Chapter 8

Detection, coding, and decoding

8.1 Introduction

The previous chapter showed how to characterize noise as a random process and this chapter uses that characterization to retrieve the signal from the noise corrupted received waveform. As one might guess, this is not possible without occasional errors when the noise is unusually large. The objective then, is to retrieve the data while minimizing the effect of these errors. This process of retrieving data from a noise corrupted version is known as *detection*.

Detection, decision making, hypothesis testing, and decoding are synonyms. The word *detection* refers to the effort to detect whether some phenomenon is present or not on the basis of observations. For example, a radar system uses the observations to detect whether or not a target is present; a quality control system attempts to *detect* whether a unit is defective; a medical test *detects* whether a given disease is present. The meaning of detection has been extended in the digital communication field from a yes/no decision to a decision at the receiver from a finite set of possible transmitted signals. Such a decision from a set of possible transmitted signals is also called *decoding*, but here the possible set is usually regarded as the codewords in a code rather than the signals in a signal set.[1] *Decision making* is, again, the process of deciding between a number of mutually exclusive alternatives. *Hypothesis testing* is the same, and here the mutually exclusive alternatives are called hypotheses. We use the word hypotheses for the possible choices in what follows, since the word conjures up the appropriate intuitive image of making a choice between a set of alternatives, where only one alternative is correct and there is a possibility of erroneous choice.

These problems will be studied initially in a purely probabilistic setting. That is, there is a probability model within which each hypothesis is an event. These events are mutually exclusive and collectively exhaustive, i.e., the sample outcome of the experiment lies in one and only one of these events, which means that in each performance of the experiment, one and only one hypothesis is correct. Assume there are M hypotheses[2], labeled $a_0, \ldots, a_{M-1}$. The sample outcome of the experiment will lie in one of these M events. This defines a random symbol U

[1] As explained more fully later, there is no fundamental difference between a code and a signal set.

[2] The principles here apply essentially without change for a countably infinite set of hypotheses; for an uncountably infinite set of hypotheses, the process of choosing an hypothesis from an observation is called *estimation*. Typically, the probability of choosing correctly in this case is 0 and the emphasis is on making an estimate that is close in some sense to the correct hypothesis.

which, for each m, takes the sample value a_m when event a_m occurs. The marginal probability $p_U(a_m)$ of hypothesis a_m is denoted p_m and is usually referred to as the *a priori probability* of a_m. There is also a random variable (rv) V, called the observation. This is the data on which the decision must be based. A sample value v of V is observed, and on the basis of that observation, the detector selects one of the possible M hypotheses. The observation could equally well be a complex random variable, a random vector, a random process, or a random symbol, and these generalizations are discussed in what follows.

Before discussing how to make decisions, it is important to understand when and why decisions must be made. As a binary example, assume that the conditional probability of hypothesis a_0, given the observation, is 2/3 and that of hypothesis a_1 is 1/3. Simply deciding on hypothesis a_0 and forgetting about the probabilities throws away the information about the probability that the decision is correct. However, actual decisions sometimes must be made. In a communication system, the user usually wants to receive the message (even partly garbled) rather than a set of probabilities. In a control system, the controls must occasionally take action. Similarly managers must occasionally choose between courses of action, between products, and between people to hire. In a sense, it is by making decisions that we return from the world of mathematical probability models to the world being modeled.

There are a number of possible criteria to use in making decisions. Initially assume that the criterion is to maximize the probability of correct choice. That is, when the experiment is performed, the resulting experimental outcome maps into both a sample value a_m for U and a sample value v for V. The decision maker observes v (but not a_m) and maps v into a decision $\tilde{u}(v)$. The decision is correct if $\tilde{u}(v) = a_m$. In principal, maximizing the probability of correct choice is almost trivially simple. Given v, calculate $p_{U|V}(a_m \,|\, v)$ for each possible hypothesis a_m. This is the probability that a_m is the correct hypothesis conditional on v. Thus the rule for maximizing the probability of being correct is to choose $\tilde{u}(v)$ to be that a_m for which $p_{U|V}(a_m \,|\, v)$ is maximized. For each possible observation v, this is denoted

$$\tilde{u}(v) = \arg\max_m [p_{U|V}(a_m \,|\, v)] \qquad \text{(MAP rule)}, \tag{8.1}$$

where $\arg\max_m$ means the argument m that maximizes the function. If the maximum is not unique, the probability of being correct is the same no matter which maximizing m is chosen, so to be explicit, the smallest such m will be chosen.[3] Since the rule (8.1) applies to each possible sample output v of the random variable V, (8.1) also defines the selected hypothesis as a random symbol $\tilde{U}(V)$. The conditional probability $p_{U|V}$ is called an *a posteriori probability*. This is in contrast to the *a priori probability* p_U of the hypothesis before the observation of V. The decision rule in (8.1) is thus called the maximum a posteriori probability (MAP) rule.

An important consequence of (8.1) is that the MAP rule depends only on the conditional probability $p_{U|V}$ and thus is completely determined by the joint distribution of U and V. Everything else in the probability space is irrelevant to making a MAP decision.

When distinguishing between different decision rules, the MAP decision rule in (8.1) is denoted as $\tilde{u}_{\text{MAP}}(v)$. Since the MAP rule maximizes the probability of correct decision for each sample value v, it also maximizes the probability of correct decision averaged over all v. To see this

[3] As discussed in the appendix, it is sometimes desirable to choose randomly among the maximum aposteriori choices when the maximum in (8.1) is not unique. There are often situations (such as with discrete coding and decoding) where non-uniqueness occurs with positive probability.

analytically, let $\tilde{u}_{_D}(v)$ be an arbitrary decision rule. Since $\tilde{u}_{\text{MAP}}$ maximizes $p_{U|V}(m\,|\,v)]$ over m,

$$p_{U|V}(\tilde{u}_{\text{MAP}}(v)\,|\,v) - p_{U|V}(\tilde{u}_{_D}(v)\,|\,v) \geq 0; \qquad \text{for each rule D and observation } v. \tag{8.2}$$

Taking the expected value of the first term on the left over the observation V, we get the probability of correct decision using the MAP decision rule. The expected value of the second term on the left, for any given D is the probability of correct decision using that rule. Thus, taking the expected value of (8.2) over V shows that the MAP rule maximizes the probability of correct decision over the observation space. The above results are very simple, but also important and fundamental. They are summarized in the following theorem.

Theorem 8.1.1. *The MAP rule, given in (8.1), maximizes the probability of correct decision, both for each observed sample value v and as an average over V. The MAP rule is determined solely by the joint distribution of U and V.*

Before discussing the implications and use of the MAP rule, the above assumptions are reviewed. First, a probability model was assumed in which all probabilities are known, and in which, for each performance of the experiment, one and only one hypothesis is correct. This conforms very well to the communication model in which a transmitter sends one of a set of possible signals, and the receiver, given signal plus noise, makes a decision on the signal actually sent. It does not always conform well to a scientific experiment attempting to verify the existence of some new phenomenon; in such situations, there is often no sensible way to model a priori probabilities. Detection in the absence of known a priori probabilities is discussed in the appendix.

The next assumption was that maximizing the probability of correct decision is an appropriate decision criterion. In many situations, the cost of a wrong decision is highly asymmetric. For example, when testing for a treatable but deadly disease, making an error when the disease is present is far more costly than making an error when the disease is not present. As shown in Exercise 8.1, it is easy to extend the theory to account for relative costs of errors.

With the present assumptions, the detection problem can be stated concisely in the following probabilistic terms. There is an underlying sample space Ω, a probability measure, and two rv's U and V of interest. The corresponding experiment is performed, an observer sees the sample value v of rv V, but does not observe anything else, particularly not the sample value of U, say a_m. The observer uses a detection rule, $\tilde{u}(v)$, which is a function mapping each possible value of v to a possible value, a_0 to a_{M-1}, of U. If $\tilde{v}(v) = a_m$, the detection is correct, and otherwise an error has been made. The above MAP rule maximizes the probability of correct detection conditional on each v and also maximizes the unconditional probability of correct detection. Obviously, the observer must know the conditional probability assignment $p_{_{U|V}}$ in order to use the MAP rule.

The next two sections are restricted to the case of binary hypotheses, $(M = 2)$. This allows us to understand most of the important ideas but simplifies the notation considerably. This is then generalized to an arbitrary number of hypotheses; fortunately this extension is almost trivial.

8.2 Binary detection

Assume a probability model in which the correct hypothesis U is a binary random variable with possible values $\{a_0, a_1\}$ and a priori probabilities p_0 and p_1. In the communication context,

the a priori probabilities are usually modeled as equiprobable, but occasionally there are multi-stage detection processes in which the result of the first stage leads to non-equiprobable a priori probabilities in subsequent stages. Thus let p_0 and $p_1 = 1 - p_0$ be arbitrary. Let V be a rv with a conditional probability density $f_{V|U}(v \,|\, a_m)$ that is finite and non-zero for all $v \in \mathbb{R}$ and $m \in \{0, 1\}$. The modifications for zero densities, discrete V, complex V, or vector V are relatively straight-forward and discussed later.

The conditional densities $f_{V|U}(v \,|\, a_m)$, $m \in \{0, 1\}$ are called *likelihoods* in the jargon of hypothesis testing. The marginal density of V is given by $f_V(v) = p_0 f_{V|U}(v \,|\, a_0) + p_1 f_{V|U}(v \,|\, a_1)$. The a posteriori probability of U, for $m = 0$ or 1, is given by

$$p_{U|V}(a_m \,|\, v) = \frac{p_m f_{V|U}(v \,|\, a_m)}{f_V(v)}. \tag{8.3}$$

Writing out (8.1) explicitly for this case,

$$\frac{p_0 f_{V|U}(v \,|\, a_0)}{f_V(v)} \underset{<_{\tilde{U}=a_1}}{\overset{\geq^{\tilde{U}=a_0}}{}} \frac{p_1 f_{V|U}(v \,|\, a_1)}{f_V(v)}. \tag{8.4}$$

This "equation" indicates that the MAP decision is a_0 if the left side is greater than or equal to the right, and is a_1 if the left side is less than the right. Choosing the decision $\tilde{U} = a_0$ when equality holds in (8.4) is an arbitrary choice and does not affect the probability of being correct. Canceling $f_V(v)$ and rearranging,

$$\Lambda(v) = \frac{f_{V|U}(v \,|\, a_0)}{f_{V|U}(v \,|\, a_1)} \underset{<_{\tilde{U}=a_1}}{\overset{\geq^{\tilde{U}=a_0}}{}} \frac{p_1}{p_0} = \eta. \tag{8.5}$$

$\Lambda(v) = f_{V|U}(v \,|\, a_0)/f_{V|U}(v \,|\, a_1)$ is called the *likelihood ratio*, and is a function only of v. The ratio $\eta = p_1/p_0$ is called the *threshold* and depends only on the a priori probabilities. The binary MAP rule (or MAP test, as it is usually called) then compares the likelihood ratio to the threshold, and decides on hypothesis a_0 if the threshold is reached, and on hypothesis a_1 otherwise. Note that if the a priori probability p_0 is increased, the threshold decreases, and the set of v for which hypothesis a_0 is chosen increases; this corresponds to our intuition—the more certain we are initially that U is 0, the stronger the evidence required to make us change our minds. As shown in Exercise 8.1, the only effect of minimizing over costs rather than error probability is to change the threshold η in (8.5).

An important special case of (8.5) is that in which $p_0 = p_1$. In this case $\eta = 1$, and the rule chooses $\tilde{U}(v) = a_0$ for $f_{V|U}(v \,|\, a_0) \geq f_{V|U}(v \,|\, a_1)$ and chooses $\tilde{U}(v) = 1$ otherwise. This is called a *maximum likelihood (ML) rule or test*. In the communication case, as mentioned above, the a priori probabilities are usually equal, so MAP then reduces to ML. The maximum likelihood test is also often used when p_0 and p_1 are unknown.

The *probability of error*, *i.e.*, one minus the probability of choosing correctly, is now derived for MAP detection. First we find the probability of error conditional on each hypothesis, $\Pr\{e \,|\, U{=}a_1\}$ and $\Pr\{e \,|\, U{=}a_0\}$. The overall probability of error is then given by

$$\Pr\{e\} = p_0 \Pr\{e \,|\, U{=}a_0\} + p_1 \Pr\{e \,|\, U{=}a_1\}.$$

In the radar field, $\Pr\{e \,|\, U{=}a_0\}$ is called the probability of false alarm, and $\Pr\{e \,|\, U{=}a_1\}$ is called the probability of a miss. Also $1 - \Pr\{e \,|\, U{=}a_1\}$ is called the probability of detection. In

statistics, $\Pr\{e \,|\, U{=}a_1\}$ is called the probability of error of the second kind, and $\Pr\{e \,|\, U{=}a_0\}$ is the probability of error of the first kind. These terms are not used here.

Note that (8.5) partitions the space of observed sample values into 2 regions. $R_0 = \{v : \Lambda(v) \geq \eta\}$ is the region for which $\tilde{U} = a_0$ and $R_1 = \{v : \Lambda(v) < \eta\}$ is the region for which $\tilde{U} = a_1$. For $U = a_1$, an error occurs if and only if v is in R_0, and for $U = a_0$, an error occurs if and only if v is in R_1. Thus,

$$\Pr\{e \,|\, U{=}a_0\} = \int_{R_1} f_{V|U}(v \,|\, a_0)\, dv. \tag{8.6}$$

$$\Pr\{e \,|\, U{=}a_1\} = \int_{R_0} f_{V|U}(v \,|\, a_1)\, dv. \tag{8.7}$$

Another, often simpler, approach is to work directly with the likelihood ratio. Since $\Lambda(v)$ is a function of the observed sample value v, the random variable, $\Lambda(V)$, also called a likelihood ratio, is defined as follows: for every sample point ω, $V(\omega)$ is the corresponding sample value v, and $\Lambda(V)$ is then shorthand for $\Lambda(V(\omega))$. In the same way, $\tilde{U}(V)$ (or more briefly $\tilde{U}$) is the decision random variable. In these terms, (8.5) states that

$$\tilde{U} = a_0 \quad \text{if and only if} \quad \Lambda(V) \geq \eta. \tag{8.8}$$

Thus, for MAP detection with a threshold η,

$$\Pr\{e \,|\, U{=}a_0\} = \Pr\{\tilde{U}{=}a_1 \,|\, U{=}a_0\} = \Pr\{\Lambda(V) < \eta \,|\, U{=}a_0\}. \tag{8.9}$$

$$\Pr\{e \,|\, U{=}a_1\} = \Pr\{\tilde{U}{=}a_0 \,|\, U{=}a_1\} = \Pr\{\Lambda(V) \geq \eta \,|\, U{=}a_1\}. \tag{8.10}$$

A *sufficient statistic* is defined as any function of the observation v from which the likelihood ratio can be calculated. As examples, v itself, $\Lambda(v)$, and any one-to-one function of $\Lambda(v)$ are sufficient statistics. $\Lambda(v)$, and functions of $\Lambda(v)$, are often simpler to work with than v in calculating the probability of error. This will be particularly true when vector or process observations are discussed, since $\Lambda(v)$ is always one dimensional and real.

We have seen that the MAP rule (and thus also the ML rule) is a threshold test on the likelihood ratio. Similarly the min-cost rule, (see Exercise 8.1), and the Neyman-Pearson test (which, as shown in the appendix, makes no assumptions about a priori probabilities), are threshold tests on the likelihood ratio. Not only are all these binary decision rules based only on threshold tests on the likelihood ratio, but the properties of these rules, such as the conditional error probabilities in (8.9) and (8.10) are based only on $\Lambda(V)$ and η. In fact, it is difficult to imagine any sensible binary decision procedure, especially in the digital communication context, that is not a threshold test on the likelihood ratio. Thus, once a sufficient statistic has been calculated from the observed vector, that observed vector has no further value in any decision rule of interest here.

The log likelihood ratio, $\mathrm{LLR}(V) = \ln[\Lambda(V)]$ is an important sufficient statistic which is often easier to work with than the likelihood ratio itself. As seen in the next section, the LLR is particularly convenient with Gaussian noise statistics.

8.3 Binary signals in white Gaussian noise

This section first treats standard 2-PAM, then 2-PAM with an offset, then binary signals with vector observations, and finally binary signals with waveform observations.

8.3.1 Detection for PAM antipodal signals

Consider PAM antipodal modulation (*i.e.*, 2 -PAM), as illustrated in Figure 8.1.

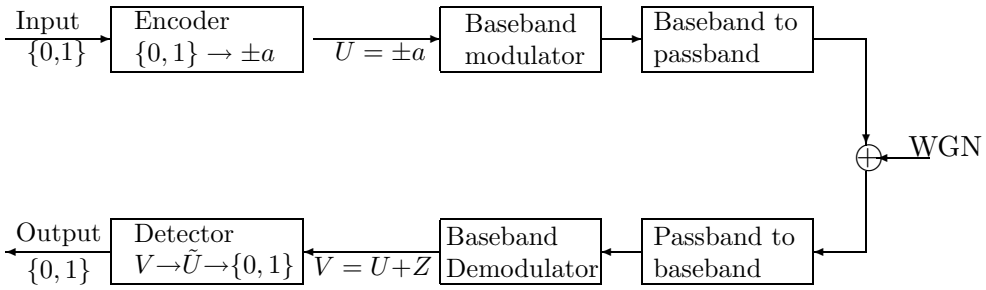

Figure 8.1: The source produces a binary digit which is mapped into $U = \pm a$. This is modulated into a waveform, WGN is added, the resultant waveform is demodulated and sampled, resulting in a noisy received value $V = U + Z$. From Section 7.8, $Z \sim \mathcal{N}(0, N_0/2)$. This is explained more fully later. Based on this observation the receiver makes a decision $\tilde{U}$ and maps this back to the binary output, which is the hypothesized version of the binary input.

The correct hypothesis U is either $a_0 = a$ or $a_1 = -a$. Let $Z \sim \mathcal{N}(0, N_0/2)$ be a Gaussian noise rv of mean 0 and variance $N_0/2$, independent of U. That is,

$$f_{_Z}(z) = \frac{1}{\sqrt{2\pi N_0/2}} \exp\left[\frac{-z^2}{N_0}\right].$$

Assume that 2-PAM is simplified by sending only a single binary symbol (rather than a sequence over time) and by observing only the single sample value v corresponding to that input. As seen later, these simplifications are unnecessary, but they permit the problem to be viewed in the simplest possible context. The observation V (*i.e.*, the channel output prior to detection) is $a+Z$ or $-a+Z$, depending on whether $U = a$ or $-a$. Thus, conditional on $U = a$, $V \sim \mathcal{N}(a, N_0/2)$ and, conditional on $U = -a$, $V \sim \mathcal{N}(-a, N_0/2)$.

$$f_{V|U}(v \,|\, a) = \frac{1}{\sqrt{\pi N_0}} \exp\left[\frac{-(v-a)^2}{N_0}\right]; \qquad f_{V|U}(v \,|\, -a) = \frac{1}{\sqrt{\pi N_0}} \exp\left[\frac{-(v+a)^2}{N_0}\right].$$

The likelihood ratio is the ratio of these likelihoods, and given by

$$\Lambda(v) = \exp\left[\frac{-(v-a)^2 + (v+a)^2}{N_0}\right] = \exp\left[\frac{4av}{N_0}\right]. \tag{8.11}$$

Substituting this into (8.5),

$$\exp\left[\frac{4av}{N_0}\right] \begin{matrix} \geq^{\tilde{U}=a} \\ <_{\tilde{U}=-a} \end{matrix} \frac{p_1}{p_0} = \eta. \tag{8.12}$$

This is further simplified by taking the logarithm, yielding

$$\mathrm{LLR}(v) = \left[\frac{4av}{N_0}\right] \begin{matrix} \geq^{\tilde{U}=a} \\ <_{\tilde{U}=-a} \end{matrix} \ln(\eta). \tag{8.13}$$

$$v \begin{matrix} \geq^{\tilde{U}=a} \\ <_{\tilde{U}=-a} \end{matrix} \frac{N_0 \ln(\eta)}{4a}. \tag{8.14}$$

Figure 8.2 interprets this decision rule.

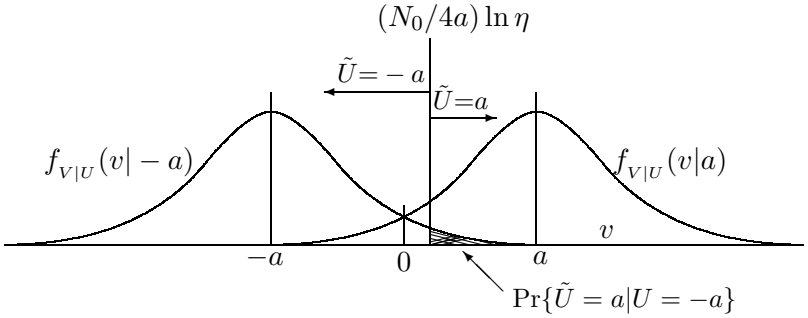

Figure 8.2: Binary hypothesis testing for antipodal signal, $0 \to a, 1 \to -a$. The a priori probabilities are p_0 and p_1, the threshold is $\eta = p_0/p_1$, and the noise is $\mathcal{N}(0, N_0/2)$.

The probability of error, given $U=-a$, is seen to be the probability that the noise value is greater than $a + \frac{N_0 \ln(\eta)}{4a}$. Since the noise has variance $N_0/2$, this is the probability that the normalized Gaussian rv $Z/\sqrt{N_0/2}$ exceeds $a/\sqrt{N_0/2} + \sqrt{N_0/2}\,\ln(\eta)/(2a)$. Thus,

$$\Pr\{e \,|\, U=-a\} = Q\left(\frac{a}{\sqrt{N_0/2}} + \frac{\sqrt{N_0/2}\ln\eta}{2a}\right), \tag{8.15}$$

where $Q(x)$, the complementary distribution function of $\mathcal{N}(0,1)$, is given by

$$Q(x) = \int_x^\infty \frac{1}{\sqrt{2\pi}} \exp\left(\frac{-z^2}{2}\right) dz.$$

The probability of error given $U=a$ is calculated the same way, but is the probability that Z is less than or equal to $-a + \frac{N_0 \ln(\eta)}{4a}$. Since $-Z$ has the same distribution as Z,

$$\Pr\{e \,|\, U=a\} = Q\left(\frac{a}{\sqrt{N_0/2}} - \frac{\sqrt{N_0/2}\ln\eta}{2a}\right). \tag{8.16}$$

It is more insightful to express $a/\sqrt{N_0/2}$ as $\sqrt{2a^2/N_0}$. As seen before, a^2 can be viewed as the energy per bit, E_b, so that (8.15) and (8.16) become

$$\Pr\{e \,|\, U=-a\} = Q\left(\sqrt{\frac{2E_b}{N_0}} + \frac{\ln\eta}{2\sqrt{2E_b/N_0}}\right), \tag{8.17}$$

$$\Pr\{e \,|\, U=a\} = Q\left(\sqrt{\frac{2E_b}{N_0}} - \frac{\ln\eta}{2\sqrt{2E_b/N_0}}\right). \tag{8.18}$$

Note that these formulas involve only the ratio E_b/N_0 rather than E_b or N_0 separately. If the signal, observation, and noise had been measured on a different scale, then both E_b and N_0 would change by the same factor, helping explain why only the ratio is relevant. In fact, the scale could be normalized so that either the noise has variance 1 or the signal has variance 1.

The hypotheses in these communication problems are usually modeled as equiprobable, $p_0 = p_1 = 1/2$. In this case, $\ln \eta = 0$ and MAP detection is equivalent to ML. Eqns. (8.17) and (8.18) then simplify to

$$\Pr\{e\} = \Pr\{e \,|\, U{=}-a\} = \Pr\{e \,|\, U{=}a\} = Q\left(\sqrt{\frac{2E_b}{N_0}}\right). \tag{8.19}$$

In terms of Figure 8.2, this is the tail of either Gaussian distribution from the point 0 where they cross. This equation keeps reappearing in different guises, and it will soon seem like a completely obvious result for a variety of Gaussian detection problems.

8.3.2 Detection for binary non-antipodal signals

Next consider the slightly more complex case illustrated in Figure 8.3. Instead of mapping 0 to $+a$ and 1 to $-a$, 0 is mapped to an arbitrary number b_0 and 1 to an arbitrary number b_1. To analyze this, let c be the mid-point between b_0 and b_1, $c = (b_0 + b_1)/2$. Assuming $b_1 < b_0$, let $a = b_0 - c = c - b_1$. Conditional on $U{=}b_0$, the observation is $V = c + a + Z$; conditional on $U{=}b_1$, it is $V = c - a + Z$. In other words, this more general case is simply the result of shifting the previous signals by the constant c.

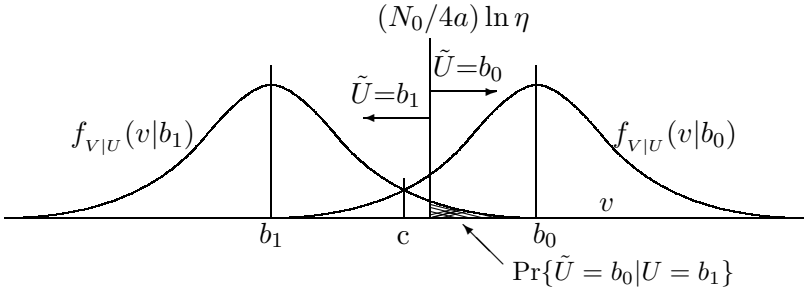

Figure 8.3: Binary hypothesis testing for arbitrary signals, $0 \to b_0, 1 \to b_1$, for $b_0 > b_1$. With $c = (b_0{+}b_1)/2$ and $a = |b_0 - b_1|/2$, this is the same as Figure 8.2 shifted by c. For $b_0 < b_1$, the picture must be reversed, but the answer is the same.

Define $\tilde{V} = V - c$ as the result of shifting the observation by $-c$. $\tilde{V}$ is a sufficient statistic and $\tilde{V} = \pm a + Z$. This is the same as the problem above, so the error probability is again given by (8.15) and (8.16).

The energy used in achieving this error probability has changed from the antipodal case. Assuming equal a priori probabilities, the energy per bit is now $(b_0^2 + {b_1}^2)/2 = a^2 + c^2$. A center value c is frequently used as a 'pilot tone' in communication for tracking the channel. We see that E_b is then the sum of the energy used for the actual binary transmission (a^2) plus the energy used for the pilot tone (c^2). The fraction of energy E_b used for the signal is $\gamma = \frac{a^2}{a^2+c^2}$. This changes (8.19) to

$$\Pr\{e \,|\, U{=}b_1\} = \Pr\{e \,|\, U{=}b_0\} = Q\left(\sqrt{\frac{2\gamma E_b}{N_0}}\right) \tag{8.20}$$

For example, a common binary communication technique called *on-off keying* uses the binary signals 0 and $2a$. In this case, $\gamma = 1/2$ and there is an energy loss of 3 dB from the antipodal case. For ML, the probability of error then becomes, $Q(\sqrt{E_b/N_0})$.

