ELEMENTS OF DIGITAL COMMUNICATIONS

Instructor: Dr. Mustafa El-Halabi

Fall 2020

Theory is the first term in the Taylor series expansion of practice.

T. COVER

TABLE OF CONTENTS

Page

LIST OF FIGURES

Page

CHAPTER 1

PULSE CODE MODULATION

1.1 Digital Versus Analog Communication

Communications can be either *analog* or *digital*. We speak of *analog communication* when the transmitter sends one of a *continuum* of possible signals. The transmitted signal could be the output of a microphone. Any tiny variation of the signal can constitute another valid signal. More likely, in analog communication we use the source signal to vary a parameter of a *carrier signal*. As we have seen in earlier courses, two popular ways to do analog communication are *amplitude modulation* (AM) and *frequency modulation* (FM). In AM we let the carrier's amplitude depend on the source signal. In FM it is the carrier's frequency that varies as a function of the source signal.

We speak of digital communication when the transmitted sends one of a finite set of possible signals. For instance, if we communicate 1000 bits, we are communicating one out of 2^{1000} possible binary sequences of length 1000. To communicate our choice, we use signals that are appropriate for the channel at hand. No matter which signals we use, the result will be digital communication. One of the simplest ways to do this is that each bit determines the amplitude of a carrier over a certain duration of time. So the first bit could determine the amplitude from time 0 to T, the second from T to $2T$, etc. This is the simplest form of pulse amplitude modulation (PAM). There are many sensible ways to map bits to waveforms that are suitable to channel, and regardless of the choice, it will be a form of digital communication.

It is important to note that the meaning of *digital* versus *analog* communication should not be confused with their meaning in the context of electronic circuits. We can communicate digitally by means of analog or digital electronics and the same is true for analog communication.

The difference between analog and digital communication might seem to be minimal at this point, but actually it is not. It all boils down to the fact that in digital communication the receiver has a chance to exactly reconstruct the transmitted signal because there is a finite number of possibilities to choose from. The signals used by the transmitter are chosen to facilitate the receiver's decision. One of the performance criteria is the error probability, and we can design systems that have such a small error probability that for all practical purposes it is zero. The situation is quite different in analog communications. As there is a continuum of signals that the transmitter could possible send, there is no chance for the receiver to reconstruct an exact replica of the transmitted signal from the noisy received signal. It no longer makes sense to talk about error probability. If we say that an error occurs every time that there is a difference between the transmitted signal and the reconstruction provided by the receiver, then the error probability is always 1.

Example 1. *Consider a very basic transmitter that maps a sequence b_0, b_1, b_2, b_3 of numbers into a sequence $w(t)$ of rectangular pulses of a fixed duration. The i^{th} pulse has amplitude b_i. Is this analog or digital communication? It depends on the alphabet of b_i, $i = 0 \ldots, 3$. If it is a discrete alphabet, like $\{-1.3, 0, 9, 2\}$, then we speak of digital communication. In this case there are only m^4 valid sequences b_0, b_1, b_2, b_3, where m is the alphabet size (in this case $m = 4$), and equal many possibilities for $w(t)$.*

In principle, the receiver can compare the noisy channel output waveform against all these possibilities and choose the most likely sequence. If the alphabet is $\mathbb{R}$, then the communication is analog. In this case the noise will make it virtually impossible for the receiver to guess the correct sequence.

The following is an example that illustrates the difference between analog and digital communication. Compare faxing a text to sending an email over the same telephone line. The fax uses analog technology. It treats the document as a continuum of gray levels. It does not differentiate between text or images. The receiver prints a degraded version of the original. And if we repeat the operation multiple times by re-faxing the latest reproduction it will not take long until the result is dismal. Email on the other hand is a form of digital communication. It is almost certain that the receiver reconstructs an identical replica of the transmitted text.

1.1.1 Advantages of Digital Communications versus Analog Communications

Digital transmission of information has sufficiently overwhelming advantages that it increasingly dominates communication systems, and certainly all new designs. In computer-to-computer communication, the information to be transported is inherently digital. But information that at its source is inherently continuous time and continuous amplitude, like voice, music, pictures, and video, can be represented, not exactly but accurately, by collection of bits. Some of the advantages of digital communications over analog communication are listed below:

1. Digital communication transmits signals from finite alphabet, whereas digital communications transmits signals from an uncountable infinite alphabet.
2. Digital communication is more rugged than analog communication because it can withstand channel noise and distortion much better as long as the noise and distortion are within limits. Such is not the case with analog messages. Any distortion or noise, no matter how small, will distort the received signal. In digital communication, we can use error-correction codes to fight noise.
3. The greatest advantage of digital communication over analog communication, however, is the viability of regenerative repeaters in the former. In an analog communication system, a message signal, as it travels along the channel, grows progressively weaker, whereas the channel noise and the signal distortion, being cumulative, become progressively stronger. Ultimately, the signal, overwhelmed by noise and distortion, is mutilated. Amplification is of little help because it enhances the signal and the noise in the same proportion. Consequently, the distance over which an analog message can be transmitted is limited by the transmitted power. If a transmission path is long enough, the channel distortion and noise will accumulate sufficiently to overwhelm even a digital signal. The trick is to set up repeater stations along the transmission path a distances short enough to be bale to detect signal pulses before the noise and distortion have a chance to accumulate sufficiently. At each repeater station, the pulses are detected, and new, clean pulses are transmitted to the next repeater station, which, in turn, duplicates the same process. If noise and distortion are within limits, pulses can be detected correctly. This way the digital messages can be transmitted over long distances with greater reliability. In contrast, analog messages cannot be cleaned up periodically, and the transmission is therefore less reliable. The most significant error in PCM comes from quantizing.
4. Digital hardware implementation is flexible and permits the use of microprocessors, multiprocessors, digital switching, and large-scale integrated circuits.
5. Digital signals can be coded to yield extremely low error rates and high fidelity as well. It is also easy to encrypt whereas analog signals are hard to encrypt.
6. Digital signals are easy to compress, whereas analog signals are hard to compress.
7. It is easier and more efficient to multiplex several digital signals.
8. Digital communication is inherently more efficient than analog in realizing the exchange of SNR for bandwidth.
9. The cost of digital hardware continues to halve every two or three years, while performance or capacity doubles over the same time period (relentless exponential progress in digital technology).

1.2 Formatting Textual Data

The goal of the first essential signal-processing step, *formatting*, is to ensure that the message (or source signal) is compatible with digital processing. Transmit formatting is a transformation from source information to digital symbols. When data compression in addition to formatting is employed, the process is termed source coding.

Data already in digital format would bypass the formatting function. Textual information is transformed into binary digits by use of a coder. Analog information is formatted using three separate processes: sampling, quantization, and coding. In all cases, the formatting step results in a sequence of binary digits. These digits are to be transmitted through a baseband channel, such as a pair of wires or a coaxial cable. However, no channel can be used for the transmission of binary digits without first transforming the digits to waveforms that are compatible with the channel. For baseband channels, compatible waveforms are pulses. After transmission through the channel, the pulse waveforms are recovered (demodulated) and detected to produce an estimate of the transmitted digits; the final step, (reverse) formatting, recovers an estimate of the source information.

The original form of most communicated data (except from computer-to-computer transmissions) is either textual or analog. If the data consists of alphanumeric text, they will be character encoded with one of several standard formats (ASCII, EBCDIC, etc.). The textual material is thereby transformed into a digital format. Character coding, then, is the step that transforms text into binary digits (bits).

Textual messages comprise a sequence of alphanumeric characters. When digitally transmitted, the characters are first encoded into a sequence of bits, called bit stream or baseband signal. Groups of k bits can then be combined to form new digits, or symbols, from a finite symbol set or alphabet of $M = 2^k$ such symbols. A system using a symbol set of size M is referred to as M-ary system.

Example 2. *For $k = 1$, the system is termed binary, the size of the symbol set is $M = 2$, and the modulator uses one of the two different waveforms to represent the binary "one" and the other to represent the binary "zero". For this example, the bit rate and the symbol rate is the same.*

Example 3. *For $k = 2$, the system is termed quaternary or 4-ary, the size of the symbol set is $M = 4$, and at each time the modulator uses one out of 4 different waveforms that represents the symbol.*

If the information is analog, it cannot be character encoded as in the case of textual data; the information must first be transformed into a digital format. The process of transforming an analog waveform into a form that is compatible with digital communication system starts with sampling the waveform to produce a discrete pulse-amplitude-modulated waveform.

1.3 Analog Pulse Modulation

Pulse modulation involves communication using a train of recurring pulses. The key advantage in pulse modulation is that one can send multiple signals using Time Division Multiplexing. There are several pulse modulation techniques

1. Pulse Amplitude Modulation (PAM)
2. Pulse Width Modulation (PWM)
3. Pulse Position Modulation (PPM)
4. Pulse Code Modulation (PCM)

1.3.1 Pulse Amplitude Modulation (PAM)

Now that we understand the essence of the sampling process, we are ready to formally define *pulse-amplitude modulation* (PAM), which is the simplest and most basic form of analog pulse modulation techniques. In pulse-amplitude modulation (PAM), the amplitudes of regularly spaced pulses are varied in proportion to the corresponding sample values of a continuous

message signal; the pulses can be of a rectangular form or some other appropriate shape. Pulse-amplitude modulation as defined here is somewhat similar to natural sampling, where the message signal is multiplied by a periodic train of rectangular pulses. In natural sampling, however, the top of each modulated rectangular pulse is permitted to vary with the message signal, whereas in PAM it is maintained flat. The waveform of a PAM signal is illustrated in Fig.1.1.

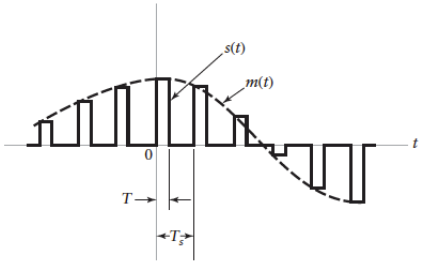


Figure 1.1: Flat-top sampling of a message signal.

The dashed curve in this figure depicts the waveform of the message signal and the sequence of amplitude-modulated rectangular pulses shown as solid lines represents the corresponding PAM signal $s(t)$. There are two operations involved in the generation of the PAM signal

1. Instantaneous sampling of the message signal every seconds, where the sampling rate is chosen in accordance with the sampling theorem.

2. Lengthening the duration of each sample, so that it occupies some finite value T.

In digital circuit technology, these two operations are jointly referred to as *sample-and-hold*. One important reason for intentionally lengthening the duration of each sample is to avoid the use of an excessive channel bandwidth, since bandwidth is inversely proportional to pulse duration. We will show that by using flat-top samples to generate a PAM signal, we introduce amplitude distortion as well as a delay of $T/2$. Hence, we should be careful in how long we make the sample duration T. The transmission of a PAM signal imposes rather stringent requirements on the amplitude and phase responses of the channel, because of the relatively short duration of the transmitted pulses. Also, PAM suffers from serious DC component.

Furthermore, it may be shown that the noise performance of a PAM system can never be better than direct transmission of the message signal. It should be evident from the waveform that a PAM signal has significant dc component and that the bandwidth required to preserve the pulse shape far exceeds the message bandwidth. Consequently you seldom encounter a single-channel communication system with PAM or, other analog pulse-modulated methods. But analog pulse modulation deserves attention for its major role in time-division multiplexing (TDM), data telemetry, and instrumentation systems.

Mathematically, we can represent the PAM wave $m(t)$ in Fig.1.1 as

$$s(t) = \sum_{n=-\infty}^{\infty} m(nT_s)h(t - nT_s) \tag{1.1}$$

where $h(t)$ is a rectangular pulse of unit amplitude and duration T defined as follows

$$h(t) = \begin{cases} 1 & 0 < t < T \\ 1/2 & t = 0, t = T \\ 0 & \textit{otherwise} \end{cases}$$

The instantaneously sampled version of the input signal $m(t)$ is given by

$$m_s(t) = \sum_{n=-\infty}^{\infty} m(nT_s)\delta(t - nT_s) \tag{1.2}$$

Convolving $m_s(t)$ with the pulse $h(t)$, we get

$$\begin{aligned} m_s(t) \star h(t) &= \int_{-\infty}^{\infty} m_s(\tau) h(t-\tau) d\tau \\ &= \int_{-\infty}^{\infty} \sum_{-\infty}^{\infty} m(nT_s)\delta(\tau - nT_s) h(t-\tau) d\tau \\ &= \sum_{-\infty}^{\infty} m(nT_s) \int_{-\infty}^{\infty} \delta(\tau - nT_s) h(t-\tau) d\tau \\ &= \sum_{-\infty}^{\infty} m(nT_s) h(t - nT_s) \\ &= s(t) \end{aligned} \tag{1.3}$$

Taking the Fourier transform of both sides of Eq. (1.3), we get

$$S(f) = M_s(f)H(f) = f_s H(f) \sum_{-\infty}^{\infty} M(f - mf_s) \tag{1.4}$$

Finally, suppose that $m(t)$ is strictly band-limited and that the sampling rate f_s is greater than the Nyquist rate. Then, passing $s(t)$ through a low-pass reconstruction filter, we find that the spectrum of the resulting filter output is equal to $M(f)H(f)$. This is equivalent to passing the original analog signal $m(t)$ through a low-pass filter of transfer function $H(f)$. Since the H(f) is given by

$$H(f) = Tsinc(fT)e^{-j\pi fT} \tag{1.5}$$

we can infer that by using PAM to represent an analog message signal, we introduce *amplitude distortion* as well as *delay* of $T/2$ (multiplication by $e^{-j\pi fT} = e^{-j2\pi f(T/2)}$ in frequency domain corresponds to a shift of $T/2$ in time domain).

1.3.2 Pulse Width Modulation (PWM) and Pulse Position Modulation (PPM)

In pulse-amplitude modulation, pulse amplitude is the variable parameter. Pulse duration is the next logical parameter available for modulation. In *pulse-duration modulation* (PDM), the samples of the message signal are used to vary the duration of the individual pulses. This form of modulation is also referred to as *pulse-width modulation* (PWM). In Fig.1.2(a), the modulating signal (a sawtooth signal) indicates the width of each pulse in the corresponding modulated pulse. See Fig.1.2(b). PDM is wasteful of power, in that long pulses expend considerable power during the pulse while bearing no additional information.

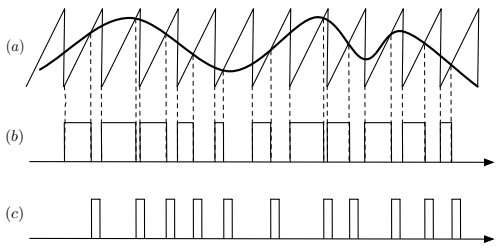


Figure 1.2: Illustration of two different forms of pulse-time modulation for the case of a saw-tooth modulating wave. (a) Modulating wave and Pulse carrier. (c) PDM wave. (d) PPM wave.

If this unused power is subtracted from PDM, so that only time transitions are essentially preserved, we obtain a more efficient type of pulse modulation known as *pulse-position modulation* (PPM). In PPM, the position of a pulse relative to its unmodulated time of occurrence is varied in accordance with the message signal, as illustrated in Fig.1.2(c).

Like PM and FM CW modulation, PPM has the advantage over PAM and PDM in that it has a higher noise immunity since all the receiver needs to do is detect the presence of a pulse at the correct time; the duration and amplitude of the

pulse are not important. Also, it requires constant transmitter power since the pulses are of constant amplitude and duration. It is widely used in fiber optic communications and deep space communications, but has the disadvantage of depending on transmitter-receiver synchronization.

In both PDM and PPM the amplitude remains constant, which offers robustness to non-linear amplitude distortion. Thus, PPM is the best form among the pulse analog modulation techniques. Recall that FM is the best form of continuous wave (CW) modulation. PPM and FM have a common feature represented by the fact that their noise performance, assessed by calculation of the figure of merit, is proportional to the square of the transmission bandwidth normalized with respect to the message bandwidth. The principle of improved noise performance with the increase in transmission bandwidth is called *Bandwidth-Noise trade off*. Hence, the square-law is the best we can achieve using CW and analog pulse modulation in terms of Bandwidth-Noise trade-off. Can an improved law of B-N trade-off be achieved? The answer can be found in pulse code modulation, which can give an exponential law for B-N trade off.

1.4 Quantization

Discrete sources are a subject of interest in their own right (for text, computer files, etc.) and also serve as the inner layer for encoding analog source sequences and waveform sources (see Fig.1.3). This section treats coding and decoding for a sequence of analog values. Source coding for analog values is usually called *quantization*. Quantization is the process of representing a large, possibly infinite, set of values with a smaller set.

Example 4. *(Real-to-integer conversion)*
Consider a source x of real number $[-10, 10]$ which is to be quantized using a quantizer $Q(x) = \lfloor x+0.5 \rfloor$. Hence, $[-10, 10] \rightarrow \{-10, -9, -8, \dots, -1, 0, 1, \dots, 8, 9, 10\}$.

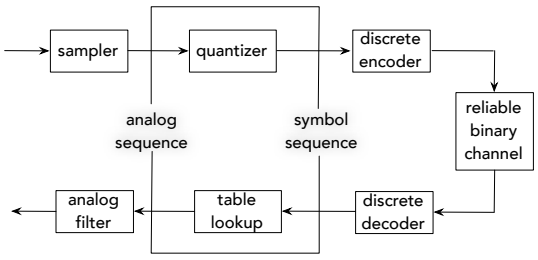


Figure 1.3: Encoding and decoding of discrete sources, analog sequence sources, and waveform sources.

The second step in digitizing an analog signal is to quantize the amplitude of the sampled signal $x_s(t)$. Quantization is the process of mapping a set of continuous amplitudes (infinite number of levels) into a finite number of discrete values. Obviously, this is a many-to-one mapping and, thus, in general we cannot recover exactly the analog signal from its quantized version. We can, however, through appropriate design, minimize this distortion. We will refer to the distortion introduced by quantization as *quantization noise*.
Let us assume that the analog signal to be quantized has amplitudes in the range $-V_p \leq V \leq V_p$ volts, and that we map all voltages in $[-V_p; V_p]$ into L discrete levels. The obvious question (which does not in general have an obvious answer) is: *how do we choose these L discrete levels such that the quantization noise is minimized?* In general, quantization is of two kinds: *scalar quantization* and *vector quantization*. We will mainly focus on scalar quantization which also consists of two types: *uniform quantization* and *non-uniform quantization*.

1.4.1 Scalar Quantization

A scalar quantizer partitions the set $\mathbb{R}$ of real numbers into M subsets $\mathcal{R}_1, \dots, \mathcal{R}_M$, called *quantization regions*. Assume that each quantization region is an interval. Each region $\mathcal{R}_j$ is then represented by a representation point $a_j \in \mathbb{R}$. When the

source produces a number $u \in \mathcal{R}_j$, that number is quantized into the point a_j. A scalar quantizer can be viewed as a function $\{q(u) : \mathbb{R} \to \mathbb{R}\}$ that maps analog real values u into discrete real values $q(u)$ where $q(u) = a_j$ for $u \in \mathcal{R}_j$. The quantization function q is a many-to-few, non-linear, irreversible, and deterministic mapping from the input u to the output $q(u)$.

An analog sequence $u_1, u_2, \ldots$ of real-valued symbols is mapped by such a quantizer into the discrete sequence $q(u_1), q(u_2), \ldots$. Taking $u_1, u_2, \ldots$, as sample values of a random sequence $U_1, U_2, \ldots$, the map $q(u)$ generates a random variable Q_k for each U_k; Q_k takes the value a_j if $U_k \in \mathcal{R}_j$. Thus each quantized output Q_k is a discrete random variable with the alphabet $\{a_1, \ldots, a_L\}$. The discrete random sequence $Q_1, Q_2, \ldots$, is encoded into binary digits, transmitted, and then decoded back into the same discrete sequence. For now, assume that transmission is error-free.

We first investigate how to choose the quantization regions $\mathcal{R}_1, \ldots, \mathcal{R}_L$, and how to choose the corresponding representation points. Initially assume that the regions are intervals, ordered as in Figure.1.4, with $\mathcal{R}_1 = (-\infty, b_1)$, $\mathcal{R}_2 = (b_1, b_2)$,..., $\mathcal{R}_L = (b_{L-1}, \infty)$. Thus an $L-$level quantizer is specified by $L-1$ interval endpoints, $b_1, \ldots, b_{L-1}$, and L representation points, $a_1, \ldots, a_L$.

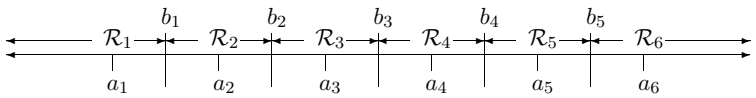


Figure 1.4: Quantization regions and representation points.

It needs to be mentioned at this stage that the quantization process introduces noise represented by the error or difference between the input signal u and the quantized output signal $q(u)$. This error is called *quantization noise*, and it introduces distortion. When this error is made sufficiently small, the original message signal and the quantized signal become practically indistinguishable to the human ear or eye depending on whether u is a voice or picture signal. This means that the analog message can be approximated by a signal constructed at discrete amplitudes which are selected on a minimum error basis from an available set. Clearly, the quantization noise can be reduced further by reducing the spacing between the adjacent quantization levels or step-size. An important question that needs to be answered in order to design a quantizer is the following: for a given value of L, how can the regions and representation points be chosen to minimize the mean-square quantization error?

To answer the previous question, the probabilistic model for $U_1, U_2, \ldots$ is important. For example, if it is known that each U_k is discrete and has only one sample value in each interval, then the representation points would be chosen as those sample values. Suppose now that the random variables $\{U_k\}$ are i.i.d analog random variables with the pdf $f_U(u)$. For a given set of points $\{a_j\}$, $Q(U)$ maps each sample value $u \in \mathcal{R}_j$ into a_j. The distortion of the quantization noise is given by the random variable $D = U - Q(U)$. The mean-square distortion, or mean-squared quantization error (MSQE) is then

$$MSQE = E[D^2] = E\left[(U - Q(U))^2\right] = \int_{-\infty}^{\infty} (u - q(u))^2 f_U(u) du = \sum_{j=1}^{L} \int_{\mathcal{R}_j} f_U(u)(u - a_j)^2 du \tag{1.6}$$

In order to minimize (1.6) over the set of a_j, it is simply necessary to choose a_j to minimize the corresponding integral, assuming that the regions are considered fixed. If the regions are not fixed, then we need to minimize (1.6) over the set of b_j as well.

In order to design an $L-$level quantizer, it is important to notice that the distortion equation contains in general $(2L-1)$ unknowns, L quantization levels a_j and $(L-1)$ quantization interval boundaries b_j. Taking the derivatives of the distortion given by Eq.(1.6) with respect to the $(2L-1)$ parameters and setting to zero every single equation, we obtain the following conditions for the optimum quantization levels a_j and quantization interval boundaries b_j:

The MSQE can be written as

$$MSQE = \int_{-\infty}^{b_1} (u - a_1)^2 f_U(u) du + \sum_{j=2}^{L-2} \int_{b_j}^{b_{j+1}} (u - a_{i+1})^2 f_U(u) du + \int_{b_{L-1}}^{+\infty} (u - a_L)^2 f_U(u) du \tag{1.7}$$

Differentiating (1.7) with respect to b_j yields

$$\frac{\partial}{\partial b_j} MSQE = f_U(b_j)\left[(b_j - a_j)^2 - (b_j - a_{j+1})^2\right] = 0 \tag{1.8}$$

which results in

$$b_j = \frac{1}{2}(a_j + a_{j+1}) \tag{1.9}$$

To determine the quantized values a_j, we differentiate (1.7) with respect to a_j and define $b_0 = -\infty$ and $b_L = +\infty$. Thus, we obtain

$$\frac{\partial}{\partial a_j} MSQE = \int_{b_{j-1}}^{b_j} 2(u - a_j) f_U(u) du = 0 \tag{1.10}$$

which results in

$$a_j = \frac{\int_{b_{j-1}}^{b_j} u f_U(u) du}{\int_{b_{j-1}}^{b_j} f_U(u) du} = \frac{\int_{b_{j-1}}^{b_j} u f_U(u) du}{P(b_{j-1} \le X \le b_j)} = \frac{\int_{b_{j-1}}^{b_j} u f_U(u) du}{p_i} \tag{1.11}$$

where $p_i \triangleq P(b_{j-1} \le X \le b_j)$.
Thus, for the optimum uniform quantizer, we have:

1. The optimum quantization interval boundaries are at the midpoints of the optimum quantization values

$$b_j = \frac{1}{2}(a_j + a_{j+1})$$

2. The optimum quantization values are the centroids of the quantization intervals

$$a_j = \frac{\int_{b_{j-1}}^{b_j} u f_U(u) du}{\int_{b_{j-1}}^{b_j} f_U(u) du}$$

Although the above rules are very simple, they do not result in analytical solutions to the optimal quantizer design. The usual method of designing the optimal quantizer is to start with a set of quantization regions and then using the second criterion, to find the quantized values. Then, we design new quantization regions for the new quantized values, and alternating between the two steps until convergence (when the distortion does not change much from one step to the next). This iterative numerical method is known as the **Max-Lloyd Algorithm**. Based on this method, one can design the optimal quantizer for various source statistics.

The Max-Lloyd algorithm

The *Max-Lloyd algorithm* is an algorithm for finding the endpoints $\{b_j\}$ and the representation points $\{a_j\}$ to meet the above necessary conditions. The algorithm is almost obvious given the necessary conditions; the contribution of *Lloyd* and *Max* was to define the problem and develop the necessary conditions. The algorithm simply alternates between the optimizations of the previous subsections, namely optimizing the endpoints $\{b_j\}$ for a given set of $\{a_j\}$, and then optimizing the points $\{a_j\}$ for the new endpoints. The *Max-Lloyd* algorithm is as follows. Assume that the number M of quantizer levels and the pdf $F_U(u)$ are given.

1. Choose an arbitrary initial set of M representation points $a_1 < a_2 < \cdots < a_M$.
2. For each j; $1 \le j \le M-1$, set $b_j = \frac{1}{2}(a_j + a_{j+1})$.
3. For each j; $1 \le j \le M$, set a_j equal to the conditional mean of U given $U \in (b_{j-1}, b_j)$ (where b_0 and b_M are taken to be $-\infty$ and $+\infty$ respectively).
4. Repeat steps (2) and (3) until further improvement in MSE is negligible; then stop.

The MSQE decreases (or remains the same) for each execution of step (2) and step (3). Since the MSQE is nonnegative, it approaches some limit. Thus if the algorithm terminates when the MSE improvement is less than some given $\epsilon > 0$, then the algorithm must terminate after a finite number of iterations.

The problem with the *Max-Lloyd algorithm* is that the algorithm might reach a local minimum of MSQE instead of the global minimum. This algorithm is a type of hill-climbing algorithm; starting with an arbitrary set of values, these values are modified until reaching the top of a hill where no more local improvements are possible. A reasonable approach in this sort of situation is to try many randomly chosen starting points, perform the *Max-Lloyd algorithm* on each and then take the best solution. This is somewhat unsatisfying since there is no general technique for determining when the optimal solution has been found.

1.4.2 Minimum Mean-Square Quantization Error (MMSQE)

Let us now derive an expression for the minimum distortion (MMSQE) incurred by an optimum scalar quantizer. Expanding the quadratic term in Eq.(1.6) we have

$$\begin{aligned} MMSQE &= \sum_{j=1}^{L} \int_{\mathcal{R}_j} (u - a_j)^2 f_U(u) du \\ &= \int_{-\infty}^{\infty} u^2 f_U(u) du + \sum_{j=1}^{L} a_j^2 \cdot \int_{\mathcal{R}_j} f_U(u) du - 2 \sum_{j=1}^{L} a_j \cdot \int_{\mathcal{R}_j} u f_U(u) du \\ &= \sigma_U^2 + \sum_{j=1}^{L} a_j^2 \cdot p_j - 2 \sum_{j=1}^{L} a_j \cdot \frac{\int_{\mathcal{R}_j} u f_U(u) du}{p_j} \cdot p_j \\ &= \sigma_U^2 - \sum_{j=1}^{L} a_j^2 \cdot p_j \end{aligned} \tag{1.12}$$

where σ_U^2 is the variance of U (we are assuming without loss of generality that U is zero-mean), and $p_j = \int_{\mathcal{R}_j} f_U(u) du$ is the probability of quantization values. The discrete random variable $Q(U)$ takes values from the set of L optimum quantization values with respective probabilities p_j. Then, the mean of $Q(U)$ is the same as the mean of U. This can be shown easily:

$$E[Q(U)] = \sum_{j=1}^{L} a_j \cdot p_j = \sum_{j=1}^{L} \frac{\int_{\mathcal{R}_j} u f_U(u) du}{p_j} \cdot p_j = \int_{-\infty}^{\infty} u \cdot f_U(u) du = E[U]$$

Since U is zero-mean, this means $Q(U)$ is zero-mean and thus the sum in the last equality in (1.12) is in fact the variance of $Q(U)$; thus the MMSQE (Minimum mean-square quantization error) for the optimum quantizer is

$$MMSQE = \sigma_U^2 - \sigma_{Q(U)}^2 \tag{1.13}$$

Example 5. *(Two-levels optimum scalar quantizer.)*
Consider a signal $x(t)$ having a PDF $f_X(x) = 1 - \frac{x}{2}$ for $0 \le x \le 2$. Design a 2-levels optimum quantizer for $x(t)$ and compute its minimum MSQE (MMSQE).

Solution. □

The dynamic range of the signal is equal to 2 Volts. Let the quantization regions be denoted by $\mathcal{R}_1 = [0, b]$ and $\mathcal{R}_2 = [1, 2]$, with respective quantization values a_1 and a_2. Hence, we can write the following equations corresponding to the design of an optimum quantizer

$$b = \frac{a_1 + a_2}{2}, \qquad a_1 = \frac{\int_0^b x(1 - x/2) dx}{\int_0^b (1 - x/2) dx}, \qquad a_2 = \frac{\int_b^2 x(1 - x/2) dx}{\int_b^2 (1 - x/2) dx}$$

Evaluating a_1 and a_2, we get

$$a_1 = \frac{-2b^3 + 6b^2}{-3b^2 + 12b}, \qquad a_2 = \frac{2b^3 - 6b^2 + 8}{3b^2 - 12b + 12}$$

Substituting in $2b = a_1 + a_2$, we get the following polynomial in b

$$b^4 - 10b^3 + 32b^2 - 40b + 16 = 0$$

The roots of the previous polynomials are: $b = 2$, $b = 3 - \sqrt{5} \simeq 0.76$, $b = 3 + \sqrt{5} \simeq 5.23$. The only feasible solution is $b = 3 - \sqrt{5}$, because $0 < b < 2$. This value of b yields

$$a_1 = 0.35, \qquad a_2 = 1.17$$

As a result, the MMSQE can be computed as follows

$$MMSQE = \int_0^{0.76} (x - 0.35)(1 - x/2)dx + \int_{0.76}^{2} (x - 1.17)(1 - x/2)dx = 0.0619$$

1.4.3 Uniform Quantization

This section analyzes the performance of uniform scalar quantizers. For a uniform scalar quantizer, every quantization interval $\mathcal{R}_j$ has the same length $|\mathcal{R}_j| = \Delta$. In other words, $\mathbb{R}$ (or the portion of $\mathbb{R}$ over which $f_U(u) > 0$), is partitioned into equal intervals, each of length Δ. See Fig. 1.5.

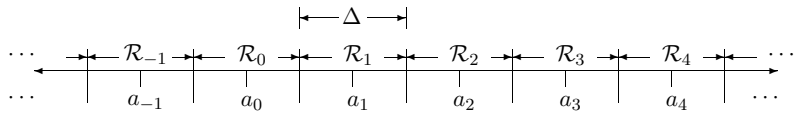


Figure 1.5: Uniform scalar quantizer.

A uniform or linear quantizer has all the quantization levels uniformly distributed in the interval $[-V_p; V_p]$ (except possibly the two intervals at the boundaries when the range of possible amplitudes is infinite). In other words, the interval $[-V_p; V_p]$ is subdivided into L quantization intervals, and the quantization amplitudes are assigned at the center of each quantization interval.

Example 6. *(Uniform quantizer)*
Consider a 4-level quantization $(L = 4)$. For a uniform quantizer, the quantization intervals and their corresponding quantized amplitudes are as shown in Fig. 1.6. Any amplitude x within a quantization interval is assigned to the voltage in the middle of that interval (except again possibly for the boundary intervals).

Thus, the input-output characteristic of a *uniform quantizer* is a stair case type characteristic and the spacing between two adjacent quantization levels a_{k-1} and a_k is called a "quantum" or "step-size", and is denoted by Δ. A look at *non-uniform* quantization will be taken later.

Definition 1. *(Quantizer Bit-rate)*
If the number of possible quantizer's outputs is L (L-level quantizer), then the quantizer bit rate *is $R = \lceil \log_2 L \rceil$. Alternatively, we refer to the $L-$level quantizer as an $R-$bit quantizer.*

Uniform quantizers are usually of two types:

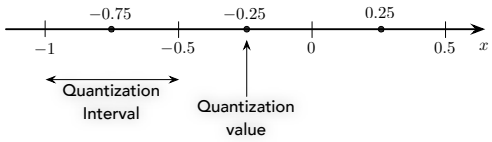


Figure 1.6: 4-Level uniform quantization.

1. *Midrise quantizer*: zero is not an output level. See Fig. 1.7(a).
2. *Midtread quantizer* (*Dead-Zone quantizer*): zero is an output level. See Fig. 1.7(b).

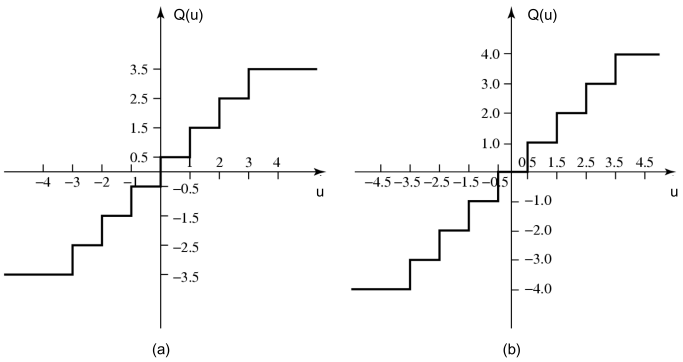


Figure 1.7: (a) Midrise uniform quantizer. (b) Midtread uniform quantizer

1.4.4 Quantization Noise

An important performance measure for quantizers is the *signal-to-quantization noise ratio* (SQNR), defined by

$$SQNR = \frac{P_U}{D} \tag{1.14}$$

where for stationary stochastic processes $U(t)$: $P_U = E[U^2]$ and $D = E[(U - Q(U))^2]$. The previous definitions can be extended to non-stationary processes:

$$P_U = \lim_{T\to\infty} \frac{1}{T} \int_{-T/2}^{T/2} E[U^2(t)]dt$$

$$D = \lim_{T\to\infty} \frac{1}{T} \int_{-T/2}^{T/2} E\left\{[U(t) - Q(U(t))]^2\right\} dt$$

For the optimum quantizer of a zero-mean source ($P_U = \sigma_U^2$), the SQNR becomes:

$$SQNR = \frac{\sigma_U^2}{\sigma_U^2 - \sigma_{Q(U)}^2} \tag{1.15}$$

Example 7. *(SQNR for a uniform quantizer)*
Let the input $x(t)$ be a sinusoid of amplitude V volts. It can be argued that all amplitudes in $[-V, V]$ are equally likely. Then,

if the step size of a uniform quantizer is Δ, the quantization error $d(x) = x - \hat{x}$ can be argued to be uniformly distributed in the interval $\left(-\frac{\Delta}{2}, \frac{\Delta}{2}\right)$. Then,

$$D = E[d^2(X)] = \frac{1}{\Delta}\int_{-\Delta/2}^{\Delta/2} y^2 dy = \frac{\Delta^2}{12} \tag{1.16}$$

Also, for a sinusoidal signal

$$P_X = \frac{V^2}{2}$$

Assuming we have an N-bit quantizer, then we have $L = 2^N$ quantization intervals that partition the 2V amplitude range. Thus,

$$\Delta = \frac{2V}{2^N}$$

and the SQNR in dB units is

$$SQNR = 10\log_{10}\left(\frac{P_X}{D}\right) = 10\log_{10}\left(\frac{3\times 4^N}{2}\right) = 6.02N + 1.76dB. \tag{1.17}$$

Therefore, it can be seen here that to reduce the quantization noise, Δ needs to be reduced. With the assumption that the quantization levels need to cover the entire dynamic range of the analog message, the reduction of Δ is equivalent to an increase in the number of quantization levels. Also, for every additional bit of quantization, we improve the SQNR performance by about 6 dB (not a small amount).
It needs to be noted here that each quantization level is to be represented by a binary codeword formed by a specific number of binary digits, or bits. This representation permits the transmission of the quantization levels in binary form. Let R be the number of bits per sample used in the construction of the binary code. Then, we can write: $L = 2^R$, under the assumption of a fixed length coding.
Also, the average power of the quantization noise; i.e., $E(d^2) = \frac{\Delta^2}{12}$, becomes:

$$E(d^2) = \frac{4V^2}{12L^2} = \frac{1}{3}V^2 2^{-2R} \Rightarrow SQNR = \frac{3P_X 2^{2R}}{V^2}$$

Example 8. *Consider an audio signal $m(t) = 3\cos(500\pi t)$. How many bits of quantization are needed to achieve an SQNR of at least 40 dB?*

Solution.

$$SQNR = \frac{P}{(\Delta^2/12)} = \frac{12\times 4.5}{\Delta^2} = \frac{54}{\Delta^2}$$

Since $SQNR \geq 10^4$, hence $\Delta \leq 7.35\times 10^{-2}$. Since $\Delta = \frac{6}{2^N}$, then $2^N > 81.6$. Choosing $N = 7$ will achieve an SQNR of at least 40 dB. □

Example 9. *(Darkening and contouring effect of image quantization.)*

1.4.5 Non-Uniform Quantization/Companding

As long as the statistics of the input signal are close to the uniform distribution, uniform quantization works fine. However, in coding for certain signals such as speech, the input distribution is far from being uniformly distributed. For a speech waveform,

Figure 1.8: The effect of increasing the quantization levels on reconstructing an image.

in particular, there exists a higher probability for smaller amplitudes and lower probability for larger amplitudes. If we use a uniform quantizer for such signals, the distortion will be high. In fact, speech signals are modeled as having a Gamma or Laplacian distribution, peaking about zero (that does not mean that 0 Volt has the peak highest probability).

