ECE8771
Information Theory & Coding for Digital Communications

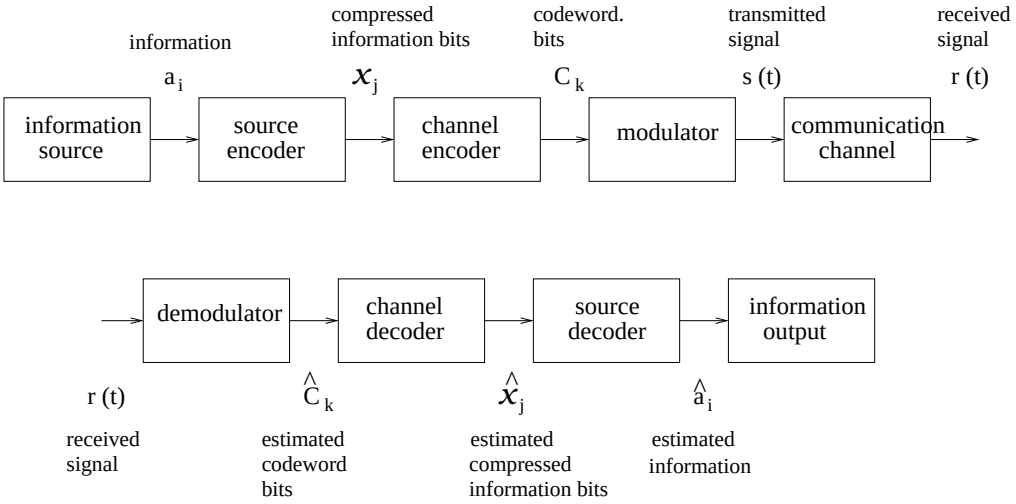
Villanova University
ECE Department

Prof. Kevin M. Buckley

Lecture Set 1

Review of Digital Communications, Introduction to Information Theory

---



Contents

List of Figures

1 Course Introduction

This Section of the Notes corresponds to Chapter 1 of the Text.

For over 60 years digital communication has had a substantial and a growing influence on society. With the recent worldwide growth of cellular and satellite telephone, and with the internet and multimedia applications, digital communication now has a daily impact on our lives and plays a central role in the global economy. Digital communication has become both a driving force and a principal product of a global society.

Digital communication is a broad, practical, highly technical, deeply theoretical, dynamically changing engineering discipline. These characteristics make digital communication a very challenging and interesting topic of study. Command of this topic is necessarily a long term challenge, and any course in digital communication must provide some tradeoff between overview and more in-depth treatment of selective topics. That said, the aim of this Course is to provide:

- an introduction to information theory, directed towards understanding the performance of communications systems;
- an overview of signal compression bounds and techniques;
- an overview of channel capacity, directed towards motivating the study of channel coding; and
- a somewhat in-depth study of channel coding, including an emphasis on recent practical developments.

Information theory is the principal foundation of modern digital communication. It provides a theoretical basis for lossless source coding, lossy source coding and channel coding. Specifically, it provides theoretical performance limitations and, conversely, achievable performance capabilities. For channel coding, given a specific channel, information theory tells us what the maximum channel capacity is for transmission of information without error. A very important aspect of this channel capacity bound is that the various forms of its proof are typically not constructive. That is, the proofs do not suggest how the bound can be practically achieved. As a consequence, since the inception of information theory over 60 years ago with Shannon's ground breaking work [1], there has been a large volume of practical and theoretical research aimed at achieving capacity bounds for various types of information sources and communication channels.

1.1 Overview of Shannon's Contributions to Information Theory

Claude Shannon, with his two part 1948 paper in the Bell System Technical Journal [1], is the recognized father of information theory and coding. It should be noted that the fundamental impact of the information theory described in this paper goes well beyond communications. It has significantly influenced probability and statistics, computer science, mathematics, physics, economics, biology and chemistry. For a comprehensive overview of the impact of Shannon's work, refer to Verdu's survey paper [2]. It should also be noted that subsequent, substantial contributions have been made by others.

As noted above, through the establishment of performance bounds, information theory has contributed to communications in the areas of source coding and channel coding. *Source coding*, which address the issue of information compression prior to transmission so as to reduce communication system requirements, is the topic of Section 5 of this Course. It is based on the information theory topics of source entropy and mutual information established and employed by Shannon, which are overviewed in Section 4 of this Course. The key source coding results of Shannon are captured in his source coding theorems related to: lossless source coding for discrete memoryless channels with fixed and variable length codes for sources without and with memory; and lossy source coding, based on rate distortion measures, for again a variety of code and source types. These results have motivated a vast amount of results on the topics of bounds for specific source types, and source coding algorithm development and evaluation.

Channel coding addresses the issue of incorporating redundancy into data prior to transmission over a channel so as to control transmission errors, while approaching the channel capacity for transmitting information without error. Shannon's channel capacity theorem establishes that for a given channel there is an upper bound, called the channel capacity, on the rate of error free information transmission. This bound is a function of channel characteristics such and bandwidth and SNR. This theorem has motivated the development and analysis of large number of approaches to channel coding. Channel capacity is the topic of Section 6 of this Course, and channel coding is covered in Sections 7, 8, 9 and 10.

1.2 The Digital Communication System

Figure 1 is a block diagram of a typical digital communication system. This figure is followed by a description of each block, and by accompanying comments on their relationship to this Course.

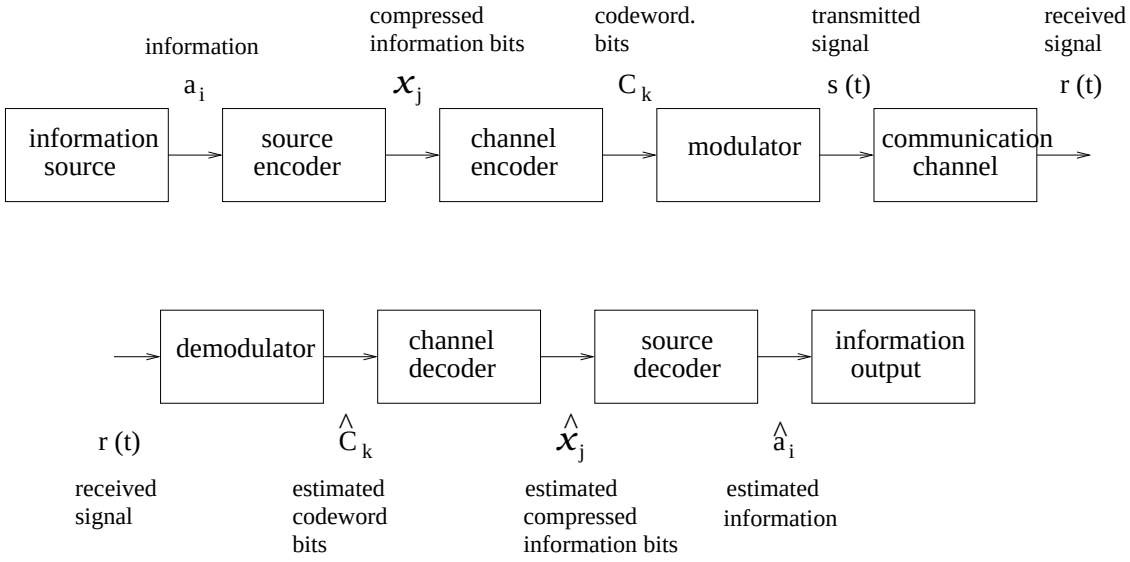


Figure 1: Digital Communication system block diagram.

The source encoder transforms signals to be transmitted into information bits, x_j, while implementing data compression for efficient representation for transmission. Source coding techniques include: fixed length codes (*lossless*); variable length Huffman codes (*lossless*); Lempel Ziv coding (*lossless*); sampling & quantization (*lossy*); adaptive differential pulse code modulation (ADPCM) (*lossy*); and transform coding (*lossy*). We will briefly overview source coding in Section 5 this Course.

The channel encoder introduces redundancy into the information bits to form the *codewords or code sequences*, C_k, so as to accommodate receiver error management. Channel coding approaches included: block coding; convolutional coding, turbo coding, space-time coding and coded modulation. Channel coding is the major topic of this Course, covering Sections 7-10. We will consider a variety of channel encoding/decoding methods.

The digital modulator transforms codewords or codeword sequences into M-ary waveforms (symbols) which can be transmitted over a communication channel. Digital modulation schemes include: Pulse Amplitude Modulation (PAM); Frequency Shift Keying (FSK); M-ary Quadrature Amplitude Modulation (M-QAM);and Binary Phase Shift Keying (BPSK) & Quadrature Phase Shift Keying (QPSK).

To evaluate the performance of channel coders/decoders operating in a digital communication system, and to understand coded modulation methods, it is necessary to understand digital modulation schemes that are used in communication systems. In Section 3 of this Course we will review several modulation schemes which we will subsequently use to illustrate channel coder/decoder performance characteristics. We will summarize performance analyzes of these modulation schemes as a precursor to the analyze of channel code methods.

The communication channel is at the heart of the communication problem. Additive channel noise corrupts the transmitted digital communication signal, causing unavoidable symbol decoding errors at the receiver. The channel can also distort the transmitted signal, for example as characterized by the channel impulse response. We further discuss these forms of signal corruption in Subsection 1.3.1 below. Additionally, interfering signals are often superimposed on the transmitted signal along with the noise. In this Course we are primarily interested in the control of errors caused by additive noise. However, to introduce the channel distortion issue and to illustrate how it is addressed in conjunction with channel coding, we overview selected approaches to joint channel equalization and decoding – namely turbo equalization and space-time coding.

The digital demodulator is the signal processor that transforms the distorted, noisy received symbol waveforms into discrete time data from which binary or M-ary symbols are estimated. Demodulator components include: correlators or matched filters (which include the receiver front end); nearest neighbor threshold detectors; channel equalizers; symbol detectors and sequence estimators. Design of the digital demodulator is not a principal topic of this Course. However, to understand channel decoding performance, we do need to know how to optimally demodulate the modulation schemes we use to illustrate channel coding performance. So this topic will be overviewed in Section 3.

The channel decoder works in conjunction with the channel encoder to manage digital communication errors. It is thus a principal topic of this Course. In Sections 7-10 we will describe several basic approaches to channel code decoding (e.g. soft and hard decoding), and evaluate performance for various coding and modulation schemes.

The source decoder is the receiver component that reverses, as much as possible or reasonable, the source coder. Source decoding is not explicitly covered in this Course, though decoding procedures are directly dictated by the corresponding source encoder.

1.2.1 The Communication Channel & Digital Demodulator

As noted above, the channel corrupts the transmitted symbols. Specifically, the channel distorts the transmitted symbols and superimposes noise and interference. The symbol *distortion* can be either linear or nonlinear, though linear distortion is much more common and easier to deal with. Distortion often results in *intersymbol interference (ISI)*, i.e. adjacent symbols overlapping in time at the receiver. In applications such as cellular phones, *fading* of the transmitted signal is also a major concern. The *noise* that is superimposed onto the transmitted symbols is typically Gaussian receiver noise. In practice, this additive noise makes it impossible to perfectly determine which symbols are sent. In some applications *interference* is also superimposed onto the transmitted symbols. For example, this can be in the form of: crosstalk from bundled wires; or interference from symbols on adjacent tracks of a magnetic disk; or competing users in a multi-user electromagnetic channel; or electromagnetic radiation from man made or natural sources; or jamming signals. The digital demodulator estimates the transmitted symbols. As much as possible or practical, it compensates for the channel, through optimum detection or sequence (of symbols) estimation, or through channel equalization and interference suppression (e.g. optimum and adaptive filtering). The function of channel coding/decoding is to correct or otherwise manage the inevitable demodulator symbol errors.

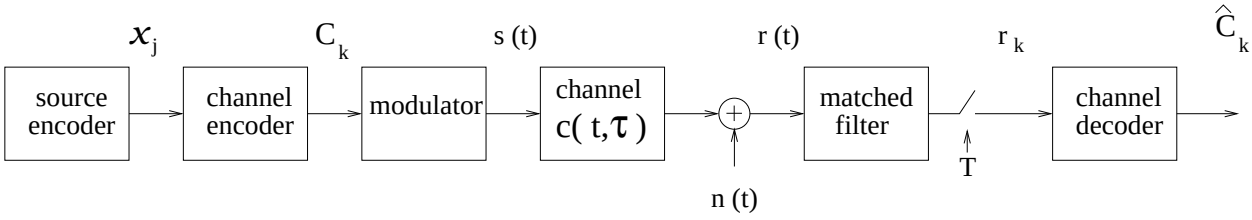


Figure 2: Digital communication channel with channel distortion.

Figure 2 is a block diagram model of the transmitter, channel and receiver front end of a typical digital communications system. The bit sequence x_j is the information to be communicated. It is processed by the channel encoder to from the codeword sequence $\mathbf{C}_k$. In the modulator $\mathbf{C}_k$ is mapped to a sequence of symbols, I_k, which modulates a carrier sinusoid to form the signal $s(t)$, e.g.

$$s(t) \;=\; \sum_k \; I_k \; g(t - kT) \quad , \tag{1}$$

which is transmitted across the channel. Here $g(t)$ is the analog symbol pulse shape and T is the symbol duration (i.e. the inverse of the symbol rate). The channel shown here is linear and time varying, with impulse response $c(t,\tau)$ at time t (i.e. τ is the channel memory variable). Noise and interference, denoted $n(t)$, is superimposed onto the channel output to form the received signal

$$r(t) \;=\; \int \; s(\tau)\; c(t, t-\tau)\; d\tau \;+\; n(t) \quad . \tag{2}$$

Typically, the received signal is matched filtered and sampled to form the discrete-time sequence r_k which is a distorted, noisy version of the desired symbol sequence I_k. The channel decoder processes r_k to form an estimate $\hat{\mathbf{C}}_k$ of $\mathbf{C}_k$.

Figure 3 is an equivalent discrete-time model, from I_k to r_k, of the digital communication system depicted in Figure 2. For the most part, in this Course we will consider a simple special case of this model, for which the noise is Additive White Gaussian Noise (AWGN) and the channel is distortionless. However, in Section 8 on turbo equalization and Section 9 on space-time coding, we will characterize and address channel distortion.

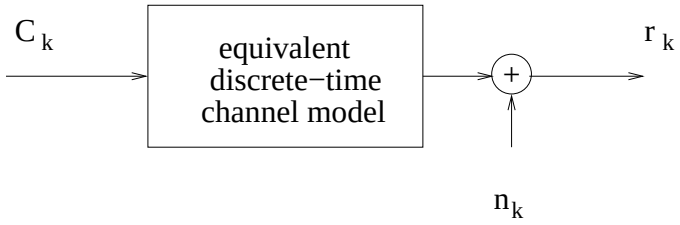


Figure 3: Equivalent discrete time model.

2 Background in Probability

This Section of the Notes corresponds to Sections 2.1-6,9 of the Text. Pur objective here is a selected review of those probability and random process concepts which provide the probabilistic background necessary for discussing coding topics at the level targeted for this Course. Note that the digital communication receiver problem is to process a received signal

$$r(t) = s(t) * c(t) + n(t) \quad , \tag{1}$$

where $s(t)$ is the transmitted signal which is constructed by the transmitter as a carrier signal modulated by a sequence of symbols that represent the digital information to be transmitted. $c(t)$ is the impulse response of the channel, and $n(t)$ is the additive noise and interference. Throughout most of this Course we will assume that the channel is distortionless (i.e. $c(t) = \delta(t)$ and that there are no interference). The noise $n(t)$ is assumed random, as is the symbol sequence that modulates the carrier to form the transmitted signal $s(t)$. Thus, the received signal $r(t)$ is random, as are data samples derived from it.

2.1 Probability

Given a random event B and mutually exclusive, exhaustive (i.e. comprehensive) random events $\{A_i; i = 1, 2, ..., n\}$, with individual probabilities $P(B)$ and $\{P(A_i); i = 1, 2, ..., n\}$, and joint probabilities $\{P(A_i, B); i = 1, 2, ..., n\}$, we have that

$$P(A_i, A_j) = 0 \quad i \neq j \tag{2}$$

because the A_i are mutually exclusive. We also have that

$$\sum_{i=1}^{n} P(A_i) = 1 \tag{3}$$

because the A_i are mutually exclusive and exhaustive. Also,

$$P(A_i/B) = \frac{P(A_i, B)}{P(B)} \tag{4}$$

is the conditional probability equation. $P(A_i/B)$ reads – the probability of event A_i given event B (has occurred). The relation

$$P(A_i/B) = \frac{P(B/A_i)\ P(A_i)}{P(B)} \tag{5}$$

is Bayes' theorem relating the conditional probabilities $P(A_i/B)$ and $P(B/A_i)$. The equation

$$P(B) = \sum_{i=1}^{n} P(B/A_i)\ P(A_i) \tag{6}$$

is the total probability (of B in terms of its conditional probabilities $P(B/A_i)$). Finally,

$$P(A_i/B) = \frac{P(B/A_i)\ P(A_i)}{\sum_{j=1}^{n} P(B/A_j)\ P(A_j)} \tag{7}$$

is Bayes' theorem using the total probability for $P(B)$.

Within the context of this Course, we are often interested in the above relationships, where $\{A_i; i = 1, 2, \cdots, n\}$ is the set of symbols used to representing binary data, and the event B is related to received data. Since one and only one symbol is sent at a time, the symbol set is mutually exclusive and exhaustive. These notions can be extended to a sequence of transmitted symbols.

Union Bound

As we will see in Section 3.3 of these Notes and in Example 2.1 below (and Chapter 4 of the Course Text), the union bound on probability is useful in the performance analysis of digital modulation schemes.

Let $E_i; i = 1, 2, \cdots N$ be events which are not necessarily mutually exclusive or exhaustive. We are often interested in the probability of the union of these events:

$$P\left(\bigcup_{i=1}^{N} E_i \right) \quad . \tag{8}$$

If the E_i were mutually exclusive, then

$$P\left(\bigcup_{i=1}^{N} E_i \right) \; = \; \sum_{i=1}^{N} P(E_i) \quad . \tag{9}$$

This is illustrated in Figure 4(a) for the two event case. However, in general,

$$P\left(\bigcup_{i=1}^{N} E_i \right) \; \leq \; \sum_{i=1}^{N} P(E_i) \quad , \tag{10}$$

since if the events share some outcomes (elements), the probabilities are counted more than once with the summation over events. This inequality is called the *union bound*. Figure 4(b) illustrates the union bound for the two event case.

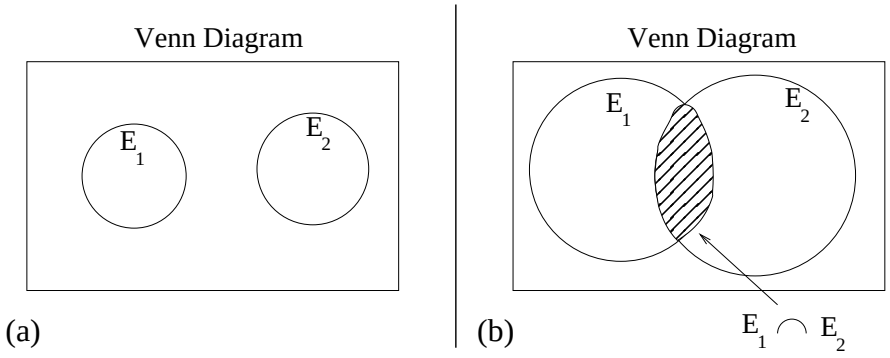


Figure 4: An illustration of the union bound.

Example 2.1: Let $I_i;\ i = 1, 2, \cdots, M$ represent the M possible symbols of a digital modulation scheme. Say symbol I_1 was transmitted, and let $I_i/I_1;\ i = 2, 3, \cdots, M$ each denote the event that a symbol I_i is decided over event I_1 at the receiver. These $M-1$ events are typically not mutual exclusive. $P(I_i/I_1)$, the probability of event I_i/I_1, is usually easy to identify. Often, of interest is $P(e/I_1)$, the probability of error given that I_1 was transmitted. This is typically difficult to identify. However, the union bound

$$P(e/I_1) \ \leq \ \sum_{i=2}^{M} \ P(I_i/I_1) \ \ , \tag{11}$$

is easy enough to identify, and often useful as a guideline for performance.

2.2 Random Variables

Let X_1 be a random variable (RV) which takes on values x_1. The Probability Density Function (PDF) [1] $p(x_1)$ and probability distribution function $F(x_1)$ are related as follows:

$$F(x_1) = P(-\infty < \mathsf{x}_1 \leq x_1) = \int_{-\infty}^{x_1} p(u_1)\, du_1 \tag{12}$$

$$p(x_1) = \frac{\partial}{\partial x_1} F(x_1) \ \ . \tag{13}$$

Given n RV's, $\{X_i;\ i = 1, 2, ..., n\}$ (in vector form $\underline{X} = [X_1, X_2, \cdots, X_n]^T$), their joint PDF is denoted

$$p(x_1, x_2, ..., x_n) = p(\underline{x}) \ ; \qquad \underline{x} = [x_1, x_2, ..., x_n]^T \tag{14}$$

where the superscript $''T''$ denotes the matrix or vector transpose operation, and the "underbar" indicates a vector or matrix (lower case represents a vector, while uppercase represents a matrix). Random variables can be either continuous (e.g. a sample of a received signal) or discrete (e.g. a communication symbol). So, joint PDF's can be either smooth (for continuous RV's) or impulsive (for discrete RV's) or combined smooth/impulsive (for a mix of continuous/discrete RV's). We determine joint probabilities of RV's by integrating their joint PDF, i.e.

$$P(\underline{a} < \underline{X} \leq \underline{b}) = \int_{\underline{a}}^{\underline{b}} p(\underline{x})\ d\underline{x} \quad . \tag{15}$$

This is illustrated below in Figure 5 for both continuous and discrete-valued RVs.

It follows from Bayes' theorem that a conditional PDF of X_1 given a value x_2 of RV X_2 is

$$p(x_1/x_2) = \frac{p(x_1, x_2)}{p(x_2)} = \frac{p(x_2/x_1)\ p(x_1)}{p(x_2)} \tag{16}$$

where it is assumed that the value x_2 is possible (i.e. $p(x_2) \neq 0$). This last equation is particularly useful for symbol detection and sequence estimation.

[1] We will use a lower case p to denote a PDF of a continuous-valued RV, and an upper case P to represent the PDF of a discrete-valued RV.

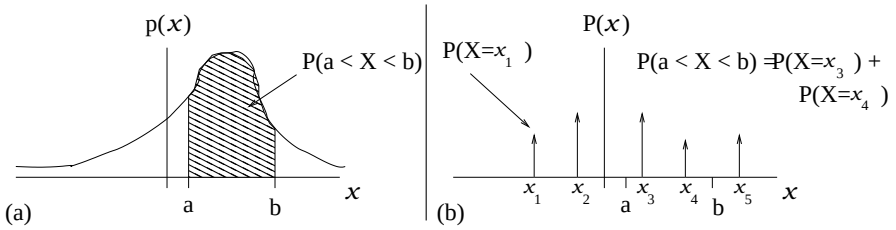


Figure 5: A PDF of a single random variable X, and the probability $P(a < X < b)$: (a) continuous-valued; (b) discrete-valued.

Note that if a RV X_1 is discrete (say a digital symbol) and X_2 is continuous (e.g. a sample of a received signal), we write

$$P(x_1/x_2) = \frac{p(x_2/x_1)\ P(x_1)}{p(x_2)} \quad . \tag{17}$$

Again, this assumes that for the value of x_2 considered, the PDF $p(x_2)$ is nonzero (i.e. that value x_2 can occur).

2.3 Statistical Independence and the Markov Property

Note that in general a joint PDF of $\underline{X}$ does not factor into a product of the individual PDF's, i.e. in general

$$p(\underline{x}) \neq \prod_{i=1}^{n} p(x_i) \quad . \tag{18}$$

However, if it does for a particular $\underline{X}$, then we say that the RV's are *statistically independent*, and the joint PDF will be a lot easier to work with (and the random vector $\underline{X}$ will be easier to optimally process).

Let $X_j;\ j = 1, 2, \cdots, n$ be a random sequence. Let $\underline{X} = [x_1, x_2, ..., x_n]^T$. We say this this sequence is a Markov process if joint PDF of $\underline{X}$ has the following factorization:

$$p(\underline{x}) \ = \ p(x_1) \cdot p(x_2/x_1) \cdot p(x_3/x_2) \cdots p(x_n/x_{n-1}) \quad . \tag{19}$$

You can imagine that Markov random process joint PDF's are easier to work with than general joint PDF's, but not quite as easy to work with as statistically independent random variable PDF's.

2.4 Gaussian Random Variables

Consider a random variable X with PDF $p(x)$. Its mean m_x and variance σ_x^2 are, respectively

$$m_x \ = \ E\{X\} \ = \ \int_{-\infty}^{\infty} x\, p(x)\, dx\ ; \quad \sigma_x^2 \ = \ E\{|X - m_x|^2\} \ = \ \int_{-\infty}^{\infty} |x - m_x|^2\, p(x)\, dx\ . \tag{20}$$

A real-valued (as opposed to a complex-valued) Gaussian RV X has a PDF of the following form:

$$p(x) = \frac{1}{\sqrt{2\pi\sigma_x^2}}\ e^{-(x-m_x)^2/2\sigma_x^2} \quad . \tag{21}$$

A complex-valued random variable $X = X_r + jX_i$ is Gaussian if its PDF is

$$p(x) = \frac{1}{2\pi\sigma_x^2}\ e^{-|x-m_x|^2/\sigma_x^2} \quad . \tag{22}$$

A complex-valued X is really a 2-dimensional variable (i.e. we can consider $X = [X_r, X_i]^T$) and its PDF is actually the 2-dimensional joint PDF of $[X_r,\ X_i]$.

Let $\underline{X}$ be a real-valued Gaussian random vector. Then, its joint PDF is of the form

$$p(\underline{x}) = \frac{1}{(2\pi)^{n/2}(\det(\underline{C}_x))^{1/2}}\ e^{-\frac{1}{2}(\underline{x}-\underline{m}_x)^T\underline{C}_x^{-1}(\underline{x}-\underline{m}_x)} \tag{23}$$

where " det " denotes the determinant, $\underline{m}_x = E\{\underline{X}\}$ is the mean vector, and $\underline{C}_x^{-1}$ is the matrix inverse of the covariance matrix $\underline{C}_x = (\underline{X} - \underline{m}_x)\ E\{(\underline{X} - \underline{m}_x)^H\}$. If all the random variables in X are mutually uncorrelated, then the joint PDF reduces to

$$p(\underline{x}) = \frac{1}{\prod_{i=1}^n (2\pi\sigma_{x_i}^2)^{1/2}}\ e^{-\frac{1}{2}\sum_{i=1}^n (x_i-m_{x_i})^2/\sigma_{x_i}^2} = \prod_{i=1}^n p(x_i) \quad , \tag{24}$$

i.e. mutually uncorrelated Gaussian RV's are statistically independent. The fact that uncorrelated Gaussian RV's are also statistically independent is a significant advantage.

If $\underline{X}$ is complex-valued Gaussian, then its joint PDF is

$$p(\underline{x}) = \frac{1}{(2\pi)^n \det(\underline{C}_x)}\ e^{-(\underline{x}-\underline{m}_x)^H\underline{C}_x^{-1}(\underline{x}-\underline{m}_x)} \quad . \tag{25}$$

Uncorrelated complex-valued Gaussian RV's are also statistically independent.

Gaussian RV's, both real and complex valued, are often encountered in communication systems. For example, when the additive noise is receiver noise (thermal noise from the front-end amplifier), a sample of this noise is real-valued Gaussian. For a band pass communication system, the in-phase/quadrature demodulator that is often applied to the output of the receiver front-end amplifier, generates a "complex-valued" signal whose samples are complex-valued Gaussian if the front-in amplifier output is simply receiver noise.

Example 2.2: Problem: Determine $P(0 \le X \le 2)$ for a real-valued Gaussian RV X with mean $m_x = 1$ and variance $\sigma_x^2 = 2$.

Solution:

$$P(0 \le X \le 2) \ = \ \int_a^b \frac{1}{\sqrt{2\pi\sigma_x^2}}\ e^{(x-m_x)^2/2\sigma_x^2}\ dx \ = \ Q\left(\frac{a-m_x}{\sigma_x}\right) \ - \ Q\left(\frac{b-m_x}{\sigma_x}\right) \tag{26}$$

where $a = 0$, $b = 2$ and $Q(x)$ is the Gaussian tail probability function

$$Q(x) \ = \ \frac{1}{\sqrt{2\pi}} \int_x^\infty e^{\lambda^2/2}\ d\lambda \quad . \tag{27}$$

Using a Q function table, we get

$$P(0 \le X \le 2) \ = \ Q\left(\frac{0-1}{\sqrt{2}}\right) \ - \ Q\left(\frac{2-1}{\sqrt{2}}\right) \ \approx \ 0.4988 \quad . \tag{28}$$

2.5 Bounds on Tail Probabilities

See pp. 56-63 of the Course Text for a complementary discussion on bounds on PDF tail probabilities. As with the union bound introduced earlier, the bounds described here are useful for performance analysis of coding schemes and decoding algorithms. Consider a random variable X with mean m_x, variance σ_x^2, and PDF illustrated below in Figure 6(a). Consider a positive constant δ. Say we are interested in the probability

$$P(X \geq m_x + \delta) \;=\; P(X - m_x \geq \delta) \quad , \tag{29}$$

i.e. the probability that the random variable will be greater than or equal to δ above its mean. If we know the PDF we can find this probability, exactly. Alternatively, we may look for a useful bound on this probability.

Chebyshev Inequality: For the two-sided tail probability illustrated in Figure 6(b),

$$P(|X - m_x| \geq \delta) \;\leq\; \frac{\sigma_x^2}{\delta^2} \quad . \tag{30}$$

Note that for symmetric PDF's, we have $P(X - m_x \geq \delta) \;\leq\; \frac{\sigma_x^2}{2\delta^2}$.
Proof: Consider zero mean $Y = X - m_x$. The Chebyshev inequality in terms of Y is

$$P(|Y| \geq \delta) \;\leq\; \frac{\sigma_x^2}{\delta^2} \quad . \tag{31}$$

As illustrated in Figure 7(a), let

$$g(Y) \;=\; \begin{cases} 1 & |Y| \geq \delta \\ 0 & otherwise \end{cases} \quad . \tag{32}$$

Let $p_Y(y)$ denote the PDF of Y. Then,

$$E\{g(Y)\} \;=\; \int_{-\infty}^{\infty} g(y)\, p_Y(y)\, dy \;=\; \int_{|y| \geq \delta} 1\, p_Y(y)\, dy \;=\; P(|Y| \geq \delta) \quad . \tag{33}$$

Since $g(Y) \leq \left(\frac{Y}{\delta}\right)^2$ for all Y, we have

$$E\{g(Y)\} \;\leq\; E\{\left(\frac{Y}{\delta}\right)^2\} \;=\; \frac{E\{Y^2\}}{\delta^2} \;=\; \frac{\sigma_x^2}{\delta^2} \quad . \tag{34}$$

So,

$$P(|X - m_x| \geq \delta) \;\leq\; \frac{\sigma_x^2}{\delta^2} \quad . \tag{35}$$

This derivation of the Chebyshev inequality (bound) leads to the following tighter bound.

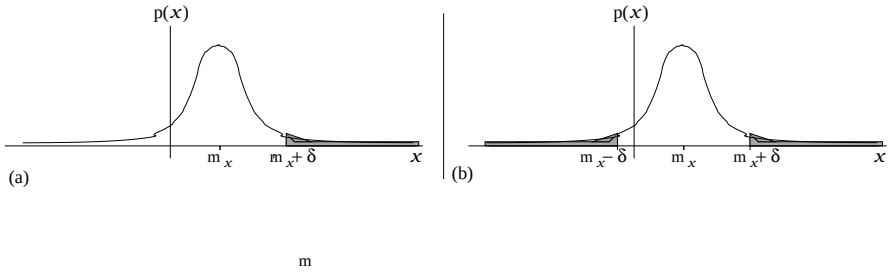


Figure 6: (a) A tail probability; (b) a two-sided tail probability for the Chebyshev inequality.

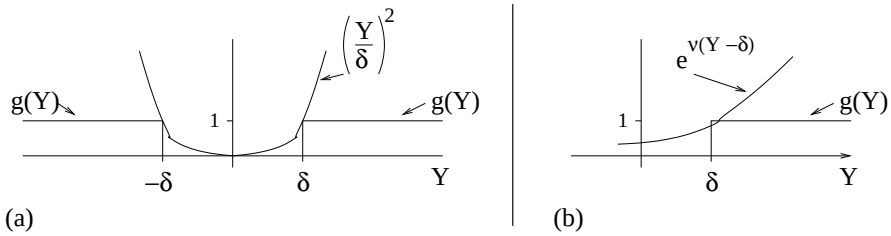


Figure 7: $g(Y)$ function for (a) the Chebyshev bound, (b) the Chernov bound.

Chernov Bound: Considering the proof above of the Chebyshev inequality, but instead of using $g(Y) \leq \left(\frac{Y}{\delta}\right)^2$ with

$$g(Y) \;=\; \begin{cases} 1 & |Y| \geq \delta \\ 0 & otherwise \end{cases} \quad , \tag{36}$$

let $g(Y) \leq e^{\nu(Y-\delta)}$ where ν is a constant to be determined and

$$g(Y) \;=\; \begin{cases} 1 & Y \geq \delta \\ 0 & otherwise \end{cases} \quad , \tag{37}$$

as illustrated in Figure 7(b). Then

$$P(Y \geq \delta) \;=\; E\{g(Y)\} \;\leq\; E\{e^{\nu(Y-\delta)}\} \quad , \tag{38}$$

where for the tightest bound we want $E\{e^{\nu(Y-\delta)}\}$ as small as possible. So first minimize $E\{e^{\nu(Y-\delta)}\}$ with respect to ν. Setting

$$\frac{\partial}{\partial \nu} E\{e^{\nu(Y-\delta)}\} \;=\; 0 \quad , \tag{39}$$

we have

$$E\{Y e^{\nu Y}\} \;-\; \delta E\{e^{\nu Y}\} \;=\; 0 \quad . \tag{40}$$

First solve for $\nu = \hat{\nu}$. Then,

$$P(Y \geq \delta) \;\leq\; E\{e^{\hat{\nu}(Y-\delta)}\} \;=\; e^{-\hat{\nu}\delta} E\{e^{\hat{\nu} Y}\} \quad . \tag{41}$$

This is the Chernov bound.

Example 2.3: Determine the Chernov bound for the Gaussian tail probability function $Q(x)$.

Solution: For a zero mean, unit variance Gaussian random variable X, the Chernov bound is

$$Q(x) \;=\; P(X \geq x) \;<\; e^{-\hat{\nu}x} E\{e^{\hat{\nu}X}\} \quad , \tag{42}$$

where $\hat{\nu}$ is the solution to

$$E\{Xe^{\nu X}\} \;-\; x\; E\{e^{\nu X}\} \;=\; 0 \quad . \tag{43}$$

It is straightforward to show that, for the PDF considered here, $E\{e^{\nu X}\} = e^{\nu^2/2}$ and $E\{Xe^{\nu X}\} = \nu e^{\nu^2/2}$. The solution to Eq (43) is $\nu \;=\; \hat{\nu} \;=\; x$. Eq (42) becomes

$$Q(x) \;\leq\; e^{-x^2/2} \quad . \tag{44}$$

2.6 Random Processes

For wide-sense continuous-time (CT) stationary random process $X(t)$, the mean, autocorrelation, and autocovariance functions are defined, respectively, as

$$m = E\{X(t)\} \tag{45}$$

$$R(\tau) = E\{X(t)\; X^*(t-\tau)\} \tag{46}$$

$$C(\tau) = E\{(X(t)-m)\; (X(t-\tau)-m)^*\} = R(\tau) - |m|^2 \quad . \tag{47}$$

Note that, because the process is wide-sense stationary, these functions are not a function of time t. That is, the mean is constant, and the correlation and covariance functions are functions of only the distance in time τ between the random variables being producted. Often a wide-sense stationary random process is zero-mean. Then, $m = 0$ and $R(\tau) = C(\tau)$, and we use the terms correlation and covariance interchangeably. Zero mean processes are easier to work with, so that in practice if a process is not zero mean, the mean is often filtered out. The power spectral density (PSD) of a CT wide-sense stationary process is[2]

$$S(\omega) = \int_{-\infty}^{\infty} R(\tau)\; e^{-j\omega\tau} d\tau \;; \qquad R(\tau) = \frac{1}{2\pi} \int_{-\infty}^{\infty} S(\omega)\; e^{j\omega\tau} d\tau \quad , \tag{49}$$

i.e. the continuous-time Fourier transform (CTFT) of the autocorrelation function.

