5.10 PERFORMANCE EVALUATION FOR LINEAR BLOCK CODES

In this section we present specific numerical results concerning the performance of various coding techniques on several channels. Our coverage is certainly not exhaustive, but it should serve to illustrate the important methodology for other codes and channel scenarios. Several other cases are addressed in the Exercises. Using hard-decision decoding, we begin with the performance in an AWGN environment.

5.10.1 AWGN Channel, Hard-decision Decoding

In the case of hard-decision decoding, the demodulator provides a symbol-by-symbol estimate of the codeword, and, because the presumption is that these decisions are independent, the probability of block error for linear codes may be easily expressed in terms of the channel symbol error probability P_s. Our treatment in Chapter 3 provides extensive results on P_s for different modulation formats as a function of signal-to-noise ratio.

Suppose we employ an (n, k) linear block code over GF(q). We assume a bounded-distance decoder that decodes correctly if and only if $t = \lfloor \frac{d_{\min}-1}{2} \rfloor$ or fewer errors occur. (The standard BCH and RS decoding procedures are bounded-distance, and the performance of complete decoders is usually well approximated by assuming correct decoding ensues only when t or fewer errors occur.) With such a decoder, two types of events are possible when the number of errors exceeds t: The decoder may *decode incorrectly* when the received q-ary sequence **r** is within t units of another codeword, or the decoder may fail to decode, causing a *detected error* when **r** is not within t units of any codeword. We denote the respective probabilities of these events by P_{ICD} and P_{DE}. In some applications it is of little importance which event occurs. In others, where retransmission is perfectly acceptable when failures occur, and where undetected errors can be nearly intolerable, it is important to know the respective probabilities of the two events.

For the present, we shall focus on the probability of correct decoding, P_{CD}; this is related to the above quantities by

$$1 - P_{\text{CD}} = P_E = P_{\text{ICD}} + P_{\text{DE}}. \tag{5.10.1}$$

(The symbol P_E denotes the probability of not decoding a block correctly but should not be understood necessarily as the probability of a decoding error.)

Because the code is assumed linear, and we suppose the channel is uniform from the input (UFI), so that the symbol error probability is invariant to which code symbol is sent,[28] it is no loss of generality to assume that the codeword of 0's is selected for transmission. We then have

$$\begin{aligned} 1 - P_{\text{CD}} = P_E &= P(wt(\mathbf{e}) > t) \\ &= \sum_{i=t+1}^{n} C_i^n P_s^i (1 - P_s)^{n-i}. \end{aligned} \tag{5.10.2}$$

[28] Even if this is not the case, as say in QAM, use of the worst-case error probability will preserve an upper bound in the calculation.

Equation (5.10.2) and the relevant expression for P_s provide the means to calculate performance of any block code in conjunction with any hard-decision demodulation technique. Often, one finds presentations of P_E versus P_s, leaving the modulation and channel aspects outside the study. In many cases, however, we are interested in comparing the performance of the system with coding against one without, when the same modulation/demodulation strategy and channel setting are in force, and when the information throughput is held *fixed*. Normally, then, we seek P_E versus available signal-to-noise ratio, E_b/N_0.

To compute P_s we must first determine E_s/N_0, the code symbol energy-to-noise density ratio. This is in turn related to E_b/N_0, our standard measure of comparison, by

$$\frac{E_s}{N_0} = \frac{k}{n}(\log_2 q)\frac{E_b}{N_0}, \tag{5.10.3}$$

since the energy available per codeword transmission is $k(\log_2 q)E_b$ joules, and this is distributed among n code symbols.

To illustrate the calculations and introduce performance comparisons, we consider the binary (23, 12) Golay code. This code serves as a useful vehicle because it yields impressive coding performance, and because of the code's perfect nature, all error patterns with four or more errors in 23 bits produce incorrect decodings. Thus, a complete decoder and a bounded-distance decoder are equivalent here. We assume binary antipodal signaling, say with PSK, and binary demodulator quantization. The relevant P_s relation is given by

$$P_s = Q\left(\left(\frac{2E_s}{N_0}\right)^{1/2}\right) = Q\left(\left(\frac{24E_b}{23N_0}\right)^{1/2}\right). \tag{5.10.4}$$

In Figure 5.10.1 we present results for $P_E = 1 - P_{\text{CD}}$ versus E_b/N_0, and alongside compare the message error probability for transmission of an *uncoded* 12-bit message over the same channel employing the *same* E_b/N_0. We find that the coded system achieves a certain small message error probability, say 10^{-5}, with about 2.0 dB less E_b/N_0 than is required for the uncoded system. This gain is achieved at the expense of increased bandwidth (by $\frac{23}{12}$) and with some additional complexity. Nonetheless, the savings in energy is extremely beneficial on certain power-limited channels. In Exercise 5.10.1, a similar analysis is invited for the (7, 4) code, where the energy saving is not as large.

To compare different block coding techniques and their ability to deliver messages that are many blocklengths long, it is traditional to compute the delivered *symbol error probability* observed at the output of the decoder. We shall denote this quantity by P_o. This quantity may be calculated for linear block codes used on uniform-from-the-input channels by again assuming that the all-zeros word is selected for transmission. Given that $i > t$ errors occur, the decoder will fail to decode correctly, but when this occurs, only some (typically a small fraction) of the delivered message symbols are erroneous. In general, the exact calculation of the output error probability is tedious and requires information equivalent to the entire standard array for the code, as in Section 5.2. Given this array, it is possible to count how many errors are produced for each error pattern,[29]

[29] Note that the number of errors produced upon decoding is just $wt(\mathbf{e} - \hat{\mathbf{e}})$.

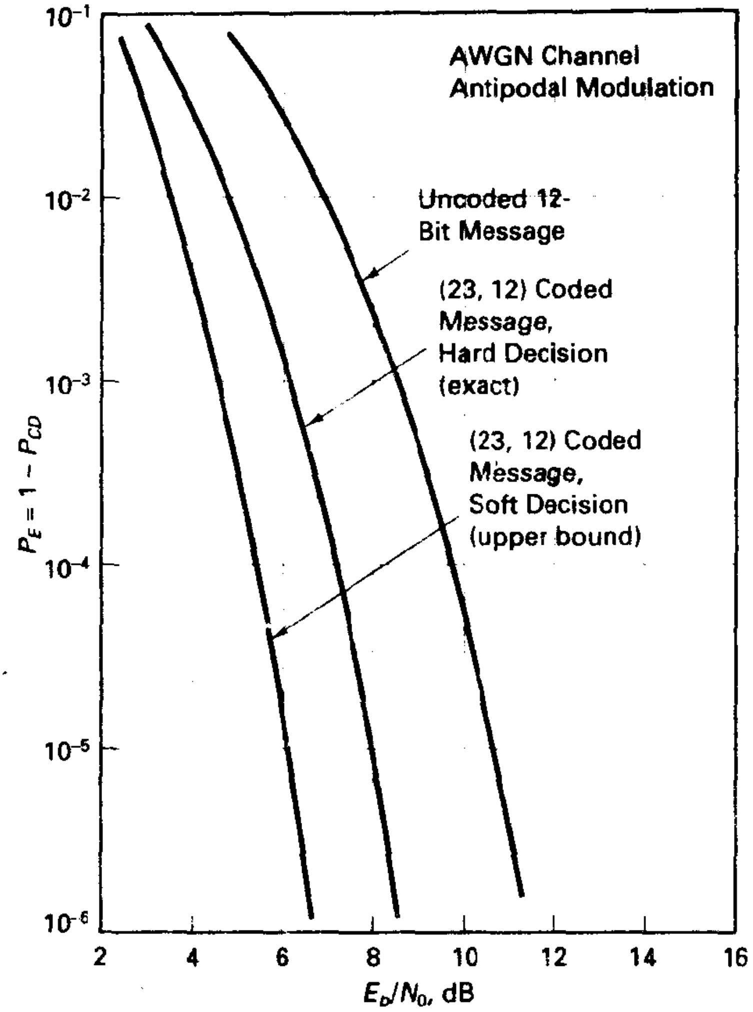


Figure 5.10.1 Probability of block error, Golay (23, 12) code.

weight these by the probability of the error pattern occurence, and sum. Such an analysis can also incorporate the distinction between decoding failures and incorrect decodings, if desired.

Generally this information is not available, or the computation would be prohibitively difficult, and an accurate approximation is the follcwing. Given j errors, if the decoder can decode, it will produce a code vector within t units of the received vector **r**. The triangle inequrlity states that the Hamming distance between the transmitted word and the recovered codeword is less than or equal $j+t$. (This is pessimistic in including decoding "failures," or detected error events, into the post-decoding error probability calculation.) For example, with the two-error-correcting (15, 7) code, if four errors occur, we can be assured that no more than 6 of the 15 code symbols are incorrect at the decoder output. Over selection of codewords for transmission and error patterns, there is no preference for location of symbol errors after decoding, and we can thus bound the output symbol error probability by

$$P_o \leq \sum_{j=t+1}^{n} \frac{j+t}{n} C_j^n P_s^j (1 - P_s)^{n-j}. \tag{5.10.5}$$

Another frequent approximation, good for small P_s, is that decoding errors are always to nearest-neighbor vectors, leaving $d_{\min}$ erroneous symbols in n positions, hence $P_o \approx (d_{\min}/n) P_E$. In the simplest view, we have the upper-bound

$$P_o \leq P_E, \tag{5.10.6}$$

since a block error produces message errors in at most all k symbol positions.

In Figure 5.10.2 we present results on P_o obtained using (5.10.5) for several binary codes with rate near $\frac{1}{2}$ and varying blocklengths. The codes are by now familiar: the (7, 4) Hamming code, a (15, 7) BCH code, the (23, 12) Golay code, and a (127, 64)

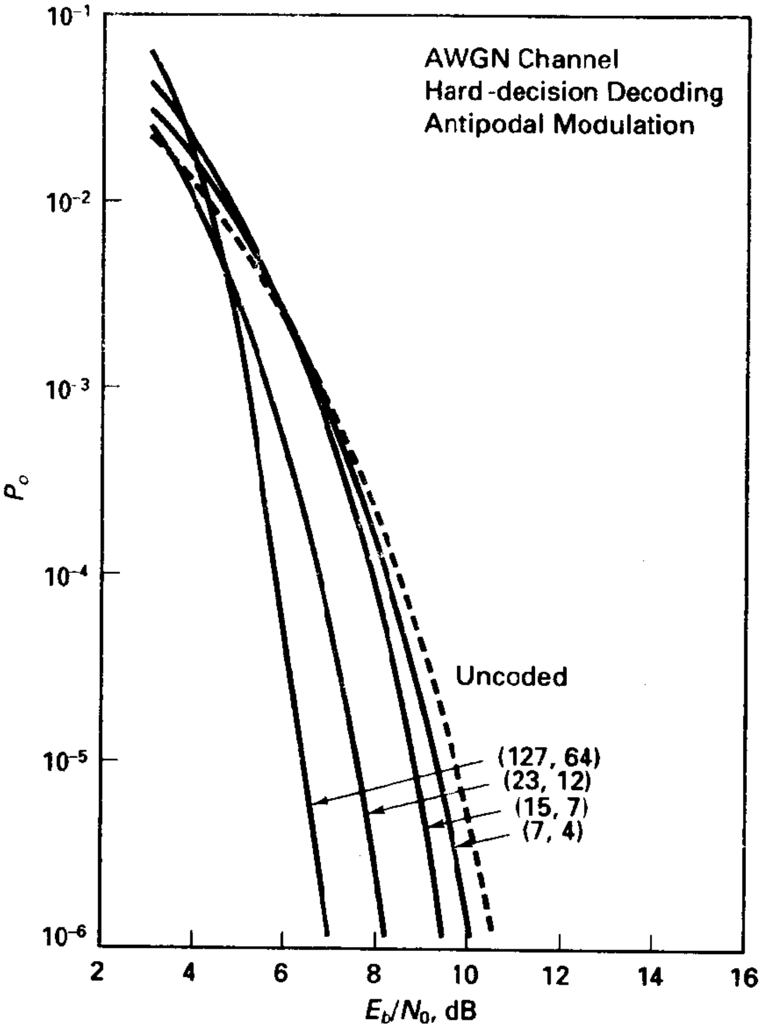


Figure 5.10.2 Probability of output error for block codes.

10-error-correcting BCH code. Again, binary antipodal signaling and binary demodulator quantization are assumed. We observe the benefits of increased blocklength, at least in small error probability regions of the plot.

Communication engineers refer to the savings in E_b/N_0 over uncoded signaling as ***coding gain***. We shall soon discuss the quantity known as asymptotic coding gain, which may be rather different from the "real" coding gain. Coding gain must always be specified at some operating error probability, typically 10^{-5}, and is expressed in decibels. The (23, 12) code, for example, from Figure 5.10.2, attains a coding gain of 2.0 dB at $P_b = 10^{-5}$. Such coding gain translates into an effective improvement in the communication link—allowing either a reduction of transmitter power or antenna gain or an increase in receiver noise level or an increase in the bit rate for a given power.

The crossover of coded and uncoded performance curves in Figure 5.10.2 is perhaps surprising, but rather common with coding,[30] indicating we should not always expect coding to provide a panacea for improving error performance. The reason is simply that as E_b/N_0 decreases, so does the code symbol E_s/N_0 value, and P_s rapidly increases. This latter degradation is more than enough to offset the error-correcting capability of the code at some point, and the coded system can perform worse than an uncoded one. Generally, the more powerful the code, the lower the SNR at which crossover occurs. Incidentally, this effect is not some artifact of the symbol error probability approximations just made and is not related to bounded-distance decoding assumptions. The same effect is observed in comparing *message* error probabilities for coded and uncoded systems, as, for example, in the case of the Golay code and 12-bit messages (Figure 5.10.1). We will also encounter this effect when soft-decision decoding is analyzed.

Another observation about coded system performance is that as the block length n increases the performance curves become steeper, essentially leading to an all-or-nothing behavior. This can be attributed to the law-of-large numbers: A block code is capable of correcting a certain number of errors, t, out of n symbols. As n becomes large, with high probability that the number of errors is either less than t (when channel quality is such that $P_s < t/n$), or the actual number of errors exceeds t when the opposite is true. The sharpness of this transition from good to bad thus becomes more precipitous for the longer codes. Information-theoretic arguments are similarly very sharp; decoding either succeeds with very high probability if the rate is below a critical value or fails in the opposite case.

The effect of varying code rate is demonstrated for binary cod'ng and antipodal signaling in Figure 5.10.3. Here we show the performance of different BCH codes with block length $n = 127$, and we show performance for codes with rates of nearly $\frac{1}{4}$, $\frac{1}{2}$, and $\frac{3}{4}$. Although R_0 analysis of Chapter 4 pointed to a slight preference for low rate codes in this case, the performance with $n = 127$ BCH codes is relatively insensitive to R.

Low-rate coding is not always to be preferred (also forecast in Chapter 4). we apply the same analysis to the case of binary FSK (orthogonal) modulation with *noncoherent* detection and binary quantization, we obtain the results of Figure 5.10.4.

[30] This assumes the information rate is held fixed in the comparison.

