ESE 471
Communications Theory and Systems
Lecture Notes

Spring 2021

Dr. Neal Patwari
Washington University in St. Louis
McKelvey School of Engineering

Contents

Lecture 1

Today: (1) Syllabus (2) Intro to Digital Communications

1 About My Notes

I type my lecture notes. My previous version of these lecture notes for my entire Spring 2020 course is on Canvas. I will be updating my notes each time I give a lecture, even up to the lecture time. I will post my updated lecture notes online, which you can use after lecture to help you follow the lecture. However, you must accept these conditions:

1. **Taking notes is important**: I find most learning requires your active recording, not just watching. Please take your own notes. I have a big right margin on my notes, so you have room to copy your own notes next to mine.

2. **My written notes do not and cannot reflect everything said during lecture**: I answer questions and understand your perspective better after hearing your questions, and I try to tailor my approach during the lecture. If I didn't, you could just watch a recording.

2 Introduction

A digital communication system repeatedly conveys discrete-valued information across a physical (analog, noisy, attenuating) channel. The mismatch between the digital information and the analog channel and the loss and randomness introduced by the channel, are the primary challenges for the design of reliable and efficient digital communications systems. This is the challenge that you'll learn to address in this class. You learn in circuits how to design a matching circuit for maximum power transfer, You want, for power efficiency, to have the source impedance match the destination impedance. This course teaches you how to design the "matching network" to most efficiently send and receive digital information across an analog channel which we cannot really change.

What is an information source? It might be audio, video, text, or data. It might be a continuous-time (analog) signal (audio, images) and it might be 1-D or 2-D. Or, it may already be digital (discrete-time, discrete-valued). Our object is to convey the signals or data to another place (or time) with as faithful of a representation as possible.

In this section we talk about what we'll cover in this class, and more importantly, what we won't cover.

2.1 Executive Summary

Here is the one sentence version: **We will study how to efficiently encode digital data on a noisy analog medium, without interfering with other signals sent on the same medium, so that decoding the data (*i.e.*, reception) at a receiver is simple, efficient, and high-fidelity.**

The keys points stuffed into that one sentence are:

1. Digital information on an analog medium: We can send analog signals, *i.e.*, real-valued, continuous-time functions, on the medium. The medium is, for example, EM waves on the

air, EM waves in an optical fiber or metal wire, or magnetic field on a drive. It is also called a "channel". The signals sent on the medium are composed of waveforms. In particular, we will send **linear combinations of orthogonal waveforms**. In digital communications systems, there are a discrete list of possible linear combinations we might send, one for each possible symbol. We will discuss why and what linear combinations, and what orthogonal waveforms are and why they are useful.

2. Decoding the data: When receiving a signal (a function) in noise, it will not match exactly what was sent. How should a receiver make a decision about which signal was sent?

3. What makes a receiver difficult to realize? What choices of waveforms make a receiver simpler to implement? What techniques are used in a receiver to compensate?

4. Efficiency, Bandwidth, and Fidelity: Fidelity is the correctness of the received data (*i.e.*, the opposite of error rate). What is the tradeoff between energy, bandwidth, and bit error rate? We want all three to be low (low consumption, low bandwidth usage, and low bit error rate). Energy and bandwidth are the costs of our communication system, and fidelity is how well it performs.

2.2 Why not Analog Information?

Analog systems still exist and will continue to exist; however, new systems you will design during your career will almost always be digital communication systems. Why?

- Fidelity: Analog systems will always have noise in the received signal. Digital communications systems can be designed to achieve an arbitrarily low error rate.
- Energy: transmit power, and device power consumption
- Bandwidth efficiency: digital communications systems can be significantly more efficient in use of the spectrum than older analog systems.
- Moore's Law decreases device costs for digital hardware
- Increasingly the information needed to be transferred is fundamentally digital
- More powerful information security is possible

2.3 Networking Stack

In this course, we study digital communications from bits to bits. That is, we study how to take ones and zeros from a transmitter (TX), send them through a medium, and then (mostly) correctly identify the same ones and zeros at the receiver (RX). There's a lot more than this to the digital communication systems which you use on a daily basis (*e.g.*, cellular phone, Wi-Fi, Bluetooth, wireless keyboard, wireless car key).

To manage complexity, we (engineers) don't try to build a system to do everything all at once. We typically start with an application, and we build a layered network to handle the application. Each layer has particular inputs and outputs that are standardized so that we can change the operation of one layer (ideally) without change to the other layers. The 5-layer OSI stack, which you would study in a computer networking class, names the layers as:

- Application Layer
- Transport Layer
- Network Layer
- Link Layer
- Physical (PHY) Layer

(Note that there is also a 7-layer model in which application layer is further separated.) ESE 471 is part of the bottom layer, the physical layer. In fact, the physical layer has much more detail. It is primarily divided into:

- Multiple Access Control (MAC)
- Encoding
- Channel / Medium

The MAC layer ensures that users who are on the same channel don't interfere too much with each other. The encoding layer is how the transmitter and receiver agree to send digital bits, and the choice of the channel / medium, as we talked about, both enables the signal to arrive at the receiver, but degrades the signal in random ways (noise, filtering) as discussed above.

Our class covers part of the MAC layer, and focuses on the Encoding and the Channel / Medium layers.

2.4 Channels and Media

As communication system designers, we can chose from a few media, but we largely can't change the properties of the channel. That's why I'm going to start by discussing the problems of the channel before I discuss how we can "match" our transmitter and receiver to the channel.

Here are some example channels:

- Wireless EM Media: (anything above 0 Hz) Radio "Microwave", mm-wave, light
- Acoustic: ultrasound
- Wired EM Media: Transmission lines, waveguides, optical fiber, coaxial cable, wire pairs, ...
- Disk (data storage applications)

A channel poses particular problems:

1. Noise and interference: Anything transmitted into the channel adds together, so other devices using the same channel have their signals also received by our receiver, which we call *interference* if we don't want that signal. There is thermal noise in every channel, due to the physics of living above 0 Kelvin. It is typically called *additive noise*.
2. Attenuation: There is typically many orders of magnitude of reduction of the transmit power between the transmitter and receiver. For example, the power in the signal may be multiplied by 10^{-10} for a typical cellular communication channel. It can be successfully received even when it is received with on the order of the same power as the additive noise and interference.

3. Frequency dependence / multipath: The linear filtering of the channel results from the physics and EM of the medium. For example, attenuation in telephone wires varies by frequency. Narrowband wireless channels experience fading that varies as a function of frequency. Wideband wireless channels display multipath, due to multiple time-delayed reflections, diffractions, and scattering of the signal from objects in the environment. We often model a channel as a linear filter, although the impulse response is random and unknown ahead of time.

4. Doppler: When the transmitter and receiver (or objects in the channel) are moving, the channel filter response changes as a result. One effect is the Doppler effect, that the transmitted signal changes frequency as it goes through the channel. In this case, case our channel is neither linear nor time-invariant.

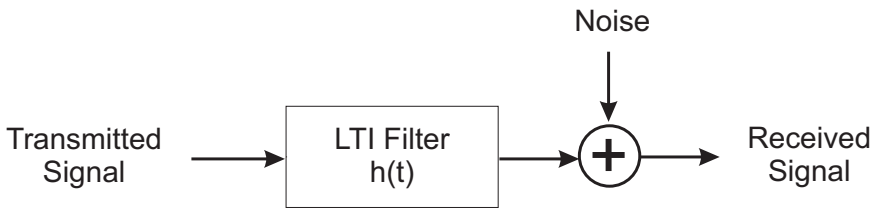


Figure 1: Linear filter and additive noise channel model.

In this course, we will focus primarily on the additive noise channel, but we will mention what variations exist for particular channels, and how they are addressed.

2.5 Topics in ESE 471

2.5.1 Topic: Random Processes

What are things that can be considered random in a communication system?
Solution: Random things in a communication system:

- Noise in the channel
- Signal (bits)
- Channel filtering, attenuation, and fading
- Device frequency, phase, and timing offsets between transmitter and receiver

We want to build the best receiver possible despite the random impediments. Optimal detection and optimal receiver design is something that we study using probability theory.

We have to tolerate errors. Noise and attenuation of the channel will cause bit errors to be made by the demodulator and even the channel decoder. This may be tolerated, or a higher layer networking protocol (*e.g.*, TCP) can determine that an error occurred and then re-request the data.

2.5.2 Topic: Signals, Signal Space, and Bandwidth

To show that signals sharing the same channel don't interfere with each other, we need to show that they are *orthogonal*. This means, in short, that a receiver can uniquely separate them. Signals

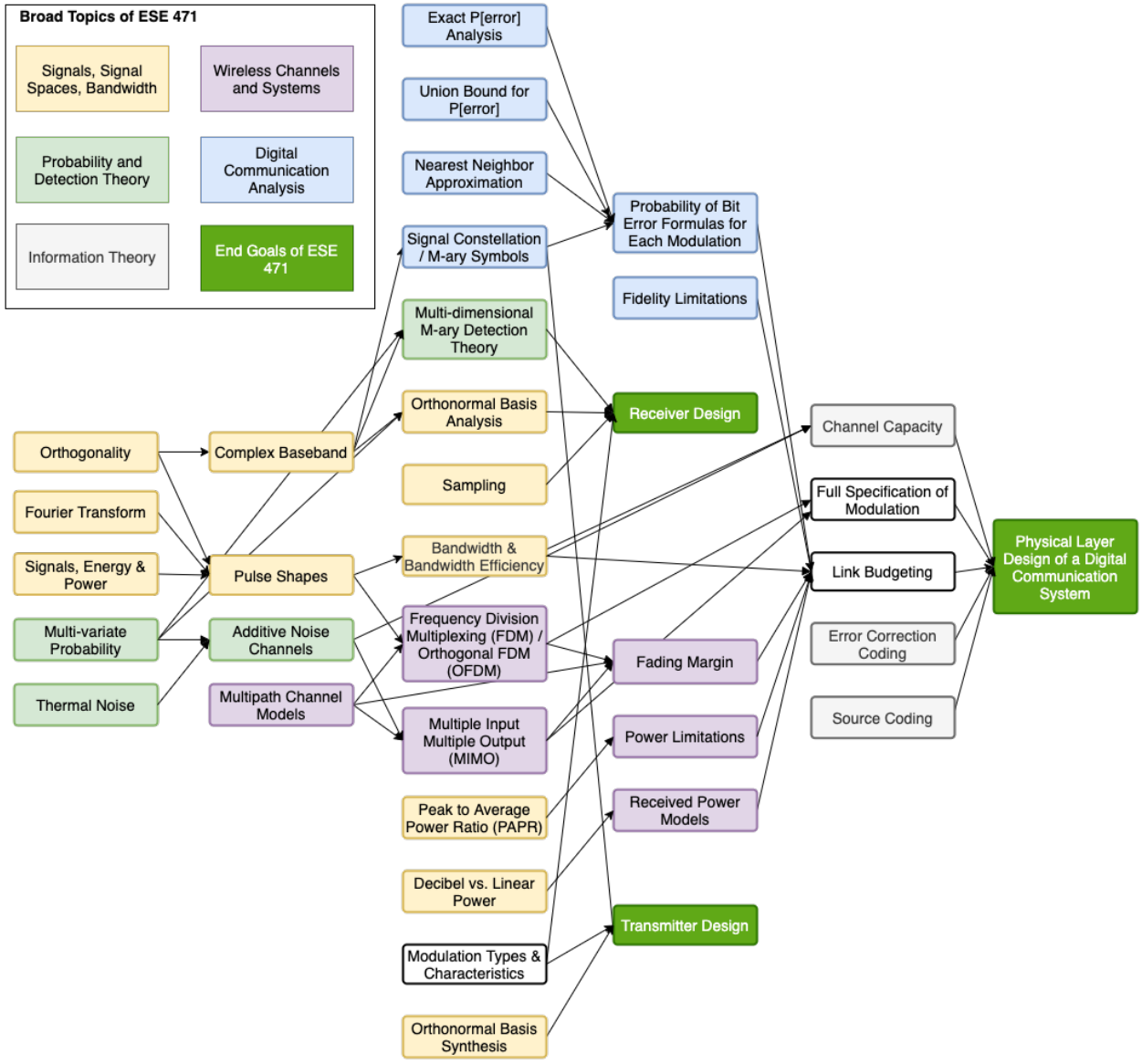


Figure 2: The topics we cover in ESE 471 include a) signals, signal spaces, and bandwidth; b) probability and detection theory; c) wireless channels; d) analysis of digital communications; e) information theory. These lead us to be able to design a transmitter and receiver, and to design and analyze the physical layer of a communication system to meet a specification.

in different frequency bands are orthogonal. We talk about the width of the band that a signal occupies so that we know how much of the spectrum we must reserve for it in the vicinity of the receiver.

We also send multiple orthogonal signals from a single transmitter in order to provide multiple independent dimensions on which to send information. We'll send signals that are orthogonal to the signals we send at a later time, so that we can convey different information over time and not have our own information signal interfere with itself.

2.6 Topic: Digital Communication Analysis

We'll apply probability theory to judge the probability of bit error in a given communication system. These formulas will help us decide what modulation to use to "match" the channel, and judge how efficient our system is compared to the best we can do. You'll learn how to calculate such a formula for any digital modulation, present or future.

A closely related topic is information theory. One of the most useful results for us is Shannon's capacity theorem, which gives us the "best we can do" bound I mentioned. It also provides us with coding methods which help us get closer to that bound. We introduce information theory in this course.

2.6.1 Topic: Wireless Channels

The wireless channel is particularly challenging. Multipath causes random fading. One can be conservative and send lots of extra power to account for the worst case, but this is inefficient. Special techniques have been developed to deal with, and in some ways *benefit from*, the randomness of the wireless channel. We cover two topics in this regard: 1) orthogonal frequency division multiplexing (OFDM), and 2) multiple-input multiple-output (MIMO) systems. Full study of such technologies requires more reading and study, but we'll cover the basics. Standard MIMO uses 2-4 antennas per device to allow bandwidth efficiency to multiply by 2-4. The next generation may require "massive MIMO", which uses ¿100 antennas at a base station to increase spectral efficiency.

2.7 Do We Need Communications Engineers?

Wireless communications has been a key enabling technology for living and working during this pandemic and stay-at-home push. We expect to have many parallel video calls going on simultaneously from each household, all day long. We expect our broadband connection to handle this traffic and be up without a hitch all day. I used to quote a statistic that broadband IP traffic will triple every 5 years and wireless traffic will increase twice as fast as wired, but I'm guessing that we're now growing faster than that.

How will we achieve these data rates? The spectrum is well utilized; there is little additional bandwidth available to handle this rate of increase, so engineers will have to make communication systems more spectrum efficient, or make use of spectrum that is currently available at higher frequencies (20 GHz and above) where there is more spectrum available (but with more severe channel challenges). Cellular communications systems have increased capacity largely by making cell coverage areas smaller; but there are limits to how small one can make a cell. Massive MIMO may be another way to increase bandwidth efficiency.

The number of wireless devices we own and use is also increasing rapidly. Many such devices may have lower data rates (e.g., wireless thermostats) and need to be able to operate with low

energy consumption. Today's protocols (e.g., WiFi) do not allow them to be energy efficient, and the protocols themselves operate inefficiently when serving large numbers of devices. There will be multiple new generations of wireless protocols to handle our Internet-of-Things devices and allow them to be as efficient as they need to be.

All of these new technologies, and those which will be developed to meet future demand, will require engineering.

Finally, the main objective of digital communications is to reliable *classification* at a receiver. A receiver must take in high-dimensional noisy measurements and make a decision about what is true about the information that was encoded. Thus I argue that ESE 471 is the original framework for "big data" – tools we develop are broadly useful across data science.

Lecture 2

Today: (1) Power, Energy, dB, (2) Orthogonality, (3) Periodicity and Impulse Functions

- Lecture videos for this lecture are at:
  `https://youtube.com/playlist?list=PLQuDEk4rPDOOunXRzrfMHdgRaATSVgnV8`

3 Power and Energy

Two of the biggest limitations in communications systems are (1) *energy / power*; and (2) *bandwidth*. This section provides some tools to deal with power and energy.