Consider a wide-sense stationary random process $X(t)$ as the input to a linear time-invariant (LTI) system (e.g. a transmitted signal through a channel, or a received signal

[2]Here we express the PSD as a function of angular frequency ω. The autocorrelation/PSD relationship is the CTFT as shown. In the Course Text, the PSD we expressed as a function of frequency f (in Hz), in which case the CTFT pair is

$$S(f) = \int_{-\infty}^{\infty} R(\tau)\; e^{-j2\pi f\tau} d\tau \;; \qquad R(\tau) = \int_{-\infty}^{\infty} S(f)\; e^{j2\pi f\tau} d\tau \quad . \tag{48}$$

through a receiver filter). Denote the LTI system impulse response $h(t)$ and corresponding frequency response $H(\omega)$. The output $Y(t)$ is also wide-sense stationary with

$$m_y = m_x \int_{-\infty}^{\infty} h(t)\ dt \tag{50}$$

$$R_y(\tau) = R_x(\tau) * h(\tau) * h^*(-\tau) \tag{51}$$

$$S_y(\omega) = S_x(\omega)\ |H(\omega)|^2 \quad . \tag{52}$$

Real-Valued Bandpass (Narrowband) Signals & Their Lowpass Equivalents

This topic is covered on Sections 2.1,9 of the Course Text. First consider a deterministic, real-valued, bandpass, narrowband signal $x(t)$ with center frequency ω_c and CTFT

$$X(\omega) = \int_{-\infty}^{\infty} x(t)\ e^{-j\omega t} dt \quad , \tag{53}$$

where $X(\omega)$, as illustrated in Figure 8, is complex symmetric. In the context of this Course, $x(t)$ will be a transmitted signal (i.e. the modulated signal that is the input to the channel).

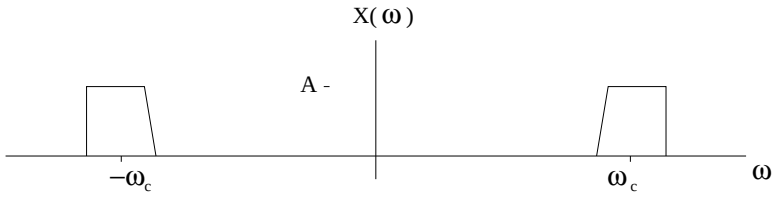


Figure 8: The spectrum of a bandpass real-valued signal.

Let $U(\omega)$ be the step function (i.e. $U(\omega) = 0,\ \omega < 0;\ U(\omega) = 1,\ \omega > 0$). The *analytic signal* for $x(t)$ is defined as follows:

$$X_+(\omega) \ = \ U(\omega)\ X(\omega) \tag{54}$$

$$x_+(t) = \int_{-\infty}^{\infty} X_+(\omega)\ e^{j\omega t} d\omega \quad . \tag{55}$$

$|X_+(\omega)|$ is sketched below for the $X(\omega)$ illustrated previously.

Note that the inverse CTFT of the frequency domain step used above is

$$u_f(t) = \frac{1}{2}\delta(t) + \frac{j}{2}h(t)\ , \qquad h(t) = \frac{1}{\pi t} \tag{56}$$

where $\delta(t)$ is the impulse function. It can be shown that $h(t)$ is a 90^o phase shifter. So, by the convolution property of the CTFT,

$$x_+(t) = x(t) * u_f(t) = \frac{1}{2}x(t) + \frac{j}{2}\ x(t) * h(t) = \frac{1}{2}\ x(t) + \frac{j}{2}\ \hat{x}(t) \tag{57}$$

where $x(t)$ and $\hat{x}(t)$ are real-valued. Also, from the definition of $x_+(t)$ and CTFT properties, note that

$$x(t) \ = \ 2\ Re\{x_+(t)\} \ = \ x_+(t) + x_+^*(t) \quad . \tag{58}$$

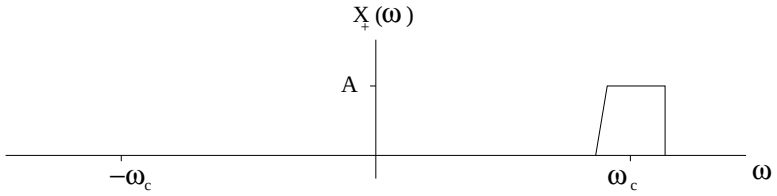


Figure 9: The spectrum of the complex analytic signal corresponding to the bandpass real-valued signal illustrated in Figure 8.

Figure 9 shows the complex analytic signal spectrum for the real-valued bandpass signal illustrated in Figure 8.

The *equivalent lowpass* (also termed *complex envelope*) of $x(t)$ is, by definition

$$x_l(t) \;=\; 2\; x_+(t)\; e^{-j\omega_c t} \tag{59}$$

$$X_l(\omega) \;=\; 2\; X_+(\omega + \omega_c) \tag{60}$$

where ω_c is the center frequency of the real-valued bandpass signal $x(t)$. We term this signal the lowpass equivalent because, as illustrated below for the example sketched out previously, $x_l(t)$ is lowpass and it preserves sufficient information to reconstruct $x(t)$ (i.e. it is the positive, translated frequency content). Note that

$$x_+(t) \;=\; \frac{1}{2}\; x_l(t)\; e^{j\omega_c t} \quad . \tag{61}$$

So,

$$x(t) = Re\{x_l(t)\; e^{j\omega_c t}\} \quad , \tag{62}$$

and also

$$X(\omega) \;=\; \frac{1}{2}\; [X_l(\omega - \omega_c) + X_l^*(-\omega - \omega_c)] \quad . \tag{63}$$

Then, given $x_l(t)$ (say it was designed), $x(t)$ is easily identified (as is $x_l(t)$ from $x(t)$).

Figure 10 shows the complex lowpass signal spectrum for the real-valued bandpass signal illustrated in Figure 8.

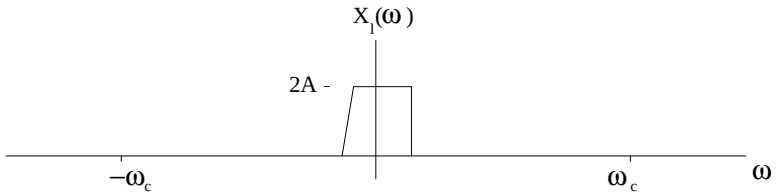


Figure 10: The spectrum of the complex lowpass signal corresponding to the bandpass real-valued signal illustrated in Figure 8.

Figure 11 illustrates how to generate $x_l(t)$ from $x(t)$. In Figure 11(a), $h(t) = \frac{1}{\pi t}$ is the impulse response of a Hilbert transform. Figure 11(b) shows a circuit based on a quadrature receiver which is essentially equivalent to the Hilbert transform circuit in Figure 11(a). The bandpass

filter is included to separate the desired signal, centered at frequency ω_c, from other signals in different frequency bands. The frequency response of the bandpass filter is

$$H_{bp}(\omega) = \begin{cases} 1 & \omega_c - \omega_m \leq \omega \leq \omega_c + \omega_m \\ 0 & otherwise \end{cases} \quad . \tag{64}$$

ω_m is the one-sided bandwidth of the desired signal.

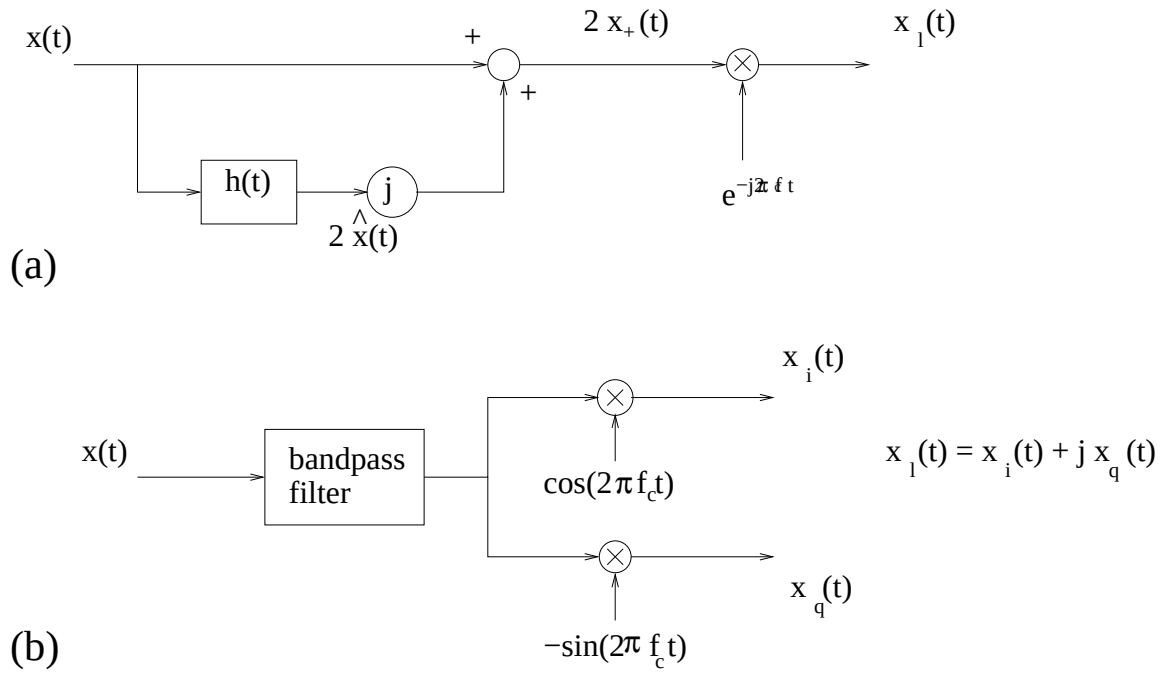


Figure 11: A receiver (complex demodulator) that generates the the complex lowpass equivalent signal $x_l(t)$ from the original real-valued bandpass signal $x(t)$: (a) Hilbert transform based; (b) quadrature receiver based.

As noted above, we will often use the lowpass equivalent $x_l(t)$ to represent $x(t)$. In general $x_l(t)$ is complex-valued. Realizing this, we introduce the following notation:

$$x_l(t) = x_i(t) + jx_q(t) \; ; \quad x_i(t) = Re\{x_l(t)\} \; ; \quad x_q(t) = Im\{x_l(t)\} \quad . \tag{65}$$

$x_i(t)$ and $x_q(t)$ are termed, respectively, the in-phase and quadrature components of $x_l(t)$. Writing $x_l(t)$ in terms of its magnitude and phase, we have

$$x_l(t) = r_x(t)\, e^{j\theta_x(t)} \; ; \quad r_x(t) = \sqrt{x_i^2(t) + x_q^2(t)} \; ; \quad \theta_x(t) = -\tan^{-1}\left(\frac{x_q(t)}{x_i(t)}\right) \quad , \tag{66}$$

and $x_i(t) = r_x(t)\ \cos(\theta_x(t))$ and $x_q(t) = r_x(t)\ \sin(\theta_x(t))$. Now we have that

$$x(t) = Re\{r_x(t)\ e^{j(\omega_c t + \theta_x(t))}\} = r_x(t)\cos(\omega t + \theta_x(t)) \quad . \tag{67}$$

Thus, $r_x(t)$ and $\theta_x(t)$ are, respectively, the *envelope* and *phase* of $x(t)$.

The *energy* of $x(t)$ is, by Parseval's theorem,

$$\mathcal{E} \;=\; \frac{1}{2\pi} \int_{-\infty}^{\infty} |X(\omega)|^2 \, d\omega \quad . \tag{68}$$

As illustrated in Figure 12, it can be calculated, from the lowpass equivalent, as

$$\mathcal{E} \;=\; \frac{1}{2}\mathcal{E}_l \;=\; \frac{1}{2}\,\frac{1}{2\pi} \int_{-\infty}^{\infty} |X_l(\omega)|^2 \, d\omega \quad . \tag{69}$$

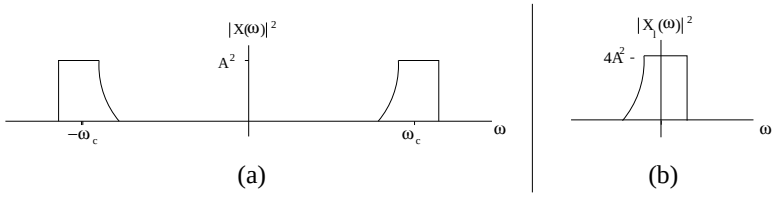


Figure 12: Energy spectra for: (a) the real-valued bandpass signal $x(t)$; (b) its complex lowpass equivalent $x_l(t)$.

Note the need for the $\frac{1}{2}$ factor. This is because the spectral levels of $X_l(\omega)$ are twice that of the positive frequency components of $x(t)$, but the negative frequency components of $x(t)$ are not present in $X_l(\omega)$.

We now extend this discussion on equivalent lowpass representation to random processes, which in the context of this Course represent modulated carriers and additive noise. Let $X(t)$ be a CT wide-sense stationary random process which is narrowband bandpass, such that its power spectral density $S_x(\omega) = 0$ for $|(\omega - \omega_c)| > W$ and $W << \omega_c$. Assume that $X(t)$ is real-valued, so that the correlation function

$$R_x(\tau) = E\{X(t)\; X(t-\tau)\} \;=\; \frac{1}{2\pi} \int_{-\infty}^{\infty} S_x(\omega)\; e^{j\omega\tau} \; d\omega \tag{70}$$

is real-valued. From the discussion above, we have that

$$\begin{aligned} X(t) \;&=\; X_i(t) \;+\; j\; X_q(t) \;-\; r_x(t)\cos(\omega_c t + \theta_x(t)) \\ &=\; Re\{X_l(t)e^{j\omega_c t}\} \end{aligned} \tag{71}$$

where $X_l(t)$ is the lowpass equivalent lowpass of $X(t)$.

Summarizing properties of the autocorrelation/PSD functions of $X(t)$ and its lowpass equivalent $X_l(t)$, from Sect. 2.9 of the Course Text we have that

$$R_x(\tau) = Re\{R_{x_l}(\tau)e^{j\omega_c\tau}\} \tag{72}$$

$$S_x(\omega) = \frac{1}{2}\left[S_{x_l}(\omega - \omega_c) + S_{x_l}(-\omega - \omega_c)\right] \quad . \tag{73}$$

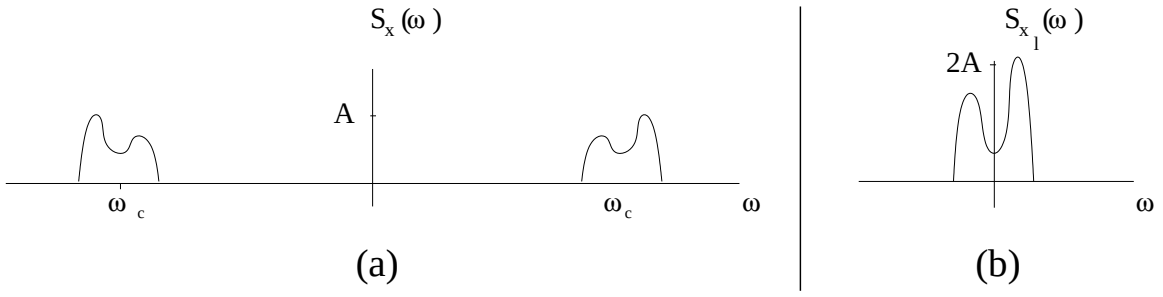


Figure 13: Bandpass and equivalent lowpass random process power spectra.

Bandpass White Noise and Power

Additive receiver noise $N(t)$ will often be bandpass white. Figure 14 illustrates its power spectral density. Its autocorrelation function is

$$R_n(\tau) \;=\; N_0 \; \frac{\sin(\pi B \tau)}{\pi \tau} \; \cos(\omega_c \tau) \quad . \tag{74}$$

Considering the power of the bandpass white noise process $N(t)$ and its lowpass representation $N_l(t)$, we see from the figures or equations above that

$$\mathcal{P} = \frac{1}{2\pi} \; \int_{-\infty}^{\infty} S_n(\omega) \; d\omega = N_0 \; \frac{B}{2\pi} \tag{75}$$

and

$$\mathcal{P}_l = \frac{1}{2\pi} \; \int_{-\infty}^{\infty} S_{n_l}(\omega) \; d\omega = N_0 \; \frac{B}{2\pi} \quad . \tag{76}$$

It is interesting and expected that the noise power be proportional to the spectral level and bandwidth. What is of particular importance is that $\mathcal{P}_l = \mathcal{P}$.

Recall from our previous discussion on deterministic bandpass signals that, for a deterministic signal $x(t)$ and its equivalent lowpass representation $x_l(t)$, the energy relationship $\mathcal{E}_l = 2\mathcal{E}$. We see that there is a *factor of 2* difference between the signal energy to noise power ratios (SNR's) of the bandpass and equivalent lowpass representations. The equivalent lowpass representation shows an SNR that is factor of 2 larger than the actual SNR (of the bandpass signals). *This is an important detail that is difficult to ignore when studying communication system performance.*

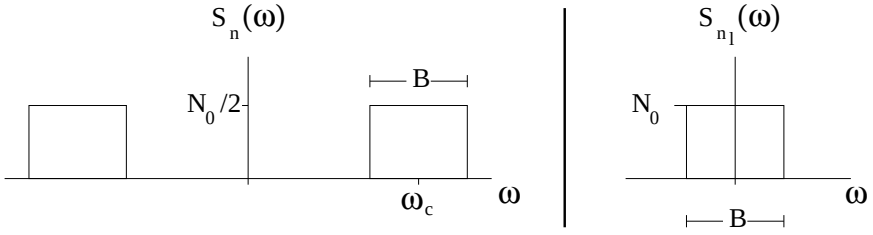


Figure 14: Power spectrum density of bandlimited white noise.

3 Modulation & Detection

This Section of the Notes corresponds to discussions in Sections 3.1-2 and 4.1-6 of the Course Text. In this Section of the Course we overview digital modulation schemes. That is, we consider approaches to mapping a binary source onto a carrier sinusoid for transmission over a communication channel. This topic is not central to this Course. Rather, we need to cover it to understand the subsequent central Course topics of channel capacity, channel coding, and trellis coded modulation.

We will need familiarity with several modulation schemes, including pulse amplitude modulation (PAM), phase shift keying (PSK), quadrature amplitude modulation (QAM) frequency shift keying (FSK), and multidimensional orthogonal modulation. For performance analysis of various block and convolutional channel coding techniques, we will review performance analyses of binary versions of PAM, PSK and FSK with both coherent and noncoherent reception.

For a discussion on channel capacity, we will need an understanding of multidimensional orthogonal modulation. It is with an argument based on this approach to modulation that Shannon developed channel capacity. In this Section of the Course we will describe this modulation approach.

For analysis of nonbinary codes (such as Reed-Solomon block codes) and to understand and analyze trellis coded modulation, we need exposure to higher order (nonbinary) modulation schemes. For this purpose, we will cover M-ary FSK ($M > 2$) and M-ary quadrature amplitude modulation (QAM) in this Section.

To understand a principal motivation for trellis coded modulation, we need to consider bandwidth characteristics of the various modulation schemes. We will overview this topic in this Section of the Course.

We begin in Subsection 3.1 with a development of a mathematical framework for representing and analyzing modulation schemes, which will result in what is termed the *signal space*. In In this Subsection 3.2 we also describe the modulation schemes listed above. In Subsection [3.2] we describe optimum detectors for these schemes. We will see that they are composed of matched filters followed by threshold detectors. We close with Subsection [3.3] which develops the performance analyses of the binary modulation schemes mentioned above. Included in this will be comments concerning modulated signal bandwidth.

3.1 Digital Modulation

The objective here is to classify digital modulation schemes and introduce several schemes which will be considered later in the Course, and to describe them in terms of their signal space representations.

The digital communication problem is to transmit a binary information sequence $\{x_n\}$. Digital modulation is the mapping of the binary symbols x_n to a transmitted waveform that carries this sequence. As part of a modulation scheme, the binary sequence $\{x_n\}$ is *mapped* to a sequence of symbols (waveforms). Given M symbols, the binary data is arranged into blocks of $k = log_2(M)$ bits (M is assumed to be a power of 2, i.e. $M = 2^k$). Then each symbol represents k bits. The symbol rate is $\frac{R}{k}$ where R is the bit rate (in bits/sec.).

Let $s_m(t);\ m = 1, 2, \cdots, M$ denote the set of symbols. The transmitted signal $s(t)$ is derived from the sequence of symbols representing the information sequence $\{x_n\}$. Thus, we can consider digital modulation to be a cascade of two mappings: first from the blocks of k binary values to the symbols, and then from the symbols to the transmitted waveform. In this Section we will focus on the second mapping.

3.1.1 Modulation Classifications

Digital modulation schemes can be classified as either memoryless or memory, or as either linear or nonlinear. Below we will first generally discuss digital modulation within the context of these classifications. We will then specifically consider several linear, memoryless schemes (e.g. PAM, PSK, QAM, FSK).

Linear, Memoryless Modulation

In a linear modulation scheme, the principle of superposition applies in the mapping from the blocks of k binary values x_n to the transmitted waveform $s(t)$.

We assume that the mapping from the blocks of x_n's to the symbols will be linear (this basically means that there is no nonlinear operation on the sequence $\{x_n\}$ before the mapping to symbols). Then, linearity will hold if the mapping from the sequence of symbols representing $\{x_n\}$ to $s(t)$ is linear; e.g.

$$s(t) = \sum_n s_{m(n)}(t - nT) \quad , \tag{1}$$

where $m(n)$ indicates that the symbol selected (at time nT) depends on the k information bits at that time. A modulation scheme will be memoryless if there is no memory in either the mapping from the blocks of x_n to the symbols, or the mapping from the symbols to $s(t)$. Thus a one-to-one mapping from each block of k x_n's to a $s_m(t)$, followed by a superposition of symbols as shown in Eq. (1) would constitute a linear, memoryless modulation scheme.

Memory

Memory can be introduced either in the mapping of the information sequence to the symbols or in the mapping of the symbols to the transmitted signal. An example of the former is differential encoding (such as NRZI). An example of the latter is CPM (which is also nonlinear) in which the sequence of symbols is integrated (to form the phase of the transmitted waveform).

3.1.2 Signal Space Representation & The Symbol Constellation

Let $\{s_m(t);\ m = 1, 2, \cdots, M\}$ be M waveforms used to represent the M symbols of some modulation scheme. These symbols are limited in time to the range $[0, T]$. Consider orthonormal expansion of these waveforms in terms of the $N \leq M$ orthonormal functions $\phi_k(t); k = 1, 2, ..., N$ which form a basis for the $s_m(t)'s$. The constant N is called the *dimension* of the modulation scheme. (The symbol waveforms and corresponding basis functions could be either real-valued bandpass or their complex-valued (in-phase/quadrature) lowpass

equivalents. Here will use complex notation.) The symbol expansions are

$$s_m(t) = \sum_{k=1}^{N} s_{mk}\ \phi_k(t) = \underline{\phi}(t)\ \underline{s}_m \tag{2}$$

where

$$s_{mk}\ = < s_m(t), \phi_k(t) > = \int_0^T\ s_m(t)\ \phi_k^*(t)\ dt \tag{3}$$

is the inner product of $s_m(t)$ and $\phi_k(t)$, $\underline{s}_m = [s_{m1}, s_{m2}, \cdots, s_{mN}]^T$, and $\underline{\phi}(t) = [\phi_1(t), \phi_2(t), \cdots, \phi_N(t)]$.

Signal Space Diagram

A signal space diagram is a plot, in N-dimensional space, of the symbol expansion coefficient vectors $\underline{s}_m$, the signal space representation of $s_m(t)$. The configuration of these vectors in the signal space is called the *constellation* of the modulation scheme. Below we will illustrate the constellations for several linear memoryless modulation schemes.

Noting that $\int_0^T \underline{\phi}^H(t)\ \underline{\phi}(t)\ dt = \underline{I}_N$ (the N-dimensional identity matrix), since these basis functions are assumer normalized, the *energy* of a symbol waveform $s_m(t)$ can be written as

$$\mathcal{E}_m\ =\ \int_0^T |s_m(t)|^2\ dt\ =\ \int_0^T |\underline{\phi}(t)\ \underline{s}_m|^2\ dt\ =\ \underline{s}_m^H\ \underline{s}_m \tag{4}$$

(i.e. the sum of the squares of the signal space representation coefficients). The *Euclidean distance* between two symbol waveforms an important indicator of the ability to distinguish between them at the receiver when observed in noise. The Euclidean distance between symbols $s_m(t)$ and $s_k(t)$ is

$$\begin{aligned} d_{km}^{(e)}\ &=\ \left(\int_0^T |\underline{\phi}(t)\ \underline{s}_m - \underline{\phi}(t)\ \underline{s}_k|^2\ dt\right)^{\frac{1}{2}} \\ &=\ \left(\underline{s}_m^H\ \underline{s}_m + \underline{s}_k^H\ \underline{s}_k - \underline{s}_k^H\ \underline{s}_m - \underline{s}_m^H\ \underline{s}_k\right)^{\frac{1}{2}} \\ &=\ \left(\mathcal{E}_m + \mathcal{E}_k - 2Re\{\underline{s}_m^H\ \underline{s}_k\}\right)^{\frac{1}{2}} \\ &=\ \left(\mathcal{E}_m + \mathcal{E}_k - 2\sqrt{\mathcal{E}_m\ \mathcal{E}_k}\ \rho_{mk}\right)^{\frac{1}{2}} \end{aligned} \tag{5}$$

where

$$\rho_{mk} = \cos\theta_{mk} = \frac{Re\{\underline{s}_m^H\ \underline{s}_k\}}{||\underline{s}_m||\ ||\underline{s}_k||}\quad , \tag{6}$$

termed the correlation coefficient for $s_m(t)$ and $s_k(t)$, is the cosine of the angle θ_{mk} between the two signal space representations $\underline{s}_m$ and $\underline{s}_k$. For equal energy waveforms (i.e. for $\mathcal{E}_m = \mathcal{E}_k = \mathcal{E}$),

$$d_{km}^{(e)} = (2\mathcal{E}(1 - \cos\theta_{mk}))^{\frac{1}{2}} \tag{7}$$

which is maximized for $\theta_{mk} = 180^o$ (e.g. when $\underline{s}_m$ and $\underline{s}_k$ are collinear but of opposite sign).

As we will see, effective digital communications occurs when Euclidean distances between digital transmission symbol waveforms are large. Typically, for multiple symbol digital modulation schemes, bit-error-rate is dominated by the minimum of the distances between all of the symbols.

3.1.3 Linear Memoryless Modulation Scheme Examples

In this Subsection we describe several linear memoryless modulation schemes that will be used in this Course to explore channel coding performance.

Pulse Amplitude Modulation (PAM)

Pulse Amplitude Modulation (PAM) is a $N = 1$ dimensional, M-symbol, memoryless, linear modulation scheme. The M symbols are as follows:

$$s_m(t) = A_m g(t)\ \cos(2\pi f_c t) = Re\{A_m g(t)\ e^{j2\pi f_c t}\}\ ; 0 \le t \le T \quad m = 1, 2, \cdots, M \tag{8}$$

where $\cos(2\pi f_c t)$ is the carrier sinusoid, and $g(t)$ is a real-valued pulse shaping waveform. Symbols are distinguished by the different amplitudes of the carrier,

$$A_m = (2m - 1 - M)d \qquad m = 1, 2, \cdots, M \quad , \tag{9}$$

where $2d$ is the distance between symbol amplitudes. Let $\mathcal{E}_g = \int_0^T g^2(t)dt$ be the energy of the pulse shaping waveform. The energy of the a symbol $s_m(t)$ is $\mathcal{E}_{s_m} = \frac{A_m^2}{2}\mathcal{E}_g$.

In terms of signal space representation, the normalized basis function for the symbol waveforms is

$$\phi(t) = \sqrt{2/\mathcal{E}_g}\ g(t)\ \cos(2\pi f_c t) \quad . \tag{10}$$

A transmitted PAM symbol can be written as

$$s_m(t) = s_m\ \phi(t)\ ; \qquad s_m = A_m\sqrt{\mathcal{E}_g/2} \quad . \tag{11}$$

The 1-dimensional signal space diagram (i.e. the constellation) for PAM is illustrated in Figure 15 for $M = 2$ and $M = 4$. The Euclidean distance between 2 adjacent samples is the minimum distance, which is

$$d_{min}^{(e)} = \sqrt{g(s_m - s_{m-1})^2} = \left((\mathcal{E}_g/2)\ (2d(m - (m-1)))^2\right)^{1/2} = d\sqrt{2\mathcal{E}_g} \quad . \tag{12}$$

Noise perturbs the received symbol and therefore its signal space representation. If perturbed too much, the symbol will be mistaken for some other symbol, most likely an adjacent one.

Given a binary information sequence $\{x_n\}$, at time n, $k = \log^2(M)$ bits are mapped to a corresponding symbol $s_{m(n)}(t)$. Then, the transmitted signal is

$$s(t) = \sum_n s_{m(n)}(t - nT) = \sum_n A_{m(n)}\ \sqrt{\mathcal{E}_g/2}\ \phi(t - nT) \quad . \tag{13}$$

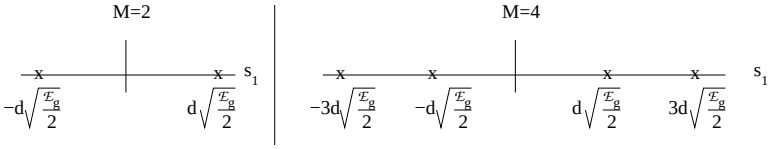


Figure 15: PAM signal space representation for $M = 2$, $M = 4$.

Phase Shift Keying (PSK)

For PSK, the M symbols are

$$s_m(t) = g(t)\cos\left(2\pi f_c t + 2\pi(m-1)/M\right) \;\; ; \;\; 0 \le t \le T \quad m = 1, 2, \cdots, M \;\; . \tag{14}$$

Symbols are distinguished by the different phases of the carrier. As with PAM, $g(t)$ is a real-valued pulse shaping waveform. To derive the signal space representation of PSK, we can use trigonometric identities to rewrite Eq. 14 as

$$\begin{aligned} s_m(t) &= g(t)\left[\cos\left(\frac{2\pi(m-1)}{M}\right)\cos\left(2\pi f_c t\right) - \sin\left(\frac{2\pi(m-1)}{M}\right)\sin\left(2\pi f_c t\right)\right] \\ &= s_{m1}\;\phi_1(t) + s_{m2}\;\phi_2(t) \;=\; [\phi_1(t),\;\phi_2(t)]\;\underline{s}_m \end{aligned} \tag{15}$$

where the orthonormal basis functions are

$$\phi_1(t) = \sqrt{\frac{2}{\mathcal{E}_g}}g(t)\cos(2\pi f_c t) \;\; ; \qquad \phi_2(t) = -\sqrt{\frac{2}{\mathcal{E}_g}}g(t)\sin(2\pi f_c t) \;\; ; \qquad 0 \le t \le T \tag{16}$$

and the signal space representation for the m^{th} symbol is

$$\underline{s}_m = [s_{m1}, s_{m2}]^T = \left[\sqrt{\frac{\mathcal{E}_g}{2}}\cos\left(2\pi(m-1)/M\right) \;\; , \;\; \sqrt{\frac{\mathcal{E}_g}{2}}\sin\left(2\pi(m-1)/M\right)\right]^T \;\; . \tag{17}$$

This modulation scheme is $N = 2$-dimensional because any symbol can be represented as linear combinations of $\phi_1(t)$ and $\phi_2(t)$. These two basis functions are referred to as the in-phase and quadrature components, respectively. PSK is a linear modulation scheme because the transmitted signal $s(t)$ is constructed as a superposition of time shifted $s_m(t)$'s, which are in turn formed as a linear combination of basis functions. It is memoryless because an $\underline{s}_m$ depends on only one block of x_n's, and the superposition of the time shifted $s_m(t)$'s is memoryless.

The energy of a PSK symbol can be determined from any of the representations above. Using Eq. 15, the energy is the sum of the squares for the symbol coefficients, which from Eq. 17 is

$$\mathcal{E}_m = \frac{\mathcal{E}_g}{2} \;\; . \tag{18}$$

All symbols have the same energy.

For $M = 2$, we see that

$$\underline{s}_1 = \sqrt{\frac{\mathcal{E}_g}{2}}\;\;[1\;,\;0]^T \;\; , \quad \underline{s}_2 = \sqrt{\frac{\mathcal{E}_g}{2}}\;\;[-1\;,\;0]^T \;\; . \tag{19}$$

So, $\phi_2(t)$ is not used, and thus for this case the modulation scheme is only 1-dimensional. Comparing it to PAM with $M = 2$, we see that the two schemes are identical.

For $M = 4$, we have

$$\underline{s}_1 = \sqrt{\frac{\mathcal{E}_g}{2}}\;[1,\;0]^T; \quad \underline{s}_2 = \sqrt{\frac{\mathcal{E}_g}{2}}\;[0,\;1]^T; \quad \underline{s}_3 = \sqrt{\frac{\mathcal{E}_g}{2}}\;[-1,\;0]^T; \quad \underline{s}_4 = \sqrt{\frac{\mathcal{E}_g}{2}}\;[0,\;-1]^T \;\; . \tag{20}$$

Figure 16 illustrates the signal spaces for $M = 2$ and $M = 4$ PSK.

For a given M, the symbols are equal-distance from the origin in 2-dimensional signal space, and evenly distributed in phase. The minimum Euclidean distance is the distance between adjacent symbols,

$$d_{min}^{(e)} = \sqrt{\mathcal{E}_g(1 - cos(2\pi/M))} \quad . \tag{21}$$

The transmitted signal $s(t)$ is constructed in a manner similar to PAM, i.e.

$$s(t) = \sum_n s_{m(n)}(t - nT) = \sum_n g(t - nT) \cos\left(2\pi f_c t + 2\pi(m(n) - 1)/M\right) \tag{22}$$

where $2\pi(m(n) - 1)/M$ is the phase used at symbol time n to represent the block of $k = \log_2(M)$ information samples from $\{x_n\}$.

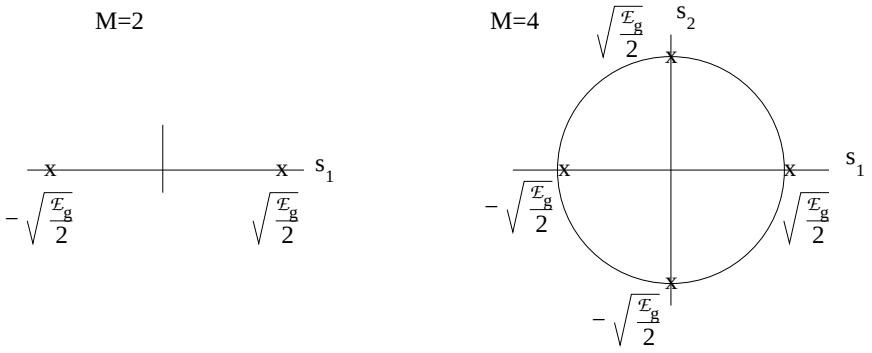


Figure 16: PSK signal space representation for $M = 2$, $M = 4$.

Quadrature Amplitude Modulation (QAM)

QAM is a generalization of the 2-dimensional PSK modulation scheme, where symbols are distinguished by varying both the amplitude and phase of the carrier (see Eq. 14), or equivalently the coefficients of both the in-phase and quadrature basis functions (see Eq. 15). The M symbols are

$$s_m(t) = V_m g(t) \cos\left(2\pi f_c t + \theta_m\right) \;\; ; \;\; 0 \le t \le T \quad m = 1, 2, \cdots, M \quad , \tag{23}$$

where V_m and θ_m are the magnitude and phase of the m^{th} symbol. To derive the signal space representation of QAM, we can rewrite Eq. 23 as

$$s_m(t) = s_{m1}\ \phi_1(t) + s_{m2}\ \phi_2(t) \ = \ [\phi_1(t),\ \phi_2(t)]\ \underline{s}_m \tag{24}$$

where the orthonormal basis functions are the same as for PSK (i.e. see Eq. 16).

