

CHAPTER 4: COMMUNICATION SYSTEMS

4.1 INTRODUCTION

Communication systems convey information from one point to another via physical channels that propagate electromagnetic, acoustic, particle density, or other waves. This information is usually manifest as voltages or currents; these may be continuous (often called *analog*) variables with an infinite number of possible values or discrete (often called *digital*) variables with a finite set of known possible values. Communications systems linking machines include *telemetry systems* that convey sensory data one way, *networks* that convey data two-way among multiple nodes, and *memory systems* that store and recall information.

Communications systems can not only link people or systems at great distances via audio, visual, computer, or other messages, but may link the various parts within systems, and even within single semiconductor chips. They may communicate information in two directions, or only one way, and they may involve one node broadcasting to many, one node receiving from many, or a finite set of nodes communicating among themselves in a network. Even active measurement and remote sensing systems can be regarded as communications systems. In this case the transmitted signals are designed to be maximally sensitive to the channel characteristics, rather than insensitive, and the receiver's task is to extract these channel characteristics knowing what was transmitted.

A two-node, one-way communication system consists of the channel that conveys the waves, together with a modulator and a demodulator. All communications systems can be regarded as aggregates of these basic two-node units. The modulator transforms the signal or symbol to be transmitted into the signal that is propagated across the channel; the channel may add noise and distortion. The task of the demodulator is to analyze the channel output and to make the best possible estimate of the exact symbol transmitted, accounting for the known channel characteristics and any user concerns about the relative importance of different sorts of errors. A sequence of symbols constitutes a message. A complete communications system is formed by combining many two-node, one-way systems in the desired configuration.

For convenience we shall designate systems that involve a finite, discrete set of intended messages as “digital”, and systems communicating continuous variables with an infinite number of possibilities as “analog”. We further designate digital systems as being *binary systems* if only two possible symbols exist, and *M-ary systems* otherwise, where M is the number of possible alternative symbols. Figure 4.1-1 presents a block diagram for a two-node, one-way M -ary communications system. In the figure the transducer at the output of the modulator develops a voltage output $v_o(t)$ that is a unique function of time depending on which of the M possible symbols are to be sent: S_1, S_2, \dots, S_M . This output signal could also be a current waveform, a

pressure waveform, a torsion waveform, a particle density waveform, or any other signal intended to propagate across the channel.

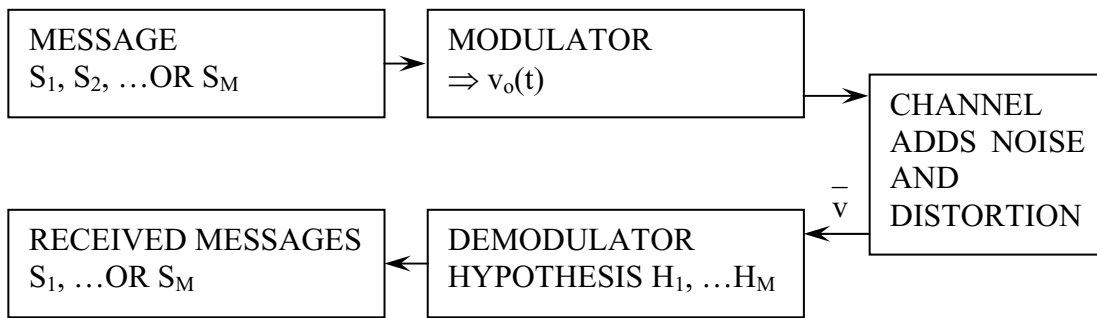


Figure 4.1-1: Communications system block diagram

In this general representation the channel might include an amplifier, transmitting antenna, propagation path, a receiving antenna, more amplifiers, and a detector. The antennas act as transducers between the waves that propagate in the channel and the voltages or other signals within the receiver and transmitter. Channel noise can be added at the transmitter, along the path, and in the receiver. This noise may be additive Gaussian white noise, various forms of multiplicative noise, or combinations thereof. Channels can also introduce systematic and random distortions, fading, and variations in propagation delay, possibly even leading to transpositional noise where some portions of the signal arrive out of sequence. The distinction between multiplicative noise and distortion is sometimes ambiguous.

A well-designed demodulator compensates for all the systematic distortions in the signal and hypothesizes which waveform was actually transmitted. This estimate is designed not only to choose the symbol most probably transmitted, but also to take account of any *cost function* that assigns different penalties to various sorts of errors. For an M-ary system a cost function is easily represented as a matrix, where the rows correspond to possible transmitted symbols, and the columns correspond to possible received symbols; in general this is an $M \times M$ matrix. Each entry in the matrix consists of a number representing the cost to the user associated with that possibility. As discussed later, the optimum demodulator is designed to minimize this average cost over the estimated probability distribution of transmitted symbols.

Section 4.2 discusses the design of optimum binary and M-ary communication systems, followed by the design of transmitted signals in Section 4.3. How well such systems perform depends partly on the channel characteristics. The system performance for various M-ary communication systems is evaluated in Section 4.4. Communication systems can be improved by coding the signals to survive channel impairments (channel coding) and to reduce their redundancy (source coding); these techniques are discussed in Sections 4.5 and 4.6, respectively. Representative analog communication systems, such as amplitude modulation (AM), frequency modulation (FM), single sideband (SSB), and others, are discussed in Section 4.7. Finally, system design issues are reviewed in Section 4.8 for a variety of wired, wireless, satellite, and

optical fiber communication systems. In Chapter 5 these communication system perspectives are also applied to active remote sensing systems.

4.2 BINARY AND M-ARY COMMUNICATIONS

In this section we focus on the design of an optimum demodulator for binary or M-ary communication systems. Signal design and system performance analyses for such systems are discussed in Sections 4.3 and 4.4, respectively.

In binary systems there are two possible transmitted symbols, designated here as S_1 and S_2 . We assume these have *a priori* probabilities of P_1 and P_2 , respectively, where “*a priori*” means prior to any information about the received signal. The modulator transmits $\bar{v}_1(t)$ or $\bar{v}_2(t)$, depending on the symbols; these can also be represented as vectors \bar{v}_1 and \bar{v}_2 , where the transmitted symbol waveforms are considered to be sampled signals. The demodulator must interpret the waveform emerging from the channel, represented as $v(t)$ or its sampled equivalent \bar{v} .

The task of designing an optimum demodulator can be viewed very simply. For example, suppose the received voltage is known to range between zero and 10 volts. The demodulator design can be characterized simply as dividing this possible range of voltages into two regions, V_1 and V_2 . If the received voltage v falls in the range of V_1 , we hypothesize that S_1 was transmitted; we designate this hypothesis as H_1 . Similarly, if the received voltage v falls in the region V_2 , then this leads to the alternate hypothesis H_2 . In a simple system V_1 might correspond to all voltages between zero and 5 volts, while V_2 corresponds to the rest.

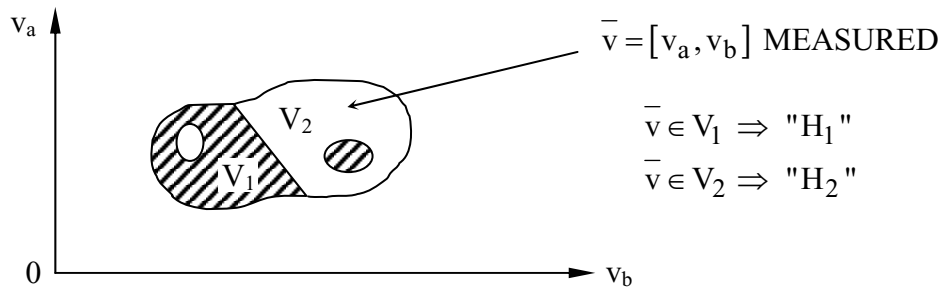


Figure 4.2-1: Demodulator decision regions in voltage space v_a, v_b

The situation becomes a little more complicated when the received voltage is a waveform. In the special case where the received waveform consists of two voltage samples ($\bar{v} = [v_a, v_b]$), the demodulator design can be represented by a two-dimensional figure where the axes correspond to the two voltage samples, v_a and v_b . The assignment of possible combinations of these two samples to the hypotheses H_1 and H_2 is illustrated in Figure 4.2-1. The situation pictured here is quite general and shows how all combinations of v_a and v_b may not be feasible, and also shows how the two regions V_1 and V_2 (corresponding to H_1 and H_2 , respectively) can

even be scrambled. This concept of partitioning the space of possible received signal waveforms into two or more regions corresponding to the set of possible transmitted symbols can clearly be extended to an arbitrary number of dimensions.

The task of demodulator design then consists in the binary case of defining the regions V_1 and V_2 . The cost function we wish to minimize when we optimize this demodulator is frequently simply the probability of error P_e . This error probability depends on the *a priori* probabilities of S_1 and S_2 (P_1 and P_2 , respectively) and on the conditional probabilities of receiving the voltage waveform \bar{v} when S_1 is sent ($p\{\bar{v}|S_1\}$). Thus

$$P_e = P_1 \int_{V_2} p\{\bar{v}|S_1\} d\bar{v} + P_2 \int_{V_1} p\{\bar{v}|S_2\} d\bar{v} \quad (4.2.1)$$

To simplify this expression to an integral over only one subspace, V_1 , we note that the total probability that some voltage waveform \bar{v} is transmitted is unity:

$$\int_{V_1} p\{\bar{v}|S_1\} d\bar{v} + \int_{V_2} p\{\bar{v}|S_1\} d\bar{v} = 1 \quad (4.2.2)$$

Therefore,

$$P_e = P_1 + \int_{V_1} [P_2 p\{\bar{v}|S_2\} - P_1 p\{\bar{v}|S_1\}] d\bar{v} \quad (4.2.3)$$

It is clear from (4.2.3) that to minimize the probability of error we simply assign a given point in the received-signal space to the region V_1 when

$$P_1 p\{\bar{v}|S_1\} > P_2 p\{\bar{v}|S_2\} \quad (4.2.4)$$

That is, in this binary case, we simply choose hypothesis 1 or 2 according to which received signal \bar{v}_i has a greater *a posteriori* probability $P_i p\{\bar{v}|S_i\}$.

A simple example is the one-dimensional binary case with additive Gaussian noise. Assume that the message S_1 corresponds to a transmitted signal of A volts, while S_2 corresponds to zero volts. If the additive Gaussian noise has variance $\sigma_n^2 = N$, then the conditional probabilities are

$$\begin{aligned} p\{\bar{v}|S_1\} &= \frac{1}{\sqrt{2\pi N}} e^{-(v-A)^2/2N} \\ p\{\bar{v}|S_2\} &= \frac{1}{\sqrt{2\pi N}} e^{-v^2/2N} \end{aligned} \quad (4.2.5)$$

These conditional probabilities are plotted in Figure 4.2-2, but are scaled by the *a priori* probabilities P_1 they show how the possible voltages v are assigned to the two regions V_1 and V_2 in accord with (4.2.4). In Figure 4.2-2a the *a priori* probabilities are equal for the two possible messages, and the decision threshold $A/2$ given by (4.2.4) falls halfway between zero and A volts. Figure 4.2-2b shows how the decision threshold is biased so as to shrink the region V_1 assigned when $P_1 < P_2$. That is, P_e is minimized by biasing the outcome to favor the message more likely to have been transmitted *a priori*.

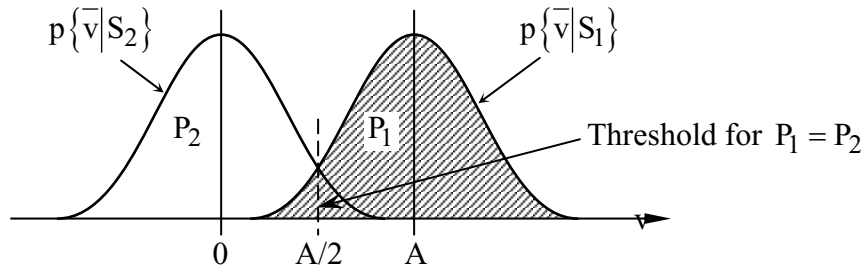


Figure 4.2-2a: *A posteriori* probabilities for equal *a priori* probabilities.

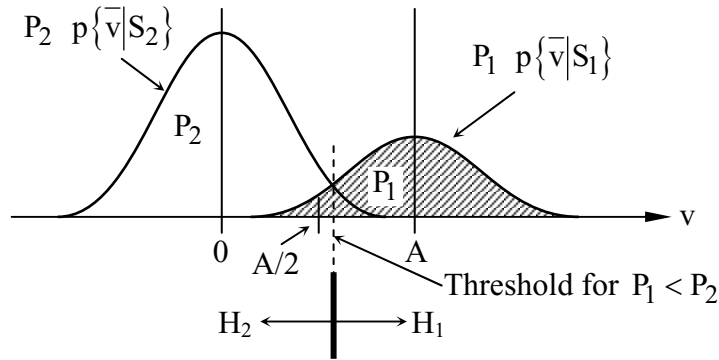


Figure 4.2-2b: *A posteriori* probabilities for unequal *a priori* probabilities.

This rule for defining the region V_1 given by (4.2.4) can be expressed in terms of the likelihood ratio ℓ :

$$\ell \equiv \frac{p\{\bar{v}|S_1\}}{p\{\bar{v}|S_2\}} > \frac{P_2}{P_1} \Rightarrow V_1 \quad (4.2.6)$$

The decision rule in (4.2.6) can equivalently be expressed in terms of the logarithm of the likelihood ratio

$$\ln \ell > \ln(P_2/P_1) \Rightarrow V_1 \quad (4.2.7)$$

The advantage of using the logarithm for additive Gaussian noise follows from its simple form

$$\begin{aligned}\ln \ell &= \left[-(v - A)^2 / 2N \right] + v^2 / 2N \\ &= (2vA - A^2) / 2N\end{aligned}\tag{4.2.8}$$

Combining (4.2.7) and (4.2.8) leads us to a simple decision rule: choose V_1 if

$$v > \frac{A^2 + 2N \ln(P_2/P_1)}{2A}, \text{ or}\tag{4.2.9}$$

$$v > \frac{A}{2} + \frac{N}{A} \ln(P_2/P_1)\tag{4.2.10}$$

The binary decision rule of (4.2.10) is simple; the decision threshold lies halfway between the two transmitted voltages, zero and A , but biased by a term proportional to the imbalance in *a priori* probabilities, $\ln(P_2/P_1)$, and also proportional to the ratio of noise variance N to the voltage difference A . Thus the bias is zero if either the noise N is zero or if the *a priori* probabilities P_1 and P_2 are equal.

If a certain rms voltage is being added to the received signal, superior performance can be obtained by averaging many such independent samples. Let the transmitted signals S_1 and S_2 each consist of m samples. The general decision rule (4.2.6) still applies, where \bar{v} is now a vector consisting of m entries. To compute this likelihood ratio we need to know the conditional probabilities $P\{\bar{v}|S_1\}$. In the case where the noise bandwidth is sufficiently broad that each sample v_1 is independent, then the joint probability equals the product of the probability distributions for each sample separately:

$$p\{v_1, v_2, \dots, v_m | S_1\} = \prod_{i=1}^m p\{v_i | S_1\}\tag{4.2.11}$$

Figure 4.2-3 illustrates two representative waveforms S_1 and S_2 consisting of m samples. If each waveform lasts T seconds, then every T seconds we can send one or the other, so as to convey ultimately a complex message.

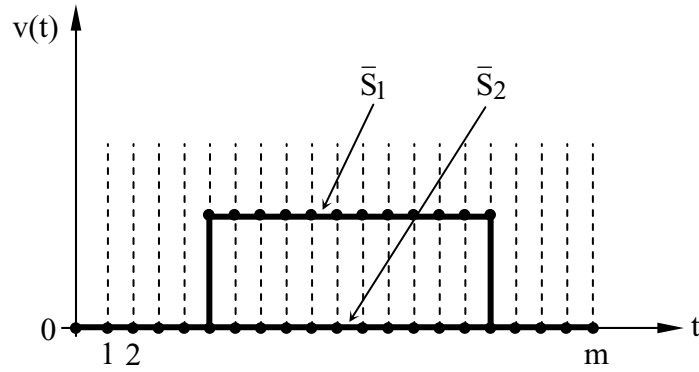


Figure 4.2-3: Alternate waveforms with m samples.

To evaluate (4.2.11) under the Gaussian noise assumption, we note

$$p\{v_i | S_1\} = \frac{1}{\sqrt{2\pi N}} e^{-(v_i - S_1)^2 / 2N} \quad (4.2.12)$$

Therefore the conditional probability distribution we need for evaluating our decision rule (4.2.6) is

$$p\{\bar{v} | S_1\} = \frac{1}{(\sqrt{2\pi N})^m} e^{-\sum_{i=1}^m (v_i - S_1)^2 / 2N} \quad (4.2.13)$$

and the decision rule (4.2.7) becomes

$$\ln \ell = \frac{1}{2N} \left[\sum_{i=1}^m (v_i - S_{2i})^2 - \sum_{j=1}^m (v_j - S_{1j})^2 \right] > \ln \frac{P_2}{P_1} \Rightarrow V_1 \quad (4.2.14)$$

where we note

$$\sum_{i=1}^m (v_i - S_{2i})^2 = |\bar{v} - \bar{S}_2|^2 \quad (4.2.15)$$

Since

$$|\bar{v} - \bar{S}_2|^2 - |\bar{v} - \bar{S}_1|^2 = -2\bar{v} \cdot \bar{S}_2 + \bar{S}_2 \cdot \bar{S}_2 + 2\bar{v} \cdot \bar{S}_1 - \bar{S}_1 \cdot \bar{S}_1 \quad (4.2.16)$$

it follows that the decision rule (4.2.14) can be rearranged to give

$$\bar{v} \cdot (\bar{S}_1 - \bar{S}_2) > \frac{\bar{S}_1 \cdot \bar{S}_1^* - \bar{S}_2 \cdot \bar{S}_2^*}{2} + N \ln(P_2/P_1) \quad (4.2.17)$$

The right-hand side of (4.2.17) is a bias term, the first part of which is zero if the energies in the two waveforms S_1 and S_2 are equal, and the second part is zero if either the additive noise N is zero or if the *a priori* probabilities P_1 and P_2 are equal. The vector dot product (or correlation between the received signal \bar{v} and the two possible noiseless transmitted signals \bar{S}_1 and \bar{S}_2) performs the function of a *matched filter*. A simple block diagram of a communications receiver that computes (4.2.17) to yield one of two hypotheses H_1 or H_2 is illustrated in Figure 4.2-4. This receiver compares the output of a filter matched to \bar{S}_1 with the output from a filter matched to \bar{S}_2 .

The decision rule (7.2.17) that minimizes the probability of error can be generalized to M possible hypotheses by shifting the right-hand side of (4.2.17) to the left. That is, we hypothesize that the transmitted signal is the symbol i that yields the largest biased matched filter response:

$$f_i = \bar{v} \cdot \bar{S}_i - \frac{\bar{S}_i \cdot \bar{S}_i^*}{2} + N \ln P_i > \text{all } f_{j \neq i} \quad (4.2.18)$$

The receiver illustrated in Figure 4.2-4 can be generalized for the M -ary case simply by adding more matched filters in parallel feeding the same final amplitude measurement unit that chooses the receiver hypothesis H .

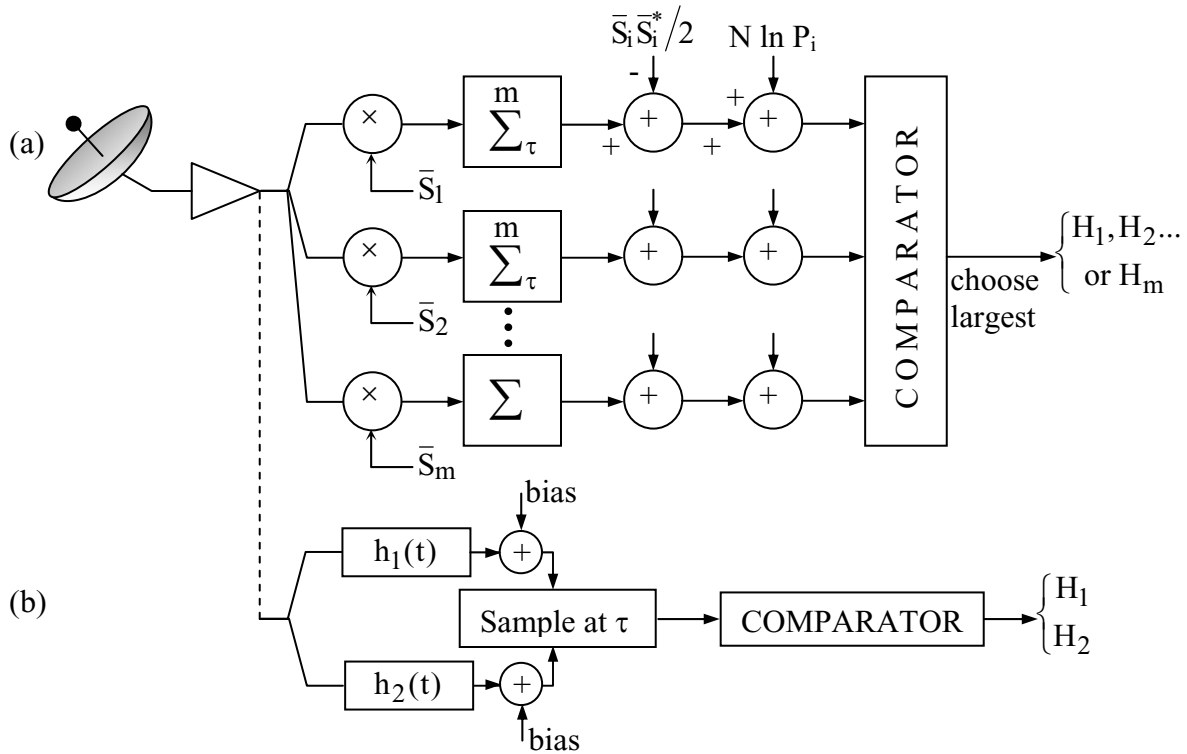


Figure 4.2-4: Decision-making receiver. The biases added to $\bar{v} \cdot \bar{s}_i$ are those given in (4.2.18)

Example 4.2.1

A transducer sends either 1 volt (state 1) or 3 volt (state 2) depending on its state, which has a probability of 0.1 or 0.9, respectively. Uniformly distributed noise ranging from -3 to $+3$ volts is added to the signal. How should the range -2 to $+6$ volts be assigned to H_1 and H_2 in order to minimize the probability of error P_e ? Suppose $P_1 = 0.9$?