Recall that energy is power times time. **Use the units**: energy is measured in Joules (J); power is measured in Watts (W) which is the same as Joules/second (J/sec). Also, recall that our standard in signals and systems is define our signals, such as $x(t)$, as voltage signals (V). When we want to know the power of a signal we assume it is being dissipated in a 1 Ohm resistor, so $|x(t)|^2$ is the power dissipated at time t (since power is equal to the voltage squared divided by the resistance). The actual received power a

A signal $x(t)$ has energy defined as

$$E = \int_{-\infty}^{\infty} |x(t)|^2 dt$$

A signal with finite energy is called a *waveform* or *energy signal*. For some signals, E will be infinite because the signal is non-zero for an infinite duration of time (it is always on). These signals we call *power signals* and we compute their power as

$$P = \lim_{T\to\infty} \frac{1}{2T} \int_{-T}^{T} |x(t)|^2 dt$$

3.1 Discrete-Time Signals

In the Rice book [11], it refers to discrete samples of the sampled signal x as $x(n)$. You may be more familiar with the $x[n]$ notation. But, Matlab uses parentheses also; so we'll follow the Rice text notation. Essentially, whenever you see a function of n (or k, l, m), it is a discrete-time function; whenever you see a function of t (or perhaps τ) it is a continuous-time function. I'm sorry this is not more obvious in the notation.

For discrete-time signals, energy and power are defined as:

$$E = \sum_{n=-\infty}^{\infty} |x(n)|^2 \quad (1)$$

$$P = \lim_{N\to\infty} \frac{1}{2N+1} \sum_{n=-N}^{N} |x(n)|^2 \quad (2)$$

4 Orthogonality

In the introductory lecture, we described a couple reasons why orthogonal signals are used in digital communication systems. In short, we use orthogonal signals for:

1. *Multiple Access*: Multiple users can access the same medium, and a receiver can separate one user's signal from the rest.
2. *Increasing Signal Dimension*: A single user can send information along multiple dimensions at the same time, which is useful for increasing the bit rate or fidelity.

Also, I gave my "engineering" definition of a set of orthogonal waveforms: *They are waveforms that can be separated at the receiver.* Now, let's provide the mathematical definition.

Def'n: *Orthogonal*
Two real-valued waveforms $\phi_0(t)$ and $\phi_1(t)$ are *orthogonal* if

$$\int_{-\infty}^{\infty} \phi_0(t)\phi_1(t)dt = 0.$$

Two complex-valued waveforms $\phi_0(t)$ and $\phi_1(t)$ are orthogonal if

$$\int_{-\infty}^{\infty} \phi_0(t)\phi_1^*(t)dt = 0,$$

where $\phi_1^*(t)$ is the complex conjugate of $\phi_1(t)$.

What does this integral remind you of?

Def'n: *Orthogonal Set*
K waveforms $\phi_0(t), \ldots, \phi_{K-1}(t)$ are mutually orthogonal, or form an orthogonal set, if every pair of waveforms $\phi_i(t)$, $\phi_j(t)$, for $i \neq j$, is orthogonal.

Example: Sine and Cosine
Let

$$\phi_0(t) = \begin{cases} \cos(2\pi t), & 0 < t \leq 1 \\ 0, & o.w. \end{cases}$$

$$\phi_1(t) = \begin{cases} \sin(2\pi t), & 0 < t \leq 1 \\ 0, & o.w. \end{cases}$$

Are $\phi_0(t)$ and $\phi_1(t)$ orthogonal?

Solution: Using $\sin 2x = 2\cos x \sin x$,

$$\begin{aligned}\int_{-\infty}^{\infty} \phi_0(t)\phi_1(t)dt &= \int_0^1 \cos(2\pi t)\sin(2\pi t)dt \\ &= \int_0^1 \frac{1}{2}\sin(4\pi t)dt \\ &= \frac{-1}{8\pi}\cos(4\pi t)\Big|_0^1 = \frac{-1}{8\pi}(1-1) = 0\end{aligned}$$

So, yes, $\phi_0(t)$ and $\phi_1(t)$ are orthogonal.

Example: Frequency Shift Keying
Assume $T_s >> 1/f_c$, and show that these two are orthogonal.

$$\begin{aligned}\phi_0(t) &= \begin{cases}\cos(2\pi f_c t), & 0 \le t \le T_s \\ 0, & o.w.\end{cases} \\ \phi_1(t) &= \begin{cases}\cos\left(2\pi\left[f_c + \frac{1}{T_s}\right]t\right), & 0 \le t \le T_s \\ 0, & o.w.\end{cases}\end{aligned}$$

Solution: The integral of the product of the two must be zero. Checking, and using the identity for the product of two cosines,

$$\begin{aligned}&\int_0^{T_s} \cos(2\pi f_c t)\cos\left(2\pi\left[f_c + \frac{1}{T_s}\right]t\right)dt \\ = &\frac{1}{2}\int_0^{T_s}\cos(2\pi t/T_s)\,dt + \frac{1}{2}\int_0^{T_s}\cos(4\pi f_c t + 2\pi t/T_s)\,dt \\ = &0 + \frac{1}{2}\left[\frac{1}{2\pi(2f_c + 1/T_s)}\sin(2\pi(2f_c + 1/T_s)t)\right|_0^{T_s}\end{aligned}$$

The remaining term has a $\frac{1}{2\pi(2f_c+1/T_s)}$ constant out front. Because f_c is very high, this term will be very very low. The sine term is limited to between -1 and +1 so it will not cause the second term to be large. Thus,

$$\int_{-\infty}^{\infty}\phi_0(t)\phi_1(t)dt \le \frac{1}{\pi(2f_c + 1/T_s)} \approx 0$$

Thus the two different frequency waveforms are orthogonal.

5 Time-Domain Concept Review

5.1 Impulse Functions

Def'n: *Impulse Function*
The (Dirac) impulse function $\delta(t)$ is the function which makes

$$\int_{-\infty}^{\infty} x(t)\delta(t)dt = x(0) \tag{3}$$

true for any function $x(t)$ which is continuous at $t = 0$.

We are defining a function by its most important property, the 'sifting property'. Is the following definition more familiar?

$$\delta(t) = \lim_{T\to 0} \begin{cases} 1/(2T), & -T \le t \le T \\ 0, & o.w. \end{cases}$$

You can visualize $\delta(t)$ here as an infinitely high, infinitesimally wide pulse at the origin, with area one. This is why it 'pulls out' the value of $x(t)$ in the integral in (3).

Other properties of the impulse function:

- Time scaling,
- Symmetry,
- Sifting at arbitrary time t_0,

The continuous-time unit step function is

$$u(t) = \begin{cases} 1, & t \ge 0 \\ 0, & o.w. \end{cases}$$

Example: Sifting Property
What is $\int_{-\infty}^{\infty} \frac{\sin(\pi t)}{\pi t}\delta(1-t)dt$?
Solution: The integral pulls out the value of the fraction when the argument of the δ function is zero, in this case, when $t = 1$. Thus plug in $t = 1$ into $\frac{\sin(\pi t)}{\pi t}$ and get $\frac{\sin(\pi)}{\pi} = 0/\pi = 0$.

The discrete-time impulse function (also called the Kronecker delta or δ_K) is defined as:

$$\delta(n) = \begin{cases} 1, & n = 0 \\ 0, & o.w. \end{cases}$$

(There is no need to get complicated with the math; this is well defined.) Also,

$$u(n) = \begin{cases} 1, & n \ge 0 \\ 0, & o.w. \end{cases}$$

Lecture 3

Today: Orthogonal Signal Representations: (1) Span, (2) Synthesis, (3) Analysis

- Reading for these notes: Rice [11] Section 5.1
- Lecture videos for this lecture are at:
  `https://youtube.com/playlist?list=PLQuDEk4rPDONW82OXdbHZ6yjWT8nqJa7S`

6 Orthonormal Bases: Transmission and Reception

We've been talking about orthogonal waveforms. A shorthand notation: the integral of the product of two functions is also called the inner product, and sometimes the notation is used, where $f(t)$ and $g(t)$ are two waveforms:

$$\langle f(t), g(t) \rangle = \int_{t=-\infty}^{\infty} f(t)g(t)dt$$

Taking orthogonality one step further, we define an orthonormal basis:

Def'n: *Orthonormal Basis*
A set of functions $\phi_0(t), \ldots, \phi_{K-1}(t)$ are *orthonormal* (also referred to as an *orthonormal basis*) if they are mutually orthogonal and each has unit energy, *i.e.*, $\int_{t=-\infty}^{\infty} |\phi_k(t)|^2 dt = 1$ for $k = 0, 1$.

Sometimes an orthonormal basis is also called an orthonormal set. There is no difference between the two terms. The word "basis" is nice because it implies you can build something out of the set – and in fact, you can build a variety of signals by taking some linear combination of the functions in the orthonormal basis.

For shorthand, we use a set name to describe the orthonormal basis. The Rice book uses $\mathcal{B}$:

$$\mathcal{B} = \{\phi_0(t), \phi_1(t), \ldots, \phi_{K-1}(t)\}$$

Each function in $\mathcal{B}$ is called a basis function. You can think of $\mathcal{B}$ as an unambiguous and useful language.

What are some common orthonormal bases?

- Nyquist sampling: sinc functions centered at sampling times.
- Fourier series: complex sinusoids at different frequencies
- Sine and cosine at the same frequency
- Wavelets

And, we will come up with others. Each one has a limitation – only a certain set of signals can be exactly represented as a linear combination of its waveforms. Essentially, we must limit the set of possible signals to a set. That is, some subset of all possible signals.

6.1 Linear Combinations

What is a linear combination of orthogonal waveforms? Well, consider the orthogonal set $\phi_0(t), \ldots, \phi_{K-1}(t)$. A linear combination $s_m(t)$ is

$$s_m(t) = a_{m,0}\phi_0(t) + a_{m,1}\phi_1(t) + \cdots + a_{m,K-1}\phi_{K-1}(t) = \sum_{m=k}^{K-1} a_{m,k}\phi_k(t)$$

We also call the a linear combination a *symbol*. We use subscript m to indicate that it's not the only possible linear combination (or symbol). In fact, we will use M different symbols, so $i = 0, \ldots, M-1$, and we will use $s_0(t), \ldots, s_{M-1}(t)$.

We represent the mth symbol (linear combination of the orthogonal waveforms), $s_m(t)$, as a vector for ease of notation:

$$\mathbf{s}_m = [a_{m,0}, a_{m,1}, \ldots, a_{m,K-1}]^T$$

The superscript T is for transpose – $\mathbf{s}_m$ is a column vector. Vectors are easy to deal with because they can be plotted in vector space, to show graphically what is going on. We call the plot of all possible $\mathbf{s}_m$, that is, for $m = 0, \ldots M-1$, the *constellation diagram*. Some examples are shown in Figure 3.

6.2 Span of an Orthonormal Basis

Def'n: *Span*
The span of the set $\mathcal{B}$ is the set of all functions which are linear combinations of the functions in $\mathcal{B}$. The span is referred to as Span$\{\mathcal{B}\}$ and is

$$\text{Span}\{\mathcal{B}\} = \left\{ \sum_{k=0}^{K-1} a_k \phi_k(t) \text{ such that } a_0, \ldots, a_{K-1} \in \mathbb{R} \right\}$$

Another way to define this set is to say that a function $f(t) \in \text{Span}\{\mathcal{B}\}$ if and only if there exists $a_0, \ldots, a_{K-1} \in \mathbb{R}$ such that

$$f(t) = \sum_{k=0}^{K-1} a_k \phi_k(t)$$

In short, the Span of $\mathcal{B}$ is the space of possible symbol waveforms. The symbols $s_m(t)$ for $m = 0, \ldots M-1$ are the ones we choose to use to convey information between the transmitter and receiver. We call the set of the M possible symbols $\mathcal{S}$. Since every element of $\mathcal{S} \in \text{Span}\{\mathcal{B}\}$, it is clear that $\mathcal{S} \subset \text{Span}\{\mathcal{B}\}$.

6.3 Using M Different Linear Combinations

Here is how a transmitter *uses* the different linear combinations to convey digital bits to the receiver. First, consider that there are M different symbols for the TX to chose from. Each symbol is described by a $\log_2 M$-length bit sequence. For example, if there are 8 possible combinations, we would label them 000, 001, 011, 010, 110, 111, 101, 100.

The transmitter knows which $\log_2 M$-bit sequence it wants to send. It picks the symbol that corresponds to that bit sequence, let's call it symbol m. Then it sends $s_m(t)$.

If the receiver is able to determine that symbol m was sent, it will correctly receive those $\log_2 M$ bits of information. In this example, it will receive three bits of information.

Next we will talk about how a receiver is able to separate the received signal into components, each corresponding to one of the orthogonal waveforms $\phi_k(t)$. From this separation, it will be able to decide which symbol was sent.

6.4 How to Choose a Modulation

A *digital modulation* is the choice of: (1) the linear combinations $\mathbf{s}_0, \ldots, \mathbf{s}_{M-1}$ and, (2) the orthogonal waveforms $\phi_0(t), \ldots, \phi_{K-1}(t)$. We will choose a digital modulation as a tradeoff between the following characteristics:

1. Bandwidth efficiency: How many bits per second (bps) can be sent per Hertz of signal bandwidth. Thus the bandwidth efficiency has units of bps/Hz.

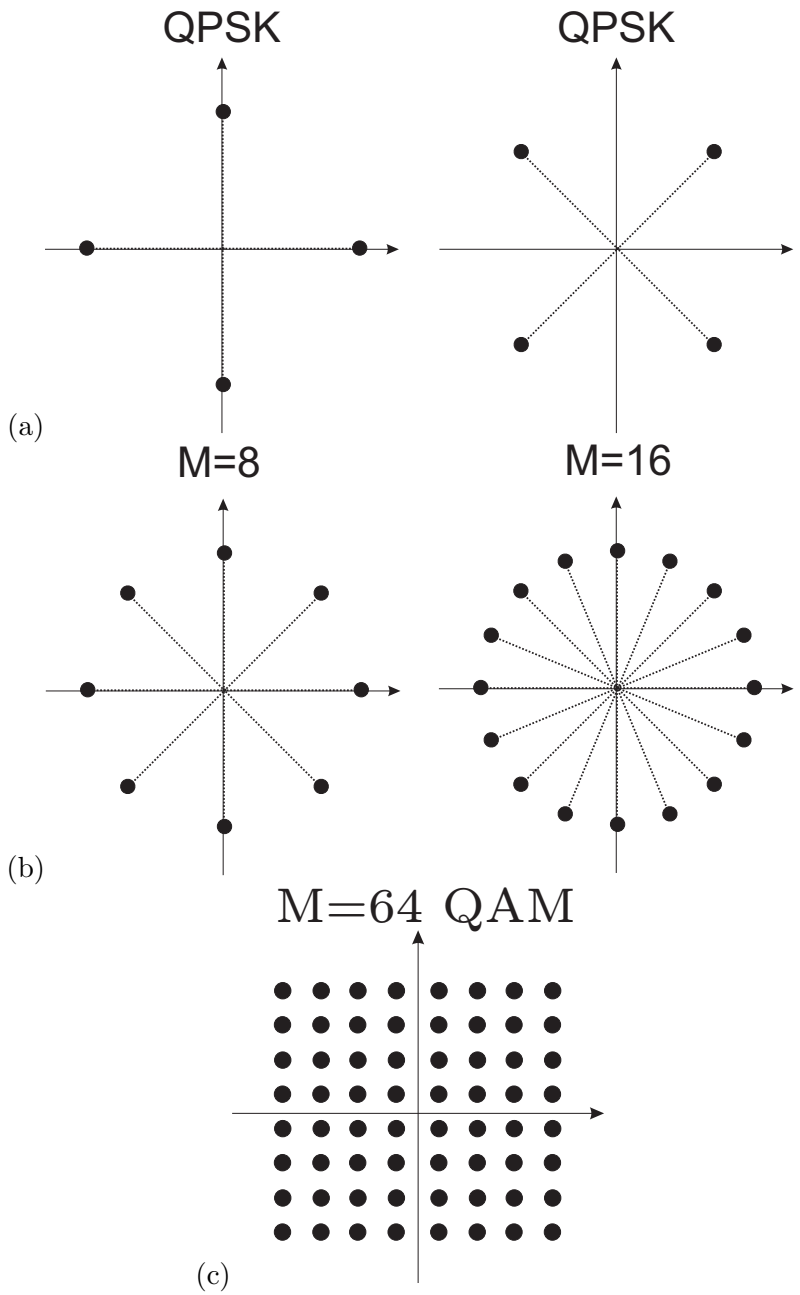


Figure 3: Signal constellations for (a) $M = 4$ PSK (a.k.a. BPSK), (b) $M = 8$ and $M = 16$ PSK, and (c) 64-QAM.