For the m^{th} symbol, the signal space representation depends on both V_m and θ_m:

$$\underline{s}_m = [s_{m1}, s_{m2}]^T = \left[\sqrt{\frac{\mathcal{E}_g}{2}}\ \ V_m \cos\theta_m\ ,\ \sqrt{\frac{\mathcal{E}_g}{2}}\ \ V_m \sin\theta_m\right]^T \quad . \tag{25}$$

From Eq. 24, the energy of a QAM symbol is the sum of the squares of the symbol coefficients, which from Eq. 25 is

$$\mathcal{E}_m = \frac{V_m^2 \mathcal{E}_g}{2} \quad . \tag{26}$$

Symbols will not all have the same energy since amplitude as well as phase varies.

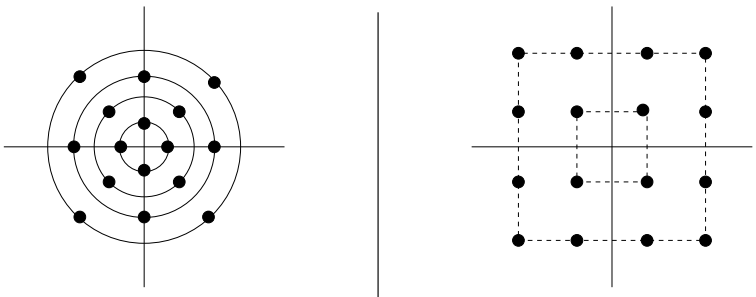
Although the magnitude and phase of QAM symbols can be selected in any way, as illustrated in Figure 17 the two common schemes are to:

1. select symbols on a circular grid in the signal space; or
2. select symbols on a rectangular grid in the signal space.

The transmitted signal $s(t)$ is constructed in a manner similar to PAM and PSK,

$$s(t) = \sum_n s_{m(n)}(t - nT) = \sum_n V_{m(n)} g(t - nT) \cos\Big(2\pi f_c (t - nT) + \theta_{m(n)}\Big) \tag{27}$$

where $V_{m(n)}$ and $\theta_{m(n)}$ at symbol time n are selected to represent the block of k information samples from $\{x_n\}$.



(a) M=16 QAM on circular grid (b) M−16 QAM on rectangular grid

Figure 17: Signal space representations for two QAM schemes.

Frequency Shift Keying (FSK)

Several higher dimensional linear modulation schemes are commonly used, including frequency, time and code division multiplexing schemes. FSK allocates different frequencies to different symbols, using basis functions

$$\phi_k(t) = \sqrt{\frac{2}{T}} \cos(2\pi(f_c + k\Delta f)t) \; ; \quad 0 \le t \le T \; ; \quad k = 1, 2, \cdots, N \tag{28}$$

with Δf the frequency spacing. The $M = N$ symbols are simply

$$s_m(t) = \sqrt{\mathcal{E}}\, \phi_m(t) \; ; \quad m = 1, 2, \cdots, M \quad , \tag{29}$$

where $\mathcal{E}$ is the energy of a symbol. For $M = 2$ FSK, termed Binary FSK, the values **0** and **1** of a binary sample x_n are transmitted by pulses represented as follows:

$$\begin{aligned} &0 \text{ is represented by a } \sqrt{\tfrac{2\mathcal{E}}{T}} cos((2\pi f_c - 2\pi\Delta f)t) && 0 \le t \le T \; , \\ &1 \text{ is represented by a } \sqrt{\tfrac{2\mathcal{E}}{T}} cos((2\pi f_c + 2\pi\Delta f)t) && 0 \le t \le T. \end{aligned}$$

The *modulation index* in FSK is defined as $f_k = \Delta f / f_c$. Power Spectral Density (PSD) is an important issue in selection of a modulation scheme. For large values of f_k, the PSD will effectively consist of two non-overlapping PSD's. For small f_k, the two spectra merge.

One obvious way to generate a binary FSK signal is to switch between two independent oscillators according to whether the data bit is a 0 or 1. Normally, this form of FSK generation results in a waveform that is *discontinuous* at the switching times. Because of these possible discontinuities, FSK can have undesirable PSD spread (i.e. it uses frequency resources inefficiently). This motivates continuous phase approaches, which employ memory to assure that phase transitions between symbols are continuous. Continuous phase approaches are nonlinear and have memory. They will not be considered in this Course.

3.2 Optimum Detection

This Section covers selected topics form Sections 4.1-5 of the Course Text. We address the problem of symbol detection, assuming that:

1. the modulation scheme is memoryless,
2. successive symbol waveforms do not overlap in time,
3. the symbols are uncorrelated in time, and
4. the channel has no memory (i.e. it introduces no symbol waveform distortion so as to result in intersymbol interference (ISI)).

In this case, the optimum receiver processes one symbol at a time, deriving a symbol estimate each symbol interval by processing the noisy received signal over that symbol's interval. We further assume that the signal waveforms are received in additive white Gaussian noise (AWGN).

If any of the assumptions stated above do not hold, then the optimum receiver must in general process the received signal over its entire duration to estimate all of the symbols concurrently. That is, for optimum reception we must perform sequence estimation. Concerning assumption 3, channel encoding does introduce correlation between symbols. We will consider optimum receiver structures for channel encoded digital communication in Sections 7-10 of the Course. Throughout most of this Course we are not concerned with situations in which assumptions 1, 2 or 4 do not apply. That said, the decoder algorithms we develop for convolution codes are generally applicable for any digital communication system with memory (i.e. when any or all of the above assumptions do not apply). Also, time permitting, we discuss ISI mitigation in Section 9 of the Course when covering turbo coding.

In this Subsection we show that the optimum digital communication receiver under the assumptions stated above has the basic structure illustrated in Figure 18. We first describe a commonly used receiver filter structure, which consists of N parallel correlators or matched filters – one for each of the basis functions $\phi_i(t);\ i = 1, 2, \cdots, N$ of the N dimensional linear modulation scheme. We then address the issue of optimum detection, considering two commonly considered optimality criteria – maximum likelihood (ML) and maximum a posterior (MAP). We show that a receiver front end consisting of matched filters followed by symbol-rate samplers generates sufficient receiver signal statistics for these optimality criteria. Optimum detection is then implemented as a threshold detector, where the threshold depends on which optimality criterion is desired. The coherent receiver and noncoherent receiver detection problems are treated separately.

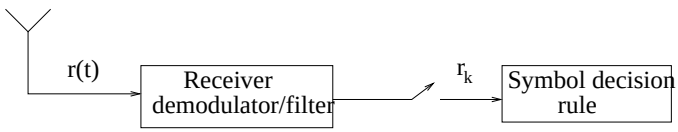


Figure 18: Receiver block diagram considered in this Subsection.

Signal Representation Review:

1. *Lowpass Signal Representation:*

   Real-valued passband signal: $s(t) \overset{CTFT}{\Longleftrightarrow} S(f)$

   Equivalent lowpass signal: $s_l(t) \overset{CTFT}{\Longleftrightarrow} S_l(f)$

$$s_l(t) \;=\; s_+(t)\; e^{-j2\pi f_c t}\;; \qquad s(t) = Re\{s_l(t)\; e^{j2\pi f_c t}\} \tag{30}$$

   where $s_+(t)$ is the positive frequency component of passband $s(t)$ (see Section 2.1 of the Course Text). $s_l(t)$ is typically derived from $s(t)$ using a quadrature receiver.

$$S(f) = \frac{1}{2}\left[S_l(f - f_c) + S_l^*(-f - f_c)\right] \tag{31}$$

   The signal energy in the bandpass signal, $\mathcal{E}$, and that in the lowpass equivalent, $\mathcal{E}_l$, are related as

$$\mathcal{E} \;=\; \frac{1}{2}\,\mathcal{E}_l \quad . \tag{32}$$

   The spectral level of the additive white noise is defined as $\frac{N_0}{2}$ and the corresponding lowpass equivalent has spectral level N_0.

2. *Signal Space Concept:*

   The symbol waveforms are:

$$s_m(t) = \sum_{k=1}^{N} s_{mk}\;\phi_k(t) \;=\; \underline{\phi}(t)\;\underline{s}_m \qquad m = 1, 2, \cdots, M \quad . \tag{33}$$

   Note that the signal space representation for the m^{th} symbol is $\underline{s}_m$, which is an N dimensional column vector. Since the basis functions are normalized, the symbol energies are $\mathcal{E}_m = ||\underline{s}_m||^2 = \underline{s}_m^H \underline{s}_m$. (These are the energies of the bandpass signals.)

3. *Inner Product:*

   The inner product between two real-valued signals $r(t)$ and $f(t)$, each limited in duration to $0 \le t \le T$, is

$$< r(t), \phi(t) >= \int_0^T r(t)\;\phi(t)\;dt = Re\{\frac{1}{2}\int_0^T r_l(t)\;\phi_l^*(t)\;dt\} \tag{34}$$

3.2.1 Correlation Demodulator & Matched Filter

Consider a set of symbols $s_m(t);\ m = 1, 2, \cdots, M;\ 0 \le t \le T$ received in Additive White Gaussian Noise (AWGN). The assumption that the noise is white, with spectral level $\frac{N_0}{2}$, effectively means that it has a much broader bandwidth than the receiver (which has bandwidth dictated by the symbols), and it has a flat PSD of level $\frac{N_0}{2}$ over this receiver bandwidth. Given the received signal

$$r(t) = s_m(t) + n(t)\ ; \qquad 0 \le t \le T \tag{35}$$

the *symbol detection* objective is to process $r(t);\ 0 \le t \le T$ to decide which of the M symbols was transmitted.

Correlation Demodulator

This receiver filter structure correlates the received signal $r(t);\ 0 \le t \le T$ with each of the basis function $\phi_k(t);\ k = 1, 2, \cdots, N$. (Note that for a $N = 1$ dimensional modulation scheme, $r(t)$ is simply correlated with the symbol shape.) The correlator computes

$$\begin{aligned} r_k &= \int_0^T r(t)\ \phi_k(t)\ dt \qquad k = 1, 2, ..., N \\ &= \frac{1}{2}\ Re\{\int_0^T r_l(t)\ \phi_{lk}^*(t)\ dt\}\ =\ \frac{1}{2}\ \int_0^T Re\{r_l(t)\ \phi_{lk}^*(t)\}\ dt\ , \end{aligned} \tag{36}$$

where $r_l(t)$ is the *equivalent lowpass* of $r(t)$, generated by a basebanding quadrature receiver, and the $\phi_{lk}(t)$ are the equivalent baseband basis functions. Note that the implementation of this correlator assumes that the $\phi_k(t)$ are known exactly, which assumes that the carrier phase is known. That is, the receiver is *carrier-phase synchronized* and subsequent processing is termed *coherent processing*. It also assumes that the received signal temporal location is known (i.e. the receiver is symbol synchronized).

As we will see, an optimum decision is then made based on the output vector

$$\underline{r} = [r_1, r_2, ..., r_N]^T \quad . \tag{37}$$

Figure 19 shows both the bandpass and equivalent lowpass implementation. Note that receivers for bandpass communication systems often quadrature demodulate the received signal $r(t)$ and then process as shown in this figure.

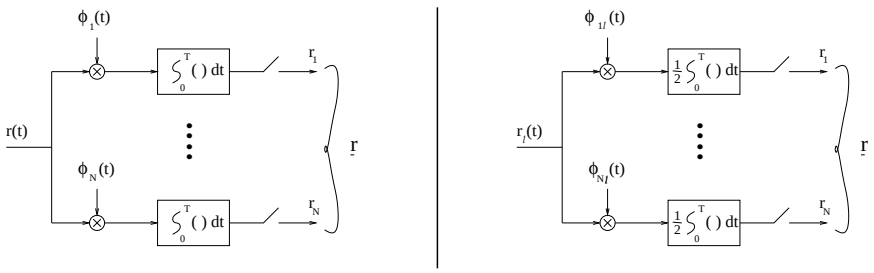


Figure 19: The correlator receiver.

So, why correlate $r(t)$ with the $\phi_k(t)$? We will answer this formally in Subsection 3.2.2. For now, consider correlating each $s_m(t)$ with $r(t)$ (perhaps so as to decide which $s_m(t)$ is most correlated with $r(t)$). We have

$$\begin{aligned} \int_0^T r(t)\, s_m(t)\, dt &= \int_0^T r(t) \sum_{k=1}^{N} s_{mk}\, \phi_k(t)\, dt \\ &= \sum_{k=1}^{N} s_{mk} \int_0^T r(t)\, \phi_k(t)\, dt \\ &= \sum_{k=1}^{N} s_{mk}\, r_k \;=\; \underline{s}_m^T\, \underline{r}\ . \end{aligned} \tag{38}$$

This establishes that instead of correlating $r(t)$ which each $s_m(t)$ we can correlate $r(t)$ with each $\phi_k(t)$ instead. This is advantageous whenever $N < M$, which will be the case in PAM, PSK and QAM with large M.

Some Characteristics of the Correlation Demodulator

Since

$$r(t) = s_m(t) + n(t) \quad , \tag{39}$$

we have that

$$\begin{aligned} r_k &= \int_0^T r(t)\, \phi_k(t)\, dt \\ &= \int_0^T s_m(t)\, \phi_k(t)\, dt \;+\; \int_0^T n(t)\, \phi_k(t)\, dt \\ &= s_{mk} \;+\; n_k \end{aligned} \tag{40}$$

where s_{mk} is a signal space coefficient and $n_k = \int_0^T n(t)\, \phi_k(t)\, dt$ is the correlation between the noise and the basis function. So,

$$\underline{r} \;=\; \underline{s}_m \;+\; \underline{n}\ . \tag{41}$$

Assume that $n(t)$ is zero mean, and that it is statistically independent of the $s_m(t)$'s . This latter assumption, that the additive noise is independent of the information, is reasonable. Then the *mean* of r_k is

$$\begin{aligned} E\{r_k\} &= E\{s_{mk} + n_k\} \\ &= E\{s_{mk}\} + E\{n_k\} \\ &= s_{mk} + E\left\{ \int_0^T n(t)\, \phi_k(t)\, dt \right\} \\ &= s_{mk} + \int_0^T E\{n(t)\}\, \phi_k(t)\, dt \\ &= s_{mk} \quad . \end{aligned} \tag{42}$$

Thus,

$$E\{\underline{r}\} \;=\; \underline{s}_m \quad , \tag{43}$$

and

$$E\{\underline{n}\} \;=\; \underline{0}_M \quad . \tag{44}$$

The *covariance* of a pair $\{r_k, r_l\}$ is

$$Cov\{r_k, r_l\} \;=\; E\{(r_k - E\{r_k\})(r_l - E\{r_l\})\} \;=\; E\{n_k\, n_l\} \;\;, \tag{45}$$

so that

$$Cov\{\underline{r}, \underline{r}\} \;=\; E\{\underline{n}\;\underline{n}^T\} \;\;. \tag{46}$$

We have that

$$\begin{aligned} Cov\{n_k, n_l\} &= E\left\{\int_0^T n(t)\;\phi_k(t)\;dt \cdot \int_0^T n(\tau)\;\phi_l(\tau)\;d\tau\right\} \\ &= \int_0^T \int_0^T E\{n(t)\;n(\tau)\}\;\phi_k(t)\;\phi_l(\tau)\;dt\;d\tau \\ &= \frac{N_0}{2}\int_0^T \int_0^T \delta(t-\tau)\;\phi_k(t)\;\phi_l(\tau)\;dt\;d\tau = \frac{N_0}{2}\int_0^T \phi_k(\tau)\;\phi_l(\tau)\;d\tau \\ &= \frac{N_0}{2}\;\delta(k-l) \quad . \end{aligned} \tag{47}$$

In going from line 2 to 3 in Eq. 47 above, we make use of the fact that

$$R_{nn}(\tau) = \int_{-\infty}^{\infty} S_{nn}(f)\;e^{j2\pi fT}\;df \;=\; \frac{N_0}{2}\;\delta(\tau) \quad . \tag{48}$$

Thus, given that $r(t) = s_m(t) + n(t)$, the correlator output vector has covariance matrix

$$E\{(\underline{r} - \underline{s}_m)(\underline{r} - \underline{s}_m)^T\} = \underline{C}_x = \frac{N_0}{2}\;\underline{I}_N \quad , \tag{49}$$

where $\underline{I}_N$ is the $N \times N$ identity matrix. (Note the noise variance is $\sigma_n^2 = \frac{N_0}{2}$.)

It is well known that if the correlator input $r(t)$ is Gaussian, then so will be the correlator outputs $r_k; k = 1, 2, \cdots, N$. This is due to the fact that the correlator is a linear operator, and linear operators derive Gaussian outputs from Gaussian inputs. Thus, for a given symbol $\underline{s}_m$, since $\underline{r}$ has mean $\underline{s}_m$ and covariance matrix $\underline{C}_x$, its joint Probability Density Function (PDF), "conditioned" on $\underline{s}_m$, is

$$\begin{aligned} p(\underline{r}/\underline{s}_m) &= \frac{1}{(2\pi)^{N/2}(\det \underline{C}_x)^{1/2}}\;\; e^{-(\underline{r}-\underline{s}_m)^T \underline{C}_x^{-1}(\underline{r}-\underline{s}_m)/2} \\ &= \frac{1}{(2\pi)^{N/2}(N_0/2)^{N/2}}\;\; e^{-|\underline{r}-\underline{s}_m|^2/2(N_0/2)} \\ &= \prod_{k=1}^{N}\; p(r_k/s_{mk}) \;; \qquad p(r_k/s_{mk}) = \frac{1}{\sqrt{\pi N_0}}\;\; e^{-(r_k - s_{mk})^2/N_0} \quad . \end{aligned} \tag{50}$$

This joint PDF of $\underline{r}$ will be used in the design of optimum detectors.

Figure 20 depicts the PDF's of $\underline{r}$, conditioned on different transmitted symbols, for an $N = 2$ dimensional modulation scheme. Notice that this figure resembles the signal space representation of the modulation scheme.

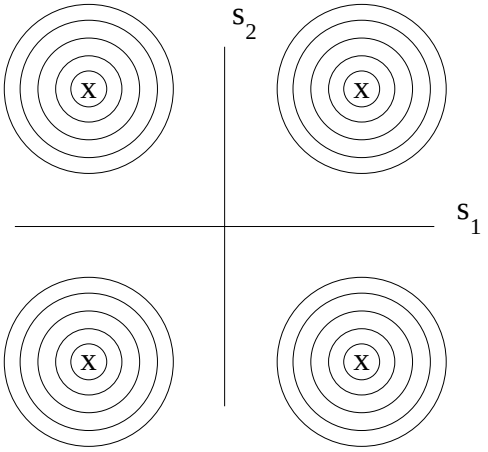


Figure 20: A contour plot of the correlator output vector PDF for $N = 2$.

OK, so in addressing the question "why correlate $r(t)$ with the $\phi_k(t)$?" we established the fact that, with the vector $\underline{r}$ of correlations between $r(t)$ and the $\phi_k(t)$ inner products, we can compute the inner products

$$\int_0^T r(t)\ s_m(t)\ dt\ =\ \underline{r}^H \underline{s}_m \qquad\qquad m = 1, 2, \cdots M\ \ . \tag{51}$$

We can not reconstruct $r(t)$ from $\underline{r}$, so clearly we loose something in going from $r(t)$ to $\underline{r}$. But what is it that we loose, and is it anything useful? Consider the following decomposition of $r(t)$:

$$r(t) = \hat{r}(t) + r^{'}(t)\ \ , \tag{52}$$

where

$$\hat{r}(t) = \underline{\phi}(t)\ \underline{r} \qquad\qquad 0 \le t \le T \tag{53}$$

is the projection of $r(t)$ onto the span of $\underline{\phi}(t)$, and $r^{'}(t)$ is what is left over. $\hat{r}(t)$ is the rank N approximation of $r(t)$ given the functions $\phi_k(t);\ k = 1, 2, \cdots, N$ which form a basis for the $s_m(t);\ m = 1, 2, \cdots, M$. If $r(t) = s_m(t)$, then $\hat{r}(t) = s_m(t)$. That is, there is no loss of signal – $\underline{r}$ contains all the information of the symbols. Also, $r^{'}(t)$ contains no symbol component.

Let $n_s(t) = \sum_{k=0}^{N} n_k\ \phi_k(t)$ denote the part of $n(t)$ in the span of the $\phi_k(t)$ and let $n^{'}(t) = r^{'}(t)$ be what is left over. Then

$$\begin{aligned} r(t) &= s_m(t) + n(t) \\ &= s_m(t) + n_s(t) + n^{'}(t)\ \ . \end{aligned} \tag{54}$$

Does $n^{'}(t)$ (i.e. $r^{'}(t)$) provide us with any information about the noise and/or signal in $\underline{r}$ which is useful? We have that

$$\begin{aligned} E\{n^{'}(t)\ r_k\} &= E\{n^{'}(t)\ s_{mk}\} + E\{n^{'}(t)\ n_k\} = E\{n^{'}(t)\ n_k\} \\ &= E\left\{ \left[n(t) - \sum_{j=0}^{N} n_j\ \phi_j(t) \right]\ n_k \right\} \end{aligned} \tag{55}$$

$$
\begin{aligned}
&= E\left\{n(t)\ n_k\right\} - E\left\{\sum_{j=0}^{N} n_j\ n_k\ \phi_j(t)\right\} \\
&= \int_0^T E\{n(t)\ n(\tau)\}\phi_k(\tau)\ d\tau - \sum_{j=0}^{N} E\{n_j\ n_k\}\ \phi_j(t) \\
&= \frac{1}{2}N_0\ \phi_k(t) - \frac{1}{2}N_0\ \phi_k(t) = 0 \quad .
\end{aligned}
$$

So, $r^{'}(t)$ and the r_k are uncorrelated, and since they are also Gaussian, they are statistically independent. This suggests that $r^{'}(t)$ is not useful. Later we will show that, in terms of symbol detection, this means that $\underline{r}$ is a sufficient statistic of $r(t)$. That is, an optimum detector based on $r(t)$ needs only $\underline{r}$.

For a transmitted signal composed of a superposition of nonoverlapping symbols, the received signal is

$$r(t) = \sum_n s_{m(n)}(t - nT) + n(t) \quad . \tag{56}$$

Given this, Figure 21 shows an implementation of the correlator for a ongoing sequence of symbols. $\tilde{\phi}_k(t) = \sum_n \phi_k(t - nT)$ is the periodic extension of $\phi_k(t)$, the symbols and $\tilde{\phi}_k(t)$ are assumed synchronized, the integrator integrates over the past T seconds, and the integrator output is sampled at $t = nT;\ n = \cdots, 0, 1, 2, \cdots$.

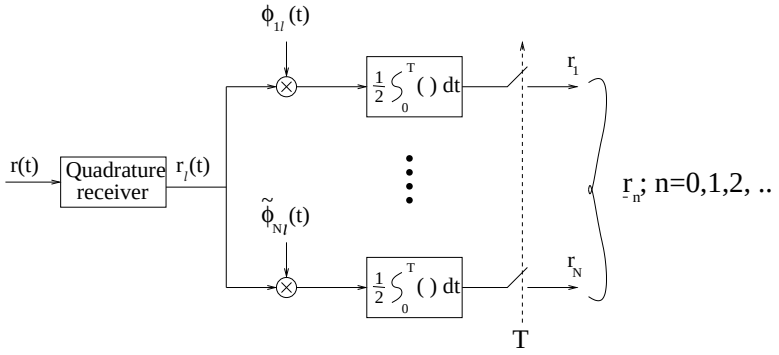


Figure 21: Correlator receiver for a sequence of symbols.

The Matched Filter

As illustrated in Figure 22(a), consider a linear filter operating on the received signal $r(t)$, followed by a sampler.

Let the filter be a matched filter – matched to the basis function $\phi_k(t)$. A matched filter maximizes the output SNR – in this case $\frac{s_{mk}^2}{E\{n_k^2\}} = \frac{\mathcal{E}_m}{E\{n_k^2\}}$. Its impulse response is

$$h_k(t) = \begin{cases} \phi_k(T-t) & 0 \le t \le T \\ 0 & otherwise \end{cases} \quad . \tag{57}$$

Then the matched filter output is

$$
\begin{aligned}
y_k(t) &= r(t) * h_k(t) = \int_{-\infty}^{\infty} r(\tau)\ h_k(t-\tau)\ d\tau \qquad (58) \\
&= \int_{t-T}^{t} r(\tau)\ \phi_k(T - t + \tau)\ d\tau \quad ,
\end{aligned}
$$

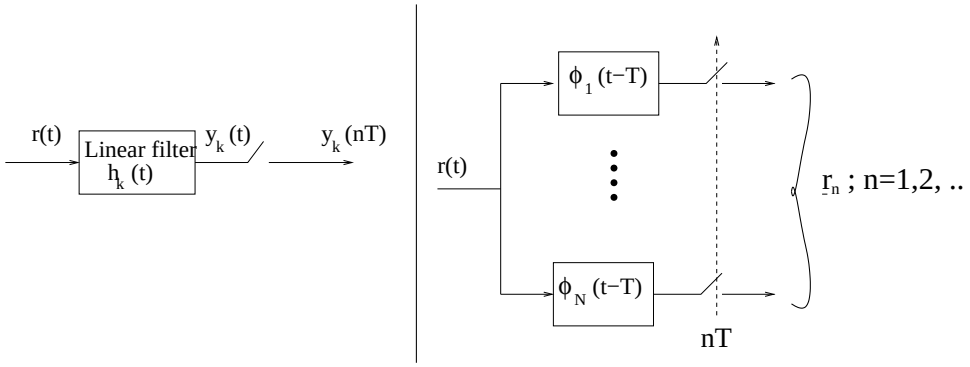


Figure 22: (a) Linear filter operating on $r(t)$, (b) the matched filter receiver.

and the output of the sampler is

$$\begin{aligned} y_k(nT) &= \int_{(n-1)T}^{nT} r(\tau)\ \phi_k(T - nT + \tau)\ d\tau \\ &= \int_{(n-1)T}^{nT} r(\tau)\ \tilde{\phi}_k(\tau)\ d\tau \quad , \end{aligned} \tag{59}$$

where as before, $\tilde{\phi}_k(t)$ is the periodic extension of $\phi_k(t)$. Referring to an earlier discussion, we see that the matched filter implements the correlator receiver.

Figure 22(b) shows the bandpass matched filter receiver for an N-dimensional modulation scheme where $s(t) = \sum_n s_{m(n)}(t - nT)$, and the $s_m(t)$ are limited to $0 \leq t \leq T$.

3.2.2 Optimum Symbol Detectors

Now we address the problem of optimum symbol detection. As noted earlier, by symbol detection we mean the decision as to which symbol was transmitted. We make the following assumptions: N-dimensional modulation scheme; noise is AWGN; there is no memory in the symbol sequence (see the assumptions listed at the start of Subsection 3.2).

We first consider the coherent Maximum Likelihood (ML) criterion of optimality, and resulting coherent ML detector, for carrier-phase synchronous reception. We then consider the coherent Maximum A Posterior (MAP) detector, which assures minimum Symbol Error Probability (SEP). The coherent ML detector is equivalent to the coherent MAP detector under an additional assumption. Finally we consider noncoherent detection.

Coherent Maximum Likelihood (ML) Detector

First we will consider the ML symbol detector considering the matched filters output vector $\underline{r}$ as the observed data. We will then show, by considering the received signal $r(t)$ as the observed data, that the optimum detection algorithm reduces to a matched filter preprocessor followed by the ML detector designed with $\underline{r}$ as the observation data (i.e. for this detection problem, $\underline{r}$ forms a sufficient statistic for $r(t)$).

ML Detector for Observation Data $\underline{r}$

Our starting point here is with $\underline{r}$. That is, given the sampled matched filter output, what

is the optimum decision rule? Consider the joint PDF of $\underline{r}$, conditioned on the transmitted symbol being $s_m(t)$:

$$p(\underline{r}/\underline{s}_m) = \frac{1}{(2\pi)^{N/2}(\sigma_n^2)^{N/2}} \; e^{-\sum_{k=1}^{N}(r_k - s_{mk})^2/2\sigma_n^2} \quad , \tag{60}$$

where $\sigma_n^2 = \frac{N_0}{2}$ is the noise power in each r_k. It is important to to stress that the joint conditional PDF $p(\underline{r}/\underline{s}_m)$ is a function of $\underline{r}$ where the elements of $\underline{s}_m$ are given parameters.

The ML detector [3] consists of the following two steps:

1. Plug the available data $\underline{r}$ into $p(\underline{r}/\underline{s}_m)$. Consider the result to be a function of $\underline{s}_m$, the symbol parameters to be detected. This function of $\underline{s}_m$ is called the *likelihood function.*
2. Determine the symbol $\underline{s}_m$ that maximizes the likelihood function. This symbol is the ML detection.

So, the ML detection problem statement is:

$$\max_{\underline{s}_m} \quad p(\underline{r}/\underline{s}_m) = \frac{1}{(2\pi)^{N/2}(\sigma_n^2)^{N/2}} \; e^{-\sum_{k=1}^{N}(r_k - s_{mk})^2/2\sigma_n^2} \quad . \tag{61}$$

Since the natural log function $\ln(\cdot)$ is monotonically increasing,

$$p(\underline{r}/\underline{s}_l) \; > \; p(\underline{r}/\underline{s}_k) \tag{62}$$

implies

$$\ln\{p(\underline{r}/\underline{s}_l)\} \; > \; \ln\{p(\underline{r}/\underline{s}_k)\} \quad . \tag{63}$$

So, an alternative form of the ML detector is:

$$\max_{\underline{s}_m} \quad \ln\{p(\underline{r}/\underline{s}_m)\} = -\frac{N}{2}\ln(2\pi\sigma_n^2) - \frac{1}{2\sigma_n^2}\sum_{k=1}^{N}(r_k - s_{mk})^2 \quad . \tag{64}$$

Taking the negative of this, and discarding terms that do not effect the optimization problem, we have the following equivalent problem:

$$\min_{\underline{s}_m} \quad \sum_{k=1}^{N}(r_k - s_{mk})^2 \; = \; ||\underline{r} - \underline{s}_m||^2 \quad . \tag{65}$$

This third form, basically the negative log likelihood function (absent a few non effecting terms), is the simplest to compute and therefore the one used.

Considering Eq 65, we see that the ML estimator for this problem is a "nearest neighbor" estimator. That is, the ML estimate is the symbol whose signal space representation is the nearest to the matched filter output vector (in the Euclidean norm sense). We term this Euclidean distance cost,

$$D(\underline{r}, \underline{s}_m) \; = \; ||\underline{r} - \underline{s}_m||^2 \quad , \tag{66}$$

the *distance metric.* Figure 23 illustrates the ML symbol detector.

[3] The ML method is a *parameter estimation* method. It is common in the signal processing community to refer it the objective as *estimation* when the parameters are continuous, and as *detection* when the parameters are discrete. Here, the parameters we wish to determine discrete (i.e. the symbols).

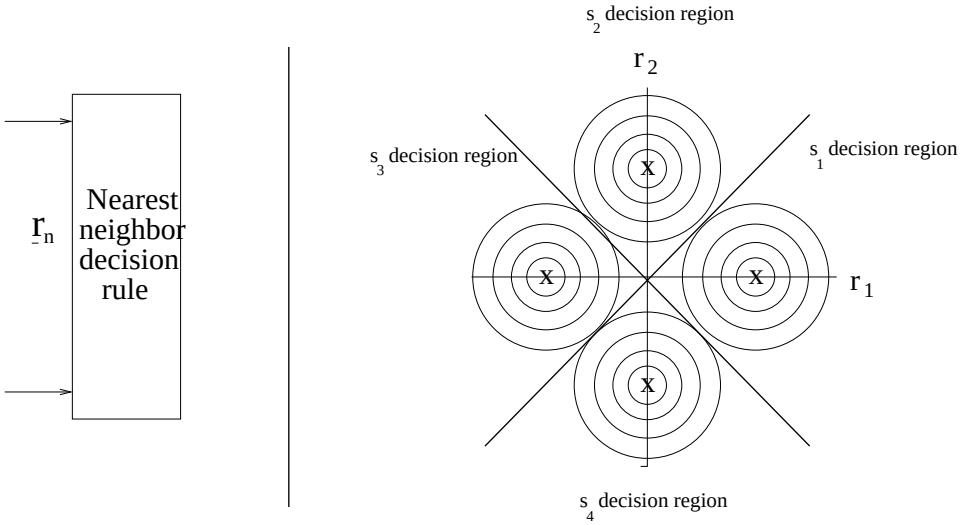


Figure 23: The ML detector starting with the sampled matched filter outputs as the observation data ($N = 2$): (a) block diagram, (b) signal space.

Detector for Observation Data $r(t)$

Now consider the receiver observation $r(t) = s_m(t) + n(t);\ 0 \le t \le T$. Let $\phi_k(t);\ 0 \le t \le T; k = 1, 2, \cdots, \infty$ be a set of orthonormal functions, with $\phi_k(t);\ k = 1, 2, \cdots, N$ being the basis functions for $s_m(t);\ m = 1, 2, \cdots, M$. Under realistic assumptions of $r(t)$,

$$E\left\{ |r(t) - \sum_{k=1}^{\infty} r_k\ \phi_k(t)|^2 \right\} = 0 \quad , \tag{67}$$

where

$$r_k =< r(t), \phi_k(t) >= \int_{-\infty}^{\infty} r(t)\ \phi_k(t)\ dt \quad . \tag{68}$$

So the coefficients r_k of this basis expansion are an equivalent representation of $r(t)$. In Figure 24 illustrate this orthonormal expansion representation of $r(t)$, where the coefficient vector $\underline{r}$ is infinite dimensional.

To derive an expression for $\underline{r} = [r_1, r_2, \cdots]^T$ in terms of the symbol and noise components, we have that

$$\begin{aligned} r_k &= \int_0^T (s_m(t)\ +\ n(t))\ \phi_k(t)\ dt \\ &= \begin{cases} s_{mk} + n_k & k = 1, 2, \cdots, N \\ n_k & otherwise \end{cases} , \end{aligned} \tag{69}$$

where $n_k =< n(t),\ \phi_k(t) >$. z_k is real-valued zero-mean Gaussian with PDF

$$p(z_k) = \frac{1}{\sqrt{2\pi\sigma_n^2}}\ \ e^{-(z_k)^2/2\sigma_n^2} \quad . \tag{70}$$

With this, we have

$$p(\underline{r}/\underline{s}_m) = \prod_{k=1}^{\infty} p(r_k/\underline{s}_m) = \prod_{k=1}^{N} p(r_k/\underline{s}_m) \prod_{k=N+1}^{\infty} p(r_k/\underline{s}_m) \quad . \tag{71}$$

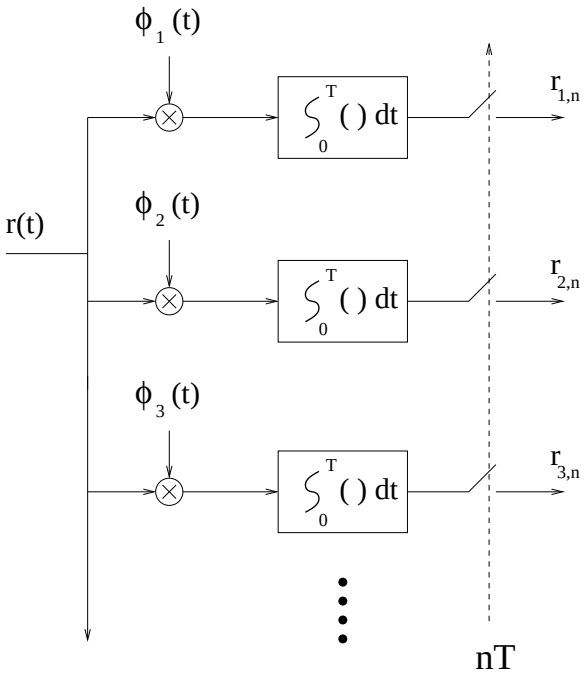


Figure 24: Orthogonal expansion representation of $r(t)$.

The $p(r_k/\underline{s}_m);\ k = N+1, N+2, \cdots, \infty$ are not a function of $\underline{s}_m$. Thus, letting $\underline{r}_N = [r_1, r_2, \cdots, r_N]^T$, the problem of ML detection starting with $r(t)$ reduces to that of maximizing $p(\underline{r}_N/\underline{s}_m)$, which is the likelihood function for the ML estimation problem starting with the vector of matched filter outputs. The Figure 25 illustrates the ML estimator based on $r(t);\ 0 \le t \le T$ as the starting data.

