Digital Communications: A Discrete Time View

(Lecture notes for CSE 4214)

Andrew W. Eckford

Department of Computer Science and Engineering

York University, Toronto, Ontario, Canada

Version: November 24, 2009

Contents

CHAPTER 1

Review: Probability, Random Processes, and Linear Systems

1.1. Probability

In this section, we briefly review some necessary concepts of probability that will be used throughout this text.

1.1.1. Foundations. Basics of probability; joint probability; independence; conditional probability; Bayes' rule.

1.1.2. Discrete-valued random variables. Probability mass function; expected value; mean and variance; examples.

1.1.3. Continuous-valued random variables. Probability density function; expected value; mean and variance; examples.

1.1.4. The Gaussian distribution. Definition; properties (e.g., even function).

A Gaussian random variable x with with mean μ and variance σ^2 has a probability density function given by

$$f(x) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{1}{2\sigma^2}(x-\mu)^2\right). \tag{1.1}$$

Integrals over this pdf may be expressed in terms of the *error function complementary*, erfc$(\cdot)$, which is defined as

$$\text{erfc}(z) = \frac{2}{\sqrt{\pi}} \int_{t=z}^{\infty} \exp(-t^2)dt. \tag{1.2}$$

The function erfc$(\cdot)$ has the following mathematical interpretation: if t is a Gaussian random variable with mean $\mu = 0$ and variance $\sigma^2 = 1/2$, then erfc$(z) = \Pr(|t| > z)$. Furthermore, due to the symmetry of the Gaussian pdf about the mean, we

illustrate in Figure X that

$$\Pr(t > z) = \Pr(t < z) = \frac{1}{2}\text{erfc}(z). \tag{1.3}$$

Using a change of variables, erfc$(\cdot)$ may be used to calculate an arbitrary Gaussian integral. For instance, for the random variable x with pdf $f(x)$ in (1.1), suppose we want to calculate the probability $\Pr(x > z)$. This probability can be expressed as

$$\Pr(x > z) = \int_{x=z}^{\infty} f(x)dx \tag{1.4}$$

$$= \int_{x=z}^{\infty} \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{1}{2\sigma^2}(x-\mu)^2\right) dx. \tag{1.5}$$

Now we make the substitution

$$t = \frac{x-\mu}{\sqrt{2\sigma^2}}. \tag{1.6}$$

To perform a change of variables in an integral, we need to replace both x and dx with the equivalent functions of t. Solving for x, we have that

$$x = \sqrt{2\sigma^2}t + \mu, \tag{1.7}$$

so taking the first derivative of x with respect to t, dx is given by

$$dx = \sqrt{2\sigma^2}dt. \tag{1.8}$$

Substituting (1.7)-(1.8) into (1.5), we get

$$\Pr(x > z) = \int_{x=z}^{\infty} \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{1}{2\sigma^2}(x-\mu)^2\right) dx \tag{1.9}$$

$$= \int_{\sqrt{2\sigma^2}t+\mu=z}^{\infty} \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-t^2\right) \sqrt{2\sigma^2}dt \tag{1.10}$$

$$= \int_{t=(z-\mu)/\sqrt{2\sigma^2}}^{\infty} \frac{1}{\sqrt{\pi}} \exp\left(-t^2\right) dt \tag{1.11}$$

$$= \frac{1}{2}\text{erfc}\left(\frac{z-\mu}{\sqrt{2\sigma^2}}\right). \tag{1.12}$$

1.2. Discrete-Time Random Processes

There are many ways to define a random process, but for our purposes, the following is sufficient:

- A random process is a function of time $X(t)$, so that for each fixed time t^*, $X(t^*)$ is a random variable.

As a result, we can write the probability density function (pdf) of the random process at any given time. For example, $f_{X(t^*)}(x)$ represents the pdf of the random process at time t^*. Joint probability density functions measure the joint probability of the process at k different times; these are called *kth order statistics* of the random process. For example, for $k = 2$ and times t_1 and t_2, we can write the second order statistics as $f_{X(t_1),X(t_2)}(x_1, x_2)$.

1.2.1. Definition, Mean, and Variance. It's easy to imagine a random process in discrete time, as merely a sequence of random variables, one for each time interval. For instance, consider the following two random processes defined at integer times $t \in \{\ldots, -2, -1, 0, 1, 2, \ldots\}$:

EXAMPLE 1.1. *At each time $t \in \{\ldots, -2, -1, 0, 1, 2, \ldots\}$, a fair coin is flipped. If the coin shows heads after the flip at time t, then $X(t) = 1$; otherwise, $X(t) = -1$. Thus, for any integer t^*, we can write*

$$f_{X(t^*)}(x) = \begin{cases} 0.5, & x = +1; \\ 0.5, & x = -1; \\ 0 & \text{otherwise.} \end{cases}$$

Since, at each fixed time t, the random process is a random variable, we can calculate the mean and variance of the process at each fixed time as usual for random variables. Thus, for the process as a whole, the mean and variance for a random process are calculated as functions of time. For instance, for the process in Example 1.1, the mean of this process is given by

$$\begin{aligned} \mu(t) &= \sum_{x \in \{+1,-1\}} x f_{X(t)}(x) \\ &= (+1)(0.5) + (-1)(0.5) \\ &= 0 \end{aligned}$$

for all t. The variance of the process is given by

$$\begin{aligned} \sigma^2(t) &= \sum_{x \in \{+1,-1\}} (x - \mu(t))^2 f_{X(t)}(x) \\ &= (+1 - 0)^2(0.5) + (-1 - 0)^2(0.5) \\ &= 1 \end{aligned}$$

for all t.

As an alternative, the following more compicated example has mean and variance that are non-trivial functions of time:

EXAMPLE 1.2. *Let $X(0) = 0$. For each $t \in \{1, 2, \ldots\}$, a fair coin is flipped. If the coin shows heads after the flip at time t, then $X(t) = X(t-1) + 1$; otherwise, $X(t) = X(t-1)$.*

For any t, it is clear that $X(t)$ is the number of heads in the previous t trials. Such random variables are represented by the binomial distribution [**1**]. *Thus,*

$$f_{X(t)}(x) = \binom{t}{x} \frac{1}{2^t}.$$

The mean of this random process is given by

$$\mu(t) = \frac{t}{2},$$

and the variance is given by

$$\sigma^2(t) = \frac{t}{4}.$$

The reader is asked to prove these values in the exercises.

Instances of the random processes from Examples 1.1 and 1.2 are given in Figure 1.1.

1.2.2. Autocorrelation. Suppose you wanted a measure of correlation between two random variables, X_1 and X_2, with the same mean $\mu = 0$ and the same variance $\sigma^2 > 0$. As a candidate for this measure, consider

$$R = E[X_1 X_2]. \tag{1.13}$$

If the random variables are independent (i.e., uncorrelated), then since $E[X_1 X_2] = E[X_1]E[X_2]$ for independent random variables, we would have

$$R = E[X_1]E[X_2] = \mu^2 = 0,$$

bearing in mind that each of the random variables are zero mean. On the other hand, if the two random variables are completely correlated (i.e., $X_1 = X_2$), we would have

$$R = E[X_1 X_2] = E[X_1^2] = \sigma^2.$$

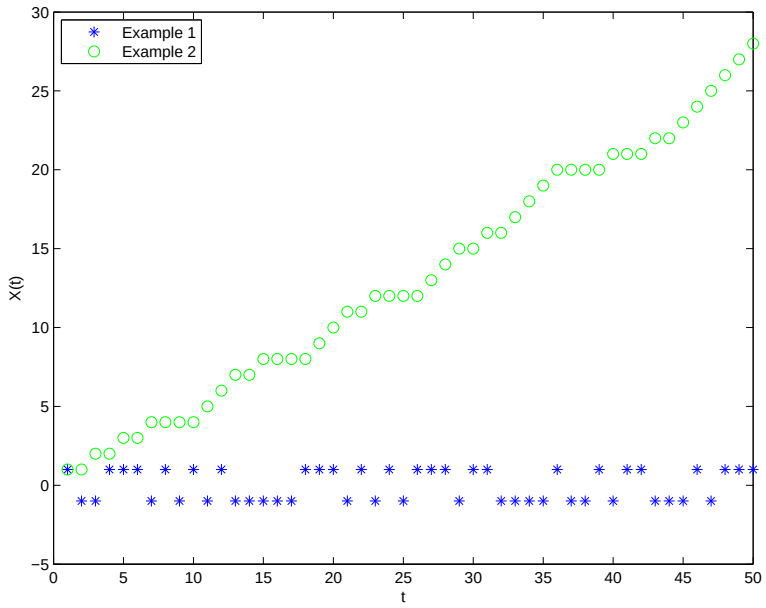

FIGURE 1.1. Illustration of the discrete-time random processes from Examples 1.1 and 1.2.

Further, if they were completely anticorrelated (i.e., $X_1 = -X_2$), it is easy to see that $R = -\sigma^2$.

This measure of correlation also has the following nice property:

THEOREM 1.1. *Given the above definitions,* $|R| \leq \sigma^2$.

Proof: Start with $E[(X_1 + X_2)^2]$. We can write:

$$\begin{aligned} E[(X_1 + X_2)^2] &= E[X_1^2 + 2X_1X_2 + X_2^2] \\ &= E[X_1^2] + 2E[X_1X_2] + E[X_2^2] \\ &= \sigma^2 + 2R + \sigma^2 \\ &= 2\sigma^2 + 2R. \end{aligned}$$

Since $(X_1 + X_2)^2 \geq 0$ for all X_1 and X_2, it is true that $E[(X_1 + X_2)^2] \geq 0$. Thus, $2\sigma^2 + 2R \geq 0$, so $R \geq -\sigma^2$. Repeating the same procedure but starting with $E[(X_1 - X_2)^2]$, we have that $R \leq \sigma^2$, and the theorem follows. ■

Since $R = 0$ when X_1 and X_2 are independent, $R = \sigma^2$ (the maximum possible value) when they are completely correlated, and $R = -\sigma^2$ (the minimum possible

value) when they are completely anticorrelated, R is a good candidate for a correlation measure. The magnitude of R indicates the degree of correlation between X_1 and X_2, while the sign indicates whether the variables are correlated or anticorrelated. Properties of this correlation measure when the variances are unequal, or when the means are nonzero, are considered in the exercises.

We apply this correlation measure to different time instants of the same random process, which we refer to as the *autocorrelation*. In particular, let $X(t)$ be a discrete-time random process defined on $t \in \{\ldots, -2, -1, 0, 1, 2, \ldots\}$. Then the autocorrelation between $X(t_1)$ and $X(t_2)$ is defined as

$$R(t_1, t_2) = E[X(t_1)X(t_2)]. \tag{1.14}$$

Note the similarity with (1.13), since $X(t)$ is merely a random variable for each time t. For the same reason, $R(t_1, t_2)$ has all the same properties as R.

1.2.3. Stationary random processes. A *stationary* discrete-time random process is a process for which the statistics do not change with time. Formally, a process is stationary if and only if

$$f_{X(t_1),X(t_2),\ldots,X(t_k)}(x_1, x_2, \ldots, x_k) = f_{X(t_1+\tau),X(t_2+\tau),\ldots,X(t_k+\tau)}(x_1, x_2, \ldots, x_k) \tag{1.15}$$

for all $k \in \{1, 2, \ldots\}$ and all $\tau \in \{\ldots, -2, -1, 0, 1, 2, \ldots\}$. This does *not* imply that the process $X(t)$ is constant with respect to time, only that the statistical variation of the process is the same, regardless of when you examine the process. The process in Example 1.1 is stationary; intuitively, this is because we keep flipping the same unchanging coin, and recording the outcome in the same way at all t.

We now examine the effects of stationarity on the mean, variance, and autocorrelation of a discrete-time random process $X(t)$. The mean $\mu(t)$ is calculated as follows:

$$\begin{aligned} \mu(t) &= \int_x x f_{X(t)}(x) \mathrm{d}x \\ &= \int_x x f_{X(t+\tau)}(x) \mathrm{d}x \\ &= \mu(t+\tau), \end{aligned}$$

where the second line follows from the fact that $f_{X(t)} = f_{X(t+\tau)}$ for all $\tau \in \{\ldots, -2, -1, 0, 1, 2, \ldots\}$. Thus, $\mu(t) = \mu(t+\tau)$ for all τ, so $\mu(t)$ must be a constant with respect to t. Using a similar line of reasoning, we can show that $\sigma^2(t)$ is a constant with respect to t. Thus, for stationary random processes, we will write $\mu(t) = \mu$ and $\sigma^2(t) = \sigma^2$ for all t.

For the autocorrelation, we can write

$$\begin{aligned} R(t_1, t_2) &= E[X(t_1)X(t_2)] \\ &= \int_{x_1}\int_{x_2} x_1 x_2 f_{X(t_1),X(t_2)}(x_1, x_2)\mathrm{d}x_2\mathrm{d}x_1 \qquad (1.16) \\ &= \int_{x_1}\int_{x_2} x_1 x_2 f_{X(t_1+\tau),X(t_2+\tau)}(x_1, x_2)\mathrm{d}x_2\mathrm{d}x_1. \qquad (1.17) \end{aligned}$$

Let $\tau = \tau' - t_1$. Substituting back into (1.17), we have

$$\begin{aligned} R(t_1, t_2) &= \int_{x_1}\int_{x_2} x_1 x_2 f_{X(t_1+\tau'-t_1),X(t_2+\tau'-t_1)}(x_1, x_2)\mathrm{d}x_2\mathrm{d}x_1 \\ &= \int_{x_1}\int_{x_2} x_1 x_2 f_{X(\tau'),X(t_2-t_1+\tau')}(x_1, x_2)\mathrm{d}x_2\mathrm{d}x_1. \qquad (1.18) \end{aligned}$$

However, in (1.18), since $X(t)$ is stationary, $f_{X(\tau'),X(t_2-t_1+\tau')}(x_1, x_2)$ does not change for any value of τ'. Thus, setting $\tau' = 0$, we can write

$$R(t_1, t_2) = \int_{x_1}\int_{x_2} x_1 x_2 f_{X(0),X(t_2-t_1)}(x_1, x_2)\mathrm{d}x_2\mathrm{d}x_1,$$

which is not dependent on the exact values of t_1 or t_2, but only on the difference $t_2 - t_1$. As a result, we can redefine the autocorrelation function for stationary random processes as $R(t_2 - t_1)$; further, reusing τ to represent this difference, we will usually write $R(\tau)$, where

$$R(\tau) = E[X(t)X(t+\tau)]$$

for all t.

The properties that $\mu(t) = \mu$, $\sigma^2(t) = \sigma^2$, and $R(t_1, t_2) = R(t_2 - t_1)$ apply only to the first and second order statistics of the process $X(t)$. In order to verify whether a process is stationary, it is necessary to prove the condition (1.15) for every order of statistics. In general this is a difficult task. However, in some circumstances, only first and second order statistics are required. In this case, we define a *wide-sense stationary* (WSS) process as any process which satisfies the first and second order conditions of $\mu(t) = \mu$, $\sigma^2(t) = \sigma^2$, and $R(t_1, t_2) = R(t_2 - t_1)$. We have shown that

all stationary processes are WSS, but it should seem clear that a WSS process is not necessarily stationary.

1.2.4. Power spectral density. For a wide-sense stationary random process, the *power spectral density* (PSD) of that process is the Fourier transform of the autocorrelation function:

$$S_x(j\omega) = \mathcal{F}\left[R_x(\tau)\right] = \int_{\tau=-\infty}^{\infty} R_x(\tau) e^{-j\omega\tau} d\tau. \tag{1.19}$$

Properties of PSD:

$$\mathrm{Var}(x[k]) = R_x(0) = \frac{1}{2\pi}\int_{-\infty}^{\infty} S_x(j\omega) d\omega. \tag{1.20}$$

1.3. Linear systems

1.3.1. Review of linear systems.

1.3.2. Linear systems and random processes. Apply a linear filter with frequency-domain transfer function $H(j\omega)$ to a wide-sense stationary random process with PSD $S_x(j\omega)$. The output is a random process with PSD $S_w(j\omega)$, where

$$S_w(j\omega) = S_x(j\omega)|H(j\omega)|^2. \tag{1.21}$$

1.4. Problems

(1) For the random process in Example 1.2, show that $\mu(t) = t/2$, and $\sigma^2(t) = t/4$. Is this process stationary? Explain.