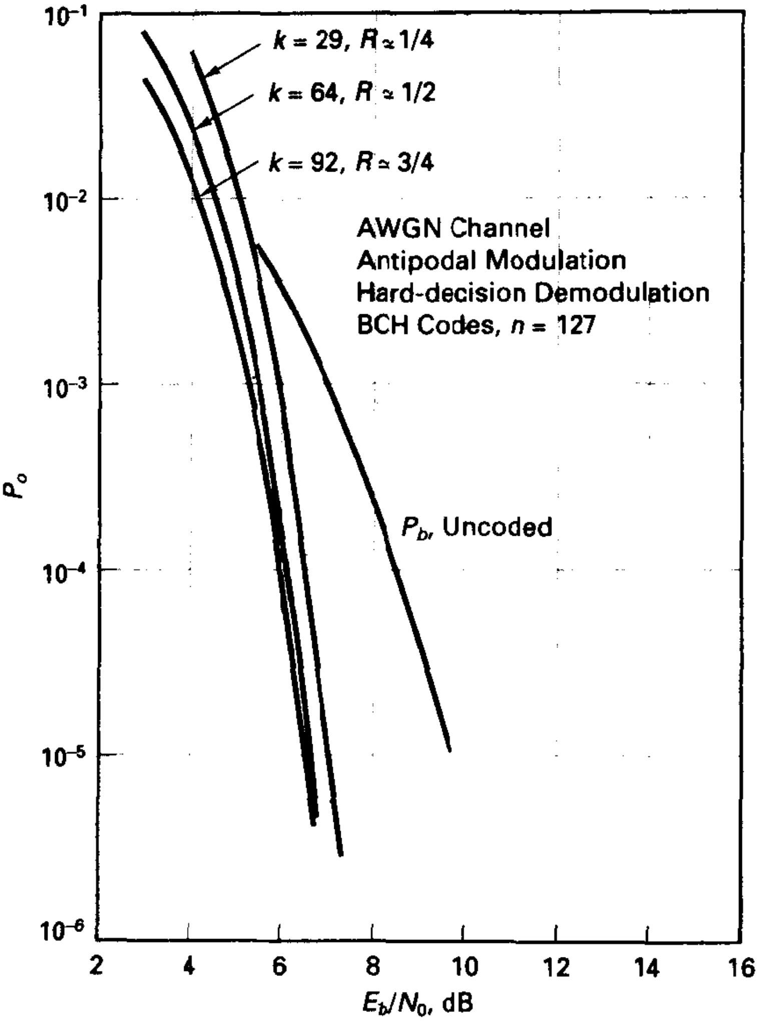


Figure 5.10.3 Probability of output error for block codes, antipodal modulation, coherent detection.

Notice that all curves are shifted to the right relative to those for antipodal signaling; this is to be expected based on the relative energy efficiencies in obtaining a given P_s of the coherent and orthogonal signal sets, in addition to the degradation due to noncoherent detection. Moreover, low-rate coding is inferior here; the qualitative explanation is that as code rate drops, the available E_b/N_0 is spread more thinly among code symbols, so that E_s/N_0 is relatively small. In the small SNR regime, noncoherent detection degrades relatively quickly, and even though low-rate codes achieve larger Hamming distance, the increased channel error probability P_s more than offsets this gain.

Example 5.36 Analysis of RS (15, 9) Code over GF(16) with Orthogonal Signaling

The (15, 9) RS code has $d_{\min} = 7$, so that with hard decisions, the decoder is capable of repairing three or fewer errors. For a modulation/demodulation technique, we adopt 16-ary orthogonal modulation and noncoherent detection. The symbol error probability for this situation is presented in Section 3.4, and in particular P_s for uncoded transmission as

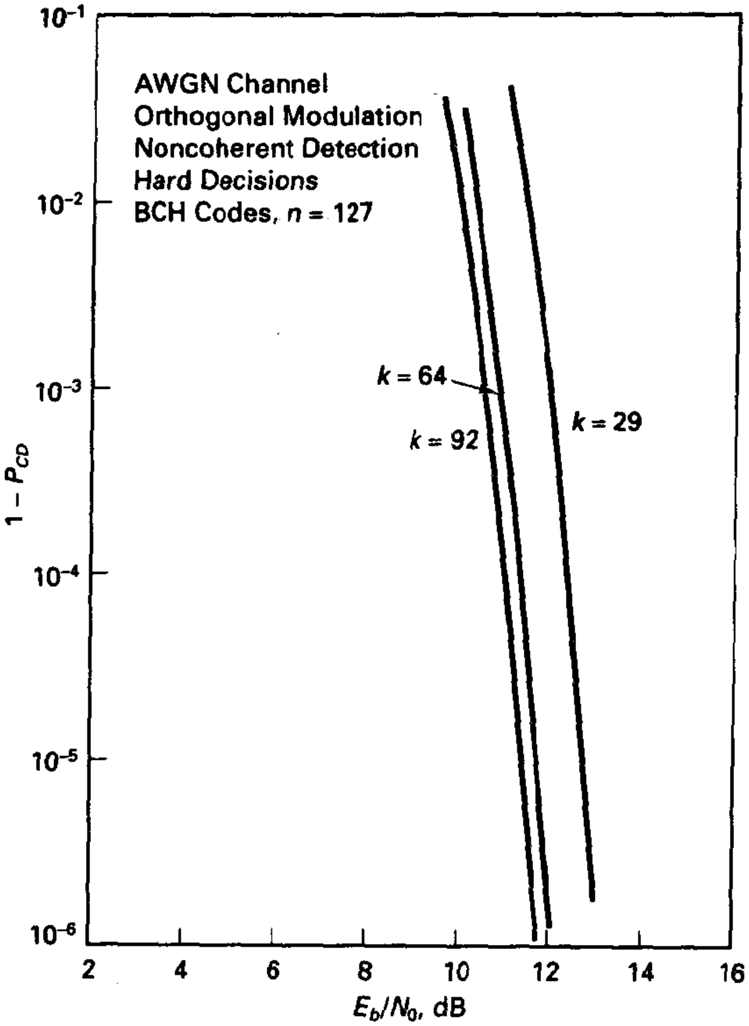


Figure 5.10.4 Probability of block error, BCH codes, orthogonal modulation, noncoherent detection.

a function of E_b/N_0 is given in Figure 3.4.5. To properly incorporate the influence of coding, assuming we keep the information throughput fixed with and without coding, we must realize that the effective E_s/N_0 per code symbol is

$$\frac{E_s}{N_0} = \frac{9}{15}(\log_2 16)\left(\frac{E_b}{N_0}\right) \tag{5.10.7}$$

The probability of not correctly decoding is given by

$$1 - P_{\text{CD}} = \sum_{i=4}^{15} C_i^{15} P_s^i (1 - P_s)^{15-i} \tag{5.10.8}$$

For small values of P_s, (5.10.8) can be well approximated by the first term in the sum, yielding

$$1 - P_{\text{CDcoding}} \approx 1365 P_s^4. \tag{5.10.9a}$$

With no coding, but the same modulation/detection strategy, the probability of not delivering a correct 9-symbol message is

$$1 - P_{\mathrm{CD}_{\text{no coding}}} \approx 9P_s, \tag{5.10.9b}$$

where P_s is related to E_b/N_0 as in Chapter 3. Note the values of P_s differ in these two expressions due to the power sharing as in (5.10.3).

Probability of incorrect decoding

Unless the code is perfect, and the attempted error correction radius is the maximum, t, $1 - P_{\mathrm{CD}} \neq P_{\mathrm{ICD}}$. It is often important to ascertain the actual probability of incorrect decoding, from which P_{DE} can be determined if necessary. Let us assume that the decoder performs bounded-distance decoding out to a radius $t_1 \leq t$, i.e., a decoding is only produced if the received vector $\mathbf{r}$ is within t_1 Hamming units of some codeword. If the decoder attempts no error correction, but only seeks to detect errors, we set $t_1 = 0$. The incorrect decoding event is just that event where the error pattern moves the received vector inside another decoding sphere of radius t_1 about some incorrect codeword.

In the case of $t_1 = 0$, knowledge of the weight spectrum $\{A_w, w = d_{\min}, \ldots, n\}$, of the code provides an exact assessment of P_{ICD}, using

$$P_{\mathrm{UE}} = P_{\mathrm{ICD}} = \sum_{i=d_{min}}^{n} A_i P_s^i (1 - P_s)^{n-i}. \tag{5.10.10}$$

For large codes, it is usually convenient to employ the MacWilliams relation (Section 5.2) and the weight spectrum of the smaller dual code to compute the required weight spectrum.

In the general case of error correction and detection, with reference to Figure 5.10.5, it is easy to develop a bound on P_{ICD} by realizing that incorrect decoding cannot occur unless $d_{\min} - t_1$ or more channel errors are present. Consequently,

$$P_{\mathrm{ICD}} \leq \sum_{i=d_{\min}-t_1}^{n} C_i^n P_s^i (1 - P_s)^{n-i}. \tag{5.10.11}$$

This is typically rather pessimistic. If necessary, exact results on P_{ICD} are available using methodology developed in the text of Michelson and Levesque [6]. By lengthy

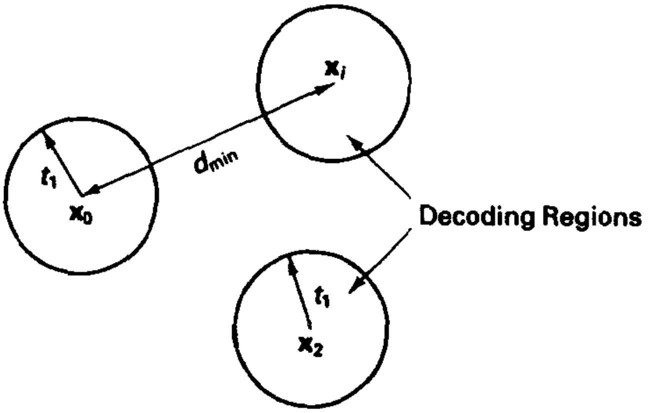


Figure 5.10.5 Decoding regions for correction of up to t_1 errors. Interstitial space constitutes detected errors.

combinatoric arguments it is shown there, for $q > 2$, that

$$P_{\mathrm{ICD}} = \sum_{i=d_{\min}}^{n} A_i P(i) \sum_{s=0}^{t_1} \sum_{k=i-s}^{i+s} \sum_{m=r_1}^{r_2} C_{i-s+m}^{i} C_{k-i+s-2m}^{s-m} C_{m}^{n-i} \cdots (q-2)^{k-i+s-2m}(q-1)^m. \tag{5.10.12}$$

In this expression, the summation limits r_1 and r_2 are

$$r_1 = \max(0, k-i) \tag{5.10.13a}$$

and

$$r_2 = \left\lfloor \frac{k-i+s}{2} \right\rfloor \tag{5.10.13b}$$

Also,

$$P(i) = \left(\frac{P_s}{q-1}\right)^i \left(1 - \frac{P_s}{q-1}\right)^{n-i}$$

is the probability of a *specific* error pattern of weight i.

In the binary code case, the expression simplifies to

$$P_{\mathrm{ICD}} = \sum_{i=d_{\min}}^{n} A_i P(i) \sum_{s=0}^{t_1} \sum_{k=i-s}^{i+s} C_{(i+k-s)/2}^{h} C_{(k-i+s)/2}^{n-i} \tag{5.10.13c}$$

Example 5.36 Continued

P_{ICD} as a function of E_b/N_0 is calculated for $t_1 = 3$ using (5.10.12) and the known weight spectrum of the (15, 9) RS code and is presented in Figure 5.10.6. 16-ary orthogonal modulation with noncoherent detection is assumed. Comparison of this result with that of Figure 5.10.4 reveals that the large majority of cases where decoding is not correct are attributable to decoding failures, and not to incorrect decodings. This is attributable in large part to the relatively small volume occupied by the 16^9 15-dimensional decoding regions of radius 3; specifically, these regions occupy less than 10% of the total volume.

We can also see the benefit of being less ambitious in error correction. If instead we set the correction radius to $t_1 = 2$, P_{ICD} drops sharply, but at the expense of a smaller probability of correct decoding.

5.10.2 Soft-decision (ML) Decoding, AWGN Channel

True ML decoding of block codes for the AWGN channel remains relatively uncommon and currently is only feasible for "small" codes. Wolf [47] has provided a general trellis decoding procedure for organizing ML decoding for general (n, k) linear codes, for which the trellis length is n levels and the maximum breadth is q^k or q^{n-k}, whichever is smaller. Recently, van Tilborg et al. [66] have shown branch-and-bound procedures for decoding Hamming codes in ML fashion. The case of binary $(n, n-1)$ single-parity check codes is a case wherein ML decoding is relatively straightforward and is known as Wagner decoding [67]. The decoder first checks the parity of the hard-decision vector produced by the demodulator, and if the parity check equation is satisfied, we have the ML codeword. If the parity check fails, then the decoder locates the single position of the

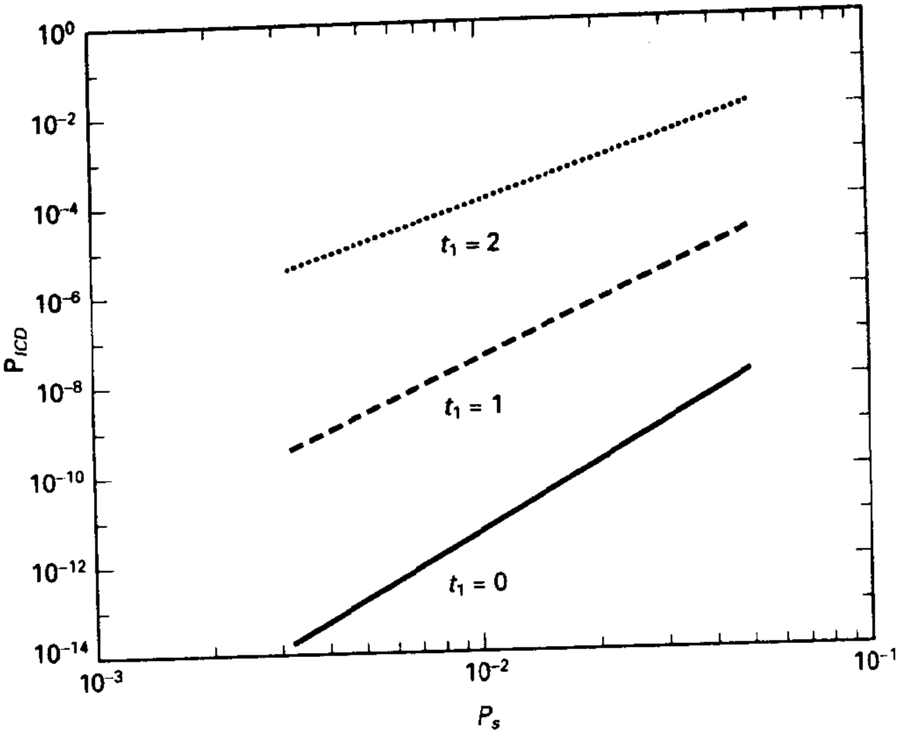


Figure 5.10.6 Probability of incorrect decoding for RS (15, 11) code versus symbol error probability.

codeword with smallest symbol likelihood and "flips" the binary decision to the second-best choice. This restores correct parity and gives the ML choice over all codewords.

The ML decoding problem is a member of a class of computationally tough problems, known as NP-complete problems, for which no algorithm for solving the problem having solution time a polynomial function of the problem size has yet been found. Complexity theorists hold such a polynomial-time algorithm is unlikely, given the concerted effort directed at some of these problems, for example, the traveling salesman problem, the knapsack problem, etc. Thus, efficient, true ML algorithms for large block codes seem prohibitively difficult. (This is not only a matter for soft-decision decoding; even the problem of ML decoding on a hard-decision channel is "difficult" in the same sense.)

The algorithms of Chase [49] and the trellis decoding algorithm of Wolf [47] provide, in principle, ways to approximate ML decoding. Certainly near-ML decoders will become more common as device advances continue and as algorithm research progresses. In any case, the potential coding performance of ML decoding is of interest as a measure of goodness for suboptimal approaches.

Let us once again assume the all-0's codeword is transmitted via q-ary modulation and demodulation, but that the decoder is presented with full information sufficient to compute codeword likelihoods $\Lambda(\mathbf{r}, \mathbf{x}_i)$. Of course, the nature of this codeword metric depends on the channel, the modulation set used, and the form of demodulation.