Therefore, it makes sense to design a quantizer with more quantization regions at lower amplitudes and less quantization regions at larger amplitudes. The resulting quantizer will be a *nonuniform quantizer* having quantization regions of various sizes. See Fig. 1.9.

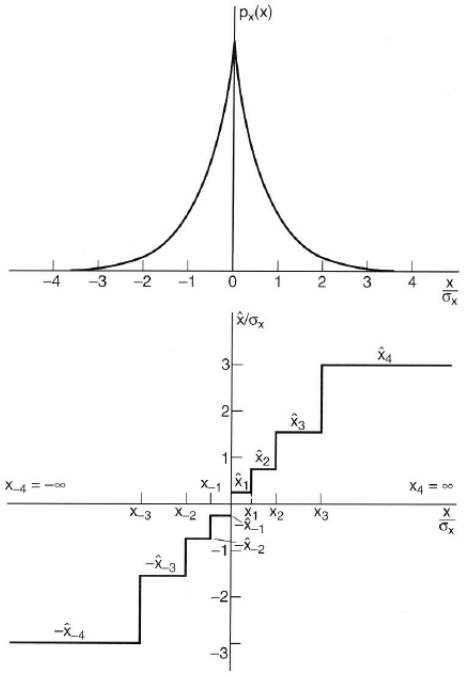


Figure 1.9: 3-bit non-uniform quantizer. (a) Laplacian pdf. (b) Input-output characteristic.

For example, the range of voltages covered by voice signals, from the peaks of loud talk to the weak passages of weak talk, is on the order of 1000 to 1. By using a non-uniform quantizer with the feature that the step size increases as the separation from the origin of the input-output amplitude characteristic is increased, the large end-step of the quantizer can take care of possible excursions of the voice signal into the large amplitude ranges that occur relatively infrequently. In other words, the weak passages that need more protection are favored at the expense of the loud passages. In this way, a nearly uniform percentage precision is achieved throughout the greater part of the amplitude range of the input signal, with the result that fewer steps

are needed than would be the case if a uniform quantizer were used. Fig. 1.10 illustrates a comparison between uniform and non-uniform quantization for a speech voltage signal.

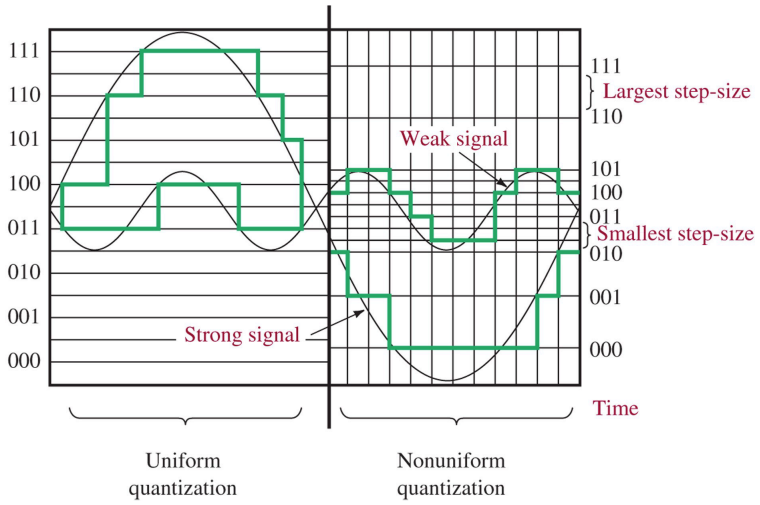


Figure 1.10: Comparison between uniform and non-uniform quantization for a speech voltage signal.

We saw above that uniform quantizers are easy to built and implement. However, their performance can be poor for practical sources. Non-uniform, optimum quantizers on the other hand have optimum performance, but their optimality assumes perfect knowledge of the source statistics and they are not robust to variations in these statistics. In many practical systems, the need for robustness and good performance (although not optimal) can be met by pre-distorting the source signals through an invertible non-linearity in order to make the amplitudes at the output of the non-linearity be more uniform. In this case, a simple uniform quantizer can be used. The process of pre-distorting the signal at the transmitter is known as (signal) compression and is performed using a *compressor*. To restore the reconstructed quantized and compressed signal to its correct amplitude levels, a device, called an *expander*, is used at the receiver. The expander law is the inverse of the compressor law. The two operations together are typically referred to as *companding*.

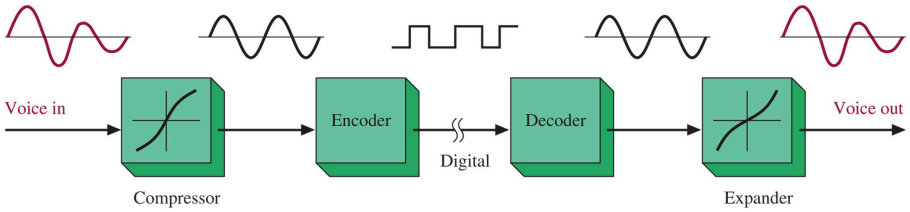


Figure 1.11: Companding of voice signal.

The use of a nonuniform quantizer is equivalent to passing the message signal through a compressor and then applying the compressed signal to a uniform quantizer. A particular form of compression law that is used in practice is the so called μ-*law* defined by

$$|v| = \frac{\ln(1+\mu|m|)}{\ln(1+\mu)} \tag{1.18}$$

where the logarithm is the natural logarithm; m and v are respectively the normalized input and output voltages, and μ is a positive constant. For convenience of presentation, the input to the quantizer and its output are both normalized so as to occupy a dimensionless range of values from zero to one, as shown in Fig.1.12(a); here we have plotted the μ-law for varying μ. Practical values of μ tend to be in the vicinity of 255. The case of uniform quantization corresponds to $\mu = 0$. For a given μ, the reciprocal slope of the compression curve, which defines the quantum steps, is given by the derivative of $|m|$ which respect

to $|v|$; that is,

$$\frac{d|m|}{d|v|} = \frac{\ln(1+\mu)}{\mu}(1+\mu|m|) \tag{1.19}$$

We see therefore that the μ-law is neither strictly linear nor strictly logarithmic, but it is approximately linear at low input levels corresponding to $\mu|m| << 1$, and approximately logarithmic at high input levels corresponding to $\mu|m| >> 1$.

The μ-law used for signal compression is used in the United States, Canada, and Japan. In Europe, another compression law known as the *A-law* is used for signal compression. The A-law is defined by

$$|v| = \begin{cases} \frac{A|m|}{1+\ln A} & 0 \le |m| \le \frac{1}{A} \\ \frac{1+\ln(A|m|)}{1+\ln A} & \frac{1}{A} \le |m| \le 1 \end{cases} \tag{1.20}$$

which is shown in Fig.1.12(b). Typical values of A used in practice tend to be in the vicinity of 100. The case of uniform quantization corresponds to $A = 1$. The reciprocal slope of this second compression curve is given by the derivative of $|m|$ with respect to $|v|$, as shown by

$$\frac{d|m|}{d|v|} = \begin{cases} \frac{1+\ln A}{A} & 0 \le |m| \le \frac{1}{A} \\ (1+\ln A)|m| & \frac{1}{A} \le |m| \le 1 \end{cases} \tag{1.21}$$

To restore the reconstructed quantized and compressed signal to its correct amplitude levels, a device, called an expander, is used at the receiver. The expander law is the inverse of the compressor law.

Note that the $\mu-$law is used in $T1$ digital telephony systems (using twisted cables) that achieve a bit rate of 1.544 Mbits/s, and the $A-$law is used in $E1$ digital telephony systems (using coaxial cables or twisted cables) that achieve a bit rate of 2.048 Mbits/s.

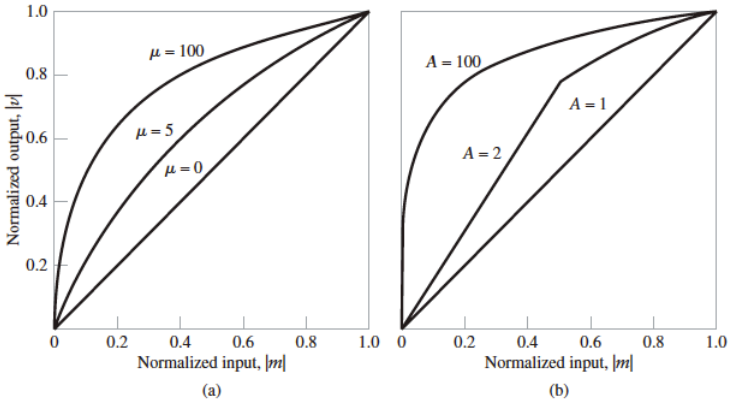


Figure 1.12: The μ-law and the A-law.

Example 10. *Consider a message signal $m(t)$ with a dynamic range between 2 and 8 volts.The signal $m(t)$ is inputed to a compressor using the μ-law with $\mu = 255$. Assume that the number of quantization levels used without compression is 64. Determine the improvement in SQNR that is made available by the use of the compressor assuming that the message signal average power prior to and after the use of the compressor did not change.*

Solution. The normalized input signal, $|m|$, is between $(2/8) = 0.25$ and $(8/8) = 1$ volts. The normalized output is between $|v| = \frac{\ln(1+255\times 0.25)}{\ln(1+255)} = 0.752$ and $|v| = 1$. Before compression, $(SQNR)_i = \frac{P_i}{(\Delta^2/12)} = \frac{12P_i}{(0.75/64)^2}$. After compression, $(SQNR)_0 = \frac{12P_0}{(0.248/64)^2}$. Since $P_i = P_0$, then the improvement in the SQNR is:

$$10\log_{10}\frac{(SQNR)_0}{(SQNR)_i} = 9.6dB$$

□

Example 11. *Consider a 16-level uniform quantizer designed for a signal with dynamic range* ± 10 *Volts. Consider an input signal of 1.2 V.*

1. *Find the step size* Δ.
2. *Find the minimum quantization error.*
3. *Find the quantization error for the input signal.*
4. *Assume the use of a* $\mu-$*law compander. Take* $\mu = 255$.
   (a) *Find the compressor's output.*
   (b) *Find the uniform quantizer's output.*
   (c) *Find the quantization error for the input signal.*

Solution.

$$L = 16, \quad 2A = 10$$

1. $\Delta = \frac{20}{16} = 1.25$
2. $-\Delta/2 = -0.625$
3. $d = \hat{m} - m = \frac{1.25}{2} - 1.2 = -0.575$
4. (a) $m = 1.2$, hence the normalized input is $|m| = \frac{1.2}{10} = 0.12$

$$|v| = \frac{ln(1 + 255(0.12))}{ln(1 + 255)} = 0.6227 \simeq 0.623$$

   Hence, the input to the uniform quantizer is $10(0.623) = 6.23$.

   (b) The uniform quantizer output is now $4.5\Delta = 4.5(1.25) = 5.625$.

   (c) 5.625 is now at the input of the expander:

$$\frac{1}{255}\left[(1 + 255)^{\frac{5.625}{10}} - 1\right] = 0.0848$$

   Multiplying back by the normalization factor, we get $0.084 \times 10 = 0.848$. Hence, the quantization error is $e = 0.848 - 1.2 = -0.352$.

□

1.5 Pulse Code Modulation (PCM)

With the sampling and quantization processes at our disposal, we are now ready to describe *pulse-code modulation* (PCM), which is the most basic form of digital pulse modulation. In PCM, a message signal is represented by a sequence of coded pulses, which is accomplished by representing the signal in discrete form in both time and amplitude.The basic operations performed in the transmitter of a PCM system are sampling, quantization,and encoding, as shown in Fig.1.13; the low-pass filter prior to sampling is included merely to prevent aliasing of the message signal. The quantizing and encoding operations are usually performed in the same circuit, which is called an *analog-to-digital converter*. The analog message is sampled and quantized to one of L levels; then each quantized sample is digitally encoded into an ℓ-bit ($\ell = \log_2 L$) codewords. For baseband transmission, the codeword bits will then be transformed to pulse waveforms. The essential features of binary PCM are shown in Fig.1.14. Assume that an analog signal $x(t)$ is limited in its excursions to the range -4 to $+4$. The step size between quantization levels has been set at 1 V. Thus, eight quantization levels are employed; these are located at $-3.5, -2.5, \cdots, +3.5$ V. We assign the code number 0 to the level at -3.5 V, the code number 1 to the level at -2.5 V, and so on, until the level at

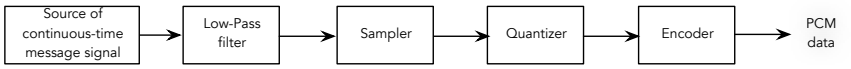


Figure 1.13: Basic steps in a PCM transmitter.

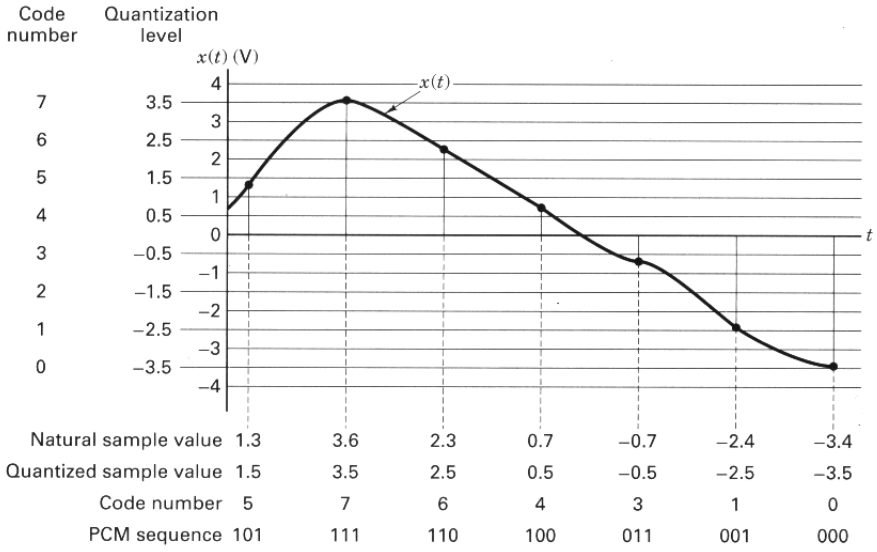


Figure 1.14: Natural sampling, uniform quantization and PCM

3.5 V, which is assigned the code number 7. Each code number has its representation in binary arithmetic, ranging from 000 for code number 0 to 111 for code number 7. Why have the voltage levels been chosen in this manner, compared with using a sequence of consecutive integers, $1, 2, 3, ...$? The choice of voltage levels is guided by two constraints. First, the quantile intervals between the levels should be equal; and second, it is convenient for the levels to be symmetrical about zero.

Note that each sample is assigned to one of eight levels or three-bit PCM sequence. Suppose that the analog signal is a musical passage, which is sampled at the Nyquist rate. And, suppose that when we listen to the music in digital form, it sounds terrible. What would we do to improve the fidelity? Increasing the number of levels will reduce the quantization noise. If we double the number of levels to 16, what are the consequences? In that case, each analog sample will be represented as a four-bit sequence. Will that cost anything? In a real-time communication system, the messages must not be delayed. Hence, the transmission time for each sample must be the same, regardless of how many bits represent the sample. Hence, when there are more bits per sample, the bits must move faster; in other words, they must be replaced by "skinnier" bits. The data rate is thus increased, and the cost is a greater transmission bandwidth. This explains how one can generally obtain better fidelity at the cost of more transmission bandwidth. Be aware, however, that there are some communication applications where delay is permissible. For example, consider the transmission of planetary images from spacecraft. The Galileo project, launched in 1989, was on such a mission to photograph and transmit images of the planet Jupiter. The Galileo spacecraft arrived at its Jupiter destination in 1995. The journey took several years; therefore, any excess signal of several minutes (or hours or days) would certainly not be a problem. In such cases, the cost of more quantization levels and greater fidelity need not be bandwidth; it can be time delay.

1.5.1 Regenerative Repeaters

The most important feature of a PCM system lies in the ability to control the effects of distortion and noise produced by transmitting a PCM signal over a channel. This capability is accomplished by reconstructing the PCM signal by means of a chain of regenerative repeaters located at sufficiently close spacing along the transmission route. Three basic functions are performed by a regenerative repeater: equalization, timing, and decision making.

The equalizer shapes the received pulses so as to compensate for the effects of amplitude and phase distortions produced by the transmission characteristics of the channel. The timing circuitry provides a periodic pulse train, derived from the received pulses; this is done for renewed sampling of the equalized pulses at the instants of time where the signal-to-noise ratio is a maximum. The sample so extracted is compared to a predetermined threshold in the decision-making device. In each bit interval, a decision is then made on whether the received symbol is a 1 or 0 on the basis of whether the threshold is exceeded or not. If the threshold is exceeded, a clean new pulse representing symbol 1 is transmitted to the next repeater. Otherwise, another clean new pulse representing symbol 0 is transmitted. In this way, the accumulation of distortion and noise in a repeater span is removed, provided the disturbance is not too large to cause an error in the decision-making process. Ideally, except for delay, the regenerated signal is exactly the same as the information-bearing signal that was originally transmitted.

The *repeater* is formed by a *matched filter* followed by a sampler and a decision-making device. In fact, this combination of devices is also used at the front end of the PCM decoder. The matched filter has the role of maximizing the output signal-to-noise ratio. It will be studied in the next chapter. The sampler, which is supplied with a timing circuit, samples the matched filter output at the time instants where the signal-to-noise ratio is maximum.

Example 12. *Consider the following bit stream received at the front end of a receiver:*

$$10110010101101001100010011000010101101 01$$

Assume that the bit 1 on the right-hand side of the sequence is the first received bit. Let also, the digit on the right of the codeword be transmitted first. Hence, the binary sequence is to be decoded as follows with $R = 8$*:*

$$10110101 \rightarrow 1 \times 2^7 + 0 \times 2^6 + 1 \times 2^5 + 1 \times 2^4 + 0 \times 2^3 + 1 \times 2^2 + 0 \times 2^1 + 1 \times 2^0 = 181$$

$11000010 \rightarrow 194$*;* $11000100 \rightarrow 196$*;* $10110100 \rightarrow 180$*;* $10110010 \rightarrow 178$*. The regeneration of the quantized signal can be shown as below (generating a PAM signal):*

1.6 Baseband Modulation

It was shown in previous section how analog waveforms are transformed into binary digits via the use of PCM. There is nothing "physical" about the digits resulting from this process. Digits are just abstractions- a way to describe the message information. Thus, we need something physical that will represent or "carry" the digits.

We will represent the binary digits with electrical pulses in order to transmit them through a baseband channel. See Fig.1.15. The sequence of electrical pulses having the pattern shown in Fig.1.15(b) can be used to transmit the information in the PCM bit stream, and hence the information in the quantized samples of a message.

The presence or absence of a pulse is a *symbol*. A particular arrangement of symbols used in a code to represent a single value of the discrete set is called a *codeword*. In a binary code, each symbol may be either of two distinct values, such as a negative pulse or positive pulse. The two symbols of the binary code are customarily denoted as 0 and 1. In practice, a binary code is preferred over other codes (e.g., ternary code) for two reasons:

1. The maximum advantage over the effects of noise in a transmission medium is obtained by using a binary code, because a binary symbol withstands a relatively high level of noise.

2. The binary code is easy to generate and regenerate.

Suppose that, in a binary code, each code word consists of R bits (the bit is an acronym for binary digit). Then R denotes the number of bits per sample. Hence, by using such a code, we represent a total of 2^R distinct numbers. For example, a sample quantized into one of 256 levels may be represented by an 8-bit code word.

1.6.1 PCM Waveforms Types

When pulse modulation is applied to a binary symbol, the resulting binary waveform is called a PCM waveform. There are several types of PCM waveforms that are described and illustrated in Fig.1.16; in telephony applications, these waveforms are

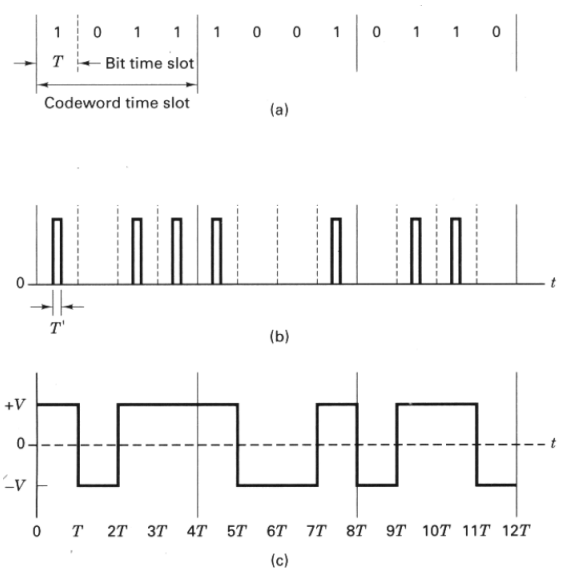


Figure 1.15: Example of waveform representation of binary digits. (a) PCM sequence. (b) Pulse representation of PCM. (c) Pulse wave-form

often called *line codes*. When pulse modulation is applied to non-binary symbol, the resulting waveform is called an M-ary pulse waveform, of which there are several types.The PCM waveforms fall into the following four groups:

1. Non-return to zero (NRZ)
2. Return to Zero (RZ)
3. Phase encoded
4. Multilevel binary

The NRZ group is probably the most commonly used PCM waveform. It can be partitioned into the following subgroups: NRZ-L (L for level), NRZ-M (M for mark), and NRZ-S (S for space). A binary one is represented by one voltage level and a binary zero is represented by another voltage level. There is a change in level whenever the data change from a one to a zero or from a zero to a one. With NRZ-M, the one, or *mark*, is represented by a change in level, and the zero, or *space*, is represented by no change in level. NRZ-L is used in digital logic circuits, NRZ-M is used primarily in magnetic tape recording.

The RZ waveforms consist of unipolar-RZ, bipolar-RZ, and RZ-AMI. These codes find application in baseband data transmission and in magnetic recording. With unipolar-RZ, a one is represented by a half-bit-wide pulse, and a zero is represented by the absence of a pulse. With bipolar-RZ, the ones and zeros are represented by opposite-level pulses that are one-half bit wide. There is a pulse present in each bit interval. RZ-AMI ("alternate mark inversion") is a signaling scheme used in telephone systems. The ones are represented by equal-amplitude alternative pulses. The zeros are represented by the absence of pulses.

The phase-encoded group consists of bi-ϕ-L (bi-phase-level), better known as *Manchester coding*; bi-ϕ-M (bi-phase-mark); bi-ϕ-S (bi-phase-space); and *delay modulation* (DM), or *Miller coding*. The phase-encoding schemes are used in magnetic recording systems and optical communications and in some satellite telemetry links. With bi-ϕ-L, a one is represented by a half-bit-wide pulse positioned during the first half of the bit interval; a zero is represented by half-bit-wide pulse positioned during the second half of the bit interval. With bi-ϕ-M, a transition occurs at the beginning of every bit interval. A one is represented by a second transition one-half bit interval later; a zero is represented by no second transition. With bi-ϕ-S, a transition also occurs at the beginning of every bit interval. A one is represented by no second transition; a zero is represented by a second transition one-half bit interval later. With delay modulation, a one is represented by a transition at the mid-point of the interval. A zero is represented by no transition, unless it is followed by another zero. In this case, a transition is placed at the end of the bit interval of the first zero. The previously shown line codes differ not only in their time domain representations but also in their power spectra as to whether they contain DC components represented by impulse functions (RZ contains

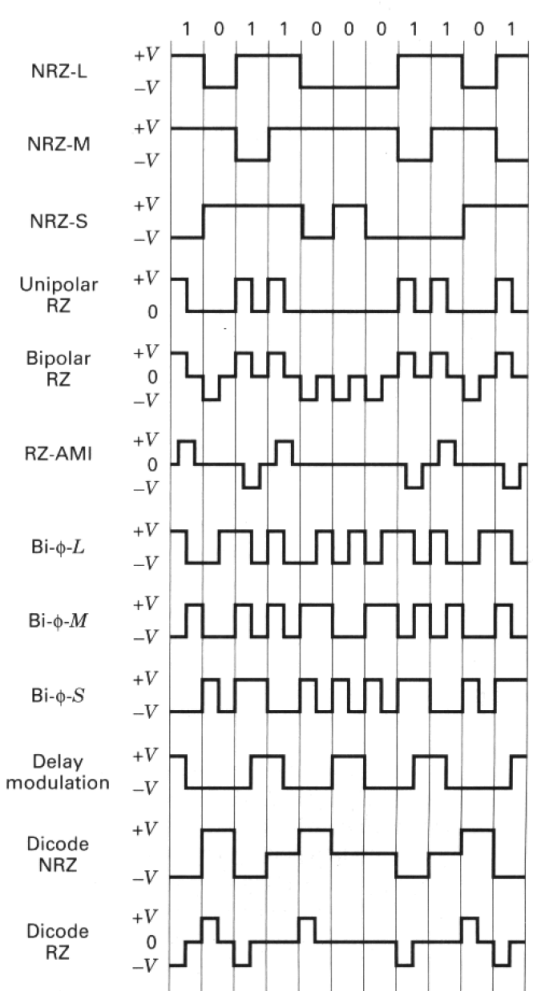


Figure 1.16: Various PCM Line Codes

DC components which cause a loss of power). Also, the line spectra differ in the required transmission bandwidth. Since the transmission bandwidth is inversely proportional to the bit duration, then RZ needs twice the bandwidth required for NRZ.

1.6.2 Bit Rate, Bit Duration and Bandwidth in PCM

The *bit rate* in PCM depends on the sampling rate, $1/T_s$, the number of quantization levels L, and the numbers of bits per sample R. The bit rate is given by

$$R_b = \frac{R}{T_s} \tag{1.22}$$

For a fixed length binary code, $L = 2^R$. Hence, $R = \log_2 L$ and $R_b = \frac{\log_2 L}{T_s}$. The bit duration is the inverse of the bit rate:

$$T_b = \frac{1}{R_b} = \frac{T_s}{R} \tag{1.23}$$

For the case of NRZ line codes, the transmission bandwidth is:

$$B_T = \frac{1}{T_b} = \frac{R}{T_s} \tag{1.24}$$

For the case of RZ line codes, the transmission bandwidth is twice as much:

$$B_T = \frac{2R}{T_s} \tag{1.25}$$

Of course the bandwidth computed as above considers the baseband binary transmission case without the involvement of a modulation carrier. Also, the bandwidth is defined by accounting for the significant frequency components of the different line codes. These components are assumed contained between 0 and the first or second zero crossing of the power spectra of the line codes with the frequency axis.

Example 13. *Consider an analog message of bandwidth 4KHz. The message is sampled at a rate equal to the Nyquist rate and quantized to 256 levels. Determine the bit rate, bit duration and the required transmission bandwidth under the use of binary ON-OFF (Unipolar RZ) baseband transmission technique.*

Solution. $R_b = \log_2 L/T_s = 2 \times 4 \times \log_2(256) = 64$ Kbits/s. Bit duration is $T_b = 1/R_b = 1/64000 = 15.6\mu s$. The transmission bandwidth is $B_T = 128$ KHz. □

Example 14. *A sinusoidal signal $m(t)$ band-limited to 3 KHz is sampled at a rate 33.33% higher than the Nyquist rate. The maximum quantization error is 0.5% of the peak amplitude. The quantized samples are binary coded. Find the minimum bandwidth of the channel required to transmit the encoded binary signal.*

Solution. The Nyquist rate is $f_N = 2 \times 3000 = 6$ KHz. The sampling rate is $f_s = 6000 \times 1.33 = 8$ KHz. The quantization step is Δ, and the maximum quantization error is $\Delta/2$. Therefore, $\Delta/2 = m_p/L = (0.5/100)m_p$. Hence, $L = 200$.
For binary coding, L must be a power of 2. Therefore, the next higher value of L that is a power of 2 is $L = 256 = 2^n$, giving $n = 8$ bits per sample. We require to transmit a total of $8 \times 8000 = 64$ Kbits/s.
Noiseless channels of bandwidth B Hz can transmit a signal of bandwidth B Hz. To reconstruct the signal, we need a minimum of $2B$ samples (Nyquist rate). Thus, a channel of B Hz can transmit $2B$ pieces of information, i.e., 2 pieces of information per Hz. Hence, in binary we can send 2 bits/s per Hz of bandwidth. Therefore, we require a minimum transmission bandwidth of $64/2 = 32KHz$. □

1.7 Virtues, Limitations and Modifications of PCM

The following advantages can be noted for PCM systems:

1. Good performance in the presence of channel noise and interference.
2. Efficient regeneration of the coded signal along the transmission path.
3. Efficient exchange of increased channel bandwidth for improved signal- to-noise ratio obeying an exponential law.
4. A uniform format for the transmission of different kinds of baseband signals. This allows the integration of these signals in a common network.

These advantages, however, are attained at the cost of increased system complexity and channel bandwidth. For instance, if we desire to to send a 4 KHz voice signal using PCM ($\mu-$law), it requires 8000 samples/sec times 8 bits/s sample or 64000 bits/sec. Hence, depending upon the type of line coding and pulse shaping used, the digitized voice signal could require a bandwidth of roughly 64 KHz, or 16 times that of the analog signal. Although certain advantages accrue with this bandwidth expansion, engineers began to wonder if this bit rate could be reduced without affecting the quality and intelligibility of the speech. Therefore, we will examine techniques for reducing the bit rate required to represent speech, images, or other messages, with some minimum or acceptable loss in fidelity. Signals such as speech and images are called *sources*, and the methods employed for bit rate reduction are variously said to be performing redundancy removal, entropy reduction, data compression, source coding, or source coding with a fidelity criterion. In the next sections we present DPCM and DM compression techniques which permit the removal of redundancies which are usually present in a PCM signal and this leads to a reduction in the bit rate of the transmitted data without a serious degradation in system performance.

In fact, the use of data compression techniques adds to the system complexity and, thus, to the cost of implementation. But, this cost increase is traded off for a reduced bit rate and therefore reduced bandwidth requirement. Although PCM involves the

use of complex operations, today they can be implemented using commercially available VLSI chips. If, however, the simplicity of implementation is desired, then *Differential Pulse Code Modulation (DPCM)* or *Delta Modulation (DM)* can be used as alternatives to PCM. In DPCM and DM, an intentional oversampling of the message signal is performed to allow for the use of a simple quantization strategy. The increase in transmission bandwidth was a reason for concern in the past. Today, however, it is not a real concern for two reasons: the first is the availability of wideband communication channels. This has been made possible by the deployment of communications satellites for broadcasting and ever increasing use of fiber optics for networking. The second is the use of data compression techniques.

1.8 Differential Pulse-Code Modulation (DPCM)

Differential Pulse-Code Modulation is a data compression technique aimed at reducing the bit rate as compared to PCM while maintaining the same signal quality. When a voice or video signal is sampled at a rate slightly higher than the Nyquist rate, the resulting sampled signal is found to exhibit high correlation between adjacent samples. The exception is the case when the spectrum of the process is flat within its bandwidth. The meaning of this high correlation is that, in an average sense, the signal does not change rapidly from one sample to the next. When these highly correlated samples are encoded, as in standard PCM system, the resulting encoded signal contains redundant information. Actually, some number of adjacent samples could be quantized to the same level and this leads to the generation of the same codeword in successive sampling intervals. This means that symbols that are not absolutely essential to the transmission of information are generated as a result of the encoding process. By removing this redundancy before encoding, we obtain a more efficient coded signal.

Now, if we know a sufficient part of a redundant signal, we may infer the rest, or at least make the most probable estimate. In particular, if we know the past behavior of a signal up to a certain point in time, it is possible to make some inference about its future values; such a process is commonly called *prediction*. Suppose that a message signal $m(t)$ is sampled at the rate $1/T_s$ to produce a sequence of correlated samples T_s seconds apart; which is denoted by $\{m(kT_s)\}$ (we will drop the T_s term in the following analysis just to simplify the notations). The fact that it is possible to predict future values of the signal $m(t)$ provides motivation for the *differential quantization* scheme shown in Fig.1.17.

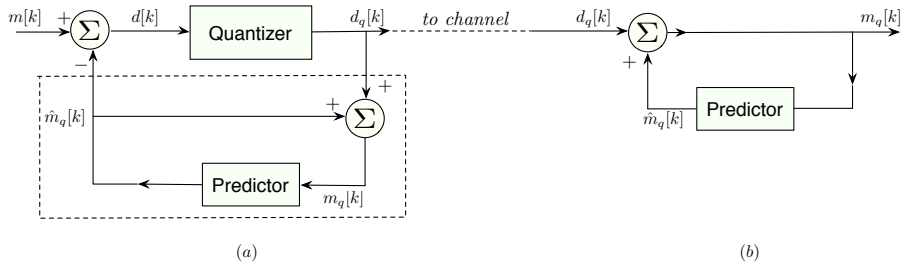


Figure 1.17: DPCM System. (a) Transmitter. (b) Receiver.

In DPCM, we do not transmit the present sample $m[k]$, but $d[k]$ (the difference between $m[k]$ and its predicted value $\hat{m}[k]$). At the receiver, we generate $\hat{m}[k]$ from the past samples values to which the received $d[k]$ is added to generate $m[k]$. There is; however, one difficulty in this scheme. In order to estimate $\hat{m}[k]$ at the receiver, we should have $m[k-1]$, $m[k-2]$, ... as well as $d[k]$, but instead, we have their quantized versions $m_q[k-1]$, $m_q[k-2]$, ... (because of the quantization before transmission).Hence, we cannot determine $\hat{m}[k]$, but we can determine $\hat{m}_q[k]$, the estimate of the quantized sample $m_q[k]$. For sure this will increase the error in reconstruction.

In order to handle this issue, a better strategy is to determine $\hat{m}_q[k]$ (instead of $m[k]$) at the transmitter from the quantized samples $m_q[k-1]$, $m_q[k-2]$, ... The difference $d[k] = m_q[k] - \hat{m}_q[k]$ is now transmitted using PCM instead of transmitting $m[k] - \hat{m}[k]$. At the receiver, we can generate $\hat{m}_q[k]$, and given $d[k]$, we can reconstruct $m_q[k]$, and finally get $m[k]$.

At the transmitter, for the predictor output to be $\hat{m}_q[k]$, the predictor input should be $m_q[k]$. How would we achieve that? Since the the difference $d[k] = m[k] - \hat{m}_q[k]$ is quantized to yield $d_q[k]$, then

$$\begin{aligned} d_q[k] &= Q\left(d[k]\right) \\ &= Q\left(m[k] - \hat{m}_q[k]\right) \\ &= m[k] + q[k] - \hat{m}_q[k] \\ &= d[k] + q[k] \end{aligned}$$

where $q[k]$ is the quantization error. The predictor output $\hat{m}_q[k]$ is fed back to its input so that the predictor input $m_q[k]$ is

$$\begin{aligned} m_q[k] &= \hat{m}_q[k] + d[k] + q[k] \\ &= m[k] - d[k] + d_q[k] \\ &= m[k] + q[k] \end{aligned}$$

This shows that $m_q[k]$ is a quantized version of $m[k]$. The quantized signal $d_q[k]$ is now transmitted over the channel. The receiver shown in Fig.1.17 is identical to the dotted portion of the transmitter. The inputs in both cases are also the same, viz., $d_q[k]$. Therefore, the predictor output must be $\hat{m}_q[k]$ (the same as the predictor output at the transmitter). Hence, the receiver output (which is the predictor input) is the also the same, viz., $m_q[k] = m[k] + q[k]$. This shows that we are able to to receive the desired signal $m[k]$ plus the quantization noise $q[k]$. This is the quantization noise associated with the difference signal $d[k]$, which is generally smaller than $m[k]$.

SQNR Improvement

To determine the improvement in DPCM over PCM, let m_p and d_p be the peak amplitudes of $m(t)$ and $d(t)$, respectively. If we use the same value of L in both cases, the quantization step Δ in DPCM is reduced by the factor d_p/m_p. Because the quantization noise power is $\Delta^2/12$, the quantization noise in DPCM reduces by the factor $(d_p/d_m)^2$, and the SQNR increases by the same factor. Moreover, the signal power is proportional to its peak value squared. Therefore, the *processing gain* G_p (SQNR improvement due to prediction) is

$$G_p = \frac{P_m}{P_d}$$

where P_m and P_d are the powers of $m(t)$ and $d(t)$, respectively. Now, for a given message signal, the average power P_m is fixed, so that G_p is maximized by minimizing the average prediction error power P_d. Accordingly, our objective should be to design the prediction filter so as to minimize P_d.

In the case of voice signals, it has been found that the improvement in DPCM over PCM can be as high as 25 dB. Alternately, for the same SQNR, the bit rate for DPCM could be lower than that for PCM by 3 to 4 bits per sample. Thus, telephone systems using DPCM can often operate at 32 kbits/s or even 24 kbits/s.

Example 15. *In a DPCM system, it is assumed that the dynamic range of the quantizer input has been reduced to 1/20 of the dynamic range of the message signal. Determine the relationship between the SQNR (in dB) in DPCM and PCM if both techniques use the same number of quantization levels.*

Solution.

$$\Delta_{PCM} = \frac{2V}{L} \Rightarrow (SQNR)_{PCM} = \frac{P_M}{\Delta_{PCM}^2/12} = \frac{12P_ML^2}{4V^2} = \frac{3P_ML^2}{V^2}$$

$$\Delta_{DPCM} = \frac{2V/12}{L} \Rightarrow (SQNR)_{DPCM} = \frac{P_M}{\Delta_{DPCM}^2/12} = \frac{12P_ML^2}{4V^2} \times 400 = \frac{3P_ML^2}{V^2} \times 400$$

Hence,

$$(SQNR)_{DPCM} = 400 \times (SQNR)_{PCM}$$

$$10\log_{10}(SQNR)_{DPCM} - 10\log_{10}(SQNR)_{PCM} = 26dB$$

□

1.9 Linear Prediction

The *linear predictor* used in DPCM to obtain the predicted value of the sample $m(n)$ using past p sample values or past quantized sample values is as shown in Fig.1.18. The predictor is formed by p unit delay elements, a set of multipliers involving the filter coefficients w_1, w_2,...,w_p and an adder to sum the coefficients-multiplied delayed inputs $m(n-1)$, $m(n-2)$,...,$m(n-p)$ to provide the output $\hat{m}(n)$.