Solution:

Using (4.2.4) we choose H_1 when $P_1 p\{v|S_1\} > P_2 p\{v|S_2\}$. Here $p\{v|S_1\} = 0$ for $v > 4$, and $p\{v|S_1\} = 1/6$ for $-2 < v < 4$. Similarly, $p\{v|S_2\} = 0$ for $v < 0$, and $p\{v|S_2\} = 1/6$ for $0 < v < 6$. Since $P_1 = 0.1$ and $P_2 = 0.9$, we choose H_1 for $v < 0$; H_2 otherwise. If $P_1 = 0.9$ and $P_2 = 0.1$, we choose H_1 for $v < 4$ volts.

Example 4.2.2

A communications system employs 4 symbols: $\sin \omega t$, $2 \cos \omega t$, $-2 \sin \omega t$, and $-\cos \omega t$; they have *a priori* probabilities of 0.2, 0.2, 0.3, and 0.3, respectively. They last 1 second, and have additive Gaussian white noise of $N_0 = kT_{\text{sys}}/2 \text{ WHZ}^{-1}$; $\omega/2\pi = 10 \text{ MHz}$. Design an optimum receiver (see Figure 4.2-4b for the case of two hypotheses) for estimating H_1 , H_2 , H_3 , or H_4 , and evaluate any bias terms.

Solution:

Using (4.2.18), the matched filter impulse responses are $h_i(t) = s_i(1-t) = a(t)s_i(1-t)$ where $a(t) = 1$ for $0 < t < 1$, and zero otherwise. The biases b_i are:

$$b_i = N \ln P_i - \frac{\bar{S}_i \cdot \bar{S}_i}{2}$$

$$b_1 = N \ln 0.2 - 1/4, \quad b_2 = N \ln 0.2 - 1, \quad b_3 = N \ln 0.3 - 1, \quad b_4 = N \ln 0.3 - 1/4$$

The noise variance $N = N_0 B = \sigma_n^2$. The filter acting on $kT_{\text{sys}}/2 \text{ WHZ}^{-1}$ (double-sided spectrum) is formed by $h_i(t)$. Its output noise is given by

$$N = \int_{-\infty}^{\infty} \frac{kT_{\text{sys}}}{2} |H(f)|^2 df = kT_{\text{sys}}/4$$

where $\int_{-\infty}^{\infty} |H(f)|^2 df = \int_{-\infty}^{\infty} h^2(t) dt = 1/2$ for S_1 and S_4 by Parseval's theorem.

Each second the decision circuit selects the most probable message i .

4.3 SIGNAL DESIGN

Although Section 4.2 described how to design optimum receivers that minimize the probability of error in an M-ary communication system, the problem of signal design remains. We wish to design the possible set of transmitted signals $\bar{S}_1, \bar{S}_2, \dots$, so as to minimize the probability of error P_e , which is given by (4.2.1) for the binary case. Since a trivial solution to this problem is to employ arbitrarily large signals compared to the noise, we must operate within a constraint such as a fixed average signal energy. If the units of \bar{S}_1 are volts, then we might choose to limit $\sum_i |\bar{S}_i|^2 P_i$; this is proportional to the average signal energy. Consider the simple

case of a binary communication system where the two possible signals \bar{S}_1 and \bar{S}_2 each consist of three samples, and therefore can be represented in three-dimensional space. The noise vector \bar{n} , which is superimposed by the channel, adds to the received signal vector to give a total received signal \bar{v} . This is graphically represented in Figure 4.3-1.

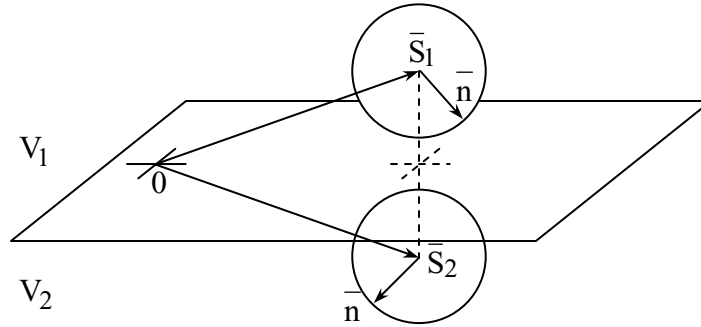


Figure 4.3-1: A pair of 3-sample signals.

Since the constraint on the average energy of the signals effectively limits the length of the vectors \bar{S}_1 and \bar{S}_2 in Figure 4.3-1, the design of the signals consists of determining their direction so as to minimize the expected probability of error P_e . Since the additive noise \bar{n} is uncorrelated with the signals, the probability distributions for the total received signal $\bar{v}_i = \bar{S}_i + \bar{n}$ are spherically symmetric and centered on \bar{S}_1 and \bar{S}_2 , as suggested in Figure 4.3-1. Since errors only occur when the noise vector \bar{n} causes the received signal \bar{v}_i to fall on the wrong side of the decision boundary separating V_1 and V_2 , the probability of error is clearly minimized by choosing the signal so that $\bar{S}_2 = -\bar{S}_1$.

In Figure 4.3-2 two alternative signal designs are compared, both of which have equal probabilities of error P_e . In each case $|\bar{S}_1 - \bar{S}_2| = 2$, but the case where $\bar{S}_1 = -\bar{S}_2$ requires half the average signal energy and power that is required when $\bar{S}_2 = 0$, assuming $P_1 = P_2$.

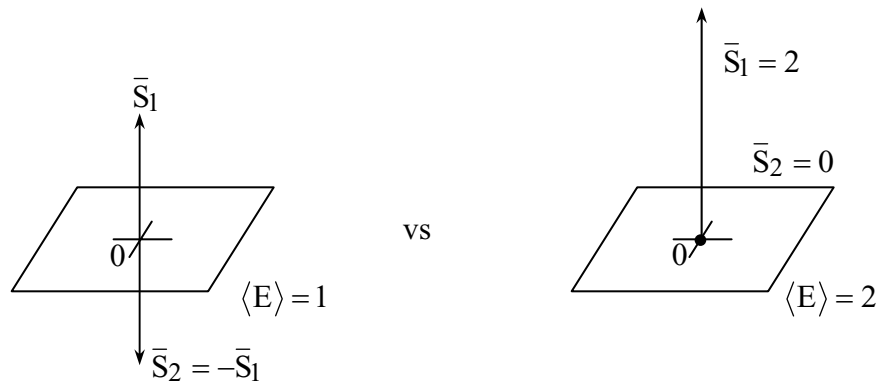


Figure 4.3-2: Binary signal design alternatives.

Figure 4.3-3 illustrates how a set of four possible signals (S_1 , S_2 , S_3 , and S_4) might be designed for two- and three- dimensional spaces.

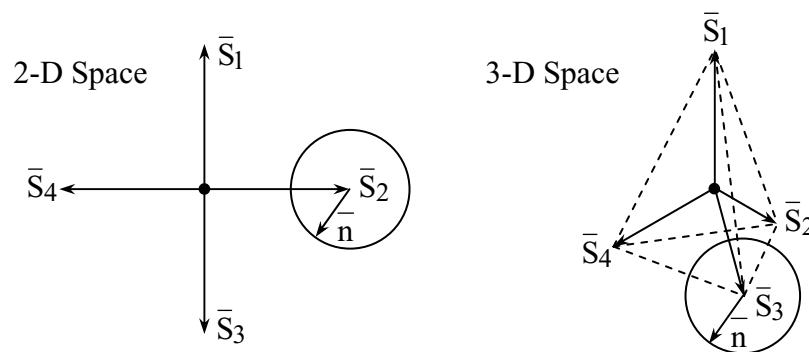


Figure 4.3-3: Quarternary signal design alternatives.

The signals in Figure 4.3-3 can be represented as two or three amplitudes for successive samples of the transmitted signal, i.e. in a 2-D or 3-D space. Often the transmitted signals are sinusoids that can also be characterized in terms of magnitude and phase, or in terms of the amplitudes of a real and an imaginary part. One possible arrangement for an ensemble of sixteen alternative sinusoidal transmitted signals is suggested in Figure 4.3-4a where their real and imaginary parts form a two-dimensional array of sixteen points arranged in a four-by-four grid. If the noise probability spheres for the received signal form circles about each of these points, as normally occurs¹, then it is clear we could slightly improve the relationship between average signal power and probability of error in two different ways. First, we could omit those possibilities that use more power, where the energy required for the symbol \bar{S}_1 is proportional to

¹ Spherically symmetric distributions in n dimensions form if the variance of the noise added in each dimension (e.g., to each sample value) is equal and the noise is uncorrelated with other noise samples and with the signal.

its magnitude squared. Thus, by rounding off the corners of the constellation illustrated in Figure 4.3-4a, we arrive at the slightly more efficient constellation illustrated in Figure 4.3-4b. Still further improvement can be obtained by using a hexagonal grid, as suggested in Figure 4.3-4c. These three constellations contain sixteen, twelve, and twenty-one possible symbols, respectively.

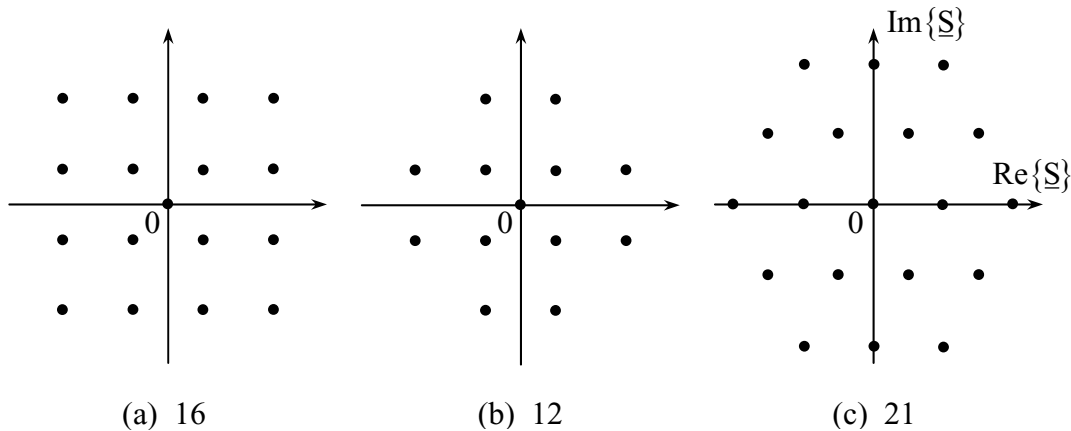


Figure 4.3-4: Phasor constellations.

The most general case does not employ sinusoids, but uses arbitrary waveforms instead, characterized by many samples. Although the n -dimensional sphere packing optimization problem is generally unsolved, various trial-and-error and hill-climbing techniques can be used to arrive at an optimum set. In practice this is increasingly done for large-order constellations, although simple ensembles of sinusoids of various magnitudes and phases are often used instead.

Example 4.3.1

A synchronized set of three two-voltage symbols is used to telemeter the state of a switch over a noisy channel. One symbol consists of a +1-volt signal for 1 second followed by a -1-volt signal for 1 second. What two other two-voltage symbols (one second each) would complete this trio of equal-energy symbols with minimum average P_e ?

Solution:

The trio of symbols S_i should be as nearly opposite as possible. In two dimensions S_1 , S_2 , and S_3 can be readily plotted. The energy in each symbol equals $a^2 + b^2 = 2$, so the vector length is $\sqrt{2}$ for each. Assign S_1 to the vector $(1 - j)$, which is at -45° . Therefore S_2 is $[a_2, b_2]$ volts, where $a_2 = \sqrt{2} \cos 75^\circ = 0.37$, $b_2 = j\sqrt{2} \sin 75^\circ = 1.37j$, and $S_3 = [a_3, b_3]$ where $a_3 = \sqrt{2} \cos 15^\circ = -1.37$, $b_3 = -j\sqrt{2} \sin 15^\circ = -0.37j$.

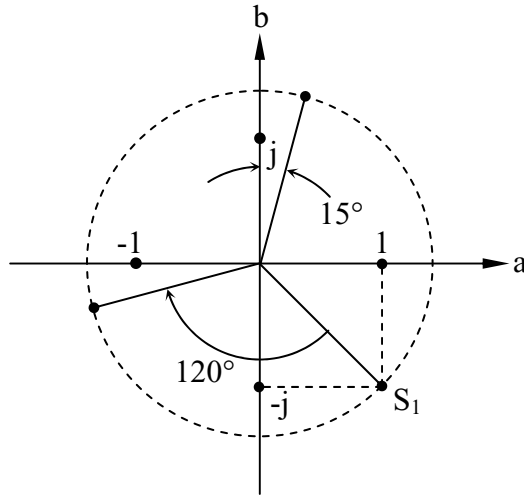


Figure 4.3-5: Vector symbol set

4.4 PERFORMANCE OF DIGITAL COMMUNICATION SYSTEMS

The performance of a digital communication system can be characterized by its probability of message error for a given signal power and noise environment. Another important performance parameter is the bandwidth required by the system. In some cases this available bandwidth is limited by law or physics. In other cases it is limited by economics, and bandwidth may be bought and sold as a commodity. Telephone companies, cable television companies, and over-the-air broadcasters all assign economic value to their available bandwidth. Here we consider first the relationship between error probabilities and signal power, and later we consider issues of spectrum utilization.

Perhaps the simplest communication system involves binary communications over a linear channel perturbed by additive Gaussian white noise of $N_o/2$ (watts/Hz) for the double-sided spectrum representation (employing positive and negative frequencies). For a nominal one-ohm resistive load and signal bandwidth B Hertz, the total noise power is $N_o B = E[n^2(t)]$, which we define as N .

The optimum decision rule for a binary signal (4.2.17) was to choose H_1 if

$$\bar{v} \cdot (\bar{S}_1 - \bar{S}_2) > \frac{|\bar{S}_1|^2 - |\bar{S}_2|^2}{2} + N \ln(P_2/P_1) \quad (4.4.1)$$

$$\begin{aligned}\sigma_y^2 &\equiv E[y^2] = E\left\{\left[\frac{1}{2B} \sum_{j=1}^{2BT} n_j (S_{1j} - S_{2j})\right]^2\right\} \\ &= \left(\frac{1}{2B}\right)^2 E\left\{\sum_{i=1}^{2BT} \sum_{j=1}^{2BT} n_i n_j (S_{1i} - S_{2i})(S_{1j} - S_{2j})\right\}\end{aligned}\quad (4.4.5)$$

Since

$$E[n_i n_j] = N\delta_{ij} \quad (4.4.6)$$

it follows that (4.4.5) becomes simply

$$\sigma_y^2 = \left(\frac{1}{2B}\right)^2 N |\bar{S}_1 - \bar{S}_2|^2 = \frac{N_0}{2} \int_0^T [S_1(t) - S_2(t)]^2 dt \quad (4.4.7)$$

Therefore, the probability of error $P_e|_{S_1}$ becomes simply the integrated tail of the Gaussian distribution:

$$P_e|_{S_1} = \int_{-\infty}^{-b} \frac{1}{\sqrt{2\pi\sigma_y^2}} e^{-y^2/2\sigma_y^2} dy \quad (4.4.8)$$

as suggested in Figure 4.4.1a.

This integrated area under the tail of a Gaussian distribution is so commonly used that it has been defined in terms of a function $\text{ERF}(A)$, which is tabulated in reference books because it has no simple closed-form expression. This *error function* $\text{ERF}(A)$, for historical reasons, was defined not as the tail of the Gaussian distribution but as the complementary core, as suggested in Figure 4.4.1b. Its algebraic definition is

$$\text{ERF}(A) \equiv \frac{1}{\sqrt{\pi}} \int_{-A}^A e^{-x^2} dx \quad (4.4.9)$$

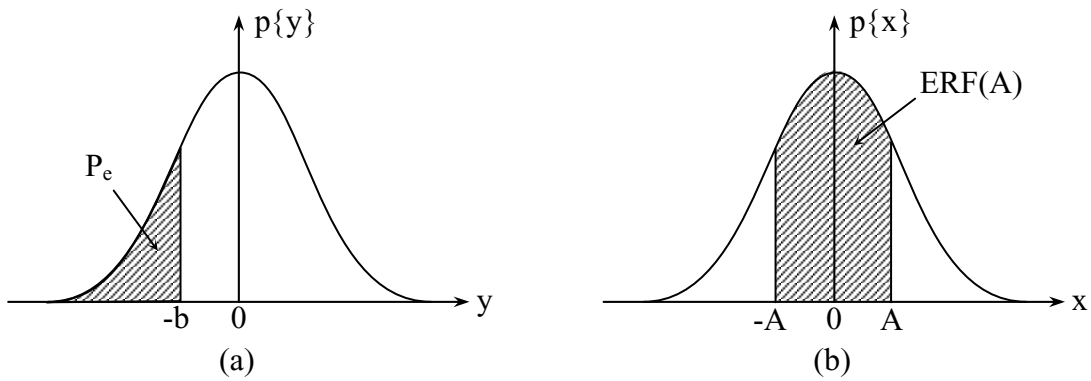


Figure 4.4.1 a) P_e is the area in one tail of a Gaussian distribution
 b) Area corresponding to $\text{ERF}(A)$

The tail of the distribution is given by the complementary error function

$$\text{ERFC}(A) \equiv 1 - \text{ERF}(A) \quad (4.4.10)$$

To relate (4.4.8) to (4.4.9) we note

$$\text{ERF}(K/\sqrt{2}) = \frac{1}{\sqrt{\pi}} \int_{-K/\sqrt{2}}^{K/\sqrt{2}} e^{-x^2} dx \quad (4.4.11)$$

$$\text{ERF}(K/\sqrt{2}) = \int_{-K\sigma_y}^{K\sigma_y} \frac{1}{\sqrt{2\pi\sigma_y^2}} e^{-y^2/2\sigma_y^2} dy \quad (4.4.12)$$

where the integrand of (4.4.12) was made identical to the integrand of (4.4.8) by letting $x^2 = y^2/2\sigma_y^2$, which implies $x = y/\sigma_y\sqrt{2}$ and $dx = dy/\sqrt{2\sigma_y^2}$. As a result, when $x = K/\sqrt{2}$, then $y = K\sigma_y = b$. Therefore $K/\sqrt{2} = b/\sigma_y\sqrt{2}$ and (4.4.8) becomes

$$P_e|_{S_1} = \frac{1}{2} \text{ERFC}(b/\sigma_y\sqrt{2}) \quad (4.4.13)$$

with the help of (4.4.10) and (4.4.12).

The total probability of error is the probability-weighted sum of the probability of error if S_1 is sent and the probability of error if S_2 is sent:

$$P_e = P_1 P_e|_{S_1} + P_2 P_e|_{S_2} \quad (4.4.14)$$

If the probabilities of symbols 1 and 2 equal 0.5, then

$$P_e = \frac{1}{2} \text{ERFC}(b/\sqrt{2}\sigma_y) \quad (4.4.15)$$

Using the expressions for b (4.4.4) and σ_y (4.4.7) we find

$$P_e = \frac{1}{2} \text{ERFC} \left[\frac{\frac{1}{4B} |\bar{S}_1 - \bar{S}_2|^2 - \frac{N}{2B} \ln(P_2/P_1)}{\frac{\sqrt{2N}}{2B} |\bar{S}_1 - \bar{S}_2|} \right] \quad (4.4.16)$$

which can be simplified to

$$P_e = \frac{1}{2} \text{ERFC} \left[\frac{1}{\sqrt{8N}} |\bar{S}_1 - \bar{S}_2| \right] \quad (4.4.17)$$

by noting that if $P_1 = P_2$, then $\ln(P_2/P_1) = 0$.

Not only is this expression useful because it yields the probability of error in a simple form, but it also suggests immediately how we might choose the signals \bar{S}_1 and \bar{S}_2 so as to minimize P_e . That is, if we choose two symbols with equal energy, so that $|\bar{S}_1| = |\bar{S}_2|$, then clearly P_e is minimized if

$$|\bar{S}_1| = |\bar{S}_2| \Rightarrow \bar{S}_2 = -\bar{S}_1 \quad (4.4.18)$$

That is, to minimize the probability of error for equal-energy and equal-probability symbols, one symbol should be the negative of the other.

The optimum receiver for continuous signals $S(t)$ and noise $N(t)$ can be simply found by evaluating (4.4.2) in terms of its continuous-signal equivalent. We assume the noise is white but band-limited to B Hertz as suggested in Figure 4.4-2, with a double-sided spectrum noise power density of $N_o/2$ [WHz⁻¹]. Thus the average total noise power is $N_o B$, and $E[n^2(t)] = N_o B = N$.

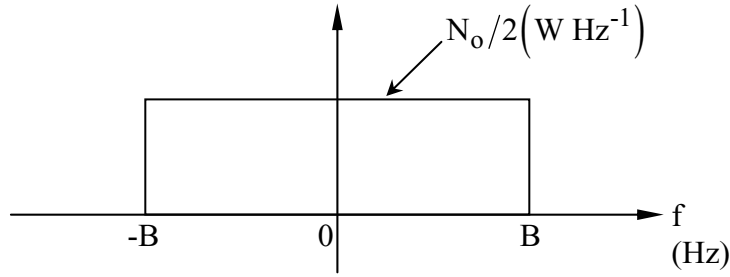


Figure 4.4-2: Noise power spectrum, band limited.