In this decoding regime, we are attempting complete decoding, by definition, and we let P_E denote the probability of decoding error. This is the probability that *some* nonzero codeword has greater metric than the all-0's codeword, but, in general, this is too difficult to evaluate exactly. However, a simple union bound generally provides sufficient accuracy. Specifically, we upper-bound the probability of decoding error by the sum of two-codeword error probabilities:

$$P_E \leq \sum_{\mathbf{x}_i \neq \mathbf{x}_0} P(\Lambda(\mathbf{r}, \mathbf{x}_i) > \Lambda(\mathbf{r}, \mathbf{x}_0)) = \sum_{\mathbf{x}_i \neq \mathbf{x}_0} P(\mathbf{x}_0 \rightarrow \mathbf{x}_i). \tag{5.10.14}$$

This bound only requires the ability to compute the probability of confusing two codewords with the given modulator/channel/demodulator setup and the spectrum of codeword distances; yet it is known to be asymptotically correct for increasing SNR. We shall illustrate this bounding procedure with the Golay (23, 12) code.

Example 5.37 Performance for ML Decoding of the (23, 12) Code with Antipodal Signaling

The probability of decoding a weight-i codeword instead of the all-0's word, assuming antipodal transmission on the AWGN channel, is given by

$$P(\text{weight } i \text{ error}) = Q((i2E_s/N_0)^{1/2}),$$

since for every unit of Hamming distance, the squared Euclidean distance between coded signals increases by $4E_s$. The union bound on P_E then becomes

$$P_E \leq \sum_{i=d_{\min}}^{n} A_i Q((i2E_s/N_0)^{1/2}). \tag{5.10.15}$$

The weight spectrum of the code (Figure 5.4.3) tells that 253 vectors are at distance 7, 506 are at distance 8, and so on. Figure 5.10.1 presents the union upper-bound for ML decoding on the Gaussian channel versus E_b/N_0; again recall that the symbol energy-to-noise density ratio is given by

$$\frac{E_s}{N_0} = \left(\frac{12}{23}\right)\frac{E_b}{N_0}. \tag{5.10.16}$$

Also shown on this plot are the performance of hard-decision decoding given earlier, and it may be seen that soft-decision decoding buys about 2 dB in energy efficiency over the range of the plot. Chase [49] shows that Algorithm II performs within 0.2 dB of the ML detector in this case.

As the signal-to-noise ratio increases, it may be seen that (5.10.15) is increasingly dominated by the first term of the sum, or by the minimum Hamming distance decoding possibilities. (This will frequently carry over to other modulation/demodulation options and to other codes.) Thus we often say that the asymptotic performance of a code is given by

$$P_E \approx A_{d_{\min}} Q\left(\left(\frac{d_{\min}2E_s}{N_0}\right)^{1/2}\right) = A_{d_{\min}} Q\left(\left(\frac{2E_b}{N_0}(d_{\min}R)\right)^{1/2}\right). \tag{5.10.17a}$$

Comparing this performance with that of uncoded antipodal signaling of a k-bit message, for which

$$P_{E_{\text{no coding}}} \approx kQ\left(\left(\frac{2E_b}{N_0}\right)^{1/2}\right) \tag{5.10.17b}$$

at high SNR, we see that the relative energy efficiency governed by the Q-function argument is $d_{\min}R$. This quantity, often converted to decibels, is frequently dubbed the ***asymptotic coding gain*** (ACG), for it represents the relative energy efficiency at *large* SNR, where multiplier coefficients are relatively insignificant and where the first term of the union bound becomes an accurate estimator of P_E. More specifically, if we replace the Q-functions in (5.10.17a) by the exponential bound $(1/2)e^{-x^2/2}$, we find that

$$\lim_{E_b/N_0 \to \infty} \log_e P_E = -ACG(E_b/N_0), \tag{5.10.18a}$$

whereas the corresponding expression for uncoded antipodal transmission is

$$\lim_{E_b/N_0 \to \infty} \log_e P_{E_{\text{no coding}}} = -(E_b/N_0). \tag{5.10.18b}$$

Thus, the ACG parameter does asymptotically predict performance correctly. Graphically, the interpretation is that on a logarithmic presentation, at sufficiently small P_E, the performance curve is shifted from that of antipodal signaling by $10\log_{10} ACG$. The convenient aspect of ACG is that only the minimum Hamming distance of the code and the code rate are required for its calculation.

Example 5.38 Asymptotic Coding Gain for Several $R = 1/2$ Binary Codes

Consider the following binary codes: the (8, 4) extended Hamming code, with $d_{\min} = 4$; a (16, 8) self-dual code with $d_{min} = 6$; the extended Golay (24, 12) code with $d_{min} = 8$; and the (48, 24) extended quadratic residue code, with $d_{\min} = 12$, [5]. Assuming maximum likelihood decoding and antipodal transmission, the ACG's of these codes are, respectively, 3, 4.8, 6, and 7.6 decibels.

This progression may suggest that arbitrarily large coding gains are possible by further increase in block length, but, of course, information-theoretic limits disallow these to be "real" gains. For example, when coding using two signal-space dimensions per bit on the AWGN channel, channel capacity calculations of Chapter 2 show that the theoretical minimum SNR for "arbitrarily-reliable" communication is $E_b/N_0 = 1$, or 0 dB. We must interpret ACG as the increase in energy efficiency, relative to that of uncoded transmission, for vanishingly small error probability. Of course, the performance of uncoded transmission is increasingly inefficient relative to the channel capacity limit as we move to smaller error probabilities. This allows large ACGs to be consistent with the information theory dictums of Chapter 2. We should also observe that for typical error probability levels, say 10^{-5}, the ACG usually is slightly optimistic in its assessment of true coding gain; this is due to the fact that the ACG formulation overlooks the multiplier attached to the many-nearest-neighbor situation, and this may not become truly insignificant until extremely small error probabilities are studied.

Bhargava [68] has plotted the performance of an ML decoder for these codes, using the union bound above. Each of the codes is a so-called extremal self-dual code, for which the weight spectrum is known. As a point of interest, the coding gain for the (48, 24) code, at a decoded bit error probability of $P_b = 10^{-5}$, is only 4.8 dB, quite short of the 7.6 dB above. At $P_b = 10^{-8}$ the calculated gain is 5.8 dB. Reference [69] includes related material on soft-decoding these codes.

On the AWGN energy efficiency of soft versus hard decoding

We have observed in the case of the Golay code approximately a 2-dB gain in energy efficiency simply by employing soft (unquantized) decoding rather than algebraic decoding on binary decisions. This result corroborates the earlier R_0 and channel capacity theory attached to this channel setting. Another perspective is provided by comparing the asymptotic coding gain parameters for the case of soft and hard decoding.

For hard-decision decoding, we can approximate (5.10.2) for small P_s by

$$P_{E_{\text{hard}}} \approx C^n_{t+1} P_s^{t+1}. \tag{5.10.19}$$

Recalling that for antipodal signaling $P_s = Q[(2E_s/N_0)^{1/2}] = Q[(R2E_b/N_0)^{1/2}]$, and using $Q(x) \le (1/2)e^{-x^2/2}$, we arrive at

$$P_{E_{\text{hard}}} \approx \frac{C^n_{t+1}}{2^{t+1}} e^{-(E_b/N_0)[R(t+1)]} \tag{5.10.20}$$

The approximate performance for soft-decision decoding is obtained by use of the same Q-function approximation in (5.10.17a), yielding

$$P_{E_{ML}} \approx \left(\frac{A_{d_{\min}}}{2}\right) e^{-(E_b/N_0)(d_{\min}R)} \tag{5.10.21}$$

Comparison of this expression with that of (5.10.20) reveals an efficiency ratio of

$$ACG = \frac{(d_{\min}R)}{(t+1)R} \to 2 \tag{5.10.22}$$

as $d_{\min}$ becomes large, independent of code rate R. Thus, one can argue that for high SNR, and for reasonably large-distance binary codes, on the antipodal AWGN channel, hard-decision decoding costs roughly 3 dB in energy efficiency. Caution is again in order: Typical experience tends to give a slightly smaller penalty, due to the importance of the nonexponential terms, namely, the error multiplier coefficient and the fact that $d_{\min} < 2(t+1)$.

5.10.3 Hard-decision Decoding, Rayleigh Channel

In Chapter 3 we saw that error probability for uncoded transmission on the Rayleigh channel exhibits a weak inverse dependence on E_b/N_0, regardless of modulation format. However, we demonstrated in Chapter 4 that the channel capacity of the interleaved Rayleigh channel is only marginally less than that of the nonfading channel. Block codes are indeed able in many cases to improve the situation; however, careless application of coding techniques may produce poor results. We first consider what not to do!

Consider a slowly fading channel and use of an (n, k) code with block length such that the fading process may be viewed as fixed over a codeword. In this case, just as in Chapter 3, we may determine the probability of not correctly decoding, P_E, by first evaluating the error probability conditioned upon a specific fading strength and then averaging this with respect to the fading random variable. In effect, we may think of moving the operating point up and down one of the P_E curves of the previous section, weighting the results by the probability of a given level of SNR. In the case of a code whose performance curve is steep, we essentially have that the error probability is 1 if the SNR is below a certain threshold and that $P_E \approx 0$ if SNR is above this threshold. Thus, P_E is the probability that the SNR for the given block is below a critical number. For the Rayleigh p.d.f., this probability depends inversely on mean E_b/N_0, and consequently we have not significantly improved the situation at all—a 10-dB increase in SNR is required to effect a drop in average P_E by a factor of 10, although we may have lowered the absolute error probability some.

Interleaving, as discussed in Section 5.9, is a principal remedy for this behavior. Rather than allow one bad fading event to cause decoding error, interleaving scrambles the transmission of codewords and then reshuffles following demodulation so that channel actions (fading variables in particular) are roughly independent and multiple *independent* fading events are required to cause decoding error. This is a less probable event. Of course, interleaver delay and/or memory limitations may preclude attainment of independent samples; the possibilities depend on symbol rate, fading rate, block length, and delay constraints.

With perfect interleaving assumed, performance analysis is relatively simple. In the hard-decision case, we first determine P_s, the symbol error probability on the Rayleigh channel, as performed in Chapter 3. Remember that the effective energy per code symbol is reduced by an amount proportional to code rate R. The resulting symbol error probability then becomes an input to the P_E calculation performed in Section 5.10.1, since the channel is now memoryless.

Example 5.39 (23, 12) Code on Interleaved Rayleigh Channel

Suppose for illustration we elect binary coding with the Golay code and transmit symbols with binary DPSK so that differentially coherent demodulation can be used. (Again, there is reluctance toward coherent demodulation on fading channels due to problems of maintaining absolute carrier phase synchronization.) We assume no other side information is given the decoder. With E_b/N_0 denoting the average SNR available per uncoded information bit, we realize that $E_s/N_0 = (12/23)E_b/N_0$. Furthermore, the symbol error probability is

$$P_s = \frac{1}{2 + 2E_s/N_0} \qquad (5.10.23)$$

as derived in Section 3.6. The Golay decoder will not decode correctly if four or more errors occur in 23 positions. Thus,

$$P_E = \sum_{i=4}^{23} C_i^{23} P_s^i (1 - P_s)^{23-i}. \qquad (5.10.24a)$$

For E_b/N_0 reasonably large, where we are interested in the performance, (5.10.24a) is dominated by the first term, and we have

$$P_E \approx C_4^{23} \left[\left(\frac{23}{12} \right) \frac{1}{2E_b/N_0} \right]^4 = 7469(E_b/N_0)^{-4}. \qquad (5.10.24b)$$

This asymptotic expression is shown in Figure 5.10.7, along with the result for uncoded DPSK transmission. Notice very different results here from those obtained for the AWGN environment. Namely, the effective energy savings is very large (20–40 dB) at typical performance levels of interest, and, furthermore, the gain grows as we seek more reliable operation. Thus, the concept of asymptotic coding gain is meaningless in this case. In graphical terms, the slope of the P_E curve on a log–log plot has been increased to -4, instead of -1, by virtue of coding and interleaving. In general, the value of the slope for hard-decision decoding is $-(t + 1)$, where t is the guaranteed error-correcting power.

A related topic is that of ***diversity transmission***, a classic method of improving performance on fading channels. In nth-order diversity transmission, we send n replicas of a message symbol through separate channels (time, frequency, or space diversity

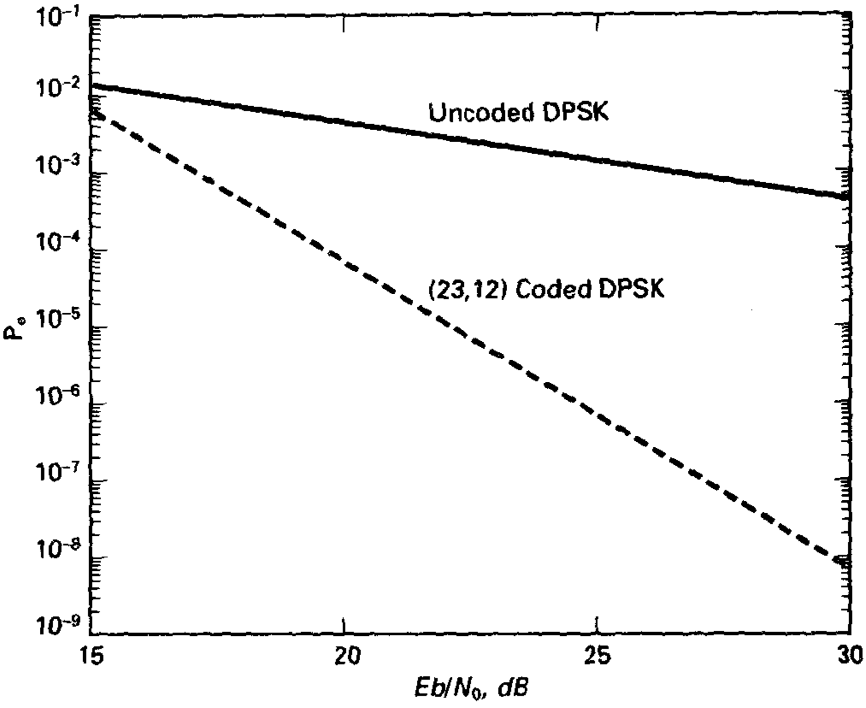


Figure 5.10.7 Error probability for (23, 12) code on Rayleigh channel.

are common), trusting that the various channels exhibit independent behavior. In time or frequency diversity, an energy sharing among transmissions is implied, again meaning $E_s/N_0 = (1/n)E_b/N_0$. We should see that diversity transmission is little more than repetition coding, with rate $R = 1/n$ message symbols per code symbol. To put this in the present hard-decision decoding context, let n be odd and use majority voting in the decoder as an error-correction policy. The decoder will err if $(n + 1)/2$ or more errors occur. Thus,

$$P_{E_{\text{diversity}}} \approx C^{n}_{(n+1)/2} P_s^{(n-1)/2}. \tag{5.10.25}$$

Substituting for P_s a relation such as (5.10.23) shows that a performance curve having slope $-(n + 1)/2$, roughly half the diversity order, is attainable (this again is equivalent to $\lfloor(d_{\min} - 1)/2\rfloor$ for this repetition code). However, the penalty for this improvement is a drop in system throughput to $1/n$ times the uncoded value. Error-control coding is now understood as the way to more efficiently achieve *implicit diversity* protection against fading and thus avoid this drastic drop in throughput.[31]

5.10.4 Soft-decision Decoding, Rayleigh Channel

Though soft-decision decoding of block codes is more difficult, it offers even greater potential for coded systems on fading channels. In soft-decision decoding the analog demodulator outputs for each symbol are supplied to the decoder for further processing;

[31] On the topic of diversity, we should note that there are "soft" analogs of the voting scheme just analyzed, going under the name of diversity combiners.

in addition, "side information" in the form of channel gain may be supplied, although this is somewhat more problematic.