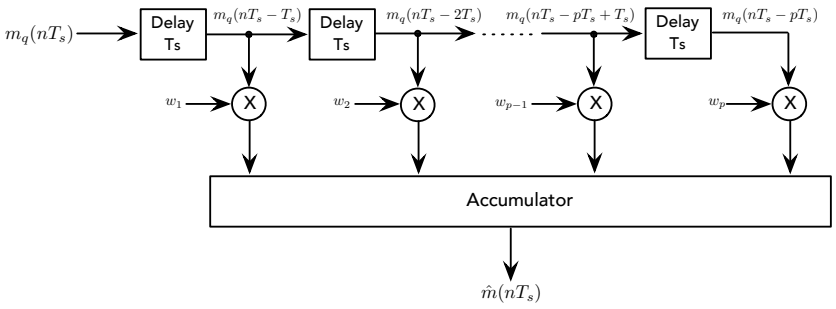


Figure 1.18: Tapped-delay linear prediction filter of order p.

The predictor output is given by

$$\hat{m}(n) = \sum_{k=1}^{p} w_k m(n-k) \tag{1.26}$$

The number of delay elements is called the *prediction order* (p). The difference $e(n) = m(n) - \hat{m}(n)$ is called the predictor error. The objective in the design of the predictor is to choose the filter coefficients so as to minimize the mean-square value of the error $e(n)$. Thus, the performance index $J = \sigma_E^2 = E[e^2(n)]$, which represents the average power of the error signal, needs to be minimized. If this is achieved, then the processing gain G_p in DPCM, which is equal to σ_M^2/σ_E^2 is maximized. This leads to SQNR improvements over a PCM system using the same number of quantization levels. The following is a derivation of the performance index

$$\begin{aligned}
\sigma_E^2 &= E\left[(m(n) - \hat{m}(n))^2\right] \\
&= E\left[\left(m(n) - \sum_{k=1}^{p} w_k m(n-k)\right)^2\right] \\
&= E\left[m^2(n)\right] - 2\sum_{k=1}^{p} w_k E\left[m(n)m(n-k)\right] + E\left[\sum_{k=1}^{p} w_k m(n-k) \sum_{j=1}^{p} w_j m(n-j)\right] \\
&= E\left[m^2(n)\right] - 2\sum_{k=1}^{p} w_k E\left[m(n)m(n-k)\right] + \sum_{j=1}^{p}\sum_{k=1}^{p} w_j w_k E\left[m(n-j)m(n-k)\right] \\
&= \sigma_M^2 - 2\sum_{k=1}^{p} w_k R_M(k) + \sum_{j=1}^{p}\sum_{k=1}^{p} w_j w_k R_M(k-j)
\end{aligned} \tag{1.27}$$

where σ_M^2 is the average power of the message signal $m(t)$ (assumed to be zero-mean), and R_M is the autocorrelation function of the $m(t)$. Taking the partial derivative of Eq.(1.27) with respect to the coefficients w_k and equating to zero

$$\frac{\partial \sigma_E^2}{\partial w_k} = -2R_M(k) + 2\sum_{j=1}^{p} w_j R_M(k-j) = 0 \quad \textit{for every } w_k$$

Hence,

$$\sum_{j=1}^{p} w_j R_M(k-j) = R_M(k), \quad k = 1, 2, ..., p \tag{1.28}$$

The set of p linear equations in (1.28), having p unknowns $w_1, w_2, ..., w_p$, are known as the *Yule-Walker* equations. A matrix form can also be used for the linear equations. Let $\underline{w}_0 = \begin{bmatrix} w_1 & w_2 & \dots & w_p \end{bmatrix}^T$, which is a $p \times 1$ coefficient vector,

$\underline{r}_M = \begin{bmatrix} R_M(1) & R_M(2) & \ldots R_M(p) \end{bmatrix}^T$ which is a $p \times 1$ autocorrelation vector, and

$$R_M = \begin{bmatrix} R_M(0) & R_M(1) & \cdot & \cdot & \cdot & R_M(p-1) \\ R_M(1) & R_M(0) & \cdot & \cdot & \cdot & R_M(p-2) \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ R_M(p-1) & R_M(p-2) & \cdot & \cdot & \cdot & R_M(0) \end{bmatrix}$$

which is a $p \times p$ autocorrelation matrix.
Hence, the matrix form representation of the *Yule-Walker* equations is

$$R_M \underline{w}_0 = \underline{r}_M \tag{1.29}$$

Assume that matrix R_M is non-singular, then R_M^{-1} exists. Then,

$$\underline{w}_0 = R_M^{-1} \underline{r}_M \tag{1.30}$$

Replacing Eq.(1.28) in Eq.(1.27) gives the minimum value of σ_E^2; i.e., $\sigma_{E,min}^2$

$$\begin{aligned} \sigma_{E,min}^2 &= \sigma_M^2 - 2\sum_{k=1}^{p} w_k R_M(k) + \sum_{k=1}^{p} w_k \sum_{j=1}^{p} w_j R_M(k-j) \\ &= \sigma_M^2 - 2\sum_{k=1}^{p} w_k R_M(k) + \sum_{k=1}^{p} w_k R_M(k) \\ &= \sigma_M^2 - \sum_{k=1}^{p} w_k R_M(k) \end{aligned} \tag{1.31}$$

Using the vector forms we re-write Eq.(1.31) as

$$\sigma_{E,min}^2 = \sigma_M^2 - \underline{w}_0^T \underline{r}_M \tag{1.32}$$

Using Eq.(1.30),

$$\begin{aligned} \sigma_{E,min}^2 &= \sigma_M^2 - \left[R_M^{-1} \underline{r}_M\right]^T \underline{r}_M \\ &= \sigma_M^2 - \underline{r}_M{}^T R_M^{-1\,T} \underline{r}_M \\ &= \sigma_M^2 - \underline{r}_M{}^T R_M^{-1} \underline{r}_M \end{aligned} \tag{1.33}$$

The quantity $\underline{r}_M^T R_M^{-1} \underline{r}_M$ is always positive and less than σ_M^2. As a result, $\sigma_{E,min}^2 < \sigma_M^2$, i.e., the processing gain $G_p = \sigma_M^2/\sigma_E^2$, in DPCM is always greater than 1.

Example 16. *Consider a second order linear predictor, such that $R_M(0) = 2$, $R_M(1) = 1.5$ and $R_M(2) = 1$. The message signal is assumed to be stationary and of zero-mean. Find the predictor coefficients and the ratio $\sigma_{E,min}^2/\sigma_M^2$.*

Solution. Using Eq.(1.30), we write $\begin{bmatrix} R_M(0) & R_M(1) \\ R_M(1) & R_M(0) \end{bmatrix} \begin{bmatrix} w_1 \\ w_2 \end{bmatrix} = \begin{bmatrix} R_M(1) \\ R_M(2) \end{bmatrix} \Rightarrow \begin{bmatrix} 2 & 1.5 \\ 1.5 & 2 \end{bmatrix} \begin{bmatrix} w_1 \\ w_2 \end{bmatrix} = \begin{bmatrix} 1.5 \\ 1 \end{bmatrix}$ This corresponds to solving the following system of equations $\begin{cases} 2w_1 + 1.5w_2 = 1.5 \\ 1.5w_1 + 2w_2 = 1 \end{cases}$
Thus, $w_1 = 0.857$ and $w_2 = -0.142$.
Using Eq.(1.33), $\sigma_{E,min}^2 = R_M(0) - \begin{bmatrix} 1.5 & 1 \end{bmatrix} \begin{bmatrix} 2 & 1.5 \\ 1.5 & 2 \end{bmatrix}^{-1} \begin{bmatrix} 1.5 \\ 1 \end{bmatrix} = 2 - 1.1428 = 0.8572.$
Hence, $\frac{\sigma_E^2}{\sigma_M^2} = \frac{0.8572}{2} = 0.4286.$ □

1.10 Delta Modulation (DM)

The exploitation of signal correlations in DPCM suggests the further possibility of oversampling a message signal (typically 4 times the Nyquist rate) to purposely increase the correlation between adjacent samples of the signal. This would permit the use of a simple quantization strategy for constructing the encoded signal. *Delta Modulation* (DM), which is the one-bit (two level) version of DPCM, is precisely such a scheme. In DM, we use a first-order predictor, which, as seen earlier, is just a time

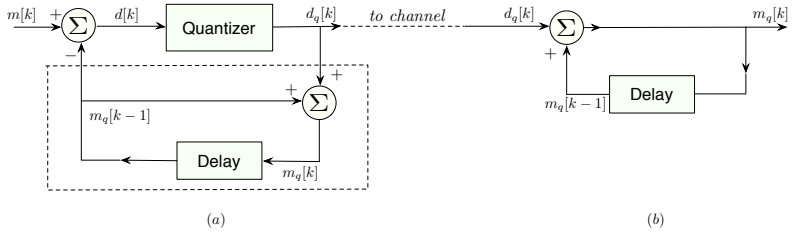


Figure 1.19: Delta Modulation. (a)Transmitter. (b) Receiver.

delay of T_s. Thus, the DM transmitter and receiver are identical to those of the DPCM, with a time delay for the predictor. See Fig. 1.19. From this figure, we obtain $m_q[k] = m_q[k-1] + d_q[k]$. Hence, $m_q[k-1] = m_q[k-2] + d_q[k-1]$ which yields

$$m_q[k] = m_q[k-2] + d_q[k] + d_q[k-1]$$

Proceeding iteratively in this manner, and assuming zero initial condition, that is, $m_q[0] = 0$, yields

$$m_q[k] = \sum_{m=0}^{k} d_q[m] \tag{1.34}$$

This shows that the receiver (demodulator) is an accumulator (adder). If the output $d_q[k]$ is represented by impulses, then the accumulator may be realized by an integrator because its output is the sum of the strengths of the input impulses (sum of the areas under the impulses). We may also replace the feedback portion of the modulator (which is identical to the demodulator) by an integrator. The demodulator output is $m_q[k]$, which when passed through a LPF yields the desired signal reconstructed from the quantized samples.

Fig. 1.20 shows a practical implementation of the delta modulator and demodulator.

The first-order predictor is replaced by a low-cost integrator circuit (such as an RC integrator). The modulator (Fig. 1.20a) consists of a comparator and a sampler in the direct path and an integrator-amplifier in the feedback path. Let us see how this delta modulator works.

The analog signal $m(t)$ is compared with the feedback signal (which serves as a predicted signal) $\hat{m}_q(t)$. The error signal $m(t) = m(t) - \hat{m}_q(t)$ is applied to a comparator. If $d(t)$ is positive, the comparator's output is a constant signal of amplitude E (or Δ), and if $d(t)$ is negative, the comparator's output is $-E$ (or $-\Delta$). Thus, the difference is a binary signal ($L = 2$) that is needed to generate a 1-bit DPCM sequence. The comparator's output is sampled by a sampler at a rate of f_s samples per second, where f_s is typically much higher than the Nyquist rate. The sampler thus produces a train of narrow pulses $d_q[k]$ (to simulate impulses) with a positive pulse when $m(t) > \hat{m}_q(t)$ and a negative pulse when $m(t) < \hat{m}_q(t)$. Note that each pulse is coded by a single binary pulse (1-bit DPCM), as required. The pulse train $d_q[k]$ is the delta-modulated pulse train signal (Fig. 1.20d), which tries to follow $m(t)$.

To understand how this works, we note that each pulse in $d_q[k]$ at the input of the integrator gives rise to a step function (positive or negative, depending on the pulse polarity) in $\hat{m}_q(t)$. If, for example, $m(t) > \hat{m}_q(t)$, a positive pulse is generated in $d_q[k]$, which gives rise to a positive step in $\hat{m}_q(t)$, trying to equalize $\hat{m}_q(t)$ to $m(t)$ in small steps at every sampling instant, as shown in Fig. 1.20c). It can be seen that $\hat{m}_q(t)$ is a kind of staircase approximation of $m(t)$. When $\hat{m}_q(t)$ is passed through a low-pass filter, the coarseness of the staircase in $\hat{m}_q(t)$ is eliminated, and we get a smoother and better approximation to $m(t)$.

The difference between the input $m(t)$ and the approximation $\hat{m}_q(t)$, as seen in Fig. 1.20c, is quantized into only two representation levels, namely $\pm\Delta$, corresponding to positive and negative differences. Thus, if the approximation falls below the signal at any sampling epoch, it is increased by Δ. If, on the other hand, the approximation lies above the signal, it is diminished

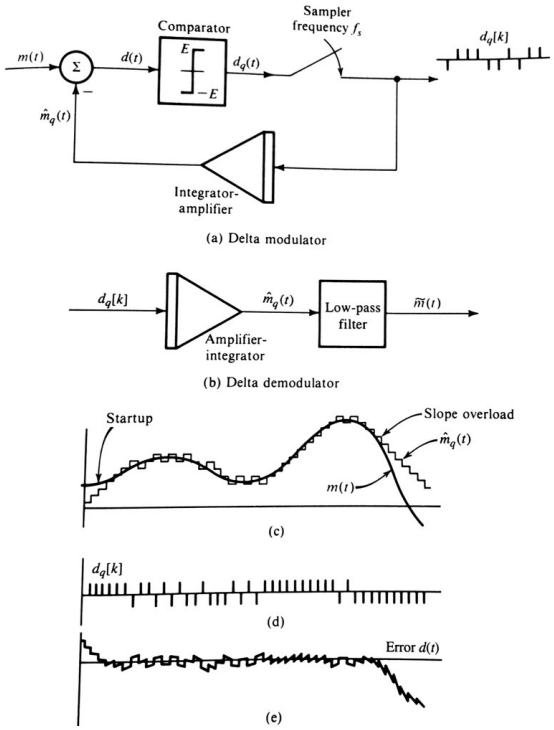


Figure 1.20: Practical Implementation of Delta Modulation.

by Δ. Provided that the signal does not change too rapidly from sample to sample, we find that the staircase approximation remains within $\pm\Delta$ of the input signal.

1.10.1 Quantization Noise

Delta modulation is subject to two types of quantization error:

1. *slope overload distortion*
2. *granular noise*

We first discuss the cause of slope overload distortion and then granular noise. We observe that $d_q[k] = m_q[k] - m_q[k-1]$ is the digital equivalent of integration in the sense that it represents the accumulation of positive and negative increments of magnitude Δ: $m_q[k] = m_q[k-1] \pm \Delta$. Hence, between one sample and the next, $m(t)$ increases or decreases by an amount equal to $|m[k] - m[k-1]|$, whereas $\hat{m}_q(t)$ increases or decreases by an amount equal to Δ. Hence, in order that $\hat{m}_q(t)$ varies as fast as $m(t)$, we need Δ or Δ/T_s to be of the order of $|m[k] - m[k-1]|$ or $|m[k] - m[k-1]|/T_s$. This ratio is the average slope of $m(t)$ between the time instants $(k-1)T_s$ and kT_s. Generally, if we consider a region of high slope for $m(t)$, then in this region we can require that Δ/T_s be of the order of the average derivative of $m(t)$ in this region. Or, to be on the safe side, Δ/T_s needs to be of the order of the maximum value of the derivative of $m(t)$ in the region. This condition is usually written as

$$\frac{\Delta}{T_s} \geq \max \left| \frac{dm(t)}{dt} \right| \tag{1.35}$$

If the above condition is not observed, then Δ is considered too small to make $\hat{m}_q(t)$ follow a steep segment of $m(t)$. The result of this phenomenon is called *slope overload distortion*, and is represented in the front end of Fig.1.21.

Note that since the maximum slope of the staircase approximation $m_q(t)$ is fixed by the step size Δ, increases and decreases in $m_q(t)$ tend to occur along straight lines, as illustrated in the front end of Fig.1.21. For this reason, a delta modulator using a fixed value for the step size Δ is often referred to as a *linear delta modulator*.

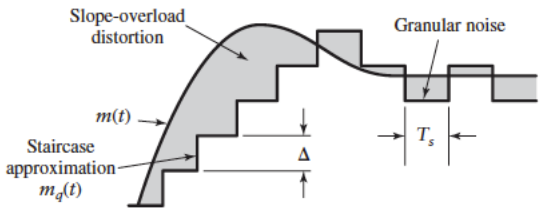


Figure 1.21: Illustration of quantization errors, slope-overload distortion and granular noise, in delta modulation.

In contrast to slope-overload distortion, *granular noise* occurs when the step size is too large relative to the local slope characteristic of the original message signal $m(t)$. This second situation causes the staircase approximation $m_q(t)$ to oscillate around a relatively flat segment of $m(t)$ which is illustrated in the back end of Fig.1.21. Granular noise in delta modulation may be viewed as the analog of quantization noise in pulse-code modulation.

So, it appears here that a large step size Δ, is needed to accommodate large variations in $m(t)$ and a small step size is needed for small variations in the message signal $m(t)$. Hence, the choice of the optimum size leads to the minimization of the mean-square value of the quantization noise should be obtained by a compromise between slope overload distortion and granular noise. This is to be done if we want to use fixed step size. Another solution is to make the delta modulator adaptive. That is, to make the step size varies in accordance with the variation of the input signal. The principle used in *adaptive delta modulation* (ADM) algorithms is as follows:

1. If successive errors e_n are of opposite polarity, then the DM is operating in its granular mode. The step-size should be reduced.
2. If successive errors e_n are of the same polarity, then the DM is operating in its slope-overload mode. The step-size should be increased.

Example 17. *Consider a sinusoidal message signal $m(t) = 3\cos(4000\pi t)$. The signal is sampled at 4 times the Nyquist rate. Determine the step-size Δ required so that if $m(t)$ is applied to a DM, no slope overload distortion would occur. Repeat the calculation for a signal sampled at 8 times the Nyquist rate.*

Solution. The condition for no slope overload distortion is

$$\frac{\Delta}{T_s} = \Delta \times f_s \geq \left|\frac{dm(t)}{dt}\right| = \max|12000\pi \sin(4000\pi t| \Rightarrow \Delta f_s \geq 12000\pi \Rightarrow \Delta \geq \frac{12000\pi}{16000} = 2.356V$$

If we sample at $f_s = 8 \times 4000 = 32$ KHz, then $\Delta \geq 1.178$ V. □

Of course , the approximation of the message can be improved if we sample faster. So, as it can be seen in the previous example, that the optimum depends on the sampling frequency f_s and also on the signal $m(t)$; i.e., its amplitude and frequency. For a general message signal that is not sinusoidal, the optimum Δ would depend on the message frequency components and their corresponding amplitudes. Alternatively, given a message signal with a specific bandwidth, and sampled at a specific rate (much higher than the Nyquist rate), the problem becomes on of finding the maximum amplitude of the message that results in no slope overload distortion. Of course a specific Δ needs to be adopted.

For a sinusoidal signal $m(t) = A\cos(2\pi f_c t)$, A_{max} can be determined using $\Delta \geq \frac{2\pi f_c A}{f_s}$, which gives $A \leq \frac{\Delta f_s}{2\pi f_c}$. Hence, $A_{max} = \frac{\Delta f_s}{2\pi f_c}$. We conclude that for a given Δ, f_s, and f, the smaller signal amplitude the better the fight against slope overload distortion would be. Also, for a given Δ, and f_s, when f increases there is a need for a smaller amplitude combat slope overload distortion. Fortunately, voice signals have a decreasing amplitude spectrum with increasing frequency.

1.11 Time-Division Multiplexing (TDM)

Definition 2. *Time-division multiplexing (TDM) is the time interleaving of samples from several sources so that the information from these sources can be transmitted serially over a single communication channel.*

Fig. 1.22 illustrates the TDM concept as applied to three analog sources that are multiplexed over a PCM system. For convenience, natural sampling is shown together with the corresponding gated TDM PAM waveform. In practice, an electronic switch (commutator) is used for the commutation (sampler). In this example, the pulse width of the TDM PAM signal is $T_s/3 = 1/(3f_s)$, and the pulse width of the TDM PCM signal is $T_s/(3n)$, where n is the number of bits used in the PCM word. Here $f_s = 1/T_s$ denotes the frequency of rotation for the commutator, and f_s satisfies the Nyquist rate for the analog source with the largest bandwidth. In some applications in which the bandwidth of the sources is markedly different, the larger bandwidth sources may be connected to several switch positions on the sampler so that they will be sampled more often than the smaller bandwidth sources.

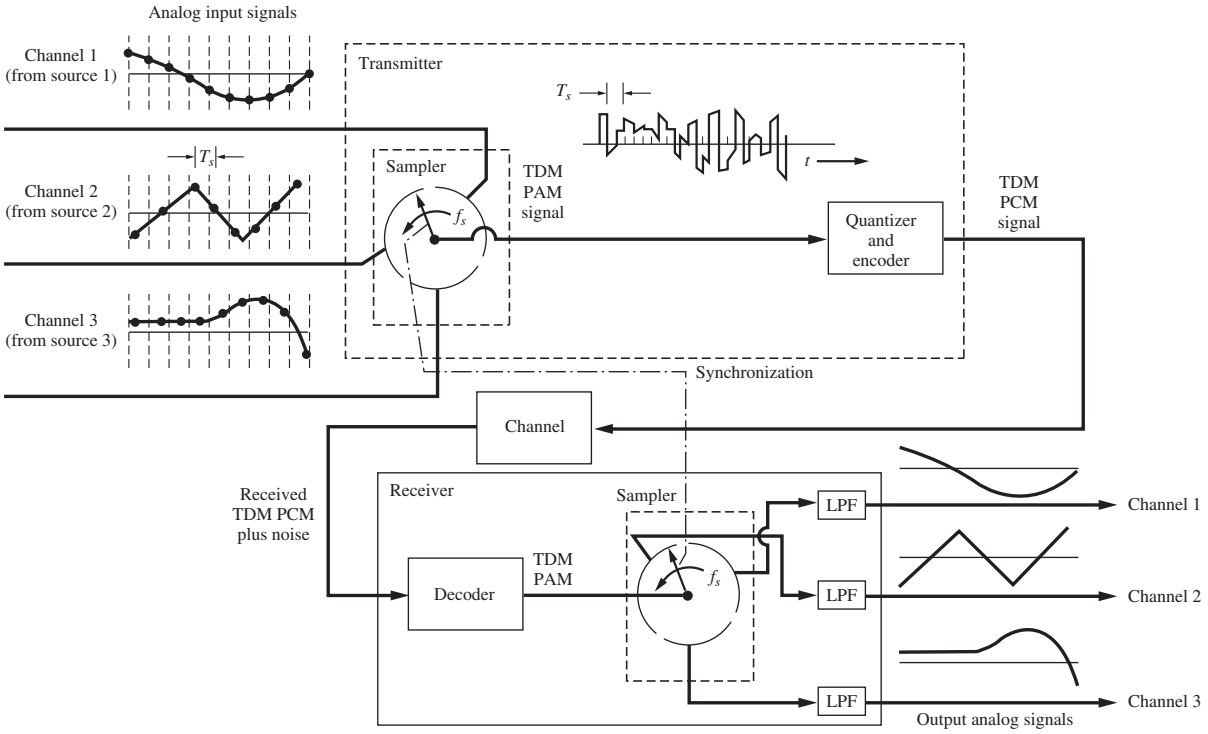


Figure 1.22: Three-channel TDM PCM system

We will start first by assuming all messages to have the same bandwidth. Usually anti-aliasing filters are applied to each message before sampling. The function of the commutator is twofold: (1) to take a narrow sample of each of the N input messages at a rate that is slightly higher than $2W$, where W is the cutoff frequency of the anti-aliasing filter, and (2) to sequentially interleave these N samples inside the sampling interval T_s. Indeed, this latter function is the essence of the time-division multiplexing operation. It is clear that the use of time-division multiplexing introduces a bandwidth expansion factor N, because the scheme must squeeze N samples derived from N independent message sources into a time slot equal to one sampling interval.

At the receiver, the decommutator (sampler) has to be synchronized with the incoming waveform so that the PAM samples corresponding to source 1, for example, will appear on the channel 1 output. This is called *frame synchronization*. Low-pass filters are used to reconstruct the analog signals from the PAM samples. Inter-Symbol-Interference (ISI) (to be discussed later) resulting from poor channel filtering would cause PCM samples from one channel to appear on another channel, even though perfect bit and frame synchronization were maintained. Feedthrough of one channel?s signal into another channel is called crosstalk.

Example 18. *(TDM using PAM)*
Consider time division multiplexing of 40 messages using PAM. These signals are to be transmitted over a channel having a bandwidth equal to 200 KHz. The channel bandwidth is assumed to be defined by the multiplexed PAM pulse duration. Find the maximum sampling rate that can be used so that the multiplexed PAM signal can be transmitted over the channel.

Solution. The duration of each pulse in the multiplexed signal is $T_s/40$, where T_s is the sampling period. Hence, $40f_s$ is the required transmission bandwidth. Since $40f_s \leq 200KHz \Rightarrow f_s \leq 5KHz$. □

Example 19. *(TDM using PCM)*
Consider time division multiplexing of 5 messages using PCM. Each message has a bandwidth of 2 KHz and is sampled at Nyquist rate. After sampling, each sample is coded as 3 bits. Find the transmission bandwidth.

Solution. The number of bits in the sampling period is $5 \times 3 = 15bits$. Hence, the bit duration is $T_s/15 \Rightarrow B_W = 15/T_s = 15f_s = 15 \times 2 \times 2 = 60KHz$. □

1.11.1 Frame Synchronization

Frame synchronization is needed at the TDM receiver so that the received multiplexed data can be sorted and directed to the appropriate output channel. The frame sync can be provided to the receiver demultiplexer (demux) circuit either by sending a frame sync signal from the transmitter over a separate channel or by deriving the frame sync from the TDM signal itself. Because the implementation of the first approach is obvious, we will concentrate on that of the latter approach, which is usually more economical, since a separate sync channel is not needed. As illustrated in Fig. 1.23, frame sync may be multiplexed along with the information words in an N-channel TDM system by transmitting a unique $K-$bit sync word at the beginning of each frame.

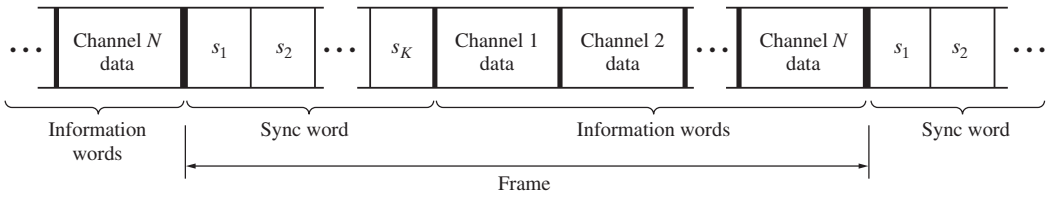


Figure 1.23: TDM frame sync format.

The frame sync is recovered from the corrupted TDM signal by using a frame synchronizer circuit that cross-correlates the regenerated TDM signal with the expected unique sync word.

Example 20. *(Design of a Time-Division Multiplexer)*
Design a time-division multiplexer that will accommodate 11 sources. Assume that the sources have the following specifications: Source 1: Analog, 2 KHz bandwidth, Source 2: Analog, 4 KHz bandwidth, Source 3: Analog, 2 KHz bandwidth, Source $4-11$: Digital, synchronous at 7200 bits/s. Suppose that the analog sources will be converted into 4-bit PCM words and, for simplicity, that frame sync will be provided via a separate channel and synchronous TDM lines are used. Design a TDM system to accommodate the 11 sources.

Solution. To satisfy the Nyquist rate for the analog sources, sources 1, 2, and 3 need to be sampled at 4, 8, and 4 KHz, respectively. As shown in Fig. 1.24, this can be accomplished by rotating the first commutator at $f_1 = 4$ *KHz* and sampling

source 2 twice on each revolution. This produces a 16 *ksamples/s* TDM PAM signal on the commutator output. Each of the analog sample values is converted into a 4-bit PCM word, so that the rate of the TDM PCM signal on the ADC output is 64 *Kbits/s*. The digital data on the ADC output may be merged with the data from the digital sources by using a second commutator rotating at $f_2 = 8$ *KHz* and wired so that the 64 *kbits/s* PCM signal is present on 8 of 16 terminals. This arrangement provides an effective sampling rate of 64 *kbits/s*. On the other eight terminals, the digital sources are connected to provide a data transfer rate of 8 *Kbits/s* for each source. Since the digital sources are supplying a 7.2 *kbit/s* data stream, pulse stuffing is used to raise the source rate to 8 *kbits/s*.

□

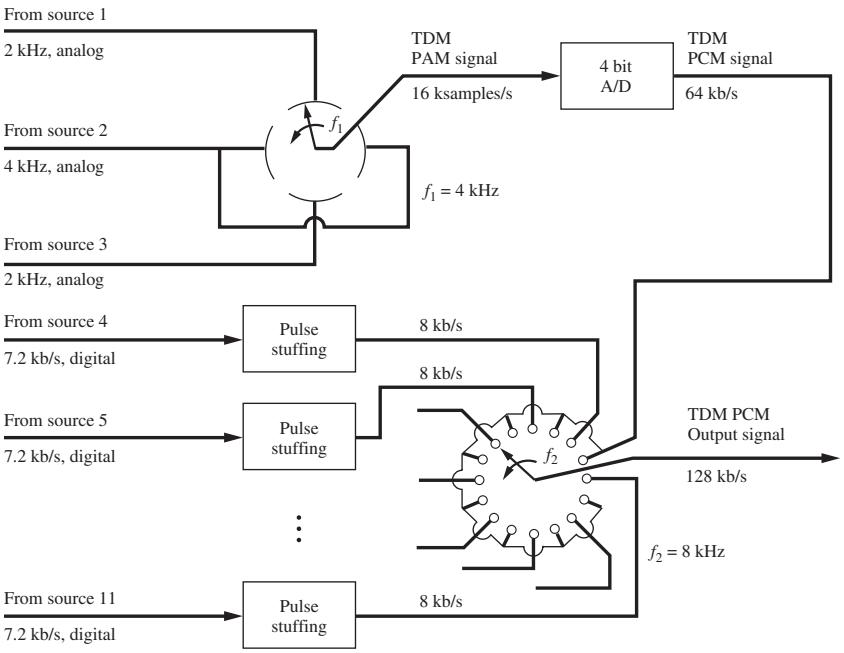


Figure 1.24: TDM with analog and digital inputs.

CHAPTER 3

INFORMATION THEORY

3.1 Introduction

So far, the sampler and quantizer have reduced the analog source into a discrete source whose output consists of sequences of quantization levels taking a discrete set of values. The next step in efficiently digitizing the analog signal is to map each quantization level at the output of the quantizer into a unique binary sequence. This is done by a source encoder, whose purpose is to assign to each quantized sample $\hat{x}_i$ a binary sequence b_i. The source encoder converts the sequence of symbols from the source to a sequence of binary digits, preferably using as few binary digits per symbol as possible. The source decoder performs the inverse operation. Initially, in the spirit of source/channel separation, we ignore the possibility that errors are made in the channel decoder and assume that the source decoder operates on the source encoder output.

One way to accomplish the mapping into a sequence of binary digits is to assign to each of the L possible quantized outputs a distinct binary sequence of length $N = \log_2(L)$ bits. In case L is not a power of 2, in which case N is not an integer, we must instead use the next integer N' greater than N. The resulting encoder is referred to as a *fixed-length encoder*, and its rate is N' bits/sample. Fixed-length encoders, although easy to implement, are not in general efficient. If we allow the length of a binary sequence assigned to a quantized sample to be variable, we can usually achieve rates less that N' bits/sample. These source encoders, to be studied next, are know as *variable-length encoders*. Variable-length encoders can achieve a smaller average number of encoded bits per source symbol (quantized value), which is desirable since this leads to a smaller bit-rate to communicate the source. A smaller bit-rate, in turn, means a smaller required channel bandwidth for transmission. If the source bandwidth is W Hz, and an $N-$bit quantizer is used, then the bandwidth, B, required to transmit the source is (assuming approximately that signaling bandwidth equals the inverse of the signaling rate): $B \simeq N \times f_s \geq 2WN$, where W is the bandwidth of the analog source. Clearly, bandwidth requirements increase linearly with N thus the need to keep N as small as possible. To summarize, the goal of this chapter is to represent a source with the fewest possible bits such that best recovery of the source from the compressed data can be achieved. Before we can analyze variable-length source encoders, we need some information-theory background.

3.2 Measures of Information

In everyday life, events can surprise us. Usually, the more likely or unexpected an event is, the more surprising it is. Thus, the more information it conveys when it happens. As we saw above, an analog source can be converted into a discrete source through sampling and quantization. Consider a discrete source which produces outputs that belong to a set of L possible symbols x_i, $i = 1, 2..., L$ (in our particular case, for example, quantization levels). Each of the L outputs from the source occurs with some probability p_i, $i = 1, 2, ..., L$ The source can be modeled as a discrete random variable X which takes values from the set of possible source outputs with a given probability p_i, i.e., $Pr(X = x_i) = p_i$.

Definition 3. *(Self-information) If x_i is an event produced by a discrete source X, the amount of information gained after*

observing x_i is called self-information *and is given by*

$$I(x_i) = \log_2 \frac{1}{p_i} \quad \textit{bits} \tag{3.1}$$

Note that a *bit* is now a unit of information. Equivalently, $I(x_i)$ is the amount of uncertainty that we have about whether x_i will occur, before an output occurs. Thus, $I(x_i)$ can be thought of either as the amount of uncertainty about x_i before it occurs, or the amount of information that we receive when we actually observe x_i. Clearly uncertainty and information go hand-in-hand and they are related to the probabilities of occurrence of the various symbols produced by the source.

Example 21. *Considering tossing a fair dice. The probability density function is given by $p_i = 1/6$ for $i = 1, ..., 6$. Hence, $I(x_i) = \log_2 6 = 2.585$ bits. Thus, the occurrence of say, a 5, conveys 2.585 bits of information, as would the occurrence of any other symbol in this case.*

Self-information has the following properties:

1. Events which are certain to occur, i.e., $p_i = 1$, have $I(x_i) = 0$, thus zero surprise and no informations.
2. Events which are impossible, that is, $p_i = 0$, have $I(x_i) = \infty$, thus infinite surprise.
3. $I(x_i) \geq 0$, that is the occurrence of an event either provides some or no information, but never brings about a loss of information.
4. $I(x_k) > I(x_i)$ for $p_k < p_i$, that is the less probable an event is, the more information we get when it occurs.
5. $I(x_i x_k) = I(x_i) + I(x_k)$, given that the occurrence of x_i is independent of x_k.

A natural question that might arises is: Why do we use Log function to define a metric that measures information? Given a set of independent events $A_1, A_2, \ldots, A_n$ with PMF $p_i = P(A_i)$, we want the definition of information measure to satisfy:

1. A small change in p_i should cause small change in information.
2. If $p_i = 1/n$, for all i, then information should be a monotonically increasing function of n.
3. Dividing the outcome of a source into several groups does not change the information.

Claude Shannon showed that the only way all these condition could be satisfied was if the measure is $H = -k \sum p_i \log p_i$, where k is an arbitrary positive constant. This measure is called the *entropy*.

Definition 4. *(Entropy) The* entropy*, $H(X)$, of a source is the average amount of self-information produced by the source, i.e.,*

$$H(X) = E[I(X)] = E\left[\log_2 \frac{1}{p(x)}\right] \tag{3.2}$$

The entropy of s discrete source is always non-negative; $H(X) \geq 0$.

Example 22. *Consider a source with symbols from the set $\{x_1, x_2, x_3, x_4\}$. Let $p_1 = 1/2$, $p_2 = 1/4$, $p_3 = 1/8$, $p_4 = 1/8$. Then, $H(x) = \sum_{i=1}^{4} p_i \log_2(p_i) = 1.75$ bits. Suppose we want to determine the value of X with the minimum number of binary questions (Yes/No). An efficient first question is: is $X = a$? If the answer is no, the second question can be:is $X = b$?. If the answer is no, the third question can be: is $X = c$?, and so on. The resulting expected number of binary questions required is 1.75 bits. We show later that the minimum expected number of binary questions required to determine X lies between $H(X)$ and $H(X) + 1$.*

So, the entropy can be thought of as a measure of the following things about X:

1. The average length of the shortest description of the random variable
2. The amount of information provided by an observation of X
3. Our uncertainty about X
4. The randomness of X

When the random variable X has only two possible outcomes, one with probability p and the other with probability $(1-p)$, then

$$H(X) = H(p, 1-p) \triangleq h(p) = -p\log_2 p - (1-p)\log_2(1-p) \tag{3.3}$$

where $h(p)$ is the *binary entropy function* shown in Fig.3.1.

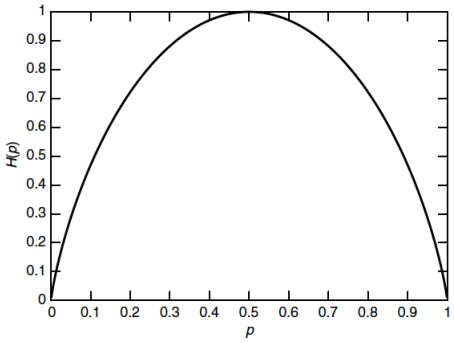


Figure 3.1: Binary entropy function.

Definition 5. *As we can see from Fig.3.1, the maximum uncertainty, $H(X) = 1$, occurs when$p = 1/2$ (maximum confusion), and $H(X) = 0$ when $p = 0$ or 1 (i.e., X is* deterministic*). Also, it is important to note that $h(p)$ is a* concave *function of p. This can be easily proven since $h''(p) = -\frac{1}{p^2-p}$, which is negative for all $0 < p < 1$.*
If the source outputs symbols at a fixed rate of R_s symbols per second, then the information rate of the source is: $R_b = H(X)R_s$ bits/s

Theorem 1. *(Bounds on the Entropy of a discrete random variable)*
For a source producing $|\mathcal{X}|$ symbols,

$$0 \leq H(X) \leq \log_2(|\mathcal{X}|) \tag{3.4}$$

with the upper bound being achieved if and only if $p_i = 1/|\mathcal{X}|)$, $i = 1, 2, ..., |\mathcal{X}|)$ (uniform distribution), and the lower bound achieved for X being deterministic.