Thus to evaluate (4.4.2) in the continuous-signal limit, we define the Gaussian random variable y of zero mean as

$$y \equiv \int_0^T n(t)[S_1(t) - S_2(t)] dt \quad (4.4.19)$$

which is analogous to (4.4.3). If the symbols $S_1(t)$ and $S_2(t)$ each last T seconds, then their sampled versions would contain $2BT$ samples. Because the noise signal $\bar{n}(t)$ is sampled at the Nyquist rate ($2B$ samples per second), the noise samples in the vector \bar{n} are independent and identically distributed, consistent with the assumptions we made earlier; thus the continuous-signal case and the sampled-signal case are exactly analogous. Similarly we define

$$b \equiv \frac{1}{2} \int_0^T [S_1(t) - S_2(t)]^2 dt - \frac{N_o}{2} (P_2/P_1) \quad (4.4.20)$$

analogous to (4.4.4). The variance of y then becomes

$$\sigma_y^2 = \left(\frac{1}{2B} \right)^2 N |\bar{S}_1 - \bar{S}_2|^2 = \frac{N_o}{2} \int_0^T [S_1(t) - S_2(t)]^2 dt \quad (4.4.21)$$

analogous to (4.4.7).

Thus, in this continuous-signal case, the expression for probability of error given by (4.4.2) reduces to

$$P_e = \frac{1}{2} \text{ERFC} \left[\frac{1}{2} \sqrt{\frac{\int_0^T [S_1(t) - S_2(t)]^2 dt}{N_o}} \right] \quad (4.4.22)$$

analogous to (4.4.17). In this case also, the probability of error is minimized for equal-energy and equal-probability symbols if $S_2(t) = -S_1(t)$.

Using these expressions for P_e given by (4.4.17) and (4.4.22), several simple communication systems can be evaluated.

Perhaps the simplest binary communication system is *on-off keying*, which is abbreviated “OOK”. In this case a pure tone is either turned on or turned off so the signal $S_1(t)$ is $A \cos \omega_0 t$, while $S_2(t)$ is simply zero. These definitions for the two symbols $S_1(t)$ and $S_2(t)$ are given in Table 4.4.1.

Table 4.4.1: Definition of modulation types and P_e

Abbreviation	Modulation Type	$S_1(t)$	$S_2(t)$	P_e
OOK	On-off keying	$A \cos \omega_0 t$	0	$0.5 \operatorname{ERFC} \sqrt{E/4N_0}$
FSK	Frequency-shift keying	$A \cos \omega_1 t$	$A \cos \omega_2 t$	$0.5 \operatorname{ERFC} \sqrt{E/2N_0}$
PSK	Phase-shift keying	$A \cos \omega t$	$-A \cos \omega t$	$0.5 \operatorname{ERFC} \sqrt{E/N_0}$

Two other common types of signaling are called *frequency-shifting keying*, “FSK”, and *phase-shift keying*, “PSK”, for which the waveforms are also presented in Table 4.4.1. *Binary frequency-shift keying (BFSK)* involves two sinusoidal symbols at different frequencies, whereas binary phase-shift (BFSK) keying involves sinusoids at the same frequency, but one is 180 degrees out of phase with the other, and therefore each is the negative of the other.

The expressions for error probability P_e are readily evaluated for each of these binary communications systems using (4.4.22), and these expressions are contained in the last column of Table 4.4.1. Note that the resulting expressions for error are extremely simple, and differ only slightly. Each involves the complementary error function and the ratio E/N_0 , where E is the energy in each symbol (or in $S_1(t)$ for OOK).

$$E = \int_0^T S_1^2(t) dt = |\bar{S}_1 \cdot \bar{S}_1| \quad (4.4.23)$$

Thus P_e is a simple function of the ratio of average symbol energy E (Joules) and the noise power spectral density N_0 (Watts/Hertz = Joules). Therefore the cost of communicating over a given channel can be expressed as the cost of energy, where a certain level of performance requires a certain received energy (Joules) per bit. The required transmitter power is therefore generally proportional to the desired symbol or bit rate, and communications systems employing very low bit rates can successfully use very low power transmitters and very small antennas.

The expressions for P_e given by Table 4.4.1 are also plotted approximately in Figure 4.4-3 as a function of the ratio E/N_o (dB). The best performance is given, as might be expected from (4.4.18), by binary phase-shift keying, “BPSK”. To achieve the same P_e using frequency-shift keying (FSK), twice the energy is required. This form of FSK is sometimes called *coherent FSK*, where we have assumed up until now that the receiver knows exactly the two possible symbol waveforms, including the phase of any sinusoids. Sometimes receivers are built where such phase coherence between the transmitter and receiver cannot be maintained, and the receiver must therefore test the hypothesis $S_i(t)$ for various phases, which slightly increases P_e unless somewhat more energy is used, as suggested by the curve labeled *non-coherent FSK*. The worst performance is obtained for on-off keying (OOK), which again is assumed to be coherent. Incoherent OOK would require still more energy.

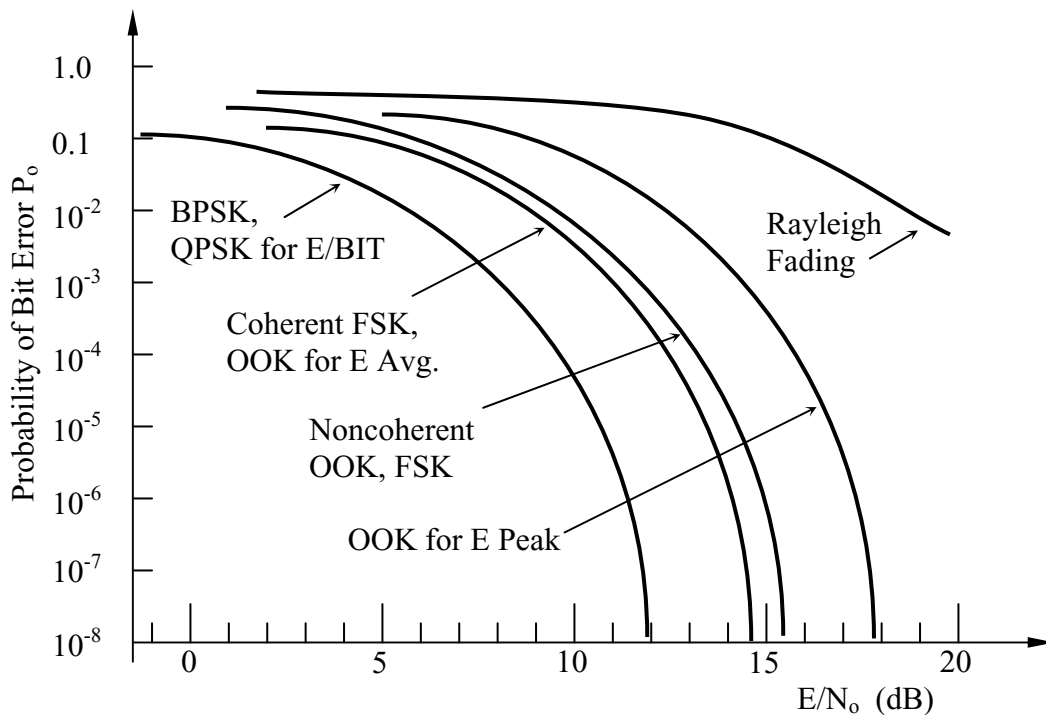


Figure 4.4-3: Approximate error probabilities versus E/N_o

Receivers for which the possible transmitted signals are known exactly, and that compute the overlap integrals between the received signals and the reference waveforms are called *matched-filter receivers*, and are generally implemented by passing the received signal through various linear filters with impulse responses $h_i(t)$ that are time-inverted versions of the corresponding possible transmitted symbols $S_i(t)$, as suggested earlier in Figure 4.2-4b.

More than two possible symbols can be transmitted. For simplicity, most such systems employ sinusoids of different amplitudes and phases. One variation of this is called *multi-phase-shift keying*, “MPSK”. The most common variations of MPSK are BPSK, *QPSK*, and *8PSK*, corresponding to two, four, and eight phases, respectively, at the same frequency and amplitude.

Each symbol therefore conveys one, two, and three bits of information, respectively, where the number of bits per symbol $L = \log_2 M$. Each such symbol is sometimes called a *baud*, where the number of bauds per second is equal to, or less than, the number of bits per second. Figure 4.4-4 shows the probability of bit error as a function of E/N_0 for various MPSK systems. It illustrates how the average energy per baud must increase as the number of bits per baud increases. Note that the required energy per bit for a given P_e is relatively small as we move from $M = 2$ to $M = 4$, but begins to increase substantially as M grows beyond 16. The principal incentive for increasing the number of bits per baud is to reduce the bandwidth required per bit/second.

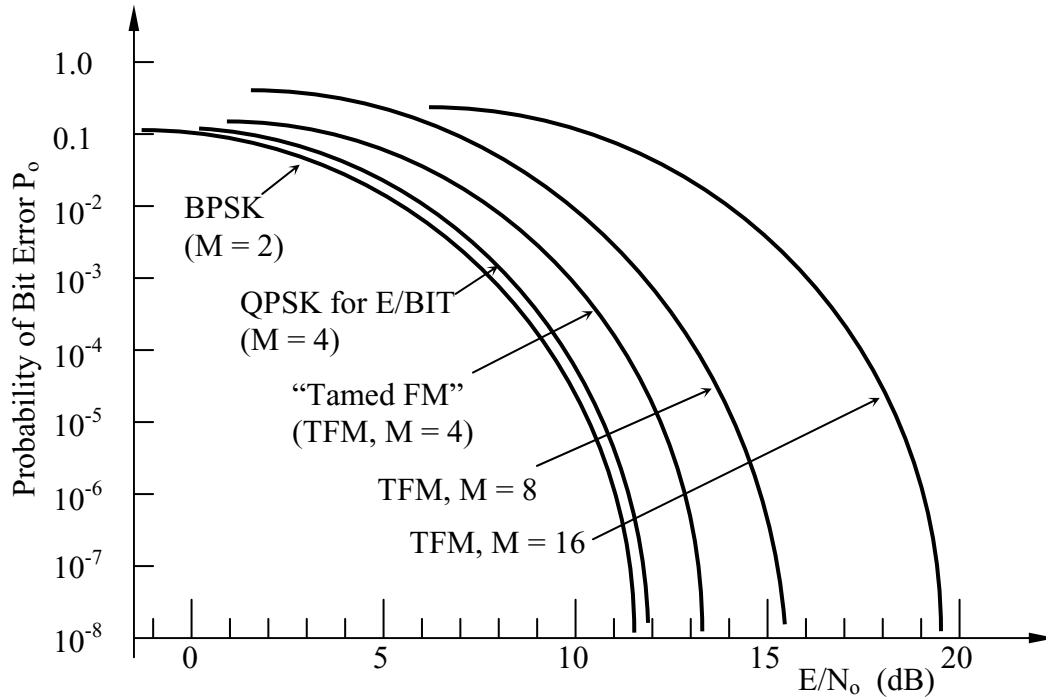


Figure 4.4-4: Approximate bit-error probabilities versus E/N_0 for MPSK modulation

Figure 4.4-5a shows a typical symbol pair for a BPSK system, and Figure 4.4-5b illustrates a typical set for QPSK. We may estimate the spectral distribution of the transmitted signal by Fourier transforming a random sequence of such bauds. Note that the symbols can involve abrupt steps at the end of the baud, therefore yielding more spectral sidelobes than BPSK.

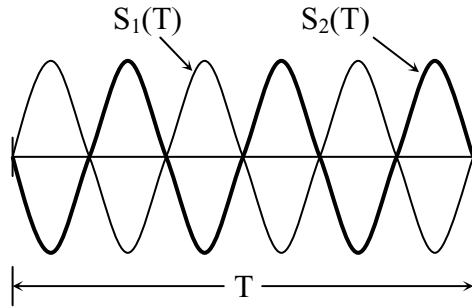


Figure 4.4-5a: BPSK symbols.

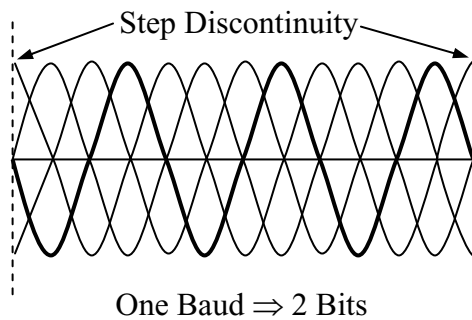


Figure 4.4-5b: QPSK symbols.

We can now consider the required bandwidth per bit/second; this equals $B(\text{Hertz})/R(\text{bits/second})$. Figure 4.4-6 illustrates the improved spectral efficiency that results as M increases for MPSK systems. The required bandwidth per bit is illustrated for coherent MPSK with a baud error rate (BER) of 10^{-7} . Note that the bandwidth required per bit drops by a factor of two as M increases from two to four, but the corresponding increase required in baud energy is very small, less than 2 dB. Although additional increases are available in B/R , they require increasingly expensive increases in E/N_0 . This graph suggests the primary reason QPSK ($M = 4$) is so popular, it reduces bandwidth without sacrificing much power.

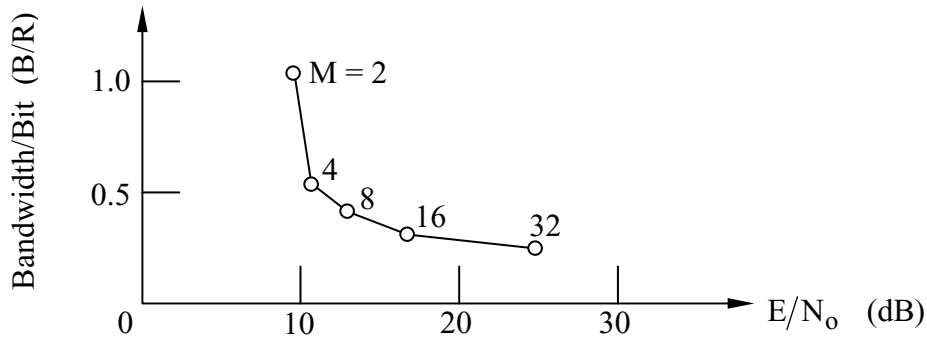


Figure 4.4-6: Approximate bandwidth per bit for MPSK versus E/N_0 .

The reason MPSK is not commonly used for $M > 8$ is suggested in Figure 4.4-7. In Figure 4.4-7a four possible phasors are presented corresponding to the four possible symbols. The dotted lines bound the decision regions, the one corresponding to S_1 being cross-hatched. Since the bounds of V_1 corresponding to S_1 are nearly as far from S_1 as would be the boundaries for BPSK, the required signal power is comparable. The corresponding diagram for 16-PSK is suggested in Figure 4.4-7b, where the 16 possible symbols correspond to phasors arranged in a circle. Now the cross-hatched region corresponding to S_1 assumes an inefficient shape vulnerable to noise. A better arrangement of 16 symbols in phase space is suggested in Figure 4.4-7c, where the same average energy per symbol spaces the phasors farther apart, reducing P_e . In this case the phasors differ in both phase and amplitude and are called 16-QAM for Quadrature Amplitude Modulation.

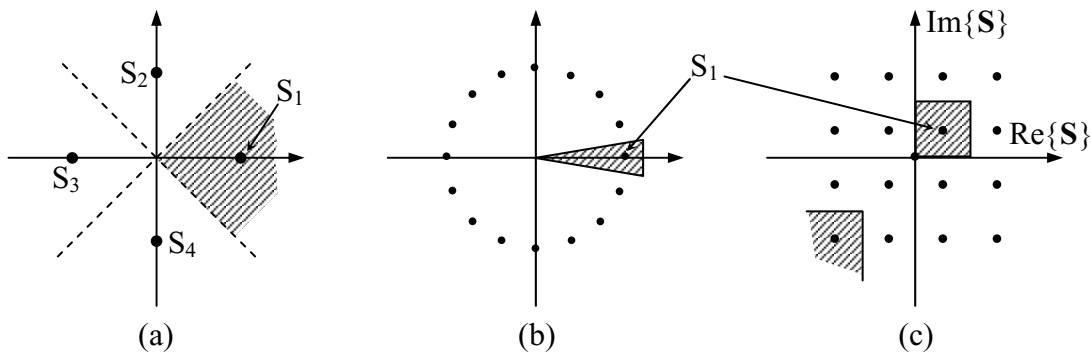


Figure 4.4-7: Phasor diagrams for (a) QPSK, (b) 16-PSK, and (c) 16 phase-amplitude modulation.

Another way to reduce spectral sidelobes and improve spectrum efficiency is to window the set of possible sinusoids with other than a boxcar envelope, as suggested in Figure 4.4-8. Figure 4.4-8d illustrates how a boxcar envelope produces a narrow spectral feature at the expense of higher sidelobes in the spectral domain, where the energy spectral density is proportional to

$\left[\frac{\sin^2(f-f_0)}{(f-f_0)^2}\right]$. The cosine envelope employed in Figure 4.4-8b yields somewhat broader spectral features but with reduced sidelobes.

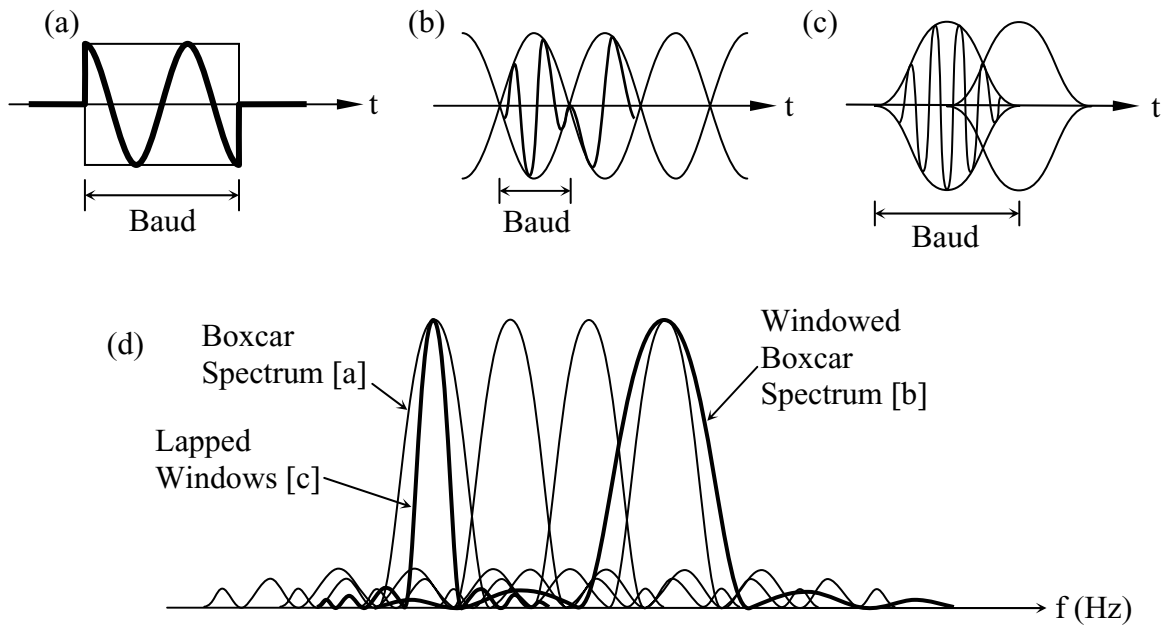


Figure 4.4-8: (a) MPSK with a boxcar envelope, (b) MPSK with a windowed boxcar envelope, (c) MPSK with overlapped windowed boxcar envelopes, (d) spectral widths of windowed and lapped sinusoids.

Both narrower mainlobes and lower sidelobes can be obtained simultaneously if the window functions overlap, as in Figure 4.4-8(c). In this case the symbol set should be chosen so as to be orthogonal not only within each window, but also all possible combinations of symbols within adjacent windows should be orthogonal. That is, $\bar{S}_{i,k} \cdot \bar{S}_{j,k+1} = 0$ for all possible symbol pairs $\bar{S}_{i,j}$ that can occur in adjacent time slots $k, k+1$. Powerful digital signal processing (DSP) circuits make real-time generation and reception of such symbol sets relatively straightforward. The spectral benefits of using lapped windows are suggested in Figure 4.4-8(d).

One advantage of designing transmission signals with higher spectral efficiency and low sidelobes is that many adjacent channels can be packed side by side with little cross-talk penalty. One example of this is illustrated in Figure 4.4-9, which presents the possible degradation that results for various channel spacings Δf (Hz). The signal degradation is the degree to which the transmitter power must be increased to yield the same probability of error that would have been achieved for an isolated channel. The horizontal axis of the figure is the dimensionless ratio $\Delta f/R'$, where R' represents the channel baud rate in bauds (or symbols) per second. The signal degradation is shown for cases where $N = 1$ interfering channel and $N = \infty$, corresponding to a channel imbedded in an infinite number of equally spaced channels. Note that the signal

degradation increases as the channels are spaced more closely together. The figure also shows the relative performance for QPSK modulation with boxcar envelopes, and alternatively for one of several possible QPSK-like overlapping window schemes, called “tamed FM”. The reason one can overcome the effects of sidelobe interference by boosting signal power is that the error probability is controlled by the combination of thermal noise in the receiver plus the coherent non-random effects of spectral sidelobes from adjacent channels. Clearly, if the thermal noise $N_0 = 0$, boosting signal power would yield no improvement because the transmitted signal sidelobes could be increased proportionately. Thus boosting transmitter power works only to the point where the thermal noise becomes negligible in comparison.

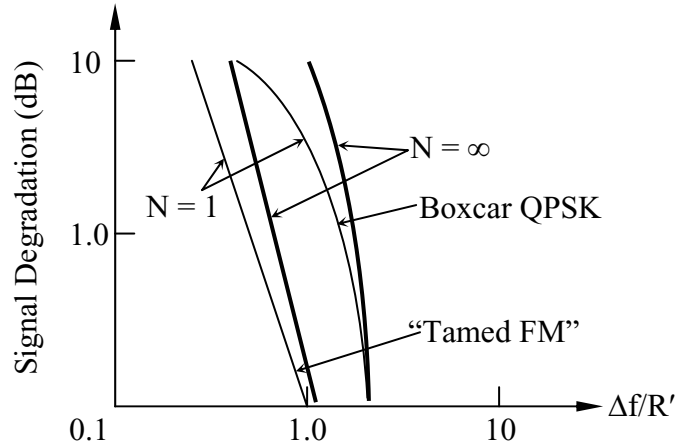


Figure 4.4-9: Approximate signal degradation versus channel separation.