(2) Suppose X_1 and X_2 are zero-mean random variables with variances σ_1^2 and σ_2^2, respectively. For the correlation measure R defined in (1.13), show that
$$|R| \leq \frac{\sigma_1^2 + \sigma_2^2}{2}.$$

(3) Suppose X_1 and X_2 have the same *nonzero* mean μ, and the same variance σ^2. For the correlation measure R defined in (1.13), show that $|R| \leq \sigma^2 + \mu^2$.

(4) Give an example of a discrete-time random process for which $\mu(t) = \mu$ and $\sigma^2(t) = \sigma^2$ for all t, but there exist t_1 and t_2 such that $R(t_1, t_2) \neq R(t_2 - t_1)$.

(5) Calculate $\mu(t)$ and $R(t_1, t_2)$ for the continuous time random process given in Example **??**. Is this process stationary? Explain.

(6) Let $X(t) = X\sin(2\pi t)$, where X is a random variable corresponding to the result of a single fair coin flip: $X = 1$ if the coin is heads, and $X = -1$ is the coin is tails. Does $X(t)$ satisfy the definition of a continuous-time random process? If so, calculate $f_{X(t)}(x)$; if not, explain why not.

1.5. Laboratory Exercise: Probability and Random Processes

In this laboratory exercise, you will investigate the properties of discrete-valued random variables and random processes.

1.5.1. Generating arbitrary random variables. Let x be a discrete-valued random variable, taking values on 1, 2, ..., 6, with probability mass function $p(x)$.

- MATLAB provides a routine, `rand`, which generates uniformly distributed random variables on the range from 0 to 1. Given $p(x)$, propose a way to generate instances of x, with probabilities $p(x)$, from `rand`.
- Write a MATLAB function, called `xrand`, implementing the method you describe. The routine takes a 1×6 vector, where the first element of the vector is $p(1)$, the second is $p(2)$, and so on. The routine returns a value on 1, 2, ..., 6 at random according to the probabilities $p(x)$.

Discussion of empirical distributions.

Given a distribution, write a function to calculate the mean and variance, both empirically and theoretically.

Consider the following Gaussian random process: ... Plot the autocorrelation, both empirically and

CHAPTER 2

Baseband Data Transmission

2.1. Hardware Model of Communication

2.1.1. Traditional communication system model.

2.1.2. Transmitter hardware model.

2.1.3. Receiver hardware model.

2.2. Noise

Although there are many sources of random distortion in communication systems, we will assume that the dominant source is *thermal noise*, arising from the random motion of electrons in electrical components. This random motion results in small current fluctuations, which can be significant in the presence of a very weak signal. Because there are many independently-moving electrons, all of which contribute randomly to the current fluctuations, the *central limit theorem* [**2**] can be used to model noise as a Gaussian random variable.

Gaussian noise has zero mean. The autocorrelation of a continuous-time Gaussian noise process $n(t)$ is given by

$$R_n(\tau) = \frac{N_0}{2}\delta(\tau), \tag{2.1}$$

and is thus wide-sense stationary. Taking the Fourier transform of $R_n(\tau)$, its power spectral density is given by

$$S_n(j\omega) = \frac{N_0}{2}, \tag{2.2}$$

where N_0 is a constant proportional to the temperature of the device. From (2.2), the power spectrum is the same at all frequencies ω, so the noise is "white", analogously to white light; furthermore, the current fluctuations are added to whatever deterministic signal is present. Thus, we refer to this type of noise as *additive white Gaussian noise* (AWGN).

In discrete time, the received signal $y[k]$ is given by

$$y[k] = As[k] + n[k], \tag{2.3}$$

where $s[k]$ represents the signal, A is a scaling factor representing signal attenuation/amplification, and $n[k]$ is the sampled version of the continuous-time AWGN process $n(t)$.

We will now determine the properties of $n[k]$. In the hardware model, we precede the A-to-D converter with an ideal lowpass filter having frequency-domain transfer function $H_{LP}(j\omega)$, where

$$H_{LP}(j\omega) = \begin{cases} 1, & |\omega| < \pi/T_s, \\ 0 & \text{otherwise.} \end{cases} \tag{2.4}$$

That is, the cutoff frequency of the filter is the Nyquist frequency $1/2T_s$. Let w(t) represent the noise random process at the output of the ideal lowpass filter, and let $S_w(j\omega)$ represent its PSD. From (1.21), $S_w(j\omega)$ is given by

$$S_w(j\omega) = S_n(j\omega)|H_{LP}(j\omega)|^2 \tag{2.5}$$

$$= \begin{cases} N_0/2, & |\omega| < \pi/T_s, \\ 0 & \text{otherwise.} \end{cases} \tag{2.6}$$

Thus, from (1.20), the variance of $w(t)$ (and hence each sample $w[k]$) is given by

$$\text{Var}(w(t)) = \text{Var}(w[k]) \tag{2.7}$$

$$= \frac{1}{2\pi}\int_{\omega=-\infty}^{\infty} S_w(j\omega)d\omega \tag{2.8}$$

$$= \frac{1}{2\pi}\cdot\frac{2\pi}{T_s}\cdot\frac{N_0}{2} \tag{2.9}$$

$$= \frac{N_0}{2T_s}. \tag{2.10}$$

We also need to know whether the samples $w[k]$ are independent. Taking the inverse Fourier transform of $S_w(j\omega)$, since $S_w(j\omega)$ is rectangular, we have that (see Appendix A)

$$R_x(\tau) = \mathcal{F}^{-1}\left[S_w(j\omega)\right] \tag{2.11}$$

$$= \frac{1}{T_s}\text{sinc}\left(\frac{\tau}{T_s}\right). \tag{2.12}$$

If τ is a multiple of the sampling frequency, we have

$$R_x(kT_s) = \frac{1}{T_s}\text{sinc}\,(k) \tag{2.13}$$

$$= \begin{cases} \frac{1}{T_s}, & k = 0, \\ 0, & k \neq 0. \end{cases} \tag{2.14}$$

Thus, recalling our discussion of correlation in Chapter 1, the noise process $w[k]$ is *uncorrelated* from sample to sample. Since $w[k]$ is a Gaussian random process, this is sufficient to show that it is *independent* from sample to sample.

2.3. Modulation and Detection

2.3.1. Modulation. Data can be represented as an arbitrarily long vector of binary $\{0, 1\}$ symbols, as in Figure X. The goal of *modulation* is to transform such a vector into a function of time, which is necessary before the bit can be transmitted over the medium.

Since this book deals with discrete-time signal processing, the modulator will replace each bit with a non-overlapping sequence of samples; the function of time will then be generated in digital-to-analog conversion. From Section 2.1, each bit consists of $n_b = T_b/T_s$ samples, so we should define two sequences of n_b samples each: one to represent 0, and one to represent 1. Let $s_0[k]$ and $s_1[k]$ represent these sequences for bits 0 and 1, respectively.

The following examples illustrate modulation in detail.

EXAMPLE 2.1 (Polar nonreturn-to-zero). *In polar nonreturn-to-zero (NRZ), for any n_b, let*

$$s_0[k] = \begin{cases} 1, & 1 \leq k \leq n_b, \\ 0, & \text{otherwise}, \end{cases} \tag{2.15}$$

and

$$s_1[k] = -s_0[k]. \tag{2.16}$$

EXAMPLE 2.2 (Binary phase shift keying). *In binary phase shift keying (BPSK), suppose for 0, the bit time T_b contains exactly one cycle of a sinusoid; thus, there*

would be exactly one cycle over n_b samples. We can write

$$s_0[k] = \begin{cases} \sin(2\pi k/n_b), & 1 \le k \le n_b, \\ 0, & \text{otherwise.} \end{cases} \tag{2.17}$$

Furthermore, let

$$s_1[k] = \begin{cases} \sin(2\pi k/n_b + \pi), & 1 \le k \le n_b, \\ 0, & \text{otherwise,} \end{cases} \tag{2.18}$$

as depicted in Figure X. From (2.18), the phase is shifted by π radians in order to transmit a 1, hence the terminology. However, also note that $s_1[k] = -s_0[k]$.

Now let $\mathbf{b} = [0, 1, 1, 0, 1]$, and let $n_b = 5$. Replacing 0 and 1 with $s_0[k]$ and $s_1[k]$, respectively, from each example, we obtain discrete-time signals depicted in Figure X.

2.3.2. Detection. From (2.3), the received signal is corrupted by an AWGN random process $n[k]$. The detector's job is to extract the value of the bit, 0 or 1, from the noisy signal $y[k]$, as accurately as possible.

The detector consists of two components: a *filter*, which performs signal processing on $y[k]$, and a *decision device*, which takes the output of the filter and determines whether a 0 or 1 was sent. Typically, the decision device examines the output value of the filter after each bit has been sent (i.e., at integer multiples of the bit time T_b, or integer multiples of n_b in discrete time); we will call these values the *filter outputs*. We will assume that the filter is linear and time invariant, and has impulse response $h[k]$. The filter outputs $\phi(j)$ are given by

$$\phi(j) = [y[k] \star h[k]]_{jn_b}, \tag{2.19}$$

where the notation $\star$ represents discrete-time convlution, and the notation $[\cdot]_{jn_b}$ indicates that the expression is evaluated at time jn_b.

Given $\phi(j)$, the decision device then assigns a bit, 0 or 1, to each possible output of the filter. This is usually done through a threshold (i.e., the bit is a 0 if the filter output exceeds the threshold, or a 1 if the filter output is less than the threshold). Let $d_z(x)$ represent the decision function on x with threshold z, where

$$d_z(x) = \begin{cases} 0, & x \ge z, \\ 1, & x < z. \end{cases} \tag{2.20}$$

Thus, combining (2.19)-(2.20), the estimated bits $\hat{b}_j$ are given by

$$\hat{b}_j = d_z(\phi(j)). \tag{2.21}$$

Selection of optimal $h[k]$ and z are non-trivial design problems, which will be discussed extensively in the remainder of the book. However, the following example illustrates a correctly designed detector, and its outputs in a noise-free channel.

EXAMPLE 2.3. *Let $s_0[t]$ and $s_1[t]$ be polar NRZ pulses, as defined in (2.15)-(2.16). Let*

$$h[k] = \begin{cases} 1, & 0 \le k < n_b, \\ 0, & \text{otherwise.} \end{cases} \tag{2.22}$$

Note that $h[k] = s_0[k+1]$.

In the absence of noise, the filter output in response to $s_0[k]$ at time n_b is

$$[s_0[k] \star h[k]]_{n_b} = \sum_{i=-\infty}^{\infty} s_0[i]h[n_b - i] \tag{2.23}$$

$$= \sum_{i=1}^{n_b} 1 \tag{2.24}$$

$$= n_b. \tag{2.25}$$

Since $s_1[k] = -s_0[k]$, the filter output in response to $s_1[k]$ is given by

$$[s_1[k] \star h[k]]_{n_b} = -\sum_{i=-\infty}^{\infty} s_0[i]h[n_b - i] \tag{2.26}$$

$$= -\sum_{i=1}^{n_b} 1 \tag{2.27}$$

$$= -n_b. \tag{2.28}$$

From now on, we will let s_0 represent the noise-free filter output when 0 is sent (and, respectively, s_1 when 1 is sent). Thus,

$$s_0 = [s_0[k] \star h[k]]_{n_b}, \tag{2.29}$$

and

$$s_1 = [s_1[k] \star h[k]]_{n_b}. \tag{2.30}$$

Because the detection filter is linear, the effect of noise will be added to the noise-free output.

Now consider the effect of noise. From (2.3), since convolution distributes over addition, applying the filter $h[k]$ to $y[k]$ will result in

$$y[k] \star h[k] = (s[k] + n[k]) \star h[k] \tag{2.31}$$

$$= s[k] \star h[k] + n[k] \star h[k], \tag{2.32}$$

which consists of a signal term $s[k]\star h[k]$ and a noise term $n[k]\star h[k]$. The signal term in (2.32) can be obtained by calculating $h[k] \star s_0[k]$ and $h[k] \star s_1[k]$, as in Example 2.3 for polar NRZ. Considering the noise term, evaluating the filter output at time n_b, we can write

$$[n[k] \star h[k]]_{n_b} = \sum_i h[i]n[n_b - i], \tag{2.33}$$

where the sum is over all possible values of i (which can be restricted to those values of i for which $h[i] \neq 0$, i.e., the "support" of $h[i]$). Since $n[k]$ is an AWGN random process with mean zero and variance $N_0/2T_s$ (from (2.10)), the sum in (2.33) is a weighted sum of independent Gaussian random variables, with mean zero and variance

$$\sigma^2 = \frac{N_0}{2T_s} \sum_i h[i]^2. \tag{2.34}$$

Recalling Chapter 1, the sum of Gaussian random variables is itself a Gaussian random variable. Thus, if the transmitted symbol is known to be 0, the filter output has mean $[s_0[k] \star h[k]]_{n_b}$ and variance σ^2. Similarly, if the transmitted symbol is known to be 1, the filter output has mean $[s_1[k] \star h[k]]_{n_b}$ and variance σ^2.

2.4. Error analysis

2.4.1. General form of the probability of error. An error occurs if $\hat{b}_j \neq b_j$ (we will refer to this event as error). Using the decision function $d_z(\phi[j])$ from (2.20), we have that $\hat{b}_j = 0$ if $\phi[j] \geq z$; thus, an error occurs if $b_j = 1$ and $\phi[j] \geq z$. Similarly, an error occurs if $b_j = 0$ and $\phi[j] < z$.