8.3.3 Detection for binary real vectors in WGN

Next consider the vector version of the Gaussian detection problem. Suppose the observation is a random n-vector $\boldsymbol{V} = \boldsymbol{U} + \boldsymbol{Z}$. The noise $\boldsymbol{Z}$ is a random n-vector $(Z_1, Z_2, \ldots, Z_n)^\mathsf{T}$, independent of $\boldsymbol{U}$, with iid components given by $Z_k \sim \mathcal{N}(0, N_0/2)$. The input $\boldsymbol{U}$ is a random n-vector with M possible values (hypotheses). The mth hypothesis, $0 \le m \le M-1$, is denoted by $\boldsymbol{a}_m = (a_{m1}, a_{m2}, \ldots, a_{mn})^\mathsf{T}$. A sample value $\boldsymbol{v}$ of $\boldsymbol{V}$ is observed and the problem is to make a MAP decision, denoted $\tilde{\boldsymbol{U}}$, about $\boldsymbol{U}$.

Initially assume the binary antipodal case where $\boldsymbol{a}_1 = -\boldsymbol{a}_0$. For notational simplicity, let $\boldsymbol{a}_0$ be denoted as $\boldsymbol{a} = (a_1, a_2, \ldots, a_n)^\mathsf{T}$. Thus the two hypotheses are $\boldsymbol{U} = \boldsymbol{a}$ and $\boldsymbol{U} = -\boldsymbol{a}$ and the observation is either $\boldsymbol{a} + \boldsymbol{Z}$ or $-\boldsymbol{a} + \boldsymbol{Z}$. The likelihoods are then given by

$$\begin{aligned} f_{\boldsymbol{V}|U}(\boldsymbol{v} \mid \boldsymbol{a}) &= \frac{1}{(\pi N_0)^{n/2}} \exp \sum_{k=1}^{n} \frac{-(v_k - a_k)^2}{N_0} = \frac{1}{(\pi N_0)^{n/2}} \exp\left(\frac{-\|\boldsymbol{v} - \boldsymbol{a}\|^2}{N_0}\right) \\ f_{\boldsymbol{V}|U}(\boldsymbol{v} \mid \text{-}a) &= \frac{1}{(\pi N_0)^{n/2}} \exp \sum_{k=1}^{n} \frac{-(v_k + a_k)^2}{N_0} = \frac{1}{(\pi N_0)^{n/2}} \exp\left(\frac{-\|\boldsymbol{v} + \boldsymbol{a}\|^2}{N_0}\right). \end{aligned}$$

The log likelihood ratio is thus given by

$$\mathrm{LLR}(\boldsymbol{v}) = \frac{-\|\boldsymbol{v} - \boldsymbol{a}\|^2 + \|\boldsymbol{v} + \boldsymbol{a}\|^2}{N_0} = \frac{4\langle \boldsymbol{v},\, \boldsymbol{a}\rangle}{N_0}, \tag{8.21}$$

and the MAP test is

$$\mathrm{LLR}(\boldsymbol{v}) = \frac{4\langle \boldsymbol{v},\, \boldsymbol{a}\rangle}{N_0} \underset{\tilde{\boldsymbol{U}}=-\boldsymbol{a}}{\overset{\tilde{\boldsymbol{U}}=\boldsymbol{a}}{\gtrless}} \ln \frac{p_1}{p_0} = \ln(\eta).$$

This can be restated as

$$\frac{\langle \boldsymbol{v}, \boldsymbol{a}\rangle}{\|\boldsymbol{a}\|} \underset{\tilde{\boldsymbol{U}}=-\boldsymbol{a}}{\overset{\tilde{\boldsymbol{U}}=\boldsymbol{a}}{\gtrless}} \frac{N_0 \ln(\eta)}{4\|\boldsymbol{a}\|}. \tag{8.22}$$

The projection of the observation $\boldsymbol{v}$ onto the signal $\boldsymbol{a}$ is $\frac{\langle \boldsymbol{v},\boldsymbol{a}\rangle}{\|\boldsymbol{a}\|}\frac{\boldsymbol{a}}{\|\boldsymbol{a}\|}$. Thus the left side of (8.22) is the component of $\boldsymbol{v}$ in the direction of $\boldsymbol{a}$, thus showing that the decision is based solely on that component of $\boldsymbol{v}$. This result is rather natural; the noise is independent in different orthogonal directions, and only the noise in the direction of the signal should be relevant in detecting the signal.

The geometry of the situation is particularly clear in the ML case (see Figure 8.4). The noise is spherically symmetric around the origin, and the likelihoods depend only on the distance from the origin. The ML detection rule is then equivalent to choosing the hypothesis closest to the received point. The set of points equidistant from the two hypotheses, as illustrated in Figure 8.4, is the perpendicular bisector between them; this bisector is the set of $\boldsymbol{v}$ satisfying $\langle \boldsymbol{v}, \boldsymbol{a}\rangle = 0$. The set of points closer to $\boldsymbol{a}$ is on the $\boldsymbol{a}$ side of this perpendicular bisector; it is determined by $\langle \boldsymbol{v}, \boldsymbol{a}\rangle > 0$ and is mapped into $\boldsymbol{a}$ by the ML rule. Similarly, the set of points closer to $-\boldsymbol{a}$ is

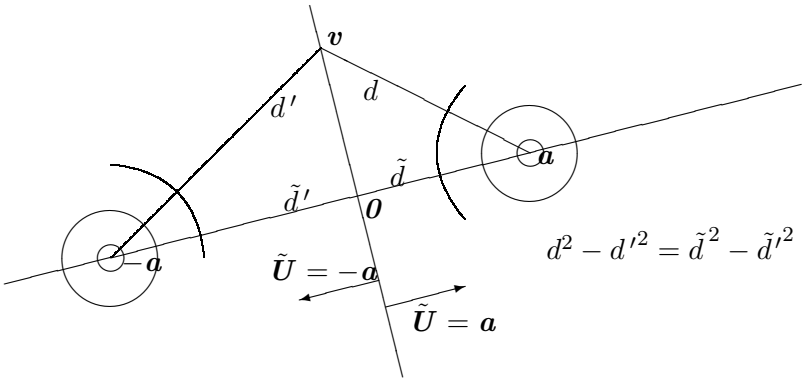

Figure 8.4: ML decision regions for binary signals in WGN. A vector $\boldsymbol{v}$ on the threshold boundary is shown. The distance from $\boldsymbol{v}$ to $\boldsymbol{a}$ is $d = \|\boldsymbol{v} - \boldsymbol{a}\|$. Similarly the distance to $-\boldsymbol{a}$ is $d' = \|\boldsymbol{v} + \boldsymbol{a}\|$. As shown algebraically in (8.21), any point at which $d^2 - d'^2 = 0$ is a point at which $\langle \boldsymbol{v}, \boldsymbol{a} \rangle = 0$, and thus at which the LLR is 0. Geometrically, from the Pythagorean theorem, however, $d^2 - d'^2 = \tilde{d}^2 - \tilde{d}'^2$, where $\tilde{d}$ and $\tilde{d}'$ are the distances from $\boldsymbol{a}$ and $-\boldsymbol{a}$ to the projection of $\boldsymbol{v}$ on the straight line generated by $\boldsymbol{a}$. This demonstrates geometrically why it is only the projection of $\boldsymbol{v}$ onto $\boldsymbol{a}$ that is relevant .

determined by $\langle \boldsymbol{v}, \boldsymbol{a} \rangle < 0$, and is mapped into $-\boldsymbol{a}$. In the general MAP case, the region mapped into $\boldsymbol{a}$ is again separated from the region mapped into $-\boldsymbol{a}$ by a perpendicular to $\boldsymbol{a}$, but in this case it is the perpendicular defined by $\langle \boldsymbol{v}, \boldsymbol{a} \rangle = N_0 \ln(\eta)/4$.

Another way of interpreting (8.22) is to view it in a different co-ordinate system. That is, choose $\boldsymbol{\phi}_1 = \boldsymbol{a}/\|\boldsymbol{a}\|$ as one element of an orthonormal basis for the n-vectors and choose another $n-1$ orthonormal vectors by the Gram-Schmidt procedure. In this new co-ordinate system $\boldsymbol{v}$ can be expressed as $(v'_1, v'_2, \ldots, v'_n)^{\mathsf{T}}$, where for $1 \le k \le n$, $v'_k = \langle \boldsymbol{v}, \phi_k \rangle$. Since $\langle \boldsymbol{v}, \boldsymbol{a} \rangle = \|\boldsymbol{a}\| \langle \boldsymbol{v}, \boldsymbol{\phi}_1 \rangle = \|\boldsymbol{a}\| v'_1$, the left side of (8.22) is simply v'_1, *i.e.*, the size of the projection of $\boldsymbol{v}$ onto $\boldsymbol{a}$. Thus (8.22) becomes

$$v'_1 \underset{<_{\tilde{U}=1}}{\overset{\geq^{\tilde{U}=0}}{}} \frac{N_0 \ln(\eta)}{4\|\boldsymbol{a}\|}.$$

This is the same as the one-dimensional MAP test in (8.14). In other words, the n-dimensional problem is the same as the one dimensional problem when the appropriate co-ordinate system is chosen. Actually, the derivation of (8.22) has shown something more, namely that v'_1 is a sufficient statistic. The components $v'_2, \ldots, v'_n$, which contain only noise, cancel out in (8.21) if (8.21) is expressed in the new co-ordinate system. The fact that the co-ordinates of $\boldsymbol{v}$ in directions orthogonal to the signal do not affect the LLR is sometimes called the *theorem of irrelevance*. A generalized form of this theorem is stated later as Theorem 8.4.2.

Some additional insight into (8.22) (in the original co-ordinate system) can be gained by writing $\langle \boldsymbol{v}, \boldsymbol{a} \rangle$ as $\sum_k v_k a_k$. This says that the MAP test weights each co-ordinate linearly by the amount of signal in that co-ordinate. This is not surprising, since the two hypotheses are separated more by the larger components of $\boldsymbol{a}$ than by the smaller.

Next consider the error probability conditional on $\boldsymbol{U} = -\boldsymbol{a}$. Given $\boldsymbol{U} = -\boldsymbol{a}$, $\boldsymbol{V} = -\boldsymbol{a} + \boldsymbol{Z}$, and

thus

$$\frac{\langle \boldsymbol{V}, \boldsymbol{a}\rangle}{\|\boldsymbol{a}\|} = -\|\boldsymbol{a}\| + \langle \boldsymbol{Z}, \boldsymbol{\phi}_1\rangle.$$

The mean and variance of this, given $\boldsymbol{U}=-\boldsymbol{a}$, are $-\|a\|$ and $N_0/2$. Thus, $\langle \boldsymbol{V}, \boldsymbol{a}\rangle/\|\boldsymbol{a}\|$ is $\mathcal{N}(-\|\boldsymbol{a}\|, N_0/2)$. From (8.22), the probability of error, given $\boldsymbol{U}=-\boldsymbol{a}$, is the probability that $\mathcal{N}(-\|\boldsymbol{a}\|, N_0/2)$ exceeds $N_0 \ln(\eta)/(4\,\|\boldsymbol{a}\|)$. This is the probability that Z is greater than $\|a\| + N_0 \ln(\eta)/(4\,\|\boldsymbol{a}\|)$. Normalizing as in subsection 8.3.1,

$$\Pr\{e \mid \boldsymbol{U}=-\boldsymbol{a}\} = Q\left(\sqrt{\frac{2\|\boldsymbol{a}\|^2}{N_0}} + \frac{\ln \eta}{2\sqrt{2\|\boldsymbol{a}\|^2/N_0}}\right). \tag{8.23}$$

By the same argument,

$$\Pr\{e \mid \boldsymbol{U}=\boldsymbol{a}\} = Q\left(\sqrt{\frac{2\|\boldsymbol{a}\|^2}{N_0}} - \frac{\ln \eta}{2\sqrt{2\|\boldsymbol{a}\|^2/N_0}}\right). \tag{8.24}$$

It can be seen that this is the same answer as given by (8.15) and (8.16) when the problem is converted to a coordinate system where $\boldsymbol{a}$ is collinear with a coordinate vector. The energy per bit is $E_b = \|\boldsymbol{a}\|^2$, so that (8.17) and (8.18) follow as before. This is not surprising, of course, since this vector decision problem is identical to the scalar problem when the appropriate basis is used.

For most communication problems, the a priori probabilities are assumed to be equal so that $\eta = 1$. Thus, as in (8.19),

$$\Pr\{e\} = Q\left(\sqrt{\frac{2E_b}{N_0}}\right). \tag{8.25}$$

This gives us a useful sanity check - the probability of error does not depend on the orthonormal coordinate basis.

Now suppose that the binary hypotheses correspond to non-antipodal vector signals, say $\boldsymbol{b}_0$ and $\boldsymbol{b}_1$. We analyze this in the same way as the scalar case. Namely, let $\boldsymbol{c} = (\boldsymbol{b}_0 + \boldsymbol{b}_1)/2$ and $\boldsymbol{a} = \boldsymbol{b}_0 - \boldsymbol{c}$. Then the two signals are $\boldsymbol{b}_0 = \boldsymbol{a} + \boldsymbol{c}$ and $\boldsymbol{b}_1 = -\boldsymbol{a} + \boldsymbol{c}$. As before, converting the observation $\boldsymbol{V}$ to $\tilde{\boldsymbol{V}} = \boldsymbol{V} - \boldsymbol{c}$ shifts the midpoint and converts the problem back to the antipodal case. The error probability depends only on the distance $2\|\boldsymbol{a}\|$ between the signals and is given by (8.23) and (8.24). The energy per bit is again different, and assuming equiprobable input vectors, the energy per bit is $E_b = \|\boldsymbol{a}\|^2 + \|\boldsymbol{c}\|^2$. Thus the center point $\boldsymbol{c}$ contributes to the energy, but not to the error probability.

It is often more convenient, especially when generalizing to $M > 2$ hypotheses, to express the LLR for the non-antipodal case directly in terms of $\boldsymbol{b}_0$ and $\boldsymbol{b}_1$. Using (8.21) for the shifted vector $\tilde{\boldsymbol{V}}$, the LLR can be expressed as

$$\mathrm{LLR}(\boldsymbol{v}) = \frac{-\|\boldsymbol{v} - \boldsymbol{b}_0\|^2 + \|\boldsymbol{v} - \boldsymbol{b}_1^2\|^2}{N_0}. \tag{8.26}$$

For ML detection, this is simply the minimum distance rule, and for MAP, the interpretation is the same as for the antipodal case.

8.3.4 Detection for binary complex vectors in WGN

Next consider the complex vector version of the same problem. Assume the observation is a complex random n-vector $\boldsymbol{V} = \boldsymbol{U} + \boldsymbol{Z}$. The noise, $\boldsymbol{Z} = (Z_1, \ldots, Z_n)^\mathsf{T}$, is a complex random vector of n zero-mean complex iid Gaussian rv's with iid real and imaginary parts, each $\mathcal{N}(0, N_0/2)$. Thus each Z_k is circularly symmetric and denoted by $\mathcal{CN}(0, N_0)$. The input $\boldsymbol{U}$ is independent of $\boldsymbol{Z}$ and binary, taking on value $\boldsymbol{a}$ with probability p_0 and $-\boldsymbol{a}$ with probability p_1 where $\boldsymbol{a} = (a_1, \ldots, a_n)^\mathsf{T}$ is an arbitrary complex n-vector.

This problem can be reduced to that of the last subsection by letting $\boldsymbol{Z'}$ be the $2n$ dimensional real random vector with components $\Re(Z_k)$ and $\Im(Z_k)$ for $1 \le k \le n$. Similarly let $\boldsymbol{a'}$ be the $2n$ dimensional real vector with components $\Re(a_k)$ and $\Im(a_k)$ for $1 \le k \le n$ and let $\boldsymbol{U'}$ be the real random vector that takes on values $\boldsymbol{a'}$ or $-\boldsymbol{a'}$. Finally, let $\boldsymbol{V'} = \boldsymbol{U'} + \boldsymbol{Z'}$.

Recalling that probability densities for complex random variables or vectors are equal to the joint probability densities for the real and imaginary parts,

$$\begin{aligned} f_{V|U}(\boldsymbol{v}|\boldsymbol{a}) &= f_{V'|U'}(\boldsymbol{v'}|\boldsymbol{a'}) = \frac{1}{(\pi N_0)^n} \exp \sum_{k=1}^{n} \frac{-\Re(v_k - a_k)^2 - \Im(v_k - a_k)^2}{N_0} \\ f_{V|U}(\boldsymbol{v}|-\boldsymbol{a}) &= f_{V'|U'}(\boldsymbol{v'}|-\boldsymbol{a'}) = \frac{1}{(\pi N_0)^n} \exp \sum_{k=1}^{n} \frac{-\Re(v_k + a_k)^2 - \Im(v_k + a_k)^2}{N_0}. \end{aligned}$$

The LLR is then

$$\mathrm{LLR}(\boldsymbol{v}) = \frac{-\|\boldsymbol{v} - \boldsymbol{a}\|^2 + \|\boldsymbol{v} + \boldsymbol{a}\|^2}{N_0}. \tag{8.27}$$

Note that

$$\|\boldsymbol{v} - \boldsymbol{a}\|^2 = \|\boldsymbol{v}\|^2 - \langle \boldsymbol{v}, \boldsymbol{a}\rangle - \langle \boldsymbol{a}, \boldsymbol{v}\rangle + \|\boldsymbol{a}\|^2 = \|\boldsymbol{v}\|^2 - 2\Re[\langle \boldsymbol{v}, \boldsymbol{a}\rangle] + \|\boldsymbol{a}\|^2$$

Using this and the analagous expression for $\|\boldsymbol{v} + \boldsymbol{a}\|^2$, (8.27) becomes

$$\mathrm{LLR}(\boldsymbol{v}) = \frac{4\Re[\langle \boldsymbol{v}, \boldsymbol{a}\rangle]}{N_0} \tag{8.28}$$

The MAP test can now be stated as

$$\frac{\Re[\langle \boldsymbol{v}, \boldsymbol{a}\rangle]}{\|\boldsymbol{a}\|} \underset{\tilde{U}=-\boldsymbol{a}}{\overset{\tilde{U}=\boldsymbol{a}}{\gtrless}} \frac{N_0 \ln(\eta)}{4\|\boldsymbol{a}\|}. \tag{8.29}$$

Note that the value of the LLR and the form of the MAP test are the same as the real vector case except for the real part of $\langle \boldsymbol{v}, \boldsymbol{a}\rangle$. The significance of this real part operation is now discussed.

In the n-dimensional complex vector space, $\langle \boldsymbol{v}, \boldsymbol{a}\rangle/\|\boldsymbol{a}\|$ is the complex value of the projection of $\boldsymbol{v}$ in the direction of $\boldsymbol{a}$. In order to understand this projection better, consider an orthonormal basis in which $\boldsymbol{a} = (1, 0, 0, \ldots, 0)^\mathsf{T}$. Then $\langle \boldsymbol{v}, \boldsymbol{a}\rangle/\|\boldsymbol{a}\| = v_1$. Thus $\Re(v_1) = \pm 1 + \Re(z_1)$ and $\Im(v_1) = \Im(z_1)$. Clearly, only $\Re(v_1)$ is relevant to the binary decision. Using $\Re[\langle \boldsymbol{v}, \boldsymbol{a}\rangle/\|\boldsymbol{a}\|]$ in (8.29) is simply the general way of stating this elementary idea. If the complex plane is viewed as a 2-dimensional real space, then taking the real part of $\langle \boldsymbol{v}, \boldsymbol{a}\rangle$ is equivalent to taking the further projection of this two dimensional real vector in the direction of $\boldsymbol{a}$ (see Exercise 8.12).

The other results and interpretations of the last subsection remain unchanged. In particular, since $\|\boldsymbol{a}'\| = \|\boldsymbol{a}\|$, the error probability results are given by

$$\Pr\{e \mid \boldsymbol{U}=-\boldsymbol{a}\} = Q\left(\sqrt{\frac{2\|\boldsymbol{a}\|^2}{N_0}} + \frac{\ln\eta}{2\sqrt{2\|\boldsymbol{a}\|^2/N_0}}\right) \tag{8.30}$$

$$\Pr\{e \mid \boldsymbol{U}=\boldsymbol{a}\} = Q\left(\sqrt{\frac{2\|\boldsymbol{a}\|^2}{N_0}} - \frac{\ln\eta}{2\sqrt{2\|\boldsymbol{a}\|^2/N_0}}\right). \tag{8.31}$$

For the ML case, recognizing that $\|\boldsymbol{a}\|^2 = E_b$, we have the familiar result

$$\Pr\{e\} = Q\left(\sqrt{\frac{2E_b}{N_0}}\right). \tag{8.32}$$

Finally, for the non-antipodal case with hypotheses $\boldsymbol{b}_0$ and $\boldsymbol{b}_1$, the LLR is again given by (8.26).

8.3.5 Detection of binary antipodal waveforms in WGN

This section extends the vector case of the previous two subsections to the waveform case. It will be instructive to do this simultaneously for both passband real random processes and baseband complex random processes. Let $U(t)$ be the baseband modulated waveform. As before, the situation is simplified by transmitting a single bit rather than a sequence of bits, so for some arbitrary, perhaps complex, baseband waveform $a(t)$, the binary input 0 is mapped into $U(t) = a(t)$ and 1 is mapped into $U(t) = -a(t)$; the a priori probabilities are denoted by p_0 and p_1. Let $\{\theta_k(t); k \in \mathbb{Z}\}$ be a complex orthonormal expansion covering the baseband region of interest, and let $a(t) = \sum_k a_k\theta_k(t)$.

Assume $U(t) = \pm a(t)$ is modulated onto a carrier f_c larger than the baseband bandwidth. The resulting bandpass waveform is denoted $X(t) = \pm b(t)$ where, from Section 7.8, the modulated form of $a(t)$, denoted $b(t)$, can be represented as

$$b(t) = \sum_k b_{k,1}\psi_{k,1}(t) + b_{k,2}\psi_{k,2}(t)$$

where

$$\begin{aligned} b_{k,1} = \Re(a_k); \qquad \psi_{k,1}(t) &= \Re\{2\theta_k(t)\,\exp[2\pi i f_c t]\}; \\ b_{k,2} = \Im(a_k); \qquad \psi_{k,2}(t) &= -\Im\{2\theta_k(t)\,\exp[2\pi i f_c t]\}. \end{aligned}$$

From Theorem 6.6.1, the set of waveforms $\{\psi_{k,j}(t); k \in \mathbb{Z}, j \in \{1,2\}\}$ are orthogonal, each with energy 2. Let $\{\phi_m(t); m \in \mathbb{Z}\}$ be a set of orthogonal functions, each of energy 2 and each orthogonal to each of the $\psi_{k,j}(t)$. Assume that $\{\phi_m(t); m \in \mathbb{Z}\}$, together with the $\psi_{k,j}(t)$, span $\mathcal{L}_2$.

The noise $W(t)$, by assumption, is WGN. It can be represented as

$$W(t) = \sum_k \left(Z_{k,1}\psi_{k,1}(t) + Z_{k,2}\psi_{k,2}(t)\right) + \sum_m W_m\phi_m(t),$$

where $\{Z_{k,m}; k \in \mathbb{Z}, m \in \{1,2\}\}$ is the set of scaled linear functionals of the noise in the $\mathcal{L}_2$ vector space spanned by the $\psi_{k,m}(t)$, and $\{W_m; m \in \mathbb{Z}\}$ is the set of linear functionals of the noise in

the orthogonal complement of the space. As will be seen shortly, the joint distribution of the W_m makes no difference in choosing between $a(t)$ and $-a(t)$, so long as the W_m are independent of the $Z_{k,j}$ and the transmitted binary digit. The observed random process at passband is then $Y(t) = X(t) + W(t)$,

$$Y(t) = \sum_k [Y_{k,1}\psi_{k,1}(t) + Y_{k,2}\psi_{k,2}(t)] + \sum_m W_m\phi_m(t) \qquad \text{where}$$

$$Y_{k,1} = (\pm b_{k,1} + Z_{k,1})\,; \qquad Y_{k,2} = (\pm b_{k,2} + Z_{k,2})\,.$$

First assume that a finite number n of orthonormal functions are used to represent $a(t)$. This is no loss of generality, since the single function $a(t)/\|a(t)\|$ would be sufficient. Suppose also, initially, that only a finite set, say $W_1, \ldots, W_\ell$ of the orthogonal noise functionals are observed. Assume also that the noise variables, $Z_{k,j}$ and W_m are independent and each[4] $\mathcal{N}(0, N_0/2)$. Then the likelihoods are given by

$$f_{\boldsymbol{Y}|\boldsymbol{X}}(\boldsymbol{y} \mid \boldsymbol{b}) = \frac{1}{(\pi N_0)^n} \exp\left[\sum_{k=1}^{n}\sum_{j=1}^{2} \frac{-(y_{k,j} - b_{k,j})^2}{N_0} + \sum_{m=1}^{\ell} \frac{-w_m^2}{N_0}\right],$$

$$f_{\boldsymbol{Y}|\boldsymbol{X}}(\boldsymbol{y} \mid -\boldsymbol{b}) = \frac{1}{(\pi N_0)^n} \exp\left[\sum_{k=1}^{n}\sum_{j=1}^{2} \frac{-(y_{k,j} + b_{k,j})^2}{N_0} + \sum_{m=1}^{\ell} \frac{-w_m^2}{N_0}\right].$$

The log likelihood ratio is thus given by

$$\begin{aligned}
\mathrm{LLR}(\boldsymbol{y}) &= \sum_{k=1}^{n}\sum_{j=1}^{2} \frac{-(y_{k,j} - b_{k,j})^2 + (y_{k,j} + b_{k,j})^2}{N_0} \\
&= \frac{-\|\boldsymbol{y} - \boldsymbol{b}\|^2 + \|\boldsymbol{y} + \boldsymbol{b}\|^2}{N_0} \qquad (8.33) \\
&= \sum_{k=1}^{n}\sum_{j=1}^{2} \frac{4y_{k,j}b_{k,j}}{N_0} = \frac{4\langle \boldsymbol{y},\, \boldsymbol{b}\rangle}{N_0}. \qquad (8.34)
\end{aligned}$$

and the MAP test is

$$\langle \boldsymbol{y},\, \boldsymbol{b}\rangle \underset{<}{\overset{\geq}{}}{}^{\tilde{\boldsymbol{X}}=\boldsymbol{b}}_{\tilde{\boldsymbol{X}}=-\boldsymbol{b}} \frac{N_0 \ln(\eta)}{4\|\boldsymbol{b}\|}.$$

This is the same as the real vector case analyzed in Subsection 8.3.3. In fact, the only difference is that the observation here includes noise in the degrees of freedom orthogonal to the range of interest, and the derivation of the LLR shows clearly why these noise variables do not appear in the LLR. In fact, the number ℓ of rv's W_m can be taken to be arbitrarily large, and they can have any joint density. So long as they are independent of the $Z_{k,j}$ (and of $X(t)$), they cancel out in the LLR. In other words, WGN is noise that is iid Gaussian over a large enough space to represent the signal, and is independent of the signal and noise elsewhere.

