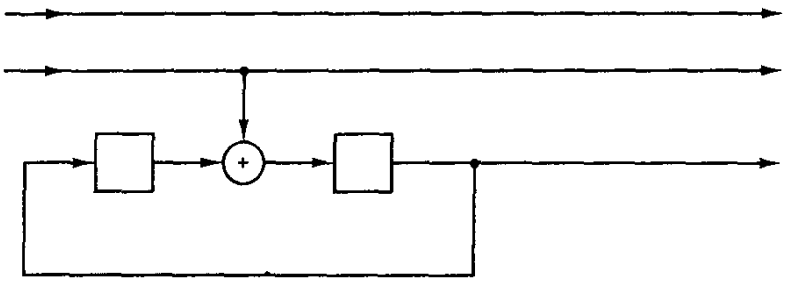


Figure 6.6.8 Four-state encoder for 8-PSK in systematic form with feedback, $h^0(D) = 1 + D^2$, $h^1(D) = D$, $h^2(D) = 0$.

Figure 6.6.8 shows the diagram of this encoder, with $h^0(D)$ providing the feedback connections, and the remaining polynomials specifying connection of the input bit lines to the various adders. The state variables are the contents of the memory cells, and it can be seen that any state may transition in one step to two next-states. The modulator input bits b_3 and b_2 are defined directly by the two message bits. The subset specification is provided by (b_1, b_2), and this generation process has memory on only one of the input bits.

Usually, the polynomials $h^i(D)$ are specified in octal notation for compactness and would be $h^0 = 5_8$, $h^1 = 2_8$ and $h^2 = 0_8$ in this case.

In Tables 6.12 and 6.13, information is summarized for best codes for 8-PSK and QAM with $S = 4$ to $S = 256$ states. The data are taken from [4] and [58].

TABLE 6.12 CODE SUMMARY FOR CODED 8-PSK

ν	$\tilde{k}$	$h^0(D)$	$h^1(D)$	$h^2(D)$	d_f^2/E_b	ACG, dB
2	1	5	2	—	4.0	3.0
3	2	11	2	4	4.58	3.6
4	2	23	4	16	5.17	4.1
5	2	45	16	34	5.76	4.6
6	2	105	36	74	6.00	4.8
7	2	203	14	16	6.34	5.0
8	2	405	250	176	6.93	5.4

ACG is relative to uncoded 4-PSK.

TABLE 6.13 CODE SUMMARY FOR CODED 16-QAM

ν	$\tilde{k}$	$h^0(D)$	$h^1(D)$	h^2 (D)	d_f^2/E_b	ACG, dB
2	1	5	2	—	4.80	4.4
3	2	11	2	4	6.00	5.3
4	2	23	4	16	7.20	6.1
5	2	45	16	34	7.20	6.1
6	2	105	36	74	8.40	6.8
7	2	203	14	16	9.60	7.4
8	2	405	250	176	9.60	7.4

ACG is relative to uncoded 8-PSK.

Codes are also optimal for higher throughput cases by using additional uncoded bits.

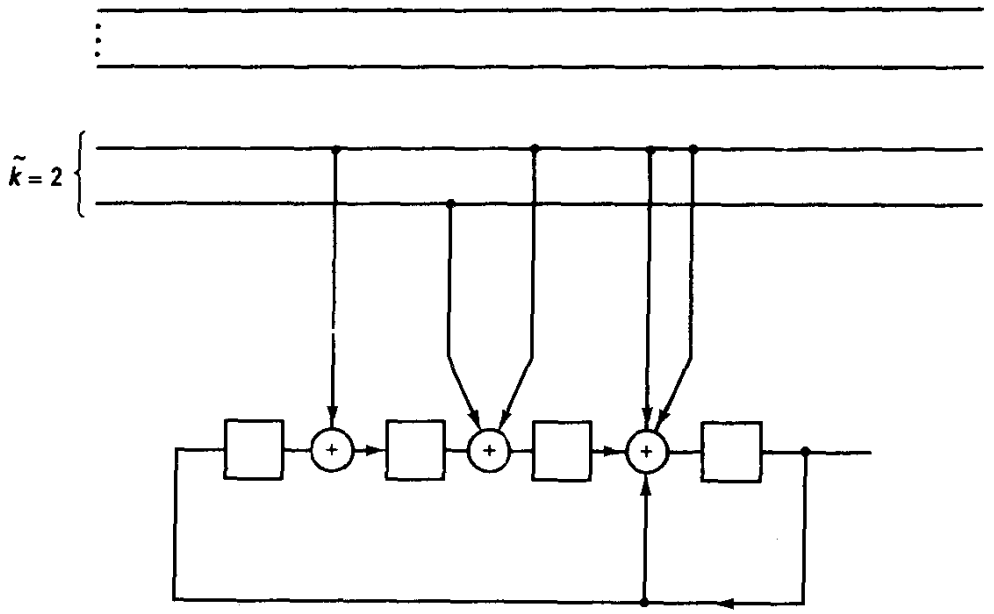


Figure 6.6.9 Sixteen-state encoder for QAM constellations. $h^0(D) = 23_8$, $h^1 = 4_8$, $h^2 = 16_8$, other $h^i = 0$.

To illustrate the use of these tables, Figure 6.6.9 shows the optimal 16-state encoder for QAM. Notice that two of the input bits now influence the state vector. The asymptotic coding gain is 6.1 dB over an uncoded technique having the same spectral efficiency, 8-PSK.

TCM for 16-PSK constellations has been studied as a means of achieving higher spectral efficiency with constant envelope transmission. Due to the special growth of squared distance within subsets when 16-PSK is partitioned, the optimal codes differ from those found for 8-PSK. A summary of these codes is found in [4] and [59].

In the matter of power spectrum, we earlier suggested that the power spectrum is unchanged if we simply enlarge the modulator constellation and keep the dimensionality per bit as it would be for an uncoded system. Biglieri [60] has shown in fact that for Ungerboeck-style TCM, that is, set-partitioned constellations preceded by convolutional encoders, the power spectrum is in fact unaltered. This clearly has much to do with the symmetries of the signaling process.

Slightly better opportunities for good trellis codes exist when the constellation is a multidimensional constellation, say a lattice in N-dimensional space [58] or an L-fold product of M-PSK designs [61]. The conceptual nature of the codes is similar: k bits enter the encoder per trellis level, and some bits become part of the state vector while others are uncoded. The encoder produces a signal point from an N-dimensional constellation.[13] To correctly assess a given technique, particularly against two-dimensional coded modulation, we must measure the information throughput per signal space dimension, and also calculate free distance in terms of energy per information bit. It is also important to attempt to measure the decoder complexity fairly. One way of doing so is to

[13] This may be realized by sending multiple two-dimensional symbols perhaps.

count the number of decoder branches per decoded bit, without regard to dimensionality, for this registers the number of metric updates, metric comparisons, and the like, per information bit in the decoder. Interested readers are referered to [58] for precise details.

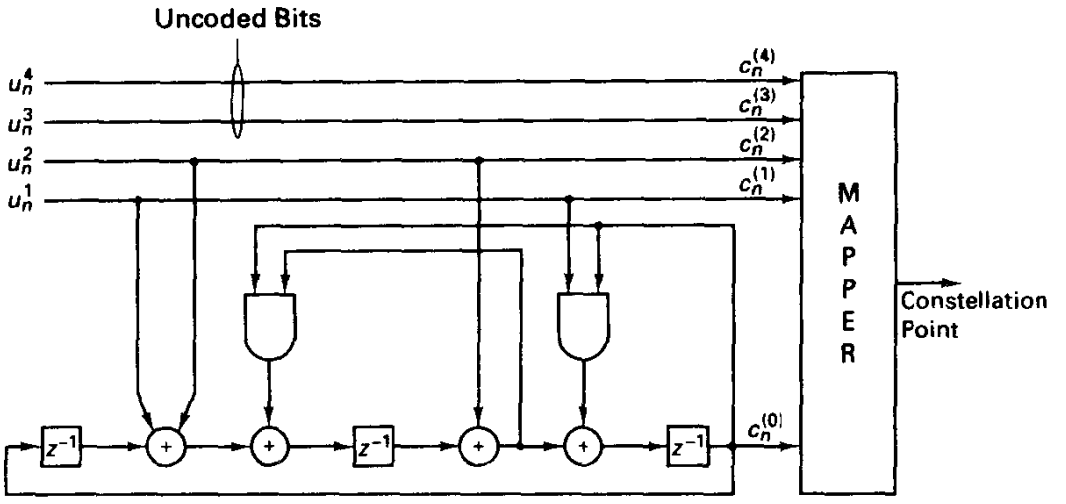
One benefit of higher-dimensional constellations is ease in obtaining ***rotational invariance***, a property that allows the carrier phase synchronization process to leave so-called phase ambiguities, for example, the acquired phase could be in-phase with that of the modulator or 180 degrees out-of-phase. Rotational invariance also aids in fast reacquisition once noise or jitter has caused the carrier synchronization loop to momentarily lose lock. The requirement is that for any signal pattern producible by the TCM system it must be true that a rotated version of this sequence is also a valid code sequence. Differential encoding ultimately is required to resolve the ambiguity.

Wei [62] presents trellis codes for two or more dimensions that achieve 0/90/180/270 degree rotational invariance in carrier phase synchronization. It is typically the case that attaining rotational invariance requires some sacrifice in coding gain, at a given state complexity. One case where this proved not to be true is the case of eight states with coded QAM. Wei was able to produce a rotationally invariant code with the same free distance as the best Ungerboeck 2-D code; the encoder is nonlinear, that is, it contains AND gates in addition to mod-2 adders and delay cells. This coding scheme ultimately became part of a CCITT standard V.32 for high-speed voice-band modems. Figure 6.6.10 presents the encoder and constellation in this standard.

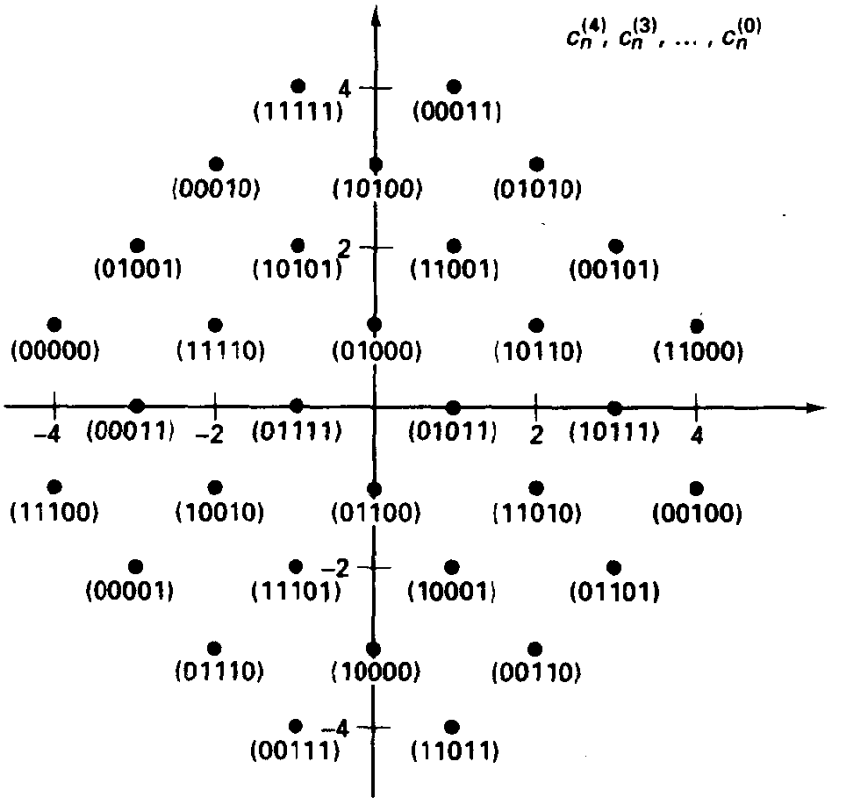
Figure 6.6.11 shows the concept behind a four-dimensional, 16-state TCM scheme developed by Wei [62], mapping 12 input bits onto a constellation from $\mathbf{Z}^4$ having $2^{13} = 8192$ points. The system would have the same spectral efficiency as uncoded 64-QAM (6 bits/two-dimensional symbol), yet yields several decibels of coding gain over 64-QAM. Calderbank and Sloane [63] discuss a similar eight-dimensional trellis code. Pietrobon et al. [61] discuss rotationally invariant L-dimensional trellis codes for PSK and show that higher dimensional signal sets are useful in obtaining this invariance property.

Performance evaluation of trellis codes for the Gaussian channel follows similar lines as that done for binary convolutional codes coupled with binary modulation, discussed earlier. There is, however, one important difference related to the general lack of the complete invariance of performance to transmitted sequence that is present in the binary convolutional code case. With trellis codes, it *may* turn out that some transmitted sequences in signal space are more prone to error than others, that is, have closer nearest-neighbor sequences, or the multiplicities of neighbors differ. It turns out that, at least with regard to free distance, Ungerboeck-style codes (binary convolutional codes coupled to a set partitioned constellation) have the usual invariance property, and any trellis path can be adopted as the reference for free-distance analysis. In general, precise bounding of the error probability must resort to a generalized transfer function bounding approach, which correctly enumerates both nearest neighbor sequences in a trellis code design, as well as numbers of bit errors on these error events. Zehavi and Wolf [64] introduced this technique, and Benedetto [65] has summarized the performance evaluation issue in detail, analyzing different symmetry properties of trellis code designs.

Representing state-of-the art in trellis coding for the bandlimited Gaussian channel is the V.34 modem standard, called V.Fast prior to standardization. The method is described in [66] and invokes numerous advanced features, including multidimensional trellis-coded modulation. Other items include constellation shaping, transmitter/receiver



(a)



(b)

Figure 6.6.10 V.32 coding format, 8-state trellis code mapping 4 bits/interval to 32-QAM constellation. (a) Encoder and mapper; (b) constellation.

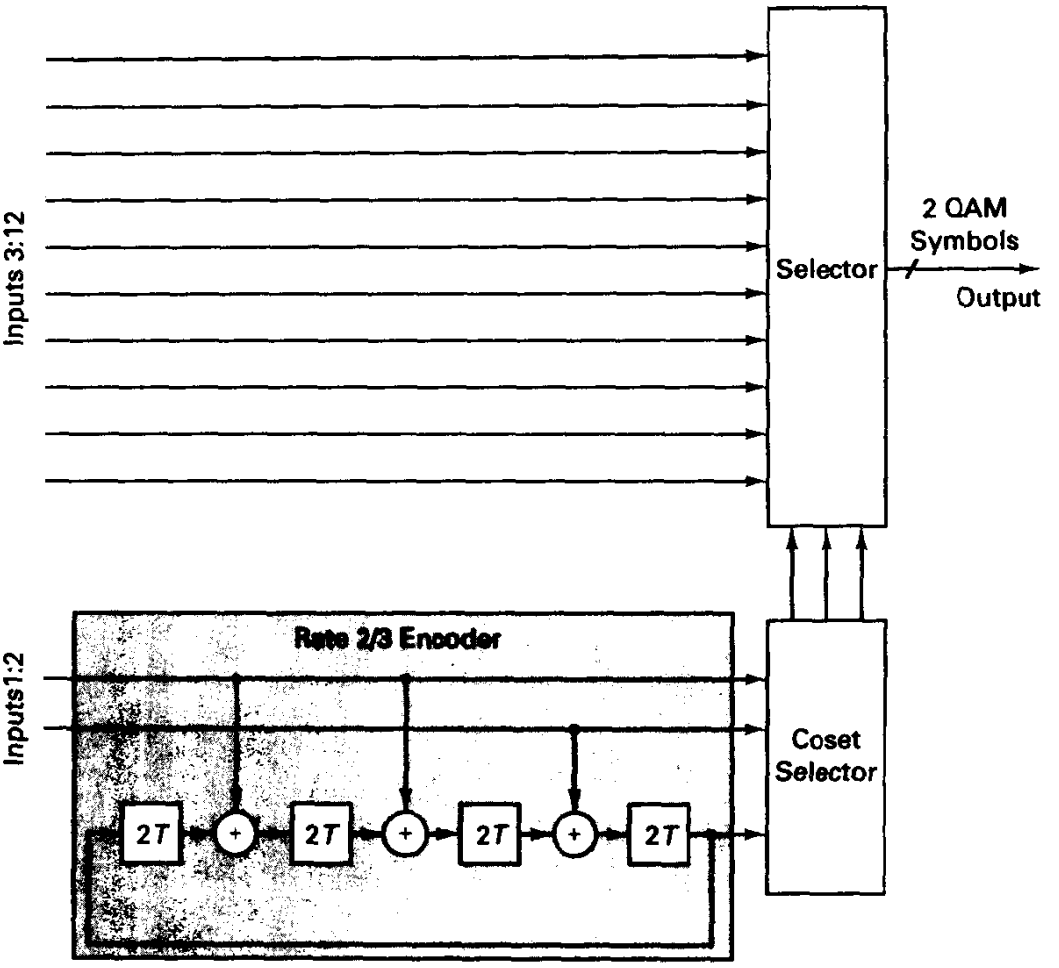


Figure 6.6.11 Wei's 16-state four-dimensional trellis code mapping 12 bits to 8192 points in Z^4.

