

The following comments apply to Example 18.2:

- The linear relationship between Hamming distance and Euclidean distance in Gray-mapped QPSK is unique among nonbinary signal sets. In all other cases, no such linear relationship exists, and the best TCM schemes must be determined by jointly designing the code and the signal set mapping.

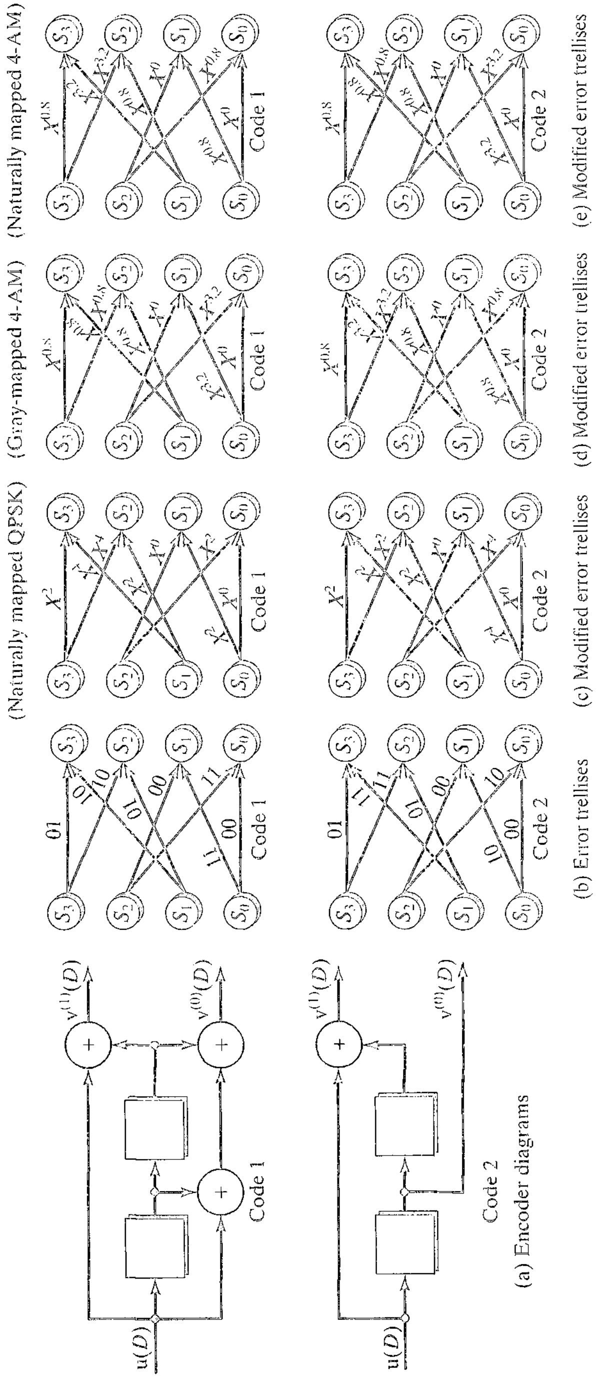


FIGURE 18.4: Encoder diagrams and error trellises for two (2, 1, 2) binary convolutional codes.

- The nonlinearity of most TCM systems arises from the mapping function $f(\cdot)$, which does not preserve a linear relationship between Hamming distance and Euclidean distance.
- Both signal mappings in Example 18.2 are *regular*; that is, each error vector $\mathbf{e}$ has a unique SE distance associated with it, and the AEWEs are equal to the MEWEs for all $\mathbf{e}$. For regular mappings, the Euclidean weight enumerating function $A(X)$ of the code is independent of the transmitted sequence. Thus, $A(X)$ and d^2_{free} can be computed in the same way as for linear convolutional codes with binary signal sets, that is, by assuming that the code sequence corresponding to the all-zero information sequence is transmitted.
- The critical step in the design of code 2 for naturally mapped QPSK was to assign the error vector $\mathbf{e} = (10)$ with maximum Euclidean weight to the two branches in the trellis that diverge from and remerge with the all-zero state S_0. This assignment guarantees the best possible Euclidean distance in the first and last branches of an error event and is one of the key rules of good TCM system design.
- Each of the coding gains quoted in this example came at the expense of bandwidth expansion, since the coded systems have a spectral efficiency of $\eta = 1$ bit/symbol $= 1/2$ bit/dimension, and the spectral efficiency of uncoded BPSK is $\eta = 1$ bit/dimension. Most of the comparisons with uncoded systems in the remainder of this chapter will involve TCM schemes that do not require bandwidth expansion; that is, they are bandwidth efficient.
- The design of good rate $R = 1/2$ codes for use with naturally mapped QPSK will be considered again in Section 18.4, when we take up the issue of rotationally invariant code designs.
- The QPSK signal set is equivalent to two independent uses of BPSK, denoted by $2 \times$ BPSK. This can be considered a simple form of multidimensional signaling, a subject that will be covered in Section 18.5.

EXAMPLE 18.3 Rate $R = 1/2$ Trellis-Coded 4-AM

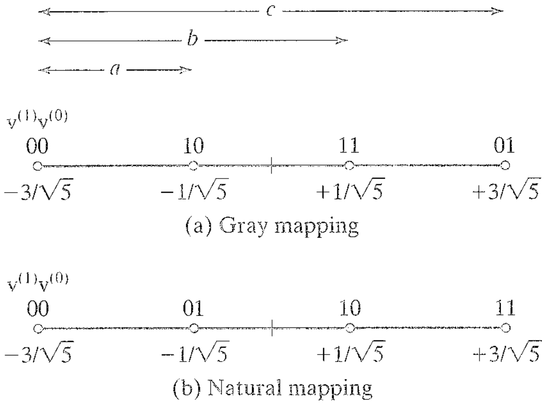
In this example we consider the same two rate $R = 1/2$ convolutional codes as in Example 18.2, but this time with the encoder output vector $\mathbf{v} = (v^{(1)}v^{(0)})$ mapped into the one-dimensional 4-AM signal set. Both Gray mapping and natural mapping of the 4-AM signal set are illustrated in Figure 18.5, where the signal amplitudes are assigned in such a way that the average signal energy $E_s = 1$. Using the signal point labeled $\mathbf{v} = (00)$ as a reference, we see that there are three distinct SE distances between the 4-AM signal points:

$$a^2 = [(-1/\sqrt{5}) - (-3/\sqrt{5})]^2 = [2/\sqrt{5}]^2 = 0.8, \tag{18.15a}$$

$$b^2 = [(1/\sqrt{5}) - (-3/\sqrt{5})]^2 = [4/\sqrt{5}]^2 = 3.2, \tag{18.15b}$$

$$c^2 = [(3/\sqrt{5}) - (-3/\sqrt{5})]^2 = [6/\sqrt{5}]^2 = 7.2. \tag{18.15c}$$

Clearly, the MSE distance between signal points in this case is $\Delta^2_{min} = 0.8$.



(a) Gray mapping

(b) Natural mapping

FIGURE 18.5: Gray and natural mapping of the 4-AM signal set.

TABLE 18.4: Euclidean distance structure for Gray- and naturally mapped 4-AM.

	(a) Gray mapping		(b) Natural mapping	
$\mathbb{e}$	$\Delta^2_{\mathbb{e}}(X)$	$\delta^2_{\mathbb{e}}(X)$	$\Delta^2_{\mathbb{e}}(X)$	$\delta^2_{\mathbb{e}}(X)$
00	X^0	X^0	X^0	X^0
01	$\frac{1}{2}X^{0.8}+\frac{1}{2}X^{7.2}$	$X^{0.8}$	$X^{0.8}$	$X^{0.8}$
10	$X^{0.8}$	$X^{0.8}$	$X^{3.2}$	$X^{3.2}$
11	$X^{3.2}$	$X^{3.2}$	$\frac{1}{2}X^{0.8}+\frac{1}{2}X^{7.2}$	$X^{0.8}$

In Table 18.4 we list, for both Gray and natural mapping of 4-AM, the four possible error vectors $\mathbb{e}$ and the four corresponding AEWEs $\Delta^2_{\mathbb{e}}(X)$ and MEWEs $\delta^2_{\mathbb{e}}(X)$. In Problem 18.6 it is shown that $\Delta^2_{\mathbb{e},0}(X) = \Delta^2_{\mathbb{e},1}(X)$ in both cases, and thus the mappings are uniform. We note that in each case, however, there is exactly one error vector $\mathbb{e}$ for which $\Delta^2_{\mathbb{e}}(X)$ is not a monomial, and thus the mappings are not regular.

If we now replace the binary labels on the error trellises shown in Figure 18.4(b) with the MEWEs of Gray- and naturally mapped 4-AM from Table 18.4, we obtain the modified error trellises of Figures 18.4(d) and 18.4(e), respectively. Examining the modified error trellises for the minimum-weight error events, we see that for Gray-mapped 4-AM (Figure 18.4(d)) $d^2_{free} = 7.2$ for code 1, resulting in a coding gain of $\gamma = 10\log_{10}(d^2_{free}/d^2_{min}) = 2.55$ dB compared with uncoded 2-AM with unit energy signals and $d^2_{min} = 4$, whereas code 2 achieves only $d^2_{free} = 2.4$, resulting in a *coding loss* of $\gamma = -2.22$ dB. Thus, code 1 is clearly the better choice for Gray-mapped 4-AM. For 4-AM with natural mapping (Figure 18.4(e)), the situation is exactly reversed, and the best choice is code 2, which results in a coding gain of $\gamma = 2.55$ dB compared with uncoded 2-AM.

The following observations relate to Example 18.3:

- In both cases, the mappings are *nonregular*; that is, for some error vectors $\mathbf{e}$, the MEWE does not equal the AEWE. This implies that the weight enumerating function $A(X)$ of the TCM system changes depending on the transmitted sequence; however, since $\Delta^2_{\mathbf{e},0}(X) = \Delta^2_{\mathbf{e},1}(X)$ for all $\mathbf{e}$ in both cases, the mappings are *uniform*, and the MFSE distance d^2_{free} can be computed by replacing the labels $\mathbf{e}$ in the binary error trellis with their corresponding MEWEs $\delta^2_{\mathbf{e}}(X)$ and using the transfer function method.

- By definition, all regular mappings must be uniform, but the reverse is not true.

- As in Example 18.2, the critical step in designing the best codes for both mappings was to assign the error vector with maximum Euclidean weight to the branches in the trellis that diverge from and remerge with the state S_0.

- In Example 18.3, unlike in Example 18.2, coding gain is achieved without bandwidth expansion, since the coded signal set, 4-AM, has the same dimensionality as the uncoded signal set, 2-AM. This explains the somewhat smaller coding gain, 2.55 dB versus 3.98 dB, achieved in Example 18.3 compared with Example 18.2.

EXAMPLE 18.4 Rate $R = 2/3$ Trellis-Coded 8-PSK

Now, consider a rate $R = 2/3$ convolutional code with 8-PSK modulation in which we denote the three encoder output bits by the vector $\mathbf{v} = (v^{(2)}v^{(1)}v^{(0)})$. In Figure 18.6 these three bits are shown mapped into the 8-PSK signal set according to the natural mapping rule. Each signal is again assumed to have unit energy, but in this case the MSE distance between signal points, computed in (18.10a), is $\Delta^2_{min} = 0.586$. Thus, compared with the QPSK signal set with the same average energy, the MSE distance of 8-PSK is reduced from 2.0 to 0.586.

In Table 18.5 we list the eight possible error vectors $\mathbf{e}$ and the eight corresponding AEWEs, $\Delta^2_{\mathbf{e},0}(X)$, $\Delta^2_{\mathbf{e},1}(X)$, and $\Delta^2_{\mathbf{e}}(X)$, and MEWEs, $\delta^2_{\mathbf{e},0}(X)$, $\delta^2_{\mathbf{e},1}(X)$, and

TABLE 18.5: Euclidean distance structure for naturally mapped 8-PSK.

$\mathbf{e}$	$\Delta^2_{\mathbf{e},0}(X)$	$\Delta^2_{\mathbf{e},1}(X)$	$\Delta^2_{\mathbf{e}}(X)$	$\delta^2_{\mathbf{e},0}(X)$	$\delta^2_{\mathbf{e},1}(X)$	$\delta^2_{\mathbf{e}}(X)$
000	X^0	X^0	X^0	X^0	X^0	X^0
001	$X^{0.586}$	$X^{0.586}$	$X^{0.586}$	$X^{0.586}$	$X^{0.586}$	$X^{0.586}$
010	X^2	X^2	X^2	X^2	X^2	X^2
011	$\frac{1}{2}X^{0.586} + \frac{1}{2}X^{3.414}$	$\frac{1}{2}X^{0.586} + \frac{1}{2}X^{3.414}$	$\frac{1}{2}X^{0.586} + \frac{1}{2}X^{3.414}$	$X^{0.586}$	$X^{0.586}$	$X^{0.586}$
100	X^4	X^4	X^4	X^4	X^4	X^4
101	$X^{3.414}$	$X^{3.414}$	$X^{3.414}$	$X^{3.414}$	$X^{3.414}$	$X^{3.414}$
110	X^2	X^2	X^2	X^2	X^2	X^2
111	$\frac{1}{2}X^{0.586} + \frac{1}{2}X^{3.414}$	$\frac{1}{2}X^{0.586} + \frac{1}{2}X^{3.414}$	$\frac{1}{2}X^{0.586} + \frac{1}{2}X^{3.414}$	$X^{0.586}$	$X^{0.586}$	$X^{0.586}$

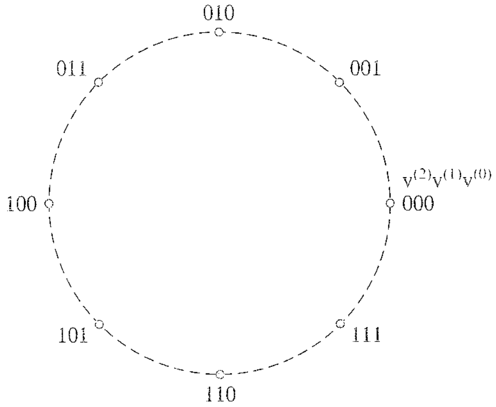


FIGURE 18.6: Natural mapping rule for the 8-PSK signal set.

$\delta_{\mathbf{e}}^2(X)$, for naturally mapped 8-PSK. In this case we see that natural mapping of 8-PSK is *uniform*. (In Problem 18.8 it is shown that Gray mapping of 8-PSK is not uniform.) Unlike the uniform 8-PSK mapping shown in Figure 18.2(a), however, natural mapping has only two error vectors that result in different Euclidean distances; that is, for natural mapping the error vectors $\mathbf{e} = (011)$ and $\mathbf{e} = (111)$ result in $\Delta_{\mathbf{v}}^2(\mathbf{e}) = 0.586$ or 3.414, whereas the other six error vectors correspond to only a single Euclidean distance.

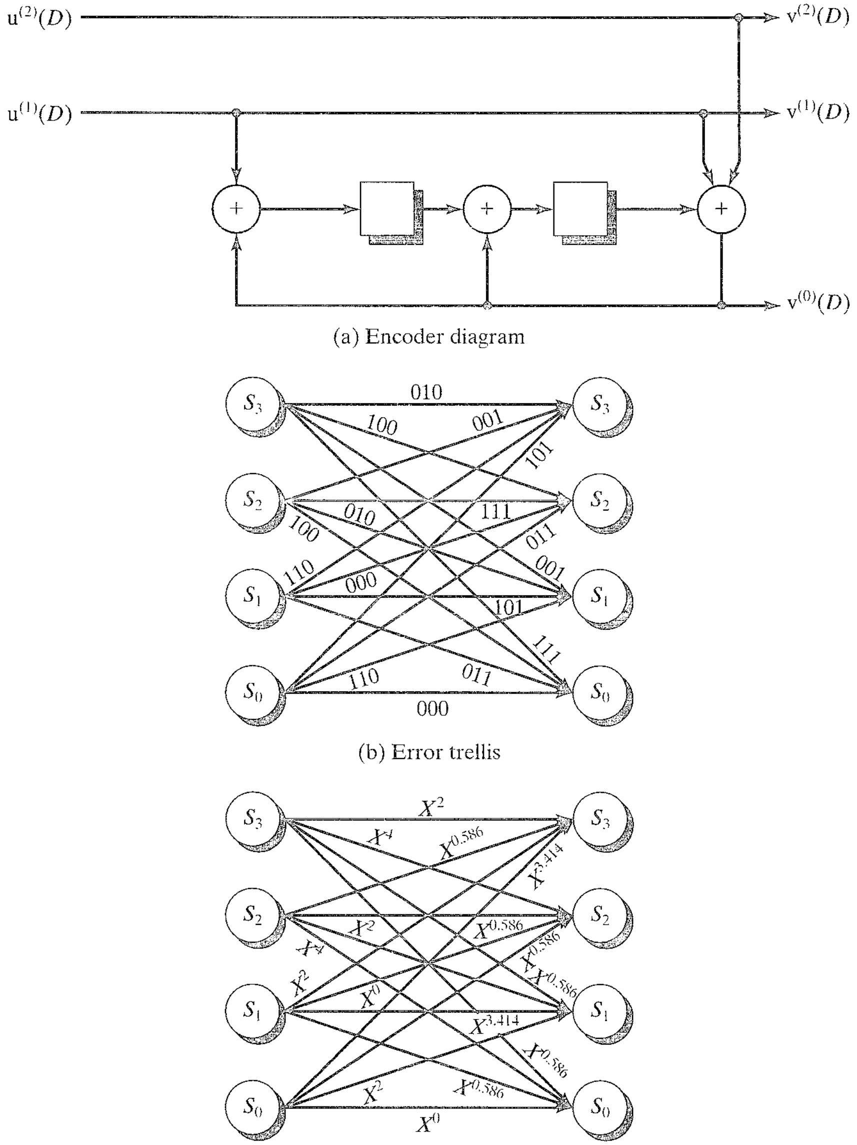
We now consider several possible code designs for naturally mapped 8-PSK and evaluate their MFSE distances d_{free}^2 using the error trellis labeled with MEWEs. We begin with a rate $R = 2/3$ convolutional code whose parity-check matrix in systematic feedback form is given by

$$\mathbb{H}(D) = \left[\begin{array}{ccc} 1/(D^2 + D + 1) & (D^2 + 1)/(D^2 + D + 1) & 1 \end{array} \right]. \tag{18.16}$$

This is the optimum free distance $(3, 2, 1)$ code with constraint length $\nu = 2$ and $d_{H,free} = 3$. The encoder diagram is shown in Figure 18.7(a), the $2^\nu = 4$-state binary error trellis is given in Figure 18.7(b), and the modified error trellis labeled with the MEWEs for naturally mapped 8-PSK is shown in Figure 18.7(c).

From Figure 18.7(c) we see that the nonzero path associated with the sequence of states $S_0S_2S_3S_0$ results in an MFSE distance of $d_{free}^2 = 1.758$. Because this TCM scheme has a spectral efficiency of $\eta = 2$ bits/symbol, the appropriate uncoded system with which to compare is QPSK with an average signal energy $E_s = 1$. For this signal set, $d_{min}^2 = 2.0$, and thus naturally mapped TCM suffers a *coding loss* of $\gamma = 10\log_{10}(d_{free}^2/d_{min}^2) = 10\log_{10}(1.758/2) = -0.56$ dB in this case!