2. Power efficiency: How much energy per bit is required at the receiver in order to achieve a desired fidelity (low bit error rate). We typically use S/N or E_s/N_0 or E_b/N_0 as our figure of merit. This will be a major topic of the 2nd part of this course.

3. Cost of implementation: Things like symbol and carrier synchronization, and linear transmitters, require additional device cost, which might be unacceptable in a particular system.

6.5 Synthesis

Consider that you have a symbol waveform you want to send, $s_m(t)$, but that you don't know what constants $a_{m,k}$ for $k = 0, \ldots, K-1$ to use to generate it from your basis. Assuming that it is in Span$\{\mathcal{B}\}$, it can be represented as a linear combination of the basis functions,

$$s_m(t) = \sum_{k=0}^{K-1} a_{m,k}\phi_k(t)$$

Further, we can calculate the particular constants $a_{m,k}$ using the inner product:

$$a_{m,k} = \langle s_m(t), \phi_k(t) \rangle$$

The *projection* of the mth symbol $s_m(t)$ onto basis function k is defined as $a_{m,k}\phi_k(t) = \langle s_m(t), \phi_k(t) \rangle \phi_k(t)$. Then the signal is equal to the sum of its projections onto the basis functions.

Why is this?

Solution: Since $s_m(t) \in$ Span$\{\mathcal{B}\}$, we know there are some constants $\{a_{m,k}\}_k^{K-1}$ such that

$$s_m(t) = \sum_{k=0}^{K-1} a_{m,k}\phi_k(t)$$

Taking the inner product of both sides with $\phi_j(t)$,

$$\begin{aligned}
\langle s_m(t), \phi_j(t) \rangle &= \left\langle \sum_{k=0}^{K-1} a_{m,k}\phi_k(t), \phi_j(t) \right\rangle \\
\langle s_m(t), \phi_j(t) \rangle &= \sum_{k=0}^{K-1} a_{m,k} \langle \phi_k(t), \phi_j(t) \rangle \\
\langle s_m(t), \phi_j(t) \rangle &= a_{m,j}
\end{aligned}$$

So, now we can now represent a signal by a vector,

$$\mathbf{s}_m = [a_{m,0}, a_{m,1}, \ldots, a_{m,K-1}]^T$$

This and the basis functions **completely represent each signal**. Plotting $\{\mathbf{s}_m\}_m$ in an K dimensional grid is termed a *constellation diagram*. Generally, this space that we're plotting in is called *signal space*.

We can also synthesize any of our M signals in the signal set by adding the proper linear combination of the K orthonormal waveforms. By choosing one of the M signals, we convey information, specifically, $\log_2 M$ bits of information.

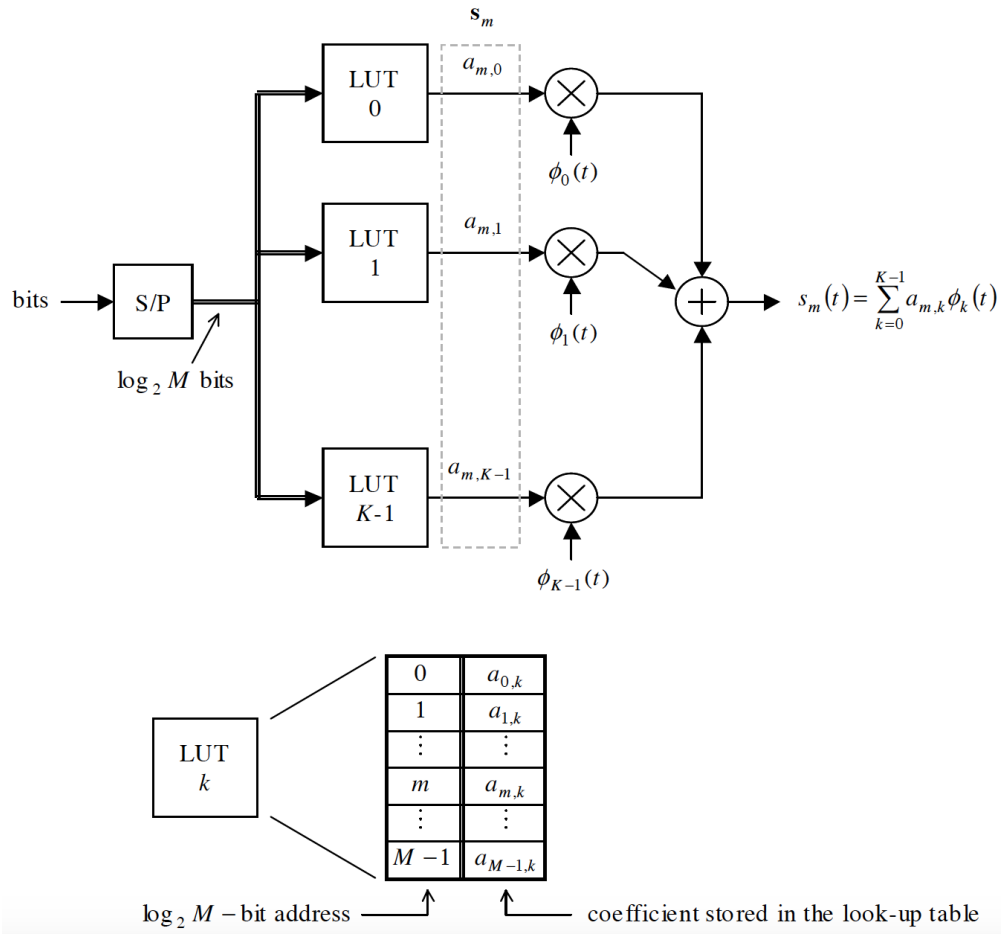


Figure 4: Rice book Figure 5.5: Block diagram of a modulator for M-ary linear modulation based on the synthesis equation.

See Figure 4, from Rice Chapter 5, which shows a block diagram of how a transmitter would synthesize one of the M signals to send, based on an input bit stream.

Example: Position-shifted pulses
Plot the signal space diagram for the signals,

$$\begin{aligned} s_0(t) &= u(t) - u(t-1) \\ s_1(t) &= u(t-1) - u(t-2) \\ s_2(t) &= u(t) - u(t-2) \end{aligned}$$

given the orthonormal basis,

$$\begin{aligned} \phi_0(t) &= u(t) - u(t-1) \\ \phi_1(t) &= u(t-1) - u(t-2) \end{aligned}$$

What are the signal space vectors, $\mathbf{s}_0$, $\mathbf{s}_1$, and $\mathbf{s}_2$?
Solution: They are $\mathbf{s}_0 = [1, 0]^T$, $\mathbf{s}_1 = [0, 1]^T$, and $\mathbf{s}_2 = [1, 1]^T$. They are plotted in the signal space diagram in Figure 5.

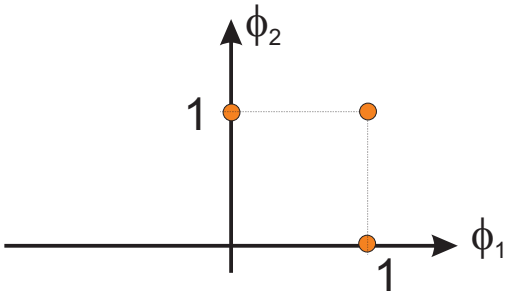


Figure 5: Signal space diagram for position-shifted signals example.

Energy: Energy can be calculated in signal space as

$$\text{Energy}\{s_m(t)\} = \int_{-\infty}^{\infty} s_i^2(t)dt = \sum_{k=0}^{K-1} a_{m,k}^2$$

Proof? (To do on your own).

Although different orthonormal bases can be used (one can represent the same symbols with a different orthonormal basis and different symbol vectors), the energy and distance between points in the constellation diagram will not change.

Example: Amplitude-shifted signals
Now consider

$$\begin{aligned} s_0(t) &= 1.5[u(t) - u(t-1)] \\ s_1(t) &= 0.5[u(t) - u(t-1)] \\ s_2(t) &= -0.5[u(t) - u(t-1)] \\ s_3(t) &= -1.5[u(t) - u(t-1)] \end{aligned}$$

and the orthonormal basis,

$$\phi_1(t) \quad = \quad u(t) - u(t-1)$$

What are the signal space vectors for the signals $\{s_m(t)\}$? What are their energies?
Solution: $\mathbf{s}_0 = [1.5]$, $\mathbf{s}_1 = [0.5]$, $\mathbf{s}_2 = [-0.5]$, $\mathbf{s}_3 = [-1.5]$. See Figure 6.

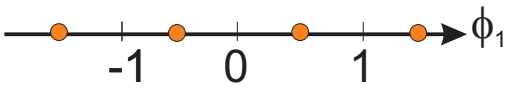


Figure 6: Signal space diagram for amplitude-shifted signals example.

Energies are just the squared magnitude of the vector: 2.25, 0.25, 0.25, and 2.25, respectively.

6.6 Analysis

At a receiver, our job will be to analyze the received signal (a function) and to decide which of the M possible signals was sent. This is the task of *analysis*. It turns out that an orthonormal bases makes our analysis very straightforward and elegant.

We won't receive exactly what we sent - there will be additional functions added to the signal function we sent due to noise and interference. We might say that if we send signal m, *i.e.*, $s_m(t)$ from our signal set, then we would receive

$$r(t) = s_m(t) + w(t)$$

where the $w(t)$ is the sum of all of the additive signals that we did not intend to receive. But $w(t)$ would generally not be totally in the span of our basis $\mathcal{B}$, so $r(t)$ would not be in Span$\{\mathcal{B}\}$ either.

6.6.1 Symbol Closest to Received Signal

One main question we ask at the receiver is, what is the symbol $s_m(t)$ that is closest to the received signal? The term "closest" here is somewhat qualitative until we define it. Without proof (in this lecture) we are going to use squared error to measure "closeness":

$$\mathcal{E}_m = \int_{-\infty}^{\infty} |r(t) - s_m(t)|^2 dt \tag{4}$$

That is, for any given m, we would integrate across time the squared difference between $r(t)$ and $s_m(t)$. Our decision will be the m that has minimum $\mathcal{E}_m$:

$$\hat{m} = \arg \min_{m \in \{0 \ldots M-1\}} \left\{ \int_{-\infty}^{\infty} |r(t) - s_m(t)|^2 dt \right\} \tag{5}$$

The term $\hat{s}(t)$ is the result, the receiver's guess of the transmitted symbol. Because $s_m(t) \in \mathcal{S}$, we know that $s_m(t) = \sum_{k=0}^{K-1} a_{m,k}\phi_{m,k}$. So,

$$\begin{aligned}\hat{m} &= \arg\min_{m\in\{0\ldots M-1\}} \left\{ \int_{t=-\infty}^{\infty} r^2(t)dt \right. \\ &\quad -2\sum_{k=0}^{K-1} a_{m,k}\int_{t=-\infty}^{\infty} r(t)\phi_k(t)dt \\ &\quad \left. +\sum_{k=0}^{K-1}\sum_{k'=0}^{K-1} a_{m,k}a_{m,k'}\int_{t=-\infty}^{\infty}\phi_k(t)\phi_{k'}(t)dt \right\}. \end{aligned} \tag{6}$$

Although there are a lot of terms, this begins to simplify. The first term is not a function of m. The integral in the second term is the inner product $\langle r(t), \phi_k(t)\rangle$. Let's call this x_k. The integral in the third term is one when $k = k'$ and zero otherwise, so we can just consider the terms when $k = k'$,

$$\arg\min_{m\in\{0\ldots M-1\}} \left\{ -2\sum_{k=0}^{K-1} a_{m,k}x_k + \sum_{k=0}^{K-1} a_{m,k}^2 \right\}. \tag{7}$$

We can make this easier for us by including a constant $\sum_{k=0}^{K-1} x_k^2$. Since this constant is not a function of m, it doesn't affect the arg min.

$$\begin{aligned} &= \arg\min_{m\in\{0\ldots M-1\}} \left\{ \sum_{k=0}^{K-1} \left(x_k^2 - 2a_{m,k}x_k + a_{m,k}^2 \right) \right\}. \\ &= \arg\min_{m\in\{0\ldots M-1\}} \left\{ \sum_{k=0}^{K-1} \left(x_k - a_{m,k} \right)^2 \right\}. \end{aligned} \tag{8}$$

To find the m that makes this the smallest, then, we should calculate the x_k values by finding the inner product between the received signal and $\phi_k(t)$, and forming the vector

$$\mathbf{x} = [x_0, x_1, \ldots, x_{K-1}]^T$$

Keep a list of the symbol vectors $\mathbf{s}_m$ for each m. The final expression above translates to

$$\hat{m} = \arg\min_m \|\mathbf{x} - \mathbf{s}_m\|^2 \tag{9}$$

where $\|\cdot\|^2$ is the squared norm of a vector. Thus, just find the squared Euclidean distance between each $\mathbf{s}_m$ and $\mathbf{x}$. This lowest-squared distance vector corresponds to the m that is closest to the received signal.

Note that $\mathbf{x}$, that is, the inner products

$$x_k = \int_{t=-\infty}^{\infty} r(t)\phi_k(t)dt$$

are all that matters in the decision about which symbol was sent.

6.6.2 Best Approximation for Received Signal in the Basis

What is the best approximation to $r(t)$ in the signal space? Specifically, what is $\hat{r}(t) \in \text{Span}\{\mathcal{B}\}$ such that the energy of the difference between $\hat{r}(t)$ and $r(t)$ is minimized, *i.e.*,

$$\arg \min_{\hat{r}(t) \in \text{Span}\{\mathcal{B}\}} \int_{-\infty}^{\infty} |\hat{r}(t) - r(t)|^2 dt \tag{10}$$

Solution: Since $\hat{r}(t) \in \text{Span}\{\mathcal{B}\}$, it can be represented as a vector in signal space,

$$\mathbf{x} = [x_0, x_1, \ldots, x_{K-1}]^T.$$

and the synthesis equation is

$$\hat{r}(t) = \sum_{k=0}^{K-1} x_k \phi_k(t)$$

If you plug in the above expression for $\hat{r}(t)$ into (10), and then find the minimum with respect to each x_k, you'd see that the minimum error is at

$$x_k = \int_{-\infty}^{\infty} r(t)\phi_k(t)dt$$

that is, $x_k = \langle r(t), \phi_k(t) \rangle$, for $k = 0, \ldots, K-1$.

Example: Analysis using a Walsh-Hadamard 2 Basis
See Figure 7. Let $s_0(t) = \phi_0(t)$ and $s_1(t) = \phi_1(t)$. What is $\hat{r}(t)$?
Solution:

$$\hat{r}(t) = \begin{cases} 1, & 0 \le t < 1 \\ -1/2, & 1 \le t < 2 \\ 0, & o.w. \end{cases}$$

Lecture 4

Today: (1) Fourier Transform and Bandwidth, (2) Sampling Intro

- Lecture videos for this lecture are at:
  `https://youtube.com/playlist?list=PLQuDEk4rPDOOmA07EcEiplvuAbD1uP4kB`

So far we've been talking about continuous-time waveforms. We just had a lecture on signal space representations, which is a discrete representation of a continuous-time signal. But we have not talked specifically about discrete-time signals, nor have we talked about the bandwidth of the waveforms we are using.