"handshaking" for optimization, and transmitter pre-coding for receiver equalization of dispersive channel effects. On certain voice-band channels, this standard allows transmission at a rate of 28.8 kbps. Automatic fallback to lower speeds is accomplished on degraded channels.

Finally, readers are referred to the work of Forney [58, 67] which greatly unifies trellis-coded modulation theory. Lately, more general algebraic systems of rings and groups have been invoked to design, analyze, and unify TCM systems.

6.6.3 Trellis Codes for Fading Channels

We have recently emphasized the importance of designing codes so that free Euclidean distance is maximized. This is appropriate for the nonfading, band-limited Gaussian channel. On the other hand, when channel fading is introduced and symbol interleaving and deinterleaving are employed to render the channel effectively memoryless, the design reverts to a somewhat more traditional objective, maximizing Hamming symbol distance.

We will study the case of Rayleigh fading in detail and consider only the case of coherent detection with perfect side information on channel phase and gain. This is admittedly problematic, but it is indicative of design rules for, say, differentially coherent detection without side information.

We begin by writing a union bound for the node error probability in a trellis decoder:

$$P_e \leq \sum_{i \in I} P_2(i), \tag{6.6.7}$$

where i indexes an error event in the incorrect set I. We can ultimately perform some form of counting of error events to determine the number in various classes and now focus on the two-codeword error probability.

Given any M-PSK/M-QAM signal sequences $\mathbf{x}_i$ (in complex form), received complex data $\mathbf{r}$, and *known* channel fading amplitude $\mathbf{a}$, the appropriate decoder metric is a weighted correlation:

$$\Lambda(\tilde{\mathbf{r}}, \mathbf{x}_i \mid \mathbf{a}) = \sum_j a_j \operatorname{Re}\{x_{ij} r_j^*\} + a_j x_{aj} x_{ij}^*. \tag{6.6.8}$$

This derives from writing the likelihood function conditioned upon a given test sequence and known channel amplitudes, as discussed in Chapter 4. Note that, given a known vector of channel gains, the metric is a Gaussian random variable, and it is not difficult to show that the two-codeword probability of error is

$$P_2(i|\mathbf{a}) = Q\left[\left(\frac{2E_s}{N_0}\sum_j a_j^2 d_{i_j}^2\right)^{1/2}\right], \tag{6.6.9}$$

which involves an amplitude-weighted squared Euclidean distance between sequences. If we invoke an exponential bound on the Q-function, we obtain

$$\begin{aligned} P_2(i|\mathbf{a}) &= \frac{1}{2} e^{-\left(\frac{E_s}{N_0}\sum_j a_j^2 d_{i_j}^2\right)} \\ &= \frac{1}{2}\prod_j e^{-\left(\frac{E_s}{N_0} a_j^2 d_{i_j}^2\right)}. \end{aligned} \tag{6.6.10}$$

To determine the upper bound on node error probability, we weight the conditional error probability in (6.6.10) by the p.d.f. for the fading and then integrate to obtain the two-codeword probability $P_2(i)$. By the perfect interleaving assumption, these amplitudes are independent Rayleigh random variables:

$$f(\mathbf{a}) = \prod_j 2a_j e^{-a_j^2}. \tag{6.6.11}$$

Doing this calculation for each event i and then summing as in (6.6.7) gives

$$P_e \leq \sum_i \prod_j \frac{1}{\left(\frac{E_s}{N_0}\right) d_{i_j}^2 + 1}. \tag{6.6.12}$$

Notice first that the dependence on E_s/N_0 is non-exponential; we have seen this routinely

on the Rayleigh channel. A simpler upper bound, most accurate at high SNR, is

$$P_e \leq \sum_i \prod_j \frac{1}{\left(\frac{kE_b}{N_0}\right) d_{ij}^2}. \tag{6.6.13}$$

Two things should be noted from this formulation. First, we pick up an additional exponent on E_b/N_0 for every unit of symbol Hamming distance between sequences. This translates to the fact that the effective *diversity order* obtained for a two-sequence test is equivalent to the symbol Hamming distance. (Diversity order is the effective number of independent propagation channels obtained in time, frequency, or space diversity transmission systems, and this number represents the slope of error probability versus SNR on a log-log display.) At high SNR, it is crucial that this diversity order be maximized over all possible error events. Subsequent to this, we should design codes so that the *product* of squared Euclidean distances is maximal.

In general, codes that emerge as good for the Rayleigh fading channel are different from the good AWGN channel codes because of the differing optimization criterion. In particular, AWGN codes often have a large parallel transition multiplicity, and if any such parallel transitions exists, the minimum Hamming symbol distance is 1, leaving the code with no effective diversity. Equivalently, a single badly faded symbol is enough to cause a decoding error. A simple case in point is that of coded 8-PSK. The best 4-state AWGN code was illustrated in Figure 6.6.4 and possesses parallel transitions. A better 4-state code for the Rayleigh channel is shown in Figure 6.6.12 [68, 69]; notice the branching structure is typical of good binary convolutional codes with $R = \frac{2}{3}$ and four states. The minimum symbol Hamming distance is 2 for this code (there are multiple occurrences of these), so the coding scheme has twofold diversity, assuming once again nearly perfect interleaving is achieved. To estimate the node error probability, we can dispense with error events having 3 or greater symbol distance in the trellis and itemize the product distance on all the distance-2 events to obtain

$$\begin{aligned} P_e &\approx \frac{1}{(kE_b/N_0)^2} \sum_i \prod_j \frac{1}{d_{ij}^2} \\ &= \frac{C}{(kE_b/N_0)^2}, \end{aligned} \tag{6.6.14}$$

again pointing to second-order diversity. The coding gain of this simple 4-state code is

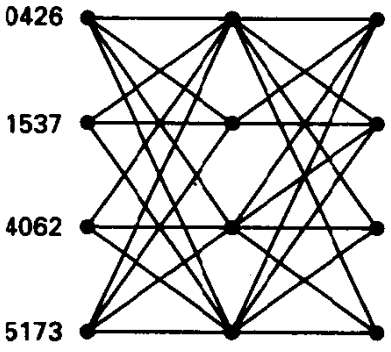


Figure 6.6.12 Trellis code for sending 2 bits/interval using 8-PSK. Minimum Hamming distance is 2, which is diversity order with perfect interleaving.

very large (on the order of 20 decibels at $P_b = 10^{-4}$), illustrating the immense payoff that coding may produce on fading channels.

Bigger trellises can, of course, achieve larger free Hamming distance. Another alternative that has been intoduced by Divsalar and Simon [69, 70] is "multiple trellis coded modulation," wherein more than one modulator symbol is assigned per branch. For example, we could view the constellation as pairs of 8-PSK symbols, giving 64 candidates. This can be set partitioned so that Hamming distance 2 is obtained on parallel transitions in the trellis, offering dual-diversity behavior. Interested readers are referred to the performance analysis and design discussion found in [69, 70]. Ho and Fung [71] examine the effect of time-correlated fading (or imperfect interleaving) on the performance of trellis codes on fading channels. Sundberg and Seshadri [72] provide a fine overview of coding for fading channels.

6.7 CONTINUOUS-PHASE MODULATION (CODING)

Continuous-phase modulation (CPM) has become a popular transmission technique in the last decade, primarily owing to the fact that the transmitted signal possesses a constant envelope (thus being suitable for nonlinear amplification) and low spectral sidelobes intrinsic to the modulation process. The reader is referred to [73] for the definitive treatment of this material. A more compact summary is available in Anderson and Sundberg's tutorial article [74].

Although the adopted name includes modulation, the memory of the signal construction makes the discussion more appropriate for our discussion of coding, and indeed the usual cases of interest exhibit a finite-state trellis description for which Viterbi decoding forms the optimal decoder. As we proceed into the discussion, we will suggest the proper way to view CPM is as a trellis code combined with a memoryless modulator, where the modulator alphabet is rather large. This viewpoint was first advanced by Rimoldi [75].

6.7.1 Signal Description

To begin, we write the transmitted CPM signal as a phase-modulated sinusoid

$$s(t, \mathbf{u}) = \left(\frac{2E_s}{T_s}\right)^{1/2} \cos\left[\omega_c t + \phi(t, \mathbf{u}) + \phi_0\right], \tag{6.7.1}$$

where $\mathbf{u} = (u_0, u_1, \ldots, u_j, \ldots)$ is a semiinfinite M-ary data sequence selected in equiprobable, independent fashion from the symmetric set $\{\pm 1, \pm 3, \ldots, \pm(M-1)\}$. The energy expended per symbol interval T_s is E_s,[14] as may be verified by integration of the square of the signal in (6.7.1), and ϕ_0 is an initial phase angle of the carrier at time zero.

Only the phase function $\phi(t, \mathbf{u})$ is of real interest in (6.7.1), for it conveys the message sequence, and the transmitted signal is exactly specified once other fixed parameters are known. The nature of evolution of this phase function defines a particular CPM scheme.

[14] Again, this technically requires that either $\omega_c = n2\pi/T_s$ or $\omega_c \gg n2\pi/T_s$.

We define the phase function to be a superposition of responses produced by each symbol:

$$\phi(t, \mathbf{u}) = 2\pi h \sum_{j=0}^{n} u_j q(t - jT_s), \tag{6.7.2}$$

where $q(t)$ is a ***phase shaping pulse*** and h is the ***modulation index***. Thus, by definition, the phase modulation $\phi(t, \mathbf{u})$ is a linear function of the message. By convention, we choose the modulation to be causal, that is, $q(t) = 0$, $t \leq 0$, and let the final value be $q(t) = \frac{1}{2}$, $t \geq LT_s$, where L is the number of *symbol* intervals when a given data symbol exerts a *changing* influence on the cumulative phase.

From (6.7.2) and the preceding boundary conditions, we see that over the interval $[jT_s, (j + L)T_s]$, a given data symbol, u_j, induces a net phase change of $u_j 2\pi h(\frac{1}{2}) = u_j h\pi$ radians, and this induced phase modulation persists over time in CPM, by virtue of (6.7.2). Since each symbol's eventual phase influence is proportional through h to the data symbol, it becomes obvious that h is a modulation index, just as in analog angle modulation schemes. Because of implementation considerations and because a finite-state trellis follows, as we will see, h is assumed to be rational; that is, $h = p/r$, with both p and r normally small, relatively prime integers. Furthermore, for reasons of spectrum economy, it will be the usual case that $h \leq \frac{2}{3}$.

The phase shaping pulse, $q(t)$, may be related to the ***frequency-shaping pulse***, $g(t)$, by

$$q(t) = \int_0^t g(x)\, dx, \tag{6.7.3}$$

and we may therefore view CPM as either frequency or phase modulation.

A CPM scheme is completely defined by h, M, and either $q(t)$ or $g(t)$. Two families of CPM schemes will be covered here and are known as ***L*-REC** (for rectangular):

$$g(t) = \begin{cases} \dfrac{1}{2LT_s}, & 0 \leq t \leq LT_s, \\ 0, & \text{else}, \end{cases} \tag{6.7.4a}$$

and ***L*-RC** (for raised cosine):[15]

$$g(t) = \begin{cases} \dfrac{1}{2LT_s}\left[1 - \cos\left(\dfrac{2\pi t}{LT_s}\right)\right], & 0 \leq t \leq LT_s, \\ 0, & \text{else}. \end{cases} \tag{6.7.4b}$$

Other pulse shapes are possible, but L-REC and L-RC bracket the range of options and exhibit the essential properties obtainable with other choices. Figure 6.7.1 illustrates the two generic frequency pulse shapes and the corresponding phase pulse shapes. The nomenclature, however, derives from the shape of the frequency pulse.

A rather well known special case of CPM is called MSK, for minimum shift keying. It is equivalent to the use of 1-REC pulse shaping and adoption of $h = \frac{1}{2}$. 1-REC CPM is equivalent to standard frequency-shift-keying modulation, and with $h = \frac{1}{2}$ the frequency

[15]Here raised-cosine refers to a time-domain characteristic, not a frequency-domain shape.

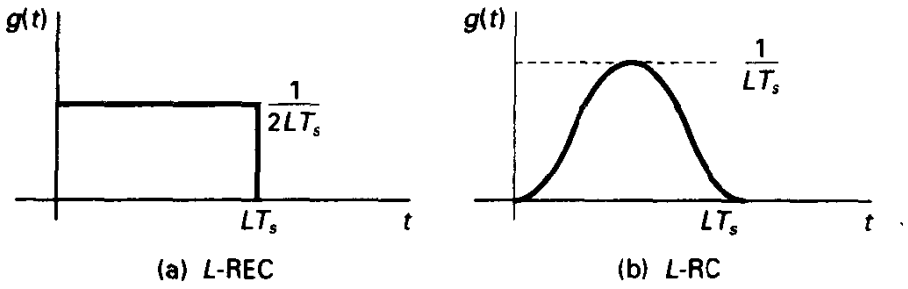


Figure 6.7.1 Frequency pulses for CPM. (a) L-REC; (b) L-RC.

shift ($\pm\frac{1}{4}T_s$) is the minimum such that the two modulated signals are orthogonal over one symbol interval, hence the name.

As an aside, we may incorporate standard DPSK modulation into the CPM framework if we allow discontinuous phase. We obtain M-DPSK by choosing $q(t)$ to be the step function with height $\frac{1}{2}$ and $h = \frac{1}{M}$. This would mean that the new phase in each modulation interval changes by $\pm\pi/M, \pm 3\pi/M, \ldots, \pm(M-1)\pi/M$. (Actually, what we have implemented is M-DPSK with an additional phase rotation of π/M radians in each interval, but this does not materially affect the signal's error probability or power spectrum.) It is worth noting that this $\pi/4$-QPSK technique has been adopted for the North American TDMA cellular standard known as IS-54.

A helpful way of visualizing CPM is provided by the ***phase tree***, the set of all phase paths $\phi(t, \mathbf{u})$ commencing with some common initial conditions at $t = 0$. Because a new M-ary symbol begins to influence the phase trajectory every T_s seconds, the phase tree exhibits M-way branching, and over N symbol intervals, there are M^N distinct phase paths or codewords.[16] In Figure 6.7.2 we show the phase tree for binary 1-REC (Figure 6.7.2a), binary 3-RC (Figure 6.7.2b), and quaternary ($M = 4$) 3-RC (Figure 6.7.2c) cases for four levels of the tree. Note that the phase axis is scaled in terms of $2\pi h$, so the trees shown may be employed for any modulation index. Of course, phase angles are rendered modulo-2π, and it should be remembered that two phase paths that appear very dissimilar on a planar presentation may in fact be identical as modulo-2π trajectories and thus produce identical signals. This fact is made more clear by wrapping the planar graphs shown in Figure 6.7.2 onto a cylinder, as shown in Figure 6.7.3. In so doing, we must choose a modulation index h, for this controls the *stretch factor* in the wrapping of the planar tree onto a cylinder.

Two properties of CPM are readily apparent in either the phase tree or phase cylinder. First, any specific phase trajectory is relatively smooth, say compared to the phase process produced by a PSK modulator. In particular, the phase is everywhere continuous [by definition of $q(t)$], and for appropriate pulse shaping, derivatives of the phase process may be continuous as well. Baker [76] has shown that the asymptotic rate of decay of spectral sidelobes in digital FM (or PM) signaling is related to the number of continuous derivatives of the pulse $q(t)$. That is, if m is the *smallest* order of derivative that is *discontinuous*, meaning that m is the smallest value for which $d^m q(t)/dt^m \neq 0$ for some time point, then the power spectrum behaves for large frequency separation

[16]This is perhaps the first encounter with codewords that are in fact complicated waveform patterns.