Alternatively, we may fix the ratio E/N_0 and let the bit error rate (BER) vary as we change $\Delta f/R$. One such trade-off is suggested in Figure 4.4-10 for tamed FM for the cases $N = 1$ and $N = \infty$. We therefore typically use $\Delta f/R$ values greater than 0.6. For QPSK designed for use in a multichannel system, we prefer values of $\Delta f/R' > 1.5$. Since QPSK yields two bits per baud, this corresponds to ratios of Δf to bit rate R greater than 0.75.

So far we have considered only channels that are not subject to random fading. Fading can be introduced when absorbing or reflecting objects like trucks or clouds move across a transmission path, or when multiple beams interfere at a receiving antenna after moving along trajectories for which the length varies randomly due to changes in refractive index, the motion of reflectors, or the influence of other varying inhomogeneities. For any fixed transmitter power, each fade will increase the probability of bit error substantially, and this increase must be averaged over all possible fades in accord with their probability. The net effect is to increase P_e for any given value of E/N_0 . Figure 4.4-3 shows the increase in P_e for Rayleigh fading and nonfading channels. These errors can be reduced by a variety of signal processing techniques that are discussed in the next section.

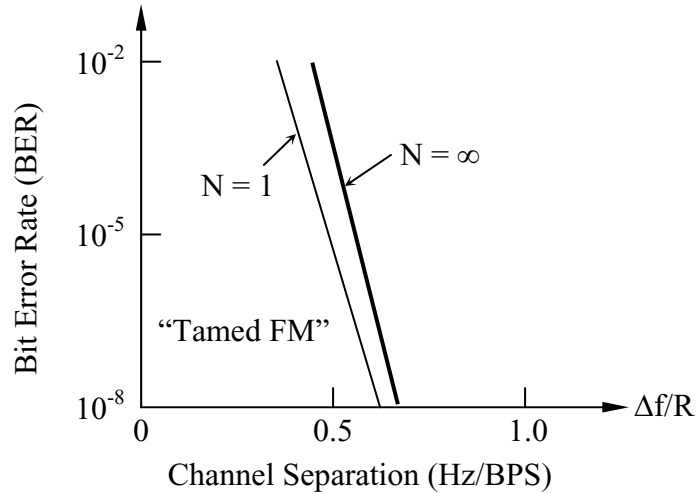


Figure 4.4-10: Bit-error-rate versus channel separation.

Example 4.4.1

The Gaussian white noise in a receiver is $10^{-15} \text{ WHz}^{-1}$. What is the maximum data rate R (bits/sec) which can be communicated via a 10^{-10} watt signal at $P_e = 10^{-6}$ for coherent binary phase shift keying? What is R for QPSK?

Solution:

Figure 4.4-4 suggests that E/N_o must be ~ 10.5 dB for BPSK. Therefore $E \cong 11.2N_o = 1.12 \times 10^{-14}$ Joules per symbol and per bit. $R(\text{bits/sec}) = P(\text{watts})/E(\text{J/bit}) = 10^{-10}/1.12 \times 10^{-14} \cong 8.9$ kbps. For QPSK $E/N_o \cong 11$ dB and $R(\text{bits/sec}) = 2(\text{bits/symbol}) \times P/E(\text{J/symbol}) = 2 \times 10^{-10}/1.26 \times 10^{-14} \cong 15.9$ kbps.

4.5 CHANNEL CODING AND CAPACITY

Errors in communications systems introduced by noise, fading, or other means are undesirable and sometimes unacceptable. Fortunately, signal processing and coding can reduce P_e without altering channel noise or transmitter power. More precisely, Claude Shannon proved that, in principle, the probability of error P_e can be made to approach zero if the *channel capacity* C (bits/second) is not exceeded, where

$$C = B \log_2(1 + S/N) \quad (4.5.1)$$

and where B is the channel bandwidth in Hertz, S is the average signal power, and N is the noise power, where $N = N_o B$. For example, (4.5.1) suggests that a channel might convey 3 bits/Hz if $S/N = 10$, and convey 7 bits/Hz if $S/N = 127$ (~ 21 dB).

Although Shannon did not present a simple way to achieve this channel capacity, many techniques have been devised to approach these limits with varying degrees of complexity and delay. Numerous journals and texts have discussed these various methods, and we shall only present a few of the relevant conclusions here.

For example, one approach suggests the important role of time delay introduced by coding. It can be shown that for some coding techniques

$$P_e \leq 2^{-Tk(C,R)} \quad (4.5.2)$$

where T is the time delay in the coding process, and $k(C,R)$ is some function of the channel capacity C and desired data rate R in bits per second. One coding technique characterized by (4.5.2) is the following. Let the number of possible symbols of length T seconds be $M = 2^{RT}$, where RT is the number of bits of information communicated in T seconds. We choose as our symbol set $M = 2^{RT}$ frequencies spaced at $\sim 1/T$ Hz, consistent with the spectral width of a sinusoid lasting T seconds. The nominal total bandwidth occupied by this communication system is therefore the number of possible frequencies times the bandwidth of each. That is, $B = 2^{RT}/T$, which we can, in principle, let approach infinity.

Using this approach, the channel capacity with infinite bandwidth C_∞ , which is defined as the limit of C as B approaches infinity (see (4.5.1)), is given by

$$C_\infty = S/(N_o \ln 2) \geq R_{(P_e \rightarrow 0)} \quad (4.5.3)$$

which implies that the data rate R cannot exceed $S/(N_o \ln 2)$ if zero error probability is to be approached. Therefore,

$$S/N_o R \geq \ln 2 = 0.69 \quad (P_e \rightarrow 0) \quad (4.5.4)$$

The limit given by (4.5.4) is presented in Figure 4.5-1, together with the more familiar relationship between P_e and E/N_o for $M = 2$, which is BFSK. Note that the relation $P_e(E/N_o)$ for $M=10^6$ is not enormously different. Figure 4.5-1 simply represents the performance for the classical MFSK communication system discussed in Section 4.4. Here we see for the first time the benefits of letting M approach infinity, which are achieved at the expense of longer delays because T must increase to accommodate the extremely large numbers of possible frequencies. For practical reasons M typically is less than 100 so the ability of this approach to produce small values P_e are limited, as suggested by the figure. Note, however, that the time delay penalties are modest because $M = 2^{RT}$, where T appears in the exponent.

Two important variations of this technique are phase modulation (PM) and frequency modulation (FM). In each case the bandwidth employed for communications is expanded beyond the original bandwidth of the signal so that the probability of error and the effects of channel noise can be effectively reduced. Such analog systems are discussed later in Section 4.7.

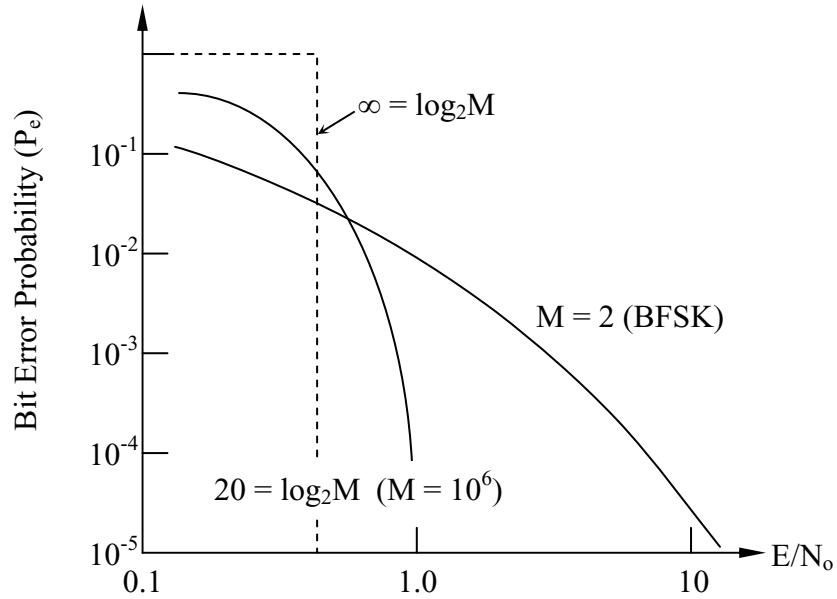


Figure 4.5-1: Approximate error probability versus E/N_0 and M .

A third approach leading to a relatively simple expression for P_e in terms of C , R , and T employs a noiseless analog feedback channel by which the receiver can tell the transmitter what has been received. The transmitter can then compensate for the previous channel errors in the next transmission. The system in Figure 4.5-2 communicates R bits per second in a block lasting T seconds by communicating a single analog voltage accurate to one part in M where $M = 2^{RT}$ and RT is the number of bits sent per block.

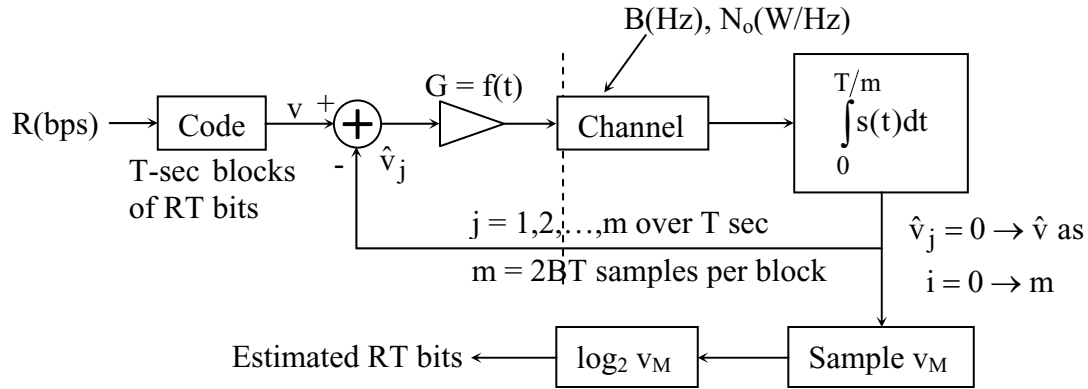


Figure 4.5-2: Noise-reducing analog feedback communications system.

The system communicates this precise analog voltage A by sending a succession of M analog voltages during T seconds at the Nyquist rate, where $M = 2BT$. The noiseless feedback channel is continuously sending the receiver's best estimate of the desired voltage v , which the transmitter subtracts from the actual voltage to obtain a difference that the transmitter then communicates after amplifying it by some gain G . This gain is steadily increased over the duration of the transmission in a programmed way so as to maintain roughly constant transmitter power. Thus by a method of successive approximations, the receiver deduces the transmitted voltage v , which can then be converted into the desired message containing RT bits. These bits are deduced by the receiver simply by computing $\log_2 v_M$. It can be shown that P_e for this system, assuming an optimum strategy for increasing the gain G during each block is employed, is given as

$$P_e \cong e^{-(3/2) \left[2^{2(C-R)T} \right]} \ll 1 \quad (4.5.5)$$

Even in this ideal case of a noiseless feedback channel, the probability of error P_e approaches zero only as T approaches infinity. Fortunately, as we increase the channel rate R from negligible values to $C/2$, this delay T must increase only a factor of two. In this case, the delay problem becomes severe only as we try to approach the channel capacity itself.

A variation on the feedback-channel approach to reducing transmission errors involves requests for retransmission when the received signal is ambiguous due to noise or other effects. A wide variety of retransmission and successive approximation techniques have been devised but none of them permit the channel capacity to be exceeded.

With the exception of frequency modulation (FM), most schemes for approaching the channel capacity involve the use of digital *channel coding*. Channel codes are distinct from "source codes" which reduce redundancy in the transmitted signal, and "cryptographic" codes that are designed to conceal information from interpretation by unauthorized parties. One of the

most basic approaches to channel coding is described by a short poem attributed to Solomon Golomb,

“A message with content and clarity
has gotten to be quite a rarity,
to combat the terror
of serious error,
use bits of appropriate parity”

Many channel codes can be viewed as simple extensions of *parity-check codes* which have been widely used. One simple parity check code involves breaking any bit stream (0's and 1's) into eight-bit blocks containing seven message bits and one *parity bit*. The encoder chooses the value for the parity bit such that the total number of ones in the eight-bit block is an even number for an even parity code, of odd for an odd parity code. If a single bit error occurs in such a block, then the block is transformed to an “illegal” message, and the recipient is therefore warned that one of the seven message bits may be in error (the parity bit itself could also be the one at fault), and should be treated with caution. During periods where the probability of bit error rises substantially, the recipient of the message may instead choose to discard a series of blocks when even some of them indicate parity errors. Such simple *error-detection* codes are usually used in situations where errors are so rare that an occasional warning suffices. Usually the entire transaction is repeated or the message is retransmitted when such an error is detected. Such repetition implies a timely feedback path, however. These codes are used not only for transmission of data over short and long distances, but also for storing and recalling data for various storage media.

This simple parity check concept can be extended in several ways. For example, the parity bit may become a *check sum*, which is a multi-bit number designed to force the sum of several multi-bit numbers comprising the message to total one of several possible legal numbers. These additions can be arithmetic or confined to columns, which are summed modulo-2, or even modulo-n if binary numbers are not employed.

Perhaps the most useful extension of the parity check concept has been to more general error detection and correction codes. Consider a binary message composed of a long string of bits divided into blocks of fixed length. Assume each block consists of K message bits and R check bits. Conceptually, error correction can be accomplished if the number of possible illegal blocks is sufficiently large compared to the number of possible legal ones that errors of one or a few bits in any given block necessarily produces an illegal message sufficiently close to the original that the receiver can confidently guess it.

First consider how many check bits R are required to permit correction of N bits for a block of length $K + R$ bits. First, consider only received blocks that have zero or one bit in error, where this bit can be either a message or a check bit. The receiver must then make one of $K + R + 1$ choices. One possible choice is that the received message is perfect. The other $K + R$ choices correspond to changing the one message or check bit believed to be in error. The block

must be sufficiently redundant that the receiver is capable of making this decision. That is, the block must contain K message bits plus enough additional check bits to instruct the receiver as to its proper choice. Therefore, the number of check bits R must equal or exceed $\log_2(K + R + 1)$.

Perhaps the simplest example is a message consisting of $K = 1$ bit accompanied by $R = 2$ check bits. In this case $K + R + 1 = 4$ and $R = \log_2(K + R + 1)$. This code is not very efficient because the number of check bits R is a large fraction $R/(R + K) = 2/3$ of the total data stream. Table 4.5.1 shows how the communication efficiency improves for a single-bit correction code as the block size increases. Until each message block comprises at least a few bytes (a byte is 8 bits), such codes are relatively inefficient. The extremely high efficiencies implied by the table for long blocks are generally not employed in practice, however, because such long blocks are more subject to errors and perhaps to several errors, unless the probability of bit error P_e is extremely low.

Table 4.5.1 Number of required parity bits for single-error correction

K	R ≥	R/(R + K)
1	2	2/3
2	3	0.6
3	3	0.5
4	3	0.4
56	6	0.1
10^3	10	0.01
10^6	20	2×10^{-5}

The same logical approach permits simple calculation of the number of check bits required to correct two errors. In this case each block would contain the K message bits plus enough check bits to accommodate the increased number of choices faced by the receiver. The receiver must now decide whether there are 0, 1, or 2 errors present. If one error is present, it could occur in $K + R$ possible locations. Two errors could be arranged in any of $(K + R)(K + R + 1)/2$ locations. Therefore, in order for a block code to correct up to two errors, we need

$$K + R \geq K + \log_2(1 + K + R + (K + R)(K + R + 1)/2) \quad (4.5.6)$$

The number of check bits required in order to correct two errors, and the corresponding fraction of the bits devoted to this purpose, are suggested in Table 4.5.2.

Table 4.5.2 Parity bits required versus block size for double-error correction

K	R ≥	R/(R + K)
5	7	0.6
10^3	~20	0.02
10^6	~40	4×10^{-5}

The following example illustrates one way such an error-correction code might be implemented. Consider the special case of a single-error correction code where the number of message bits K is 4, implying $R = 3$ using Table 4.5.1. Note that this is an efficient combination in the sense that $R = \log_2(K + R + 1)$.

For this ($K = 4, R = 3$) code we might represent each block as $[m_1 \ m_2 \ m_3 \ m_4 \ c_1 \ c_2 \ c_3]$ where m_i represents the four message bits and c_j represents the three check bits. Each check bit is then related in a different way to the various message bits. For example, we might let

$$c_1 \equiv m_1 \oplus m_2 \oplus m_3 \quad (4.5.7)$$

where the symbol \oplus means *sum modulo-2*, and is defined so that the sum modulo-2 of two one-bit numbers is 0 if they are identical, and 1 if they are different (i.e., $0 \oplus 0 = 0$, $1 \oplus 1 = 0$, and $0 \oplus 1 = 1 \oplus 0 = 1$).

In similar fashion we can define c_2 and c_3 . Note that for each set of four message bits, one and only one set of check bits is legal; that is, only one-eighth of all seven-bit blocks are allowable.

For this example, we have

$$c_1 \oplus c_1 \equiv m_1 \oplus m_2 \oplus m_3 \oplus c_1 \equiv 0 \quad (4.5.8)$$

Similar expressions can be constructed for c_2 and c_3 . Three such equations can be represented compactly in matrix form, as suggested in (4.5.9).

$$\begin{bmatrix} 1 & 1 & 1 & 0 & 1 & 0 & 0 \\ 1 & 1 & 0 & 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 1 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} m_1 \\ m_2 \\ m_3 \\ m_4 \\ c_1 \\ c_2 \\ c_3 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \quad (4.5.9)$$

which can be abbreviated $\mathbf{H}\mathbf{Q} = 0$, which defines the legal seven-bit code words \mathbf{Q} . In modulo-2 matrix multiplication, the multiplication of the first row of \mathbf{H} with \mathbf{Q} corresponds directly to (4.5.8). Thus it follows from (4.5.9) that we have defined c_2 and c_3 as

$$\begin{aligned} c_2 &= m_1 \oplus m_2 \oplus m_4 \\ c_3 &= m_1 \oplus m_3 \oplus m_4 \end{aligned} \quad (4.5.10)$$

Note that the modulo-2 operator \oplus commutes², and is associative³ and distributive⁴ with respect to binary one-bit multiplication ($0 \cdot 0 = 0 \cdot 1 = 1 \cdot 0 = 0$; $1 \cdot 1 = 1$).

Equation (4.5.9) defines how check bits c_i are defined, and thus defines the legal code words Q where $HQ = 0$.

Suppose the received binary message vector $V \in Q_i$ ⁵, where Q_i is the set of legal seven-bit blocks; and where the sixteen legal blocks correspond to all possible combinations of the four message bits m_i . We can define an error vector E with seven entries, where the locations of 1's defines the locations of the bits in error. Thus an illegal vector V with one bit error can be represented as

$$V = Q \oplus E \quad (4.5.11)$$

The decoder in a single-error-correction coded system then can compute for each received vector V the sum-modulo-2 vector product

$$HV = HQ \oplus HE = HE \quad (4.5.12)$$

and use the resulting vector product HE in a look-up table to determine which bit, if any, is in error. If $HE = 0$, the received vector V is error free and one of the legal code words.

Although it certainly is not necessary, there may be conceptual or implementation advantages in defining H so that the product HE is a three-bit number specifying the location of the error in the seven-bit word V . Shuffling the check and message bits as suggested in (4.5.13) accomplishes this for this particular example. The columns of the matrix H in (4.5.13) have been arranged so that they sequentially represent the binary numbers for one through seven (S). Note from (4.5.9) and (4.5.13) that HE , by definition, singles out that column j of H , which corresponds to the one bit in E which is nonzero. If there are no errors, $HE = 0$.

² The operator \oplus is commutative if $a \oplus b = b \oplus a$.

³ The operator \oplus is associative if $(a \oplus b) \oplus c = a \oplus (b \oplus c)$.

⁴ The matrix multiplication operator distributes over the vector sum-modulo-2 operator \oplus if $H(Q_1 \oplus Q_2) = HQ_1 \oplus HQ_2$.

⁵ $a \in B$ means the element a is in the set B .

$$\begin{matrix} \begin{bmatrix} 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 & 1 & 0 & 1 \end{bmatrix} \\ (j=1 \ 2 \ 3 \ 4 \ 5 \ 6 \ 7) \end{matrix} \begin{bmatrix} c_3 \\ c_2 \\ m_4 \\ c_1 \\ m_3 \\ m_2 \\ m_1 \end{bmatrix} = \mathbf{HE} \quad (4.5.13)$$

These same principles for designing error-correction codes can be extended to blocks of any size and to corrections of any number of errors, provided that the number of check bits R is sufficiently large relative to the number of message bits K .

Channel coding has two desirable effects that can be combined when designing any particular communications system. One benefit is to reduce the probability of error for a given communications channel to any particular desired level, although this may slow the rate at which information is communicated. This reduction in P_e can also be exchanged for reductions in costly transmitter power, antenna gain, or signal-to-noise ratio. If coding is used to reduce transmitter power or such related system parameters while keeping P_e constant, these benefits are called *coding gain*.