Now, we ask the question, Is it possible to achieve a positive coding gain without bandwidth expansion with 4-state, rate $R = 2/3$ coded, naturally mapped 8-PSK? Because the naturally mapped 8-PSK signal set is nonregular, we may find a better TCM scheme by considering suboptimum rate $R = 2/3$ codes. In addition, we may consider a rate $R = 1/2$ code with one uncoded information bit as equivalent to a rate $R = 2/3$ code; that is, both have a spectral efficiency of $\eta = 2$ bits/symbol when combined with 8-PSK modulation. To illustrate this latter approach, we consider



(a) Encoder diagram

(b) Error trellis

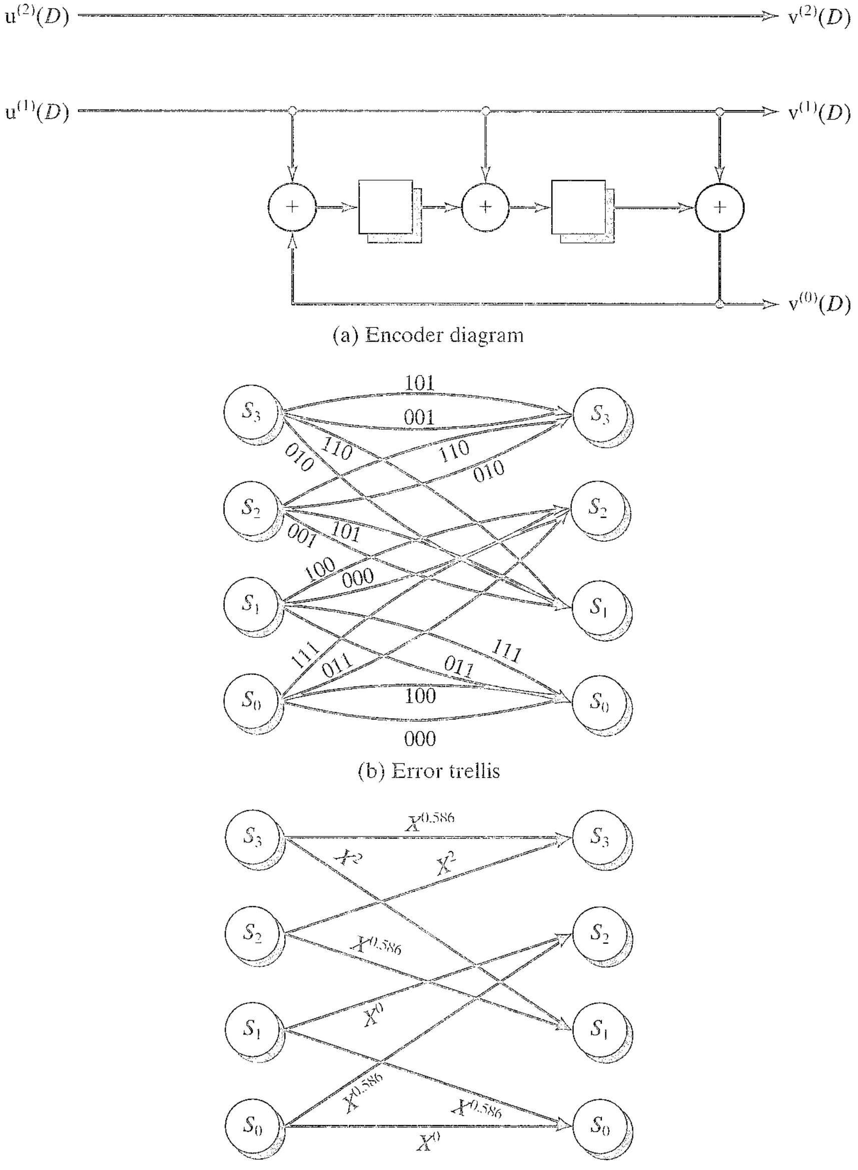
(c) Modified error trellis

FIGURE 18.7: Encoder diagram and error trellises for rate $R = 2/3$ coded 8-PSK.

the same two (2, 1, 2) codes as in Example 18.2, although this time we include an uncoded information bit and use the systematic feedback form of the encoders. Thus, the two rate $R = 1/2$ generator matrices are given by

$$\text{Code 1} : \mathbb{G}_1(D) = \left[\begin{array}{cc} 1 & (D^2 + D + 1)/(D^2 + 1) \end{array}\right], \tag{18.17a}$$

$$\text{Code 2} : \mathbb{G}_2(D) = \left[\begin{array}{cc} 1 & D/(D^2 + 1) \end{array}\right]. \tag{18.17b}$$



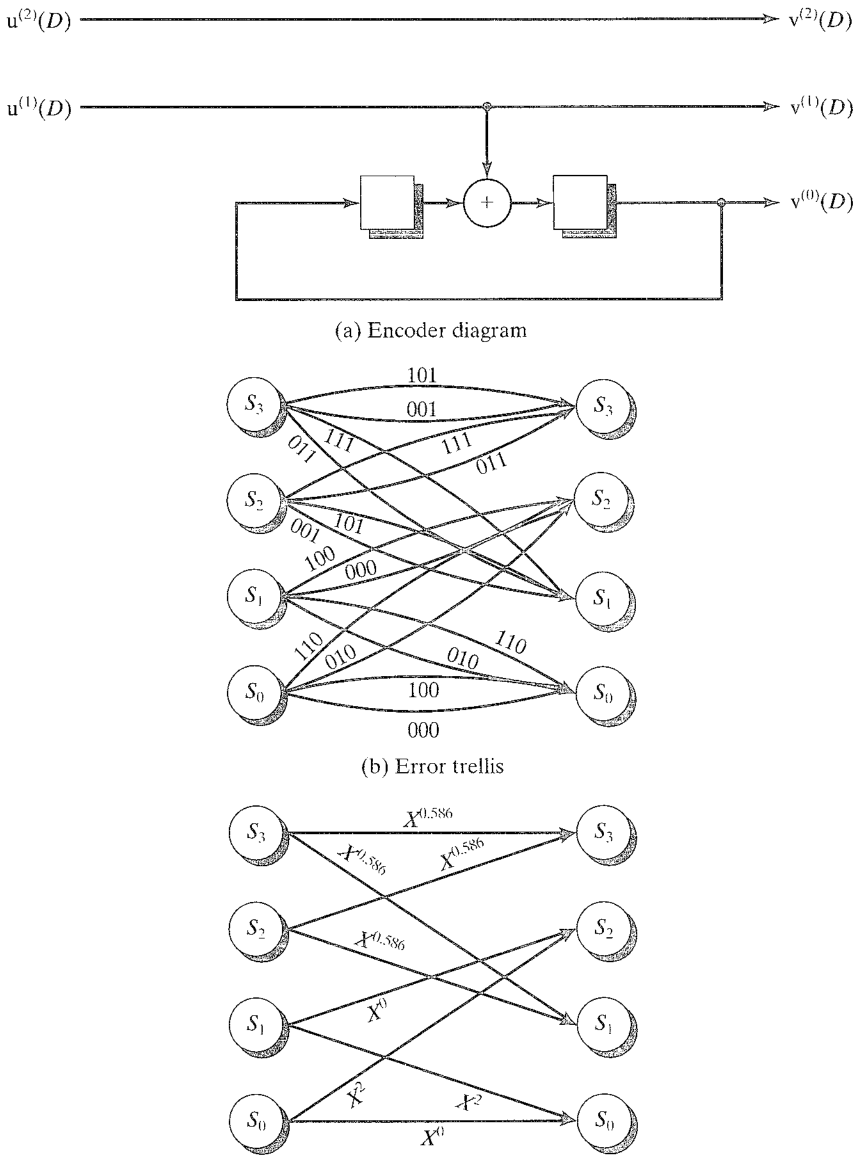
(a) Encoder diagram

(b) Error trellis

(c) Modified error trellis

FIGURE 18.8: Encoder diagram and error trellises for rate $R = 1/2$ coded 8-PSK (code 1).

The encoder diagrams for these two codes are shown in Figures 18.8(a) and 18.9(a), their binary error trellises are given in Figures 18.8(b) and 18.9(b), and the modified error trellises labeled with MEWEs for naturally mapped 8-PSK are shown in Figures 18.8(c) and 18.9(c), respectively. The uncoded information bit is handled by adding a *parallel transition* to each branch in the binary trellis of the rate $R = 1/2$



(a) Encoder diagram

(b) Error trellis

(c) Modified error trellis

FIGURE 18.9: Encoder diagram and error trellises for rate $R = 1/2$ coded 8-PSK (code 2).

code. Thus, there are two branches connecting each pair of states in the binary error trellis, one for each of the two possible values of the uncoded bit. We follow the convention that the first bit listed on each branch of the binary error trellis is the uncoded bit. In the modified error trellis, we show only one branch connecting each pair of states; that is, it has the same structure as the trellis for the rate $R = 1/2$ code, but its label is the minimum-weight label of the two MEWEs for the corresponding parallel branches in the binary error trellis. For example, in Figure 18.8(b), the two parallel branches connecting state S_0 to itself in the binary error trellis are labeled (000) and (100). Thus, in Figure 18.8(c), the two corresponding MEWEs are X^0 and X^4, and the single branch connecting state S_0 to itself is labeled X^0.

For any TCM scheme with parallel transitions, the calculation of the MFSE distance d^2_{free} involves two terms: (1) the MFSE distance δ^2_{free} between distinct trellis paths longer than one branch and (2) the MSE distance δ^2_{min} between distinct trellis paths one branch in length. Because δ^2_{free} is the free distance between trellis paths associated with the coded bits, it can be computed from the error trellis labeled with the MEWEs. Because δ^2_{min}, on the other hand, is the minimum distance between parallel transitions associated with the uncoded bits, it must be computed separately. Then, the overall MFSE distance is given by

$$d^2_{free} = \min\{\delta^2_{free}, \delta^2_{min}\}. \tag{18.18}$$

EXAMPLE 18.4 (Continued)

The parallel transition distance δ^2_{min} is independent of the code and depends only on the mapping used. From Figures 18.8(b) and 18.9(b) it is clear that the parallel branch labels always differ by the error vector (100). Thus, from Table 18.5 we conclude that $\delta^2_{min} = 4.0$. Now, we can see from Figures 18.8(c) and 18.9(c) that $\delta^2_{free} = 3(0.586) = 1.758$ for code 1, and $\delta^2_{free} = 2(2.0) + 0.586 = 4.586$ for code 2. Thus,

$$d^2_{free} = \min\{\delta^2_{free}, \delta^2_{min}\} = \min\{1.758, 4.0\} = 1.758, \quad \text{(code 1)} \tag{18.19a}$$

and

$$d^2_{free} = \min\{\delta^2_{free}, \delta^2_{min}\} = \min\{4.586, 4.0\} = 4.0, \quad \text{(code 2)} \tag{18.19b}$$

and the asymptotic coding losses (gains) compared with uncoded QPSK are $\gamma = -0.56$ dB for code 1 and $\gamma = +3.01$ dB for code 2. Thus, for the three different codes considered in this example, the best performance, and the only coding gain, is achieved by the suboptimum (in terms of $d_{H,free}$) rate $R = 1/2$ code with one uncoded bit. This simple 4-state code achieves a 3.01-dB coding gain compared with uncoded QPSK without bandwidth expansion. (Problem 18.9 illustrates that other mapping rules for 8-PSK result in less coding gain than natural mapping.)

The following remarks relate to Example 18.4:

- All mappings for the 8-PSK signal set are *nonregular*. Thus, the weight enumerating function $A(X)$ depends on the transmitted code sequence for all 8-PSK–based TCM systems; however, if the mapping is *uniform*, the average weight enumerating function $A_{av}(X)$ can be computed by labeling the branches of the error trellis with the AEWEs and using the transfer function method.

- Virtually all signal sets and mappings used in practical TCM systems are nonregular, although symmetries usually exist that allow a uniform mapping.

- The MEWEs can be used to compute the MFSE distance d^2_{free} of TCM systems with uniform mappings, as shown in Examples 18.2, 18.3, and 18.4; however, to determine the average weight enumerating function $A_{av}(X)$, the AEWEs must be used, as will be illustrated in Section 18.3.

- The critical advantage of naturally mapped 8-PSK over other uniform mappings for 8-PSK is that the error vector for all parallel transitions, $\mathbb{e} = (100)$, is assigned to the largest possible EW, $\Delta^2(\mathbb{e}) = 4.0$, by the natural mapping rule (see Problem 18.9). In other words, for 8-PSK, the MSE distance between signal points on parallel transition paths is maximized by natural mapping, thus minimizing the probability of a one-branch (parallel transition) error event.

- An exhaustive search of all possible 8-PSK TCM schemes with $\eta = 2$ bits/symbol and 4 states indicates that the best scheme is code 2 in Example 18.4, that is, the suboptimum rate $R = 1/2$ code with one uncoded bit, combined with natural mapping. This illustrates that, unlike code designs for binary modulation, the best TCM designs often include uncoded information bits resulting in parallel transitions in the trellis. (If uncoded bits are employed in the design of codes for binary modulation, the minimum free Hamming distance can never exceed the minimum Hamming distance between the parallel transition branches, which equals 1.)

- All the encoders in Example 18.4 were given in systematic feedback form. Equivalent nonsystematic feedforward encoders exist that give slightly different BER performance because of the different (encoder) mapping between information bits and code bits. Systematic feedback encoders are usually preferred in TCM system design because they represent a convenient canonical form for representing minimal rate $R = k/(k+1)$ encoders in terms of a single parity-check equation. This canonical representation simplifies the search for the best encoders.

- Larger coding gains can be achieved by employing more powerful codes, that is, longer constraint lengths. Tables of the best TCM code designs for a number of important signal constellations are given in Section 18.2.

The rate $R = k/(k+1)$ code lemma guarantees that if the mapping is uniform, any error sequence $\mathbb{e}(D)$ in the binary error trellis with a given Euclidean weight $\Delta^2[\mathbb{e}(D)]$ corresponds to a pair of signal sequences $\mathbb{y}(D)$ and $\mathbb{y}'(D)$ in the trellis separated by a free squared Euclidean distance of $\Delta^2[\mathbb{e}(D)]$. In this case, the MFSE distance d^2_{free} of a TCM system can be computed using the method of Euclidean weights; however, if the mapping is not uniform, the rate $R = k/(k+1)$ code lemma does not hold, and the method of Euclidean weights will, in general, give only a lower bound on the actual d^2_{free}. This point is illustrated in the following example.

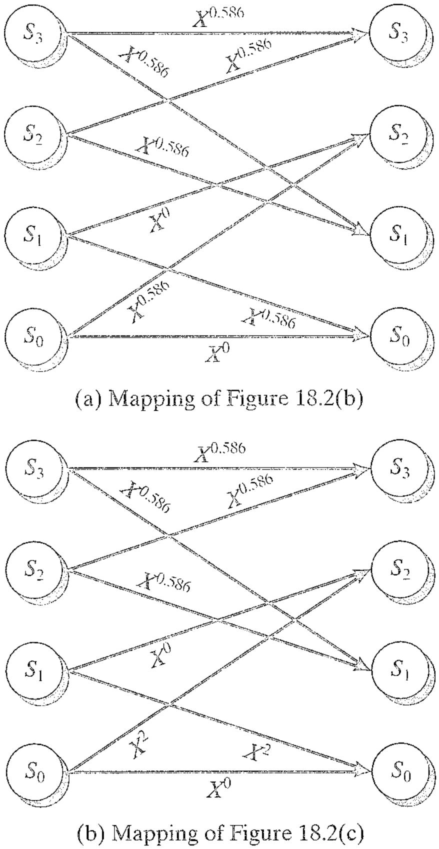
EXAMPLE 18.5 Nonuniform Mappings

Consider the two nonuniform mappings of 8-PSK shown in Figures 18.2(b) and (c), along with their AEWEs and MEWEs listed in Tables 18.2(b) and (c). If these mappings are used along with code 2 from Example 18.4, whose encoder diagram and binary error trellis are shown in Figures 18.9(a) and (b), respectively, we obtain the modified error trellises shown in Figure 18.10.

First, consider the nonuniform mapping of Figure 18.2(b) and Table 18.2(b), in which there is no isometry between the subsets $Q(0)$ and $Q(1)$. Let $\mathbf{e}(D) = \mathbf{e}_0 + \mathbf{e}_1 D + \mathbf{e}_2 D^2 + \mathbf{e}_3 D^3 = (110) + (011)D + (111)D^2 + (110)D^3$ be a path through the binary error trellis of Figure 18.9(b) that starts and ends in state S_0. From the modified error trellis of Figure 18.10(a), we can compute the EW of $\mathbf{e}(D)$ as follows:

$$\Delta^2[\mathbf{e}(D)] = 0.586 + 0.586 + 0.586 + 0.586 = 2.344. \tag{18.20}$$

For the rate $R = k/(k+1)$ code lemma to be satisfied, there must exist a pair of 4-branch trellis paths, $\mathbf{v}(D)$ and $\mathbf{v}'(D)$, starting and stopping in the same state, that differ by the error path $\mathbf{e}(D)$ and whose corresponding signal sequences $\mathbf{y}(D)$ and $\mathbf{y}'(D)$ are distance 2.344 apart. From Figure 18.2(b) and Table 18.2(b) we see that the desired path pair must start with the branches $\mathbf{v}_0 = (101)$ and $\mathbf{v}'_0 = (011)$, since this is the only pair of binary labels such that $\mathbf{e}_0 = \mathbf{v}_0 \oplus \mathbf{v}'_0 = (110)$, and



(a) Mapping of Figure 18.2(b)

(b) Mapping of Figure 18.2(c)

FIGURE 18.10: Modified error trellises for two nonuniform mappings.

$d_E^2[f(\mathbf{v}_0), f(\mathbf{v}_0')] = d_E^2[\mathbf{y}_0, \mathbf{y}_0'] = 0.586$. (The branches assigned to $\mathbf{v}_0$ and $\mathbf{v}_0'$ can also be reversed without changing the result.) Thus, from Figure 18.9(b), the path pair must start either from state S_2 or from state S_3. Similarly, the next three pairs of branch labels must be $\mathbf{v}_1 = (001)$ and $\mathbf{v}_1' = (010)$, $\mathbf{v}_2 = (111)$ and $\mathbf{v}_2' = (000)$, and $\mathbf{v}_3 = (101)$ and $\mathbf{v}_3' = (011)$ (or the reverse of these labels) to satisfy the distance conditions; but a close examination of Figure 18.9(b) reveals that no pair of paths with these labels and starting either from state S_2 or from state S_3 exists in the trellis. Thus, it is impossible to find a pair of paths $\mathbf{v}(D)$ and $\mathbf{v}'(D)$, starting and stopping in the same state, that differ by the error path $\mathbf{e}(D)$ and whose corresponding signal sequences $\mathbf{y}(D)$ and $\mathbf{y}'(D)$ are distance 2.344 apart; hence, the rate $R = k/(k+1)$ code lemma is not satisfied.

Next, consider the nonuniform mapping of Figure 18.2(c) and Table 18.2(c), in which there is an isometry between the subsets $Q(0)$ and $Q(1)$. Let $\mathbf{e}(D) = \mathbf{e}_0 + \mathbf{e}_1 D + \mathbf{e}_2 D^2 + \mathbf{e}_3 D^3 + \mathbf{e}_4 D^4 = (110) + (101)D + (100)D^2 + (101)D^3 + (110)D^4$ be a path through the binary error trellis of Figure 18.9(b) that starts and ends in state S_0. From the modified error trellis of Figure 18.10(b) we can compute the EW of $\mathbf{e}(D)$ as follows:

$$\Delta^2[\mathbf{e}(D)] = 2.0 + 0.586 + 2.0 + 0.586 + 2.0 = 7.172. \tag{18.21}$$

For the rate $R = k/(k+1)$ code lemma to be satisfied, there must exist a pair of 5-branch trellis paths, $\mathbf{v}(D)$ and $\mathbf{v}'(D)$, starting and stopping in the same state, that differ by the error path $\mathbf{e}(D)$ and whose corresponding signal sequences $\mathbf{y}(D)$ and $\mathbf{y}'(D)$ are distance 7.172 apart. From Figure 18.2(c) and Table 18.2(c) we see that the desired path pair must start either with the branch pair $\mathbf{v}_0 = (000)$ and $\mathbf{v}_0' = (110)$ or with the branch pair $\mathbf{v}_0 = (100)$ and $\mathbf{v}_0' = (010)$, since these are the only pairs of binary labels such that $\mathbf{e}_0 = \mathbf{v}_0 \oplus \mathbf{v}_0' = (110)$, and $d_E^2[f(\mathbf{v}_0), f(\mathbf{v}_0')] = d_E^2[\mathbf{y}_0, \mathbf{y}_0'] = 2.0$. From Figure 18.9(b) we see that the path pair must start either from state S_0 or from state S_1 in both cases. As in the previous case considered, the next four pairs of branch labels are similarly constrained to satisfy the distance conditions. It is easily seen that there is only one possible branch pair corresponding to the error vector $\mathbf{e}_1 = \mathbf{e}_3 = (101)$, but there are two possible branch pairs corresponding to the error vectors $\mathbf{e}_2 = (100)$ and $\mathbf{e}_4 = (110)$. Again, a close examination of Figure 18.9(b) reveals that no pair of paths with these labels and starting either from state S_0 or from state S_1 exists in the trellis. Thus, it is impossible to find a pair of paths $\mathbf{v}(D)$ and $\mathbf{v}'(D)$, starting and stopping in the same state, that differ by the error path $\mathbf{e}(D)$ and whose corresponding signal sequences $\mathbf{y}(D)$ and $\mathbf{y}'(D)$ are distance 7.172 apart; hence, the rate $R = k/(k+1)$ code lemma is again not satisfied.

Example 18.5 leads to the following observations:

- When the mapping is nonuniform, there are still many error sequences for which the rate $R = k/(k+1)$ code lemma is satisfied; however, Example 18.5 illustrates that this is not true for all error sequences.

- Example 18.5 shows that an isometry between the subsets $Q(0)$ and $Q(1)$ is necessary, but not sufficient, to guarantee that the rate $R = k/(k+1)$ code lemma is satisfied. See Problem 18.10 for an example illustrating this fact that uses a different signal constellation.

- Because the rate $R = k/(k+1)$ code lemma is not satisfied for nonuniform mappings, the method of Euclidean weights provides only a lower bound on d^2_{free} in this case. This is also true of the method to be presented in Section 18.3 for determining the AWEF $A_{av}(X)$ of a TCM system from the AEWEs.

- Using a nonuniform mapping does not necessarily imply an inferior TCM system, just one that is more difficult to analyze. In this case, a supertrellis of $(2^\nu)^2 = 2^{2\nu}$ states must be used to determine the set of distances between all possible path pairs; however, uniform mappings result in the best designs for most practical TCM systems (see Problem 18.11).

- A more stringent uniformity condition, called *geometric uniformity*, was introduced by Forney [23]. When this condition is satisfied, the computation of weight enumerating functions is simplified, but many practical TCM systems are not geometrically uniform.

Examples 18.2, 18.3, and 18.4 illustrate two basic rules of good TCM system design:

Rule 1: Signal set mapping should be designed so that the MSE distance between parallel transition branches is maximized.

Rule 2: The convolutional code should be designed so that the branches in the modified error trellis leaving and entering the same state have the largest possible MSE distance.