Today's objectives are to provide the tools needed to study:

1. The frequency content of waveforms that we will use for digital communications systems,
2. Sampling of continuous-time signals, and the frequency content of discrete-time signals, and
3. Operations we will do on discrete-time signals, for example, frequency translation and filtering.

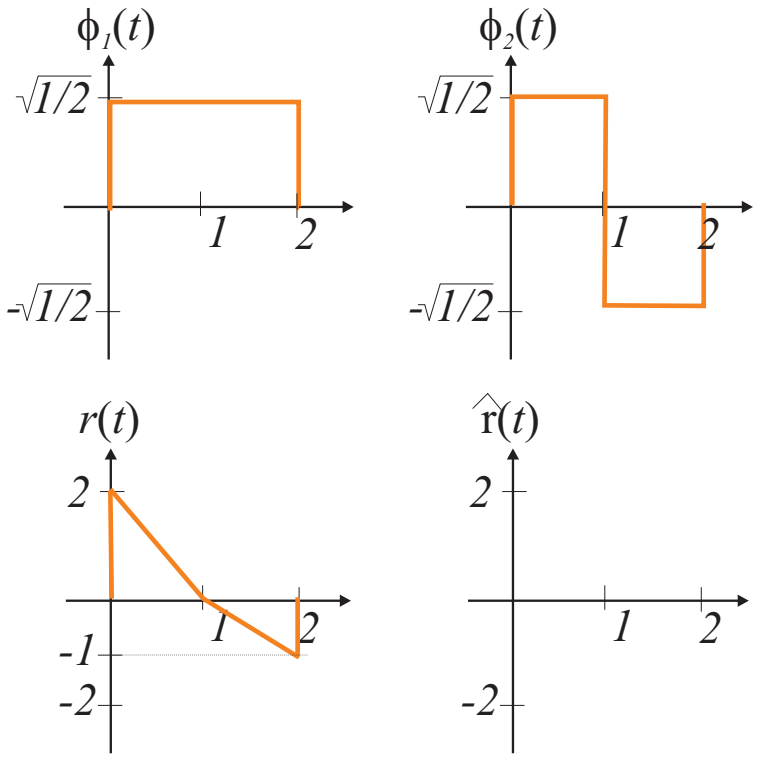


Figure 7: Signal and basis functions for Analysis example.

7 Fourier Transform

The Fourier transform (FT) is a transform used to show the frequency content of continuous-time, aperiodic signals.

7.1 Periodicity

Def'n: *Periodic (continuous-time)*
A signal $x(t)$ is periodic if $x(t) = x(t + T_0)$ for some constant $T_0 \neq 0$ for all $t \in \mathbb{R}$. The smallest such constant $T_0 > 0$ is the *period*.

If a signal is not periodic it is *aperiodic*.
Periodic signals have *Fourier series* representations, as defined in Rice Ch. 2 [11].

Def'n: *Periodic (discrete-time)*
A DT signal $x(n)$ is periodic if $x(n) = x(n + N_0)$ for some integer $N_0 \neq 0$, for all integers n. The smallest positive integer N_0 is the period.

Notes about continuous-time frequency transforms:

1. You might be most familiar with the Laplace Transform. To convert it to the Fourier transform, we replace s with $j\omega$, where ω is the radial frequency, with units radians per second (rad/s).

Time:	**Periodicity:** *Periodic*	*Aperiodic*
Continuous-Time		Laplace Transform $x(t) \leftrightarrow X(s)$
	Fourier Series $x(t) \leftrightarrow a_k$	Fourier Transform $x(t) \leftrightarrow X(j\omega)$ $X(j\omega) = \int_{t=-\infty}^{\infty} x(t)e^{-j\omega t}dt$ $x(t) = \frac{1}{2\pi}\int_{\omega=-\infty}^{\infty} X(j\omega)e^{j\omega t}d\omega$
Discrete-Time		z-Transform $x(n) \leftrightarrow X(z)$
	Discrete Fourier Transform (DFT) $x(n) \leftrightarrow X[k]$ $X[k] = \sum_{n=0}^{N-1} x(n)e^{-j\frac{2\pi}{N}kn}$ $x(n) = \frac{1}{N}\sum_{k=0}^{N-1} X[k]e^{j\frac{2\pi}{N}nk}$	Discrete Time Fourier Transform (DTFT) $x(n) \leftrightarrow X(e^{j\Omega})$ $X(e^{j\Omega}) = \sum_{n=-\infty}^{\infty} x(n)e^{-j\Omega n}$ $x(n) = \frac{1}{2\pi}\int_{n=-\pi}^{\pi} X(e^{j\Omega})d\Omega$

Table 1: Frequency Transforms

2. You may prefer the radial frequency representation, but also feel free to use the rotational frequency f (which has units of cycles per sec, or Hz. Frequency in Hz is more standard for communications; you should use it for intuition. In this case, just substitute $\omega = 2\pi f$. You could write $X(j2\pi f)$ as the notation for this, but typically you'll see it written as $X(f)$. Note that the definition of the Fourier transform in the f domain removes the $\frac{1}{2\pi}$ in the inverse Fourier transform definition.

$$\begin{aligned} X(j2\pi f) &= \int_{t=-\infty}^{\infty} x(t)e^{-j2\pi ft}dt \\ x(t) &= \int_{f=-\infty}^{\infty} X(j2\pi f)e^{j2\pi ft}df \end{aligned}$$

3. The Fourier series is limited to purely periodic signals. Both Laplace and Fourier transforms are *not* limited to periodic signals.

4. Note that $e^{j\alpha} = \cos(\alpha) + j\sin(\alpha)$.

See Table 2.4.4 in the Rice book.

Example: Square Wave
Given a rectangular pulse $x(t) = \text{rect}(t/T_s)$,

$$x(t) = \begin{cases} 1, & -T_s/2 < t \leq T_s/2 \\ 0, & o.w. \end{cases}$$

What is the Fourier transform $X(f)$? Calculate both from the definition and from a table.

Solution: Method 1: From the definition:

$$\begin{aligned} X(j\omega) &= \int_{t=-T_s/2}^{T_s/2} e^{-j\omega t} dt \\ &= \frac{1}{-j\omega} e^{-j\omega t}\Big|_{t=-T_s/2}^{T_s/2} \\ &= \frac{1}{-j\omega}\left(e^{-j\omega T_s/2} - e^{j\omega T_s/2}\right) \end{aligned}$$

Using the fact that $\frac{1}{-2j}\left(e^{-j\alpha} - e^{j\alpha}\right) = \sin(\alpha)$, we have:

$$X(j\omega) = 2\frac{\sin(\omega T_s/2)}{\omega} = T_s \frac{\sin(\omega T_s/2)}{\omega T_s/2}.$$

While it is sometimes convenient to replace $\frac{\sin(\pi x)}{\pi x}$ with $\mathrm{sinc}(x)$, it is confusing because $\mathrm{sinc}(x)$ is sometimes defined as $\frac{\sin(\pi x)}{\pi x}$ and sometimes defined as $\frac{\sin x}{x}$. No standard definition for 'sinc' exists! Rather than make a mistake because of this, the Rice book always writes out the expression fully. I will try to follow suit.

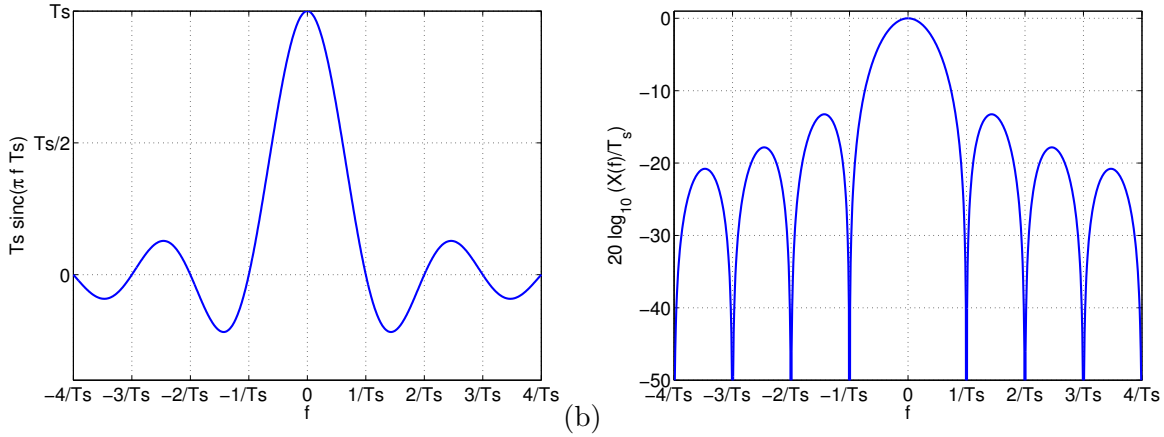
Method 2: From the tables and properties. Let $T = T_s/2$. Then

$$x(t) = \begin{cases} 1, & |t| < T \\ 0, & o.w. \end{cases}, \tag{11}$$

which is in Table 2.4.4. From the table,

$$X(j\omega) = 2T\frac{\sin(\omega T)}{\omega T} = T_s \frac{\sin(\omega T_s/2)}{\omega T_s/2}.$$

See Figure 8(a).



(a) (b)

Figure 8: (a) Fourier transform $X(j2\pi f)$ of rect pulse with period T_s, and (b) Power vs. frequency $20\log_{10}(X(j2\pi f)/T_s)$.

Question: What if $Y(j\omega)$ was a rect function? What would the inverse Fourier transform $y(t)$ be?

One can see a problem of using waveforms that are rectangular-shaped in *either* the time or frequency domains. If 100% limited in the time domain, then the signal spreads infinitely in frequency; if 100% limited in the frequency domain, then the signal spreads out infinitely in time.

7.2 Fourier Transform Properties

See Table 2.4.3 in the Rice book.

Assume that $\mathfrak{F}\{x(t)\} = X(j\omega)$. Important properties of the Fourier transform:

1. Time shift property:
$$\mathfrak{F}\{x(t-t_0)\} = e^{-j\omega t_0}X(j\omega)$$

2. Scaling property: for any real $a \neq 0$,
$$\mathfrak{F}\{x(at)\} = \frac{1}{|a|}X\left(j\frac{\omega}{a}\right)$$

3. Convolution property: if, additionally $y(t)$ has Fourier transform $X(j\omega)$,
$$\mathfrak{F}\{x(t) \star y(t)\} = X(j\omega) \cdot Y(j\omega)$$

4. Modulation property:
$$\mathfrak{F}\{x(t)cos(\omega_0 t)\} = \frac{1}{2}X(\omega-\omega_0) + \frac{1}{2}X(\omega+\omega_0)$$

5. Duality property:
$$\begin{aligned} x(j\omega) &= \mathfrak{F}\{X(-t)\} \\ x(-j\omega) &= \mathfrak{F}\{X(t)\} \end{aligned}$$
This says is that you can go backwards in a Fourier transform table – Replace the ω with t in the frequency column, and replace t with $-\omega$ in the time-domain column.

6. Parceval's theorem: The energy calculated in the frequency domain is equal to the energy calculated in the time domain.
$$\int_{t=-\infty}^{\infty}|x(t)|^2dt = \int_{f=-\infty}^{\infty}|X(f)|^2df = \frac{1}{2\pi}\int_{\omega=-\infty}^{\infty}|X(j\omega)|^2d\omega$$
So do whichever one is easiest! Or, check your answer by doing both.

Example: Applying FT Properties
If $w(t)$ has the Fourier transform $W(j\omega)$, find $X(j\omega)$ for the following: $x(t) = w(2t+2)$.
Solution: Let $x(t) = z(2t)$ where $z(t) = w(t+2)$. Then $Z(j\omega) = e^{j2\omega}W(j\omega)$. Then $X(j\omega) = \frac{1}{2}Z\left(\frac{j\omega}{2}\right)$. So plugging in the first result,
$$X(j\omega) = \frac{1}{2}e^{j\omega}W\left(\frac{j\omega}{2}\right).$$

Alternatively, let $x(t) = y(t+1)$ where $y(t) = w(2t)$. Then $X(j\omega) = e^{j\omega}Y(j\omega)$. Then $Y(j\omega) = \frac{1}{2}W\left(\frac{j\omega}{2}\right)$. Plugging in,

$$X(j\omega) = e^{j\omega}\frac{1}{2}W\left(\frac{j\omega}{2}\right).$$

The two approaches come up with the same answer, as they should.

7.3 Bandwidth

Bandwidth is a critical resource for a digital communications system; we have various definitions to quantify it. In short, it isn't easy to describe a signal in the frequency domain with a single number. And, in the end, a system will be designed to meet a spectral mask required by the FCC or system standard.

Intuitively, bandwidth is the maximum extent of our signal's frequency domain characterization $X(f)$. A baseband signal absolute bandwidth is often defined as the W such that $X(f) = 0$ for all f except for the range $-W \le f \le W$. Other definitions for bandwidth are

- 3-dB bandwidth: B_{3dB} is the value of f such that $|X(f)|^2 = |X(0)|^2/2$.
- 90% bandwidth: $B_{90\%}$ is the narrowest range which captures 90% of the energy in the signal:

$$\int_{-B_{90\%}}^{B_{90\%}} |X(f)|^2 df = 0.90 \int_{f=-\infty}^{\infty} |X(f)|^2 df$$

As a motivating example, I mention the square-root raised cosine (SRRC) pulse, which has the following desirable Fourier transform:

$$H_{RRC}(f) = \begin{cases} \sqrt{T_s}, & 0 \le |f| \le \frac{1-\alpha}{2T_s} \\ \sqrt{\frac{T_s}{2}\left\{1+\cos\left[\frac{\pi T_s}{\alpha}\left(|f| - \frac{1-\alpha}{2T_s}\right)\right]\right\}}, & \frac{1-\alpha}{2T_s} \le |f| \le \frac{1+\alpha}{2T_s} \\ 0, & o.w. \end{cases} \tag{12}$$

where α is a parameter called the "rolloff factor". We can actually analyze this using the properties of the Fourier transform and many of the standard transforms you'll find in a Fourier transform table.

The SRRC and other pulse shapes are discussed in Appendix A, and we will go into more detail later on. The purpose so far is to motivate practicing up on frequency transforms.

Example: Butterworth Filter
A Butterworth low-pass filter has a frequency response with magnitude,

$$|H(f)| = \frac{1}{\sqrt{1+(f/f_0)^{2n}}}$$

where n is the number of reactive components (*i.e.*, inductors or capacitors).

1. What is the 3dB bandwidth of this filter?
2. Find the lowest n so that $|H(f)|^2$ is constant to within 1 dB over the frequency range $-0.8f_0 < f < 0.8f_0$. Hint: $|H(f)|^2$ is strictly decreasing as $|f|$ increases. So the requirement is that $|H(0)|^2$ is no more than 1.26 times $|H(0.8f_0)|^2$.

Solution:

1. The squared magnitude, $|H(f)|^2$ is equal to 1/2 when $f = f_0$, regardless of n. Thus the 3dB bandwidth is always f_0.

2. $|H(f)|$ is strictly decreasing from $f = 0$. We need to find the n such that $|H(\pm 0.8 f_0)|^2$ is more than 1 dB down from the filter's maximum value (at $f = 0$). Note that

$$10 \log_{10} |H(f)|^2 = -10 \log_{10} \left(1 + (f/f_0)^{2n}\right)$$

which makes $10 \log_{10} |H(0)|^2 = 0$ and thus we are looking for the n that has:

$$\begin{aligned} -1 &= 10 \log_{10} |H(0.8 f_0)|^2 = -10 \log_{10}(1 + 0.8^{2n}) \\ 0.1 &= \log_{10}(1 + 0.8^{2n}) \\ 10^{0.1} - 1 &= 0.8^{2n} \\ n &= \frac{\log(10^{0.1} - 1)}{2 \log(0.8)} \approx 3.03 \end{aligned}$$

While $n = 3$ is a good engineering answer, $n = 4$ is a good math answer, since $n = 3$, technically, would lead to > 1 dB variation.