Remember that if b_j is known, then $\phi[j]$ is a Gaussian random variable. Thus, using properties of Gaussian random variables, we can calculate the conditional error probabilities $\Pr(\mathsf{error}|b_j = 0)$ and $\Pr(\mathsf{error}|b_j = 1)$. The average error probability is then given by

$$\Pr(\mathsf{error}) = \Pr(\mathsf{error}|b_j = 0)\Pr(b_j = 0) + \Pr(\mathsf{error}|b_j = 1)\Pr(b_j = 1). \tag{2.35}$$

Let's start with $\Pr(\mathsf{error}|b_j = 1)$. Clearly

$$\Pr(\mathsf{error}|b_j = 1) = \Pr(\phi[j] \geq z)|b_j = 1). \tag{2.36}$$

This event is illustrated in Figure X. Given $b_j = 1$, $\phi[j]$ is a Gaussian random variable with mean s_1 (from (2.30)) and variance σ^2 (from (2.34)). The PDF of this random variable is given by

$$f(\phi[k]|b_j = 1) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{1}{2\sigma^2}(\phi[k] - s_1)^2\right). \tag{2.37}$$

Thus, since $\phi[k]$ is a continuous-valued random variable,

$$\Pr(\mathsf{error}|b_j = 1) = \Pr(\phi[j] \geq z)|b_j = 1) \tag{2.38}$$

$$= \int_{\phi[k]=z}^{\infty} \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{1}{2\sigma^2}(\phi[k] - s_1)^2\right) d\phi[k]. \tag{2.39}$$

As we mentioned in Chapter 1, integrals over the Gaussian PDF, such as the one in (2.39), cannot be directly computed. However, we can use a special function known as the error function complementary, $\mathrm{erfc}(\cdot)$, defined in (1.2), to represent this integral. With a change of variables, the integral in (2.39) can be expressed in terms of $\mathrm{erfc}(\cdot)$ as

$$\Pr(\mathsf{error}|b_j = 1) = \frac{1}{2}\mathrm{erfc}\left(\frac{z - s_1}{\sqrt{2\sigma^2}}\right). \tag{2.40}$$

By a similar derivation, $\Pr(\mathsf{error}|b_j = 0)$ is given by

$$\Pr(\mathsf{error}|b_j = 0) = \frac{1}{2}\mathrm{erfc}\left(\frac{s_0 - z}{\sqrt{2\sigma^2}}\right). \tag{2.41}$$

Showing the correctness of (2.40) and (2.41) are left as exercises for the reader. Finally, substituting into (2.35), we have

$$\Pr(\mathsf{error})$$

$$= \Pr(\mathsf{error}|b_j = 0)\Pr(b_j = 0) + \Pr(\mathsf{error}|b_j = 1)\Pr(b_j = 1) \tag{2.42}$$

$$= \frac{1}{2}\mathrm{erfc}\left(\frac{s_0 - z}{\sqrt{2\sigma^2}}\right)\Pr(b_j = 0) + \frac{1}{2}\mathrm{erfc}\left(\frac{z - s_1}{\sqrt{2\sigma^2}}\right)\Pr(b_j = 1). \tag{2.43}$$

2.4.2. Probability of error for Polar NRZ. Let's return to Example 2.3, which illustrated detection with Polar NRZ signals, and use a threshold $z = 0$. In the example, we showed that $s_0 = -s_1 = n_b$. Thus, substituting into (2.43), we have

$$\begin{aligned}
&\Pr(\mathsf{error}) \\
&= \frac{1}{2}\mathrm{erfc}\left(\frac{n_b - 0}{\sqrt{2\sigma^2}}\right)\Pr(b_j = 0) + \frac{1}{2}\mathrm{erfc}\left(\frac{0 - (-n_b)}{\sqrt{2\sigma^2}}\right)\Pr(b_j = 1) && (2.44)\\
&= \frac{1}{2}\mathrm{erfc}\left(\frac{n_b}{\sqrt{2\sigma^2}}\right)\Pr(b_j = 0) + \frac{1}{2}\mathrm{erfc}\left(\frac{n_b}{\sqrt{2\sigma^2}}\right)\Pr(b_j = 1) && (2.45)\\
&= \frac{1}{2}\mathrm{erfc}\left(\frac{n_b}{\sqrt{2\sigma^2}}\right)(\Pr(b_j = 0) + \Pr(b_j = 1)) && (2.46)\\
&= \frac{1}{2}\mathrm{erfc}\left(\frac{n_b}{\sqrt{2\sigma^2}}\right). && (2.47)
\end{aligned}$$

From (2.47), since $\mathrm{erfc}(\cdot)$ is a decreasing function, it seems like increasing n_b leads to a decrease in $\Pr(\mathsf{error})$. However, by examining (2.34), we can show that the probability of error is independent of the sampling rate. In Example 2.3, we used a filter impulse response $h[k] = s_0[k]$, so (2.34) becomes

$$\begin{aligned}
\sigma^2 &= \frac{N_0}{2T_s}\sum_{i=1}^{n_b} s_0[k]^2 && (2.48)\\
&= \frac{N_0}{2T_s}\sum_{i=1}^{n_b} 1 && (2.49)\\
&= \frac{N_0 n_b}{2T_s}. && (2.50)
\end{aligned}$$

However, the sample time T_s can be expressed as

$$T_s = \frac{T_b}{n_b}. \tag{2.51}$$

Substituting into (2.50), we have

$$\sigma^2 = \frac{N_0 n_b^2}{2T_b}, \tag{2.52}$$

and finally, substituting into (2.47), we have

$$\begin{aligned}
\Pr(\mathsf{error}) &= \frac{1}{2}\mathrm{erfc}\left(\frac{n_b}{\sqrt{2(N_0 n_b^2)/(2T_b)}}\right) && (2.53)\\
&= \frac{1}{2}\mathrm{erfc}\left(\sqrt{\frac{T_b}{N_0}}\right). && (2.54)
\end{aligned}$$

Thus, in this example, the only important factors in determining the probability of error are the bit duration, T_b, and the AWGN power spectral density coefficient, N_0.

2.4.3. Summary and Examples. Our calculation of probability of error for Polar NRZ followed a procedure that can be generalized to other signalling schemes. Given modulation signals $s_0[k]$ and $s_1[k]$, detection filter $h[k]$, threshold z, and all relevant system parameters (e.g., i.e., probabilities $\Pr(b_j = 0)$ and $\Pr(b_j = 1)$, noise coefficient N_0, sample time T_s, samples per bit n_b), the following procedure may be used to calculate the probability of error:

(1) Calculate the noise-free filter outputs s_0 and s_1, using equations (2.29)-(2.30).
(2) Calculate the variance of the noise σ^2 at the output of the filter, using equation (2.34). Given that the input bit is 0 (or 1), the output of the filter is then a Gaussian random variable with mean $\mu = s_0$ (resp., s_1) and variance σ^2.
(3) Obtain probability of error by substituting all these quantities into equation (2.43).

Using this procedure, we now present two additional examples.

EXAMPLE 2.4 (Binary Phase Shift Keying). *Returning to example 2.2, we now calculate the probability of error for BPSK. Let*

$$h[k] = \begin{cases} \sin\left(-\frac{2\pi k}{n_b}\right), & 1 \le k \le n_b \\ 0 & \text{otherwise} \end{cases}, \tag{2.55}$$

and let the threshold $z = 0$. We now follow the procedure given above.

Noise-free filter outputs: *Using (2.29), s_0 is given by*

$$s_0 = [s_0[k] \star h[k]]_{n_b} \tag{2.56}$$

$$= \sum_{i=1}^{n_b} s_0[i]h[n_b - i] \tag{2.57}$$

$$= \sum_{i=1}^{n_b} \sin\left(\frac{2\pi i}{n_b}\right) \sin\left(\frac{2\pi(i-n_b)}{n_b}\right) \tag{2.58}$$

$$= \sum_{i=1}^{n_b} \sin\left(\frac{2\pi i}{n_b}\right) \sin\left(\frac{2\pi i}{n_b} + 2\pi\right) \tag{2.59}$$

$$= \sum_{i=1}^{n_b} \sin\left(\frac{2\pi i}{n_b}\right)^2. \tag{2.60}$$

Furthermore, it is easy to show that

$$s_1 = -s_0. \tag{2.61}$$

Variance of the noise: *Using (2.34), σ^2 is given by*

$$\sigma^2 = \frac{N_0}{2T_s} \sum_{i=1}^{n_b} h[i]^2 \tag{2.62}$$

$$= \frac{N_0}{2T_s} \sum_{i=1}^{n_b} \sin\left(-\frac{2\pi k}{n_b}\right)^2 \tag{2.63}$$

$$= \frac{N_0}{2T_s} \sum_{i=1}^{n_b} \sin\left(\frac{2\pi k}{n_b}\right)^2 \tag{2.64}$$

$$= \frac{N_0}{2T_s} s_0, \tag{2.65}$$

where (2.64) follows from (2.63) since $\sin(-x) = -\sin(x)$.

Probability of error. *Substituting into (2.43), we have*

$$\Pr(\mathsf{error})$$

$$= \frac{1}{2}\mathrm{erfc}\left(\frac{s_0 - z}{\sqrt{2\sigma^2}}\right) \Pr(b_j = 0) + \frac{1}{2}\mathrm{erfc}\left(\frac{z - s_1}{\sqrt{2\sigma^2}}\right) \Pr(b_j = 1) \tag{2.66}$$

$$= \frac{1}{2}\mathrm{erfc}\left(\sqrt{\frac{T_s s_0}{N_0}}\right) \Pr(b_j = 0) + \frac{1}{2}\mathrm{erfc}\left(\sqrt{\frac{T_s s_0}{N_0}}\right) \Pr(b_j = 1) \tag{2.67}$$

$$= \frac{1}{2}\mathrm{erfc}\left(\sqrt{\frac{T_s s_0}{N_0}}\right). \tag{2.68}$$

To obtain specific numbers for this system, let $T_b = 10^{-4}$, $n_b = 8$, *and* $N_0 = 1.25 \cdot 10^{-5}$. *Then from (2.60),* $s_0 = 4$, *so*

$$\Pr(\mathsf{error}) = \frac{1}{2}\mathrm{erfc}\left(\sqrt{\frac{(10^{-4}/8)\cdot 4}{1.25\cdot 10^{-5}}}\right) \tag{2.69}$$

$$= 0.00234. \tag{2.70}$$

EXAMPLE 2.5 (On-off keying). *Let* $s_0[k]$ *be the same as (2.15), and let* $h[k] = s_0[k+1]$, *as in Polar NRZ. However, let* $s_1[k] = 0$ *for all* k. *This is referred to as* on-off keying, *since the transmitter is "on" (all +1) to transmit 0, and "off" (all zero) to transmit 1. Suppose* $\Pr(b_j = 0) = \Pr(b_j = 1) = 1/2$. *We consider two cases: first,* $z = 0$, *and second,* $z = n_b/2$.

Noise-free filter outputs: *Since* $s_0[k]$ *and* $h[k]$ *are the same as in Polar NRZ,* s_0 *is also the same, so*

$$s_0 = [s_0[k] \star h[k]]_{n_b} = n_b. \tag{2.71}$$

Since $s_1[k] = 0$, *then*

$$s_1 = [0 \star h[k]]_{n_b} = 0. \tag{2.72}$$

Variance of the noise: *Since* $h[k]$ *is the same as in Polar NRZ, then*

$$\sigma^2 = \frac{N_0}{2T_s}\sum_i h[k]^2 = \frac{N_0}{2T_s}n_b. \tag{2.73}$$

Probability of error. *Substituting into (2.43), and using* $z = 0$, *we have*

$$\Pr(\mathsf{error})$$

$$= \frac{1}{2}\mathrm{erfc}\left(\frac{s_0 - z}{\sqrt{2\sigma^2}}\right)\Pr(b_j = 0) + \frac{1}{2}\mathrm{erfc}\left(\frac{z - s_1}{\sqrt{2\sigma^2}}\right)\Pr(b_j = 1) \tag{2.74}$$

$$= \frac{1}{4}\mathrm{erfc}\left(\sqrt{\frac{T_s n_b}{N_0}}\right) + \frac{1}{4}\mathrm{erfc}\,(0) \tag{2.75}$$

$$= \frac{1}{4}\left(\mathrm{erfc}\left(\sqrt{\frac{T_s s_0}{N_0}}\right) + 1\right). \tag{2.76}$$

On the other hand, using $z = n_b/2$, *we have*

$$\Pr(\mathsf{error}) = \frac{1}{2}\mathrm{erfc}\left(\sqrt{\frac{n_b T_s}{4N_0}}\right). \tag{2.77}$$

The probability of error in (2.77) is generally smaller (and therefore better) than (2.76), which we illustrate in Figure X.

2.5. Probability of error and energy per bit

Probability of error is frequently expressed in terms of the *average energy per bit* E_b, which allows the system designer to compare two systems on the basis of the same energy expenditure.

In continuous time, the energy E_0 and E_1 contained in the continuous-time modulation functions $s_0(t)$ and $s_1(t)$, respectively, are expressed by

$$E_0 = \int_{t=0}^{T_b} s_0(t)^2 dt \tag{2.78}$$

and

$$E_1 = \int_{t=0}^{T_b} s_1(t)^2 dt. \tag{2.79}$$

Average energy per bit is then given by

$$E_b = E_0 \Pr(b_j = 0) + E_1 \Pr(b_j = 1). \tag{2.80}$$

In discrete time, calculation of the energy per bit is dependent on the digital-to-analog hardware that is used to transform $s_0[k]$ and $s_1[k]$ into continuous-time functions $s_0(t)$ and $s_1(t)$, respectively. In this book, we will use the following method: each sample will be replaced with a rectangular function of width T_s and amplitude equal to the sample value, where the rectangle corresponding to the kth sample occupies time between $t = (k-1)T_s$ and $t = kT_s$. This scheme is depicted in Figure X.

More formally, let $r_{T_s}(t)$ be a rectangular function over the interval T_s, given by

$$r_{T_s}(t) = \begin{cases} 1, & -T_s \leq t < 0 \\ 0 & \text{otherwise} \end{cases} \tag{2.81}$$

This is a more convenient form of the rect$(\cdot)$ function, defined in the appendix. Furthermore, the rectangle is defined on the interval $[-T_s, 0)$ because, from the definition above, the rectangle "lags" the sample. Then $s_0(t)$ is given by

$$s_0(t) = \sum_{i=1}^{n_b} s_0[i] r_{T_s}(t - iT_s). \tag{2.82}$$

To calculate energy E_0 for bit 0, we can now use (2.78):

$$E_0 = \int_{t=0}^{T_b} \left(\sum_{i=1}^{n_b} s_0[i] r_{T_s}(t - iT_s) \right)^2 dt \tag{2.83}$$

$$= \sum_{i=1}^{n_b} s_0[i]^2 \int_{t=0}^{T_b} r_{T_s}(t - iT_s) dt \tag{2.84}$$

$$= T_s \sum_{i=1}^{n_b} s_0[i]^2, \tag{2.85}$$

where (2.84) follows from (2.83) because the rectangles $r_{T_s}(t - iT_s)$ do not overlap, and have unit amplitude; and where (2.85) follows from (2.84) because the area under $r_{T_s}(t - iT_s)$ is always T_s. Similarly for E_1, we have that

$$E_1 = T_s \sum_{i=1}^{n_b} s_1[i]^2. \tag{2.86}$$

For the three error calculations we gave in this chapter, we can now restate the probability of error as a function of energy per bit. For polar NRZ, we have that

$$E_0 = T_s \sum_{i=1}^{n_b} s_0[i]^2 \tag{2.87}$$

$$= T_s n_b \tag{2.88}$$

$$= T_b, \tag{2.89}$$

and $E_1 = E_0$. Thus, $E_b = T_b$. Substituting back into (2.53), we have

$$\Pr(\mathsf{error}) = \frac{1}{2} \mathrm{erfc} \left(\sqrt{\frac{E_b}{N_0}} \right), \tag{2.90}$$

which directly relates the probability of error to the average energy consumed in transmitting a bit. Restating the equations for the other two modulation schemes are left as exercises.

2.6. Problems

(1) Making the changes of variables described in Chapter 1, show that (2.40) and (2.41) are correct.

(2) Restate the error calculation for binary phase shift keying (Example 2.4) and on-off keying (Example 2.5) in terms of average energy per bit E_b. How do these schemes compare with Polar NRZ in terms of energy efficiency?

2.7. Laboratory exercise

CHAPTER 3

Optimal System Design

In Chapter 2, we outlined the basic problems of modulation and detection, without discussing how parameters such as z and $h[k]$ should be selected. The digital communication system design problem is to minimize the probability of detection error, subject to constraints on the energy per bit E_b. In this chapter, we present the optimal solution to this design problem.

3.1. Optimizing the decision threshold

In the on-off keying example from Chapter 2 (Example 2.5), we saw that the choice of threshold had an impact on the probability of error. Given a pair of signals $s_0[k]$ and $s_1[k]$, and a filter $h[k]$, the threshold z should obviously be selected so as to minimize $\Pr(\mathsf{error})$. How can we do this?