A general block diagram of a TCM system is shown in Figure 18.11. At each time unit l, a total of k information bits, $\mathbf{u}_l = \left(u_l^{(k)}, u_l^{(k-1)}, \cdots, u_l^{(1)}\right)$, enter the

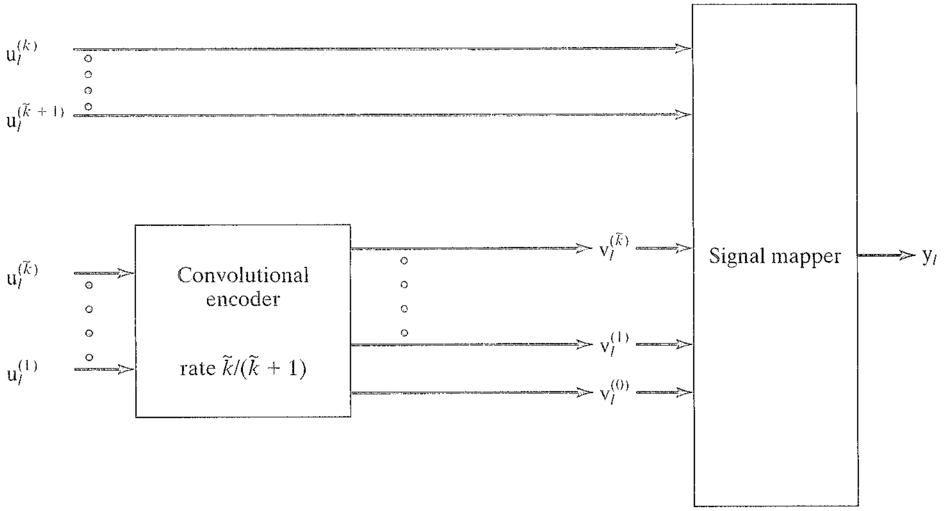


FIGURE 18.11: General TCM encoder diagram and signal mapper.

system. Of these, a total of $\tilde{k} \leq k$ bits, namely, $u_l^{(\tilde{k})}, u_l^{(\tilde{k}-1)}, \cdots, u_l^{(1)}$, enter a rate $R = \tilde{k}/(\tilde{k}+1)$ systematic feedback convolutional encoder, producing the output bits $v_l^{(\tilde{k})}, v_l^{(\tilde{k}-1)}, \cdots, v_l^{(1)}, v_l^{(0)}$, where $v_l^{(0)}$ is the parity bit, and $v_l^{(\tilde{k})}, v_l^{(\tilde{k}-1)}, \cdots, v_l^{(1)}$ are information bits. These $\tilde{k}+1$ bits enter the signal mapper along with the $k - \tilde{k}$ uncoded information bits $u_l^{(k)} = v_l^{(k)}, u_l^{(k-1)} = v_l^{(k-1)}, \cdots, u_l^{(\tilde{k}+1)} = v_l^{(\tilde{k}+1)}$. Finally, the $k+1$ bit vector $\mathbf{v}_l = (v_l^{(k)}, v_l^{(k-1)}, \cdots, v_l^{(1)}, v_l^{(0)})$ is mapped into one of the $M = 2^{k+1}$ possible points in the signal set S. If $k = \tilde{k}$, then there are no uncoded information bits and no parallel transitions in the trellis diagram.

In the next section we will study a technique called *mapping by set partitioning* [1] in which the $\tilde{k}+1$ coded bits $v_l^{(\tilde{k})}, v_l^{(\tilde{k}-1)}, \cdots, v_l^{(1)}, v_l^{(0)}$ are used to select a subset of size $2^{k-\tilde{k}}$ from the signal set S, and then the $k-\tilde{k}$ uncoded bits $v_l^{(k)}, v_l^{(k-1)}, \cdots, v_l^{(\tilde{k}+1)}$ are used to choose a particular signal point from within the selected subset. Thus, a path through the trellis indicates the particular sequence of selected subsets, and the $2^{k-\tilde{k}}$ parallel transitions associated with each trellis branch indicate the choice of signal points within the corresponding subset. This mapping technique allows us to design TCM systems that satisfy the two basic design rules noted.

18.2 TCM CODE CONSTRUCTION

There are three basic steps in designing a TCM system:

1. Signal set selection
2. Labeling of the signal set
3. Code selection

A signal set is chosen primarily to satisfy system constraints on spectral efficiency and modulator design. For example, if a spectral efficiency of $\eta = k$ bits/symbol is desired, a signal set with 2^{k+1} points must be selected. Similarly, if, because of nonlinearities in the transmission path (e.g., a traveling wave tube amplifier), a constant-amplitude signaling scheme is required, then a PSK signal set must be chosen. If amplitude modulation can be accommodated, then a rectangular or QAM signal set will give better performance. Several typical signal sets were shown in Figure 18.1. As an example, consider a linear transmission path and a spectral efficiency requirement of $\eta = 4$ bits/symbol, the specifications for the CCITT V.32 modem standard that can achieve data rates up to 14.4 Kbps over voice-grade telephone lines. In this case, the 32-CROSS signal set was chosen for implementation.

The next step in the design process is to assign binary labels, representing encoder output blocks, to the signal points in such a way that the MFSE distance d_{free}^2 of the overall TCM system is maximized. These labels are assigned by using a technique called *mapping by set partitioning* [1]. This technique successively partitions the signal set into smaller subsets of equal size, thereby generating a tree structure in which each signal point corresponds to a unique path through the tree. If binary partitioning is used, that is, at every level in the partitioning tree each subset from the previous level is divided into two subsets of equal size, the tree has

$k+1$ levels. Thus, each path through the tree can be represented by a $(k+1)$-bit label, which can then be assigned to the corresponding signal point. To maximize d^2_{free}, the partitioning must be done in such a way that the two basic rules for good TCM system design discussed in Section 18.1 are satisfied. This requires that the *minimum squared subset distance* (MSSD) Δ^2_p, that is, the MSE distance between signal points within the same subset, be maximized at each level p of the partition tree. The approach is illustrated with two examples.

EXAMPLE 18.6 Partitioning of 8-PSK

Consider the binary partitioning tree for the 8-PSK signal set S shown in Figure 18.12. Level 0 of the partitioning tree contains the full 8-PSK signal set S. Assuming unit energy signals, the MSSD at level 0 was computed in (18.10a) and is denoted by $\Delta^2_0 = 0.586$. (Δ^2_0 is the same as the previously defined Δ^2_{min}, the MSE distance between signal points. The notation Δ^2_0 indicates that this term corresponds to the MSSD at level 0 in the set-partitioning tree.) Label bit $v^{(0)}$ then divides the set S into two subsets, $Q\left(v^{(0)}\right) = Q(0)$ and $Q(1)$, each containing four signal points such that the MSSD of both subsets at level 1 is given by $\Delta^2_1 = 2.0$. It is important to point out here two properties of this partition:

1. There is no partition of 8-PSK into two equal-size subsets that achieves a larger MSSD.
2. Subset $Q(0)$ is isomorphic to subset $Q(1)$ in the sense that $Q(1)$ can be obtained from $Q(0)$ by rotating the points in $Q(0)$ by 45°.

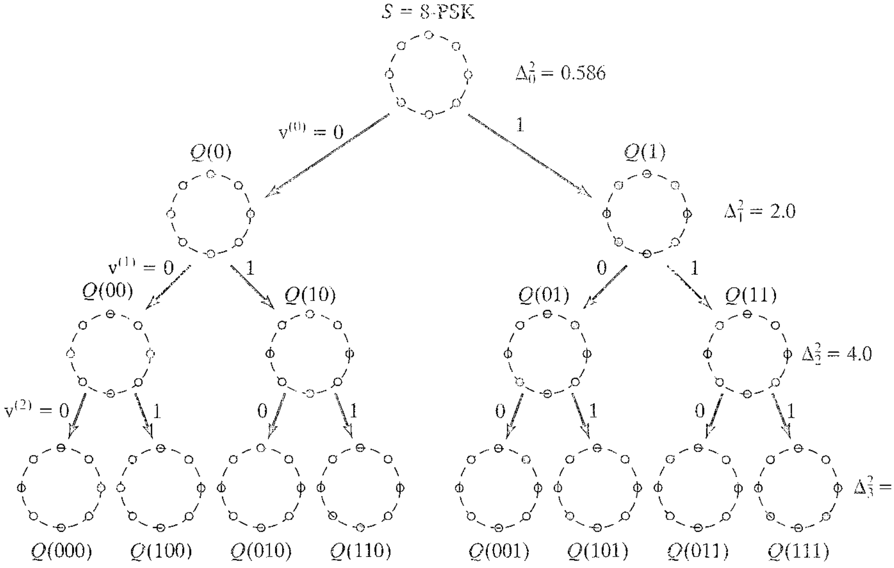


FIGURE 18.12: Partitioning of 8-PSK.

These two properties of 8-PSK partitioning, namely, maximizing the MSSD and maintaining an isometry among all subsets at the same level, are characteristic of most practical signal set partitionings. The isometry property implies that the MSSDs are the same for all subsets at a given level.

Continuing with the example, we see that label bit $v^{(1)}$ now divides each of the subsets $Q\,(0)$ and $Q\,(1)$ at level 1 into two subsets, containing two signal points each, such that $\Delta_2^2 = 4.0$ for each subset at level 2. We see again at level 2 that the subset distance has been increased and that the four subsets are isomorphic and thus have the same MSSD. The four subsets are denoted as $Q\left(v^{(1)}v^{(0)}\right) = Q\,(00)$, $Q\,(10)$, $Q\,(01)$, and $Q\,(11)$, representing the four possible values of the binary label $(v^{(1)}v^{(0)})$. Finally, label bit $v^{(2)}$ divides each of the subsets at level 2 into two subsets containing one signal point each at level 3. This is the lowest level in the partitioning tree, and the MSSD Δ_3^2 at this level is infinite, since there is only one signal point in each subset. The eight subsets at level 3, $Q\left(v^{(2)}v^{(1)}v^{(0)}\right) = Q\,(000)$, $Q\,(100)$, $Q\,(010)$, $Q\,(110)$, $Q\,(001)$, $Q\,(101)$, $Q\,(111)$, and $Q\,(011)$ are represented by a unique binary label $(v^{(2)}v^{(1)}v^{(0)})$ that corresponds to a path through the partitioning tree. This binary label then defines the mapping between a 3-bit encoder output block and a corresponding signal point in the 8-PSK signal set.

As noted in Section 18.1, a TCM system using 8-PSK can employ either a rate $R = 2/3$ code or a rate $R = 1/2$ code with one uncoded bit. To best describe the code design procedure, we consider the case of a 4-state, rate $R = 1/2$ code with one uncoded bit in the remainder of this example. In this case only the first two levels of the partitioning tree are used, and each of the four subsets at level 2, that is, the subsets $Q\,(00)$, $Q\,(10)$, $Q\,(01)$, and $Q\,(11)$, contains two signal points separated by the distance $\Delta_2^2 = 4.0$. First, the two coded bits $(v^{(1)}v^{(0)})$ are used to select a subset, and the uncoded bit $v^{(2)}$ is then used to select the signal point to be transmitted. This means that each branch in the code trellis, which represents a parallel transition, is assigned one of the level-2 subsets $Q\,(00)$, $Q\,(10)$, $Q\,(01)$, or $Q\,(11)$ with subset distance $\Delta_2^2 = 4.0$. Note that, since Δ_2^2 was maximized by the partitioning procedure, this guarantees that the MSE distance δ_{min}^2 between parallel transition branches is maximized, thus satisfying rule 1 for good TCM code design.

Now, we consider the assignment of the level-2 subsets $Q\,(00)$, $Q\,(10)$, $Q\,(01)$, and $Q\,(11)$ to the branches of the code trellis. Note that the trellis is completely defined by the set of branches leaving each state. In this example there are a total of $2^{\tilde{k}} = 2$ branches leaving each of the $2^{\nu} = 4$ states. Because there are only four level-2 subsets from which to choose, exactly half of these subsets must be assigned to each set of two branches leaving a state. From Figure 18.12 we can see that the distance between diverging branches is maximized if the two branches leaving each state are assigned subsets belonging to the same level-1 subset, $Q\,(0)$ or $Q\,(1)$. In other words, the level-2 subsets $Q\,(00)$ and $Q\,(10)$ (belonging to $Q\,(0)$) should be paired, and the level-2 subsets $Q\,(01)$ and $Q\,(11)$ (belonging to $Q\,(1)$) should be paired. To ensure that the distance between remerging branches will also be maximized, the same level-2 subset pair (either $\{Q\,(00)\,,\, Q\,(10)\}$ or $\{Q\,(01)\,,\, Q\,(11)\}$ should be used to label the diverging branches of both states in each trellis "butterfly," and the level-2 subset pair should be assigned in such a way that the two remerging branches of

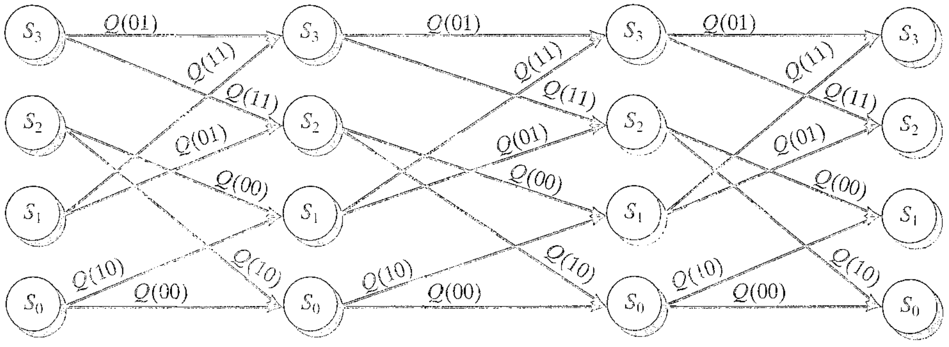


FIGURE 18.13: Branch labels for 4-state, rate $R = 1/2$ coded 8-PSK.

both states in the butterfly are labeled by the same pair (see Figure 18.13).[2] Finally, in order to ensure that all signal points are used equally often, subset $Q(0)$ (pair $\{Q(00), Q(10)\}$) should be assigned to half the states (one butterfly), and subset $Q(1)$ (pair $\{Q(01), Q(11)\}$) to the other half (the other butterfly). Because each of the level-1 subsets ($Q(0)$ and $Q(1)$) contains $2^{\tilde{k}} = 4$ signal points, and their MSSD $\Delta_1^2 = 2.0$ is the largest possible for a subset of four points, this guarantees that the MSE distance between branches leaving and entering the same state is maximized, thus satisfying rule 2 for good TCM system design. The final labeling of branches for this example is shown in Figure 18.13, where the trellis represents a 4-state, rate $R = 1/2$, feedforward encoder.

The following remarks relate to Example 18.6:

- The assignment of signal points from only one level-1 subset ($Q(0)$ or $Q(1)$) to all the branches leaving and entering each state implies that the code bit $v^{(0)}$, which determines the subset chosen at level 1, must be the same for each set of branches leaving or entering a particular state. This places some restrictions on the codes that yield good TCM designs.

- In general, half of the $2^{\nu-\tilde{k}}$ butterflies in the code trellis are assigned to subset $Q(0)$ and the other half to subset $Q(1)$. This ensures that all signal points are used with equal probability.

- It is always possible, in the manner described here, to ensure that the diverging and remerging branch distance equals Δ_1^2, thus guaranteeing that $\delta_{free}^2 \geq 2\Delta_1^2$, except in the special case $\nu = \tilde{k}$. In this case, the trellis is fully connected and contains only a single butterfly, thus implying that either the diverging or

[2] A trellis section of any $(n, \tilde{k}, \nu)$ encoder can be decomposed into a set of $2^{\nu-\tilde{k}}$ fully connected subtrellises containing $2^{\tilde{k}}$ states each. These subtrellises, called *butterflies*, connect a subset of $2^{\tilde{k}}$ states at one time to a (in general, different) subset of $2^{\tilde{k}}$ states at the next time. For example, in Figure 18.13, the pair $(2^{\tilde{k}} = 2)$ of states S_0 and S_2 connect to the state pair S_0 and S_1, forming one of the $2^{\nu-\tilde{k}} = 2$ butterflies, and the other butterfly is formed by the state pair S_1 and S_3 connecting to the state pair S_2 and S_3.

remerging distance must equal only Δ_0^2. Hence, 2-state trellises ($\nu = 1$) with rate $R = 1/2$ codes ($\tilde{k} = 1$) do not yield good TCM designs.

- If a rate $R = 2/3$ code is used in the preceding example, then $k = \tilde{k} = \nu$, and the trellis is fully connected. This implies that δ_{free}^2 is at most equal to $\Delta_0^2 + \Delta_1^2 = 2.586$, no matter which code is selected. Thus, for 4-state 8-PSK TCM schemes with $\eta = 2$ bits/symbol, rate $R = 2/3$ codes are suboptimal compared with rate $R = 1/2$ codes with one uncoded bit.

- In the partitioning of 8-PSK, the two subsets at level 1 are equivalent to QPSK signal sets, and the four subsets at level 2 are equivalent to BPSK signal sets. This isometry between subsets at the same level of the partition tree is characteristic of all PSK signal set partitionings.

- For the 8-PSK partition shown in Figure 18.12, mapping by set partitioning results in the natural mapping rule discussed in Section 18.1. If the order of the subsets at any level in the partitioning tree is changed, the resulting mapping is isomorphic to natural mapping.

- Mapping by set partitioning always results in the distance relation $\Delta_0^2 \leq \Delta_1^2 \leq \cdots \leq \Delta_{\tilde{k}}^2$, which, along with the proper assignment of subsets to trellis branches, guarantees that the two rules of good TCM system design are satisfied.

- The separate tasks assigned to coded and uncoded bits by set partitioning, namely, the selection of subset labels for the trellis branches and the selection of a signal point from a subset, respectively, imply that the general TCM encoder and mapper in Figure 18.11 can be redrawn as shown in Figure 18.14.

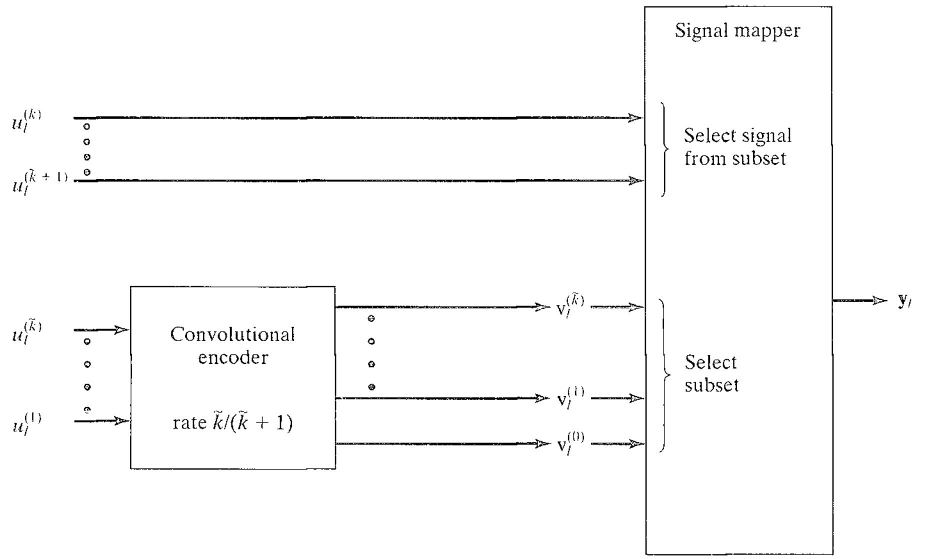


FIGURE 18.14: Set-partitioning TCM encoder diagram and signal mapper.

If $\tilde{k} = k$, then there are no parallel transitions, and the subset labels on the trellis become signal point labels.

EXAMPLE 18.7 Partitioning of 16-QAM

As a second example of set partitioning, we consider the 16-QAM signal set, denoted by S, shown in Figure 18.15. Letting Δ_0^2 represent the MSSD at level 0, we see that the average signal energy is given by the expression

$$\begin{aligned} E_s &= (1/16)\left\{4\left[(\Delta_0/2)^2 + (\Delta_0/2)^2\right] + 8\left[(3\Delta_0/2)^2 + (\Delta_0/2)^2\right]\right. \\ &\quad \left. + 4\left[(3\Delta_0/2)^2 + (3\Delta_0/2)^2\right]\right\} \\ &= (1/16)\left[2\Delta_0^2 + 20\Delta_0^2 + 18\Delta_0^2\right] = 5\Delta_0^2/2. \end{aligned} \tag{18.22}$$

Thus, $\Delta_0^2 = 2/5$ if the average energy $E_s = 1$. At level 1 of the partitioning tree, we obtain the subsets $Q(0)$ and $Q(1)$, each isomorphic to an 8-AM/PM constellation, and it is easy to see that $\Delta_1^2 = 2\Delta_0^2$. Continuing down the partitioning tree, we obtain four subsets at level 2, each isomorphic to 4-QAM, with $\Delta_2^2 = 2\Delta_1^2$; eight subsets at level 3, each isomorphic to 2-AM, with $\Delta_3^2 = 2\Delta_2^2$; and, finally, the 16 signal points at level 4, each labeled according to the set-partitioning mapping rule.

The following observations relate to Example 18.7:

- The 16-QAM signal set can be considered a multidimensional version of 4-AM, that is, 2 × 4-AM.
- In the 16-QAM case, the MSSD doubles at each level of the partitioning tree; that is, $\Delta_i^2 = 2\Delta_{i-1}^2$, $i = 1, 2, \cdots, k$. This is characteristic of most partitionings of rectangular-type signal constellations used in practice.
- 16-QAM is a (translated) subset of the two-dimensional integer lattice $\mathbb{Z}^2$, and the subsets at each level of the partitioning are isomorphic.
- It is not always possible to partition signal sets based on a lattice in such a way that all subsets at a given partition level are isomorphic. In this case, although the subsets are no longer distance invariant, they all still have the same MSSD Δ_i^2. An example of this situation is shown in Section 18.4 for the 32-CROSS constellation.
- TCM systems based on 16-QAM modulation can employ code rates R of 3/4 or 2/3 with one uncoded bit, or 1/2 with two uncoded bits.