[4]Recall that $N_0/2$ is the noise variance using the same scale as used for the signal waveform. Since the input energy is measured at baseband, the noise is also. At passband, the signal energy is scaled up by a factor of 2, and the noise energy is similarly scaled.

The argument above leading to (8.33) and (8.34) is not entirely satisfying mathematically, since it is based on the slightly vague notion of the signal space of interest, but in fact it is just this feature that makes it useful in practice, since physical noise characteristics do change over large changes in time and frequency.

The inner product in (8.34) is the inner product over the $\mathcal{L}_2$ space of real sequences. Since these sequences are coefficients in an orthogonal (rather than orthonormal) expansion, the conversion to an inner product over the corresonding functions (see Exercise 8.5) is given by

$$\sum_{k,j} y_{k,j} b_{k,j} = \frac{1}{2} \int y(t) b(t)\, dt. \tag{8.35}$$

This shows that the LLR is independent of the basis, and that this waveform problem reduces to the single dimensional problem if $b(t)$ is a multiple of one of the basis functions. Also, if a countably infinite basis for the signal space of interest is used, (8.35) is still valid.

Next consider what happens when $Y(t) = \pm b(t) + W(t)$ is demodulated to the baseband waveform $V(t)$. The component $\sum_m W_m(t)$ of $Y(t)$ extends to frequencies outside the passband, and thus $Y(t)$ is filtered before demodulation, preventing an aliasing like effect between $\sum_m W_m(t)$ and the signal part of $Y(t)$ (see Exercise 6.11). Assuming that this filtering does not affect $b(t)$, $b(t)$ maps back into $a(t) = \sum_k a_k \theta_k(t)$ where $a_k = b_{k,1} + i b_{k,2}$. Similarly $W(t)$ maps into

$$Z(t) = \sum_k Z_k \theta_k(t) + Z_\perp(t)$$

where $Z_k = Z_{k,1} + iZ_{k,2}$ and $Z_\perp(t)$ is the result of filtering and frequency demodulation on $\sum_m W_m \phi_m(t)$. The received baseband complex process is then

$$V(t) = \sum_k V_k \theta_k(t) + Z_\perp(t) \qquad \text{where} \quad V_k = \pm a_k + Z_k. \tag{8.36}$$

By the filtering assumption above, the sample functions of $Z_\perp(t)$ are orthogonal to the space spanned by the $\theta_k(t)$ and thus the sequence $\{V_k; k \in \mathbb{Z}\}$ is determined from $V(t)$. Since $V_k = Y_{k,1} + iY_{k,2}$, the sample value $\mathrm{LLR}(\boldsymbol{y})$ in (8.34) is determined as follows by the sample values of $\{v_k; k \in \mathbb{Z}\}$,

$$\mathrm{LLR}(\boldsymbol{y}) = \frac{4\langle \boldsymbol{y},\, \boldsymbol{b}\rangle}{N_0} = \frac{4\Re[\langle \boldsymbol{v}, \boldsymbol{a}\rangle]}{N_0}. \tag{8.37}$$

Thus $\{v_k; k \in \mathbb{Z}\}$ is a sufficient statistic for $y(t)$, and thus the MAP test based on $y(t)$ can be done using $v(t)$. Now an implementation that first finds the sample function $v(t)$ from $y(t)$ and then does a MAP test on $v(t)$ is simply a particular kind of test on $y(t)$, and thus cannot achieve a smaller error probability than the MAP test on $\boldsymbol{y}$. Finally, since $\{v_k; k \in \mathbb{Z}\}$ is a sufficient statistic for $y(t)$, it is also a sufficient statistic for $v(t)$ and thus the orthogonal noise $Z_\perp(t)$ is irrelevant.

Note that the LLR in (8.37) is the same as the complex vector result in (8.28). One could repeat the argument there, adding in an orthogonal expansion for $Z_\perp(t)$ to verify the argument that $Z_\perp(t)$ is irrelevant. Since $Z_\perp(t)$ could take on virtually any form, the argument above, based on the fact that $Z_\perp(t)$ is a function of $\sum_m W_m \phi_M(t)$, which is independent of the signal and noise in the signal space, is more insightful.

To summarize this subsection, the detection of a single bit sent by generating antipodal signals at baseband and modulating to passband has been analyzed. After adding WGN, the received waveform is demodulated to baseband and then the single bit is detected. The MAP detector at passband is a threshold test on $\int y(t)b(t)\,dt$. This is equivalent to a threshold test at baseband on $\Re[\int v(t)a^*(t)\,dt]$. This shows that no loss of optimality occurs by demodulating to baseband and also shows that detection can be done either at passband or at baseband. In the passband case, the result is an immediate extension of binary detection for real vectors, and at baseband, it is an immediate extension of binary detection of complex vectors.

The results of this section can now be interpreted in terms of PAM and QAM, while still assuming a "one-shot" system in which only one binary digit is actually sent. Recall that for both PAM and QAM modulation, the modulation pulse $p(t)$ is orthogonal to its T-spaced time shifts if $|\hat{p}(f)|^2$ satisfies the Nyquist criterion. Thus, if the corresponding received baseband waveform is passed through a matched filter (a filter with impulse response $p^*(t)$) and sampled at times kT, the received samples will have no intersymbol interference. For a single bit transmitted at discrete time 0, $u(t) = \pm a(t) = ap(t)$. The output of the matched filter at receiver time 0 is then

$$\int v(t)p^*(t)\,dt = \frac{\Re[\langle \boldsymbol{v}, \boldsymbol{a}\rangle]}{a},$$

which is a scaled version of the LLR. Thus the receiver from Chapter 6 that avoids intersymbol interference also calculates the LLR, from which a threshold test yields the MAP detection.

The next section shows that this continues to provide MAP tests on successive signals. It should be noted also that sampling the output of the matched filter at time 0 yields the MAP test whether or not $p(t)$ has been chosen to avoid intersymbol interference.

It is important to note that the performance of binary antipodal communication in WGN depends only on the energy of the transmitted waveform. With ML detection, the error probability is the familiar expression $Q(\frac{2E_b}{N_0})$ where $E_b = \int |a(t)|^2\,dt$ and the variance of the noise in each real degree of freedom in the region of interest is $N_0/2$.

This completes the analysis of binary detection in WGN, including the relationship between the vector case and waveform case and that between complex waveforms or vectors at basebande and real waveforms or vectors at passband.

The following sections analyze M-ary detection. The relationships between vector and waveform and between real and complex is the same as above, so the following sections each assume whichever of these cases is most instructive without further discussion of these relationships.

8.4 M-ary detection and sequence detection

The analysis in the previous section was limited in several ways. First, only binary signal sets were considered, and second, only the 'one-shot' problem where a single bit rather than a sequence of bits was considered. In this section, M-ary signal sets for arbitrary M will be considered, and this will then be used to study the transmission of a sequence of signals and to study arbitrary modulation schemes.

8.4.1 M-ary detection

Going from binary to M-ary hypothesis testing is a simple extension. To be specific, this will be analyzed for the complex random vector case. Let the observation be a complex random n-vector $\boldsymbol{V}$ and let the complex random n-vector $\boldsymbol{U}$ to be detected take on a value from the set $\{\boldsymbol{a}_0, \ldots, \boldsymbol{a}_{M-1}\}$ with a priori probabilities $p_0, \ldots, p_{M-1}$. Denote the a posteriori probabilities by $p_{\boldsymbol{U}|\boldsymbol{V}}(\boldsymbol{a}_m|\boldsymbol{v})$. The MAP rule (see Section 8.1) then chooses $\tilde{\boldsymbol{U}}(\boldsymbol{v}) = \arg\max_m p_{\boldsymbol{U}|\boldsymbol{V}}(\boldsymbol{a}_m|\boldsymbol{v})$. Assuming that the likelihoods can be represented as probability densities $f_{\boldsymbol{V}|\boldsymbol{U}}$, the MAP rule can be expressed as

$$\tilde{\boldsymbol{U}}(\boldsymbol{v}) = \arg\max_m p_m f_{\boldsymbol{V}|\boldsymbol{U}}(\boldsymbol{v}|\boldsymbol{a}_m).$$

Usually, the simplest approach to this M-ary rule is to consider multiple binary hypothesis testing problems. That is, $\tilde{\boldsymbol{U}}(v)$ is that $\boldsymbol{a}_m$ for which

$$\Lambda_{m,m'}(\boldsymbol{v}) = \frac{f_{\boldsymbol{V}|\boldsymbol{U}}(\boldsymbol{v}|\boldsymbol{a}_m)}{f_{\boldsymbol{V}|\boldsymbol{U}}(v|\boldsymbol{a}_{m'})} \geq \frac{p_{m'}}{p_m}$$

for all m'. In the case of ties, it makes no difference which of the maximizing hypotheses are chosen.

For the complex vector additive WGN case, the observation is $\boldsymbol{V} = \boldsymbol{U} + \boldsymbol{Z}$ where $\boldsymbol{Z}$ is complex Gaussian noise with iid real and imaginary components. As derived in (8.27), the log likelihood ratio (LLR) between each pair of hypotheses $\boldsymbol{a}_m$ and $\boldsymbol{a}_{m'}$ is given by

$$\mathrm{LLR}_{m,m'}(\boldsymbol{v}) = \frac{-\|\boldsymbol{v} - \boldsymbol{a}_m\|^2 + \|\boldsymbol{v} - \boldsymbol{a}_{m'}^2\|^2}{N_0}. \tag{8.38}$$

Thus each binary test separates the observation space[5] into two regions separated by the perpendicular bisector between the two points. With M hypotheses, the space is separated into the Voronoi regions of points closest to each of the signals (hypotheses) (see Figure 8.5). If the a priori probabilities are unequal, then these perpendicular bisectors are shifted, remaining perpendicular to the axis joining the two signals, but no longer being bisectors.

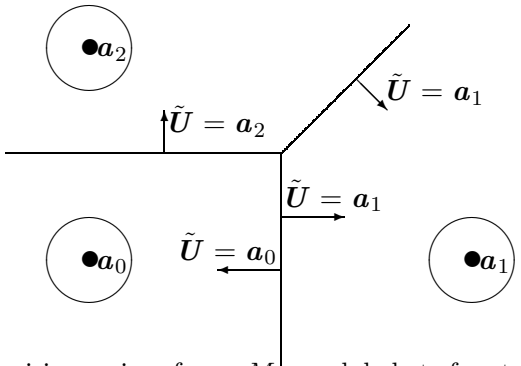

Figure 8.5: Decision regions for an M-ary alphabet of vector signals in iid Gaussian noise. For ML detection, the decision regions are Voronoi regions, *i.e.*, regions separated by perpendicular bisectors between the signal points.

[5]For an n dimensional complex vector space, it is simplest to view the observation space as the corresponding $2n$ dimensional real vector space.

The probability that noise carries the observation across one of these perpendicular bisectors is given in (8.29). The only new problem that arises with M-ary hypothesis testing is that the error probability, given $\boldsymbol{U} = m$, is the union of $M-1$ events, namely crossing the corresponding perpendicular to each other point. This can be found exactly by integrating over the n dimensional vector space, but is usually upper bounded and approximated by the union bound, where the probability of crossing each perpendicular is summed over the $M-1$ incorrect hypotheses. This is usually a good approximation (if M is not too large), because the Gaussian density decreases so rapidly with distance; thus in the ML case, most errors are made when observations occur roughly half way between the transmitted and the detected signal point.

8.4.2 Successive transmissions of QAM signals in WGN

This subsection extends the 'single-shot' analysis of detection for QAM and PAM in the presence of WGN to the case in which an n-tuple of successive independent symbols are transmitted. We shall find that under many conditions, both the detection rule and the corresponding probability of symbol error can be analyzed by looking at one symbol at a time.

First consider a QAM modulation system using a modulation pulse $p(t)$. Assume that $p(t)$ has unit energy and is orthonormal to its T-spaced shifts $\{p(t-kT);\, k \in \mathbb{Z}\}$, *i.e.*, that $\{p(t-kT);\, k \in \mathbb{Z}\}$ is a set of orthonormal functions. Let $\mathcal{A} = \{a_0, \ldots, a_{M-1}\}$ be the alphabet of complex input signals and denote the input waveform over an arbitrary n-tuple of successive input signals as

$$u(t) = \sum_{k=1}^{n} u_k p(t - kT),$$

where each u_k is a selection from the input alphabet $\mathcal{A}$.

Let $\{\phi_k(t);\, k \geq 1\}$ be an orthonormal basis of complex $\mathcal{L}_2$ waveforms such that the first n waveforms in that basis are given by $\phi_k(t) = p(t - kT)$, $1 \leq k \leq n$. The received baseband waveform is then

$$V(t) = \sum_{k=1}^{\infty} V_k \phi_k(t) = \sum_{k=1}^{n} (u_k + Z_k) p(t - kT) + \sum_{k>n} Z_k \phi_k(t). \tag{8.39}$$

We now compare two different detection schemes. In the first, a single ML decision between the M^n hypotheses for all possible joint values of $U_1, \ldots, U_n$ is made based on $V(t)$. In the second scheme, for each $k, 1 \leq k \leq n$, an ML decision between the M possible hypotheses $a_0 \ldots, a_{M-1}$ is made for input U_k based on the observation $V(t)$. Thus in this scheme, n separate M-ary decisions are made, one for each of the n successive inputs.

For the first alternative, each hypothesis corresponds to an n dimensional vector of inputs, $\boldsymbol{u} = (u_1, \ldots, u_n)^{\mathsf{T}}$. As in Subsection 8.3.5, the sample value $v(t) = \sum_k v_k \phi_k(t)$ of the received waveform can be taken as an ℓ-tuple $\boldsymbol{v} = (v_1, v_2, \ldots, v_\ell)^{\mathsf{T}}$ with $\ell \geq n$. The likelihood of $\boldsymbol{v}$ conditional on $\boldsymbol{u}$ is then given by

$$f_{V|U}(\boldsymbol{v}|\boldsymbol{u}) = \prod_{k=1}^{n} f_Z(v_k - u_k) \prod_{k=n+1}^{\ell} f_Z(v_k).$$

For any two hypotheses, say $\boldsymbol{u} = (u_1, \ldots, u_n)^{\mathsf{T}}$ and $\boldsymbol{u}' = (u'_1, \ldots, u'_n)^{\mathsf{T}}$, the likelihood ratio and

LLR are

$$\Lambda_{\boldsymbol{u},\boldsymbol{u}'}(\boldsymbol{v}) = \prod_{k=1}^{n} \frac{f_Z(v_k - u_k)}{f_Z(v_k - u'_k)} \tag{8.40}$$

$$\mathrm{LLR}_{\boldsymbol{u},\boldsymbol{u}'}(\boldsymbol{v}) = \frac{-\|\boldsymbol{v}-\boldsymbol{u}\|^2 + \|\boldsymbol{v}-\boldsymbol{u}'\|^2}{N_0}. \tag{8.41}$$

Note that for each $k > n$, v_k does not appear in this likelihood ratio. Thus this likelihood ratio is still valid[6] in the limit $\ell \to \infty$, but the only relevant terms in the decision are $v_1, \ldots, v_n$. Therefore let $\boldsymbol{v} = (v_1, \ldots, v_n)^\mathsf{T}$ in what follows. From (8.41), this likelihood ratio is positive if and only if $\|\boldsymbol{v}-\boldsymbol{u}\| < \|\boldsymbol{v}-\boldsymbol{u}'\|$. The conclusion is that for M^n-ary detection, done jointly on $u_1, \ldots, u_n$, the ML decision is the vector $\boldsymbol{u}$ that minimizes the distance $\|\boldsymbol{v}-\boldsymbol{u}\|$.

Consider how to minimize $\|\boldsymbol{v}-\boldsymbol{u}\|$. Note that

$$\|\boldsymbol{v}-\boldsymbol{u}\|^2 = \sum_{k=1}^{n} (v_k - u_k)^2. \tag{8.42}$$

Suppose that $\tilde{\boldsymbol{u}} = (\tilde{u}_1, \ldots, \tilde{u}_n)^\mathsf{T}$ minimizes this sum. Then for each k, $\tilde{u}_k$ minimizes $(v_k - u_k)^2$ over the M choices for u_k (otherwise some $a_m \neq \tilde{u}_k$ could be substituted for $\tilde{u}_k$ to reduce $(v_k - u_k)^2$ and therefore reduce the sum in (8.42)). Thus the ML sequence detector with M^n hypotheses detects each U_k by minimizing $(v_k - u_k)^2$ over the M hypotheses for that U_k.

Next consider the second alternative above. For a given sample observation $\boldsymbol{v} = v_1, \ldots, v_\ell$ and a given k, $1 \leq k \leq n$, the likelihood of $\boldsymbol{v}$ conditional on $U_k = u_k$ is

$$f_{\boldsymbol{V}|U_k}(\boldsymbol{v}|u_k) = f_Z(v_k - u_k) \prod_{j\neq k, 1\leq j\leq n} f_{V_j}(v_j) \prod_{j=n+1}^{\ell} f_Z(v_j)$$

where $f_{V_j}(v_j) = \sum_m p_m f_{V_j|U_j}(v_j|a_m)$ is the marginal probability of V_j. The likelihood ratio of $\boldsymbol{v}$ between the hypotheses $U_k = a_m$ and $U_k = a_{m'}$ is then

$$\Lambda^{(k)}_{m,m'}(\boldsymbol{v}) = \frac{f_Z(v_k - a_m)}{f_Z(v_k - a_{m'})}$$

This is the familiar one-dimensional non-antipodal Gaussian detection problem, and the ML decision is to choose $\tilde{u}_k$ as the a_m closest to u_k. Thus, given the sample observation $v(t)$, the vector $(\tilde{u}_1, \ldots, \tilde{u}_n)^\mathsf{T}$ of individual M-ary ML detectors for each U_k is the same as the M^n-ary ML sequence detector for the sequence $\boldsymbol{U} = (U_1, \ldots, U_n)^\mathsf{T}$. Moreover, each of these detectors are equivalent to a vector of ML decisions on each U_k based solely on the observation V_k.

Summarizing, we have proved the following theorem:

Theorem 8.4.1. *Let $U(t) = \sum_{k=1}^{n} U_k p(t-kT)$ be a QAM (or PAM) baseband input to a WGN channel and assume that $\{p(t-nT); 1 \leq k \leq n\}$ is an orthonormal sequence. Then the M^n-ary ML decision on $\boldsymbol{U} = (U_1, \ldots U_n)^\mathsf{T}$ is equivalent to making separate M-ary ML decisions on each U_k, $1 \leq k \leq n$, where the decision on each U_k can be based either on the observation $v(t)$ or the observation of v_k.*

[6] In fact, these final $\ell - n$ components do not have to be independent or equally distributed, they simply must be independent of the signals and noise for $1 \leq k \leq n$.

Note that the theorem states that the same decision is made for both sequence detection and separate detection for each signal. It *does not* say that the probability of an error within the sequence is the same as the error for a single signal. Letting P be the probability of error for a single signal, the probability of error for the sequence is $1-(1-P)^n$.

The theorem makes no assumptions about the probabilities of the successive inputs, although the use of ML detection would not minimize the probability of error if the inputs were not independent and equally likely. If coding is used between the n input signals, then not all of these M^n n-tuples are possible. In this case, ML detection on the *possible encoded* sequences (as opposed to all M^n sequences) is different from separate detection on each signal. As an example, if the transmitter always repeats each signal, with $u_1 = u_2$, $u_3 = u_4$, etc., then the detection of u_1 should be based on both v_1 and v_2. Similarly, the detection of u_3 should be based on v_3 and v_4, etc.

When coding is used, it is possible to make ML decisions on each signal separately, and then to use the coding constraints to correct errors in the detected sequence. These individual signal decisions are then called *hard decisions*. It is also possible, for each k, to save a sufficient statistic (such as v_k) for the decision on U_k. This is called a *soft decision* since it saves all the relevant information needed for an ML decision between the set of possible codewords. Since the ML decision between possible encoded sequences minimizes the error probability (assuming equi-probable codewords), soft decisions allow for smaller error probabilities than hard decisions.

Theorem 8.4.1 can be extended to MAP detection if the input signals are statistically independent of each other (see Exercise 8.15). One can see this intuitively by drawing the decision boundaries for the two-dimensional real case; these decision boundaries are then horizontal and vertical lines.

A nice way to interpret Theorem 8.4.1 is to observe that the detection of each signal U_k depends only on the corresponding received signal V_k; all other components of the received vector are irrelevant to the decision on U_k. The next subsection generalizes from QAM to arbitrary modulation schemes and also generalizes this notion of irrelevance.

8.4.3 Detection with arbitrary modulation schemes

The previous sections have concentrated on detection of PAM and QAM systems, using real hypotheses $\mathcal{A} = \{a_0, \ldots, a_{M-1}\}$ for PAM and complex hypotheses $\mathcal{A} = a_0, \ldots, a_{M-1}$ for QAM. In each case, a sequence $\{u_k; k \in \mathbb{Z}\}$ of signals from $\mathcal{A}$ is modulated into a baseband waveform $u(t) = \sum_k u_k p(t-kT)$. The PAM waveform is then either transmitted or first modulated to passband. The complex QAM waveform is necessarily modulated to a real passband waveform.

This is now generalized by considering a signal set $\mathcal{A}$ to be an M-ary alphabet, $\{\boldsymbol{a}_0, \ldots, \boldsymbol{a}_{M-1}\}$, of real n-tuples. Thus each $\boldsymbol{a}_m$ is an element of $\mathbb{R}^n$. The n components of the mth signal vector are denoted by $\boldsymbol{a}_m = (a_{m,1}, \ldots, a_{m,n})^\mathsf{T}$. The selected signal vector $\boldsymbol{a}_m$ is then modulated into a signal waveform $\boldsymbol{b}_m(t) = \sum_{k=1}^n a_{m,k}\phi_k(t)$ where $\{\phi_1(t), \ldots, \phi_n(t)\}$ is a set of n orthonormal waveforms.

The above provides a general scenario for mapping the symbols 0 to $M-1$ into a set of signal waveforms $\boldsymbol{b}_0(t)$ to $\boldsymbol{b}_{M-1}(t)$. A provision must also be made for transmitting a sequence of such M-ary symbols. If these symbols are to be transmitted at T-spaced intervals, the most straightforward way of accomplishing this is to choose the orthonormal waveforms $\phi_1(t), \ldots, \phi_n(t)$ in such a way that $\phi_k(t-\ell T)$ and $\phi_j(t-\ell' T)$ are orthonormal for all j, k, $1 \le j, k \le n$ and all

integer ℓ, ℓ'. In this case, a sequence of symbols, say $s_1, s_2, \ldots$, each drawn from the alphabet $\{0, \ldots, M-1\}$, could be mapped into a sequence of waveforms $\boldsymbol{b}_{s_1}(t), \boldsymbol{b}_{s_2}(t-T), \ldots$. The transmitted waveform would then be $\sum_\ell \boldsymbol{b}_{s_\ell}(t-\ell T)$.

PAM is a special case of this scenario where the dimension n is 1. The function $\phi_1(t)$ in this case is the real modulation pulse $p(t)$ for baseband transmission and $\sqrt{2}\,p(t)\cos(2\pi f_c t)$ for passband transmission. QAM is another special case where n is 2 at passband. In this case, the complex signals a_m are viewed as 2-dimensional real signals. The orthonormal waveforms (assuming real $p(t)$) are $\phi_1(t) = \sqrt{2}\,p(t)\cos(2\pi f_c t)$ and $\sqrt{2}\,p(t)\sin(2\pi f_c t)$.

More generally, it is not necessary to start at baseband and shift to passband[7], and it is not necessary for successive signals to be transmitted as time shifts of a basic waveform set. For example, in frequency-hopping systems, successive n-dimensional signals can be modulated to different carrier frequencies. What is important is that the successive transmitted signal waveforms are all orthogonal to each other.

Let $X(t)$ be the first signal waveform in such a sequence of successive waveforms. Then $X(t)$ is a choice from the set of M waveforms, $\boldsymbol{b}_0(t), \ldots, \boldsymbol{b}_{M-1}(t)$. We can represent $X(t)$ as $\sum_{k=1}^n X_k \phi_k(t)$ where, under hypothesis m, $X_k = a_{m,k}$ for $1 \le k \le n$. Let $\phi_{n+1}(t), \phi_{n+2}(t) \ldots$ be an additional set of orthonormal functions such that the entire set $\{\phi_k(t); k \ge 1\}$ spans the space of real $\mathcal{L}_2$ waveforms. The subsequence $\phi_{n+1}(t), \phi_{n+2}(t) \ldots$ might include the successive time shifts of $\phi_1(t), \ldots, \phi_n(t)$ for the example above, but in general can be arbitrary. We do assume, however, that successive signal waveforms are orthogonal to $\phi_1(t), \ldots, \phi_n(t)$, and thus that they can be expanded in terms of $\phi_{n+1}(t), \phi_{n+2}(t), \ldots, .$ The received random waveform $Y(t)$ is assumed to be the sum of $X(t)$, the WGN $Z(t)$, and contributions of signal waveforms other than X. These other waveforms could include successive signals from the given channel input and also signals from other users. This sum can be expanded over an arbitrarily large number, say ℓ, of these orthonormal functions as

$$Y(t) = \sum_{k=1}^{\ell} Y_k \phi_k(t) = \sum_{k=1}^{n} (X_k + Z_k)\phi_k(t) + \sum_{k=n+1}^{\ell} Y_k \phi_k(t). \tag{8.43}$$

Note that in (8.43), the random process $\{Y(t);\, t \in \mathbb{R}\}$ specifies the random variables $Y_1, \ldots, Y_\ell$. Assuming that the sample waveforms of $Y(t)$ are $\mathcal{L}_2$, it also follows that the limit as $\ell \to 0$ of $Y_1, \ldots, Y_\ell$ specifies $Y(t)$ in the $\mathcal{L}_2$ sense. Thus we consider $Y_1, \ldots, Y_\ell$ to be the observation at the channel output. It is convenient to separate these terms into two vectors, $\boldsymbol{Y} = (Y_1, \ldots, Y_n)^\mathsf{T}$ and $\boldsymbol{Y'} = (Y_{n+1}, \ldots, Y_\ell)^\mathsf{T}$.