Recall the average error probability expression

$$\Pr(\mathsf{error}) = \frac{1}{2}\mathrm{erfc}\left(\frac{s_0 - z}{\sqrt{2\sigma^2}}\right)\Pr(b_j = 0) + \frac{1}{2}\mathrm{erfc}\left(\frac{z - s_1}{\sqrt{2\sigma^2}}\right)\Pr(b_j = 1), \tag{3.1}$$

and differentiate with respect to z. Doing so, we get an expression closely related to the Gaussian integral: let

$$f_T(t) = \frac{2}{\sqrt{\pi}}\exp(-t^2), \tag{3.2}$$

and let $F_T(t)$ represent the indefinite integral of $f_T(t)$ (which is not available in closed form). Then

$$\mathrm{erfc}(z) = \int_z^\infty f_T(t)dt \tag{3.3}$$

$$= F_T(\infty) - F_T(z). \tag{3.4}$$

However, by the fundamental theorem of calculus, it is true that

$$f_T(t) = \frac{d}{dt}F_T(t), \tag{3.5}$$

so taking the first derivative of $\text{erfc}(z)$ with respect to z, we get

$$\frac{d}{dz}\text{erfc}(z) = -f_T(z). \tag{3.6}$$

Applying this to (3.1), and using the chain rule for derivatives, we get

$$\begin{aligned}
&\frac{d}{dz}\Pr(\mathsf{error}) \\
&= \frac{1}{2}\Pr(b_j = 0)\frac{d}{dz}\text{erfc}\left(\frac{s_0 - z}{\sqrt{2\sigma^2}}\right) + \frac{1}{2}\Pr(b_j = 1)\frac{d}{dz}\text{erfc}\left(\frac{z - s_1}{\sqrt{2\sigma^2}}\right) \qquad (3.7)\\
&= \frac{1}{2\sqrt{\sigma^2}}\Pr(b_j = 0) f_T\left(\frac{s_0 - z}{\sqrt{2\sigma^2}}\right) - \frac{1}{2\sqrt{\sigma^2}}\Pr(b_j = 1) f_T\left(\frac{z - s_1}{\sqrt{2\sigma^2}}\right). \qquad (3.8)
\end{aligned}$$

To find the minimum, we set the expression in (3.8) to zero. As a result, the minimizing value of z is the value satisfying

$$\Pr(b_j = 0) f_T\left(\frac{s_0 - z}{\sqrt{2\sigma^2}}\right) = \Pr(b_j = 1) f_T\left(\frac{z - s_1}{\sqrt{2\sigma^2}}\right). \tag{3.9}$$

It is left as an exercise for the reader to show that this value is a minimum. Substituting into (3.9) with the expansion of $f_T(t)$, and collecting exponential terms, we have

$$\exp\left(-\frac{1}{2\sigma^2}(s_0 - z)^2 + \frac{1}{2\sigma^2}(z - s_1)^2\right) = \frac{\Pr(b_j = 1)}{\Pr(b_j = 0)}. \tag{3.10}$$

Taking the natural logarithm, log, of both sides results in

$$\frac{1}{2\sigma^2}\left((z - s_1)^2 - (s_0 - z)^2\right) = \log\frac{\Pr(b_j = 1)}{\Pr(b_j = 0)}, \tag{3.11}$$

and collecting terms on the left, we have

$$\frac{1}{2\sigma^2}\left(2(s_0 - s_1)z - (s_0^2 - s_1^2)\right) = \log\frac{\Pr(b_j = 1)}{\Pr(b_j = 0)}. \tag{3.12}$$

Finally, solving for z gives

$$z = \frac{\sigma^2}{s_0 - s_1}\log\frac{\Pr(b_j = 1)}{\Pr(b_j = 0)} + \frac{1}{2}(s_0 + s_1). \tag{3.13}$$

Thus, in the polar NRZ example we presented above, $z = 0$ is indeed the optimal threshold, since $s_1 = -s_0$ and $\Pr(b_j = 0) = \Pr(b_j = 1) = 1/2$.

We make two remarks on (3.13). First, if the two binary values 0 and 1 are equiprobable (i.e., $\Pr(b_j = 0) = \Pr(b_j = 1) = 1/2$),then (3.13) reduces to

$$z = \frac{1}{2}(s_0 + s_1), \tag{3.14}$$

which is exactly halfway between the two mean values s_0 and s_1. Thus, for any received value $y[k]$, the decision $\hat{b}[k]$ is made based on the *closest point* to $y[k]$,

either s_0 or s_1. Second, if 0 and 1 are not equiprobable, then the threshold is biased towards the less likely bit, thereby expanding the region of $y[k]$ that maps to the more likely bit. This reduces error since, in case of uncertainty, it is safer to select the more likely bit. This is illustrated in Figure X.

EXAMPLE 3.1. *Suppose* $s_0 = 1$, $s_1 = -1$, *and* $\sigma^2 = 1$. *Let* $z = 1/2$. *For what values of* $\Pr(b_j = 0)$ *and* $\Pr(b_j = 1)$ *is this setting of* z *optimal?*

Substituting into (3.13), we have

$$z = \frac{1}{2} = \frac{\sigma^2}{s_0 - s_1} \log \frac{\Pr(b_j = 1)}{\Pr(b_j = 0)} + \frac{1}{2}(s_0 + s_1) \tag{3.15}$$

$$= \frac{1}{2} \log \frac{\Pr(b_j = 1)}{\Pr(b_j = 0)} + \frac{1}{2}(0). \tag{3.16}$$

Simplifying, we have

$$\log \frac{\Pr(b_j = 1)}{\Pr(b_j = 0)} = 1. \tag{3.17}$$

Thus, $\Pr(b_j = 1)/\Pr(b_j = 0) = e^1 = e \simeq 2.718$. *However, remember that* $\Pr(b_j = 1) + \Pr(b_j = 0) = 1$. *Thus,*

$$\frac{\Pr(b_j = 1)}{1 - \Pr(b_j = 1)} = e, \tag{3.18}$$

the solution for which is $\Pr(b_j = 1) = e/(1 + e) \simeq 0.731$, *so* $\Pr(b_j = 1) \simeq 0.269$.

In general, the optimal way to distinguish between two signals in noise is to employ the maximum *a posteriori* probability (MAP) criterion

3.2. Receiver filter design: The matched filter

We now consider how to design the optimal receiver filter $h[k]$. The following assumptions are used to simplify the derivation:

- The input bits are equiprobable: $\Pr(b_j = 0) = \Pr(b_j = 1) = 1/2$; and
- Modulation waveform $s_1[k]$ is a scalar multiple of $s_0[k]$; i.e., there exists α such that

$$s_1[k] = \alpha s_0[k]. \tag{3.19}$$

Note that (3.19) is true of all three modulation schemes we have studied thus far: in Polar NRZ and BPSK, we had $\alpha = -1$, while in on-off keying, we had $\alpha = 0$.

To simplify the notation, let

$$\hat{h}[k] = h[n_b - k]. \tag{3.20}$$

Using $\hat{h}[k]$, s_0 becomes

$$\begin{aligned} s_0 &= [s_0[k] \star h[k]]_{n_b} && (3.21)\\ &= \sum_{i=1}^{n_b} s_0[i]h[n_b - i] && (3.22)\\ &= \sum_{i=1}^{n_b} s_0[i]\hat{h}[i]. && (3.23) \end{aligned}$$

Furthermore, using (3.19), s_1 becomes

$$\begin{aligned} s_1 &= \sum_{i=1}^{n_b} s_1[i]\hat{h}[i] && (3.24)\\ &= \alpha \sum_{i=1}^{n_b} s_0[i]\hat{h}[i] && (3.25)\\ &= \alpha s_0. && (3.26) \end{aligned}$$

Furthermore, since $\hat{h}[k]$ rearranges the elements of $h[k]$, but does not change their values, it should be clear that the variance is now given by

$$\sigma^2 = \frac{N_0}{2T_s} \sum_i \hat{h}[i]^2. \tag{3.27}$$

By assumption, the bit values are equiprobable, so we use the optimal threshold from (3.14). This leads to

$$z = \frac{1}{2}(s_0 + s_1) = \frac{1+\alpha}{2} s_0. \tag{3.28}$$

Substituting into (3.1), we have

$$\begin{aligned} &\Pr(\mathsf{error}) \\ &= \frac{1}{2}\mathrm{erfc}\left(\frac{s_0 - z}{\sqrt{2\sigma^2}}\right)\Pr(b_j = 0) + \frac{1}{2}\mathrm{erfc}\left(\frac{z - s_1}{\sqrt{2\sigma^2}}\right)\Pr(b_j = 1) && (3.29)\\ &= \frac{1}{4}\mathrm{erfc}\left(\frac{s_0 - (1+\alpha)s_0/2}{\sqrt{2\sigma^2}}\right) + \frac{1}{4}\mathrm{erfc}\left(\frac{(1+\alpha)s_0/2 - \alpha s_0}{\sqrt{2\sigma^2}}\right) && (3.30)\\ &= \frac{1}{2}\mathrm{erfc}\left(\frac{(1-\alpha)s_0}{2\sqrt{2\sigma^2}}\right). && (3.31) \end{aligned}$$

The filter design problem can then be stated as follows: find $\hat{h}[k]$ satisfying

$$\min_{\hat{h}[k]} \frac{1}{2}\mathrm{erfc}\left(\frac{(1-\alpha)s_0}{2\sqrt{2\sigma^2}}\right). \tag{3.32}$$

However, note that erfc(x) is a *decreasing* function of x, so minimizing erfc is equivalent to maximizing its argument. Thus, (3.32) is equivalent to finding $\hat{h}[k]$ satisfying

$$\max_{\hat{h}[k]} \frac{(1-\alpha)s_0}{2\sqrt{2\sigma^2}}. \tag{3.33}$$

The constants do not affect the value of $\hat{h}[k]$ maximizing (3.33), and neither does squaring the expression, so (3.33) becomes

$$\max_{\hat{h}[k]} \frac{s_0^2}{\sigma^2}. \tag{3.34}$$

Substituting s_0 and σ^2 with their expansions, given by (3.23) and (3.27) respectively, the design problem becomes: find $\hat{h}[k]$ satisfying

$$\max_{\hat{h}[k]} \frac{\left(\sum_{i=1}^{n_b} s_0[i]\hat{h}[i]\right)^2}{\sum_{i=1}^{n_b} \hat{h}[i]^2}, \tag{3.35}$$

again eliminating the constants in the denominator. Remarkably, α is irrelevant to the maximization, so the filter only depends on $s_0[k]$.

To solve this problem, we use the *Cauchy-Schwartz inequality* [**3**]. There are many forms of this inequality, but the following form is most appropriate for this problem. Let $a[k]$ and $b[k]$ be discrete-time functions that are supported on $1 \leq k \leq n$. Then:

$$\left(\sum_{i=1}^{n} a[i]b[i]\right)^2 \leq \left(\sum_{i=1}^{n} a[i]^2\right)\left(\sum_{i=1}^{n} b[i]^2\right), \tag{3.36}$$

with equality *if and only if* $a[i] = b[i]$ for all i. This inequality is proved in Appendix B.

Returning to the design problem, we can apply the Cauchy-Schwartz inequality to $s_0[k]$ and $\hat{h}[k]$. Since $s_0[k]$ and $\hat{h}[k]$ are supported over $1 \leq k \leq n_b$, by substituting directly into (3.36), we can write

$$\left(\sum_{i=1}^{n_b} s_0[i]\hat{h}[i]\right)^2 \leq \left(\sum_{i=1}^{n_b} s_0[i]^2\right)\left(\sum_{i=1}^{n_b} \hat{h}[i]^2\right). \tag{3.37}$$

However, $s_0[k]$ is given, so $\sum_{i=1}^{n_b} s_0[i]^2$ is a constant with respect to $\hat{h}[i]$. Rearranging (3.37), we can write

$$\frac{\left(\sum_{i=1}^{n_b} s_0[i]\hat{h}[i]\right)^2}{\sum_{i=1}^{n_b} \hat{h}[i]^2} \leq K \tag{3.38}$$

where $K = \sum_{i=1}^{n_b} s_0[i]^2$, emphasizing that this quantity is constant. The quantity on the left of the inequality (3.38) is the same as the quantity to be maximized in (3.35). Thus, from (3.38), we conclude that the maximum possible value of this quantity is K, and by the equality condition for the Cauchy-Schwartz inequality, this value is achieved if and only if

$$\hat{h}[k] = s_0[k] \tag{3.39}$$

for all k. Letting $h^\star[k]$ represent the optimized filter, we have that

$$h^*[k] = s_0[n_b - k]. \tag{3.40}$$

The optimal filter $h^*[k]$ is called the *matched filter*, since from (3.40) it is clearly matched to $s_0[k]$.

Using $h^*[k]$, we can find the optimized values of s_0 and s_1 (which we write s_0^* and s_1^*, respectively), as follows:

$$s_0^* = \sum_{i=1}^{n_b} s_0[i]^2 \tag{3.41}$$

$$= \frac{E_0}{T_s}, \tag{3.42}$$

recalling the definition of E_0 as the energy required to send a zero. Similarly,

$$s_1^* = \alpha \frac{E_0}{T_s}. \tag{3.43}$$

Notice that $E_1 = \alpha^2 E_0$, so the average energy per bit, E_b, is given by

$$E_b = \frac{1}{2}(E_0 + E_1) \tag{3.44}$$

$$= \frac{1+\alpha^2}{2} E_0. \tag{3.45}$$

Furthermore, the optimized value of σ^2, written σ^{2^*}, is given by

$$\sigma^{2^*} = \frac{N_0}{2T_s} \sum_{i=1}^{n_b} s_0[i]^2 \tag{3.46}$$

$$= \frac{N_0}{2T_s^2} E_0. \tag{3.47}$$

Substituting all of the above into (3.31), we have

$$\Pr(\mathsf{error}) = \frac{1}{2}\mathrm{erfc}\left(\frac{(1-\alpha)s_0^*}{2\sqrt{2\sigma^{2*}}}\right) \tag{3.48}$$

$$= \frac{1}{2}\mathrm{erfc}\left(\sqrt{\frac{(1-\alpha)^2}{4}\frac{E_0}{N_0}}\right) \tag{3.49}$$

$$= \frac{1}{2}\mathrm{erfc}\left(\sqrt{\frac{(1-\alpha)^2}{2(1+\alpha^2)}\frac{E_b}{N_0}}\right). \tag{3.50}$$

Thus, under our design assumptions, the probability of error for the optimal filter can be expressed in terms of E_b/N_0. It is interesting to note that the individual values of E_b and N_0 are irrelevant – only their ratio matters. Thus, E_b/N_0 is often used as a *figure of merit* for digital communication systems.

Example 3.2.

3.3. Optimized waveform design

The parameter α relates $s_0[k]$ to $s_1[k]$. Since, from (3.50), the probability of error is a function of α, we may consider the value of α that minimizes the probability of error.

We need to find α satisfying

$$\min_\alpha \frac{1}{2}\mathrm{erfc}\left(\sqrt{\frac{(1-\alpha)^2}{2(1+\alpha^2)}\frac{E_b}{N_0}}\right). \tag{3.51}$$

Note that E_b/N_0 is independent of α, so taking the same apporach as we took leading up to (3.35): we need to find α maximizing

$$\max_\alpha \frac{(1-\alpha)^2}{1+\alpha^2}. \tag{3.52}$$

Taking the first derivative, we get

$$\frac{d}{d\alpha}\frac{(1-\alpha)^2}{1+\alpha^2} = \frac{-2(1-\alpha)(1+\alpha)}{(1+\alpha^2)^2}, \tag{3.53}$$

which has critical points at $\alpha = -1$ and $\alpha = +1$; it is straightforward to show that these are a maximum and a minimum, respectively. Thus, using the optimal signalling scheme $s_1[k] = -s_0[k]$, optimal filter $h[k]$, and optimal threshold z, the best possible probability of error is given by

$$\Pr(\mathsf{error}) = \frac{1}{2}\mathrm{erfc}\left(\sqrt{\frac{E_b}{N_0}}\right). \tag{3.54}$$

Intuitively, it makes sense that $\alpha = +1$ is a minimum, since in that case, $s_0[k] = s_1[k]$ – in other words, there is no difference between the signals used to transmit 0 and 1, so there is no way to tell them apart. On the other hand, for constant E_b, this result suggests that the best approach is to set $s_0[k] = -s_1[k]$, as we did in Polar NRZ and BPSK. Thus, on-off keying, in which $\alpha = 0$, is not an optimal signalling scheme. Furthermore, in (3.54), the details of $s_0[k]$ and $s_1[k]$ are not relevant – they only affect Pr(error) through E_b. Thus, any optimal signalling scheme with the same E_b should have the same error performance. (However, there are other criteria, such as bandwidth, that make some signalling schemes more useful than others; we will discuss these in later chapters.)