Consider the following simple example employing the single-bit-error-correction scheme described above for seven-bit blocks. Suppose a given communication system has a probability of bit error $P_e = 10^{-5}$. Then the probability of an error in a four-bit message $[m_1 \ m_2 \ m_3 \ m_4]$ is

$$1 - (1 - P_e)^4 = 1 - (1 - 10^{-5})^4 \cong 4 \times 10^{-5} \quad (4.5.14)$$

If we now add three bits to each four-bit message to produce seven bits, and then send it in the same time period at the same power, the ratio E/N_0 is reduced by a factor of 4/7, for a loss of 2.4 dB. Depending on the modulation scheme, represented by a curve such as those in Figure 4.4-3, the reduction of E/N_0 would cause P_e to increase, say to 6×10^{-4} per bit. The received seven-bit vectors V will yield the four message bits error-free provided there are not two or more errors in that seven-bit block. Because P_e is so small, we may neglect the probability of three or more errors to a reasonable approximation. The probability of two errors out of seven is the probability of perfection in 5 bits, times the probability of error in two particular bits (P_e^2), times the number n of possible ways two errors can be arranged in a set of seven ($n = 7 \cdot 6/2$); the first error can be in any of seven positions, the second in any of six, and since the sequence is irrelevant, we divide by two. Here this probability of two errors is approximately 8×10^{-6} . The probabilities of various numbers of errors in a seven-bit block are suggested in (4.5.15).

$$\begin{aligned}
p\{0 \text{ errors}\} &= (1 - P_e)^7 \cong 1 - 7P_e \\
p\{1 \text{ error}\} &= p\{6 \text{ perfect bits}\} P_e^7 = [1 - P_e]^6 P_e^7 \\
p\{2 \text{ errors}\} &= p\{5 \text{ perfect bits}\} \cdot P_e^2 \cdot 7 \cdot 6/2 = (1 - P_e)^5 P_e^2 \cdot \frac{7 \cdot 6}{2}
\end{aligned} \tag{4.5.15}$$

In the present case we began with a probability of bit error of 10^{-5} , which increased when we speeded up the signal transmission to provide extra time for sending check bits. However, once these check bits are interpreted, we find that the probability of block error (two bit errors in a single block) has been reduced approximately to 7×10^{-6} . To be fair, we should express this as an equivalent probability of message bit error. In this case, the two bit errors in an erroneous block will not be corrected if $\mathbf{H}\mathbf{Q} = 0$. Since only two out of seven bits would be in error, the corresponding probability of message bit error is two-sevenths of 7×10^{-6} , or $P_e = 2 \times 10^{-6}$.

So far we have not changed the average transmitter power; we merely reduced the energy E per bit by increasing the bit transmission rate by the factor $7/4$. We now may reduce this average transmitter power until the net effective P_e after decoding increases to the original level of 10^{-5} . It is this achievable reduction in transmitter power (or the equivalent), which we define as coding gain.

How much coding gain is available in any particular communication system depends very much on the steepness of the curve relating P_e to E/N_0 . For example, coding can actually be counterproductive unless the equivalent probability of bit error P_e is reduced sufficiently by coding to compensate for the increases in P_e due to the faster transmission needed to accommodate the check bits. When the P_e versus E/N_0 curve is steep, rather small increases in transmitter power can achieve the same reduction in P_e available through coding. The greatest opportunities for channel coding occur for channels that have much more gradual P_e curves, such as channels which fade. In these cases, the rather significant reductions in P_e which result for modest error-correction channel coding could only be achieved otherwise by substantial increases in average transmitter power. Thus channel coding is used most routinely in channels where fading, burst noise, or similar time-varying impairments yield gradual P_e curves, such as those suggested in Figure 4.5-3. In these cases, coding gains exceeding 10 dB can sometimes be achieved. For any particular channel, the coding gain increases as the block length and coding efficiency increase, introducing system complexity and increased coding delays.

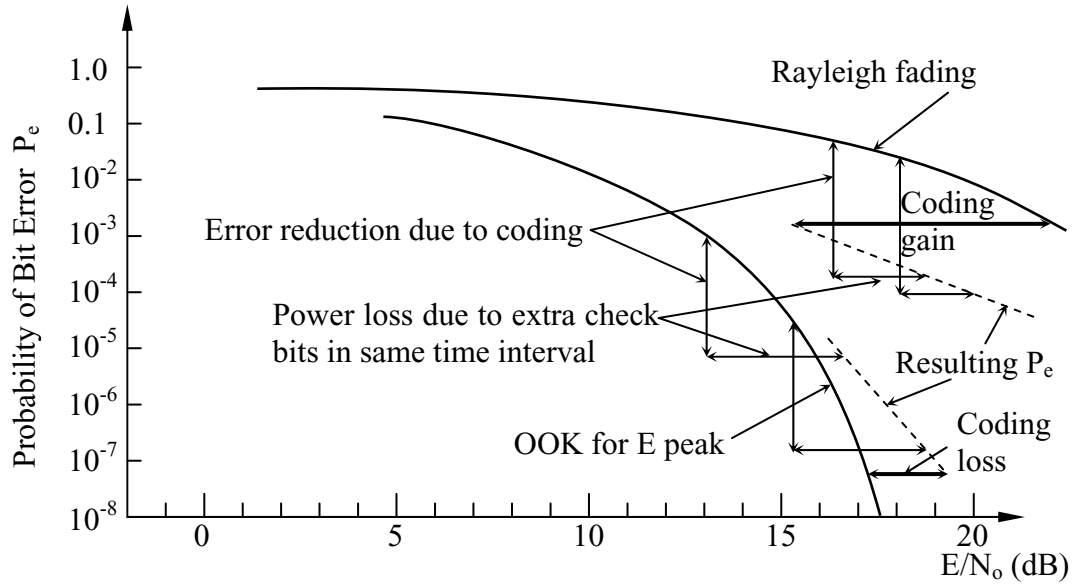


Figure 4.5-3: Origin of coding gain

Figure 4.5-3 suggests how an initial relationship between P_e and E/N_0 can be converted to a second one through channel coding. The horizontal offset between two curves is defined as coding gain if the required E/N_0 is reduced by coding.

Since the greatest opportunities for channel coding often arise in cases where random channel fading or burst errors are involved, additional accommodations are sometimes made to the burst character of the errors. One approach is to use *interleaving*, where a signal to be transmitted, including any check bits, is scrambled in a systematic and reversible way for transmission so that a single burst error is distributed over several original data blocks. After reception, the bits are unscrambled and error-correcting codes can accommodate the resulting smaller number of errors in each block. For example, if a typical burst causes three adjacent errors, then the input stream of error-corrected blocks might be grouped in sets of three for which the bits are then interleaved as suggested in Figure 4.5-4. If three bits in a row are changed, then when the interleaving is reversed to reproduce the original three blocks, each of these blocks will contain only one error, which can be accommodated.

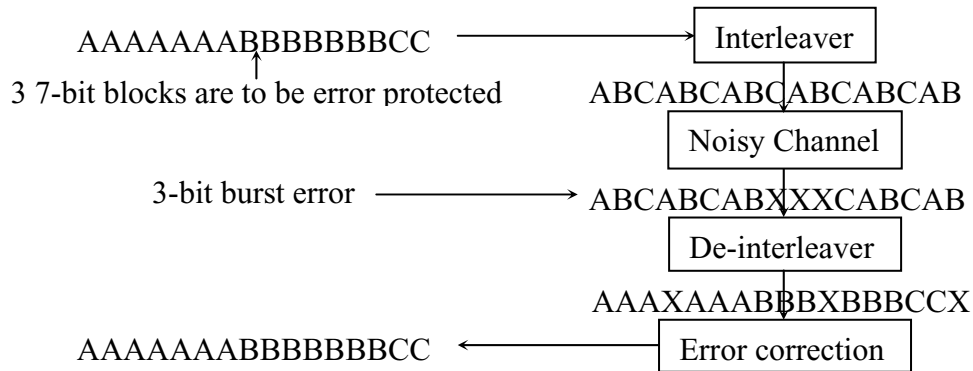


Figure 4.5-4: Interleaving to reduce burst errors.

Another approach is to use *Reed-Solomon codes*. These codes tolerate adjacent bit errors better than random ones. They are formed by grouping many bits together into single symbols, for which symbol-error-correcting codes are constructed. A single-error correcting code thus corrects any single symbol which is erroneous, but that symbol might convey four bits of information and have sixteen possibilities, for example.

One technique for dealing with burst errors which does not employ codes is *diversity*, where this can include *space diversity*, *frequency diversity*, and *polarization diversity*. In each case two or more propagation paths with independent and detectable burst errors are employed. Since the probability of simultaneous burst errors on two or more channels can become vanishingly small, rather low order diversity systems are often very satisfactory. Diversity obviously can be combined with other channel coding techniques.

The examples thus far have involved block codes, where each block as it arrives is interpreted independently of data arriving previously or subsequently, with the exception of interleaving, which simply scrambles bits in an orderly known way. *Convolutional codes* are different. Each bit transmitted is a function of the M most recent bits to have arrived at the encoder for transmission. The decoder in turn produces the hypothesized output bit as a function of the M most recent bits received from the channel. M is called the *constraint length*.

An example of a rate one-half constraint-length-3 convolutional coder is illustrated in Figure 4.5-5. In this case the transmitted output bit is taken alternately from two possible modulo-2 adders; since the message bit rate is therefore half the transmitted rate, we call it a rate-one-half coder. A simple error-checking decoder is also shown in Figure 4.5-5. The contents of both 3-bit registers are identical. In this case the decoding shift register contains the first two bits in any three-bit sequence, a and b . The value a is added to the incoming message $a \oplus c$ and $a \oplus b$ is added to $a \oplus b \oplus c$. In either case an error-free system will yield the correct value for c . If there are no errors the outputs of the two modulo-2 summers in the decoder will be identical, and when they are subsequently summed modulo-2 the result is 0 unless there is a bit error. Information about which received bit is in error can be obtained by assuming temporarily that both hypothesized values for c are correct, and then observing the subsequent behavior of two

different decoding results, one for each hypothesis; one of them will remain error-free for an extended period, whereas the other typically will indicate an error almost immediately, suggesting that hypothesis was faulty.

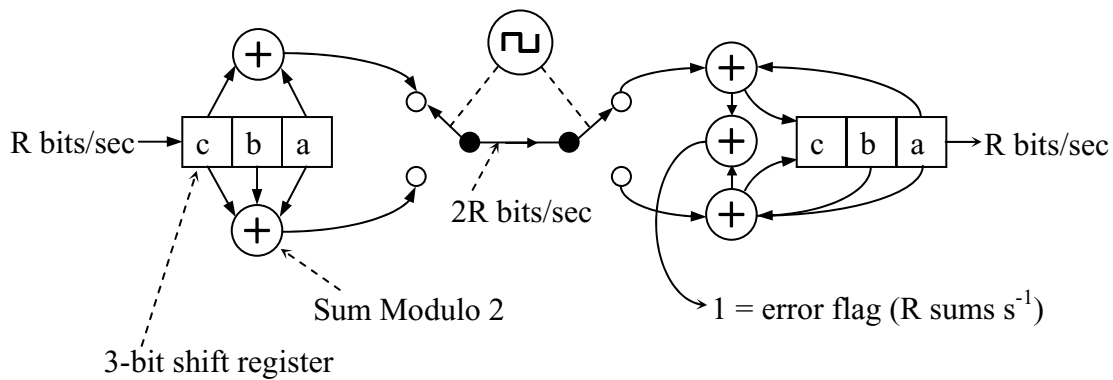


Figure 4.5-5: Convolutional coder and decoder.

Convolutional coding systems lend themselves to such deferred decision making, yielding better error performance. Such deferred or *soft decisions* can be implemented in other ways as well. For example, the binary communications systems described in Section 4.2 divided the received signal space \bar{v} into M regions, each corresponding to an explicit hypothesis concerning which symbol was transmitted. An alternative is to further subdivide this received signal space into more regions, some of which correspond to specific ambiguities. For example, a binary communications system can be divided into three spaces, two being unambiguously 0 or 1, and one corresponding to uncertainty, which could be resolved advantageously after more data had been received. The same binary decision space could be divided into even more degrees of certainty concerning whether 0 or 1 were transmitted. The use of soft decisions in modern communications systems can typically yield up to a few dB improvement in the signal-to-noise ratio for the same P_e .

Example 4.5.1

For the case of Example 4.4.1 and coherent BPSK modulation, a 10^{-10} W signal yielded $P_e \cong 10^{-6}$ for 8.9 kbps and 10^{-15} WHz^{-1} white noise in the receiver. The associated signal bandwidth is roughly 8.9 kHz for BPSK. For this bandwidth, signal power, and noise, what is the actual channel capacity? For the same bandwidth, data rate, and noise, what is the minimum possible signal power for $P_e = 0$? By how many dB can it be reduced in theory?

Solution:

Shannon's channel capacity theorem (4.5.1) says $C = B \log(1 + S/N_o B)$ (bits/sec)
 $\cong 8900 \log_2(1 + 10^{-10}/8900 \times 10^{-15}) \cong 8900 \times 3.5 \cong 31$ kbps. If $C = 8900$ (bits/sec) = B (Hz),
then $\log(1 + S/N_o B) = 1$ and therefore $S = N_o B \cong 8.9 \times 10^{-12}$ (W), which is a savings of ~ 10.5
dB even as $P_e \rightarrow 0$. This theoretical limit to coding gain is impossible to achieve in practice.

Example 4.5.2

Assume for a particular channel modulation scheme $P_e = 10^{-6}$ for $E/N_o = 18$ dB, and more generally $\log_{10} P_e \cong -E/3N_o$ dB in the operating region of interest. What coding gain results at the reduced P_e if single-error correction for 1000-bit blocks is employed? What gain results if double-error correction is employed for 1000-bit blocks?

Solution:

Single-error correction requires 10 bits per 1000-bit block ($2^{10} > 10^3 + 10 + 1$), and therefore a speed up of one percent (10/1000) with a corresponding one percent increase in average power since E/N_o remains constant if P_e does. Errors result only if two or more occur in a single block, where the probability of two errors is $(1 - 10^{-6})^{998} (10^{-6})^2 \times (1000 \cdot 999/2) \cong 5 \times 10^{-7}$ per block. The probability of three or more errors is negligible in comparison. Two or more errors per block typically results in the wrong bit being corrected, and 3-bit errors per block. One percent of these are parity bits and irrelevant. The resulting reduced message-bit error probability is $\sim 3 \times 5 \times 10^{-7} \times 10^{-3} = 1.5 \times 10^{-9}$. Had we wished to achieve this P_e without coding, E/N_o (dB) would have to be $-3 \log_{10} P_e \cong 26.5$ dB, or ~ 8.5 dB greater than with this coding scheme. At $P_e \cong 1.5 \times 10^{-9}$ the coding gain for single-error correction therefore approximates 8.5 dB.

Double-error correction reduces the probability of block error roughly to the probability of three errors, or $(1 - 10^{-6})^{997} (10^{-6})^3 (1010 \cdot 1009 \cdot 1008/3!) \cong 1.7 \times 10^{-10}$. Such a block would typically have roughly $3 + 2 = 5$ message bit errors after corrections, so $P_e \cong 5 \times 1.7 \times 10^{-10} \times 10^{-3} = 8.5 \times 10^{-13}$. Had this small P_e been our original objective, E/N_o would otherwise have to be $-3 \log_{10} P_e = 36.2$ dB, so the coding gain would be 18.2 dB. These coding gains are large because the given $P_e(E/N_o)$ relation is not steep. Computing the coding gain for a given P_e usually requires numerical or iterative methods, using arguments similar to those employed here.

4.6 SOURCE CODING AND TRANSFORMS

Source coding or *data compression* generally removes redundancies in the input signal, which can be reversibly recovered. They furthermore may discard less perceptible or important aspects of the signal stream, which may not be recovered at the decoder; this is called *lossy coding* and is discussed later. *Lossless coding* or *reversible coding* corresponds to source-coding systems where the receiver recovers the desired source message perfectly even though the bit rate conveyed over the communications channel may be less than the bit rate associated with the original signal. Lossless *entropy coding*, sometimes called *Huffman coding*, generally employs fewer bits to indicate the presence of common symbols or messages, and employs more bits for those which are more unexpected or unusual. This strategy reduces the average number of bits required to convey any signal stream.

A simple example illustrates the principle. Suppose four possible symbols are to be transmitted, which could be represented with two bits each so that the average data rate is two bits per symbol communicated. If, however, the *a priori* probabilities of these four symbols are, for example, 0.5, 0.25, 0.125, and 0.125 (totaling unity), then a more efficient coding scheme is to represent the first symbol by the transmitted message “0”, the second symbol by “11”, and the last two by “100” and “101”, respectively. Unbroken sequences of such symbols following one another endlessly form a *commaless code*, which can be subdivided unambiguously into its component parts, provided synchronization correctly establishes the location of the first bit. For example, the commaless bit stream 0100101111100100 can be divided unambiguously into 0, 100, 101, 11, 11, 0, 0, 100. Such bit streams are generally self-synchronizing.

The average bit rate communicated in this example is then one bit times its probability (0.5 for the symbol 0) plus two bits times its probability (2×0.25 for the symbol 11) plus three bits times the probability that the last two symbols are transmitted ($3 \times 2 \times 0.125$). This total average number of bits per symbol transmitted is 1.75, representing a small savings over the initial two bits per message.

This minimum average number H of bits per symbol is defined as the *entropy* of the input signal stream, where

$$H = -\sum_{i=1}^M P_i \log_2 P_i \quad \text{bits} \quad (4.6.1)$$

and the P_i is the probability that symbol i will be sent. In the four-symbol example above, the entropy H is 1.75 bits per symbol. The entropy H of a random binary stream is one bit per symbol, where $P_i = 0.5$, and the number of possible symbols M equals 2. The entropy of a typical symbol is sometimes also called its *information content*. If the probabilities of sequential bits depend on each other, then the expression for H becomes more complex; here we assume the symbol probabilities are independent.

Entropy codes can be implemented in a variety of ways. For example, the symbol stream to be transmitted can be divided into equal length blocks, each of which is transformed into a new block containing more or fewer bits than the original; the encoded data stream then comprises unequal-length blocks. Alternatively, the transmitted signal can comprise equal-length blocks, each corresponding to blocks of nonuniform length in the original symbol stream. Finally, both the input blocks and the transmitted blocks can be of unequal length.

One important type of entropy code is called *run-length coding*. Consider a two-dimensional black and white image coded by scanning the image in a two-dimensional raster and converting the sensed luminance into 1's and 0's. If the sample size represented by a single bit is small compared to the typical extent of any black or white region, the resulting bit stream will tend to consist of long series of 0's broken by similar series of 1's. Such a message can be represented alternatively as the number of 0's and 1's in each run. Figure 4.6-1 illustrates how such a conversion might be made. The transmitted signal might simply consist of the binary representations for each of these run lengths or the run lengths might be Huffman coded first.

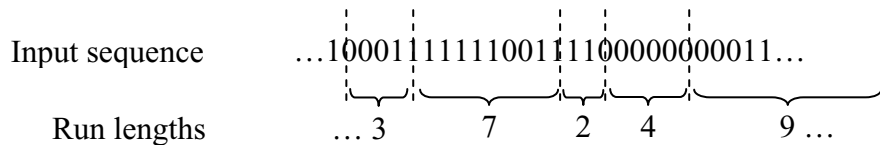


Figure 4.6-1: Run-length coding.

In general, the larger the block of data which is entropy coded, the closer the final system can approach the true entropy in the input data stream. One way to entropy code smaller blocks but still achieve some of the savings associated with using larger ones, is to use entropy codes based on probability distributions conditioned by the previously transmitted symbol. That is, for each transmitted symbol the conditional probability distributions for the subsequent symbol are computed. Since the receiver hopefully knows this previously transmitted symbol, it knows which entropy code is being used on a symbol-by-symbol basis. This technique captures some of the entropy reduction associated with the correlation between adjacent symbols.

Some of the best performing are arithmetic codes⁶ and turbo codes.

The second mechanism for reducing data rates is to use *information-lossy codes*, which achieve compaction at the expense of the signal-to-noise ratio. The more successful information-lossy codes distribute the added noise selectively so that it is least visible. These codes generally reduce redundancy in the signal, and then quantize the resulting numerical representation. The signal processing employed to reduce the redundancy must be reversed at the receiver, and can be quite complex.

⁶ For example, see IEEE Trans. Comm., 37, 2, pp. 93-97, February, 1989.

Because the sensitivity of the human visual and auditory systems to noise appears to vary with frequency over the spatial or temporal spectral representations of these signals, various frequency transforms are often computed before quantization is performed. One basic representation is the *discrete Fourier transform* (DFT). It is defined over a block of N samples as

$$\underline{X}(n) = \sum_{k=0}^{N-1} x(k)e^{-jn2\pi(k/N)} \quad (4.6.2)$$

$[n = 0, 1, \dots, N - 1]$

Its inverse transform (IDFT) is

$$x(k) = \frac{1}{N} \sum_{n=0}^{N-1} \underline{X}(n)e^{jn2\pi(k/N)} \quad (4.6.3)$$

Equation (4.6.3) defines how signal $x(k)$ can be constructed by superimposing the complex spectral coefficients $\underline{X}(n)$ by weighting them with the basis functions given by $e^{jn2\pi(k/N)}$. This basis function is unity for all values of k when $n = 0$. The DFT basis functions are presented in Figure 4.6-2(a) for $n = 0, 1$, and 2 . These basis functions correspond to real values for $\underline{X}(n)$. Note that in general $\underline{X}(n)$ is complex, and contains $2N$ scalars for a signal sampled at N points. If $x(k)$ is either pure real or pure imaginary, then $\underline{X}(n)$ is redundant by a factor of 2. The real and imaginary parts for the DFT of a real function are unique only for $n < N/2$; the axis of symmetry is near $n = n/2$, and $X(0) = X(N)$.

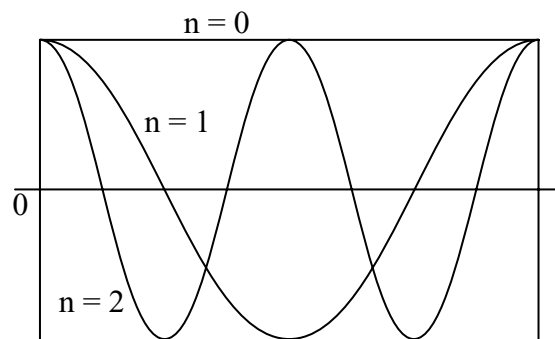


Figure 4.6-2(a): Discrete Fourier transform (DFT) basis functions for $\text{Re}\{\underline{X}_n\}$.