It is crucial that interleaving be used, at least with the standard "random" error correcting codes, to effectively make the channel memoryless. Otherwise, on slow fading channels, we suffer the effect mentioned in Section 5.9—correlated fading across an entire code block overwhelms the ability of the decoder to locate the correct codeword with high probability. With interleaving, the deinterleaver must store the analog information attached to symbols, and possibly side information if available.

To illustrate the analysis of this application, let us again consider the use of an (n, k) code, with binary PSK modulation and coherent detection. (Coherent detection is admittedly somewhat questionable on fading channels.) After interleaving to adequate depth, related to the correlation time of the channel gain process, binary symbols are sent and received with energy $a_n^2 E_s$, where a_n is the channel gain attached to the nth symbol in the channel time index sequence. By assumption, these are Rayleigh random variables.

Assuming that the demodulator is provided perfect knowledge about a_n, the maximum likelihood decoding rule is

$$\underset{\mathbf{x}_i}{\text{maximize}} \prod_{j=0}^{n-1} f(r_j | x_{ij}, a_j) \tag{5.10.26}$$

Interleaving has the effect of giving a memoryless structure to the likelihood function, and conditioned upon a given code symbol and fading amplitude, the required p.d.f. is just a Gaussian form with mean $a_j x_{ij} E_s^{1/2}$ and variance $N_0/2$. Maximizing the logarithm instead shows that the rule becomes

$$\underset{\mathbf{x}_i}{\text{minimize}} \sum_{j=0}^{n-1} (r_j - a_j x_{ij} E_s^{1/2})^2 \tag{5.10.27}$$

This can be interpreted geometrically as "find the closest codeword to $\mathbf{r}$ *after* correction by the proper signal strength in every coordinate."

If side information is not available, but interleaving is still utilized, the proper symbol metric is

$$\log f(r_j | x_{ij}) = \log \left[\int f(r_j | x_{ij}) f(a)\, da \right], \tag{5.10.28}$$

which can be evaluated "by parts," leading to a slightly different metric from above.

Efficient soft-decision decoding could be accomplished by a Chase algorithm, for example, avoiding the need for exhaustive evaluation of the likelihood of all 4096 codewords. This, however, would have to be interpreted as "near ML decoding."

Performance analysis for ML decoding on this channel follows a union bounding procedure, for which we need the two-codeword probability of error, averaged over the fading distribution. Consider two codewords $\mathbf{x}_0$ and $\mathbf{x}_i$ that differ in w positions within the block. With perfect side information and antipodal signaling, the two codeword error probability, conditioned upon a certain fading sequence $\mathbf{a}$, is

$$P[\mathbf{x}_0 \to \mathbf{x}_i | \mathbf{a}] = Q[(d_E / 2N_0)^{1/2}] \tag{5.10.29}$$

where $d_E^2 = (a_1^2 + \cdots + a_w^2)(4E_s)$ is the Euclidean distance between code sequences, modified by the channel gain in a given position. Note that only w positions contribute to the total Euclidean distance. Substituting this distance expression into (5.10.29) and using an exponential bound on the Q-function gives

$$P\,[\mathbf{x}_0 \rightarrow \mathbf{x}_i|\mathbf{a}] \leq \frac{1}{2}\prod_{i=1}^{w} e^{-a_i^2 2E_s/N_0} \tag{5.10.30}$$

To remove conditioning on the fading amplitude, we assume the fading variables are independent Rayleigh variates. Averaging of (5.10.30) then leaves the upper bound

$$P[\mathbf{x}_0 \rightarrow \mathbf{x}_i] \leq \frac{1}{2}\frac{1}{\left(1 + \frac{2E_s}{N_0}\right)^w} \tag{5.10.31}$$

showing that the probability of confusing two sequences having distance w is inversely proportional to the wth power of SNR, also meaning that effectively we have achieved wth-order diversity when sequences differ in this many positions.

The final upper bound on codeword error probability then uses the weight spectrum of the code in a union bound:

$$P_E \leq \frac{1}{2}\sum_{w=d_{\min}}^{n} \frac{A_w}{\left(1 + \frac{2E_s}{N_0}\right)^w} \approx \frac{1}{2}\frac{A_{d_{\min}}}{\left(R\frac{2E_b}{N_0}\right)^{d_{\min}}} \tag{5.10.32}$$

assuming dominance at high SNR by the minimum distance events.

In summary, the analysis points to a high SNR behavior for soft-decision decoding that diminishes as $(E_b/N_0)^{d_{\min}}$, and we thereby say the effective diversity order of the block coding strategy is equivalent to the minimum distance of the code. Recall for hard-decision decoding the effective diversity order was roughly half as large, $t+1$. Although we developed the result for the binary codes, this idea carries over to nonbinary codes on the Rayleigh channel, provided full interleaving is attained. Soft-decision decoding for other cases is in general much more difficult however.

5.11 POWER SPECTRUM OF CONVENTIONAL BLOCK CODED MODULATION

Our focus thus far in this chapter has been on the error-control aspects of block codes. One penalty attached to potential improvement to communication efficiency is hardware complexity. Another is spectrum occupancy, although as we see in the next section, coded transmission does not necessarily increase bandwidth relative to an uncoded transmission. In the traditional case of linear (n, k) block codes, the bandwidth is normally expanded, however, by virtue of the greater number of channel symbols sent per unit time, assuming fixed information rate.

Block coding techniques have actually been used to shape the power spectrum in some applications by introducing statistical dependencies into the code stream. This goes under the name of *line coding* where spectral shaping to accommodate nonideal channel response is important. A primary example is magnetic and optical recording, where both short and long runs of consecutive symbols are forbidden, the former to enhance readability of the signal and the latter to preserve symbol synchronization. Most of these coding approaches are nonlinear block codes, and in fact certain rules are put into play across codeword boundaries to ensure that the concatenation of codewords meets the desired constraints. Compact disc recordings use such a procedure, called eight-to-fourteen (EFM) modulation [32]. This application, while an important one in certain applications, is not the one we have in mind here.

We consider then the case of a linear code over GF(q), combined with q-ary modulation. As a baseline for comparison, we consider as an alternative an uncoded scheme that maps k q-ary symbols directly into modulator signals over a time interval of kT_s seconds. We assume that the message symbols are equiprobable and statistically independent, and the resulting power spectrum can then be computed using the techniques of Chapter 3. For memoryless modulation with independent symbols, we found that the power spectrum was essentially a weighted sum of magnitude-squared Fourier transforms of the various possible signals, and in the special case of linear modulation, wherein the various symbols are complex scalar multiples of a common pulse shape, for example, M-PSK or QAM, we found that the power spectrum possessed the shape of the magnitude squared of the pulse shape's Fourier transform.

An appealing way to model the power spectrum for coded modulation is the following: treat the block encoder output sequence as another equiprobable, independent, q-ary sequence, with transmission rate increased by a factor n/k relative to the message symbol rate. Under this approximation, the coded signal power spectrum has exactly the same shape as the uncoded signal would, except the frequency axis is scaled by the coding rate R. In other words, this view would hold that a $R = \frac{1}{2}$ coded binary PSK signal (with rectangular pulse shape) would have a $\sin^2(\pi f T_b/2)/(\pi f T_b/2)^2$ spectral shape, with first nulls $2R_b$ removed from the center frequency, rather than R_b hertz. As another case, use of a $(31, 27)$ RS code, in conjunction with 32-ary FSK, would produce a spectrum identical to that of uncoded FSK, except 31/27 wider.

Clearly, there is some merit in the thinking. A listing of all codewords in a q-ary linear code will find all code symbols used equally often; thus, the equiprobable approximation is valid assuming equiprobable selection of codewords and a time randomization. The independence assumption is more problematic, since the encoder obviously places certain dependencies on the symbols of a codeword—not all sequences are possible at the encoder output. It turns out, however, that the power spectrum is dependent only on the discrete-time autocorrelation function of the coded symbol stream, and for typical codes, this autocorrelation is "white," that is, successive symbols are uncorrelated when mapped to a symmetric signal constellation. Thus, although not strictly independent, it is usually the case that the symbol statistics yield a power spectrum consistent with the preceding approximation. We shall refer to this as ***spectral equivalence***.

More specifically, Wilson and Lakshman [70] have shown that if a linear code has a generator matrix G whose n columns are pairwise linearly independent (or if all columns are distinct to within a scalar factor), then when mapped to a symmetric signal set, the coded signal's power spectrum is exactly that of uncoded modulation, except frequency-stretched by the code expansion factor $1/R$. This property of generator matrices seems routinely satisfied, a corollary of good error control properties, and in fact the cases where it fails to occur are low-rate codes and repetition codes.

Example 5.39 Power Spectrum of Two (15, 5) Codes

Suppose we employ a (15, 5) triple-error correcting BCH in conjunction with PSK modulation. The generator matrix (in nonsystematic form) is, following (5.4.26),

$$\mathbf{G} = \begin{bmatrix} 1 & 1 & 1 & 0 & 1 & 1 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 & 1 & 1 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 0 & 1 & 1 & 0 & 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 0 & 1 & 1 & 0 & 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 1 & 1 & 0 & 0 & 1 & 0 & 1 \end{bmatrix} \tag{5.11.1}$$

Notice that no two columns of the generator matrix sum to the zero vector; hence the sufficient condition for spectral equivalence is satisfied. The power spectrum would be exactly the same as that of uncoded PSK, except three times wider.

Consider, on the other hand, the use of a (15, 5) repetition code produced by

$$\mathbf{G} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}. \tag{5.11.2}$$

Here there exist obvious repetitions of columns, and spectral equivalence does not follow. It happens that, in rough terms, the bandwidth expansion by three does occur; however, the fine details of the power spectrum are different with this code.

As a corollary remark, it has also been shown [70] that, when spectral equivalence does not hold, rearrangement of the columns of G does alter the spectrum. (The error control properties remain unaffected, however.)

5.12 BLOCK CODING FOR BAND-LIMITED CHANNELS

Thus far we have in essence been treating block coding from a somewhat classical perspective; that is, the block encoder appends $n - k$ additional symbols from the code alphabet to the information vector. The modulator/demodulator forms a discrete-time channel, perhaps with soft-quantized outputs, and the decoder attempts to infer the information symbols from the n demodulator outputs. In such applications, the bandwidth of the signal produced by the modulator is expanded by the ratio $1/R$, relative to a system using the same modulator without coding.

In the modern era, bandwidth has become a steadily more precious resource to the communication engineer. For this reason, multilevel modulation schemes such as M-PSK and M-QAM were developed. As seen in Chapter 3, these invariably trade spectral bandwidth (dimensionality per bit) for energy efficiency. We might ask whether coding could be combined with such modulation methods to preserve good spectral efficiency, avoiding the traditional bandwidth expansion associated with coding and at the same time increasing the energy efficiency. We know from principles of information theory that such bandwidth-efficient coding schemes do exist, and indeed the potential gains to be had over uncoded transmission are just as great in the regime of several bps per hertz spectral efficiency as they are in the more traditional regime where the spectral efficiency is lower.

The first major step in achieving this promise was made in the realm of trellis coding and will be discussed in Chapter 6. Subsequently, similar ideas permeated block coding techniques, and we shall present them here. The essential change of perspective is that we try to find sequences of signal space coefficients that are distant in Euclidean distance terms, *rather than try to find codes with good Hamming distance properties, and then* map these onto a bandwidth-efficient modulation technique. Typical of the nonpreferred design is that shown in Figure 5.12.1, where we begin with, say, 8-PSK modulation, a relatively bandwidth-efficient transmission scheme. To improve the energy efficiency, we might precode the message with a (7, 5) RS code over GF(8), which is capable of single-error correction of hard-decision demodulator outputs. The dimensionality per information bit of this scheme is 14 dimensions/$(5 \cdot 3) = \frac{14}{15}$ dimensions/bit (really 7 complex dimensions/15 bits). Uncoded 8-PSK would have a dimensionality of 1 complex dimension per 3 bits, so we have sacrificed bandwidth by a ratio of $\frac{7}{5}$.

The energy per code symbol is $E_s = 15E_b/7$, and use of the theory developed in Chapter 3 would allow us to find the symbol error probability on the AWGN channel for the coded technique. The decoder will correct all 0 or 1-error patterns, and we have

$$1 - P_{\text{CD}} \approx C_2^7 P_s^2 (1 - P_s)^5, \tag{5.12.1}$$

which will show some asymptotic coding gain over uncoded 8-PSK.

We could do better if soft-decision ML decoding of the RS code was performed, but this is rather difficult except for simple codes. A better approach is to design coded modulation schemes to maximize the smallest signal space distance. Furthermore, by increasing the modulator alphabet size relative to what is needed for uncoded transmission, we can avoid the bandwidth penalty of coding. We will take up this topic in Chapter 6 under trellis codes, although the concept extends easily to block codes as well. It is more straightforward, however, to implement ML decoding within the trellis coding framework. Interested readers are invited to consult [71] on analysis of this particular coding scheme.

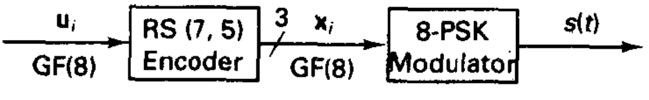


Figure 5.12.1 Simple means of combining nonbinary coding with bandwidth-efficient modulation.

5.12.1 Multilevel Coding

A hybrid approach that recognizes the importance of signal space distance in coding for the AWGN channel, yet that retains simple decoding, is known as ***multilevel coding*** [72, 73]. Essentially, the idea is that we wish to select codewords as sequences of points from a bandwidth-efficient constellation, say 8-PSK. To maximize the vector Euclidean distance over codewords, we can have a few positions where codewords differ by symbols having large intersignal distance, or we can have a relatively larger number of positions where the symbol distance is small, say the minimum distance between points in the constellation.

To build multilevel codes, we envision the encoder as putting constraints on the various bit lines of an M-ary modulator. For example in 8-PSK, three modulator input bits label a constellation point and four in the case of 16-QAM. If we adopt a labeling of points for 8-PSK corresponding to natural binary labeling around the circle, then the least significant bit labels which of two QPSK sets the signal lies within. Within these sets of four, the second bit labels one of two sets of antipodal pairs. Finally, the most significant bit labels which of the members of the selected set is actually transmitted. Viewed this way, it is clear that the LSB needs relatively large error protection, due to its small Euclidean distance to a neighbor. Resolving just the second bit is more reliable, and the least likely bit to be in error is the MSB.

This has led to the concept of block coding each bit line with binary block codes having common length n, but varying k parameter. A simple example, due to Sayegh [74], is shown in Figure 5.12.2, wherein we use an $(n, k_0) = (7, 1)$ code on the LSB line, an $(n, k_1) = (7, 6)$ code on the second bit, and an $(n, k_2) = (7, 7)$ (no coding) code on the MSB line.

If d_{E_i} represents the minimum Euclidean distance between cosets at level i in a partitioning of the original constellation, and d_{H_i} represents the Hamming distance for the binary code at level i, it can be shown that the minimum squared Euclidean distance between valid sequences at the output of the modulator is

$$d^2_{\min_E} = \min(d_{H_0} d^2_{E_0}, d_{H_1} d^2_{E_1}, \ldots, d_{H_{m-1}} d^2_{E_{m-1}}),$$

where m is the number of bits at the modulator input. In the case of the example just presented, the three codes have minimum Hamming distances of 7, 2, and 1, respectively. The corresponding squared Euclidean distances between points are $0.585E_s$, $2E_s$, and $4E_s$. (These are, respectively, the squared distances in the original constellation, in QPSK sets, and in PSK sets.) Thus, the minimum squared Euclidean distance is

$$d^2_{m_E} = \min[7(0.585E_s), 2(2E_s), 1(4E_s)] = 4E_s = 8E_b, \tag{5.12.2}$$

since 14 information bits produce 7 modulator symbols.