Coherent Maximum A Posterior (MAP) Detector

The MAP detector is based on maximizing the posterior PDF $P(\underline{s}_m/\underline{r})$ of $\underline{s}_m$ given $\underline{r}$. Using Bayes rule, we have

$$P(\underline{s}_m/\underline{r}) \ = \ \frac{p(\underline{r}/\underline{s}_m)P(\underline{s}_m)}{p(\underline{r})} \tag{72}$$

where $P(\underline{s}_m)$ is the probability of $\underline{s}_m$, and $p(\underline{r})$ is the joint PDF of $\underline{r}$. The MAP detector consists of the following two steps:

1. Plug the available data $\underline{r}$ into $p(\underline{s}_m/\underline{r})$. Consider the result to be a function of $\underline{s}_m$, the symbol parameters to be detected.

2. Determine the symbol $\underline{s}_m$ that maximizes Eq. 72. This symbol is the MAP detection.

Since the denominator term in Eq. 72 is independent of $\underline{s}_m$, the MAP detector can be stated as:

$$\max_{\underline{s}_m} \quad p(\underline{r}/\underline{s}_m)\ P(\underline{s}_m) \quad . \tag{73}$$

Comparing Eqs. 61 and 73, we see that the difference lies in the MAP detector's weighting of the likelihood function by the symbol probabilities $P(\underline{s}_m)$. *If the symbols are equally*

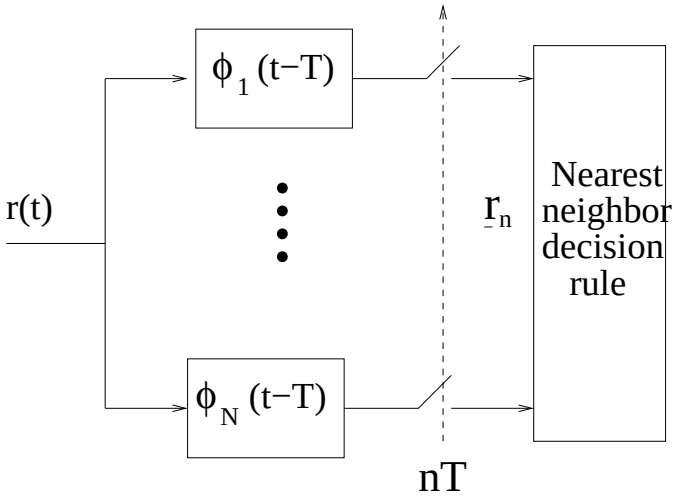


Figure 25: The ML detector starting with $r(t)$ as the observation data.

likely, then the ML and MAP detectors are equal. However, in general they are different. In terms of the primary objective of symbol detection, *the MAP estimator is optimum in that it minimizes symbol error probability.*

Noncoherent Detectors

This discussion parallels one in Section 4.5 of the Course Text. Consider a symbol waveform,

$$s_m(t) \;=\; Re\{s_{ml}(t)\; e^{j2\pi f_c t}\} \quad . \tag{74}$$

If this signal is received with an altered carrier phase (e.g. due to propagation delay), then the receiver input can be written as

$$r(t) \;=\; Re\{s_{ml}(t)\; e^{j(2\pi f_c t+\phi)}\} \;+\; n(t) \quad , \tag{75}$$

which has equivalent lowpass representation

$$r_l(t) \;=\; s_{ml}(t)\; e^{j\phi} \;+\; n_l(t) \quad . \tag{76}$$

If ϕ is known, then it can be incorporated into the receiver matched filters and optimum detection can be performed as described above. However, if ϕ is unknown, it is a random nuisance parameter that must be estimated or otherwise dealt with. This problem is referred to as the noncoherent detection problem. In this Subsection we address the issue of optimum noncoherent detection.

For the coherent receiver case, we have seen that the optimum detector consists of a bank of matched filters (matched to the modulation scheme basis functions), followed by symbol-rate samplers, followed by a decision rule (e.g. the nearest neighbor decision rule for ML detection) operating on the matched-filter sampled outputs which are sufficient statistics. Note that an equivalent alternative detector structure consists of a bank of matched filters (matched to the symbol waveforms) followed by symbol-rate samplers, followed by an equivalent decision rule. The former structure is usually preferred because it is less complex when $N < M$.

For the noncoherent receiver case, it can be shown that the optimum receiver has a similar matched-filter/symbol-rate-sampler/decision-rule structure. However the decision rule is different, and formulation in terms of filters matched to the symbol waveforms is more convenient. For the m^{th} symbol waveform, the lowpass equivalent matched filter has impulse response $s^*_{ml}(t-T)$. The optimum receiver structure is shown in Figure 26.

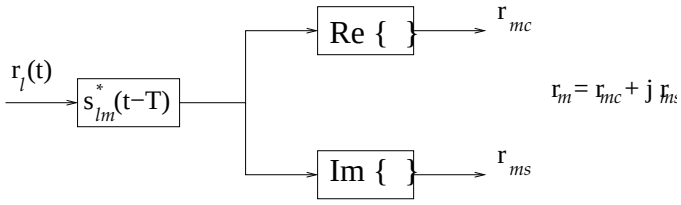


Figure 26: Optimum receiver structure for noncoherent detection.

Let $\underline{r}$ be the sampled matched filter output vector. Consider the MAP symbol detection problem

$$\max_{\underline{s}_m} \qquad P(\underline{s}_m/\underline{r}) \;=\; \frac{p(\underline{r}/\underline{s}_m)\; P(\underline{s}_m)}{p(\underline{r})} \tag{77}$$

(i.e. ML if the symbols are equally likely). For the random ϕ case under consider here, we need $p(\underline{r}/\underline{s}_m)$. Given $p(\underline{r}/\underline{s}_m, \phi)$, we can derive the required conditional density by marginalizing out ϕ as follows:

$$p(\underline{r}/\underline{s}_m) \;=\; \int_0^{2\pi} \; p(\underline{r}/\underline{s}_m, \phi)\; p(\phi)\; d\phi \quad , \tag{78}$$

where the range of integration reflects the maximum range of values for ϕ. Using this, the MAP estimator reduces to a likelihood ratio test. For equally likely symbols (or for ML detection), the optimum decision rule reduces to the *envelope detector*

$$\max_m \qquad \sqrt{r_{mc}^2 + r_{ms}^2} \quad , \tag{79}$$

or equivalently the *square-law detector*

$$\max_m \qquad r_{mc}^2 + r_{ms}^2 \quad . \tag{80}$$

3.3 Performance Analysis of Linear, Memoryless Modulation Schemes

Sections 4.1-4 of the Course Text describe analyses of various linear, memoryless modulation schemes. Here we consider several of these results. All but the last example considered assume coherent reception. At the end of this Subsection we will bring transmission bandwidth into the discussion of performance, overviewing digital communications bandwidth characteristics and commenting on the summary performance plot shown in Figure 4.6-1 of the Course Text.

We assume the symbols are equally likely, the noise is AWGN (additive, white, Gaussian noise), and that nearest neighbor (equivalently ML and MAP) detection is applied. The performance measures of interest are:

1. BER (bit error rate), denoted P_b (i.e. the bit error probability); and

2. SEP (symbol error probability), denoted P_e

as a function of SNR/bit. SNR/bit is defined as $\gamma_b = \frac{\mathcal{E}_b}{N_0} = \frac{\mathcal{E}_b}{2\sigma_n^2}$ where $\mathcal{E}_b$ is the average bit energy and, as before, N_0 is the AWGN lowpass spectral level. Note that for an $M = 2$ symbol modulation scheme, $P_b = P_e$. This is generally not true for $M > 2$. We will focus primarily of P_e since, compared to P_b, it more directly and thus more easily identified.

To understand the relationship between P_b and P_e, consider the 8-PSK constellation shown in Figure 27(a). Consider a nearest-neighbor symbol error, since as noted before this type is the most likely to occur. Consider transmission of symbol $\underline{s}_1$ and reception of symbol $\underline{s}_2$. The corresponding probability, call it $P(2/1)$, contributes to the SEP P_e. For example, if all bits represented by $\underline{s}_2$ are different from those represented by $\underline{s}_1$ (e.g. $\underline{s}_1 = (000)$ and $\underline{s}_2 = (111)$), then the contribution to P_b and P_e will be the same. Otherwise, the contribution to P_b will be less. On the other hand, if $\underline{s}_1$ and $\underline{s}_2$ differ by only one bit (e.g. $\underline{s}_1 = (000)$ and $\underline{s}_2 = (001)$), then the contribution to P_b will be $\frac{1}{k}P_e$, where $k = \log_2 M$ is the number of bits per symbol. Figure 27(b) shows a Gray code labeling of the 8-PSK constellation which is efficient in that all nearest-neighbor symbol pairs differ by only one bit.

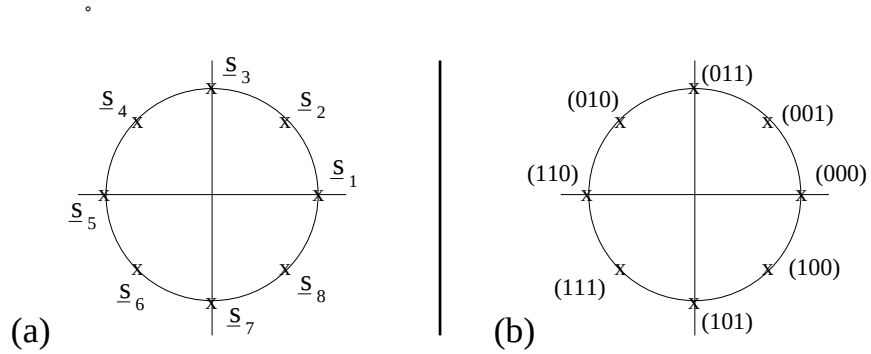


Figure 27: (a) the 8-PSK constellation; and (b) a Gray code bit mapping.

The point is that the P_b will be more difficult to determine, being dependent on the symbol to bit assignments. Also, generalizing the 8-PSK example above, we can conclude that

$$P_b \;\leq\; P_e \;\leq\; k\,P_b \quad . \tag{81}$$

3.3.1 Binary PSK

Here we consider the performance of 2-PSK ($M = 2$, $N = 1$; the same as binary PAM) with a coherent receiver. This is covered in Subsection 4.2, pp. 173-4 of the Course Text. This modulation scheme is also referred to as *antipodal signaling*, since the two symbol waveforms (and signal space representations) are negatives of one another. Figure 28 illustrates the PDF's conditioned on the two symbols, where $x = r$ in the correlation receiver output statistic. In terms of the bit energy $\mathcal{E}_b$, the signal space representations are $s_0 = -\sqrt{\mathcal{E}_b}$ (i.e. H0) and $s_1 = \sqrt{\mathcal{E}_b}$ (i.e. H1).

Performance:

The SEP and BER are

$$P_e \;=\; P_b \;=\; Q\left(\sqrt{2\gamma_b}\right) \quad . \tag{82}$$

Derivation:

The probability of error given symbol s_i is

$$P(e/s_1) \;=\; P(e/s_0) \;=\; \int_0^\infty p(x/s_0)\,dx \;= Q\left(\frac{\sqrt{\mathcal{E}_b}}{\sigma_n}\right) \tag{83}$$

where $\sigma_n^2 = (N_0/2)$. By total probability,

$$P_e \;=\; P(s_0)\,P(e/s_0) \;+\; P(s_1)\,P(e/s_1) \quad . \tag{84}$$

Under the equiprobable symbol assumption we have Eq (82).

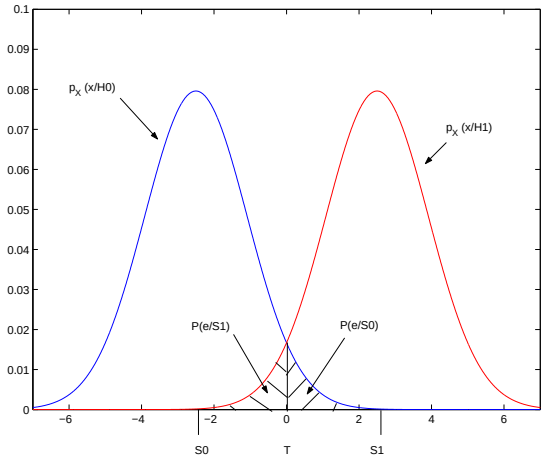


Figure 28: The receiver statistic ($r = x$) conditional PDF's. For ML, $T = 0$.

3.3.2 Binary Orthogonal Modulation

Binary orthogonal modulation is a $M = 2$, $N = 2$ scheme. Each symbol is represented by its own orthonormal basis waveform. The symbols have equal energy. The signal space representations are then $\underline{s}_1 = [\sqrt{\mathcal{E}_b}, 0]^T$ and $\underline{s}_2 = [0, \sqrt{\mathcal{E}_b}]^T$ as illustrated in Figure 29. The noises added onto r_1 and r_2 are mutually uncorrelated, each with variance σ_n^2. Under the coherent receiver assumption, performance analysis is presented on p. 176 of the Course Text, as a special case of more general equiprobable binary signaling scheme described and analyzed on pp. 174-5.

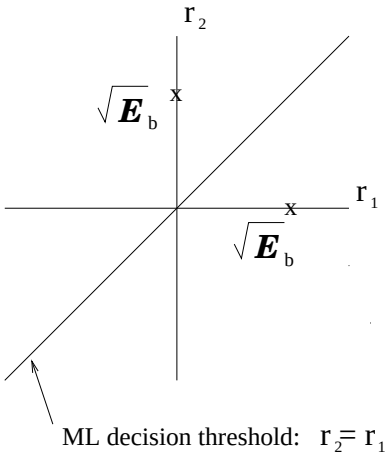


Figure 29: Signal space representation for binary orthogonal modulation.

Performance:

The SEP and BER are

$$P_e \;=\; P_b \;=\; Q\left(\sqrt{\gamma_b}\right) \quad . \tag{85}$$

Compared to binary PSK, twice the SNR/bit is needed for the same BER.

Derivation 1:

Figure 29 shows that the ML decision rule can be implemented by comparing r_1 to r_2, deciding on s_1 if $r_1 > r_2$ (and s_2 if $r_2 > r_1$). Equivalently we can compare the statistic $r = r_2 - r_1$ to the threshold $T = 0$. The noise variance for r is twice that of r_1 or r_2 (i.e. the variance of the sum is the sum of the variances for uncorrelated random variables), whereas the signal levels are same. The conditional PDF are the same as those in Figure 28 except the noise variance is doubled. Thus,

$$P_e \;=\; Q\left(\frac{\sqrt{\mathcal{E}_b}}{\sqrt{2\sigma_n^2}}\right) \;=\; Q\left(\sqrt{\frac{\mathcal{E}_b}{2\sigma_n^2}}\right) \;=\; Q\left(\sqrt{\gamma_b}\right) \quad . \tag{86}$$

Derivation 2:

This follows the general M orthogonal modulation performance analysis on pp. 204-5 of the Course Text, for $M = 2$. First note that $P_e = 1 - P_c$ where P_c is the probability of the correct detection of a symbol. From Figure 29,

$$P_c \;=\; P(r_1 > r_2/s_1) \;=\; \int_{-\infty}^{\infty}\int_{-\infty}^{r_1} p(r_1, r_2/s_1)\; dr_2\; dr_1 \quad , \tag{87}$$

where $p(r_1, r_2/s_1)$ is joint uncorrelated Gaussian, i.e.

$$p(r_1, r_2/s_1) \;=\; \mathcal{N}_{r_2}(0, \sigma_n^2)\; \mathcal{N}_{r_1}(\sqrt{\mathcal{E}_b}, \sigma_n^2) \quad . \tag{88}$$

So

$$P_c \;=\; \int_{-\infty}^{\infty} \frac{1}{\sqrt{2\pi\sigma_n^2}}\; e^{-(r_1-\sqrt{\mathcal{E}_b}\;)^2/2\sigma_n^2} \left\{ \int_{-\infty}^{r_1} \frac{1}{\sqrt{2\pi\sigma_n^2}}\; e^{-(r_2)^2/2\sigma_n^2}\; dr_2 \right\}\; dr_1 \tag{89}$$

$$= \int_{-\infty}^{\infty} \frac{1}{\sqrt{2\pi\sigma_n^2}}\; e^{-(r_1-\sqrt{\mathcal{E}_b}\;)^2/2\sigma_n^2} \left\{ 1 \;-\; Q\left(\sqrt{\frac{r_1}{\sigma_n}} \right) \right\}\; dr_1 \tag{90}$$

$$= \int_{-\infty}^{\infty} \frac{1}{\sqrt{2\pi}}\; e^{-(y-\sqrt{\mathcal{E}_b/\sigma_n^2}\;)^2/2} \left\{ 1 \;-\; Q\left(y\right) \right\}\; dy \tag{91}$$

$$= 1 \;-\; \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\infty} Q\left(y\right)\; e^{-(y-\sqrt{\mathcal{E}_b/\sigma_n^2}\;)^2/2}\; dy \quad . \tag{92}$$

For the next to last equation we let $y = \frac{r_1}{\sigma_n}$. Thus,

$$P_e \;=\; 1 - P_c \;=\; \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\infty} Q\left(y\right)\; e^{-(y-\sqrt{\mathcal{E}_b/\sigma_n^2}\;)^2/2}\; dy \quad . \tag{93}$$

3.3.3 M-ary Orthogonal Modulation

This is a generalization of binary orthogonal modulation, for general $M = N$. Again, each symbol is represented by its own orthonormal basis waveform, and The symbols have equal energy. M-ary orthogonal FSK is one example. Assuming a coherent receiver, SEP and BER equations are presented of p. 205 of the Course Text. You will explore analysis of this modulation scheme in more detail as a problem in Homework #1. This analysis is a generalization of that presented above for binary orthogonal modulation. The signal space representation of the 1^{st} symbol is

$$\underline{s}_1 \;=\; [\sqrt{\mathcal{E}}, 0, 0, \cdots, 0] \quad , \tag{94}$$

where $\mathcal{E}$ is the symbol energy, so that the energy/bit is $\mathcal{E}_b = \frac{\mathcal{E}}{k}$ where $k = \log_2(M)$. The representations of other symbols are defined as the obvious extension of this. Then we have that $\gamma_b = \frac{\mathcal{E}_b}{N_0} = \frac{\mathcal{E}}{2k\sigma_n^2}$. The BER is,

$$P_b \;=\; \frac{2^{k-1}}{(2^k-1)}\; P_e$$

$$= \frac{2^{k-1}}{(2^k-1)\sqrt{2\pi}} \int_{-\infty}^{\infty} \left[1 - \left(\int_{-\infty}^{x} e^{-y^2/2} dy \right)^{M-1} \right] e^{-\frac{1}{2}\left(x-\sqrt{2k\gamma_b}\right)^2} dx \quad . \tag{95}$$

We defer discussion on BER to Subsection 3.3.7 on noncoherent orthogonal FSK.

3.3.4 *M*-ary PSK

Analysis of M-ary PSK for a coherent receiver is presented in pp. 190-5 of the Course Text. For this modulation scheme, the signal space representations are

$$\underline{s}_m = \left[\sqrt{\mathcal{E}}\cos\left(2\pi(m-1)/M\right)\ ,\ \sqrt{\mathcal{E}}\sin\left(2\pi(m-1)/M\right)\right]^T\ ;\quad m=1,2,\cdots,M \tag{96}$$

where $\mathcal{E}$ is the symbol energy. The symbol error probability is

$$P_e\ =\ 1\ -\ \int_{-\pi/M}^{\pi/M}\ p_\Theta(\theta)\ d\theta\quad , \tag{97}$$

where Θ is the observation signal space representation vector phase, under the $m = 1$ assumption, which has PDF

$$p_\Theta(\theta)\ = \frac{1}{2\pi}e^{-k\gamma_b\sin^2\theta}\ \int_0^\infty v\ e^{-(v-\sqrt{2k\gamma_b}\cos\theta)^2/2}\ dv\quad , \tag{98}$$

where V is the vector magnitude random variable and γ_b is SNR/bit. For $M = 2$, P_e reduces to the 2-PSK equation derived earlier. For $M = 4$, it can be shown that

$$P_e\ =\ 2\ Q\left(\sqrt{2\gamma_b}\right)\ \left[1\ -\ \frac{1}{2}\ Q\left(\sqrt{2\gamma_b}\right)\right]\quad . \tag{99}$$

For $M > 4$, P_e can obtained by evaluating Eq (97) numerically. The approximation,

$$P_e\ \approx\ 2\ q\left(\sqrt{2k\gamma_b}\ \sin(\pi/M)\right)\quad , \tag{100}$$

is derived in the Course Text on p. 194. As pointed out on p. 195, fore Gray code bit to symbol mapping,

$$P_b\ \approx\ \frac{1}{k}\ P_e\quad . \tag{101}$$

3.3.5 *M*-ary PAM

Performance for this $N = 1$ dimensional modulation scheme, with coherent reception, is presented on pp. 188-90 of the For this modulation scheme, the signal space representations are

$$s_m\ =\ (2m-1-M)\ d\ \sqrt{\frac{\mathcal{E}_g}{2}}\ ;\qquad m=1,2,\cdots,M\quad , \tag{102}$$

and the average energy/symbol is

$$\mathcal{E}_{av}\ =\ \frac{1}{6}\ (M^2-1)\ d^2\ \mathcal{E}_g\quad . \tag{103}$$

The average probability of symbol error is

$$P_M\ =\ \frac{2(M-1)}{M}\ Q\left(\sqrt{\frac{6\ k\ \gamma_{b,av}}{M^2-1}}\right) \tag{104}$$

where $\gamma_{b,av} = \frac{\mathcal{E}_{b,av}}{N_0}$ and $\mathcal{E}_{b,av} = \frac{\mathcal{E}_{av}}{k}$. Note that BER is not given since, unlike the M-ary orthogonal modulation case, it is a complicated calculation which depends on how the bits values are assigned to the different symbols.

3.3.6 M-ary QAM

Performance of QAM with coherent reception is considered in Subsection 4.3-3 of the Course Text. For this modulation scheme, the signal space representations are

$$\underline{s}_m = \left[\sqrt{\frac{\mathcal{E}_g}{2}} \;\; V_m \cos\theta_m \;\; , \;\; \sqrt{\frac{\mathcal{E}_g}{2}} \;\; V_m \sin\theta_m \right]^T \quad . \tag{105}$$

If the constellation of symbol points is on square grid, and if k is even (i.e. for 4-QAM, 16-QAM, 64-QAM ...), then QAM can be interpreted as two $\sqrt{M}$-ary PAM modulations, one on the in-phase basis and the other on the quadrature. For correct detection, both in-phase and quadrature must be detected correctly. So the symbol error probability is

$$P_e \;=\; 1 \;-\; (1 - P_{e,\sqrt{M}})^2 \quad , \tag{106}$$

where $P_{e,\sqrt{M}}$ is the SEP for $\sqrt{M}$-ary PAM, i.e. from Eq 104,

$$P_{e,\sqrt{M}} \;=\; \frac{2(\sqrt{M}-1)}{\sqrt{M}} \; Q\left(\sqrt{\frac{3k\gamma_{b,av}}{M-1}} \right) \quad , \tag{107}$$

and $\gamma_{b,av}$ is the average SNR/bit. As with M-PAM, in general for QAM it is difficult determine an expression for P_b.

3.3.7 M-ary Orthogonal FSK Modulation

Now we consider *noncoherent* reception of M-ary orthogonal FSK. For coherent reception, we have already considered this modulation scheme in Subsection 3.3.3. Assume M equiprobable, equal energy orthogonal FSK symbols. In pp. 216-218 of the Course Text, symbol error probability is shown to be

$$P_e \;=\; \sum_{n=1}^{M-1} (-1)^{n+1} \left(\begin{array}{c} M-1 \\ n \end{array} \right) \frac{1}{n+1} \;\; e^{-\frac{nk\gamma_b}{n+1}} \quad . \tag{108}$$

Concerning BER, first let $P(i/j)$ denote the probability of deciding symbol i given symbol j was transmitted. Note that with orthogonal modulation, all $p(i/j);\; i \neq j$ are equal. Thus,

$$P(i/j) \;=\; \frac{P_e}{M-1} \;=\; \frac{P_e}{2^k - 1} \qquad i \neq j \quad . \tag{109}$$

For any bit represented by any transmitted symbol, there are 2^{k-1} other symbols that, if incorrectly detected, will result in an error in that bit. Orthogonal symbol errors events are independent, the probability of error of that bit is the sum of the individual probabilities of events resulting in that bit being in error. So, for equally probable bits,

$$P_b \;=\; 2^{k-1} \; \frac{P_e}{2^k - 1} \;\approx\; \frac{1}{2} \; P_e \quad . \tag{110}$$

3.3.8 Examples of Performance Analysis

Example 3.1: For M-ary orthogonal modulation, determine the SNR/bit required to achieve BER $= 10^{-4}$ for $M = 2$ and $M = 64$. Then, Using the union/Chernov bound on P_e derived in the Course Text on pp. 206-7 and considered as a problem in Homework 1, determine a bound on SNR/bit, γ_b, that assures $P_e \longrightarrow 0$ as $M \longrightarrow \infty$.

Solution: Using the orthogonal modulation P_e vs. γ_b plot in the Coruse Text, Figure 4.4-1 on p. 206, we get that, to achieve BER $= 10^{-4}$ for $M = 2$ we need $\gamma_b = 11dB$. For $M = 64$ we need $\gamma_b = 5.8dB$. With $M = 64$ we save $5.2dB$ in SNR/bit to achieve the same level of performance. Of course, in this case the price paid would be a significantly larger bandwidth (e.g. $M = 2$ FSK vs. $M = 64$ FSK). Considering the union/Chernov bound

$$P_e \;<\; e^{-k(\gamma_b \;-\; 2\ln 2)/2} \tag{111}$$

from the Course Text, note that as $k \longrightarrow \infty$ (i.e $M \longrightarrow \infty$), $P_e \longrightarrow 0$ as long as $\gamma_b > 2\ln 2 = 1.42dB$. In words, we can assure reliable communications (arbitrarily low P_e) using orthogonal symbols, as long as the SNR/bit is greater than $1.42dB$ (and assuming we are willing to use a lot of orthogonal symbols). This leads to two important questions: 1) Is this bound tight, or can we achieve reliable communications at lower SNR/bit? and 2) Can we achieve reliable communications at this SNR/bit level, or better, without having to resort to large numbers of orthogonal symbols? These questions have motivated extensive research over the past 60 years. As we shall see later in the Course, the answer to the first question is that this bound is not very tight. We will also see that the answer to the second question is yes, there are more practical approaches to achieving performance close to even tighter performance bounds.

Example 3.2: Figure 30 shows performance curves for digital several modulation schemes with ML symbol detection and coherent reception. These plots, of symbol error probability vs. SNR/bit, were generated using the performance equations presented in this Subsection. Comparing binary PAM with binary orthogonal symbols, binary PAM performs $\gamma_b = 3dB$ better for any level of performance. Also, for SEP at moderate (e.g. 10^{-3}) to very good (e.g. $<\; 10^{-6}$) levels, 8-PAM requires about $8dB$ more SNR/bit.

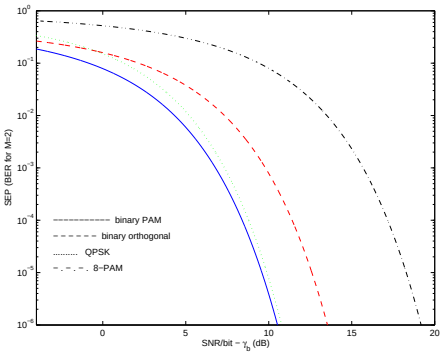


Figure 30: Performance curves for several modulation schemes.

3.3.9 Power Spectral Density (PSD) of Digitally Modulated Signals

This Subsection of the Notes corresponds to topics in Section 3.4 Course Text. We consider the frequency characteristics of digitally modulated signals. This is a critically important issue in most digital communications applications because of the need to efficiently utilize limited channel bandwidth. Concerning our specific interests, certain classes of channel coding schemes are specifically designed to utilize bandwidth efficiently. Here we will restrict the discussion to linear modulation schemes – in particular PAM, PSK and QAM.

Consider the following representation of symbols

$$s_m(t) = Re\{s_{ml}(t)\ e^{j2\pi f_c t}\} \quad , \tag{112}$$

where $s_{ml}(t)$ is the equivalent lowpass representation. Let, $s_{ml}(t) = I_n\ g(t)$, where I_n is a complex value representing the symbol m at time n. For PAM, PSK and QAM we have, respectively, I_n is of the form:

$$\begin{aligned} I_n &= A_m \\ I_n &= e^{j2\pi(m-1)/M} \\ I_n &= V_m\ e^{j\theta_m} \quad . \end{aligned} \tag{113}$$

Using this equivalent lowpass representation, the transmitted signal is

$$s(t) = Re\{v(t)\ e^{j2\pi f_c t}\} \tag{114}$$

where $v(t)$ is the equivalent lowpass representation of $s(t)$:

$$s_l(t) \ = \ v(t) \ = \ \sum_n I_n\ g(t - nT) \quad . \tag{115}$$

The transmitted signal $s(t)$ is a random process since the I_n are random. It can be shown that $v(t)$ and therefore $s(t)$ are not wide-sense stationary. This is because of the within-symbol structure (i.e. the $g(t)$ structure). However, if I_n is wide-sense stationary, which we assume it is, then $v(t)$ and therefore $s(t)$ are *cyclostationary*, and we can identify their 2^{nd} order statistical characterizations.

We know, from our previous discussion of equivalent lowpass signals, that the spectral characteristics of $s(t)$ can be determined from those of $v(t)$, e.g. for wise-sense stationary $s(t)$ the power density spectrum relationship is

$$S_{ss}(\omega) = \frac{1}{2}\left[S_{vv}(\omega - \omega_c) + S_{vv}(-\omega - \omega_c)\right] \quad . \tag{116}$$

We will proceed to characterize the frequency characteristics of $v(t)$ and then deduce those of $s(t)$.

Let m_i be the mean of I_n. Then,

$$E\{v(t)\} = m_i \sum_{n=-\infty}^{\infty} g(t-nT) \quad . \tag{117}$$

So, the mean is periodic with period T, and $E\{v(t)\} = 0$ if $m_i = 0$.

By definition, a cyclostationary signal has a mean and autocorrelation function that are periodic in t with some period T. The autocorrelation function of the equivalent lowpass signal $v(t)$ is defined as

$$R_{vv}(t, t-\tau) \;=\; E\{v^*(t)v(t-\tau)\} \tag{118}$$

Plugging in $v(t) = \sum_{n=-\infty}^{\infty} I_n g(t-nT)$, and letting

$$R_{ii}(m) = E\{I_n^* I_{n-m}\} \tag{119}$$

denote the discrete-time autocorrelation function of wide-sense stationary I_n, we get an expression for $R_{vv}(t, t-\tau)$. In this case, $R_{vv}(t, t-\tau)$ has period T.

So $v(t)$ is cyclostationary. For such a signal, it makes sense to define a time averaged autocorrelation function

$$\overline{R}_{vv}(\tau) = \frac{1}{T} \int_{-T/2}^{T/2} R_{vv}(t, t-\tau)\; dt \quad . \tag{120}$$

Defining

$$R_{gg}(\tau) = \int_{-\infty}^{\infty} g^*(t)\; g(t-\tau)\; dt \quad , \tag{121}$$

we have that

$$\overline{R}_{vv}(\tau) = \frac{1}{T} \sum_{m=-\infty}^{\infty} R_{ii}(m)\; R_{gg}(\tau - mT) \;=\; R_{ii}^c(\tau) * R_{gg}(\tau) \quad , \tag{122}$$

where $R_{ii}^c(\tau) = \frac{1}{T} \sum_{n=-\infty}^{\infty} R_{ii}(n)\; \delta(\tau - nT)$.

Define $S_{vv}(\omega)$, the continuous-time Fourier transform (CTFT) of $\overline{R}_{vv}(\tau)$, as the "average PSD". Then, from the convolution property of Fourier transforms,

$$S_{vv}(\omega) = S_{ii}^c(\omega)\; S_{gg}(\omega) \tag{123}$$

where the two terms on the right are the CTFT's of the respective autocorrelation functions. $S_{ii}^c(\omega)$ is periodic with period $\frac{1}{T}$ since $R_{ii}^c(\tau)$ consists of impulses at integer multiples of T. You may recall for studying sampling theory that $S_{ii}^c(\omega) = \frac{1}{T} S_{ii}(\omega)$ where $S_{ii}(\omega) = DTFT\{R_{ii}(m)\}$, where $DTFT$ stands for discrete-time Fourier transform. From the definition of $R_{gg}(\tau)$, $S_{gg}(\omega) = |G(\omega)|^2$, the magnitude-squared of the CTFT of $g(t)$. So,

$$S_{vv}(\omega) = \frac{1}{T}\; |G(\omega)|^2\; S_{ii}(\omega) \quad . \tag{124}$$

Assume, for example, that the I_n are uncorrelated sample-to-sample so that

$$R_{ii}(m) = \begin{cases} \sigma_i^2 + m_i^2 & m = 0 \\ m_i^2 & m \neq 0 \end{cases} \quad . \tag{125}$$

Then the PSD of I_n is

$$S_{ii}(\omega) = \sigma_i^2 + m_i^2 \sum_{m=-\infty}^{\infty} e^{-j\omega mT} = \sigma_i^2 + \frac{m_i^2}{T} \sum_{m=-\infty}^{\infty} \delta(\omega - \frac{2\pi m}{T}) \quad . \tag{126}$$

Then, from Eq. 124,

$$S_{vv}(\omega) = \frac{\sigma_i^2}{T} \, |G(\omega)|^2 \; + \; \frac{m_i^2}{T^2} \sum_{m=-\infty}^{\infty} \left| G(2\pi \frac{m}{T}) \right|^2 \, \delta(\omega - 2\pi \frac{m}{T}) \quad . \tag{127}$$

Form this derivation, we observe the following:

1. We can use the pulse shaping waveform $g(t)$ to control the spectrum $S_{vv}(\omega)$ and therefore of
$$S_{ss}(\omega) = \frac{1}{2} \left[S_{vv}(\omega - \omega_c) + S_{vv}(-\omega - \omega_c) \right] \quad . \tag{128}$$
2. We can use correlation in I_n, i.e. memory in the generation of the I_n sequence, to control the spectrum of $s(t)$.
3. We want I_n to be zero-mean so there are no impulses in $S_{ii}(\omega)$ at integer multiples of $\frac{2\pi}{T}$.

If I_n is zero-mean, then the bandwidth of $s(t)$ is the two-sided bandwidth of $g(t)$. In Example 3.4-1 (p. 134) and in Figure 9.2-7 (p. 608) of the Course Text, the authors illustrate $|G(\omega)|^2$ for two common pulse shapes, the rectangular and raised-cosine pulses respectively. Notice that the zero-crossing bandwidths of both pulses are proportional to $\frac{1}{T}$, with the raised-cosine pulse having twice the zero-crossing bandwidth and lower "side lobe" levels.

Note that for linear modulation the bandwidth of $s(t)$ is not effected by the number of symbol levels.

3.3.10 A Performance/SNR/Bandwidth Comparison of Modulation Schemes

In the selection of channel codes to control symbol errors, bandwidth and power requirements are important considerations. For a given channel noise level, the power requirement is equivalent to an SNR requirement. SNR and bandwidth requirements differ for different modulation schemes.

In Subsection 3.3.1-8 we summarized symbol and bit error rates vs. SNR for several linear, memoryless modulation schemes that we will consider when evaluating channel coding strategies. In Subsection 3.3.9 we developed a foundation from which bandwidth characteristics of different modulation schemes can be derived. Some useful approximate bandwidth requirements are stated in Subsection 4.6 of Course Text, and summarized in the table below. W is the approximate bandwidth, in Hz., and R is the bit rate.

Modulation	Bandwidth W	Bite Rate R	R/W
PAM (SSB)	$\frac{1}{2T}$	$\frac{1}{T}k = \frac{1}{T}\log_2 M$	$2\ \log_2 M$
PSK	$\frac{1}{T}$	$\frac{1}{T}k = \frac{1}{T}\log_2 M$	$\log_2 M$
QAM	$\frac{1}{T}$	$\frac{1}{T}k = \frac{1}{T}\log_2 M$	$\log_2 M$
FSK	$\frac{M}{2T}$	$\frac{1}{T}k = \frac{1}{T}\log_2 M$	$\frac{2\ \log_2 M}{M}$

Table 3.1: Approximate Bandwidth Requirements for Different Modulation Schemes.