Similarly, the WGN $Z(t) = \sum_k Z_k \phi_k(t)$ can be represented by $\boldsymbol{Z} = (Z_1, \ldots, Z_n)^\mathsf{T}$ and $\boldsymbol{Z'} = (Z_{n+1}, \ldots, Z_\ell)^\mathsf{T}$ and $X(t)$ can be represented as $\boldsymbol{X} = (X_1, \ldots, X_n)^\mathsf{T}$. Finally let $V(t) = \sum_k V_k \phi_k(t)$ be the contributions from other users and successive signals from the given user. Since these terms are orthogonal to $\phi_1(t), \ldots, \phi_n(t)$, $V(t)$ can be represented by $\boldsymbol{V'} = (V_{n+1}, \ldots, V_\ell)^\mathsf{T}$. With these changes, (8.43) becomes

$$\boldsymbol{Y} = \boldsymbol{X} + \boldsymbol{Z}; \qquad \boldsymbol{Y'} = \boldsymbol{Z'} + \boldsymbol{V'}. \tag{8.44}$$

The observation is a sample value of $(\boldsymbol{Y}, \boldsymbol{Y'})$, and the detector must choose the MAP value of $\boldsymbol{X}$. Assuming that $\boldsymbol{X}, \boldsymbol{Z}, \boldsymbol{Z'}$, and $\boldsymbol{V'}$ are statistically independent, the likelihoods can be

[7] It seems strange at first that the real vector and real waveform case here is more general than the complex case, but the complex case is used for notational and conceptual simplifications at baseband, where the baseband waveform will be modulated to passsband and converted to a real waveform.

expressed as

$$f_{\boldsymbol{Y}\,\boldsymbol{Y}'|\boldsymbol{X}}(\boldsymbol{y}\boldsymbol{y}'|\boldsymbol{a}_m) = f_{\boldsymbol{Z}}(\boldsymbol{y}-\boldsymbol{a}_m) f_{\boldsymbol{Y}'}(\boldsymbol{y}').$$

The likelihood ratio between hypothesis $\boldsymbol{a}_m$ and $\boldsymbol{a}_{m'}$ is then given by

$$\Lambda_{m,m'}(\boldsymbol{y}) = \frac{f_{\boldsymbol{Z}}(\boldsymbol{y}-\boldsymbol{a}_m)}{f_{\boldsymbol{Z}}(\boldsymbol{y}-\boldsymbol{a}_{m'})}. \tag{8.45}$$

The important thing here is that all the likelihood ratios (for $0 \le m, m' \le M-1$) depend only on $\boldsymbol{Y}$ and thus $\boldsymbol{Y}$ is a sufficient statistic for a MAP decision on $\boldsymbol{X}$. $\boldsymbol{Y}'$ is irrelevant to the decision, and thus its probability density is irrelevant (other than the need to assume that $\boldsymbol{Y}'$ is statistically independent of $(\boldsymbol{Z}, \boldsymbol{X})$). This also shows that the size of ℓ is irrelevant. This is summarized (and slightly generalized by dropping the Gaussian noise assumption) in the following theorem.

Theorem 8.4.2 (Theorem of irrelevance). *Let $\{\phi_k(t); k \ge 1\}$ be a set of real orthonormal functions. Let $X(t) = \sum_{k=1}^{n} X_k \phi_k(t)$ and $Z(t) = \sum_{k=1}^{n} Z_k \phi_k(t)$ be the input to a channel and the corresponding noise respectively, where $\boldsymbol{X} = (X_1, \ldots, X_n)^\mathsf{T}$ and $\boldsymbol{Z} = (Z_1, \ldots, Z_n)^\mathsf{T}$ are random vectors. Let $Y'(t) = \sum_{k>n} Y_k \phi_k(t)$ where for each $\ell > n$, $\boldsymbol{Y}' = (Y_{n+1}, \ldots, Y_\ell)^\mathsf{T}$ is a random vector that is statistically independent of the pair $\boldsymbol{X}$, $\boldsymbol{Z}$. Let $\boldsymbol{Y} = \boldsymbol{X} + \boldsymbol{Z}$. Then the LLR and the MAP detection of $\boldsymbol{X}$ from the observation of $\boldsymbol{Y}$, $\boldsymbol{Y}'$ depends only on $\boldsymbol{Y}$. That is, the observed sample value of $\boldsymbol{Y}'$ is irrelevant.*

The orthonormal set $\{\phi_1(t), \ldots, \phi_n(t)\}$ chosen above appears to have a more central importance than it really has. What is important is the existence of an n-dimensional subspace of real $\mathcal{L}_2$ that includes the signal set and has the property that the noise and signals orthogonal to this subspace are independent of the noise and signal within the subspace. In the usual case, we choose this subspace to be the space spanned by the signal set, but there are also cases where the subspace must be somewhat larger to provide for the independence between the subspace and its complement.

The irrelevance theorem does not specify how to do MAP detection based on the observed waveform, but rather shows how to reduce the problem to a finite dimensional problem. Since the likelihood ratios specify both the decision regions and the error probability for MAP detection, it is clear that the choice of orthonormal set cannot influence either the error probability or the mapping of received waveforms to hypotheses.

One important constraint in the above analysis is that both the noise and the interference (from successive transmissions and from other users) are additive. The other important constraint is that the interference is both orthogonal to the signal $X(t)$ and also statistically independent of $X(t)$. The orthogonality is why $\boldsymbol{Y} = \boldsymbol{X} + \boldsymbol{Z}$, with no contribution from the interference. The statistical independence is what makes $\boldsymbol{Y}'$ irrelevant.

If the interference is orthogonal but not independent, then a MAP decision based on $\boldsymbol{Y}$ alone could still be made. The resulting error probability, however, would be greater than or equal to that for a MAP decision based on $\{\boldsymbol{Y}, \boldsymbol{Y}'\}$. Thus the dependence generally permits a decrease in error probability

On the other hand, if the interference is non-orthogonal but independent, then $\boldsymbol{Y}$ would include both noise and a contribution from the interference, and the error probability would typically be larger, but never smaller, than in the orthogonal case. As a rule of thumb, then, non-orthogonal

interference tends to increase error probability, whereas dependence (if the receiver makes use of it) tends to reduce error probability.

If successive statistically independent signals, $\boldsymbol{X}_1, \boldsymbol{X}_2, \ldots$, are modulated onto distinct sets of orthonormal waveforms (*i.e.*, if X_1 is modulated onto the orthonormal waveforms $\phi_1(t)$ to $\phi_n(t)$), X_2 is modulated onto $\phi_{n+1}(t)$ to $\phi_{2n}(t)$, etc.) then it also follows, as in Subsection 8.4.2, that ML detection on a sequence $X_1, \ldots, X_\ell$ is equivalent to separate ML decisions on each input signal X_j, $1 \leq j \leq \ell$. The details are omitted since the only new feature in this extension is more complicated notation.

The higher dimensional mappings allowed in this subsection are sometimes called *channel codes*, and are sometimes simply viewed as more complex forms of modulation. The coding field is very large, but the following sections provide an introduction.

8.5 Orthogonal signal sets and simple channel coding

An orthogonal signal set is a set $\boldsymbol{a}_0, \ldots, \boldsymbol{a}_{M-1}$ of M real orthogonal M-vectors, each with the same energy E. Without loss of generality we choose a basis for $\mathbb{R}^M$ in which the mth basis vector is $\boldsymbol{a}_m/\sqrt{E}$. In this basis, $\boldsymbol{a}_0 = (\sqrt{E}, 0, 0, \ldots, 0)^\mathsf{T}$, $\boldsymbol{a}_1 = (0, \sqrt{E}, 0, \ldots, 0)^\mathsf{T}$, etc. Modulation onto an orthonormal set $\{\phi_m(t)\}$ of waveforms then maps hypothesis $\boldsymbol{a}_m$ $(0 \leq m \leq M-1)$ into the waveform $\sqrt{E}\phi_m(t)$. After addition of WGN, the sufficient statistic for detection is a sample value $\boldsymbol{y}$ of $\boldsymbol{Y} = \boldsymbol{A} + \boldsymbol{Z}$ where $\boldsymbol{A}$ takes on the values $\boldsymbol{a}_0, \ldots, \boldsymbol{a}_{M-1}$ with equal probability and $\boldsymbol{Z} = (Z_0, \ldots, Z_{M-1})^\mathsf{T}$ has iid components $\mathcal{N}(0, N_0/2)$. It can be seen that the ML decision is to decide on that m for which y_m is largest.

The major case of interest for orthogonal signals is where M is a power of 2, say $M = 2^b$. Thus the signal set can be used to transmit b binary digits, so the energy per bit is $E_b = E/b$. The number of required degrees of freedom for the signal set, however, is $M = 2^b$,so the spectral efficiency ρ (the number of bits per pair of degrees of freedom) is then $\rho = b/2^{b-1}$. As b gets large, ρ gets small at almost an exponential rate. It will be shown, however, that for large enough E_b, as b gets large holding E_b constant, the ML error probabiliity goes to 0. In particular, for any $E_b/N_0 < \ln 2 = 0.693$, the error probability goes to 0 exponentially as $b \to \infty$. Recall that $\ln 2 = 0.693$, *i.e.*, -1.59 dB, is the Shannon limit for reliable communication on a WGN channel with unlimited bandwidth. Thus the derivation to follow will establish the Shannon theorem for WGN and unlimited bandwidth. Before doing that, however, two closely related types of signal sets are discussed.

8.5.1 Simplex signal sets

Consider the random vector $\boldsymbol{A}$ with orthogonal equiprobable sample values $\boldsymbol{a}_0, \ldots, \boldsymbol{a}_{M-1}$ as described above. The mean value of $\boldsymbol{A}$ is then

$$\overline{\boldsymbol{A}} = \left(\frac{\sqrt{E}}{M}, \frac{\sqrt{E}}{M}, \ldots, \frac{\sqrt{E}}{M}\right)^\mathsf{T}.$$

We have seen that if a signal set is shifted by a constant vector, the Voronoi detection regions are also shifted and the error probability remains the same. However, such a shift can change the expected energy of the random signal vector. In particular, if the signals are shifted to remove the mean, then the signal energy is reduced by the energy (norm squared) of the mean. In this

case, the energy of the mean is E/M. A *simplex signal set* is an orthogonal signal set with the mean removed. That is,

$$\boldsymbol{S} = \boldsymbol{A} - \overline{\boldsymbol{A}}; \quad \boldsymbol{s}_m = \boldsymbol{a}_m - \overline{\boldsymbol{A}}; \quad 0 \leq m \leq M-1.$$

In other words, the mth component of $\boldsymbol{s}_m$ is $\sqrt{E}\,(1-1/M)$ and each other component is $-\sqrt{E}\,/M$. Each simplex signal has energy $E(1-1/M)$, so the simplex set has the same error probability as the related orthogonal set, but requires less energy by a factor of $(1-1/M)$. The simplex set of size M has dimensionality $M-1$, as can be seen from the fact that the sum of all the signals is 0, so the signals are linearly dependent. Figure 8.6 illustrates the orthogonal and simplex sets for $M = 2$ and 3.

For small M, the simplex set is a substantial improvement over the orthogonal set. For example, for $M = 2$, it has a 3 dB energy advantage (it is simply the antipodal one dimensional set). Also it uses half the dimensions of the orthogonal set. For large M, however, the improvement becomes almost negligible.

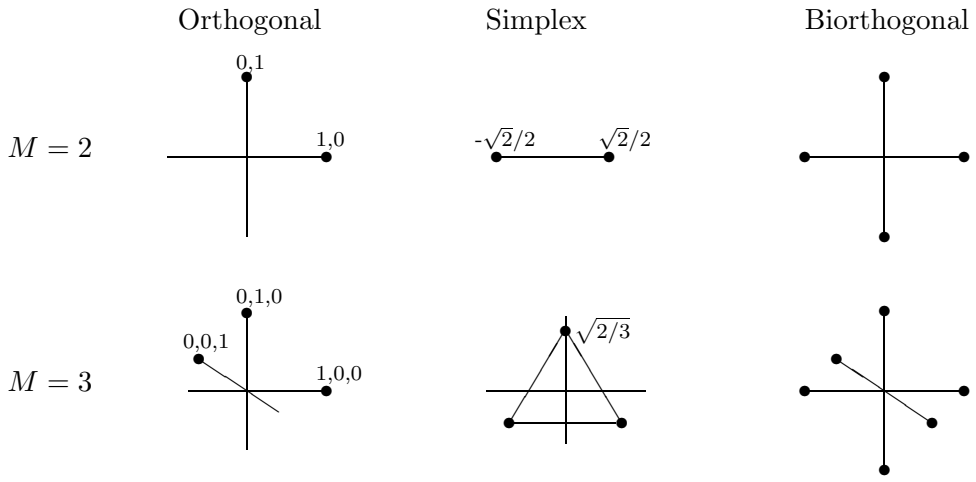

Figure 8.6: Orthogonal, simplex, and bi-orthogonal signal constellations, normalized to unit energy.

8.5.2 Bi-orthogonal signal sets

If $\boldsymbol{a}_0, \ldots, \boldsymbol{a}_{M-1}$ is a set of orthogonal signals, we call the set of $2M$ signals consisting of $\pm\boldsymbol{a}_0, \ldots, \pm\boldsymbol{a}_{M-1}$ a *bi-orthogonal signal set*. Two and three dimensional examples of bi-orthognal signals sets are given in figure 8.6.

It can be seen by the same argument used for orthogonal signal sets that the ML detection rule for such a set is to first choose the dimension m for which $|y_m|$ is largest, and then choose $\boldsymbol{a}_m$ or $-\boldsymbol{a}_m$ depending on whether y_m is positive or negative. Orthogonal signal sets and simplex signal sets each have the property that each signal is equidistant from every other signal. For bi-orthogonal sets, each signal is equidistant from all but one of the other signals. The exception, for the signal $\boldsymbol{a}_m$, is the signal $-\boldsymbol{a}_m$.

The bi-orthogonal signal set of M dimensions contains twice as many signals as the orthogonal set (thus sending one extra bit per signal), but has the same minimum distance between signals. It is hard to imagine[8] a situation where we would prefer an orthogonal signal set to a bi-orthogonal set, since one extra bit per signal is achieved at essentially no cost. However, for the limiting argument to follow, an orthogonal set is used since it is simpler to treat analytically. As M gets very large, the advantage of bi-orthogonal signals becomes smaller, which is why, asymptotically, the two are equivalent.

8.5.3 Error probability for orthogonal signal sets

Since the signals differ only by the ordering of the coordinates, the probability of error does not depend on which signal is sent; thus $\Pr(e) = \Pr(e \mid \boldsymbol{A}{=}\boldsymbol{a}_0)$. Conditional on $\boldsymbol{A} = \boldsymbol{a}_0$, Y_0 is $\mathcal{N}(\sqrt{E}, N_0/2)$ and Y_m is $\mathcal{N}(0, N_0/2)$ for $1 \le m \le M{-}1$. Note that if $\boldsymbol{A}{=}\boldsymbol{a}_0$ and $Y_0{=}y_0$, then an error is made if $Y_m \ge y_0$ for any m, $1 \le m \le M{-}1$. Thus

$$\Pr(e) = \int_{-\infty}^{\infty} f_{Y_0|\boldsymbol{A}}(y_0 \mid \boldsymbol{a}_0) \Pr\left(\bigcup_{m=1}^{M-1} (Y_m \ge y_0 \mid \boldsymbol{A} = \boldsymbol{a}_0)\right) dy_0. \tag{8.46}$$

The rest of the derivation of $\Pr(e)$, and its asymptotic behavior as M gets large, is simplified if we normalize the outputs to $W_m = Y_m\sqrt{2/N_0}$. Then, conditional on signal $\boldsymbol{a}_0$ being sent, W_0 is $\mathcal{N}(\sqrt{2E/N_0},\, 1) = \mathcal{N}(\alpha, 1)$, where α is an abbreviation for $\sqrt{2E/N_0}$. Also, conditional on $A = \boldsymbol{a}_0$, W_m is $\mathcal{N}(0,1)$ for $1 \le m \le M{-}1$.

$$\Pr(e) = \int_{-\infty}^{\infty} f_{W_0|\boldsymbol{A}}(w_0 \mid \boldsymbol{a}_0) \Pr\left(\bigcup_{m=1}^{M-1} (W_m \ge w_0 \mid \boldsymbol{A} = \boldsymbol{a}_0)\right) dw_0. \tag{8.47}$$

Using the union bound on the union above,

$$\Pr\left(\bigcup_{m=1}^{M-1} (W_m \ge w_0 \mid \boldsymbol{A} = \boldsymbol{a}_0)\right) \le (M-1)Q(w_0). \tag{8.48}$$

The union bound is quite tight when applied to independent quantitities that have small aggregate probability. Thus this bound will be quite tight when w_0 is large and M is not too large. When w_0 is small, however, the bound becomes loose. For example, for $w_0 = 0$, $Q(w_0) = 1/2$ and the bound in (8.48) is $(M-1)/2$, much larger than the obvious bound of 1 for any probability. Thus, in the analysis to follow, the left side of (8.48) will be upper-bounded by 1 for small w_0 and by $(M-1)Q(w_0)$ for large w_0. Since both 1 and $(M-1)Q(w_0)$ are valid upper bounds for all w_0, the dividing point γ between small and large can be chosen arbitrarily. It is chosen in what follows to satisfy

$$\exp(-\gamma^2/2) = 1; \qquad \gamma = \sqrt{2\ln M} \tag{8.49}$$

It might seem more natural in light of (8.48) to replace γ above by the γ_1 that satisfies $(M-1)Q(\gamma_1) = 1$, and that turns out to be the natural choice in the lower bound to $\Pr(e)$ developed

[8] One possibility is that at passband a phase error of π can turn $\boldsymbol{a}_m$ into $-\boldsymbol{a}_m$. Thus with bi-orthogonal signals it is necessary to track phase or use differential phase.

in Exercise 8.10. It is not hard to see, however, that γ/γ_1 goes to 1 as $M \to \infty$, so the difference is not of major importance. Splitting the integral in (8.47) into $w_0 \le \gamma$ and $w_0 > \gamma$,

$$\begin{aligned}
\Pr(e) &\le \int_{-\infty}^{\gamma} f_{W_0|\boldsymbol{A}}(w_0 \mid \boldsymbol{a}_0)\, dw_0 + \int_{\gamma}^{\infty} f_{W_0|\boldsymbol{A}}(w_0 \mid \boldsymbol{a}_0)(M-1)Q(w_0)\, dw_0 && (8.50)\\
&\le Q(\alpha-\gamma) + \int_{\gamma}^{\infty} f_{W_0|\boldsymbol{A}}(w_0 \mid \boldsymbol{a}_0)(M-1)Q(\gamma)\exp\left[\frac{\gamma^2}{2} - \frac{w_0^2}{2}\right] dw_0 && (8.51)\\
&\le Q(\alpha-\gamma) + \int_{\gamma}^{\infty} \frac{1}{\sqrt{2\pi}}\exp\left[\frac{-(w_0-\alpha)^2 + \gamma^2 - w_0^2}{2}\right] dw_0 && (8.52)\\
&= Q(\alpha-\gamma) + \int_{\gamma}^{\infty} \frac{1}{\sqrt{2\pi}}\exp\left[\frac{-2(w_0-\alpha/2)^2 + \gamma^2 - \alpha^2/2}{2}\right] dw_0 && (8.53)\\
&= Q(\alpha-\gamma) + \frac{1}{\sqrt{2}} Q\left(\sqrt{2}\left(\gamma - \frac{\alpha}{2}\right)\right)\exp\left[\frac{\gamma^2}{2} - \frac{\alpha^2}{4}\right]. && (8.54)
\end{aligned}$$

The first term on the right side of (8.50) is the lower tail of the distribution of W_0, and is the probability that the *negative* of the fluctuation of W_0 exceeds $\alpha-\gamma$, *i.e.*, $Q(\alpha-\gamma)$. In the second term, $Q(w_0)$ is upper bounded using Exercise 8.7c, thus resulting in (8.51). This is simplified by $(M-1)Q(\gamma) \le M\exp(-\gamma^2/2) = 1$, resulting in (8.52). The exponent is then manipulated to 'complete the square' in (8.53), leading to an integral of a Gaussian density, as given in (8.54).

The analysis now breaks into three special cases, the first where $\alpha \le \gamma$, the second where $\alpha/2 \le \gamma < \alpha$, and the third where $\gamma \le \alpha/2$. We explain the significance of these cases after completing the bounds.

Case (1): ($\alpha \le \gamma$) The argument of the first Q function in (8.53) is less than or equal to 0, so its value lies between 1/2 and 1. This means that $\Pr(e) \le 1/2$, which is a useless result. As seen later, this is the case where the rate is greater than or equal to capacity. It is also shown in Exercise 8.10 that the error probability must be large in this case.

Case (2): ($\alpha/2 \le \gamma < \alpha$) Each Q function in (8.53) has a non-negative argument, so the bound $Q(x) \le \frac{1}{2}\exp(\frac{-x^2}{2})$ applies (see Exercise 8.7b).

$$\begin{aligned}
\Pr(e) &\le \frac{1}{2}\exp\left[\frac{-(\alpha-\gamma)^2}{2}\right] + \frac{1}{2\sqrt{2}}\exp\left(\frac{-\alpha^2}{4} + \frac{\gamma^2}{2} - (\gamma-\alpha/2)^2\right) && (8.55)\\
&\le \left[\frac{1}{2} + \frac{1}{2\sqrt{2}}\right]\exp\left[\frac{-(\alpha-\gamma)^2}{2}\right] \le \exp\left(\frac{-(\alpha-\gamma)^2}{2}\right). && (8.56)
\end{aligned}$$

Note that (8.56) follows (8.55) from combining the terms in the exponent of the second term. The fact that exponents are equal is not too surprising, since γ was chosen to approximately equalize the integrands in (8.50) at $w_0 = \gamma$.

Case (3): ($\gamma \le \alpha/2$) The argument of the second Q function in (8.53) is less than or equal to 0, so its value lies between 1/2 and 1 and is upper bounded by 1, yielding

$$\begin{aligned}
\Pr(e) &\le \frac{1}{2}\exp\left[\frac{-(\alpha-\gamma)^2}{2}\right] + \frac{1}{2\sqrt{2}}\exp\left[\frac{-\alpha^2}{4} + \frac{\gamma^2}{2}\right] && (8.57)\\
&\le \exp\left(\frac{-\alpha^2}{4} + \frac{\gamma^2}{2}\right). && (8.58)
\end{aligned}$$

Since the two exponents in (8.55) are equal, the first exponent in (8.57) must be smaller than the second, leading to (8.58). This is essentially the union bound derived in Exercise 8.8.

The lower bound in Exercise 8.10 shows that these bounds are quite tight, but the sense in which they are tight will be explained later.

We now explore what α and γ are in terms of the number of codewords M and the energy per bit, E_b. Recall that $\alpha = \sqrt{2E/N_0}$. Also $\log_2 M = b$ where b is the number of bits per signal. Thus $\alpha = \sqrt{2bE_b/N_0}$. From (8.49), $\gamma^2 = 2\ln M = 2b\ln(2)$. Thus

$$\alpha - \gamma = \sqrt{2b}\left[\sqrt{E_b/N_0} - \sqrt{\ln 2}\right].$$

Substituting these values into (8.56) and (8.58),

$$\Pr(e) \leq \exp\left[-b\left(\sqrt{E_b/N_0} - \sqrt{\ln 2}\right)^2\right] \quad \text{for } \frac{E_b}{4N_0} \leq \ln 2 < \frac{E_b}{N_0} \tag{8.59}$$

$$\Pr(e) \leq \exp\left[-b\left(\frac{E_b}{2N_0} - \ln 2\right)\right] \quad \text{for } \ln 2 < \frac{E_b}{4N_0}. \tag{8.60}$$

We see from this that if $E_b/N_0 > \ln 2$, then as $b \to \infty$ holding E_b constant, $\Pr(e) \to 0$.

Recall that in (7.86), we stated that the capacity (in bits per second) of a WGN channel of bandwidth W, noise spectral density $N_0/2$, and power P is

$$C = W \log\left(1 + \frac{P}{\mathsf{W}N_0}\right). \tag{8.61}$$

With no bandwidth constraint, *i.e.*, in the limit $\mathsf{W} \to \infty$, the ultimate capacity is $C = \frac{P}{N_0 \ln 2}$. This means that, according to Shannon's theorem, for any rate $R < C = \frac{P}{N_0 \ln 2}$, there are codes of rate R bits per second for which the error probability is arbitrarily close to 0. Now $P/R = E_b$, so Shannon says that if $\frac{E_b}{N_0 \ln 2} > 1$, then codes exist with arbitrarily small error.

The orthogonal codes provide a concrete proof of this ultimate capacity result, since (8.59) shows that $\Pr(e)$ can be made arbitrarily small (by increasing b) so long as $\frac{E_b}{N_0 \ln 2} > 1$. Shannon's theorem also says that the error probability can not be made small if $\frac{E_b}{N_0 \ln 2} < 1$. We have not quite proven that here, although Exercise 8.10 shows that the error probability cannot be made arbitrarily small for an orthogonal code[9] if $\frac{E_b}{N_0 \ln 2} < 1$.

The limiting operation here is slightly unconventional. As b increases, E_b is held constant. This means that the energy E in the signal increases linearly with b, but the size of the constellation increases exponentially with b. Thus the bandwidth required for this scheme is infinite in the limit, and going to infinity very rapidly. This means that this is not a practical scheme for approaching capacity, although sets of 64 or even 256 bi-orthogonal waveforms are used in practice.

The point of the analysis, then, is first to show that this infinite bandwidth capacity can be approached, but second to show also that using large but finite sets of orthogonal (or bi-orthogonal or simplex) waveforms does decrease error probability for fixed signal to noise ratio, and decreases it as much as desired (for rates below capacity) if enough bandwidth is used.

[9]Since a simplex code has the same error probability as the corresponding orthogonal code, but differs in energy from the orthogonal code by a vanishingly small amount as $M \to \infty$, the error probability for simplex codes also cannot be made arbitrarily small for any given $\frac{E_b}{N_0 \ln 2}$ less than 1. It is widely believed, but never proven, that simplex codes are optimal in terms of ML error probability whenever the error probability is small. There is a known example, however, [30], for all $M \geq 7$, where the simplex is non-optimal, but in this example, the signal to noise ratio is very small and the error probability is very large.

The different forms of solution in (8.59) and (8.60) are interesting, and not simply consequences of the upper bounding used. For case (2), which leads to (8.59), the typical way that errors occur is when $w_0 \approx \gamma$. In this situation, the union bound is on the order of 1, which indicates that, conditional on $y_0 \approx \gamma$, it is quite likely that an error will occur. In other words, the typical error event involves an unusually large negative value for w_0 rather than any unusual values for the other noise terms. In case (3), which leads to (8.60), the typical way for errors to occur is when $w_0 \approx \alpha/2$ and when some other noise term is also at about $\alpha/2$. In this case, an unusual event is needed both in the signal direction and in some other direction.