Proof. Since $-\log p_X(x)$ is always nonnegative for all $x \in \mathcal{X}$, it follows that $H(X) \geq 0$. We write $H(X)$ as follows

$$\begin{aligned} H(X) &= \sum_{x\in\mathcal{X}} p_X(x)\log\frac{1}{p_X(x)} = \sum_{x\in\mathcal{X}} p_X(x)\log\frac{1/|\mathcal{X}|}{p_X(x)\cdot 1/|\mathcal{X}|} \\ &= \sum_{x\in\mathcal{X}} p_X(x)\log\frac{1}{1/|\mathcal{X}|} + \sum_{x\in\mathcal{X}} p_X(x)\log\frac{1/|\mathcal{X}|}{p_X(x)} \\ &= \log|\mathcal{X}| + \sum_{x\in\mathcal{X}} p_X(x)\log\frac{1/|\mathcal{X}|}{p_X(x)} \end{aligned}$$

Assuming for now that the logarithm has base e. Using the fact that $\ln x \leq x - 1$, we can bound $H(X)$ by

$$\begin{aligned} H(X) &\leq \ln|\mathcal{X}| + \sum_{x\in\mathcal{X}} p_X(x)\left(\frac{1/|\mathcal{X}|}{p_X(x)} - 1\right) \\ &= \ln|\mathcal{X}| + \sum_{x\in\mathcal{X}} p_X(x)\left(\frac{1}{|\mathcal{X}|} - p_X(x)\right) \\ &= \ln|\mathcal{X}| + 1 - 1 = \ln|\mathcal{X}| \end{aligned}$$

The bound $\ln x \leq x - 1$ holds with equality if and only if $x = 1$. In the derivation above, we see that $H(X) = \ln|\mathcal{X}|$ if and only if $p_X(x) = 1/|\mathcal{X}|$ for all $x \in \mathcal{X}$, i.e, the symbols are equally likely. Finally, if the logarithm has base 2, then we can use the bound $\log_2 x = \frac{\ln x}{\ln 2} \leq \frac{x-1}{\ln 2}$ to show that $H(X) \leq \log_2 |\mathcal{X}|$. □

Definition 6. *(Joint Entropy) The* joint entropy $H(X, Y)$ *of a pair of discrete random variables* (X, Y) *with a joint distribution* $p(x, y)$ *is defined as*

$$H(X, Y) = -\sum_{x\in\mathcal{X}}\sum_{y\in\mathcal{Y}} p(x, y)\log p(x, y) = -E\left[\log p(x, y)\right] \tag{3.5}$$

We also define the *conditional entropy* of a random variable given another as the expected value of the entropies of the conditional distributions, averaged over the conditioning random variable.

Definition 7. *(Conditional Entropy) If* $(X, Y) \sim p(x, y)$*, the* conditional entropy $H(Y|X)$ *is defined as*

$$\begin{aligned} H(Y|X) &= \sum_{x\in\mathcal{X}} p(x)H(Y|X = x) && (3.6)\\ &= -\sum_{x\in\mathcal{X}} p(x)\sum_{y\in\mathcal{Y}} p(y|x)\log p(y|x) && (3.7)\\ &= -\sum_{x\in\mathcal{X}}\sum_{y\in\mathcal{Y}} p(x, y)\log p(y|x) && (3.8)\\ &= -E\left[\log p(Y|X)\right] && (3.9) \end{aligned}$$

Theorem 2. *(Entropy Chain Rule)*

$$H(X, Y) = H(X) + H(Y|X) \tag{3.10}$$

Proof.

$$\begin{aligned} H(X, Y) &= -\sum_{x\in\mathcal{X}}\sum_{y\in\mathcal{Y}} p(x, y)\log p(x, y) \\ &= -\sum_{x\in\mathcal{X}}\sum_{y\in\mathcal{Y}} p(x, y)\log p(x)p(y|x) \\ &= -\sum_{x\in\mathcal{X}}\sum_{y\in\mathcal{Y}} p(x, y)\log p(x) - \sum_{x\in\mathcal{X}}\sum_{y\in\mathcal{Y}} p(x, y)\log p(y|x) \\ &= -\sum_{x\in\mathcal{X}} p(x)\log p(x) - \sum_{x\in\mathcal{X}}\sum_{y\in\mathcal{Y}} p(x, y)\log p(y|x) \\ &= H(X) + H(Y|X) \end{aligned}$$

□

Corollary 1.

$$H(X, Y|Z) = H(X|Z) + H(Y|X, Z) \tag{3.11}$$

Example 23. *Let* (X, Y) *have the following joint distribution:*

Y \ X	1	2	3	4
1	$\frac{1}{8}$	$\frac{1}{16}$	$\frac{1}{32}$	$\frac{1}{32}$
2	$\frac{1}{16}$	$\frac{1}{8}$	$\frac{1}{32}$	$\frac{1}{32}$
3	$\frac{1}{16}$	$\frac{1}{16}$	$\frac{1}{16}$	$\frac{1}{16}$
4	$\frac{1}{4}$	0	0	0

Find $H(X)$, $H(Y)$, and $H(X|Y)$, and $H(X, Y)$.

Solution. The marginal distribution of X is $\left(\frac{1}{2}, \frac{1}{4}, \frac{1}{8}, \frac{1}{8}\right)$ and the marginal distribution of Y is $\left(\frac{1}{4}, \frac{1}{4}, \frac{1}{4}, \frac{1}{4}\right)$, and hence $H(X) = 7/4$ bits and $H(Y) = 2$ bits.

$$\begin{aligned} H(X|Y) &= \sum_{i=1}^{4} p(Y=i)H(X|Y=i) \\ &= \frac{1}{4}H\left(\frac{1}{2}, \frac{1}{4}, \frac{1}{8}, \frac{1}{8}\right) + \frac{1}{4}H\left(\frac{1}{4}, \frac{1}{2}, \frac{1}{8}, \frac{1}{8}\right) + \frac{1}{4}H\left(\frac{1}{4}, \frac{1}{4}, \frac{1}{4}, \frac{1}{4}\right) + \frac{1}{4}H\left(1, 0, 0, 0\right) \\ &= \frac{1}{4} \times \frac{7}{4} + \frac{1}{4} \times \frac{7}{4} + \frac{1}{4} \times 2 + \frac{1}{4} \times 0 \\ &= \frac{11}{8} \textit{bits} \end{aligned}$$

Similarly, $H(Y|X) = 13/8$ bits and $H(X, Y) = H(X|Y) + H(Y) = 27/8$ bits. □

Theorem 3. *(Conditioning Reduces Entropy)(Information can't hurt)*

$$H(X|Y) \leq H(X) \tag{3.12}$$

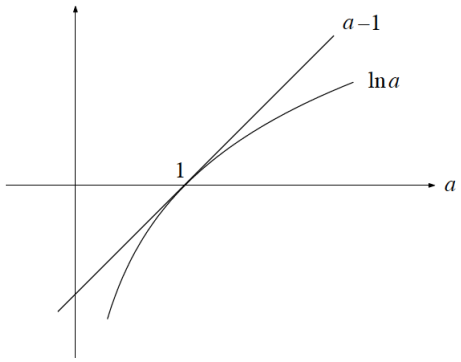
with equality if and only if X and Y are independent.

Proof. We first rewrite $H(X|Y)$ as follows

$$\begin{aligned} H(X|Y) &= \sum_{x\in\mathcal{X}}\sum_{y\in\mathcal{Y}} p(x,y) \log \frac{1}{p_{X|Y}(x|y)} \\ &= \sum_{x\in\mathcal{X}}\sum_{y\in\mathcal{Y}} p(x,y) \log \frac{p_X(x)}{p_{X|Y}(x|y)p_X(x)} \\ &= \sum_{x\in\mathcal{X}}\sum_{y\in\mathcal{Y}} p(x,y) \log \frac{1}{p_X(x)} + \sum_{x\in\mathcal{X}}\sum_{y\in\mathcal{Y}} p(x,y) \log \frac{p_X(x)}{p_{X|Y}(x|y)} \\ &= H(X) + \sum_{x\in\mathcal{X}}\sum_{y\in\mathcal{Y}} p(x,y) \log \frac{p_X(x)p_Y(y)}{p_{X,Y}(x,y)} \end{aligned}$$

Assuming the natural logarithm, we can use a fundamental inequality on the natural log function; $\ln a \leq a-1$ to bound $H(X|Y)$ by

$$\begin{aligned} H(X|Y) &\leq H(X) + \sum_{x\in\mathcal{X}}\sum_{y\in\mathcal{Y}} p(x,y) \left(\frac{p_X(x)p_Y(y)}{p_{X,Y}(x,y)} - 1\right) \\ &= H(X) + \sum_{x\in\mathcal{X}} p_X(x) \sum_{y\in\mathcal{Y}} p_Y(y) - \sum_{x\in\mathcal{X}}\sum_{y\in\mathcal{Y}} p(x,y) \\ &= H(X) + 1 - 1 = H(X) \end{aligned}$$



Note that the equality $H(Y|X) = H(X)$ holds if and only if the logarithm argument is equal to1 while applying $\ln x \leq x - 1$. This happens when $p_{XY}(x, y) = p_X(x)p_Y(y)$, i.e., X and Y are independent. □

Intuitively, the theorem says that knowing another random variable Y can only reduce the uncertainty in X. Note that this is true only on the average. Specifically, $H(X|Y = y)$ may be greater than or less than or equal to $H(X)$, but on the average $H(X|Y) = \sum_y p(y)H(X|Y = y) \leq H(X)$. For example, in a court case, specific new evidence might increase uncertainty, but on the average evidence decreases uncertainty.

Corollary 2. *Let X_1, X_2,. . . ,X_n be drawn according to $p(x_1, x_2, \dots, x_n)$. Then,*

$$H(X_1, X_2, \dots, X_n) = \sum_{i=1}^{n} H(X_i|X_{i-1}, \dots, X_1) \tag{3.13}$$

Theorem 4. *(Independence bound on entropy)*
Let X_1, X_2,. . . ,X_n be drawn according to $p(x_1, x_2, \dots, x_n)$. Then,

$$H(X_1, X_2, \dots, X_n) \leq \sum_{i=1}^{n} H(X_i) \tag{3.14}$$

with equality if and only if the X_i are independent.

Proof. By the chain rule for entropies,

$$\begin{aligned} H(X_1, X_2, \dots, X_n) &= \sum_{i=1}^{n} H(X_i|X_{i-1}, \dots, X_1) \\ &\leq \sum_{i=1}^{n} H(X_i) \end{aligned}$$

where the last inequality follows from the result that conditioning reduces entropy. We have equality if and only if X_i is independent of $X_{i-1}, \dots, X_1$ for all i (i.e., if and only if the X_i's are independent). □

An essential parameter in measuring the amount of information shared between two points in a communication system (two random variables X and Y) is the *mutual information* defined as follows.

Definition 8. *(Mutual Information)*
If $(X, Y) \sim p(x, y)$, the mutual information *between X and Y is defined as*

$$I(X; Y) = E_{(X,Y)}\left[\log \frac{p(X, Y)}{p(X)p(Y)}\right] \tag{3.15}$$

Mutual information measures how much knowing one of the random variables reduces uncertainty about the other. For example, if X and Y are independent, then knowing X does not give any information about Y and vice versa, so their mutual information is zero. At the other extreme, if X is a deterministic function of Y and Y is a deterministic function of X then all information conveyed by X is shared with Y: knowing X determines the value of Y and vice versa. As a result, in this case the mutual information is the same as the uncertainty contained in Y (or X) alone, namely the entropy of Y (or X). Moreover, this mutual information is the same as $H(X)$ and as $H(Y)$. (A very special case of this is when X and Y are the same random variable.) As seen from the definition, mutual information is a measure of the inherent dependence expressed in the joint distribution of X and Y relative to the joint distribution of X and Y under the assumption of independence (the product of marginals). Mutual information therefore measures dependence in the following sense: $I(X;Y) = 0$ if and only if X and Y are independent random variables. The aforementioned is concretized in the following properties and theorems.

Corollary 3.

$$I(X;Y) = H(X) + H(Y) - H(X,Y) = H(X) - H(X|Y) \tag{3.16}$$

Proof.

$$\begin{aligned}
I(X;Y) &= E_{(X,Y)}\left[\log \frac{p(X,Y)}{p(X)p(Y)}\right] \\
&= E_{(X,Y)}\left[\log \frac{1}{p(X)}\right] + E_{(X,Y)}\left[\log \frac{1}{p(y)}\right] - E_{(X,Y)}\left[\log \frac{1}{p(X,Y)}\right] \\
&= E_X\left[\log \frac{1}{p(X)}\right] + E_Y\left[\log \frac{1}{p(Y)}\right] - H(X,Y) \\
&= H(X) + H(Y) - H(X,Y) \\
&= H(X) + H(Y) - (H(Y) + H(X|Y)) \\
&= H(X) - H(X|Y)
\end{aligned}$$

□

Hence, $I(X;Y)$ measures the amount of information that Y carries about X.

Note that $I(X;X) = H(X) - H(X|X) = H(X) - 0 = H(X)$. So, entropy is also called *self-information*.
Properties of $I(X;Y)$

1. The Non-negativity of mutual information: $I(X;Y) \geq 0$ (although this is pretty intuitive, the proof of this result is beyond the scope of this course)
2. Mutual information is symmetric: $I(X;Y) = I(Y;X)$

3.3 Source Codes

Entropy as a measure of information is not only intuitively satisfying but also appears universally in characterizing the fundamental limits of communication systems:

- Data compression
- Communication over (point-to-point) noisy channels
- Wireless uplink and downlink communication
- Coding for computer networks

This chapter treats first the problem of *data compression* (or *source coding*) and then treats the problem of *channel capacity*.

Definition 9. *A discrete source is* memoryless *if successive symbols, $X_1, X_2, \ldots$, produced by the source are independent of one another;*

$$p(\{x_i\}_{i=1}^n) = \prod_{i=1}^n p(x_i)$$

Suppose we assign binary codewords to each symbol or to groups of source symbols:

- L_I = input word length (source symbols)
- L_O = output code word length (code bits)
- $R = L_O/L_I$ bits/source symbol is the *code rate* of the source code

The source-coding theorem is one of the three fundamental theorems of information theory introduced by Shannon (1948). The source-coding theorem establishes a fundamental limit on the rate at which the output of an information source can be compressed without causing a large error probability. We have seen already that the entropy of an information source is a measure of the uncertainty or, equivalently, the information content of the source. Therefore, it is natural that in the statement of the source-coding theorem, the entropy of the source plays a major role.

Theorem 5. ***(Source Coding Theorem: Shannon'48)***
If a discrete source has an entropy of $H(X)$ bits/source symbol, it can be encoded with some lossless source code of rate R, provided that $R > H(X)$. Furthermore, if $R < H(X)$ then a lossless representation of the source is not possible. The smallest average number, $\overline{L}$, of bits per source symbol that any source encoder can achieve equals the entropy of the source, i.e., $\overline{L} \geq H(X)$.

This theorem indicates the fundamental nature of $H(X)$ in that no source encoder can achieve a rate less than $H(X)$. In fact, most practical encoders can only hope to approach $H(X)$.

From the source coding theorem we observe that the entropy H gives a sharp bound on the rate at which a source can be compressed for reliable reconstruction. This means that at rates above the entropy, it is possible to design a code with an error probability as small as desired, whereas at rates below entropy, such a code does not exist. This important result; however, does not provide specific algorithms to design codes approaching this bound. In this section, we will introduce algorithms that perform very close to the entropy bound, and that achieves that bound under particular cases.

First, we need to investigate some desired features that these codes should satisfy. We start by defining a *fixed-length code* and a *variable-length code*.

3.3.1 Fixed-Length Codes

The simplest approach to encoding a discrete source into binary digits is to create a code C that maps each symbol x of the alphabet $\mathcal{X}$ into a distinct codeword $C(x)$, where $C(x)$ is a block of binary digits. Each such block is restricted to have the same block length L, which is why such a code is called a fixed-length code. For example, if the alphabet $\mathcal{X}$ consists of the 7 symbols $\{a, b, c, d, e, f, g\}$, then the following fixed-length code of block length $L = 3$ could be used.

$$\begin{aligned} C(a) &= 000 \\ C(b) &= 001 \\ C(c) &= 010 \\ C(d) &= 011 \\ C(e) &= 100 \\ C(f) &= 101 \\ C(g) &= 110 \end{aligned}$$

The source output, x_1, x_2,..., would then be encoded into the encoded output $C(x_1)C(x_2)...$ and thus the encoded output contains L bits per source symbol. For the above example the source sequence $bad...$ would be encoded into $001000011...$. Note that the output bits are concatenated.

There are 2^L different combinations of values for a block of L bits. Thus, if the number of symbols in the source alphabet, $M = |\mathcal{X}|$, satisfies $M \leq 2^L$, then a different binary L-tuple maybe be assigned to each symbol. Assuming that the decoder knows where the beginning of the encoded sequence is, the decoder can segment the sequence into $L-$ bit blocks and then decode each block into the corresponding source symbol.

In summary, if the source alphabet has size M, then this coding method requires $L = \lceil \log_2 M \rceil$ bits to encode each source symbol. Thus, $\log_2 M \leq L < \log_2 M + 1$. The lower bound, $\log_2 M$, can be achieved with equality if and only if M is a power of 2.

This method is non-probabilistic; it takes no account of whether some symbols occur more frequently than others, and it works robustly regardless of the symbol frequencies. But if it is known that some symbols occur more frequently than others, then the rate $\overline{L}$ of coded bits per source symbol can be reduced by assigning shorter bit sequences to more common symbols in a variable-length source code.

3.3.2 Variable-Length Codes

Data compression can be achieved by assigning short descriptions to the most frequent outcomes of the data source, and necessarily longer descriptions to the less frequent outcomes.

Definition 10. *The* expected length, $\overline{L}$, *of a source code* $C(x)$ *for a random variable* X *with probability mass function* $p(x)$ *is given by*

$$\overline{L} = \sum_{x \in \mathcal{X}} p(x)\ell(x) \tag{3.17}$$

where $\ell(x)$ *is the length of the codeword associate with* $\mathcal{X}$.

Example 24. *Let us assume that there are only four quantization levels:* ± 0.75, ± 0.25. *Moreover, let us assume that levels* ± 0.25 *occur with probability 3/8 each, and levels* ± 0.75 *with probability 1/8 each. Now consider the following source code:*

$$\begin{aligned} +0.25 &\Leftrightarrow 0 \\ -0.25 &\Leftrightarrow 11 \\ +0.75 &\Leftrightarrow 100 \\ -0.75 &\Leftrightarrow 101 \end{aligned}$$

The average number of bits per source symbol (expected length), $\overline{L}$, *is*

$$\overline{L} = 1 \times \frac{3}{8} + 2 \times \frac{3}{8} + 3 \times \frac{1}{8} + 3 \times \frac{1}{8} = 1.875 \textit{bits/source symbol}$$

With a fixed-length code, $\overline{L} = 2 \textit{bits/symbol} > 1.875$. *We will see that, in general, larger improvements can be expected than above. The smallest average number of bits per source symbol is actually* 1.8113 *bits/source symbol.*

Definition 11. *(Non-Singular Code) A code is said to be* non-singular *if for different symbols, we have different codeword representation:* $x_1 \neq x_2 \Rightarrow C(x_1) \neq C(x_2)$

Definition 12. *(Code Extension) The* extension of a code *is a concatenation of the codeword representation of many symbols:* $C(x_1 x_2 ... x_n) = C(x_1)C(x_2)...C(x_n)$

Definition 13. *(Uniquely Decodable Codes) A code is called* uniquely decodable *(U.D) if its extension is non-singular. In other words, if the original symbols can be recovered uniquely from sequences of encoded symbols. However, one may have to look at the entire string to determine even the first symbol in the corresponding source string.*

Example 25. *Consider the following scenarios*

1.

$$\begin{aligned} \ell_1 &\Leftrightarrow 00 \\ \ell_2 &\Leftrightarrow 00 \quad \textit{Not uniquely decodable} \\ \ell_3 &\Leftrightarrow 11 \end{aligned}$$

2.

$$\begin{aligned} \ell_1 &\Leftrightarrow 0 \\ \ell_2 &\Leftrightarrow 1 \quad \textit{Not uniquely decodable} \\ \ell_3 &\Leftrightarrow 11 \end{aligned}$$

For example, the sequence $...011001...$ *can be decoded either as* $...\ell_1\ell_3\ell_1\ell_1\ell_2...$ *or* $...\ell_1\ell_2\ell_2\ell_1\ell_1\ell_2...$ *(not unique)*

3.

$$\begin{aligned} \ell_1 &\Leftrightarrow 00 \\ \ell_2 &\Leftrightarrow 01 \quad \textit{Uniquely decodable} \\ \ell_3 &\Leftrightarrow 11 \end{aligned}$$

Definition 14. *(Prefix-Free Codes) A* prefix-free code *is one in which no codeword is a prefix to any other codeword*

Example 26.

$$\begin{aligned} \ell_1 &\Leftrightarrow 0 \\ \ell_2 &\Leftrightarrow 11 \\ \ell_3 &\Leftrightarrow 100 \quad \textit{Code is prefix-free} \\ \ell_4 &\Leftrightarrow 101 \end{aligned}$$

Theorem 6. *A sufficient condition for a code to be uniquely decodable is that it be prefix-free. In other words, all prefix-free codes are uniquely decodable, but not all uniquely decodable codes are necessarily prefix-free.*

Example 27.

$$\begin{aligned} \ell_1 &\Leftrightarrow 1 \\ \ell_2 &\Leftrightarrow 10 \quad \textit{Code is U.D. but not prefix-free} \\ \ell_3 &\Leftrightarrow 100 \end{aligned}$$

Prefix-free codes are also called *instantaneous codes* because a symbol can be decoded by the time the last bit is reached. Notice that the code in the previous example is not instantaneous, since we have to wait and see what the first bit in the next symbol is before we can decode the previous symbol.

In a uniquely decodable code which is not instantaneous, we may have to wait a long time before we know the identity of the first symbol. There is a testing procedure that can always be used to determine whether or not a code is uniquely decodable. We explain this test through an example.

Example 28. *Consider a code:* $\{a, c, ad, abb, bad, deb, bbcde\}$*, and we want to test whether is it U.D. or not. We start by constructing a sequence of sets* S_0*,* S_1*, ..., as follows:*
Let S_0 *be the original set of codewords. To form* S_1*, we look at all pairs of codewords in* S_0*. If a codeword* w_i *is a prefix of another codeword* w_j*;* $w_j = w_iA$*, we place the suffix* A *in* S_1*. In our case* $S_1 = \{d, bb\}$*. In general, to form* S_n*,* $n > 1$*, we*

compare S_0 to S_{n-1}. If a codeword $w \in S_0$ is a prefix of a sequence $A = wB \in S_{n-1}$, the suffix B is placed in S_n, and if a sequence $A' \in S_{n-1}$ is a prefix of a codeword $w' = A'B' \in S_0$, we place the suffix B' in S_n. We get the following

$$\begin{aligned} S_0 &= \{a, c, ad, abb, bad, deb, bbcde\} \\ S_1 &= \{a, bb\} \\ S_2 &= \{eb, cde\} \\ S_3 &= \{de\} \\ S_4 &= \{b\} \\ S_5 &= \{ad, bcde\} \\ S_6 &= \{d\} \\ S_7 &= \{eb\} \end{aligned}$$

A code is uniquely decodable if and only if none of the sets S_1, S_2,...,S_7 contains a codeword that is a member of S_0. Hence, our code is uniquely decodable. In fact, the sequence abbcdebad is ambiguous, having the two possible interpretations: a, bbcde, bad or abb, c, deb, ad.

We wish now to construct instantaneous codes of minimum expected length to describe a given source. It is clear that we cannot assign short codewords to all source symbols and still be prefix-free. The set of codewords lengths possible for instantaneous codes is limited by the following inequality.

Theorem 7. *(Kraft Inequality) For any instantaneous code over the binary alphabet, the codewords length ℓ_1, ℓ_2, ..., ℓ_m must satisfy the inequality*

$$\sum_{i}^{m} 2^{-\ell_i} \leq 1$$

Conversely, given a set of codeword lengths that satisfy the inequality, there exists an instantaneous code with these word lengths.

Definition 15. *(Optimal Code) A code having the minimum expected length $\overline{L}$ and which is subject to Kraft inequality is called* optimal code

Theorem 8. *(The Lossless Coding Theorem) The expected length $\overline{L}$ of a prefix-free code for a random variable X is greater than or equal to the entropy $H(X)$, $\overline{L} \geq H(X)$, with equality if and only if $p_i = 2^{-\ell_i}$.*

Definition 16. *The merit of any code is measured by its average length in comparison to $H(X)$. The* code efficiency η *is defined as*

$$\eta = \frac{H(X)}{\overline{L}} \tag{3.18}$$

where $\overline{L}$ is the average length of the code. The redundancy γ *is defined as*

$$\gamma = 1 - \eta \tag{3.19}$$

3.4 Huffman Coding Algorithm

An optimal prefix code for a given distribution can be constructed by a simple algorithm discovered by *Huffman*. Any other code for the same alphabet cannot have a lower expected length than the code constructed by this algorithm. Huffman code is an optimal code, but not the optimal. Huffman code archives Kraft inequality with equality. The Huffman code is a variable length, prefix-free (and thus U.D.) code that can asymptotically achieve an average length as close to the entropy of the source as desired. The price we pay for getting closer to the entropy is complexity. Huffman code is optimum for memoryless sources with known probabilities. Construction of Huffman codes is based on two ideas:

- In an optimum code, symbols with higher probability should have shorter codewords

- In an optimum prefix code, the two symbols that occur least frequently will have the same length

The following algorithm results in a Huffman code:

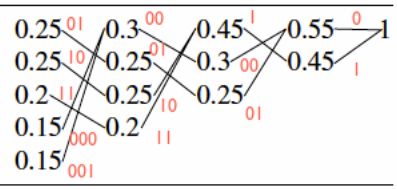
1. List the message symbols (source symbols) vertically and in such a way that symbols higher on the list are more probable than symbols following. On a parallel vertical column, list the corresponding probability of each symbol.
2. Assign to the two least probable symbols one a"0" and the other a "1" (it doesn't matter which). Add the probabilities of the two least-probable symbols, and consider the new list of probabilities.
3. Repeat part b) until only two probabilities are left (that should add up to one). Assign to one a "0" and to the other a "1".
4. Trace the branches of the resulting binary tree and read off the binary sequence corresponding to each branch. Assign this sequence to the symbol at the end of the branch.

Example 29. *Consider a random variable X taking values in the set $\mathcal{X} = \{1, 2, 3, 4, 5\}$ with probabilities 0.25, 0.25, 0.2, 0.15, 0.15, respectively. This code has an average length:*

$$\overline{L} = 2 \times 0.25 + 2 \times 0.25 + 2 \times 0.2 + 3 \times 0.15 + 3 \times 0.15 = 2.3 \quad \textit{bits}$$

The entropy of X is: $H(X) = 2.285 < 2.3$. The code efficiency is $\eta = \frac{2.285}{2.3} = 0.9934$. The redundancy $\gamma = 1 - 0.9934 = 0.00652$.

Codeword Length	Codeword	X	Probability
2	01	1	0.25
2	10	2	0.25
2	11	3	0.2
3	000	4	0.15
3	001	5	0.15



Remark 1. *For Huffman code, the redundancy is zero when the probabilities are negative powers of two.*

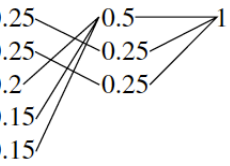
Remark 2. *When more than two "symbols" in a Huffman tree have the same probability, different merge orders produce different Huffman codes. Although the average length could be the same, the length variances are different.*

A similar procedure is used to find an optimal M-ary Huffman code. In this case we arrange the messages in descending order of probability, combine the last r messages into one message, and rearrange the new set (reduced set) in the order of descending probability. We repeat the procedure until the final set reduces to M messages. Each of these messages is not assigned one of the M numbers $0, 1, 2, \ldots, M-1$. We now regress in exactly the same way as in the binary case until each of the original messages is assigned a code.

Example 30. *Consider a ternary code for the same random variable as before. Now we combine the three least likely symbols into one supersymbol and obtain the following table: This code has an average length: $\overline{L} = 1 \times 0.25 + 1 \times 0.25 + 2 \times 0.2 + 2 \times 0.15 + 2 \times 0.15 = 1.5$ ternary digits.*

For an M-ary code, we will have exactly M messages left in the last reduced set if, and only if, the total number of original messages is equal to $M + k(M-1)$, where k is an integer. This is because each reduction decreases the number of messages

Codeword	X	Probability
1	1	0.25
2	2	0.25
00	3	0.2
01	4	0.15
02	5	0.15



by $M-1$. Hence, if there is a total of k reductions, the total number of original messages must be $M+k(M-1)$. In case the original messages do not satisfy this condition, we must add some dummy messages with zero probability of occurrence until this condition is fulfilled. For example, if $M=4$ and the number of messages is 6, then we must add one dummy message with zero probability of occurrence to make the total number of messages 7, that is $[4+1(4-1)]$, and proceed as usual.

3.5 Tunstall Codes

The Huffman code was the first variable length code that we looked at in this chapter. It encodes letters from the source alphabet using codewords with varying numbers of bits: codewords with fewer bits for letters that occur more frequently and codewords with more bits for letters that occur less frequently. The *Tunstall code* is an important exception. In the Tunstall code, all codewords are of equal length. However, each codeword represents a different number of letters. An example of a 2-bit Tunstall code for an alphabet $\{A, B\}$ is shown below. The main advantage of a Tunstall code is that errors in codewords do not propagate, unlike Huffman codes, in which an error in one codeword will cause a series of errors to occur.

Example 31. *Let's encode the sequence AAABAABAABAABAAA using the code in the following table.*

Starting at the left, we can see that the string AAA occurs in our codebook and has a code of 00. We then code B as 11, AAB as 01, and so on. We finally end up with coded string 001101010100.

Sequence	Codeword
AAA	00
AAB	01
AB	10
B	11

The Tunstall coding algorithm is as follows:
Suppose we want an n-bit Tunstall code for a source that generates i.i.d letters from an alphabet of size N. The number of codewords is 2^n. We start with the N letters of the source alphabet in our codebook. Remove the entry in the codebook that has the highest probability and add the N strings obtained by concatenating this letter with every letter in the alphabet (including itself). This will increase the size of the codebook from N to $N+(N-1)$. The probabilities of the new entries will be the product of the probabilities of the letters concatenated to form the new entry. Now look through the $N+(N-1)$ entries in the codebook and find the entry that has the highest probability, keeping in mind that the entry with the highest probability may be a concatenation of symbols. Each time we perform this operation we increase the size of the codebook by $N-1$. Therefore, this operation can be performed K times, where $N+K(N-1)\leq 2^n$.

Example 32. *Let us design a $3-$bit Tunstall code for a memoryless source with the following alphabet: $\{A, B, C\}$ with probabilities given in the following table.*

We start out with the codebook and associated probabilities. Since the letter A has the highest probability, we remove it from the list and add all two-letter strings beginning with A as shown below.

After one iteration we have 5 entries in our codebook. Going through one more iteration will increase the size of the codebook by 2, and we will have 7 entries, which is still less than the final codebook size. Going through another iteration after that would bring the codebook size to 10, which is greater than the maximum size of 8. Therefore, we will go through just one more iteration. The find $3-$bit Tunstall code is shown in the table below.

Letter	Probability
A	0.60
B	0.30
C	0.10

Sequence	Probability
B	0.30
C	0.10
AA	0.36
AB	0.18
AC	0.06

Sequence	Probability
B	000
C	001
AB	010
AC	011
AAA	100
AAB	101
AAC	110

3.6 Lempel-Ziv Coding Algorithm

Huffman coding has two important drawbacks. First, the source statistics are used to design a Huffman code. If one only has access to the source outputs, the design procedure requires two passes through the data, one to estimate the statistics of the source, and a second one for encoding. To overcome this, one can use adaptive Huffman codes where the code is updated dynamically to match the statistics of the sequence as it is observed. This is a problem because one must jointly encode multiple symbols to take advantage of source memory and reduce length rounding loss. In this case, one finds that the complexity increases exponentially with the number of symbols that are encoded together. Also, Tunstall codes depends on source statistics. To provide a partial solution to these drawbacks, we study an example of *universal source-coding algorithms*, namely the *Lempel- Ziv algorithm*. This type of universal data compression is the basis for standard file compression algorithms (e.g., winzip, gzip, unix compress).

In many applications, the output of the source consists of recurring patterns. A classic example is a text source in which certain patterns or words recur constantly. Also, there are certain patterns that simply do not occur, or if they do, occur with great rarity. For example, we can be reasonably sure that the word Limpopo1 occurs in a very small fraction of the text sources in existence.

A very reasonable approach to encoding such sources is to keep a list, or dictionary, of frequently occurring patterns. When these patterns appear in the source output, they are encoded with a reference to the dictionary. If the pattern does not appear

in the dictionary, then it can be encoded using some other, less efficient, method. In effect we are splitting the input into two classes, frequently occurring patterns and infrequently occurring patterns. For this technique to be effective, the class of frequently occurring patterns, and hence the size of the dictionary, must be much smaller than the number of all possible patterns.The most widely used dictionary compression techniques (also called adaptive dictionary) are the Lempel-Ziv family of codes.

The basic idea behind the Lempel-Ziv algorithm is to parse the input sequence into non-overlapping strings of different lengths while constructing a dictionary of the strings seen thus far. There are many versions of this algorithm and we discuss the variant known as LZ78 (It was the algorithm of the widely used Unix file compression utility *compress*, and is used in the GIF image format) that was described in a 1978 paper by Lempel and Ziv. The encoding algorithm works as follows. First, initialize the dictionary to contain all strings of length one and set the input pointer to the beginning of the string. Then, apply the following iterative procedure

1. Starting at the input pointer, find the longest substring w that is already in the dictionary.
2. Concatenate w with the next symbol y in the string and add *wy* to the first empty location in the dictionary.
3. Encode the pair by sending the dictionary index of w and the value of y.
4. Set the input pointer to the symbol after y.

There are a number of practical variants of this algorithm that improve performance and/or reduce the implementation complexity.

Decompression works in the reverse fashion. Each received index and symbol can be immediately decoded and used to build a copy of the dictionary at the receiver. In this fashion, one can resolve the input without ambiguity.

Example 33. *Suppose that we are to use a Lempel-Ziv algorithm with dictionary size $2^3 = 8$. The dictionary is initialized to contain 0 and 1 in the first two positions. Then, the source sequence is sequentially parsed into strings that have not appeared so far. For example,*

$$10110101000101... \rightarrow 10, 11, 01, 010, 00, 101$$

The dictionary table at this point has eight elements.

Each phrase (the bit string contained between two commas) is coded by giving the location of its prefix in the dictionary table, and the value of the additional bit. This results in the coded sequence

$$10, 11, 01, 010, 00, 101 \rightarrow (001, 0)(001, 1)(000, 1)(100, 0)(000, 0)(010, 1)$$

where the first number of each pair gives the index of the prefix in the table and the second number gives the last bit of the new phrase. When applied to sequences generated by any stationary ergodic source, the Lempel-Ziv coding algorithm asymptotically achieves the optimal encoding rate (known as the entropy rate). Most readers will notice that this algorithm, as stated, requires prior knowledge of the total number of phrases in the dictionary. In fact, this problem can be solved easily and the solution actually requires fewer transmitted bits. The key point is that both the transmitter and receiver know the number of phrases currently in the dictionary. Let M be the current number of phrases in the dictionary. Then, the transmitter can be simply send the $\lceil \log_2 M \rceil$ least significant bits of the index. Since the receiver also knows M, there will be no confusion. In this case, the encoded sequence will be

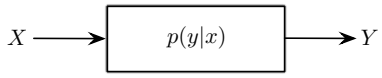
$$10, 11, 01, 010, 00, 101 \rightarrow (1, 0)(01, 1)(00, 1)(100, 0)(000, 0)(010, 1)$$

3.7 Channel Coding

We shift gears now and turn our attention to the fundamental limits of communication across a noisy channel. Consider the following discrete channel $p(y|x)$ between X and Y depicting noisy communication between a transmitter and a receiver.

An important question to address here is: What is the maximum rate of reliable information transfer between X and Y and how to achieve this maximum? To answer this question, we need the following definitions.

Index	*Dictionary String*	*Encoded Index*	*Added Bit*
000	*0*	*N/A*	*N/A*
001	*1*	*N/A*	*N/A*
010	*10*	*001*	*0*
011	*11*	*001*	*1*
100	*01*	*000*	*1*
101	*010*	*100*	*0*
110	*00*	*000*	*0*
111	*101*	*010*	*1*



Definition 17. *(Information Capacity) The* information capacity *between two random variables X and Y is defined as the maximum mutual information $I(X;Y)$ over all probability distribution on the input X*

$$C = \max_{p(x)} I(X;Y) \tag{3.20}$$

Definition 18. *(Code Rate)*
The rate of a code is defined as $R = \frac{\log_2 M}{n}$, where M is the number of messages and n is the number of channel use. Intuitively, R is the ratio between how many bits of messages are transmitted and how many bits are used for encoding.

Definition 19. *(Reliable Communication)*
A communication is said to be reliable *is the probability of transmission error is ideally zero.*

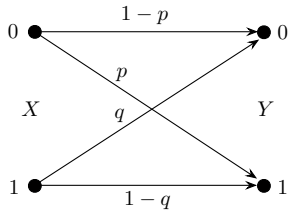
Definition 20. *(Achievable Rate)*
A rate R is achievable if there exists a code (inducing a probability distribution $p(x)$) of rate R that can be used to reliably *transmit M messages across the channel $p(y|x)$.*

Definition 21. *(Operational Capacity)*
The operational capacity *between two random variables X and Y is defined as the maximum of all achievable rates for the channel $p(y|X)$.*

Theorem 9. *(Shannon Channel Coding Theorem.)*
The Shannon theorem states that given a noisy channel $p(y|x)$ with information capacity C and information transmitted at a rate R, if $R < C$ there exist codes that allow the probability of error at the receiver to be made arbitrarily small. This means that, theoretically, it is possible to transmit information nearly without error at any rate below a limiting rate, C. Also, If $R > C$, an arbitrarily small probability of error is not achievable. All codes will have a probability of error greater than a certain positive minimal level, and this level increases as the rate increases. So, information cannot be guaranteed to be transmitted reliably across a channel at rates beyond the channel capacity. Intuitively, the theorem shows that the operational capacity and the information capacity are equal.