A performance quantity of principal concern in digital communication systems is bandwidth efficiency, which is the rate-to-bandwidth ratio

$$\frac{R}{W} \tag{129}$$

with units (bits/sec./Hz.). Bandwidth efficiency tell us how many bits per second we can push through the system per Hertz of system bandwidth. Figure 4.6-1 of the Course Text (reproduced below as Figure 31) compares, for a symbol error rate of 10^{-5}, efficiency for some of the modulation schemes we've considered. The channel capacity bound and its asymptotic value are topics of the next Section of the course. The relevance of the "bandwidth-limited region" $\frac{R}{W} > 1$ and the "power-limited region" $\frac{R}{W} < 1$ will become more clear when we discuss trellis coded modulation later in the course.

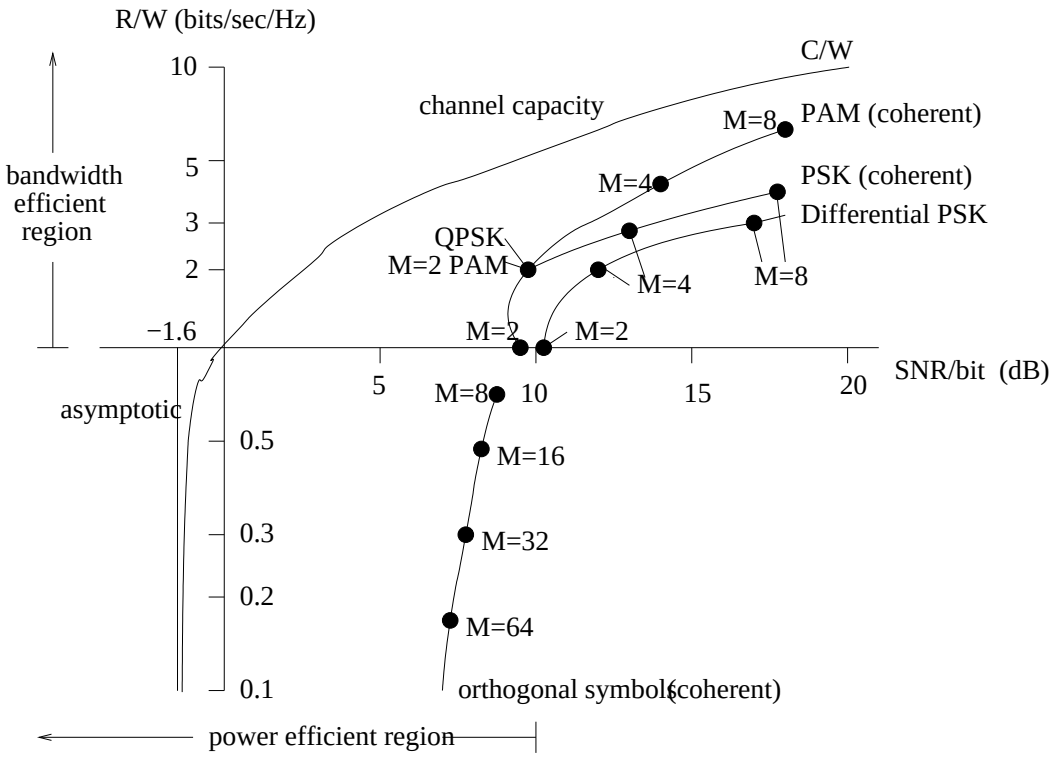


Figure 31: Comparison of SNR and bandwidth characteristics of several modulation schemes at SEP $= 10^{-5}$.

4 Information Theory - an Overview

According to *Webster's New Twentieth Century Unabridged Dictionary*, information is

> *1. an informing or being informed; especially a telling or being told something. 2. something told; news; intelligence; word. 3. knowledge acquired in any manner; facts; data; learning; lore...*

The term information suggests both an object and its conveyance. We are interested in systems which convey objects of information. Though such systems have been around for a long time, as indicated in Section 1 of the Course it was not until 1948 that Shannon introduced information theory, providing the foundation for quantitatively considering information in the context of engineering systems. This theory quantifies the object conveyed (e.g. as bits), and established bounds guiding its conveyance.

For a formal treatment of information theory and digital communication, we desire a measure of information. Qualitatively, information is the representation/conveyance of the previously or otherwise unknown. It replaces uncertainty with certainty. So, concerning communication signals, an information measure should be a measure of the randomness of a random variable or process which is replaced with certainty when a realization of the random variable or process is observed. A quantitative measure of the information in a signal should therefore indicate how much uncertainty that signal replaces when it is observed. This idea leads to the concept and measures of *entropy*.

In information theory, entropy is the measure of average uncertainty. It is our measure of the average information in realizations of random variables or processes. Reflecting on our background in probability, it is common when introducing probability to discuss foundations from which it can be built. For example, relative frequency of sets of outcomes of a random experiment is one foundation. A more powerful foundation is the set of three axioms of probability (e.g. Peebles, [3], p. 12). Although an axiomatic basis for entropy has been developed, it is practical and usually adequate to simply define entropy and built from there. This is what we do in this Section.

In this Section of the Course we expand on Sections 6.1-2 of the Course Text. A good reference on Information Theory is the Cover & Thomas book [4], which was the primary reference used in writing this Section. In this Section of the Course we cover the most basic concepts of information theory. In sequence, we consider the following measures of information and average information:

1. for a single discrete (valued) source, modeled as a random variable with discrete values termed symbols, *entropy* – the average *self-information* of the different symbol values;
2. for a single continuous (valued) random variable, a useful measure *differential entropy*;
3. for two random variables, the principle measure *average mutual information* – the average of the *mutual information* between all pairs of symbols from the two random variables; and additionally *joint entropy, conditional self-entropy*, and *conditional entropy*.
4. for multiple random variables, *mutual information & joint entropy*.
5. for stationary random processes, *entropy rate*.

4.1 A Single Random Variable

In this Subsection we consider the entropy of a random variable and the information of its realizations. We look at discrete-valued random variables first, and then continuous-valued.

4.1.1 A Discrete-Valued Random Variable

We will start with the definition established in information theory for the entropy of a discrete random variable.

> Let X be a discrete-valued random variable that can take on symbol values $\{x_i;\ i = 1, 2, \cdots, L\}$ with corresponding probabilities $\{p_i = P(x_i);\ i = 1, 2, \cdots, L\}$. The entropy of X is defined as
>
> $$\begin{aligned} H(X) &= -\sum_{i=1}^{L} p_i \log\left(p_i\right) \\ &= \sum_{i=1}^{L} p_i \log\left(\frac{1}{p_i}\right) \quad . \end{aligned} \tag{1}$$

See [4], Chapter 2 for a more in-depth discussion of this definition.

In Eq (1), the log is usually taken to be base 2, in which case entropy is in *bits*. For base 10 and base e, respectively, entropy is in *Hartleys* and *nats*. In this Course, unless otherwise indicated, within the context of entropy "log" will denote $\log_2$ and we will talk about "bits" of information.

> *Example 4.1:* Consider a single discrete-valued variable X of L equally likely values. The entropy, in bits, is
>
> $$H(X) = -\sum_{i=1}^{L} \frac{1}{L} \log\frac{1}{L} = \log(L) \ \ bits \tag{2}$$
>
> For L a power of 2 (i.e. $L = 2^N$), we have
>
> $$H(X) = N \ \ bits\ . \tag{3}$$
>
> For $L = 2$ symbols, the entropy is 1 bit. For $L = 4$, $H(X) = 2$ bits; for $L = 8$, $H(X) = 3$ bits; .... This has intuitive appeal since 2^N symbols can be represented uniquely with N bits.

Example 4.2: Consider Example 4.1, specifically with $L = 2$, but let the symbols probabilities be general. The 2 symbols are $\{x_1, x_2\}$ with probabilities

$$p_i \;=\; \begin{cases} \rho & i = 1 \\ 1-\rho & i = 2 \end{cases} \quad . \tag{4}$$

Then the entropy is

$$H(X) \;=\; H(\rho) \;=\; -\rho\,\log\rho \;-\; (1-\rho)\,\log(1-\rho) \;\; bits \;\; . \tag{5}$$

Figure 32 (6.2-1, p. 334 of the Course Text) shows $H(X)$ vs. ρ. Observe that $H(X)$ peaks at $\rho = 0.5$ (which corresponds to Example 4.1 with $L = 2$). So, the entropy is at most 1 bit. If the symbols are not equally likely, entropy is less than a bit. As ρ approaches 0 or 1 the entropy goes to zero, indicating the amount of uncertainty goes to zero.

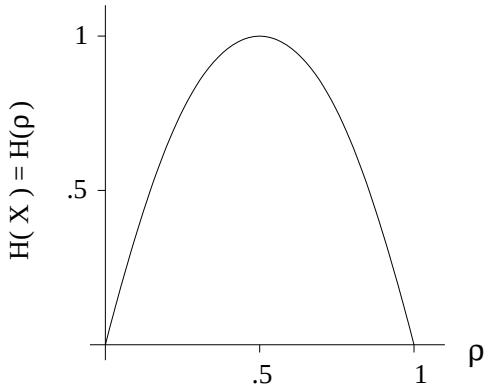


Figure 32: Entropy vs. symbol probability for a binary random variable.

Example 4.3: Consider a random X composed of 4 symbols, $\{x_1, x_2, x_3, x_4\}$, and let

$$p_i \;=\; \begin{cases} \frac{1}{2} & i = 1 \\ \frac{1}{4} & i = 2 \\ \frac{1}{8} & i = 3, 4 \end{cases} \quad . \tag{6}$$

The entropy is

$$H(X) \;=\; -\frac{1}{2}\,\log\frac{1}{2} - \frac{1}{4}\,\log\frac{1}{4} - \frac{1}{8}\,\log\frac{1}{8} - \frac{1}{8}\,\log\frac{1}{8} \;=\; \frac{7}{4}\;bits \;\; . \tag{7}$$

Because the symbols are not equally likely, the entropy is less than two (the number of bits required to uniquely represent the symbols).

Consider a random variable X with symbols $\{x_1, x_2, \cdots, x_L\}$ and corresponding probabilities $\{p_1, p_2, \cdots, p_L\}$. We define the *self-information* of symbol x_i as

$$I(x_i) \;=\; -\log\left(p_i\right) \quad . \tag{8}$$

$I(x_i) \geq 0$ since symbol probabilities are less than 1. Notice that if a p_i is close to zero, the self-information of symbol x_i is high. In terms of the symbol self-information, the entropy is

$$H(X) \;=\; \sum_{i=1}^{n} \; p_i \; I(x_i) \quad . \tag{9}$$

The *entropy* $H(X)$ *is the average self-information* of random variable X.

> *Example 4.4:* In Example 4.3, the self-informations of the 4 symbols are $I(x_1) = 1$, $I(x_2) = 2$, $I(x_3) = 3$ and $I(x_4) = 3$. Symbols x_3 and x_4 have the highest self-information, meaning that their occurrence conveys more information since they are less likely. However these symbols contribute less to entropy (average information) than do x_1 and x_2 since they occur with less probability.

Entropy Bound Lemma: Let X be a discrete=valued random variable with values $\{x_1, x_2, \cdots, x_L\}$ and corresponding probabilities $\{p_1, p_2, \cdots, p_L\}$.

$$0 \leq H(X) \leq \log(L) \quad . \tag{10}$$

$H(X) = 0$ if X is not random. $H(X) = \log(L)$ if X is uniformly distributed.

To prove this, first consider the lower bound. Since self information and probabilities are nonnegative, from Eq (9), $H(X) \geq 0$. If only one symbol, say x_1, occurs (i.e. $p_1 = 1$), then noting that $\lim_{p_i \to 0} p_i \log(p_i) = 0$, we have that $H(X) = -p_1 \log\left(p_1\right) = 0$. In this case the average self information (i.e. entropy) is zero because there is no uncertainty in X.

Now we prove the upper bound. We must show that $H(X) \;-\; \log(L) \;\leq\; 0$.

$$\begin{aligned}
H(X) \;-\; \log(L) \;&=\; \sum_{i=1}^{L} \; p_i \log\left(\frac{1}{p_i}\right) \;-\; \log(L) \\
&=\; \sum_{i=1}^{L} \; p_i \log\left(\frac{1}{p_i}\right) \;-\; \sum_{i=1}^{L} p_i \; \log(L) \\
&=\; \sum_{i=1}^{L} \; p_i \log\left(\frac{1}{L\; p_i}\right) \;=\; \frac{1}{\ln(2)} \sum_{i=1}^{L} \; p_i \ln\left(\frac{1}{L\; p_i}\right) \\
&\leq\; \frac{1}{\ln(2)} \sum_{i=1}^{L} \; p_i \left(\frac{1}{L\; p_i} - 1\right) \;=\; 0.
\end{aligned}$$

In Example 4.1 we have already shown that $H(X) = \log(L)$ for uniform distributed X.

4.1.2 A Continuous-Valued Random Variable

Strictly speaking, the concept of entropy developed for a discrete random variable does not generalize to the continuous-valued random variable case. This can be explained by the fact that in general a continuous-valued random variable will require an infinite number of bits to represent. For a continuous-valued random variable, the concept of entropy is quantified as *differential entropy* which is defined as follows.

Let X be a random variable that can take on values x with probabilities described by its PDF $p(x)$. The *differential entropy* is

$$H(X) \;=\; -\int_{-\infty}^{\infty} p(x)\log\left(p(x)\right)\,dx \quad . \tag{11}$$

Comparing Eqs (1) and (11), note that the entropy bound has no meaning here since L has no meaning. Also note that the differential entropy can be negative. (See [4], Chapter 9, for a more in-depth discussion on differential entropy).

Example 4.5: Let X be a zero-mean Gaussian random variable with variance σ_x^2. Its PDF is

$$p(x) = \frac{1}{\sqrt{2\pi\sigma_x^2}}\; e^{-x^2/2\sigma_x^2} \quad . \tag{12}$$

Let us determine the differential entropy in nats (by taking log base e) and then convert to bits.

$$\begin{aligned}
H_e(X) \;&=\; -\int_{-\infty}^{\infty} p(x)\;\ln\left(p(x)\right)\,dx \\
&=\; -\int_{-\infty}^{\infty} p(x)\;\left[-\ln\left(\sqrt{2\pi\sigma_x^2}\right) - \frac{x^2}{2\sigma_x^2}\right]\,dx \\
&=\; \frac{1}{2}\ln\left(2\pi\sigma_x^2\right) \;+\; \frac{E\{X^2\}}{2\sigma_x^2} \\
&=\; \frac{1}{2}\;\left[\ln\left(2\pi\sigma_x^2\right) \;+\; \ln(e)\right] \\
&=\; \frac{1}{2}\;\ln\left(2\pi e\sigma_x^2\right) \quad nats\;.
\end{aligned} \tag{13}$$

Utilizing $\ln(x) = \frac{1}{\log(e)}\log(x)$ to convert this to bits, we have

$$H_e(X) \;=\; \frac{1}{2}\,\frac{1}{\log(e)}\;\ln\left(2\pi e\sigma_x^2\right) \quad nats \;\;, \tag{14}$$

and

$$H(X) \;=\; H_b(X) \;=\; \log(e)\;H_e(X) \;=\; \frac{1}{2}\log\left(2\pi e\sigma_x^2\right) \quad bits. \tag{15}$$

Notice that the differential entropy increases as the variance increases, which supports the notion that the uncertainty associated with X increases as the variance increases.

An Asside:

Consider a discrete-valued random variable X that can take on values $\{x_1, x_2, \cdots, x_n\}$. Let $p(X)$ and $q(X)$ be two PDF's. The *relative entropy* or *Kullback Leibler distance* between two discrete PDF's is defined as

$$D(p||q) \;=\; \sum_{i=1}^{n} \; p(x_i) \; \log\left(\frac{p(x_i)}{q(x_i)}\right) \quad . \tag{16}$$

The relative entropy is a useful indicator of the "distance" between the two PDF's. $D(p||q) \geq 0$ ([4], p. 26). By inspection of Eq (16), if $p(\cdot) = q(\cdot)$, $D(p||q) = 0$.

Example 4.6: Consider a random variable X that takes on values $\{0, 1\}$. Consider two PDF's

$$p(x) \;=\; (1-r)\delta(x) \;+\; r\delta(x-1) \; ; \qquad q(x) \;=\; (1-s)\delta(x) \;+\; s\delta(x-1) \quad . \tag{17}$$

Give the general expressions for $D(p||q)$ and $D(q||p)$, and determine the specific relative entropies for the cases: $r = s$ (i.e. equivalent PDF's); and $r = \frac{1}{2},\ s = \frac{1}{4}$.

Solution:

$$D(p||q) \;=\; (1-r)\log\left(\frac{(1-r)}{(1-s)}\right) \;+\; r\log\left(\frac{r}{s}\right) \quad . \tag{18}$$

Also,

$$D(q||p) \;=\; (1-s)\log\left(\frac{(1-s)}{(1-r)}\right) \;+\; s\log\left(\frac{s}{r}\right) \quad . \tag{19}$$

For $r = s$, both terms of either relative entropy expression have log(1) terms, so $D(p||q) = D(q||p) = 0$. The relative entropy or Kullback Leibler distance is zero.

For $r = \frac{1}{2},\ s = \frac{1}{4}$,

$$D(p||q) \;=\; 0.2075 \; bits \; ; \qquad D(q||p) \;=\; 0.1887 \; bits \quad . \tag{20}$$

Note that in general $D(p||q) \neq D(q||p)$.

4.2 Two Random Variables

In this Subsection we consider information and entropy concepts for two random variables. As in Subsection [4.1], we will look at the discrete-valued case first, then the continuous-valued case. We will conclude the subsection with a consideration of one continuous-valued, one discrete-valued random variable case.

4.2.1 Two Discrete-Valued Random Variables

Let X and Y be two discrete-valued random variables that can take on values $\{x_i; i = 1, 2, \cdots, n\}$ and $\{y_j; j = 1, 2, \cdots, m\}$ with corresponding probabilities $\{P(x_i); i = 1, 2, \cdots, n\}$ and $\{P(y_j); j = 1, 2, \cdots, m\}$. The *mutual information* between the value x_i of X and the value y_j of Y is defined as

$$I(x_i; y_j) \;=\; \log\left(\frac{P(x_i/y_j)}{P(x_i)}\right) \;=\; \log\left(\frac{P(y_j, x_i)}{P(x_i)P(y_j)}\right) \;=\; \log\left(\frac{P(y_j/x_i)}{P(y_j)}\right) \tag{21}$$

where $P(x_i/y_j)$ is the conditional probability $P(X/y_j)$ evaluated at $X = x_i$.

Note from Eq (21) that $I(x_i; y_j) = I(y_j; x_i)$. $I(x_i; y_j)$ can be negative. If X and Y are statistically independent, so that $P(x_i/y_j) = P(x_i)$, then $I(x_i; y_j) = \log(1) = 0$. The mutual information is zero. On the other hand, if y_j completely determines x_i, so that $P(x_i/y_j) = 1$, then $I(x_i; y_j) = I(x_i)$ – the self-information is equal to the mutual information.

> *Example 4.7:* Consider the Binary Symmetric Channel (BSC) illustrated in Figure 33. The random variables X an Y are the binary channel input and output, respectively. Assume the binary input symbols $\{x_1, x_2\}$ are equally likely. The conditional bit error probabilities are $P(y_2/x_1) = P(y_1/x_2) = \rho$, while the correct decision probabilities are $P(y_1/x_1) = P(y_2/x_2) = (1 - \rho)$. The problem is to determine the mutual information between the input symbols and the output symbols.

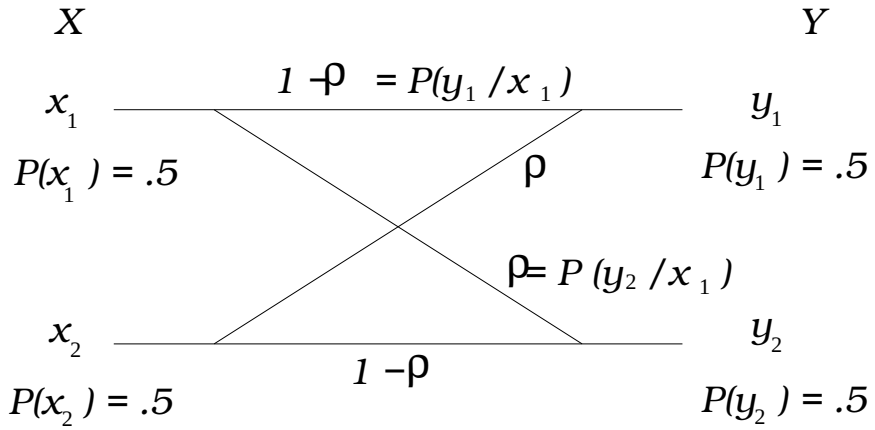


Figure 33: Binary symmetric channel.

Note that the binary modulation schemes considered earlier in Sections 2,3, under the assumptions considered and using an ML receiver, can be modeled as BSC's.

Solution: Since $P(y_i) = P(y_1/x_1)P(x_1) + P(y_1/x_2)P(x_2)$, for the binary symmetric channel with equally likely inputs, the outputs are equally likely. Thus,

$$I(x_1;y_1) \;=\; I(x_2;y_2) \;=\; \log\left(\frac{(1-\rho)}{1/2}\right) \;=\; \log(2(1-\rho)) \quad , \tag{22}$$

and

$$I(x_1;y_2) \;=\; I(x_2;y_1) \;=\; \log\left(\frac{\rho}{1/2}\right) \;=\; \log(2\rho) \quad . \tag{23}$$

Note that for $\rho = 0$, x_i completely determines y_i. In this case $I(x_i;y_i) = 1$ bit, the mutual information is equal to the self-information of x_i or y_i, and $I(x_i;y_j) = -\infty$ for $i \neq j$ indicating that y_j precludes x_i for $i \neq j$.

On the other hand, for $\rho = \frac{1}{2}$, $I(x_i;y_i) = 0$ bit, and $I(x_i;y_j) = 0;\ i \neq j$. The mutual information between either input and either output is zero. This makes sense since transmitting one symbol is equally likely to result in receiving either.

The *average mutual information* of discrete random variables X and Y is

$$\begin{aligned} I(X;Y) \;&=\; \sum_{i=1}^{n}\sum_{j=1}^{m} P(x_i;y_j)\; I(x_i;y_j) \\ &=\; \sum_{i=1}^{n}\sum_{j=1}^{m} P(x_i;y_j)\; \log\left(\frac{P(x_i,y_j)}{P(x_i)P(y_j)}\right) \\ &=\; \sum_{i=1}^{n} P(x_i)\; \sum_{j=1}^{m} P(y_j/x_i)\; \log\left(\frac{P(y_j/x_i)}{P(y_j)}\right) \\ &=\; \sum_{i=1}^{n} P(x_i)\; I(x_i;Y) \quad . \end{aligned} \tag{24}$$

It can be shown that $I(X;Y) \geq 0$ (see [4], p. 28), with equality for statistically independent X and Y.

Example 4.8: For the digital communication system considered in Example 4.7 (i.e. a BSC), determine the average mutual information between X and Y (see the Course Text problem 6.2).

Solution: The joint probabilities are

$$P(x_i;y_j) \;=\; \begin{cases} \frac{1-\rho}{2} & i = j \\ \frac{\rho}{2} & i \neq j \end{cases} \tag{25}$$

With this,

$$\begin{aligned} I(X;Y) \;&=\; \sum_{i=1}^{2}\sum_{j=1}^{2} P(x_i;y_j)\; I(x_i;y_j) \\ &=\; 2\frac{1-\rho}{2}\log(2(1-\rho)) \;+\; 2\frac{\rho}{2}\log(2\rho) \\ &=\; 1 + (1-\rho)\log(1-\rho) + \rho\log(\rho) \\ &=\; 1 \;-\; H(\rho) \quad , \end{aligned} \tag{26}$$

where $H(\rho)$ is defined in Example 4.2.

Examples 4.7,8 suggest that mutual information can be useful in comparing two random variables, e.g. the input and output of a channel or quantizer. We will see later that Shannon used it to derive channel and source coding theorems. Now we identify measures that further help us to study information content of multiple random variables.

Consider discrete-valued random variables X and Y. The *joint entropy* is defined as

$$H(X,Y) \;=\; -\sum_{i=1}^{n}\sum_{j=1}^{m} P(x_i,y_j)\;\log\left(P(x_i,y_j)\right) \quad , \tag{27}$$

i.e. the average of $\{-\log(P(X,Y))\}$ as opposed to the average mutual information $I(X;Y)$ which is the average of $\{I(x_i,y_j)\}$.

The *conditional self-information* of a value x_i given a value y_j is defined as

$$I(x_i/y_j) \;=\; -\;\log(P(x_i/y_j)) \quad . \tag{28}$$

The *conditional entropy* of X given Y is the average of the conditional self-information of X over both X and Y, i.e.

$$H(X/Y) \;=\; \sum_{i=1}^{n}\sum_{j=1}^{m} P(x_i;y_j)\;I(x_i/y_j) \quad . \tag{29}$$

The conditional entropy is the average self-information in X after Y has been observed. Note that

$$\begin{aligned} H(X/Y) \;&=\; \sum_{i=1}^{n} P(x_i)\;\sum_{j=1}^{m} P(y_j/x_i)\;(-1)\;\log(P(y_j/x_i) \\ &=\; \sum_{i=1}^{n} P(x_i)\;H(Y/x_i) \quad , \end{aligned} \tag{30}$$

where $H(Y/x_i) = \sum_{j=1}^{m}\; P(y_j/x_i)\;(-1)\;\log(P(y_j/x_i)$.

The chain rule for joint entropy: Considering the joint entropy, we have

$$\begin{aligned} H(X,Y) \;&=\; -\sum_{i=1}^{n}\sum_{j=1}^{m} P(x_i,y_j)\;\log\left(P(x_i,y_j)\right) \\ &=\; -\sum_{i=1}^{n}\sum_{j=1}^{m} P(x_i,y_j)\;\log\left(P(x_i)P(y_j/x_i)\right) \\ &=\; -\sum_{i=1}^{n}\sum_{j=1}^{m} P(x_i,y_j)\;\log\left(P(x_i)\right) \;-\; \sum_{i=1}^{n}\sum_{j=1}^{m} P(x_i,y_j)\;\log\left(P(y_j/x_i)\right) \\ &=\; -\sum_{i=1}^{n} P(x_i)\;\log\left(P(x_i)\right) \;-\; \sum_{i=1}^{n}\sum_{j=1}^{m} P(x_i,y_j)\;\log\left(P(y_j/x_i)\right) \\ &=\; H(X) \;+\; H(Y/X) \quad . \end{aligned} \tag{31}$$

Similarly,

$$H(X,Y) \;=\; H(Y) \;+\; H(X/Y) \quad . \tag{32}$$

So, the joint entropy is the entropy of one random variable plus the conditional entropy of the other. This is called the chain rule because it allows us to "build" joint entropy one variable at a time.

Example 4.9: Consider statistically independent discrete-valued random variables X and Y. Their joint PDF is $P(x,y) = P_x(x)P_y(y)$. Determine simplified expressions for joint and conditional entropy.

Solution: The joint entropy is,

$$\begin{aligned}
H(X,Y) &= -\sum_{i=1}^{n}\sum_{j=1}^{m} P_x(x_i)P_y(y_j)\ \log\left(P_x(x_i)P_y(y_j)\right) \\
&= -\sum_{i=1}^{n}\sum_{j=1}^{m} P_x(x_i)P_y(y_j)\ \left[\log\left(P_x(x_i)\right) + \log\left(P_y(y_j)\right)\right] \\
&= \sum_{i=1}^{n} P_x(x_i)\log\left(P_x(x_i)\right)\ \sum_{j=1}^{m} P_y(y_j)\ +\ \sum_{i=1}^{n} P_x(x_i)\ \sum_{j=1}^{m} P_y(y_j)\log\left(P_y(y_j)\right) \\
&= \sum_{i=1}^{n} P_x(x_i)\log\left(P_x(x_i)\right)\ +\ \sum_{j=1}^{m} P_y(y_j)\log\left(P_y(y_j)\right) \\
&= H(X)\ +\ H(Y)\quad . \qquad (33)
\end{aligned}$$

For statistically independent random variables, the joint entropy is the sum of the individual entropies. From Eq (33) and the general chain rule equation, we have that $H(Y/X) = H(Y)$.

A relationship between average mutual information $I(X;Y)$ and entropy measures: Considering average mutual information, we have

$$\begin{aligned}
I(X;Y) &= \sum_{i=1}^{n}\sum_{j=1}^{m} P(x_i;y_j)\ \log\left(\frac{P(x_i/y_j)}{P(x_i)}\right) \\
&= -\sum_{i=1}^{n}\sum_{j=1}^{m} P(x_i;y_j)\ \log\left(P(x_i)\right)\ +\sum_{i=1}^{n}\sum_{j=1}^{m} P(x_i;y_j)\ \log\left(P(x_i/y_j)\right) \\
&= -\sum_{i=1}^{n} P(x_i)\ \log\left(P(x_i)\right)\ -\left(-\sum_{i=1}^{n}\sum_{j=1}^{m} P(x_i;y_j)\ \log\left(P(x_i/y_j)\right)\right) \\
&= H(X)\ -\ H(X/Y)\quad . \qquad (34)
\end{aligned}$$

Similarly,

$$I(X;Y)\ =\ H(Y)\ -\ H(Y/X)\quad . \qquad (35)$$

Considering this relationship and the chain rule, Figure 34 illustrates the relationships between entropy, joint entropy, conditional entropy and average mutual information. Notice, for example, that

$$H(X;Y) \;=\; H(X) \;+\; H(Y) \;-\; I(X;Y) \quad . \tag{36}$$

The joint entropy is the sum of the individual entropies minus the mutual information. Also, note that the average mutual information corresponds to the intersection of the information of the two random variables.

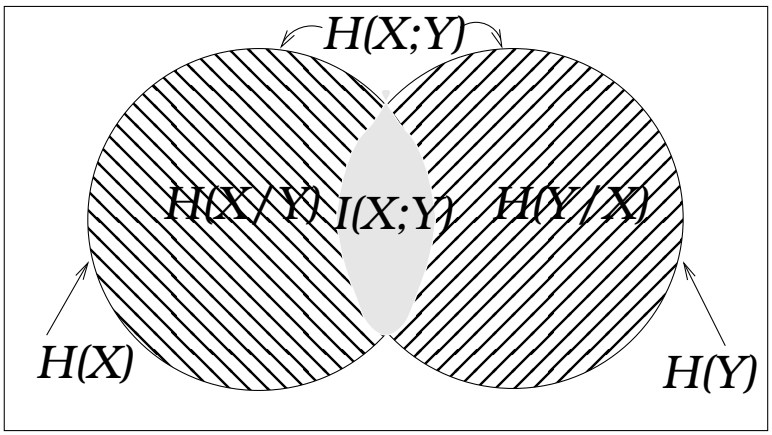


Figure 34: Relationship between average mutual information and entropy.

Example 4.10: Consider two random variables X and Y that take on the values which are given, along with their joint probabilities, in Table 4.1.

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

Table 4.1

Determine the entropies $H(X)$ & $H(Y)$, the joint entropy $H(X,Y)$, the conditional entropies $H(X/Y)$ & $H(Y/X)$, and the average mutual information $I(X,Y)$.

Solution:

The marginal probabilities for X and Y are, respectively, $\{\frac{1}{4},\frac{1}{4},\frac{1}{4},\frac{1}{4}\}$ and $\{\frac{1}{2},\frac{1}{4},\frac{1}{8},\frac{1}{8}\}$. From these, $H(X)=\frac{7}{4}$ and $H(Y)=2$. The one conditional entropy can be obtained using Eq (29). For example $H(Y/X)=\frac{11}{8}$ bits. Then, from the chain rule, Eq (31), $H(X,Y)=\frac{27}{8}$ bits. Again from the chain rule, $H(X/Y)=H(Y/X)+H(X)-H(Y)=\frac{13}{8}$. From the average mutual-information/entropy relationship developed above, Eq (34), $I(X;Y)=H(Y)-H(Y/X)=\frac{3}{8}$.

4.2.2 Continuous-Valued Random Variables

For any two continuous-valued random variables X and Y, the average mutual information is defined as

$$\begin{aligned} I(X;Y) \;=\; Ave\{I(x;y)\} \;&=\; \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} p(x,y)\; I(x;y)\; dx\; dy \\ &=\; \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} p(x,y)\; \log\left(\frac{p(x,y)}{p(x)p(y)}\right) dx\; dy \\ &=\; \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} p(x)p(y/x)\; \log\left(\frac{p(y/x)}{p(y)}\right) dx\; dy \quad . \end{aligned} \tag{37}$$

It can be shown that $I(X;Y) \geq 0$, with equality for statistically independent X and Y.

Other useful measures are: conditional differential entropy

$$H(X/Y) \;=\; -\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} p(x,y)\; \log\left(p(x/y)\right)\; dx\; dy \quad ; \tag{38}$$

joint differential entropy

$$H(X;Y) \;=\; -\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} p(x,y)\; \log\left(p(x,y)\right) dx\; dy \quad ; \tag{39}$$

and relative differential entropy or Kullback Leibler distance

$$D(p||q) \;=\; \int_{-\infty}^{\infty} p(x)\; \log\left(\frac{p(x)}{q(x)}\right)\; dx \quad . \tag{40}$$

Paralleling the discrete random variables case, we have

$$I(X;Y) \;=\; H(X) \;-\; H(X/Y) \quad , \tag{41}$$

and the chain rule

$$H(X,Y) \;=\; H(X) \;+\; H(Y/X) \quad . \tag{42}$$

4.2.3 One Discrete-Valued, One Continuous=Valued Random Variable

There are several situations where communications engineers are interested in the mutual information between a discrete-valued random variable and a continuous-valued one. We will employ it when considering the capacity of a channel for which the input is a discrete-valued symbol and the output is that symbol superimposed onto AWGN. Let X be discrete-valued, with values $x_i; j = 1, 2, \cdots, n$ and PDF $P(x)$. Let Y be continuous-valued with PDF $p(y)$. The average mutual information between X and Y is

$$I(X;Y) \;=\; \sum_{i=1}^{n} \int_{-\infty}^{\infty} P(x_i)\; p(y/x_i)\; \log\left(\frac{p(y/x_i)}{p(y)}\right)\; dy \quad . \tag{43}$$

Example 4.11: Let X be a discrete-valued random variable that can take on equally probable values $x_1 = A$ and $x_2 = -A$, and let $Y = X + N$ where N is zero-mean Gaussian with variance σ_n^2. Determine an expression for their average mutual information $I(X,Y)$, plot it vs. SNR/bit $\gamma_b = \frac{A^2}{2\sigma_n^2}$, and compare it to the average mutual information of a binary PSK implementation of a BSC (i.e. see Example 4.8 with $\rho = Q(\sqrt{2\gamma_b})$).

Solution: With $P(x_i) = \frac{1}{2};\; i = 1, 2$ and

$$p(y/x_1) \;=\; \frac{1}{\sqrt{2\pi\sigma_n^2}}\; e^{(y-A)^2/2\sigma_n^2}\;; \qquad p(y/x_2) \;=\; \frac{1}{\sqrt{2\pi\sigma_n^2}}\; e^{(y+A)^2/2\sigma_n^2} \quad , \tag{44}$$

Eq (43) reduces to

$$I(X;Y) \;=\; \frac{1}{2}\int_{-\infty}^{\infty} p(y/A) \log\frac{p(y/A)}{p(y)}\; dy \;+\; \frac{1}{2}\; \int_{-\infty}^{\infty} p(y/-A) \log\frac{p(y/-A)}{p(y)}\; dy \quad , \tag{45}$$

where by total probability $p(y) = \frac{1}{2}[p(y/A) + p(y/-A)]$. This can be evaluated numerically.

Figures 6.5-5 and 6.5-6 of the Course Text[4] show $I(X;Y)$, plotted as a function of SNR/bit $\gamma = \frac{A^2}{2\sigma^2}$, for the BSC and AWGN channels respectively[5]. Compared to the average mutual information between the input and output of a BSC, the AWGN channel has increased average mutual information. This should be expected since the BSC output is the AWGN channel output quantized.

[4]The reason that the independent axes of these figures is labeled "Capacity" will become apparent when we discuss capacity a little later. For now, just assume that $I(X,Y)$ is capacity in these figures.

[5]Actually, Figure 6.5-6 of the Course Text is incorrect. It shows a mutual information curve for the AWGN channel which is worse than that of the BSC – which can not be. See Figure 48 of these Course Notes for an accurate comparison

Example 4.12: Consider a binary PSK modulation scheme, and denote the binary input X which takes on values $x_1 = -1$ and $x_2 = 1$. Let R denote the received signal, after matched filtering and sampling. Due to AWGN, it has PDF

$$p(r/x_i) \;=\; \frac{1}{\sqrt{2\pi\sigma_n^2}}\, e^{-(r-x_i)^2/2\sigma_n^2} \tag{46}$$

where $\sigma_n^2 = 0.2$ is the sampled noise variance. Assume that $P(x_1) = P(x_2) = 0.5$.