We now consider the last step in the design process, that of code selection. Assume that the code is generated by a rate $R = \tilde{k}/(\tilde{k}+1)$ systematic feedback convolutional encoder with parity-check matrix

$$\mathbb{H}(D) = \left[\mathbb{h}^{(\tilde{k})}(D)/\mathbb{h}^{(0)}(D) \ \cdots \ \mathbb{h}^{(1)}(D)/\mathbb{h}^{(0)}(D) \quad 1\right], \tag{18.23}$$

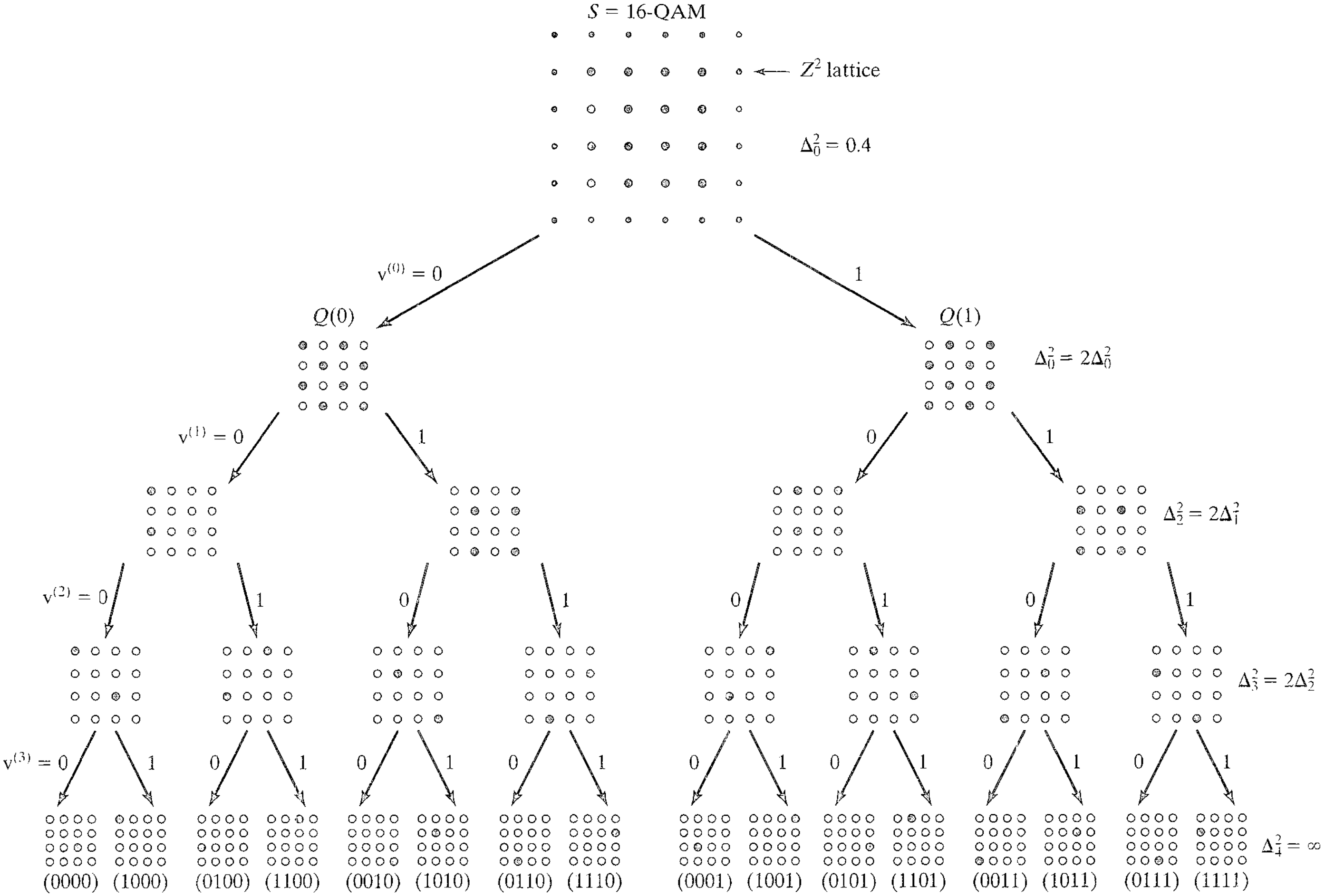


FIGURE 18.15: Partitioning of 16-QAM.

where $\mathbb{h}^{(j)}(D) = h_0^{(j)} + h_1^{(j)}D + \cdots + h_\nu^{(j)}D^\nu$, $j = 0, 1, \cdots, \tilde{k}$; ν is the constraint length; and $h_0^{(0)} = h_\nu^{(0)} = 1$ is required for a minimal encoder realization. Then, the vector $\mathbb{V}(D) = [\mathbb{v}^{(\tilde{k})}(D), \cdots, \mathbb{v}^{(1)}(D), \mathbb{v}^{(0)}(D)]$ is a codeword if and only if

$$\mathbb{V}(D)\mathbb{H}^T(D) = \sum_{(0\le j\le \tilde{k})} \mathbb{v}^{(j)}(D)\mathbb{h}^{(j)}(D) = \mathbb{0}(D), \qquad (18.24)$$

where $\mathbb{v}^{(j)}(D) \triangleq v_0^{(j)} + v_1^{(j)}D + v_2^{(j)}D^2 + \cdots + v_l^{(j)}D^l + \cdots$, $j = 0, 1, \cdots, \tilde{k}$; $\mathbb{v}_l = [v_l^{(\tilde{k})}, \cdots, v_l^{(1)}, v_l^{(0)}]$ represents the encoder output that selects the subset at time l; $\sum$ denotes modulo-2 addition; and $\mathbb{0}(D)$ represents the all-zero sequence. The general realization of a rate $R = \tilde{k}/(\tilde{k}+1)$ systematic feedback convolutional encoder with parity-check matrix given by (18.23) is shown in Figure 18.16(a).

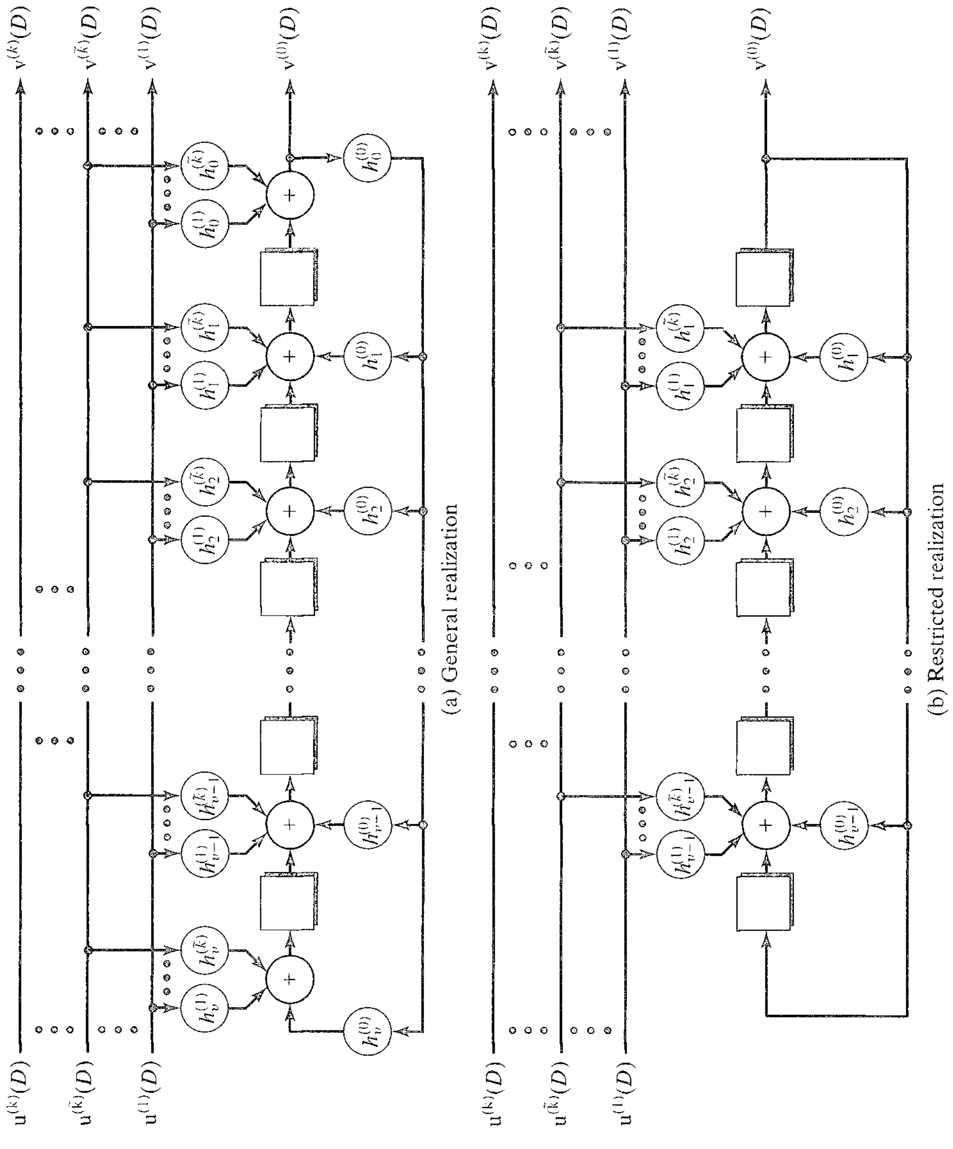
We now place some restrictions on the general encoder realization of Figure 18.16(a) that are appropriate for good TCM code design. Recall the set-partitioning requirement that the parity bit $v^{(0)}$ be the same for all branches leaving and entering a given state. It is easy to see from Figure 18.16(a) that to guarantee that $v^{(0)}$ be the same for all branches leaving a state, there must be no connections from any information sequence to the shift-register output; that is, we require that $h_0^{(1)} = h_0^{(2)} = \cdots = h_0^{(\tilde{k})} = 0$. Also, since $v^{(0)}$ is an input to the first (leftmost) register stage, and the output of the first register stage must be the same for all branches entering a given state, to guarantee that $v^{(0)}$ is the same for all branches entering a state, there must be no connections from an information sequence to the shift-register input; that is, we require that $h_\nu^{(1)} = h_\nu^{(2)} = \cdots = h_\nu^{(\tilde{k})} = 0$. These restrictions lead to the systematic feedback convolutional encoder realization shown in Figure 18.16(b) that is used to search for good TCM code designs.

The criterion for selecting good codes, based on the approximate expression for event-error probability given in (18.5), is to select codes that maximize the MFSE distance d_{free}^2 and minimize the average number of nearest neighbors $A_{d_{free}}$. An appropriate search algorithm must first find the codes with the largest d_{free}^2 and then select those with the smallest $A_{d_{free}}$. Assuming that the MSE distance δ_{min}^2 between parallel transitions is computed separately, we can use (18.13) to express the MFSE distance between trellis paths as

$$\delta_{free}^2 = \min_{\mathbb{e}(D)\neq\mathbb{0}(D)} \Delta^2[\mathbb{e}(D)] = \min_{\mathbb{e}(D)\neq\mathbb{0}(D)} \sum_l \Delta^2(\mathbb{e}_l). \qquad (18.25)$$

It is also possible to compute a lower bound on δ_{free}^2 directly from the binary error trellis of the code and the MSSDs Δ_p^2 in the set-partitioning tree using the following lemma.

> Lemma 18.2 (Set-Partitioning Lemma) [1] Let $q(\mathbb{e})$ denote the number of trailing zeros in the error vector $\mathbb{e}$, for example, $q(e^{(k)}, \cdots, e^{(3)}, 1, 0, 0) = 2$. Then,
>
> $$\Delta^2(\mathbb{e}) \ge \Delta_{q(\mathbb{e})}^2. \qquad (18.26)$$
>
> ***Proof.*** The proof of (18.26) follows from the fact that if $\mathbb{v}$ and $\mathbb{v}' = \mathbb{v} \oplus \mathbb{e}$ represent two trellis branch labels, then their corresponding signal point



(a) General realization

(b) Restricted realization

FIGURE 18.16: Two systematic feedback convolutional encoder realizations.

labels in the set-partitioning tree will agree in the trailing $q(\mathbf{e})$ positions. This implies that they follow the same path through the tree for the first $q(\mathbf{e})$ levels, and thus $\Delta_{\mathbf{v}}^2(\mathbf{e}) \geq \Delta_{q(\mathbf{e})}^2$. Because this condition holds for all $\mathbf{v}$, $\Delta^2(\mathbf{e}) = \min_{\mathbf{v}} \Delta_{\mathbf{v}}^2(\mathbf{e}) \geq \Delta_{q(\mathbf{e})}^2$. **Q.E.D.**

Using the set-partitioning lemma, we can now write

$$\delta_{free}^2 \geq \min_{\mathbf{e}(D) \neq \mathbf{0}(D)} \sum_l \Delta_{q(\mathbf{e}_l)}^2. \tag{18.27}$$

The following comments apply to the set-partitioning lemma.

- For the case $\mathbb{e} = \mathbb{0}$, we take $\Delta^2(\mathbb{e}) = \Delta^2_{q(\mathbb{0})} = 0$.

- Inequality (18.26) is satisfied with equality for most $\mathbb{e}$. For example, the only exception for 8-PSK is $\Delta^2(101) > \Delta_0^2$, and the only exceptions for 16-QAM are $\Delta^2(1001) > \Delta_0^2$, $\Delta^2(1101) > \Delta_0^2$, and $\Delta^2(1111) > \Delta_0^2$.

- Inequality (18.27) is almost always satisfied with equality, since there are usually several paths $\mathbb{e}(D) \neq \mathbb{0}(D)$ that achieve the minimum value of δ^2_{free}, including at least one that does not include any error vector $\mathbb{e}_l$ for which (18.26) is not satisfied with equality.

- Thus, either (18.25) or (18.27) can be used to compute the MFSE distance between trellis paths; but (18.27) is simpler, since it does not require computation of the Euclidean weight of every error vector. This can be especially advantageous for large signal sets or if codes are designed based on a lattice partitioning without specifying a particular signal set.

Sets of optimum TCM code designs based on the foregoing search procedure are listed in Tables 18.6(a)–(d). The codes were found by computer search [4]. Each table gives the following information:

- The MSSDs Δ_i^2, $i = 0, 1, \cdots, k$.

- The encoder constraint length ν.

- The number of coded information bits $\tilde{k}$.

- The parity-check coefficients $\mathbb{h}^{(j)} = \left[h_\nu^{(j)}, h_{\nu-1}^{(j)}, \cdots, h_1^{(j)}, h_0^{(j)}\right]$, $j = 0, 1, \cdots, \tilde{k}$, in octal form.

- The MFSE distance d^2_{free}. An asterisk (*) indicates that d^2_{free} occurs only along parallel transitions; that is, $\delta^2_{free} > \delta^2_{min}$. In Tables 18.6(a) and (b), the ratio of d^2_{free} to Δ_0^2, the MSSD at level 0, assuming the average energy $E_s = 1$, is given. (In these cases, Δ_0^2 varies with the signal constellation considered, but the ratio d^2_{free}/Δ_0^2 is constant.)

- The asymptotic coding gain in decibels compared with an uncoded modulation system with the same spectral efficiency. The notation denotes the two signal constellations being compared; for example, $\gamma_{32CR/16QAM}$ denotes the coding gain of a coded 32-CROSS constellation compared with uncoded 16-QAM. The number of information bits k transmitted per coded symbol, which equals the spectral efficiency η in bits/symbol, is also given. In Tables 18.6(a) and (b), coding gains are given for several different spectral efficiencies based on constellations chosen from the same lattice. The notation $\gamma_{C/U}$ denotes the coding gain of a coded lattice of infinite size compared with the uncoded lattice.

- The average number of nearest neighbors $A_{d_{free}}$. In Tables 18.6(a) and (b), $A_{d_{free}}$ is given only for the infinite spectral efficiency case, that is, $k \to \infty$.

TABLE 18.6: List of optimum TCM codes.

(a) Codes for one-dimensional AM based on $\mathbb{Z}^1$

$$\{\Delta_i^2, 0 \le i \le 2\} = \Delta_0^2, 4\Delta_0^2, 16\Delta_0^2$$

ν	$\tilde{k}$	$\mathbf{h}^{(1)}$	$\mathbf{h}^{(0)}$	d_{free}^2/Δ_0^2	$\gamma_{4\text{-AM}/2\text{-AM}}$ $k = 1$	$\gamma_{8\text{-AM}/4\text{-AM}}$ $k = 2$	$\gamma_{C/U}$ $(k \to \infty)$	$A_{d_{free}}$ $(k \to \infty)$
2	1	2	5	9.0	2.55	3.31	3.52	4
2	1	2	5	9.0	2.55	3.31	3.52	4
3	1	04	13	10.0	3.01	3.77	3.97	4
4	1	04	23	11.0	3.42	4.18	4.39	8
5	1	10	45	13.0	4.15	4.91	5.11	12
6	1	024	103	14.0	4.47	5.23	5.44	36
7	1	126	235	16.0	5.05	5.81	6.02	66
8	1	362	515	16.0*	—	5.81	6.02	2
	1	362	515	17.0	5.30	—	—	

(b) Codes for two-dimensional AM/PM based on $\mathbb{Z}^2$

$$\{\Delta_i^2, 0 \le i \le 3\} = \Delta_0^2, 2\Delta_0^2, 4\Delta_0^2, 8\Delta_0^2$$

ν	$\tilde{k}$	$\mathbf{h}^{(2)}$	$\mathbf{h}^{(1)}$	$\mathbf{h}^{(0)}$	d_{free}^2/Δ_0^2	$\gamma_{16\text{-QAM}/8\text{-PSK}}$ $k = 3$	$\gamma_{32\text{-CR}/16\text{-QAM}}$ $k = 4$	$\gamma_{64\text{-QAM}/32\text{-CR}}$ $k = 5$	$\gamma_{C/U}$ $(k \to \infty)$	$A_{d_{free}}$ $(k \to \infty)$
2	1	—	2	5	4.0*	4.36	3.01	2.80	3.01	4
3	2	04	02	11	5.0	5.33	3.98	3.77	3.98	16
4	2	16	04	23	6.0	6.12	4.77	4.56	4.77	56
5	2	10	06	41	6.0	6.12	4.77	4.56	4.77	16
6	2	064	016	101	7.0	6.79	5.44	5.23	5.44	56
7	2	042	014	203	8.0	7.37	6.02	5.81	6.02	344
8	2	304	056	401	8.0	7.37	6.02	5.81	6.02	44
9	2	0510	0346	1001	8.0*	7.37	6.02	5.81	6.02	4

(c) Codes for 8-PSK

$$\{\Delta_i^2, 0 \le i \le 2\} = 4\sin^2(\pi/8), 2, 4$$

ν	$\tilde{k}$	$\mathbf{h}^{(2)}$	$\mathbf{h}^{(1)}$	$\mathbf{h}^{(0)}$	d_{free}^2/Δ_0^2	$\gamma_{8\text{-PSK}/4\text{-PSK}}$ $k = 2$	$A_{d_{free}}$
2	1	—	2	5	4.000*	3.01	1
3	2	04	02	11	4.586	3.60	2
4	2	16	04	23	5.172	4.13	≈2.3
5	2	34	16	45	5.758	4.59	4
6	2	066	030	103	6.343	5.01	≈5.3
7	2	122	054	277	6.586	5.17	≈0.5
8	2	130	072	435	7.515	5.75	≈1.5

(d) Codes for 16-PSK

$$\{\Delta_i^2, 0 \le i \le 3\} = 4\sin^2(\pi/16), 4\sin^2(\pi/8), 2, 4$$

ν	$\tilde{k}$	$\mathbf{h}^{(2)}$	$\mathbf{h}^{(1)}$	$\mathbf{h}^{(0)}$	d_{free}^2/Δ_0^2	$\gamma_{16\text{-PSK}/8\text{-PSK}}$ $k = 3$	$A_{d_{free}}$
2	1	—	2	5	1.324	3.54	4
3	1	—	04	13	1.476	4.01	4
4	1	—	04	23	1.628	4.44	8
5	1	—	10	45	1.910	5.13	8
6	1	—	024	103	2.000*	5.33	2
7	1	—	024	203	2.000*	5.33	2
8	2	374	176	427	2.085	5.51	≈8.0

Adapted from [4].

The codes listed for one-dimensional AM are based on the one-dimensional integer lattice $\mathbb{Z}^1$, and the codes listed for two-dimensional AM/PM are based on the two-dimensional integer lattice $\mathbb{Z}^2$, where these lattices are infinite extensions of the one- and two-dimensional signal constellations shown in Figures 18.1(a) and 18.1(b). In these cases the same codes yield the same maximum d^2_{free}/Δ^2_0 independent of the size of the signal constellation chosen from the lattice, although the minimum number of nearest neighbors can vary owing to the effect of the signal constellation boundaries. In Tables 18.6(a) and (b), to negate the effect of constellation boundaries, we list only the average multiplicities $A_{d_{free}}$ assuming a signal constellation of infinite size. Because set partitioning of an infinite lattice results in a regular mapping, the values of $A_{d_{free}}$ in Tables 18.6(a) and (b) are all integers. In Table 18.6(a), we see that for codes based on $\mathbb{Z}^1$, the asymptotic coding gain γ increases with the spectral efficiency k; that is, the largest coding gains are achieved in the limit as $k \to \infty$. Also, two optimum 256-state codes are listed. The first code, whose d^2_{free} occurs along parallel transitions, is optimum when the number of information bits $k \geq 2$; that is, when the trellis contains parallel transitions. The second code, which achieves a larger d^2_{free}, is optimum only when $k = 1$, that is, when the trellis does not contain parallel transitions. In Table 18.6(b) we note the relatively large asymptotic coding gains of coded 16-QAM compared with uncoded 8-PSK. This difference is due to the restriction that PSK signals must all have the same energy. The coding gains of 16-QAM compared with uncoded rectangular constellations are not as large, as shown in Problem 18.15. In contrast with the lattice-based codes, in Tables 18.6(c) and (d) we see that different codes are optimum for 8-PSK and 16-PSK constellations, and that the nonregular mapping can result in noninteger values of the average multiplicities $A_{d_{free}}$.