In a maximum likelihood decoder of these codes, the asymptotic coding gain is given by the minimum squared Euclidean distance, properly normalized in energy. Here we use

$$\begin{aligned} P_E &\approx NQ\left[\left(\frac{d_{\min_E}}{2N_0}\right)^{1/2}\right] \\ &= NQ\left[\left(\frac{4E_b}{N_0}\right)^{1/2}\right] \end{aligned} \tag{5.12.3}$$

after substitution of the distance stated. This points to a 3-dB gain over antipodal signaling (or uncoded QPSK as well), yet the technique has the same spectral efficiency as uncoded QPSK!

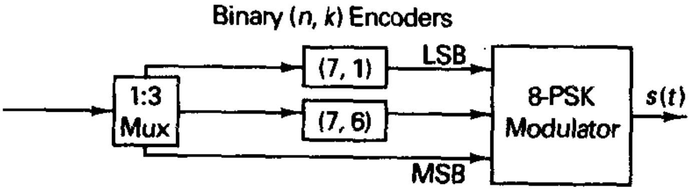


Figure 5.12.2 Multilevel coded 8-PSK (after Sayegh [74]).

Multilevel codes are actually decoded hierarchically, and the decoding is not strictly ML, although generally the performance is asymptotically as good. (This equivalence may not set in until extremely small error probabilities in some cases.) First, the demodulator releases a sequence of n observations, appropriate for the scheme in effect. If the modulation is two-dimensional, then a two-dimensional basis function receiver produces a pair of Gaussian random variables at each code position. A decoder for the LSB bit then performs an ML or near-ML decision on the *sequence* of LSB bits presented to the modulator, using the soft-decision outputs of the demodulator. This decision has chosen k_0 information bits. At the same time, the decision has selected n cosets wherein the remaining message bits are constrained to lie. Given the choice of cosets, a second decoder, matched to the second-level code, performs another ML decision using the original demodulator output sequence, but with side information on the coset membership of the transmitted sequence. Once this decision is made, we have decided another k_1 information bits. Finally, the corresponding choice has zeroed in on a coset sequence for the remaining bits, and ML decoding if finally performed on this MSB code.

In the case of Sayegh's code, the first decoder is for a (7, 1) code, and ML decisions are easy because there are just two codewords for which metrics must be computed. The next-level decoder is for a (7, 6) single parity bit code, and Wagner decoding can easily decode this code in ML fashion by first making hard symbol decisions on whether $\mathbf{r}_n$ lies in coset 1 or coset 0 at the first partition level. The parity of these coset bit decisions is checked, and, if odd, we change the decision of the worst (largest) distance decision among these hard decisions. Finally, the (7, 7) code requires no decoding obviously; we simply perform hard decisions within the selected coset sequence. Each of these is a simple antipodal decision process.

5.12.2 Simple LSB Coding and Hard-decision Decoding

A very simple technique capable of modest gains on the AWGN channel is one that employs coding only on the LSB line of a modulator set and *hard-decision* decoding of this line at the decoder [75]. The coding power of the LSB code is made large enough to give negligible probability of decoding error for the coded portion of the system negligible, relative to the probability of decision error for remaining uncoded bits, which label constellation points that are more distant. Algebraic decoding is less efficient, as we have emphasized, but capable of very high speeds, and this is the strength of this technique. In [75] it is shown that for 8- or 16-PSK designs, triple-error correction is sufficient for the LSB code and that the asymptotic coding gain can approach

$$\text{ACG} \to 10\log_{10}(d_1^2/d_0^2), \tag{5.12.4}$$

where d_0 and d_1 are the original constellation distance and the minimum distance of the first subset partition, respectively. This ACG is approached as the code rate tends toward 1, while still maintaining three-error correction. Suggested codes are (23, 12), (63, 45), and (127, 106).

Similar coding can be applied to the one-stage partitions of any coset-decomposable constellation. For example, sets built from the lattice $\mathbf{Z}^2$ decompose easily into two sets with squared-distance twice that of the original set. Here it turns out that, asymptotically, single-error correction is sufficient to balance the coded segment with the uncoded segment. This simplifies decoding. However, the potential coding gain is smaller, only 3 dB.

5.12.3 Multilevel Codes for Fading Channels

On an interleaved Rayleigh fading channel, the code designs that maximize minimum Euclidean distance are no longer necessarily optimal. What is more crucial is that the code be such that multiple bad fading events are required to cause a decoding error. This diversity effect is far more important than is the maximization of minimum distance. In the example cited, we have codewords that differ in only one position of the codeword (there the distance between these points is large, $4E_s$), and it takes but one bad fading event to cause a decision error. The net result is that performance versus E_b/N_0 still behaves as $c/(E_b/N_0)$; that is, the slope is -1 on a log–log plot, as in Section 3.6. Coding has bought a slight shift of the curve, 3 dB in this case, but the overall impact of coding is very discouraging.

If instead we increase the minimum Hamming distance in symbols between codewords, say to 2, then we can show that second-order diversity is obtained, presuming optimal metric decoding is employed. This might be accomplished here by using less redundancy on the first-level code and more on a (7, 6) code on the MSB line. Such a design gives minimum Hamming distance 2 in symbols.

More extensive discussion on this topic is postponed to Chapter 6, where similar ideas surface in the design of trellis codes for fading channels. Interested readers are referred to a recent survey article by Seshadri et al. [76] for further study.

APPENDIX 5A1: DATA SCRAMBLERS

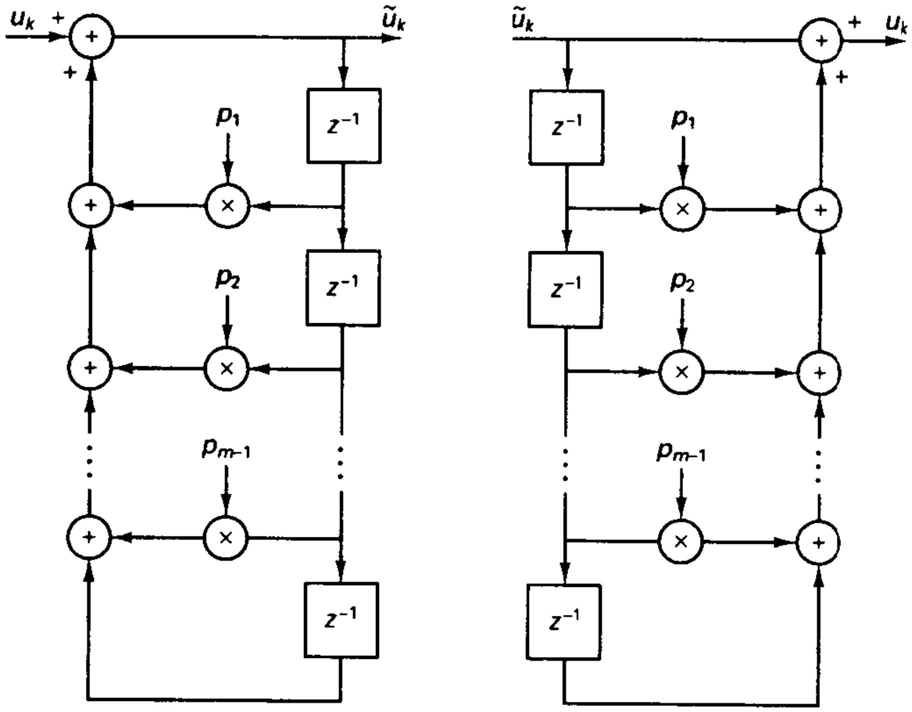
A practical consideration in data communications is ensuring that the transmitted signal exhibit reasonable statistical behavior and in particular avoid long strings on a certain symbol or certain short period sequences. Such may occur during temporary pauses in a communication session, especially in digital coding of speech, or merely due to predominance of some symbols in a message sequence and may lead to two harmful effects:

1. The power spectrum may exhibit undesirable concentration of power at certain discrete frequencies. For example, in binary NRZ transmission, long strings of either 0 or 1 would lead to a spectral line at zero frequency (or the carrier frequency in a carrier modulated system); periodic patterns in the data will also produce spectral lines.
2. Synchronization circuits in the receiver, which depend on symbol transitions to work effectively, may show poor tracking performance or break lock altogether. Again, NRZ provides an example: all clock synchronization circuits require relatively frequent level transitions to locate the timing epoch.

A common remedy for this problem is to add ***data scrambling*** prior to coding and modulation, along with corresponding descrambling following demodulation/decoding. (Scrambling as discussed here is not for purposes of message security.) Most commonly, this is performed on a binary bit stream version of the message and can be done in a ***self-synchronizing*** manner; that is, the descrambler automatically produces (after a delay of only a few bits) the proper output sequence without need to search for a proper sequence phase. Furthermore, channel errors will be seen not to cause catastrophic loss. This type of message scrambling is not intended as a message security technique, for unscrambling the message is far too easy and common to all users!

Self-synchronizing scramblers employ a primitive polynomial $p(D)$ to divide the desired message polynomial $u(D)$, producing the output sequence $\tilde{u}(D)$ over the same field. Although the division process is performed in a manner seen for encoding of cyclic block codes, the scrambling process can proceed for an indefinitely long time. Figure 5A1.1a shows a generic scrambler built from an m-stage shift register with feedback, where m is the degree of $p(D)$. By writing a difference equation for $\tilde{u}_k$ and then representing sequences in polynomial notation, it is readily seen that

$$\tilde{u}(D) = \frac{u(D)}{p(D)}. \tag{5A1.1}$$



(a) Scrambler (b) Descrambler

Figure 5A1.1 Scrambling circuits using $p(D) = p_0 + p_1 D + \cdots + p_m D^m$.

To see why this device has the desired effect, suppose that the register holds at least a single 1 and that the input is frozen at $u_k = 0$ thereafter. The register will sequence through a series of states and repeat after $2^m - 1$ clock cycles. This is the maximum length possible and follows from adoption of a primitive polynomial. Similar results occur if the input is held at 1. Not only is the period of repetition very long, but the relative frequency of 0's and 1's is nearly balanced, as are the occurrences of runs of various types. Of course, if we wish to be pathological, we can find an input for any given starting state of the register that will hold the output at a constant level indefinitely; however, this is obviously a rare event.

Whereas scrambling divides the input sequence by $p(D)$, the descrambler multiplies by $p(D)$, recovering the original $u(D)$ sequence. The generic descrambler is shown in Figure 5A1.1b. A difference equation will again reveal that

$$v(D) = p(D)\tilde{u}(D) = p(D)\frac{u(D)}{p(D)} = u(D), \tag{5A1.2}$$

as desired.

Notice that the system is self-synchronizing from end to end and will produce correct output from a random starting state after m cycles. Alternatively, the scrambler and descrambler can be set to a prescribed initial condition to synchronize immediately. In any case, once synchronized, there is zero delay between the input and output of a given bit.

Since descramblers are feed-forward, finite-length filters, unlimited error propagation is avoided in the face of channel errors. Assuming that the channel error rate is small, it is readily seen that the output error rate is magnified by a factor equivalent to the number of taps in the feed-forward descrambler or, equivalently, the number of nonzero terms in $s(D)$. It is important to note the order of division followed by multiplication. On an error-free channel the order is arbitrary, but reversing the order on a channel with error introduces potential for unlimited error propagation.

Typical applications use scramblers with degree 10 or more. For example, in the X.25 CCITT telecommunications standard for packet communication, the polynomial $p(D) = D^{17}+D^2+1$ is specified. For the data modem standard V.22bis, the polynomial $p(D) = D^{17} + D^{14} + 1$ is employed. Thus, the magnification is 3, considered to be a tolerable price for the benefits obtained. In packetized communication, three errors are no worse than one error anyway.

BIBLIOGRAPHY

1. Peterson, W. W., and Weldon, E. J., Jr., *Error Correcting Codes*, 2nd ed, Cambridge, MA: MIT Press, 1972.
2. Berlekamp, E., *Algebraic Coding Theory*, New York: McGraw-Hill, 1968.
3. Lin, J., and Costello, D. J., *Error Control Coding: Fundamentals and Applications*, Englewood Cliffs, NJ: Prentice Hall, 1983.
4. Blahut, R., *Theory and Practice of Error Control Codes*, Reading, MA: Addison-Wesley, 1983.

5. MacWilliams, F. J., and Sloane, N. J. A., *The Theory of Error Correcting Codes*, Amsterdam: North-Holland, 1977.

6. Michelson, A. M., and Levesque, A. H., *Error-control Techniques for Digital Communications*, New York: Wiley, 1985.

7. Gallager, R., *Information Theory and Reliable Communication*, New York: Wiley, 1968.

8. MacWilliams, F. J., "A Theorem on the Distribution of Weights in a Systematic Code," *Bell System Tech. J.*, vol. 42, pp. 79–94, 1963.

9. Muller, D. E., "Application of Boolean Algebra to Switching Circuit Design and to Error Detection," *IRE Trans. Electronic Computing*, vol. 3, pp. 6–12, 1954; see also Reed, I. S., "A Class of Multiple-error-correcting Codes and the Decoding Scheme," *IRE Trans. Information Theory*, vol. 4, pp. 38–49, 1954.

10. Helgert, H. J., and Stinaff, R. D., "Minimum Distance Bounds for Binary Linear Codes," *IEEE Trans. Information Theory*, pp. 344–356, May 1973.

11. Verhoeff, T., "An Updated Table of Minimum-distance Bounds for Binary Linear Codes," *IEEE Trans. Information Theory*, vol. IT-33, no. 5, pp. 665–680, September 1987.

12. Nordstrom, A. W., and Robinson, J. P., "An Optimum Nonlinear Code," *Information and Control*, vol. 11, pp. 613–616, 1967.

13. Golay, M. J. E., "Binary Coding," *IRE Trans. Information Theory*, vol. 4, pp. 23–28, 1954.

14. Tietavainen, A., "A Short Proof for the Nonexistence of Unknown Perfect Codes over GF(q), $q > 2$," *Annales Acad. Scient. Fennicae*, series A, no. 580, pp. 1–6, 1974.

15. Singleton, R. C., "Maximum Distance q-Nary Codes," *IEEE Trans. Information Theory*, vol. 10, pp. 116–118, 1964.

16. Plotkin, M., "Binary Codes with Specified Minimum Distance," *IRE Trans. Information Theory*, vol. 6, pp. 445–450, 1960.

17. Elias, P., "Coding for Noisy Channels," *IRE Conv. Record*, pp. 37–46, 1955.

18. McEliece, R. J., Rodemich, E. R., Rumsey, H. C., Jr., and Welch, L. R., "New Upper Bounds on the Rate of a Code via the Delsarte–MacWilliams Inequalities," *IEEE Trans. Information Theory*, vol. 23, pp. 157–166, 1977.

19. Gilbert, E. N., The Gilbert bound apparently appeared in unpublished notes in 1953. See ref. [1] for a proof.

20. Varshamov, R. R., "Estimate of the Number of Signals in Error Correcting Codes," *Dokl. Akad. Nauk SSSR*, vol. 117, pp. 739–741, 1957 (in Russian).

21. Chen, C. L., "Computer Results on the Minimum Distance of Some Binary Cyclic Codes," *IEEE Trans. Information Theory*, pp. 359–360, May 1970.

22. Promhouse, G., and Tavares, S. E., "The Minimum Distance of All Binary Cyclic Codes of Odd Length from 69 to 99," *IEEE Trans. Information Theory*, pp. 438–442, 1978.