7.4 Linear Time Invariant (LTI) Filters

If a (deterministic) signal $x(t)$ is input to a LTI filter with impulse response $h(t)$, the output signal is

$$y(t) = h(t) \star x(t)$$

Using the above convolution property,

$$Y(j\omega) = X(j\omega) H(j\omega)$$

8 Sampling

A common statement of the Nyquist sampling theorem is that a signal can be sampled at twice its bandwidth. But the theorem really has to do with signal reconstruction from a sampled signal.

Theorem: (Nyquist Sampling Theorem.) Let $x_c(t)$ be a baseband, continuous signal with bandwidth B (in Hz), *i.e.*, $X_c(j\omega) = 0$ for all $|\omega| \geq 2\pi B$. Let $x_c(t)$ be sampled at multiples of T, where $\frac{1}{T} \geq 2B$ to yield the sequence $\{x_c(nT)\}_{n=-\infty}^{\infty}$. Then

$$x_c(t) = 2BT \sum_{n=-\infty}^{\infty} x_c(nT) \frac{\sin(2\pi B(t - nT))}{2\pi B(t - nT)}. \tag{13}$$

Proof: Not covered.

Notes:

- This is an interpolation procedure.
- This is a synthesis equation with $2BT \frac{\sin(2\pi B(t-nT))}{2\pi B(t-nT)}$, for integer n, as the orthogonal basis functions, for the space of bandlimited signals.
- This is only precise when $X(j\omega) = 0$ for all $|\omega| \geq 2\pi B$.

8.1 Aliasing Due To Sampling

Essentially, sampling is the multiplication of a impulse train (at period T with the desired signal $x(t)$:

$$\begin{aligned} x_{sa}(t) &= x(t) \sum_{n=-\infty}^{\infty} \delta(t - nT) \\ x_{sa}(t) &= \sum_{n=-\infty}^{\infty} x(nT)\delta(t - nT) \end{aligned}$$

What is the Fourier transform of $x_{sa}(t)$? In the frequency domain, this is a convolution:

$$\begin{aligned} X_{sa}(j\omega) &= X(j\omega) \star \frac{2\pi}{T} \sum_{n=-\infty}^{\infty} \delta\left(\omega - \frac{2\pi n}{T}\right) \\ &= \frac{1}{T} \sum_{n=-\infty}^{\infty} X\left(j\left(\omega - \frac{2\pi n}{T}\right)\right) \quad \text{for all } \omega \\ &= \frac{1}{T} X(j\omega) \quad \text{for } |\omega| < 2\pi B \text{ if } X(j\omega) \text{is bandlimited.} \end{aligned} \tag{14}$$

This is shown graphically in the Rice book [11] in Section 2.6.1 "The Sampling Theorem", Figure 2.12.

The Fourier transform of the sampled signal is many copies of $X(j\omega)$ strung at integer multiples of $2\pi/T$, as shown in Fig. 9.

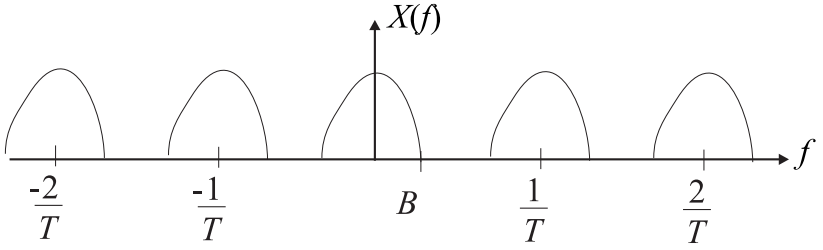


Figure 9: The effect of sampling on the frequency spectrum in terms of frequency f in Hz.

Example: Sinusoid sampled above and below Nyquist rate
Consider two sinusoidal signals sampled at $1/T = 25$ Hz:

$$\begin{aligned} x_1(nT) &= \sin(2\pi 5nT) \\ x_2(nT) &= \sin(2\pi 20nT) \end{aligned}$$

What are the two frequencies of the sinusoids, and what is the Nyquist rate? Which of them does the Nyquist theorem apply to? Draw the spectrums of the continuous signals $x_1(t)$ and $x_2(t)$, and indicate what the spectrum is of the sampled signals.

Figure 10 shows what happens when the Nyquist theorem is applied to the each signal (whether or not it is valid). What observations would you make about Figure 10(b), compared to Figure 10(a)?

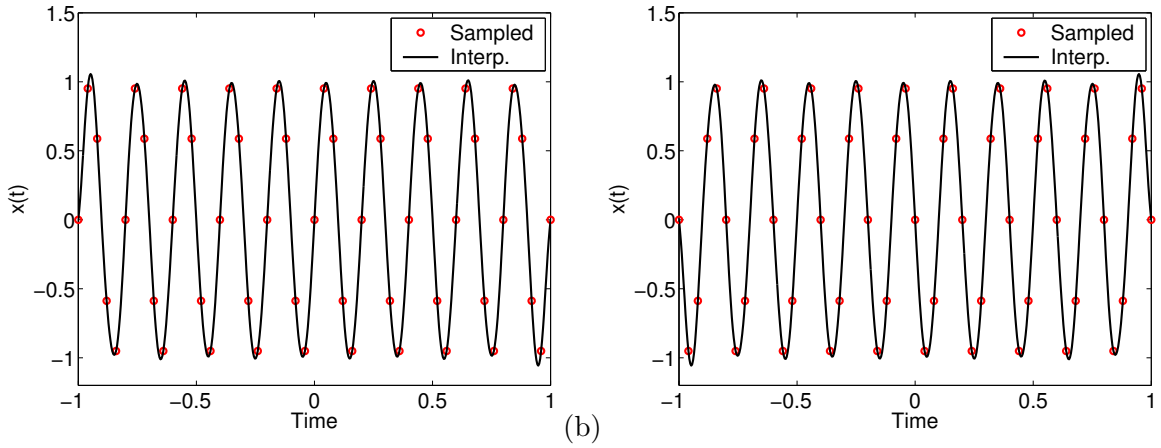


Figure 10: Sampled (a) $x_1(nT)$ and (b) $x_2(nT)$ are interpolated (—) using the Nyquist interpolation formula.

Example: Square vs. round pulse shape
Consider the square pulse considered before, $x_1(t) = \text{rect}(t/T_s)$. Also consider a parabola pulse (this doesn't really exist in the wild – I'm making it up for an example.)

$$x_2(t) = \begin{cases} 1 - \left(\frac{2t}{T_s}\right)^2, & -\frac{1}{2T_s} \leq t \leq \frac{1}{2T_s} \\ 0, & o.w. \end{cases}$$

What happens to $x_1(t)$ and $x_2(t)$ when they are sampled at rate T?

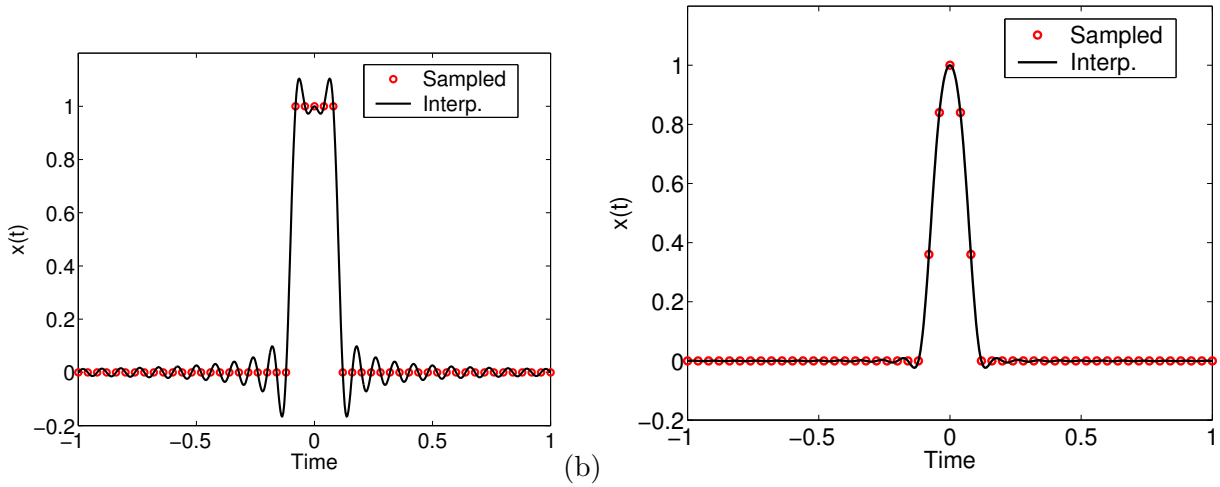


Figure 11: Sampled (a) $x_1(nT)$ and (b) $x_2(nT)$ are interpolated (—) using the Nyquist interpolation formula.

In the Matlab code `EgNyquistInterpolation.m` we set $T_s = 1/5$ and $T = 1/25$. Then, the sampled pulses are interpolated using (13). Even though we've sampled at a pretty high rate, the reconstructed signals will not be perfect representations of the original, in particular for $x_1(t)$. See Figure 11.

Lecture 5

Today: (1) Intro to PAM, (2) ISI / Nyquist Filtering Theorem

- Reading for these notes: Rice [11] A.2, 3.2.
- Lecture videos for this lecture are at:
  `https://youtube.com/playlist?list=PLQuDEk4rPDOOePIbyLGaqaygf-2fR5d4W`

9 Intro to PAM

Pulse amplitude modulation is a one-dimensional modulation, that is, it has one waveform $\phi_0(t)$ in its basis. We write this as $\phi_0(t) = p(t)$, where $p(t)$ is a unit-energy waveform called the pulse shape. (The reason to introduce the new function name $p(t)$ will become clear later when we introduce other modulations.) To encode symbols, we will choose an amplitude $a_{m,0}$ to indicate the mth symbol, and make the $a_{m,0}$ different. For example, for $M = 2$, we might choose amplitudes $\{-A, A\}$ (this is also called *binary bipolar PAM* or because it is so common usually we shorten it to *binary PAM*). The variable A then determines the energy of the symbol. As another example, for $M = 4$, we could choose from the set $\{-3A, -A, A, 3A\}$. These are equally spaced, but for $M > 2$, different symbols have different energies. In particular, for M-PAM, the constellation diagram is shown in Rice Section 5.2.1, and amplitudes $\{a_{m,0}\}_m$ are,

$$-(M-1)A, -(M-3)A, \ldots, -A, A, \ldots, (M-3)A, (M-1)A$$

The average energy, assuming that each symbol is equally likely, can be shown to be:

$$E_{avg} = \frac{M^2 - 1}{3} A^2$$

To send multiple symbols over time, we transmit a signal $s(t)$,

$$s(t) = \sum_n a(n) p(t - nT_s),$$

where $a(n)$ is the amplitude (one of the $\{a_{m,0}\}_m$) of the nth symbol we choose to send, and T_s is the symbol period. Each subsequent pulse is delayed in time by T_s.

Example: Binary Bipolar PAM
Figure 12 shows a 4-ary PAM signal set using amplitudes $a_{0,0} = -A$, $a_{1,0} = A$. It shows a signal set using square pulses,

$$\phi_0(t) = p(t) = \begin{cases} 1/\sqrt{T_s}, & 0 \le t < T_s \\ 0, & o.w. \end{cases}$$

At the receiver, what we do is multiply the received signal $r(t)$ by $\phi_0(t)$ and integrate:

$$x = \int_{t=-\infty}^{\infty} r(t)\phi_0(t)dt$$

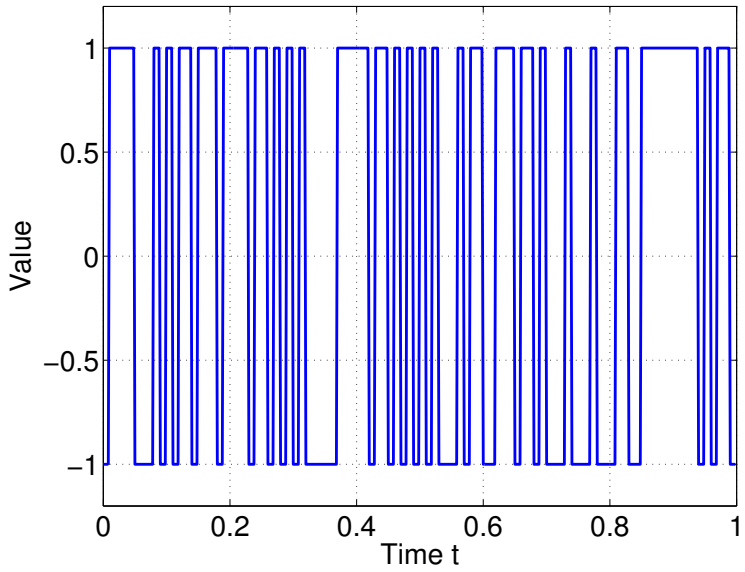


Figure 12: Example signal for binary bipolar PAM example for $A = \sqrt{T_s}$.

This is also called *correlation*. This needs to happen at each multiple of T_s. But there really are finite limits – let's say that the signal has a duration T_s, and then rewrite the integral as

$$x = \int_{t=0}^{T_s} r(t)\phi_0(t)dt \tag{15}$$

Essentially, this is as if we had multiple orthogonal waveforms, and thus our receiver is a bunch of parallel correlations, each time delayed by T_s, as shown in Figure 13. Note that my integral limits in Figure 13 may be wider if the pulse is wider than T_s (e.g., the nth branch would start before nT_s and end after $(n+1)T_s$).

In reality, instead of a correlation, we can equivalently do a matched filter. Equation (15) can be written as

$$x = \int_{t=0}^{T_s} r(t)h(T_s - t)dt$$

where $h(t) = \phi_0(T_s - t)$. (Plug in $(T_s - t)$ in this formula and the T_ss cancel out and only positive t is left.) This is the output of a convolution, taken at time T_s,

$$x = r(t) \star h(t)|_{t=T_s}$$

Or equivalently

$$x = r(t) \star \phi_0(T_s - t)|_{t=T_s}$$

Notes:

- $x(n)$ is the output of a matched filter at time nT_s.
- We might, for example, have a physical filter with the impulse response $\phi_0(T_s - t)$, in which case it would be easy to do a matched filter implementation.

Try out the Matlab code, `correlation_and_matched_filter_rx.m`, which is posted on Canvas, for some examples of a pulse shape, adding noise, and performing the filtering for a correlation and matched filter receiver.

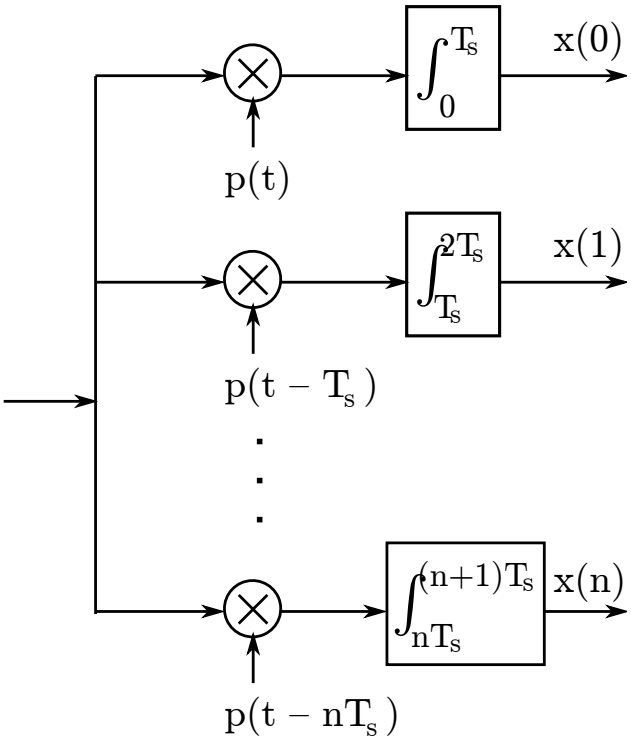


Figure 13: The pulse shape $p(t)$ is chosen to be orthogonal to itself at integer multiples of the time delay. Thus a PAM receiver might be implemented as a correlation receiver, in which each correlation (multiplication and integration) is done in parallel.