When a trellis contains parallel transitions, care must be taken in computing the value of $A_{d_{free}}$, since each parallel branch may contribute to a minimum-distance path. For example, in Table 18.6(a), $A_{d_{free}} = 4$ for the 4-state coded integer lattice $\mathbb{Z}^1$. Referring to the error trellis in Figure 18.4(b) for code 2, which is equivalent to the 4-state code in Table 18.6(a), we note that the error trellis for the coded lattice $\mathbb{Z}^1$ is formed by replacing each branch with an (infinite) set of parallel transitions. In this case the trellis branches labeled $\mathbb{e} = (00)$ will now contain the set of parallel transitions representing all error vectors $\mathbb{e} = \left(\cdots e^{(3)}e^{(2)}00\right)$, the trellis branches labeled $\mathbb{e} = (10)$ will now contain the set of parallel transitions representing all error vectors $\mathbb{e} = \left(\cdots e^{(3)}e^{(2)}10\right)$, and so on. The MSE distance $9\Delta^2_0$ is achieved by a path that diverges from state S_0 along the branch labeled $\mathbb{e} = (10)$ to state S_1 (a squared distance of $4\Delta^2_0$), continues to state S_2 along the branch labeled $\mathbb{e} = (01)$ (a squared distance of Δ^2_0), and remerges with state S_0 along the branch labeled $\mathbb{e} = (10)$ (a squared distance of $4\Delta^2_0$). Now, note that for a given parallel transition on the branch labeled $\mathbb{e} = (00)$ leaving state S_0, say $\mathbb{e} = \left(\cdots e^{(4)}e^{(3)}000\right)$, there are two parallel transitions, $\mathbb{e} = \left(\cdots e^{(4)}e^{(3)}110\right)$ and $\mathbb{e} = \left(\cdots e^{(4)}e^{(3)}010\right)$, with squared distance $4\Delta^2_0$ on the diverging branch labeled $\mathbb{e} = (10)$. The same situation holds when the minimum-weight path remerges with state S_0 along the branch labeled $\mathbb{e} = (10)$. For the middle branch on the minimum-weight path, there is only one parallel transition, $\mathbb{e} = \left(\cdots e^{(4)}e^{(3)}001\right)$, with squared distance Δ^2_0 along the branch labeled $\mathbb{e} = (01)$. Because there are four possible combinations of minimum-weight paths, in this case, $A_{d_{free}} = 4$. Another example of computing $A_{d_{free}}$ for a trellis with parallel transitions is given in Problem 18.16. Finally, we recall that when $A_{d_{free}}$ of the coded system exceeds A_{min} of the uncoded system, the real coding gain at practical BERs of around 10^{-5} is somewhat reduced compared with the asymptotic coding gain γ.

It is interesting to note that many of the optimum codes listed in Table 18.6 contain one or more uncoded bits. This is because, particularly for short constraint lengths, the parallel transition distance $\delta_{min}^2 = \Delta_{\tilde{k}+1}^2$ is already larger than the free distance δ_{free}^2 between trellis paths, and thus using a higher rate code cannot improve the overall free distance d_{free}^2. For longer constraint lengths, however, the free distance δ_{free}^2 between trellis paths increases, and then more coded bits, that is, a larger $\tilde{k}$, must be used to increase the parallel transition distance δ_{min}^2 and consequently the overall free distance d_{free}^2.

18.3 TCM PERFORMANCE ANALYSIS

The *average weight enumerating function* (AWEF) $A_{av}(X)$ and the *average input output weight enumerating function* (AIOWEF) $A_{av}(W, X)$ of a TCM system can be computed by labeling the branches of the binary error trellis with their corresponding AEWEs, augmented by the input weight enumerators when computing $A_{av}(W, X)$, and then forming the modified state diagram and using the transfer function approach developed in Chapter 11. Once $A_{av}(X)$ and $A_{av}(W, X)$ have been evaluated, the event-error probability $P(E)$ and the bit-error probability $P_b(E)$ can be estimated using the union bounding techniques developed in Chapter 12. For an unquantized-output AWGN channel whose inputs are drawn from the TCM signal set, this process gives the expressions

$$P(E) \le f(d_{free}^2 E_s/4N_0) A_{av}(X)\Big|_{X=e^{-E_s/4N_0}} \tag{18.28a}$$

and

$$P_b(E) \le (1/k)\ f(d_{free}^2 E_s/4N_0) \partial A_{av}(W, X)/\partial W\Big|_{X=e^{-E_s/4N_0},\, W=1}, \tag{18.28b}$$

where $f(x) = e^x Q(\sqrt{2x})$, and d_{free}^2 is computed under the assumption of a unit average energy signal set. The reader should note the similarity between the expressions in (18.28) and those derived for binary convolutional codes in Chapter 12. In fact, they are identical except that the WEFs are replaced by *average* WEFs, and the Hamming distance, $d_{H,free}$, in Chapter 12 is replaced by $d_{free}^2/4$, where d_{free}^2 is SE distance, in the preceding expressions. This reflects the fact that, for unit energy binary signals, $d_{free}^2 = 4d_{H,free}$, as noted in (18.8).

The bounds in (18.28) are valid for any TCM system without parallel transitions, that is, the case for which each error event represents a path through the trellis at least two branches in length. In the case of parallel transitions, that is, one-branch error events, the bounds are modified as follows:

$$\begin{aligned} P(E) \le\ & f(\delta_{min}^2 E_s/4N_0) A_{av}^p(X)\Big|_{X=e^{-E_s/4N_0}} \\ & + f(\delta_{free}^2 E_s/4N_0) A_{av}^t(X)\Big|_{X=e^{-E_s/4N_0}} \end{aligned} \tag{18.29a}$$

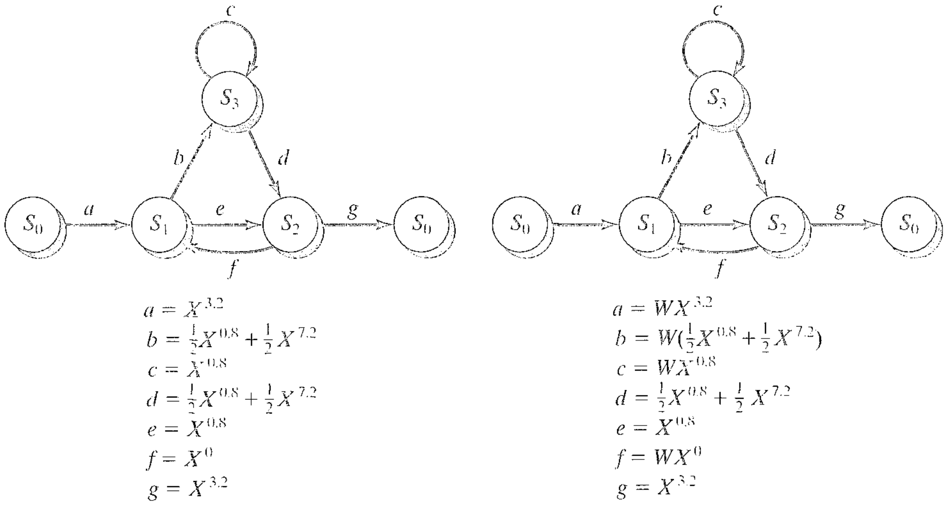
and

$$P_b(E) \leq (1/k)\ f(\delta^2_{min} E_s/4N_0)\partial A^p_{av}(W,X)/\partial W\Big|_{X=e^{-E_s/4N_0},W=1}$$
$$+ (1/k)\ f(\delta^2_{free} E_s/4N_0)\partial A^t_{av}(W,X)/\partial W\Big|_{X=e^{-E_s/4N_0},W=1}, \tag{18.29b}$$

where $A^p_{av}(X)$ and $A^p_{av}(W,X)$ represent the AWEF and AIOWEF for the parallel transition paths, and $A^t_{av}(X)$ and $A^t_{av}(W,X)$ represent the AWEF and AIOWEF for the trellis paths, respectively. (It should be noted that $A^p_{av}(X)$ and $A^p_{av}(W,X)$ are simply the AWEFs of the subsets at the last level, that is, level $\tilde{k}+1$, in the set-partitioning tree.) The use of WEFs to evaluate the performance of TCM systems was introduced by Zehavi and Wolf [24], and an algorithm for computing the AWEF was presented in [25]. We now illustrate the application of the bounds with two examples.

EXAMPLE 18.8 4-State, Rate $R = 1/2$ Trellis-Coded 4-AM

We consider the 4-state, rate $R = 1/2$ binary feedforward encoder shown for code 2 in Figure 18.4(a) along with naturally mapped 4-AM. The binary error trellis of this encoder was shown in Figure 18.4(b), and the AEWEs of naturally mapped 4-AM were listed in Table 18.4(b). In Figure 18.17(a) we show the modified state diagram labeled with the AEWEs. We now compute the AWEF using the standard transfer



$a = X^{3.2}$
$b = \frac{1}{2}X^{0.8} + \frac{1}{2}X^{7.2}$
$c = X^{0.8}$
$d = \frac{1}{2}X^{0.8} + \frac{1}{2}X^{7.2}$
$e = X^{0.8}$
$f = X^{0}$
$g = X^{3.2}$

(a) Modified state diagram

$a = WX^{3.2}$
$b = W(\frac{1}{2}X^{0.8} + \frac{1}{2}X^{7.2})$
$c = WX^{0.8}$
$d = \frac{1}{2}X^{0.8} + \frac{1}{2}X^{7.2}$
$e = X^{0.8}$
$f = WX^{0}$
$g = X^{3.2}$

(b) Augmented modified state diagram

FIGURE 18.17: Modified state diagrams for naturally mapped, 4-state, rate $R = 1/2$ trellis-coded 4-AM.

function approach as follows:

$$\begin{aligned}A_{av}(X) &= \frac{X^{3.2}X^{0.8}X^{3.2}(1 - X^{0.8}) + X^{3.2}(0.5X^{0.8} + 0.5X^{7.2})^2 X^{3.2}}{(1 - X^{0.8})(1 - X^{0.8}X^{0}) - (0.5X^{0.8} + 0.5X^{7.2})^2 X^{0}} \\ &= \frac{X^{7.2} - 0.75X^{8.0} + 0.5X^{14.4} + 0.25X^{20.8}}{1 - 2X^{0.8} + 0.75X^{1.6} - 0.5X^{8.0} - 0.25X^{14.4}} \\ &= X^{7.2} + 1.25X^{8.0} + 1.75X^{8.8} + 2.0625X^{9.6} + \cdots . \qquad (18.30\text{a})\end{aligned}$$

Equation (18.30a) implies that for an arbitrary transmitted sequence $\mathbf{y}$, there is an *average* of 1 error path $\mathbf{y}'$ with MFSE distance $d^2_{free} = d^2_E(\mathbf{y}, \mathbf{y}') = 7.2$, an *average* of 1.25 error paths $\mathbf{y}'$ with SE distance $d^2_E(\mathbf{y}, \mathbf{y}') = 8.0$, an *average* of 1.75 error paths $\mathbf{y}'$ with SE distance $d^2_E(\mathbf{y}, \mathbf{y}') = 8.8$, and so on.

In Figure 18.17(b) we show the modified state diagram augmented by the input weight enumerators. In this case, following the same procedure as before, we find that the AIOWEF is given by

$$A_{av}(W, X) = WX^{7.2} + 1.25W^2X^{8.0} + 1.75W^3X^{8.8} + 2.0625W^4X^{9.6} + \cdots . \qquad (18.30\text{b})$$

Here we see that the error paths at a distance of 7.2 from the correct path are always associated with 1 information bit error, those at a distance of 8.0 are always associated with 2 information bit errors, those at a distance of 8.8 with 3 information bit errors, and so on.

Finally, the expressions of (18.30) can be used in (18.28) to evaluate $P(E)$ and $P_b(E)$ as functions of the channel SNR E_s/N_0. The bounds are sketched in Figure 18.18 along with uncoded BPSK, which has the same spectral efficiency of $\eta = 1$ bit/dimension.

The following observations relate to Example 18.8:

- The codeword multiplicities are averages because TCM systems are nonlinear, and the number of codewords at a particular distance from the correct sequence depends on the transmitted path.

- The codeword multiplicities are fractional because the signal constellation is finite, and the mapping is nonregular. Thus, the multiplicity of 1.25 associated with incorrect paths at distance 8.0 means that, depending on the correct path, there may be either 1 or 2 incorrect paths at distance 8.0.

- If the same rate $R = 1/2$ code was used, along with an infinite number of uncoded information bits, to code the one-dimensional integer lattice $\mathbb{Z}^1$, the average multiplicities would be integers, since a regular signal mapping can then be achieved with set partitioning, as noted in the previous section. In this case the average number of nearest neighbors is $A_{d_{free}} = 4$, since the parallel transition subsets in both the first and last branches of the shortest error event contain exactly two signal points distance 3.2 away from any given reference point (see Problem 18.15).

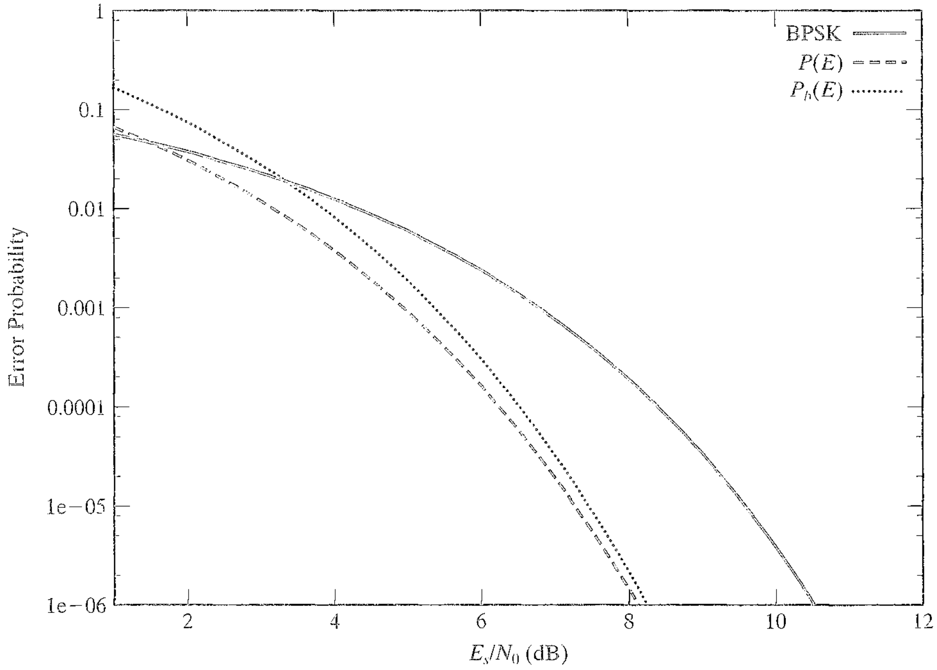


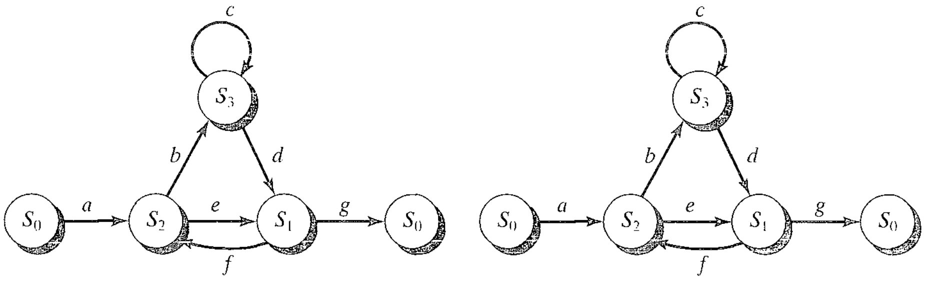
FIGURE 18.18: Error probability bounds for naturally mapped, 4-state, rate $R = 1/2$ trellis-coded 4-AM.

- Because of the particular structure of the encoder in this example, a deterministic relationship exists between codeword distance from the correct path and information weight; that is, $d_E^2 = 6.4 + 0.8w_I$, where d_E^2 represents the SE distance from the correct path, and w_I represents the corresponding information weight. For example, all codewords distance 14.4 from the correct path have information weight 10.

- From the bound on $P_b(E)$ plotted in Figure 18.18, we see that the *real coding gain* at a BER of 10^{-5} of this TCM system compared with uncoded BPSK is approximately 2.1 dB. This coding gain is achieved without bandwidth expansion.

EXAMPLE 18.9 4-State, Rate $R = 1/2$ Trellis-Coded 8-PSK

Now, consider the 4-state, rate $R = 1/2$ binary feedback encoder shown for code 2 in Figure 18.9(a) along with one uncoded information bit and naturally mapped 8-PSK modulation. The binary error trellis of this encoder was shown in Figure 18.9(b), where there was a parallel transition connecting each pair of states, and the AEWEs of naturally mapped 8-PSK modulation were listed in Table 18.5. In Figure 18.19(a) we show the modified state diagram, in which each branch is labeled with the sum of the AEWEs for the corresponding parallel transition branch labels in the binary trellis. We can now compute the AWEF for the trellis paths $A_{av}^t(X)$ using the



$$\begin{aligned} a &= 2X^2 \\ b &= X^{0.586} + X^{3.414} \\ c &= X^{0.586} + X^{3.414} \\ d &= X^{0.586} + X^{3.414} \\ e &= X^{0.586} + X^{3.414} \\ f &= 1 + X^4 \\ g &= 2X^2 \end{aligned}$$

(a) Modified state diagram

$$\begin{aligned} a &= (W + W^2)X^2 \\ b &= \tfrac{1}{2}(W + W^2)(X^{0.586} + X^{3.414}) \\ c &= X^{0.586} + WX^{3.414} \\ d &= \tfrac{1}{2}(W + W^2)(X^{0.586} + X^{3.414}) \\ e &= X^{0.586} + WX^{3.414} \\ f &= 1 + WX^4 \\ g &= (W + W^2)X^2 \end{aligned}$$

(b) Augmented modified state diagram

FIGURE 18.19: Modified state diagrams for naturally mapped, 4-state, rate $R = 1/2$ trellis-coded 8-PSK.

standard transfer function approach as follows:

$$\begin{aligned} A_{av}^t(X) &= \frac{\begin{array}{c}2X^2(X^{0.586} + X^{3.414})2X^2(1 - X^{0.586} - X^{3.414}) + \\ 2X^2(X^{0.586} + X^{3.414})^2 2X^2\end{array}}{\begin{array}{c}[1 - X^{0.586} - X^{3.414}][1 - (X^{0.586} - X^{3.414})(X^0 + X^4)] - \\ [X^{0.586} + X^{3.414}]^2[X^0 + X^4]\end{array}} \\ &= \frac{4X^{4.586} + 4X^{7.414}}{1 - 2X^{0.586} - 2X^{3.414} - X^{4.586} - X^{7.414}} \\ &= 4X^{4.586} + 8X^{5.172} + 16X^{5.758} + 32X^{6.344} + \cdots . \end{aligned} \tag{18.31a}$$

Equation (18.31a) implies that for an arbitrary transmitted sequence $\mathbf{y}$, there is an *average* of 4 error paths $\mathbf{y}'$ with a MFSE distance *between trellis paths* of $\delta_{free}^2 = d_E^2(\mathbf{y}, \mathbf{y}') = 4.586$, an *average* of 8 error paths $\mathbf{y}'$ with free SE distance $d_E^2(\mathbf{y}, \mathbf{y}') = 5.172$, and so on. Because this TCM system includes parallel transitions, we must also compute the parallel transition AWEF $A_{av}^p(X)$. From the 8-PSK set-partitioning tree, we see that there are only two signal points in each parallel transition, and that

$$A_{av}^p(X) = X^{4.0}. \tag{18.31b}$$

Equation (18.31b) implies that the MSE distance *between parallel transitions* is $\delta_{min}^2 = 4.0$, and hence the MFSE distance of the TCM system is

$$d_{free}^2 = \min\left\{\delta_{free}^2, \delta_{min}^2\right\} = 4.0, \tag{18.31c}$$

as noted earlier in Example 18.4.

In Figure 18.19(b) we show the modified state diagram augmented by the input weight enumerators. In this case, following the same procedure as above, we find that the AIOWEF is given by (see Problem 18.17)

$$\begin{aligned} A^{t}_{av}(W, X) &= \left(W + W^2\right)^2 X^{4.586} + \left[\left(W + W^2\right)^2 + 0.25\left(W + W^2\right)^4\right] X^{5.172} + \cdots \\ &= \left(W^2 + 2W^3 + W^4\right) X^{4.586} + \left(W^2 + 2W^3 + 1.25W^4 + W^5\right. \\ &\quad \left. + 1.5W^6 + W^7 + 0.25W^8\right) X^{5.172} + \cdots . \end{aligned} \tag{18.31d}$$

Here we see that the error paths at a distance of 4.586 from the correct path are associated with 2, 3, or 4 information bit errors, those at a distance of 5.172 are associated with between 2 and 8 information bit errors, and so on. In addition, the coefficients of the W terms denote the relative likelihood that a certain number of information bit errors will correspond to error paths of a given weight. For example, the terms $1.25W^4X^{5.172}$ and $0.25W^8X^{5.172}$ indicate that error paths at a distance of 5.172 from the correct path are five times more likely to have 4 information bit errors than 8. Finally, the parallel transition AIOWEF is given by

$$A^{p}_{av}(W, X) = WX^{4.0}, \tag{18.31e}$$

which indicates that all parallel transition error events are associated with one information bit error.

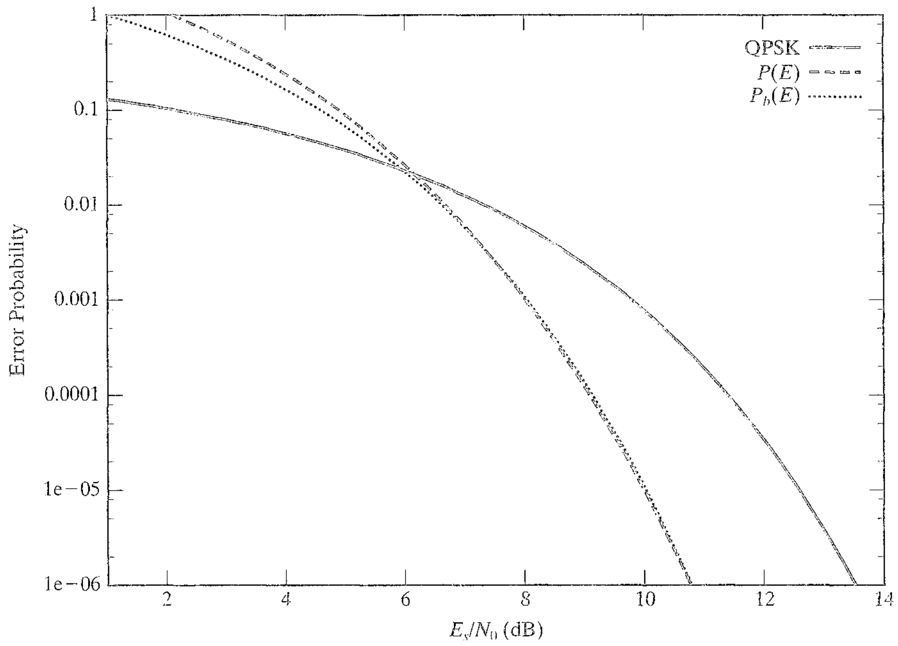


FIGURE 18.20: Error probability bounds for naturally mapped, 4-state, rate $R = 1/2$ trellis-coded 8-PSK.