23. Bose, R. C., and Ray-Chaudhuri, D. K., "On a Class of Error Correcting Binary Group Codes," *Information and Control*, vol. 3, pp. 68–79, 1960.

24. Hocquenghem, A., "Codes correcteurs d'erreurs," Chiffres, vol. 2, pp. 147–156, 1959 (in French).

25. Kasmai, T., and Lin, S., "Some Results on the Minimum Weight of BCH Codes," *IEEE Trans. Information Theory*, vol. 18, pp. 824–825, 1972.

26. Van Lint, J. H., and Wilson, R. M., "On the Minimum Distance of Cyclic Codes," *IEEE Trans. Information Theory*, vol. IT-32, pp. 23–40, 1986.

27. Kasami, T., "A Decoding Procedure for Multiple-error Correcting Cyclic Codes," *IEEE Trans. Information Theory*, vol. 10, pp. 134–138, 1964.

28. Lin, S., and Weldon, E. J., Jr., "Long BCH Codes Are Bad," *Information and Control*, vol. 11, pp. 445–451, 1967.

29. Berlekamp, E. R., "Long Primitive Binary BCH Codes Have Distance $d \sim 2n \ln R^{-1} / \log n$," *IEEE Trans. Information Theory*, vol. 8, pp. 415–426, 1972.

30. Golub, G. H., and Van Loan, C. F., *Matrix Computations*, Baltimore, MD: Johns Hopkins Press, 1983.

31. Reed, I. S., and Solomon, G., "Polynomial Codes over Certain Finite Fields," J. SIAM, vol. 8, pp. 300–304, 1960.

32. Peek, J. B. H., "Communications Aspects of the Compact Disc Digital Audio System," *IEEE Communication Society Magazine*, vol. 23, pp. 7–15, 1985.

33. Advanced Micro Devices, data sheet on AM95c94, Advanced Burst Processor (similar products are manufactured by Western Digital, NCR, and Ampex).

34. Wolf, J. K., "Adding Two Information Symbols to Certain Nonbinary BCH Codes and Some Applications," *Bell Systems Tech. J.*, vol. 48, pp. 2405–2424, 1969.

35. "The VLSI Implementation of a Reed–Solomon Encoder," *IEEE Trans. Computers*, vol. 33, pp. 906–911, 1984.

36. Blahut, R. E., "A Universal Reed–Solomon Decoder," *IBM J. Research Development*, vol. 28, pp. 150–158, 1984.

37. Chen, C. L., "Byte-oriented Error-correcting Codes for Semiconductor Memory Systems," *IEEE Trans. Computers*, vol. C-35, pp. 646–648, July 1986.

38. Forney, G. D., Jr., *Concatenated Codes*, Cambridge, MA: MIT Press, 1966.

39. Meggitt, J. E., "Error Correcting Codes and Their Implementation," *IRE Trans. Information Theory*, vol. 7, pp. 232–244, 1961.

40. Peterson, W. W., "Encoding and Decoding Procedures for the Bose-Chaudhuri Codes," *IRE Trans. Information Theory*, vol. 6, pp. 459–470, 1960.

41. Gorenstein, D. C., and Zierler, N. "A Class of Error-correcting Codes in p^m Symbols," *J. Soc. Indus. Applied Math.*, vol. 9, pp. 207–214, 1961.

42. Chien, R. T., "Cyclic Decoding Procedure for the BCH Codes," *IEEE Trans. Information Theory*, vol. 10, pp. 357–363, 1964.

43. Berlekamp, E. R., "On Decoding Binary BCH Codes," *IEEE Trans. Information Theory*, vol. 11, pp. 577–580, 1965.

44. Forney, G. D., Jr., "On Decoding BCH Codes," *IEEE Trans. Information Theory*, vol. 11, pp. 549–551, 1965.

45. Massey, J. L., "Shift Register Synthesis and BCH Decoding," *IEEE Trans. Information Theory*, vol. 15, pp. 122–127, 1969.

46. Blahut, R. E., "Transform Techniques for Error Control Codes," *IBM J. Research Development*, vol. 23, pp. 299–315, 1979.

47. Wolf, J. K., "Efficient Maximum likelihood Decoding of Linear Block Codes Using a Trellis," *IEEE Trans. Information Theory*, vol. 24, pp. 76–80, 1978.

48. Forney, G. D., Jr., "Coset Codes—Binary Lattices and Related Codes," *IEEE Trans. Information Theory*, vol. IT-34, pt II, pp. 1152–1187, September 1988.

49. Chase, D., "A Class of Algorithms for Decoding Block Codes with Channel Measurement Information," *IEEE Trans. Information Theory*, vol. 18, pp. 170–182, 1972.

50. Forney, G. D., Jr., "Generalized Minimum Distance Decoding," *IEEE Trans. Information Theory*, vol. 12, pp. 125–131, 1966.

51. Korznik, V. I., "Bound on Undetected Error Probability and Optimum Group Codes in a Channel with Feedback," *Telecomm. Radio Engineering*, vol. 2, pp. 87–92, 1965.

52. Leung-Yan-Cheong, S. K., Barnes, E. R., and Friedman, D. U., "On Some Properties of the Undetected Error Probability for Linear Codes," *IEEE Trans. Information Theory*, vol. 25, pp. 110–112, 1979.

53. Fujiwara, T., Kasami, T., Kitai, A., and Lin, S., "On the Undetected Error Probability for Shortened Hamming Codes'," *IEEE Trans. Communications*, vol. 33, pp. 570–574, 1985.

54. Witzke, K. A., and Leung, C., "A Comparison of Some Error Detecting CRC Code Standards," *IEEE Trans. Communications*, vol. 33, pp. 996–998, 1985.

55. Goodman, R. M., McEliece, R. J., and Sayano, M., "Phased Burst Error-correcting Array Codes," *IEEE Trans. Information Theory*, vol. 39, pp. 684–693, March 1993.

56. Justesen, J., "A Class of Constructive Asymptotically Good Algebraic Codes," *IEEE Trans. Information Theory*, vol. 18, pp. 652–656, 1972.

57. Kanal, L. N., and Sastry, A. R. K., "Models for Channels with Memory and Their Applications to Error Control," *Proc. IEEE*, vol. 66, pp. 724–743, July 1978.

58. Drukarev, A. T., and Yiu, K. P., "Performance of Error Correcting Codes on Channels with Memory," *IEEE Trans. Communications*, vol. COM-34, pp. 513–521, June 1986.

59. Berlekamp, E. R., Peile, R. E., and Pope, S. P., "The Applications of Error Control to Communications," *IEEE Communications Magazine*, vol. 25, pp. 44–57, 1987.

60. Gilbert, E. N., "Capacity of a Burst-noise Channel," *Bell Sys. Tech. J.*, pp. 1253–1265, September 1960.

61. Forney, G. D., Jr., "Burst-correcting Codes for the Classic Bursty Channel," *IEEE Trans. Communications*, vol. COM-19, pp. 772–781, October 1971.

62. Patel, A. M., "Error Recovery Scheme for the IBM 3850 Mass Storage System," *IBM J. of Res. and Dev.*, vol. 24, pp. 32–42, January 1980.

63. Ramsey, J. L., "Realization of Optimum Interleavers," *IEEE Trans. Information Theory*, vol. IT-16, pp. 338–345, May 1970.

64. Wolfowitz, J., "Memory Increases Capacity," *Information and Control*, vol. 11, pp. 423–428, 1967.

65. Wu, W. W., Haccoun, D., Piele, R., and Hirata, Y., "Coding for Satellite Communication," *IEEE Journal on Selected Areas in Communication*, vol. SAC-5, pp. 724–748, May 1987.

66. Van Tilborg, H., et al., "Maximum Likelihood Decoding of Hamming Codes," in submission.

67. Silverman, R. A., and Balser, M., "Coding for Constant Date Rate Systems," *IRE Trans. Information Theory*, vol. 4, pp. 50–63, September 1954.

68. Bhargava, V. J., "Soft Decoding Performance of Extremal Self-dual Codes," *Proc. IEEE*, vol. 71, pp. 183–184, 1983.

69. Baumert, L. J., and McEliece, R. J., "Soft Decision Decoding of Block Codes," *Proc. ITC Conference*, Los Angeles, pp. 879–882, 1978.

70. Wilson, S. G., and Lakshman, M., "Power Spectrum of Linear Block Code Modulation," submitted to *IEEE Trans. Communications*, 1994.

71. Einarsson, G., and Sundberg, C-E., "A Note on Soft Decision Decoding with Successive Erasures," *IEEE Trans. Information Theory*, vol. IT-22, pp. 88–95, January 1976.

72. Imai, H., and Hirakawa, S., "A New Multilievel Coding Method Using Error Correcting Codes,"·*IEEE Trans. Information Theory*, vol. IT-23, pp. 371–377, May 1977.

73. Pottie, G. J., and Taylor, D. P., "Multilevel Codes Based on Partitioning," *IEEE Trans. Information Theory*, vol. 35, pp. 87–98, January 1989.

74. Sayegh, S. L., "A Class of Optimum Block Codes in Signal Space," *IEEE Trans. Communications*, vol. COM-34, pp. 1043–1045, October 1986.

75. Wilson, S. G., Livingston, J. N., and McCanless, J. C., "LSB-coded Modulation with Hard-decision Decoding," *Proc. of Conf. on Info. Science and Systems*, Princeton, NJ, March 1994, submitted to *IEEE Trans. Communications*.

76. Seshadri, N., Sundberg, C-. E. W., and Weeraclody, V., "Advanced Techniques for Modulation, Error Correction, Channel Equalization and Diversity," *AT&T Technical Journal*, vol. 72, pp. 48–63, July–August 1993.

EXERCISES

5.0.1. The Hamming code of Section 5.0.1 was shown to be single error correcting on a binary symmetric channel. Argue the following by means of the Venn diagram for the code:

(a) If instead of trying to correct errors we merely report "bad data" when one or more of the circle checks fails, then single- and double-error patterns are detected perfectly, while some three-error patterns escape detection.

(b) Suppose that the channel simply removes tokens occasionally, which we could think of as an erasure, but that remaining tokens are known to be correct. Argue that two or fewer such erasures can be filled perfectly, but that some three-erasure patterns produce erroneous decoding in an ML decoder.

5.0.2. Suppose that we add another parity constraint to the Venn diagram describing the $(7, 4)$ code. Draw a circle completely enclosing the previous diagram and require that the fourth parity symbol residing in this new region be chosen such that the number of red tokens in the eight positions be an even number. We now have an $(8, 4)$ code.

(a) Write the four parity check equations.

(b) Argue that this code is capable of correcting any single error among the eight code bits while still detecting any double-error pattern. What is the indication of a double-error pattern?

5.1.1. Verify that the set of rational numbers of the form p/q, together with ordinary addition and multiplication of fractions, constitutes a field.

5.1.2. Construct GF(7) and form addition and multiplication tables. Determine the order of all nonzero field elements. What is the characteristic of this field?

5.1.3. Solve the following linear system of equations:

(a) Over GF(3): $x + y = 0$; $2x - 2y = 1$

(b) Over GF(4): $\alpha x + \alpha^2 y = 1$; $x - \alpha^2 y = 0$

Note: All standard algebraic procedures, including Cramer's rule and Gaussian elimination, are valid for finite fields.

5.1.4. **(a)** Show that $f(D) = D^4 + D^3 + D^2 + D^1 + 1$ over GF(2) is irreducible by testing factors of degree 2 or less. Show, however, that $f(D)$ is not a primitive polynomial by examining consecutive powers of α, defined as a solution of the polynomial $f(D) = 0$.

(b) The polynomial $f(D) = D^2 + D + 2$ over GF(3) is primitive. Use it to provide a construction of GF(9). What are the orders of the nonzero field elements?

5.1.5. Verify that the axioms listed in (5.1.1) are satisfied for the polynomial construction of GF(p^m) as described in Section 5.1. That is, show that addition and multiplication of polynomials of degree m over GF(p), modulo an irreducible polynomial of degree m, yields consistent arithmetic.

5.1.6. Find a subfield of size 4 in GF(16) introduced in Example 5.2. What subfields could we find in GF(256)?

5.1.7. A combinational logic circuit is to be implemented for performing multiplication in GF(8), as in Example 5.1. Such a circuit would have three input lines for each operand and three binary output lines, yielding the coefficients of the polynomial representation of the product. Use standard minimization techniques on the required truth table to design such a circuit. [As a technological note, programmable array logic technology now makes such implementations inexpensive, even if needed to be replicated for several GF(q) multipliers. Addition could be done even more simply.]

5.1.8. Design a circuit that multiplies an arbitrary input element in GF(16) by the fixed element α^3 and then adds this to a previous sum, also an element in GF(16). This multiply-and-accumulate operation is important in computing the syndrome symbols for decoding of cyclic codes.

5.1.9. Verify (5.1.12), on which hinges the invertibility of the DFT. (*Hint*: Use the fact that the result for the sum of a geometric series $\sum_{j=0}^{n-1} \alpha^j = (\alpha^n - 1)/(\alpha - 1)$ ~~.

5.1.10. Compute the DFT of the length-7 binary sequence 1000100 using as a primitive seventh root of unity $\alpha \in \text{GF}(8)$ satisfying $\alpha = 2$ in Figure 5.1.1. Verify that the inverse transform produces the original sequence.

5.1.11. Prove the convolution theorem for sequences over finite fields, (5.1.22); that is, the DFT of the cyclic convolution of two sequences is obtained as the product of their respective DFTs.

5.2.1. Suppose that we alter the (7, 4) code with generator matrix given in (5.2.1) to obtain a (7, 3) code. Specifically, remove from the original code all codewords that have an odd weight. Argue that the new parity check matrix is obtained from the former by augmenting it with an additional row of seven 1's.

(a) Find the parity check matrix and a new generator matrix.

(b) Find the weight spectrum of the (7, 3) code and give an expression for the probability of decoding error on a BSC, assuming decoding with the use of a syndrome table and complete decoding.

(c) If we operate the decoder in error-detection only mode, that is, we do not output a decision whenever the syndrome is nonzero, determine the probability of undetected error.

5.2.2. Prove that in a linear code C either all codewords are of even weight or exactly half are odd weight and half are even weight.

5.2.3. In the now-antiquated two-out-of-five code employed in early teleprinters for transmitting decimal data, the digits 0 through 9 were represented by five-bit binary patterns with exactly two 1's.

(a) Show that there are 10 codewords.

(b) What is the rate of this code?

(c) Show that the code is nonlinear.

(d) Determine $d_{\min}$ and the distance spectrum. Despite the fact the code is nonlinear, the distance spectrum is invariant to choice of reference code vector.

(e) We decode only if the received pattern has two 1's; find the probability of incorrect decoding and the probability of correct decoding.

5.2.4. An (11, 4) binary code[32] has a generator matrix given by

$$\mathbf{G} = \begin{bmatrix} 1 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 1 & 1 \\ 0 & 0 & 1 & 0 & 1 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 & 0 & 1 & 1 & 0 & 1 & 1 & 1 \end{bmatrix}$$

(a) G informs us how to connect the four input bits to binary adders to produce the seven parity bits. Draw such a diagram.

(b) Find $d_{\min}$ by calculating the weights of all nonzero codewords or, equivalently, the weights of all (nonzero) linear combinations of rows of **G**.

(c) Describe a syndrome table or standard array. How many single-, double-, and triple-error patterns are correctable in a complete decoder (mode 2)?

(d) If we operate a decoder in bounded-distance decoding mode, decoding any vector within two Hamming units of a codeword and flagging the remainder, find the probability of correct decoding and the probability of incorrect decoding, or undetected error.