Two other block transforms which are widely used are the *discrete cosine transform* (DCT) and the *discrete sine transform* (DST), for which the basis functions for low values of n are illustrated in Figure 4.6-2(b) and (c), respectively. Note that these transforms are linear and lossless; the number of unique scalars produced by these transforms equals the numbers of

unique (real-valued) scalars in the input signal. Compaction comes only as we quantize these output spectral coefficients and possibly even discard some.

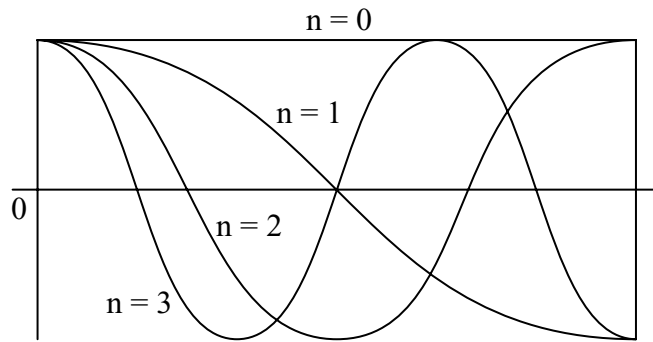


Figure 4.6-2(b): Discrete cosine transform (DCT) basis functions.

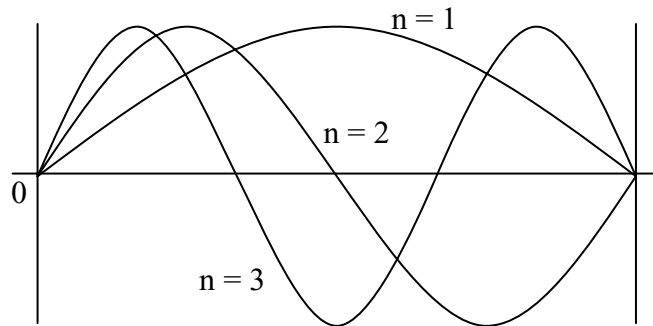


Figure 4.6-2(c): Discrete sine transform (DST) basis functions.

Note that the DCT, given by (4.6.4), transforms N real samples into N real spectral coefficients, whereas the DFT produces complex spectral coefficients; this simplicity of the DCT has contributed to its popularity.

$$Y(n) = c_0 \sqrt{\frac{2}{N}} \sum_{k=0}^{N-1} y(k) \cos(n\pi(k+0.5)/N) \quad (4.6.4)$$

where $c_0 = 2^{-0.5}$ if $n = 0$, and 1 otherwise.

Since the number of output coefficients generally equals the number of input coefficients, the signal compaction benefits of transforms lie elsewhere. Consider a signal which is sampled well above its Nyquist rate. Its transform will have significant coefficients only at lower frequencies, and thus the number of significant parameters needed to approximate this signal in the spectral domain has indeed been reduced. Even if these high-frequency coefficients are not reduced to zero, fewer bits are needed to represent them to any desired level of accuracy. Since most of the redundancy in real signals lies in the correlations between adjacent samples in one,

two, or more dimensions, spectral techniques are efficient at representing them in a compact fashion.

The *Karhounen-Loeve transform* (KLT) is designed to achieve the maximum compaction possible for a signal whose variations are characterized by a jointly Gaussian random process. Its basis functions are the eigenvectors of the signal correlation matrix. Among other transforms of special interest are two families called *lapped orthogonal transforms* (LOT) and *modulated lapped transforms* (MLT). Both the LOT and MLT employ basis functions that extend into adjacent blocks while remaining orthogonal within their own block and orthogonal between blocks. Because they avoid the sharp discontinuities which occur in the DCT and DFT basis functions, they effectively exhibit lower sidelobes when representing multi-block signals; this makes them more efficient at compaction. Although the DST employs basis functions without sharp discontinuities at block boundaries, it is less efficient at representing constant signals because many basis functions must be employed in order to minimize artifacts at the block boundaries. For both the LOT and MLT, constant signals across many blocks can be represented by linear superpositions of only the first term, as in the case of DFT and DCT; this is true in both one and two dimensions. Lower order basis functions for both the LOT and MLT are suggested in Figure 4.6-3.

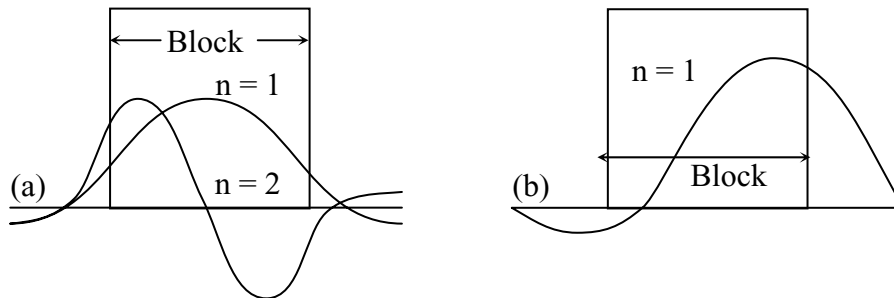


Figure 4.6-3: Lapped transform basis functions (a) LOT, (b) MLT.

Figure 4.6-4 illustrates the average and standard deviation of the energy in each of the basis functions for various orthogonal representations for a typical image or auditory signal that has been divided into a large sequence of blocks. Note that the least efficient representation is given by Figure 4.6-4(a), which is the original representation in space or time. In this case the energy is distributed equally among the various single-impulse basis functions, each corresponding to a different point in space or time. The energy is distributed equally because we have assumed the signal is stationary.

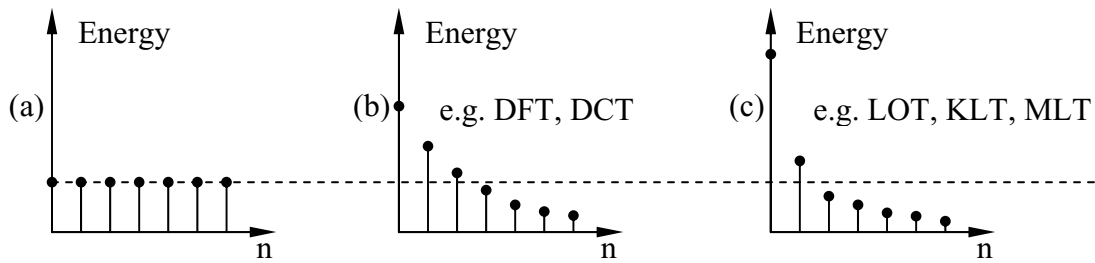


Figure 4.6-4: Energy compaction for transforms of increasing coding efficiency, a-c. n represents space or time in (a), and frequency in (b) and (c).

The total energy is, by Parseval's theorem, identical for representations in the other transforms illustrated in the figure. The transforms differ, however, in their ability to concentrate energy in the lower order functions. Note that the LOT and MLT typically achieve higher levels of compaction than the DFT or DCT because of their overlapping nature into adjacent blocks. The KLT is superior to the DFT and DCT by virtue of its mathematical definition as the optimum transform for a given ensemble of waveforms governed by a jointly Gaussian random variable process. However, it can be shown that for first-order 1-D or 2-D Markovian jointly Gaussian processes, the DCT is the KLT. Best of all is a lapped KLT.

Figure 4.6-5 suggests what the two-dimensional DCT coefficients might be for 8×8 pixel blocks comprising a typical image. The magnitude contours for the coefficients corresponding to the smoother portion of the image are indicated by blocks of type (c), where the significant entries are all in the upper-left-hand corner, as shown in the figure. Regions of the image containing strong horizontal striations might have coefficient magnitudes characterized by (d), where most of the energy is in the low, horizontal frequency terms. Vertical striations and diagonal striations might produce coefficients as suggested in transforms (e) and (f), respectively.

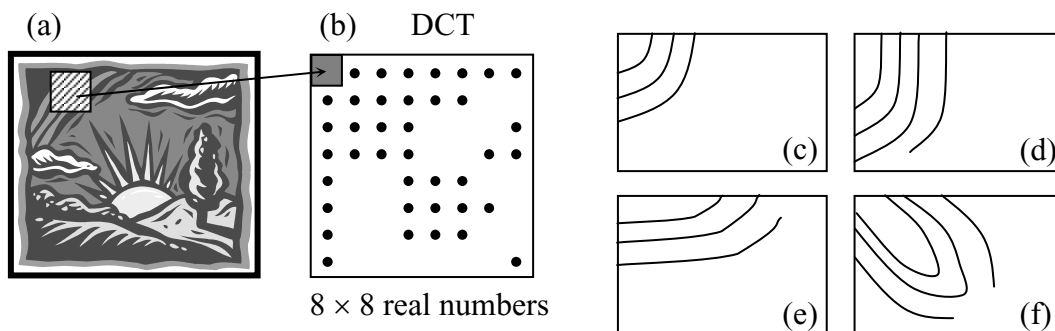


Figure 4.6-5: Adaptive DCT image coding.

Once the representation for a given signal is chosen, a quantization scheme must be selected. When all coefficients for a signal have identical statistics, the number of quantization

bits assigned to each variable is generally identical. However, when the energy in the different coefficients is different, as in the cases illustrated in Figure 4.6-4 and 4.6-5, then more bits are assigned to those terms for which the average energy is greatest, and relatively fewer bits to those for which the average energy is less.

The advantages of this are illustrated by the simple example of a smooth image for which the luminance is uniform over each block of M samples, but randomly distributed from block to block between the values of zero and one. If we wish to represent this image with eight-bit accuracy, then we must use approximately eight bits for each of the M samples across the block. In this special DC case, the total number of bits (8) required to represent this signal to the same accuracy in the spectral domain using the DCT is reduced roughly by a factor of M , which can be very significant. Adaptive coding techniques that vary both the number of bits assigned to each coefficient and the amplitude of each bit as a function of frequency can extract most of the redundancy and realize most of the opportunities for adding imperceptible coding noise to natural signal streams. Although additional reductions in transmitted data rates can be achieved by systems that physically model the signal source and predict its behavior-- for example, by motion compensation or by full knowledge of the signal's likely evolution-- most such systems employing more than motion compensation are still in the research stage.

Additional savings can be obtained by clever choice of quantization scheme. Usually the range of interest for any variable, such as signal voltage, is divided uniformly into equal-size quantized bins, the number of bins normally being some power of two or ten. Because any value falling within a particular bin is ultimately represented by a single symbol, an RMS error is introduced which approximates 0.29 of the full bin width for a uniform distribution. If we fix the number of bits n used to represent a given variable, and therefore the number of bins 2^n , then there is a potential reduction in the RMS error if we make the bins of nonuniform width, using smaller ones in regions where the probability density of the variable is greater. For a given number of average bits and a given probability density distribution, there is an optimum way to do this. The nature of the result is suggested in Figure 4.6-6. A trivial example is one where the random variable is uniformly distributed over only half its possible range. If the same number of bits is now distributed over this half of the range only, each bin is now half as wide and the RMS quantization error is now halved. Alternatively, one fewer bit can be used without degrading the RMS error.

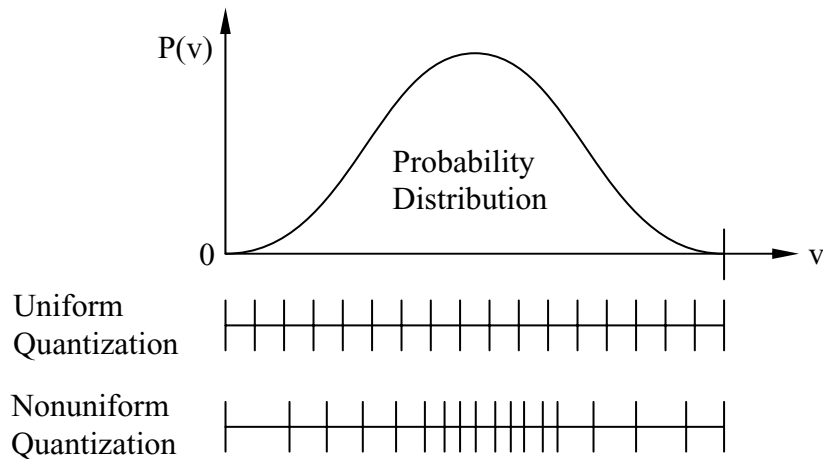


Figure 4.6-6: Non-uniform quantizers

An important variation of this technique of nonuniform quantization is *vector quantization* (VQ). Normally, scalars are quantized, whereas in VQ it is vectors. The simplest case is a two-element vector $\vec{v} = [a, b]$, where a, b might be, for example, adjacent samples in a sampled data stream. The joint probability distribution of a, b is suggested by the contours in Figure 4.6-7, where $P(a, b)$ is a jointly Gaussian process and a, b are clearly correlated. Conventional scalar quantization is illustrated in Figure 4.6-7(a), where three bits for each a and b define which cell in the two-dimensional space is of interest. In this simple case the numerical addressing of each cell for VQ can be made identical to conventional scalar quantization. Figure 4.6-7(b) illustrates a variation, where the same three bits for each axis address nonuniform quantizers having finer resolution near the center of the probability distribution. Because the probability distribution is skewed here by the correlation between a and b , still better performance can be obtained with nonlinear quantizers by rotating the a, b space into the c, d space by a matrix multiplication, as illustrated in Figure 4.6-7(c). The more highly correlated the variable a and b , the greater the relative efficiency of the VQ scheme illustrated in (c). In each of these three cases the numerical identifier for each possible vector has been divided into two subparts, one for each of two axes. Figure 4.6-7(d) illustrates an even more general VQ for this same problem, which achieves still higher performance; each equiprobable region has a unique identifier.

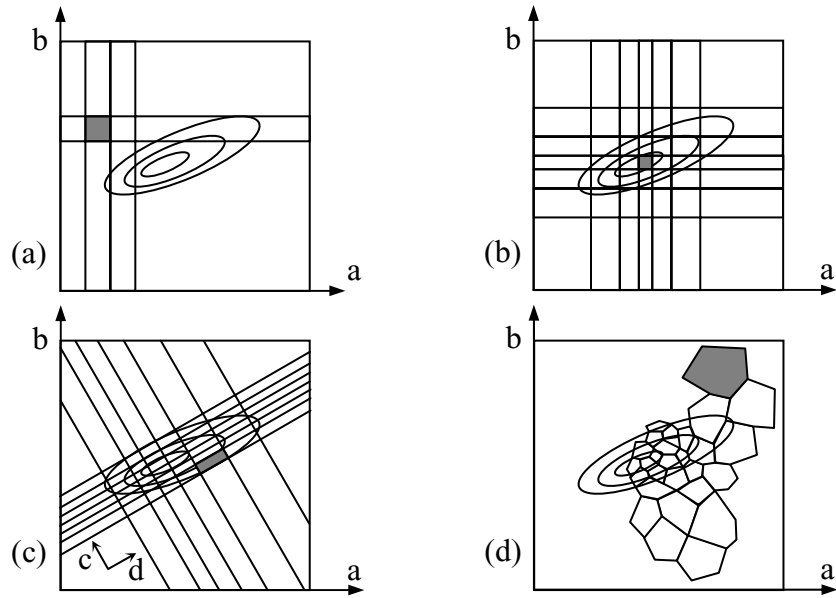


Figure 4.6-7: Vector quantization schemes: (a) standard uniform, (b) standard non-uniform, (c) rotated non-uniform, (d) arbitrary regions.

In the practical design of vector quantizers, there is usually a significant trade-off between system performance and the computational encoding costs associated with assigning the variables to a particular vector identification number, and the decoding cost of interpreting that identification number in terms of the original parameters. Vector quantizers can involve up to sixteen and more dimensions, although as the dimensionality increases the incentive is to use the simpler variants, such as illustrated in (b) and (c). In general, the performance of VQ is roughly comparable to that of systems employing an efficient transform followed by an efficient entropy code such as arithmetic coding. Such design choices are often made on the basis of practical engineering considerations rather than fundamental theoretical issues.

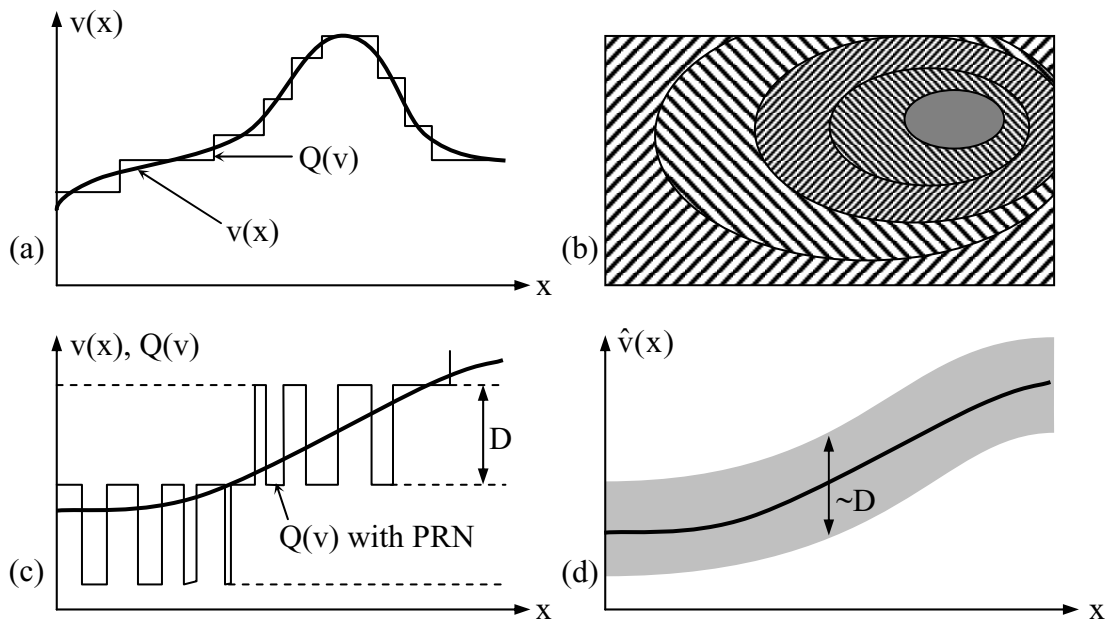


Figure 4.6-8: Pseudo-random noise (PRN) quantization of images.

A third type of quantization introduces randomness into the decision process to improve the perceptual quality of the result. A common variant of this technique is use of *pseudo-random noise* (PRN), which is uniformly distributed over the quantization interval. Consider an image with a smoothly varying luminance across its width. A coarse quantizer would divide it into bands of varying shades of gray with sharp and visibly annoying boundaries between them. It can be shown that if PRN is added to the analog signal prior to quantization, where the uniformly distributed random signal has a total dynamic range D equal to one quantization interval, and if the same PRN is subsequently subtracted from the reconstructed quantized signal, then the mean value of the resulting reconstructed quantized luminance image equals the original luminance curve without discontinuities in either the mean or the variance of the noisy reconstructed image. Figure 4.6-8(a) illustrates a luminance curve $v(x)$ and its quantized rendition $Q(v)$. The resulting two-dimensional reconstructed quantized image is shown in Figure 4.6-8(b), and its disturbing character is clearly evident. If we add pseudo-random noise n_{prn} then $Q(v + n_{\text{prn}})$ takes the form illustrated in Figure 4.6-8(c). When the receiver reconstructs the image, it knows exactly the pseudo-random noise sequence used to generate n_{prn} , and can subtract it to yield $Q(v + n_{\text{prn}}) - n_{\text{prn}}$, as illustrated in Figure 4.6-8(d). So long as the pseudo-random noise sequence is sufficiently scrambled as not to assume an identifiable character on its own, then this technique completely obliterates the artificial boundary illustrated in (b), typically without the expense of added noise. Low-pass filtering can now restore the signal if the sampling rate was sufficiently high.

A generalization of this technique involves for each independent signal sample the pseudo-random selection of a quantizer from a suite of possible quantizers, each with its own slightly

different distribution of quantization levels. The signal reconstruction levels associated with any quantized variable correspond to the original selected quantizer. Pseudo-random noise sequences can be generated by simple digital circuits; the sequence used in the receiver must be synchronized, however, with the sequence used at the transmitter. The improvements offered by such random quantization schemes are generally perceptual, unless low-pass filtering of the output can be employed to reduce rms error.

A final form of quantization is also nonlinear, where the inter-level spacing is small at low signal levels where human sight or hearing is more sensitive (e.g., in quiet musical passages), and larger at higher levels (e.g., loud passages). This technique is used to reduce the perception of quantization noise, and is called *companding*. Using such quantizers for audio or video signals permits, for example, use of eight-bit quantizers to yield ten or more bits of equivalent noise performance.

Lossy coding systems generally include some transformation that maximally concentrates the variation of the signal into the smallest possible number of degrees of freedom, and follows this with an appropriate quantization stage. Often, for simplicity, these orthogonal transforms are replaced with simpler *predictive coding* or *pre-filtering* and *post-filtering* schemes. These are generally much simpler to implement and, until recently, constituted the approach most commonly used.

The general form of a predictive coder is illustrated in Figure 4.6-9. As the signal $s(t)$ to be transmitted enters from the left, a prediction of its value \hat{s} based on prior data is made and subtracted from the original, leaving a residual signal, hopefully much smaller than the nominal excursions of the signal about its mean. This residual is then coded and transmitted over the channel. The receiver decodes this signal to reconstruct the original residual signal $s - \hat{s}$. This residual is then added to the predicted signal \hat{s} to reconstruct the original signal s . Both the transmitter and the receiver construct the estimate \hat{s} in the same way, as illustrated in the figure. The coded signal, which already suffered some coding delay Δ , is decoded to produce the coded signal s perfectly, but delayed by an additional decoding time of Δ . Thus the prediction \hat{s} for $t = t_0 + \Delta$ must be based on true values of s at a time 2Δ in the past. The 3Δ predictor used for this purpose can range from very simple to very complex. Use of the same decoders and predictors in the receiver as in the transmitter guarantees perfect reconstruction if there are no errors in transmission or execution. The time delay in the reconstructed signal s is 2Δ .