Now, we can use the expressions of (18.31) in (18.29) to evaluate $P(E)$ and $P_b(E)$ as functions of the channel SNR E_s/N_0. The bounds are sketched in Figure 18.20 along with uncoded QPSK, which has the same spectral efficiency of $\eta = 2$ bits/symbol.

The following remarks apply to Example 18.9:

- The WEFs for each possible parallel transition, that is, for each subset at level 2 of the partition tree, are identical, because the four BPSK subsets at level 2 are isomorphic. In general, however, this may not be the case, and $A_{av}^{p}(X)$ is computed by taking the average of the WEFs of each subset at level $\tilde{k}+1$ in the set-partitioning tree.
- The MFSE distance path is a parallel transition. This implies that at high SNRs, $A_{av}^{p}(X)$ and $A_{av}^{p}(W, X)$ are the dominant terms in the error probability bounds, thus allowing approximate bounds on $P(E)$ and $P_b(E)$ to be obtained very simply.
- The possible codeword weights represented in (18.31a) and (18.31b) are separated by the value $\Delta_{\text{min}}^2 = 0.586$, the MSE distance between signal points. This is characteristic of any TCM system; that is, codeword weights increase by multiples of Δ_{min}^2.
- From the bound on $P_b(E)$ plotted in Figure 18.20, we see that the *real coding gain* at a BER of 10^{-5} of this TCM system compared with uncoded QPSK is approximately 2.6 dB. This coding gain is achieved without bandwidth expansion.

As a final comment before leaving this section, we note that whereas the *asymptotic coding gain* of a TCM system can be obtained by computing d_{free}^2, as shown in Section 18.1, the *real coding gain* at a particular BER must be obtained either from computer simulations or estimated from bounds such as the ones presented in this section.

18.4 ROTATIONALLY INVARIANT TCM

The typical signal set used in a coded modulation system has several phase symmetries; that is, phase rotations by certain angles replicate the signal set. For example, 8-PSK has eight phase symmetries, spaced 45° apart, and any QAM constellation has four phase symmetries spaced by 90°. In general, when a receiver locks onto a particular phase, a trial-and-error procedure must be initiated to determine whether it is in the correct phase. If a particular system is prone to frequent loss of carrier synchronization, reacquiring sync can be a time-consuming exercise. Thus, it is desirable in many applications that a coded modulation system be invariant to phase rotations. In other words, if the receiver locks onto the wrong phase, the system should still be able to operate properly. Hence, in case of temporary loss of synchronization, the receiver must lock onto only one of the possible symmetries and need not initiate a procedure to reacquire the correct phase.

The basic requirement for a coded modulation system to be invariant to a particular phase rotation is that when the symbols of each code sequence are replaced by the corresponding symbols in the rotated signal set, it is still a valid code sequence. In other words, in the absence of noise, the decoder would still decode a proper code sequence, albeit an incorrect code sequence; however, differential encoding of the information sequence and differential decoding of the decoder output sequence can be employed to ensure that the correct sequence is decoded. Thus, a rotationally invariant code combined with differential encoding/decoding can be used to provide reliable communication even when the receiver is locked onto the wrong phase, although a small penalty in BER performance is incurred because isolated single-bit errors at the decoder output are doubled by differential decoding. (Note that if a code is not rotationally invariant, rotated code sequences are, in general, not code sequences, and this property may be used to detect an out-of-phase lock condition.)

To illustrate the idea of rotationally invariant codes, we start with the simple case of a binary code with BPSK modulation in which the encoder output bits are represented as 0 → -1 and 1 → $+1$. The only phase symmetry of the signal set is caused by a 180° rotation, which has the effect of inverting the sign of every modulation symbol; that is, $-1 \to +1$ and $+1 \to -1$. Thus, every codeword is replaced by its complement. For any linear code, the complement of a codeword is a codeword if and only if the all-one sequence is a codeword. Thus, the simple condition for 180° rotational invariance for any linear binary code with BPSK modulation is that the all-one sequence is a codeword.

We now consider the case of QPSK modulation and 90° phase symmetries, beginning with an example.

EXAMPLE 18.10 Rate $R = 1/2$ Coded QPSK

Consider a rate $R = 1/2$ convolutional code with generator matrix

$$\mathbb{G}(D) = \left[\mathbb{h}^{(0)}(D) \quad \mathbb{h}^{(1)}(D)\right] \tag{18.32a}$$

and parity-check matrix

$$\mathbb{H}(D) = \left[\mathbb{h}^{(1)}(D) \quad \mathbb{h}^{(0)}(D)\right], \tag{18.32b}$$

whose two encoder output bits are Gray mapped into QPSK, as shown in Figure 18.3(a). In this case the *parity-check equation* (PCE) is given by

$$\mathbb{V}(D)\mathbb{H}^T(D) = \mathbb{v}^{(1)}(D)\mathbb{h}^{(1)}(D) \oplus \mathbb{v}^{(0)}(D)\mathbb{h}^0(D) = \mathbb{0}(D), \tag{18.33}$$

where $\mathbb{V}(D) = [\mathbb{v}^{(1)}(D), \mathbb{v}^{(0)}(D)]$ represents a codeword. Now, note that after a 90° rotation of the signal set, the rotated code sequences become

$$\mathbb{v}_r^{(1)}(D) = \mathbb{v}^{(0)}(D) \quad \text{and} \quad \mathbb{v}_r^{(0)}(D) = \mathbb{v}^{(1)}(D) \oplus \mathbb{1}(D), \tag{18.34}$$

where $\mathbb{1}(D)$ represents the all-one sequence. Using the rotated code sequences in (18.33) we obtain

$$\begin{aligned} \mathbb{V}_r(D)\mathbb{H}^T(D) &= \mathbb{v}_r^{(1)}(D)\mathbb{h}^{(1)}(D) \oplus \mathbb{v}_r^{(0)}(D)\mathbb{h}^{(0)}(D) \\ &= \mathbb{v}^{(0)}(D)\mathbb{h}^{(1)}(D) \oplus \left[\mathbb{v}^{(1)}(D) \oplus \mathbb{1}(D)\right]\mathbb{h}^{(0)}(D), \end{aligned} \tag{18.35}$$

which must equal $\mathbb{0}(D)$ for all $\mathbb{V}_r(D) = [\mathbb{v}_r^{(1)}(D), \mathbb{v}_r^{(0)}(D)]$ for the code to be invariant to 90° rotations. From (18.33) we see that for any information sequence $\mathbb{u}(D)$, $\mathbb{V}(D) = [\mathbb{v}^{(1)}(D), \mathbb{v}^{(0)}(D)] = [\mathbb{u}(D)\mathbb{h}^{(0)}(D), \mathbb{u}(D)\mathbb{h}^{(1)}(D)]$ is a valid code sequence. Substituting this $\mathbb{V}(D)$ into (18.35) we have

$$\mathbb{V}_r(D)\mathbb{H}^T(D) = \mathbb{u}(D)\left\{\left[\mathbb{h}^{(1)}(D)\right]^2 \oplus \left[\mathbb{h}^{(0)}(D)\right]^2\right\} \oplus \mathbb{h}^{(0)}(D)\mathbb{1}(D). \tag{18.36}$$

Examining (18.36) closely we see that the first term equals $\mathbb{u}(D)$ times a nonzero binary polynomial of degree at most 2ν, where ν is the constraint length. Considering $\mathbb{1}(D)$ to extend infinitely in both directions, we see that we can write the second term as $\mathbb{h}^0(D)\mathbb{1}(D) = \mathbb{h}^{(0)}(1)(D)$, which equals either $\mathbb{0}(D)$ or $\mathbb{1}(D)$ depending on whether the Hamming weight of $\mathbb{h}^{(0)}(D)$ is even or odd, respectively. For example, code 1 in Example 18.2 has $\mathbb{h}^{(0)}(D) = 1 + D^2$ and $\mathbb{h}^{(1)}(D) = 1 + D + D^2$, and (18.36) becomes

$$\mathbb{V}_r(D)\mathbb{H}^T(D) = D^2\mathbb{u}(D) \oplus \mathbb{0}(D) = D^2\mathbb{u}(D). \tag{18.37}$$

Clearly, (18.37) is not equal to $\mathbb{0}(D)$ for any nonzero $\mathbb{u}(D)$, and thus the coded modulation system is not invariant to 90° rotations.

In the case of a 180° phase rotation, we can see directly from Figure 18.3(a) that both rotated code sequences are complements of their respective correct code sequences. This situation is exactly analogous to the BPSK case, and thus rate $R = 1/2$ convolutional codes with Gray-mapped QPSK are invariant to 180° phase rotations if and only if the all-one sequence $\mathbb{1}(D)$ is a codeword.

Now, consider the same example with natural mapping. In this case the 90° rotated code sequences become

$$\mathbb{v}_r^{(1)}(D) = \mathbb{v}^{(1)}(D) \oplus \mathbb{v}^{(0)}(D) \quad \text{and} \quad \mathbb{v}_r^{(0)}(D) = \mathbb{v}^{(0)}(D) \oplus \mathbb{1}(D). \tag{18.38}$$

Again, substituting in (18.33) and using $\mathbb{V}(D) = [\mathbb{u}(D)\mathbb{h}^{(0)}(D), \mathbb{u}(D)\mathbb{h}^{(1)}(D)]$, we obtain

$$\mathbb{V}_r(D)\mathbb{H}^T(D) = \mathbb{u}(D)\left\{\left[\mathbb{h}^{(1)}(D)\right]^2\right\} \oplus \mathbb{h}^{(0)}(D)\mathbb{1}(D), \tag{18.39}$$

which for code 1 in Example 18.2 becomes

$$\mathbb{V}_r(D)\mathbb{H}^T(D) = \left(1 + D + D^2\right)^2 \mathbb{u}(D) \oplus \mathbb{0}(D) = \left(1 + D^2 + D^4\right)\mathbb{u}(D). \tag{18.40}$$

As in the case of Gray-mapped QPSK, we see that for naturally mapped QPSK (18.40) does not equal $\mathbb{0}(D)$ for any nonzero $\mathbb{u}(D)$, and thus the coded modulation system is not invariant to 90° rotations.

For a 180° phase rotation, we see from Figure 18.3(b) that $\mathbb{v}_r^{(1)}(D) = \mathbb{v}^{(1)}(D) \oplus \mathbb{1}(D)$ and $\mathbb{v}_r^{(0)}(D) = \mathbb{v}^{(0)}(D)$; that is, $\mathbb{v}_r^{(1)}(D)$ is the complement of the correct sequence, and $\mathbb{v}_r^{(0)}(D)$ equals the correct sequence. In this case, $\mathbb{V}_r(D)\mathbb{H}^T(D) = \mathbb{v}_r^{(1)}(D)\mathbb{h}^{(1)}(D) \oplus \mathbb{v}_r^{(0)}(D)\mathbb{h}^{(0)}(D) = \mathbb{h}^{(1)}(D)\mathbb{1}(D)$. Thus rate $R = 1/2$ convolutional codes with naturally mapped QPSK are invariant to 180° phase rotations if and only if $\mathbb{h}^{(1)}(D)$ has even weight.

The following remarks relate to Example 18.10:

- Rotational invariance is a property of the code, not the encoder, so the results of Example 18.10 also hold for the equivalent systematic feedback encoder with $\mathbb{H}(D) = \left[\mathbb{h}^{(1)}(D)/\mathbb{h}^{(0)}(D) \ \ 1\right]$.

- If we consider $\mathbb{1}(D)$ to start at time 0, then terms of the form $\mathbb{h}^{(0)}(D)\mathbb{1}(D)$ would have a short (degree $< \nu$) preamble before reaching their steady-state value of $\mathbb{0}(D)$ or $\mathbb{1}(D)$. This would affect rotational invariance only in the first ν time units, and thus it makes sense to ignore this transient condition by assuming that $\mathbb{1}(D)$ extends infinitely in both directions (see Problem 18.19).

- This example can be generalized to show that any rate $R = k/(k+1)$ convolutional code with a *linear binary* PCE $\mathbb{V}(D)\mathbb{H}^T(D) = \mathbb{v}^{(k)}(D)\mathbb{h}^{(k)}(D) \oplus \cdots \oplus \mathbb{v}^{(1)}(D)\mathbb{h}^{(1)}(D) \oplus \mathbb{v}^{(0)}(D)\mathbb{h}^{(0)}(D) = \mathbb{0}(D)$ that maps into a two-dimensional signal set with 90° phase symmetries can at best be invariant to 180° phase rotations [26].

- As will be seen next, binary rate $R = k/(k+1)$ convolutional codes can achieve 90° rotational invariance only by making use of a *nonlinear* PCE.

To see how nonlinear PCEs can achieve a greater degree of rotational invariance than linear PCEs, we consider the special case of a rate $R = 1/2$ convolutional code with naturally mapped QPSK modulation. As noted in Example 18.10, linear PCEs are not capable of achieving 90° rotational invariance in this case. Before specifying a nonlinear PCE, we write the two encoded sequences in *integer* form as follows:

$$\mathbb{v}(D) = \mathbb{v}^{(0)}(D) + 2\mathbb{v}^{(1)}(D), \tag{18.41}$$

where the coefficients of $\mathbb{v}(D)$ are elements in the *ring* of integers $\mathbb{Z}_4$. (Throughout the remainder of this section, we will use the symbol $+$ to denote addition in a ring of integers and the symbol $\oplus$ to denote addition modulo-2, i.e., binary addition.) Using this formulation, we can express the effect of a 90° phase rotation on naturally mapped QPSK simply as

$$\mathbb{v}_r(D) = \mathbb{v}(D) + \mathbb{1}(D) \pmod 4. \tag{18.42}$$

Similarly, we can write the two parity-check polynomials in integer form as

$$\mathbb{h}(D) = \mathbb{h}^{(1)}(D) + 2\mathbb{h}^{(0)}(D). \tag{18.43}$$

Now, consider the PCE given by

$$[\mathbb{h}(D)\mathbb{v}(D) \pmod 4]^1 = \mathbb{0}(D), \tag{18.44}$$

where the notation $[\alpha(D)]^1$ means that from the binary representation of every element $\alpha_l = 2\alpha_l^{(1)} + \alpha_l^{(0)} \in \mathbb{Z}_4$ in $\alpha(D)$ the most significant bit $\alpha_l^{(1)}$ is chosen; that is, $[\alpha(D)]^1 = \alpha_l^{(1)}$. (The operation represented as $[\alpha(D)]^1$ in (18.44) causes this PCE to be both binary and nonlinear.) For the PCE represented by (18.44) to be invariant to 90° rotations, it must still be satisfied when $\mathbb{v}_r(D)$ is substituted for $\mathbb{v}(D)$, which requires that

$$\begin{aligned}\mathbb{h}(D)\mathbb{v}(D) = \mathbb{h}(D)\mathbb{v}_r(D) &= \mathbb{h}(D)\left[\mathbb{v}(D) + \mathbb{1}(D)\right] \pmod 4\\ &= \mathbb{h}(D)\mathbb{v}(D) + \mathbb{h}(D)\mathbb{1}(D) \pmod 4 \qquad (18.45)\\ &= \mathbb{h}(D)\mathbb{v}(D) + \mathbb{h}(1)(D) \pmod 4.\end{aligned}$$

Because $\mathbb{h}(1)(D)$ is a constant sequence with $\mathbb{h}(1) \in \mathbb{Z}_4$, (18.45) is satisfied if and only if

$$\mathbb{h}(1) \pmod 4 = \mathbb{h}^{(1)}(1) + 2\mathbb{h}^{(0)}(1) \pmod 4 = 0. \qquad (18.46)$$

Note, for example, that if $\mathbb{h}^{(1)}(D)$ has two delay terms, $\mathbb{h}^{(0)}(D)$ must have an odd number of delay terms to satisfy (18.46) (see Problem 18.20).

We can rewrite the PCE of (18.44) as

$$\begin{aligned}[\mathbb{h}(D)\mathbb{v}(D) \pmod 4]^1 &= \Big[\mathbb{h}^{(1)}(D)\mathbb{v}^{(0)}(D) + 2\mathbb{h}^{(1)}(D)\mathbb{v}^{(1)}(D)\\ &\quad + 2\mathbb{h}^{(0)}(D)\mathbb{v}^{(0)}(D) + 4\mathbb{h}^{(0)}(D)\mathbb{v}^{(1)}(D) \pmod 4\Big]^1\\ &= \Big\{\mathbb{h}^{(1)}(D)\mathbb{v}^{(0)}(D) + 2\Big[\mathbb{h}^{(1)}(D)\mathbb{v}^{(1)}(D)\\ &\quad + \mathbb{h}^{(0)}(D)\mathbb{v}^{(0)}(D)\Big] \pmod 4\Big\}^1\\ &= \mathbb{0}(D), \qquad (18.47)\end{aligned}$$

where we have simplified (18.47) by noting that the term $4\mathbb{h}^{(0)}(D)\mathbb{v}^{(1)}(D) \pmod 4 = \mathbb{0}(D)$. Following the restrictions of Figure 18.16(b) for good TCM code design leads us to search for codes with parity-check polynomials

$$\mathbb{h}^{(1)}(D) = h^{(1)}_{\nu-1}D^{\nu-1} + \cdots + h^{(1)}_2 D^2 + h^{(1)}_1 D \qquad (18.48a)$$

and

$$\mathbb{h}^{(0)}(D) = D^{\nu} + h^{(0)}_{\nu-1}D^{\nu-1} + \cdots + h^{(0)}_2 D^2 + h^{(0)}_1 D + 1 \qquad (18.48b)$$

that satisfy (18.46), thus guaranteeing 90° rotational invariance. We must then substitute these polynomials into (18.47) to specify the binary nonlinear PCE.

We now consider an example in which we choose $\mathbb{h}^{(1)}(D)$ to have only two delay terms and proceed to derive a general binary nonlinear PCE that guarantees 90° rotational invariance for rate $R = 1/2$ codes with naturally mapped QPSK.

EXAMPLE 18.11 Rotationally Invariant Rate $R = 1/2$ Codes for Naturally Mapped QPSK

Consider the choice

$$\mathbb{h}^{(1)}(D) = D^b + D^a, \qquad (18.49)$$

where $\nu > b > a > 0$. Note that in this case, since $\mathbb{h}^{(1)}(1) = 2 \pmod 4$, $\mathbb{h}^{(0)}(D)$ must have an odd number of nonzero terms to satisfy (18.46). Now, substituting (18.49) into (18.47) we obtain the PCE

$$\left\{(D^b + D^a)\mathbb{v}^{(0)}(D) + 2\left[(D^b + D^a)\mathbb{v}^{(1)}(D) + \mathbb{h}^{(0)}(D)\mathbb{v}^{(0)}(D)\right] \pmod 4\right\}^1 = 0(D). \tag{18.50}$$

To express (18.50) using binary (mod-2) arithmetic, we note that we can write the mod-4 sum of any two binary sequences $\mathbb{m}(D)$ and $\mathbb{n}(D)$ as

$$\mathbb{m}(D) + \mathbb{n}(D) \pmod 4 = \mathbb{m}(D) \oplus \mathbb{n}(D) + 2\mathbb{m}(D) \circ \mathbb{n}(D), \tag{18.51}$$

where $\mathbb{m}(D) \circ \mathbb{n}(D)$ represents the logical AND of the sequences $\mathbb{m}(D)$ and $\mathbb{n}(D)$. Using (18.51) repeatedly in (18.50) and recalling that $4\mathbb{m}(D) \pmod 4 = \mathbb{0}(D)$ for any binary sequence $\mathbb{m}(D)$, we obtain the binary PCE

$$\begin{aligned}&\Big\{(D^b \oplus D^a)\mathbb{v}^{(0)}(D) + 2\Big[D^b\mathbb{v}^{(0)}(D) \circ D^a\mathbb{v}^{(0)}(D) \oplus (D^b \oplus D^a)\mathbb{v}^{(1)}(D) \oplus \\ &\qquad \mathbb{h}^{(0)}(D)\mathbb{v}^{(0)}(D)\Big]\Big\}^1 \\ &\quad = D^b\mathbb{v}^{(0)}(D) \circ D^a\mathbb{v}^{(0)}(D) \oplus (D^b \oplus D^a)\mathbb{v}^{(1)}(D) \oplus \mathbb{h}^{(0)}(D)\mathbb{v}^{(0)}(D) = \mathbb{0}(D),\end{aligned} \tag{18.52}$$

where (18.52) has been simplified by using the fact that the term $(D^b \oplus D^a)\mathbb{v}^{(0)}(D)$ has no effect on the most significant bits of the sequence in braces. Equation (18.52) represents a binary nonlinear PCE that guarantees 90° rotational invariance for rate $R = 1/2$ codes with $\mathbb{h}^{(1)}(D)$ defined by (18.49) and any $\mathbb{h}^{(0)}(D)$ with an odd number of nonzero terms, where the nonlinearity is represented by the logical AND operation. (In Problem 18.21 it is shown that the preceding nonlinear PCE is satisfied when the rotated binary sequences given in (18.38) for natural mapping are substituted into (18.52), and $\mathbb{h}^0(D)$ is assumed to have an odd number of nonzero terms.)