3.4. Summary

Optimal parameter selections derived in this chapter are given as follows:

- **Optimal threshold.** Given in (3.13). If 0 and 1 are equiprobable, the optimal threshold is $z = (s_0 + s_1)/2$.
- **Optimal filter.** The *matched filter* is optimal, with $h^*[k] = s_0[n_b - k]$.
- **Signal selection.** Given $s_0[k]$, set $s_1[k] = -s_0[k]$. Every such setting of $s_0[k]$ and $s_1[k]$ with the same E_b has the same Pr(error).

Unless otherwise noted, these optimal settings will be used throughout the rest of the book.

3.5. Problems

(1) Show that the value of z satisfying (3.9) is a minimum of Pr(error).
(2) For the three modulation schemes introduced in Chapter 2, demonstrate that the optimal threshold and matched filter were correctly chosen in each example.

3.6. Laboratory exercise

CHAPTER 4

Signal Space and Nonbinary Data Transmission

In Chapters 2 and 3, we introduced the basic elements of the digital communcations problem, including modulation, detection, and optimized receiver design. However, our analysis in those chapters was binary and one-dimensional: we could only transmit one bit at a time. Furthermore, we restricted ourselves to the case where $s_1[k]$ was a scalar multiple of $s_0[k]$, which is not necessarily optimal when nonbinary signals are transmitted. In this chapter, we introduce *signal space*, which provides a mathematical framework for nonbinary and multi-dimensional modulation schemes. Furthermore, we introduce limitations on bandwidth, and discuss their importance on signal design.

4.1. Introduction to Signal Space

4.1.1. Vector spaces. Chapter 2 established that the modulation functions $s_0[k]$ and $s_1[k]$ are discrete functions supported on $1 \leq k \leq n_b$. These functions can instead be represented as $1 \times k$ row vectors $\mathbf{s}_0$ and $\mathbf{s}_1$, respectively, where

$$\mathbf{s}_0 = [s_0[1], s_0[2], \ldots, s_0[n_b]], \tag{4.1}$$

and

$$\mathbf{s}_1 = [s_1[1], s_2[1], \ldots, s_2[n_b]]. \tag{4.2}$$

In Chapter 3, we assumed that $\mathbf{s}_1$ was a scalar multiple of $\mathbf{s}_0$; let us now relax that assumption. Recall the definition of vector dot product: if $\mathbf{a} = [a_1, a_2, \ldots, a_n]$ and $\mathbf{b} = [b_1, b_2, \ldots, b_n]$ are $1 \times n$ vectors, then

$$\mathbf{a} \cdot \mathbf{b} = \sum_{i=1}^{n} a_i b_i. \tag{4.3}$$

Now suppose there exists a vector $\bar{\mathbf{s}}_0$ and constants α and β such that

$$\mathbf{s}_1 = \alpha \mathbf{s}_0 + \beta \bar{\mathbf{s}}_0, \tag{4.4}$$

where $\bar{\mathbf{s}}_0 \cdot \mathbf{s}_0 = 0$. That is, $\bar{\mathbf{s}}_0$ is *orthogonal* to $\mathbf{s}_0$.

The pair $\mathbf{s}_0$, $\bar{\mathbf{s}}_0$ thus form a two-dimensional *vector space* containing both $\mathbf{s}_0$ and $\mathbf{s}_1$. Vector spaces consist of a set of *basis vectors*, where any vector in the space can be composed of a linear combination of the basis vectors. Furthermore, vector spaces are *closed*, in the sense that any linear combination of vectors in the vector space is also in the vector space. This is illustrated in the following example, using the well-known Cartesian space.

EXAMPLE 4.1 (Cartesian vector space). *Let* $\mathbf{x} = [1, 0]$ *and* $\mathbf{y} = [0, 1]$ *be* 1×2 *basis vectors. Firstly, note that* $\mathbf{x}$ *and* $\mathbf{y}$ *are orthogonal:*

$$\mathbf{x} \cdot \mathbf{y} = x_1 y_1 + x_2 y_2 \tag{4.5}$$

$$= 1 \cdot 0 + 0 \cdot 1 \tag{4.6}$$

$$= 0. \tag{4.7}$$

Clearly, any two-dimensional vector can be expressed as

$$[\alpha, \beta] = \alpha\mathbf{x} + \beta\mathbf{y}, \tag{4.8}$$

and therefore all such vectors are in the two-dimensional Cartesian vector space. Furthermore, from (4.8), the summation of any pair of vectors in this space is also in the space. To see this, we can write

$$[\alpha_1, \beta_1] + [\alpha_2, \beta_2] = \alpha_1\mathbf{x} + \beta_1\mathbf{y} + \alpha_2\mathbf{x} + \beta_2\mathbf{y} \tag{4.9}$$

$$= (\alpha_1 + \alpha_2)\mathbf{x} + (\beta_1 + \beta_2)\mathbf{y} \tag{4.10}$$

$$= [\alpha_1 + \alpha_2, \beta_1 + \beta_2]. \tag{4.11}$$

The same arguments apply to any vector space with orthogonal basis vectors.

The *norm* of a $1 \times n$ vector $\mathbf{a}$ is given by

$$|\mathbf{a}| = \sqrt{\mathbf{a} \cdot \mathbf{a}} = \sqrt{\sum_{i=1}^{n} a_i^2}. \tag{4.12}$$

From the above example, the vectors $\mathbf{x}$ and $\mathbf{y}$ have the additional useful property that $|\mathbf{x}| = |\mathbf{y}| = 1$. A basis for a vector space for which the basis vectors are all orthogonal to each other, and all have unit norm, is called an *orthonormal basis*. Given any m linearly independent vectors (i.e., none of the m vectors can

be expressed as a linear combination of the others), it always possible to generate an m-dimensional orthonormal basis using the *Gram-Schmidt procedure*. Here we give this procedure for $m = 2$, which is the largest case that we will require in this book. Let $\mathbf{a}$ and $\mathbf{b}$ represent the two (linearly independent) vectors, and suppose they are both $1 \times n$:

(1) Normalize $\mathbf{a}$: let

$$\hat{\mathbf{a}} = \frac{\mathbf{a}}{|\mathbf{a}|}. \tag{4.13}$$

Using (4.12), it is easy to show that $|\hat{\mathbf{a}}| = 1$, so $\hat{\mathbf{a}}$ is the first basis vector.

(2) Remove the component of $\mathbf{b}$ in the direction of $\hat{\mathbf{a}}$: let

$$\mathbf{b}' = \mathbf{b} - \hat{\mathbf{a}}(\mathbf{b} \cdot \hat{\mathbf{a}}). \tag{4.14}$$

Using (4.3), it is easy to show that $\mathbf{b}'$ is orthogonal to $\hat{\mathbf{a}}$, i.e., $\mathbf{b}' \cdot \hat{\mathbf{a}} = 0$.

(3) Normalize $\mathbf{b}'$: let

$$\hat{\mathbf{b}} = \frac{\mathbf{b}'}{|\mathbf{b}'|}. \tag{4.15}$$

Again using (4.12), it is easy to show that $|\hat{\mathbf{b}}| = 1$, and $\hat{\mathbf{b}}$ is still orthogonal to $\hat{\mathbf{a}}$, so $\hat{\mathbf{b}}$ is the second basis vector.

(4) The orthonormal basis is finally given by the pair of vectors $\hat{\mathbf{a}}$ and $\hat{\mathbf{b}}$.

We now verify that $\mathbf{a}$ and $\mathbf{b}$ can be represented in terms of this basis: from (4.13), $\mathbf{a}$ is given by

$$\mathbf{a} = |\mathbf{a}|\hat{\mathbf{a}}, \tag{4.16}$$

(with a coefficient of zero in the $\hat{\mathbf{b}}$ direction), and from (4.14)-(4.15), $\mathbf{b}$ is given by

$$\mathbf{b} = (\mathbf{b} \cdot \hat{\mathbf{a}})\hat{\mathbf{a}} + |\mathbf{b}'|\hat{\mathbf{b}}. \tag{4.17}$$

Thus, $\mathbf{a}$ and $\mathbf{b}$ are in the vector space formed by $\hat{\mathbf{a}}$ and $\hat{\mathbf{b}}$.

Since we will be treating signals (like $s_0[k]$) as vectors throughout this chapter, vector notation may be applied to a signal as well as a vector (for example, $|s_0[k]|$ will signify the norm of $s_0[k]$).

4.2. Signal Space in Digital Communication

4.2.1. Introduction to M-ary digital communication. In introducing signal space, we are mostly interested in discussing schemes where *more than one bit is transmitted at once*. Whereas in *binary* communication, only one of two symbols can be used (i.e., 0 and 1), in *M-ary* communication, any one of M symbols can be transmitted. Without loss of generality, we will call these symbols $\{0, 1, 2, \ldots, M-1\}$.

One simple way to think about M-ary communication is to start with a binary data stream, and group adjacent bits. Consider the following example:

EXAMPLE 4.2. *Suppose we wish to transmit the binary vector*

$$\mathbf{b} = [0, 1, 0, 0, 1, 1, 1, 0, 1, 0, 1, 1]. \tag{4.18}$$

Suppose this vector is to be transmitted as 4-ary data, i.e., using symbols $\{0, 1, 2, 3\}$. *We can group the bits in* $\mathbf{b}$ *into pairs, and map them into elements of* $\{0, 1, 2, 3\}$ *using the usual mapping, as follows:*

$$\mathbf{b} = [(0, 1), (0, 0), (1, 1), (1, 0), (1, 0), (1, 1)] \tag{4.19}$$

$$= [1, 0, 3, 2, 2, 3]. \tag{4.20}$$

Similarly, grouping the bits in $\mathbf{b}$ *into triplets, we can transmit the vector as 8-ary data using symbols* $\{0, 1, 2, 3, 4, 5, 6, 7\}$, *as follows:*

$$\mathbf{b} = [(0, 1, 0), (0, 1, 1), (1, 0, 1), (0, 1, 1)] \tag{4.21}$$

$$= [2, 3, 5, 3]. \tag{4.22}$$

However, M-ary transmission need not have a direct binary translation, i.e., M need not be a power of 2. Consider the following example.

EXAMPLE 4.3. *Suppose we want to encode the 26 letters of the English alphabet, as well as the space character, for a total of 27 encoding symbols. We will assign these to numbers as follows: (space)=0, A=1, B=2, ..., Z=26.*

Suppose that this data is to be transmitted as 3-ary data. Note that $27 = 3^3$, *so each symbol can be represented by exactly three 3-ary symbols.*

Suppose we encode the phrase "BROWN FOX". This can be represented as the vector

$$\mathbf{b} = [2, 18, 15, 23, 14, 0, 6, 15, 24]. \tag{4.23}$$

Representing each number $\{0, 1, \ldots, 26\}$ *as a triplet of* $\{0, 1, 2\}$*, we have*

(4.24)

$$\mathbf{b} = [(0,0,2), (2,0,0), (1,2,0), (2,1,2), (1,1,2), (0,0,0), (0,2,0), (1,2,0), (2,2,0)].$$

In any M-ary system, the number of *bits per symbol* is given by

$$\beta = \log_2 M. \tag{4.25}$$

This quantity is frequently used to allow two systems to be compared on a per-bit basis, rather than a per-symbol basis (since bits are the fundamental unit of digital information, and symbols may be used to convey different numbers of bits). For example, in the 3-ary system from Example 4.3, each symbol conveys $\log_2 3 = 1.585$ bits.

In the remainder of this section, we will use the ideas of signal space to design M-ary communication systems. Extending our notation from Chapter 2, for each $i \in \{0, 1, \ldots, M-1\}$, we will write $s_i[k]$ to represent the modulation signal corresponding to the ith symbol.

4.2.2. One-dimensional signal space. In a one-dimensional signal space, there is only one vector in the basis, and all valid signals are linear multiples of that basis vector. We have implicitly used one-dimensional signal space in the previous chapters of this book – e.g., see (3.19) in Chapter 3.

Let $s[k]$ represent the single basis vector in the space, where $|s[k]| = 1$. For each $i \in \{0, 1, \ldots, M-1\}$, assign a coefficient α_i, such that

$$s_i[k] = \alpha_i s[k]. \tag{4.26}$$

Without loss of generality, suppose the coefficients α_i are arranged so that

$$\alpha_0 \geq \alpha_1 \geq \ldots \geq \alpha_{M-1}. \tag{4.27}$$

We now consider the probability of error in M-ary transmission schemes. To do so, we will follow a similar approach to the one we saw in Chapter 2: namely, we will obtain the noise-free filter outputs and the noise variance, and we will use

these to calculate the probability of error given that each symbol $\{0, 1, \ldots, M-1\}$ was sent.

We start with the filter outputs. Suppose the signal is passed through a detection filter $h[k]$. Let s_i, $i \in \{0, 1, \ldots, M-1\}$, represent the noise-free filter outputs; then

$$s_i = [s_i[k] \star h[k]]_{n_b} \tag{4.28}$$

$$= [\alpha_i s[k] \star h[k]]_{n_b} \tag{4.29}$$

$$= \alpha_i [s[k] \star h[k]]_{n_b}. \tag{4.30}$$

Letting $H = [s[k] \star h[k]]_{n_b}$, we then have $s_i = H\alpha_i$. Assuming that $H > 0$, we then have that

$$s_0 \geq s_1 \geq \ldots \geq s_{M-1}. \tag{4.31}$$

This one-dimensional arrangement is depicted in Figure X.

The noise variance is dependent only on the detection filter $h[k]$, and not on the transmitted signals. As a result, we can still use (2.34) from Chapter 2.

In order to calculate probability of error, we need to specify the decision thresholds on the filter output. From (4.31), the noise-free filter outputs are arranged along the real number line. Suppose symbol j, $0 < j < M-1$ is transmitted, so that the noise-free filter output is s_j. There are noise-free filter outputs both less than s_j (e.g., s_{j+1}) and greater than s_j (e.g., s_{j-1}). Thus, a single threshold, like in the binary case, is inadequate: we need to distinguish j from *both* $j+1$ and $j-1$. Instead, we will use a vector of thresholds $[z_1, z_2, \ldots, z_{M-1}]$ to distinguish among the M signals.

We will require that

$$z_1 > z_2 > \ldots > z_{M-1}. \tag{4.32}$$

Thus, given filter outputs $\phi[j]$, the decision function $d(\cdot)$ has the form

$$d(\phi[k]) = \begin{cases} 0, & \phi[k] > z_1 \\ 1, & z_1 \geq \phi[k] > z_2 \\ \vdots & \vdots \\ j, & z_j \geq \phi[k] > z_{j+1} \\ \vdots & \vdots \\ M-1, & z_{M-1} \geq \phi[k]. \end{cases} \tag{4.33}$$

Note that this rule is a generalization of the binary case: if $M = 2$, then we have two symbols $\{0, 1\}$, and we only use one threshold z_1, deciding 0 if $\phi[k] > 0$ and deciding 1 if $\phi[k] \leq 0$. These thresholds are also depicted on Figure X.