5.2.5. Using the MacWilliams identity, compute the weight spectrum for the (15, 11) Hamming code, which is the dual of the (15, 4) maximal length code. For maximal length codes, there is one word of weight 0 and $2^k - 1$ words of weight 2^{k-1}. Check your result against the known weight spectrum for Hamming codes.

5.2.6. The dual code of the (11, 4) code in Exercise 5.2.4 is an (11, 7) binary code.

(a) What is its generator matrix?

(b) Find the weight spectrum of the (11, 4) code; then use the MacWilliams identity to find the weight spectrum of the 128 codewords in the (11, 7) code.

5.2.7. Michelson and Levesque [6] discuss a simple approximation to the weight spectrum for binary codes having the all 1's vector as a codeword. There are $2^k - 2$ words whose weights lie between $d_{\min}$ and $n - d_{\min}$, and the approximation is to apply a binomial distribution over this range, suitably normalized. Thus,

$$A_w \approx [C^n_m 2^k - 2] / \sum_{j=d_{\min}}^{n-d_{\min}} \binom{n}{j} \approx 2^{-(n-k)} C^n_w.$$

(a) Test the accuracy of this approximation on a (17, 9) code having $d_{\min} = 5$ and weight spectrum for weights 5 through 12 of 34/68/68/85/85/68/68/34.

(b) Test the approximation on a (21, 16) code with $d_{\min} = 3$ and weight spectrum for weights 3 through 18 of 42/210/651/1638/3570/6468/9310/10878/10878/9310/6468/3570/1638/651/210/42. Generally, the approximation is best for large, high-rate codes.

5.2.8. Show that syndrome decoding of linear codes is equivalent to ML decoding directly from **r**, in the sense that the solution set for the error pattern based on the syndrome $\mathbf{s} = \mathbf{r}\mathbf{H}^T = \mathbf{e}\mathbf{H}^T$ is the same set as obtained by considering the test error patterns of the form $\mathbf{e}_j = \mathbf{r} - \mathbf{x}_j$.

5.2.9. Show that q-ary Hamming codes can be structured so that the syndrome $\mathbf{s} = \mathbf{y}\mathbf{H}^T$ yields the base q representation of the error location and the value of the error, assuming that zero or one error occurs. *Hint*: The parity check matrix should have its columns ordered lexicographically.

5.2.10. (R. Gallager) Consider two systematic (7, 3) binary codes. Both are systematic. For code I, the parity equations are $x_3 = u_0 + u_1$, $x_4 = u_0 + u_2$, $x_5 = u_1 + u_2$, $x_6 = u_0 + u_1 + u_2$.

[32] D. Slepian, "A Class of Binary Signaling Alphabets," *Bell System Tech. J.*, vol. 35, pp. 203–234, 1956.

For code II, everything is the same, except that $x_5 = u_1$.

(a) Determine the generator matrices and parity check matrices for both codes.

(b) Argue that code I has minimum distance 4, while code II has minimum distance 3.

(c) Despite the result part of (b), show that code II has (slightly) smaller error probability for complete decoding on a BSC. You will need to generate the syndrome decoding table.

This exercise illustrates that greater minimum distance is not a strict indicator of superior error performance. The second code is a quasi-perfect code.

5.2.11. A (15, 5) binary code has generator matrix whose five rows are the 11-place vector (11101100101) and four right shifts of this vector, zero padding assumed. The resulting code is not in systematic form.

(a) Put **G** into systematic form by elementary row operations on **G**.

(b) Determine **H**.

(c) What is the size of the standard array for syndrome decoding?

(d) Argue that all sets of six columns in **H** are linearly independent; hence $d_{\min} \geq 7$.

(e) Show that in fact $d_{\min} = 7$.

(f) How many syndromes are consumed by error patterns with weight ≤ 3, and what actions could be taken for the remaining syndromes?

5.2.12. Form the generator matrix for the first-order Reed–Muller code with block length 32. What is k and $d_{\min}$?

5.2.13. Prove that on a q-input, $(q+1)$-output symmetric channel with erasure declaration a code having minimum distance $d_{\min}$ is capable of correctly processing any combination of t_1 errors and t_2 erasures, provided that $2t_1 + t_2 \leq d_{\min} - 1$. In particular, show that such a code can fill $d_{\min} - 1$ erasures if no other errors occur.

5.2.14. The ISBN numbering scheme for cataloging books uses a linear block code over GF(11) to supply error-detection capability for decimal numbers. Specifically, each book is assigned a nine-digit ($k = 9$) decimal string, $(x_0, x_1, \ldots, x_8)$ indicating language, publisher, and book number. A tenth digit from the field, x_9, is appended so that

$$\sum_{j=0}^{9}(j+1)x_j = 0, \quad \text{modulo } 11.$$

(The use of a field with 11 elements is apparently superfluous for a decimal code, but there are no fields of size 10.) If the parity symbol indeed is the eleventh field element (not 0 through 9), then the symbol X is assigned. For example, Hill's book *A First Course in Coding Theory*, from which this example is borrowed, has ISBN 0-19-853803-0, satisfying the preceding parity check equation.

(a) Verify that the encoding is correct for the ISBN code assigned to this book.

(b) Specify the parity check matrix for this code.

(c) Show that the code is capable of detecting any single-digit error (made by a typist or a computer) *and* detecting any transposition of two symbols (a frequent type of error, by humans at least). This requires showing that all such error types produce nonzero sum in the parity check process (or, equivalently, produce nonzero syndrome).

5.2.15. The weight enumerator polynomial for the binary (23, 12) Golay code is

$$A(z) = 1 + 253z^7 + 506z^8 + 1288z^{11} + 1288z^{12} + 506z^{15} + 253z^{16} + z^{23}.$$

(a) Evaluate the probability of decoding error for a complete decoder on a BSC with $\epsilon = 0.005$.

(b) If the code were used only for error detection, calculate P_{UE} on this same channel.

(c) Suppose that we use the code for error correction and detection and agree to "correct" up to two errors, instead of the guaranteed three. Calculate $1 - P_{\text{CD}}$ and P_{UE}. Note the tradeoff between undetected error probability and correct decoding probability.

5.2.16. Repeat Exercise 5.2.15 if the same code is used on a pure erasure channel with $\delta = 0.05$. The code allows up to six-erasure correction.

5.2.17. A $(6, 3)$ code over GF(4) is generated by

$$\mathbf{G} = \begin{bmatrix} 1 & 0 & 0 & 1 & 1 & 1 \\ 0 & 1 & 0 & 1 & \alpha & \alpha^2 \\ 0 & 0 & 1 & 1 & \alpha^2 & \alpha \end{bmatrix}$$

(a) Show that the code is self-dual.

(b) Show that the code is MDS; that is, $d_{\min} = n - k + 1 = 4$.

5.2.18. Suppose that we use two codewords $(1, 1, 1)$ and $(0, 0, 0)$ with antipodal signaling on an AWGN channel. Let the demodulator quantize the correlator output for each symbol to one of four levels, with thresholds set at 0 and $\pm 0.5\sqrt{N_0/2}$. We thus have a binary-input/quaternary-output DMC. Assume that $E_s/N_0 = 3$ dB. Specify the log-likelihood metric table for each quantized symbol and then a decoding algorithm based on this that is maximum likelihood for the quantized channel.

5.3.1. Consider the $(6, 3)$ binary code over GF(2) whose generator matrix is given in (5.2.20). The code is single error correcting ($t = 1$), but it is not a perfect code. Show by appeal to the standard array for this code shown in the text that the code is quasi-perfect.

5.3.2. We know that a $(15, 11)$ Hamming code has $d_{\min} = 3$ and that the expurgated code $(15, 10)$ has $d_{\min} = 4$. We may wonder whether a $(15, 9)$ code exists with distance 5 (if so it would be double error correcting). Apply the Hamming and Gilbert bounds to the question of the existence of such a code. What do these bounds say about the possibility of a $(15, 8)$ code with distance 5? *Remark*: We know that a $(15, 7)$ (linear) BCH code has $d_{\min} = 5$ and that there is a nonlinear $(15, 8)$ code due to Nordstrom and Robinson [12] that has distance 5.

5.3.3. Verify the claim used in the proof of the Plotkin bound that in each column of a complete listing of codewords in a linear code each of the q symbols is employed exactly q^{k-1} times, provided that no column of the **G** matrix is the zero vector.

5.3.4. Suppose that we are interested in binary codes for 24-bit messages and wish the code rate to be $\frac{1}{2}$; that is, we wish the block length to be 48. Use the Hamming and Varshamov/Gilbert bounds to place limits on the minimum distance for such a code. *Remark*: A $(48, 24)$ extended quadratic residue (QR) code has minimum distance of 12 [5].

5.3.5. Using the argument of the Gilbert bound, construct a binary $(7, 3)$ $d_{\min} = 4$ code.

5.3.6. Use the Varshamov bound argument to construct a parity check matrix for a binary $(8, 4)$ code with $d_{\min} = 4$.

5.3.7. Still another upper bound on minimum distance for (n, k) codes over GF(q) is the Griesmer bound,[33] which states that

$$n \geq d_{\min} + \left\lceil \frac{d_{\min}}{q} \right\rceil + \left\lceil \frac{d_{\min}}{q^2} \right\rceil + \cdots + \left\lceil \frac{d_{\min}}{q^{k-1}} \right\rceil = \sum_{i=0}^{k-1} \left\lceil \frac{d_{\min}}{q^i} \right\rceil$$

(a) Show that this implies the Singleton bound $n - k + 1 \geq d_{\min}$.

(b) If a code is MDS, that is, the code achieves the Singleton bound, what does the Griesmer bound say about the alphabet size q?

(c) Show that the upper bound on $d_{\min}$ for binary $(15, 7)$ codes is 6 by this result.

[33] J. H. Griesmer, "A Bound for Error Correcting Codes," *IBM J. Research Development*, pp. 532–542, November 1960.

5.3.8. Plot the asymptotic (large block length) forms of the Hamming and Varshamov–Gilbert bounds on $d_{\min}/n$ as a function of R for codes over GF(8). Also, show the Singleton and Plotkin upper bounds on this plot.

5.3.9. A (32, 16) Reed–Muller code has $d_{\min} = 8$. Assess this in light of the Hamming and Varshamov bounds.

5.4.1. Verify that $D - 1$ is a divisor of $D^n - 1$ for any n in any field and thus that there exists a simple $(n, n-1)$ cyclic code of any length over GF(q). This is just the code formed by appending an overall parity symbol to an $(n-1)$-symbol message. Diagram an encoder and syndrome former, each of which employs a one-cell feedback shift register.

5.4.2. Show that the only cyclic $(2k, k)$ rate $\frac{1}{2}$ codes are those for which $g(D) = D^k - 1$ (which obviously factors $D^{2k} - 1$). These cyclic codes all have $d_{\min} = 2$ for any block length and are thus very poor codes. If we need exactly $R = 1/2$, it is far better to modify another code and accept the loss of the cyclic property.

5.4.3. Produce a length-63 narrow-sense BCH code with design distance $\delta = 5$. Use the information about GF(64) in Figure 5.1.4 to determine minimal polynomials for α and for α^3. Find the generator polynomial and determine that the number of information symbols in the code is $k = 51$. (The actual minimum distance is in fact 5.)

5.4.4. With the advent of inexpensive, fast semiconductor memory, encoder/decoder design sometimes is effectively done with use of read-only-memory (ROM) tables. For the extended Golay (24, 12) code, describe a ROM implementation of an encoder for the systematic code and a ROM-assisted syndrome decoder (we still use a feedback shift register to compute the syndrome). Repeat for the (48, 24) extended quadratic residue code, and comment on feasibility of this approach.

5.4.5. There are many "tricks" associated with actual implementation of encoders and decoders, particularly over larger fields. Consider implementation of the encoder for a (255, 252) RS code over GF(256), which is single error correcting/double error detecting. The field elements are represented as 8-bit bytes. The generator polynomial is of the form

$$g(D) = (D - \beta)(D - \alpha\beta)(D - \alpha^2\beta),$$

where β is an arbitrary field element and α is a primitive element. Thus, three consecutive powers of a field element are roots of the generator polynomial, as required. Remember for RS codes that the starting element in the root string is arbitrary.

(a) Show that choice of $\beta = \alpha^{-1}$ leaves $g(D)$ in the form

$$g(D) = D^3 + (1 + \alpha + \alpha^{-1})D^2 + (1 + \alpha + \alpha^{-1})D + 1,$$

so in implementing the encoder as a feedback shift register, only a *single* GF(256) multiplier is required. (Such "reversible" generators are discussed in *IEEE Trans. Information Theory*, vol. IT-28, pp. 869–874, 1982.)

(b) Diagram the encoder, and show a gate-level diagram of the binary hardware needed to perform the addition and multiplication by the given field element. Assume that a primitive polynomial $f(X) = X^8 + X^4 + X^3 + X^2 + 1$ is used for defining the field GF(256).

5.5.1. **(a)** Prove the distributive property for the modulus operator with respect to polynomials; that is,

$$[a(D) + b(D)] \bmod g(D) = [a(D) \bmod g(D) + b(D) \bmod g(D)] \bmod g(D)$$

$$= a(D) \bmod g(D) + b(D) \bmod g(D).$$

[*Hint*: define $s(D)$ by Euclid's theorem: $a(D)+b(D) = q(D)g(D)+s(D)$, and likewise for $s_a(D)$ and $s_b(D)$; for example, $a(D) = q_a(D)g(D) + s_a(D)$. Then use uniqueness of the quotient and remainder polynomials to prove that $s(D) = s_a(D) + s_b(D)$.]

(b) Prove that a similar property also holds for the multiplication of polynomials: $[a(D)b(D)] \bmod g(D) = [[a(D) \bmod g(D)][b(D) \bmod g(D)]] \bmod g(D)$.

5.5.2. (a) Calculate the fractional volume contained in the seven-dimensional space of seven-tuples over GF(8) that is contained in the spheres of radius 2 around each codeword in the (7, 3) RS code. This should illustrate why the probability of decoding failure can be rather large compared to the probability of incorrect decoding.

(b) Repeat this calculation for a binary (127, 106) triple-error-correcting BCH code.

5.5.3. A (15, 7) BCH code is generated by

$$g(D) = D^8 + D^7 + D^6 + D^4 + 1.$$

(a) What is the parity check polynomial $h(D)$, and what are the parameters of the dual cyclic code it generates?

(b) Diagram a systematic encoder and the syndrome former for the (15, 7) code.

(c) Determine if $\mathbf{r}$ = (000110111001000) is a valid codeword, with the rightmost bit representing the leading information symbol, by computing the syndrome [dividing $r(D)$ by $g(D)$].

(d) Decode the previous received vector by computing the syndromes $S_i, i = 0, \ldots, 3$, and then the connection polynomial $B(D)$, and use this to extrapolate to the remaining error transform digits. How many errors does the decoder perceive to have occurred? Does the final syndrome check produce the zero vector? [Partial answer: $B(D) = 1 - \alpha^{10}D - \alpha^6 D^2$.]

5.5.4. Argue that the following errors-and-erasures procedure decodes correctly for *binary* codes when r errors and s erasures occur, if $2r + s \leq \delta - 1$. Replace all erasures with 0's and decode, if possible. Determine the weight of the resulting error pattern. Replace all erasures with 1's and decode if possible. Determine the weight of the resulting error pattern. Decide in favor of the lower-weight error pattern. (*Hint*: Assume that of the s erasures b were originally 0 symbols and $s - b$ were originally 1's.)

5.5.5. Repeat the decoding exercise in Example 5.25 with $e(D) = 1 + D + \alpha^2 D^2$; that is, add an additional error in position 1. The decoding attempt should probably fail since the true number of errors exceeds 2.