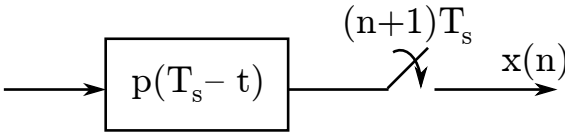


Figure 14: A matched filter receiver inputs $r(t)$ to a filter with impulse response $p(T_s - t)$, and then samples the output each multiple of T_s.

10 Inter-Symbol Interference (ISI)

1. Consider the spectrum of the ideal 1-D PAM system with a square pulse.
2. Consider the time-domain of the signal which is close to a rect function in the frequency domain.

We don't want either: (a) the pulse occupies too much spectrum, and (b) the pulse occupies too much time.

1. If we try 1. above and use FDMA (frequency division multiple access) then the interference is *out-of-band interference*.
2. If we try 2. above and put symbols right next to each other in time, our own symbols can experience interference called *inter-symbol interference* whenever we don't sample the output of the pulse shape filter at exactly the right time (or when the multipath in the radio channel make multiple pulses arrive at small time delays).

In reality, we want to compromise between (1) and (2) and experience only a small amount of both.

10.1 Nyquist Filtering

We want to extend the pulse $p(t)$ to be longer in duration than only between zero and T_s, but we want to do so in a way that preserves the property that $p(t)$ and $p(t-T_s)$, and for that matter, $p(t-nT_s)$ for integers $n \neq 0$, are orthogonal.

Key insight: we don't need $p(t)$ and $p(t-\tau)$ to be orthogonal for all real-valued times τ – only for $\tau = nT_s$, for integer n. In other words, the set:

$$\ldots, p(t+2T_s), p(t+T_s), p(t), p(t-T_s), p(t-2T_s), \ldots$$

form an orthonormal set. The Nyquist filtering condition is a frequency-domain rule that can be used to design arbitrary pulse shapes $p(t)$ that meet this condition.

I'm going to use frequency in Hertz f, rather than radial frequency ω, for this discussion, as it makes expressions a little easier.

Theorem: Nyquist Filtering
Proof: Let $r_p(t) = \int_{\tau=-\infty}^{\infty} p(\tau)p(\tau-t)d\tau$ be the autocorrelation function of pulse shape $p(t)$. A necessary and sufficient condition for $r_p(t)$ to satisfy

$$r_p(nT_s) = \begin{cases} 1, & n = 0 \\ 0, & \text{other integer } n \end{cases}$$

is that its Fourier transform $R_p(f)$ satisfy

$$\sum_{m=-\infty}^{\infty} R_p\left(f + \frac{m}{T_s}\right) = T_s$$

Proof: on page 677, Appendix A, of Rice book.

Note that $r_p(t)$ doesn't need to be any particular value at any real-valued t other than nT_s.

Some comments on what $R_p(f)$ could be:

- $R_p(f) = \text{rect}(fT_s)$, *i.e.*, exactly constant within $-\frac{1}{2T_s} < f < \frac{1}{2T_s}$ band and zero outside.
- It may bleed into the next 'channel' but the sum of

$$\cdots + R_p\left(f - \frac{1}{T_s}\right) + R_p(f) + R_p\left(f + \frac{1}{T_s}\right) + \cdots$$

must be constant across all f. But the neighboring frequency-shifted copy of $R_p(f)$ must be lower s.t. the sum is constant. See Figure 15.

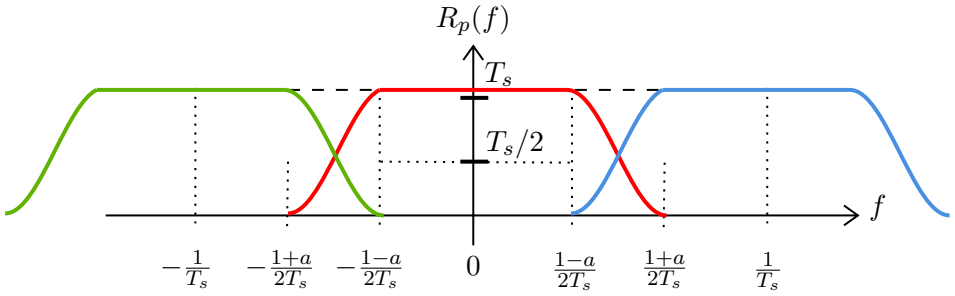


Figure 15: The Nyquist filtering criterion: $1/T_s$-frequency-shifted copies of $R_p(f)$ must add up to a constant (T_s). This plot shows $R_p(f)$ for the "square root raised cosine" pulse shape.

If $R_p(f)$ only bleeds into one neighboring channel (that is, $R_p(f) = 0$ for all $|f| > \frac{1}{T_s}$), we denote the difference between the ideal rect function and our $R_p(f)$ as $\Delta(f)$,

$$\Delta(f) = |R_p(f) - T_s\text{rect}(fT_s)|$$

then we can rewrite the Nyquist Filtering condition as,

$$\Delta\left(\frac{1}{2T_s} - f\right) = \Delta\left(\frac{1}{2T_s} + f\right), \quad \text{for all } -\frac{1}{T_s} \le f < \frac{1}{T_s}$$

Essentially, $R_p(f)$ is symmetric about $f = \frac{1}{2T_s}$ and $R_p(f) = \frac{T_s}{2}$, i.e., if it was folded twice at those lines it would line up.

Andy Bateman (*Digital Communications: Design for the Real World*, 1998) presents this condition as "A Nyquist channel response is characterized by the transfer function having a transition band that is symmetrical about a frequency equal to $0.5 \times 1/T_s$."

Activity: Do-it-yourself Nyquist filter. Take a sheet of paper and fold it in half on the longest side, and then in half the other direction. Cut a function in the thickest side (the edge that you just folded). Leave a tiny bit so that it is not completely disconnected into two halves. Unfold. Drawing a horizontal line for the frequency f axis, the middle is $0.5/T_s$, and the vertical axis it $R_p(f)$. One half (bottom or top) will be a plot of $R_p(f)$, and the other half will be an (inverse) plot of $R_p(f - 1/T_s)$.

10.2 How to get $p(t)$ from $R_p(f)$

Assuming now $R_p(f)$ meets the Nyquist filtering condition, how do we go from that to $p(t)$? The pulse shape $p(t)$ must be such that $r_p(t)$, the autocorrelation function, has the Fourier transform

$R_p(f)$. Recall that convolution in the time domain is multiplication in the frequency domain. And autocorrelation (correlation with the same function itself) is convolution with a time-reversed version of itself: $r_p(t) = p(t) \star p(-t)$. Let's assume that the pulse shape is real-valued and symmetric about $t = 0$ since that's the only kind of pulse shape I've ever seen used. Thus

$$R_p(f) = \mathfrak{F}\{p(t) \star p(-t)\} = P(f)P(f) = |P(f)|^2.$$

This means that $|P(f)| = |R_p(f)|^{1/2}$ or

$$p(t) = \mathfrak{F}^{-1}\left\{|R_p(f)|^{1/2}\right\}. \tag{16}$$

Technically we could have $|P(f)| = \pm|R_p(f)|^{1/2}$, but the sign is arbitrary, as long as the transmitter and receiver agree on it.

10.3 Square Root Raised Cosine Pulse Shapes

The raised-cosine function has an autocorrelation function of:

$$R_p(f) = \begin{cases} T_s, & 0 \le |f| \le \frac{1-\alpha}{2T_s} \\ \frac{T_s}{2}\left\{1 + \cos\left[\frac{\pi T_s}{\alpha}\left(|f| - \frac{1-\alpha}{2T_s}\right)\right]\right\}, & \frac{1-\alpha}{2T_s} \le |f| \le \frac{1+\alpha}{2T_s} \\ 0, & o.w. \end{cases} \tag{17}$$

where α is a parameter between 0 and 1 that indicates how quickly (α close to 0) or how slowly (α close to 1) the pulse shape's frequency content transitions from its maximum T_s to zero.

From (16) we need the square root of $R_p(f)$, which is this pulse shape is called the square root raised cosine (SRRC):

$$|R_p(f)|^{1/2} = \begin{cases} \sqrt{T_s}, & 0 \le |f| \le \frac{1-\alpha}{2T_s} \\ \sqrt{\frac{T_s}{2}\left\{1 + \cos\left[\frac{\pi T_s}{\alpha}\left(|f| - \frac{1-\alpha}{2T_s}\right)\right]\right\}}, & \frac{1-\alpha}{2T_s} \le |f| \le \frac{1+\alpha}{2T_s} \\ 0, & o.w. \end{cases} \tag{18}$$

Finally, taking the inverse Fourier transform of this, with some manipulation and use of our favorite Fourier transform properties and pairs, we get the $p(t)$ result in the Rice book (A.30):

$$p(t) = \frac{1}{\sqrt{T_s}} \frac{\sin\left(\frac{\pi(1-\alpha)t}{T_s}\right) + \frac{4\alpha t}{T_s}\cos\left(\frac{\pi(1+\alpha)t}{T_s}\right)}{\frac{\pi t}{T_s}\left[1 - \left(\frac{4\alpha t}{T_s}\right)^2\right]}, \tag{19}$$

where t is the time variable, and α and T_s are constants. Incidentally, this would be a good proof for a motivated student. Please note that the function is 0/0 for $t = 0$ which one can avoid by calculating it for some very small $t > 0$ (the engineering solution) or by using L'Hôpital's rule (the math solution).

Lecture 7

Today: (1) QAM, (2) PSK

- Reading for these notes: Rice [11] Chapter 5.2, 5.3
- Lecture videos for this lecture are at:
  `https://youtube.com/playlist?list=PLQuDEk4rPDOPFMfseH2UOsKMAEOwB_2Ob`

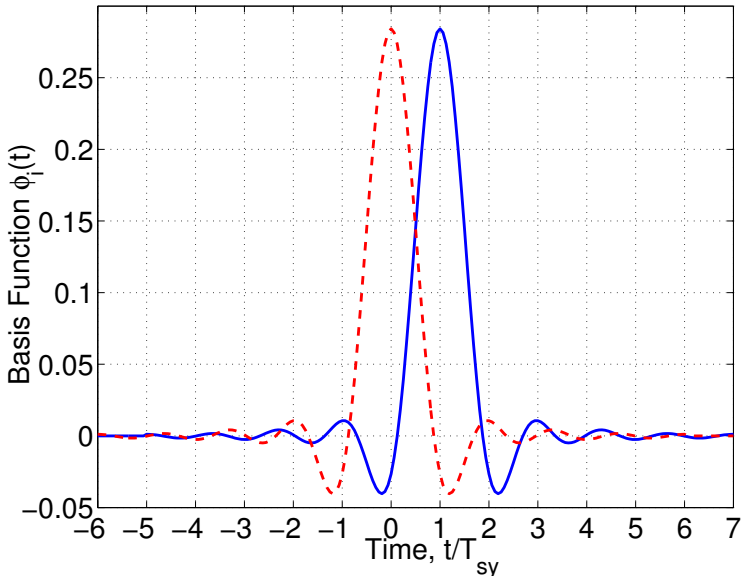


Figure 16: Two SRRC Pulses, delayed in time by nT_s for any integer n, are orthogonal to each other.

11 Quadrature Amplitude Modulation (QAM)

Quadrature amplitude modulation (QAM) is a two-dimensional bandpass signalling method which uses the in-phase and quadrature (cosine and sine, respectively) at the same frequency as the two basis functions. In other words, QAM's two basis functions are:

$$\begin{aligned} \phi_0(t) &= \sqrt{2}p(t)\cos(\omega_0 t) \\ \phi_1(t) &= -\sqrt{2}p(t)\sin(\omega_0 t) \end{aligned} \tag{20}$$

where $p(t)$ is a pulse shape that meets the Nyquist filtering theorem with support on $T_1 \leq t \leq T_2$ for some real constants $T_1 < T_2$. That is, $p(t)$ is only non-zero within that window, and $T_1 < 0$ and $T_2 > T_s$. (We remove the assumption that it is non-zero for all time, as this is not a realistic assumption.)

In most flow chart-type drawings of transmitters and receivers, frequency up-conversion and down-conversion are separate from pulse shaping, which is largely true to how the device operates. The definition of the orthonormal basis in (20) specifically considers a pulse shape at a frequency ω_0. We include it here because it is *critical* to see how, with the same pulse shape $p(t)$, we can have two orthogonal basis functions. (This is not intuitive!)

We have two restrictions on $p(t)$ that makes these two basis functions orthonormal:

- $p(t)$ is unit-energy.
- $p(t)$ is 'low pass'; that is, it has low frequency content compared to ω_0.

11.1 Showing Orthogonality

Let's show that the basis functions are orthogonal. Rather than assume that the waveforms are infinite in duration, let's put finite limits T_1 and T_2 on them, and say that they are zero before T_1

and after T_2. Starting the proof, and using $\sin(2A) = 2\cos A \sin A$,

$$\begin{aligned}\int_{t=-\infty}^{\infty} \phi_0(t)\phi_1(t)dt &= -\int_{T_1}^{T_2} \sqrt{2}p(t)\cos(\omega_0 t)\sqrt{2}p(t)\sin(\omega_0 t)dt \\ &= -2\int_{T_1}^{T_2} p^2(t)\cos(\omega_0 t)\sin(\omega_0 t)dt \\ &= -\int_{T_1}^{T_2} p^2(t)\sin(2\omega_0 t)dt.\end{aligned}$$

But we don't know $p(t)$, so we can't proceed. For a simple example, let $p(t) = c$ for a constant c , for $T_1 \leq t \leq T_2$. Are the two bases orthogonal?

$$\begin{aligned}\int_{t=-\infty}^{\infty} \phi_0(t)\phi_1(t)dt &= -c^2\int_{T_1}^{T_2} \sin(2\omega_0 t)dt \\ &= -c^2\left[-\frac{\cos(2\omega_0 t)}{2\omega_0}\right|_{T_1}^{T_2} \\ &= -c^2\frac{\cos(2\omega_0 T_2) - \cos(2\omega_0 T_1)}{2\omega_0} \qquad (21)\end{aligned}$$

Next, using the relationship,

$$\cos A - \cos B = -2\sin\frac{A+B}{2}\sin\frac{A-B}{2}, \qquad (22)$$

we can write (21) as,

$$\int_{t=-\infty}^{\infty} \phi_0(t)\phi_1(t)dt = c^2\frac{\sin(\omega_0(T_2+T_1))\sin(\omega_0(T_2-T_1))}{\omega_0}$$

We only need one of the sinusoids to be zero to make the functions orthogonal. There are two cases:

1. The pulse duration $T_2 - T_1$ is an integer number of periods. That is, $\omega_0(T_2 - T_1) = \pi k$ for some integer k. In this case, the right sin is zero, and so the correlation is exactly zero.

2. Otherwise, the numerator bounded above and below by +1 and -1, because it is a sinusoid. That is,
$$-\frac{c^2}{\omega_0} \leq \langle\phi_0(t), \phi_1(t)\rangle \leq \frac{c^2}{\omega_0}$$
Typically, ω_0 is a large number. For example, frequencies $2\pi\omega_0$ could be in the MHz or GHz ranges. Certainly, when we divide by numbers on the order of 10^6 or 10^9, we're going to get a very small inner product. For engineering purposes, then, $\phi_0(t)$ and $\phi_1(t)$ are orthogonal.

Finally, we can attempt the proof for the case of arbitrary pulse shape $p(t)$. In this case, we use the 'low-pass' assumption that the maximum frequency content of $p(t)$ is much lower than $2\pi\omega_0$. This assumption allows us to assume that $p^2(t)$ is nearly constant over the course of one cycle of the carrier sinusoid.