Example 34. *Let $X \in \mathcal{X} = \{0,1\}$ be a $Ber(\alpha)$ random variable, and let Y be generated from X via a binary channel with crossover probabilities $p(y|x)$ as shown below*

1. *Assume that $p = q$ (i.e., a binary symmetric channel (BSC)). Find the channel capacity C and the capacity achieving distribution in terms of p.*



2. *Let $p = 1/3$ and $q = 1/4$. Find the channel capacity C and the capacity achieving distribution.*

Solution

1. *Let $P(X = 0) = \alpha$ and $P(X = 1) = 1 - \alpha$. Then, $P(Y = 0) = \alpha + p - 2\alpha p$ and $P(Y = 1) = 1 - \alpha - p + 2\alpha p$. The mutual information $I(X; Y)$ is*

$$\begin{aligned} I(X;Y) &= H(Y) - H(Y|X) \\ &= H(Y) - P(X=0)H(Y|X=0) - P(X=1)H(Y|X=1) \\ &= H(Y) - \alpha h(p) - (1-\alpha)h(p) \\ &= H(Y) - h(p) \end{aligned}$$

Since p is a constant (related to channel modeling), then $I(X; Y)$ is maximized by maximizing $H(Y)$. Since $H(Y)$ is a binary entropy, then it attains the maximum (which is equal to 1) when $P(Y = 0) = P(Y = 1) = 1/2$. Since Y is induced by X through $p(y|x)$, we need to make sure that there exists a probability distribution on the input X that produces a $Ber(1/2)$ distribution at the channel's output Y. Solving $P(Y = 0) = \alpha + p - 2\alpha p = 1/2$, we get $\alpha = 1/2$. Hence, $X \sim Ber(1/2)$ is the capacity achieving distribution, because the resulting distribution at the output $Y \sim Ber(1/2)$ maximizes $H(Y)$ and in turn maximizes $I(X; Y)$. Therefore, the capacity is $C = 1 - h(p)$ bits/channel use.

2. *$P(Y = 0) = \frac{5\alpha}{12} + \frac{1}{4}$, $P(Y = 1) = -\frac{5\alpha}{12} + \frac{3}{4}$.*

$$\begin{aligned} I(X;Y) &= H(Y) - H(Y|X) \\ &= h\left(\frac{5\alpha}{12} + \frac{1}{4}\right) - P(X=0)H(Y|X=0) - P(X=1)H(Y|X=1) \\ &= h\left(\frac{5\alpha}{12} + \frac{1}{4}\right) - \alpha h\left(\frac{1}{3}\right) - (1-\alpha)h\left(\frac{1}{4}\right) \end{aligned}$$

$I(X; Y)$ is a function of α. In order to find the capacity, we need to find the maximum of $I(X; Y)$ with respect to α.

$$\frac{dI(X;Y)}{da} = \frac{5}{12}\log_2 \frac{\frac{5\alpha}{12} + \frac{1}{4}}{-\frac{5\alpha}{12} + \frac{3}{4}} - h\left(\frac{1}{3}\right) + h\left(\frac{1}{4}\right) = \frac{5}{12}\log_2 \frac{5\alpha + 3}{-5\alpha + 9} - h\left(\frac{1}{3}\right) + h\left(\frac{1}{4}\right)$$

Setting the derivative equal to zero

$$\frac{5}{12}\log_2 \frac{5\alpha + 3}{-5\alpha + 9} = h\left(\frac{1}{3}\right) - h\left(\frac{1}{4}\right) \Rightarrow \log_2 \frac{5\alpha + 3}{-5\alpha + 9} = \frac{12}{5}(0.91 - 0.81) = 0.24$$

Hence, $\frac{5\alpha+3}{-5\alpha+9} = 2^{0.24} = 1.18$, giving $\alpha = 0.7$. Thus, the capacity achieving distribution is $X \sim Ber(0.7)$ and the capacity is

$$\begin{aligned} C &= h\left(\frac{5(0.7)}{12} + \frac{1}{4}\right) - 0.7h\left(\frac{1}{3}\right) - 0.3h\left(\frac{1}{4}\right) \\ &= h(0.54) - 0.7h\left(\frac{1}{3}\right) - 0.3h\left(\frac{1}{4}\right) \\ &= 0.995 - (0.7)(0.91) - (0.3)(0.81) = 0.115 \text{ bits/channel use} \end{aligned}$$

3.7.1 Capacity of Bandlimited Channels

A common model for communication over a radio network or a telephone line is a bandlimited channel with white noise. This is a continuous-time channel. The output os such a channel can be described as the convolution

$$Y(t) = (X(t) + Z(t)) \star h(t) \tag{3.21}$$

where $X(t)$ is the signal waveform, $Z(t)$ is the waveform of the white Gaussian noise, and $h(t)$ is the impulse response of an ideal bandpass filter of bandwidth W. If the noise has power spectral density $N_0/2$ W/Hz, we can derive the capacity of such a channel, as given in the following theorem. This theorem is considered one of the most important results in digital communications.

Theorem 10. *(Capacity of AWGN Channel)*
For an AWGN channel of bandwidth W and received power P, the channel capacity is given by the formula

$$C = W \log_2 \left(1 + \frac{P}{N_0 W}\right) \quad bit/s \tag{3.22}$$

Proof. The proof of this theorem is beyond the scope of this course. □

Example 35. *Find the minimum signal-to-noise ratio (in dB) that can be tolerated in order to reliably transmit a digital bit stream at a rate of 1.544 Mbps over a 96 KHz band-limited channel.*

Solution. $C = 1.544$ *Mbps*, $W = 96$ *KHz*. Since

$$\frac{C}{W} = \log_2 \left(1 + \frac{P}{N}\right) \Rightarrow \frac{P}{N} = 65586 \Rightarrow SNR_{min} = 48.4 \text{ dB}$$

□

CHAPTER 4

RECEIVER DESIGN FOR AWGN BASEBAND COMMUNICATION

4.1 Introduction

In the case of baseband signaling, the received waveforms are already in a pulse-like form. One might ask, why then, is a demodulator needed to recover the pulse waveforms? The answer is that the arriving baseband pulses are not in the form of ideal pulse shapes, each one occupying its own symbol interval. The filtering at the transmitter and the channel typically cause the received pulse sequence to suffer from inter-symbol interference (ISI) and thus appear as an amorphous "smeared" signal, not quite ready for sampling and detection. The goal of the demodulator (receiving filter) is to recover a baseband pulse with the best possible signal-to-noise ratio (SNR), free of any ISI. Equalization, is a technique used to help accomplish this goal. The equalization process is not required for every type of communication channel. However, since equalization embodies a sophisticated set of signal-processing techniques, making it possible to compensate for channel-induced interference, it is an important area of many systems.

The bandpass model of the detection process (to be covered in the next chapter) is virtually identical to the baseband model considered in this chapter. That is because a received bandpass waveform is first transformed to a baseband waveform before the final detection step takes place. For linear systems, the mathematics of detection is unaffected by a shift in frequency. In fact, we can define an equivalence theorem as follows: Performing bandpass linear signal processing followed by heterodyning (frequency conversion or mixing that yields a spectral shift) the signal to baseband, yields the same results as heterodyning the bandpass signal to baseband, followed by baseband linear signal processing. As a result of this equivalence theorem, all linear signal-processing simulations can take place at baseband (which is preferred for simplicity) with the same results as at bandpass. This means that the performance of most digital communication systems will often be described and analyzed as if the transmission channel is a baseband channel.

The task of the detector is to retrieve the bit stream from the received waveform, as error free as possible, notwithstanding the impairments to which the signal may have been subjected. There are two primary causes for error-performance degradation. The first is the effect of filtering at the transmitter, channel, and receiver (causing smearing and ISI), and the second is the noise effect produced by a variety of sources, such as galactic and atmospheric noise, switching transients, intermodulation noise, as well as interfering signals from other sources. With proper precautions, much of the noise and interference entering a receiver can be reduced in intensity or even eliminated. However, there is one noise source that cannot be eliminated, and that is the noise caused by the thermal motion of electrons in any conducting media. This motion produces *thermal noise* in amplifiers and circuits, and corrupts the signal in an additive fashion. The primary statistical characteristic of thermal noise is that the noise amplitudes are distributed according to a normal or Gaussian distribution. It can be seen that the most probable noise amplitudes are those with small positive or negative values. In theory, the noise can be infinitely large, but very large noise amplitudes are rare. Using quantum mechanics, we can show that thermal noise is white, i.e., has a constant spectral density given by $N_0/2$. Since thermal noise is present in all communication systems and it the predominant noise source for many systems, the thermal noise characteristics (additive, white, and Gaussian, giving rise to the name AWGN) are most often

used to model the noise in the detection process and in the design of receivers. Fig. 4.1 shows a typical scenario of baseband pulses being affected by Gaussian noise and causing detection errors to occur at the receiver.

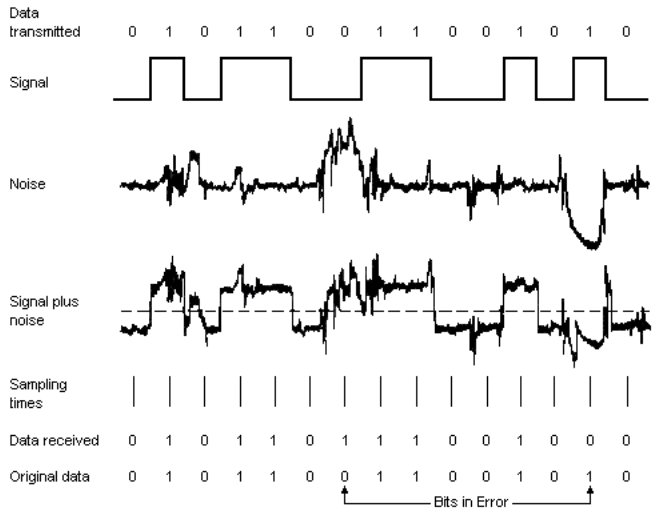


Figure 4.1: Baseband pulses affected by Gaussian noise.

4.2 Hypothesis Testing

Hypothesis testing refers to the problem of guessing the outcome of a random variable H that takes values in a finite alphabet $\mathcal{H} = \{0, 1, \dots, m-1\}$, based on the outcome of a random variable Y called *observable*.

This problem comes up in various applications under different names. Hypothesis testing is the terminology used in statistics. A receiver does hypothesis testing, but communication people call it *decoding*. An alarm system such as a smoke detector also does hypothesis testing, but people would call it *detection*. A more appealing name for hypothesis testing is *decision making*. Hypothesis testing, decoding, detection, and decision making are all synonyms.

In communication, the hypothesis H is the message to be transmitted and the observable Y is the channel output (or a sequence of channel outputs). The receiver guesses the realization of H based on the realization of Y. Unless stated otherwise, we assume that, for all $i \in \mathcal{H}$, the system designer knows $P_H(i)$ (called the*a priori probability*) and $f_{Y|H}(\cdot|i)$[1].

The receiver's decision will be denoted by $\hat{i}$ and the corresponding random variable $\hat{H} \in \mathcal{H}$. If we could, we would ensure that $\hat{H} = H$, but this is generally not possible. The goal is to devise a decision strategy that maximizes the probability $P_c = Pr\{\hat{H} = H\}$ that the decision is correct. An equivalent goal is to minimize the *error probability* $P_e = Pr\{\hat{H} \neq H\} = 1 - P_e$.

Hypothesis testing is at the heart of the communication problem. As described by *Claude Shannon*, "The fundamental problem of communication is that of reproducing at one point either exactly or approximately a message selected at another point".

4.2.1 MAP Decision Rule

From P_H and $f_{Y|H}$, via *Baye's rule*, we obtain

$$P_{H|Y}(i|y) = \frac{P_H(i) f_{Y|H}(y|i)}{f_Y(y)}$$

where $f_Y(y) = \sum_i P_H(i) f_{Y|H}(y|i)$. In the above expression $P_{H|Y}(i|y)$ is *a posteriori probability* of H give Y. If the decision is $\hat{H} = i$, the probability that it is the correct decision is the probability that $H = i$, i.e., $P_{H|Y}(i|y)$. As our goal is to maximize the probability of being correct, the *optimum decision rule*, also known as *Maximum A Posteriori (MAP) decision rule* is

$$\hat{H}(y) = arg \max_{i \in \mathcal{H}} P_{H|Y}(i|y) = arg \max_{i \in \mathcal{H}} P_H(i) f_{Y|H}(y|i) \tag{4.1}$$

[1]We assume that Y is a continuous random variable. If it is discrete, then we use $P_{Y|H}(\cdot|i)$ instead of $f_{Y|H}(\cdot|i)$

where $arg\max_i g(i)$ stands for "one of the arguments i for which the function $g(i)$ achieves its maximum". In case of ties, i.e., if $P_{H|Y}(j|y)$ equals $P_{H|Y}(k|y)$ equals $\max_i P_{H|Y}(i|y)$, then it does not matter if we decide for $\hat{H} = j$ or for $\hat{H} = k$. In either case, the probability that we have decided correctly is the same.

Because the MAP rule maximizes the probability of being correct for each observation y, it also maximizes the unconditional probability P_c of being correct. The former is $P_{H|Y}(\hat{H}(y)|y)$. If we plug in the random variable Y instead of y, then we obtain a random variable. The expected value of this random variable is the unconditional probability of being correct, i.e.,

$$P_c = \mathbb{E}\left[P_{H|Y}(\hat{H}(y)|y)\right] = \int_y P_{H|Y}(\hat{H}(y)|y) f_Y(y) dy \tag{4.2}$$

4.2.2 ML Decision Rule

There is an important special case, namely when H is uniformly distributed . In this case $P_{H|Y}(i|y)$, as a function of i, is proportional to $f_{Y|H}(y|i)$. Therefore, the argument that maximizes $P_{H|Y}(i|y)$ also maximizes $f_{Y|H}(y|i)$. Then, the MAP decision rule is equivalent to the following *Maximum Likelihood (ML) decision rule*[2].

$$\hat{H}(y) = arg\max_{i \in \mathcal{H}} f_{Y|H}(y|i) \tag{4.3}$$

Notice that the ML decision rule is defined even if we do not know P_H. Hence, it is the solution of choice when the prior is not known.

4.2.3 Binary Hypothesis Testing

The special case in which we have to make a binary decision, i.e., $\mathcal{H} = \{0, 1\}$, is both instructive and of practical relevance. As there are only two alternatives to be tested, the MAP test may now be written as

$$P_H(1) f_{Y|H}(y|1) \underset{\hat{H}_0}{\overset{\hat{H}_1}{\gtrless}} P_H(0) f_{Y|H}(y|0) \Rightarrow \Lambda(y) \triangleq \frac{f_{Y|H}(y|1)}{f_{Y|H}(y|0)} \underset{\hat{H}_0}{\overset{\hat{H}_1}{\gtrless}} \frac{P_H(0)}{P_H(1)} \triangleq \eta$$

The above test is depicted in Fig. 4.2 assuming $y \in \mathbb{R}$. This is a very important figure that helps us visualize what goes on and, as we will see, will be helpful to compute the probability of error. The left side of the above test is called the *likelihood ratio*, denoted by $\Lambda(y)$, whereas the right side is the *threshold* η.

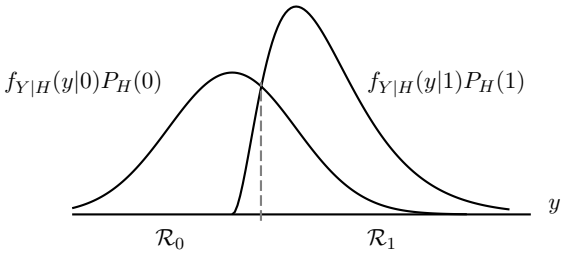


Figure 4.2: Binary MAP decision.

When $P_H(0) = P_H(1) = 1/2$, then $\eta = 1$ and the MAP test becomes a binary ML test:

$$f_{Y|H}(y|1) \underset{\hat{H}_0}{\overset{\hat{H}_1}{\gtrless}} f_{Y|H}(y|0)$$

The ML decision rule has a straightforward graphical interpretation. If the curve corresponding to the likelihood $f_{Y|H}(y|1)$ is above the curve corresponding to the likelihood $f_{Y|H}(y|0)$, then decide in favor of H_1, otherwise decide in favor of H_0. The MAP rule does not have this simple graphical interpretation because $\eta \neq 1$.

[2]The name stems from the fact that $f_{Y|H}(y|i)$, as a function of i, is called the *likelihood function*

Example 36. *(Optimal Decision Rules)*
Consider the following two likelihoods corresponding to hypotheses H_1 and H_0

$$\begin{aligned} f_{Y|H}(y|1) &= e^{-y}U(y) \\ f_{Y|H}(y|0) &= 2e^{-2y}U(y) \end{aligned}$$

1. *Find the ML decision rule.*
2. *Find the MAP decision rule with $P_H(0) = 2P_H(1)$.*

Solution.

1. The ML decision rule is $\hat{H}(y) = arg\max_{i\in\mathcal{H}} f_{Y|H}(y|i)$ for $i = 0$ and $i = 1$. In order to find the ML decision rule, we need to compare between the likelihoods $f_{Y|H}(y|1)$ and $f_{Y|H}(y|0)$, and find for what values of y one is larger than the other:

$$f_{Y|H}(y|1) \underset{\hat{H}_0}{\overset{\hat{H}_1}{\gtrless}} f_{Y|H}(y|0) \Rightarrow e^{-y} \underset{\hat{H}_0}{\overset{H_1}{\gtrless}} 2e^{-2y} \Rightarrow e^{y} \underset{\hat{H}_0}{\overset{\hat{H}_1}{\gtrless}} 2 \Rightarrow y \underset{\hat{H}_0}{\overset{\hat{H}_1}{\gtrless}} \ln(2)$$

   This implies that under the ML decision rule, if $y > \ln(2)$, then decide in favor of hypothesis H_1 and if $0 < y < \ln(2)$, decide in favor of hypothesis H_0.

2. The MAP decision rule is $\hat{H}(y) = arg\max_{i\in\mathcal{H}} P_H(i)f_{Y|H}(y|i)$ for $i = 0$ and $i = 1$. In order to find the MAP decision rule, we need to compare $P_H(0)f_{Y|H}(y|0)$ to $P_H(1)f_{Y|H}(y|1)$:

$$P_H(1)f_{Y|H}(y|1) \underset{\hat{H}_0}{\overset{\hat{H}_1}{\gtrless}} P_H(0)f_{Y|H}(y|0) \Rightarrow \frac{f_{Y|H}(y|1)}{f_{Y|H}(y|0)} \underset{\hat{H}_0}{\overset{\hat{H}_1}{\gtrless}} \frac{P_H(0)}{P_H(1)} \Rightarrow \frac{e^{y}}{2} \underset{\hat{H}_0}{\overset{\hat{H}_1}{\gtrless}} 2 \Rightarrow y \underset{\hat{H}_0}{\overset{\hat{H}_1}{\gtrless}} \ln(4)$$

   This implies that under the MAP decision rule, if $y > \ln(4)$, then decide in favor of hypothesis H_1 and if $0 < y < \ln(4)$, decide in favor of hypothesis H_0.

□

4.2.4 Performance Measure: Probability of Error

A function $\hat{H} : \mathcal{Y} \to \mathcal{H} = \{0, \dots, m-1\}$ is called a *decision function* (also called *decoding function*). One way to describe a decision function is by means of the decision regions $\mathcal{R}_i = \{u \in \mathcal{Y} : \hat{H}(y) = i\}$, $i \in \mathcal{H}$. Hence, $\mathcal{R}_i$ is the set of $y \in \mathcal{Y}$ for which $\hat{H}(y) = i$.

To compute the probability of error, it is often convenient to compute the error probability for each hypothesis and then take the average. When $H = 0$, the decision is incorrect if $Y \in \mathcal{R}_1$, or equivalently, if $\Lambda(y) \geq \eta$. Hence, denoting by $P_e(i)$ the error probability when $H = i$,

$$P_e(0) = Pr\{Y \in \mathcal{R}_i | H = 0\} = Pr\{\Lambda(Y) \geq \eta\} = \int_{\mathcal{R}_1} f_{Y|H}(y|0)dy \tag{4.4}$$

Similar expressions hold for the probability of error conditioned on $H = 1$, denoted by $P_e(1)$. Using the total law of probability, we obtain the unconditional error probability

$$P_e = P_e(1)P_H(1) + P_e(0)P_H(0)$$

Example 37. *(Probability of Error for a MAP Decision Rule)*
Consider the following two likelihoods corresponding to hypotheses H_1 and H_0

$$\begin{aligned} f_{Y|H}(y|1) &= e^{-y}U(y) \\ f_{Y|H}(y|0) &= 2e^{-2y}U(y) \end{aligned}$$

Find the probability of error corresponding to the MAP decision rule with $P_H(0) = 2P_H(1)$.

Solution. The MAP decision rule was previously derived and found to be

$$y \underset{\hat{H}_0}{\overset{\hat{H}_1}{\gtrless}} \ln(4)$$

Hence, $\mathcal{R}_0 = (0, \ln 4)$ and $\mathcal{R}_1 = (\ln 4, \infty)$. The conditional probability of error $P_e(1) = Pr\{Y \in \mathcal{R}_0 | H = 1\}$ can be computed as follows

$$P_e(1) = \int_{\mathcal{R}_0} f_{Y|H}(y|1)dy = \int_0^{\ln 4} e^{-y}dy = \frac{3}{4}$$

The conditional probability of error $P_e(0) = Pr\{Y \in \mathcal{R}_1 | H = 0\}$ can be computed as follows

$$P_e(0) = \int_{\mathcal{R}_1} f_{Y|H}(y|0)dy = \int_{\ln 4}^{\infty} 2e^{-2y}dy = \frac{1}{16}$$

Since $P_H(0) = 2P_H(1)$ and $P_H(0) + P_H(1) = 1$, then $P_H(0) = \frac{2}{3}$ and $P_H(1) = \frac{1}{3}$. As a result,

$$P_e = \frac{2}{3} \times \frac{1}{16} + \frac{1}{3} \times \frac{3}{4} = \frac{7}{24}$$

□

4.3 Demodulation and Detection for the AWGN Channel

During a given interval T, a binary baseband system will transmit one of two waveforms, denoted $s_1(t)$ and $s_0(t)$. Then, for any binary channel, the transmitted signal over a symbol interval $(0, T)$ is given by

$$s_i(t) = \begin{cases} s_1(t), & 0 \le t \le T \quad \textit{for a binary 1} \\ s_0(t), & 0 \le t \le T \quad \textit{for a binary 0} \end{cases}$$

The received signal $r(t)$, degraded by noise $n(t)$ and possibly degraded by the impulse response of the channel $h_c(t)$, is

$$r(t) = s_i(t) * h_c(t) + n(t), \quad i=1,...,M \tag{4.5}$$

where $n(t)$ is assumed to be a zero mean (additive White and Gaussian noise) AWGN process. For binary transmission over an ideal distortionless channel, the representation for $r(t)$ simplifies to

$$r(t) = s_i(t) + n(t), \quad i=1,...,M \tag{4.6}$$

We define *demodulation* as recovery of a waveform (to an undistorted baseband pulse), and we define *detection* to mean the decision-making process of selecting the digital meaning of that waveform. If error-correcting coding is not present, the detector output consists of estimates of message symbols (or bits), $\hat{m}_i$ (also called *hard decisions*). If error-correction coding is used, the detector output consists of estimates of channel symbols (or coded bits) $\hat{u}_i$, which can take the form of *hard* or *soft* decisions. For brevity, the term *detection* is occasionally used loosely to encompass all the receiver signal-processing steps through the decision making step. Fig.4.3 shows the two basic steps in demodulation and detection of digital signals.

Within the *demodulator* and *sample* block of Fig.4.3 is the *receiving filter* (essentially the demodulator), which performs waveform recovery in preparation for the next important step−detection. The *frequency down-conversion* is meant for bandpass

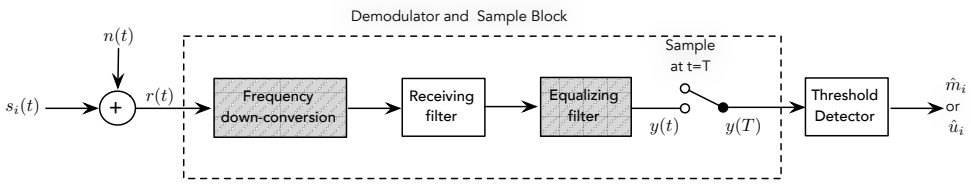


Figure 4.3: Basic steps in demodulation/detection of digital signals

signals. The filtering at the transmitter and the channel typically cause the received pulse sequence to suffer from ISI, and thus it is not quite ready for sampling and detection. The goal of the receiving filter is to recover a baseband pulse with the best possible SNR, free of any ISI. The optimum receiving filter for accomplishing this is called a *matched filter* or *correlator*. An optional *equalizing filter* follows the receiving filter; it is only needed for those systems where channel-induced ISI can distort the signals. The receiving filter and equalizing filter are shown as two separate blocks in order to emphasize their separate functions. In most cases, however, when an equalizer is used, a single filter would be designed to incorporate both functions and thereby compensate for the distortion caused by both the transmitter and the channel. Such a composite filter is sometimes referred to simply as the *equalizing filter*.

In this chapter, we will only focus on baseband communication and thus the down-conversion and equalization blocks will not be considered in our present treatment of demodulation and detection. Hence, we will consider the simplified block diagram shown in Fig. 4.4.

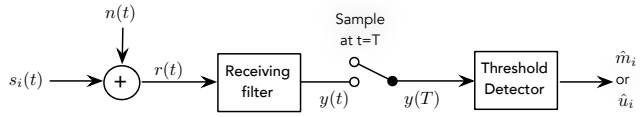


Figure 4.4: Demodulation/detection of baseband signals

4.3.1 The Matched Filter

The matched filter is a linear filter designed to provide the maximum SNR at its output for a given transmitted symbol waveform. Consider that a known signal $s(t)$ plus AWGN $n(t)$ is the input to a linear, time-invariant (receiving) filter followed by a sampler, as shown in Fig.4.4. We will determine the optimum receiver (*matched filter*) for detecting the known pulse $s(t)$ of duration T. The pulse shape is assumed to be known by the receiver. Hence, the only source of uncertainty is the noise. The AWGN noise process $n(t)$ is assumed to be zero-mean and of spectral height $N_0/2$. The received signal at the filter input is

$$r(t) = s(t) + n(t), \quad 0 \le t \le T$$

The signal $s(t)$ is modeled as a deterministic signal and the noise $n(t)$ is a AWGN process, which results in the signal $r(t)$ being modeled as a Gaussian process as well. The filter's output $y(t)$ can be written as

$$y(t) = \underbrace{s(t) * h(t)}_{\triangleq x(t)} + \underbrace{n(t) * h(t)}_{\triangleq w(t)} = x(t) + w(t)$$

where $h(t)$ denotes the impulse response of the receiving filter, assumed to be LTI. The signal $s(t)$ processed by an LTI system produces the signal $x(t)$ (a deterministic signal) and the random noise process $n(t)$ processed by an LTI system produces a Gaussian noise process $w(t)$ of zero-mean and of variance σ_w^2. At the sampler's output, we get

$$y(T) = x(T) + w(T)$$

where $x(T)$ is a constant and $w(T)$ is a Gaussian random variable of zero-mean and of variance σ_w^2. For simplicity, we will drop the T, and denote the threshold detector's in Fig. 4.3 by

$$Y = x + W \tag{4.7}$$

The receiver output signal-to-noise ratio SNR at time $t = T$ can be expressed as

$$SNR = \frac{P_x}{E[W^2(T)]} \tag{4.8}$$

Since the spectrum of $x(t)$ is $X(f) = S(f)H(f)$, then $x(t)$ can be expressed using the inverse Fourier transform as follows

$$x(t) = \int_{-\infty}^{\infty} S(f)H(f)e^{j2\pi ft}df$$

The average power of the signal, sampled at time T, at the output of the filter can be expressed as

$$P_x = |X|^2 = \left| \int_{-\infty}^{\infty} S(f)H(f)e^{j2\pi fT}df \right|^2 \tag{4.9}$$

The average power of the noise $w(t)$ at the filter output is

$$E\left[w^2(t)\right] = \frac{N_0}{2}\int_{-\infty}^{\infty} |H(f)|^2\, df \tag{4.10}$$

Replacing Eq.(4.9) and Eq.(4.10) in Eq.(4.8)

$$\begin{aligned} SNR &= \frac{\left| \int\limits_{-\infty}^{\infty} H(f)S(f)e^{j2\pi fT}df \right|^2}{\frac{N_0}{2}\int\limits_{-\infty}^{\infty} |H(f)|^2\, df} \\ &\leq \frac{\int\limits_{-\infty}^{\infty} |H(f)|^2\, df \int\limits_{-\infty}^{\infty} |S(f)|^2\, df}{\frac{N_0}{2}\int\limits_{-\infty}^{\infty} |H(f)|^2\, df} \qquad (4.11) \\ &\leq \frac{2}{N_0}\int_{-\infty}^{\infty} |S(f)|^2\, df \qquad (4.12) \end{aligned}$$

where Eq.(4.11) is due to *Schwarz's inequality* which states the following

$$\left| \int_{-\infty}^{\infty} f(a)g(a)da \right|^2 \leq \int_{-\infty}^{\infty} |f(a)|^2\, da \int_{-\infty}^{\infty} |g(a)|^2\, da$$

with equality if $f(a) = kg^*(a)$, where k is an arbitrary constant and $*$ denotes the complex conjugate. The right-hand side of Eq.(4.12) represents the maximum value that can be assumed by the receiver output SNR. To achieve this maximum value, we use the condition under which *Schwarz's inequality* is satisfied with equality. Let $f(a) \triangleq H(f)$ and $g(a) \triangleq S(f)e^{j2\pi fT}$, we obtain

$$H(f)_{opt} = k\ S^*(f)e^{-j2\pi fT} \tag{4.13}$$

Under this condition,

$$SNR = \frac{2}{N_0}\int_{-\infty}^{\infty} |S(f)|^2\, df$$

and the filter is called optimum filter. To obtain the impulse response of the optimum filter, we use the inverse Fourier transform

$$h_{opt}(t) = k\int_{-\infty}^{\infty} S^*(f)e^{-j2\pi fT}e^{j2\pi ft}df = k\int_{-\infty}^{\infty} S^*(f)e^{-j2\pi f(T-t)}df$$

Assume that $s(t)$ is a real valued signal, then $S^*(f) = S(-f)$. Hence,

$$h_{opt}(t) = \int_{-\infty}^{\infty} S(-f)e^{-j2\pi f(T-t)}df = \int_{-\infty}^{\infty} S(f)e^{j2\pi f(T-t)}df$$

Thus,

$$h_{opt}(t) = k\ s(T-t) \tag{4.14}$$

the constant k can be set to 1 or it can be considered as a normalization constant used to make the energy of $h_{opt}(t)$ equal to 1. The optimum filter is called a *matched filter* because $h_{opt}(t)$ is a time reversed and delayed version of $s(t)$; that is, it is "matched" to the input signal $s(t)$. In general the delay can be a time t_0 of the peak signal output, i.e., $h_{opt}(t) = s(t_0 - t)$.

We finally note that the maximum value of the output SNR can be expressed in the following manner

$$(SNR)_{max} = \frac{2}{N_0}\int_{-\infty}^{\infty} |S(f)|^2\, df = \frac{2E}{N_0} \tag{4.15}$$

where E is the energy of the pulse present at the input of the receiver (due to *Plancherel's theorem*). The use of the matched filter removes the dependence on the shape of the input signal $s(t)$, in the sense that all signals $s(t)$ having the same energy E produce the same output signal-to-noise ratio irrespective of their shapes. This is, of course, true provided that for each signal shape a corresponding matched filter is used. In conclusion, if a signal $s(t)$ is corrupted by AWGN, the filter with an impulse response matched to $s(t)$ maximizes the output SNR.

Example 38. *Consider a matched filter receiver with input $s(t) = A$, $0 \le t \le T$ under AWGN. The filter has an impulse response that is normalized in $[0, T]$ and matched to $s(t)$. Determine the maximum value of the matched filter output and the time instant at which this maximum is reached.*

Solution. Let $h(t)$ be the impulse response of the matched filter. Since it is matched to $s(t) = A$, then $h(t) = k\,A$, $0 \le t \le T$. The matched filter's power is normalized to 1, hence $k^2A^2T = 1 \Rightarrow k = 1/A\sqrt{T}$. The matched filter output is $y(t) = s(t) * h(t)$ (the convolution of two rect functions):

$$y(t) = \int_{-\infty}^{\infty} s(\tau)h(t-\tau)d\tau = \begin{cases} \frac{A}{\sqrt{T}}t, & 0 \le t \le T \\ -\frac{A}{\sqrt{T}}t + 2A\sqrt{T}, & T \le t \le 2T \end{cases}$$

Sampling the output $y(t)$ at $t = T$, we get the maximum value $y(T) = A\sqrt{T}$, which occurs at $t = T$. □

Example 39. *Suppose that the known signal is the rectangular pulse*

$$s(t) = \begin{cases} 1, & t_1 \le t \le t_2 \\ 0, & \textit{otherwise} \end{cases}$$

The duration is $T = t_2 - t_1$. Then, for the case of White noise, the impulse response required for the matched filter is

$$h(t) = s(t_0 - t)$$

It is obvious that for the matched filter to be causal we need $t_0 \ge t_2$. In order to minimize the time that we have to wait before the maximum signal level occurs at the filter's output, we should pick $t_0 = t_2$. Hence, $h(t) = s(t_2 - t)$.

4.3.2 Threshold Detector and Error Probability

In this section, we derive a formula that can be used to compute the *bit error rate* (BER) (or bit probability of error) that results from the detection process that needs to be implemented at the receiver. Consider a transmitter which uses the polar

non-return to zero (PNRZ) signaling technique, and that the additive noise process at the receiver input is a zero-mean white and Gaussian process with a spectral height equal to $N_0/2$. The pulse shape used in the PNRZ is considered rectangular with positive pulse amplitude representing the bit 1 and negative pulse amplitude representing the bit 0. The pulse shape is known at the receiver. However, due to the presence of the corruptive noise process, the polarity of the pulse during each signaling interval (bit duration, T) is unknown. It is this polarity that needs to be determined or decided upon. This leads to a decision about the transmitted bit (a binary decision-making).The PNRZ binary data plus noise is present at the input of a receiver, which is chosen to be matched filter since the noise is white. The output of the matched filter is sampled at $t = T$. The sampled output $y(T)$ is fed to a decision device to decide on the polarity of the transmitted bit.

In general, we can represent the matched filter's input $r(t)$ as:

$$r(t) = \begin{cases} s_1 + n(t), & 0 \leq t \leq T \quad \leftarrow \textit{bit 1 is sent} \\ s_0 + n(t), & 0 \leq t \leq T \quad \leftarrow \textit{bit 0 is sent} \end{cases}$$

Hence, the filter's output yields

$$Y = \begin{cases} x_1 + W, & 0 \leq t \leq T \quad \leftarrow \textit{bit 1 is sent} \\ x_0 + W, & 0 \leq t \leq T \quad \leftarrow \textit{bit 0 is sent} \end{cases} \tag{4.16}$$

where x_i is the desired signal component, and W is the noise component. The noise component n_0 is a zero mean Gaussian random variable, and thus Y is a Gaussian random variable with a mean of either x_0 or x_1 depending on whether a binary one or a binary zero was sent. The variance of Y is equal to the variance of W:

$$\sigma_Y^2 = \sigma_W^2 = \frac{N_0}{2} E_h \tag{4.17}$$

where E_h is the energy of the matched filter.

For the AWGN channel, the problem of detection can be re-formulated as the following binary hypothesis testing problem

$$H_1 \quad : \quad Y \sim \mathcal{N}(x_1, \sigma_W^2) \tag{4.18}$$

$$H_0 \quad : \quad Y \sim \mathcal{N}(x_0, \sigma_W^2) \tag{4.19}$$

As a result, the output statistic for each hypothesis is

$$f_{Y|H}(y|0) \quad = \quad \frac{1}{\sqrt{2\pi\sigma_W^2}} exp\left\{-\frac{(y-x_0)^2}{2\sigma_W^2}\right\} \tag{4.20}$$

$$f_{Y|H}(y|1) \quad = \quad \frac{1}{\sqrt{2\pi\sigma_W^2}} exp\left\{-\frac{(y-x_1)^2}{2\sigma_W^2}\right\} \tag{4.21}$$

Assuming the optimal detector used in a MAP detector, we can compute the likelihood ratio

$$\Lambda(y) = \frac{f_{Y|H}(y|1)}{f_{Y|H}(y|0)} = exp\left\{y\frac{x_1 - x_0}{\sigma_W^2} + \frac{x_0^2 - x_1^2}{2\sigma_W^2}\right\}$$

The threshold is

$$\eta = \frac{P_H(0)}{P_H(1)} = \frac{p_0}{p_1}$$

Now we have all the ingredients to derive the MAP detection rule. Instead of comparing $\Lambda(y)$ to the threshold η, we can compare $\ln \Lambda(y)$ (the *log likelihood function*) to $\ln \eta$. Hence, for the binary AWGN channel, the MAP detection rule can be expressed as

$$y\frac{x_1 - x_0}{\sigma_W^2} + \frac{x_0^2 - x_1^2}{2\sigma_W^2} \underset{\hat{H}_0}{\overset{\hat{H}_1}{\gtrless}} \ln \eta \tag{4.22}$$

Without loss of generality, assume $x_1 > x_0$. Then we can divide both sides by $\frac{x_1 - x_0}{\sigma_W^2}$ (which is positive) without changing the outcome of the above comparison. We can further simplify by moving the constants to the right. The result is the simple test

$$\hat{H}_{MAP}(y) = \begin{cases} 1, & y \geq \theta \\ 0, & y < \theta \end{cases} \tag{4.23}$$

where

$$\theta = \frac{\sigma_W^2}{x_1 - x_0} \ln \eta + \frac{x_0 + x_1}{2} \tag{4.24}$$

Assuming now that the optimal detector used is an ML detector ($p_0 = p_1 = 1/2$, i.e., $\eta = 1$). The threshold θ becomes the midpoint $\frac{x_0+x_1}{2}$. See Fig. 4.5. This threshold is the *optimum threshold* for minimizing the probability of making an incorrect decision for this important special case of ML detection.