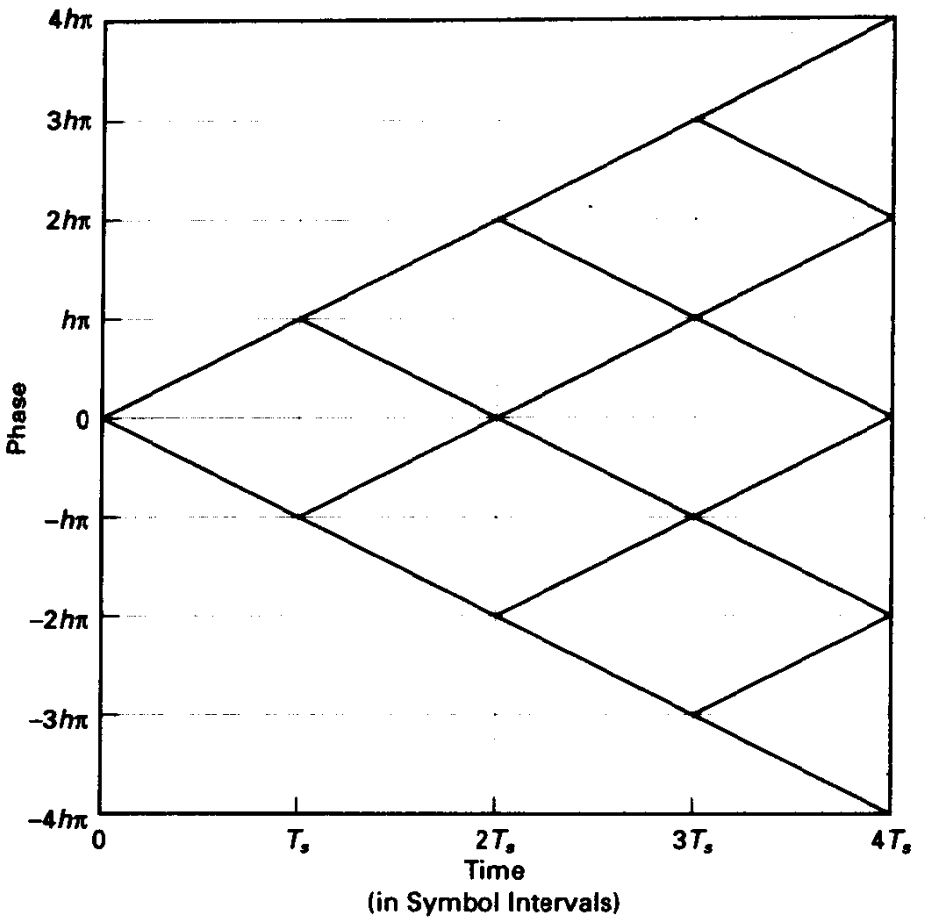


Figure 6.7.2a Phase tree for 1-REC and $M = 2$.

from the carrier frequency as $O(f^{-2m-2})$; that is,

$$G_s(f) \approx \frac{K}{|f - f_c|^{2m+2}}, \qquad |f - f_c| \gg 0. \tag{6.7.5}$$

In the case of L-REC pulse shaping, Figure 6.7.2 shows that although the function $q(t)$ is continuous the first-derivative is not everywhere continuous; hence $m = 1$ for this pulse, and the asymptotic rate of decay of spectral sidelobes for all such CPM schemes is therefore of order $|f - f_c|^{-4}$ for positive frequencies. For L-RC signaling, first and second derivatives of $q(t)$ are continuous at all points of time, but the third derivative is not at the two points $t = 0$ and $t = LT_s$ (Exercise 6.7.2). Therefore, power spectra for M-ary L-RC modulation decrease as the eighth power of frequency for large separation from the carrier. Too much should not be made of this, however, for this consideration reveals little about the distribution of the bulk of the signal power. More will be said about this shortly.

The second key observation regarding the phase tree is that two phase paths that split at $t = 0$ (or any symbol boundary for that matter), due to distinct data symbols, may not remerge (and stay remerged) until after $L + 1$ symbol intervals have elapsed.

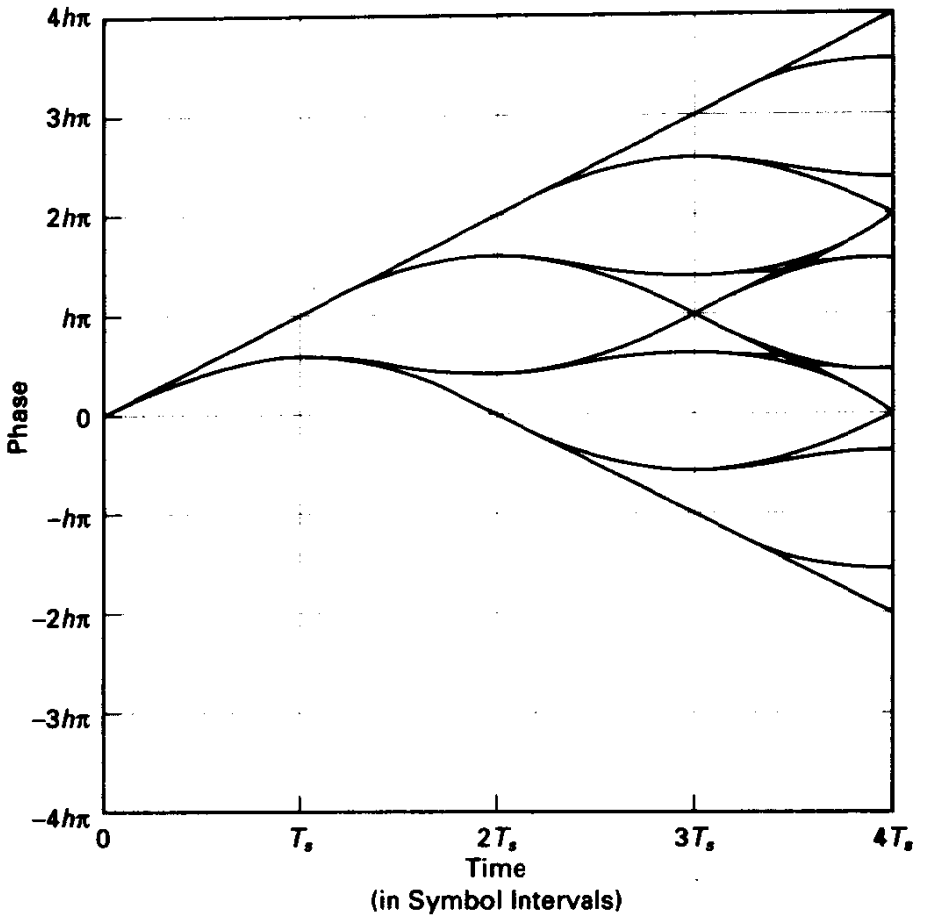


Figure 6.7.2b Phase tree for 3-RC and $M = 2$.

Therefore, the parameter L plays a key role of memory order for CPM schemes. With the 1-REC scheme, phase paths diverging at $t = 0$ may merge again after two symbol intervals, while for 3-RC, the quickest merger happens after four intervals.[17] The implication of this CPM property lies in detection of the message: it is intuitively clear that an optimal decoder should utilize an interval of the received waveform $r(t)$ with duration at least $(L + 1)T_s$ seconds to decide any given data symbol and that the energy associated with this span of time is useful to the detection of a given symbol. This principle is germane to any signaling technique with memory.

6.7.2 State Representation

The phase process $\phi(t, \mathbf{u})$ is cumulative on message symbols and exhibits infinite memory, even though the frequency pulse lasts only LT_s seconds. In studying the phase process over the interval $nT_s \leq t \leq (n + 1)T_s$, we observe that the current symbol u_n,

[17] Quicker mergers may happen for certain modulation indexes, the *weak* modulation indexes, but these are to be generally avoided, as discussed later.

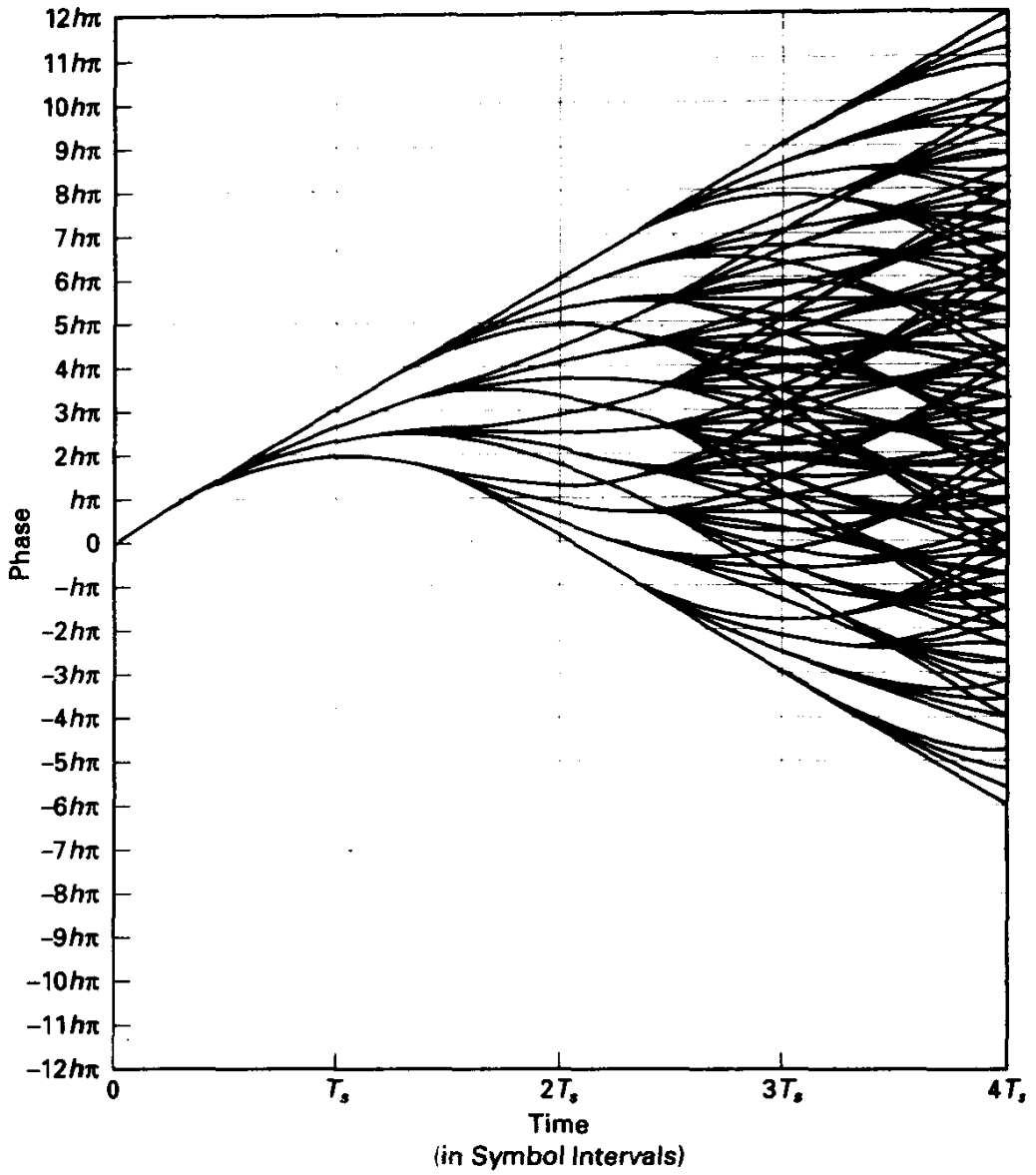


Figure 6.7.2c Phase tree for 3-RC and $M = 4$.

as well as those symbols indexed back to $n + 1 - L$, will exert a *changing* influence on $\phi(t, \mathbf{u})$, while symbols prior to u_{n+1-L} have produced their final phase influence, $u_j h\pi$. Hence, we may conveniently separate the total phase of (6.7.1) into two components:

$$\phi(t, \mathbf{u}) = (\phi_0 + \sum_{j=0}^{n-L} u_j h\pi) + 2\pi h \sum_{j=n-L+1}^{n} u_j q(t - nT_s), \qquad nT_s \le t \le (n+1)T_s$$

$$= \theta_n + 2\pi h \sum_{j=n-L+1}^{n} u_j q(t - nT_s), \qquad nT_s \le t \le (n+1)T_s. \tag{6.7.6}$$

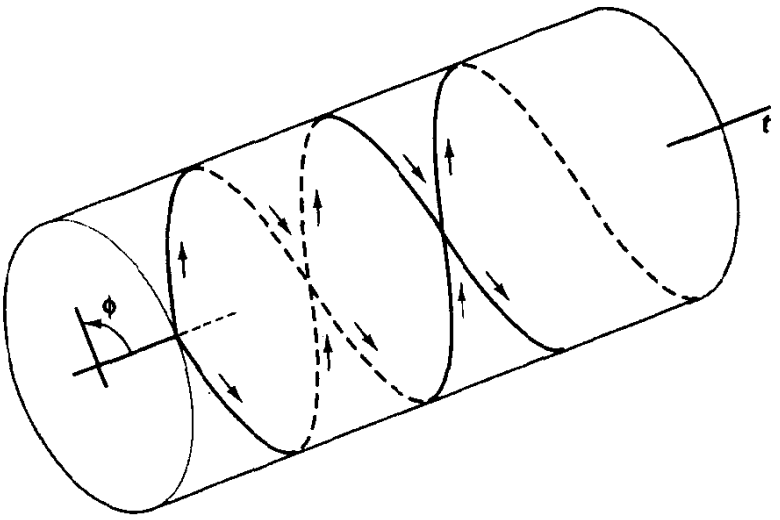


Figure 6.7.3 Phase trajectories drawn on cylinder. Not all paths shown.

This suggests a state-variable description of the CPM signal: by specifying the cumulative phase term θ_n and the data vector $[u_{n-L}, u_{n-L+1}, \ldots, u_n]$, we may exactly represent the CPM signal over $[nT_s, (n+1)T_s]$ using (6.7.6) in (6.7.1). Furthermore, we can see by inspection how to update the state under the influence of message inputs. This representation would not be noteworthy unless we observe that for $h = p/r$ the cumulative phase term in (6.7.6) may take on exactly r phase positions at each value of n, relative to the initial phase. This follows from the fact that

$$\theta_n = \left(h\pi \sum_{j=0}^{n-L} u_j\right) \bmod 2\pi = \left(\frac{p}{r} \sum_{j=0}^{n-L} u_j\right) \bmod 2. \tag{6.7.7}$$

The latter sum, taken modulo-2, is contained in $\{0, \pm 2\pi/r, \ldots, \pm(r-1)\pi/r\}$ for an even number of summands or in $\{\pm\pi/r, \ldots, \pm(r-1)\pi/r\}$ for an odd number of summands, so in either case r phase positions exist. A technical nuisance is that if p is odd the r phase positions are the same set at each time, while if p is even, the r phases cycle between two r-ary sets. The latter is tantamount to a time-varying state description.

To summarize, a finite-state description for the CPM modulator is provided by specifying the vector $\sigma_n = (\theta_n, u_n, \ldots, u_{n-L+1})$. Since this vector may assume one of rM^{L-1} values, we say that the state complexity of a given CPM scheme is $S = rM^{L-1}$. Notice that this depends only on h, M, and L, and not the particular nature of the shaping pulse. Naturally, a finite-state trellis description follows, depicting all possible state transitions of a CPM modulator.

The updating of the state vector is specified by rules for determining the new cumulative phase component and the new data vector component. These are

$$\theta_{n+1} = \theta_n + h\pi u_{n-L+1} \tag{6.7.8a}$$

and

$$\mathbf{u}_{n+1} = (u_n, u_{n-1}, \ldots, u_{n-L+1}), \tag{6.7.8b}$$

which is merely a right shift of $\mathbf{u}_n$, with the most recent data symbol u_n preaffixed.

It should also be apparent that the state specified allows a complete description of the signal waveform in any given interval, since the state, together with knowledge of a new input symbol, provides all the information needed to compute $s(t, \mathbf{u})$ through (6.7.1). The output equation is not so simple now as in the case of, say, trellis-coded PSK, but can certainly be expressed as

$$\phi(t, \mathbf{u}) = f(t, \sigma_n, u_n), \tag{6.7.9}$$

producing only a finite number of possible phase trajectories over any symbol interval. In particular, the number of states is $S = rM^{L-1}$, and each state produces M transitions to next states, so the number of distinct signal trajectories in any one interval is rM^L. (These are, in fact, distinct signal trajectories, as contrasted with the case of trellis coding with a memoryless modulator, wherein the same modulator signal may be produced with several state transitions.)

Example 6.25 State Description for Binary, 2-REC Signal

Let there be $M = 2$ symbols possible at each symbol time, and let 2-REC be the frequency pulse shape. Thus, the phase tree exhibits piecewise-linear phase segments having slopes $h\pi/T_s, 0$, and $-h\pi/T_s$. With $h = \frac{2}{3}$, the number of phase states is 3, and with $L = 2$, the total number of states is $3 \cdot 2^{(2-1)} = 6$. In Figure 6.7.4 we show the trellis form of the state-transition diagram. Note that states are labeled with a cumulative phase component and a data component, the latter a scalar variable in this case. For example, if the modulator state is $\sigma_n = (2\pi/3, +1)$ and the next input is a -1 symbol, then the cumulative phase entry becomes $4\pi/3$ radians (adding the contribution of the previous $+1$ symbol into the cumulative phase), and the next data state is -1.