A more intuitive way to look at this distinction is to visualize what happens when E/N_0 is held fixed and M is varied. Case 3 corresponds to small M, case 2 to larger M, and case 1 to very large M. For small M, one can visualize the Voronoi region around the transmitted signal point. Errors occur when the noise carries the signal point outside the Voronoi region, and that is most likely at the points in the Voronoi surface closest to the transmitted signal, *i.e.*, at points half way between the transmitted point and some other signal point. As M increases, the number of these midway points increases until one of them is almost certain to cause an error when the noise in the signal direction becomes too large.

8.6 Block Coding

This section provides a brief introduction to the subject of coding for error correction on noisy channels. Coding is a major topic in modern digital communication, certainly far more major than suggested by the length of this introduction. In fact, coding is a topic that deserves its own text and its own academic subject in any serious communication curriculum. Suggested texts are [6] and [15]. Our purpose here is to provide enough background and examples to understand the role of coding in digital communication, rather than to prepare the student for coding research. We start by viewing orthogonal codes as block codes using a binary alphabet. This is followed by the Reed-Muller codes, which provide considerable insight into coding for the WGN channel. This then leads into Shannon's celebrated noisy-channel coding theorem.

A *block code* is a code for which the incoming sequence of binary digits is segmented into blocks of some given length m and then these binary m-tuples are mapped into *codewords*. There are thus 2^m codewords in the code; these codewords might be binary n-tuples of some *block length* $n > m$, or might be vectors of signals, or might be waveforms. There is no fundamental difference between coding and modulation; for example the orthogonal code above with $M = 2^m$ codewords can be viewed either as modulation with a large signal set or coding using binary m-tuples as input.

8.6.1 Binary orthogonal codes and Hadamard matrices

When orthogonal codewords are used on a WGN channel, any orthogonal set is equally good from the standpoint of error probability. One possibility, for example, is the use of orthogonal sine waves. From an implementation standpoint, however, there are simpler choices than orthogonal sine waves. Conceptually, also, it is helpful to see that orthogonal codewords can be constructed from binary codewords. This digital approach will turn out to be conceptually important in the study of fading channels and diversity in the next chapter. It also helps in implementation, since it postpones the point at which digital hardware gives way to analog waveforms.

One digital approach to generating a large set of orthogonal waveforms comes from first generating a set of M binary codewords, each of length M and each distinct pair differing in exactly $M/2$ places. Each binary digit can then be mapped into an antipodal signal, $0 \to +a$ and $1 \to -a$. This yields an M-tuple of real-valued antipodal signals, $s_1, \ldots, s_M$, which is then mapped into the waveform $\sum_j s_j \phi_j(t)$ where $\{\phi_j(t); 1 \leq j \leq M\}$ is an orthonormal set (such as sinc functions or Nyquist pulses). Since each pair of binary codewords differs in $M/2$ places, the corresponding pair of waveforms are orthogonal and each waveform has equal energy. A binary code with the above properties is called an *binary orthogonal code*.

There are many ways to generate binary orthogonal codes. Probably the simplest is from a *Hadamard matrix*. For each integer $m \geq 1$, there is a 2^m by 2^m Hadamard matrix H_m. Each distinct pair of rows in the Hadamard matrix H_m differs in exactly 2^{m-1} places, so the 2^m rows of H_m constitute an binary orthogonal code with 2^m codewords.

It turns out that there is a simple algorithm for generating the Hadamard matrices. The Hadamard matrix H_1 is defined to have the rows 00 and 01 which trivially satisfy the condition that each pair of distinct rows differ in half the positions. For any integer $m > 1$, the Hadamard matrix H_{m+1} of order 2^{m+1} can be expressed as four 2^m by 2^m submatrices. Each of the upper two submatrices is H_m, and the lower two submatrices are H_m and $\overline{H}_m$, where $\overline{H}_m$ is the complement of H_m. This is illustrated in Figure 8.7 below.

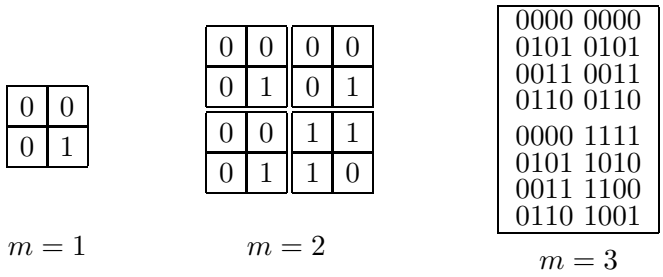

Figure 8.7: Hadamard Matrices.

Note that each row of each matrix in Figure 8.7, other than the all zero row, contains half zeroes and half ones. To see that this remains true for all larger values of m, we can use induction. Thus assume, for given m, that H_m contains a single row of all zeros and $2^m - 1$ rows, each with exactly half ones. To prove the same for H_{m+1}, first consider the first 2^m rows of H_{m+1}. Each row has twice the length and twice the number of ones as the corresponding row in H_m. Next consider the final 2^m rows. Note that $\overline{H}_m$ has all ones in the first row and 2^{m-1} ones in each other row. Thus the first row in the second set of 2^m rows of H_{m+1} has no ones in the first 2^m positions and 2^m ones in the final 2^m positions, yielding 2^m ones in 2^{m+1} positions. Each remaining row has 2^{m-1} ones in the first 2^m positions and 2^{m-1} ones in the final 2^m positions, totaling 2^m ones as required.

By a similar inductive argument (See Exercise 8.18), the mod-2 sum[10] of any two rows of H_m is another row of H_m. Since the mod-2 sum of two rows gives the positions in which the rows differ, and only the mod-2 sum of a codeword with itself gives the all 0 codeword, this means that the set of rows is a binary orthogonal set.

The fact that the mod-2 sum of any two rows is another row makes the corresponding code a

[10]The mod-2 sum of two binary numbers is defined by $0 \oplus 0 = 0$, $0 \oplus 1 = 1$, $1 \oplus 0 = 1$, and $1 \oplus 1 = 0$. The mod-2 sum of two rows (or vectors) or binary numbers is the component-wise row (or vector) of mod-2 sums.

special kind of binary code called a *linear code*, *parity-check code* or *group code* (these are all synonyms). Binary M-tuples can be regarded as vectors in a vector space over the binary scalar field. It is not necessary here to be precise about what a field is; so far it has been sufficient to consider vector spaces defined over the real or complex fields. However, the binary numbers, using mod-two addition and ordinary multiplication, also form a field and the familiar properties of vector spaces apply here also.

Since the set of codewords in a linear code is closed under mod-2 sums (and also closed under scalar multiplication by 1 or 0), a linear code is a binary vector subspace of the binary vector space of binary M-tuples. An important property of such a subspace, and thus of a linear code, is that the set of positions in which two codewords differ is the set of positions in which the mod-2 sum of those codewords contains 1's. This means that the distance between two codewords (*i.e.*, the number of positions in which they differ) is equal to the weight (the number of positions containing 1's) of their mod-2 sum. This means, in turn, that for a linear code, the minimum distance $d_{\min}$, taken between all distinct pairs of codewords, is equal to the minimum weight (minimum number of 1's) of any non-zero codeword.

Another important property of a linear code (other than the trivial code consisting of all binary M-tuples) is that some components x_k of each codeword $\boldsymbol{x} = (x_1, \ldots, x_M)^{\mathsf{T}}$ can be represented as mod-2 sums of other components. For example, in the $m = 3$ case of Figure 8.7, $x_4 = x_2 \oplus x_3$, $x_6 = x_2 \oplus x_5$, $x_7 = x_3 \oplus x_5$, $x_8 = x_4 \oplus x_5$, and $x_1 = 0$, Thus only 3 of the components can be independently chosen, leading to a 3-dimensional binary subspace. Since each component is binary, such a 3-dimensional subspace contains $2^3 = 8$ vectors. The components that are mod-2 combinations of previous components are called 'parity checks' and often play an important role in decoding. The first component, x_1, can be viewed as a parity check since it cannot be chosen independently, but its only role in the code is to help achieve the orthogonality property. It is irrelevant in decoding.

It is easy to modify the binary orthogonal code to generate a binary simplex code, *i.e.*, a binary code which, after the mapping $0 \to a, 1 \to -a$, forms a simplex in Euclidean space. The first component of each binary codeword is dropped, turning the code into M codewords over $M-1$ dimensions. Note that in terms of the antipodal signals generated by the binary digits, dropping the first component converts the signal $+a$ (corresponding to the first binary component 0) into the signal 0 (which corresponds neither to the binary 0 or 1) . The generation of the binary biorthogonal code is equally simple; the rows of H_m yield half of the codewords and the rows of $\overline{H}_m$ yield the other half. Both the simplex and the biorthogonal code, as expressed in binary form here, are linear binary block codes.

Two things have been accomplished with this representation of orthogonal codes. First, orthogonal codes can be generated from a binary sequence mapped into an antipodal sequence, and second, an example has been given where modulation over a large alphabet can be viewed as a binary block code followed by modulation over a binary or very small alphabet.

8.6.2 Reed-Muller codes

Orthogonal codes (and the corresponding simplex and biorthgonal codes) use enormous bandwidth for large M. The Reed-Muller codes constitute a class of binary linear block codes that include large bandwidth expansion (in fact they include the binary biorthogonal codes) but also allow for much smaller bandwidth expansion, *i.e.*, they allow for binary codes with M codewords where $\log M$ is much closer to the number of dimensions used by the code.

The Reed-Muller codes are specified by two integer parameters, $m \geq 1$ and $0 \leq r \leq m$; a binary linear block code, denoted $\mathrm{RM}(r,m)$, exists for each such choice. The parameter m specifies the block length to be $n = 2^m$. The minimum distance $d_{\min}(r,m)$ of the code and the number of binary information digits $k(r,m)$ required to specify a codeword are given by

$$d_{\min}(r,m) = 2^{m-r} \qquad k(r,m) = \sum_{j=0}^{r} \binom{m}{j} \tag{8.62}$$

where $\binom{m}{j} = \frac{m!}{j!(m-j)!}$. Thus these codes, like the binary orthogonal codes, exist only at block lengths equal to a power of 2. While there is only one binary orthogonal code (as defined through H_m) for each m, there is a range of RM codes for each m ranging from large $d_{\min}$ and small k to small $d_{\min}$ and large k as r increases.

For each m, these codes are trivial for $r = 0$ and $r = m$. For $r = 0$ the code consists of two codewords selected by a single bit, so $k(0,m) = 1$; one codeword is all 0's and the other is all 1's, leading to $d_{\min}(0,m) = 2^m$. For $r = m$, the code is the set of all binary 2^m tuples, leading to $d_{\min}(m,m) = 1$ and $k(m,m) = 2^m$. For $m = 1$, then, there are two RM codes. $\mathrm{RM}(0,1)$ consists of the two codewords (0,0) and (1,1), and $\mathrm{RM}(1,1)$ consists of the four codewords (0,0), (0,1), (1,0), and (1,1).

For $m > 1$ and intermediate values of r, there is a simple algorithm, much like that for Hadamard matrices, that specifies the set of codewords. The algorithm is recursive, and, for each $m > 1$ and $0 < r < m$, specifies $\mathrm{RM}(r,m)$ in terms of $\mathrm{RM}(r,m{-}1)$ and $\mathrm{RM}(r{-}1,m{-}1)$. Specifically, $\boldsymbol{x} \in \mathrm{RM}(r,m)$ if $\boldsymbol{x}$ is the concatenation of $\boldsymbol{u}$ and $\boldsymbol{u} \oplus \boldsymbol{v}$, denoted $\boldsymbol{x} = (\boldsymbol{u}, \boldsymbol{u} \oplus \boldsymbol{v})$, for some $\boldsymbol{u} \in RM(r,m{-}1,)$ and $\boldsymbol{v} \in RM(r{-}1,m{-}1)$. More formally, for $0 < r < m$,

$$\mathrm{RM}(r,m) = \{(\boldsymbol{u}, \boldsymbol{u} \oplus \boldsymbol{v}) \mid \boldsymbol{u} \in \mathrm{RM}(r,m{-}1), \boldsymbol{v} \in \mathrm{RM}(r{-}1,m{-}1)\}. \tag{8.63}$$

The analogy with Hadamard matrices is that $\boldsymbol{x}$ is a row of H_m if $\boldsymbol{u}$ is a row of H_{m-1} and $\boldsymbol{v}$ is either all ones or all zeros.

The first thing to observe about this definition is that if $\mathrm{RM}(r,m{-}1)$ and $\mathrm{RM}(r{-}1,m{-}1)$ are linear codes, then $\mathrm{RM}(r,m)$ is also. To see this, let $\boldsymbol{x} = (\boldsymbol{u}, \boldsymbol{u} \oplus \boldsymbol{v})$ and $\boldsymbol{x}' = (\boldsymbol{u}', \boldsymbol{u}' \oplus \boldsymbol{v}')$. Then

$$\boldsymbol{x} \oplus \boldsymbol{x}' = (\boldsymbol{u} \oplus \boldsymbol{u}', \boldsymbol{u} \oplus \boldsymbol{u}' \oplus \boldsymbol{v} \oplus \boldsymbol{v}') = (\boldsymbol{u}'', \boldsymbol{u}'' \oplus \boldsymbol{v}'')$$

where $\boldsymbol{u}'' = \boldsymbol{u} \oplus \boldsymbol{u}' \in \mathrm{RM}(r,m{-}1)$ and $\boldsymbol{v}'' = \boldsymbol{v} \oplus \boldsymbol{v}' \in \mathrm{RM}(r{-}1,m{-}1)$. This shows that $\boldsymbol{x} \oplus \boldsymbol{x}' \in \mathrm{RM}(r,m)$, and it follows that $\mathrm{RM}(r,m)$ is a linear code if $\mathrm{RM}(r,m{-}1)$ and $\mathrm{RM}(r{-}1,m{-}1)$ are. Since both $\mathrm{RM}(0,m)$ and $\mathrm{RM}(m,m)$ are linear for all $m \geq 1$, it follows by induction on m that all the Reed-Muller codes are linear.

Another observation is that different choices of the pair $\boldsymbol{u}$ and $\boldsymbol{v}$ cannot lead to the same value of $\boldsymbol{x} = (\boldsymbol{u}, \boldsymbol{u} \oplus v)$. To see this, let $\boldsymbol{x}' = (\boldsymbol{u}', \boldsymbol{v}')$. Then if $\boldsymbol{u} \neq \boldsymbol{u}'$, it follows that the first half of $\boldsymbol{x}$ differs from that of $\boldsymbol{x}'$. Similarly if $\boldsymbol{u} = \boldsymbol{u}'$, and $\boldsymbol{v} \neq \boldsymbol{v}'$, then the second half of $\boldsymbol{x}$ differs from that of $\boldsymbol{x}'$. Thus $\boldsymbol{x} = \boldsymbol{x}'$ only if both $\boldsymbol{u} = \boldsymbol{u}'$ and $\boldsymbol{v} = \boldsymbol{v}'$. As a consequence of this, the number of information bits required to specify a codeword in $\mathrm{RM}(r,m)$, denoted $k(r,m)$ is equal to the number required to specify a codeword in $\mathrm{RM}(r,m{-}1)$ plus that to specify a codeword in $\mathrm{RM}(r{-}1,m{-}1)$, *i.e.*, for $0 < r < m$,

$$k(r,m) = k(r,m{-}1) + k(r{-}1,m{-}1)$$

Exercise 8.19 shows that this relationship implies the explicit form for $k(r,m)$ given in (8.62). Finally Exercise 8.20 verifies the explicit form for $d_{\min}(r,m)$ in (8.62).

The RM$(1, m)$ codes are the binary bi-orthogonal codes and one can view the construction in (8.63) as being equivalent to the Hadamard matrix algorithm by replacing the M by M matrix H_m in the Hadamard algorithm by the $2M$ by M matrix $\begin{bmatrix} H_m \\ G_m \end{bmatrix}$ where $G_m = \overline{H}_m$.

Another interesting case is the RM$(m-2, m)$ codes. These have $d_{\min}(m-2, m) = 4$ and $k(m-2, m) = 2^m - m - 1$ information bits. In other words, they have $m+1$ parity checks. As explained below, these codes are called *extended Hamming codes*. A property of all RM codes is that all codewords have an even number[11] of 1's and thus the last component in each codeword can be viewed as an overall parity check which is chosen to ensure that the codeword contains an even number of 1's.

If this final parity check is omitted from RM$(m-2, m)$ for any given m, the resulting code is still linear and must have a minimum distance of at least 3, since only one component has been omitted. This code is called the Hamming code of block length $2^m - 1$ with m parity checks. It has the remarkable property that every binary $2^m - 1$ tuple is either a codeword in this code or distance 1 from a codeword[12].

The Hamming codes are not particularly useful in practice for the following reasons. If one uses a Hamming code at the input to a modulator and then makes hard decisions on the individual bits before decoding, then a block decoding error is made whenever 2 or more bit errors occur. This is a small improvement in reliability at a very substantial cost in transmission rate. On the other hand, if soft decisions are made, using the extended Hamming code (*i.e.*, RM$(m-2, m)$ extends $d_{\min}$ from 3 to 4, greatly decreasing the error probability with a marginal cost in added redundant bits.

8.7 The noisy-channel coding theorem

The previous sections provided a brief introduction to coding. It provided several examples showing that the use of binary codes could accomplish the same thing, for example, as the use of large sets of orthogonal, simplex, or bi-orthogonal waveforms. There was an ad hoc nature to the development, however, illustrating a number of schemes with various interesting properties, but little in the way of general results.

The earlier results on $\Pr(e)$ for orthogonal codes were more fundamental, showing that $\Pr(e)$ could be made arbitrarily small for a WGN channel with no bandwidth constraint if $\frac{E_b}{N_0}$ is greater than $\ln 2$. This constituted a special case of the noisy-channel coding theorem, saying that arbitrarily small $\Pr(e)$ can be achieved for that very special channel and set of constraints.

8.7.1 Discrete memoryless channels

This section states and proves the noisy-channel coding theorem for another special case, that of discrete memoryless channels (DMC's). This may seem a little peculiar after all the emphasis in this and the last chapter on WGN. There are two major reasons for this choice. The first is that the argument is particularly clear in the DMC case, particularly after studying the AEP for

[11] This property can be easily verified by induction.

[12] To see this, note that there are 2^{2^m-1-m} codewords, and each codeword has $2^m - 1$ neighbors; these are distinct from the neighbors of other codewords since $d_{\min}$ is at least 3. Adding the codewords and the neighbors, we get the entire set of 2^{2^m-1} vectors. This also shows that the minimum distance is exactly 3.

discrete memoryless sources. The second is that the argument can be generalized easily, as will be discussed later. A DMC has a discrete input sequence $\boldsymbol{X} = X_1, \ldots, X_k, \ldots$. At each discrete time k, the input to the channel belongs to a finite alphabet $\mathcal{X}$ of symbols. For example, in the last section, the input alphabet could be viewed as the signals $\pm a$. The question of interest would then be whether it is possible to communicate reliably over a channel when the decision to use the alphabet $\mathcal{X} = \{a, -a\}$ has already been made. The channel would then be regarded as the part of the channel from signal selection to an output sequence from which detection would be done. In a more general case, the signal set could be an arbitrary QAM set.

A DMC is also defined to have a discrete output sequence $\boldsymbol{Y} = Y_1, \ldots, Y_k, \ldots$, where each output Y_k in the output sequence is a selection from a finite alphabet $\mathcal{Y}$ and is a probabilistic function of the input and noise in a way to be described shortly. In the example above, the output alphabet could be chosen as $\mathcal{Y} = \{a, -a\}$ corresponding to the case in which hard decisions are made on each signal at the receiver. The channel would then include the modulation and detection as an internal part, and the question of interest would be whether coding at the input and decoding from the single-letter hard decisions at the output could yield reliable communication.

Another choice would be to use the pre-decision outputs, first quantized to satisfy the finite alphabet constraint. Another, almost identical choice, would be a detector that produced a quantized LLR as opposed to a decision.

In summary, the choice of discrete memoryless channel alphabets depends on what part of the overall communication problem is being addressed.

In general, a channel is described not only by the input and output alphabets but also the probabilistic description of the outputs conditional on the inputs (the probabilistic description of the inputs is selected by the channel user). Let $\boldsymbol{X}^n = (X_1, X_2, \ldots X_n)^{\mathsf{T}}$ be the channel input, here viewed either over the lifetime of the channel or any time greater than or equal to the duration of interest. Similarly the output is denoted by $\boldsymbol{Y}^n = (Y_1, \ldots, Y_n)$. For a DMC, the probability of the output n-tuple, conditional on the input n-tuple, is defined to satisfy

$$p_{\boldsymbol{Y}^n|\boldsymbol{X}^n}(y_1, \ldots, y_n \mid x_1, \ldots, x_n) = \prod_{k=1}^{n} p_{Y_k|X_k}(y_k|x_k) \tag{8.64}$$

where $p_{Y_k|X_k}(y_k = j|x_k = i)$, for each $j \in \mathcal{Y}$ and $i \in \mathcal{X}$ is a function only of i and j and not of the time k. Thus, conditional on a given input sequence, the output symbols are independent and each has a conditional distribution depending only on the corresponding input symbol. This conditional distribution is denoted $P_{i,j}$ for all $i \in \mathcal{X}$ and $j \in \mathcal{Y}$, *i.e.*, $p_{Y_k|X_k}(y_k{=}j|x_k{=}i) = P_{i,j}$. Thus the channel is completely described by the input alphabet, the output alphabet, and the conditional distribution function $P_{i,j}$. The conditional distribution function is usually called the *transition* function or matrix.

The most intensely studied DMC over the past 60 years is the binary symmetric channel (BSC), which has $\mathcal{X} = \{0, 1\}, \mathcal{Y} = \{0, 1\}$ and satisfies $P_{0,1} = P_{1,0}$. This single number $P_{0,1}$ thus specifies the BSC. The WGN channel with antipodal inputs and ML hard decisions at the output is an example of the BSC. Despite the intense study of the BSC and its inherent simplicity, the question of optimal codes of long block length (optimal in the sense of minimum error probability) is largely unanswered. Thus, the noisy-channel coding theorem, which describes various properties of the achievable error probability through coding plays a particularly important role in coding.

8.7.2 Capacity

The capacity C of a DMC is defined in this subsection. The following subsection, after defining the rate R at which information enters the modulator, shows that reliable communication is impossible on a channel if $R > C$. This is known as the converse to the noisy-channel coding theorem, and is in contrast to the final subsection which shows that arbitrarily reliable communication is possible for any $R < C$. As in the analysis of orthogonal codes, communication at rates below capacity can be made increasingly reliable with increasing block length, while this is not possible for $R > C$.

The capacity is defined in terms of various entropies. For a given DMC and given sequence length n, let $p_{\boldsymbol{Y}^n|\boldsymbol{X}^n}(\boldsymbol{y}^n|\boldsymbol{x}^n)$ be given by (8.64) and let $p_{\boldsymbol{X}^n}(\boldsymbol{x}^n)$ denote an arbitrary probability mass function chosen by the user on the input $X_1 \ldots, X_n$. This leads to a joint entropy $\mathsf{H}[\boldsymbol{X}^n\,\boldsymbol{Y}^n]$. From (2.37), this can be broken up as

$$\mathsf{H}[\boldsymbol{X}^n\,\boldsymbol{Y}^n] = \mathsf{H}[\boldsymbol{X}^n] + \mathsf{H}[\boldsymbol{Y}^n|\boldsymbol{X}^n], \tag{8.65}$$

where $\mathsf{H}[\boldsymbol{Y}^n|\boldsymbol{X}^n] = \mathsf{E}[-\log p_{\boldsymbol{Y}^n|\boldsymbol{X}^n}(\boldsymbol{Y}^n|\boldsymbol{X}^n)]$. Note that because $\mathsf{H}[\boldsymbol{Y}^n|\boldsymbol{X}^n]$ is defined as an expectation over both $\boldsymbol{X}^n$ and $\boldsymbol{Y}^n$, $\mathsf{H}[\boldsymbol{Y}^n|\boldsymbol{X}^n]$ depends on the distribution of $\boldsymbol{X}^n$ as well as the conditional distribution of $\boldsymbol{Y}^n$ given $\boldsymbol{X}^n$. The joint entropy $\mathsf{H}[\boldsymbol{X}^n\,\boldsymbol{Y}^n]$ can also be broken up the opposite way as

$$\mathsf{H}[\boldsymbol{X}^n\,\boldsymbol{Y}^n] = \mathsf{H}[\boldsymbol{Y}^n] + \mathsf{H}[\boldsymbol{X}^n|\,\boldsymbol{Y}^n], \tag{8.66}$$

Combining (8.65) and (8.66),it is seen that $\mathsf{H}[\boldsymbol{X}^n] - \mathsf{H}[\boldsymbol{X}^n|\,\boldsymbol{Y}^n] = \mathsf{H}[\boldsymbol{Y}^n] - \mathsf{H}[\boldsymbol{Y}^n|\boldsymbol{X}^n]$. This difference of entropies is called the *mutual information* between $\boldsymbol{X}^n$ and $\boldsymbol{Y}^n$ and denoted $I(\boldsymbol{X}^n; \boldsymbol{Y}^n)$, so

$$I(\boldsymbol{X}^n; \boldsymbol{Y}^n) = \mathsf{H}[\boldsymbol{X}^n] - \mathsf{H}[\boldsymbol{X}^n|\,\boldsymbol{Y}^n] = \mathsf{H}[\boldsymbol{Y}^n] - \mathsf{H}[\boldsymbol{Y}^n|\boldsymbol{X}^n] \tag{8.67}$$

The first expression for $I(\boldsymbol{X}^n; \boldsymbol{Y}^n)$ has a nice intuitive interpretation. $\mathsf{H}[\boldsymbol{X}^n]$ is understood from source coding as representing the number of bits required to represent the channel input. If we look at a particular sample value $\boldsymbol{y}^n$ of the output, $\mathsf{H}[\boldsymbol{X}^n|\,\boldsymbol{Y}^n{=}\boldsymbol{y}^n]$ can be interpreted as the number of bits required to represent $\boldsymbol{X}^n$ after observing the output sample value $\boldsymbol{y}^n$. Note that $\mathsf{H}[\boldsymbol{X}^n|\,\boldsymbol{Y}^n]$ is the expected value of this over $\boldsymbol{Y}^n$. Thus $I(\boldsymbol{X}^n; \boldsymbol{Y}^n)$ can be interpreted as the reduction in uncertainty, or number of required bits for specification, after passing through the channel. This intuition will lead to the converse to the noisy-channel coding theorem in the next subsection.