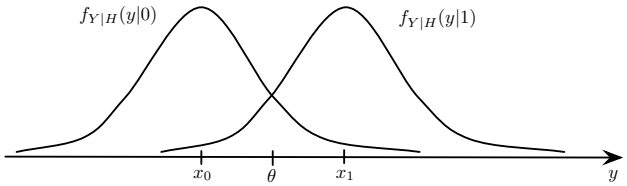


Figure 4.5: For the ML detector, the decision threshold θ is the midpoint between x_0 and x_1.

We now determine the probability of error. For the binary decision-making depicted in Fig.4.5, there are two ways errors can occur. The first type of error results from deciding that $s_0(t)$ was transmitted, while $s_1(t)$ was the transmitted signal (i.e., $s_1(t)$ was sent, and the channel noise results in Y being less than θ). The second type of error results from deciding that $s_1(t)$ was transmitted, while $s_0(t)$ was the transmitted signal (i.e., $s_0(t)$ was sent, and the channel noise results in Y being greater than θ).

$$\begin{aligned} P_e &= P_e(0)P_H(0) + P_e(1)P_H(1) \\ &= P_e(0)p_0 + P_e(1)p_1 \\ &= Pr\{Y > \theta | H = 0\}p_0 + Pr\{Y < \theta | H = 1\}p_1 \\ &= p_0 \int_\theta^\infty f_{Y|H}(y|0)dy + p_1 \int_{-\infty}^\theta f_{Y|H}(y|1)dy \\ &= p_0 Q\left(\frac{\theta - x_0}{\sigma_W}\right) + p_1 Q\left(\frac{x_1 - \theta}{\sigma_W}\right) \end{aligned} \tag{4.25}$$

For the case of equal priors (ML detection), $\theta = \frac{x_0+x_1}{2}$, and the probability of error is reduced to

$$P_e = Q\left(\frac{x_1 - x_0}{2\sigma_W}\right) \tag{4.26}$$

As we can see, the probability of error depends on the difference $(x_1 - x_0)$. The bigger this difference, the smaller the error probability p_e (the Q function is non-increasing). There is a nice geometrical interpretation for this result. The difference $(x_1 - x_0)$ represents the distance between the two signal $s_1(t)$ and $s_0(t)$, the bigger this difference is, the less we can confuse between them, hence, the smaller the probability of error, and vice versa. But how big can we make this distance?

4.3.3 Optimizing the Error Performance

To optimize P_e in the context of AWGN channel and the receiver in Fig. 4.4, we need to select the optimum receiving filter and the optimum decision threshold. For the binary case, the optimum decision threshold has been derived, and it was shown that this threshold results in $P_e = Q[(x_1 - x_0)/2\sigma_W]$. Next, for minimizing P_e, it is necessary to choose the matched filter that maximizes the argument of $Q(\cdot)$. Thus, we need to determine the linear filter that maximizes $(x_1 - x_0)/2\sigma_W$, or equivalently, that maximizes

$$\frac{(x_1 - x_0)^2}{\sigma_W^2} \tag{4.27}$$

where $(x_1 - x_0)$ is the difference between the desired signal components at the filter output at time $t = T$, and the square of this difference signal is the instantaneous power of the difference signal.

Since a matched filter maximizes the output SNR for a given known signal, here we view the optimum filter as one that maximizes the difference between two possible signal outputs. From Equation.(4.15), it was shown that a matched filter achieves the maximum possible output SNR equal to $2E/N_0$. Consider that the filter is matched to the input difference $[s_1(t) - s_0(t)]$; thus we can write an output SNR at time $t = T$ as

$$\left(\frac{S}{N}\right)_T = \frac{(x_1 - x_0)^2}{\sigma_W^2} = \frac{2E_d}{N_0} \tag{4.28}$$

where $N_0/2$ is the two-sided power spectral density and E_d is the energy of the difference signal at the filter input given by

$$E_d = \int_0^T [s_1(t) - s_0(t)]^2 dt \tag{4.29}$$

By maximizing the output SNR as shown in Equation.(4.28), the matched filter provides the maximum distance (normalized by noise) between the two candidate outputs.

Next, combining Eq. (4.25) and Eq. (4.28) yields

$$P_e = Q\left(\sqrt{\frac{E_d}{2N_0}}\right) \tag{4.30}$$

Returning to Eq. (4.29), we can expand it as follows

$$\begin{aligned} E_d &= \int_0^T [s_1(t) - s_2(t)]^2 dt \\ &= \int_0^T s_1^2(t)dt + \int_0^T s_2^2(t)dt - 2\int_0^T s_1(t)s_2(t)dt \\ &= E_b + E_b - 2\int_0^T s_1(t)s_2(t)dt \\ &= 2E_b - 2\int_0^T s_1(t)s_2(t)dt \end{aligned} \tag{4.31}$$

where E_b is the bit energy (energy of $s_i(t)$, $i = 1, 2$).
Here we distinguish between two general class of signals; *antipodal signaling* and *orthogonal signaling*.
Antipodal Signaling
Antipodal signals are signals which are mirror images to each other, i.e., $s_1(t) = -s_2(t)$. In this case, Eq. (4.31) becomes

$$E_d = 2E_b + 2\int_0^T s_1^2 dt = 4E_b \tag{4.32}$$

Replacing Eq. (4.32) in Eq. (4.30) yields

$$P_e = Q\left(\sqrt{\frac{2E_b}{N_0}}\right) \tag{4.33}$$

Orthogonal Signaling
Orthogonal signals are signals satisfying

$$\int_0^T s_1(t)s_2(t)dt = 0$$

Replacing in Eq.(4.31) we get

$$E_d = 2E_b \tag{4.34}$$

Replacing Eq.(4.34) in Eq.(4.30) yields

$$P_e = Q\left(\sqrt{\frac{E_b}{N_0}}\right) \tag{4.35}$$

As we can see by comparing Eq. (4.33) and Eq. (4.35), antipodal signaling requires a factor of 2 increase in energy compared to orthogonal signaling. Since $10\log_{10}(2) \simeq 3$ *dB*, we say that antipodal signaling offers a 3 *dB* better error performance than orthogonal signaling. This result is illustrated in Fig. 4.6.

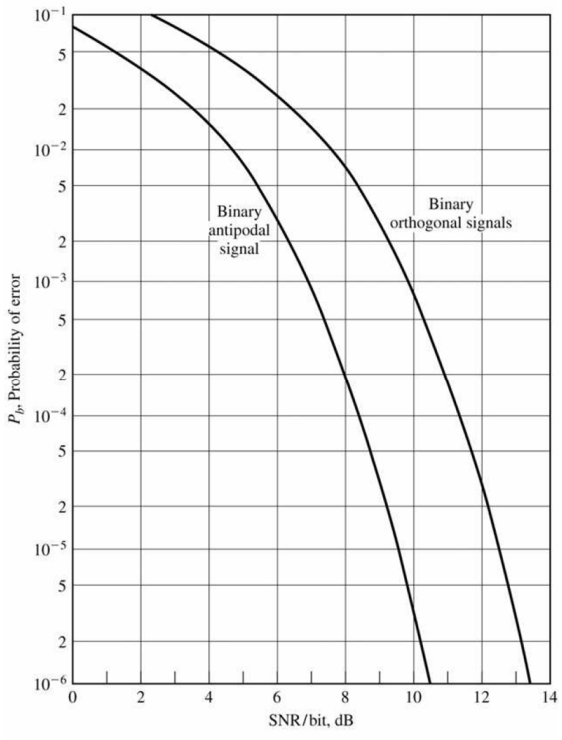


Figure 4.6: Error probability comparison between Antipodal and orthogonal signaling.

4.3.4 Correlation Realization of the Matched Filter

Eq. (4.14) illustrates the matched filter's basic property: the impulse response $h(t)$ of the filter is a delayed version of the mirror image of the signal $s(t)$, i.e., $h(t) = s(T - t)$. Let the received input waveform $r(t) = s(t) + n(t)$ be at the input of the matched filter and denote by $y(t)$ its output. Then, we can write

$$y(t) = r(t) * h(t) = \int_0^t r(\tau)h(t - \tau)d\tau \tag{4.36}$$

Substituting $h(t) = s(T - t)$ into $h(t - \tau)$ in Eq.(4.36) we get

$$y(t) = \int_0^t r(\tau)s[T - (t - \tau)]d\tau = \int_0^t r(\tau)s(T - t + \tau)d\tau \tag{4.37}$$

When $t = T$, we can write Eq.(4.37) as

$$y(T) = \int_0^T r(\tau)s(\tau)d\tau = \langle r, s\rangle \tag{4.38}$$

The operation of Eq. (4.38), the product integration of the received signal $r(t)$ with a replica of the transmitted waveform $s(t)$ over one symbol interval, is the *correlation* of $r(t)$ with $s(t)$. Consider that a received signal $r(t)$ is correlated with each prototype signal $s_i(t)$ $(i = 1, ..., M)$, using a bank of M correlators. See Fig.4.7. The signal $s_i(t)$ whose product integration or correlation with $r(t)$ yields the maximum output $z_i(T)$ is the signal that matches $r(t)$ better than all the other $s_j(t), j \neq i$. We will subsequently use this correlation characteristic for the optimum detection of signals.

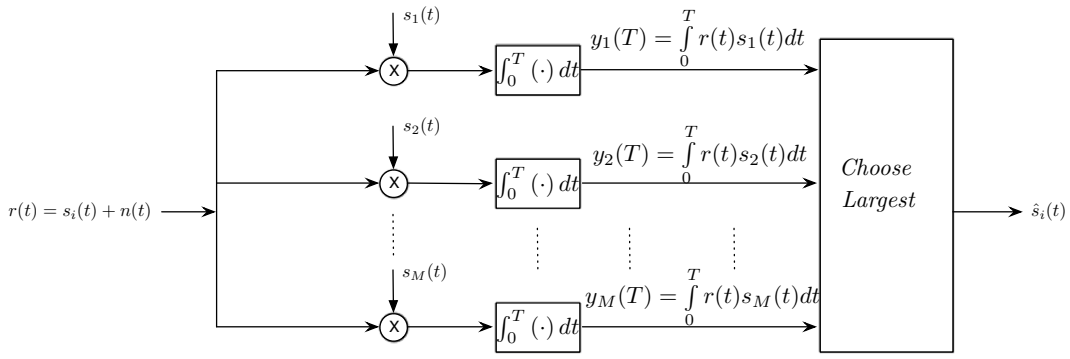


Figure 4.7: Bank of Correlators.

CHAPTER 5

BANDPASS COMMUNICATION

In baseband data transmission an incoming serial data stream is represented in the form of a discrete pulse-amplitude modulated wave that can be transmitted over a low-pass channel (e.g., a coaxial cable). What if the requirement is to transmit the data stream over a band-pass channel, exemplified by wireless and satellite channels? In applications of this kind, we usually resort to the use of a modulation strategy configured around a sinusoidal carrier whose amplitude, phase, or frequency is varied in accordance with the information-bearing data stream. Digital modulation/demodulation techniques dealing with band-pass data transmission are studied in this chapter.

The primary aim of the chapter is to describe some important digital band-pass modulation techniques used in practice. In particular, we describe three basic modulation schemes: namely, *amplitude-shift keying* (ASK), *phase-shift keying* (PSK), and *frequency-shift keying* (FSK), followed by some of their variants. Another issue that will receive particular attention is that of coherent versus non-coherent detection. A digital communication system is said to be coherent if the receiver is synchronized to the transmitter with respect to carrier phase; otherwise, the system is said to be non-coherent. Naturally, a non-coherent system offers the practical advantage of reduced complexity but at the cost of degraded performance.

5.1 Introduction

Given a binary source that emits symbols 0 and 1, the modulation process involves switching or keying the amplitude, phase, or frequency of a sinusoidal carrier wave between a pair of possible values in accordance with symbols 0 and 1. To be more specific, consider the sinusoidal carrier

$$c(t) = A_c \cos(2\pi f_c t + \phi_c) \tag{5.1}$$

where A_c is the carrier amplitude, f_c is the carrier frequency, and f_c is the carrier phase. Given these three parameters of the carrier $c(t)$, we may now identify three distinct forms of binary modulation:

1. *Binary amplitude shift-keying (BASK)*, in which the carrier frequency and carrier phase are both maintained constant, while the carrier amplitude is keyed between the two possible values used to represent symbols 0 and 1.

2. *Binary phase-shift keying (BPSK)*, in which the carrier amplitude and carrier frequency are both maintained constant, while the carrier phase is keyed between the two possible values (e.g., 0° and 180°) used to represent symbols 0 and 1.

3. *Binary frequency-shift keying (BFSK)*, in which the carrier amplitude and carrier phase are both maintained constant, while the carrier frequency is keyed between the two possible values used to represent symbols 0 and 1.

In light of these definitions, we see that BASK, BPSK, and BFSK are special cases of amplitude modulation, phase modulation, and frequency modulation, respectively. Figure.5.1 illustrates these three basic forms of binary signaling.

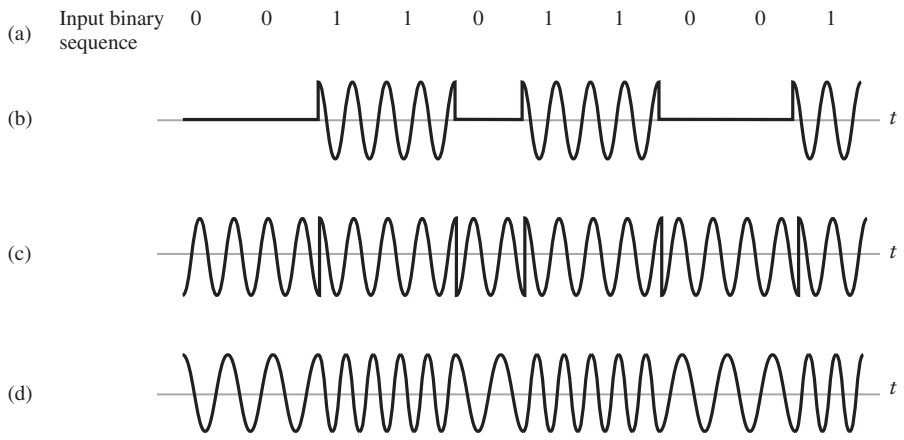


Figure 5.1: The three basic forms of signaling binary information. (a) Binary data stream. (b) Amplitude-shift keying. (c) Phase-shift keying. (d) Frequency-shift keying with continuous phase.

In the digital communications literature, the usual practice is to assume that the carrier $c(t)$ has unit energy measured over one symbol (bit) duration. Specifically, the carrier amplitude expressed in terms of the bit duration T_b is

$$A_c = \sqrt{\frac{2}{T_b}} \tag{5.2}$$

We may thus express the carrier $c(t)$ in the equivalent form

$$c(t) = \sqrt{\frac{2}{T_b}} \cos(2\pi f_c t + \phi_c) \tag{5.3}$$

One lesson learned from the material covered in previous chapters is the fact that the transmission bandwidth requirement of an angle-modulated wave is greater than that of the corresponding amplitude-modulated wave. In light of that lesson, we may say that the transmission bandwidth requirement of BFSK is greater than that of BASK for a given binary source. However, the same does not hold for BPSK, as we shall see from the material presented in this chapter. This is one of many differences that distinguish digital modulation from analog modulation.

The spectrum of a digitally modulated wave, exemplified by BASK, BPSK and BFSK, is centered on the carrier frequency f_c, implicitly or explicitly. Moreover, as with analog modulation, it is normal practice to assume that the carrier frequency f_c is large compared with the "bandwidth" of the incoming binary data stream that acts as the modulating signal. This band-pass assumption has certain implications, as discussed next. To be specific, consider a linear modulation scheme for which the modulated wave is defined by

$$s(t) = b(t)c(t) \tag{5.4}$$

where $b(t)$ denotes an incoming binary wave. Then, setting the carrier phase $\phi_c = 0$ for convenience of presentation, we may express $s(t)$ as

$$s(t) = \sqrt{\frac{2}{T_b}} b(t) \cos(2\pi f_c t) \tag{5.5}$$

Under the assumption $f_c \gg W$, where W is the bandwidth of the binary wave $b(t)$, there will be no spectral overlap in the generation of $s(t)$ (i.e., the spectral content of the modulated wave for positive frequencies is essentially separated from its spectral content for negative frequencies). Another implication of the band-pass assumption is that we may express the transmitted signal energy per bit as

$$E_b = \int_0^{T_b} |s(t)|^2 dt = \frac{1}{T_b} \int_0^{T_b} |b(t)|^2 dt + \underbrace{\frac{1}{T_b} \int_0^{T_b} |b(t)|^2 \cos(4\pi f_c t) dt}_{\simeq 0} \simeq \frac{1}{T_b} \int_0^{T_b} |b(t)|^2 dt \tag{5.6}$$

5.2 Bandpass Modulation Schemes

5.2.1 Binary Amplitude-Shift Keying (ASK)

Binary amplitude-shift keying (BASK) is one of the earliest forms of digital modulation used in radio telegraphy at the beginning of the twentieth century. To formally describe BASK, consider a binary data stream $b(t)$ which is of the ON-OFF signaling variety. That is, $b(t)$ is defined by

$$b(t) = \begin{cases} \sqrt{E_b}, & \textit{for binary symbol 1} \\ 0, & \textit{for binary symbol 0} \end{cases} \tag{5.7}$$

Then, multiplying $b(t)$ by the sinusoidal carrier wave $c(t)$ with the phase ϕ_c set equal to zero for convenience of presentation, we get the BASK wave

$$s(t) = \begin{cases} \sqrt{\frac{E_b}{T_b}}\cos(2\pi f_c t), & \textit{for binary symbol 1} \\ 0, & \textit{for binary symbol 0} \end{cases} \tag{5.8}$$

The carrier frequency f_c may have an arbitrary value, consistent with transmitting the modulated signal anywhere in the electromagnetic radio spectrum, so long as it satisfies the band-pass assumption.

A property of BASK that is immediately apparent from Fig.5.1(b), which depicts the BASK waveform corresponding to the incoming binary data stream of Fig.5.1(a), is the non-constancy of the envelope of the modulated wave. Accordingly, insofar as detection of the BASK wave is concerned, the simplest way is to use an envelope detector, exploiting the non-constant-envelope property of the BASK signal.

5.2.2 Binary Phase-Shift Keying (PSK)

In the simplest form of phase-shift keying known as binary phase-shift keying (BPSK), the pair of signals $s_1(t)$ and $s_2(t)$ used to represent symbols 1 and 0, respectively, are defined by

$$s_i(t) = \begin{cases} \sqrt{\frac{2E_b}{T_b}}\cos(2\pi f_c t), & \textit{for symbol 1 corresponding to } i = 1 \\ \sqrt{\frac{2E_b}{T_b}}\cos(2\pi f_c t + \pi) = -\sqrt{\frac{2E_b}{T_b}}\cos(2\pi f_c t), & \textit{for symbol 0 corresponding to } i = 2 \end{cases} \tag{5.9}$$

where $0 \leq t \leq T_b$. See Figure.5.1(c) for a representation example of BPSK. The pair of sinusoidal waves, $s_1(t)$ and $s_2(t)$, which differ only in a relative phase-shift of π are antipodal signals. We see that BPSK is a special case of double-sideband suppressed-carried (DSB-SC) modulation. BPSK differs from BASK in an important respect: the envelope of the modulated signal $s(t)$ is maintained constant for all time t. This property has two important consequences:

- The transmitted energy per bit, E_b is constant; equivalently, the average transmitted power is constant.
- Demodulation of BPSK cannot be performed using envelope detection; rather, we have to look to coherent detection.

5.2.3 Quadriphase-Shift Keying (QPSK)

An important goal of digital communication is the efficient utilization of channel bandwidth. This goal is attained by a bandwidth-conserving modulation scheme known as quadriphase-shift keying, which builds on the same idea as that of quadrature-carrier multiplexing. In quadriphase-shift keying (QPSK), as with BPSK, information carried by the transmitted signal is contained in the phase of a sinusoidal carrier. In particular, the phase of the sinusoidal carrier takes on one of four equally spaced values, such as $\pi/4$, $3\pi/4$, $5\pi/4$, and $7\pi/4$. For this set of values, we define the transmitted signal as

$$s_i(t) = \begin{cases} \sqrt{\frac{2E}{T}}\cos\left[2\pi f_c t + (2i-1)\frac{\pi}{4}\right], & 0 \leq t \leq T \\ 0, & \textit{elsewhere} \end{cases} \tag{5.10}$$

where $i = 1, 2, 3, 4$; E is the transmitted signal energy per symbol and T is the symbol duration. Each one of the four equally spaced phase values corresponds to a unique pair of bits called a *dibit*. For example, we may choose the foregoing set of phase values to represent the Gray encoded set of dibits: 10, 00, 01, and 11. In this form of encoding, we see that only a single bit is

changed from one dibit to the next. Note that the symbol duration (i.e., the duration of each dibit) is twice the bit duration, as shown by

$$T = 2T_b$$

Using a well-known trigonometric identity, we may recast the transmitted signal in the interval $0 \leq t \leq T$ in the expanded form

$$s_i(t) = \sqrt{\frac{2E}{T}} \cos\left[(2i-1)\frac{\pi}{4}\right] \cos(2\pi f_c t) - \sqrt{\frac{2E}{T}} \sin\left[(2i-1)\frac{\pi}{4}\right] \sin(2\pi f_c t) \tag{5.11}$$

In fact, the QPSK signal consists of the sum of two BPSK signals.

The QPSK receiver consists of an in-phase (I)-channel and quadrature (Q)-channel with a common input, as depicted in Fig.5.2. Each channel is itself made up of a product modulator, low-pass filter, sampler, and decision-making device. Under ideal conditions, the $I-$ and $Q-$channels of the receiver, respectively, recover the demultiplexed components $a_1(t)$ and $a_2(t)$ responsible for modulating the orthogonal pair of carriers in the transmitter. Accordingly, by applying the outputs of these two channels to a multiplexer (consisting of a parallel-to-serial converter), the receiver recovers the original binary sequence. The design of the QPSK receiver builds on the strategy described for the coherent BPSK receiver. Specifically, each of the two low-pass filters in the coherent QPSK receiver of Fig.5.2 must be assigned a bandwidth equal to or greater than the reciprocal of the symbol duration T for satisfactory operation of the receiver.

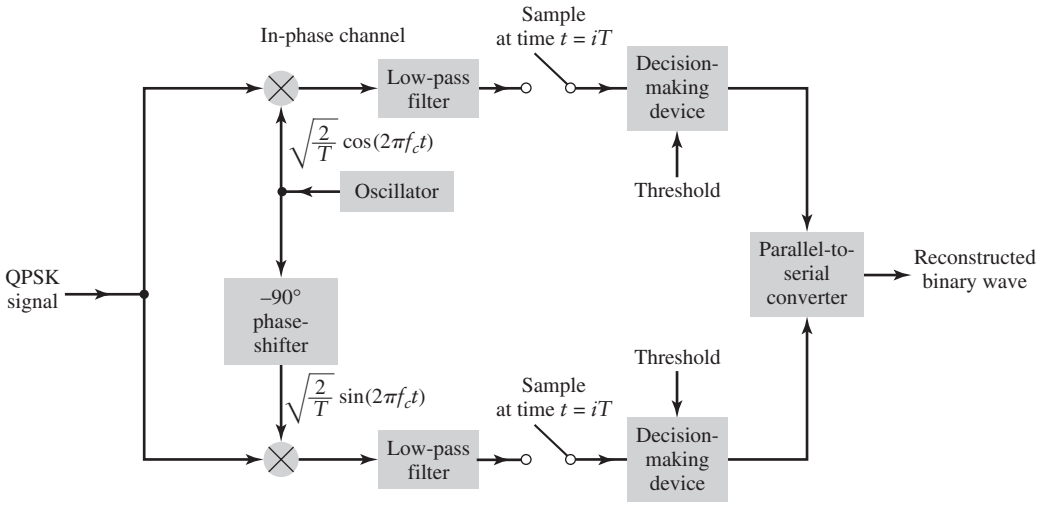


Figure 5.2: Block diagram of a QPSK receiver.

5.2.4 Binary Frequency-Shift Keying (FSK)

In the simplest form of frequency-shift keying known as binary frequency-shift keying (BFSK), symbols 0 and 1 are distinguished from each other by transmitting one of two sinusoidal waves that differ in frequency by a fixed amount. A typical pair of sinusoidal waves is described by

$$s_i(t) = \begin{cases} \sqrt{\frac{2E_b}{T_b}} \cos(2\pi f_1 t), & \textit{for symbol 1 corresponding to } i = 1 \\ \sqrt{\frac{2E_b}{T_b}} \cos(2\pi f_2 t), & \textit{for symbol 0 corresponding to } i = 2 \end{cases} \tag{5.12}$$

When the frequencies f_1 and f_2 are chosen in such a way that they differ from each other by an amount equal to the reciprocal of the bit duration T_b, the BFSK signal is referred to as *Sunde's* BFSK after its originator. This modulated signal is a continuous-phase signal in the sense that phase continuity is always maintained, including the inter-bit switching times.

Fig. 5.3 plots the waveform of Sunde's BFSK produced by the input binary sequence 0011011001 for a bit duration $T_b = 1$ s. Part (a) of the figure displays the waveform of the input sequence, and part (b) displays the corresponding waveform of the BFSK signal. The latter part of the figure clearly displays the phase-continuous property of BFSK.

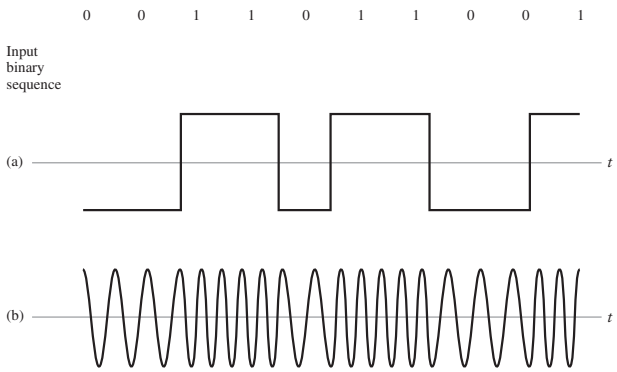


Figure 5.3: (a) Binary sequence and its non-return-to-zero level-encoded waveform. (b) Sunde's BFSK signal.

5.3 M-ary Digital Modulation Schemes

By definition, in an M-ary digital modulation scheme, we send any one of M possible signals $s_1(t)$, $s_2(t)$,... , $s_M(t)$ during each signaling (symbol) interval of duration T. In almost all applications, $M = 2^m$ where m is an integer. Under this condition, the symbol duration $T = mT_b$, where T_b is the bit duration. M-ary modulation schemes are preferred over binary modulation schemes for transmitting digital data over band-pass channels when the requirement is to conserve bandwidth at the expense of both increased power and increased system complexity. In practice, we rarely find a communication channel that has the exact bandwidth required for transmitting the output of an information-bearing source by means of binary modulation schemes. Thus, when the bandwidth of the channel is less than the required value, we resort to an M-ary modulation scheme for maximum bandwidth conservation.

5.3.1 M-ary Phase-Shift Keying

To illustrate the capability of M-ary modulation schemes for bandwidth conservation, consider first the transmission of information consisting of a binary sequence with bit duration T_b. If we were to transmit this information by means of binary PSK, for example, we would require a channel bandwidth that is inversely proportional to the bit duration T_b. However, if we take blocks of m bits to produce a symbol and use an M-ary PSK scheme with $M = 2^m$ and symbol duration $T = mT_b$, then the bandwidth required is proportional to $1/(mT_b)$. This simple argument shows that the use of M-ary PSK provides a reduction in transmission bandwidth by a factor $m = \log_2(M)$ over binary PSK.

In M-ary PSK, the available phase of 2π radians is apportioned equally and in a discrete way among the M transmitted signals, as shown by the phase-modulated signal

$$s_i(t) = \sqrt{\frac{2E}{T}} \cos\left(2\pi f_c t + \frac{2\pi}{M} i\right), \quad i = 0, 1, 2, \dots, M-1, \quad 0 \le t \le T \tag{5.13}$$

We can express $s_i(t)$ in terms of the in-phase and quadrature components

$$s_i(t) = \left[\sqrt{E}\cos\left(\frac{2\pi}{M} i\right)\right]\left[\sqrt{\frac{2E}{T}}\cos(2\pi f_c t)\right] - \left[\sqrt{E}\sin\left(\frac{2\pi}{M} i\right)\right]\left[\sqrt{\frac{2E}{T}}\sin(2\pi f_c t)\right] \tag{5.14}$$

The discrete coefficients $\sqrt{E}\cos\left(\frac{2\pi}{M} i\right)$ and $-\sqrt{E}\sin\left(\frac{2\pi}{M} i\right)$ are respectively the in-phase and quadrature components of the M-ary PSK signal $s_i(t)$. We can easily verify that the envelope of $s_i(t)$ is a constant equal to $\sqrt{E}$ for all M. The modulation strategy of QPSK discussed earlier is an example of M-ary PSK with the number of phase levels $M = 4$.

The previous discussion leads to an insightful geometric portrayal of M-ary PSK. To explain, suppose we construct a two-dimensional diagram with the horizontal and vertical axes respectively defined by the following pair of orthonormal functions

(also called *basis*):

$$\phi_1(t) = \sqrt{\frac{2}{T}}\cos(2\pi f_c t), \quad 0 \le t \le T \tag{5.15}$$

$$\phi_2(t) = \sqrt{\frac{2}{T}}\sin(2\pi f_c t), \quad 0 \le t \le T \tag{5.16}$$

where the band-pass assumption implies orthogonality; the scaling factor $\sqrt{\frac{2}{T}}$ assures unit energy over the interval T for both $\phi_1(t)$ and $\phi_2(t)$. On this basis, we may represent the in-phase and quadrature components for $i = 0, 1, 2, \ldots, M-1$ as a set of points in this two-dimensional diagram, as illustrated in Fig.5.4 for $M = 8$. Such a diagram is referred to as *signal-space diagram* or *signal constellation*.

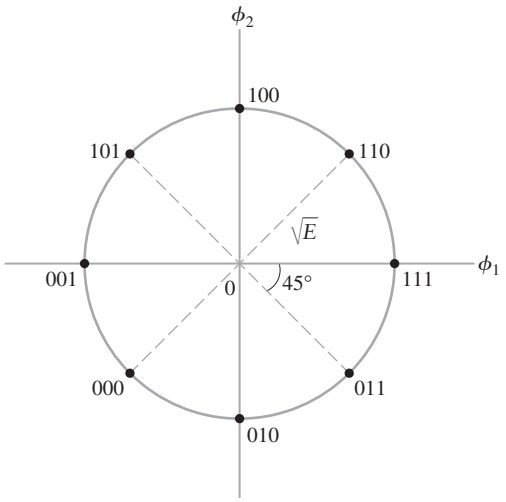


Figure 5.4: Signal-space diagram of 8-PSK.

Figure.5.4 leads us to make three important observations:

1. M-ary PSK is described in geometric terms by a constellation of M signal points distributed uniformly on a circle of radius $\sqrt{E}$.
2. Each signal point in the figure corresponds to the signals $s_i(t)$ for a particular value of the index i.
3. The squared length from the origin to each signal point is equal to the signal energy E.

In light of these observations, we may now formally state that the signal-space-diagram of Figure.5.4 completely sums up the geometric description of M-ary PSK in an insightful manner. Note that the 3-bit sequences corresponding to the 8 signal points are Gray-encoded, with only a single bit changing as we move along the constellation in the figure from one signal point to an adjacent one.

5.3.2 M-ary Quadrature Amplitude Modulation (QAM)

Suppose next that the constraint that the envelope of $s_i(t)$ is a constant for all M is removed. Then, the in-phase and quadrature components of the resulting M-ary modulated signal are permitted to be independent of each other. Specifically, the mathematical description of the new modulated signal assumes the form

$$s_i(t) = \sqrt{\frac{2E_0}{T}}a_i\cos(2\pi f_c t) - \sqrt{\frac{2E_0}{T}}b_i\sin(2\pi f_c t), \quad i = 0, 1, \ldots, M-1, \quad 0 \le t \le T \tag{5.17}$$

where the level parameter a_i in the in-phase component and the level parameter b_i in the quadrature component are independent of each other for all i. This new modulation scheme is called M-ary quadrature amplitude modulation (QAM). Note also that the

constant E_0 is the energy of the signal pertaining to a particular value of the index i for which the amplitude of the modulated signal is the lowest.

M-ary QAM is a hybrid form of M-ary modulation, in the sense that it combines amplitude-shift keying and phase-shift keying. It includes two special cases:

1. If $b_i = 0$ for all i, the modulated signal $s_i(t)$ reduces to

$$s_i(t) = \sqrt{\frac{2E_0}{T}} a_i \cos(2\pi f_c t), \quad i = 0, 1, 2 \ldots, M-1$$

   which defined M-ary *amplitude-shift keying* (M-ary ASK)

2. If $E_0 = E$ and the constraint $(Ea_i^2 + Eb_i^2)^{1/2} = \sqrt{E}$ for all i is satisfied, then the modulated signal $s_i(t)$ reduces to M-ary PSK.

Figure.5.5 portrays the signal-space representation of M-ary QAM for $M = 16$, with each signal point being defined by a pair of level parameters a_i and b_i, where $i = 1, 2, 3, 4$. This time, we see that the signal points are distributed uniformly on a rectangular grid. The rectangular property of the signal-space diagram is testimony to the fact that the in-phase and quadrature components of M-ary QAM are independent of each other. Moreover, we see from Figure.5.5 that, unlike M-ary PSK, the different signal points of M-ary QAM are characterized by different energy levels, and so they should be. Note also that each signal point in the constellation corresponds to a specific *quadbit*, which is made up of 4 bits. Assuming the use of Gray encoding, only one bit is changed as we go from each signal point in the constellation horizontally or vertically to an adjacent point, as illustrated in Figure.5.5.

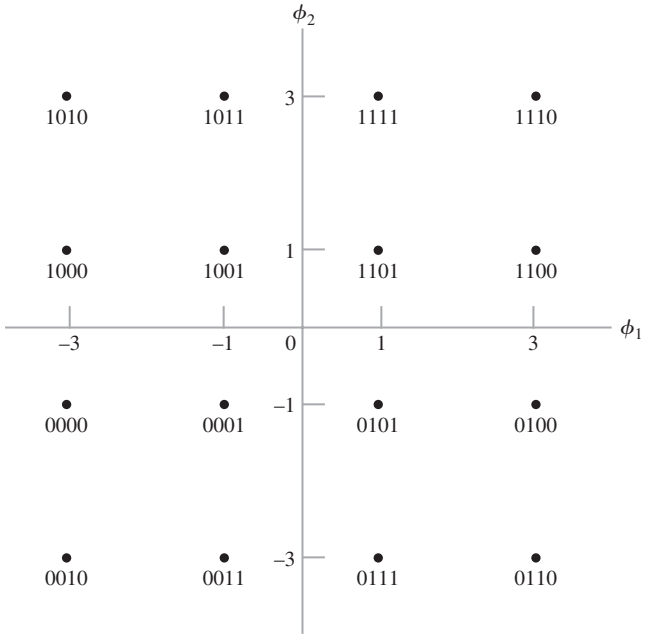


Figure 5.5: Signal-space diagram of Gray-encoded M-ary QAM for $M = 16$.

5.3.3 M-ary Frequency-Shift Keying

When we consider the M-ary version of frequency-shift keying, the picture is quite different from that described for M-ary PSK or M-ary QAM. Specifically, in one form of M-ary FSK, the transmitted signals are defined for some fixed integer n as follows:

$$s_i(t) = \sqrt{\frac{2E}{T}} \cos\left[\frac{\pi}{T}(n+i)t\right], \quad i = 0, 1, \ldots, M-1, \quad 0 \le t \le T \tag{5.18}$$

The M transmitted signals are all of equal duration T and equal energy E. With the individual signal frequencies separated from each other by $1/(2T)$ hertz, the signals $s_i(t)$ are orthogonal. Like M-ary PSK, the envelope of M-ary FSK is constant for all M. Hence, both of these M-ary modulation strategies can be used over non-linear channels. On the other hand, M-ary QAM can only be used over linear channels because its discrete envelope varies with the index i (i.e., the particular signal point chosen for transmission).

To develop a geometric representation of M-ary FSK, we start with Eq.5.18. In terms of the signals $s_i(t)$ defined therein, we introduce a complete set of orthonormal functions:

$$\phi_i(t) = \frac{1}{\sqrt{E}} s_i(t), \quad i = 0, 1, \ldots, M-1, \quad 0 \leq t \leq T \tag{5.19}$$

Unlike M-ary PSK and M-ary QAM, we now find that M-ary FSK is described by an M-dimensional signal-space diagram, where the number of signal points is equal to the number of coordinates. The visualization of such a diagram is difficult beyond $M = 3$. Figure.5.6 illustrates the geometric representation of M-ary FSK for $M = 3$.

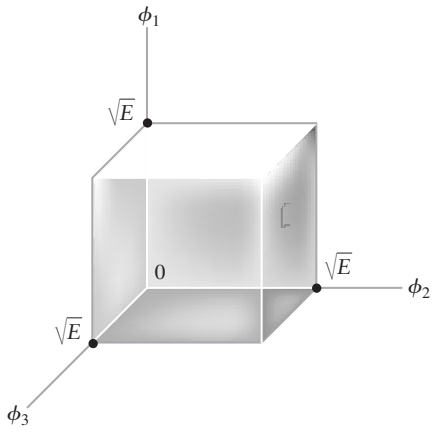


Figure 5.6: Signal constellation for M-ary FSK for $M = 3$.

5.4 Discrete Data Detection

In practice, the channel output waveform $y(t)$ is not equal to the modulated signal $x(t)$. In many cases, the "essential" information of the channel output $y(t)$ is captured by a finite set of vector components, i.e. a vector $\underline{y}$ generated by the demodulation described earlier. Specific important examples appear later in this chapter, but presently the analysis shall presume the existence of the vector $\underline{y}$ and proceed to study the detector for the channel. The detector decides which of the discrete channel input vectors x_i, $i = 1, \ldots, M$ was transmitted based on the observation of the channel output vector $\underline{y}$.