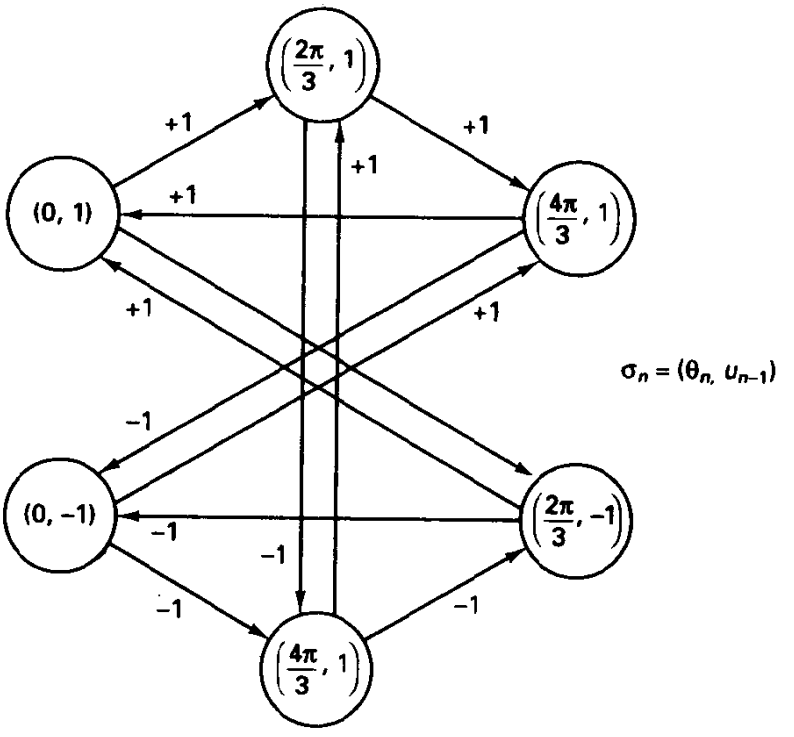


Figure 6.7.4 State transition diagram for 2-REC, $h = \frac{2}{3}$, $M = 2$.

6.7.3 Modulator Implementations

CPM modulators may be implemented in several forms. The first is suggested by the definition of the waveform: implement phase modulation with a baseband, pulse-shaped input to the modulator, and scale the phase modulation index appropriately. Equivalently, we could fashion a frequency modulator and excite it by a superposition of frequency pulses. By definition, use of a single oscillator with frequency modulation supplies the continuous-phase property, and pulse shaping in the form of $g(t)$ provides extra smoothness in the phase if desired. This is the standard means of implementing FSK, with a 1-REC pulse shape. The only possible deficiency of this technique is lack of precise control of the modulation index h under the influence of aging of components, temperature effects, and the like. It is not absolutely critical that the modulation index be exactly its prescribed value, say $h = \frac{1}{2}$, but if it is significantly different, then the receiver designed for the nominal index may be significantly mismatched. The power spectrum will be slightly different than expected as well.

A second implementation is shown in Figure 6.7.5, derived from expanding (6.7.1) into quadrature form. We can employ quadrature modulation of the carrier by the sine and cosine of the phase function $\phi(t, \mathbf{u})$, employing a table-lookup process to fetch the desired quadrature waveforms. An alternative provides a discrete-time representation of the $N = M^L$ generic phase progressions, modifying these by the desired phase offset θ_n in the table-lookup process. The time discretization is certainly a design issue, but eight samples per T_s-second interval are probably adequate, particularly with filter smoothing prior to modulation. The advantage of this implementation is that the phase increments are carefully controlled and drift of the modulation index cannot occur. The reader is referred to Maseng [77] for further discussion of this implementation.

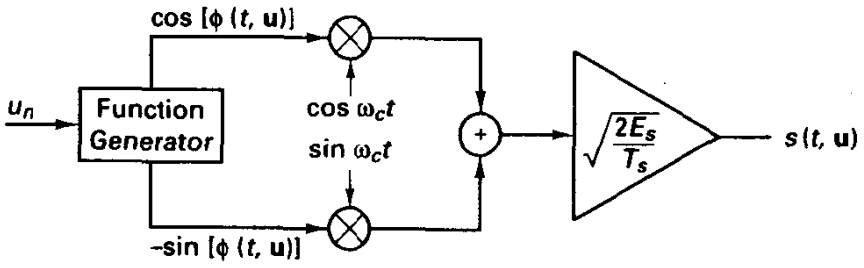


Figure 6.7.5 Quadrature modulation realization of CPM.

6.7.4 Description of CPM as Memoryless Modulation Preceded by Coding

Given our previous discussion, it becomes clear that one way of visualizing the CPM signaling process is as the cascade of a section having memory and a memoryless modulator or signal generator. That is, the state equation subsumes all the essential memory of the modulation process and defines the output signal as soon as the new input symbol is specified.

In systems with $L = 1$, the only memory present in the modulation is the cumulative phase term, which enforces the continuous-phase property. In such cases the state of the modulator is

$$\sigma_{n+1} = \theta_{n+1} = \theta_n + h\pi u_n, \text{ modulo } 2\pi. \tag{6.7.10}$$

The output signal is clearly only dependent on σ_n and u_n, and this leads to a realization of the modulator in the form shown in Figure 6.7.6. The modulator has Mr inputs at any given time, where again r is the denominator of h, and in response produces one of Mr waveforms over the next modulation interval. It should be emphasized that the standard CPM modulator would be viewed as having M possible inputs at each interval, whereas our alternative view allows a larger number; this is no inconsistency, for given a certain prehistory, only M possible waveforms are possible; a finite-state encoder enforces the CPM constraints. It is also worth noting that the multiplicity r stems from a phase rotation of the basic set of M^{L-1} waveforms. Thus, only a relatively small set of basic waveforms must be synthesized.

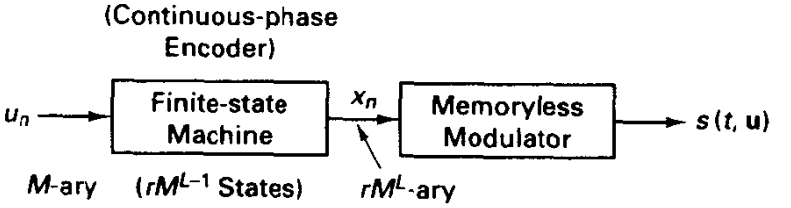


Figure 6.7.6 *Decomposition of CPM into a unit with memory and a unit without memory. After Rimoldi, [75].*

Generalizing this viewpoint, we can think of the CPM process as a ***memoryless modulation*** preceded by a ***low-rate digital encoder***. The latter accepts $\log_2 M$ bits per symbol interval and selects a symbol from a modulator set of size rM^L symbols. Thus, in classical terms the encoder's rate is

$$R = \frac{\log M}{\log(rM^L)} \text{ bits/modulator input bit.} \tag{6.7.11}$$

Thus, for 4-ary 3-RC modulation with $h = \frac{1}{2}$, the apparent rate is $R = \frac{2}{7}$.

The low rate is precisely a result of the desired phase smoothness constraints. Consideration of merely energy efficiency would suggest that expansion of the modulator set by a small factor is adequate, as achieved in trellis coding with expanded signal constellations [4]. Achieving high spectral efficiency with the constant envelope property, however, requires larger set expansion.

Rimoldi [78] has referred to this representaion of CPM as a cascade of a part having memory [a continous phase encoder (CPE)] and a part that is memoryless [memoryless modulator (MM)]. This decomposition is more than idle thought; recently it has been shown [78, 79] that such a view opens the way to optimizing additional trellis coding that might be utilized, whereas when memory is left in the modulator, it is difficult to directly optimize codes with a desired number of receiver trellis states.

6.7.5 Power Spectra of CPM Modulation

CPM is a nonlinear digital modulation technique whose spectral analysis is rather complicated. The modulation process is a Markov process, when h is a rational number, so in principle the methods of Chapter 3 can be invoked to determine the power spectrum. However, the state complexity may be large, and the numerical accuracy of this procedure may be poor.

More direct attacks are described in [80] for the case of 1-REC modulation and in [81] for general CPM. Both are based on determination of the autocorrelation function for the complex envelope of the CPM signal. In general, the power spectrum may include spectral lines, in addition to the continuum spectrum.

Some general remarks follow:

1. The power spectrum generally widens with increasing h, although the data led shape changes. Spectral lines emerge in the power spectrum for integer-valued h, although this is normally not a case of interest.
2. Use of small h is not a recipe for extremely bandwidth efficient communication, despite the small values of B_{99}. This is analogous to the early misunderstanding of FM transmission, where it was conjectured that small frequency deviation could compress bandwidth below that of the modulating signal. In fact, the small power contained in the sidelobes of the signal is precisely where the information resides.
3. Smoothness of the CPM signal does indeed increase the rate of decay of spectral sidelobes. This may be advantageous in multicarrier systems, in reducing adjacent channel interference. However, one possible drawback to this increased sidelobe decay is a wider spectral mainlobe. MSK is a case in point: the sidelobes decay as $|f - f_c|^4$, rather than $|f - f_c|^2$ for PSK/QPSK, but the mainlobe width for MSK is 50% larger than that of QPSK at equal bit rate. Thus, the claims for improved spectral efficiency of CPM must be judged carefully.

Figure 6.7.7 presents power spectra for binary CPFSK and binary 3-RC, with $h = \frac{1}{4}$ and $h = \frac{1}{2}$.

6.7.6 Coherent Decoding of CPM

We now consider the optimal decoding of continuous-phase modulation, under the assumption that the signal is received by a nonfading, nondispersive channel, together with additive white Gaussian noise. We further assume that the receiver has been able to determine the received signal's reference phase angle, ϕ_0, so coherent detection is possible. We seek the maximum likelihood *sequence* estimate for the data sequence $\mathbf{u}$.

Our discussion in Chapter 2 immediately implies that the optimal receiver for a sequence of length N symbols embedded in white Gaussian noise would correlate the received waveform, $r(t)$, against the M^N possible signal trajectories and choose that sequence producing the largest correlation. (Note that pure correlation without any additional bias terms follows from assumption of equiprobable message sequences and the fact that each signal contains equal energy.) Correlation over an interval of length

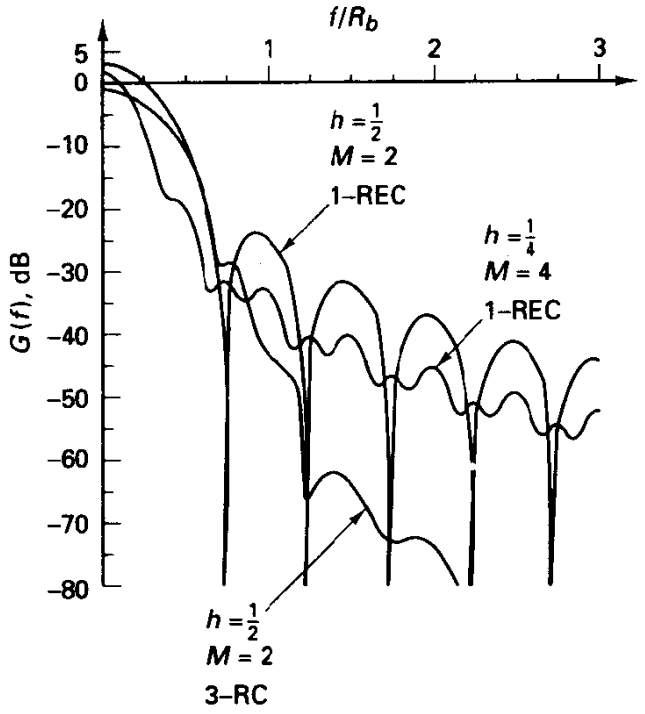


Figure 6.7.7 Power spectra for $M = 2$, 1-REC, $h = \frac{1}{2}$; $M = 4$, 1-REC, $h = \frac{1}{4}$; M = 2, 3-RC; $h = \frac{1}{2}$. Note rapid decay of 3-RC spectrum.

NT_s seconds may be decomposed into a sum of N correlations performed over T_s-second intervals. Thus, the optimal receiver may be implemented as shown in Figure 6.7.8, where we show a bank of analog symbol-duration correlators, each performing a correlation with one of rM^L signals, coupled with a discrete-time processor that sums appropriate outputs of the correlator bank to form global sequence correlations. The complexity of this receiver still grows exponentially with blocklength N, however.

Because of the finite-state nature of the encoder, we may apply the principle of optimality once again and use the Viterbi algorithm to perform sequence correlations and avoid the complexity that grows exponentially with message length N. The Viterbi receiver uses outputs of the correlator (or matched filter) bank as metrics for scoring various trellis hypotheses. The trellis decoder has rM^{L-1} states, as described previously, and for each state performs an M-way add/compare/select operation to determine the survivor state. Such a trellis decoder is little different from those we have seen earlier

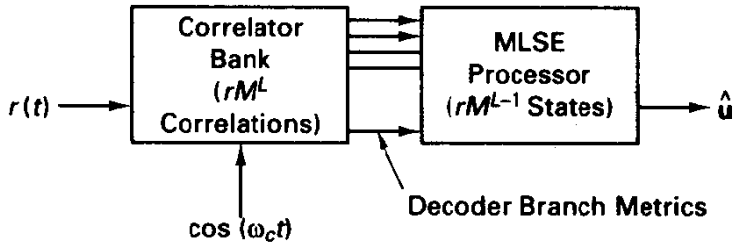


Figure 6.7.8 Generic maximum likelihood decoder for continuous phase modulation.

in the chapter; perhaps the notable complication here is that the number of fundamental correlations needed to form metrics is larger in general for CPM, that is, rM^L versus, say, only one for antipodal signaling of binary convolutionally coded sequences. Recall, however, that many of these correlations are mere phase rotations of others, allowing complexity reduction. An important engineering question is whether the number of correlators can be reduced to a smaller state by approximating classes of waveforms by single functions; this is akin to a quantization strategy.

The concept of free distance is important for decoding performance in CPM, just as it was in earlier cases for trellis codes. For deciding among arbitrary signals in AWGN, the fundamental quantity of interest is the minimum L_2 distance between any pair of signals to be considered. We define the ***free distance*** of a CPM scheme as the minimum L_2 distance among all pairs of signal paths that are initially merged, then split in the trellis, and later remerged. For any two signals $s_i(t)$ and $s_j(t)$ that are unmerged from nT_s to $(n+l)T_s$, the L_2 distance between them is

$$d_{12} = \int_{nT_s}^{(n+l)T_s} \left[s_i(t) - s_j(t)\right]^2 dt = \frac{2E_s}{T_s} \int (1 - \cos[\Delta\phi(t)]\, dt, \tag{6.7.12}$$

where $\Delta\phi(t)$ is the ***phase difference function*** between the two signals. Technically, this distance expression pertains under the assumption that there are either (1) an integral number of cycles of the carrier frequency per symbol interval or, more typically, (2) the number of carrier cycles per symbol is large.

It is a simple matter to determine the L_2 distance between two CPM signals by (6.7.12) merely by following the phase difference trajectory. For certain coding schemes having adequate symmetry, we have found it is sufficient to let any one sequence, typically the all-zeros sequence, be the reference point for calculation of distances to other sequences. Unfortunately, in the case of CPM this does not follow, and some transmitted sequences have different distance sets to neighboring sequences. For example, consider 1-REC signaling with $h = \frac{1}{4}$. The input sequence 100000 has a neighboring sequence 0100000 that merges after two symbol intervals, with squared distance $d_{12}^2 = 2E_s[1 - \sin(h\pi)/h\pi]$, as follows from (6.7.12). On the other hand, the input sequence 11000000 does not have a two-step error event for general h. This fact complicates determination of the free distance for a given CPM configuration, and in essence all possible pair sequences must be considered by a trellis sweep on *pair states*. Mulligan and Wilson [82] have reported a procedure for efficiently performing this computation. In the case of *uncoded* CPM, since the distance function depends only on phase difference, which in turn depends on the difference sequence between the two messages, it is sufficient in searching for d_f^2 to examine all *difference sequences* obtained by placing one of $2M - 1$ difference symbols in each position of the sequence.

In CPM, we are concerned with signal paths that split and remerge at some later time. If we denote by $\mathbf{u}^{(1)}$ and $\mathbf{u}^{(2)}$ two such information sequences, then it is readily seen that the phase difference between the two signals will be

$$\Delta\phi(t) = 2\pi h \sum_{j=0}^{n} \gamma_j q(t - jT_s), \tag{6.7.13}$$

where $\gamma_j = u_j^{(1)} - u_j^{(2)}$ is the ***symbol difference sequence***. If M-ary transmission is employed, the possible differences in each symbol position are $\{0, \pm 2, \pm 4, \ldots, \pm 2M - 1\}$. Two information sequences that differ over a span of N symbols, that is, $\gamma_0 \neq 0$ and $\gamma_{N-1} \neq 0$, may not merge is the signal sense due to phase memory. The earliest time the signals may become equal (and stay equal or merged) is at time $j = N - 1 + L$, or $L - 1$ intervals beyond the unmerged information span. Whether they do in fact merge or not depends on the difference sequence γ. In particular, we require that the running sum of the γ_j sequence depart from zero and return to zero for $L - 1$ additional intervals to effect a state merger.