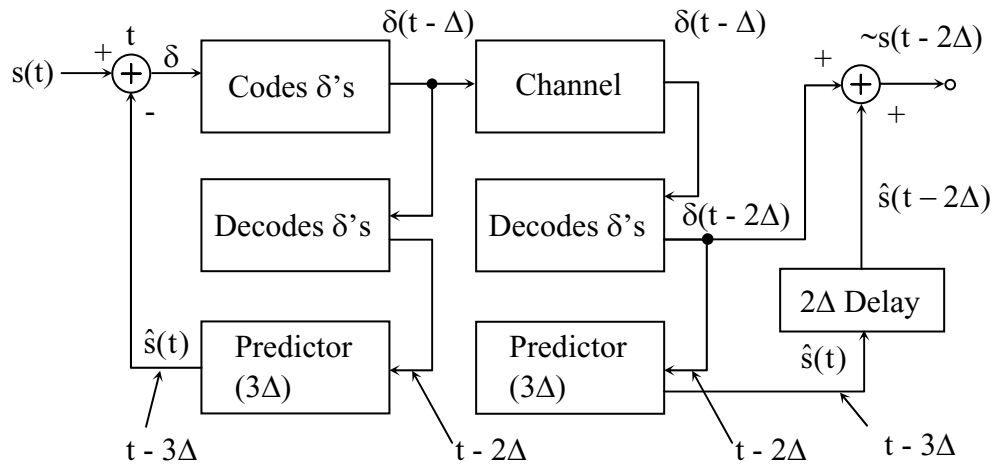


Figure 4.6-9: Predictive coder and decoder.

Because the effects of errors in this system are cumulative, and therefore inevitable, various techniques have been devised to eliminate them. The simplest such approach is to restart the system at regular intervals from a synchronized prediction of the signal mean or some other initialization value. Alternatively, various techniques exist for continuously devoting some fraction of the transmitted signal to this reinitialization and renormalization process.

A widely used technique for improving the perceptual performance of sample data systems is to use *prefiltering* and *postfiltering* to minimize the perceived mean square error. For a given system for communications or signal storage and retrieval, we often seek to approximate the ideal desired reconstructed signal $s(t)$ as viewed with an ideal observer impulse response function $r(t)$. A typical total system is illustrated in Figure 4.6-10. We may model the communications or memory system as having quantization noise plus two additive noises over which we have no control. One additive noise $n_1(t)$ (e.g., sensor noise) occurs in the transmitter prior to prefiltering, and the other is additive channel noise $n_2(t)$, which is effectively combined with the quantization noise. We further assume that the input signal is filtered by the linear filter $f(t)$ (e.g., a lense transfer or microphone response function) just prior to addition of $n_1(t)$, where we also have no control over $f(t)$. The challenge is to design the optimum linear prefilter $g(t)$ which precedes the sampler, and the ideal postfilter $h(t)$ which follows the channel. Again, the output signal will be observed by some observer characterized by the impulse response function $r(t)$. These filters may be implemented as block filters operating on predefined blocks of data, or as infinite impulse response (IIR) filters, or as finite impulse response (FIR) filters. The impulse responses of these three types of filters tend to resemble one another for any particular application.

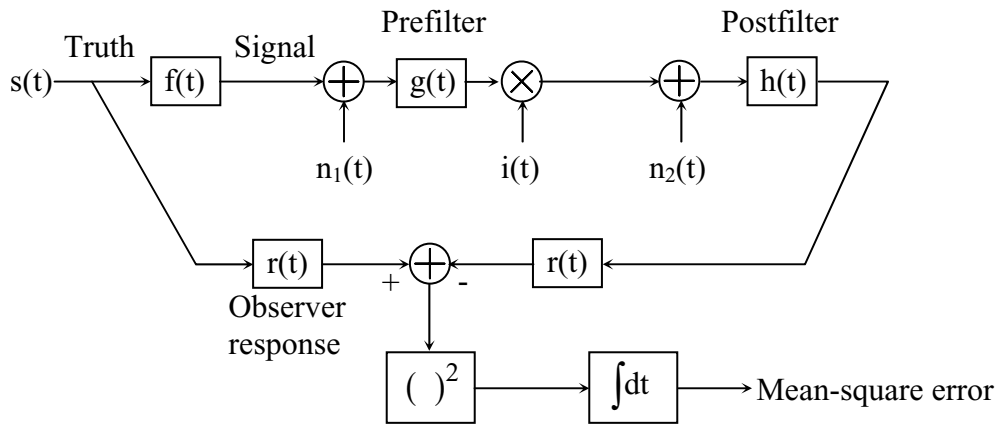


Figure 4.6-10: System flow diagram for prefiltering and postfiltering.

For a wide variety of signal types, the optimum pre- and post filters often take the form illustrated in Figure 4.6-11. The prefilter typically resembles a Gaussian with a little bit of overshoot, sometimes called a Mexican-hat function (Figure 4.6-11(a)), whereas the ideal postfilter (Figure 4.6-11(b)) generally resembles a Gaussian function. The reason why this form of prefilter and postfilter is generally preferred is suggested in Figure 4.6-11(c), where the typical signal spectrum (spatial or temporal) is shown to roll off toward the Nyquist frequency N_f , possibly even dropping below the additive noise level $N_o(f)$. In this figure we are neglecting aliasing noise and $n_1(t)$.

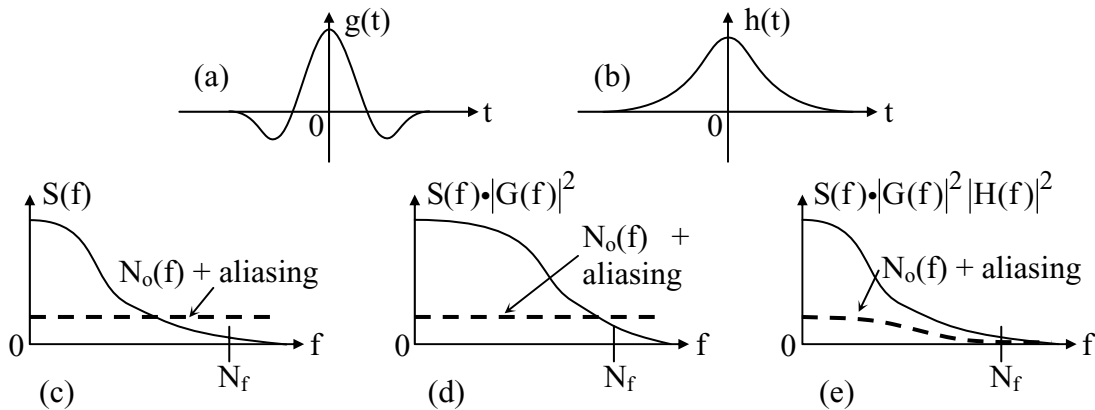


Figure 4.6-11: Prefiltering and postfiltering signals: (a) prefilter, (b) postfilters, (c) prefiltered spectrum, (d) postfiltered spectra.

The effect of the prefilter $g(t)$ illustrated in Figure 4.6-11(a) is to slightly enhance the input spectrum $S(f)$ at higher frequencies, as suggested in Figure 4.6-11(d). The optimum postfilter $H(f)$ is typically a low-pass filter, almost perfectly canceling the effects of the prefilter $G(t)$. Thus the spectrum of the reconstructed signal is approximately that shown in Figure 4.6-11(e),

where the original signal spectrum $S(f)$ has been essentially restored to its original value, but the post-filter $H(f)$ has reduced the high-frequency noise introduced in the channel and otherwise. The optimum prefilter enhances weaker portions of the signal spectrum so that the additive channel noise is relatively smaller in comparison, and the postfilter compensates for this distortion.

Example 4.6.1

Estimate roughly the number of bits required to FAX a typical typewritten page with (a) no coding, (b) run-length coding, (c) entropy-coded run-length coding, and (d) character coding.

Solution:

A black-or-white (one bit) representation of a page requires $\sim 1500 \times 2000 = 3$ Mbits, where a 5-inch line of text at 300 pixels (picture elements) per inch requires 1500 pixels or bits, and an 8.5×11 -inch page needs more pixels vertically (~ 2000) than horizontally (~ 1500). A typical line with 80 characters might require 240 runs of 0's and 1's if the characters average 1.5 vertical strokes each. If we represent each run with 11 bits (to accommodate the maximum run of 1500 pixels) then each such line requires roughly $11 \times 240 = 2600$ bits, more than the original 1500! However, in typical text over one-third the lines are pure white, requiring less than 30 bits, and another third might contain only the dangling tails of y's, g's, etc., or the tops of d's, f's, h's, etc. For these lines only ~ 40 runs might be required, or ~ 440 bits. The average is roughly $(1500 + 30 + 440)/3 = 657$ bits per line, a savings of ~ 2.3 . Entropy coding of run lengths of $\sim 7 - 20$ white pixels and runs of $1 - 13$ black pixels, yields a span of 13 which could be represented by 4 bits. Thus 11 bits per run could probably be Huffman-coded here to yield roughly another factor of two compression for the 40- or 240-run lines, but little savings for the empty lines. We might expect in this case a total of $(1500 + 440)/6 + 30/3 = 333$ average bits per line, a further savings of ~ 2.0 , or a total savings of ~ 4.5 . Character coding would require only 26 letters in two forms, plus punctuation, etc., or an alphabet of $64 - 128$ characters requiring 6 or 7 bits. A page of 80×40 characters would require $3200 \times 7 = 22,400$ bits versus 3×10^6 , a savings of 134. Entropy coding might increase this to a factor of 200 or more, particularly when we entropy-code combinations of characters or even the words themselves.

4.7 ANALOG COMMUNICATIONS

Analog communication systems convey analog, or continuously varying signals from one point to another, or into and out of a memory or recording system. We consider such systems to be "analog", even when implemented digitally, if they contain vulnerable points where additive noise readily blurs one parameter value into adjacent values. One essential character of analog systems, then, is that successive processing of the same signal gradually degrades it with each pass through any system. In contrast, digital systems are usually constructed so that multi-pass processing introduces little or no cumulative degradation. The degradation can, in fact, be zero an arbitrarily large fraction of the time.

Clever design of transmitted analog signals can, however, achieve many of the benefits of channel coding by similarly exchanging bandwidth for signal-to-noise ratio. Analog systems can also provide source coding benefits, for example, by dividing an analog signal into subbands that are transmitted over different subchannels with signal-to-noise ratios adapted to the amplitudes and perceptual significance of these various subbands.

One very simple analog communications system is a *double-sideband synchronous carrier* (DSBSC) system. To convey this signal $s(t)$ over a channel with additive Gaussian white noise $n(t)$ near a carrier frequency ω_c requires us to recover $s(t)$ from the received signal $x(t)$, where

$$x(t) = A_c s(t) \cos \omega_c t + n(t) \quad (4.7.1)$$

Multiplication of the signal $s(t)$ with the carrier ω_c at the transmitter produces sum and difference spectra which appear as upper and lower sidebands on either side of the carrier, as illustrated in Figure 4.7-1. The double-sided noise spectrum at the receiver has amplitude $kT_R/2$ (W Hz^{-1}), and the mean squared value of the noise $n(t)$ is

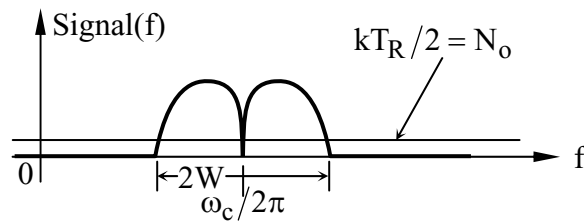


Figure 4.7-1: Signal and noise spectra.

$$\langle n^2(t) \rangle = (kT_R/2)4W = N_o 4W \quad (4.7.2)$$

The base bandwidth of one sideband is W Hz. We assume here a nominal 1-ohm impedance so that we may relate the voltage $n(t)$ to noise power. The receiver noise temperature T_R , is defined and discussed in Section 2.2.1.

Figure 4.7-2 illustrates the basic transmitter and receiver configurations which can be used to recover the estimated original signal $\hat{s}(t)$ from the noisy received signal $x(t)$. A simple receiver simply multiplies the received signal $x(t)$ by $\cos \omega_c t$. This translates the double-sideband transmitted signal to dc, and the signals produced by this multiplier at $2\omega_c$ are eliminated by the low-pass filter which follows, as illustrated in Figure 4.7-2(b).

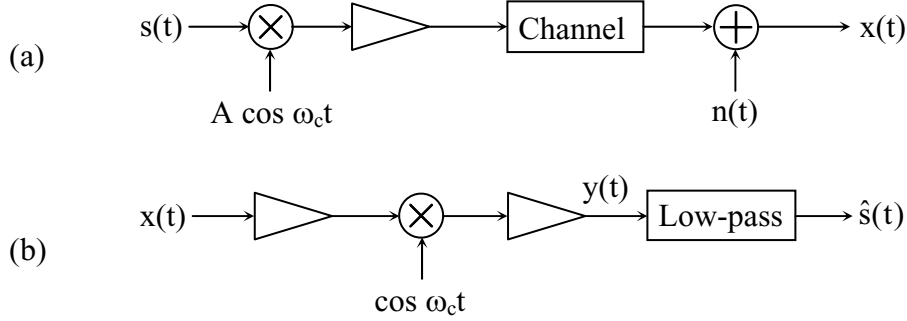


Figure 4.7-2: Transmitter and receiver for double-sideband synchronous carrier transmission.

The question of interest is how the signal-to-noise ratio of the detected output is related to the signal-to-noise ratio which would have resulted had we simply transmitted the signal $s(t)$ directly, adding $n(t)$. To evaluate this output signal-to-noise ratio, we need to define the noise more accurately as

$$n(t) = n_c(t) \cos \omega_c t - n_s(t) \sin \omega_c t \quad (4.7.3)$$

where the cosine and sine terms of the noise, $n_c(t)$ and $n_s(t)$, are slowly varying with bandwidths W over which the noise is white. As a result

$$\overline{n^2(t)} = \left[\overline{n_c^2(t)} + \overline{n_s^2(t)} \right] / 2 = \overline{n_c^2} = \overline{n_s^2} \quad (4.7.4)$$

$$y(t) = [A_c s(t) + n_c(t)] \cos^2 \omega_c t - n_s(t) \sin \omega_c t \cos \omega_c t \quad (4.7.5)$$

where the second term containing the product $\sin \omega_c t \cos \omega_c t$ is centered at $2\omega_c$, where it is removed by the low-pass filter. The first term is proportional to $\cos^2 \omega_c t$, which equals $(1 + \cos 2\omega_c t)/2$. Only the factor $1/2$ survives the low-pass filter, so that the low-pass filtered output $y_{\text{lpf}}(t)$ is

$$y_{\text{lpf}}(t) = \frac{1}{2} [A_c s(t) + n_c(t)] \quad (4.7.6)$$

The output signal-to-noise ratio $S_{\text{out}}/N_{\text{out}}$ can be found from the ratio of the powers associated with the signal and noise terms in (4.7.6) yielding

$$\begin{aligned} S_{\text{out}}/N_{\text{out}} &= \left[(A_c)^2 \overline{s^2(t)/2} \right] / \left[n_c^2(t) \right] \\ &= [P_c/2N_oW] \overline{s^2(t)} \end{aligned} \quad (4.7.7)$$

By convention we define the average signal power $\overline{s^2(t)}$ to be unity, and the carrier power P_c is $A_c^2/2$. Although in this modulation scheme the carrier is not transmitted as a separate intact sinusoid, we can nevertheless define the *carrier-to-noise ratio* (CNR) for this system as the ratio of the output signal power to the output noise power within the accepted bandwidth, where $\text{CNR}_{\text{DSBSC}} = S_{\text{out}}/N_{\text{out}}$.

An important variation of the previous modulation scheme is *single-sideband communications* (SSB). In this case we have assumed the carrier at the receiver is synchronous with that at the transmitter, which we can designate as SSBSC for its synchronous carrier⁷. This system differs from DSBSC only in that we filter out one of the transmitted sidebands, halving the transmitted bandwidth to W hertz. Since both the transmitted power and the noise power are each halved, the resulting signal-to-noise ratio and CNR are unchanged.

A more primitive but more widely used form of modulation is *amplitude modulation* (AM). In this case a carrier signal of peak amplitude A_c is modulated by the zero-mean signal $s(t)$, the magnitude of which is restricted to values less than unity in the expression

$$\begin{aligned} y(t) &= A_c [1 + m s(t)] \cos \omega_c t + n_c(t) \cos \omega_c t + n_s(t) \sin \omega_c t \\ &= \text{Re} \left\{ \underline{Y}(t) e^{j\omega_c t} \right\} \end{aligned} \quad (4.7.8)$$

where the modulation index m is also restricted to values less than unity, and the additive Gaussian white channel noise is again divided into its cosine and sine terms. The first term in (4.7.8) corresponds to a carrier frequency plus a double-sideband signal, as illustrated in Figure 4.7-3(c). This received signal $y(t)$ can also be usefully represented as a slowly varying complex phasor $\underline{Y}(t)$, as suggested in (4.7.8) and illustrated in Figure 4.7-3(b). In the limit where the signal-to-noise ratio is large, an envelope detector can be used which approximates the received signal as the magnitude of the phasor $\underline{Y}(t)$ minus a DC term. The resulting output signal plus noise $\underline{Y}(t)$ for a large signal-to-noise ratio is then approximately

$$|\underline{Y}(t)| \cong A_c [1 + m s(t)] + n_c(t) \quad (4.7.9)$$

⁷ Sometimes SC signifies “suppressed carrier”, which we also have here; in effect, no carrier is transmitted.

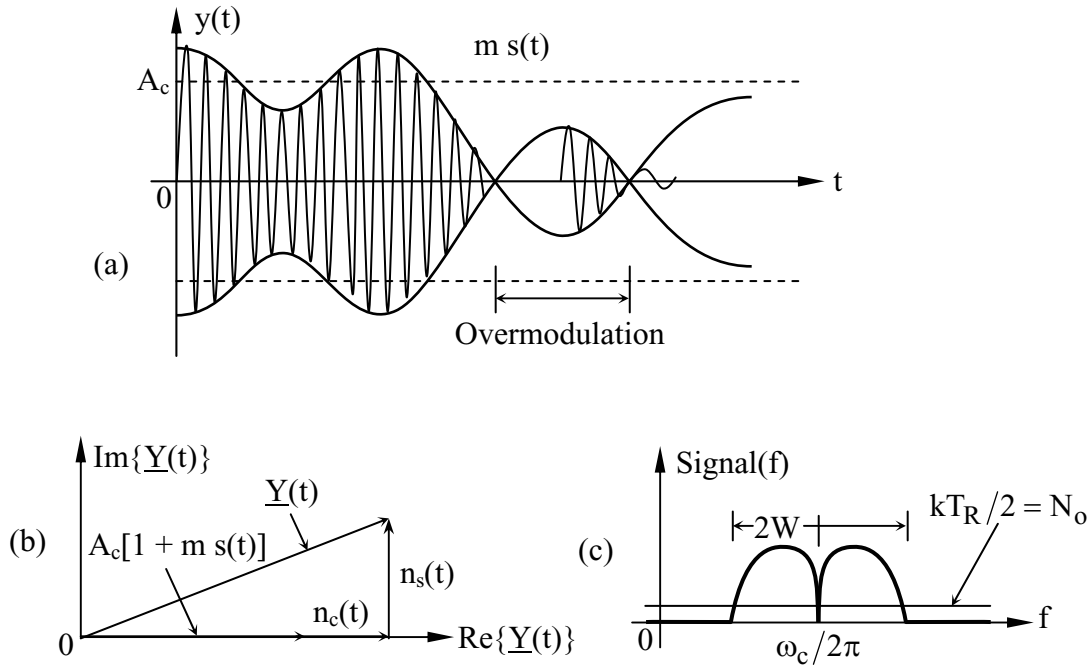


Figure 4.7-3: Amplitude modulated signals.

The signal-to-noise ratio S_{out}/N_{out} can be found directly from (4.7.9) to be

$$\frac{S_{out}}{N_{out}} \cong A_c^2 m^2 \overline{s^2(t)} / \overline{n_c^2(t)} = \frac{2P_c m^2 \overline{s^2(t)}}{4N_o W} \quad (4.7.10)$$

The output signal-to-noise ratio can be contrasted to the signal-to-noise ratio at the input, where

$$\frac{S_{in}}{N_{in}} = \frac{(A_c^2/2) \overline{(1 + m s(t))^2}}{4WN_o} \quad (4.7.11)$$

An effective noise figure for an AM system relative to an ideal one can be computed by taking the ratio of these input and output signal-to-noise ratios, analogous to the noise figure for an amplifier. This noise figure for AM is

$$F_{AM} = \frac{S_i/N_i}{S_{out}/N_{out}} = \frac{1 + m^2 \overline{s^2(t)}}{2m^2 \overline{s^2(t)}} \geq \frac{1 + 0.5}{1} = 3/2 \quad (4.7.12)$$

where it is slightly larger than unity because of the power wasted in the carrier. If the product $m^2 \overline{s^2}$ associated with the modulation index m and the signal power $\overline{s^2}$ is much less than unity, the noise figure F_{AM} deteriorates further as the percentage of power wasted in the carrier increases. As a result, AM broadcasters attempt to use the largest possible modulation indexes short of saturating the transmission system, whereupon the system is *overmodulated*, as suggested in Figure 4.7-3(a), and serious distortion results.

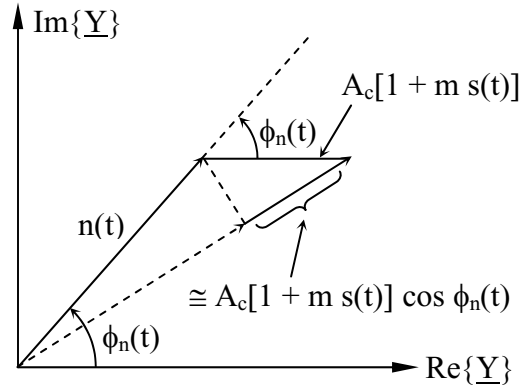


Figure 4.7-4: Amplitude modulation phasors in the limit of small S/N.