The addition of the nonlinear term $D^b\mathbb{v}^{(0)}(D) \circ D^a\mathbb{v}^{(0)}(D)$ makes (18.52) different from a linear PCE for a rate $R = 1/2$ code. Given this difference, it is not clear if (18.52) will, in general, result in an encoder realization with only 2^ν states. Considering the specific example

$$\mathbb{H}(D) = \left[(D^2 + D)/(D^3 + D + 1) \;\; 1\right], \tag{18.53}$$

we show in Figure 18.21(a) an 8-state encoder realization in which the nonlinear term $D^2\mathbb{v}^{(0)}(D) \circ D\mathbb{v}^{(0)}(D)$ can be obtained directly from the feedback shift register that forms $\mathbb{v}^{(0)}(D)$; that is, no additional states are needed in this case. (The separate 2-state differential encoder for the information sequence is also shown in Figure 18.21(a).) This nonlinear encoder results in a 90° rotationally invariant code, since $\mathbb{h}^{(0)}(D) = D^3 + D + 1$ has an odd number of delay terms, and thus the rotated code sequences satisfy (18.52) (see Problem 18.22).

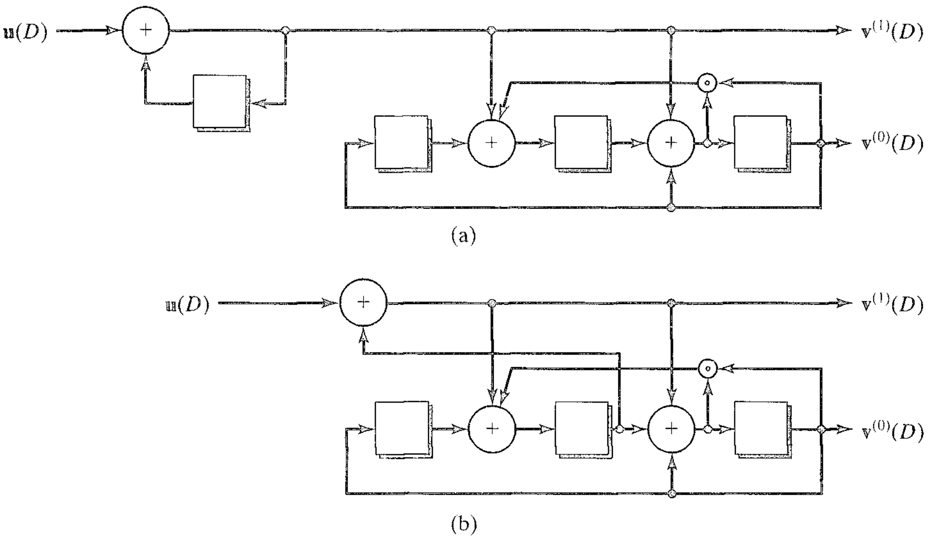


FIGURE 18.21: Realization of a rotationally invariant, 8-state, rate $R = 1/2$ QPSK encoder (a) with separate differential encoding and (b) with embedded differential encoding.

The following comments apply to Example 18.11:

- The nonlinear PCE represented by (18.52) can always be realized using $\nu' = \max[\nu, 2(b-a)]$ delay elements [26]. In this example, $\nu' = \nu = 3$ (see Problem 18.23 for an example that includes the case $\nu < 2(b-a)$).
- More classes of rotationally invariant codes can be found by dropping the condition of (18.49) and merely requiring that (18.46) be satisfied; however, in this more general case, it is not as easy to determine the conditions under which additional states are not required in the encoder realization.
- Natural mapping is assumed and (18.44) is used as the PCE because these choices result in simple conditions on the parity-check polynomials to guarantee rotational invariance and because if $\mathbb{h}^{(1)}(D)$ is chosen as in (18.49), the PCE contains only one nonlinear term. Additional classes of 90° rotationally invariant codes can be found if other mappings are assumed or different PCEs are used, but the code specification and realization is, in general, more complex.

For a given ν, a large family of 90° rotationally invariant rate $R = 1/2$ nonlinear codes for naturally mapped QPSK are defined by (18.52). A computer search can be used to select the parameters a and b and the coefficients of $\mathbb{h}^{(0)}(D)$ that satisfy the conditions of (18.46) and (18.49), maximize d^2_{free}, and minimize $A_{d_{free}}$. (The search technique cannot use the method of Euclidean weights to find d^2_{free} in this case, since the code is nonlinear: that is, all pairs of trellis paths must be compared.) A list of the best 90° rotationally invariant rate $R = 1/2$ nonlinear codes for naturally

TABLE 18.7: Rotationally invariant rate $R = 1/2$ QPSK codes.
$\eta = 1$ bit/symbol, $d^2_{min} = 4$, $A_{min} = 1$ (BPSK)

				90° Invariance		180° Invariance		360° Invariance		γ
ν	$\tilde{k}$	$\mathbb{h}^{(1)}$	$\mathbb{h}^{(0)}$	d^2_{free}	$A_{d_{free}}$	d^2_{free}	$A_{d_{free}}$	d^2_{free}	$A_{d_{free}}$	(dB)
3	1	06	13	10	0.5	12	2	12	1	3.98
4	1	06	23	12	0.5	12	1	14	2	4.77
5	1	30	45	14	1.0	16	2	16	1	5.44
6	1	050	105	16	1.875	20	11	—	—	6.02
7	1	110	217	16	0.25	20	2	20	1	6.02
8	1	220	427	18	0.312	24	11	24	9	6.53
9	1	0120	1017	20	0.75	24	2	24	1	6.99

Adapted from [26].

mapped QPSK found for constraint lengths up to $\nu = 9$ is presented in Table 18.7. The parity-check coefficients $\mathbb{h}^{(j)} = \left[h_\nu^{(j)}, h_{\nu-1}^{(j)}, \cdots, h_1^{(j)}, h_0^{(j)}\right]$, $j = 0, 1$, are given in octal form, as in Table 18.6. The values of d^2_{free} and $A_{d_{free}}$ for the best 180° and 360° rotationally invariant rate $R = 1/2$ linear codes are also listed for comparison, along with the asymptotic coding gain γ of the best nonlinear code. (The codes for 360° invariance are the optimum free distance codes found in Table 12.1 used with Gray mapping. The value of d^2_{free} given in Table 18.6 is twice as large as the value of d_{free} given in Table 12.1 because the QPSK signals have been normalized to unit energy. The missing entry for $\nu = 6$ means that the best 360° invariant code is identical to the best 180° invariant code.) The value of γ is computed with reference to an uncoded system (BPSK) with the same spectral efficiency $\eta = 1$ bit/symbol that has MSE distance $d^2_{min} = 4$ and number of nearest neighbors $A_{min} = 1$. Note that, in general, the best nonlinear codes have smaller values of d^2_{free} than the best linear codes, so there is some penalty to be paid in asymptotic coding gain to achieve full rotational invariance. On the other hand, the values of $A_{d_{free}}$ are larger for the linear codes, which lessens the suboptimality of the nonlinear codes at moderate BERs. (The values of $A_{d_{free}}$ are, in general, fractional in the nonlinear case, since not all codewords have a nearest neighbor at distance d^2_{free}.) For example, the best 8-state, 90° invariant nonlinear code (the code in Example 18.11) has $\gamma = 3.98$ dB, which is 0.79 dB less than the best 8-state, 180° invariant linear code; however, the nonlinear code has only one-fourth the number of nearest neighbors of the linear code. In Figure 18.22 we plot the simulated BER performance of these two codes. Note that the suboptimality of the nonlinear code is only about 0.3 dB at a BER of 10^{-5}.

Using the nonlinear 90° rotationally invariant codes described in this section guarantees that rotated code sequences are still valid paths through the trellis. For the coding system to operate properly (i.e., with only a small loss in decoded BER), even if the receiver has lost synchronization and locked onto the wrong phase, differential encoding and decoding must be employed. Thus, the input sequence $\mathbf{u}(D)$ must pass through the one time unit delay circuit $1/(D + 1)$ (shown in Figure 18.21(a)) prior to convolutional encoding, and the decoded sequence $\hat{\mathbf{u}}(D)$ must be processed by the inverse circuit $(D + 1)$ after decoding. For some rotationally invariant codes, it is

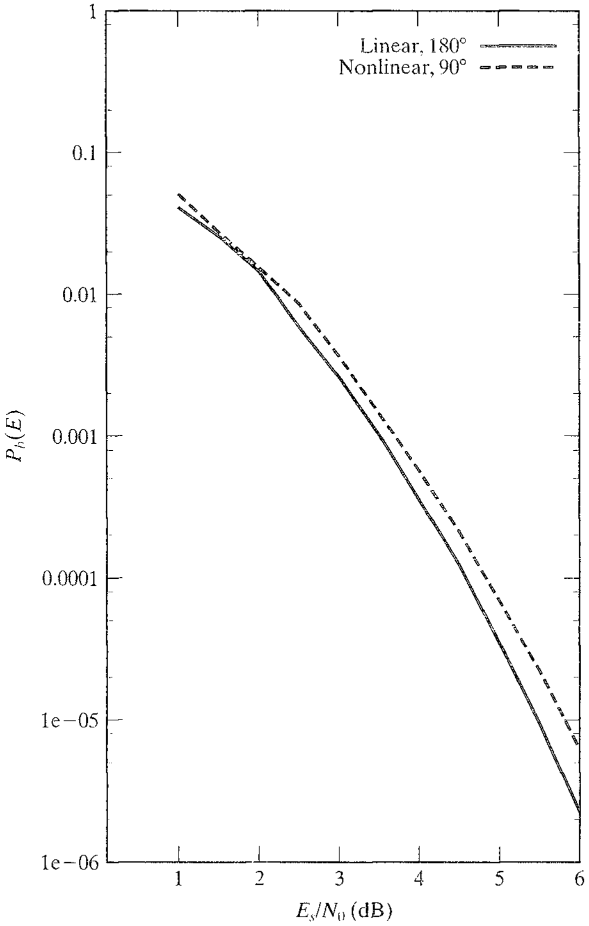


FIGURE 18.22: Performance curves for two 8-state, rate $R = 1/2$ QPSK codes.

possible to embed the differential encoding within the convolutional encoder, thus eliminating the need for separate differential encoding and decoding circuits. For the 90° rotationally invariant code of Example 18.11, an encoder realization that includes embedded differential encoding is shown in Figure 18.21(b).

Any QAM signal constellation has exactly the same four rotational symmetries as the QPSK signal set. Thus, the same approach used for rate $R = 1/2$ coded QPSK can be used to construct 90° rotationally invariant codes for rate $R = k/(k+1)$ coded 2^{k+1}-ary QAM constellations, as long as the two least significant label bits are assigned using natural mapping. In this case only the first information sequence $\mathbf{v}^{(1)}(D)$ and the parity sequence $\mathbf{v}^{(0)}(D)$ are affected by a rotation, so the check polynomials $\mathbb{h}^{(2)}(D), \cdots, \mathbb{h}^{(k)}(D)$ corresponding to the other coded information sequences $\mathbf{v}^{(2)}(D), \cdots, \mathbf{v}^{(k)}(D)$ can be chosen to maximize d^2_{free} and minimize $A_{d_{free}}$ without regard to the invariance constraints. Using the same conditions as in Example 18.11, we can write a general binary nonlinear PCE for rate $R = k/(k+1)$

coded QAM with $\tilde{k}$ coded information bits as

$$\begin{aligned}&\mathbb{h}^{(\tilde{k})}(D)\mathbb{v}^{(\tilde{k})}(D) \oplus \cdots \oplus \mathbb{h}^{(2)}(D)\mathbb{v}^{(2)}(D) \oplus (D^b \oplus D^a)\mathbb{v}^{(1)}(D) \oplus \mathbb{h}^{(0)}(D)\mathbb{v}^{(0)}(D)\\ &\quad = D^b\mathbb{v}^{(0)}(D) \circ D^a\mathbb{v}^{(0)}(D).\end{aligned} \tag{18.54}$$

Again, following the restrictions of Figure 18.16(b) for good TCM code design, we let the check polynomials $\mathbb{h}^{(2)}(D), \cdots, \mathbb{h}^{(\tilde{k})}(D)$ be denoted by

$$\mathbb{h}^{(j)}(D) = D^{b_j} + h^{(j)}_{b_j-1}D^{b_j-1} + \cdots + h^{(j)}_{a_j+1}D^{a_j+1} + D^{a_j}, \tag{18.55}$$

where $\nu > b_j \geq a_j > 0$ for $2 \leq j \leq \tilde{k}$. Further, letting

$$b' = \max\left(b_{\tilde{k}}, \cdots, b_2, b\right), \quad a' = \max\left(a_{\tilde{k}}, \cdots, a_2, a\right), \tag{18.56}$$

we can realize the nonlinear PCE represented by (18.54) using $\nu' = \max(\nu, b - a + b' - a')$ delay elements [26].

EXAMPLE 18.12 Rotationally Invariant Rate $R = 3/4$ Codes for Naturally Mapped 16-QAM with Two Coded Bits ($\tilde{k} = 2$)

Consider the following parity-check matrix for a rate $R = 2/3$ convolutional code:

$$\mathbb{H}(D) = [D/(D^3 + D + 1) \quad (D^2 + D)/(D^3 + D + 1) \quad 1], \tag{18.57}$$

in which $\mathbb{h}^{(1)}(D)$ and $\mathbb{h}^{(0)}(D)$ are chosen as in Example 18.11 to ensure 90° rotational invariance. In this case $\nu = 3, b = 2, a = 1$, and $b_2 = a_2 = 1$, so $b' = 2, a' = 1$, and $\nu' = \max(3, 2) = 3$. Hence, we can use this code along with one uncoded bit to provide a 90° rotationally invariant 8-state encoder realization of a rate $R = 3/4$ 16-QAM TCM system. The encoder realization (without differential encoding) is shown in Figure 18.23.

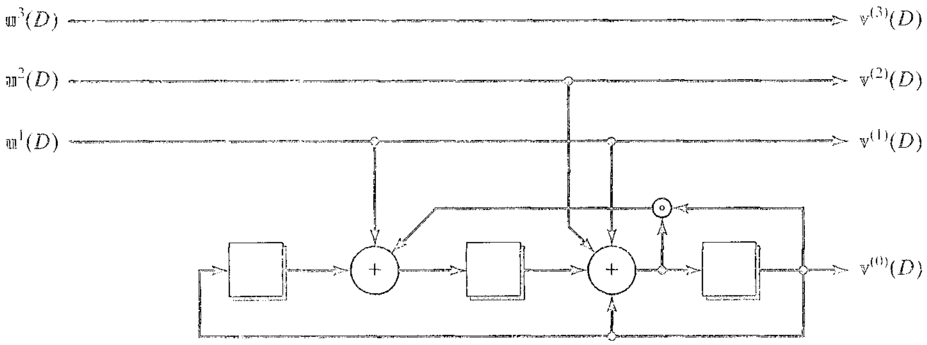


FIGURE 18.23: Realization of a rotationally invariant, 8-state, rate $R = 3/4$ 16-QAM encoder.

TABLE 18.8: Rotationally invariant rate $R = k/(k+1)$ QAM codes.
$\eta = k$ bits/symbol, $d^2_{min} = 2$, $A_{min} = 4(\mathbb{Z}^2)$

					90° Invariance		**180° Invariance**		**360° Invariance**		γ
ν	$\tilde{k}$	$\mathbb{h}^{(2)}$	$\mathbb{h}^{(1)}$	$\mathbb{h}^{(0)}$	d^2_{free}	$A_{d_{free}}$	d^2_{free}	$A_{d_{free}}$	d^2_{free}	$A_{d_{free}}$	**(dB)**
3	2	02	06	13	5	16	5	16	—	—	3.98
4	2	04	12	23	5	8	6	56	—	—	3.98
5	2	02	14	45	6	16	6	8	—	—	4.77
6	2	020	014	103	7	80	7	44	7	40	5.44
7	2	100	060	205	7	16	8	204	8	172	5.44
8	2	100	210	417	8	60	8	28	—	—	6.02

Adapted from [26].

Because (18.54) defines a large family of 90° rotationally invariant rate $R = k/(k+1)$ nonlinear codes for naturally mapped QAM signal constellations, a computer search can be used to select the parameters a and b and the coefficients of $\mathbb{h}^{(0)}(D)$ and $\mathbb{h}^{(2)}(D), \cdots, \mathbb{h}^{(\tilde{k})}(D)$ that satisfy the conditions of (18.46) and (18.49), maximize d^2_{free}, and minimize $A_{d_{free}}$. A list of the best 90° rotationally invariant rate $R = k/(k+1)$ nonlinear codes for naturally mapped QAM found for constraint lengths up to $\nu = 8$ is presented in Table 18.8, where the uncoded reference system is the (scaled) infinite two-dimensional integer lattice $\mathbb{Z}^2$ with $d^2_{min} = 2$ and $A_{min} = 4$. The values of d^2_{free} and $A_{d_{free}}$ for the best 180° and 360° rotationally invariant rate $R = k/(k+1)$ linear codes are also listed for comparison, along with the asymptotic coding gain γ of the best nonlinear code. In all cases the best nonlinear codes found had only two coded bits, that is, $\tilde{k} = 2$, and the same codes resulted in the same maximum d^2_{free} independent of the size of the signal constellation chosen from the lattice, that is, independent of the number of uncoded bits and the spectral efficiency $\eta = k$ bits/symbol. Because boundary effects cause the (in general, fractional) values of $A_{d_{free}}$ to differ depending on k and the size of the signal constellation, only the (integer) values of $A_{d_{free}}$ corresponding to an infinite-size constellation are listed in the table. Note that the best nonlinear codes have smaller values of d^2_{free} than the best linear codes in only two cases, namely, $\nu = 4$ and $\nu = 7$, and that in the other cases the only penalty to be paid for full rotational invariance is a somewhat larger value of $A_{d_{free}}$. In fact, the best 8-state 90° invariant nonlinear code (the code in Example 18.12) has exactly the same parameters as the best 8-state 180° invariant linear code. In Figure 18.24 we plot the simulated BER performance of these two codes with one uncoded bit and 16-QAM, where we see that the 90° invariant nonlinear code is actually slightly better than the 180° invariant linear code (owing to the effect of higher-order terms in the distance spectrum of the two codes). Finally, we note that for larger values of ν, more than $\tilde{k} = 2$ coded bits will be needed to achieve the maximum d^2_{free}.

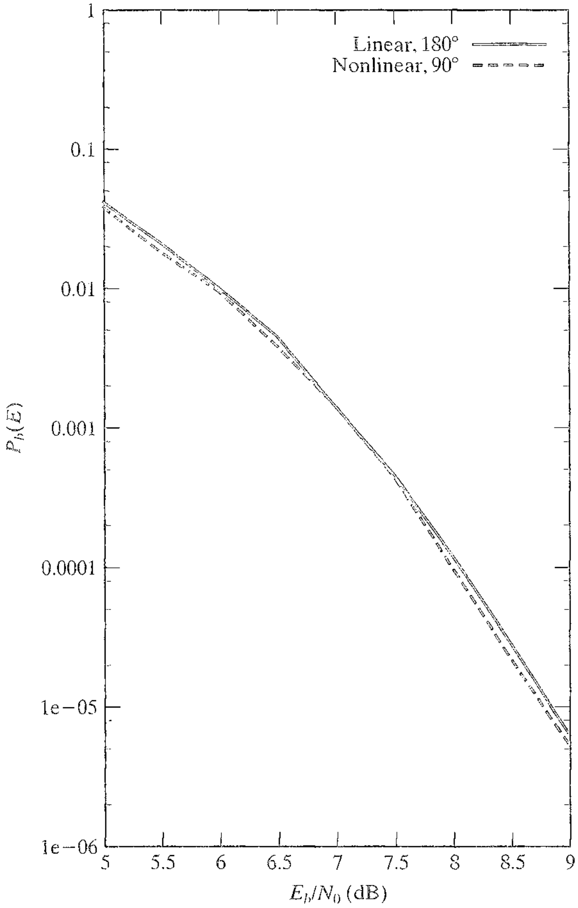


FIGURE 18.24: Performance curves for two 8-state, rate $R = 3/4$ 16-QAM codes.