Example 6.26 Merger Events for Binary 3-RC

Consider the sequence $\mathbf{u}^{(1)} = (1, -1, 1, 1, 1, 1, \ldots)$ and $\mathbf{u}^{(2)} = (-1, 1, 1, 1, 1, 1, \ldots)$. The corresponding difference sequence is

$$\gamma = (2, -2, 0, 0, 0, 0, \ldots) \tag{6.7.14}$$

for which the running sum is $2, 2 - 2 = 0, 2 - 2 + 0 = 0, \ldots$. Thus, although the information sequences merge after two steps, the additional memory implicit in 3-RC delays the true signal merger until the end of the fourth symbol interval. Such may be seen in Figure 6.7.9 as the ***quickest merger event***. This merger event happens for any modulation index h and is called an ***inevitable merger event***. Other inevitable merger events are specified by any difference sequence whose running sum becomes 0 and remains so. For example, the difference sequence $\gamma = (2, 2, -2, -2, 0, 0, \ldots)$ provides a signal merge at depth $6T_s$.

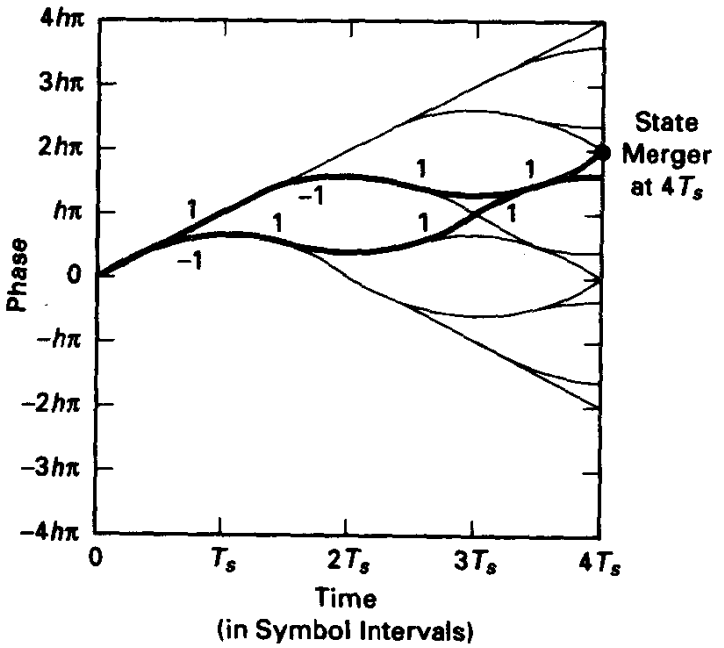


Figure 6.7.9 Shortest split-and-merge event for binary 3-RC (see darker curve).

It may happen that for *some* modulation indexes merges occur that are not inevitable merges, but occur because

$$2\pi h \sum_{j=0}^{n} \gamma_j = m2\pi, \qquad m \neq 0. \tag{6.7.15}$$

In these cases the modulo-2π rendition of the phase makes it possible for sequences to merge on the "backside" of the phase cylinder for certain h. It may be that such merges dominate the minimum distance, that is, induce smaller L_2 distance than inevitable merger events and, in such cases, these modulation indexes are known as ***weak*** modulation indexes.

It is to be expected that the free distance of a CPM scheme will depend on M; the modulation index h (which determines the rapidity of phase separations); the pulse shape, $q(t)$; and on L. Figure 6.7.10 depicts the squared free distance normalized to $2E_b$ as a function of h for 1-REC signaling with various choices of M. Figure 6.7.11 does the same for 3-RC signaling. In both we note an expected growth with distance as the modulation index increases and then a kind of saturation effect wherein distance no longer grows. This is due to the modulo-2π rendition of phase angle. Sequences that appear to be widely separated on a planar drawing of the phase tree are perhaps not separated widely in phase angle, giving small effective distance by (6.7.12).

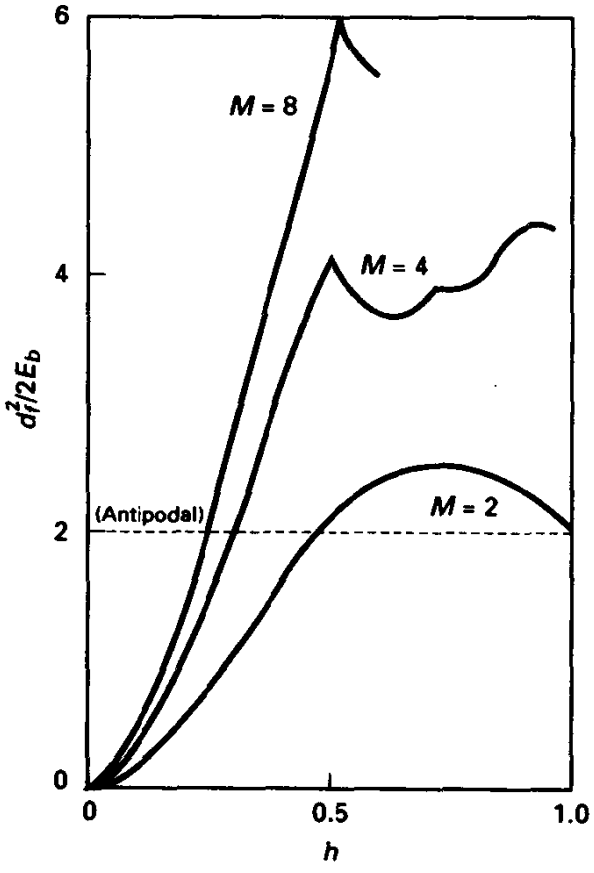


Figure 6.7.10 Normalized free distance for 1-REC CPM versus h.

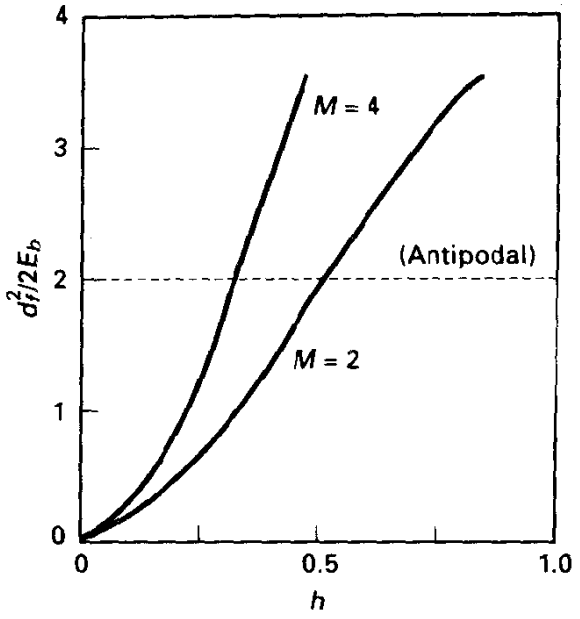


Figure 6.7.11 Normalized free distance for 3-RC CPM versus h.

It is also true that for different values of h the dominant (smallest distance) pair of sequences changes. Typically, for small h the dominant distance event is the pair of sequences 100000 and 010000, which lead to the quickest inevitable merger of the signal paths. In any case, the distance for this pair serves as a simply computed upper bound on free distance.

Aulin [83] has employed transfer function bounding approaches to analyze the decoded symbol error probability for optimal CPM decoders. This process is somewhat analogous to that for convolutional codes, but requires a more careful averaging with respect to choice of transmitted sequences.

6.7.7 Related Topics in CPM

A large amount of research has been directed at the problem of receiver simplification, that is, providing alternatives to a Viterbi-type decoder with $S = rM^{L-1}$ states. Most of this is summarized in the text of Anderson, Aulin, and Şundberg [73], but we provide a capsule summary.

First, the fact that we are dealing with a frequency-modulated signal suggests discriminator detection as a simple expedient. The structure of such a receiver is a chain of bandpass filter/discriminator/low-pass filter, with the low-pass filter sampled at the symbol rate. If the pulse used has $L = 1$, then decisions can be based on single samples; otherwise, some sort of decision feedback is required to reconvert the instantaneous frequency back to a symbol sequence. Since such receivers do not derive from decision-theoretic bases, we must expect them to be somewhat suboptimal. Analysis of such receivers is complicated by the nonlinearity of the discriminator, as is the case for any FM demodulator. In particular, proper treatment of the click noise component is at best approximate. Simmons and Wittke [84] have provided a treatment of the case of binary 1-REC, showing the dependence on prefilter bandwidth.

A related suboptimal detector is motivated by the fact that information is conveyed in the phase progressions, and a detector based on phase differences thereby follows. These detectors are known as differential detectors and are indicated in Figure 6.7.12. Usually, such techniques are very inefficient for $M > 2$ and for longer pulse shapes.

A technique due to Svensson [85] decodes CPM with an intentionally mismatched trellis, which has fewer states than a matched trellis. By referring to Figure 6.7.13, we can see a binary 1-REC phase tree hiding in the binary 3-RC phase tree, the former indicated by the dashed lines. This would suggest that detection can be performed with one-fourth the number of trellis states. The sacrifice in error performance is discussed in [85], but is not simply summarized. However, for small h it is known that fractional-decibel losses are possible for significant savings in complexity. Of course, we retain completely the attractive spectral attributes of the transmitted signal.

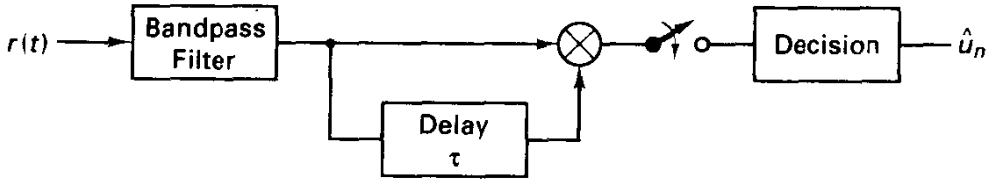


Figure 6.7.12 Differential detector for CPM; τ is optimized for a given scheme. Note similarity to DPSK detector.

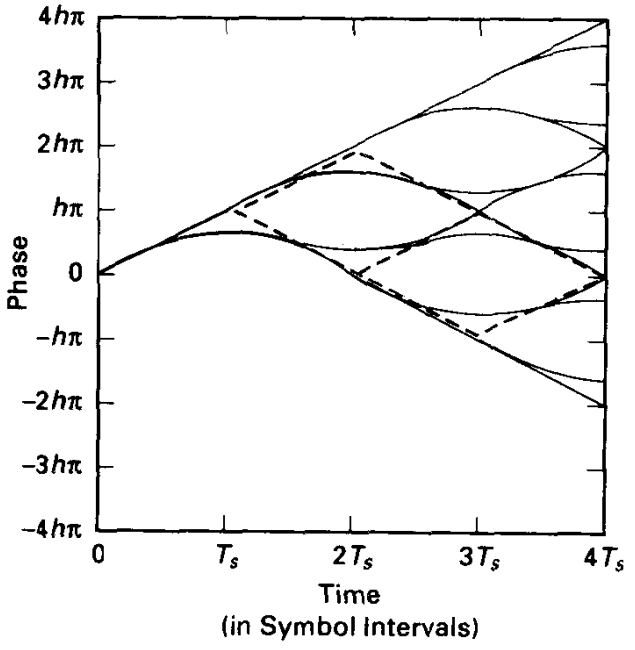


Figure 6.7.13 Binary 1-REC phase tree superimposed on binary 3-REC phase tree, suggesting simplified, suboptimal trellis decoder.

Multi-h codes is the name given to CPM modulation wherein the modulation index is made to vary with time in periodic fashion [85]. Proper choice of the set of modulation indexes can prolong the quickest remerger time in the trellis, thus increasing the free distance. An example is the alternation between $h = \frac{2}{4}$ and $h = \frac{3}{4}$ with binary 1-REC. This can be decoded with a four-state (time-varying) trellis and has shortest

merger events of length-3 symbols, rather than two with standard 1-REC signaling. The power spectrum is well approximated by assuming a fixed-h analysis, using the average value of h over the cycle [87].

As with any modulation technique, we can redundantly code the inputs to a CPM modulator, achieving further increase in the free distance of the system. For example, we could adopt 4-ary 1-REC signaling with $h = \frac{1}{4}$ as the signaling base but code these inputs with an $R = \frac{1}{2}$ trellis code accepting one bit per interval. Figure 6.7.14 illustrates the best 4-state coding scheme with these parameters, [87, 88]. The combined transmission method is a super-encoder combining the memory of the explicit encoder with the memory of the CPM scheme. The net state complexity of such systems is at most $S = rk^{m+L}$, where m is the memory order of the trellis encoder and k is the number of input bits per trellis level. Correspondingly, the complexity of an ML receiver increases with additional precoding.

Sometimes trellis coding of CPM does not increase the number of states in an obvious way, that is, the minimal state description may have only 8 states rather than the apparent 16 obtained by multiplying state complexities of the trellis encoder and the CPM modulator. Rimoldi's decomposition always makes clear how many states a coded CPM system will have, and it allows direct search for the optimal free distance system, given a specified number of receiver states [78]. Related material on this issue is also given in [79], where "matching" the encoder to the CPM modulator is found to be a subtle issue.

We close our CPM study with a depiction of the energy efficiency/spectral efficiency tradeoff in Figure 6.7.15. As usual in digital communications, one resource must be traded for the other when complexity is held fixed. To measure spectral efficiency, we adopt the B_{99} definition; that is, we define bandwidth as that range of frequencies that captures 99% of the total signal power and normalize by the *information rate* to obtain spectral efficiency. (This choice is admittedly arguable.) To measure energy efficiency, we normalize the free Euclidean distance of the signaling scheme, relative to $2E_b$. Recall antipodal PSK would have a normalized efficiency of 1 on this basis, as does MSK (which is 1-REC with $h = \frac{1}{2}$.) Schemes that are more energy efficient than these have $d_f^2/2E_b > 1$; some schemes may be less energy efficient but preferably more bandwidth efficient.

Figure 6.7.15 shows this resource tradeoff for three schemes: (1) uncoded binary 1-REC, with h varying; (2) uncoded 4-ary 3-RC signaling; and (3) $r = \frac{1}{2}$ coded 4-ary 1-REC, with h varying. As we might anticipate, the schemes whose energy-vs-spectrum occupancy curves are most efficient have the greatest state complexities.

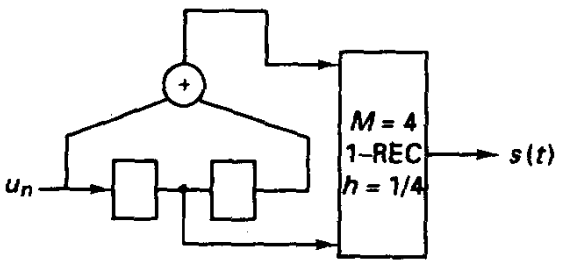


Figure 6.7.14 Best $r = \frac{1}{2}$, memory 2 convolutional encoder combined with $M = 4$, 1-REC, $h = \frac{1}{4}$ CPM. Natural binary labeling assumed, $11 \rightarrow +3$, $01 \rightarrow -1$.

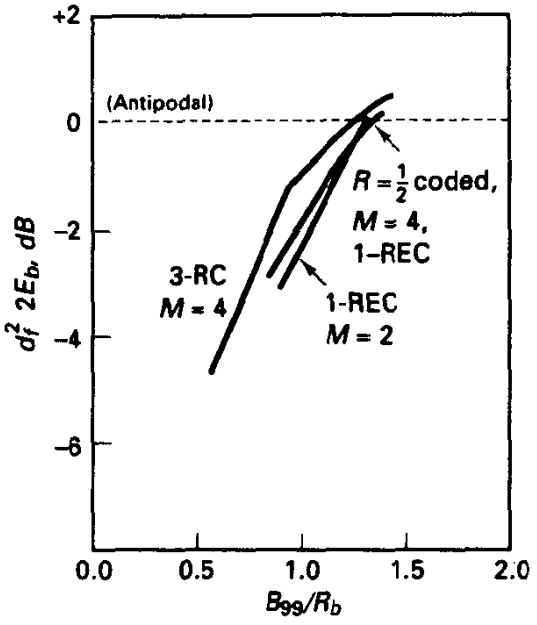


Figure 6.7.15 Energy and spectral efficiency of some CPM schemes; h varies along curves.

APPENDIX 6A1: NUMERICAL EVALUATION OF TRANSFER FUNCTION BOUNDS

Whereas direct algebraic solution for $T(W, I)$ was reasonable for the memory-2, $R = \frac{1}{2}$ encoder used as an example in this chapter, such techniques rapidly become unwieldy as the number of states increases, and these cases are, after all, of primary interest. Thus, it is important to demonstrate that efficient numerical methods exist for computing the transfer function bound.

If we consider the approach used in Example 6.14 in a more general setting, note that an S-state convolutional code has a split-state diagram with an input node, assigned value 1, an output node, whose value is the desired transfer function $T(W, I)$, and $S - 1$ internal nodes. The system of equations can be written in the form

$$\begin{aligned} \mathbf{v} &= \mathbf{A}\mathbf{v} + \mathbf{b}, \\ T(W, I) &= \mathbf{c}\mathbf{v}. \end{aligned} \tag{6A1.1}$$

(This assumes that there are no 1-step events from input to output, or parallel transitions, discussed later.) In this formulation, $\mathbf{v}$ is a vector with $S - 1$ elements representing internal node values. $\mathbf{A}$ is an $S - 1$ by $S - 1$ matrix of path gains in the variables W, I, where A_{ij} specifies the gain from state j to state i in the graph (caution here!). Finally, $\mathbf{b}$ and $\mathbf{c}$ are, respectively, column and row vectors with $S - 1$ entries specifying the connection of the unit input to the internal nodes and connection from internal nodes to the output node.