From (4.33), we decide j if $\phi[k]$ is on the interval $[z_j, z_{j+1})$ (except for the two end points, 0 and $M-1$). We can now calculate the probability of error given that j was sent. (For now, we ignore the possibility that $j = 0$ or $j = M-1$.) An error occurs if j was sent *and* $\phi[k]$ is not on the interval $[z_j, z_{j+1})$. Furthermore, since the noise is not affected by the transmitted signal, the noise is Gaussian with mean zero and variance σ^2, like in Chapter 2; thus, $\phi[k]$ is a Gaussian random variable with mean s_j and variance σ^2. Thus, if b represents the transmitted symbol,

$$\begin{aligned} \Pr(\mathsf{error}|b=j) &= \Pr(\phi[k] > z_j | b = j) + \Pr(\phi[k] \leq z_{j+1} | b = j) \\ &= \int_{z_j}^{\infty} \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{1}{2\sigma^2}(\phi[k] - s_j)^2\right) d\phi[k] \end{aligned} \tag{4.34}$$

$$+ \int_{-\infty}^{z_{j+1}} \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{1}{2\sigma^2}(\phi[k] - s_j)^2\right) d\phi[k]. \tag{4.35}$$

In Chapter 1, we showed how to replace each of the Gaussian inegrals in (4.35) with the erfc$(\cdot)$ function. Using this transformation, we have

$$\Pr(\mathsf{error}|b=j) = \frac{1}{2}\mathrm{erfc}\left(\frac{z_j - s_j}{\sqrt{2\sigma^2}}\right) + \frac{1}{2}\mathrm{erfc}\left(\frac{s_j - z_{j+1}}{\sqrt{2\sigma^2}}\right). \tag{4.36}$$

Now let's reconsider the case of $b = 0$. Because anything greater than z_1 is mapped into 0, an error occurs if $\phi[k] < z_1$. This is equivalent to the probability of error given $b = 0$ in the binary case, resulting in

$$\Pr(\mathsf{error}|b=0) = \frac{1}{2}\mathrm{erfc}\left(\frac{s_0 - z_1}{\sqrt{2\sigma^2}}\right). \tag{4.37}$$

Similarly, if $b = M-1$, an error occurs if $\phi[k] \geq z_{M-1}$, which is analogous to the binary case where $b = 1$. Thus, we have that

$$\Pr(\mathsf{error}|b = M-1) = \frac{1}{2}\mathrm{erfc}\left(\frac{z_{M-1} - s_{M-1}}{\sqrt{2\sigma^2}}\right). \tag{4.38}$$

Finally, we can write

$$\begin{aligned}
&\Pr(\mathsf{error}) \\
&= \sum_{j=0}^{M-1} \Pr(b=j)\Pr(\mathsf{error}|b=j) \qquad (4.39)\\
&= \Pr(b=0)\frac{1}{2}\mathrm{erfc}\left(\frac{s_0 - z_1}{\sqrt{2\sigma^2}}\right) \\
&\quad + \Pr(b=M-1)\frac{1}{2}\mathrm{erfc}\left(\frac{z_{M-1} - s_{M-1}}{\sqrt{2\sigma^2}}\right) \\
&\quad + \sum_{j=1}^{M-2} \Pr(b=j)\left[\frac{1}{2}\mathrm{erfc}\left(\frac{z_j - s_j}{\sqrt{2\sigma^2}}\right) + \frac{1}{2}\mathrm{erfc}\left(\frac{s_j - z_{j+1}}{\sqrt{2\sigma^2}}\right)\right]. \qquad (4.40)
\end{aligned}$$

From now on, we will assume that the input symbols are equiprobable, so

$$\Pr(b=0) = \Pr(b=1) = \ldots = \Pr(b=M-1) = \frac{1}{M}. \tag{4.41}$$

It is natural to ask how to select the optimal thresholds $z_1, z_2, \ldots, z_{M-1}$. Recall in the binary case that if 0 and 1 were equiprobable, then the optimal threshold z was $z = (s_0 + s_1)/2$, i.e., halfway between s_0 and s_1. Thus, assuming that the input symbols are equiprobable, it is intuitive that the thresholds should be halfway between the noise-free filter outputs (i.e., z_1 is halfway between s_0 and s_1, z_1 is halfway between z_1 and z_2, and so on). Using the techniques in Chapter 3, it can be shown that this intuition is correct. The decision boundaries then become

$$z_i = \frac{1}{2}(s_{i-1} + s_i), \tag{4.42}$$

and (4.40) becomes

$$\begin{aligned}
&\Pr(\mathsf{error}) \\
&= \frac{1}{2M}\mathrm{erfc}\left(\frac{s_0 - s_1}{2\sqrt{2\sigma^2}}\right) \\
&\quad + \frac{1}{2M}\mathrm{erfc}\left(\frac{s_{M-2} - s_{M-1}}{2\sqrt{2\sigma^2}}\right) \\
&\quad + \frac{1}{2M}\sum_{j=1}^{M-2}\left[\mathrm{erfc}\left(\frac{s_{j-1} - s_j}{2\sqrt{2\sigma^2}}\right) + \mathrm{erfc}\left(\frac{s_j - s_{j+1}}{2\sqrt{2\sigma^2}}\right)\right]. \qquad (4.43)
\end{aligned}$$

EXAMPLE 4.4. *3-ary example ...*

Finally, suppose $s_i - s_{i+1}$ is equal for all i, $0 \leq i < M-1$, and let

$$D = s_i - s_{i+1}. \tag{4.44}$$

That is, all the noise-free filter outputs are the same distance D apart. In this case, (4.43) becomes

$$\begin{aligned}
&\Pr(\mathsf{error}) \\
&= \frac{1}{2M}\mathrm{erfc}\left(\frac{s_0 - s_1}{2\sqrt{2\sigma^2}}\right) \\
&\quad + \frac{1}{2M}\mathrm{erfc}\left(\frac{s_{M-2} - s_{M-1}}{2\sqrt{2\sigma^2}}\right) \\
&\quad + \frac{1}{2M}\sum_{j=1}^{M-2}\left[\mathrm{erfc}\left(\frac{s_{j-1} - s_j}{2\sqrt{2\sigma^2}}\right) + \mathrm{erfc}\left(\frac{s_j - s_{j+1}}{2\sqrt{2\sigma^2}}\right)\right] && (4.45) \\
&= \frac{1}{2M}\mathrm{erfc}\left(\frac{D}{2\sqrt{2\sigma^2}}\right) + \frac{1}{2M}\mathrm{erfc}\left(\frac{D}{2\sqrt{2\sigma^2}}\right) \\
&\quad + \frac{1}{2M}\sum_{j=1}^{M-2}\left[\mathrm{erfc}\left(\frac{D}{2\sqrt{2\sigma^2}}\right) + \mathrm{erfc}\left(\frac{D}{2\sqrt{2\sigma^2}}\right)\right] && (4.46) \\
&= 2\frac{1}{2M}\mathrm{erfc}\left(\frac{D}{2\sqrt{2\sigma^2}}\right) + \frac{1}{2M}2(M-2)\mathrm{erfc}\left(\frac{D}{2\sqrt{2\sigma^2}}\right) && (4.47) \\
&= \frac{M-1}{M}\mathrm{erfc}\left(\frac{D}{2\sqrt{2\sigma^2}}\right), && (4.48)
\end{aligned}$$

where (4.46) follows from the definition of D, and (4.47)-(4.48) follow from collecting identical terms.

In the following example, we show that the binary ($M = 2$) probability of error function emerges directly from this analysis:

EXAMPLE 4.5. *Suppose $M = 2$. Then since the only two noise-free filter outputs are s_0 and s_1, there is only one interval between them, so "all" intervals are identical. Further, there is only one threshold; from (4.42), this is given by $z_1 = (s_0 + s_1)/2$, halfway between the two noise-free filter outputs. Under these circumstances, if $s_1 = \alpha s_0$, the probability of error is given by (3.31) in Chapter 3. However, by substituting these values into (4.48), we get the same result, namely*

$$\frac{1}{2}\mathrm{erfc}\left(\frac{s_0 - \alpha s_0}{2\sqrt{2\sigma^2}}\right). \tag{4.49}$$

Thus, the expression in (4.48) generalizes the optimal binary probability of error equation.

It is worth noting that $\Pr(\mathsf{error})$ is now given in terms of probability of error *per symbol* rather than probability of error *per bit*. In cases where the M-ary scheme is created by joining binary symbols (like 4-ary and 8-ary in Example 4.2), the probability of error per bit is dependent on the mapping from binary to M-ary, and is not available in closed form. However, in general, there are $\log_2 M$ bits per M-ary symbol. If a symbol error occurs, the best case scenario is that only one of these bits is different, and the worst case scenario is that *all* of the bits are different. Thus, if $\Pr_{\text{bit}}(\mathsf{error})$ represents the probability of error per bit, it must be true that

$$\frac{\Pr(\mathsf{error})}{\log_2 M} \leq \Pr_{\text{bit}}(\mathsf{error}) \leq \Pr(\mathsf{error}). \tag{4.50}$$

Evaluation of probability of error per bit is covered in this chapter's laboratory exercises.

Finally, to determine the optimal filter $h[k]$, we return to the notion of signal space. Since (4.48) has the same form as the binary probability of error function, it should be clear that the matched filter $h[k] = s[n_b - k]$ is optimal. In fact this is correct and can be shown using the same arguments that were used in Chapter 3. However, we can now show that this filter is optimal using signal space arguments. Recall the definition of $\hat{h}[k]$ from Chapter 3, and let $\hat{\mathbf{h}}$ represent the vector

$$\hat{\mathbf{h}} = [\hat{h}[1], \hat{h}[2], \ldots, \hat{h}[n_b]]. \tag{4.51}$$

Then variance is given by

$$\sigma^2 = \frac{N_0}{2T_s} \sum_i \hat{h}[i]^2 \tag{4.52}$$

$$= \frac{N_0}{2T_s} |\hat{\mathbf{h}}|^2. \tag{4.53}$$

Thus, if the norm $|\hat{\mathbf{h}}|$ is fixed, the variance σ^2 is constant. We will therefore fix $|\hat{\mathbf{h}}| = 1$, so that $\hat{\mathbf{h}}$ is a normalized vector.

From (4.48), with constant σ^2, the problem is to maximize $D = s_i - s_{i-1}$, which is constant for all i. Remember that

$$s_i = \alpha_i \sum_{i=1}^{n_b} s[k]\hat{h}[k] \tag{4.54}$$

$$= \alpha_i \mathbf{s} \cdot \hat{\mathbf{h}}. \tag{4.55}$$

In Figure X, we give a geometric interpretation of the vectors $\mathbf{s}_0$ and $\hat{\mathbf{h}}$. Since $\hat{\mathbf{h}}$ is normalized, the dot product $\mathbf{s} \cdot \hat{\mathbf{h}}$ is the projection of $\mathbf{s}$ onto $\hat{\mathbf{h}}$. Clearly, from Figure X, the distance D is maximized if the projection has maximum length, and this occurs if $\hat{\mathbf{h}}$ is parallel to $\mathbf{s}$. (In fact, this is an intuitive explanation of the Cauchy-Schwartz inequality.) Thus, we choose

$$\hat{\mathbf{h}} = \frac{\mathbf{s}}{|\mathbf{s}|} \tag{4.56}$$

$$= \mathbf{s}, \tag{4.57}$$

since we defined $\mathbf{s}$ with unit norm. Since $\mathbf{s} = \hat{\mathbf{h}}$, we have that

$$\mathbf{s} \cdot \hat{\mathbf{h}} = |\mathbf{s}|^2 = 1, \tag{4.58}$$

and thus $s_i = \alpha_i$. In other words, the noise-free filter output is *the component of the signal in the* $\mathbf{s}$ *direction.* Furthermore, $D = \alpha_i - \alpha_{i-1}$ for any i.

4.2.3. Two-dimensional signal space. Suppose we now have two basis signals, $\mathbf{s}_x$ and $\mathbf{s}_y$, with the following properties:

- $|\mathbf{s}_x| = |\mathbf{s}_y| = 1$, and
- $\mathbf{s}_x \cdot \mathbf{s}_y = 0$.

In other words, $\mathbf{s}_x$ and $\mathbf{s}_y$ form a two-dimensional *orthonormal basis.* Now suppose any signal in our M-ary scheme has a component in both the $\mathbf{s}_x$ and the $\mathbf{s}_y$ direction, i.e.,

$$\mathbf{s}_i = \alpha_i \mathbf{s}_x + \beta_i \mathbf{s}_y. \tag{4.59}$$

In the one-dimensional case, we used a matched filter to extract the component of $\mathbf{s}_i$ in the direction of the basis signal $\mathbf{s}$. That is now impossible: we have two basis signals that are orthogonal to each other, so a single matched filter cannot possibly be parallel with both. Instead, we use two matched filters: one aligned with $\mathbf{s}_x$, and one aligned with $\mathbf{s}_y$. We therefore have an *ordered pair* of filter

outputs. Thus, the decision device needs to map *regions* of the two-dimensional space into elements from $\{0, 1, \ldots, M-1\}$; furthermore, the thresholds become one-dimensional boundaries between the regions. This is illustrated in Figure X.

Analogously to the one-dimensional case, the filter outputs, ϕ_x and ϕ_y, are given by

$$\phi_x = \mathbf{y} \cdot \mathbf{s}_x \tag{4.60}$$

$$\phi_y = \mathbf{y} \cdot \mathbf{s}_y, \tag{4.61}$$

where $\mathbf{y}$ represents the vector of samples observed from the channel. The noise-free filter output for symbol i, $0 \leq i \leq M-1$, are then given by the pair $(s_{i,x}, s_{i,y})$, where

$$s_{i,x} = (\alpha_i \mathbf{s}_x + \beta_i \mathbf{s}_y) \cdot \mathbf{s}_x \tag{4.62}$$

$$= \alpha_i (\mathbf{s}_x \cdot \mathbf{s}_x) + \beta_i (\mathbf{s}_y \cdot \mathbf{s}_x) \tag{4.63}$$

$$= \alpha_i, \tag{4.64}$$

bearing in mind that $\mathbf{s}_x \cdot \mathbf{s}_x = |\mathbf{s}_x|^2 = 1$, and $\mathbf{s}_y \cdot \mathbf{s}_x = 0$, both by assumption; similarly,

$$s_{i,y} = (\alpha_i \mathbf{s}_x + \beta_i \mathbf{s}_y) \cdot \mathbf{s}_y = \beta_i. \tag{4.65}$$

Thus, the ith noise-free filter output is

$$s_i = (\alpha_i, \beta_i), \tag{4.66}$$

i.e., the coordinates of the signal in signal space.

EXAMPLE 4.6. *Suppose*

$$\mathbf{s}_x = \left[+\frac{1}{2}, +\frac{1}{2}, +\frac{1}{2}, +\frac{1}{2} \right], \tag{4.67}$$

and

$$\mathbf{s}_y = \left[+\frac{1}{2}, +\frac{1}{2}, -\frac{1}{2}, -\frac{1}{2} \right]. \tag{4.68}$$

It is straightforward to show that

Again, we use the "nearest neighbor" principle to determine the locations of the decision boundaries. In other words, for any pair of filter outputs (ϕ_x, ϕ_y), we calculate the distance to each noise-free filter output, given by

$$|(\phi_x - \alpha_i, \phi_y - \beta_i)|. \tag{4.69}$$

We then decide that the symbol is equal to the *closest signal* in signal space to the received values (ϕ_x, ϕ_y), i.e., the decision funciton $d(\phi_x, \phi_y)$ is given by

$$d(\phi_x, \phi_y) = \arg\min_i |(\phi_x - \alpha_i, \phi_y - \beta_i)|. \tag{4.70}$$

Thus, decision boundaries are always located halfway between each signal point (α_i, β_i), and its nearest neighbors in signal space. This is also depicted in Figure X. The regions in space closest to each signal point are called the *Voronoi regions*.