The second expression for $I(\boldsymbol{X}^n; \boldsymbol{Y}^n)$ is the one most easily manipulated. Taking the log of the expression in (8.64),

$$\mathsf{H}[\boldsymbol{Y}^n|\boldsymbol{X}^n] = \sum_{k=1}^{n} \mathsf{H}[Y_k|X_k] \tag{8.68}$$

Since the entropy of a sequence of random symbols is upper bounded by the sum of the corresponding terms (see Exercise 2.19)

$$\mathsf{H}[\boldsymbol{Y}^n] \leq \sum_{k=1}^{n} \mathsf{H}[Y_k] \tag{8.69}$$

Substituting this and (8.68) in (8.67),

$$I(\boldsymbol{X}^n; \boldsymbol{Y}^n) \leq \sum_{k=1}^{n} I(X_k; Y_k) \tag{8.70}$$

If the inputs are independent, then the outputs are also and (8.69) and (8.70) are satisfied with equality. The mutual information $I(X_k; Y_k)$ at each time k is a function only of the pmf for X_k, since the output probabilities conditional on the input are determined by the channel. Thus, each mutual information term in (8.70) is upper bounded by the maximum of the mutual information over the input distribution. This maximum is defined as the *capacity* of the channel,

$$C = \max_{\boldsymbol{p}} \sum_{i\in\mathcal{X}} \sum_{j\in\mathcal{Y}} p_i P_{i,j} \log \frac{P_{i,j}}{\sum_{\ell\in\mathcal{X}} p_\ell P_{\ell,j}}, \tag{8.71}$$

where $\boldsymbol{p} = (p_0, p_1, \ldots, p_{\mathcal{X}-1})$ is the set (over the alphabet $\mathcal{X}$) of input probabilities. The maximum is over this set of input probabilities, subject to $p_i \geq 0$ for each $i \in \mathcal{X}$ and $\sum_{i\in\mathcal{X}} p_i = 1$. The above function is concave in $\boldsymbol{p}$, and thus the maximimization is straight-forward; for the BSC, for example, the maximum is at $p_0 = p_1 = 1/2$ and $C = 1 + P_{0,1} \log P_{0,1} + P_{0,0} \log P_{0,0}$. Since C upper bounds $I(X_k; Y_k)$ for each k, with equality if the distribution for X_k is the maximizing distribution,

$$I(\boldsymbol{X}^n; \boldsymbol{Y}^n) \leq nC, \tag{8.72}$$

with equality if all inputs are independent and chosen with the maximizing probabilities in (8.71).

8.7.3 Converse to the noisy-channel coding theorem

Define the rate R for the DMC above as the number of iid equiprobable binary source digits that enter the channel per channel use. More specifically assume that nR bits enter the source and are transmitted over the n channel uses under discussion. Assume also that these bits are mapped into the channel input $\boldsymbol{X}^n$ in a one-to-one way. Thus $\mathsf{H}[\boldsymbol{X}^n] = nR$ and $\boldsymbol{X}^n$ can take on $M = 2^{nR}$ equiprobable values. The following theorem now bounds $\Pr(e)$ away from 0 if $R > C$.

Theorem 8.7.1. *Consider a DMC with capacity C. Assume that the rate R satisfies $R > C$. Then for any block length n, the ML probability of error,* i.e.*, the probability that the decoded n-tuple $\tilde{\boldsymbol{X}}^n$ is unequal to the transmitted n-tuple $\boldsymbol{X}^n$, is lower bounded by*

$$R - C \leq H_b(\Pr(e)) + R \Pr(e), \tag{8.73}$$

where $H_b(\alpha)$ is the binary entropy, $-\alpha \log \alpha - (1-\alpha)\log(1-\alpha)$.

Remark: The right hand side of (8.73) is 0 at $\Pr(e) = 0$ and is increasing for $\Pr(e) \leq 1/2$, so (8.73) provides a lower bound to $\Pr(e)$ that depends only on C and R.

Proof: Note that $\mathsf{H}[\boldsymbol{X}^n] = nR$ and, from (8.70) and (8.67), $\mathsf{H}(\boldsymbol{X}^n) - \mathsf{H}(\boldsymbol{X}^n | \boldsymbol{Y}^n) \leq nC$. Thus

$$\mathsf{H}(\boldsymbol{X}^n | \boldsymbol{Y}^n) \geq nR - nC. \tag{8.74}$$

For each sample value $\boldsymbol{y}^n$ of $\boldsymbol{Y}^n$, $\mathsf{H}(\boldsymbol{X}^n \mid \boldsymbol{Y}^n{=}\boldsymbol{y}^n)$ is an ordinary entropy. The received $\boldsymbol{y}^n$ is decoded into some $\tilde{\boldsymbol{x}}^n$ and the corresonding probability of error is $\Pr(\boldsymbol{X}^n \neq \tilde{\boldsymbol{x}}^n \mid \boldsymbol{Y}^n{=}\boldsymbol{y}^n)$.

As in Exercise 2.20, the entropy $\mathsf{H}(\boldsymbol{X}^n \mid \boldsymbol{Y}^n{=}\boldsymbol{y}^n)$ can be upper bounded as the sum of two terms, first the binary entropy of whether or not $\boldsymbol{X}^n = \tilde{\boldsymbol{x}}^n$, and second, the entropy of all $M-1$ possible errors in the case $\boldsymbol{X}^n \neq \tilde{\boldsymbol{x}}^n$, *i.e.*,

$$\mathsf{H}(X \mid \boldsymbol{Y}^n{=}\boldsymbol{y}^n) \leq H_b(\Pr(e|\boldsymbol{y}^n)) + \Pr(e|\boldsymbol{y}^n)\log(M-1).$$

Upper bounding $\log(M-1)$ by $\log M = nR$ and averaging over $\boldsymbol{Y}^n$,

$$\mathsf{H}(\boldsymbol{X}^n|\,\boldsymbol{Y}^n) \leq H_b(\Pr(e)) + nR\Pr(e). \tag{8.75}$$

Combining (8.74 and (8.75),

$$R - C \leq \frac{H_b(\Pr(e))}{n} + R\Pr(e),$$

and upper bounding $1/n$ by 1 yields (8.73). □

The above theorem is not entirely satisfactory, since it shows that block errors cannot be made negligible at rates above capacity, but does not rule out the possibility that each block error causes only one bit error, say, and thus the probability of bit error might go to 0 as $n \to \infty$. As shown in Theorem 4.3.4 of [7], this cannot happen, but the proof doesn't add much insight and will be omitted here.

8.7.4 noisy-channel coding theorem, forward part

There are two critical ideas in the forward part of the coding theorem. The first is to use the AEP on the joint ensemble $\boldsymbol{X}^n\boldsymbol{Y}^n$. The second, however, is what shows the true genius of Shannon. His approach, rather than an effort to find and analyze good codes, was to simply choose each codeword of a code randomly, choosing each letter in each codeword to be iid with the capacity yielding input distribution.

One would think initially that the codewords should be chosen to be maximally different in some sense, but Shannon's intuition said that independence would be enough. Some initial sense of why this might be true comes from looking at the binary orthogonal codes. Here each codeword of length n differs from each other codeword in $n/2$ positions, which is equal to the average number of differences with random choice. Another initial intuition comes from the fact that mutual information between input and output n-tuples is maximized by iid inputs. Truly independent inputs do not allow for coding constraints, but choosing a limited number of codewords using an iid distribution is at least a plausible approach. In any case, the following theorem proves that this approach works.

It clearly makes no sense for the encoder to choose codewords randomly if the decoder doesn't know what those codewords are, so we visualize the designer of the modem as choosing these codewords and building them into both transmitter and receiver. Presumably the designer is smart enough to test her code before shipping a million copies around the world, but we won't worry about that. We simply average the performance over all random choices. Thus the probability space consists of M independent iid codewords of block length n, followed by a randomly chosen message m, $0 \leq m \leq M-1$ that enters the encoder. The corresponding sample value $\boldsymbol{x}_m^n$ of the mth randomly chosen codeword is transmitted and combined with noise to yield a received sample sequence $\boldsymbol{y}^n$. The decoder then compares $\boldsymbol{y}^n$ with the M possible randomly chosen messages (the decoder knows $\boldsymbol{x}_0^n, \ldots, \boldsymbol{x}_{M-1}^n$, but doesn't know m) and chooses

the most likely of them. It appears that a simple problem has been replaced with a complex problem, but since there is so much independence between all the random symbols, the new problem is surprisingly simple.

These randomly chosen codewords and channel outputs are now analyzed with the help of the AEP. For this particular problem, however, it is simpler to use a slightly different form of AEP, called the *strong AEP*, than that of Chapter 2. The strong AEP was analyzed in Exercise 2.28 and is reviewed here. Let $\boldsymbol{U}^n = U_1, \ldots, U_n$ be an n-tuple of iid discrete random symbols with alphabet $\mathcal{U}$ and letter probabilities p_j for each $j \in \mathcal{U}$. Then for any $\varepsilon > 0$, the strongly typical set $S_\varepsilon(\boldsymbol{U}^n)$ of sample n-tuples is defined as

$$S_\varepsilon(\boldsymbol{U}^n) = \left\{ \boldsymbol{u}^n : p_j(1-\varepsilon) < \frac{N_j(\boldsymbol{u}^n)}{n} < p_j(1+\varepsilon); \quad \text{for all } j \in \mathcal{U} \right\}, \tag{8.76}$$

where $N_j(\boldsymbol{u}^n)$ is the number of appearances of letter j in the n-tuple $\boldsymbol{u}^n$. The double inequality in (8.76) will be abbreviated as $N_j(\boldsymbol{u}^n) = np_j(1 \pm \varepsilon)$, so (8.76) becomes

$$S_\varepsilon(\boldsymbol{U}^n) = \{ \boldsymbol{u}^n : N_j(\boldsymbol{u}^n) = np_j(1 \pm \varepsilon); \quad \text{for all } j \in \mathcal{U} \} \tag{8.77}$$

Thus the strongly typical set is the set of n-tuples for which each letter appears with approximately the right relative frequency. For any given ε, the law of large numbers says that $\lim_{n\to\infty} \Pr(N_j(\boldsymbol{U}^n) = p_j(1 \pm \varepsilon)) = 1$ for each j. Thus (see Exercise 2.28)

$$\lim_{n\to\infty} \Pr(\boldsymbol{U}^n \in S_\varepsilon(\boldsymbol{U}^n)) = 1. \tag{8.78}$$

Next consider the probability of n-tuples in $S_\varepsilon(\boldsymbol{U}^n)$. Note that $p_{U^n}(\boldsymbol{u}^n) = \prod_j p_j^{N_j(\boldsymbol{u}^n)}$. Taking the log of this,

$$\log p_{U^n}(\boldsymbol{u}^n) = -n\mathsf{H}(U)(1 \pm \varepsilon) \qquad \text{for } \boldsymbol{u}^n \in S_\varepsilon(\boldsymbol{U}^n). \tag{8.79}$$

Thus the strongly typical set has the same basic properties as the typical set defined in Chapter 2. Because of the requirement that each letter has a typical number of appearances, however, it has additional properties that are useful in the coding theorem below.

Consider an n-tuple of channel input/output pairs, $\boldsymbol{X}^n \boldsymbol{Y}^n = (X_1Y_1), (X_2Y_2), \ldots, (X_nY_n)$ where successive pairs are iid. For each pair, XY, let X have the pmf $\{p_i; i \in \mathcal{X}\}$ which achieves capacity in (8.71). Let the pair XY have the pmf $\{p_iP_{i,j}; i \in \mathcal{X}, j \in \mathcal{Y}\}$ where $P_{i,j}$ is the channel transition probability from input i to output j. This is the joint pmf for the randomly chosen codeword that is transmitted and the corresponding received sequence.

The strongly typical set $S_\varepsilon(\boldsymbol{X}^n \boldsymbol{Y}^n)$ is then given by (8.77) as

$$S_\varepsilon(\boldsymbol{X}^n \boldsymbol{Y}^n) = \{ \boldsymbol{x}^n\boldsymbol{y}^n : N_{ij}(\boldsymbol{x}^n\boldsymbol{y}^n) = n\, p_iP_{i,j}(1 \pm \varepsilon); \quad \text{for all } i \in \mathcal{X}, j \in \mathcal{Y} \} \tag{8.80}$$

where $N_{ij}(\boldsymbol{x}^n\boldsymbol{y}^n)$ is the number of xy pairs in $((x_1y_1), (x_2y_2), \ldots, (x_ny_n))$ for which $x = i$ and $y = j$. The transmitted codeword $\boldsymbol{X}^n$ and the received n-tuple $\boldsymbol{Y}^n$ then satisfy

$$\lim_{n\to\infty} \Pr((\boldsymbol{X}^n \boldsymbol{Y}^n) \in S_\varepsilon(\boldsymbol{X}^n \boldsymbol{Y}^n)) = 1. \tag{8.81}$$

$$\log p_{\boldsymbol{X}^n\boldsymbol{Y}^n}(\boldsymbol{x}^n\boldsymbol{y}^n) = -n\mathsf{H}(XY)(1 \pm \varepsilon) \qquad \text{for } (\boldsymbol{x}^n\boldsymbol{y}^n) \in S_\varepsilon(\boldsymbol{X}^n \boldsymbol{Y}^n). \tag{8.82}$$

The nice feature about strong typicality is that if $\boldsymbol{x}^n\boldsymbol{y}^n$ is in the set $S_\varepsilon(\boldsymbol{X}^n\boldsymbol{Y}^n)$, then $\boldsymbol{x}^n$ must be in $S_\varepsilon(\boldsymbol{X}^n)$ and $\boldsymbol{y}^n$ must be in $S_\varepsilon(\boldsymbol{Y}^n)$. To see this, assume that $(\boldsymbol{x}^n, \boldsymbol{y}^n) \in S_\varepsilon(\boldsymbol{X}^n\boldsymbol{Y}^n)$. Then

$$\begin{aligned} N_i(\boldsymbol{x}^n) &= \sum_j N_{ij}(\boldsymbol{x}^n\boldsymbol{y}^n) \\ &\in \sum_j np_iP_{ij}(1\pm\varepsilon) = np_i(1\pm\varepsilon) \qquad \text{for all } i \end{aligned}$$

Thus $\boldsymbol{x}^n \in S_\varepsilon(\boldsymbol{X}^n)$. The same argument shows that $\boldsymbol{y}^n \in S_\varepsilon(\boldsymbol{Y}^n)$.

The noisy-channel coding theorem can now be stated and proved.

Theorem 8.7.2. *Consider a DMC with capacity C and let R be any fixed rate $R < C$. Then for any $\delta > 0$, and all sufficiently large block lengths n, there exist block codes with $M \geq 2^{nR}$ equiprobable codewords such that the ML error probability satisfies $\Pr(e) \leq \delta$.*

Proof: As suggested above, we consider the error probability averaged over the random selection of codes defined above, where for given block length n and rate R, the number of codewords will be $M = \lceil 2^{nR} \rceil$. Since at least one code must be as good as the average, the theorem can be proved by showing that $\Pr(e) \leq \delta$.

The decoding rule to be used will be different than maximum likelihood, but since ML is optimum, proving that $\Pr(e) \leq \delta$ for any decoding rule will prove the theorem. The rule to be used is strong typicality. That is, for given ε to be selected later, the decoder, given $\boldsymbol{y}^n$, determines whether there is an $\tilde{m}$ for which the pair $(\boldsymbol{x}^n_{\tilde{m}}\boldsymbol{y}^n)$ lies in $S_\varepsilon(\boldsymbol{X}^n\boldsymbol{Y}^n)$. If there is exactly one $\tilde{m}$ satisfying this test, that is the decoded message; that decoded message is in error, of course, if $\tilde{m}$ differs from the transmitted message m. If no $\tilde{m}$ or multiple $\tilde{m}$ satisfy the above test, the decoding is also counted as an error, so the actual decoded value in these cases is immaterial for the proof. The probability of error, given any transmitted message m, is then upper bounded by two terms, first, $\Pr(\boldsymbol{X}^n\boldsymbol{Y}^n \notin S_\varepsilon(\boldsymbol{X}^n\boldsymbol{Y}^n))$ where $\boldsymbol{X}^n\boldsymbol{Y}^n$ is the transmitted/received pair, and second, the probability that some other codeword is jointly typical with $\boldsymbol{Y}^n$. The other codewords are independent of $\boldsymbol{Y}^n$ and each is chosen with iid symbols using the same pmf as the transmitted codeword. Let $\overline{\boldsymbol{X}}^n$ be any one of these codewords. Using the union bound,

$$\Pr(e) \leq \Pr((\boldsymbol{X}^n\boldsymbol{Y}^n) \notin S_\varepsilon(\boldsymbol{X}^n\boldsymbol{Y}^n)) + (M-1)\Pr((\overline{\boldsymbol{X}}^n\boldsymbol{Y}^n) \in S_\varepsilon(\boldsymbol{X}^n\boldsymbol{Y}^n)) \tag{8.83}$$

For any large enough n, (8.81) shows that the first term is at most $\delta/2$. Also $M - 1 \leq 2^{nR}$. Thus

$$\Pr(e) \leq \frac{\delta}{2} + 2^{nR}\Pr((\overline{\boldsymbol{X}}^n\boldsymbol{Y}^n) \in S_\varepsilon(\boldsymbol{X}^n\boldsymbol{Y}^n)) \tag{8.84}$$

To analyze the second term above, define $F(\boldsymbol{y}^n)$ as the set of input sequences $\boldsymbol{x}^n$ that are jointly typical with the given $\boldsymbol{y}^n$. This set is empty if $\boldsymbol{y}^n \notin S_\varepsilon(\boldsymbol{Y}^n)$. Note that for $\boldsymbol{y}^n \in S_\varepsilon(Y^n)$,

$$p_{\boldsymbol{Y}^n}(\boldsymbol{y}^n) \geq \sum_{\boldsymbol{x}^n \in F(\boldsymbol{y}^n)} p_{\boldsymbol{X}^n\boldsymbol{Y}^n}(\boldsymbol{x}^n\boldsymbol{y}^n) \geq \sum_{\boldsymbol{x}^n \in F(\boldsymbol{y}^n)} 2^{-n\mathsf{H}(XY)(1+\varepsilon)}$$

where the final inequality comes from (8.82). Since $p_{\boldsymbol{Y}^n}(\boldsymbol{y}^n) \leq 2^{-n\mathsf{H}(Y)(1-\varepsilon)}$ for $\boldsymbol{y}^n \in S_\varepsilon(Y^n)$, the conclusion is that the number of n-tuples in $F(\boldsymbol{y}^n)$ for any typical $\boldsymbol{y}^n$ satisfies

$$|F(\boldsymbol{y}^n)| \leq 2^{n[\mathsf{H}(XY)(1+\varepsilon) - \mathsf{H}(Y)(1-\varepsilon)]} \tag{8.85}$$

This means that the probability that $\overline{\boldsymbol{X}}^n$ lies in $F(\boldsymbol{y}^n)$ is at most the size $|F(\boldsymbol{y}^n)|$ times the maximum probability of a typical $\overline{\boldsymbol{X}}^n$ (recall that $\overline{\boldsymbol{X}}^n$ is independent of $\boldsymbol{Y}^n$ but has the same marginal distribution as $\boldsymbol{X}^n$. Thus

$$\begin{aligned}\Pr((\overline{\boldsymbol{X}}^n\,\boldsymbol{Y}^n) \in S_\varepsilon(\boldsymbol{X}^n\,\boldsymbol{Y}^n)) &\leq 2^{-n[\mathsf{H}(X)(1-\varepsilon)+\mathsf{H}(Y)(1-\varepsilon)-\mathsf{H}(XY)(1+\varepsilon)]} \\ &= 2^{-n\{C-\varepsilon[\mathsf{H}(X)+\mathsf{H}(Y)+\mathsf{H}(XY)]\}},\end{aligned}$$

where we have used the fact that $C = \mathsf{H}(X) - \mathsf{H}(X|Y) = \mathsf{H}(X) + \mathsf{H}(Y) - \mathsf{H}(XY)$. Substituting this into (8.84),

$$\Pr(e) \leq \frac{\delta}{2} + 2^{n[R-C+\varepsilon\alpha]}$$

where $\alpha = \mathsf{H}(X) + \mathsf{H}(Y) + \mathsf{H}(XY)$. Finally, choosing $\varepsilon = (C-R)/(2\alpha)$,

$$\Pr(e) \leq \frac{\delta}{2} + 2^{-n(C-R)/2} \leq \delta$$

for sufficiently large n. □

The above proof is essentially the original proof given by Shannon, with a little added explanation of details. It will be instructive to explain the essence of the proof without any of the epsilons or deltas. The transmitted and received n-tuple pair $(\boldsymbol{X}^n\,\boldsymbol{Y}^n)$ is typical with high probability and the typical pairs essentially have probability $2^{-n\mathsf{H}(XY)}$ (including both the random choice of $\boldsymbol{X}^n$ and the random noise). Each typical output $\boldsymbol{y}^n$ essentially has a marginal probability $2^{-n\mathsf{H}(Y)}$. For each typical $\boldsymbol{y}^n$, there are essentially $2^{n\mathsf{H}(X|Y)}$ input n-tuples that are jointly typical with $\boldsymbol{y}^n$ (this is the nub of the proof). An error occurs if any of these are selected to be codewords (other than the actual transmitted codeword). Since there are about $2^{n\mathsf{H}(X)}$ typical input n-tuples altogether, a fraction $2^{-nI(X;Y)} = 2^{-nC}$ of them are jointly typical with the given received $\boldsymbol{y}^n$.

More recent proofs of the noisy-channel coding theorem also provide much better upper bounds on error probability. These bounds are exponentially decreasing with n with a rate of decrease that typically becomes vanishingly small as $R \to C$.

8.7.5 The noisy-channel coding theorem for WGN

The coding theorem for DMC's can be easily extended to discrete-time channels with arbitrary real or complex input and output alphabets, but doing this with mathematical generality and precision is difficult with our present tools.

This is done here for the discrete time Gaussian channel, which will make clear the conditions under which this generalization is easy. Let X_k and Y_k be the input and output to the channel at time k, and assume that $Y_k = X_k + Z_k$ where $Z_k \sim \mathcal{N}(0, N_0/2)$ is independent of X_k and independent of the signal and noise at all other times. Assume the input is constrained in second moment to $\mathsf{E}[X_k^2] \leq E$, so $\mathsf{E}[Y^2] \leq E + N_0/2$.

From Exercise 3.8, the differential entropy of Y is then upper bounded by

$$\mathsf{h}(Y) \leq \frac{1}{2}\log(2\pi e(E + N_0/2). \tag{8.86}$$

This is satisfied with equality if Y is $\mathcal{N}(0, E + N_0/2)$, and thus if X is $\mathcal{N}(0, E)$. For any given input x, $\mathsf{h}(Y|X=x) = \frac{1}{2}\log(2\pi e N_0/2)$, so averaging over the input space,

$$\mathsf{h}(Y|X) = \frac{1}{2}\log(2\pi e N_0/2). \tag{8.87}$$

By analogy with the DMC case, let the capacity C (in bits per channel use) be defined as the maximum of $\mathsf{h}(Y)-\mathsf{h}(Y|X)$ subject to the second moment constraint E. Thus, combining (8.86) and (8.87),

$$C = \frac{1}{2}\log\left(1+\frac{2E}{N_0}\right) \tag{8.88}$$

Theorem 8.7.2 applies quite simply to this case. For any given rate R in bits per channel use such that $R < C$, one can quantize the channel input and output space finely enough so that the corresponding discrete capacity is arbitrarily close to C and in particular larger than R. Then Theorem 8.7.2 applies, so rates arbitrarily close to C can be transmitted with arbitrarily high reliability. The converse to the coding theorem can also be extended.

For a discrete time WGN channel using $2W$ degrees of freedom per second and a power constraint P, the second moment constraint on each degree of freedom[13] becomes $E = P/(2W)$ and the capacity C_t in bits per second becomes Shannon's famous formula

$$C_t = W\log\left(1+\frac{P}{WN_0}\right). \tag{8.89}$$

This is then the capacity of a WGN channel with input power constrained to P and degrees of freedom per second constrained to $2W$.

With some careful interpretation, this is also the capacity of a continuous-time channel constrained in bandwidth to W and in power to P. The problem here is that if the input is strictly constrained in bandwidth, no information at all can be transmitted. That is, if a single bit is introduced into the channel at time 0, the difference in the waveform generated by symbol 1 and that generated by symbol 0 must be 0 before time 0, and thus, by the Paley-Wiener theorem, cannot be nonzero and strictly bandlimited. From an engineering perspective, this doesn't seem to make sense, but the waveforms used in all engineering systems have negligible but non-zero energy outside the nominal bandwidth.

Thus, to use (8.89) for a bandlimited input, it is necessary to start with the constraint that for any given $\eta > 0$, at least a fraction $(1-\eta)$ of the energy must lie within a bandwidth W. Then reliable communication is possible at all rates R_t in bits per second less than C_t as given in (8.89). Since this is true for all $\eta > 0$, no matter how small, it makes sense to call this the capacity of the bandlimited WGN channel. This is not an issue in the design of a communication system, since filters must be used and it is widely recognized that they can't be entirely bandlimited.

8.8 Convolutional codes

The theory of coding, and particularly of coding theorems, concentrate on block codes, but convolutional codes are also widely used and have essentially no block structure. These codes can be used whether bandwidth is highly constrained or not. We give an example below where there are two output bits for each input bit. Such a code is said to have rate 1/2 (in input bits per channel bit). More generally, such codes produce an m-tuple of output bits for each b-tuple of input bits for arbitrary integers $0 < b < m$. These codes are said to have rate b/m.

[13] We were careless in not specifying whether the constraint must be satisfied for each degree of freedom or overall as a time-average. It is not hard to show, however, that the mutual information is maximized when the same energy is used in each degree of freedom.

A convolutional code looks very much like a discrete filter. Instead of having a single input and output stream, however, we have b input streams and m output streams. For the example given here, the number of input streams is $b = 1$ and the number of output streams is $m = 2$, thus producing two output bits per input bit. There is another difference between a convolutional code and a discrete filter; the inputs and outputs for a convolutional code are binary and the addition is modulo 2. Consider the example below in Figure 8.8.

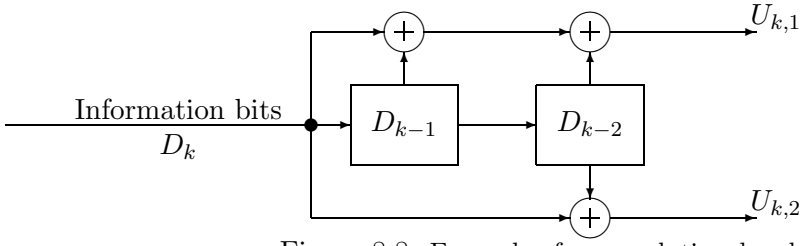

Figure 8.8: Example of a convolutional code

For the example above, the equations for the outputs are

$$\begin{aligned} U_{k,1} &= D_k \oplus D_{k-1} \oplus D_{k-2} \\ U_{k,2} &= D_k \qquad\quad \oplus D_{k-2}. \end{aligned}$$

Thus each of the two output streams are linear modulo two convolutions of the input stream. This encoded pair of binary streams can now be mapped into a pair of signal streams such as antipodal signals $\pm a$. This pair of signal streams can then be interleaved and modulated by a single stream of Nyquist pulses at twice the rate. This baseband waveform can then be modulated to passband and transmitted.