We are interested in the solution to this linear system of equations. From the first we have

$$\mathbf{v} = [\mathbf{I} - \mathbf{A}]^{-1}\mathbf{b}, \tag{6A1.2}$$

and from the second it follows that

$$T(W, I) = \mathbf{c}(\mathbf{I} - \mathbf{A})^{-1}\mathbf{b}, \tag{6A1.3}$$

which will be a polynomial in W, I.

In (6A1.3), the matrices are all expressed in terms of the form $W^w I^i$, and we could solve (6A1.3) to obtain the algebraic form of the transfer function. However, we are most frequently interested in the *numerical* value of $T(W, I)$ upon substitution of values for W and I, as described in Section 6.4. For example, on the AWGN channel with antipodal modulation and soft-decision decoding, the appropriate substitutions are $W = e^{-RE_b/N_0}$ and $I = 1$. We may as well make these substitutions at the outset, obtaining matrices of real numbers that depend on the code structure and the channel parameters. The matrix inversion and multiplication indicated in (6A1.3) are then accomplished numerically. There is a potential need to evaluate a large matrix; for example, with the $m = 6$, $R = \frac{1}{2}$ code, the inverse of a 63 by 63 matrix must be computed. This inverse can be computed, if desired, by the rapidly convergent power series

$$[\mathbf{I} - \mathbf{A}]^{-1} = \mathbf{I} + \mathbf{A} + \mathbf{A}^2 + \mathbf{A}^3 + \cdots, \tag{6A1.4}$$

where $\mathbf{A}^2$ is interpreted as $\mathbf{A} \cdot \mathbf{A}$, for example.

The following example using the four-state $R = \frac{1}{2}$ code should illustrate this numerical method.

Example 6A1.1 Application to $m = 2, R = \frac{1}{2}$ Code, Antipodal Modulation, AWGN Channel

Let's index the three internal states of the split-state transition diagram in Figure 6.4.2 as follows: (01) → 1, (10) → 2, and (11) → 3. Inspection of the flow graph equations shows that

$$\mathbf{A} = \begin{bmatrix} 0 & W & W \\ I & 0 & 0 \\ 0 & WI & WI \end{bmatrix}, \qquad \mathbf{b} = \begin{bmatrix} 0 \\ W^2 I \\ 0 \end{bmatrix}, \qquad \mathbf{c} = [W^2, 0, 0]. \tag{6A1.5}$$

$\mathbf{b}$ reflects the connection of the input to the graph, and $\mathbf{c}$ reflects the connection to the desired output node.

Suppose that we wish to bound P_e at $E_b/N_0 = 4$, or 6 dB. From the discussion of Section 6.4, we find this bound requires $T(W, I)$ with $I = 1$ and $W = e^{-RE_b/N_0} = e^{-2}$. The $\mathbf{A}$ matrix is thus

$$\mathbf{A} = \begin{bmatrix} 0 & e^{-2} & e^{-2} \\ 1 & 0 & 0 \\ 0 & e^{-2} & e^{-2} \end{bmatrix}, \tag{6A1.6a}$$

whereas

$$\mathbf{b} = \begin{bmatrix} 0 \\ e^{-4} \\ 0 \end{bmatrix}, \qquad \mathbf{c} = [e^{-4}, 0, 0]. \tag{6A1.6b}$$

Substituting these into (6A1.3) yields $\mathbf{c}(\mathbf{I} - \mathbf{A})^{-1}\mathbf{b} = 6.2 \cdot 10^{-5}$. From (6.4.22), we then have

$$P_e \le Q\left[\left(\frac{2E_b}{N_0} d_f R\right)^{1/2}\right] e^{(E_b/N_0) d_f R}\, T(W, I)|_{I=1,\, W=e^{-RE_b/N_0}} = 5.3 \cdot 10^{-6}. \tag{6A1.7}$$

To obtain numerical bounds on P_s, we recall from the discussion of Section 6.4 that the partial derivative of $T(W, I)$ is required, with appropriate substitution for W, I. The derivative may be approximated by

$$\left.\frac{\partial T(W, I)}{\partial I}\right|_{I=1} = \frac{T(W, 1+\delta) - T(W, 1)}{\delta}, \tag{6A1.8}$$

where δ is small, say 0.01. Both numerator terms may be calculated using the matrix methods discussed previously.

In summary, once the requisite matrices **A**, **b**, and **c** are carefully constructed for a given code, computation of the bounds is simple numerical work for a computer. The error probability curve is then constructed by stepping E_b/N_0 in increments of perhaps $\frac{1}{2}$ dB, each time modifying the numerical values in the various matrices. We also emphasize that the procedure is general to many channel models and modulation strategies, requiring only appropriate substitution for the W variable.

An extension to this procedure can handle situations with parallel state transitions; that is, the input node connects directly to the output node through a 1-step error event. We simply rewrite the second equation of (6A1.3) as

$$T(W, I) = \mathbf{cv} + \mathbf{d}, \tag{6A1.9}$$

where **d** specifies path gain from input to output. The solution to this system becomes

$$T(W, I) = \mathbf{c}(\mathbf{I} - \mathbf{A})^{-1}\mathbf{b} + \mathbf{d}. \tag{6A1.10}$$

In Section 6.6, we discussed a modified transfer function bounding approach due to Zehavi and Wolf [64], applicable to certain trellis codes possessing reasonable symmetry properties. This procedure formulated a modified signal flow graph, whose solution can be numerically evaluated using the methodology presented here. Benedetto [65] provides a comprehensive survey of techniques for performance evaluation of trellis codes.

BIBLIOGRAPHY

1. Elias, P., "Coding for Noisy Channels," *IRE Convention Record*, vol. 3, part 4, 1955.
2. Forney, G. D., Jr., "Convolutional Codes I: Algebraic Structure," *IEEE Trans. Information Theory*, vol. IT-16, pp. 720–738, 1970.
3. Viterbi, A. J., "Error Bounds for Convolutional Codes and Asymptotically Optimum Decoding Algorithm," *IEEE Trans. Information Theory*, vol. IT-13, pp. 260–269, 1967.
4. Ungerboeck, G., "Channel Coding with Amplitude/Phase Modulation," *IEEE Trans. Information Theory*, vol. IT-28, pp. 55–67, January 1982. See also *IEEE Communications Magazine*, vol. 25, pp. 5-21, February 1987.
5. Sundberg, C-E. W., "Continuous-phase Modulations, Parts I and II," *IEEE Communications Magazine*, vol. 24, pp. 25–38, April 1986.
6. Lin, S., and Costello, D. J., *Error Control Coding: Fundamentals and Applications*, Prentice Hall, Englewood Cliffs, NJ., 1983.

7. Odenwalder, J. P., "Dual-k Convolutional Codes for Noncoherently Demodulated Channels, *Proc. Int'l Telemetering Conference*, pp. 165–174, September 1976.
8. Trumpis, B. D., "Convolutional Coding for M-ary Channels," Ph.D. dissertation, University of California, Los Angeles, 1975.
9. Massey, J. L., and Sain, K. K., "Inverses of Sequential Circuits," *IEEE Trans. Computers*, vol. C-17, pp. 330–337, April 1968.
10. Heller, J., "Short Constraint Length Convolutional Codes," *JPL Space Programs Summary*, 37–54, vol. 3, pp. 171–177, 1967; "Improved Performance of Short Constraint Length Convolutional Codes," *JPL Space Programs Summary*, 37–54, vol. 3, pp. 83–84, 1967.
11. Costello, D. J., "Free Distance Bounds for Convolutional Codes," *IEEE Trans. Information Theory*, vol. IT-20, pp. 356–365, May 1974.
12. Larsen, K. J., "Short Convolutional Codes with Maximal Free Distance for Rates 1/2, 1/3, and 1/4," *IEEE Trans. Information Theory*, vol. IT-19, pp. 371–372, May 1973.
13. Odenwalder, J. P., "Optimal Decoding of Convolutional Codes," Ph.D. dissertation, University of California, Los Angeles, 1970.
14. Daut, D., Modestino, J. W., and Wismer, L. D., "New Short Constraint Length Convolutional Code Construction for Selected Rational Rates," *IEEE Trans. Information Theory*, vol. IT-28, pp. 793–799, September 1982.
15. Paaske, E., "Short Binary Convolutional Codes with Maximal Free Distance for Rates 2/3 and 3/4," *IEEE Trans. Information Theory*, vol. IT-20, pp. 683–689, September 1974.
16. Lee, L. N. "Short Unit-Memory Byte-Oriented Binary Convolutional Codes Having Maximal Free Distance," *IEEE Trans. Information Theory*, vol. IT-22, pp. 349–352, May 1976.
17. Conan, J., "The Weight Spectra of Some Short Low-rate Convolutional Codes," *IEEE Trans. Communications*, vol. COM-32, pp. 1050–1053, September 1984.
18. Ryan, W. E., and Wilson, S. G., "Two Classes of Convolutional Codes over GF(q) for q-ary Orthogonal Signaling," *IEEE Trans. Communications*, vol. COM-39, pp. 30–40, January 1991.
19. Cain, J. B., Clark, G. C., and Geist, J. M., "Punctured Convolutional Codes of Rate $(n-1)/n$ and Simplified Maximum Likelihood Decoding," *IEEE Trans. Information Theory*, vol. IT-25, pp. 97–100, January 1979.
20. Haccoun, D., and Begin, G., "High-rate Punctured Convolutional Codes," *IEEE Trans. Communications*, vol. COM-37, pp. 1113–1125, November 1989.
21. Yasuda, Y., Kashiki, K., and Hirata, Y., *IEEE Trans. Communications*, vol. COM-32, pp. 315–319, March 1984.
22. Hagenauer, J., "Rate-compatible Punctured Convolutional Codes and Their Applications," *IEEE Trans. Communications*, vol. COM-36, pp. 389–400, April 1988.
23. Johannesson, R., "Some Rate 1/3 and 1/4 Binary Convolutional Codes with an Optimum Distance Profile," *IEEE Trans. Information Theory*, vol. IT-23, pp. 281–283, 1977.
24. Johannesson, R., "Some Long Rate One-half Binary Convolutional Codes with an Optimum Distance Profile," *IEEE Trans. Information Theory*, vol. IT-22, pp. 629–631, 1976.
25. Johannesson, R., and Paaske, E., "Further Results on Binary Convolutional Codes with an Optimum Distance Profile," *IEEE Trans. Information Theory*, vol. IT-24, pp. 264–268, 1978.
26. Johannesson, R., "Robustly Optimal Rate One-half Binary Convolutional Codes," *IEEE Trans. Information Theory*, vol. IT-21, pp. 464–468, July 1975.
27. Omura, J. K., "On the Viterbi Decoding Algorithm," *IEEE Trans. Information Theory*, vol. IT-15, pp. 177–179, 1969.

28. Forney, G. D., Jr., "The Viterbi Algorithm," *IEEE Proceedings*, vol. 61, pp. 268–278, 1973.
29. Bellman, R., *Dynamic Programming*, Princeton University Press, Princeton, N. J., 1957.
30. Bahl, L. R., Cocke, J., Jelinek, F., and Raviv, J., "Optimal Decoding of Linear Codes for Minimizing Symbol Error Rate," *IEEE Trans. Information Theory*, vol. IT-20, pp. 248–287, March 1974.
31. Lee, L. N., "Real-time Minimum Bit Error Probability Decoding of Convolutional Codes," *IEEE Trans. Communications*, vol. COM-22, pp. 146–150, February 1974.
32. Qualcomm, Inc. data sheet, integrated circuit Viterbi decoders, San Diego, CA.
33. Stanford Telecommunications data sheet, part number STEL-2030A.
34. Hemmati, F., and Costello, D. J., "Truncation Error Probability in Viterbi Decoding," *IEEE Trans. Communications*, vol. COM-25, pp. 530–532, May 1977.
35. Rader, C. M., "Memory Management in a Viterbi Decoder," *IEEE Trans. Communications*, vol. COM-29, pp. 1399–1401, September 1981.
36. Hekstra, A. P., "An Alternative to Metric Rescaling in Viterbi Decoders," *IEEE Trans. Communications*, vol. COM-37, pp. 1220–1222, November 1989.
37. Fettweis, G., and Meyr, H., "Parallel Viterbi Algorithm Implementations: Breaking the ACS Bottleneck," *IEEE Trans. Communications*, vol. COM-37, pp. 785–790, August 1989.
38. Gulak, P. G., and Kailath, T., "Locally Connected VLSI Architectures for the Viterbi Algorithm," *IEEE Selected Areas in Communications*, vol. SAC-6, No. 3, April 1988.
39. Gulak, P. G., and Shwedk, E., "VLSI Structures for Viterbi Receivers, Part I," *IEEE Journal on Selected Areas in Communications*, vol. SAC-4, pp. 142–154, January 1986.
40. Viterbi, A. J., "Convolutional Codes and Their Performance in Communication Systems, *IEEE Trans. Communications*, vol. COM-19, pp. 751–772, October 1971.
41. Mason, S. J., "Feedback Theory—Further Properties of Signal Flow Graphs," *Proceedings of IRE*, vol. 44, pp. 920–926, July 1956.
42. Odenwalder, J. P., "Dual-*k* Convolutional Codes for Noncoherently Demodulated Channels, *Proc. Int'l Telemetering Conference*, pp. 165–174, September 1976.
43. Heller, J., and Jacobs, I. M., "Viterbi Decoding for Satellite and Space Communication, *IEEE Trans. Communications*, vol. COM-19, pp. 835–848, October 1971.
44. Van de Meeberg, L., "A Tightened Upper Bound on the Error Probability of Binary Convolutional Codes with Viterbi Decoding," *IEEE Trans. Information Theory*, vol. IT-20, pp. 389–391, May 1974.
45. Simon, M. K., Omura, J. K., Scholz, R., and Levitt, B. K., *Spread Spectrum Systems*, Computer Science Press, Rockville, MD, 1985.
46. Bucher, E. A., "Coding Options for Efficient Communication on Non-Stationary Channels," *National Telecommunications Conference Record*, December 1980.
47. Viterbi, A. J., "Orthogonal Tree Codes for Communicating in the Presence of White Gaussian Noise," *IEEE Trans. Communications*, vol. COM-15, pp. 238–242, April 1967.
48. Modestino, J. W., and Mui, S. Y., "Convolutional Codes on Rician Fading Channels," *IEEE Trans. Communications*, pp. 592ff, 1976. Also *IEEE Trans. Communications*, pp. 1075ff, 1977.
49. Lee, P. J., "New Short Constraint Length, Rate 1/*N* Convolutional Codes Which Minimize the Required SNR for Given Desired Bit Error Rates," *IEEE Trans. on Communications*, vol. COM-33, pp. 171–177, February 1985.
50. Clark, G., and Cain, J. B., *Error Correction Coding for Digital Communications*, Plenum, New York, 1981.

51. Wozencraft, J. M., and Reiffen, B., *Sequential Decoding*, MIT Press, Cambridge, MA, 1963.

52. Zigangirov, K., "Some Sequential Decoding Procedures," *Problemy Peredachi Informatsii*, vol. 2, pp. 13–25, 1966.

53. Jelinek, F., "A Fast Sequential Decoding Algorithm Using a Stack," *IBM J. Research and Development*, vol. 13, pp. 675–685, 1969.

54. Massey, J. L., "Variable Length Codes and the Fano Metric," *IEEE Trans. Information Theory*, vol. IT-18, pp. 196–198, 1972.

55. Begin, G., and Haccoun, D., "Performance of Sequential Decoding of High-rate Punctured Convolutional Codes, *IEEE Trans. Communications*, vol. COM-42, pp. 966–978, February 1994.

56. Heller, J. A., "Feedback Decoding of Convolutional Codes," in *Advances in Communication Systems*, Academic Press, New York, 1975.

57. Divsalar, D., Simon, M. K., McLane, P., and Biglieri, E.," *Trellis Coded Modulation*, Macmillan, New York, 1991.

58. Forney, G. D., Jr., "Coset Codes: Part 1," *IEEE Trans. Information Theory*, vol. IT-34, pp. 1123–1151, September 1988.

59. Wilson, S. G., Schottler, P. J., Sleeper, H. A., and Lyons, M. T., "Rate 3/4 Trellis Coded 16-PSK: Code Design and Performance Evaluation," *IEEE Trans. Communications*, vol. COM-32, pp. 1308–1315, December 1984.

60. Biglieri, E., "Ungerboeck Codes Do Not Shape the Power Spectrum," *IEEE Trans. Information Theory*, vol. IT-32, pp. 595–596, July 1986.