5.4.1 The Vector Channel Model

The vector channel model appears in Figure.5.7. This model suppresses all continuous-time waveforms, and the channel produces a discrete vector output given a discrete vector input. The detector chooses a message m_i from among the set of M

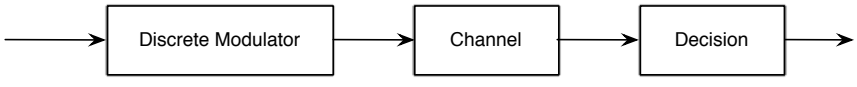


Figure 5.7: Vector Channel Model

possible messages $\{m_i\}$, $i = 1, \ldots, M$ transmitted over the vector channel. The encoder formats the messages for transmission over the vector channel by translating the message m_i into x_i, an $N-$dimensional real data symbol chosen from a signal constellation. The encoders of this text are one-to-one mappings between the message set and the signal-constellation vectors.

The channel-input vector $\underline{x}$ corresponds to a channel-output vector $\underline{y}$, an $N-$dimensional real vector. (Thus, the transformation of $y(t) \rightarrow \underline{y}$ is here assumed to occur within the channel.) The conditional probability of the output vector $\underline{y}$ given the input vector $\underline{x}$, $p_{\underline{y}|\underline{x}}$, completely describes the discrete version of the channel. The decision device then translates the output vector $\underline{y}$ into an estimate of the transmitted message $\hat{\underline{x}}$. A decoder (which is part of the decision device) reverses the process of the encoder and converts the detector output $\hat{\underline{x}}$ into the message decision $\hat{m}$.

The particular message vector corresponding to m_i is x_i, and its n^{th} component is x_{in}. The n^{th} component of $\underline{y}$ is denoted y_n, $n = 1, \ldots, N$. In the vector channel, $\underline{x}$ is a random vector, with discrete probability mass function $p_{\underline{x}}(i)$, $i = 1, \ldots, M$. The output random vector $\underline{y}$ may have a continuous probability density or a discrete probability mass function $p_{\underline{y}}(\underline{v})$, where $\underline{v}$ is a dummy variable spanning all the possible $N-$dimensional outputs for $\underline{y}$. This density is a function of the input and channel transition probability density functions:

$$p_{\underline{y}}(\underline{v}) = \sum_{i=1}^{M} p_{\underline{y}|\underline{x}}(\underline{v}|i).p_{\underline{x}}(i) \tag{5.20}$$

The average energy of the channel input symbols is

$$\mathcal{E}_{\underline{x}} = \sum_{i=1}^{M} ||x_i||^2.p_{\underline{x}}(i) \tag{5.21}$$

The corresponding average energy for the channel-output vector is

$$\mathcal{E}_{\underline{y}} = \sum_{v} ||\underline{v}||^2.p_{\underline{y}}(v) \tag{5.22}$$

An integral replaces the sum in (5.22) for the case of a continuous density function $p_{\underline{y}}(\underline{v})$. As an example, consider the simple additive noise channel $\underline{y} = \underline{x} + \underline{n}$. In this case $p_{\underline{y}|\underline{x}} = p_{\underline{n}}(\underline{y} - \underline{x})$, where $p_{\underline{n}}(\cdot)$ is the noise density, when $\underline{n}$ is independent of the input $\underline{x}$.

5.4.2 The MAP and ML Detectors

Definition 22. *(**Probability of Error**) The probability of error is defined as the probability that the decoded message $\hat{m}$ is not equal to the message that was transmitted*

$$P_e \triangleq Pr\{\hat{m} \neq m\}$$

The optimum data detector chooses $\hat{m}$ to minimize P_e, or equivalently, to maximize $P(c)$; the probability of correct decision.

The MAP Detector

Here we consider the first design problem mentioned earlier in the chapter, that of optimally deciding which of M signals is transmitted from some set of received data represented by received stochastic vector $\underline{Y}$. The case when the received data is instead a stochastic process, $Y(t)$, will be dealt with in the sequel. Let $\underline{X}_i$, $i = 1, 2. \ldots, M$ be the set of M modulation signals. Presumably, the received vector $\underline{Y}$ depend statistically on which of the M signals is transmitted. The receiver design problem is fundamentally one of partitioning the space of all received vector $\underline{Y}$ into M decision regions C_i, $i = 1, 2, \ldots, M$, such that when a received vector $\underline{Y}$ is in C_i, the receiver decides that $\underline{X}_i$ was sent. An **optimal partition** is one that minimizes the average error probability, or, equivalently, maximizes the average probability of a correct decision $P(c)$. Figure.5.8 illustrates the partitioning concept.

We have

$$p(c) = \sum_{i=1}^{M} p(c|\underline{x}_i)p(\underline{x}_i) = \sum_{i=1}^{M} \int_{C_i} f_{\underline{Y}}(\underline{y}|\underline{x}_i)p(\underline{x}_i)d\underline{y}$$

where $\underline{y}$ is a realization of the received random vector $\underline{Y}$ and $f_{\underline{Y}}(\underline{y}|\underline{x}_i)$ is the conditional density of the received random vector $\underline{Y}$ given that $\underline{X}_i$ was sent. Since the integrand in the expression above is non-negative, clearly the average probability of correct decision is maximized when we place in C_i all received vectors $\underline{y}$ for which $f_{\underline{Y}}(\underline{y}|\underline{x}_i)p(\underline{x}_i)$ is largest, for $i = 1, 2, \ldots, M$. Thus, the optimum (minimum error probability) detector implements

$$\hat{x}_{MAP} = arg \max_{i=1,2,\ldots,M} f_{\underline{Y}}(\underline{y}|\underline{x}_i)p(\underline{x}_i) \tag{5.23}$$

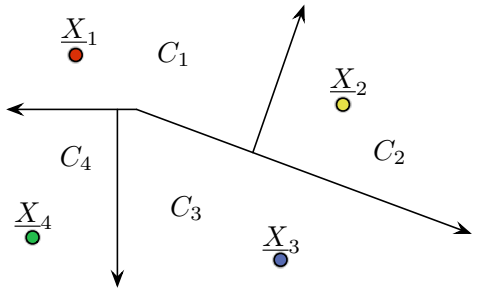


Figure 5.8: Illustration of decision regions for $M = 4$.

Equivalently, it implements

$$\hat{x}_{MAP} = arg \max_{i=1,2,\ldots,M} p(\underline{X}_i|\underline{Y} = \underline{y}) \tag{5.24}$$

in view of Baye's rule

$$p(\underline{X}_i|\underline{Y} = \underline{y}) = \frac{f_{\underline{Y}}(\underline{y}|\underline{x}_i)p(\underline{x}_i)}{f_{\underline{Y}}(\underline{y})} \tag{5.25}$$

where $f_{\underline{Y}}(\underline{y}|\underline{x}_i)$ and $f_{\underline{Y}}(\underline{y})$ are, respectively, the conditional and unconditional density functions of the received data $\underline{Y}$; $p(\underline{x}_i)$ is the *a priori probability* of transmitting signal $\underline{X}_i$ and $p(\underline{X}_i|\underline{Y} = \underline{y})$ is the *a posteriori probability* of transmitting signal $\underline{X}_i$; thus, the name *maximum a posteriori receiver* (MAP).

The ML Detector

If the a priori probabilities are equal, the the MAP receiver coincides with the *maximum likelihood* (ML) receiver that implements

$$\hat{x}_{ML} = arg \max_{i=1,2,\ldots,M} f_{\underline{Y}}(\underline{y}|\underline{x}_i) \tag{5.26}$$

As with the MAP detector, the ML detector also chooses an index i for each possible received vector $\underline{Y}$, but this index now only depends on the channel transition probabilities and is independent of the input distribution. This type of detector only minimizes p_e when the input data symbols have equal probability of occurrence. As this requirement is often met in practice, ML detection is often used. Even when the input distribution is not uniform, ML detection is still often employed as a detection rule, because the input distribution may be unknown and thus assumed to be uniform. The Minimax Theorem sometimes justifies this uniform assumption:

Theorem 11. *(**Minimax Theorem**) The ML detector minimizes the maximum possible average probability of error when the input distribution is unknown, if the conditional probability of error $Pr(error|m_i$ was sent$)$ is independent of i.*

The condition of symmetry imposed by the above theorem is not always satisfied in practical situations; but the likelihood of an application where both the inputs are nonuniform in distribution and the ML conditional error probabilities are not symmetric is rare. Thus, ML receivers have come to be of nearly ubiquitous use in place of MAP receivers.

The function $L_i = f_{\underline{Y}}(\underline{y}|\underline{x}_i)$ is known as the **likelihood function**. In general, instead of maximizing the likelihood function, it is simpler to maximizes its logarithm, $\ell_i = \ln f_{\underline{Y}}(\underline{y}|\underline{x}_i)$, referred to as the *log-likelihood function*.

5.4.3 Decision Regions

In the case of either the MAP or the ML rules, each and every possible value for the channel output $\underline{y}$ maps into one of the M possible transmitted messages. Thus, the vector space for $\underline{y}$ is partitioned into M regions corresponding to the M possible decisions. Simple communication systems have well-defined boundaries (to be shown later), so the decision regions often coincide with intuition. Nevertheless, in some well-designed communications systems, the decoding function and the regions can be more difficult to visualize.

Definition 23. *(**Decision Region**) The decision region using a MAP detector for each message m_i, $i = 1, \ldots, M$ is defined as*

$$\mathcal{D}_i \triangleq \{\underline{v}|p_{\underline{Y}|\underline{X}}(\underline{v}|i) \cdot p_{\underline{X}}(i) \geq p_{\underline{Y}|\underline{X}}(\underline{v}|j) \cdot p_{\underline{X}}(j) \ \ \forall j \neq i\} \tag{5.27}$$

With uniformly distributed input messages, the decision regions reduce to

$$\mathcal{D}_i \triangleq \{\underline{v} | p_{\underline{Y}|\underline{X}}(\underline{v}|i) \geq p_{\underline{Y}|\underline{X}}(\underline{v}|j) \ \ \forall j \neq i\} \tag{5.28}$$

In Figure.5.8, each of the four different two-dimensional transmitted vectors $\underline{X}_i$ (corresponding to the messages m_i) has a surrounding decision region in which any received value for $\underline{Y} = \underline{v}$ is mapped to the message m_i. In general, the regions need not be connected, and although such situations are rare in practice, they can occur.

5.5 Gaussian Random Vectors

In this section, we present basic definitions pertaining to the theory of Gaussian random vectors, which will prove to be handy for the upcoming sections.

Definition 24 (Vector Gaussian PDF)**.** *The joint Gaussian PDF for a vector of n random variables $\underline{X}$, with mean vector $\mu_{\underline{X}}$, and covariance matrix $K_{\underline{X}\underline{X}}$ is given by:*

$$f_{\underline{X}}(\underline{x}) = \frac{1}{\sqrt{(2\pi)^n det(K_{\underline{X}\underline{X}})}} exp\left[-\frac{1}{2}(\underline{x} - \mu_{\underline{X}})^T K_{\underline{X}\underline{X}}^{-1}(\underline{x} - \mu_{\underline{X}})\right] \tag{5.29}$$

Example 40. *For $n = 1$,*

$$f_{\underline{X}}(\underline{x}) = \frac{1}{(2\pi)^{1/2}\sigma} \exp\left[\frac{-1}{2}(\underline{x} - \underline{\mu})^T \frac{1}{\sigma^2}(\underline{x} - \underline{\mu})\right] = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left\{-\frac{1}{2}\left(\frac{x - \mu_x}{\sigma}\right)^2\right\}$$

Example 41. *For $n = 2$, $\underline{X} = (X_1, X_2)^T$ and the covariance matrix $K_{\underline{X}\underline{X}}$ is defined by*

$$K_{\underline{X}\underline{X}} = \begin{bmatrix} \sigma^2_{X_1} & Cov(X_1, X_2) \\ Cov(X_1, X_2) & \sigma^2_{X_2} \end{bmatrix} = \begin{bmatrix} \sigma^2_{X_1} & \rho\sigma_{X_1}\sigma_{X_2} \\ \rho\sigma_{X_1}\sigma_{X_2} & \sigma^2_{X_2} \end{bmatrix}$$

$$\det(K_{\underline{X}\underline{X}}) = \sigma^2_{X_1}\sigma^2_{X_2} - \rho^2\sigma^2_{X_1}\sigma^2_{X_2} = (1 - \rho^2)\sigma^2_{X_1}\sigma^2_{X_2}$$

Hence,

$$f_{X_1X_2}(x_1, x_2) = \frac{1}{(2\pi)\sigma_{X_1}\sigma_{X_2}\sqrt{1 - \rho^2}} \exp\left[\frac{-1}{2(1 - \rho^2)}\beta\right],$$

Where,

$$\beta = \left(\frac{x_1 - \mu_{X_1}}{\sigma_{X_1}}\right)^2 - 2\rho\left(\frac{x_1 - \mu_{X_1}}{\sigma_{X_1}}\right)\left(\frac{x_2 - \mu_{X_2}}{\sigma_{X_2}}\right) + \left(\frac{x_2 - \mu_{X_2}}{\sigma_{X_2}}\right)^2$$

Example 42. *Let X, Y, Z be three zero-mean jointly Gaussian random variables with the following covariance matrix*

$$K = \begin{bmatrix} 1 & 0.2 & 0.3 \\ 0.2 & 1 & 0.3 \\ 0.3 & 0.2 & 1 \end{bmatrix},$$

Find the PDF of $f_{X,Z}(x, z)$.

Solution. From the given information, X and Z are jointly Gaussian and $K_{XZ} = \begin{bmatrix} 1 & 0.3 \\ 0.3 & 1 \end{bmatrix}$. From K_{XZ} we know that:

$$\left.\begin{matrix} \sigma_X = \sigma_Z = 1 \\ Cov[XZ] = 0.3 \end{matrix}\right\} \Rightarrow \rho = \frac{0.3}{1} = 0.3.$$

Therefore,

$$f_{XZ}(x,z) = \frac{1}{(2\pi)\sqrt{0.91}} \exp\left[\frac{-1}{2(0.91)}\left(x^2 - 0.6xz + z^2\right)\right].$$

□

Example 43. *Find the expression for the PDF of the N-dimensional Gaussian vector consisting of mutually uncorrelated vectors.Verify that uncorrelated jointly Gaussian random variables are independent.*

Solution. Two vectors are mutually uncorrelated, hence, $Cov(X_i, X_i) = 0$ for all $i \neq j$. Thus, $K_{\underline{X}\underline{X}}$ is a diagonal matrix

$$K_{\underline{X}\underline{X}} = \begin{bmatrix} \sigma_1^2 & 0 & \dots & 0 \\ 0 & \sigma_2^2 & \dots & 0 \\ 0 & 0 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & \sigma_n^2 \end{bmatrix} \Rightarrow K_{\underline{X}\underline{X}}^{-1} = \begin{bmatrix} \sigma_1^{-2} & 0 & \dots & 0 \\ 0 & \sigma_2^{-2} & \dots & 0 \\ 0 & 0 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & \sigma_n^{-2} \end{bmatrix}$$

$$det(K_{\underline{X}\underline{X}}) = \prod_{i=1}^{n} \sigma_i^2$$

$$(\underline{x} - \mu_{\underline{X}})^T K_{\underline{X}\underline{X}}^{-1} (\underline{x} - \mu_{\underline{X}}) = \sum_{i=1}^{n} \left(\frac{x_i - \mu_i}{\sigma_i}\right)^2$$

Hence,

$$f_{\underline{X}}(\underline{x}) = \frac{1}{\sqrt{(2\pi)^n \prod_{i=1}^n \sigma_i^2}} exp\left[\sum_{i=1}^{n} \frac{(x_i - \mu_i)^2}{2\sigma_i^2}\right] = \prod_{i=1}^{n} \frac{1}{\sqrt{2\pi\sigma_i^2}} exp\left(-\frac{(x_i - \mu_i)^2}{2\sigma_i^2}\right) = \prod_{i=1}^{n} f_{X_i}(x_i)$$

So uncorrelated Gaussian random variables are independent. □

5.6 The Vector AWGN Channel

Additive white Gaussian noise (AWGN) is a basic noise model used in communication theory to mimic the effect of many random processes that occur in nature. In this section we consider the detection problem for a vector AWGN channel, where the problem can be cast as an m-ary hypotheses testing problem based on $n-$tuple observations.

We will assume that the noise vector $\underline{W}$ is an $n-$dimensional Gaussian random vector with zero mean, equal-variance σ^2, and with uncorrelated components in each dimension. The noise distribution is

$$f_{\underline{W}}(\underline{w}) = (\pi N_0)^{-\frac{n}{2}} \cdot e^{-\frac{1}{N_0}||\underline{w}||^2} = (2\pi\sigma^2)^{-\frac{n}{2}} \cdot e^{-\frac{1}{2\sigma^2}||\underline{w}||^2}$$

The vector channel is described as follows

$$\underline{Y} = \underline{X}_i + \underline{W}, \quad i = 1, 2 \dots, m \tag{5.30}$$

where $\underline{X}_i$ is an is an $n-$dimensional signal vector derived from an $m-$ary set (a set of m messages), independent from the noise

vector, and $\underline{Y}$ is an $n-$dimensional vector representing the received (observed) signal. Alternatively, we can write Eq. (5.30) as

$$\begin{aligned}
H_1 \quad &: \quad \underline{Y} = \underline{X}_1 + \underline{W} \ \sim \mathcal{N}(x_1, \sigma^2 \mathcal{I}_n) \Rightarrow f_{\underline{Y}|H}(\underline{y}|1) = \frac{1}{(2\pi\sigma^2)^{n/2}} exp\left\{-\frac{||\underline{y} - \underline{x}_1||^2}{2\sigma^2}\right\} \\
H_2 \quad &: \quad \underline{Y} = \underline{X}_2 + \underline{W} \ \sim \mathcal{N}(x_2, \sigma^2 \mathcal{I}_n) \Rightarrow f_{\underline{Y}|H}(\underline{y}|2) = \frac{1}{(2\pi\sigma^2)^{n/2}} exp\left\{-\frac{||\underline{y} - \underline{x}_2||^2}{2\sigma^2}\right\} \\
&\vdots \\
H_m \quad &: \quad \underline{Y} = \underline{X}_m + \underline{W} \ \sim \mathcal{N}(x_m, \sigma^2 \mathcal{I}_n) \Rightarrow f_{\underline{Y}|H}(\underline{y}|m) = \frac{1}{(2\pi\sigma^2)^{n/2}} exp\left\{-\frac{||\underline{y} - \underline{x}_m||^2}{2\sigma^2}\right\}
\end{aligned}$$

where $\mathcal{I}_n$ is the $n-$dimensional identity matrix.

To apply the MAP detection rule, we need to compute

$$\hat{H}_{MAP} = arg \max_{i=1,2,\ldots,m} L_i$$

where $L_i = f_{\underline{Y}|H}(\underline{y}|\underline{x}_i)p(H_i)$. Taking the logarithm of L_i, we get the log-likelihood function

$$\ell_i = \ln f_{\underline{Y}|H}(\underline{y}|\underline{x}_i) + \ln p(H_i)$$

Substituting with $f_{\underline{Y}|H}(\underline{y}|\underline{x}_i)$, we get

$$\begin{aligned}
\ell_i &= \ln\left[\frac{1}{(2\pi\sigma^2)^{n/2}} exp\left\{-\frac{||\underline{y} - \underline{x}_i||^2}{2\sigma^2}\right\}\right] + \ln p(H_i) \\
&= -\frac{n}{2}\ln(2\pi\sigma^2) - \frac{1}{2\sigma^2}||\underline{y} - \underline{x}_i||^2 + \ln p(H_i) \\
&= -\sum_{j=1}^{n}\left[\frac{1}{2}\ln(2\pi\sigma^2 + \frac{1}{2\sigma^2}(y_j - x_{ij})^2\right] + \ln p(H_i) \qquad (5.31)
\end{aligned}$$

Dropping terms which are irrelevant in maximizing ℓ_i from Eq. (5.31), we get

$$\ell_i = -\sum_{j=1}^{n}(y_j - x_{ij})^2 + \ln p(H_i) \qquad (5.32)$$

Expanding the square, dropping the quadratic term that does not affect the maximization and dividing the result by 2, Eq. (5.32) becomes

$$\ell_i = \sum_{j=1}^{N}\left(y_j x_{ij} - \frac{1}{2}x_{ij}^2\right) + \ln p(H_i) = \langle \underline{y}, \underline{x}_i \rangle - \frac{1}{2}||\underline{x}_i||^2 + \ln p(H_i)$$

Hence, the MAP detector for the considered vector AWGN channel is

$$\hat{H}_{MAP} = arg \max_{i=1,2,\ldots,m}\left\{\langle \underline{y}, \underline{x}_i \rangle - \frac{1}{2}||\underline{x}_i||^2 + \ln p(H_i)\right\} \qquad (5.33)$$

From the MAP decision rule, we can easily derive the ML decision rule for the vector AWGN channel, where we assume that all hypotheses are equiprobable, i.e., $p(H_i) = \frac{1}{m}$, $\forall i = 1, 2 \ldots, m$. This yields the following ML rule

$$\hat{H}_{ML} = arg \max_{i=1,2,\ldots,m}\left\{\langle \underline{y}, \underline{x}_i \rangle - \frac{1}{2}||\underline{x}_i||^2\right\} \qquad (5.34)$$

If we assume that all messages have the same energy, then the ML rule is further simplified to the following

$$\hat{H}_{ML} = arg \max_{i=1,2,\ldots,m}\langle \underline{y}, \underline{x}_i \rangle \qquad (5.35)$$

The ML detector for the AWGN channel has the intuitively appealing physical interpretation that the decision $\hat{\underline{X}} = \underline{X}_i$ corresponds to choosing the data symbol $\underline{X}_i$ that is closest, in terms of the Euclidean distance, to the received vector channel output $\underline{Y}$. To see this, consider the equiprobable transmission of two data symbols $\underline{X}_1$ and $\underline{X}_2$, then the ML decision rule is

$$-||\underline{y} - \underline{x}_1||^2 \underset{\hat{H}_2}{\overset{\hat{H}_1}{\gtrless}} -||\underline{y} - \underline{x}_2||^2 \Rightarrow ||\underline{y} - \underline{x}_1||^2 \underset{\hat{H}_1}{\overset{\hat{H}_2}{\gtrless}} ||\underline{y} - \underline{x}_2||^2 \Rightarrow d(\underline{y}, \underline{x}_1) \underset{\hat{H}_1}{\overset{\hat{H}_2}{\gtrless}} d(\underline{y}, \underline{x}_2)$$

This rule implies that the detector decides in favor of $\underline{X}_1$ if the received vector $\underline{y}$ is closest to $\underline{X}_1$ and in favor of $\underline{X}_2$ if the received vector $\underline{y}$ is closest to $\underline{X}_2$.

Without noise, the received vector is $\underline{Y} = \underline{X}_i$ the transmitted symbol, but the additive Gaussian noise results in a received symbol most likely in the neighborhood of $\underline{X}_i$. The Gaussian shape of the noise implies the probability of a received point decreases as the distance from the transmitted point increases.

Example 44. *(Optimal decision rule)*
Consider the reception of either of the following signals in additive zero-mean white and Gaussian noise process with spectral height $N_0/2$.

$$\begin{aligned} s_1(t) &= \sqrt{\frac{2E}{T}}\cos(2\pi f_c t), \quad 0 \le t \le T \\ s_2(t) &= 2\sqrt{\frac{2E}{T}}\cos(2\pi f_c t), \quad 0 \le t \le T \\ s_3(t) &= \sqrt{\frac{2E}{T}}\sin(2\pi f_c t), \quad 0 \le t \le T \end{aligned}$$

where $f_c = \frac{n}{T}$, with n being a positive integer.

1. *Determine the optimal decision rule that needs to be implemented by the vector receiver. Let x_1 and x_2 be the outputs of the correlators used in the detection of the signals.*
2. *Based on the developed decision rule, what would the decision be if the received signal is $\sqrt{E}[\cos(2\pi f_c t) - \sin(2\pi f_c t)]$?*

Solution.

1. By inspection, $\varphi_1(t) = \sqrt{\frac{2}{T}}\cos(2\pi f_c t), \quad 0 \le t \le T$ and $\varphi_2(t) = \sqrt{\frac{2}{T}}\sin(2\pi f_c t), \quad 0 \le t \le T$. From the previous example, the vector receiver implements the following:

$$\arg \max_{i=1,2,3} \left\{ \underline{x} \cdot \underline{s}_i - \frac{1}{2}E_i \right\}$$

Given the chosen basis functions, $\underline{s}_1 = [\sqrt{E} \quad 0]^T$, $\underline{s}_2 = [2\sqrt{E} \quad 0]^T$, $\underline{s}_3 = [0 \quad \sqrt{E}]^T$. Hence,

$$\begin{aligned} \underline{x} \cdot \underline{s}_1 - \frac{1}{2}E_1 &= \sqrt{E}x_1 - \frac{E}{2} \\ \underline{x} \cdot \underline{s}_2 - \frac{1}{2}E_2 &= 2\sqrt{E}x_1 - 2E \\ \underline{x} \cdot \underline{s}_3 - \frac{1}{2}E_3 &= \sqrt{E}x_2 - \frac{E}{2} \end{aligned}$$

The decision rule is then,

$$\max \left\{ x_1 - \frac{\sqrt{E}}{2}, 2x_1 - 2\sqrt{E}, x_2 - \frac{\sqrt{E}}{2} \right\}$$

2. The received signal can be written as $\sqrt{\frac{TE}{2}}\varphi_1(t) - \sqrt{\frac{TE}{2}}\varphi_2(t)$. Hence, the received vector is

$$[x_1 \quad x_2]^T = \left[\sqrt{\frac{TE}{2}} \quad -\sqrt{\frac{TE}{2}}\right]^T$$

We compute

$$\begin{aligned} x_1 - \frac{\sqrt{E}}{2} &= \sqrt{\frac{TE}{2}} - \frac{\sqrt{E}}{2} \\ 2x_1 - 2\sqrt{E} &= \sqrt{TE} - 2\sqrt{E} \\ x_2 - \frac{\sqrt{E}}{2} &= -\sqrt{\frac{TE}{2}} - \frac{\sqrt{E}}{2} \end{aligned}$$

The decision reduces to

$$\sqrt{\frac{TE}{2}} - \frac{\sqrt{E}}{2} \underset{\text{Decide } s_2(t)}{\overset{\text{Decide } s_1(t)}{\gtrless}} \sqrt{TE} - 2\sqrt{E} \Rightarrow \frac{1-\sqrt{2}}{2}\sqrt{T} + \frac{3}{4} \underset{\text{Decide } s_2(t)}{\overset{\text{Decide } s_1(t)}{\gtrless}} 0 \Rightarrow \sqrt{T} \underset{\text{Decide } s_1(t)}{\overset{\text{Decide } s_2(t)}{\gtrless}} \frac{3}{2(\sqrt{2}-1)}$$

Hence, if $T > 13.11$, decide that the signal is $s_2(t)$, otherwise, decide that the signal is $s_1(t)$.

□

5.6.1 Interpretation of the Optimum Detector for the AWGN Channel

The log-likelihood function in the waveform domain is

$$\ell_i = \int_0^T Y(t)X_i(t)dt - \frac{1}{2}\int_0^T X_i^2(t)dt, \quad i = 1, 2, \ldots, M \tag{5.36}$$

Another equivalent form of the log-likelihood function that gives further insight into the detection problem is obtained by subtracting half the energy of $Y(t)$ on the right hand side (which does not affect the detection problem), completing the squares and then multiplying the result by -2, converts the maximization into a minimization. Thus, we can equivalently minimize

$$\ell_i = \int_0^T [Y(t) - X_i(t)]^2 dt \tag{5.37}$$

The likelihood function corresponding to the log-likelihood function in (5.36) (obtained by taking the logarithms and dropping some constant terms) is

$$L_i = exp\left[\frac{2}{N_0}\int_0^T Y(t)X_i(t)dt - \frac{1}{N_0}\int_0^T X_i^2(t)dt\right] \tag{5.38}$$

We will make use of the log-likelihood function above to derive optimal detectors for channels that are perturbed by additive Gaussian noise. If the modulation signals have equal energy, then

$$\ell_i = \int_0^T Y(t)X_i(t)dt, \quad i = 1, 2, \ldots, M$$

For obvious reasons, the resulting receiver is referred to as the **correlation receiver**. In other words, when the modulation signals have equal energy, the most likely transmitted signal is one that is maximally correlated with the received signal. A block diagram of the correlation receiver is shown in Figure 5.9. It consists of a bank of correlators followed by samplers that periodically sample the correlator output every T seconds to obtain the M log-likelihood statistics that are then used to make a decision by finding the largest. Note that it is crucial that the sampling of the correlator outputs be done synchronously both

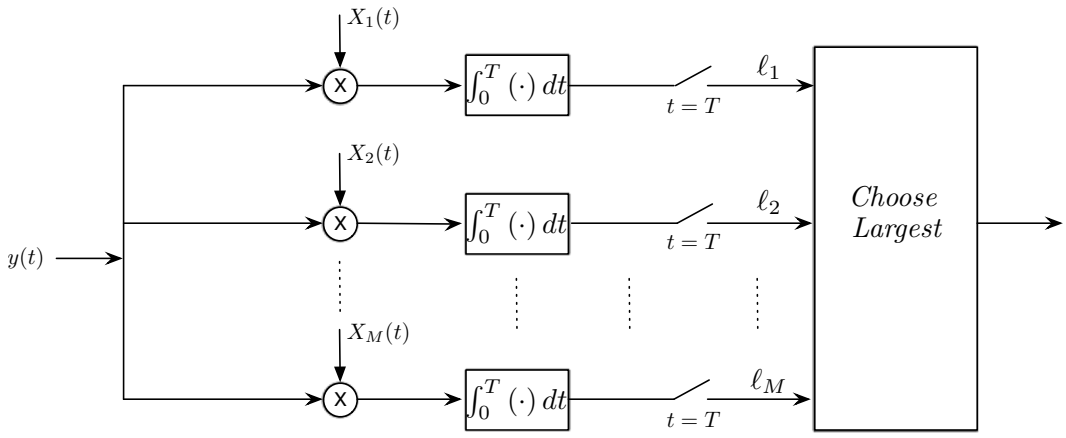


Figure 5.9: Binary ML detector

in frequency (i.e. every T seconds) and at the correct timing phase. In practice this timing information must be extracted at the receiver; it is the function of the timing-synchronization subsystem to provide timing information to the detector. Clearly, any error in timing will result in performance loss. In this chapter we will assume that time-synchronization already exists and is in fact perfect.

5.6.2 The Matched-Filter Receiver

Since the log-likelihood statistics $\ell_1, \ell_2, \ldots \ell_M$ are obtained from the received data $y(t)$ through a linear operation (correlation), it should be possible to also compute them by passing the data $y(t)$ through a linear filter of some appropriate impulse response and then sampling the output of this filter instead. To find the impulse-response of the filter, we have

$$\int_{-\infty}^{\infty} y(t) \cdot x_i(t) dt = \int_{-\infty}^{\infty} y(t) \cdot h_i(T - t) dt \Rightarrow h_i(t) = x_i(T - t) \tag{5.39}$$

Thus, the $i-$th impulse-response is a time-reversed and translated version of the $i-$th signal. We say that the impulse-response is matched to the $i-$th signal and refer to the corresponding receiver as the matched-filter receiver. The matched-filter receiver is not a new receiver but simply a different implementation of the optimum correlation receiver that offers a number of implementation advantages: it does not require a multiplier or signal generator, but only a linear filter, the design of which is a well studied problem. Figure 5.10 shows the matched-filter receiver. Note that the timing for the sampling operation is provided again by a timing-synchronizer. The matched filters shown satisfy the SNR maximization property.

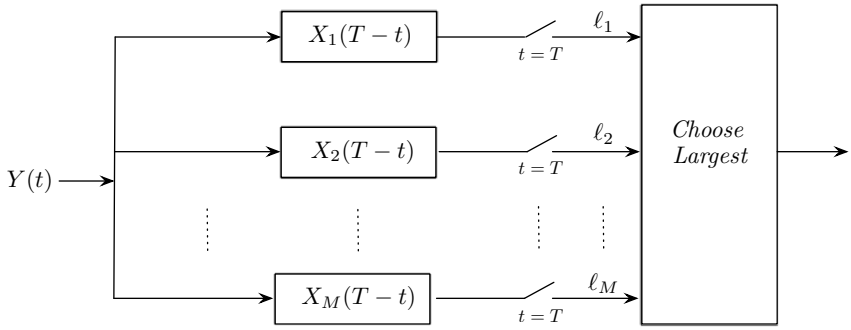


Figure 5.10: Matched Filter Receiver

Example 45. *(Error Performance for 6-ary PAM)*
Consider the transmission of 6 signals using a Pulse Amplitude Modulation (PAM) scheme (a 6-ary source) over an AWGN channel, where the noise is of zero-mean and variance σ^2. The situation could be cast as a mapping from $\mathcal{H} = \{0, 1, 2, 3, 4, 5\}$ to $\mathcal{S} = \{s_0, s_1, s_2, s_3, s_4, s_5\}$, where s_i are real valued for $i = 0, 1, \dots, 5$. Let $s_{i+1} - s_i = d$ for any $i = 0, 1, \dots, 4$.

1. *Draw the signal constellation and specify the decision regions corresponding to the ML decoder at high SNR.*
2. *Evaluate the probability of error as a function of d.*

Solution.

1. The signals are elements of $\mathbb{R}$, and the ML decoder chooses according to the minimum-distance rule. Let $\mathcal{R}_i$, $i = 0, 1 \dots, 5$, denote the decision regions corresponding to the received signals $y = s_i + w$, where w is the AWGN noise of zero-mean and variance σ^2. At high SNR, the a certain hypothesis could be decoded wrong by one of its adjacent neighbors. Using the minimum-distance decision rule, the thresholds between adjacent decision regions are the perpendicular bisectors that separate those regions; represented by dashed lines in Fig. 5.11.

2. When the hypothesis is $H = 0$, the receiver makes the wrong decision if the observation $u \in \mathbb{R}$ falls outside the decoding region $\mathcal{R}_0$. This is the case if the noise $w \in \mathbb{R}$ is larger than $d/2$. Thus,

$$P_e(0) = Pr\{w > d/2\} = Q\left(\frac{d}{2}\right)$$

By symmetry, $P_e(5) = P_e(0)$. For $i \in \{1, 2, 3, 4\}$, the probability of error when $H = i$ is the probability that the event $\{w \geq d/2\} \cup \{w < -d/2\}$ occurs. This event is the union of disjoint events. Its probability is the sum of the probabilities of the individual events. Hence,

$$P_e(i) = Pr\{\{w \geq d/2\} \cup \{w < -d/2\}\} = 2Pr\{w \geq d/2\} = 2Q\left(\frac{d}{2\sigma}\right), \quad i \in \{1, 2, 3, 4\}$$

Finally,

$$P_e = \frac{2}{6}Q\left(\frac{d}{2\sigma}\right) + \frac{4}{6}Q\left(\frac{d}{2\sigma}\right) = \frac{5}{3}Q\left(\frac{d}{2\sigma}\right)$$

□

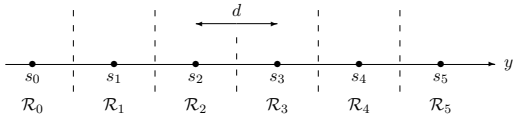


Figure 5.11: 6-ary PAM Constellation

Example 46. *(Error Performance for 4-QAM) Consider the transmission of the following 4 signals using a Quadrature Amplitude Modulation (QAM) scheme (a 4-ary source) over an AWGN channel*

$$s_i(t) = \sqrt{\frac{2E}{T}} \cos\left(2\pi f_c t + \frac{\pi}{4}(2i + 1)\right), \quad 0 \leq t \leq T, \ i = 0, 1, 2, 3$$

The signals are mapped into a 2-dimensional signal space spanned by the following orthonormal basis

$$\varphi_1(t) = \sqrt{\frac{2}{T}}\cos(2\pi f_c t),\ 0 \le t \le T$$

$$\varphi_2(t) = -\sqrt{\frac{2}{T}}\sin(2\pi f_c t),\ 0 \le t \le T$$

The received signal $y(t)$, projected on the signal space is represented by the sector $\underline{y} = [y_1 \quad y_2]^T$ give by

$$\underline{y} = \underline{s}_i + \underline{w}, \quad i = 0, 1, 2, 3$$

where $\underline{w} = [w_1 \quad w_2]^T$ is a Gaussian random vector of independent components, each of zero-mean and variance σ^2

1. *Express $\{s_i(t)\}_{i=1}^4$ in terms of $\varphi_1(t)$ and $\varphi_2(t)$.*
2. *Draw the signal constellation in the specified signal space. Draw and specify the decision regions corresponding to the ML decoder at high SNR.*
3. *For the signal constellation, compute the average energy per symbol as a function of E.*
4. *Assume the signals are encoded using a Gray code. Specify T such that the given scheme can convey a bit rate of 2 Mbps.*
5. *Compute the probability of error in terms of d, where $d = \sqrt{2E}$.*

Solution.

1.

$$\begin{aligned}
s_0(t) &= \sqrt{\frac{2E}{T}}\cos\left(2\pi f_c t + \frac{\pi}{4}\right) = \sqrt{\frac{E}{T}}\cos(2\pi f_c t) - \sqrt{\frac{E}{T}}\sin(2\pi f_c t) = \sqrt{\frac{E}{2}}\varphi_1(t) + \sqrt{\frac{E}{2}}\varphi_2(t) \\
s_1(t) &= \sqrt{\frac{2E}{T}}\cos\left(2\pi f_c t + \frac{3\pi}{4}\right) = -\sqrt{\frac{E}{T}}\cos(2\pi f_c t) - \sqrt{\frac{E}{T}}\sin(2\pi f_c t) = -\sqrt{\frac{E}{2}}\varphi_1(t) + \sqrt{\frac{E}{2}}\varphi_2(t) \\
s_2(t) &= \sqrt{\frac{2E}{T}}\cos\left(2\pi f_c t + \frac{5\pi}{4}\right) = -\sqrt{\frac{E}{T}}\cos(2\pi f_c t) + \sqrt{\frac{E}{T}}\sin(2\pi f_c t) = -\sqrt{\frac{E}{2}}\varphi_1(t) - \sqrt{\frac{E}{2}}\varphi_2(t) \\
s_3(t) &= \sqrt{\frac{2E}{T}}\cos\left(2\pi f_c t + \frac{7\pi}{4}\right) = \sqrt{\frac{E}{T}}\cos(2\pi f_c t) + \sqrt{\frac{E}{T}}\sin(2\pi f_c t) = \sqrt{\frac{E}{2}}\varphi_1(t) - \sqrt{\frac{E}{2}}\varphi_2(t)
\end{aligned}$$

2. The signals $\{s_i(t)\}$, $i = 0, 1, 2, 3$ are represented in the signal space as the following vectors

$$\begin{aligned}
\underline{s}_0 &= \left[\sqrt{\frac{E}{2}} \quad \sqrt{\frac{E}{2}}\right]^T \\
\underline{s}_1 &= \left[-\sqrt{\frac{E}{2}} \quad \sqrt{\frac{E}{2}}\right]^T \\
\underline{s}_2 &= \left[-\sqrt{\frac{E}{2}} \quad -\sqrt{\frac{E}{2}}\right]^T \\
\underline{s}_3 &= \left[\sqrt{\frac{E}{2}} \quad -\sqrt{\frac{E}{2}}\right]^T
\end{aligned}$$

The resulting signal constellation is shown in Fig. 5.12. Since the ML detector is the minimum distance decoder, then at high SNR the decision regions corresponds to the 4 quadrants in the signal space. See Fig. 5.13.