Problem:

1. Say R is threshold detected to form binary Y as follows:

$$Y \;=\; \begin{cases} y_1 & R < 0. \\ y_2 & R > 0. \end{cases} \quad . \tag{47}$$

   Determine the average mutual information $I(X;Y)$.

2. Say R is threshold detected to form trinary Z as follows:

$$Z \;=\; \begin{cases} z_1 & R < -0.3 \\ z_2 & -0.3 < R < -0.3 \\ z_3 & R > 0.3 \end{cases} \quad . \tag{48}$$

   Determine the average mutual information $I(X;Z)$.

3. Determine the average mutual information $I(X;R)$.

Compare your result in (1-3).

Solution:

1. This binary symmetric channel, with error probability $\rho = Q(1/\sqrt{0.2}) \approx 0.015$. From Example 4.8,

$$I(X;Y) \;\approx\; 1 \;-\; H(\rho) \;=\; 1 = 0.0969 \;=\; 0.9031 \quad . \tag{49}$$

2. This is a two level input, three level output channel often encountered in ARQ channel coding schemes, where the middle level output is an "erasure". Taking advantage of PDF and threshold symmetries, we have that

$$\begin{aligned} P(z_1/x_1) \;&=\; P(z_3/x_2) \;=\; 1 - Q(.7/\sqrt{0.2}) \;\approx\; 0.939 && (50)\\ P(z_3/x_1) \;&=\; P(z_1/x_2) \;=\; Q(1.3/\sqrt{0.2}) \;\approx\; 0.002 && (51)\\ P(z_2/x_1) \;&=\; P(z_2/x_2) = Q(.7/\sqrt{0.2}) - Q(1.3/\sqrt{0.2}) \approx 0059 && (52)\\ P(z_2) \;&=\; \frac{1}{2}[P(z_2/x_1) + P(z_2/x_2)] \;\approx\; 0.059 && (53)\\ P(z_1) \;&=\; P(y_3) \;=\; \frac{1}{2}[P(z_1/x_1) + P(z_1/x_2)] \;\approx\; 0.47 \quad . && (54) \end{aligned}$$

Then,

$$I(X;z_2) \;=\; \frac{1}{2}\sum_{i=1}^{2} P(z_2/x_i)\log_2\left(\frac{P(z_2/x_i)}{P(z_2)}\right) \;=\; 0 \tag{55}$$

$$I(X;z_1) \;=\; I(X;z_3) = \frac{1}{2}\left\{\sum_{i=1}^{2} P(z_1/x_i)\log_2\left(\frac{P(z_1/x_i)}{P(z_1)}\right)\right\} \approx 0.46 \tag{56}$$

Finally,

$$I(X;Z) \;=\; 2\; I(X;y_1) \;\approx\; 0.9203 \quad . \tag{57}$$

3. For R, the SNR is approximately 3.8 dB. From Figure 48 of these Notes (p. 104), we have that $I(X;R) \approx 0.94$.

Comparing results, we see that

$$I(X;R) \;>\; I(X;Z) \;>\; I(X;Y) \quad . \tag{58}$$

This should be expected since in going from R to Z to Y we have a greater degree of receiver matched-filter output quantization.

4.3 Multiple Random Variables

All information and entropy measures defined for the two random variable case addressed in Subsection 4.2.1 generalize to a general number k of random variables.

Consider a set of k discrete-valued random variables $\{X_1, X_2, \cdots, X_k\}$ with each $X_i; i = 1, 2, \cdots, k$ taking on values $\{x_{i,1}, x_{i,2}, \cdots, x_{i,n_i}\}; i = 1, 2, \cdots, k$. The joint PDF is $P(x_1, x_2, \cdots, x_k)$. The *joint entropy* of this set of random variables is defined as the following generalization of Eq (27):

$$H(X_1, X_2, \cdots, X_k) \;=\; -\sum_{j_1=1}^{n_1} \cdots \sum_{j_k=1}^{n_k} P(x_{1,j_1}, x_{2,j_2}, \cdots, x_{k,j_k}) \;\log\left(P(x_{1,j_1}, x_{2,j_2}, \cdots, x_{k,j_k})\right) \quad . \tag{59}$$

For the k random variable chain rule, we repeatedly apply the two random variable rule for each increase in k:

$$H(X_1, X_2) \;=\; H(X_1) \;+\; H(X_2/X_1) \tag{60}$$

$$\begin{aligned} H(X_1, X_2, X_3) \;&=\; H(X_1) \;+\; H(X_2, X_3/X_1) \\ &=\; H(X_1) \;+\; H(X_2/X_1) \;+\; H(X_3/X_2, X_1) \end{aligned} \tag{61}$$

$$\vdots$$

$$\begin{aligned} H(X_1, X_2, \cdots, X_k) \;&=\; H(X_1) \;+\; H(X_2/X_1) \;+\; \cdots \;+\; H(X_k/X_{k-1}, \cdots, X_2, X_1) \\ &=\; \sum_{i=1}^{k} H(X_i/X_{i-1}, \cdots, X_2, X_1) \quad . \end{aligned} \tag{62}$$

4.4 Random Sequences & Entropy Rate

Let $\{X_k; k = \cdots, -1, 0, 1, 2, \cdots\}$ be a discrete-time random process where each sample X_k is a discrete random variable. Let X denote the random process. The *entropy rate* of X is defined as

$$H_\infty(X) \;=\; \lim_{k\to\infty} \;\frac{1}{2k+1}\; H(X_{-k}, \cdots, X_{-1}, X_0, X_1, \cdots, X_k) \quad . \tag{63}$$

Example 4.13: Consider an iid random process X. Then

$$\begin{aligned} H_\infty(X) \;&=\; \lim_{k\to\infty} \;\frac{1}{2k+1}\; H(X_{-k}) + \cdots + H(X_{-1}) + H(X_0) + H(X_1) + \cdots + H(X_k) \\ &=\; \lim_{k\to\infty} \;\frac{(2k+1)H(X_0)}{(2k+1)} \;=\; H(X_0) \quad . \end{aligned} \tag{64}$$

With correlation across time, we can expect $H_\infty(X)$ to decrease.

5 Source Coding

In this Section of the Course we consider source coding, which is the processing of source information into the a format compatible with transmission over a digital communication system, while compressing it so as to reduce required communication resources (e.g. channel bandwidth, storage requirements). We will refer to a *discrete sourse* as one that is discrete in both time and value. We will in turn address successively more complex sources and corresponding source coding problems, starting with discrete memoryless sources (DMS), continuing with discrete sources with memory, then discrete-time continuous-valued, and finally addressing analog (continuous-time and valued) sources.

Source coders can be categorized as being either lossless or lossy. The output of a lossless code can be used to perfectly reconstruct the original source, whereas the original source can not be exactly reconstructed from the output of a lossy source coder since information is lost. We will use concepts from information theory to identify the maximum compression possible for lossless coding of a discrete source, and to identify the maximum compression possible for a given level of distortion for lossy coding.

Concerning source coding and compression, we are interested in the data or source rate, which is how much data there is per unit time. We are also interested in the information rate, which is a measure derived based on concepts from Section 4 of this Course that quantifies the amount of information in the compressed or uncompressed source per unit time. Qualitatively, we are interested in coding so as the reduce or minimizing the data rate for a given information rate.

This Section of the Course corresponds to Sections 6.3 and 6.4 of the Course Text.

5.1 Lossless Coding for Discrete Sources

Consider a discrete source X_k which consists of a sequence of symbols (k is the discrete-tiem index), each symbol being from the finite alphabet $\{x_i, i = 1, 2, \cdots, L\}$. If each isolated symbol is coded simply by directly representing it with bits, the number of bits required is

$$R \;=\; \log_2(L) \tag{1}$$

if L is a power of 2 and

$$R \;=\; \lfloor \log_2(L) \rfloor \;+1 \;>\; \log_2(L) \tag{2}$$

if L is not. R is the *code rate in bits/symbol* of this simple code. A question addressed here is – are there lossless codes that reduce this code rate?

The discrete source is composed of a sequence of random variables. Assume the source is stationary, and denote the probabilities of the symbols as $\{P(x_i), i = 1, 2, \cdots, L\}$ (i.e. independent of time k). The entropy (in bits) of each random variable X_k is

$$H(X_k) \;=\; -\sum_{i=1}^{L} \; P(x_i) \; \log_2(P(x_i)) \quad . \tag{3}$$

We know from Subsection 4.1 of the Course that $H(X_k) \;\leq\; \log_2(L)$, with equality only if $P(x_i) = \frac{1}{L}; i = 1, 2, \cdots, L$. For very unevenly distributed symbols, $H(X_k) \;\ll\; \log_2(L)$.

Another question addressed here is – can we find a code with rate R that decreases like the entropy $H(X_k)$ as the symbol distributions become more unevenly distributed?

Recall that if the samples of the discrete source are iid, the entropy rate is

$$H_\infty(X) \;=\; H(X_k) \quad . \tag{4}$$

If the samples are correlated, the entropy rate is less than this. This begs the following question. For sources with correlated samples, can the code rate be reduced by exploiting this correlation?

5.1.1 Discrete Memoryless Source (DMS)

In considering the coding of a discrete memoryless source (DMS), first note that memoryless implies that the samples of the discrete source random process are statistically independent, so that we can not exploit correlation between samples.

Fixed-Length Codes

The simple isolated symbol coding scheme described at the top of this Subsection (Subsection 5.1) is a fixed-length coding scheme. Each symbol is represented by R bits. Since $R \geq \log_2(L)$ and $H(X_k) \leq \log_2(L)$, we have that

$$R \;\geq\; H(X_k) \quad . \tag{5}$$

Defining the *efficiency* as entropy/rate ($H(X_k)/R$), we see that the efficiency of this code is less than or equal to one, with equality only when L is a power of 2 and the symbols are equally likely.

First, consider only the case of equally likely symbols, so that $H(X_k) = \log_2(L)$. Then, since $\log_2(L) \leq R \leq \log_2(L) + 1$,

$$H(X_k) \leq R \leq H(X_k) \;+\; 1 \quad . \tag{6}$$

> *Example 5.1:* Take the $L = 5$ case. $H(X_k) = 2.322$ and $R = 3$ bits/symbol, so efficiency is 77.4%.
>
> This situation can be improved by taking fixed-length blocks of symbols and representing each block with a fixed number of bits. For example, again taking $L = 5$, consider $J = 3$ symbols/block. Since there are $L^J = 125$ possible blocks, we need 7 bits to uniquely represent each block. The rate is $R = 7/3 = 2.333...$ bits/symbol. Since the entropy is $H(X_k) = \log_2(5) = 2.322$, the efficiency has been improved to 99.5% . We conclude that fixed-length block coding for equally likely, independent symbols can be efficient.

Considering symbols with a nonuniform distribution such that $H(X_k) \ll \log_2(L)$, we see that $R \gg H(X_k)$. This fact, along with the possibility of taking advantage of symbol-to-symbol correlations, motivate us to consider other source coding strategies and source coding bounds.

Variable-Length Codes

To overcome the inefficiency of fixed-length codes for non uniformly distributed source symbols, we now consider, for symbol-by-symbol source coding, assigning variable length code words to different symbols. The Morse and Braille codes are early examples of variable-length codes that effectively exploits nonuniform symbol probabilities. The symbols for Morse and Braille codes are, respectively, the letters of the alphabet and selected words. The idea is to represent frequently occurring (high probability) symbols with only a few bits (i.e. an "E" is represented by a single dot "." in Morse code) and less probably symbols with more bits (a "Q" is represented as "- -.-").

For a random variable X with L symbols $\{x_i, i = 1, 2, \cdots, L\}$ with corresponding probabilities $P(x_i); i = 1, 2, \cdots, L$ consider assigned variable-length code words $\mathbf{C}_i;\ i = 1, 2, \cdots, L$ of lengths $n_i;\ i = 1, 2, \cdots, L$ bits. The average number of bits/symbol is the code rate

$$\overline{R} \ = \ \sum_{i=1}^{L} \ n_i \ P(x_i) \quad . \tag{7}$$

Example 5.2: For the four symbols listed in Table 6.3-1, p. 339, of the Course Text, a fixed-length (2 bits/symbol) code would have rate $\overline{R} = 2$ bits/symbol. Code I has rate $\overline{R} = 1\frac{1}{2}$, while Code II has rate $\overline{R} = 1\frac{3}{8}$ which is the entropy of the symbol.

To be effective, a lossless code must have the property of being uniquely decodable. A desirable property of a variable-length code is that no code word be a prefix of any other code word. A code what adheres to this property is called a *prefix* code or *instantaneous code*. The latter name refers to the fact that such a code word can be uniquely decoded without referring to future code words, since the end of a code word can be instantaneously detected.

The Course Text, on pp. 339-340, provides a good example of prefix and non-prefix codes, and illustrates the fact that the code tree for prefix code words must correspond to terminal nodes.

Our objective is to find a variable-length source code, preferably a prefix code, with minimum rate $\overline{R}$. The following inequality provides a tool for designing prefix codes.

Kraft Inequality: Let D be the number of possible symbols for each element of a code word (i.e. $D = 2$ for binary code words). Let L be the number of source symbols (i.e. the number of code words needed). For a prefix code, the lengths of the code words, $n_i; i = 1, 2, \cdots, L$ must satisfy

$$\sum_{i=1}^{L} \ D^{-n_i} \ \leq \ 1 \quad . \tag{8}$$

Also, given a set of code word lengths that satisfy this inequality, there exists a prefix code of code words with these lengths.

The Course Text, pp. 340-341, provides a proof of the Kraft inequality for $D = 2$. The extension for general D is straightforward.

The Course Text also provides the following theorem (p. 342).

> *A rate bound theorem:* Let X_k be a source sample with possible symbols $\{x_i;\ i = 1, 2, \cdots, L\}$ with corresponding probabilities $P(x_i) = p_i;\ i = 1, 2, \cdots, L$ and finite entropy $H(X_k)$. A prefix code exists with rate $\overline{R}$ which is bounded as
>
> $$H(X_k) \ \leq \ \overline{R} \ < \ H(X_k) \ + \ 1 \quad . \tag{9}$$
>
> That is, a variable-length prefix code exists which will compress the source to within 1 bit of its entropy.

We'll sketch a proof for $D = 2$ though it generalizes to integers $D > 2$. First, the lower bound can by established by solving the following problem using the method of Lagrange multipliers:

$$\min_{n_1, n_2, \cdots, n_L} \quad \overline{R} = \sum_{i=1}^{L} \ n_i \ p_i \tag{10}$$

subject to the Kraft inequality

$$\sum_{i=1}^{L} \ 2^{-n_i} \ \leq \ 1 \quad . \tag{11}$$

If the n_i are not constrained to be integers, the solution is

$$n_i^* \ = \ -\log_2 p_i \ ; \qquad i = 1, 2, \cdots, L \tag{12}$$

yielding the minimum $\overline{R}$

$$\overline{R}^* \ = \ H(X_k) \quad . \tag{13}$$

In general these n_i^* will not be integers, and so rounding them up to the nearest integer will result in $\overline{R} \geq H(X_k)$.

Next, for proof the theorem's upper bound (i.e. that there exists a prefix code such that $\overline{R} < H(X_k) + 1$), see the Text p. 342 where it is shown that a prefix code with n_i adhering to

$$2^{-n_i} \ \leq \ p_i \ < \ 2^{-n_i+1} \tag{14}$$

beats this upper bound.

In summary, we have bounds on the rate of a variable-length code that are applicable to any symbol probability distribution, and which are equivalent to the fixed-length code bounds for uniform symbol probability distributions.

There are deficiencies associated with symbol-by-symbol variable length codes. For example, as with fixed-length codes for uniformly distributed symbols, for sources with a small alphabet of symbols we may not be able to get close enough to the lower bound (i.e. we may be almost 1 bit from the entropy). Also, we can't exploit symbol-to-symbol correlation. Before turning to these issues, we will now see how to design Huffman codes, which are optimum prefix codes for symbol-by-symbol coding.

Huffman Coding

Consider a discrete random variable X_k that takes on values from the alphabet $\mathcal{A} = \{x_i; i = 1, 2, \cdots, L\}$ with corresponding probabilities $\{p_i; i = 1, 2, \cdots, L\}$. Assume, without loss of generality, that $p_i \geq p_{i+1}$. Each value x_i is to be represented by a corresponding prefix code word $\mathbf{C}_i$ of length n_i bits. The average number of bits/symbol is the code rate

$$\overline{R} \ = \ \sum_{i=1}^{L} \ n_i \ P(x_i) \quad . \tag{15}$$

Recall that the entropy of this random variable is

$$H(X_k) \ = \ - \ \sum_{i=1}^{L} \ p_i \ \log(p_i) \tag{16}$$

which is bounded as

$$0 \ \leq \ H(X_k) \ \leq \ \log(L) \ , \tag{17}$$

with upper-bound equality when the values are equally likely. Recall also that a prefix code for this random variable exists that has rate

$$H(X_k) \ \leq \ \overline{R} \ < \ H(X_k) \ + \ 1 \quad . \tag{18}$$

The following point is new. If a code is uniquely decodable, then there exists a prefix code with the same code word lengths which is uniquely decodable. (See [5], p. 49, Theorem 3.3.2 and discussion that follows.)

Below we show that a Huffman code for a discrete random variable is optimum in that no other uniquely decodable code for the random variable has lower rate. An optimum code is not necessarily unique (other optimum codes, possibly with different code word length, may exist). We will see that a Huffman code can easily be designed to be a prefix code, so *only prefix codes will be considered*. Although Huffman codes exist for output code words with $D \geq 2$ element values, below we restrict our discussion the binary codes (i.e. $D = 2$).

First we establish necessary conditions for a minimum rate variable length code. This will lead directly to a procedure for designing Huffman codes.

> *Lemma 1:* An optimum code exists for which the code words $\mathbf{C}_{L-1}$ and $\mathbf{C}_L$, associated with two least likely values x_{L-1} and x_L, are of the same length and differ only in the last bit, with $\mathbf{C}_L$ ending in a "1" and $\mathbf{C}_{L-1}$ ending in a "0".

Proof: First note that for at least one optimum code $n_L \geq n_i; i = 1, 2, \cdots, L-1$ since if for some i we had $n_i > n_L$, we could swap the code words $\mathbf{C}_i$ and $\mathbf{C}_L$ without increasing the code rate $\overline{R}$. Second, note that an optimum code for which $n_L \geq n_i; i = 1, 2, \cdots, L-1$ must have a code word of length n_L which differs from $\mathbf{C}_L$ in only the last bit, otherwise we could drop the last bit of $\mathbf{C}_L$ without violating the prefix condition, lowering $\overline{R}$, and the new $\mathbf{C}_L$ would still be unique. Finally, note that if $\mathbf{C}_i$ is the code word differing from $\mathbf{C}_L$ in only the last bit, then for a least one optimum code $i = L - 1$, for the same reason as in the first note.

With Lemma 1, the problem of designing an optimum prefix code reduces to that of designing $\mathbf{C}_i; i = 1, 2, \cdots, L-2$ and identifying the first $n_L - 1$ bits of $\mathbf{C}_i, ; i = L-1, L$. You can probably see how the rest of the story goes. Define a new alphabet $\mathcal{A}^{'} = \{x_i^{'}; i = 1, 2, \cdots, L-1\}$ with corresponding probabilities

$$p_i^{'} = \begin{cases} p_i & i = 1, 2, \cdots, L-2 \\ p_{L-1} + p_L & i = L-1 \end{cases} . \tag{19}$$

Now design an optimum prefix code for this problem, after which you append the $L-1^{th}$ code word with a "0" to form C_{L-1} and then flip its last bit to form the L^{th} code word. Lemma 1 is next used to reduce this $\mathcal{A}^{'}$ alphabet problem to the next level smaller problem. Finally the problem is reduced to the trivial two value alphabet problem where one is assigned the code word "0" and the other the code word "1". The *Huffman code design procedure is the repetitive application of Lemma 1* described above.

The development above shows that at least one optimum code, a Huffman code, exists which must satisfy the repetitively described conditions established above. Conversely, it shows that a Huffman code is an optimum code.

Example 5.3: This example illustrates the basic Huffman code design procedure. Consider random variable X with the four value alphabet $\mathcal{A} = \{x_1, x_2, x_3, x_4\}$ w/ probabilities $\{p_1 = 0.6, p_2 = 0.2, p_3 = 0.1, p_4 = 0.1\}$. Figure 35 illustrates the design procedure.

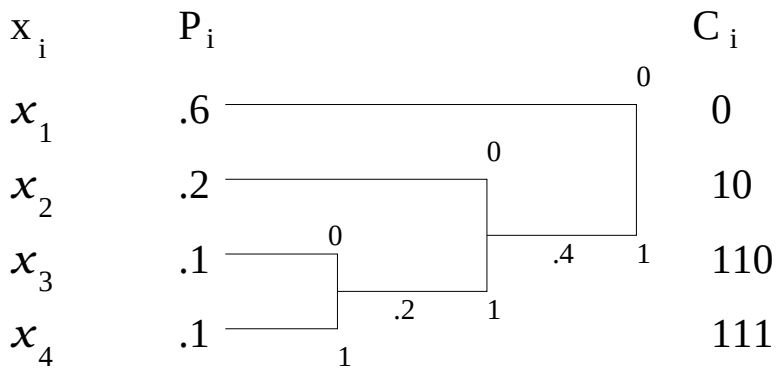


Figure 35: Huffman code design for Example 5.3.

The code words for $\{x_1, x_2, x_3, x_4\}$ are, respectively, $\mathbf{C}_1 = 0$, $\mathbf{C}_2 = 10$, $\mathbf{C}_3 = 110$ and $\mathbf{C}_4 = 111$. The entropy is $H(X) = 1.571$ bits. The code rate is $\overline{R} = 1.6$ bits (within a bit of the entropy), and the efficiency is 98.2 % .

Example 5.4: This example illustrates the Huffman code design procedure when the accumulated probability of some of the lower probability code words is greater than the probabilities of two or more other code words. Consider random variable X with the five value alphabet $\mathcal{A} = \{x_1, x_2, x_3, x_4, x_5\}$ w/ probabilities $\{p_1 = 0.3, p_2 = 0.25,\ p_3 = 0.25, p_4 = 0.1, p_5 = 0.1\}$. Figure 36 illustrates the design procedure.

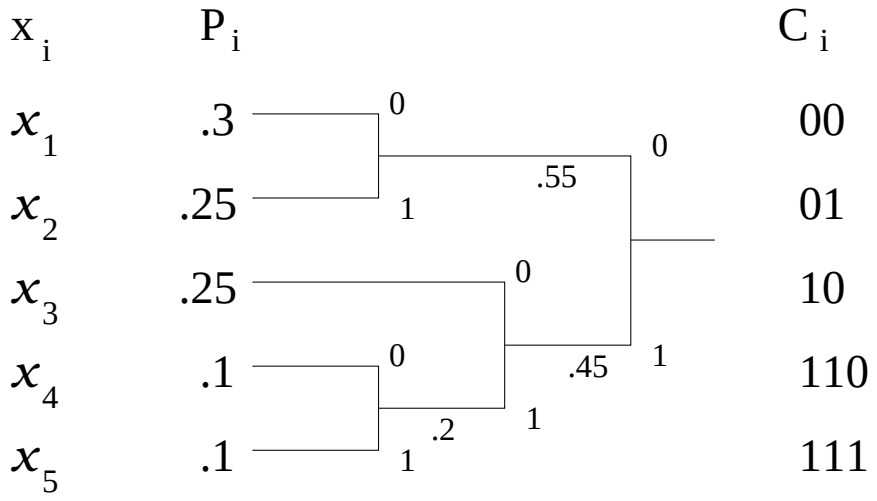


Figure 36: Huffman code design for Example 5.4.

The code words for $\{x_1, x_2, x_3, x_4\}$ are, respectively, $\mathbf{C}_1 = 00$, $\mathbf{C}_2 = 01$, $\mathbf{C}_3 = 10$, $\mathbf{C}_4 = 110$ and $\mathbf{C}_5 = 111$. The entropy is $H(X) = 2.185$ bits. The code rate is $\overline{R} = 2.2$ bits (within a bit of the entropy), and the efficiency is 99.34 % .

Example 5.5: This example suggests that if the values of probabilities are all negative integer powers of 2, the Huffman code is 100 % efficient. Consider random variable X with alphabet $\mathcal{A} = \{x_1, x_2, x_3, x_4, x_5, x_6, x_7, x_8\}$ w/ probabilities $\{p_1 = 2^{-1}, p_2 = 2^{-2}, p_3 = 2^{-4}, p_4 = 2^{-4}, p_5 = 2^{-5}, p_6 = 2^{-5}, p_7 = 2^{-5}, p_8 = 2^{-5}\}$. Figure 37 illustrates the design procedure.

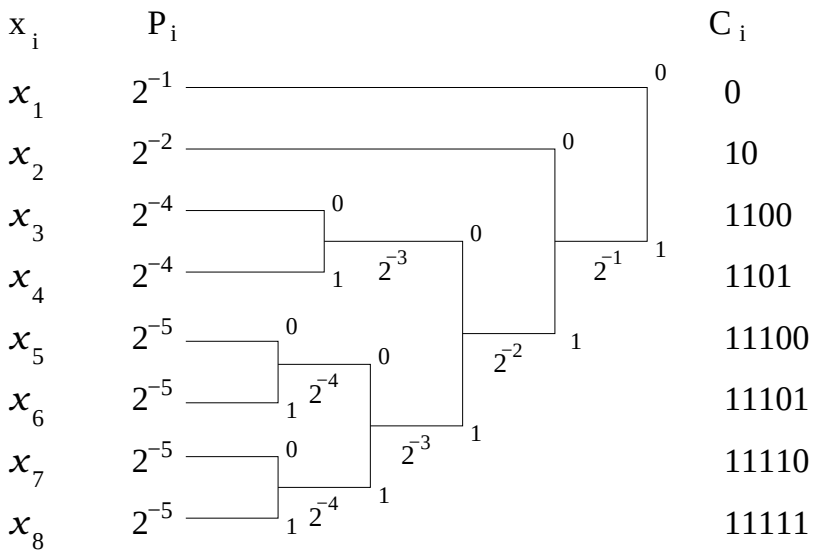


Figure 37: Huffman code design for Example 5.5.

The code words for $x_1, x_2, x_3, x_4, x_5, x_6, x_7, x_8$ are, respectively, $\mathbf{C}_1 = 0$, $\mathbf{C}_2 = 10$, $\mathbf{C}_3 = 1100$, $\mathbf{C}_4 = 1101$ $\mathbf{C}_5 = 11100$, $\mathbf{C}_6 = 11101$, $\mathbf{C}_7 = 11110$ and $\mathbf{C}_8 = 11111$. The entropy is $H(X) = 2.125$ bits, as is $\overline{R}$. Thus the efficiency is 100 % .

Example 5.6: See the Course Text, Example 6.3.1, p. 343. This example shows that for a given random variable X, a Huffman code may not be unique.

Example 5.7: Consider random variable X with five value alphabet $\mathcal{A} = \{x_1, x_2, x_3, x_4, x_5\}$ with probabilities $\{p_1 = 0.4, p_2 = .2, p_3 = 0.2, p_4 = 0.1, p_5 = .1\}$. Figure 38 illustrates the design procedure.

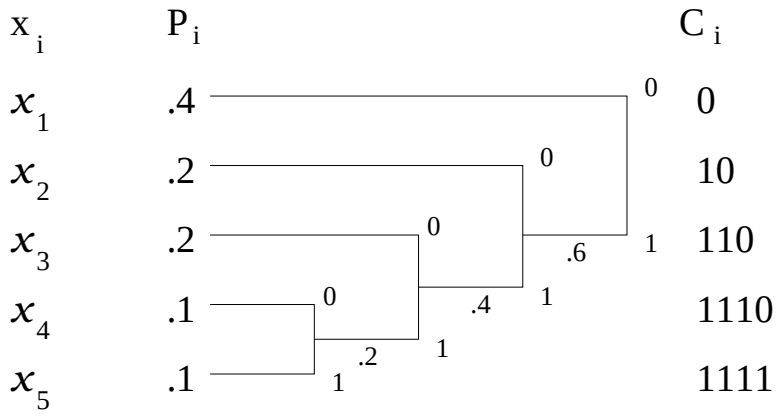


Figure 38: Huffman code design for Example 5.7.

The code words for x_1, x_2, x_3, x_4, x_5 are, respectively, $\mathbf{C}_1 = 0$, $\mathbf{C}_2 = 10$, $\mathbf{C}_3 = 110$, $\mathbf{C}_4 = 1110$ and $\mathbf{C}_5 = 1111$. The entropy is $H(X) = 2.122$ bits. The code rate is $\overline{R} = 2.2$ bits. Thus the efficiency is 96.45 % .

For this example, the code word lengths are $n_1 = 1, n_2 = 2, n_3 = 3, n_4 = 4$ and $n_5 = 4$. Plugging in this lengths into the Kraft inequality, we get

$$2^{-1} + 2^{-2} + 2^{-3} + 2^{-4} + 2^{-4} = 1 \quad . \tag{20}$$

Thus these prefix Huffman code word lengths satisfy the Kraft inequality.

Example 5.8: Finally, consider random variable X with three value alphabet $\mathcal{A} = \{x_1, x_2, x_3\}$ with probabilities $\{p_1 = 0.9, p_2 = .05, p_3 = 0.05\}$. Figure 39 illustrates the design procedure.

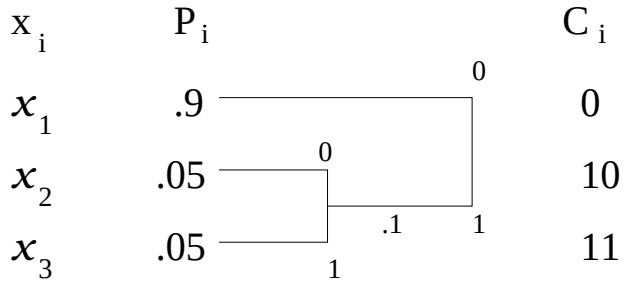


Figure 39: Huffman code design for Example 5.8.

The code words for x_1, x_2, x_3 are, respectively, $\mathbf{C}_1 = 0$, $\mathbf{C}_2 = 10$ and $\mathbf{C}_3 = 11$. The entropy is $H(X) = .3529$ bits (not much). The code rate is $\overline{R} = 1.1$ bits. Thus the efficiency is only 32.08 % .

The code rate is within a bit of the entropy, as expected of an optimum prefix code. However, the efficiency is very low because there are only a small number of values that have highly unequal probabilities.

In summary, we have developed the Huffman code design procedure, and established that a Huffman code is a minimum rate prefix code for encoding a single discrete random variable. Through an example, we have observed that Huffman codes can be inefficient for small alphabets of unequally likely symbols. Also of concern are the facts that: 1) Huffman codes require symbol probabilities, which may not be available; and 2) the Huffman code, applied to each random variable of a sequence in turn, does not take advantage of sequence correlation. We next address these concerns.

5.1.2 Discrete Stationary Source

Given the shortcomings identified above with coding the individual random variables of a discrete sequence, in this Subsection we consider both coding multiple random variables and coding the sequence directly. We begin with a discussion on *prefix codes* and then specifically the *extended Huffman code* for coding a block of random variables. We then overview *arithmetic codes* which are also used to code blocks of random variables. Next, we mention *universal codes* in general, and consider *Lempel-Ziv coding*, which is a form of universal code for a sequence of random variables.

We will restrict our discussion to stationary sources (i.e. stationary random sequences).

Prefix Codes

Consider coding a block of J discrete random variables $\underline{X} = \{X_1, X_2, \cdots, X_J\}$. Denote the joint PDF of $\underline{X}$ as $P(\underline{X})$, and let the joint entropy $H(\underline{X})$ of this block be as defined in Subsection 4.3 of these Notes. If each random variable in $\underline{X}$ takes on values from the alphabet $\mathcal{A} = \{x_1, x_2, \cdots, x_L\}$, then there are L^J possible realizations of the vector $\underline{X}$, with probabilities given by $P(\underline{X})$.

We can consider coding the block $\underline{X}$ using direct extensions of coding algorithms for a single random variable, for example by representing the L^J possible realizations of $\underline{X}$ using variable length code words that meet the prefix condition. We know from our previous discussion on prefix codes that we can find such a code that has rate, denoted $\overline{R}_J$ in bits/block, which is bounded as

$$H(\underline{X}) \;\leq\; \overline{R}_J \;\leq\; H(\underline{X}) \;+\; 1 \quad . \tag{21}$$

If for example the random variables in the block are statistically independent, so that $H(\underline{X}) \;=\; J\; H(X)$ where $H(X)$ is the entropy of each random variable in the block, then we can rewrite this bounding equation as

$$J\; H(X) \;\leq\; \overline{R}_J \;\leq\; J\; H(X) \;+\; 1 \quad , \tag{22}$$

or, in terms of the rate $\overline{R} = \frac{\overline{R}_J}{J}$ in bits/symbol,

$$H(X) \;\leq\; \overline{R} \;\leq\; H(X) \;+\; \frac{1}{J} \quad . \tag{23}$$

Notice how we have tightened the bound on $\overline{R}$, relative to coding individual random variables, in this independent random variable case. Now we are guaranteed that there is a prefix code with a rate that is within $\frac{1}{J}$ of the symbol entropy.

If there is correlation across the random variables in the random sequence, then the improvement can be even more impressive. (See the discussion above in Section 4.4 of these Notes on entropy/sample, $H_\infty(X)$, of a stationary sequence X. Also see the rate bounds for coding a stationary sequence X on p. 346 of the Course Text.)

Extended Huffman Coding

The extended Huffman code (extended to a block of random variables) has the same properties, design procedure and performance tradeoffs as the standard (one random variable) Huffman code. So, the extended Huffman code is optimum in the sense that, for the block it is designed for, no uniquely decodable code can be found that yields lower rate $\overline{R}$. We need only consider prefix codes since if a code is uniquely decodable, then there exists a prefix code with the same code word lengths which is uniquely decodable. The design procedure is exactly the same, as illustrated in the following example.

Example 5.9: Consider a sequence of iid random variables X, with each variable as described in Example 5.8. That is, each has a three value alphabet $\mathcal{A} = \{x_1, x_2, x_3\}$ w/ probabilities $\{p_1 = 0.9, p_2 = .05, p_3 = 0.05\}$. Figure 40 illustrates the design procedure for the coding $J = 2$ variables.

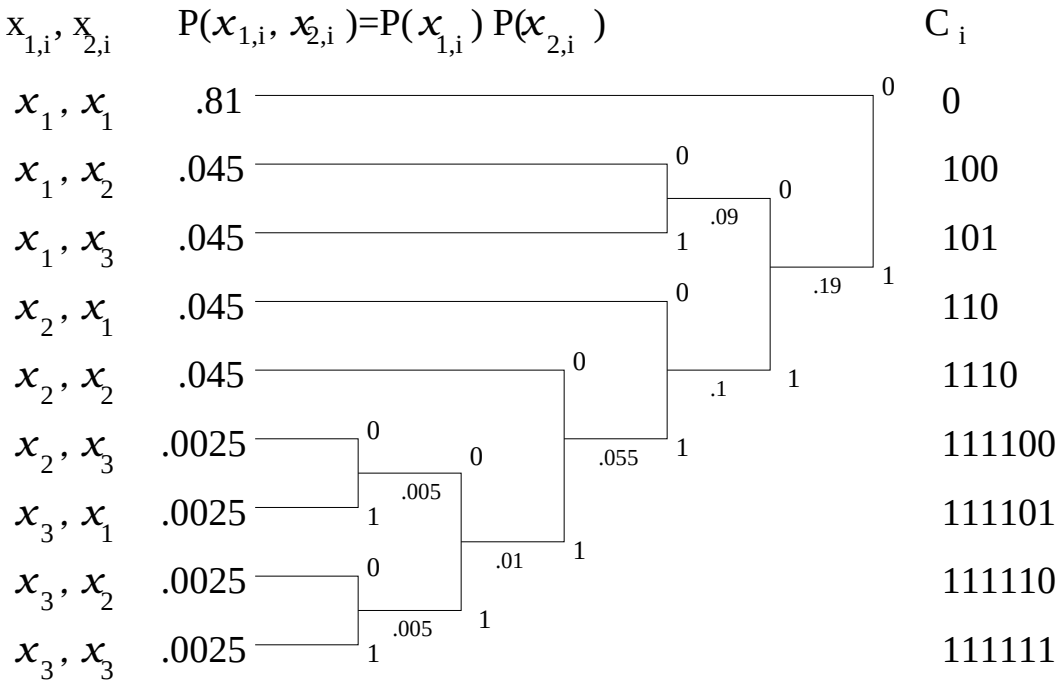


Figure 40: Extended Huffman code design for Example 5.9 – $J = 2$.