The structure of this code can be most easily visualized by a "trellis" diagram as illustrated in Figure 8.9.

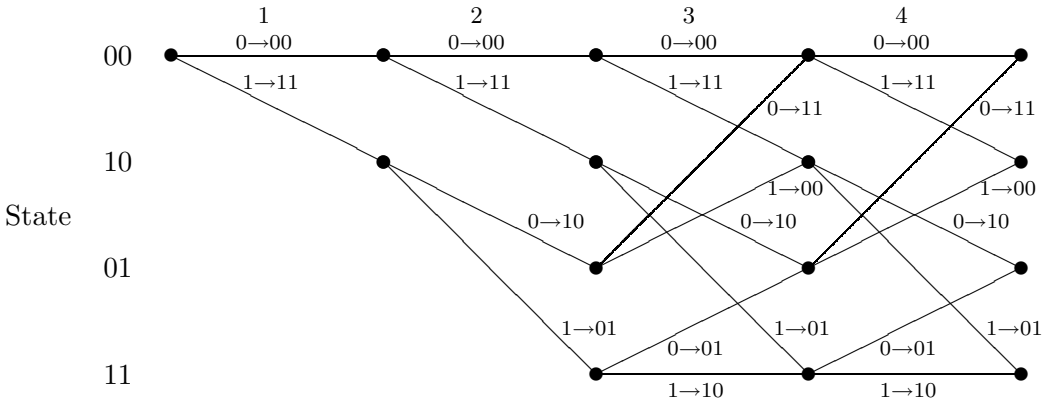

Figure 8.9: Trellis Diagram; each transition is labeled with the input and corresponding output

To understand this trellis diagram, note from Figure 8.8 that the encoder is characterized at any epoch k by the previous binary digits, D_{k-1} and D_{k-2}. Thus the encoder has four possible states, corresponding to the four possible values of the pair D_{k-1}, D_{k-2}. Given any of these four states, the encoder output and the next state depend only on the current binary input.

Figure 8.9 shows these four states arranged vertically and shows time horizontally. We assume the encoder starts at epoch 0 with $D_{-1} = D_{-2} = 0$.

In the convolutional code of the above example, the output at epoch k depends on the current input and the previous two inputs. In this case, the *constraint length* of the code is 2. In general the output could depend on the input and the previous n inputs, and the constraint length is then defined to be n. If the constraint length is n (and a single binary digit enters the encoder at each epoch k), then there are 2^n possible states, and the trellis diagram contains 2^n nodes at each time instant rather than 4.

As we have described convolutional codes above, the encoding starts at time 1 and then continues forever. In practice, because of packetization of data and various other reasons, the encoding usually comes to an end after some large number, say k_0, of binary digits have been encoded. After D_{k_0} enters the encoder, two final 0's enter the encoder, at epochs (k_0+1) and (k_0+2), and 4 final encoded digits come out of the encoder. This restores the state of the encoder to state 0, which, as we see later, is very useful for decoding. For the more general case with a constraint length of n, we need n final zeros to restore the encoder to state 0. Altogether, k_0 inputs lead to $2(k_0 + n)$ outputs, for a code rate of $k_0/[2(k_0 + n)]$. Since k_0 is usually large relative to n, this is still referred to as a rate 1/2 code. Figure 8.10 below shows the part of the trellis diagram corresponding to this termination.

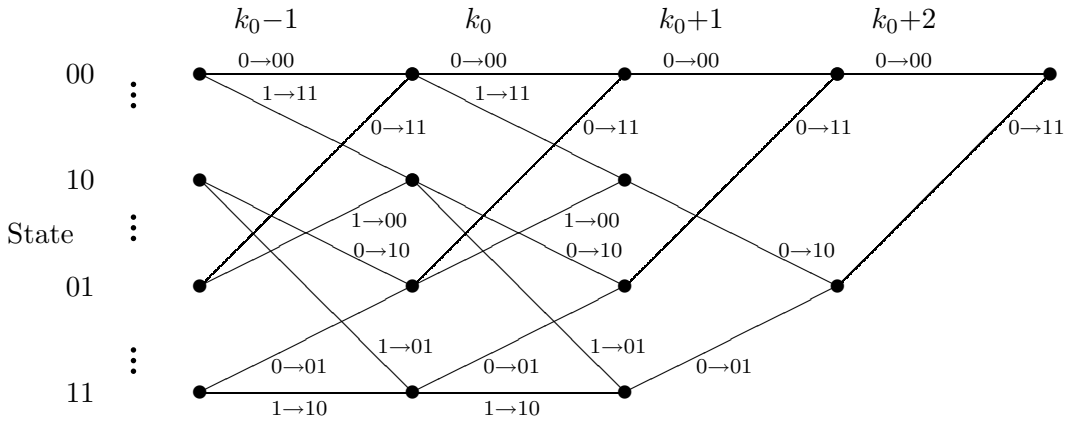

Figure 8.10: Trellis Termination

8.8.1 Decoding of convolutional codes

Decoding a convolutional code is essentially the same as using detection theory to choose between each pair of codewords, and then choosing the best overall (the same as done for the orthogonal code). There is one slight conceptual difference in that, in principle, the encoding continues forever. When the code is terminated, however, this problem does not exist, and in principle one takes the maximum likelihood choice of all the (finite length) possible codewords.

As usual, assume that the incoming binary digits are iid and equiprobable. This is reasonable if the incoming bit stream has been source encoded. This means that the codewords out to any given length are equally likely, which then justifies maximum likelihood (ML) decoding.

ML detection is also used so that codes for error correction can be designed independently of the

source data to be transmitted. For all the codes under discussion, the error probability using ML decoding is independent of the transmitted codeword. Thus ML decoding is robust in the sense that the error probability is independent of the probability distribution of the incoming bits.

Another issue, given iid inputs, is determining what is meant by probability of error. In all of the examples above, given a received sequence of symbols, we have attempted to choose the codeword that minimizes the probability of error for the entire codeword. An alternative would have been to minimize the probability of error individually for each binary information digit. It turns out to be easier to minimize the sequence error probability than the bit error probability. This in fact is what happens when we use ML detection between codewords, as suggested above.

In decoding for error correction, the objective is almost invariably to minimize the sequence probability of error. Along with the convenience suggested above, a major reason is that a binary input is usually a source coded version of some other source sequence or waveform, and thus a single output error is often as serious as multiple errors within a codeword. ML detection on sequences is assumed in what follows.

8.8.2 The Viterbi algorithm

The Viterbi algorithm is an algorithm for performing ML detection for convolutional codes. Assume for the time being that the code is terminated as in Figure 8.10. It will soon be seen that whether or not the code is terminated is irrelevant. The algorithm will now be explained for the example above and for the assumption of WGN; the extension to arbitrary convolutional codes will be obvious except for the notational complexity of the general case. For any given input $d_1, \ldots, d_{k_0}$, let the encoded sequence be $u_{1,1}, u_{1,2}, u_{2,1}, u_{2,2} \ldots, u_{k_0+2,2}$ and let the channel output, after modulation, addition of WGN, and demodulation, be $v_{1,1}, v_{1,2}, v_{2,1}, v_{2,2} \ldots, v_{k_0+2,2}$.

There are 2^{k_0} possible codewords, corresponding to the 2^{k_0} possible binary k_0-tuples $d_1, \ldots, d_{k_0}$, so an unimaginative approach to decoding would be to compare the likelihood for each of these codewords. For large k_0, even with today's technology, such an approach would be prohibitive. It turns out, however, that by using the trellis structure of Figure 8.9, this decoding effort can be greatly simplified.

Each input $d_1, \ldots, d_{k_0}$ (*i.e.*, each codeword) corresponds to a particular path through the trellis from epoch 1 to k_0+2, and each path, at each epoch k, corresponds to a particular trellis state.

Consider two paths $d_1, \ldots, d_{k_0}$ and $d'_1, \ldots, d'_{k_0}$ through the trellis that pass through the same state at time k^+ (*i.e.*, at the time immediately after the input and state change at epoch k) and remain together thereafter. Thus $d_{k+1}, \ldots, d_{k_0} = d'_{k+1}, \ldots, d'_{k_0}$. For example, from Figure 8.8, we see that $(0, \ldots, 0)$ and $1, 0, \ldots, 0$ are both in state 00 at 3^+ and both remain in the same state thereafter. Since the two paths are in the same state at k^+ and have the same inputs after this time, they both have the same encoder outputs after this time. Thus $u_{k+1,i}, \ldots, u_{k_0+2,i} = u'_{k+1,i}, \ldots, u'_{k_0+2,i}$ for $i = 1, 2$.

Since each channel output rv $V_{k,i}$ is given by $V_{k,i} = U_{k,i} + Z_{k,i}$ and the Gaussian noise variables $Z_{k,i}$ are independent, this means that for any channel output $v_{1,1}, \ldots, v_{k_0+2,2}$,

$$\frac{f(v_{1,1}, \ldots, v_{k_0+2,2} | d_1, \ldots, d_{k_0})}{f(v_{1,1}, \ldots, v_{k_0+2,2} | d'_1, \ldots, d'_{k_0})} = \frac{f(v_{1,1}, \ldots, v_{k,2} | d_1, \ldots, d_{k_0})}{f(v_{1,1}, \ldots, v_{k,2} | d'_1, \ldots, d'_{k_0})}.$$

In plain English, this says that if two paths merge at time k^+ and then stay together, the

likelihood ratio depends on only the first k output pairs. Thus if the right hand side exceeds 1, then $d_1, \ldots, d_{k_0}$ is more likely than $d'_1, \ldots, d'_{k_0}$. This conclusion holds no matter how the final inputs $d_{k+1} \ldots, d_{k_0}$ are chosen.

We then see that when two paths merge at a node, no matter what the remainder of the path is, the most likely of the paths is the one that is most likely at the point of the merger. Thus, whenever two paths merge, the least likely of the paths can be eliminated at that point. Doing this elimination successively from the smallest k for which paths merge (3 for the example), there is only one survivor for each state at each epoch.

To be specific, let $h(d_1, \ldots, d_k)$ be the state at time k^+ with input $d_1, \ldots, d_k$. For the example, $h(d_1, \ldots, d_k) = (d_{k-1}, d_k)$. Let

$$f_{\max}(k, s) = \max_{h(d_1,\ldots,d_k)=s} f(v_{1,1}, \ldots, v_{k,2} | d_1, \ldots, d_k).$$

These quantities can then be calculated iteratively for each state and each time k by the iteration

$$f_{\max}(k+1, s) = \max_{r:r\to s} f_{\max}(k, r) \cdot f(v_{k,1} | u_1(r\to s)) f(v_{k,2} | u_2(r\to s)). \tag{8.90}$$

where the maximization is over the set of states r that have a transition to state s in the trellis and $u_1(r\to s)$ and $u_2(r\to s)$ are the two outputs from the encoder corresponding to a transition from r to s.

This expression is simplified (for WGN) by taking the log, which is proportional to the negative squared distance between $\boldsymbol{v}$ and $\boldsymbol{u}$. For the antipodal signal case in the example, this is further simplified by simply taking the dot product between $\boldsymbol{v}$ and $\boldsymbol{u}$. Letting $L(k, s)$ be this dot product,

$$L(k+1, s) = \max_{r:r\to s} L(k, r) + v_{k,1} u_1(r\to s)) + v_{k,2} u_2(r\to s)). \tag{8.91}$$

What this means is that at each epoch (k+1), it is necessary to calculate the inner product in (8.91) for each link in the trellis going from k to $k+1$. These must be maximized over r for each state s at epoch (k+1). The maximum must then be saved as $L(k+1, s)$ for each s. One must, of course, also save the paths taken in arriving at each merging point.

Those familiar with dynamic programming will recognize this as an example of the dynamic programming principle.

The entire computation for decoding a block of k_0 information bits is proportional to $4(k_0+2)$. In the more general case where the constraint length of the convolutional coder is n rather than 2, there are 2^n states and the computation is proportional to $2^n(k_0+n)$. The Viterbi algorithm is usually used in cases where the constraint length is moderate, say 6 - 12, and in these situations, the computation is quite moderate, expecially compared with 2^{k_0}.

Usually one does not wait until the end of the block to start decoding. Usually when the above computation is done at epoch k, all the paths up to k' have merged for k' a few constraint lengths less than k. In this case, one can decode without any bound on k_0, and the error probability is viewed in terms of "error events" rather than block error.

8.9 Summary

This chapter analyzed the last major segment of a general point-to-point communication system in the presence of noise, namely how to detect the input signals from the noisy version presented

at the output. Initially the emphasis was on detection alone, *i.e.*, the assumption was that the rest of the system had been designed and the only question remaining was how to extract the signals.

At a very general level, the problem of detection in this context is trivial. That is, under the assumption that the statistics of the input and the noise are known, the sensible problem is maximum a posteriori probability decoding: find the a posteriori probability of all the hypotheses and choose the largest. This is somewhat complicated by questions of whether to do sequence detection or bit detection, but these questions in a sense are details.

At a more specific level, however, the detection problem led to many interesting insights and simplifications, particularly for WGN channels. A particularly important simplification is the principle of irrelevance, which says that components of the received waveform in degrees of freedom not occupied by the signal of interest (or statistically related signals) can be ignored in detection of those signals. Looked at in another way, this said that matched filters could be used to extract the degrees of freedom of interest.

The last part of the chapter introduced coding and decoding. The focus changed here from decoding/detection to the question of how coding could change the input waveforms so as to make the decoding more effective. In other words, a MAP detector can be designed for any signal structure, but the real problem is to design both signal structure and detection for effective performance.

At this point, the noisy-channel coding theorem came into the picture. If $R < C$, then the probability of error can be reduced arbitrarily by increasing block length (or constraint length in the case of convolutional codes). This means that there is no "optimal" solution to the joint problem of choosing signal structure and detection, but rather a trade-off between error probability, delay, and complexity.

Thus the problem must involve not only overcoming the noise, but doing this with reasonable delay and complexity. The following chapter considers some of these problems in the context of wireless communication.

8A Appendix: Neyman-Pearson threshold tests

We have seen above that any binary MAP test can be formulated as a comparison of a likelihood ratio with a threshold. It turns out that many other detection rules can also be viewed as threshold tests on likelihood ratios. One of the most important binary detection problems for which a threshold test turns out to be essentially optimum is the *Neyman-Pearson test*. This is often used in those situations in which there is no sensible way to choose a priori probabilities. In the Neyman-Pearson test, an acceptable value α is established for $\Pr\{e \,|\, U{=}1\}$, and, subject to the constraint, $\Pr\{e \,|\, U{=}1\} \leq \alpha$, a Neyman-Pearson test is a test that minimizes $\Pr\{e \,|\, U{=}0\}$. We shall show in what follows that such a test is essentially a threshold test. Before demonstrating this, we need some terminology and definitions.

Define $q_0(\eta)$ to be $\Pr\{e \,|\, U{=}0\}$ for a threshold test with threshold η, $0 < \eta < \infty$ and similarly define $q_1(\eta)$ as $\Pr\{e \,|\, U{=}1\}$. Thus for $0 < \eta < \infty$,

$$q_0(\eta) = \Pr\{\Lambda(V){<}\eta \,|\, U{=}0\}; \qquad q_1(\eta) = \Pr\{\Lambda(V){\geq}\eta \,|\, U{=}1\}. \tag{8.92}$$

Define $q_0(0)$ as $\lim_{\eta\to 0} q_0(\eta)$ and $q_1(0)$ as $\lim_{\eta\to 0} q_1(\eta)$. Clearly $q_0(0) = 0$ and in typical situations

$q_1(0) = 1$. More generally, $q_1(0) = \Pr\{\Lambda(V){>}0|U{=}1\}$. In other words, $q_1(0)$ is less than 1 if there is some set of observations that are impossible under $U{=}0$ but have positive probability under $U{=}1$. Similarly, define $q_0(\infty)$ as $\lim_{\eta\to\infty} q_0(\eta)$ and $q_1(\infty)$ as $\lim_{\eta\to\infty} q_1(\eta)$. We have $q_0(\infty) = \Pr\{\Lambda(V) < \infty\}$ and $q_1(\infty) = 0$.

Finally, for an arbitrary test A, threshold or not, denote $\Pr\{e\,|\,U{=}0\}$ as $q_0(A)$ and $\Pr\{e\,|\,U{=}1\}$ as $q_1(A)$.

Using (8.92), we can plot $q_0(\eta)$ and $q_1(\eta)$ as parametric functions of η; we call this the *error curve*.[3] Figure 8.11 illustrates this error curve for a typical detection problem such as (8.17) and (8.18) for antipodal binary signalling. We have already observed that, as the threshold η is increased, the set of v mapped into $\tilde{U}{=}0$ decreases, thus increasing $q_0(\eta)$ and decreasing $q_1(\eta)$. Thus, as η increases from 0 to ∞, the curve in Figure 8.11 moves from the lower right to the upper left.

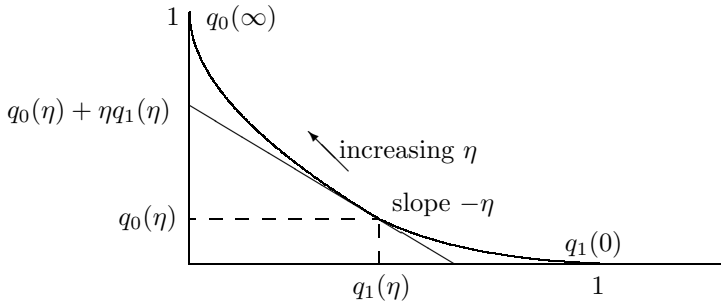

Figure 8.11: The error curve; $q_1(\eta)$ and $q_0(\eta)$ as parametric functions of η

Figure 8.11 also shows a straight line of slope $-\eta$ through the point $(q_1(\eta), q_0(\eta))$ on the error curve. The following lemma shows why this line is important.

Lemma 1: For each η, $0{<}\eta{<}\infty$, the line of slope $-\eta$ through the point $(q_1(\eta), q_0(\eta))$ lies on or beneath all other points $(q_1(\eta'), q_0(\eta'))$ on the error curve, and also lies beneath $(q_1(A), q_0(A))$ for all tests A.

Before proving this lemma, we give an example of the error curve for a discrete observation space.

Example of Discrete Observations: Figure 8.12 shows the error curve for an example in which the hypotheses 0 and 1 are again mapped $0 \to +a$ and $1 \to -a$. Assume that the observation V can take on only four discrete values $+3, +1, -1, -3$. The probabilities of each these values, conditional on $U{=}0$ and $U{=}1$, are given in the figure. As indicated there, the likelihood ratio $\Lambda(v)$ then takes the values 4, 3/2, 2/3, and 1/4, corresponding respectively to $v = 3, 1, -1$, and -3.

A threshold test at η decides $\tilde{U} = 0$ if and only if $\Lambda(V) \geq \eta$. Thus, for example, for any $\eta \leq 1/4$, all possible values of v are mapped into $\tilde{U} = 0$. In this range, $q_1(\eta) = 1$ since $U = 1$ always causes an error. Also $q_0(\eta) = 0$ since $U = 0$ never causes an error. In the range $1/4 < \eta \leq 2/3$, since $\Lambda(-3) = 1/4$, the value -3 is mapped into $\tilde{U} = 1$ and all other values into $\tilde{U} = 0$. In this range, $q_1(\eta) = 0.6$ since, when $U = 1$, an error occurs unless $V = -3$.

[3] In the radar field, one often plots $1 - q_0(\eta)$ as a function of $q_1(\eta)$. This is called the receiver operating characteristic (ROC). If one flips the error curve vertically around the point 1/2, the ROC results.

In the same way, all threshold tests with $2/3 < \eta \leq 3/2$ give rise to the decision rule that maps -1 and -3 into $\tilde{U} = 1$ and 1 and 3 into $\tilde{U} = 0$. In this range $q_1(\eta) = q_0(\eta) = 0.3$. As shown, there is another decision rule for $3/2 < \eta \leq 4$ and a final decision rule for $\eta > 4$.

v	$P(v\|0)$	$P(v\|1)$	$\Lambda(v)$
3	0.4	0.1	4
1	0.3	0.2	3/2
-1	0.2	0.3	2/3
-3	0.1	0.4	1/4

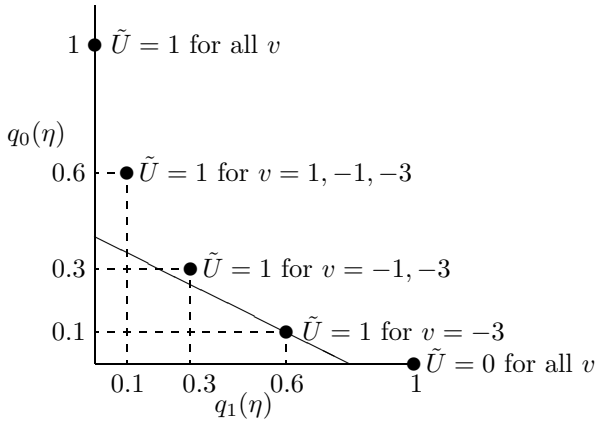

Figure 8.12: The error curve for a discrete observation space. There are only five points making up the 'curve,' one corresponding to each of the five distinct threshold rules. For example, the threshold rule $\tilde{U} = 1$ only for $v = -3$, yields $(q_1(\eta), q_0(\eta)) = (0.6, 0.1)$ for all η in the range 1/4 to 2/3. A straight line of slope $-\eta$ through that point is also shown for $\eta = 1/2$. The lemma asserts that this line lies on or beneath each point of the error curve and each point $(q_1(A), q_0(A)$ for any other test. Note that as η increases or decreases, this line will rotate around the point $(0.6, 0.1)$ until η becomes larger than 2/3 or smaller than 1/4, and then starts to rotate around the next point in the error curve.

The point of this example is that a finite observation space leads to an error curve that is simply a finite set of points. It is also possible for a continuously varying set of outputs to give rise to such an error curve when there are only finitely many possible likelihood ratios. The figure illustrates what the lemma means for error curves consisting only of a finite set of points.

Proof of lemma: Consider the line of slope $-\eta$ through the point $(q_1(\eta), q_0(\eta))$. From plane geometry, as illustrated in Figure 8.11, we see that the vertical axis intercept of this line is $q_0(\eta)+\eta q_1(\eta)$. To interpret this line, define p_0 and p_1 as a priori probabilities such that $\eta = p_1/p_0$. The overall error probability for the corresponding MAP test is then

$$\begin{aligned} q(\eta) &= p_0 q_0(\eta) + p_1 q_1(\eta) \\ &= p_0\ [q_0(\eta) + \eta q_1(\eta)]; \quad \eta = p_1/p_0. \end{aligned} \tag{8.93}$$

Similarly, the overall error probability for an arbitrary test A with the same a priori probabilities is

$$q(A) = p_0\ [q_0(A) + \eta q_1(A)]. \tag{8.94}$$

From Theorem 8.1.1, $q(\eta) \leq q(A)$, so, from (8.93) and (8.94),

$$q_0(\eta) + \eta\, q_1(\eta) \leq q_0(A) + \eta\, q_1(A). \tag{8.95}$$

We have seen that the left side of (8.95) is the vertical axis intercept of the line of slope $-\eta$ through $(q_1(\eta), q_0(\eta))$. Similarly, the right side is the vertical axis intercept of the line of slope

$-\eta$ through $(q_1(A), q_0(A))$. This says that the point $(q_1(A), q_0(A))$ lies on or above the line of slope $-\eta$ through $(q_1(\eta), q_0(\eta))$. This applies to every test A, which includes every threshold test. □

The lemma shows that if the error curve gives $q_0(\eta)$ as a differentiable function of $q_1(\eta)$ (as in the case of Figure 8.11), then the line of slope $-\eta$ through $(q_1(\eta), q_0(\eta))$ is a tangent, at point $(q_1(\eta), q_0(\eta))$, to the error curve. Thus in what follows we denote this line as the *η-tangent* to the error curve. Note that the error curve of Figure 8.12 is not really a curve at all, but the η-tangent, as defined above and illustrated in the figure for $\eta = 2/3$, still lies on or beneath all points of the error curve and all achievable points $(q_1(A), q_0(A))$, as proven above.

Since, for each test A, the point $(q_1(A), q_0(A))$ lies on or above each η-tangent, it also lies on or above the supremum of these η-tangents over $0 < \eta < \infty$. It also follows, then, that for each η', $0 < \eta' < \infty$, $(q_1(\eta'), q_0(\eta'))$ lies on or above this supremum. Since $(q_1(\eta'), q_0(\eta'))$ also lies on the η'-tangent, it lies on or beneath the supremum, and thus must lie on the supremum. We conclude that each point of the error curve lies on the supremum of the η-tangents.

Although all points of the error curve lie on the supremum of the η-tangents, all points of the supremum are not necessarily points of the error curve, as seen from Figure 8.12. We shall see shortly, however, that all points on the supremum are achievable by a simple extension of threshold tests. Thus we call this supremum the *extended error curve*.

For the example in Figure 8.11 the extended error curve is the same as the error curve itself. For the discrete example in Figure 8.12, the extended error curve is shown in Figure 8.13.

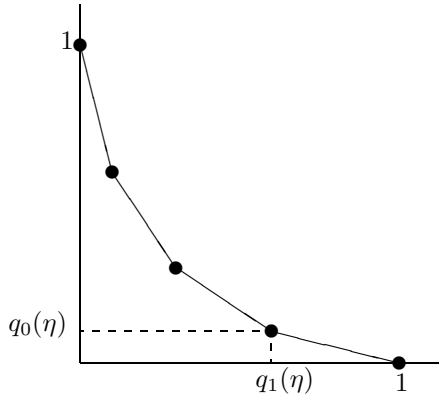

Figure 8.13: The extended error curve for the discrete observation example of Figure 8.12. From Lemma 1, for each slope $-\eta$, the η-tangent touches the error curve. Thus, the line joining two adjacent points on the error curve must be an η-tangent for its particular slope, and therefore must lie on the extended error curve.