To calculate the probability of error, consider a signal point (α_i, β_i) with Voronoi region $\mathcal{V}$. Given that symbol i was sent, then clearly the probability of a *correct* decision is

$$\int\int_{\mathcal{V}} f(\phi_x, \phi_y|i) d\phi_x d\phi_y. \tag{4.71}$$

This is analogous to the one-dimensional case, but here the integral is over a two-dimensional region in space, rather than a one-dimensional interval. Now, let $\bar{\mathcal{V}}$ represent the complement of the region $\mathcal{V}$ (i.e., everything outside $\mathcal{V}$). From (4.71), the probability of an *error* given i is

$$\Pr(\mathsf{error}|b_k = i) = \int\int_{\bar{\mathcal{V}}} f(\phi_x, \phi_y|i) d\phi_x d\phi_y \tag{4.72}$$

$$= 1 - \int\int_{\mathcal{V}} f(\phi_x, \phi_y|i) d\phi_x d\phi_y, \tag{4.73}$$

where the second line follows from the fact that total probability must be 1. (We will find it most convenient to write the probability in the form of the second line.)

Consider $f(\phi_x, \phi_y|i)$. If ϕ_x and ϕ_y are independent, then we can split this term into

$$f(\phi_x, \phi_y|i) = f(\phi_x|i) f(\phi_y|i). \tag{4.74}$$

Since ϕ_x and ϕ_y are both matched filter outputs, it should be clear that ϕ_x is a Gaussian random variable with mean α_i and variance $N_0/2T_s$, and ϕ_y is also a Gaussian random variable with mean β_i and variance $N_0/2T_s$. In fact, since

$\mathbf{s}_x \cdot \mathbf{s}_y = 0$, it can be shown that ϕ_x and ϕ_y are independent; the reader is asked to prove this in the problems. We now have

$$\Pr(\mathsf{error}|b_k = i) = 1 - \int\int_{\mathcal{V}} f(\phi_x|i) f(\phi_y|i) d\phi_x d\phi_y. \tag{4.75}$$

Now suppose that the region $\mathcal{V}$ is rectangular. That is, $\mathcal{V}$ consists of all points (x, y) satisfying $x_{\min} \leq x \leq x_{\max}$ and $y_{\min} \leq y \leq y_{\max}$ (we will allow the boundaries to be infinite). Then we can write

$$\begin{aligned} &\Pr(\mathsf{error}|b_k = i) \\ &= 1 - \int_{\phi_y=y_{\min}}^{y_{\max}} \int_{\phi_x=x_{\min}}^{x_{\max}} f(\phi_x|i) f(\phi_y|i) d\phi_x d\phi_y \qquad (4.76) \\ &= 1 - \left(\int_{\phi_x=x_{\min}}^{x_{\max}} f(\phi_x|i) d\phi_x \right) \left(\int_{\phi_y=y_{\min}}^{y_{\max}} f(\phi_y|i) d\phi_y \right). \qquad (4.77) \end{aligned}$$

(If the region $\mathcal{V}$ is not rectangular, then we cannot in general perform this separation, so the calculation is still possible but much more complicated.) Each of the two separated terms is the probability of a Gaussian random variable landing on an interval. Thus, we can use our results from the one-dimensional case, it is easy to show that

$$\int_{\phi_x=x_{\min}}^{x_{\max}} f(\phi_x|i) d\phi_x = 1 - \left(\frac{1}{2}\mathrm{erfc}\left(\frac{\alpha_i - x_{\min}}{\sqrt{N_0/T_s}} \right) + \frac{1}{2}\mathrm{erfc}\left(\frac{x_{\max} - \alpha_i}{\sqrt{N_0/T_s}} \right) \right), \tag{4.78}$$

and similarly

$$\int_{\phi_y=y_{\min}}^{y_{\max}} f(\phi_y|i) d\phi_y = 1 - \left(\frac{1}{2}\mathrm{erfc}\left(\frac{\beta_i - y_{\min}}{\sqrt{N_0/T_s}} \right) + \frac{1}{2}\mathrm{erfc}\left(\frac{y_{\max} - \beta_i}{\sqrt{N_0/T_s}} \right) \right). \tag{4.79}$$

4.2.4. Energy per bit. In a one-dimensional signal space, energy per symbol can be calculated in the same way we described in Chapter 2. The energy per symbol, E_s, is given by

$$\begin{aligned} E_s &= \sum_{i=0}^{M-1} \Pr(b_k = i) E_i \qquad (4.80) \\ &= \sum_{i=0}^{M-1} \Pr(b_k = i) \alpha_i^2 |\mathbf{s}|^2 T_s \qquad (4.81) \\ &= \frac{T_s}{M} \sum_{i=0}^{M-1} \alpha_i^2, \qquad (4.82) \end{aligned}$$

since all the symbols are equiprobable by assumption. Furthermore, the number of bits per symbol is $\log_2 M$, we have that

$$E_b = \frac{E_s}{\log_2 M} \tag{4.83}$$

$$= \frac{T_s}{M \log 2M} \sum_{i=0}^{M-1} \alpha_i^2. \tag{4.84}$$

For the same probability of symbol error, one-dimensional M-ary transmission generally requires far more energy per bit, as illustrated in the following example.

EXAMPLE 4.7. *... 1-d energy per bit ...*

We now consider the calculation of energy per bit for signals in a two-dimensional signal space. Generally, the energy in a signal $\mathbf{s}$ is given by

$$E(\mathbf{s}) = T_s|\mathbf{s}|^2 \tag{4.85}$$

$$= T_s(\mathbf{s} \cdot \mathbf{s}). \tag{4.86}$$

Thus, the energy in the ith signal, E_i, is given by

$$E_i = T_s(\mathbf{s}_i \cdot \mathbf{s}_i) \tag{4.87}$$

$$= T_s\left[(\alpha_i \mathbf{s}_x + \beta_i \mathbf{s}_y) \cdot (\alpha_i \mathbf{s}_x + \beta_i \mathbf{s}_y)\right] \tag{4.88}$$

$$= T_s\left[(\alpha_i^2 \mathbf{s}_x \cdot \mathbf{s}_x + \beta_i^2 \mathbf{s}_y \cdot \mathbf{s}_y + 2\alpha_i\beta_i \mathbf{s}_x \cdot \mathbf{s}_y)\right] \tag{4.89}$$

$$= T_s(\alpha_i^2 |\mathbf{s}_x|^2 + \beta_i^2 |\mathbf{s}_y|^2) \tag{4.90}$$

$$= T_s(\alpha_i^2 + \beta_i^2). \tag{4.91}$$

The result in (4.91) is interesting: since $E_i = |\mathbf{s}_i|^2$, energy per symbol is merely the square of the *length* of the vector $\mathbf{s}_i$. Further, the length of the vector in signal space is $\sqrt{\alpha_i^2 + \beta_i^2}$, since the coordinates α_i and β_i are with respect to an orthonormal basis. Thus, vector length is preserved under a change of basis.

Finally, energy per symbol and energy per bit are calculated similarly to the 1-d case, as illustrated in the following example.

EXAMPLE 4.8. *... 2-d energy per bit ...*

4.2.5. Hardware model.

4.3. Problems

(1) For the alphabetic encoding of Example 4.3, propose binary and 4-ary alphabets, and calculate their bit rates. Why are these bit rates higher than the 3-ary encoding?

(2) (show ϕ_x and ϕ_y are independent)

4.4. Laboratory exercises

(probability of error per bit versus per symbol)

CHAPTER 5

Multiple Access Communication Systems

5.1. Passband Data Transmission

Two-dimensional signal space is most commonly seen in *passband data transmission*, in which a carrier frequency is used in order to fit a signal in a specific frequency band.

5.1.1. Bandwidth and analog modulation. In practical communication systems, signals are allocated a certain bandwidth in frequency space. This bandwidth may result from hardware limitations (e.g., traditional telephone line signals cannot contain a DC component), or to allow the sharing of spectrum (e.g., in broadcast radio, each station is allocated a certain frequency band for its exclusive use).

Given a baseband signal, and a specified frequency band, two things must be true in order to deposit that signal into the given band:

(1) The baseband signal must be *band-limited*, i.e. to fit in a bandwidth B, the signal must start out with bandwidth not greater than B.
(2) There must be a way of *modulating* the baseband signal to fit in an arbitrary frequency band.

The first condition is satisfied by examining the Fourier transform $S(f)$ of the signal. There are many possible definitions of B, but the following is suitable for our purposes:

$$B = 2\min\{f : |S(f')| \leq \epsilon \; \forall f' > f\}. \tag{5.1}$$

There are many valid settings of ϵ, and we will not consider the selection of this parameter. However, under this definition, it is not generally feasible to select $\epsilon = 0$; all time-limted signals have infinite support in the frequency domain. (It is worth remembering that, so long as $B/2 \geq 2/T_s$, the signal satisfies the Nyquist sampling

criterion.) Furthermore, the bandwidth B is proportional to T_b, the amount of time spent to transmit one bit. Throughout the remainder of this document, we will assume that

$$B = \frac{2}{T_b}. \tag{5.2}$$

The second criterion is satisfied if the signal is multiplied by a pure sinusoid, called a *carrier*. For example, suppose we have a signal $s(t)$, with Fourier transform $S(f)$. Further, suppose the bandwidth of $S(f)$ is B. We then form the modulated signal

$$s'(t) = s(t)\cos(2\pi f_c t). \tag{5.3}$$

Taking the Fourier transform $S'(f)$ of $s'(t)$, it is well known that $S'(f)$ is centered around f_c, and has support from $f_c - B/2$ to $f_c + B/2$. That is, the signal still has bandwidth B, but in the *passband*. In this case, f_c is called the *carrier frequency*.

It makes no difference to apply an arbitrary phase shift to the carrier, so we can rewrite (5.3) as

$$s'(t) = s(t)\cos(2\pi f_c t + \theta). \tag{5.4}$$

If $\theta = -\pi/2$, then $s'(t) = s(t)\sin(2\pi f_c t)$. We will use this property to form a two-dimensional *signal space* in passband, with one basis vector formed by a $\cos(\cdot)$ carrier, and the other formed by a $\sin(\cdot)$ carrier.

5.1.2. Passband hardware model. The hardware model for passband communication is shown in Figure X. It is important to note two modifications from the baseband hardware model. First, the passband signal is *digitally downconverted* from the passband to the baseband – that is, following analog-to-digital conversion, it is multiplied by sinusoids and lowpass filtered to center the signal at zero frequency as a baseband signal. Once in the baseband, we handle the signal in the same way as we did in Chapters 2 and 4. Second, as we suggested above, we use two sinusoids to downconvert: one cosine, and one sine.

First consider the problem of downconversion. Let $s'[k]$ represent the sampled version of $s'(t)$, where (supposing $\theta = 0$)

$$s'[k] = s(kT_s)\cos(2\pi f_c k T_s) \tag{5.5}$$

$$= s[k]\cos(2\pi f_d k), \tag{5.6}$$

where $s[k] = s(kT_s)$ is the sampled version of $s(t)$, and where $f_d = f_c T_s$ is the equivalent digital carrier frequency. For convenience, we will assume that f_d is an integer multiple of $1/n_b$ (i.e., each symbol contains an integer number of cycles of the digital carrier).

The downconverter multiplies $s'[k]$ by $2\cos(2\pi f_d k)$, obtaining

$$s'[k](2\cos(2\pi f_d k)) = 2s[k]\cos(2\pi f_d k)^2 \tag{5.7}$$

$$= s[k] + s[k]\cos(4\pi f_d k^2). \tag{5.8}$$

As long as f_d is much greater than the bandwidth of $s[k]$, the term $s[k]\cos(4\pi f_d k^2)$ can be eliminated by a lowpass filter; we omit the details. Thus, the output of the downconverter is the baseband signal $s[k]$.

Second, suppose

$$s'(t) = s(t)\sin(2\pi f_c t), \tag{5.9}$$

i.e., replacing cosine with sine. Then (5.7)-(5.8) become

$$s'[k](2\cos(2\pi f_d k)) = 2s[k]\sin(2\pi f_d k)cos(2\pi f_d k)^2 \tag{5.10}$$

$$= s[k]\sin(4\pi f_d k^2). \tag{5.11}$$

which is eliminated by the lowpass filter; thus, the output of the downconverter is zero. However, suppose we multiply instead by $2\sin(2\pi f_c t)$. Then

$$s'[k](2\sin(2\pi f_d k)) = 2s[k]\sin(2\pi f_d k)^2 \tag{5.12}$$

$$= s[k] - s[k]\sin(4\pi f_d k^2), \tag{5.13}$$

which again returns $s[k]$ after lowpass filtering.

We can use this property to form a *signal space*: signal $s_I[k]$ (called the *in phase* signal) can be transmitted with carrier $\cos(2\pi f_c t)$, while signal $s_Q[k]$ (called the *quadrature* signal) can be transmitted with carrier $\sin(2\pi f_c t)$. Two downconversion

legs are used: the in phase leg extracts $s_I[k]$, while the quadrature leg extracts $s_Q[k]$.

Suppose $s[k]$ is constant ...

EXAMPLE 5.1 (Quadrature phase shift keying).

5.2. Interference-free spectrum sharing

Passband techniques may be used to share bandwidth among multiple users. In this section, we consider two *interference-free* techniques: frequency division multiple access (FDMA), and time division multiple access (TDMA).

5.2.1. Frequency division multiple access.

5.2.2. Time division multiple access.

5.3. Spread-spectrum techniques

In the previous section, we considered methods for *interference-free* spectrum sharing, in which no interference is permitted between users of the system. Although such methods provide the best possible channel to each user, they have two important disadvantages. First, these schemes are inefficient when the users' traffic is intermittent, such as in packet data systems. Since TDMA time slots and FDMA frequency bands are allocated for relatively long periods of time, it is difficult for the system to take advantage of pauses when a user is silent. Second, these schemes require a great deal of overhead to set up and maintain. Either a central controller must be used to allocate slots, or some sort of contention window must be used to allow users to compete for slots.

On the other hand, if some interference is allowed between users, both these problems are mitigated. Systems using this principle are generally called *spread spectrum* systems, as their transmissions are "spread out" over the entire system bandwidth, but in a clever way so as to minimize (though not completely eliminate) interference with other users. In this section, we will mainly discuss *direct sequence spread spectrum*, also known as code division multiple access (CDMA), but the same principles apply to other spread spectrum systems.

5.3.1. CDMA: System Model. From the perspective of a single user, CDMA may be viewed as a binary transmission scheme, where the modulation waveform $s_0[k]$ is composed of sub-waveforms, called *chips*.

Let $c_0[k]$ and $c_1[k]$ represent chips, which have support on $1 \leq k \leq n_c$. In order to form $s_0[k]$, we obtain a binary vector known as a *chip sequence* $\mathbf{h}$. For each bit in the chip sequence, we replace that bit in $s_0[k]$ with the corresponding chip. For example, if $\mathbf{h} = [0, 1, 0, 0, 1]$, then

$$s_0[k] = [c_0[k], c_1[k], c_0[k], c_0[k], c_1[k]]. \tag{5.14}$$

Thus, if the length of $\mathbf{h}$ is ℓ_c, then $n_b = \ell_c n_c$.

Unless otherwise specified, we will use $s_1[k] = -s_0[k]$, and $c_1[k] = -c_0[k]$. Except for the unusual structure of $s_0[k]$ and $s_1[k]$, all of the analysis from Chapters 2 and 3 are still valid from the perspective of a single user.

EXAMPLE 5.2. *chip sequence, bandwidth*

The main idea in CDMA is to provide each user with a unique chip sequence, carefully chosen so as to be *almost* orthogonal under any time shift. The users can then use their chip sequences at the same time, without significantly interfering with each other.

5.3.2. Feedback shift registers. Chip sequences with the required properties can be generated using feedback shift registers.

5.3.3. Probability of error analysis: Optimal. We take two approaches to the problem of analyzing CDMA's error performance: first, we consider the *optimal* approach in a two-user synchronized system, and then we consider a *suboptimal* but low-complexity approach in a multi-user unsynchronized system.