We now sketch the development of 45° rotationally invariant rate $R = 2/3$ codes for naturally mapped 8-PSK modulation by following the same approach used for the 90° invariant rate $R = 1/2$ QPSK case. We begin by considering the PCE

$$[\mathbb{h}(D)\mathbb{v}(D) \pmod 8)]^2 = \mathbb{0}(D), \tag{18.58}$$

where $\mathbb{h}(D) = \mathbb{h}^{(2)}(D) + 2\mathbb{h}^{(1)}(D) + 4\mathbb{h}^{(0)}(D)$, $\mathbb{v}(D) = \mathbb{v}^{(0)}(D) + 2\mathbb{v}^{(1)}(D) + 4\mathbb{v}^{(2)}(D)$, addition is performed in the ring of integers $\mathbb{Z}_8$, and the notation $[\alpha(D)]^2$ means that from the binary representation of every element $\alpha_l = 4\alpha_l^{(2)} + 2\alpha_l^{(1)} + \alpha_l^{(0)} \in \mathbb{Z}_8$ in $\alpha(D)$ the most significant bit $\alpha_l^{(2)}$ is chosen; that is, $[\alpha(D)]^2 = \alpha_l^{(2)}$. For the PCE represented by (18.58) to be invariant to 45° rotations, we require that

$$\mathbb{h}(1) \pmod 8) = \mathbb{h}^{(2)}(1) + 2\mathbb{h}^{(1)}(1) + 4\mathbb{h}^{(0)}(1) \pmod 8) = 0, \tag{18.59}$$

so that, for example, if $\mathbb{h}^{(2)}(D)$ has two nonzero terms, and $\mathbb{h}^{(1)}(D)$ has one nonzero term, $\mathbb{h}^{(0)}(D)$ must have an odd number of nonzero terms to satisfy (18.59) (see Problem 18.24). Now, choosing

$$\mathbb{h}^{(2)}(D) = D^c + D^b, \tag{18.60a}$$

$$\mathbb{h}^{(1)}(D) = D^a, \tag{18.60b}$$

and

$$\mathbb{h}^0(D) = D^\nu + h_{\nu-1}^{(0)} D^{\nu-1} + \cdots + h_2^{(0)} D^2 + h_1^{(0)} D + 1, \tag{18.60c}$$

where $\nu > c > b > a > 0$, and substituting (18.60a) and (18.60b) into (18.58) we obtain the rate $R = 2/3$ binary nonlinear PCE

$$\left(D^c \oplus D^b\right) \mathbb{v}^{(2)}(D) \oplus D^a \mathbb{v}^{(1)}(D) \oplus f(D) \oplus \mathbb{h}^{(0)}(D) \mathbb{v}^{(0)}(D) = \mathbb{0}(D), \tag{18.61a}$$

where

$$\begin{aligned} f(D) = {} & D^c \mathbb{v}^{(1)}(D) \circ \left[D^b \mathbb{v}^{(1)}(D) \oplus D^a \mathbb{v}^{(0)}(D) \oplus D^c \mathbb{v}^{(0)}(D) \odot D^b \mathbb{v}^{(0)}(D)\right] \\ & \oplus D^b \mathbb{v}^{(1)}(D) \odot \left[D^a \mathbb{v}^{(0)}(D) \oplus D^c \mathbb{v}^{(0)}(D) \odot D^b \mathbb{v}^{(0)}(D)\right] \\ & \oplus D^c \mathbb{v}^{(0)}(D) \odot D^b \mathbb{v}^{(0)}(D) \odot D^a \mathbb{v}^{(0)}(D). \end{aligned} \tag{18.61b}$$

Note that in this case, since $\mathbb{h}^{(2)}(1) = 2 \pmod 8$, and $\mathbb{h}^{(1)}(1) = 1 \pmod 8$, $\mathbb{h}^{(0)}(D)$ must have an odd number of nonzero terms to satisfy (18.59). Equation (18.61) represents a binary nonlinear PCE that guarantees 45° rotational invariance for rate $R = 2/3$ codes with $\mathbb{h}^{(2)}(D)$ and $\mathbb{h}^{(1)}(D)$ defined by (18.60a) and (18.60b), respectively, and any $\mathbb{h}^{(0)}(D)$ with an odd number of nonzero terms, where $f(D)$ represents the nonlinear portion of the PCE. (In Problems 18.25 and 18.26 it is shown that the preceding nonlinear PCE is satisfied when the 45° rotated binary code sequences for naturally mapped 8-PSK are substituted into (18.61), and $\mathbb{h}^{(0)}(D)$ is assumed to have an odd number of nonzero terms.)

As in the case of the rate $R = 1/2$ binary nonlinear PCE used to guarantee 90° rotational invariance for QAM constellations, certain conditions must be satisfied by the rate $R = 2/3$ binary nonlinear PCE of (18.61) for the encoder to be realized with only ν delay elements [26]. Let $h_t^{(0)}$ be the lowest-order nonzero coefficient in $\mathbb{h}^{(0)}(D)$, that is,

$$\mathbb{h}^{(0)}(D) = D^\nu + \cdots + h_t^{(0)} D^t + 1, \tag{18.62}$$

where $1 \le t \le \nu - 1$. Then, the following four conditions are required to realize (18.61) with ν delay elements:

$$(i)\ t \ge c - b, \quad (ii)\ b = 2a, \quad (iii)\ c \le 3a, \quad (iv)\ \nu \ge 2(c - a). \tag{18.63}$$

EXAMPLE 18.13 Rotationally Invariant Rate $R = 2/3$ Codes for Naturally Mapped 8-PSK with Two Coded Bits ($\tilde{k} = 2$)

Consider the following parity-check matrix for a rate $R = 2/3$ convolutional code:

$$\mathbb{H}(D) = [(D^3 + D^2)/(D^4 + D + 1) \quad D/(D^4 + D + 1) \quad 1], \tag{18.64}$$

in which $\mathbb{h}^{(2)}(D)$ and $\mathbb{h}^{(1)}(D)$ are chosen according to (18.60), and $\mathbb{h}^{(0)}(D)$ has an odd number of nonzero terms to ensure 45° rotational invariance. In this case $\nu = 4, c = 3, b = 2, a = 1$, and $t = 1$, so that the conditions of (18.63) are all satisfied, and the encoder can be realized with $\nu = 4$ delay elements. From (18.61) the PCE is given by

$$\left(D^3 \oplus D^2\right)\mathbb{v}^{(2)}(D) \oplus D\mathbb{v}^{(1)}(D) \oplus f(D) \oplus \left(D^4 + D + 1\right)\mathbb{v}^{(0)}(D) = \mathbb{0}(D). \quad (18.65a)$$

where

$$\begin{aligned} f(D) = {} & D^3\mathbb{v}^{(1)}(D) \circ \left[D^2\mathbb{v}^{(1)}(D) \oplus D\mathbb{v}^{(0)}(D) \oplus D^3\mathbb{v}^{(0)}(D) \circ D^2\mathbb{v}^{(0)}(D)\right] \\ & \oplus D^2\mathbb{v}^{(1)}(D) \circ \left[D\mathbb{v}^{(0)}(D) \oplus D^3\mathbb{v}^{(0)}(D) \circ D^2\mathbb{v}^{(0)}(D)\right] \\ & \oplus D^3\mathbb{v}^{(0)}(D) \circ D^2\mathbb{v}^{(0)}(D) \circ D\mathbb{v}^{(0)}(D). \end{aligned} \quad (18.65b)$$

This PCE can be used to provide a 45° rotationally invariant 16-state encoder realization of a rate $R = 2/3$ 8-PSK TCM system. The encoder realization (without differential encoding) is shown in Figure 18.25. (In Problem 18.27 we see how $f(D)$ can be rewritten to correspond to the encoder realization shown in Figure 18.25.)

For polynomials $\mathbb{h}^{(0)}(D)$ with an odd number of nonzero terms, (18.60) defines a family of 45° rotationally invariant rate $R = 2/3$ nonlinear codes for naturally mapped 8-PSK signal constellations, and a computer search can be used to select the parameters a, b, and c and the coefficients of $\mathbb{h}^{(0)}(D)$ that satisfy the conditions of (18.62) and (18.63), maximize d^2_{free}, and minimize $A_{d_{free}}$. A list of the best 45° rotationally invariant rate $R = 2/3$ nonlinear codes for naturally mapped 8-PSK found for constraint lengths up to $\nu = 8$ is presented in Table 18.9, where the reference system is uncoded QPSK with $d^2_{min} = 2$ and $A_{min} = 2$. The values of d^2_{free} and $A_{d_{free}}$ for the best 180° and 360° rotationally invariant rate $R = 2/3$ linear codes are also listed for comparison, along with the asymptotic coding gain γ of the best

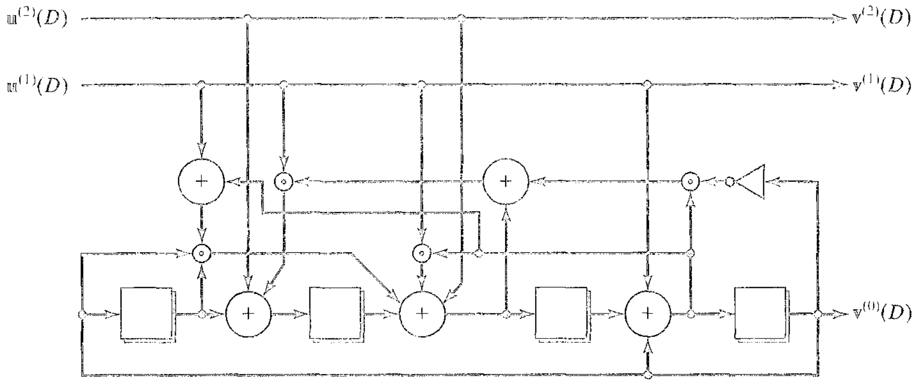


FIGURE 18.25: Realization of a rotationally invariant, 16-state, rate $R = 2/3$ 8-PSK encoder.

TABLE 18.9: Rotationally invariant rate $R = 2/3$ 8-PSK codes.
$\eta = 2$ bits/symbol, $d^2_{min} = 2$, $A_{min} = 2$ (QPSK)

					45° Invariance		180° Invariance		360° Invariance		γ
ν	$\tilde{k}$	$\mathbf{h}^{(2)}$	$\mathbf{h}^{(1)}$	$\mathbf{h}^{(0)}$	d^2_{free}	$A_{d_{free}}$	d^2_{free}	$A_{d_{free}}$	d^2_{free}	$A_{d_{free}}$	(dB)
3	1	—	06	13	4.0	1.0	4.586	2.0	—	—	3.01
4	2	14	02	23	4.586	1.0	5.172	4.0	5.172	2.25	3.60
5	2	14	02	43	4.586	0.25	5.172	0.25	5.757	2.0	3.60
6	2	060	004	127	5.172	0.469	6.343	3.25	—	—	4.13
7	2	014	002	235	5.172	0.012	6.343	0.125	6.586	0.5	4.13
8	2	120	004	721	5.757	0.016	7.515	3.375	7.515	1.5	4.59

Adapted from [26].

nonlinear code. For $\nu = 3$, the best nonlinear code has $\tilde{k} = 1$ and one uncoded bit, but for all $\nu \geq 4$, the best nonlinear code has $\tilde{k} = 2$ and no uncoded bits. The nonlinear codes have smaller values of d^2_{free} than the best linear codes, indicating that a penalty must be paid for full 45° rotational invariance; however, the nonlinear codes generally have smaller values of $A_{d_{free}}$ than the best linear codes. For example, the best 16-state 45° invariant nonlinear code (the code in Example 18.13) loses 0.52 dB in asymptotic coding gain compared with the best 16-state 180° invariant linear code. In Figure 18.26 we plot the simulated BER performance of these two codes with naturally mapped 8-PSK modulation, and we see that the 45° invariant nonlinear code is only about 0.15 dB worse than the 360° invariant linear code at a BER of 10^{-5} (owing mostly to the fact that the nonlinear code has a factor of 4 fewer nearest neighbors than the linear code).

The following comments apply to Example 18.13:

- A similar approach to that used for 8-PSK, but using modulo-16 arithmetic over the ring of integers $\mathbb{Z}_{16}$, can be used to find 22.5° rotationally invariant codes for naturally mapped 16-PSK. In this case the best codes up to $\nu = 7$ use the rate $R = 1/2$ invariant PCE with two uncoded bits, but for larger values of ν higher-rate PCEs are better [26]. As noted in Section 18.2, this is because for short constraint lengths, the parallel transition distance ($\delta^2_{min} = 2$ in this case) is already larger than the free distance δ^2_{free} between trellis paths, and thus using a higher-rate PCE cannot improve the overall free distance d^2_{free}.

- As in the QPSK case, additional classes of fully rotationally invariant codes can be found for PSK and QAM constellations if alternative mappings are assumed, different PCEs are used, or other restrictions are placed on the parity-check polynomials, but the code specification and realization is, in general, more complex.

We close this section with an example of the 8-state, rate $R = 2/3$, nonlinear, 90° invariant code designed by Wei [27, 28] and chosen for the V.32 and V.33 high-speed modem standards. The V.32 standard uses 2 uncoded bits and a 32-CROSS signal constellation for a spectral efficiency of $\eta = 4.0$ bits/symbol. In the V.33

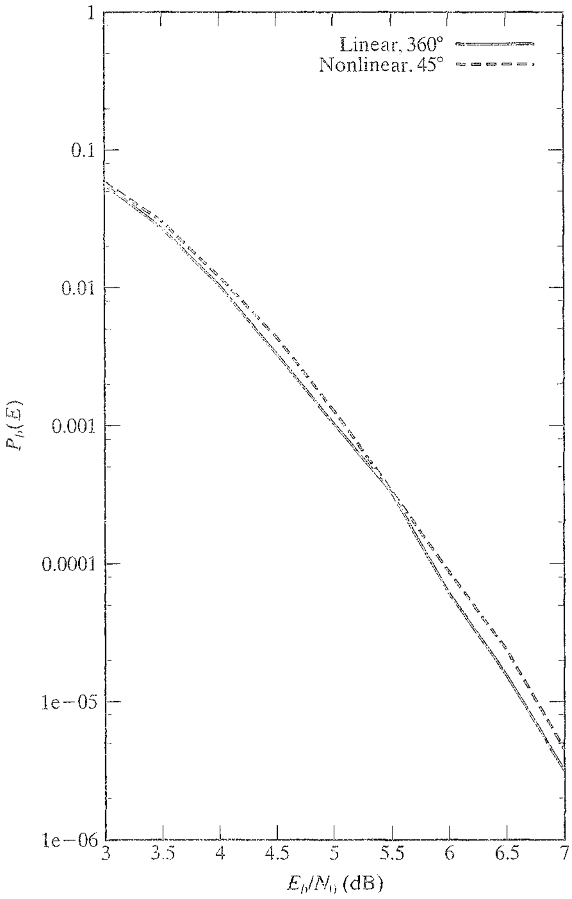
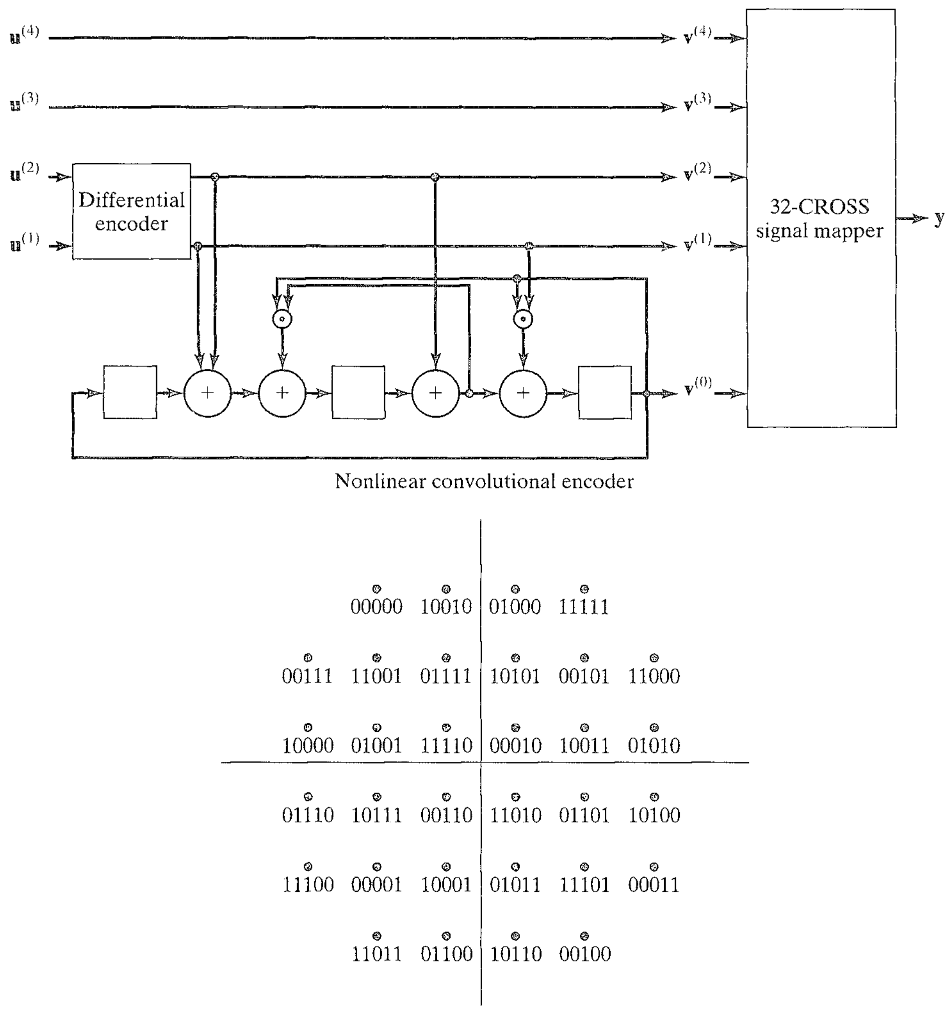


FIGURE 18.26: Performance curves for two 16-state, rate $R = 2/3$ 8-PSK codes.

standard, 4 uncoded bits and a 128-CROSS constellation are used to achieve $\eta = 6.0$ bits/symbol.

EXAMPLE 18.14 The V.32 90° Rotationally Invariant TCM System

A block diagram of the 8-state, rate $R = 2/3$, nonlinear, 90° invariant encoder and the 32-CROSS constellation used in the V.32 standard is shown in Figure 18.27. (Note that the 32-CROSS constellation is not naturally mapped, since a 90° rotation of a signal point does not alter the two least significant label bits in the same way as naturally mapped QPSK.) The encoder has four input information bits, $u^{(1)}$, $u^{(2)}$, $u^{(3)}$, and $u^{(4)}$. Bits $u^{(3)} = v^{(3)}$ and $u^{(4)} = v^{(4)}$ are uncoded and directly enter the 32-CROSS signal mapper (modulator). Bits $u^{(1)}$ and $u^{(2)}$ are first differentially encoded and then enter the 8-state, rate $R = 2/3$ systematic feedback nonlinear convolutional encoder, producing the three output bits $v^{(1)}$ and $v^{(2)}$ (information bits) and $v^{(0)}$ (a parity bit). The five encoded bits $v^{(0)}$, $v^{(1)}$, $v^{(2)}$, $v^{(3)}$, and $v^{(4)}$ then enter the modulator and are mapped into one of the 32-CROSS signals according



Nonlinear convolutional encoder

Signal constellation

FIGURE 18.27: The V.32 TCM system encoder and signal constellation.

to the mapping shown in Figure 18.27. Because one 32-CROSS signal is transmitted for every four information bits entering the encoder, the spectral efficiency of the code is $\eta = 4.0$ bits/symbol. (The V.33 standard uses the same code along with four uncoded information bits and a 128-CROSS constellation to achieve a spectral efficiency of $\eta = 6.0$ bits/symbol.) At the receiver, soft-decision Viterbi decoding, using an 8-state trellis with 4-fold (16-fold in the V.33 case) parallel transitions, is performed based on the noisy received symbols at the demodulator output. After Viterbi decoding, the decoded output bits $u^{(1)}$ and $u^{(2)}$ are differentially decoded.

The 8-state, rate $R = 2/3$ nonlinear encoder used in the V.32 and V.33 standards was designed completely by hand, without benefit of a systematic code search [27, 28]. It is invariant to 90° phase rotations of the 32-CROSS constellation

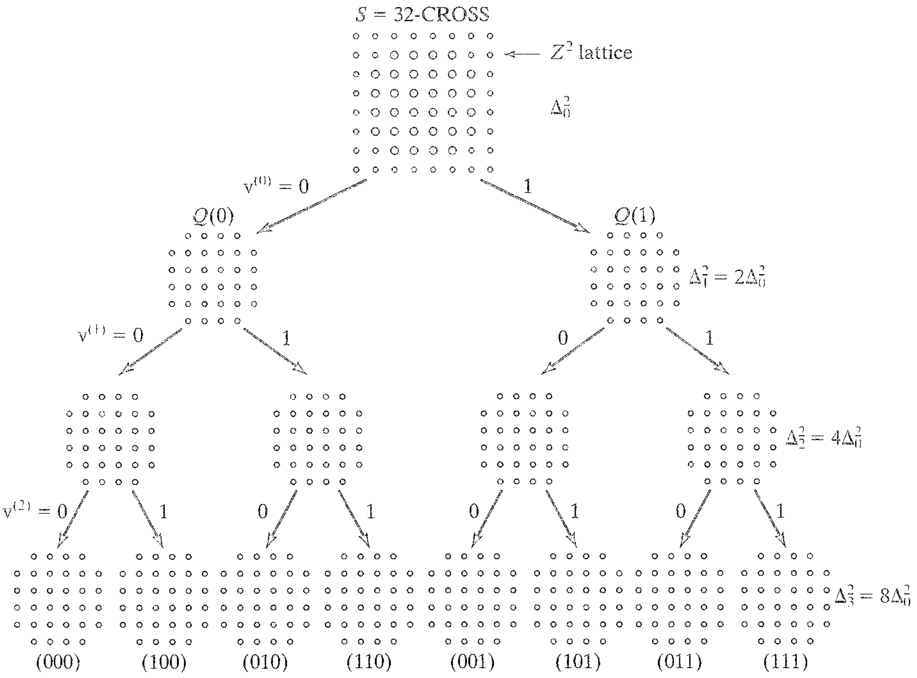


FIGURE 18.28: Three-level partitioning of the naturally mapped 32-CROSS signal constellation.

shown in Figure 18.27, has free distance $d^2_{free} = 5$ and average number of nearest neighbors $A_{d_{free}} = 6.716$, and achieves a real coding gain of 3.6 dB at a BER of 10^{-5} compared with uncoded 16-QAM ($\eta = 4.0$) and 64-QAM ($\eta = 6.0$), respectively, without bandwidth expansion. (The fractional value of $A_{d_{free}}$ is due to the nonlinearity of the code and the boundary effects of the constellation.)

An equivalent encoder, described in Example 18.12 and sketched in Figure 18.23, was designed using a systematic code search. When this encoder is used with the naturally mapped 32-CROSS constellation (see Figure 18.28), it requires only one AND gate and one differentially encoded information bit, and differential encoding can be embedded within the encoder. (Note, as mentioned in Section 18.2, that level 3 in the partition tree is an example of a case in which not all subsets at the same level are isomorphic.)

For two-dimensional signal constellations, since it is impossible to achieve 90° invariance with linear codes (the best that can be done is 180° invariance), nonlinear codes are needed for full rotational invariance. This was the crucial insight made by Wei [28] in the design of the V.32 code.

18.5 MULTIDIMENSIONAL TCM

Up to this point in our discussion of TCM we have considered only the case in which the $(k+1)$ convolutional encoder output bits at each time unit are mapped into one