To understand the discrete case better, assume that the extended error function has a straight line portion of slope $-\eta^*$ and horizontal extent γ. This implies that the distribution function of $\Lambda(V)$ given $U{=}1$ has a discontinuity of magnitude γ at η^*. Thus there is a set $\mathcal{V}^*$ of one or more v with $\Lambda(v) = \eta^*$, $\Pr\{\mathcal{V}^*|U{=}1\} = \gamma$, and $\Pr\{\mathcal{V}^*|U{=}0\} = \eta^*\gamma$. For a MAP test with threshold η^*, the overall error probability is not effected by whether $v \in \mathcal{V}^*$ is detected as $\tilde{U}{=}0$ or $\tilde{U}{=}1$. Our convention is to detect $v \in \mathcal{V}^*$ as $\tilde{U}{=}0$, which corresponds to the lower right point on the straight line portion of the extended error curve. The opposite convention, detecting $v \in \mathcal{V}^*$ as $\tilde{U}{=}1$ reduces the error probability given $U{=}1$ by γ and increases the error probability given $U{=}0$ by $\eta^*\gamma$, *i.e.*, it corresponds to the upper left point on the straight line portion of the extended error curve.

Note that when we were interested in MAP detection, it made no difference how $v \in \mathcal{V}^*$ was detected for the threshold η^*. For the Neyman-Pearson test, however, it makes a great deal of difference since $q_0(\eta^*)$ and $q_1(\eta^*)$ are changed. In fact, any point on the straight line in question can be achieved by detecting $v \in \mathcal{V}^*$ randomly. As the probabiliity of choosing $\tilde{U}$=0 is increased from 0 to 1 (given $v \in \mathcal{V}^*$), the point $(q_0(\eta), q_1(\eta))$ moves from the upper left to lower right end of the given line segment. In other words, the extended error curve is the curve relating q_1 to q_0 using a randomized threshold test. For a given η^*, of course, only those $v \in \mathcal{V}^*$ are detected randomly.

To summarize, the Neyman-Pearson test is a randomized threshold test. For a constraint α on $\Pr\{e|U{=}1\}$, we choose the point α on the abscissa of the extended error curve and achieve the corresponding ordinate as the minimum $\Pr\{e|U{=}1\}$. If that point on the extended error curve lies within a straight line segment of slope η^*, a randomized test is used for those observations with likelihood ratio η^*.

Since the extended error curve is a supremum of straight lines, it is a convex function. Since these straight lines all have negative slope, it is a monotonic decreasing[14] function. Thus, Figures 8.11 and 8.13 represent the general behavior of extended error curves, with the slight possible exception mentioned above that the end points need not have one of the error probabilities equal to 1.

The following theorem summarizes the results about Neyman-Pearson tests.

Theorem 8A.1. *The extended error curve is convex and strictly decreasing between* $(q_1(\infty), q_0(\infty))$ *and* $(q_1(0), q_0(0))$. *For a constraint* α *on* $\Pr\{e|U{=}1\}$, *the minimum value of* $\Pr\{e|U{=}0\}$ *is given by the ordinate of the extended error curve corresponding to the abscissa* α *and is achieved by a randomized threshold test.*

There is one more interesting variation on the theme of threshold tests. If the a priori probabilities are unknown, we might want to minimize the maximum probability of error. That is, we visualize choosing a test followed by nature choosing a priori probabilities to maximize the probability of error. Our objective is to minimize the probability of error under this worst case assumption. The resulting test is called a minmax test. It can be seen geometrically from Figures 8.11 or 8.13 that the minmax test is the randomized threshold test at the intersection of the extended error curve with a 45° line from the origin.

If there is symmetry between $U = 0$ and $U = 1$ (as in the Gaussian case), then the extended error curve will be symmetric around the 45° degree line, and the threshold will be at $\eta = 1$ (*i.e.*, the ML test is also the minmax test). This is an important result for Gaussian communication problems, since it says that ML detection, *i.e.*, minimum distance detection is robust in the sense of not depending on the input probabilities. If we know the a priori probabilities, we can do better than the ML test, but we can do no worse.

[14]To be more precise, it is strictly decreasing between the end points $(q_1(\infty), q_0(\infty))$ and $(q_1(0), q_0(0))$.

8.E Exercises

8.1. (Binary minimum cost detection) (a) Consider a binary hypothesis testing problem with a priori probabilities p_0, p_1 and likelihoods $f_{V|U}(v|i)$, $i = 0, 1$. Let C_{ij} be the cost of deciding on hypothesis j when i is correct. Conditional on an observation $V = v$, find the expected cost (over $U = 0, 1$) of making the decision $\tilde{U} = j$ for $j = 0, 1$. Show that the decision of minimum expected cost is given by

$$\tilde{U}_{\text{mincost}} = \arg\,\min_j \Big[C_{0j} p_{U|V}(0|v) + C_{1j} p_{U|V}(1|v)\Big]$$

(b) Show that the min cost decision above can be expressed as the following threshold test:

$$\Lambda(v) = \frac{f_{V|U}(v\,|\,0)}{f_{V|U}(v\,|\,1)} \;\underset{\tilde{U}=1}{\overset{\tilde{U}=0}{\gtrless}}\; \frac{p_1(C_{10} - C_{11})}{p_0(C_{01} - C_{00})} = \eta.$$

(c) Interpret the result above as saying that the only difference between a MAP test and a minimum cost test is an adjustment of the threshold to take account of the costs. *i.e.*, a large cost of an error of one type is equivalent to having a large a priori probability for that hypothesis.

8.2. Consider the following two equiprobable hypotheses:

$$\begin{aligned} U = 0 \quad &: \quad V_1 = a\cos\Theta + Z_1, \;\; V_2 = a\sin\Theta + Z_2, \\ U = 1 \quad &: \quad V_1 = -a\cos\Theta + Z_1, \;\; V_2 = -a\sin\Theta + Z_2. \end{aligned}$$

Z_1 and Z_2 are iid $\mathcal{N}(0, \sigma^2)$, and Θ takes on the values $\{-\pi/4, 0, \pi/4\}$ each with probability $1/3$.

Find the ML decision rule when V_1, V_2 are observed.

Hint: Sketch the possible values of V_1, V_2 for $\boldsymbol{Z} = 0$ given each hypothesis. Then, without doing any calculations try to come up with a good intuitive decision rule. Then try to verify that it is optimal.

8.3. Let

$$V_j = S_j X_j + Z_j \qquad \text{for } 1 \le j \le 4$$

where $\{X_j; 1 \le j \le 4\}$ are iid $\mathcal{N}(0, 1)$ and $\{Z_j; 1 \le j \le 4\}$ are iid $\mathcal{N}(0, \sigma^2)$ and independent of $\{X_j; 1 \le j \le 4\}$. $\{V_j; 1 \le j \le 4\}$ are observed at the output of a communication system and the input is a single binary random variable U which is independent of $\{Z_j; 1 \le j \le 4\}$ and $\{X_j; 1 \le j \le 4\}$. Given that $U = 0$, $S_1 = S_2 = 1$ and $S_3 = S_4 = 0$. Given $U = 1$, $S_1 = S_2 = 0$ and $S_3 = S_4 = 1$.

(a) Find the log likelihood ratio

$$\text{LLR}(\boldsymbol{v}) = \ln\left(\frac{f_{\boldsymbol{V}|U}(\boldsymbol{v}|0}{f_{\boldsymbol{V}|U}(\boldsymbol{x}^n|1}\right).$$

(b) Let $\mathcal{E}_a = |V_1|^2 + |V_2|^2$ and $\mathcal{E}_b = |V_3|^2 + |V_4|^2$. Explain why $\{\mathcal{E}_a,\ \mathcal{E}_b\}$ form a sufficient statistic for this problem and express the log likelihood ratio in terms of the sample values of $\{\mathcal{E}_a,\ \mathcal{E}_b\}$.

(c) Find the threshold for ML detection.

(d) Find the probability of error. Hint: Review Exercise 6.1. Note: we will later see that this corresponds to binary detection in Rayleigh fading.

8.4. Consider binary antipodal MAP detection for the real vector case. Modify the picture and argument in Figure 8.4 to verify the algebraic relation between the squared energy difference and the inner product in (8.21).

8.5. Derive (8.35), *i.e.*, that $\sum_{k,j} y_{k,j} b_{k,j} = \frac{1}{2} \int y(t) b(t)\, dt$. Explain the factor of 1/2.

8.6. In this problem, you will derive the inequalities

$$\left(1 - \frac{1}{x^2}\right) \frac{1}{x\sqrt{2\pi}}\, e^{-x^2/2} \leq Q(x) \leq \frac{1}{x\sqrt{2\pi}}\, e^{-x^2/2}; \qquad \text{for } x > 0, \tag{8.96}$$

where $Q(x) = (2\pi)^{-1/2} \int_x^\infty \exp(-z^2/2)\, dz$ is the "tail" of the Normal distribution. The purpose of this is to show that, when x is large, the right side of this inequality is a very tight upper bound on $Q(x)$.

(a) By using a simple change of variable, show that

$$Q(x) = \frac{1}{\sqrt{2\pi}} e^{-x^2/2} \int_0^\infty \exp\left(-y^2/2 - xy\right)\, dy.$$

(b) Show that

$$1 - y^2/2 \leq \exp(-y^2/2) \leq 1.$$

(c) Use parts (a) and (b) to establish (8.96)

8.7. (Other bounds on $Q(x)$) (a) Show that the following bound holds for any γ and η such that $0 \leq \gamma$ and $0 \leq \eta w$:

$$Q(\gamma + \eta) \leq Q(\gamma) \exp[-\eta\gamma - \gamma^2/2]$$

Hint: Start with $Q(\gamma + \eta) = \int_{\gamma+\eta} \exp[-x^2/2]\, dx$ and use the change of variable $y = x - \eta$.

(b) Use part (a) to show that for all $\eta \geq 0$,

$$Q(\eta) \leq \frac{1}{2} \exp[-\eta^2/2]$$

(c) Use (a) to show that for all $0 \leq \gamma \leq w$,

$$\frac{Q(w)}{\exp[-w^2/2]} \leq \frac{Q(\gamma)}{\exp[-\gamma^2/2]}$$

Note: (8.96) shows that $Q(w)$ goes to 0 with increasing w as a slowly varying coefficient time $\exp[-w^2/2]$. This demonstrates that the coefficient is decreasing for $w \geq 0$.

8.8. (Orthogonal signal sets) An *orthogonal signal set* is a set $\mathcal{A} = \{\boldsymbol{a}_m, 0 \le m \le M-1\}$ of M orthogonal vectors in $\mathbb{R}^M$ with equal energy E; *i.e.*, $\langle \boldsymbol{a}_m, \boldsymbol{a}_j \rangle = E\delta_{mj}$.

(a) Compute the normalized rate ρ of $\mathcal{A}$ in bits per two dimensions. Compute the average energy E_b per information bit.

(b) Compute the minimum squared distance $d^2_{\min}(\mathcal{A})$ between these signal points. Show that every signal has $M-1$ nearest neighbors.

(c) Let the noise variance be $N_0/2$ per dimension. Describe a ML detector on this set of M signals. Hint: Represent the signal set in an orthonormal expansion where each vector is collinear with one coordinate. Then visualize making binary decisions between each pair of possible signals.

8.9. (Orthogonal signal sets; continuation of Exercise 8.8) Consider a set $\mathcal{A} = \{\boldsymbol{a}_m, 0 \le m \le M-1\}$ of M orthogonal vectors in $\mathbb{R}^M$ with equal energy E.

(a) Use the union bound to show that $\Pr\{e\}$, using ML detection, is bounded by

$$\Pr\{e\} \le (M-1)Q(\sqrt{E/N_0}).$$

(b) Let $M \to \infty$ with $E_b = E/\log M$ held constant. Using the upper bound for $Q(x)$ in Exercise 8.7b, show that if $E_b/N_0 > 2\ln 2$ then $\lim_{M\to\infty} \Pr(e) = 0$. How close is this to the ultimate Shannon limit on E_b/N_0? What is the limit of the normalized rate ρ?

8.10. (Lower bound to $\Pr(e)$ for orthogonal signals) (a) Recall the exact expression for error probability for orthogonal signals in WGN from (8.47),

$$\Pr(e) = \int_{-\infty}^{\infty} f_{W_0|\boldsymbol{A}}(w_0|\boldsymbol{a}_0) \Pr\left(\bigcup_{m=1}^{M-1} (W_m \ge w_0 | \boldsymbol{A} = \boldsymbol{a}_0)\right) dw_0.$$

Explain why the events $W_m \ge w_0$ for $1 \le m \le M-1$ are iid conditional on $\boldsymbol{A} = \boldsymbol{a}_0$ and $W_0 = w_0$.

(b) Demonstrate the following two relations for any w_0,

$$\begin{aligned}\Pr\left(\bigcup_{m=1}^{M-1} (W_m \ge w_0 | \boldsymbol{A} = \boldsymbol{a}_0)\right) &= 1 - [1 - Q(w_0)]^{M-1} \\ &\ge (M-1)Q(w_0) - \frac{[(M-1)Q(w_0)]^2}{2}\end{aligned}$$

(c) Define γ_1 by $(M-1)Q(\gamma_1) = 1$. Demonstrate the following:

$$\Pr\left(\bigcup_{m=1}^{M-1} (W_m \ge w_0 | \boldsymbol{A} = \boldsymbol{a}_0)\right) \ge \begin{cases} \frac{(M-1)Q(w_0)}{2} & \text{for} \quad w_0 > \gamma_1 \\ \frac{1}{2} & \text{for} \quad w_0 \le \gamma_1 \end{cases}$$

(d) Show that

$$\Pr(e) \ge \frac{1}{2} Q(\alpha - \gamma_1)$$

(e) Show that $\lim_{M\to\infty} \gamma_1/\gamma = 1$ where $\gamma = \sqrt{2\ln M}$. Use this to compare the lower bound in part (d) to the upper bounds for cases 1 and 2 in Subsection 8.5.3. In particular show that $\Pr(e) \ge 1/4$ for $\gamma_1 > \alpha$ (the case where capacity is exceeded).

(f) Derive a tighter lower bound on $\Pr(e)$ than part (d) for the case where $\gamma_1 \leq \alpha$. Show that the ratio of the log of your lower bound and the log of the upper bound in Subsection 8.5.3 approaches 1 as $M \to \infty$. Note: this is much messier than the bounds above.

8.11. Section 8.3.4 discusses detection for binary complex vectors in WGN by viewing complex n-dimensional vectors as $2n$-dimensional real vectors. Here you will treat the vectors directly as n-dimensional complex vectors. Let $\boldsymbol{Z} = (Z_1, \ldots, Z_n)^\mathsf{T}$ be a vector of complex iid Gaussian rv's with iid real and imaginary parts, each $\mathcal{N}(0, N_0/2)$. The input $\boldsymbol{U}$ is binary antipodal, taking on values $\boldsymbol{a}$ or $-\boldsymbol{a}$, The observation $\boldsymbol{V}$ is $\boldsymbol{U} + \boldsymbol{Z}$,

(a) The probability density of $\boldsymbol{Z}$ is given by

$$f_{\boldsymbol{Z}}(\boldsymbol{z}) = \frac{1}{(\pi N_0)^n} \exp \sum_{j=1}^{n} \frac{-|z_j|^2}{N_0} = \frac{1}{(\pi N_0)^n} \exp \frac{-\|\boldsymbol{z}\|^2}{N_0}.$$

Explain what this probability density represents (*i.e.*, probability per unit what?).

(b) Give expressions for $f_{\boldsymbol{V}|U}(\boldsymbol{v}|\boldsymbol{a})$ and $f_{\boldsymbol{V}|U}(\boldsymbol{v}| - \boldsymbol{a})$.

(c) Show that the log likelihood ratio for the observation $\boldsymbol{v}$ is given by

$$\mathrm{LLR}(\boldsymbol{v}) = \frac{-\|\boldsymbol{v} - \boldsymbol{a}\|^2 + \|\boldsymbol{v} + \boldsymbol{a}\|^2}{N_0}.$$

(d) Explain why this implies that ML detection is minimum distance detection (defining the distance between two complex vectors as the norm of their difference).

(e) Show that $\mathrm{LLR}(\boldsymbol{v})$ can also be written as $\frac{4\Re(\langle \boldsymbol{v}, \boldsymbol{a} \rangle)}{N_0}$.

(f) The appearance of the real part, $\Re(\langle \boldsymbol{v}, \boldsymbol{a} \rangle)$, above is surprising. Point out why log likelihood ratios must be real. Also explain why replacing $\Re(\langle \boldsymbol{v}, \boldsymbol{a} \rangle)$ by $|\langle \boldsymbol{v}, \boldsymbol{a} \rangle|$ in the above expression would give a non-sensical result in the ML test.

(g) Does the set of points $\{\boldsymbol{v} : \mathrm{LLR}(\boldsymbol{v}) = 0\}$ form a complex vector space?

8.12. Let D be the function that maps vectors in $\mathcal{C}^n$ into vectors in $\mathcal{R}^{2n}$ by the mapping

$$\boldsymbol{a} = (a_1, a_2, \ldots, a_n) \to (\Re a_1, \Re a_2, \ldots, \Re a_n, \Im a_1, \Im a_2, \ldots, \Im a_n) = D(\boldsymbol{a})$$

(a) Explain why $\boldsymbol{a} \in \mathcal{C}^n$ and $i\boldsymbol{a}$ ($i = \sqrt{-1}$)are contained in the one dimensional complex subspace of $\mathcal{C}^n$ spanned by $\boldsymbol{a}$.

(b) Show that $D(\boldsymbol{a})$ and $D(i\boldsymbol{a})$ are orthogonal vectors in $\mathcal{R}^{2n}$.

(c) For $\boldsymbol{v}, \boldsymbol{a} \in \mathcal{C}^n$, the projection of $\boldsymbol{v}$ on $\boldsymbol{a}$ is given by $\boldsymbol{v}_{|\boldsymbol{a}} = \frac{\langle \boldsymbol{v}, \boldsymbol{a} \rangle}{\|\boldsymbol{a}\|} \frac{\boldsymbol{a}}{\|\boldsymbol{a}\|}$. Show that $D(\boldsymbol{v}_{|\boldsymbol{a}})$ is the projection of $D(\boldsymbol{v})$ onto the subspace of $\mathcal{R}^{2n}$ spanned by $D(\boldsymbol{a})$ and $D(i\boldsymbol{a})$.

(d) Show that $D(\frac{\Re[\langle \boldsymbol{v}, \boldsymbol{a} \rangle]}{\|\boldsymbol{a}\|} \frac{\boldsymbol{a}}{\|\boldsymbol{a}\|})$ is the further projection of $D(\boldsymbol{v})$ onto $D(\boldsymbol{a})$.

8.13. Consider 4-QAM with the 4 signal points $u = \pm a \pm ia$. Assume Gaussian noise with spectral density $N_0/2$ per dimension.

(a) Sketch the signal set and the ML decision regions for the received complex sample value y. Find the exact probability of error (in terms of the Q function) for this signal set using ML detection.

(b) Consider 4-QAM as two 2-PAM systems in parallel. That is, a ML decision is made on $\Re(u)$ from $\Re(v)$ and a decision is made on $\Im(u)$ from $\Im(v)$. Find the error probability

(in terms of the Q function) for the ML decision on $\Re(u)$ and similarly for the decision on $\Im(u)$.

(c) Explain the difference between what has been called an error in part (a) and what has been called an error in part (b).

(d) Derive the QAM error probability directly from the PAM error probability.

8.14. Consider two 4-QAM systems with the same 4-QAM constellation

$$s_0 = 1 + i, \quad s_1 = -1 + i, \quad s_2 = -1 - i, \quad s_3 = 1 - i.$$

For each system, a pair of bits is mapped into a signal, but the two mappings are different:

$$\begin{array}{lllll} \text{Mapping 1:} & 00 \to s_0, & 01 \to s_1, & 10 \to s_2, & 11 \to s_3 \\ \text{Mapping 2:} & 00 \to s_0, & 01 \to s_1, & 11 \to s_2, & 10 \to s_3 \end{array}$$

The bits are independent and 0's and 1's are equiprobable, so the constellation points are equally likely in both systems. Suppose the signals are decoded by the minimum distance decoding rule, and the signal is then mapped back into the two binary digits. Find the error probability (in terms of the Q function) for each bit in each of the two systems.

8.15. Re-state Theorem 8.4.1 for the case of MAP detection. Assume that the inputs $U_1, \ldots, U_n$ are independent and each have the a priori distribution $p_0, \ldots, p_{M-1}$. Hint: start with (8.41) and (8.42) which are still valid here.

8.16. The following problem relates to a digital modulation scheme often referred to as minimum shift keying (MSK). Let

$$s_0(t) = \begin{cases} \sqrt{\frac{2E}{T}} \cos(2\pi f_0 t) & \text{if } 0 \le t \le T, \\ 0 & \text{otherwise.} \end{cases}$$

$$s_1(t) = \begin{cases} \sqrt{\frac{2E}{T}} \cos(2\pi f_1 t) & \text{if } 0 \le t \le T, \\ 0 & \text{otherwise.} \end{cases}$$

a) Compute the energy of the signals $s_0(t), s_1(t)$. You may assume that $f_0 T \gg 1$ and $f_1 T \gg 1$.

(b) Find conditions on the frequencies f_0, f_1 and the duration T to ensure both that the signals $s_0(t)$ and $s_1(t)$ are orthogonal and that $s_0(0) = s_0(T) = s_1(0) = s_1(T)$. Why do you think a system with these parameters is called minimum shift keying?

(c) Assume that the parameters are chosen as in (b). Suppose that, under U=0, the signal $s_0(t)$ is transmitted, and under U=1, the signal $s_1(t)$ is transmitted. Assume that the hypotheses are equally likely. Let the observed signal be equal to the sum of the transmitted signal and a White Gaussian process with spectral density $N_0/2$. Find the optimal detector to minimize the probability of error. Draw a block diagram of a possible implementation.

(d) Compute the probability of error of the detector you have found in part (c).

8.17. Consider binary communication to a receiver containing k_0 antennas. The transmitted signal is $\pm a$. Each antenna has its own demodulator, and the received signal after demodulation at antenna k, $1 \le k \le k_0$, is given by

$$V_k = Ug_k + Z_k,$$

where U is $+a$ for U=0 and $-a$ for U=1. Also g_k is the gain of antenna k and $Z_k \sim \mathcal{N}(0, \sigma^2)$ is the noise at antenna k; everything is real and $U, Z_1, Z_2, \ldots, Z_{k_0}$ are independent. In vector notation, $\boldsymbol{V} = U\boldsymbol{g} + \boldsymbol{Z}$ where $\boldsymbol{V} = (v_1, \ldots, v_{k_0})^{\mathsf{T}}$ etc.

(a) Suppose that the signal at each receiving antenna k is weighted by an arbitrary real number q_k and the signals are combined as $Y = \sum_k V_k q_k = \langle \boldsymbol{V}, \boldsymbol{q} \rangle$. What is the maximum likelihood (ML) detector for U given the observation Y?

(b) What is the probability of error $\Pr(e)$ for this detector?

(c) Let $\beta = \frac{\langle \boldsymbol{g}, \boldsymbol{q} \rangle}{\|\boldsymbol{g}\|\|\boldsymbol{q}\|}$. Express $\Pr(e)$ in a form where $\boldsymbol{q}$ does not appear except for its effect on β.

(d) Give an intuitive explanation why changing $\boldsymbol{q}$ to $c\boldsymbol{q}$ for some nonzero scalar c does not change $\Pr(e)$.

(e) Minimize $\Pr(e)$ over all choices of $\boldsymbol{q}$ (or β) above.

(f) Is it possible to reduce $\Pr(e)$ further by doing ML detection on $V_1, \ldots, V_{k_0}$ rather than restricting ourselves to a linear combination of those variables?

(g) Redo part (b) under the assumption that the noise variables have different variances, *i.e.*, $Z_k \sim \mathcal{N}(0, \sigma_k^2)$. As before, $U, Z_1, \ldots, Z_{k_0}$ are independent.

(h) Minimize $\Pr(e)$ in part (g) over all choices of $\boldsymbol{q}$.

8.18. (a) The Hadamard matrix H_1 has the rows 00 and 01. Viewed as binary codewords this is rather foolish since the first binary digit is always 0 and thus carries no information at all. Map the symbols 0 and 1 into the signals a and $-a$ respectively, $a > 0$ and plot these two signals on a two dimensional plane. Explain the purpose of the first bit in terms of generating orthogonal signals.

(b) Assume that the mod-2 sum of each pair of rows of H_b is another row of H_b for any given integer $b \ge 1$. Use this to prove the same result for H_{b+1}. Hint: Look separately at the mod-2 sum of two rows in the first half of the rows, two rows in the second half, and two rows in different halves.

8.19. (RM codes) (a) Verify the following combinatorial identity for $0 < r < m$:

$$\sum_{j=0}^{r} \binom{m}{j} = \sum_{j=0}^{r-1} \binom{m-1}{j} + \sum_{j=0}^{r} \binom{m-1}{j}.$$

Hint: Note that the first term above is the number of binary m tuples with r or fewer 1's. Consider separately the number of these that end in 1 and end in 0.

(b) Use induction on m to show that $k(r, m) = \sum_{j=0}^{r} \binom{m}{j}$. Be careful how you handle $r = 0$ and $r = m$.

8.20. (RM codes) This exercise first shows that $\mathrm{RM}(r, m) \subset \mathrm{RM}(r{+}1, m)$ for $0 \le r < m$. It then shows that $d_{\min}(r, m) = 2^{m-r}$.

(a) Show that if $\mathrm{RM}(r{-}1, m{-}1) \subset \mathrm{RM}(r, m{-}1)$ for all r, $0 < r < m$, then

$$\mathrm{RM}(r{-}1, m) \subset \mathrm{RM}(r, m) \qquad \text{for all } r,\ 0 < r \leq m$$

Note: Be careful about $r = 1$ and $r = m$.

(b) Let $\boldsymbol{x} = (\boldsymbol{u}, \boldsymbol{u} \oplus \boldsymbol{v})$ where $\boldsymbol{u} \in \mathrm{RM}(r, m{-}1)$ and $\boldsymbol{v} \in \mathrm{RM}(r{-}1, m{-}1)$. Assume that $d_{\min}(r, m{-}1) \leq 2^{m-1-r}$ and $d_{\min}(r{-}1, m{-}1) \leq 2^{m-r}$. Show that if $\boldsymbol{x}$ is nonzero, it has at least 2^{m-r} 1's. Hint 1: For a linear code, $d_{\min}$ is equal to the weight (number of ones) in the minimum-weight nonzero codeword. Hint 2: First consider the case $\boldsymbol{v} = 0$, then the case $\boldsymbol{u} = 0$. Finally use part (a) in considering the case $\boldsymbol{u} \neq 0, \boldsymbol{v} \neq 0$ under the subcases $\boldsymbol{u} = \boldsymbol{v}$ and $\boldsymbol{u} \neq \boldsymbol{v}$.

(c) Use induction on m to show that $d_{\min} = 2^{m-r}$ for $0 \leq r \leq m$.