The block entropy is $H(\underline{X}) = .7058$ bits which is twice the random variable entropy $H(X) = .3529$ bits. The block code rate is $\overline{R}_2 = 1.455$ bits/block, which corresponds to a code rate of $\overline{R} = \frac{\overline{R}_2}{2} = .7275$. The resulting efficiency is 48.51 % which, although not great, is substantially better than the 32.08 % efficiency obtained from Huffman coding the individual random variables. Even higher efficiency can be realized by coding $J > 2$ blocks.

Example 5.10: In this problem we apply extended Huffman coding to a first-order binary Markov process. Consider a first-order binary Markov process X, where each random variable X_k can take on values from the binary alphabet $\mathcal{A} = \{x_1 = 0, x_2 = 1\}$ and the joint PDF of the first J random variables is

$$P(\underline{X}_J) \;=\; P(X_1, X_2, \cdots, X_J) \;=\; P(X_1) \prod_{k=2}^{J} P(X_k/X_{k-1}) \quad , \tag{24}$$

where the $P(X_k/X_{k-1})$ are conditional PDF's that are determined by the transition probabilities as

$$\begin{aligned} P(X_k/X_{k-1}) \;&=\; P(x_{k,1}/X_{k-1})\delta(x_k - x_{k,1}) \;+\; P(x_{k,2}/X_{k-1})\delta(x_k - x_{k,2}) \\ &=\; \{P(x_{k,1}/x_{k-1,1})P(x_{k-1,1}) + P(x_{k,1}/x_{k-1,2})P(x_{k-1,2})\}\;\delta(x_k - x_{k,1}) \\ &+\; \{P(x_{k,2}/x_{k-1,1})P(x_{k-1,1}) + P(x_{k,2}/x_{k-1,2})P(x_{k-1,2})\}\;\delta(x_k - x_{k,2}) \quad , \end{aligned} \tag{25}$$

where $P(x_{k,i}/x_{k-1,j})$ is the probability that $X_k = x_i$ given that $X_{k-1} = x_j$. The transition from X_{k-1} to X_k is illustrated in Figure 41.

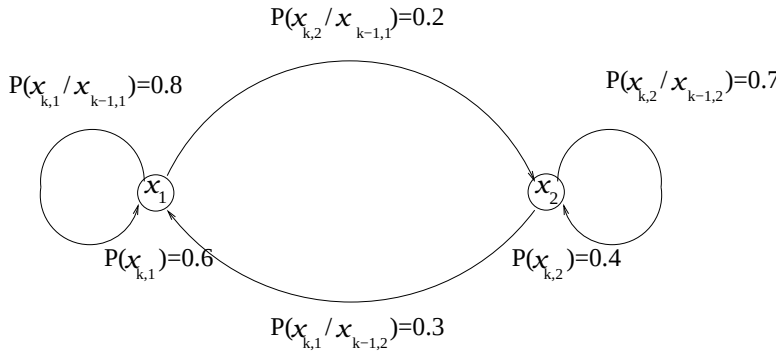


Figure 41: First order Markov process transition diagram.

Consider the extended Huffman code for the $J = 2$ case, for times $k, k-1 = 2, 1$. Let $P(x_{2,1}/x_{1,1}) = .8$, $P(x_{2,2}/x_{1,2}) = .7$, $P(x_{1,1}) = .6$ and $P(x_{1,2}) = .4$. Determine an extended Huffman code for $\underline{X}_2$. Determine the joint entropy $H(\underline{X}_2)$ in bits/block and the entropy "rate" $\frac{H(\underline{X}_2)}{2}$ in bits/symbol. Determine the code rate $\overline{R} = \frac{\overline{R}_2}{2}$. Compare $\overline{R}$ and $\frac{H(\underline{X}_2)}{2}$ to the bit rate for transmitting the bits uncoded. Figure 42. shows the extended Huffman code design for this problem.

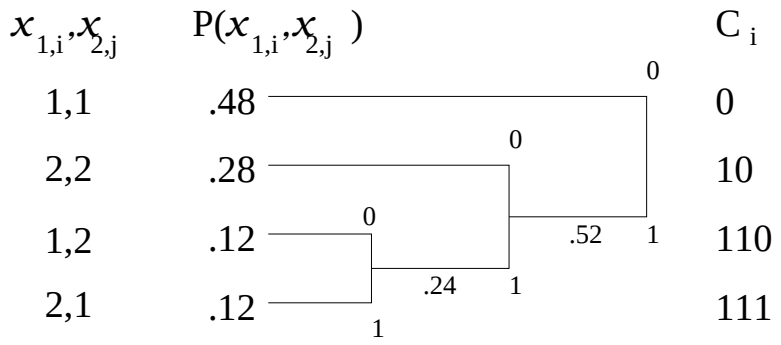


Figure 42: Extended Huffman code design for Example 5.10.

Arithmetic Coding

One problem with extended Huffman coding is that for a small alphabet with highly nonuniform probabilities, a large block size may be required to achieve the required efficiency. For a large block size, all joint probabilities must be identified, and the Huffman code design process dictates that all code words must be designed, regardless on whether or not the symbol sequence corresponding to a code word will actually be encountered. Another problem is that a complete redesign is required if an extended Huffman code is desired for a different block size. Arithmetic coding alleviates these problems. Arithmetic codes can be easily extended to longer block sizes without having to redo calculations, and only required code words need to be computed. Below, a brief overview of arithmetic coding is presented. Reference [6] provides a more extensive introduction, from which much of this summary was extracted.

Consider a block of length J of identically distributed discrete random variables (random variables need not be iid, but this assumption facilitates an easy description). We can think of the arithmetic coding procedure as one in which a *tag* is generated for each block realization to be coded, and this tag is then converted to a code word for transmission. First let's discuss the generation of the tag.

For arithmetic coding, tag generation involves a mapping from block realizations to real numbers in the range [0,1). That is, each unique block realization $\underline{x}_J$ is mapped to a unique tag $T(\underline{x}_J)$ such that $0 \leq T(\underline{x}_J) < 1$. A mapping based on the cumulative distribution function $F(\underline{x}_J)$ is employed. Rather than derive the procedure for generating the tag form the distribution, below we illustrate it with an example.

Example 5.11: Consider a block of discrete random variables $\{X_1, X_2, X_3\}$ each taking on values from the alphabet $\mathcal{A} = \{x_1, x_2, x_3, x_4\}$ with corresponding probabilities $\{p_1 = .5, p_2 = .2, p_3 = .2, p_4 = .1\}$. The first vertical line in Figure 43 illustrates the tag assignment if only X_1 were to be coded. This line represents numbers in the range [0,1) and is divided into intervals, labeled x_i, with lengths proportional the corresponding p_i. The idea is that any number in the x_1 range on the line could be used as a tag for X_1 if $X_1 = x_1$, etc.. Now consider generating the tag for representation of $\{X_1, X_2\}$. Say the $X_1 = x_1$. Then we take the x_1 interval [0,.5) and subdivide it just as we had divided [0,1) previously. The second vertical line in Figure 43 represents this. Now the idea is that any number in the x_1 range on this line could be used as a tag for $X_1 = x_1$ and $X_2 = x_1$, etc.. If instead, $X_1 = x_2$, then we would have instead divided the [.5,.7) range on the first line. Say that $X_1 = x_1$ and $X_2 = x_3$. The third vertical line shows how the $X_1 = x_1$ and $X_2 = x_3$ range, [.35,.45), is divided to represent X_3. So for example, to represent the block realization $\{x_1, x_3, x_1\}$, any number in the range [.35, .4) could be used as a tag.

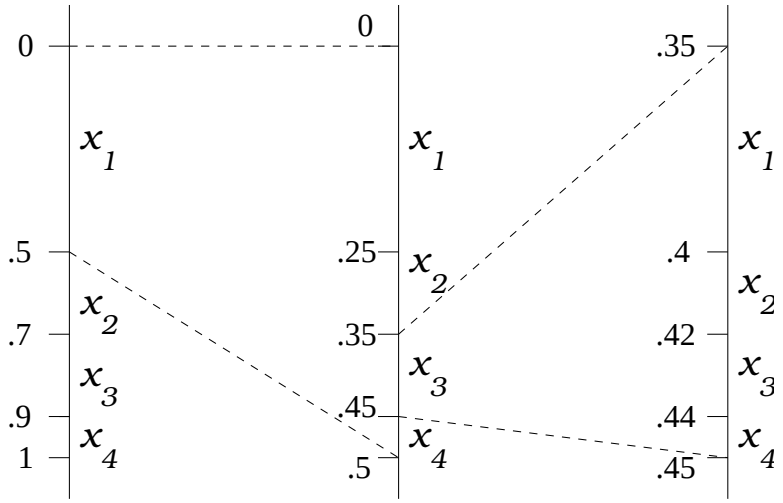


Figure 43: Arithmetic coding tag generation.

Efficient algorithms exists for generating such tags from block realizations and block realizations from tags.

Code words are generated from tags by first representing the tag in binary and then truncating the binary representation.

Example 5.12: Continuing Example 5.11, say tags are generated as the midpoint of the interval representing the block realization. Table 5.1, columns 2 & 3 shows tags and corresponding binary representations for several block realizations of length $J = 3$.

Block realization $\underline{x}$	$T(\underline{x})$	in binary	$P(\tilde{\underline{x}})$	$l(\tilde{\underline{x}})$	code word
$x_1,\ x_3,\ x_1$	.375	.011	.05	6	011
$x_4,\ x_1,\ x_2$	.93	.11101110...	.01	8	11101110
$x_2,\ x_2,\ x_1$	.61	.10011100...	.02	7	1001110

Table 5.1: Tags and binary representation.

The question now is, how many bits of the binary representation do we need to keep to generate code words that are unique (i.e. so that each block realization generates a unique code word)? For the answer, let $l(\underline{x})$ be the length of the code word corresponding to block realization $\underline{x}$. Then, the minimum number of bits is

$$l(\underline{x}) \ = \ \lceil \log \frac{1}{P(\underline{x})} \rceil \ + \ 1 \ , \tag{26}$$

where $\lceil x \rceil$ is the next integer greater than x (i.e. the "ceiling"), and $P(\underline{x})$ is the probability of the block realization $\underline{x}$ that the code word represents.

Example 5.13: Continuing Example 5.12, Table 5.1, columns 4,5 and 6 illustrate code word generation.

We can see form this overview of arithmetic coding that several problems associated with extended Huffman codes have been side stepped. We do not need to generate code words for all possible block realizations. Tags for block length J can be easily updated to give tags for block length $J+1$. However, there are still shortcomings. The major one is that joint PDF's are required, both to identify the tags, as illustrated in Table 5.1, and to identify the code words as indicated by Eq (26). Related to this, performance is sensitive to the accuracy to which the joint PDF is known. Adaptive methods exist for essentially estimating the joint PDF "on the fly", as blocks are being processed.

Given the Course time constraints, the proof of Eq (26) and an efficiency analysis of algebraic codes are beyond the scope of the course. Further consideration of these and other issues related to algebraic coding would make a good Course Project.

Universal Coding

Universal codes are source codes which are not designed for a given probabilistic description of the data to be coded. Thus, they can be applied universally, independent of the source probabilistic characteristics. Basically, universal encoders are composed of a *modeller* which derives a model form the data to be compressed, and a data compression algorithm which uses the derived model in compressing the data. Adaptive arithmetic codes, mentioned directly above, and adaptive Huffman and extended Huffman codes for that matter, can be considered universal codes.

Reference [7], Section 7.4 provides an overview of universal coding. Reference [6], Chapter 5 describes in some detail a general class of universal codes, *dictionary codes*, focusing in particular on several versions of *Lempel-Ziv* codes. Reference [4], Sections 12.3,10 presents a more theoretical treatment of universal coding in general, and Lempel-Ziv codes in particular. A binary version of the Lempel-Ziv source compression algorithm is widely used for computer file compression (e.g. the UNIX "compress" command).

Dictionary methods are applicable to sequences of discrete random variables. They take advantage of recurring patterns in a sequence, by keeping a list or dictionary of frequently occurring patterns, and by then representing one of these patterns found in the sequence by its address in the dictionary. The dictionary can be static, if the frequently recurring patterns are known *a priori*. However, this is contrary to the idea of a universal code. Adaptive dictionary techniques are more useful for general application. Lempel-Ziv coding is based on an adaptive dictionary.

A description and analysis of Lempel-Ziv source coding would make a good Course Project.

5.2 Lossy Coding for Discrete-Time Sources

In Subsection 5.1 we saw that discrete-valued sources can be encoded without loss of information. That is, uniquely decodable encoding (e.g. prefix encoding) is possible. The rate of such a code is lower-bounded by the source entropy. In this Subsection, for discrete-time sources, we establish that if we relax the unique decodability requirement by allowing some loss of information, we can reduce the code rate below the source entropy. Thus, we will see that we can trade-off code rate and information loss. We will consider continuous-valued random sources in general, and a discrete-valued source as a specific example.

For discrete-valued random sources, situations arise in which unique decodability is not a necessity. Perhaps the source is noisy, so that differentiation between adjacent values is not deemed significantly important. Or perhaps some of the possible discrete values are so unlikely that they are not considered worth keeping. Or perhaps the source is binary and some bit errors can be tolerated. In these situations, instead of insisting that the original source can be exactly reconstructed, some specified level of error is tolerated. Code rate can then be further reduced by relaxing the unique decodability requirement.

For digital communication of a continuous-valued random source, some loss of information is unavoidable since we can not represent a continuum of random source values with a countable number of code words. Again we will see that we can trade off code rate and random source distortion. Qualitatively, *distortion* is the difference between the original random source signal and the approximate signal reconstructed (decoded) from the encoded signal.

In information theory the concept of distortion has been formalized. A quantitative measure of distortion has been established and employed to develop a relationship between code rate and a specified level of source distortion. This relationship is called the *rate-distortion function*. In this Subsection we focus on the development of the rate-distortion function, and show how it is used.

Concerning lossy source encoding, we will use the standard term *quantization* in reference to the lossy assignment of a discrete or continuous source value to one of a reduced set of code words. Scalar (one variable) quantization is discussed in Section 6.4 of the Course Text. Other source coding problems, including vector (a block of variables) quantization and temporal and spectral waveform coding, are not discussed in the Course Text. In this Subsection of the Course we will briefly touch on scalar quantization, but we will not cover alternative lossy source encoding methods.

Distortion

Consider a discrete-time random source X, and let $\tilde{X}$ denote an approximation of it. At a given sample time k, let x_k be a realization of X_k. Let $\tilde{x}_k$ be the corresponding realization of $\tilde{X}_k$. We denote as $d(x_k, \tilde{x}_k)$ the measure of the distortion of x_k given $\tilde{x}_k$. Examples are the squared-error (L^2) distortion

$$d(x_k, \tilde{x}_k) \;=\; |x_k - \tilde{x}_k|^2 \quad , \tag{27}$$

the more general L^p distortion ($p = 1, 2, 3, \cdots$)

$$d(x_k, \tilde{x}_k) \;=\; |x_k - \tilde{x}_k|^p \quad , \tag{28}$$

and, specifically for binary sources, the Hamming distance

$$d(x_k, \tilde{x}_k) \;=\; \begin{cases} 0 & \tilde{x}_k \;=\; x_k \\ 1 & \tilde{x}_k \;\neq\; x_k \end{cases} \quad . \tag{29}$$

The distortion measure selected will depend on the application objective, and to some extent on the degree to which the measure can facilitate subsequent derivations.

$d(x_k, \tilde{x}_k)$ is the distortion of a realization of X at time k. Let $\underline{x}_n$ denote a block realization of X over n samples, say $\underline{x}_n = [x_1, x_2, \cdots, x_n]^T$. The distortion between block $\underline{x}_n$ and $\underline{\tilde{x}}_n$ is

$$d(\underline{x}_n, \underline{\tilde{x}}_n) \;=\; \frac{1}{n} \sum_{k=1}^{n} d(x_k, \tilde{x}_k) \quad . \tag{30}$$

Example 5.14: Using the L^2 distance measure, with $\underline{x}_5 = [3.2, -1.5, 0.22, 2.1, -0.01]^T$ and $\underline{\tilde{x}}_5 = [3, -1, 0, 2, 0]^T$, the distortion is

$$d(\underline{x}_5, \underline{\tilde{x}}_5) \;=\; \frac{1}{5} \left((.2)^2 \;+\; (-.5)^2 \;+\; (.22)^2 \;+\; (.1)^2 \;+\; (-.01)^2 \right) \;=\; 0.3485 \quad . \tag{31}$$

Example 5.15: Using the Hamming distance measure, with $\underline{x}_9 = [111111111]^T$ and $\underline{\tilde{x}}_9 = [110110011]^T$, the distortion is

$$d(\underline{x}_9, \underline{\tilde{x}}_9) \;=\; \frac{3}{9} \quad . \tag{32}$$

Now let D be the average or expected value of the distortion,

$$D \;=\; E\{d(\underline{x}_n, \underline{\tilde{x}}_n)\} \;=\; \frac{1}{n} \sum_{k=1}^{n} E\{d(x_k, \tilde{x}_k)\} \quad . \tag{33}$$

If the source random sequence is stationary, and assuming that the same quantization rule is used over the entire block, then

$$D \;=\; E\{d(x_k, \tilde{x}_k)\} \quad . \tag{34}$$

So D is the expected value of the average distortion over time. We also call D *distortion*.

Rate-Distortion Function

Above, we define distortion. We now relate distortion to average mutual information in a way that will lead to bounds on achievable code rate for a given level of distortion. We first consider a rate-distortion measure for the encoding of a single random variable (i.e. the sample-by-sample encoding of a random sequence).

Consider an iid source X and corresponding iid approximation $\tilde{X}$. Let $p(x)$ be the PDF of an X_k. Recall that the average mutual information between a X and its corresponding $\tilde{X}$ is (stated for continuous X)

$$I(X,\tilde{X}) \;=\; I(X_k,\tilde{X}_k) \;=\; \sum_{i=1}^{J} \int_{-\infty}^{\infty} p(x/\tilde{x}_i)\; P(\tilde{x}_i)\; \log\left(\frac{p(x/\tilde{x}_i)}{p(x)}\right)\; dx \quad , \tag{35}$$

where x is the value of X_k, $\tilde{x}_i$ is a value of $\tilde{X}_k$, and J is the number of possible values of discrete-valued $\tilde{X}_k$. (See the Course Text, Eq (6.4-8).) Recall that $I(X_k,\tilde{X}_k) \geq 0$ with $I(X_k,\tilde{X}_k) = 0$ when X_k and $\tilde{X}_k$ are statistically independent. As $\tilde{X}_k$ more closely represents X, the average mutual information increases towards the entropy of X.

The rate-distortion function for a X_k and corresponding $\tilde{X}_k$ is defined as

$$R(D) \;=\; \min_{p(\tilde{x}/x):E\{d(\underline{x}_n,\underline{\tilde{x}}_n)\}\leq D} \; I(X,\tilde{X}) \quad . \tag{36}$$

For a specified level of distortion D, this function gives an information rate $R(D)$ (in bits/output if log base 2 is used). This rate $R(D)$ is the minimum average mutual information over all possible[6] conditional PDF's, $p(\tilde{x}_i/x)$, consistent with the level of distortion D for the employed distortion measure $d(x = x_k, \tilde{x} = \tilde{x}_k)$. By considering the $p(\tilde{x}_i/x)$ that results in minimum $I(X,\tilde{X})$, subject to a selected level of D, we are finding the minimum average mutual information consistent with the distortion constraint. As we will see, this effectively identifies the minimum possible code rate for a given level of distortion. Note that $R(D)$ is a non increasing function of D. That is, as D increases, $R(D)$ tends to decrease (it can't increase), suggesting that if we are willing to accept more distortion, we can achieve a lower code rate.

[6]All possible $p(\tilde{x}_i/x)$ means, for a given $p(x)$, the $p(\tilde{x}_i/x)'s$ for all possible quantization rules (consistent with distortion level D).

Example 5.16: A Gaussian Source Rate-Distortion Function:

Consider a Gaussian random variable X_k with variance σ_x^2. For the L^2 distortion measure, the rate-distortion function, denoted $R_g(D)$, is

$$R_g(D) \;=\; \begin{cases} \frac{1}{2}\;\log\left(\frac{\sigma_x^2}{D}\right) & 0 \le D \le \sigma_x^2 \\ 0 & D > \sigma_x^2 \end{cases} \quad . \tag{37}$$

Proof: See Reference [4], p. 345.

So, if for example we are willing to accept a mean-squared distortion of $D = \frac{\sigma_x^2}{8}$, then the corresponding rate (i.e. the minimum possible code rate), is

$$R_g\left(\frac{\sigma_x^2}{8}\right) \;=\; \frac{1}{2}\;\log(8) \;=\; \frac{3}{2}\;\; bits/symbol \quad . \tag{38}$$

As another example, consider an acceptable distortion of $D > \sigma_x^2$. We have $R(D) = 0$, indicating that we need not send anything, since the decoder assumption $\tilde{x}_k = 0$ will meet the distortion specification.

A Bounds on $R(D)$ Theorem: Let X be any zero-mean continuous-valued source with variance σ_x^2. The rate-distortion function is upper bounded as

$$R(D) \;\le R_g(D) \;=\; \frac{1}{2}\;\log\left(\frac{\sigma_x^2}{D}\right) \quad , \tag{39}$$

and lower bounded as

$$R(D) \;\ge\; R^*(D) \;=\; H(X) \;-\; \frac{1}{2}\;\log(2\pi e D) \quad , \tag{40}$$

where $H(X)$ is the differential entropy of X.

The upper bound is met with equality for a Gaussian source, implying that of all continuous source distributions the Gaussian requires the most bits to represent it to a level of accuracy D. Table 5.2 provides information about these bounds for several common continuous-valued PDF's.

PDF	$p_X(x)$	$H(X)$	$R_g(D) - R^*(D)$ (bits/sample)
Gaussian	$\frac{1}{\sqrt{2\pi\sigma_x^2}}\ e^{-x^2/2\sigma_x^2}$	$\frac{1}{2}\ \log_2(2\pi e\sigma_x^2)$	0
Uniform	$\frac{1}{2\sqrt{3\sigma_x^2}},\ \|x\| \leq \sqrt{3\sigma_x^2}$	$\frac{1}{2}\ \log_2(12\sigma_x^2)$	0.255
Laplacian	$\frac{1}{\sqrt{2\sigma_x^2}}\ e^{-\sqrt{2}\|x\|/\sigma_x}$	$\frac{1}{2}\ \log_2(2e^2\sigma_x^2)$	0.104
Gamma	$\frac{3^{1/4}}{\sqrt{8\pi\sigma_x\|x\|}}\ e^{-\sqrt{3}\|x\|/2\sigma_x}$	$\frac{1}{2}\ \log_2(4\pi e^{0.423}\sigma_x^2/3)$	0.709

Table 5.2: Rate-distortion function bound information for continuous-valued source PDF's.

Example 5.17: A Binary Source Rate-Distortion Function:

Consider a binary random variable X_k with $P(x_i) = p\delta(x_i) + (1-p)\delta(x_i - 1)$. For the Hamming distortion measure, the rate-distortion function, denoted $R_b(D)$, is

$$R_b(D) \ = \ \begin{cases} H(p) \ - \ H(D) & 0 \leq D \leq \min\{p, 1-p\} \\ 0 & D > \min\{p, 1-p\} \end{cases} \quad . \tag{41}$$

where $H(a)$ is the entropy

$$H(a) \ = \ -\ a\ \log(a) \ - \ (1-a)\log(1-a) \ . \tag{42}$$

Proof: See Reference [4], p. 342. Also see the Course Text, p. 354.

So, if $p = 0.5$ and we are willing to accept an average bit error rate of $D = 0.25$, the minimum code rate is $R(0.25) = 0.1887$ *bits/symbol*.

> *A Source Coding with Distortion Theorem:* There exists an encoding scheme that, for allowable average distortion D, will achieve the minimum code rate $R(D)$ given by the rate-distortion function.

Proof: See Reference [4], p. 342, Theorem 13.2.1 and its proof, pp. 353-356.

Above, we have developed the rate-distortion function $R(D)$ for a single random variable. This concludes our overview of distortion as related to source coding.

It turns out that a lower per sample rate can be realized, for a specified per sample distortion level, by coding samples in blocks (see Reference [4], Subsection 13.3.3). The rate-distortion function can be extended to multiple random variables to provide a rate bound for lossy coding of a block of random variables. We will not cover this, and so you are not responsible for it. However, if you are interested, the definition of the rate-distortion function is extended to a block of random variables and to a random process in Proakis, 4-th ed. (of the Course Text), p. 116.

Lossy Codes for Discrete Time Sources - Quantization

Now that we have established that the rate-distortion function $R(D)$ provides a lower bound on the code rate for a specified level of distortion (and that this lower bound is achievable), the logical next issue is how to design a quantizer to achieve this lower bound. Quantizer design is addressed in Subsections 3.4.2 and 3.4.3 of Proakis, 4-th ed. We will not cover either scalar vector quantization (i.e. you are not responsible for it).

5.3 Lossy Coding for Continuous Time Sources

We will not explicitly cover lossy coding of continuous time sources. Note however that, after sampling a continuous time signal to create a discrete time one, the issue of source encoding becomes the discrete time problem that we have just developed a level of understanding of. The issue of sampling is captured by the well known sampling theorem (i.e. lossless sampling is achieved if we sample at a rate greater than or equal to the source bandwidth - the Nyquist rate) and the concept of aliasing (which explains the nature of the information loss when sampling under the Nyquist rate).

6 Channel Capacity & Intro. to Channel Coding

In this Section we employ information theoretic concepts to address the challenge of maximizing the rate of transmission of binary information signals over communication channels. The result will be a quantified measure, called *channel capacity*, of the maximum rate that information can be *reliably* transmitted over a given channel. The term *reliably* specifically means *with arbitrary small error rate*. Capacity will be denoted C. Usually, capacity is expressed in units of *bits per channel use*, but nats per channel use is also common. If the rate of channel use per second is given, then capacity can be specified as a *rate*, for example in *bits/second*.

The definition of channel capacity will be generally applicable. On the other hand, we will apply it to specific channel types. So we will begin this discussion with a description of six channel types of interest. We will then define channel capacity, and consider specific expressions of capacity for these six channel types. Finally we will state as one result two fundamental theorems:

- the noisy channel coding theorem, which states that reliable communication at the channel capacity is possible; and
- the converse noisy channel coding theorem, which states that reliable communication in excess of channel capacity is not possible.

Although formal proofs of these theorems are beyond the scope of this Course (background for the proofs would take too long to develop), we will note aspects of the proof that have motivated developments in channel coding.

Review: average mutual information

Let X and Y be two random variables. For review purposes, assume for now that both are discrete-valued with respective values $\{x_j; j = 0, 1, \cdots, q-1\}$ and $\{y_i; i = 0, 1, \cdots, Q-1\}$. The *average mutual information* of discrete random variables X and Y is the average of the mutual information between the values of X and Y:

$$\begin{aligned} I(X;Y) &= \sum_{j=0}^{q-1} \sum_{i=0}^{Q-1} P(x_j;y_i)\ I(x_j;y_i) \\ &= \sum_{j=0}^{q-1} \sum_{i=0}^{Q-1} P(x_j;y_i)\ \log\left(\frac{P(x_j,y_i)}{P(x_j)P(y_i)}\right) \\ &= \sum_{j=0}^{q-1} \sum_{i=0}^{Q-1} P(x_j)\ P(y_i/x_j)\ \log\left(\frac{P(y_i/x_j)}{P(y_i)}\right) \quad . \end{aligned} \tag{43}$$

Recall from Section 4 that $I(X;Y) \geq 0$, and note the following relationship between average mutual information and entropy:

$$I(X;Y) \ = \ H(Y) \ - \ H(Y/X) \ = \ H(X) \ - \ H(X/Y) \quad . \tag{44}$$

We have already used average mutual information to define a rate-distortion function for lossy source coding. Considering a source X and corresponding approximation $\tilde{X}$, the rate-distortion function is defined as

$$R(D) \;=\; \min_{p(\tilde{x}/x):E\{d(\underline{x}_n,\underline{\tilde{x}}_n)\}\leq D} \; I(X,\tilde{X}) \quad . \tag{45}$$

Recall that for a specified level of distortion D, $R(D)$ is the minimum average mutual information over all possible conditional PDF's, $p(\tilde{x}_i/x)$, consistent with that D. This identifies the minimum possible code rate for a given D, since considering all $p(\tilde{x}_i/x)$ effectively considers all possible codes. In a similar way, we will use the average mutual information between a channel input and output to define channel capacity.

6.1 Channel Models

Figure 44 illustrates a digital communication system. The focus here is on the channel encoder/decoder, the modulator/demodulator and the channel itself. Considering this diagram, different channel models result from different assumptions. If we assume that the modulator/demodulator and the receiver detector are already determined (i.e. from C_k to $\hat{C}_k$ in Figure 44), then the channel (including the modulator/demodulator and the detector) is discrete-time with both discrete-valued input and output. That is, the *channel is discrete*. Different types of discrete channels result from different digital modulation schemes (e.g. numbers of symbols), channel characteristics (e.g. noise PDF, memory), and detector functionality. Alternatively, if we separate the detector from the channel, but include the modulation/demodulator (i.e. from C_k to r_k in Figure 44), then the channel is discrete-time with discrete-valued input and continuous-valued output. If we additionally consider the digital modulation scheme to be required but unspecified (i.e. from I_k to r_k in Figure 44), then each time we use the channel, we can consider it to be discrete-time with both continuous-valued input and output. Ultimately, without any restriction of its use, we can consider the physical channel (i.e. from $s(t)$ to $r(t)$ in Figure 44) to be a *continuous waveform channel* (continuous-time with both continuous-valued input and output).

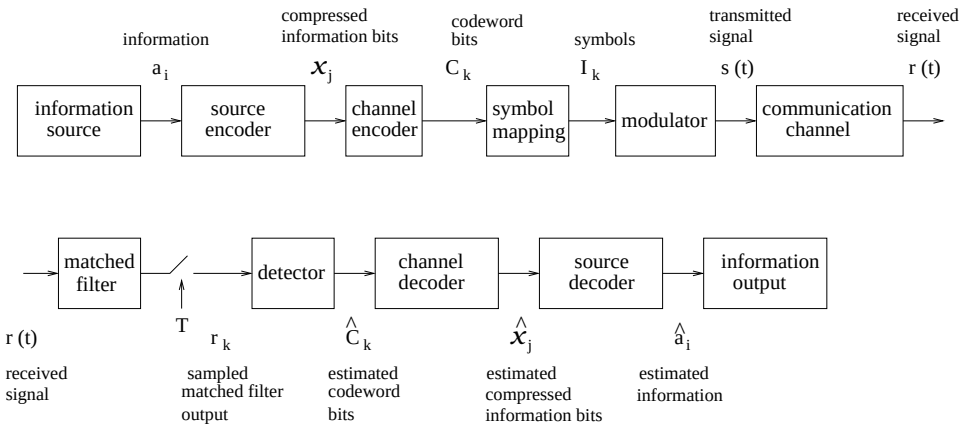


Figure 44: Channel model.

In the examples considered below we assume that the noise is additive, and when it matters it is assumed Gaussian. We focus on memoryless channels, both discrete and waveform. Specifically, we consider the following channel types which are listed in order of complexity:

1. The Binary Symmetric Channel (BSC).
2. The Discrete Memoryless Channel (DMC).
3. The discrete-valued input, additive white Gaussian noise (AWGN) channel.
4. The continuous-valued & power-limited input, AWGN channel.
5. The single-dimensional waveform channel.
6. The band-limited waveform channel. (After a general description, we focus specifically on the additive white Gaussian noise case.)

6.1.1 Binary Symmetric Channel (BSC)

Figure 45 illustrates the BSC model, where X is the binary input and Y is the binary output. Note that this model assumes that the modulation scheme is binary and fixed. The detector is fixed, being designed so that the probability of error is symmetric. That is, denote $P(Y = y_i/X = x_j) = P(0/1)$. Then for the BSC, $P(0/1) = P(1/0) = \rho$ so that $P(1/1) = P(0/0) = (1-\rho)$. The input probabilities, $P(X = 0)$ and $P(X = 1)$, are not specified. The noise PDF is not specified, other than that it, along with the received symbol energy, is consistent with the conditional error probability ρ.

For a discrete channel, the probability transition matrix $\underline{P}$ describes the input/output transition characteristics. The elements of $\underline{P}$ are $\underline{P}_{ji} = P(y_i/x_j)$. Thus,

$$\underline{P} = \begin{bmatrix} (1-\rho) & \rho \\ \rho & (1-\rho) \end{bmatrix} \quad . \tag{46}$$

Note that for the BSC, $\underline{P}$ is symmetric.

For the BSC, the input/output average mutual information is

$$I(X;Y) = \sum_{j=0}^{1} \sum_{i=0}^{1} P(x_j)\ P(y_i/x_j)\ \log\left(\frac{P(y_i/x_j)}{P(y_i)}\right) \quad . \tag{47}$$

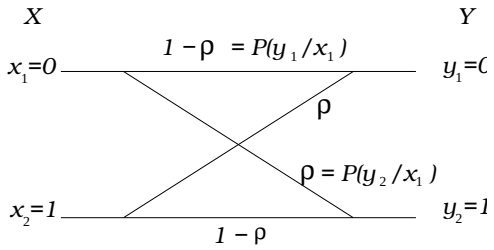


Figure 45: The binary symmetric channel (BSC).

6.1.2 Discrete Memoryless Channel (DMC)

As illustrated in Figure 46, the DMC is a generalization of the BSC to the case where there are q inputs and Q outputs. Numerous examples of DMC's are considered as problems in the back of Chapter 6 of the Course Text, including symmetry and non symmetric binary erasure channels (respectively, Figs. P6.55, (a) and (b)) and the general M-ary symmetric channel (Fig. P6.53). The $q \times Q$ probability transition matrix is

$$\underline{P} = \begin{bmatrix} P(y_1/x_1) & P(y_2/x_1) & \cdots & P(y_Q/x_1) \\ P(y_1/x_2) & P(y_2/x_2) & \cdots & P(y_Q/x_2) \\ \vdots & \vdots & \ddots & \vdots \\ P(y_1/x_q) & P(y_2/x_q) & \cdots & P(y_Q/x_q) \end{bmatrix} . \tag{48}$$

For the DMC, the input/output average mutual information is given by Eq (43).

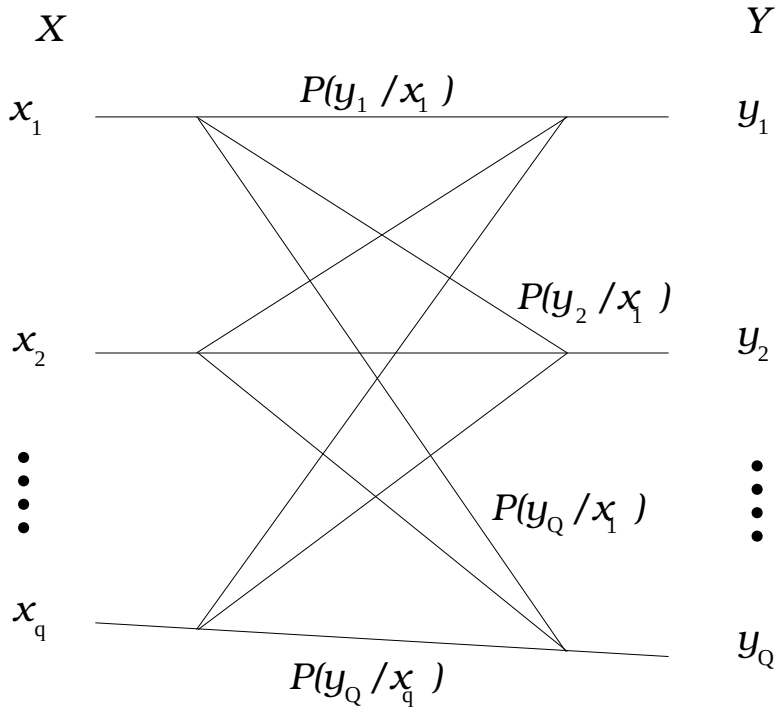


Figure 46: The discrete memoryless channel (DMC).