61. Pietrobon, S. S., Deng, R. H., Lafanechere, A., Ungerboeck, G., and Costello, D. J., "Trellis-coded Multi-dimensional Phase Modulation, *IEEE Trans. Information Theory*, vol. IT-36, January 1990.

62. Wei, L.-F., "Rotationally Invariant Convolutional Channel Coding with Expanded Signal Space, Part II: Nonlinear Codes," *IEEE Journal on Selected Areas in Communication*, vol. SAC-2, pp. 672–686, September 1984.

63. Calderbank, A. R., and Sloane, N. J. A., "An Eight-dimensional Trellis Code," *IEEE Proceedings*, vol. 74, pp. 757–759, May 1986.

64. Zehavi, E., and Wolf, J. K., "On the Performance Evaluation of Trellis Codes," *IEEE Trans. Information Theory*, vol. IT-32, March 1987.

65. Benedetto, S., Mondin, M., and Montorsi, G., "Performance Evaluation of Trellis-coded Modulation Schemes," to appear, *IEEE Proceedings*, 1995.

66. Eyuboglu, M. V., Forney, G. D., Jr., Dong, P., and Long, G., "Advanced Modulation Techniques for V. Fast," *European Transactions on Telecommunication*, vol. 4, no: 3., pp. 243–256, May 1993.

67 Forney, G. D., Jr., "Geometrically Uniform Codes," *IEEE Trans. Information Theory*, vol. IT-37, pp. 1241–1260, September 1991.

68. Leung, Y. S., and Wilson, S. G., "Trellis Coding for Fading Channels," *ICC Conference Record*, Seattle, 1987.

69 Divsalar, D., and Simon, M. K., "The Design of Trellis-coded MPSK for Fading Channels: Set Partitioning for Optimum Code Design," *IEEE Trans. Communications*, vol. COM-36, pp. 1004–1011, September 1988.

70. Divsalar, D., and Simon, M. K., "Multiple Trellis-coded Modulation," *IEEE Trans. Communications*, vol. COM-36, pp. 410–419, April 1988.

71. Ho, P., and Fung, D., "Error Performance of Interleaved Trellis-coded PSK Modulations in Correlated Rayleigh Fading Channels," *IEEE Trans. Communications*, vol.COM-40, pp.1800–1809, December 1992.

72. Sundberg, C.-E. W., and Seshadri, N., "Coded Modulation for Fading Channels: An Overview," *European Transactions on Telecommunications*, pp. 309–324, May 1993.

73. Anderson, J. B., Aulin, T., and Sundberg, C.-E. W., *Digital Phase Modulation*, Plenum, New York, 1986.

74. Anderson, J. B., and Sundberg, C-E. W., "Advances in Constant Envelope Coded Modulation," *IEEE Communications Magazine*, pp. 36–45, December 1991.

75. Rimoldi, B., "A Decomposition Approach to CPM," *IEEE Trans. Information Theory*, vol. IT-34, pp. 260–270, March 1988.

76. Baker, T. J., "Asymptotic Behavior of Digital FM Spectra," *IEEE Trans. Communications*, vol. COM-22, pp. 1585–1594, October 1974.

77. Maseng, T., "Digital Phase Modulated Signals," *IEEE Trans. Communications*, vol. COM-33, pp. 911–918, September 1985.

78. Rimoldi, B., "Design of Coded CPFSK Modulated Systems for Bandwidth and Energy Efficiency," *IEEE Trans. Communications*, vol. COM-37, pp. 297–308, September 1989.

79. Morales-Moreno, F., Holubowicz, W., and Pasupathy, S., "Optimization of Trellis-coded TFM via Matched Codes," *IEEE Trans. Communications*, vol. COM-42, pp. 1586–1594, February 1994.

80. Anderson, R. R., and Salz, J., "Spectra of Digital FM," *Bell System Tech. Journal*, vol. 44, pp. 1165–1189, 1966.

81. Aulin, T., and Sundberg, C.-E. W., "An Easy Way to Calculate the Power Spectra for Digital FM," *IEE Proceedings*, Part F, vol. 130, pp. 519–526, October 1983.

82. Mulligan, M., and Wilson, S. G., "An Improved Algorithm for Evaluating the Free Distance of Trellis Phase Codes," *IEEE Trans. Information Theory*, vol. IT-30, pp. 846–851, November 1984.

83. Aulin, T., "Symbol Error Probability Bounds for Coherently Viterbi Detected Continuous Phase Modulated Signals," *IEEE Trans. Communications*, vol. Com-29, pp. 1707–1714, November 1981.

84. Simmons, S., and Wittke, P. H., "Low-complexity Decoders for Constant Envelope Digital Modulation," *IEEE Trans. Communications*, vol. COM-31, pp. 1273–1280, December 1983.

85. Svensson, A., Sundberg, C.-E. W., and Aulin, T., "A Class of Reduced Complexity Viterbi Detectors for Partial Response Continuous Phase Modulation," *IEEE Trans. Communications*, vol. COM-32, pp. 1079–1087, October 1984.

86. Taylor, D. P., and Anderson, J. B., "A Bandwidth-efficient Class of Signal Space Codes," *IEEE Trans. Information Theory*, vol. IT-24, pp. 703–712, November 1978.

87. Wilson, S. G., and Gaus, R. C., "Power Spectrum of Multi-*h* Phase Codes," *IEEE Trans. Communications*, vol. COM-29, pp. 250–257, March 1981.

88. Lindell, G., Sundberg, C.-E. W., and Aulin, T., "Minimum Euclidean Distance for Combinations of Short-rate 1/2 Convolutional Codes and CPFSK Modulation," *IEEE Trans. Information Theory*, pp. 509–519, May 1984.

89. Pizzi, S. V., and Wilson, S. G., "Convolutional Coding Combined with Continuous Phase Modulation," *IEEE Trans. Communications*, vol. COM-33, pp. 20–29, January 1985.

90. Wilson, S. G., "Rate 5/6 Trellis Coded 8-PSK," *IEEE Trans. Communicatuions*, vol. COM-34, pp. 1045–1049, October 1986.

EXERCISES

6.1.1. An $R = \frac{1}{2}$, $\nu = 2$ systematic feedback-free encoder for a convolutional code is implemented with the shift-register encoder shown in Figure P6.1.1.

(a) Draw the state transition diagram and label each transition with $W^x I^y$.

(b) Find an expression for $T(W, I)$ and determine that the free distance is 4. (The nonsystematic encoder with the same parameters discussed in the text achieved $d_f = 5$.)

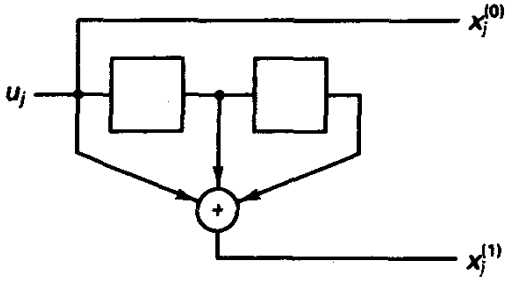


Figure P6.1.1

6.1.2. Complete the labeling of the state transition diagram shown in Figure 6.1.11 for a $R = \frac{2}{3}$ convolutional code and draw the corresponding trellis diagram for two levels.

6.1.3. An $R = \frac{2}{3}$ encoder has the diagram shown in Figure P6.1.3.

(a) What are the memory order and the encoding constraint length?

(b) Identify the state variables and argue that the resulting trellis has eight states.

(c) Formulate the matrices $\mathbf{G}_i$, $i = 0, 1, 2$.

(d) This code is known to have $d_f = 4$. Find one input sequence having output weight 4.

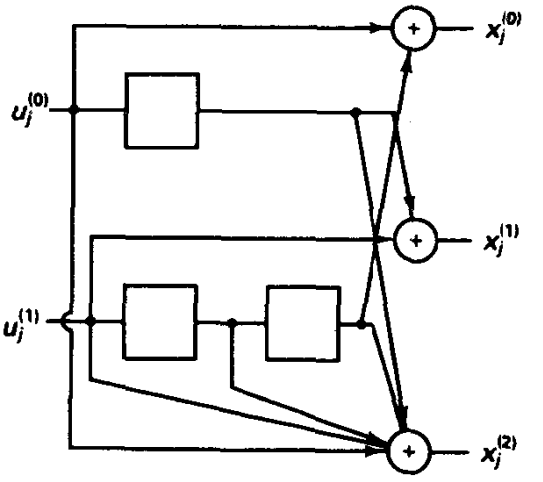


Figure P6.1.3

6.1.4. A rate $\frac{3}{4}$ binary convolutional code is defined by the generators

$$\mathbf{G}_0 = \begin{bmatrix} 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 \end{bmatrix}, \qquad \mathbf{G}_1 = \begin{bmatrix} 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

(a) Diagram the encoder.

(b) What are the encoder constraint length n_E and the memory order m?

(c) Is the code produced in systematic form?
(d) How many states does the encoder possess?
(e) How many trellis branches enter/exit each state?

6.1.5. Lee [16] discusses unit-memory convolutional codes, that is, with $m = 1$, but where k and n are not relatively prime. The G matrices for an $R = \frac{4}{8} = \frac{1}{2}$ code with unit memory are

$$G_0 = \begin{bmatrix} 1 & 0 & 0 & 1 & 0 & 1 & 1 & 1 \\ 0 & 1 & 0 & 0 & 1 & 0 & 1 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 \end{bmatrix}, \quad G_0 = \begin{bmatrix} 1 & 0 & 0 & 0 & 1 & 0 & 1 & 1 \\ 1 & 1 & 1 & 0 & 0 & 0 & 1 & 0 \\ 1 & 0 & 1 & 1 & 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 1 & 0 & 0 & 0 & 1 \end{bmatrix}$$

Draw a gate diagram of the encoder, and describe the corresponding trellis. How many information and code symbols are attached to each branch?

6.1.6. A 16-state feedback free nonsystematic convolutional code has generators $\mathbf{g}_0 = 37_8$ and $\mathbf{g}_1 = 21_8$. (This is not an OFD code.) A systematic-form encoder with output feedback is shown in Figure P6.1.6. Argue that the two code generators have equal number of states. The output sequence $(10, 10, 10, \ldots)$ is in the code produced by the first generator. If the two generators produce equivalent codes, then this same sequence must be producible by the second generator, perhaps with a different input sequence. Is this the case?

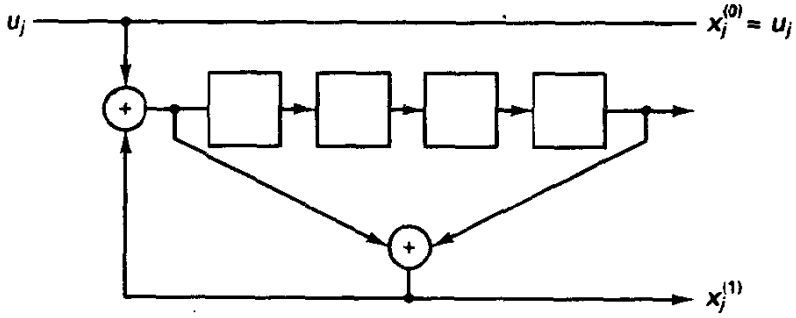


Figure P6.1.6

6.2.1. (a) For the $R = \frac{2}{3}$ code of Example 6.2, compute the column distance profile to depth $L = 4$ by finding the minimum distance corresponding to all message input sequences of length 1, 2, 3, and 4 bits. We know this code to be noncatastrophic, so find the free distance by examining short remergent sequences. Argue that $d_f = 4$.
(b) If a decoder is forced to operate with a delay corresponding to two trellis levels, determine the error-correcting capability of this decoder for a BSC. (Assume that the decoder has operated correctly until the given time.)

6.2.2. The code shown in Figure 6.1.1c is an ODP code for $R = \frac{1}{3}$ and $m = 13$. State precisely what this means. It is known that the free distance of this code is 18. Find an input yielding a code sequence with weight 18; note that it corresponds to message that has more than a single 1.

6.2.3. Johannesson [24] described ODP convolutional codes and lists a systematic $R = \frac{1}{2}$ code with $\mathbf{g}_1 = 67115143$ in right-justified octal notation. Determine the column distance function of this code to depth $L = 4$.

6.2.4. The (almost) industry standard $R = \frac{1}{2}$ convolutional code has $\nu = 6$ and is nonsystematic with generators $\mathbf{g}_0 = 133$ and $\mathbf{g}_1 = 171$. Another code found by Johannesson is specified by 135 and 147, respectively. The latter code is claimed to have optimal free distance of

10, equivalent to the first code, but apparently its distance profile is superior. Investigate whether this is the case.

6.2.5. Suppose that a binary convolutional code is employed in conjunction with hard-decision decoding. In such a case, we often refer to the decoder as an error-correcting decoder. What claims can be made for error detection and error correction given a decoder that is maximum likelihood? What about a decoder that uses a memory of L_D levels? (*Hint*: The latter is related to the distance profile.)

6.2.6. Two $R = \frac{1}{2}$, $m = 3$ codes have generators $\mathbf{g}_0 = 15$ and $\mathbf{g}_1 = 17$ and $\mathbf{g}_0 = 13$ and $\mathbf{g}_1 = 17$, respectively. Draw the encoders, and observe that the first encoder is merely the time reversal of the other. Argue that this implies that the two codes have equivalent free distance, which is 6. The first code, however, has a slightly better distance profile, which would mandate its choice in sequential decoding or trellis decoding with small memory.

6.2.7. Two $R = \frac{1}{2}$ codes with $m = 3$ are specified by generators 13 and 15 and by 17 and 15, respectively. Show that both have $d_f = 6$, but that the second has fewer paths with weight equaling the free distance.

6.2.8. Determine which if any of the encoders of Figure P6.2.8 are catastrophic by testing the generator polynomials for nontrivial common factors.

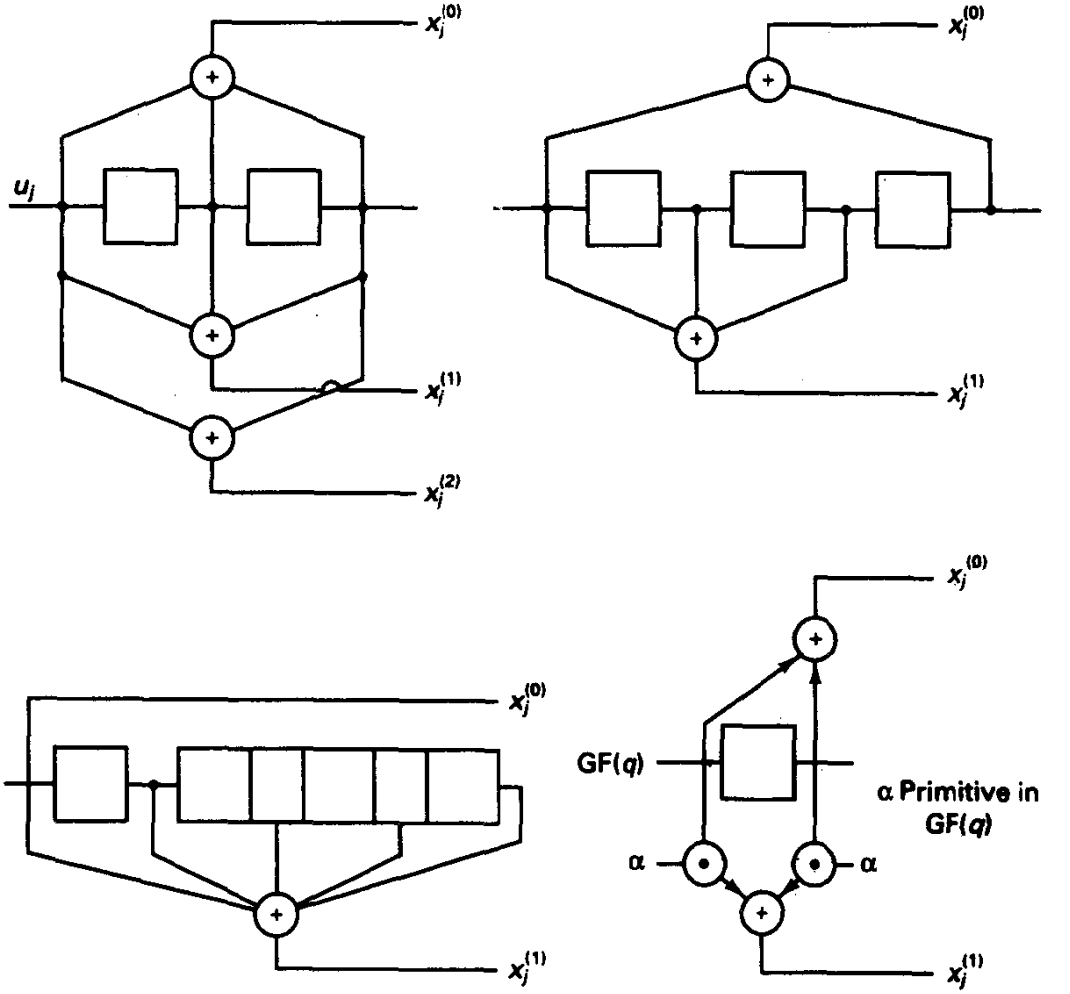


Figure P6.2.8

6.2.9. A rate $\frac{2}{3}$ binary convolutional encoder is described by the system matrix

$$\mathbf{G}(D) = \begin{bmatrix} 1+D & 1 & D \\ 1+D+D^2 & 1+D & D^2 \end{bmatrix}$$

Is the corresponding code catastrophic?