This is well-illustrated in Figure 5.3.1 in the Rice book (page 240). In this figure, we see a pulse modulated by a sine wave at frequency $2\pi\omega_0$. Zooming in on any few cycles, you can see that the pulse $p^2(t)$ is largely constant across each cycle. Thus, when we integrate $p^2(t)\sin(2\omega_0 t)$ across one cycle, we're going to end up with approximately zero.

11.2 Complex Baseband

With these two basis functions, M-ary QAM is defined as an arbitrary signal set $\mathbf{s}_0, \ldots, \mathbf{s}_{M-1}$, where each signal space vector $\mathbf{s}_m$ is two-dimensional:

$$\mathbf{s}_m = \begin{bmatrix} s_{m,0} \\ s_{m,1} \end{bmatrix}$$

The signal corresponding to symbol m in (M-ary) QAM is thus

$$\begin{aligned} s_m(t) &= a_{m,0}\phi_0(t) + a_{m,1}\phi_1(t) \\ &= a_{m,0}\sqrt{2}p(t)\cos(\omega_0 t) - a_{m,1}\sqrt{2}p(t)\sin(\omega_0 t) \\ &= \sqrt{2}p(t)\left[a_{m,0}\cos(\omega_0 t) - a_{m,1}\sin(\omega_0 t)\right] \end{aligned} \tag{23}$$

Note that we could also write the signal $s(t)$ as

$$\begin{aligned} s_m(t) &= \sqrt{2}p(t)\mathbb{R}\left\{a_{m,0}e^{j\omega_0 t} + ja_{m,1}e^{j\omega_0 t}\right\} \\ &= \sqrt{2}p(t)\mathbb{R}\left\{e^{j\omega_0 t}\left(a_{m,0} + ja_{m,1}\right)\right\} \end{aligned} \tag{24}$$

In many textbooks, you will see them write a QAM signal in shorthand as

$$s_m^{CB}(t) = p(t)(a_{m,0} + ja_{m,1})$$

This is called *complex baseband*. If you do the following operation you can recover the real signal $s_m(t)$ as

$$s_m(t) = \sqrt{2}\mathbb{R}\left\{e^{j\omega_0 t}s_m^{CB}(t)\right\}$$

In this notation symbol m is represented with a complex number

$$s_m^{CB} = a_{m,0} + ja_{m,1}$$

instead of a vector $\mathbf{s}_m = [s_{m,0}, s_{m,1}]^T$. You can think of the two versions (complex value, 2-D vector) as being equivalent. We will sometimes say the two components are real and the imaginary components. We also sometimes call the components the in-phase and quadrature components.

Many other books use complex baseband notation, so I include this primarily because you should be able to read other books and know what they're talking about.

11.3 Signal Constellations

The signal space representation $\mathbf{s}_m$ is given by

$$\mathbf{s}_m = [a_{m,0}, a_{m,1}]^T$$

for $m = 0, \ldots, M-1$.

- See Figure 5.3.3 in the Rice book for examples of square QAM [11]. These constellations use $M = 2^a$ for some **even** integer a, and arrange the points in a grid. One such diagram for $M = 64$ square QAM is also given here in Figure 17.
- Figure 5.3.4 shows examples of constellations which use $M = 2^a$ for some **odd** integer a, and arrange the points in a grid. These are either rectangular grids, or squares with the corners cut out, or more hexagonal grids.

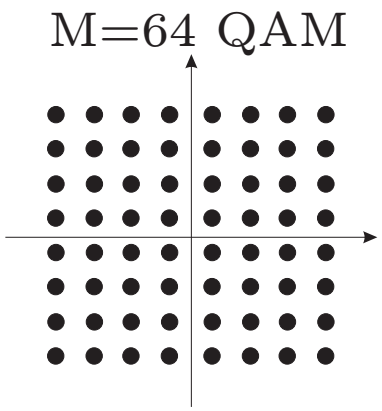


Figure 17: Square signal constellation for 64-QAM.

11.4 Angle and Magnitude Representation

You can plot $\mathbf{s}_m$ in signal space and see that it has a magnitude (distance from the origin) of $|\mathbf{s}_m| = \sqrt{a_{m,0}^2 + a_{m,1}^2}$ and angle of $\angle \mathbf{s}_m = \tan^{-1} \frac{a_{m,1}}{a_{m,0}}$. In the continuous time signal $s(t)$ this is

$$s(t) = \sqrt{2} p(t) |\mathbf{s}_m| \cos(\omega_0 t + \angle \mathbf{s}_m)$$

11.5 Average Energy in M-QAM

The average energy per symbol and average energy per bit are calculated as:

$$\begin{aligned} \mathcal{E}_s &= \frac{1}{M} \sum_{m=0}^{M-1} |\mathbf{s}_m|^2 \\ \mathcal{E}_b &= \frac{1}{M \log_2 M} \sum_{m=0}^{M-1} |\mathbf{s}_m|^2 \end{aligned} \tag{25}$$

where $\mathcal{E}_s$ is the average energy per symbol and $\mathcal{E}_b$ is the average energy per bit. We typically work from a constellation diagram, writing down the distances $|\mathbf{s}_m|^2$ for each symbol m, and then taking the average as given in (25). We'll work in class some examples of finding $\mathcal{E}_s$ and $\mathcal{E}_b$ in different constellation diagrams.

11.6 Phase-Shift Keying

Some implementations of QAM limit the constellation to include only signal space vectors with equal magnitude, *i.e.*,

$$|\mathbf{s}_0| = |\mathbf{s}_1| = \cdots = |\mathbf{s}_{M-1}|$$

The points $\mathbf{s}_m$ for $m = 0, \ldots, M-1$ are uniformly spaced on the unit circle. Some examples are shown in Figure 18.

QPSK $M = 4$ PSK is also called quadrature phase shift keying (QPSK), and is shown in Figure 18(a). Note that the rotation of the signal space diagram doesn't matter, so both 'versions' are identical in concept (although would be a slightly different implementation). Note how QPSK is the same as $M = 4$ square QAM.

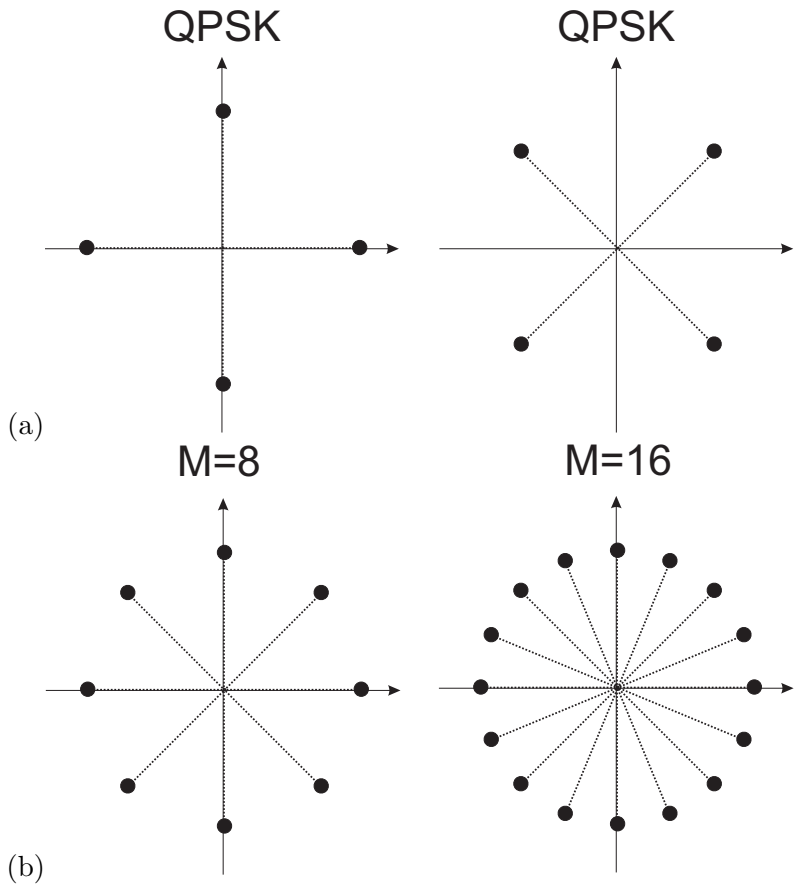


Figure 18: Signal constellations for (a) $M = 4$ PSK and (b) $M = 8$ and $M = 16$ PSK.

11.7 Systems which use QAM

See the Couch book [2], numerous Wikipedia pages, and the Rice book [11]:

- Digital Microwave Relay, various manufacturer-specific protocols. 6 GHz, and 11 GHz.
- Dial-up modems: use a $M = 16$ or $M = 8$ QAM constellation.
- DSL. G.DMT uses multicarrier (up to 256 carriers) methods (OFDM), and on each narrowband (4.3kHz) carrier, it can send up to 2^{15} QAM (32,768 QAM). G.Lite uses up to 128 carriers, each with up to $2^8 = 256$ QAM.
- Cable modems. Upstream: 6 MHz bandwidth channel, with 64 QAM or 256 QAM. Downstream: QPSK or 16 QAM.
- 802.11a, 802.11g: Adaptive modulation methods, use up to 64 QAM.
- 802.11ac, 11ax: uses up to 1024 QAM.
- Digital Video Broadcast (DVB): APSK used in ETSI standard.

11.8 Bandwidth of QAM, PAM, PSK

The bandwidth of QAM, PAM, and PSK are all determined by the bandwidth of the pulse used. For square root-raised cosine (SRRC) pulses, the null to null bandwidth is

$$B_T = \frac{1+\alpha}{T_s}$$

where α is the "rolloff" factor or "excess bandwidth" parameter of the SRRC pulse. Recall if $\alpha = 0$ then the pulse shape is a rect in the frequency domain and a sinc in the time domain, and has the smallest bandwidth.

Lecture 8

Today: (1) OQPSK (2) FSK

- Reading for these notes: Proakis & Salehi [10] FSK section, Rice [11] Sections 5.4, 5.5.
- Lecture videos for this lecture are at:
  `https://youtube.com/playlist?list=PLQuDEk4rPDOO6rDnEC05vzWWr-no_ncU4`

11.9 Complex Baseband for QAM

First, a short review. Last lecture, we described QAM and PSK by their use of two basis functions at the same frequency ω_0:

$$\begin{aligned} \phi_0(t) &= \sqrt{2}p(t)\cos(\omega_0 t) \\ \phi_1(t) &= -\sqrt{2}p(t)\sin(\omega_0 t) \end{aligned}$$

We outlined the proof that these two are orthogonal. We looked at some example symbol constellations and calculated their average symbol energies. We showed that the bandwidth is a function of the pulse shape, for example, for SRRC pulse shapes, $B = (1+\alpha)/T_s$.

Our continuous-time signal will be a sequence of amplitude-scaled versions of these bases:

$$s(t) = \sqrt{2}\sum_n \left[a_0(n)p(t-nT_s)\cos(\omega_0 t) - a_1(n)p(t-nT_s)\sin(\omega_0 t)\right], \tag{26}$$

where $a_k(n)$ is the amplitude of ϕ_0 used during the nth symbol period. We often use $I(t)$, *i.e.*, the in-phase component, as the part of the signal multiplying the $\cos(\omega_0 t)$, and $Q(t)$, *i.e.*, the quadrature component, as the part of the signal multiplying the $\sin(\omega_0 t)$:

$$I(t) = \sum_n a_0(n)p(t-nT_s), \quad Q(t) = \sum_n a_1(n)p(t-nT_s).$$
$$s(t) = \sqrt{2}I(t)\cos(\omega_0 t) - \sqrt{2}Q(t)\sin(\omega_0 t).$$

Alternatively,

$$s(t) = \sqrt{2}\sum_n |a(n)|p(t-nT_s)\cos\left(\omega_0 t + \angle a(n)\right), \tag{27}$$

where $|a(n)| = \sqrt{a_0^2(n)+a_1^2(n)}$ and $\angle a(n) = \tan^{-1}\frac{a_1(n)}{a_0(n)}$. This is a form that shows that $s(t)$ during any symbol has an "envelope" or amplitude $|a(n)|$ and a phase angle $\angle a(n)$. The envelope and angle are time-varying functions because the pulse amplitude is not constant.

We can readily connect this with our prior discussion of complex baseband. The time domain plot can be converted by taking out the $\sqrt{2}$ and putting the rest inside a Real operator:

$$s(t) = \sqrt{2}\mathbb{R}\left\{e^{j\omega_0 t}s_{CB}(t)\right\}, \tag{28}$$

where,

$$s_{CB}(t) = \sum_n |a(n)|e^{j\angle a(n)}p(t-nT_s). \tag{29}$$

That is, the time domain signal in complex baseband is a sum of time delayed pulse shapes, each weighted by a complex value $a(n)$.

When we plot the in-phase (real) vs. the quadrature (imaginary), in effect collapsing time, Rice calls it the "phase trajectory plot". Actually, it shows the trajectory of both the phase and envelope. An example for QPSK is shown in Figure 19.

12 Offset QPSK (OQPSK)

12.1 Motivation

One thing that makes a transmitter more power hungry is the need for a linear amplifier. An amplifier uses DC power to take a bandpass input signal and increase its amplitude at the output. If P_{DC} is the input power (*e.g.*, from battery) to the amplifier, and P_{out} is the output signal power, the power efficiency is rated as $\eta_P = P_{out}/P_{DC}$.

Truly linear amplifiers (class A) amplifiers are at most 50% power efficient. We are interested in the question, what signals can be amplified with nonlinear (class C) amplifiers that are around 90% power efficient? The answer is that "constant envelope" signals can. These are signals that the peak envelope is never much higher than the average envelope. The phase trajectory plot of

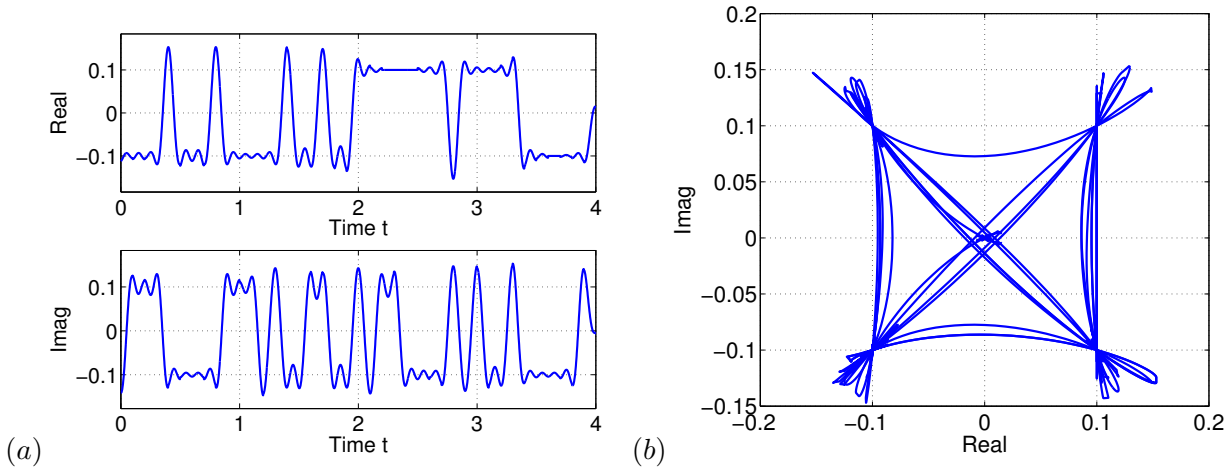


Figure 19: Matlab simulation of the complex baseband form of $s(t)$ for QPSK when using a SRRC pulse, showing the (a) in-phase and quadrature components. In the phase trajectory plot in (b), time is removed and we plot the in-phase vs. the quadrature components.

a constant envelope signal will be nearly a circle, and it will never go through or near the origin (envelope of zero).

Not all modulations are constant envelope, so this involves some modulation limitations. In order to double the power efficiency, battery-powered transmitters are often willing to use constant envelope modulations so that they can use Class C amplifiers. They can do this if their output signal has constant envelope.

12.2 Definition

The reason that QPSK is not constant envelope is that when the phase angle changes 180 degrees from one symbol to the next, the envelope will go through zero. See Figure 19 to see this graphically.