Consider the two-user case. Suppose users 1 and 2 have chip sequences $\mathbf{h}^{(1)}$ and $\mathbf{h}^{(2)}$, respectively, where

$$\mathbf{h}^{(1)} = [1, 1, 1, 0, 1, 0, 0] \tag{5.15}$$

$$\mathbf{h}^{(2)} = [0, 0, 1, 0, 1, 1, 1]. \tag{5.16}$$

Furthermore, suppose the chips are given by

$$c_0[k] = \begin{cases} 1, & 1 \le k \le n_c, \\ 0, & \text{otherwise}, \end{cases} \tag{5.17}$$

and suppose $c_1[k] = -c_0[k]$.

Similarly to our approach from Chapter 4, we can form $c_0[k]$ and $c_1[k]$ into vectors. Thus, $\mathbf{c}_0 = [1, 1, \ldots, 1]$ (of length n_c), and $\mathbf{c}_1 = -\mathbf{c}_0$. Let $\mathbf{s}^{(1)}$ and $\mathbf{s}^{(2)}$ be basis vectors corresponding to the two users, and let $s_i^{(j)}$ represent the modulation function corresponding to the jth user's ith symbol. Thus, recalling $\mathbf{h}^{(1)}$ and $\mathbf{h}^{(2)}$, we have

$$\mathbf{s}^{(1)} = [\mathbf{c}_1, \mathbf{c}_1, \mathbf{c}_1, \mathbf{c}_0, \mathbf{c}_1, \mathbf{c}_0, \mathbf{c}_0] \tag{5.18}$$

$$\mathbf{s}^{(2)} = [\mathbf{c}_0, \mathbf{c}_0, \mathbf{c}_1, \mathbf{c}_0, \mathbf{c}_1, \mathbf{c}_1, \mathbf{c}_1], \tag{5.19}$$

and

$$\mathbf{s}_0^{(1)} = \mathbf{s}^{(1)} \tag{5.20}$$

$$\mathbf{s}_1^{(1)} = -\mathbf{s}^{(1)} \tag{5.21}$$

$$\mathbf{s}_0^{(2)} = \mathbf{s}^{(2)} \tag{5.22}$$

$$\mathbf{s}_1^{(2)} = -\mathbf{s}^{(2)}. \tag{5.23}$$

We assume that the users are *bit-synchronized*, i.e., each user's bit starts at the same time. Thus, the transmitted signal can be written as a vector $\mathbf{x}$, where

$$\mathbf{x} = (1 - 2b^{(1)})\mathbf{s}^{(1)} + (1 - 2b^{(2)})\mathbf{s}^{(2)}, \tag{5.24}$$

where $b^{(1)} \in \{0, 1\}$ and $b^{(2)} \in \{0, 1\}$ are the bits transmitted by users 1 and 2, respectively; the expression $(1 - 2b)$ maps the bit b into $\{+1, -1\}$.

Although from (5.24), $\mathbf{s}^{(1)}$ and $\mathbf{s}^{(2)}$ form a basis for the signal $\mathbf{x}$, it should be clear that they do not form an *orthonormal basis*. Following the procedure from Chapter 4, we can orthonormalize the basis, as follows. First, we find $\mathbf{s}_x$ by normalizing $\mathbf{s}^{(1)}$:

$$\mathbf{s}_x = \frac{\mathbf{s}^{(1)}}{|\mathbf{s}^{(1)}|} \tag{5.25}$$

$$= \frac{1}{\sqrt{7n_c}}\mathbf{s}^{(1)}. \tag{5.26}$$

Next, we remove the component of $\mathbf{s}_x$ from $\mathbf{s}^{(2)}$, and normalize. We first need to find $\mathbf{s}^{(2)} \cdot \mathbf{s}_x$, given by

$$\mathbf{s}^{(2)} \cdot \mathbf{s}_x = \frac{1}{\sqrt{7n_c}}(\mathbf{s}^{(1)} \cdot \mathbf{s}^{(2)}) \tag{5.27}$$

$$= \frac{1}{\sqrt{7n_c}}([\mathbf{c}_1, \mathbf{c}_1, \mathbf{c}_1, \mathbf{c}_0, \mathbf{c}_1, \mathbf{c}_0, \mathbf{c}_0] \cdot [\mathbf{c}_0, \mathbf{c}_0, \mathbf{c}_1, \mathbf{c}_0, \mathbf{c}_1, \mathbf{c}_1, \mathbf{c}_1]) \tag{5.28}$$

$$= \frac{1}{\sqrt{7n_c}}(2(\mathbf{c}_1 \cdot \mathbf{c}_1) + \mathbf{c}_0 \cdot \mathbf{c}_0 + 4(\mathbf{c}_1 \cdot \mathbf{c}_0)), \tag{5.29}$$

where the last line follows from breaking up the vector into its $\mathbf{c}_0$ and $\mathbf{c}_1$ components, as well as the fact that dot product is commutative. It is easy to see that $\mathbf{c}_1 \cdot \mathbf{c}_1 = \mathbf{c}_0 \cdot \mathbf{c}_0 = n_c$, while $\mathbf{c}_1 \cdot \mathbf{c}_0 = -n_c$, so we have

$$\mathbf{s}^{(2)} \cdot \mathbf{s}_x = \frac{1}{\sqrt{7n_c}}(2n_c + n_c - 4n_c) \tag{5.30}$$

$$= -\frac{1}{\sqrt{7n_c}} n_c \tag{5.31}$$

$$= -\sqrt{\frac{n_c}{7}}. \tag{5.32}$$

Now, $\mathbf{s}_y$ is given by

$$\mathbf{s}_y = \frac{\mathbf{s}^{(2)} - (\mathbf{s}^{(2)} \cdot \mathbf{s}_x)\mathbf{s}_x}{|\mathbf{s}^{(2)} - (\mathbf{s}^{(2)} \cdot \mathbf{s}_x)\mathbf{s}_x|}. \tag{5.33}$$

The numerator of this expression is given by

$$\mathbf{s}^{(2)} - (\mathbf{s}^{(2)} \cdot \mathbf{s}_x)\mathbf{s}_x$$

$$= [\mathbf{c}_0, \mathbf{c}_0, \mathbf{c}_1, \mathbf{c}_0, \mathbf{c}_1, \mathbf{c}_1, \mathbf{c}_1] + \sqrt{\frac{n_c}{7}} \frac{1}{\sqrt{7n_c}} [\mathbf{c}_1, \mathbf{c}_1, \mathbf{c}_1, \mathbf{c}_0, \mathbf{c}_1, \mathbf{c}_0, \mathbf{c}_0] \tag{5.34}$$

$$= [\mathbf{c}_0, \mathbf{c}_0, \mathbf{c}_1, \mathbf{c}_0, \mathbf{c}_1, \mathbf{c}_1, \mathbf{c}_1] + \frac{1}{7}[\mathbf{c}_1, \mathbf{c}_1, \mathbf{c}_1, \mathbf{c}_0, \mathbf{c}_1, \mathbf{c}_0, \mathbf{c}_0] \tag{5.35}$$

$$= [\frac{6}{7}\mathbf{c}_0, \frac{6}{7}\mathbf{c}_0, -\frac{8}{7}\mathbf{c}_0, \frac{8}{7}\mathbf{c}_0, -\frac{8}{7}\mathbf{c}_0, -\frac{6}{7}\mathbf{c}_0, -\frac{6}{7}\mathbf{c}_0], \tag{5.36}$$

where the last line follows since $\mathbf{c}_1 = -\mathbf{c}_0$. Finally, the norm of the numerator is $\sqrt{48n_c/7}$. Thus,

$$\mathbf{s}_y = \sqrt{\frac{7}{48n_c}}[\frac{8}{7}\mathbf{c}_0, \frac{8}{7}\mathbf{c}_0, -\frac{6}{7}\mathbf{c}_0, \frac{6}{7}\mathbf{c}_0, -\frac{6}{7}\mathbf{c}_0, -\frac{8}{7}\mathbf{c}_0, -\frac{8}{7}\mathbf{c}_0]. \tag{5.37}$$

Expressing our original basis vectors in terms of the orthonormal basis, we have

$$\mathbf{s}^{(1)} = \sqrt{7n_c}\mathbf{s}_x, \tag{5.38}$$

and

$$\mathbf{s}^{(2)} = \sqrt{\frac{48n_c}{7}}\mathbf{s}_y + \frac{n_c}{7}\mathbf{s}_x. \tag{5.39}$$

The four possible transmitted signals, along with their corresponding Voronoi regions, are depicted in Figure X. We can see that the Voronoi regions are not rectangular. This approach, known as *multiuser detection*, is optimal but becomes intractably complex as the number of users increases beyond two.

5.3.4. Probability of error analysis: Suboptimal. The complexity of the above detection problem stems from the need to optimally detect both users' transmissions at the same time. On the other hand, we can ignore the interference from other users, and assume that it is equivalent to noise.

The quantity $\mathbf{s}^{(1)} \cdot \mathbf{s}^{(2)}$ is known as *cross-correlation*. We see from the above derivation that cross-correlation is a key parameter to determine the probability of error in multiuser detection. More generally, cross-correlation can be defined in terms of any possible relative delay between user 1 and user 2's chip sequences. First define

$$\mathbf{s}^{(2)}_{\rightarrow k} = [s^{(2)}_{k+1}, s^{(2)}_{k+2}, \ldots, s^{(2)}_{n_b}, s^{(2)}_1, s^{(2)}_2, \ldots, s^{(2)}_k] \tag{5.40}$$

as the kth circular shift of $\mathbf{s}^{(2)}$. The cross-correlation under this circular shift can be found by calculating

$$C^{(1,2)}[k] = \mathbf{s}^{(1)} \cdot \mathbf{s}^{(2)}_{\rightarrow k}. \tag{5.41}$$

If $\mathbf{s}^{(2)}$ is user 2's spreading code, then the first $n_b - k$ samples from $\mathbf{s}^{(2)}$ are modulated with bit $b_1 \in \{+1, -1\}$, while the last k samples are modulated with bit $b_2 \in \{+1, -1\}$. Thus, define

$$\mathbf{s}^{(2)}_{\rightarrow k(b_1,b_2)} = [b_1 s^{(2)}_{k+1}, b_1 s^{(2)}_{k+2}, \ldots, b_1 s^{(2)}_{n_b}, b_2 s^{(2)}_1, b_2 s^{(2)}_2, \ldots, b_2 s^{(2)}_k] \tag{5.42}$$

as user 2's interfering signal, assuming that user 1's signal is of interest.

Following the arguments from previous chapters, let $\mathbf{s}_x$ represent the normalized basis vector corresponding to $\mathbf{s}^{(1)}$, that is,

$$\mathbf{s}_x = \frac{\mathbf{s}^{(1)}}{|\mathbf{s}^{(1)}|} \tag{5.43}$$

$$= \frac{1}{\sqrt{n_b}}\mathbf{s}^{(1)}. \tag{5.44}$$

The signal observed at the input of user 1's matched filter is

$$\mathbf{y} = \alpha \mathbf{s}^{(1)} + \mathbf{s}^{(2)}_{\rightarrow k(b_1,b_2)} + \mathbf{n}, \tag{5.45}$$

where $\mathbf{n}$ is a vector of additive white Gaussian noise. The noise-free matched filter outputs are thus given by

$$\mathbf{s}_x \cdot (\alpha \mathbf{s}^{(1)} + \mathbf{s}^{(2)}_{\rightarrow k(b_1,b_2)}) = \alpha \mathbf{s}_x \cdot \mathbf{s}^{(1)} + \mathbf{s}_x \cdot \mathbf{s}^{(2)}_{\rightarrow k(b_1,b_2)} \tag{5.46}$$

$$= \sqrt{n_b}\alpha + \mathbf{s}_x \cdot \mathbf{s}^{(2)}_{\rightarrow k(b_1,b_2)}. \tag{5.47}$$

User 1's signal is contained in the $\sqrt{n_b}$ term, while the interference from user 2 is contained in the $\mathbf{s}_x \cdot \mathbf{s}^{(2)}_{\rightarrow k(b_1,b_2)}$ term.

In suboptimal analysis, we treat the interference term $\mathbf{s}_x \cdot \mathbf{s}^{(2)}_{\rightarrow k(b_1,b_2)}$ as noise. Noise in our system is Gaussian, which is characterized by its variance (since the mean is always zero). Furthermore, from Chapter 1, since the interference and noise are independent, we have that

$$\mathrm{Var}(\mathbf{s}^{(2)}_{\rightarrow k(b_1,b_2)} \cdot \mathbf{s}_x + \mathbf{n} \cdot \mathbf{s}_x) = \mathrm{Var}(\mathbf{s}^{(2)}_{\rightarrow k(b_1,b_2)} \cdot \mathbf{s}_x) + \mathrm{Var}(\mathbf{n} \cdot \mathbf{s}_x), \tag{5.48}$$

where $\mathrm{Var}(\mathbf{n} \cdot \mathbf{s}_x) = N_0/2T_s$ is the variance of the noise at the output of the matched filter. In (5.48), the variance is taken with respect to k, b_1, and b_2, since these are all random.

EXAMPLE 5.3. *Calculation of* $\mathrm{Var}(\mathbf{s}^{(2)}_{\rightarrow k(b_1,b_2)})$

We can take (5.48) further. Suppose there are $u \geq 2$ users, all interfering with user 1. By assumption, all their transmissions are statistically independent. Thus, the total variance of the interference and noise is given by

$$\sigma^2 = \frac{N_0}{2T_s} + \sum_{j=2}^{u} \mathrm{Var}(\mathbf{s}^{(j)}_{\rightarrow k(b_1,b_2)} \cdot \mathbf{s}_x) \tag{5.49}$$

Let's assume that this variance is the same for all users and is equal to v. Then (5.49) reduces to

$$\sigma^2 = \frac{N_0}{2T_s} + (u-1)v. \tag{5.50}$$

Substituting this value of σ^2 back into the probability of error expressions from Chapter 2, we have

$$\Pr(\mathsf{error}) = \frac{1}{2}\mathrm{erfc}\left(\frac{\mathbf{s}_x \cdot \mathbf{s}^{(1)}}{\sqrt{2\sigma^2}}\right) \tag{5.51}$$

$$= \frac{1}{2}\mathrm{erfc}\left(\sqrt{\frac{n_b T_s}{N_0 + 2T_s(u-1)v}}\right). \tag{5.52}$$

CHAPTER 6

An Introduction to Information Theory

6.1. Error-control coding

6.1.1. Capacity.

6.1.2. Linear block codes.

6.1.3. Convolutional codes.

6.2. Data compression

6.2.1. Entropy.

6.2.2. Huffman codes.

Bibliography

[1] A. Leon-Garcia, *Probability and Random Processes for Electrical Engineering, 2nd ed.*, Reading, MA: Addison-Wesley, 1994.

[2] A. Papoulis, *Probability, Random Variables, and Stochastic Processes, 3rd ed.*, New York, NY: McGraw-Hill, 1991.

[3] S. Haykin, *Communication Systems, 4th ed.*, New York, NY: Wiley, 2000.

APPENDIX A

Fourier Transforms

A.1. Properties

A.2. Table of fourier transform pairs

A.2.1. Definitions.

$$\text{rect}(t) = \begin{cases} 1, & |t| < 1/2 \\ 1/2, & |t| = 1/2 \\ 0, |t| > 1/2 \end{cases} \tag{A.1}$$

$$\text{sinc}(t) = \frac{\sin(\pi t)}{\pi t} \tag{A.2}$$

A.2.2. Table.

Fourier transform	$\leftrightarrow$	Time domain
$\text{rect}(j\omega)$		$\frac{1}{2\pi}\text{sinc}\left(\frac{t}{2\pi}\right)$

(A.3)

APPENDIX B

The Cauchy-Schwartz Inequality

In this appendix, we prove the Cauchy-Schwartz inequality, which we used in Chapter 3 to design the optimal detection filter.