3. The energy for each of the signals is:

$$\mathcal{E} = \left(\sqrt{\frac{E}{2}}\right)^2 + \left(\sqrt{\frac{E}{2}}\right)^2 = E$$

The average energy per symbol of the constellation is

$$\mathcal{E}_{avg} = \frac{1}{4}E + \frac{1}{4}E + \frac{1}{4}E + \frac{1}{4}E = E$$

4. The bit rate is given by

$$R_b = \frac{\log_2 4}{T} = \frac{2}{T} = 2 \times 10^6$$

Hence, $T = 10^{-6}s = 1\ \mu s$.

5. The decoding region for $\underline{s}_0$ is the first quadrant represented by $\mathcal{R}_0$, for $\underline{s}_1$ is the second quadrant represented by $\mathcal{R}_1$, etc. When $H = 0$, the decoder makes the correct decision if $\{w_1 > -\frac{d}{2}\} \cap \{w_2 \geq -\frac{d}{2}\}$ (this is the intersection of independent events). Hence,

$$P_c(0) = Pr\left\{w_1 > -\frac{d}{2}\right\} Pr\left\{w_2 \geq -\frac{d}{2}\right\} = Q^2\left(-\frac{d}{2\sigma}\right) = \left[1 - Q\left(\frac{d}{2\sigma}\right)\right]^2$$

By symmetry, for all i, $P_c(i) = P_c(0)$. Hence,

$$P_e = P_e(0) = 1 - P_c(0) = 2Q\left(\frac{d}{2\sigma}\right) - Q^2\left(\frac{d}{2\sigma}\right)$$

□

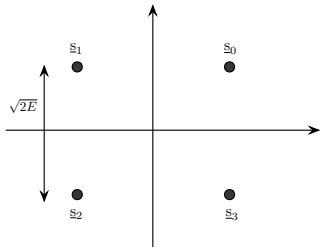


Figure 5.12: 4-ary QAM Constellation

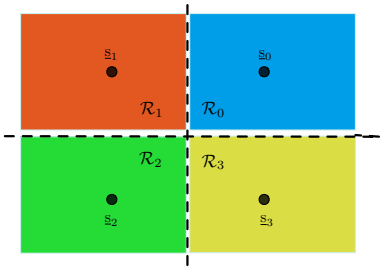


Figure 5.13: Decision region for ML detector.

CHAPTER 6

COMMUNICATION THROUGH BANDLIMITED AWGN CHANNELS

In the preceding chapter, we considered digital communication over an AWGN channel and evaluated the probability of error performance of the optimum receiver for baseband signals. However, we have assumed that the channel introduced no distortion. In this chapter, we treat digital communication over a channel that is modeled as a linear filter with a bandwidth limitation. Bandlimited channels most frequently encountered in practice are telephone channels, microwave LOS radio channels, satellite channels, and underwater acoustic channels.

In general, a linear filter channel imposes more stringent requirements on the design of modulation signals. Specifically, the transmitted signals must be designed to satisfy the bandwidth constraint imposed by the channel. The bandwidth constraint generally precludes the use of rectangular pulses at the output of the modulator. Instead, the transmitted signals must be shaped to restrict their bandwidth to that available on the channel. The design of bandlimited signals is one of the topics treated in this chapter.

We will see that a linear filter channel distorts the transmitted signal. The channel distortion results in **intersymbol interference** at the output of the demodulator and leads to an increase in the probability of error at the detector. Devices or methods for correcting or undoing the channel distortion, called **channel equalizers**, are then described.

6.1 Digital Transmission Through Bandlimited Channels.

A bandlimited channel such as a telephone wireline is characterized as a linear filter with impulse response $c(t)$ and frequency response $C(f)$, where

$$C(f) = \int_{-\infty}^{\infty} c(t) e^{-j2\pi ft} dt \tag{6.1}$$

If the channel is a baseband channel that is bandlimited to B_c Hz, then $C(f) = 0$ for $|f| > B_c$. Any frequency components at the input to the channel that are higher than B_c Hz will not be passed by the channel. For this reason, we consider the design of signals for transmission through the channel that are bandlimited to $W = B_c$ Hz, as shown in Fig. 6.1. Henceforth, W will denote the bandwidth limitation of the signal and the channel. Now, suppose that the input to a bandlimited channel is a signal waveform $g_T(t)$. Then, the response of the channel is the convolution of $g_T(t)$ with $c(t)$; i.e.,

$$h(t) = \int_{-\infty}^{\infty} c(\tau) g_T(t-\tau) d\tau = c(t) \star g_T(t) \tag{6.2}$$

or, when expressed in the frequency domain, we have

$$H(f) = C(f) G_T(f) \tag{6.3}$$

where $G_T(f)$ is the spectrum of the signal $g_T(t)$ and $H(f)$ is the spectrum of $h(t)$. Thus, the channel alters or distorts the transmitted signal $g_T(t)$.

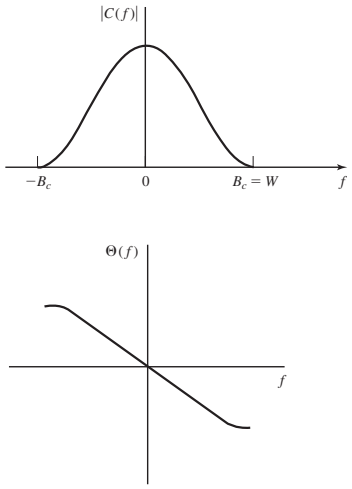


Figure 6.1: Magnitude and phase responses of bandlimited channel.

Assume now that the signal at the output of the channel is corrupted by AWGN. Then, the signal at the input to the demodulator is of the form $h(t) + n(t)$, where $n(t)$ denotes the AWGN. Recall from the preceding chapter that in the presence of AWGN, a demodulator that employs a filter which is matched to the signal h(t) maximizes the SNR at its output. Therefore, let us pass the received signal $h(t) + n(t)$ through a filter that has a frequency response

$$G_R(f) = H^*(f)e^{-j2\pi ft_0} \tag{6.4}$$

where t_0 is some nominal time delay at which we sample the filter output. The signal component at the output of the matched filter at the sampling instant $t = t_0$ is

$$y_s(t_0) = \int_{-\infty}^{\infty} |H(f)|^2 df = E_h \tag{6.5}$$

which is the energy in the channel output $h(t)$. The noise component at the output of the matched filter has a zero mean and a power-spectral density

$$S_n(f) = \frac{N_0}{2}|H(f)|^2 \tag{6.6}$$

Hence, the noise power at the output of the matched filter has a variance

$$\sigma_n^2 = \int_{-\infty}^{\infty} S_n(f)df = \frac{N_0}{2}\int_{-\infty}^{\infty} |H(f)|^2 df = \frac{N_0 E_h}{2} \tag{6.7}$$

The SNR at the output of the matched filter is

$$SNR_0 = \left(\frac{S}{N}\right)_0 = \frac{E_h^2}{N_0 E_h/2} = \frac{2E_h}{N_0} \tag{6.8}$$

This is the result for the SNR at the output of the matched filter that was obtained in the previous chapter except that the received signal energy E_h has replaced the transmitted signal energy E_s. Compared to the previous result, the major difference in this development is that the filter impulse response is matched to the received signal $h(t)$ instead of the transmitted signal. Note that the implementation of the matched filter at the receiver requires that $h(t)$ or, equivalently, the channel impulse response $c(t)$ must be known to the receiver.

Example 47. *The signal pulse $g_T(t) = \frac{1}{2}\left[1 + \cos\frac{2\pi}{T}\left(t - \frac{T}{2}\right)\right]$, $0 \le t \le T$ is transmitted through a baseband channel with frequency-response characteristic as shown in Fig. 6.2(a). The signal pulse is illustrated in Fig. 6.2(b). The channel output is corrupted by AWGN with power-spectral density $N_0/2$. Determine the matched filter to the received signal and the output SNR.*

Solution. This problem is most easily solved in the frequency domain. First, the spectrum of the signal pulse is

$$\begin{aligned} G_T(f) &= \frac{T}{2}\frac{\sin(\pi fT)}{\pi fT(1-f^2T^2}e^{-j\pi fT} \\ &= \frac{T}{2}\frac{sinc(\pi fT)}{(1-f^2T^2}e^{-j\pi fT} \end{aligned}$$

The spectrum of $|G_T(f)|^2$ is shown below.
Hence,

$$H(f) = C(f) = G_T(f) = \begin{cases} G_T(f), & |f| < W \\ 0 & otherwise \end{cases}$$

Then, the signal component at the output of the filter matched to $H(f)$ is

$$\begin{aligned} E_h &= \int_{-W}^{W} |G_T(f)|^2 df \\ &= \frac{1}{(2\pi)^2}\int_{-W}^{W} \frac{(\sin(\pi fT))^2}{f^2(1-f^2T^2)^2} df \\ &= \frac{T}{(2\pi)^2}\int_{-WT}^{WT} \frac{\sin^2(\pi\alpha)}{\alpha^2(1-\alpha^2)^2} d\alpha \end{aligned}$$

The variance of the noise component is

$$\sigma_n^2 = \frac{N_0}{2}\int_{-W}^{W} |G_T(f)|^2 df = \frac{N_0 E_h}{2}$$

Hence, the output SNR is

$$(SNR)_0 = \frac{2E_h}{N_0}$$

In this example, we observe that the signal at the input to the channel is not bandlimited. Hence, only a part of the transmitted signal energy is received. The amount of signal energy at the output of the matched filter depends on the value of the channel bandwidth W when the signal pulse duration is fixed. The maximum value of E_h, obtained as $W \to \infty$, is

$$\max E_h = \int_{-\infty}^{\infty} |G_T(t)|^2 df = \int_0^T g_T^2(t) dt$$

□

In the above development, we considered the transmission and reception of only a single signal waveform $g_T(t)$ through a bandlimited channel with impulse response $c(t)$. We observed that the performance of the system is determined by E_h, the energy in the received signal $h(t)$. To maximize the received SNR, we have to make sure that the power-spectral density of the transmitted signal matches the frequency band of the channel. To this end we must study the power-spectral density of the input signal. The impact of the channel bandwidth limitation is felt when we consider the transmission of a sequence of signal waveforms. This problem is treated in the following section.

6.2 Digital PAM Transmission Through Bandlimited Baseband Channels.

Let us consider the baseband PAM communication system illustrated by the functional block diagram in Fig. 6.3.

The system consists of a transmitting filter having an impulse response $g_T(t)$, the linear filter channel with AWGN, a receiving filter with impulse response $g_R(t)$, a sampler that periodically samples the output of the receiving filter, and a symbol detector. The sampler requires the extraction of a timing signal from the received signal (for synchronization reasons). This timing signal serves as a clock that specifies the appropriate time instants for sampling the output of the receiving filter.

First we consider digital communications by means of M-ary PAM. Hence, the input binary data sequence is subdivided into k-bit symbols and each symbol is mapped into a corresponding amplitude level that amplitude modulates the output of the

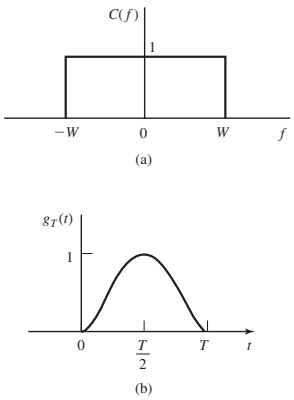
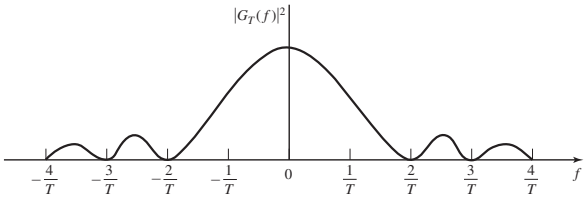


Figure 6.2: The signal pulse in (b) is transmitted through the ideal bandlimited channel shown in (a).



transmitting filter. The baseband signal at the output of the transmitting filter (the input to the channel) may be expressed as

$$v(t) = \sum_{n=-\infty}^{\infty} a_n g_T(t - nT) \tag{6.9}$$

where $T = k/R_b$ is the symbol interval ($1/T = R_b/k$ is the symbol rate), R_b is the bit rate, and $\{a_n\}$ is a sequence of amplitude levels corresponding to the sequence of $k-$bit blocks of information bits. The channel output, which is the received signal at the demodulator, may be expressed as

$$r(t) = \sum_{n=-\infty}^{\infty} a_n h(t - nT) + n(t) \tag{6.10}$$

where $h(t)$ is the impulse response of the cascade of the transmitting filter and the channel; i.e., $h(t) = c(t) \star g_T(t)$, $c(t)$ is the impulse response of the channel, and $n(t)$ represents the AWGN.

The received signal is passed through a linear receiving filter with impulse response $g_R(t)$ and frequency response $G_R(f)$.If $g_R(t)$ is matched to $h(t)$,then its output SNR is a maximum at the proper sampling instant. The output of the receiving filter may be expressed as

$$y(t) = \sum_{n=-\infty}^{\infty} a_n x(t - nT) + v(t) \tag{6.11}$$

where $x(t) = h(t) \star g_R(t) = g_T(t) \star c(t) \star g_R(t)$ and $v(t) = n(t) \star g_R(t)$ denotes the additive noise at the output of the receiving filter.

To recover the information symbols $\{a_n\}$, the output of the receiving filter is sampled periodically, every T seconds. Thus, the sampler produces

$$y(mT) = \sum_{n=-\infty}^{\infty} a_n x(mT - nT) + v(mT) \tag{6.12}$$

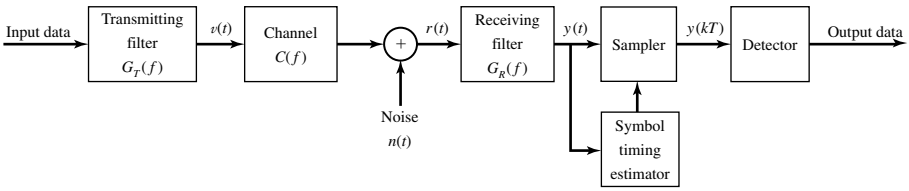


Figure 6.3: Block diagram of digital PAM system.

or, equivalently,

$$\begin{aligned} y_m &= \sum_{n=-\infty}^{\infty} a_n x_{m-n} + v_m \\ &= x_0 a_m + \sum_{n \neq m} a_n x_{m-n} + v_m \end{aligned} \tag{6.13}$$

where $x_m = x(mT)$, $v_m = v(mT)$, and $m = 0, \pm 1, \pm 2, \ldots$. A timing signal extracted from the received signal is used as a clock for sampling the received signal (covered in synchronization).

The first term on the right-hand side (RHS) of Eq.(6.13) is the desired symbol a_m , scaled by the gain parameter x_0. When the receiving filter is matched to the received signal h(t), the scale factor is

$$x_0 = \int_{-\infty}^{\infty} h^2(t)dt = \int_{-\infty}^{\infty} |H(f)|^2 df = \int_{-W}^{W} |G_T(f)|^2 |C(f)|^2 df = E_h \tag{6.14}$$

The second term on the RHS of Equation Eq.(6.13) represents the effect of the other symbols at the sampling instant $t = mT$, called the intersymbol interference (ISI). In general, ISI causes a degradation in the performance of the digital communication system. Finally, the third term, v_m, that represents the additive noise, is a zero-mean Gaussian random variable with variance $\sigma_v^2 = N_0 E_h/2$.

By appropriate design of the transmitting and receiving filters, it is possible to satisfy the condition $x_n = 0$ for $n \neq 0$, so that the intersymbol interference (ISI) term vanishes. In this case, the only term that can cause errors in the received digital sequence is the additive noise. The design of transmitting and receiving filters is considered in the next sections.

6.3 The Power Spectrum of Digitally Modulated Signals.

We will derive the power spectrum of a baseband signal. As shown above, the equivalent baseband transmitted signal for a digital PAM signal is represented in the general form as

$$v(t) = \sum_{n=-\infty}^{\infty} a_n g_T(t - nT) \tag{6.15}$$

where $\{a_n\}$ is the sequence of values selected from a PAM constellation corresponding to the information symbols from the source, and $g_T(t)$ is the impulse response of the transmitting filter. Since the information sequence $\{a_n\}$ is random, $v(t)$ is a sample function of a random process $V(t)$. In this section we evaluate the power-density spectrum of $V(t)$. Our approach is to derive the autocorrelation function of $V(t)$ and then to determine its Fourier transform in order to find the PSD of $v(t)$ using Wiener-Khinchin-Einestein Theorem.

First, the mean value of $v(t)$ is

$$\begin{aligned} E[V(t)] &= \sum_{n=-\infty}^{\infty} E(a_n) g_T(t - nT) \\ &= m_a \sum_{n=-\infty}^{\infty} g_T(t - nT) \end{aligned} \tag{6.16}$$

where m_a is the mean value of the random sequence $\{a_n\}$. Note that although m_a is a constant, the term $\sum_n g_T(t-nT)$ is a periodic function with period T. Hence, the mean value of $V(t)$ is periodic with period T. The autocorrelation function of $V(t)$ is

$$R_V(t,t+\tau) = E[V(t)V(t+\tau)] = \sum_{n=-\infty}^{\infty}\sum_{m=-\infty}^{\infty} E(a_n a_m) g_T(t-nT) g_T(t+\tau-mT) \tag{6.17}$$

In general, we assume that the information sequence $\{a_n\}$ is wide-sense stationary with autocorrelation sequence

$$R_a(n) = E[a_n a_{n+m}] \tag{6.18}$$

Hence, Eq. (6.17) may be expressed as

$$\begin{aligned} R_V(t,t+\tau) &= \sum_{n=-\infty}^{\infty}\sum_{m=-\infty}^{\infty} R_a(m-n) g_T(t-nT) g_T(t+\tau-mT) \\ &= \sum_{m=-\infty}^{\infty} R_a(m) \sum_{n=-\infty}^{\infty} g_T(t-nT) g_T(t+\tau-nT-mT) \end{aligned} \tag{6.19}$$

We observe that the second summation in Eq. (6.19) is periodic with period T. Consequently, the autocorrelation function $R_V(t+\tau,t)$ is periodic in the variable t; i.e.,

$$R_V(t+T+\tau, t+T) = R_V(t+\tau,t)$$

Therefore, the random process $V(t)$ has a periodic mean and a periodic autocorrelation. Such a random process is *cylcostationary*.

The power-spectral density of a cyclostationary process can be determined by first averaging the autocorrelation function $R_V(t+\tau,t)$ over a single period T and then computing the Fourier transform of the average autocorrelation function. Thus, we have

$$\begin{aligned} \bar{R}_V(\tau) &= \frac{1}{T}\int_{-T/2}^{T/2} R_V(t+\tau,t)dt \\ &= \sum_{m=-\infty}^{\infty} R_a(m) \sum_{n=-\infty}^{\infty} \frac{1}{T}\int_{-T/2}^{T/2} g_T(t-nT) g_T(t+\tau-nT-mT)dt \\ &= \sum_{m=-\infty}^{\infty} R_a(m) \sum_{n=-\infty}^{\infty} \frac{1}{T}\int_{nT-T/2}^{nT+T/2} g_T(t) g_T(t+\tau-mT)dt \\ &= \frac{1}{T}\sum_{m=-\infty}^{\infty} R_a(m)\int_{-\infty}^{\infty} g_T(t) g_T(t+\tau-mT)dt \end{aligned} \tag{6.20}$$

We interpret the integral in Eq. (6.20) as the time-autocorrelation function of $g_T(t)$ and define it as

$$R_g(\tau) = \int_{-\infty}^{\infty} g_T(t) g_T(t+\tau)dt \tag{6.21}$$

With this definition, the average autocorrelation function of $V(t)$ becomes

$$\bar{R}_V(\tau) = \frac{1}{T}\sum_{m=-\infty}^{\infty} R_a(m) R_g(\tau-mT) \tag{6.22}$$

Hence, the Fourier transform of the previous equation is

$$\begin{aligned} S_V(f) &= \int_{-\infty}^{\infty} \bar{R}_V(\tau) e^{-j2\pi f\tau} d\tau \\ &= \frac{1}{T}\sum_{m=-\infty}^{\infty} R_a(m)\int_{-\infty}^{\infty} R_g(\tau-mT) e^{-j2\pi f\tau} d\tau \\ &= \frac{|G_T(f)|^2}{T}\sum_{m=-\infty}^{\infty} R_a(m) e^{-j2\pi f\tau} = \frac{|G_T(f)|^2}{T} S_a(f) \end{aligned} \tag{6.23}$$

where $S_a(f)$ is the power spectrum of the information sequence $\{a_n\}$. This result illustrates the dependence of the power-spectral density $S_V(f)$ of the transmitted signal on the spectral characteristics $G_T(f)$ of the transmitting filter and the spectral characteristics $S_a(f)$ of the information sequence $\{a_n\}$. Both $G_T(f)$ and $S_a(f)$ can be designed to control the shape and form of the power spectral density of the transmitted signal.

Example 48. *Consider a binary sequence $\{b_n\}$, from which we form the symbols*

$$a_n = b_n + b_{n-1}$$

The $\{b_n\}$ are assumed to be uncorrelated binary valued (± 1) random variable, each having a zero-mean and a unit variance. Determine the power-spectral entity of the transmitted signal.

Solution. The autocorrelation function of the sequence $\{a_n\}$ is

$$\begin{aligned} R_a(m) &= E[a_n a_{n+m}] \\ &= E[(b_n + b_{n-1})(b_{n+m} + b_{n+m-1}] \\ &= \begin{cases} 2 & m = 0 \\ 1 & m = \pm 1 \\ 0 & \textit{otherwise} \end{cases} \end{aligned} \tag{6.24}$$

Hence, the PSD of the input sequence is

$$S_a(f) = 2(1 + \cos 2\pi fT) = 4\cos^2 \pi fT$$

and the corresponding PSD for the modulated signal is

$$S_V(f) = \frac{4}{T}|G_T(f)|^2 \cos^2 \pi fT$$

As demonstrated in this example, the transmitted signal spectrum can be shaped by having a correlated sequence $\{a_n\}$ as the input to the modulator. □

6.4 Signal Design For Bandlimited Channels

Recall from previous sections that the output of the transmitting filter in a digital PAM may be expressed as

$$v(t) = \sum_{n=-\infty}^{\infty} a_n g_T(t - nT) \tag{6.25}$$

and the output of the channel, which is the received signal at the demodulator, may be expressed as

$$r(t) = \sum_{n=-\infty}^{\infty} a_m h(t - nT) + n(t) \tag{6.26}$$

where $h(t) = c(t) \star g_T(t)$, $c(t)$ is the impulse response of the channel, $g_T(t)$ is the impulse response of the transmitting filter, and $n(t)$ is a sample function of an additive, white Gaussian noise process.

In this section, we consider the problem of designing a bandlimited transmitting filter. The design will be done first under the condition that there is no channel distortion. Later, we consider the problem of filter design when the channel distorts the transmitted signal. Since $H(f) = C(f)G_T(f)$, the condition for distortion-free transmission is that the frequency response characteristic $C(f)$ of the channel have a constant magnitude and a linear phase over the bandwidth of the transmitted signal; i.e.,

$$C(f) = \begin{cases} C_0 e^{-j2\pi f t_0}, & |f| \leq W \\ 0 & \textit{otherwise} \end{cases} \tag{6.27}$$

where W is the available channel bandwidth, t_0 represents an arbitrary finite delay, which we set to zero for convenience, and C_0 is a constant gain factor which we set to unity for convenience. Thus, under the condition that the channel is distortion-free,

$H(f) = G_T(f)$ for $|f| \leq W$ and zero for $|f| > W$.Consequently, the matched filter has a frequency response $H^*(f) = G_T^*(f)$ and its output at the periodic sampling times $t = mT$ has the form

$$y(mT) = x(0)a_m + \sum_{n \neq m} a_n x(mT - nT) + v(mT) \tag{6.28}$$

or, more simply,

$$y_m = x_0 a_m + \sum_{n \neq m} a_n x_{m-n} + v_m \tag{6.29}$$

where $x(t) = g_T(t) \star g_R(t)$ and $v(t)$ is the output response of the matched filter to the AWGN process $n(t)$.

The middle term on the RHS of Eq.(6.29) represents the ISI. The amount of ISI and noise that is present in the received signal can be viewed on an oscilloscope. Specifically, we may display the received signal on the vertical input with the horizontal sweep rate set at $1/T$. The resulting oscilloscope display is called an *eye pattern* because of its resemblance to the human eye. Examples of two eye patterns, one for binary PAM and the other for quaternary $(M = 4)$ PAM, are illustrated in Fig. 6.4. The

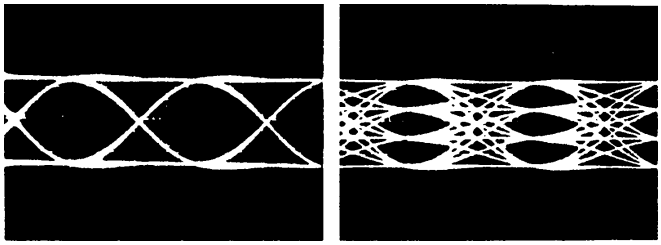


Figure 6.4: Eye patterns

effect of ISI is to cause the eye to close, thereby reducing the margin for additive noise to cause errors. Fig. 6.5 illustrates the effect of ISI in reducing the opening of the eye. Note that ISI distorts the position of the zero crossings and causes a reduction in the eye opening. As a consequence, the system is more sensitive to a synchronization error and exhibits a smaller margin against additive noise. Below we consider the problem of signal design with no ISI at the sampling instants.

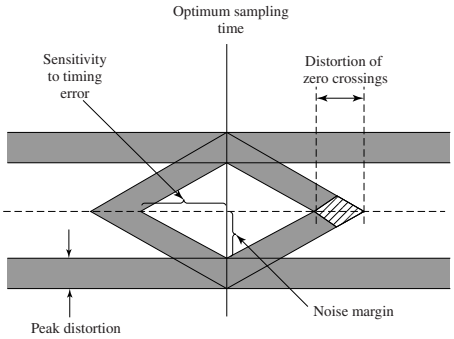


Figure 6.5: Effect of ISI on eye opening.

6.4.1 Design of Bandlimited Signals for Zero ISI – The Nyquist Criterion

As indicated above, in a general digital communication system that transmits through a bandlimited channel, the Fourier transform of the signal at the output of the receiving filter is given by $X(f) = G_T(f)C(f)G_R(f)$ where $G_T(f)$ and $G_R(f)$ denote the transmitter and receiver filters frequency responses and $C(f)$ denotes the frequency response of the channel. We have also seen that the output of the receiving filter, sampled at $t = mT$ is given by

$$y_m = x(0)a_m + \sum_{n=-\infty, n \neq m}^{\infty} x(mT - nT)a_n + v(mT) \tag{6.30}$$

To remove the effect of ISI, it is necessary and sufficient that $x(mT - nT) = 0$ for $n \neq m$ and $x(0) \neq 0$, where without loss of generality we can assume $x(0) = 1$ (The choice of x(0) is equivalent to the choice of a constant gain factor in the receiving filter. This constant gain factor has no effect on the overall system performance since it scales both the signal and the noise.).This means that the overall communication system has to be designed such that

$$x(nT) = \begin{cases} 1, & n = 0 \\ 0 & n \neq 0 \end{cases} \tag{6.31}$$

Now, we derive the necessary and sufficient condition for $X(f)$ in order for $x(t)$ to satisfy the above relation. This condition is known as the *Nyquist pulse-shaping criterion* or *Nyquist condition for zero ISI* and is stated in the following theorem.

Theorem 12 (Nyquist Pulse-Shaping Criterion)**.** *A necessary and sufficient condition for $x(t)$ to satisfy*

$$x(nT) = \begin{cases} 1, & n = 0 \\ 0 & n \neq 0 \end{cases} \tag{6.32}$$

is that its Fourier transform $X(f)$ satisfy

$$\sum_{m=-\infty}^{\infty} X\left(f + \frac{m}{T}\right) = T \tag{6.33}$$

Proof. In general, $x(t)$ is the inverse Fourier transform of $X(f)$. Hence,

$$x(t) = \int_{-\infty}^{\infty} X(f) e^{j2\pi ft} df$$

At the sampling instants $t = nT$, this relation becomes

$$x(nT) = \int_{-\infty}^{\infty} X(f) e^{j2\pi fnT} df \tag{6.34}$$

Let us break up the integral in Eq. (6.4.1) into integrals covering the finite range of $1/T$. Thus, we obtain

$$\begin{aligned} x(nT) &= \sum_{m=-\infty}^{\infty} \int_{(2m-1)/2T}^{(2m+1)/2T} X(f) e^{j2\pi fnT} df \\ &= \sum_{m=-\infty}^{\infty} \int_{1/2T}^{-1/2T} X\left(f + \frac{m}{T}\right) e^{j2\pi fnT} df \\ &= \int_{1/2T}^{-1/2T} \left[\sum_{m=-\infty}^{\infty} X\left(f + \frac{m}{T}\right)\right] e^{j2\pi fnT} df \\ &= \int_{1/2T}^{-1/2T} Z(f) e^{j2\pi fnT} df \end{aligned} \tag{6.35}$$

where $Z(f)$ is defined by

$$Z(f) = \sum_{m=-\infty}^{\infty} X\left(f + \frac{m}{T}\right) \tag{6.36}$$

Obviously, $Z(f)$ is a periodic function with period $1/T$, and therefore it can be expanded in terms of its Fourier series coefficients $\{z_n\}$ as

$$Z(f) = \sum_{m=-\infty}^{\infty} z_n e^{j2\pi nfT} df \tag{6.37}$$

where

$$z_n = T \int_{1/2T}^{-1/2T} Z(f) e^{-j2\pi nfT} df \tag{6.38}$$

Comparing Eq. (6.35) and Eq.(6.38), we obtain

$$z_n = Tx(-nT) \tag{6.39}$$

Therefore, the necessary and sufficient conditions for Eq. (6.32) to be satisfied is that

$$z_n = \begin{cases} T, & n = 0 \\ 0 & n \neq 0 \end{cases} \tag{6.40}$$

which, when substituted into Eq.(6.37), yields

$$Z(f) = T \tag{6.41}$$

or, equivalently,

$$\sum_{m=-\infty}^{\infty} X\left(f + \frac{m}{T}\right) = T \tag{6.42}$$

This concludes the proof of the theorem. □

Now, suppose that the channel has a bandwidth of W. Then $C(f) \equiv 0$ for $|f| > W$ and consequently, $X(f) = 0$ for $|f| > W$. We distinguish three cases:

1. $T < \frac{1}{2W}$, or equivalently, $\frac{1}{T} > 2W$. Since $Z(f) = \sum_{n=-\infty}^{\infty} X\left(f + \frac{n}{T}\right)$ consists of non overlapping replicas of $X(f)$, separated by $\frac{1}{T}$ as show in Fig. 6.6, there is no choice for $X(f)$ to ensure $Z(f) \equiv T$ in this case, and there is no way that we can design a system with no ISI.

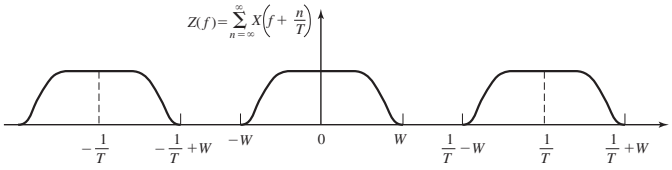


Figure 6.6: Plot of $Z(f)$ for the case $T < \frac{1}{2W}$.

2. $T = \frac{1}{2W}$, or equivalently, $\frac{1}{T} = 2W$ (Nyquist rate). In this case, the replications of $X(f)$, separated by $\frac{1}{T}$, are about to overlap as shown in Fig. 6.7.

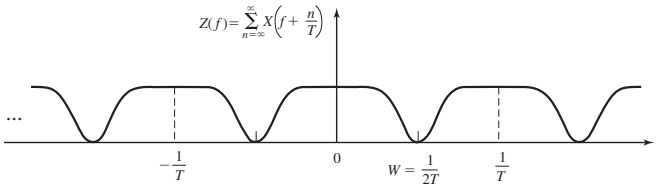


Figure 6.7: Plot of $Z(f)$ for the case $T = \frac{1}{2W}$.

It is clear that in this case there exists only one $X(f)$ that results in $Z(f) = T$, namely,

$$X(f) = \begin{cases} T, & |f| < W \\ 0 & otherwise \end{cases} \tag{6.43}$$

or, $X(f) = Trect\left(\frac{f}{2W}\right)$, which results in

$$x(t) = sinc\left(\frac{t}{T}\right)$$

This means that the smallest value of T for which transmission with zero ISI is possible is $T = \frac{1}{2W}$ and for this value, $x(t)$ has to be a sinc function. The difficulty with this choice of $x(t)$ is that it is non causal and therefore nonrealizable.

To make it realizable, usually a delay version of it; i.e., $sinc(\frac{t-t_0}{T})$ is used and t_0 is chosen such that for $t < 0$, we have $sinc(\frac{t-t_0}{T}) \simeq 0$. Of course with this choice of $x(t)$, the sampling time must also be shifted to $mT + t_0$. A second difficulty with this pulse shape is that its rate of convergence to zero is slow. The tails of $x(t)$ decay as $1/t$, consequently, a small mistiming error in sampling the output of the matched filter at the demodulator results in an infinite series of ISI components. Such a series is not absolutely summable because of the $1/t$ rate of decay of the pulse and, hence, the sum of the resulting ISI does not converge.

3. For $T < \frac{1}{2W}$, $Z(f)$ consists of overlapping replications of $X(f)$ separated by $\frac{1}{T}$, as shown in Fig. 6.8. In this case, there exist numerous choices of $X(f)$, such that $Z(f) \equiv T$.

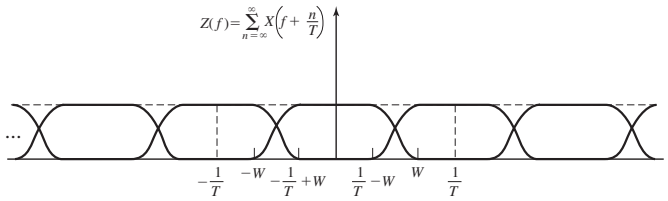


Figure 6.8: Plot of $Z(f)$ for the case $T > \frac{1}{2W}$.

A particular pulse spectrum, for the $T > \frac{1}{2W}$ case, that has desirable spectral properties and has been widely used in practice is the raised cosine spectrum. The raised cosine frequency characteristic is given as

$$X_{rc}(f) = \begin{cases} T, & 0 \le |f| < (1-\alpha)/2T \\ \frac{T}{2}\left[1+\cos\frac{\pi T}{\alpha}\left(|f| - \frac{1-\alpha}{2T}\right)\right], & \frac{1-\alpha}{2T} \le |f| \le \frac{1+\alpha}{2T} \\ 0, & |f| > \frac{1+\alpha}{2T} \end{cases} \tag{6.44}$$

where α is called the *rolloff factor*, which takes values in the range $0 \le \alpha \le 1$. The bandwidth occupied by the signal beyond the Nyquist frequency $\frac{1}{2T}$ is called the *excess bandwidth* and is usually expressed as a percentage of the Nyquist frequency. For example, when $\alpha = 1/2$, the excess bandwidth is 50%, and when $\alpha = 1$ the excess bandwidth is 100%. The pulse $x(t)$ having the raised cosine spectrum is

$$x(t) = \frac{\sin(\pi t/T)}{\pi t/T}\frac{\cos(\pi\alpha t/T)}{1-4\alpha^2t^2/T^2} = sinc(t/T)\frac{\cos(\pi\alpha t/T)}{1-4\alpha^2t^2/T^2} \tag{6.45}$$

Note that $x(t)$ is normalized so that $x(0) = 1$. Fig. **??** illustrates the raised cosine spectral characteristics and the corresponding pulses for $\alpha = 0, 1/2, 1$. We note that for $\alpha = 0$, the pulse reduces to $x(t) = sinc(t/T)$, and the symbol rate $1/T = 2W$. When $\alpha = 1$, the symbol rate is $1/T = W$. In general, the tails of x(t) decay as $1/t^3$ for $\alpha > 0$. Consequently, a mistiming error in sampling leads to a series of intersymbol interference components that converges to a finite value.

Due to the smooth characteristics of the raised cosine spectrum, it is possible to design practical filters for the transmitter and the receiver that approximate the overall desired frequency response. In the special case where the channel is ideal with $C(f) = rect\left(\frac{f}{2W}\right)$,we have

$$X_{rc}(f) = G_T(f)G_R(f) \tag{6.46}$$

In this case, if the receiver filter is matched to the transmitter filter we have $X_{rc}(f) = G_T(f)G_R(f) = |G_T(f)|^2$. Ideally,

$$G_T(f) = \sqrt{|X_{rc}(f)|}e^{-j2\pi ft_0} \tag{6.47}$$

and $G_R(f) = G_T^*(f)$, where t_0 is some nominal delay that is required to assure physical realizability of the filter. Thus, the overall raised cosine spectral characteristic is split evenly between the transmitting filter and the receiving filter. We should also note that an additional delay is necessary to ensure the physical realizability of the receiving filter.