In addition to the distortion associated with possible overmodulation, AM systems also suffer distortion at low signal-to-noise ratios, as suggested in Figure 4.7-4 where the noise phasor $\underline{n}(t)$ is large compared to the received signal vector. The envelope detector again recovers the superposition of noise plus a signal term.

$$\text{Output} = \text{noise} + A_c m s(t) \cos \phi_n(t) \quad (4.7.12)$$

where the signal term is now multiplied by $\cos \phi_n(t)$. Since $\phi_n(t)$ varies randomly over 2π , this zero-mean multiplicative noise completely destroys the intelligibility of the received signal. In order for intelligibility to be preserved in AM systems, we want the input signal-to-noise ratio to be greater than approximately 10 dB, the *AM threshold*.

A third important class of modulation schemes differs from the foregoing in two important ways. First, the amplitude of the carrier remains constant and conveys no information, and secondly, the bandwidth of the transmitted signal can exceed the bandwidth of the original input. Both *phase modulation* (PM) and *frequency modulation* (FM) are of this type, although bandwidth expansion is usually restricted to FM. For both PM and FM the transmitted signal $x(t)$ is given by

$$x(t) = A_c \cos[\omega_c t + \phi(t)] \quad (4.7.13)$$

where the signal phase $\phi(t)$ is directly proportional to the input signal $s(t)$ for phase modulation. The signal phase for PM is

$$\phi(t) = K's(t) \quad (4.7.14)$$

For FM the phase is proportional to the integral of the input signal:

$$\phi(t) = 2\pi K \int^t s(\tau) d\tau \quad (4.7.15)$$

For FM the frequency offset Δf from the nominal carrier frequency is therefore directly proportional to the input signal $s(t)$. This frequency offset in radians per second equals $d\phi/dt = 2\pi Ks(t)$. A typical FM or PM waveform is illustrated in Figure 4.7-5 together with a typical receiver.

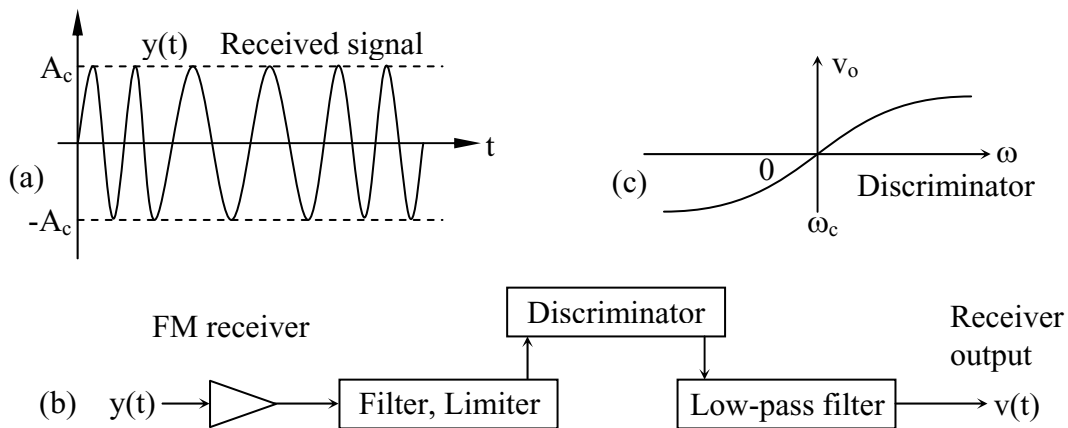


Figure 4.7-5: FM/PM signal and receiver.

After amplification the receiver selects the band of interest and limits its amplitude to a constant in one of several possible ways. This limiting process makes FM systems more immune to channel fading and to static and other forms of burst noise. Its output is a sinusoid of slowly varying frequency which passes to the discriminator, the output of which, v_o , is frequency dependent, as suggested in Figure 4.7-5(c). In this case the output voltage from the discriminator v_o is zero when the input frequency equals the carrier frequency, and is positive for frequencies above ω_c and negative for frequencies below. The output of the discriminator is low-pass filtered to obtain the desired bandwidth, which is adequate to represent the desired input signal.

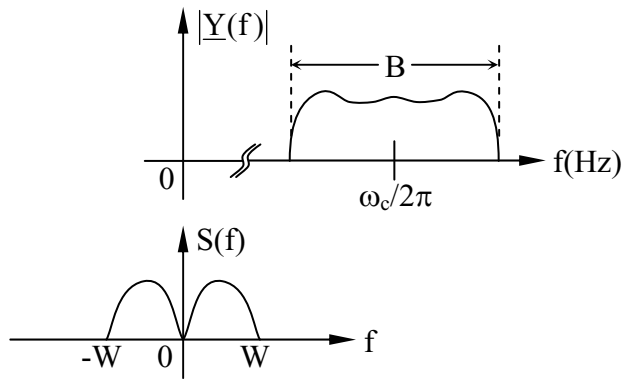


Figure 4.7-6: Spectra for FM and PM signals.

The bandwidth expansion properties of FM and PM are suggested in Figure 4.7-6, which shows how the input signal spectrum $S(f)$ at baseband has a bandwidth of W Hz. The received phasor $\underline{Y}(f)$ is centered at a carrier frequency of $\omega_c/2\pi$ Hz and has a total bandwidth B Hz. This total bandwidth B can be no less than $2W$, and is given by

$$B \cong 2W(1 + \beta^*) \quad (4.7.16)$$

where

$$\beta^* \equiv K/W \quad (4.7.17)$$

Thus the bandwidth B is roughly bounded by the limits

$$2W < B \lesssim 2W(1 + \beta^*) \lesssim 2K \quad (4.7.18)$$

where the FM bandwidth expansion factor β^* can vary between zero and factors of ten or more.

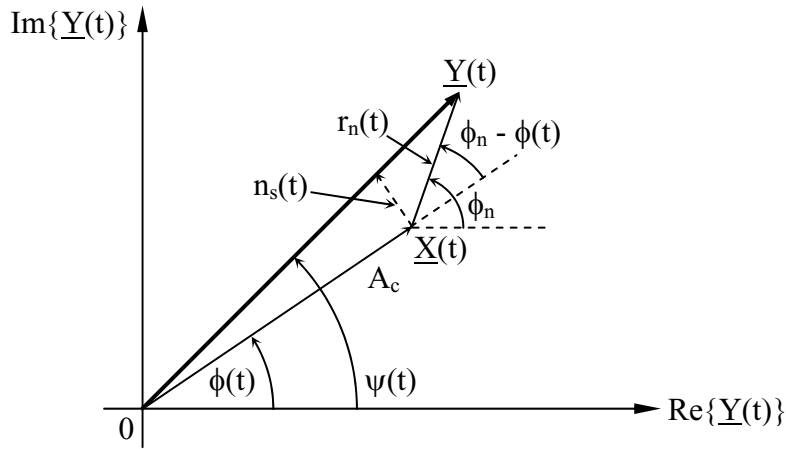


Figure 4.7-7: FM/PM phasor for large signal-to-noise ratios.

Within the bandwidth of interest the received signal $\underline{Y}(t)$ can be regarded as the phasor sum of the transmitted signal $\underline{X}(t)$ and the phasor equivalent of the received noise $r_n(t)$. How these phasors add for large signal-to-noise ratios is suggested in Figure 4.7-7, where the slowly varying received phasor $\underline{Y}(t)$ approximately equals the transmitted slowly varying phasor $\underline{X}(t)$. These phasors are defined so that $y(t) = \text{Re}\{\underline{Y}(t)e^{j\omega_c t}\}$, and $x(t) = \text{Re}\{\underline{X}(t)e^{j\omega_c t}\}$.

For both PM and FM systems, the desired phase $\phi(t)$, from which we recover the original input signal $s(t)$, is corrupted by additive noise to yield a total phase $\psi(t)$, where

$$\psi(t) \cong \phi(t) + \frac{r_n(t) \sin(\phi_n - \phi)}{A_c} \equiv \phi(t) + \frac{n_s(t)}{A_c} \quad (4.7.19)$$

The recovered signal for a PM system is simply related to $\psi(t)$ as

$$v(t) = \psi(t) = K's(t) + n_s(t)/A_c \quad (4.7.20)$$

and the recovered output voltage for FM receivers is

$$v(t) = (d\psi(t)/dt)(1/2\pi) = Ks(t) + (dn_s/dt)(1/2\pi A_c) \quad (4.7.21)$$

where we recall

$$\frac{d\psi}{dt} = \frac{d\phi}{dt} = \frac{d}{dt} \left[2\pi K \int^t s(t) dt \right] = 2\pi Ks(t) \quad (4.7.22)$$

$$n(t) = n_c(t)\cos\omega_c t - n_s(t)\sin\omega_c t \quad (4.7.23)$$

$$\overline{n^2(t)} = \overline{n_c^2(t)} = \overline{n_s^2(t)} = 2N_o B \quad (4.7.24)$$

where we have normalized the noise power spectral density N_o to equal $kT_R/2$.

The output signal-to-noise ratio for PM systems can be found from (4.7.20) by taking the ratios of the signal and noise powers. That is,

$$\frac{S_{out}}{N_{out}} = \frac{K'^2 \overline{s^2(t)}}{\left[\overline{n_s^2(t)}/A_c^2 \right]} = K'^2 \overline{s^2(t)} \left[\frac{A_c^2/2}{2N_o W} \right] \quad (4.7.25)$$

where we note $\overline{n_s^2(t)} = 2N_o B = 4N_o W$ if $B = 2W$. More generally, for PM, $B \approx 2(K'+1)W$. Thus to maximize the signal-to-noise ratio for PM, we want to maximize the carrier power and the phase gain K' relative to the noise power spectral density N_o and the input signal bandwidth W . However, as K' increases, PM systems increasingly resemble FM.

The output signal-to-noise ratio for FM systems can be similarly found using (4.7.21);

$$\frac{S_{out}}{N_{out}} = K^2 \overline{s^2(t)} \left/ \left[\left(\frac{dn_s(t)}{dt} \right)^2 / (2\pi A_c)^2 \right] \right. \quad (4.7.26)$$

where we now need to evaluate the square of the time derivative of the noise, dn/dt . The average value of this term can be evaluated most easily by transforming to the spectral domain and integrating, using Parseval's theorem, which says that power computed in the time domain must equal power computed in the spectral domain. Since $n_s(t)$ and $\underline{N}_s(f)$ constitute a Fourier transform pair, it follows that $dn_s(t)/dt$ and $j\omega \underline{N}_s(f)$ also constitute a Fourier transform pair.

The corresponding noise power spectral density is $\omega^2 |\underline{N}_s(f)|^2 = (2\pi f)^2 2N_o$ since $|\underline{N}_s(f)|$ is flat over a band of B Hz. Thus the noise power spectrum increases with the square of the frequency deviation from the carrier frequency, as suggested in Figure 4.7-8, and the denominator of (4.7.26) can be computed by integrating over the spectral domain.

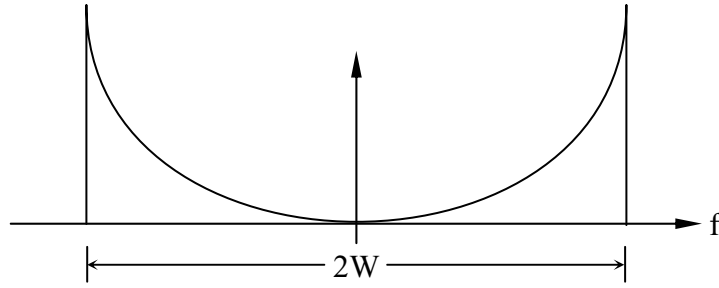


Figure 4.7-8: FM spectrum.

$$\frac{1}{(2\pi A_c)^2} \int_{-W}^W (2\pi f)^2 2N_o df = \frac{4N_o}{A_c^2} \frac{W^3}{3} \quad (4.7.27)$$

so that the output signal-to-noise ratio for an FM system is

$$\begin{aligned} \frac{S_{out}}{N_{out}} &= K^2 s^2 \left[\frac{A_c^2 3}{2} / 2N_o W^3 \right] = \frac{3P_c}{2N_o W} \beta^{*2} s^2 \\ &= 6[\text{CNR}] \beta^{*3} s^2 \end{aligned} \quad (4.7.28)$$

where the CNR here is $P_c/2BN_o$, and $B = 2W\beta^*$. In the wideband limit, sometimes called *wideband FM* (WBFM), the output signal-to-noise ratio becomes

$$\left. \frac{S_{out}}{N_{out}} \right|_{\text{WBFM}} = \left. \frac{S_{out}}{N_{out}} \right|_{\text{DSBSC}} \cdot 3\beta^{*2} \quad (4.7.29)$$

which reveals the significant advantages of FM over the highly efficient DSBSC system for which the output signal-to-noise ratio $S_{out}/N_{out} = P_c s^2 / 2N_o W$. For conventional FM radios with channel bandwidths of roughly 200 kHz, the FM signal-to-noise ratio advantage is approximately $3\beta^{*2} \cong 3 \times 5^2 \cong 19$ dB.

Further improvement in FM systems can be obtained with pre-emphasis and de-emphasis filters which amplify the input signal power roughly in proportion to its frequency deviation from zero and, more specifically, in proportion to the f^2 dependence of the noise; this pre-emphasis can be compensated at the receiver, suppressing some of the noise. Substantial improvements (perhaps 10 dB) in the output signal-to-noise ratio can be achieved in this way.

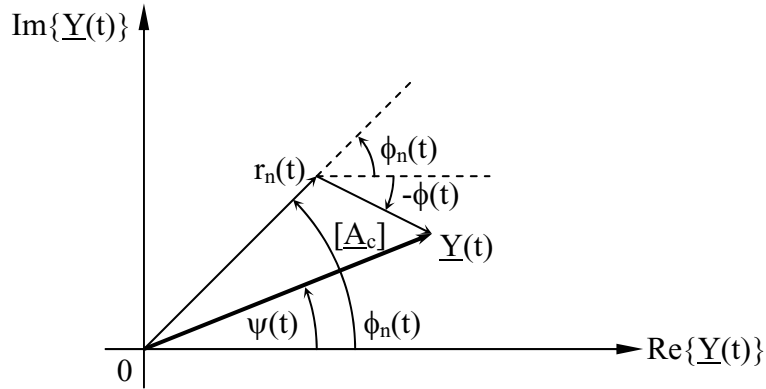


Figure 4.7-9: FM phasors below the FM threshold.

As is the case of AM, there is a signal level below which FM systems fail to deliver intelligible signals; this is called the *FM threshold*. The phasor diagram for the low SNR limit, analogous to Figure 4.7-7, is shown in Figure 4.7-9, where the signal carrier A_c is smaller than the noise phasor $r_n(t)$. In this case the phase angle we detect $\psi(t)$ has a contribution which obliterates the desired signal $\phi(t)$ because ϕ_n varies randomly over 2π radians:

$$\psi(t) \cong \phi_n + \frac{A_c}{r_n} \sin(\phi - \phi_n) \quad (4.7.30)$$

So destructive is ϕ_n that we must have carrier amplitudes much larger than the noise phasors, and the input signal-to-noise ratio must generally exceed 10 – 20 dB.

This FM threshold effect is illustrated in Figure 4.7-10, where the effective output signal-to-noise ratio is seen to substantially exceed that of the baseband signal-to-noise ratio as β increases, but this benefit is achieved only as the baseband signal-to-noise ratio rises above the FM threshold, which increases as β^* increases. These improvements in output signal-to-noise ratio as the bandwidth increases are analogous to the bandwidth-expansion channel-coding techniques discussed for digital systems in Section 4.5.

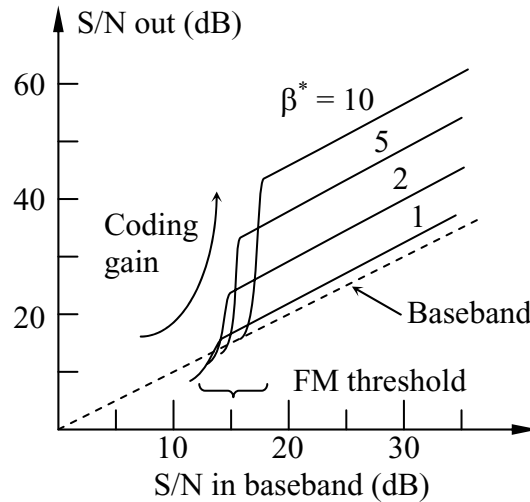


Figure 4.7-10: FM output SNR as a function of β^* .

4.8 COMMUNICATIONS SYSTEM DESIGN

Communications system design is bounded by constraints such as risk, time, signal power, receiver noise tolerance, cost and availability of technologies and bandwidth, and many others. In addition, these designs are driven by user requirements which vary considerably from case to case. For example, audio user requirements can range from low SNR to high SNR, and the desired dynamic range may also vary. In general, AM quality often provides ~25-35 dB output signal-to-noise ratios, which is satisfactory for voice communications, broadcast news, and similar purposes. FM-quality signals suitable for music and pleasant listening generally have SNR's on the order of 50 dB. Still higher audio quality might lead to a requirement for 55 dB SNR accompanied by 40 dB dynamic range, for a total SNR of 95 dB. The number of quantization levels required to yield 95 dB is approximately given by $L \approx 10^{95/20} \sigma \approx 56,000 \sigma$, where σ for uniform quantization is $12^{-0.5} = 0.29$ levels. This can be accommodated by a linear fifteen-bit quantizer having 32,000 levels, such as those used for recording music CD's.

Video quality is ~40-dB SNR (referenced to the maximum signal amplitude) in typical broadcast studios, and ranges from 20-35 dB in most homes with NTSC television receivers receiving over-the-air signals.

Bandwidth requirements for audio and video services also vary substantially. Intelligible voice generally requires 3-kHz bandwidth, and 6 kHz gives excellent service. Music can range to 15 kHz and more, where some young people can hear signals to 20 kHz. NTSC analog video and digitally coded high-definition video (HDTV) can each be transmitted in a 6-MHz bandwidth. Data requirements can range from fractions of a bit per second to millions or even

billions. Many applications are comfortably satisfied with ~10 kbps, compatible with our pervasive analog telephone system, and many more are satisfied by one form of digital telephone service, 136-kbps Integrated Services Digital Network—ISDN—with $2B + D = 2 \times 64$ kbps plus 8 kbps = 136 kbps full duplex (2 way). Even so, many consumers opt for still higher rates on cable TV or other wideband networks.

Source coding can reduce the data rate associated with these various types of signals, with greater compression often entailing some loss of signal quality. For example, uncompressed voice can be communicated well at 32-64 kbps, where 64 kbps might correspond to sampled voice at 8 kHz and 8 bits (~58 dB SNR). Compressed voice has been produced at data rates ranging between 9.6 kbps for high-quality voice down to 1.2 kbps for degraded services which still permit some speaker identification.

High-quality music uncompressed is comfortably accommodated at 1.5 Mbps, corresponding to two-channel stereo recorded with fifteen-bit accuracy and 40-kHz sampling, for example. Comparable music quality can be obtained with digital compression techniques that reduce the data rate to ~128 kbps; the losses are not perceptible to many listeners.

Standard NTSC video cameras convey approximately 500^2 pixels per frame at ~30 frames per second, where each frame consists of two fields at intervals of ~1/60 sec; one field conveys the even-numbered lines and the other conveys the odd-numbered lines. Because of the high degree of correlation between successive frames, such video data can be compressed with little degradation to 6 Mbps using motion-compensated inter-frame prediction together with lossy adaptive transform coding. Further reduction to ~1.5 Mbps yields video quality comparable to that of today's home VHS video cassette recorders or DVD players. For video conferencing purposes, a talking head imbedded in a stationary background can be satisfactorily coded using the same techniques at 384 kbps, and useful performance is obtained at 128 kbps, but artifacts are evident when the speakers move too rapidly. Acceptable color video can be coded at 56 kbps, and even 9.6 kbps yields a recognizable talking head, although somewhat jerky, blurred, and noisy, or presented in cartoon-like fashion, or in two levels—black and white.

Depending on the entropy of the data streams, typical applications can usually support lossless coding techniques yielding data rate reductions of a factor 2-4. In some cases where the data entropy is high, no useful compaction can be achieved, and in other cases where the data is highly redundant and only occasional transients are of interest, orders of magnitude improvement can be achieved.

As discussed earlier in Section 4.6, communication channels suffering from fading and burst noise can be significantly improved by digital coding and diversity in time, space, or polarization. Fading and burst noise can also be combated by interleaving the data stream and by error correction codes, yielding coding gains which can range to 10 dB and more.

Example 4.8.1

For music we desire $S/N = 50$ dB at the output. If bandwidth is free, what minimum received signal power and bandwidth is required for a baseband bandwidth of 20 kHz and a receiver noise temperature of 1000K?

Solution:

The desired S/N is enough larger than typical FM thresholds of 15 dB that wideband FM is appropriate, where β^* is chosen to leave some margin above the threshold, say 5 dB. Equation (4.7.29) suggests that $3\beta^{*2}$ should be $\sim 10^5/400$ where 10^5 is 50 dB and (S_o/N_o) is approximately 400 (26 dB, or 5 dB over the FM threshold; see Figure 4.7-10). Therefore, $\beta^* \cong 9$.

If $S/N_{DSBSC} \cong 400(26 \text{ dB}) = P_c \overline{s^2} / 2N_o W$, where $W = 20,000$, $N_o = kT_R/2$, and $\overline{s^2} = 0.5$, then:

$$P_c = 2N_o W \cdot 400 / 0.5 = 2 \cdot 1.38 \times 10^{-23} (1000/2) \cdot 2 \times 10^4 \cdot 400 / 0.5 = 2.2 \times 10^{-13} \text{ watts.}$$

The bandwidth needed is roughly $2W(1 + \beta^*) = 400$ kHz, from (4.7.16).