6.2.10. Consider the $R = \frac{1}{2}$ code produced by Figure P6.2.10.

(a) Show that this code has minimum distance among split/merge error events of 4.

(b) Show that the code is, however, catastrophic, with some information sequence of infinite weight producing a code sequence of weight that does not grow beyond 2.

(c) By drawing the trellis for this code, show an example for decoding on the BSC that exhibits catastrophic decoding behavior, assuming that the all-zeros sequence is transmitted.

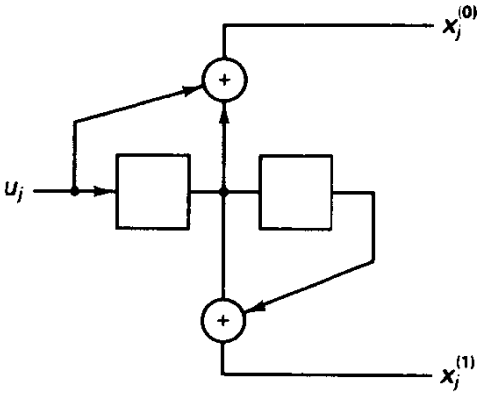


Figure P6.2.10

6.2.11. Consider an $R = \frac{1}{2}$ 8-ary convolutional code defined by the polynomials $g_0(D) = 1 + \alpha D$ and $g_1(D) = \alpha + \alpha^2 D$. How many states does the encoder have? Show that the code is catastrophic by testing whether the polynomials share a common factor other than D^m.

6.2.12. **(a)** Using the puncturing tables listed in Table 6.8, diagram how to produce an $R = \frac{7}{8}$ punctured code from a mother $R = \frac{1}{2}$ code.

(b) Describe the trellis—how many states? how does puncturing affect the number of symbols per branch?

(c) Can you find a distance-3 error event in the trellis, as claimed in the table? *Hint*: Consider free-distance events in the mother code, and examine the puncturing effect on these with different puncturing table phases.

6.2.13. Why is it that a decoder for a convolutional code can profit from having more memory than the encoding constraint length? (This is quite different than for block codes.) Couch your answer in terms of the column distance function.

6.2.14. One means of obtaining decent low-rate codes is to repeat the code symbols of a higher-rate code. For example, an $R = \frac{1}{4}$ code is obtainable from an $R = \frac{1}{2}$ code merely by duplicating the two code symbols of the latter code. Argue that the transfer function obtained in such cases is obtained by replacing W in the original transfer function by W^2. Use this construction to produce an $R = \frac{1}{4}$ code with $d_f = 20$ by modifying the memory order 6, $R = \frac{1}{2}$ OFD code listed in the text.

6.3.1. Make a longer version of the trellis diagram in Figure 6.1.10, and photocopy it for future use. Perform ML sequence decoding of the binary sequence $\tilde{\mathbf{r}} = (01, 00, 01, 10, 01, 00)$. Assume that the encoder is initialized in the zero state and terminates after six trellis levels in the same state. At each stage, for each survivor, tabulate the metric and trellis route. Can

you say whether an error occurred in decoding? How can you determine whether channel errors occurred?

6.3.2. Repeat Exercise 6.3.1 using the analog correlation metric for a received sequence $\tilde{\mathbf{r}} = (-1.2, 0.3, -0.5, -0.7, -1.3, 0.2, 0.4, -0.9, -1.4, 0.2, -0.1, -0.6)$. Note that this is a real sequence that, when binary quantized, would produce the received sequence of Exercise 6.3.1 Do both decodings produce the same estimate?

6.3.3. Use $\tilde{\mathbf{r}}$ of Exercise 6.3.2 and quantize to four levels, with thresholds at 0 and ± 0.7. We have not specified the signal-to-noise ratio of the channel producing these observations, but suppose that the four-level integer metrics shown in Figure 6.3.7 are used. Repeat the decoding exercise.

6.3.4. Investigate the decoder's selection of message in Example 6.12 if the decoder operates with four symbols of memory, releasing the oldest bit on the currently best survivor path.

6.3.5. (a) A digital signal processor chip being evaluated for implementation of the ML trellis decoding algorithm has a clock rate of 10 MHz, so the clock cycle time is 100 ns, and a butterfly section for an $R = \frac{1}{2}$ code can be evaluated in, say, 20 machine instructions. Typically, a machine instruction requires on the average 1.5 clock cycles. Ignoring time for memory traceback, metric overflow checking, and the like, how large a trellis can be decoded if the *information* rate is to be 19,200 bps?

(b) Based on the trellis size found in part (a) and adoption of an optimal code with this trellis size, determine the asymptotic coding gain (ACG) of such a code compared to uncoded antipodal signaling.

6.4.1. Verify the transfer function claimed for the dual-3 convolutional code in (6.4.10):

$$T(W) = \frac{7W^4 I}{1 - I[2W + 5W^2]}$$

Using long division, verify that the free distance is 4 and find the number of code sequences having this minimum distance.

6.4.2. Using the information weight enumeration given in Table 6.5 for the $R = \frac{1}{2}$, $\nu = 4$ OFD code, evaluate the upper bounds to decoder bit error probability for the following channel conditions:

(a) Antipodal signaling on AWGN channel with $E_b/N_0 = 6$ dB.

(b) BSC obtained from the preceding channel by binary quantizing.

(c) BSC obtained from binary orthogonal signaling on the AWGN channel with noncoherent detection at $E_b/N_0 = 10$ dB.

(d) Binary orthogonal signaling with noncoherent detection and square-law metric used in decoding, with the same SNR as in the previous case. This involves formulating $P_2(d)$ for the noncoherent, square-law combining channel; see Section 3.8.)

(e) Antipodal signaling on the coherent Rayleigh fading channel with $\bar{E}_b/N_0 = 14$ dB and perfect interleaving assumed.

(f) A jamming channel that randomly jams code symbols with probability $\frac{1}{2}$, and when noise is present the $E_b/N_0 = 3$ dB, making the "average" SNR 6 dB. Compare with part (a), which corresponds to full-time noise jamming with the same SNR. (Which is a better jamming strategy for a jammer with the given noise power resources?)

6.4.3. Repeat Exercise 6.4.2 for the $R = \frac{1}{3}$, $\nu = 6$ OFD code listed in the text.

6.4.4. For the $R = \frac{2}{3}$, $\nu = 3$ code listed in Table 6.3,

(a) Draw a nonsystematic, feedback-free encoder.

(b) Find a weight-4 sequence corresponding to a free-distance error sequence.

(c) Form the **A** matrix of one-step transition gains. Note that this is a 7×7 matrix, but in each row and column are four zeros corresponding to forbidden state transitions.

(d) Evaluate the upper bound on P_e, with $\epsilon = 0.02$, for a BSC using

$$P_e < T(W, I)_{|W=[4\epsilon(1-\epsilon)]^{1/2} I=1}$$

and

$$T(W) = \mathbf{c}(\mathbf{I} - \mathbf{A})^{-1}\mathbf{b}.$$

(See discussion of Appendix 6A1 for related material.)

6.4.5. (a) Use the numerical procedure presented in Appendix 6A1 to compute an upper bound on P_s for the $m = 2$, $R = \frac{1}{2}$ code when used with antipodal signaling on the AWGN channel. To get the tightest bound, use the expression

$$P_s \le Q\left[\left(\frac{2E_b}{N_0} d_f R\right)^{1/2}\right] e^{d_f R E_b/N_0} \left.\frac{\partial T(W, I)}{\partial I}\right|_{I=1, W=e^{-RE_b/N_0}}$$

Plot your result versus E_b/N_0 and compare with Figure 6.4.3.

(b) Repeat part (a) for the BSC obtained by quantizing the channel output to two levels.

6.4.6. Compute *lower* bounds on the decoded symbol error probability for the $R = \frac{1}{2}$ codes with 4, 16, and 64 states, and compare with the upper bounds shown in Figure 6.4.3.

6.4.7. Based on the dominant terms in the upper bounds for unquantized and hard-decision decoding on the antipodal AWGN channel, argue that the asymptotic loss in energy efficiency with hard-decision decoding is 3 dB.

6.4.8. Prove the inequality $Q\left[(x+y)^{1/2}\right] \le Q\left[(x)^{1/2}\right] e^{-y/2}$, $x, y \ge 0$. (The bound is tightest with $x > y$.)

6.4.9. An $R = \frac{1}{2}$ dual-3 encoder is to be coupled with 8-ary orthogonal modulation.

(a) Draw the encoder using GF(8) storage cells and GF(8) arithmetic symbology.

(b) Determine the number of trellis states, the number of branches per state, and the free distance in symbols.

(c) Evaluate the upper bound on symbol error probability when transmission is on an 8-ary noncoherent AWGN channel and square-law metrics are used.

(d) Repeat part (c) for the noncoherent Rayleigh channel. Note the appearance of fourth-order diversity.

(e) To achieve a better performance on the fading channel, we adopt $R = \frac{1}{4}$ coding with the dual-3 code by simply repeating both code symbols of the $R = \frac{1}{2}$ code. Argue that the free distance now doubles. (*Note*: This may be a net loss, however, on the nonfading channel due to "noncoherent" combining loss at low rates.)

6.5.1. Based on the stack algorithm description, explain why rapid increase in the column distance function is crucial for minimizing sequential decoding computation.

6.5.2. Repeat the stack decoding process found in Example 6.18, quantizing the real-valued demodulator output to eight levels and calculating appropriate metrics. Does the decoder produce the all-zeros path, corresponding to the VA solution with unquantized data?

6.5.3. (a) Consider the convolutional encoder discussed in Example 6.19, and let the all-zeros information sequence be sent. Suppose a channel error sequence corrupts an information position and the very next parity position. Perform the decoding steps to verify that this pattern is correctable.

(b) Repeat if the error pattern is in two parity positions spaced by five time steps.

6.5.4. Find a triple-error pattern that is not correctable by the feedback decoder in Example 6.19; that is, find a pattern with three errors, one of which is $e_{j-5}^{(0)}$, which gives a syndrome in the "no-correction" set of syndromes.

6.5.5. Formulate the feedback decoder for the systematic $R = \frac{1}{3}$ code with $m = 2$, having generators 4, 5, and 7.

(a) Show that the minimum distance, $d_c(3)$, for this code is 5.

(b) Determine the parity check matrix and diagram the syndrome generator, which in this case produces 2-bit sequences.

(c) Argue that, with decoder delay of $L_D = 2$, double-error correction is guaranteed.

6.6.1. The 4-state trellis code that is optimal in the free-distance sense for transmitting 2 bits per 8-PSK symbol is given in Figure P6.6.1, assuming natural binary labeling of constellation points around the circle. Show that this implements the trellis diagram of Figure 6.6.4. Note in particular that this code is structurally different than the best $R = \frac{2}{3}$, $\nu = 2$ Hamming distance code of Figure 6.1.1. Also note that, since one-step error events are dominant for the coded 8-PSK case, in moving to eight or more states, one-step events must be prevented by the encoder, meaning that additional memory placed in the encoder must be assigned to the presently uncoded input bit stream as well.

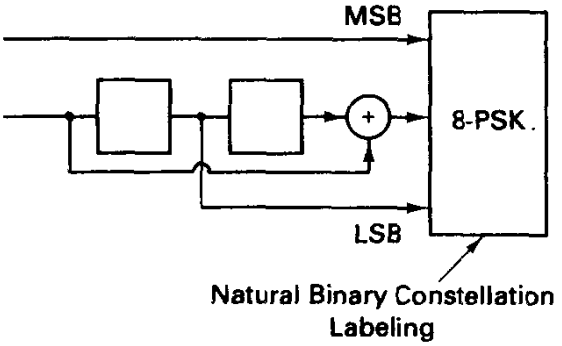


Figure P6.6.1

6.6.2. This will give practice with use of Viterbi's algorithm applied to the decoding of simple trellis codes. Suppose that we operate with the 4-state, $R = \frac{2}{3}$ coded 8-PSK design as discussed in the text. We know that the encoder begins in state 0 and ends in state 0 after the transmission of 10 bits (5 trellis levels). We receive the complex number sequence $\mathbf{r} = (1.1e^{j80}, 0.9e^{-j30}, 1.2e^{j160}, 0.7e^{j60}, 1.1e^{j200})$. All angles are in degree measure. Draw an appropriate trellis diagram and decode the sequence. In representing the signals and received data as complex numbers, the appropriate metric is of the form $\mathrm{Re}\{s_n r_n^*\}$. No bias term is necessary since all signals have the same energy. Remember to save time by first finding subset winners. What is your maximum likelihood sequence estimate?

6.6.3. (a) Perform set partitioning of the 32-point cross constellation shown in Figure P6.6.3. Continue the partitioning in divide-by-2 manner until subsets of size 4 are produced.

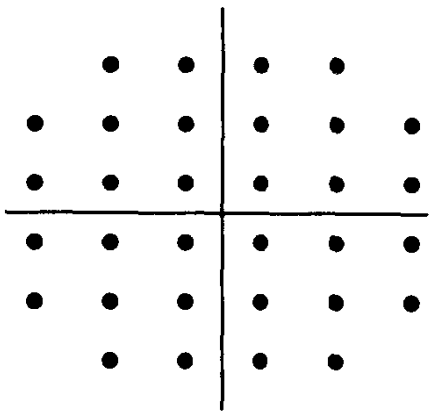


Figure P6.6.3

(b) Suppose that we wish to transmit $k = 4$ bits per modulator interval with a 4-state code. To provide the 16 branches per state, adopt two-sets-of-eight branching so that each supertransition amounts to labeling with a subset of size 8. Use Ungerboeck's rules to label the 4-state trellis. Here there are single-step, three-step, and four-step, and so on, error events. Find the minimum distance of the single (parallel) error events and the smallest three-step error event. What is the free distance of the code?

(c) If we wish an 8-state code with the same parameters, what change in structure is required to increase the free distance?

6.6.4. Using the information found in Table 6.1, diagram the optimal 64-state encoding system that sends $k = 2$ bits per 8-PSK symbol. Describe the optimal demodulator/decoder.

6.6.5. Repeat Exercise 6.6.4 for the case of $k = 3$ bits per 16-QAM symbol and 8-state encoding.

6.6.6. Show that the $R = \frac{1}{2}$, $\nu = 6$ code, when transmitted using PSK, is transparent to 180° ambiguities in the demodulator; that is, information sequences that are complements encode to code sequences related by complementation. Show, however, that if QPSK transmission is employed for the same code to conserve spectrum width the code is not transparent to 90° or 270° ambiguities.

6.6.7. Consider use of L=2 8-PSK symbol pairs to form a 64-ary constellation. Design a simple 2-state trellis code by hand that communicates $k = 5$ bits per pair of symbols by partitioning the 64-point constellation into four sets of 16 points. Calculate the free distance, and compare the spectral efficiency in bits per dimension and the energy efficiency to that of uncoded 8-PSK. (See Wilson [90] for discussion of this situation.)

6.7.1. Consider the pulse shape for CPM denoted by the half-cycle sinusoid, L-HCS, with

$$g(t) = \frac{\pi}{4LT_s} \sin\left(\frac{\pi t}{LT_s}\right), \qquad 0 \le t < LT_s$$

and zero elsewhere.

(a) Sketch the frequency pulse and the phase pulse $q(t)$, in particular showing that $q(t)$ has the adopted boundary condition $q(LT_s) = \frac{1}{2}$.

(b) Calculate successive derivatives of $q(t)$, sketching each as a function of time, and determine the smallest order of derivative for which the function is not everywhere continuous. (The answer is second-order.)

(c) Use the result of part (b) to predict the asymptotic rate of decay of spectral sidelobes for CPM with HCS shaping.

6.7.2. Verify the claim of the text that for L-RC shaping the spectral sidelobes diminish for large separation from the carrier as $(f - f_c)^{-8}$.

6.7.3. Sketch the phase tree for $N = 4$ levels when 2-REC shaping is employed (this is sometimes called duobinary FM). Assume binary ($M = 2$) transmission. Notice that in this case the phase tree presents piecewise constant segments as well as linearly increasing and decreasing segments.

6.7.4. A multi-h binary CPM code employs $h = \frac{2}{4}$ and $h = \frac{3}{4}$ on alternative bit intervals. The pulse shape is 1-REC, so phase trajectories are linear.

(a) Sketch four levels of the phase tree.

(b) Show that the shortest inevitable merger is three trellis levels, and find the free distance in terms of E_b. You should be able to show that the performance gain is about 1 dB over that of MSK, which employs fixed $h = \frac{1}{2}$.