5.5.6. The (15, 9) RS code over GF(16) is capable of correcting $t = 3$ symbol errors. Let the code have roots $\alpha^1, \alpha^2, \ldots, \alpha^6$, where α is a primitive element in GF(16), so the generator polynomial is

$$g(D) = (D - \alpha)(D - \alpha^2) \cdots (D - \alpha^6).$$

Suppose that the all-zeros message is sent, and there are two errors α and α^7 in positions 0 and 1 of the codeword.

(a) Compute the syndromes $S_0, S_1, \ldots, S_5$. Use either direct computation by (5.5.9) or recall that $S_i = R_{i+j}$, where $\mathbf{R}$ denotes the DFT of r.

(b) Use the Berlekamp–Massey algorithm to solve for the minimal-order feedback shift register capable of producing the observed sequence. (This should be a second-degree "filter".)

(c) Use the LFSR produced in part (b) to produce the remaining digits of the error transform sequence.

(d) Perform the inverse transform to determine the error pattern.

(e) Correct the received codeword.

(f) Recheck the syndromes to see if the decoded output is a valid codeword.

5.5.7. Repeat Exercise 5.5.5 if the received vector has an erasure in position 0 and an error of type α^7 in position 1. This is a correctable error situation as well.

5.5.8. Repeat Exercise 5.5.5 with the first six positions erased, all other symbols being received correctly. Show that correct decoding ensues here with the errors-and-erasures algorithm.

5.5.9. Perform Wagner decoding for an (8, 7) single parity bit code used to send ASCII characters. Let logical 0 correspond with $-A$ signal level at the demodulator output and logical 1 correspond with signal level A at the same point. Suppose the received signal strength corresponds to $A = 1$ volt and that the received analog sequence in the presence of noise is

$$\mathbf{r} = (-1.1, -0.5, 0.1, 0.2, -1.2, -1.1, -0.6, 1.2).$$

What is the decoded message? Would hard-decision decoding have produced the same estimate?

5.5.10. Perform Chase decoding (Algorithm II of the text) for the (7, 4) Hamming code in conjunction with FSK signaling and noncoherent detection. Assume that the all-zeros sequence is selected for transmission. Let the sequence of noncoherent matched filter detector outputs be

$$\vec{\mathbf{r}} = \begin{bmatrix} 2.5 & 1.5 & 0.3 & 2.0 & 0.4 & 2.0 & 0.8 \\ 0.2 & 0.9 & 0.2 & 0.5 & 0.6 & 1.2 & 0.4 \end{bmatrix},$$

where the zero channel output appear on top. Perform hard-decision decoding on $\vec{\mathbf{r}}$, and locate the $J = \lfloor \frac{3}{2} \rfloor$ lowest-confidence decision to form a single test vector $\mathbf{z}$. Perform algebraic decoding of the hard-decision vector and of the perturbed hard-decision vector. Which codeword produced has greater likelihood for this problem, using the log-likelihood metric

$$\Lambda(\vec{\mathbf{r}}, \mathbf{x}) = \sum_{j=0}^{6} \log I_0\left(\frac{\mu r_{x_j}}{\sigma^2}\right).$$

Assume that the symbol energy-to-noise density ratio is 4 dB.

5.6.1. Extend the (7, 4) code to (8, 4) by adding an overall parity bit to each codeword. Write the parity check matrix for this code by augmenting the former check matrix to reflect the new constraints.

(a) Verify that every codeword has even weight.

(b) Show that $d_{\min} = 4$.

(c) Show that this code is self-dual.

(d) Argue through use of the syndrome table that the decoder is capable of simultaneously correcting a single error and detecting two errors.

(e) If symbols 0 and 1 are mapped to antipodal signals, show that the resulting set of 16 signals is biorthogonal.

5.6.2. Draw a diagram similar to that of Figure 5.6.1 for modifications of the (15, 7) binary BCH code.

5.6.3. Prove that, when shortened, a code's minimum distance cannot decrease.

5.6.4. Show that an $(r - 1)$st-order Reed–Muller code is an expurgation of the rth-order RM code.

5.6.5. Show how to implement a simple modification of the systematic encoding technique of Section 5.4 to extend a code by 1 bit, enforcing even parity on the codeword. A single 1-bit accumulator is sufficient.

5.6.6. RS codes can be lengthened by two symbols without cost in $d_{\min}$. Show that where $d_{\min} = 3$ twice-lengthened RS codes are perfect codes over GF(q), equivalent to Hamming codes. (*Hint*: Show that all syndromes are exactly consumed by 0- and 1-error patterns.)

5.7.1. Design an error-detection scheme that operates with 256-bit messages and must correctly detect byte (8-bit) burst errors, as well as detect any four randomly placed errors in a block. (The CRC code of the text example meets the requirements, but is overdesigned.) You should be able to show that even if the code is not cyclic, but merely produced as if the code were cyclic, for example, $x(D) = u(D)g(D)$, that the decoder is capable of detecting all error patterns confined to $n - k - 1$ bits.

5.7.2. In the Ethernet protocol for local area networks, a CRC code is employed with generator polynomial given by

$$g(D) = D^{32}+D^{26}+D^{23}+D^{22}+D^{16}+D^{12}+D^{11}+D^{10}+D^{8}+D^{7}+D^{5}+D^{4}+D^{2}+D^{1}+1.$$

This polynomial is primitive of degree 32 and hence would generate a binary Hamming code of length $2^{32} - 1$, with $d_{\min} = 3$. Although this block length is much longer than used in the standard implementation, specify the random and burst-error detecting performance of the system.

(a) Fujiwara et al. [53] discuss the performance of this code under varying amounts of code shortening. In particular, it is shown that if the block length is limited to 512 bits or less by shortening, then the minimum distance increases to 5. In this case, what can be said about the random and burst-error detecting guarantees?

(b) Normally, only error detection is attempted with such CRC codes, but they could be employed as combination error-correction and error-detection codes. Discuss how the given code could be employed with $n = 512$ to correct single errors, while still guaranteeing the detection of any two- or three-error pattern.

5.7.3. An 8-bit CRC code is employed in the adaptation layer of the ATM protocol, coding a 4-byte header containing addressing and routing information. In effect, we obtain a (40, 32) code, with $d_{\min} = 4$. The CRC polynomial is $g(D) = D^8 + D^2 + D + 1$. What claims can be made about the detection of errors in the header field?

5.8.1. (a) Demonstrate that in a product coding scheme the minimum distance between two-dimensional code arrays is $d_1 d_2$, where these are the row and column minimum distances, respectively.

(b) Extend the two-dimensional product coding concept to three dimensions, and generalize the distance and error-correcting capabilities.

5.8.2. Apply row/column coding as follows for a binary symmetric channel. Let the row code be a binary (15, 11) perfect Hamming code. Let the column code be a (255, 239) binary BCH code. Estimate the probability of a block (array) error under the following regimes:

(a) Perform complete decoding of the row code; assume that when an error occurs it is most likely due to a two-error pattern and that decoding is to a message with three errors in the 15 positions distributed equally likely.

(b) Perform error detection on the row code; a single- or double-error event produces an error detection, in which case we erase the entire row prior to column decoding. In this mode, the column decoder will be presented with erasures and residual (undetected) errors.

5.8.3. Consider a concatenation of a (15, 9) Reed–Solomon outer code over GF(16) with a (7, 4) binary inner code.

(a) If the binary code symbols are transmitted using PSK, find the resultant signal bandwidth normalized to the input bit rate.

(b) Assuming reception on the coherent AWGN channel with $E_b/N_0 = 7$ dB, determine the probability of an inner codeword error for both hard-decision decoding and ML decoding of the inner code.

(c) Compute $1 - P_{\text{CD}}$ for the outer code, assuming that it is given hard decisions on the inner codewords, which it sees as GF(16) decisions.

(d) If we view this scheme as a binary code, what are n and k? What is the overall code rate? Find a shortened BCH code with roughly these same parameters and estimate its minimum distance.

5.8.4. We have seen that the two-stage decoder for product codes falls short of the capability of an ML decoder in guaranteed error-correcting power, yet many error patterns beyond the guaranteed weight are correctable. Suppose that the (15, 11) binary code is used in each dimension. Some four-error patterns cause decoding failure for a two-stage decoder. What fraction of four-error patterns are in fact correctly decoded, however?

5.8.5. Analyze a concatenated scheme with overall code rate $\frac{1}{2}$, comprised of a (256, 240) lengthened RS code over GF(256) as an outer code and a (15, 8) Nordstrom–Robinson nonlinear code with $d_{\text{min}} = 5$ as an inner code. The channel is a BSC with $\epsilon = 0.02$.

(a) Compute the probability of correct decoding for this scheme. Assume that both decoders operate as bounded-distance decoders. (Although the NR code is nonlinear, there is enough symmetry to allow every inner codeword to have equal decoding performance.)

(b) Now switch from a high-rate outer code/low-rate inner code to the opposite. Use a (12, 8) shortened Hamming code as an inner code and a (256, 192) RS outer code. Which scheme is better?

5.9.1. (a) For the Gilbert channel of Figure 5.9.1, the average error probability is 0.0167. Calculate the channel capacity for the perfectly interleaved version of this channel, which would be a BSC with $\epsilon = 0.0167$.

(b) The actual channel is a finite-state channel with memory, whose capacity may be shown to be [7]. Calculate the channel capacity of the actual channel, and observe that it is larger than that of the interleaved "equivalent" channel.

5.9.2. Design an interleaving scheme for a binary channel where block coding is used with $(n, k) = (24, 12)$ and error bursts up to length 8 bits are anticipated. Show by diagram how a block interleaver can handle up to three such bursts. What is the total memory requirement and the end-to-end delay in units of information bit times? Describe the synchronization difficulty inherent with the interleaver.

5.9.3. Show how to interlace a RS (7, 5) code over GF(8) to depth $D = 4$ by replacing each encoder call with four cells. Convince yourself that the corresponding decoder is capable of correcting any single burst of four symbols.

5.9.4. Diagram an encoder for a RS (7, 5) code over GF(8), interleaved to depth $D = 4$, using the interlacing technique shown in the text, replacing each usual encoder delay cell with four cells. Convince yourself that the decoder is capable of correcting any single burst confined to that contiguous symbols.

5.9.5. Convolutionally interleave a (7, 4) Hamming code to depth 3. Show that, by sending the all-zeros sequence, insertion of any 3-bit burst error pattern is scrambled into three separate codewords at the receiver and thus that the burst is correctable.

5.9.6. A binary code with block length $n = 63$ is to be used over a Rayleigh fading channel whose decorrelation time is 0.01 second.

(a) If the channel symbol rate is 16 kbps, design a block interleaver to effectively produce a memoryless channel as seen by the decoder. What is the total end-to-end delay of the system due to interleaving?

(b) Repeat for a convolutional interleaver.

5.9.7. A channel model commonly used to model bursty error conditions on a binary channel is the Gilbert model. We define the channel to have two states, *good* and *bad*. In the good state, the channel is perfect, while in the bad state, the channel error probability is 0.3. On each use of the channel, the state transitions are governed by $P[G \rightarrow B] = 0.01$ and $P[B \rightarrow G] = 0.2$. Using the theory of Markov chains, find the steady-state probability of being in the bad state and from this determine the error probability. (Note that this error probability would be the long-term average measured on a sample function of the process.) The mean dwell time in the bad state is $1/0.2 = 5$ symbols. Design an efficient interleaving scheme for this channel in conjunction with a binary (31, 26) error-correcting code.

5.10.1. Compute the probability of not decoding correctly when a (7, 4) Hamming code is employed on an AWGN channel with antipodal signals. The demodulator makes hard decisions on code bits. Plot this result as a function of E_b/N_0, and compare with the probability of message error for 4-bit uncoded messages on the same channel. Measure the coding gain at $P_E = 10^{-5}$

5.10.2. A (7, 3) code over GF(8) is used in conjunction with 8-PSK signaling on an AWGN channel. The code has $d_{\min} = 5$ and is therefore capable of overcoming two hard-decision errors.

(a) Let the bit rate be $R_b = 10^6$ bps. What is the encoded symbol rate?

(b) If the received power level is $P_r = 10^{-11}$ watts and the noise spectral density is $N_0/2 = 10^{-19}$ watts/hertz, determine the symbol error probability P_s and bound the decoder error probability by assuming any error pattern with more than two errors causes a decoding error.

(c) If maximum likelihood decoding were used for this channel and code, what metric should be used in evaluating codewords? How many elementary metric additions are necessary to compute all codeword likelihoods?

5.10.3. In Example 5.34, expressions were obtained for coded and uncoded performance in terms of P_s, the channel symbol error probability. Employ the union bound of Chapter 3, $P_s \leq (M-1)e^{-E_s/N_0}$, to obtain P_E as a function of E_b/N_0. Be careful to normalize the symbol energy properly in the two cases. By comparing resulting exponents, determine the relative energy efficiency.

5.10.4. A Reed–Solomon code over GF(16) is used in conjunction with 16-ary FSK modulation of code symbols. The code length is 15 and $k = 11$, so the code is double error correcting. The channel environment is white Gaussian noise, with $E_b/N_0 = 10$ dB. Assuming that noncoherent detection is performed and hard decisions are passed to the decoder, find the symbol error probability, remembering to normalize energy properly. Then calculate the probability of a block error, assuming that the decoder fails whenever three or more errors occur. Compare this performance with the probability of message error for 11-symbol messages sent uncoded on the same channel.

5.10.5. For the code of Exercise 5.10.4, repeat for a Rayleigh fading channel with an average $E_b/N_0 = 30$ dB. Assume hard decisions on code symbols and that codeword interleaving is used to make the 16-ary channel memoryless.

5.10.6. With the code of Exercise 5.10.4, assume that a jamming signal is randomly present with probability 0.01 and, when present, the effective E_b/N_0 is -20 dB. When the jammer is absent, there is zero noise in the channel. Analyze the performance of the coded system, first assuming that the decoder has no side information about jammer presence and does errors-only decoding. Then consider a decoder that is informed of which symbols have been jammed and that simply erases these positions. This decoder can err if more than four symbols are jammed.

5.10.7. Show that the asymptotic coding gain of a binary $(n, n-1)$ single parity bit code, having $d_{\min} = 2$, is 3 dB when soft-decision decoding is employed on the antipodal AWGN channel. (Assume that n is large.) In conjunction with Wagner decoding of the code, it is relatively easy to achieve this 3-dB gain in efficiency.

5.10.8. CRC codes normally used for error detection are usually extended Hamming codes, with $d_{\min} = 4$ and rate R near 1. Thus, the asymptotic coding gain is a surprising 6 dB. (Chou, D.-P., and Wilson, S. G., *International Symposium on Information Theory*, 1991.) The ML decoder could be implemented with a trellis having 2^{n-k} states, but this is too complex typically. Propose a Chase-style suboptimal scheme that uses an algebraic decoder and bit flipping to approximate ML decoding.

5.10.9. Compute the asymptotic coding gains for the following codes when soft-decision decoding is employed on the antipodal AWGN channel. All codes have rate nearly $\frac{2}{3}$.

(a) (15, 10) expurgated Hamming code, $d_{\min} = 4$.

(b) (63, 45) BCH code, $d_{\min} = 7$.

(c) (127, 85) BCH code, $d_{\min} = 13$.

5.10.10. **(a)** Calculate the effective diversity order for the codes of Exercise 5.10.9 for the fully interleaved Rayleigh channel when hard-decision decoding is used.

(b) Repeat if soft-decision decoding is accomplished, again with interleaving.

5.11.1. If the Golay (23, 12) binary code modulates a carrier using antipodal PSK, show that the power spectrum is the same as that of uncoded PSK, but scaled to be 23/12 wider.

5.11.2. Show that the first-order Reed–Muller codes have the spectral equivalence property.

5.11.3. What is the smallest binary code of length $n = 63$ that can have the spectral equivalence property? (*Hint*: All columns of **G** must remain distinct.)