6.1.3 The Discrete-Valued Input, AWGN Channel

This case is illustrated in Figure 47. The input X is discrete valued, corresponding to possible symbols, and the output is

$$Y \;=\; X \;+\; N \quad , \tag{49}$$

where N is zero-mean AWGN with variance σ_n^2 so that Y is continuous-valued. Y would typically represent a matched filter output after sampling. For q input levels, the average mutual information between X and Y is

$$I(X;Y) \;=\; \sum_{j=0}^{q-1} \int_{-\infty}^{\infty} P(x_j)\; p(y/x_j)\; \log\left(\frac{p(y/x_j)}{p(y)}\right)\; dy \quad . \tag{50}$$

This is Eq (6.4-8) of the Course Text. Here, $p(y/x_j)$ is Gaussian with mean x_j. By the total probability relation, $p(y)$ is a weighted sum of Gaussians (i.e. a Gaussian mixture).

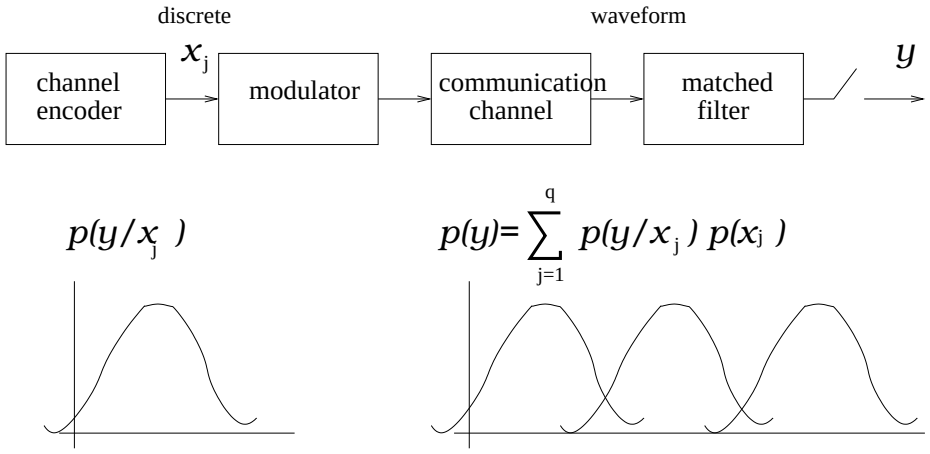


Figure 47: The discrete-valued input AWGN channel.

6.1.4 The Continuous-Valued & Power-Limited Input, AWGN Channel

As with the previous three channel types, in this case we consider one use of the channel. As with the discrete-valued input, AWGN channel just considered, we assume that the channel superimposes AWGN onto the transmitted information signal, i.e.

$$Y \;=\; X \;+\; N \quad . \tag{51}$$

However, now we assume that the input is not restricted to be discrete-valued. Now the average mutual between input X and output Y is

$$I(X;Y) \;=\; \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} p(x)\; p(y/x)\; \log\left(\frac{p(y/x)}{p(y)}\right)\; dy\; dx \quad . \tag{52}$$

For reasons that will be explained in Subsection 6.2.4, we will require that the input be power limited, i.e. $E\{X^2\} \leq P$.

6.1.5 The Single-Dimensional Waveform Channel

Consider a channel with continuous-valued, continuous-time input $X(t);\ 0 \leq t < \tau_s$ and corresponding continuous-valued, continuous-time output

$$Y(t) \ = \ X(t) \ + \ N(t) \ \ . \tag{53}$$

$N(t)$ is assumed additive noise which is statistically independent of $X(t)$. We refer to this as a waveform channel. In general, direct representation in terms of input/output conditional PDF's and average mutual information can be problematic since the input and output are defined over a continuum of time.

Assume that $X(t)$ is rank-1, i.e.

$$X(t) \ = \ X \ f(t) \ \ , \tag{54}$$

where $f(t)$ is a normalized basis function and X is a continuous-valued random variable representing the information. Assume that

$$N^{'}(t) =< f^{*}(t),\ N(t) > \ f(t); \qquad N^{''}(t) \ = \ N(t) \ - \ N^{'}(t) \tag{55}$$

where $N =< f^{*}(t),\ N(t) >$ and $N^{''}(t)$ are independent. (This is essentially the broad-band noise assumption.) Since $X(t)$ is completely specified by X, and N and $N^{''}(t)$ are independent, we can represent $Y(t)$ as

$$Y \ =< f^{*}(t),\ N(t) > = \ X \ + N \tag{56}$$

without loss of information. If $N(t)$ is white Gaussian with spectral level N_0, then N is Gaussian with variance $\sigma_n^2 = \frac{N_0}{2}$.

This channel type is equivalent to the continuous-valued & power-limited input, AWGN Channel type described above in Subsection 6.1.4. Thus Eq (52) expresses the input/output average mutual information.

6.1.6 The Band-Limited Waveform Channel

As in Subsection 6.1.5, consider a waveform channel. We can represent waveforms in terms of orthogonal basis expansions, but they are in general infinite dimensional. However, the effective finite dimensionality property of band-limited/time-limited waveforms leads to a manageable representation of a band-limited waveform channel.

Let W be the one-sided bandwidth of a random waveform (the two-sided bandwidth is assumed to be $2W$), and let τ_s be its duration (e.g. a symbol duration). Then the channel's effective dimension is the time-bandwidth product $N = 2W\tau_s$. It is effective in the sense that if the best basis (in the MSE sense) is used, the amount of signal energy not represented by this basis is an extremely small percentage. Also, as $\tau_s \rightarrow \infty$, this N dimensional representation becomes exact in the MSE sense. (This is the continuous-time Karhunan-Loeve expansion story.)

Assume that the waveform channel is N-dimensional. Consider N dimensional random waveforms $X_N(t)$, $Y_N(t)$ and $N_N(t)$ related as $Y_N(t) = X_N(t) + N_N(t)$, where

$$\begin{aligned} X_N(t) &= \sum_{i=1}^{N} X_i \; f_i(t) \\ Y_N(t) &= \sum_{i=1}^{N} Y_i \; f_i(t) \\ N_N(t) &= \sum_{i=1}^{N} N_i \; f_i(t) \quad , \end{aligned} \tag{57}$$

and the $f_i(t)$ form an orthonormal basis. The input, noise and output expansion coefficients are random variables which are related as

$$Y_i \; = \; X_i \; + \; N_i \; ; \quad i = 1, 2, \cdots, N \quad . \tag{58}$$

Assume that $N_N(t)$ is zero-mean AWGN (i.e. band-limited, but flat over its bandwidth). Then the N_i are statistically independent Gaussian random variables, each with variance $\sigma_n^2 = \frac{N_0}{2}$. For the expansion coefficient Y_i associated with the i^{th} basis function, the PDF given an input value x_i is

$$p(y_i/x_i) \; = \; \frac{1}{\sqrt{2\pi\sigma_n^2}} \; e^{-(y_i - x_i)^2/2\sigma_n^2} \; ; \qquad i = 1, 2, \cdots, N \quad . \tag{59}$$

Eq (59) describes an N dimensional Gaussian waveform channel. Although this N dimensional restriction can not be used to rigorously define the capacity of a general waveform channel, channel capacity derived based on this restriction will serve as a lower bound for the general waveform channel case. This bound will be tight in the sense that $N = 2W\tau_s$ is an effective indication of the dimension of a band-limited channel.

Let $\underline{X}_N = [X_1, X_2, \cdots, X_N]^T$ and $\underline{Y}_N = [Y_1, Y_2, \cdots, Y_N]^T$. In terms of the basis expansion coefficients, the average mutual information is

$$I(\underline{X}_N; \underline{Y}_N) = \int_{X_1} \int_{X_2} \cdots \int_{X_N} \int_{Y_1} \int_{Y_2} \cdots \int_{Y_N} p(\underline{x}) \; p(\underline{y}/\underline{x}) \log \left(\frac{p(\underline{y}/\underline{x})}{p(\underline{y})} \right) dy_N \cdots dy_2 dy_1 dx_N \cdots dx_2 dx_1. \tag{60}$$

If we *assume* that the X_i are statistically independent (e.g. $X_N(t)$ is band-limited with flat spectrum) and Gaussian, we have

$$\begin{aligned} I(\underline{X}_N; \underline{Y}_N) &= \sum_{i=1}^{N} \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} p(x_i) \; p(y_i/x_i) \; \log \left(\frac{p(y_i/x_i)}{p(y_i)} \right) \; dy_i \; dx_i \\ &= \sum_{i=1}^{N} I(X_i; Y_i) \quad . \end{aligned} \tag{61}$$

This is the Average mutual information of N statistically independent single-dimensional waveform channels.

Note that this assumption places a restriction on $X_N(t)$, and therefore this $I(\underline{X}_N;\underline{Y}_N)$ can not be used to rigorously determine channel capacity for a N dimensional waveform channel. It can be shown that, for the non statistically independent case,

$$I(\underline{X}_N;\underline{Y}_N) \leq \sum_{i=1}^{N} I(X_i;Y_i) \quad . \tag{62}$$

(Prob. 6.14 of the Course Text addresses a closely related issue, the difference being that in Prob. 6.14 the random variables are discrete valued.)

6.2 Channel Capacity

For a channel with input X and output Y, the channel capacity is denoted as C. For a discrete-time input channel (i.e. where an input random variable X is transmitted each time the channel is used), capacity is defined as

$$C \;=\; \max_{P(x)} \quad I(X;Y) \quad . \tag{63}$$

The maximum is over all valid input PDF's, and C is the best average information about the input that can be provided at the channel output.

For mutual information in bits (i.e. for $\log = \log_2$ used in the average mutual information expression), capacity is in bits (per channel use). Then, for τ_s^{-1} channel usages/second, $\frac{C}{\tau_s}$ is capacity in bits/second. For a waveform channel, where the input is continuous in both value and time, capacity has a slightly different definition which we will discuss later.

The challenge in determining a channel's capacity C is identifying the input distribution $p(x)$ which results in maximum $I(X;Y)$. Note that the maximization is over all possible input PDF's. So, qualitatively, for a given channel you pick the input that gives the largest input/output average mutual information. This average mutual information is the channel capacity. It remains to be shown that this capacity bounds information transmission in some useful sense. For the time being, it does have intuitive appeal.

We now consider channel capacity expressions for the six channels described above.

6.2.1 The BSC

In this case the input PDF $P(x)$ is discrete and of the form

$$P(x) \;=\; \alpha\;\delta(x) \;+\; (1-\alpha)\;\delta(x-1) \quad , \tag{64}$$

where $0 \leq \alpha \leq 1$. From Eq (47), the average mutual information for the BSC is

$$I(X;Y) \;=\; \alpha\;I(x_0;Y) \;+\; (1-\alpha)\;I(x_1;Y) \tag{65}$$

where

$$I(x_j;Y) = \sum_{i=0}^{1} P(y_i/x_j)\;\log\left(\frac{P(y_i/x_j)}{P(y_i)}\right) \tag{66}$$

with

$$\begin{aligned} P(y_0) &= (1-\rho)\alpha \;+\; \rho(1-\alpha) \\ P(y_1) &= \rho\alpha \;+\; (1-\rho)(1-\alpha) \quad . \end{aligned} \tag{67}$$

Plugging Eqs (66,67) into Eq (65) and setting $\frac{\partial}{\partial\alpha} I(X;Y) = 0$ to find the maximizing α (i.e. the maximizing BSC input distribution), we get $\alpha = 0.5$. The resulting capacity is

$$C \;=\; 1 \;+\; \rho\log(\rho) \;+\; (1-\rho)\log(1-\rho) \;=\; 1 \;-\; H(\rho) \qquad bits, \tag{68}$$

where $H(\rho)$ is the source entropy for a binary source with $P(X=0) = \rho$. Note that for this channel a uniform input distribution corresponds to the channel capacity.

What is Capacity?

To fully understand the significance of channel capacity, we need to formally consider Shannon's *Noisy Channel Coding Theorem*. We will do this in Subsection 6.3 below. However, now that we have an expression for capacity, at least for one channel, we can begin to consider what capacity quantifies.

The dashed curve in Figure 48 shows the channel capacity C, as a function of SNR/bit γ_b, for a 2-PSK modulation scheme with AWGN and a coherent ML (nearest-neighbor) receiver, i.e. a BSC. (Recall that for this communication system, $\rho = Q(\sqrt{2\gamma_b})$.) Considering this curve, note for example that to achieve a capacity of $C = 0.2$ bits per channel use, we need $\gamma_b \approx -6.1\ dB$. Does $C = 0.2$ mean that each time we use the channel we reliably transmit 0.2 bits? What does 0.2 bits mean?

Consider using this channel τ_s^{-1} times per second, where τ_s is the modulation symbol duration. Then, what $C = 0.2$ bits per channel use means, is that we should be able to reliably transmit at an *information rate* of $C\ \tau_s^{-1}$ bits/second. So, if we wish to reliably transmit at an information rate of 1 bit/second, we should be able to do so using this channel at a symbol rate of 5 symbols/second.

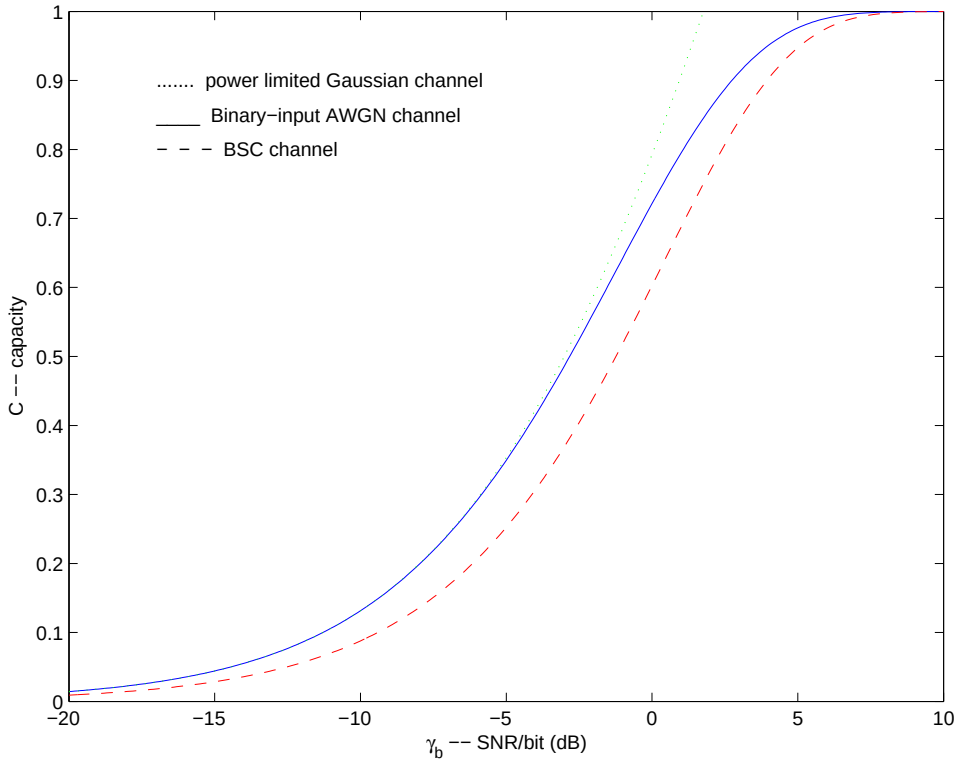


Figure 48: Capacity vs. SNR/bit for several channel models.

As γ_b increases, capacity asymptotes to 1 bit. For a BSC, if no errors are made, 1 bit of information is conveyed per channel usage.

6.2.2 The DMC

For the general DMC, there is no general expression for channel capacity. However there are established results that are useful in identifying channel capacity for any specific DMC. Here we identify two of these results.

Problem 6.52 and Eq (6.5-34) of the Course Text establish the following necessary and sufficient conditions for an input PDF $P(x)$ such that it maximizes $I(X,Y)$ (so as to determine C):

$$\begin{array}{ll} I(x_j;Y) = C & \forall\ j\ s.t.\ P(x_j) > 0 \\ I(x_j;Y) \leq C & \forall\ j\ s.t.\ P(x_j) = 0. \end{array} \tag{69}$$

This says that the $P(x)$ that results in maximum channel input/output mutual information (i.e. the $P(x)$ that determines channel capacity) is the one that balances the mutual informations $I(x_j;Y);\ j = 0, 1, \cdots, q-1$. This does not lead directly to a procedure for determining the maximizing $P(x)$ and thus C, but it does provide a test for checking if a postulated $P(x)$ yields C. On p. 365 of the Course Text the authors imply that a uniform $P(x)$ is a good first choice in determining C. So, for example, consider the DMC's described in Prob. 6.54 of the Course Text. For each, postulate equally probable input symbols, and see what happens. If this test fails to provide the maximizing $P(x)$, then optimization procedures would have to be considered to identifying the maximizing $P(x)$ and resulting C.

On pp. 363-364 of the Course Text, it is noted that a certain channel symmetry condition results in a uniform $P(x)$ maximizer. This symmetry condition can be stated as a row and column permutation property of the probability transition matrix $\underline{P}$.

Properties of Channel Capacity:

Of the channel capacity properties listed below, the second applies only to the DMC case. Properties 1. and 3. are generally applicable.

Properties:

1. $C \geq 0$. This is because $I(X;Y) \geq 0$, and it's a good thing since negative channel capacity isn't useful.

2. For DMC's, $C \leq \min\{\ \log(q),\ \log(Q)\}$. This is because, from Eq. (44), since $H(X/Y) \geq 0$ (i.e. it is an entropy), $C = \max I(X;Y) \leq \max H(X) = \log q$. Similarly, $C \leq \log Q$.

3. C is the only maximum of concave, continuous $I(X;Y)$. That is, $I(X;Y)$ is a continuous, concave function of $P(x)$ (see [4] Theorem 2.7.4). By definition, C is its maximum. Thus, given $I(X;Y)$, C can be found via constrained (i.e. $P(x \geq 0)$, $\sum_j P(x_j) = 1$) gradient search.

6.2.3 The Discrete-Valued Input, AWGN Channel

For the discrete-valued input AWGN channel, with discrete input X and continuous output Y, the average mutual information $I(X;Y)$ is given by Eq (50) above. The capacity is then

$$C \;=\; \max_{P(x)} \quad \sum_{j=0}^{q-1} \int_{-\infty}^{\infty} P(x_j)\; p(y/x_j)\; \log\left(\frac{p(y/x_j)}{p(y)}\right)\; dy \quad . \tag{70}$$

Consider binary antipodal X, with levels $x_0 = A$ and $x_1 = -A$ (e.g. binary PSK). It can be shown that uniformly distributed X yields the capacity, i.e.

$$C = \frac{1}{2}\int_{-\infty}^{\infty} \left\{ p(y/A)\log\left(\frac{p(y/A)}{p(y)}\right) + p(y/-A)\log\left(\frac{p(y/-A)}{p(y)}\right) \right\}\; dy \quad . \tag{71}$$

With $p(y/x_i)$ Gaussian with mean x_i and variance σ_n^2, we get

$$C \;=\; \frac{1}{2}\; g\left(\frac{A}{\sigma_n}\right) \;+\; \frac{1}{2}\; g\left(-\frac{A}{\sigma_n}\right) \tag{72}$$

where

$$g(x) \;=\; \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\infty} e^{(u-x)^2/2}\; \log_2\left(\frac{2}{1+e^{-2ux}}\right)\; du \quad . \tag{73}$$

(See p. 363 of the Course Text). The solid curve in Figure 48 shows the channel capacity C, vs. SNR/bit $\gamma_b = \frac{A^2}{2\sigma_n^2}$, for this channel type. Recall that in Example 4.12 of these Course Notes we explored the average mutual information between the input and output of this binary-input AWGN channel for unquantized and quantized output. We saw that the more the output was quantized, the smaller the input/output average mutual information became. With that result in mind, note that for a given γ_b the capacity of this binary-input AWGN channel is greater than that of the BSC. This should be expected since the BSC is the binary-input AWGN channel with two-level quantization of the AWGN output Y.

Example 6.1: For the three channel considered in Example 4.12 (i.e. the BCS, a DMC, and the discrete-valued input, AWGN channel), compare channel capacities.

Solution: The BSC and the DMC considered in Example 4.12 are symmetric, and so the equiprobable inputs considered for them in Example 4.12 indicate capacity. Also, for the discrete-valued input, AWGN channel considered in Example 4.12 we considered and equiprobable binary input, which we just pointed out yields capacity. Thus, the average mutual informations calculated for Example 4.12 are the channel capacities. That is:

1. for the BSC, $C \;\approx\; 0.9031$;
2. for the DMC, $C \;\approx\; 0.9203$; and
3. for the discrete-valued input, AWGN channel, $C \;=\; 0.94$.

We should expect these relative capacities since in going from the discrete-input AWGN channel to the DMC to the BSC, we have a greater degree of receiver matched-filter output quantization.

6.2.4 The Continuous-Valued & Power-Limited Input, AWGN Channel

In Subsection 6.2.3 we considered a channel model for which the input was a discrete-valued random variable and the output was the input superimposed with AWGN. We assumed given input levels, and considered the capacity in bits per channel use. We identified a formula for capacity specifically for binary antipodal input. Now we extend these results by allowing the input to be a continuous-valued. We can expect capacity to continue to improve as we increase the power of the continuous-valued random variable input, since the average mutual information between two continuous-valued random variables is in general not upper bounded. Thus, we will bound the result by constraining the input to have power $E\{X^2\} \leq P$.

The capacity for this type of channel is the solution to

$$\begin{aligned} \max_{p(x)} \quad I(X;Y) &= \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} p(x)\ p(y/x)\ \log\left(\frac{p(y/x)}{p(y)}\right)\ dy\ dx \\ &= H(Y)\ -\ H(Y/X)\ , \end{aligned} \tag{74}$$

where $H(Y)$ is the differential entropy of Y and $H(Y/X)$ the conditional differential entropy of Y given X. (The last line of Eq (74) is the average mutual information, entropy relation established in Subsection 4.2.2 of the Course Notes.) With $Y = X + N$, and assuming the N is statistically independent of X, we have that[7] So, to identify channel capacity, we need to solve

$$\max_{p(x)} \quad I(X;Y)\ =\ H(Y)\ -\ H(N)\ . \tag{75}$$

It can be shown (see [4], Theorem 8.6.5) that the Gaussian distribution maximizes entropy over all distributions with the same variance. Thus Gaussian $p(y)$ and therefore $p(x)$ solves Eq (74). Let $\sigma_x^2 = P$ be the variance of Gaussian X, so that $\sigma_y^2 = P + \sigma_n^2$ is the variance of Gaussian Y. Recall from Example 4.5 of these Course Notes that for Gaussian N, $H(N) = \frac{1}{2}\log_2(2\pi e\sigma_n^2)$. The capacity is then given by

$$C = H(Y)\ -\ H(N) \tag{76}$$

$$= \frac{1}{2}\ \log_2(2\pi e(P+\sigma_n^2))\ -\ \frac{1}{2}\ \log_2(2\pi e\sigma_n^2) \tag{77}$$

$$= \frac{1}{2}\ \log_2(1+\frac{P}{\sigma_n^2}) \qquad (\textit{bits per channel use}) \quad . \tag{78}$$

[7] For $Y = X + N$,

$$\begin{aligned} H(Y/X) &= -\int\int\ p(x,y)\ \log(p(y/x))\ dy\ dx \\ &= -\int\ p(x)\ \left\{\int\ p(y/x)\ \log(p(y/x))\ dy\right\}\ dx \\ &= -\int\ p(x)\ \left\{\int\ \mathcal{N}(x,\sigma_n^2)\ \log[\mathcal{N}(x,\sigma_n^2)]\ dy\right\}\ dx \\ &= -\int\ p(x)\ -H(N)\ dx\ =\ H(N)\ . \end{aligned}$$

The dotted curve in Figure 48 shows the channel capacity C, vs. SNR/bit $\gamma_b = \frac{P}{2\sigma_n^2}$, for this channel type. Notice that for relatively low SNR the capacities of the binary-input AWGN channel and this continuous-valued input channel are about the same. However, capacity for this continuous-valued input channel is not bounded by 1 bit. Instead, as SNR/bit increases beyond about -2dB, capacity increases without bound at a rate of about 1 bit per 6dB of SNR.

6.2.5 The Single-Dimensional Waveform Channel

In Subsection 6.1.5 it has been established that, under the assumptions stated in that Subsection, the average mutual information of a single-dimensional waveform is the same as for a continuous-valued & power-limited input, AWGN channel. Thus, for the single-dimensional waveform channel, Eq (78) is the capacity in bits per channel use. With the continued use of this waveform channel, the capacity in bits/second is

$$C \;=\; \frac{1}{2\;\tau_s}\;\log_2(1+\frac{P}{\sigma_n^2}) \qquad bits/second \quad , \tag{79}$$

where as before τ_s is the symbol (channel use) duration.

6.2.6 The Band-Limited AWGN Waveform Channel

In Subsection 6.1.6 we established that the average mutual information of a channel of bandwidth $2W$ used over duration τ_s with AWGN is bounded above by the total average mutual information of $N = 2W\tau_s$ single-dimensional waveform channels. Equality to the bound is realized when the input waveform $X(t)$ is white and Gaussian (so that the projections of the information signal $X(t)$ onto the basis functions, i.e. the X_i, are statistically independent.) Thus the capacity of this band-limited AWGN waveform channel will be $N = 2W\tau_s$ times that of the single-dimensional waveform channel.

For this band-limited channel, say that the noise spectral level is N_0 so that its power per basis dimension is $\sigma_n^2 = \frac{N_0}{2}$. Let the total input power be P, so that the power per bandwidth is $\frac{P}{2W}$. Then the total capacity, for all the single-dimensional channels combined, is

$$C \;=\; W\ \tau_s\ \log_2(1+\frac{P}{2W\sigma_n^2}) \qquad (bits\ per\ channels\ use) \quad , \tag{80}$$

or

$$\begin{aligned} C &= W\ \log_2(1+\frac{P}{2W\sigma_n^2}) \qquad (bits/second) &(81)\\ &= W\ \log_2(1+\frac{P}{N_0\ W}) \qquad (bits/second) \ . &(82) \end{aligned}$$

Comments on capacity for the band-limited AWGN waveform channel:

Consider Eq (82) – the capacity of a band-limited AWGN waveform channel. Since the noise is white with spectral level $\frac{N_0}{2}$ and bandwidth $2W$, the SNR is $\frac{P}{WN_0}$. So, the capacity is a simple function of bandwidth W and SNR $\frac{P}{WN_0}$. Capacity increases with both bandwidth and SNR. This has intuitive appeal. However the behavior of C for increasing W is more involved than it appears because, for fixed signal power, SNR decreases as W increases. Figure 50 shows C vs. W. Note that as $W \to \infty$, C approaches the asymptotic value

$$\begin{aligned} C_\infty &= \lim_{W\to\infty}\ W\ \log\left(1+\frac{P}{WN_0}\right) \;=\; \lim_{W\to\infty}\ W\ \log(e)\ln\left(1+\frac{P}{WN_0}\right)\\ &= \lim_{W\to\infty}\ W\ \log(e)\left(\frac{P}{WN_0}+\cdots\right)\\ &= W\ \log(e)\frac{P}{WN_0} \;=\; \log(e)\ \frac{P}{N_0} \;\approx 1.44\ \frac{P}{N_0} \quad . \end{aligned} \tag{83}$$

So, increasing W has diminishing returns.

In Subsection 3.3 of the Course we briefly considered bandwidth efficiency $\frac{R}{W}$ in bits/sec/Hz, where R is bit rate. The ratio $\frac{C}{W}$ is capacity, normalized by bandwidth, which also has units bits/sec/Hz. Of interest was bandwidth efficiency vs. SNR/bit $\gamma_b = \frac{\mathcal{E}_b}{N_0}$. In terms of γ_b,

$$\frac{C}{W} \;=\; \log\left(1+\frac{C}{W}\gamma_b\right) \quad bits/sec./Hz \quad . \tag{84}$$

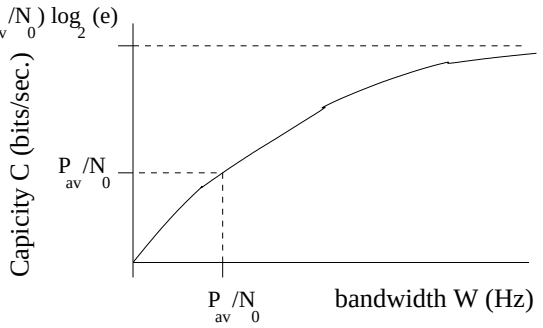


Figure 49: C vs. W for the band-limited AWGN waveform channel.

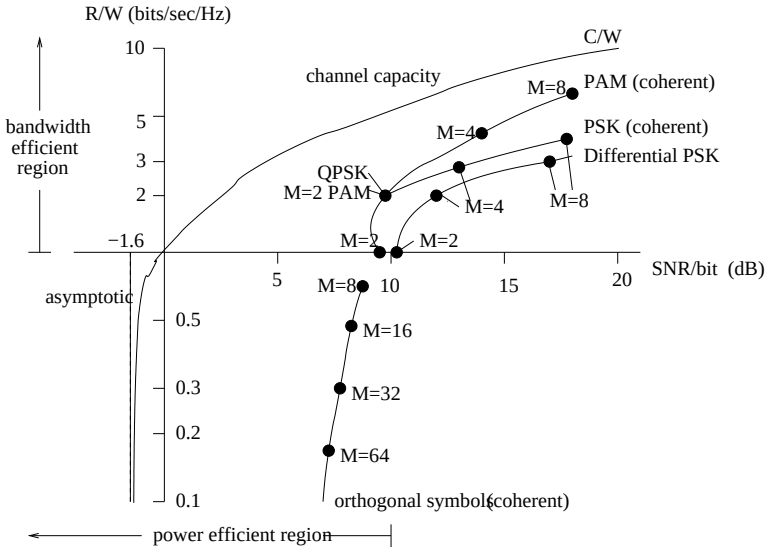


Figure 50: Bandwidth efficiency vs. SNR/bit.

From this it can be shown (Course Text, p. 367) that as $\frac{C}{W} \to 0$, $\gamma_b \to \ln 2 = -1.6dB$. Figure 50 (i.e. Figure 4.6-1 of the Course Text) contains a plot of $\frac{C}{W}$ vs. γ_b to support comparison of modulation schemes with channel capacity.
As pointed out on p. 367 of the Course Text, -1.6dB is the minimum SNR/bit possible for reliable communications. For reliable communications as SNR/bit lowers towards -1.6dB, we need bandwidth $W \to \infty$.

6.3 Relating Channel Capacity and Digital Communication Performance

Above we defined channel capacity C and derived expressions for it for several channels of interest. We argued that C, the average channel input/output mutual information, maximized over all input PDF's, has intuitive appeal. However, to this point in our discussion no specific relationship between C and communication performance has been established. In this Subsection we consider the relationship between capacity C, bit rate R and symbol error probability P_e.

We consider M-ary orthogonal modulation, for large M, to establish a key result for the AWGN waveform channel. As a generalization of this discussion to any channel with capacity C, we then introduce without proof a fundamental concept of information theory, the *noisy channel coding theorem*.[8] A key feature of the proof is then identified that has motivated extensive research into channel code design.

We start by restating a result from Section 3.3 of this Course concerning the performance of the M-ary orthogonal modulation scheme in AWGN. In that Subsection we first presented an expression for probability of symbol error P_e. We then pointed to a simple upper bound on P_e derived using the union bound on error probabilities and the Chernov bound on Gaussian probability tails. This SEP bound is,

$$P_e \; < \; e^{-k(\gamma_b - 2\ln(2))/2} \quad , \tag{85}$$

where $k = \log M$ is the number of bits represented by M orthogonal symbols. As noted earlier, Eq (85) shows that as $M \rightarrow \infty$, and correspondingly $W \rightarrow \infty$, P_e can be made arbitrarily small as long as the SNR/bit, γ_b, satisfies

$$\gamma_b > 2\ln(2) = 1.42dB \quad . \tag{86}$$

This type of result is important. It tells us that in theory we can assure reliable digital communication (i.e. communication with as few symbol errors as desired) as long as $\gamma_b > 1.42dB$ (in this case by using a very high dimensional, orthogonal modulation scheme for the AWGN waveform channel). The fact that this modulation scheme requires a very high bandwidth is a concern, but nonetheless the result is encouraging. For example, consider the modulation performance analyzes in Subsection 3.3 of the Course. To achieve a decent symbol error probability, say $P_e = 10^{-5}$, using a typical modulation scheme such as PAM of QPSK, an SNR/bit γ_b of on the order of 10dB is required. Eq (85) suggests that we might be able to do substantially better than this.

This performance bound, Eq (85), is loose in the sense that the P_e is actually substantially lower than the bound indicates, especially at low γ_b and large M. That is, the $\gamma_b > 1.42dB$ limit, which assures $P_e = 0$ is achievable, is pessimistic. A tighter bound, derived in Subsection 6.6 of the Course Text, is

$$P_e \; < \; \begin{cases} 2e^{-k(\gamma_b - 2\ln(2))/2} & \ln(M) \leq \frac{k\gamma_b}{4} \\ 2e^{-k(\sqrt{\gamma_b} - \sqrt{\ln(2)}\,)^2} & \frac{k\gamma_b}{4} \leq \ln(M) \leq k\gamma_b \end{cases} \quad . \tag{87}$$

[8] For a proof and in-depth discussion of Shannon's noisy channel coding theorem, see Cover and Thomas [4] pp. 191-203. Although given the background assumed and developed in this Course this proof is accessible, its coverage would consume time that we will use instead to study coding methods.

Besides being a tighter bound on P_e, this expression is of interest because it can be rewritten in terms of the asymptotic capacity C_∞, the bit rate R, and the symbol duration T, as

$$P_e \; < \; \begin{cases} 2 \cdot 2^{-T(\frac{C_\infty}{2} - R)} & 0 \leq R \leq \frac{C_\infty}{4} \\ 2 \cdot 2^{-T(\sqrt{C_\infty} - \sqrt{R})^2} & \frac{C_\infty}{4} \leq R \leq C_\infty \end{cases} \quad . \tag{88}$$

This form of the bound shows that we can make P_e arbitrarily small, by increasing $T = \frac{log_2 M}{R}$, as long as $R < C_\infty$.

In summary, for AWGN waveform channels at least, reliable digital communication can be achieved using high dimensional M-ary orthogonal modulation (i.e. $M \rightarrow \infty$ and $T \rightarrow \infty$) as long as $R < C_\infty$.

6.3.1 The Noisy Channel Coding Theorem

> There exists codes and decoders that achieve reliable digital communication if $C > R$. If $R > C$, reliable digital communication is not possible.

Note that unlike the results stated above for AWGN waveform channels, no assumption is made on the channel type.

Proofs of this theorem (i.e. the derivation of the results) typically make use of very long random code words. That is, very long symbol blocks are randomly assigned to be represented by an even longer binary code words. Then, if $C > R$, as the length of the symbol blocks and code words go to infinity, some realizations of the random code word assignments will result in zero probability of being incorrectly decoded.

An important characteristic of this proof is that it employs very long (infinitely long) code words. Another important point is that this proof is effectively nonconstructive. That is, it does not point to a reasonable set of codes that can be used to realize channel capacity. Infinitely long code words must be generated and tested. Motivated by this proof, extensive research has been devoted to developing very long code word codes which can be generated and decoded in a computationally efficient manner. The state-of-the-art long codes are turbo codes and Low Density Parity Check (LDPC) codes.

References

[1] C. Shannon, "A mathematical theory of communications," *Bell Systems Tech. Jour.*, vol. 27, pp. 379–423, 623–656, July, October 1948.

[2] S. Verdu, "Fifty years of shannon theory," *IEEE Trans. on Info. Theory*, vol. 14, pp. 2057–2078, October 1998.

[3] P. Peebles, *Probability, Random Variables and Random Signal Principles, 4-th ed.* McGraw Hill, 2001.

[4] T. Cover and J. Thomas, *Elements of Information Theory, 2-nd ed.* Wiley Interscience, 2006.

[5] R. Gallagher, *Information Theory and Reliable Communications.* Wiley, 1968.

[6] K. Sayood, *Introduction to Data Compression.* Morgan Kaufmann, 1996.

[7] R. Blahut, *Digital Transmission of Information.* Addison-Wesley, 1990.