Offset QPSK solves this problem by simply shifting one of the basis functions forward by half a symbol period, *i.e.*, $T_s/2$. The orthonormal basis for OQPSK are thus:

$$\begin{aligned} \phi_0(t) &= \sqrt{2}p(t)\cos(\omega_0 t) \\ \phi_1(t) &= -\sqrt{2}p(t-T_s/2)\sin(\omega_0 t) \end{aligned}$$

We still only offset subsequent symbols by T_s, so the transmitted signal is:

$$s(t) = \sqrt{2}\sum_n \left[a_0(n)p(t-nT_s)\cos(\omega_0 t) - a_1(n)p\left(t-\left(n+\frac{1}{2}\right)T_s\right)\sin(\omega_0 t)\right],$$

or equivalently the (complex baseband) in-phase and quadrature components are,

$$I(t) = \sqrt{2}\sum_n a_0(n)p(t-nT_s), \quad Q(t) = \sqrt{2}\sum_n a_1(n)p\left(t-\left(n+\frac{1}{2}\right)T_s\right).$$

Compared to (26), the in-phase component of $s(t)$ in OQPSK does not go through zero at the same times that the quadrature component does, since the pulse functions $p()$ are offset in time by half a symbol period.

Another way to look at this is to view the phase trajectory plot in Figure 20. The signal never switches from its current constellation point to the one 180^o opposite – either the in-phase or quadrature component changes during any multiple of $T_s/2$, but NOT both.

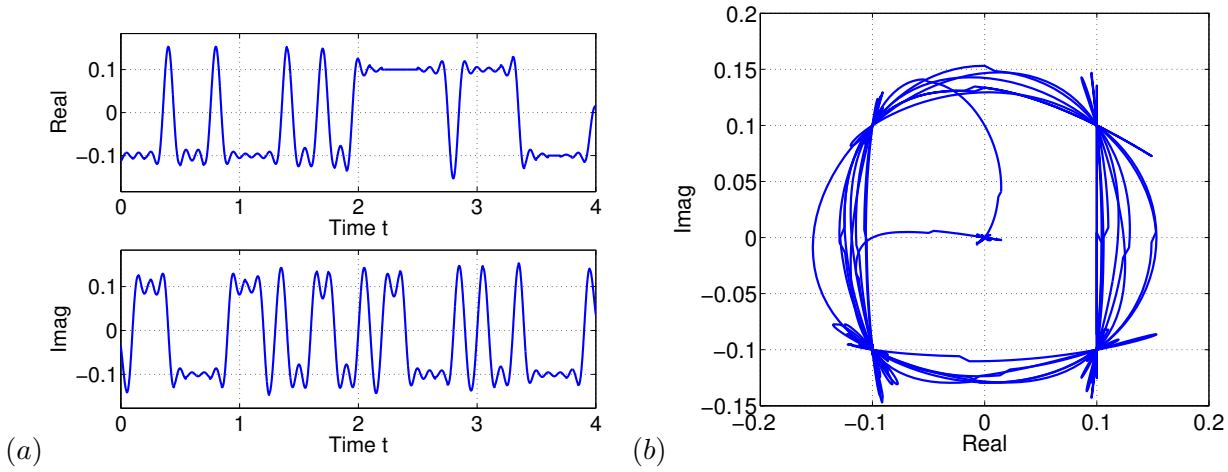


Figure 20: Matlab simulation of $s(t)$ for OQPSK, showing the (a) in-phase (cosine) and quadrature (sine) components. The phase trajectory plot in (b) shows the nearly constant envelope of OQPSK, compared to that for QPSK in Figure 19.

At the receiver, we just need to delay the sampling on the in-phase component of the signal half of a sample period with respect to the quadrature signal. The new transmitted signal takes the same bandwidth and average power, and as we will show later in the course, the same E_b/N_0 vs. probability of bit error performance. However, the envelope $|s(t)|$ is largely constant. Compare Figures 19 and 20 to see the differences between QPSK and OQPSK.

13 Frequency Shift Keying (FSK)

In frequency shift keying, symbols are selected to be sinusoids with frequency selected among a set of M different frequencies $\{f_0, f_1, \ldots f_{M-1}\}$. Note that in FSK, the number of basis functions equals the number of symbols. Consider $f_k = f_c + k\Delta f$, and thus

$$\phi_k(t) = \sqrt{2}p(t)\cos(\omega_0 t + 2\pi k\Delta f t) \tag{30}$$

where $p(t)$ is our pulse shape. We want these $\phi_k(t)$ to form an orthonormal basis. How can we make them orthonormal? First, note that these basis functions are unit energy, just like we've shown that the QAM basis functions are unit energy. Next, what do we get when we find the inner product of two different basis functions, $\phi_k(t)$ and $\phi_m(t)$ for $m \neq k$? Let's assume that $p(t)$ is a

rectangular pulse with value $\frac{1}{\sqrt{T_s}}$ between 0 and T_s:

$$\begin{aligned}
\langle \phi_k(t), \phi_m(t) \rangle &= \int_{t=0}^{T_s} (2/T_s) \cos(\omega_0 t + 2\pi k \Delta f t) \cos(\omega_0 t + 2\pi m \Delta f t) dt \\
&= 1/T_s \int_{t=0}^{T_s} \cos(2\pi(k-m)\Delta f t) dt + \\
&\quad 1/T_s \int_{t=0}^{T_s} \cos(2\omega_0 t + 2\pi(k+m)\Delta f t) dt \\
&= 1/T_s \left[\frac{\sin(2\pi(k-m)\Delta f t)}{2\pi(k-m)\Delta f} \right|_{t=0}^{T_s} \\
&= \frac{\sin(2\pi(k-m)\Delta f T_s)}{2\pi(k-m)\Delta f T_s}
\end{aligned}$$

So they are orthogonal if $2\pi(k-m)\Delta f T_s$ is a multiple of π. (They are also approximately orthogonal if Δf is really big, but we don't want to waste spectrum.) For general $k \neq m$, this requires that $\Delta f T_s = n/2$, *i.e.*,

$$\Delta f = n \frac{1}{2T_s} = n \frac{f_s}{2} \tag{31}$$

for integer n. (Otherwise, no they're not.)

Thus we need to plug into (30) for $\Delta f = n\frac{1}{2T_s}$ for some integer n in order to have an orthonormal basis. What n? In practice, we either use an n of 1 or 2. Using $n = 1$ is called *minimum shift keying* (MSK) since it is the minimum frequency spacing. Use of $n = 2$ is historically more common because it can be implemented with a non-coherent receiver, as we discuss below.

Signal space vectors $\mathbf{s}_m$ are given by

$$\begin{aligned}
\mathbf{s}_0 &= [A, 0, \ldots, 0] \\
\mathbf{s}_1 &= [0, A, \ldots, 0] \\
&\vdots \\
\mathbf{s}_{M-1} &= [0, 0, \ldots, A]
\end{aligned}$$

What is the average energy per symbol? This means that $A = \sqrt{\mathcal{E}_s}$.

For $M = 2$ and $M = 3$ these vectors are plotted in Figure 21.

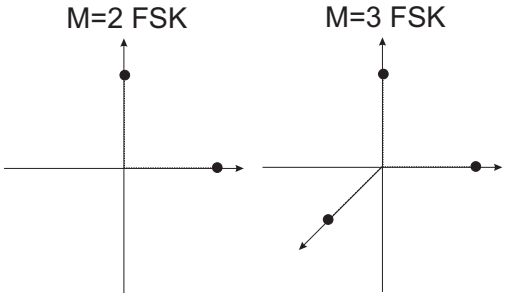


Figure 21: Signal space diagram for $M = 2$ and $M = 3$ FSK modulation.

13.1 Transmission of FSK

FSK can be seen as a sum of M different carrier signals, each multiplied by a pulse shape $p(t)$. In practice, FSK signals are usually generated from a single VCO, as seen in Figure 22.

Def'n: *Voltage Controlled Oscillator (VCO)*
A sinusoidal generator with frequency that linearly proportional to an input voltage.

Note that we don't need to (and don't want to) send square wave input into the VCO. The transition can be set to smoothly switch from one frequency to the next by pulse shaping, i.e., with $p(t)$ a slowly time-varying function.

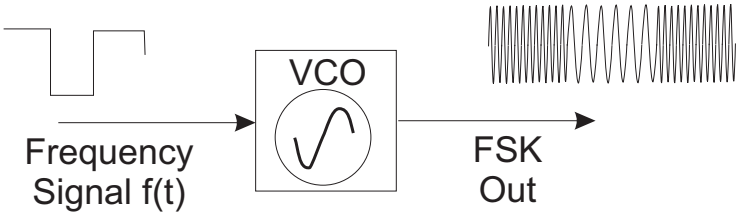


Figure 22: Block diagram of a binary FSK transmitter.

13.2 Reception of FSK

FSK reception is either phase coherent or phase non-coherent. Here, there are M possible carrier frequencies, so we'd need to know and be synchronized to M different phases θ_i, one for each symbol frequency:

$$\begin{aligned}
&\cos(\omega_0 t + \theta_0) \\
&\cos(\omega_0 t + 2\pi\Delta f t + \theta_1) \\
&\vdots \\
&\cos(\omega_0 t + 2\pi(M-1)\Delta f t + \theta_{M-1})
\end{aligned}$$

13.3 Coherent Reception

FSK reception can be done via a correlation receiver, just as we've seen for previous modulation types.

Each phase θ_k is estimated to be $\hat{\theta}_k$ by a separate phase-locked loop (PLL).

As M gets high, coherent detection becomes difficult. These M PLLs must operate even though they can only synchronize when their symbol is sent, $1/M$ of the time (assuming equally-probable symbols). Also, having M PLLs is a drawback.

13.4 Non-coherent Reception

Notice that in Figure 21, the sign or phase of the sinusoid is not very important – only one symbol exists in each dimension. In non-coherent reception, we just measure the energy in each frequency.

This is more difficult than it sounds, though – we have a fundamental problem. As we know, for every frequency, there are two orthogonal functions, cosine and sine (see QAM and PSK). Since

we will not know the phase of the received signal, we don't know whether or not the energy at frequency f_k correlates highly with the cosine wave or with the sine wave. If we only correlate it with one (the sine wave, for example), and the phase makes the signal the other (a cosine wave) we would get a inner product of zero!

The solution is that we need to correlate the received signal with both a sine and a cosine wave at the frequency f_k. This will give us two inner products, lets call them x_k^I using the capital I to denote in-phase and x_k^Q with Q denoting quadrature.

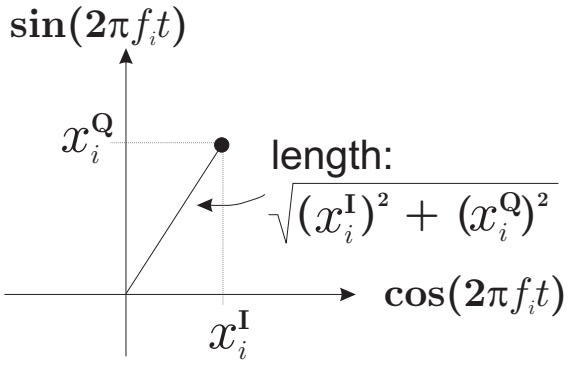


Figure 23: The energy in a non-coherent FSK receiver at one frequency f_k is calculated by finding its correlation with the cosine wave (x_k^I) and sine wave (x_k^Q) at the frequency of interest, f_k, and calculating the squared length of the vector $[x_k^I, x_k^Q]^T$.

The energy at frequency f_k, that is,

$$\mathcal{E}_{f_k} = (x_k^I)^2 + (x_k^Q)^2$$

is calculated for each frequency f_k, $k = 0, 1, \ldots, M-1$, and the detector picks the k that maximizes $\mathcal{E}_{f_k}$.

However, energy detection works only when $n = 2$ in (31). (Showing this might be a good homework or exam problem.) That is, $\Delta f = 1/T_s$. Some FSK systems trade off the extra bandwidth usage in order to allow for a less complicated receiver design (an energy detector).

13.5 Bandwidth of FSK

Carson's rule is used to calculate the bandwidth of FM signals. For M-ary FSK, it tells us that the approximate bandwidth is,

$$B_{M-FSK} = (M-1)\Delta f + B_{p(t)}$$

where $B_{p(t)}$ is the two-sided bandwidth of the pulse shape. For root raised-cosine pulse shaping, the null-to-null bandwidth is $B_{p(t)} = (1+\alpha)/T_s$.

Lecture 9

Today: (1) OFDM and Multicarrier Modulation, (2) Probability in Digital Comms, (3) Detection Threshold Activity

- Reading for these notes: Rice [11] Sections 5.5 and 4.1–4.3.
- Lecture videos for this lecture are at:
  `https://youtube.com/playlist?list=PLQuDEk4rPDOOwxVc4vQEI4fyg9_06XiMM`

14 Orthogonal Frequency Division Multiplexing (OFDM)

This is Section 5.5 in the Rice book.

In FSK, we use a single basis function at each of different frequencies. In QAM, we use two basis functions at the same frequency. *Multicarrier modulation* is the combination:

$$\begin{aligned}
\phi_{0,c}(t) &= \sqrt{2}p(t)\cos(\omega_0 t) \\
\phi_{0,s}(t) &= -\sqrt{2}p(t)\sin(\omega_0 t) \\
\phi_{1,c}(t) &= \sqrt{2}p(t)\cos(\omega_0 t + 2\pi\Delta f t) \\
\phi_{1,s}(t) &= -\sqrt{2}p(t)\sin(\omega_0 t + 2\pi\Delta f t) \\
&\vdots \\
\phi_{B-1,c}(t) &= \sqrt{2}p(t)\cos(\omega_0 t + 2\pi(B-1)\Delta f t) \\
\phi_{B-1,s}(t) &= -\sqrt{2}p(t)\sin(\omega_0 t + 2\pi(B-1)\Delta f t)
\end{aligned}$$

where $\Delta f = \frac{1}{T_s}$ and $B = K/2$.

In multicarrier modulation, we call the two waveforms (sine and cosine) at one frequency a "subcarrier". There are $B = K/2$ subcarriers. Multi-carrier modulation is a general type of modulation, of which *orthogonal frequency division multiplexing* (OFDM) is a specific version. OFDM uses the rectangular pulse, non-zero only between 0 and T_s. OFDM is thus represented as:

$$\begin{aligned}
\phi_{0,c}(t) &= \begin{cases} \sqrt{\frac{2}{T_s}}\cos(\omega_0 t), & 0 \le t \le T_s \\ 0, & o.w. \end{cases} \\
\phi_{0,s}(t) &= \begin{cases} -\sqrt{\frac{2}{T_s}}\sin(\omega_0 t), & 0 \le t \le T_s \\ 0, & o.w. \end{cases} \\
\phi_{1,c}(t) &= \begin{cases} \sqrt{\frac{2}{T_s}}\cos(\omega_0 t + 2\pi\Delta f t), & 0 \le t \le T_s \\ 0, & o.w. \end{cases} \\
\phi_{1,s}(t) &= \begin{cases} -\sqrt{\frac{2}{T_s}}\sin(\omega_0 t + 2\pi\Delta f t), & 0 \le t \le T_s \\ 0, & o.w. \end{cases} \\
&\vdots \\
\phi_{B-1,c}(t) &= \begin{cases} \sqrt{\frac{2}{T_s}}\cos(\omega_0 t + 2\pi(B-1)\Delta f t), & 0 \le t \le T_s \\ 0, & o.w. \end{cases} \\
\phi_{B-1,s}(t) &= \begin{cases} -\sqrt{\frac{2}{T_s}}\sin(\omega_0 t + 2\pi(B-1)\Delta f t), & 0 \le t \le T_s \\ 0, & o.w. \end{cases}
\end{aligned}$$

where again $\Delta f = \frac{1}{T_s}$.

You've already shown that the B cosine basis functions are mutually orthonormal while separated by $\Delta f = 1/T_s$. You will show in your Homework 4 that any two cosine and sine basis functions

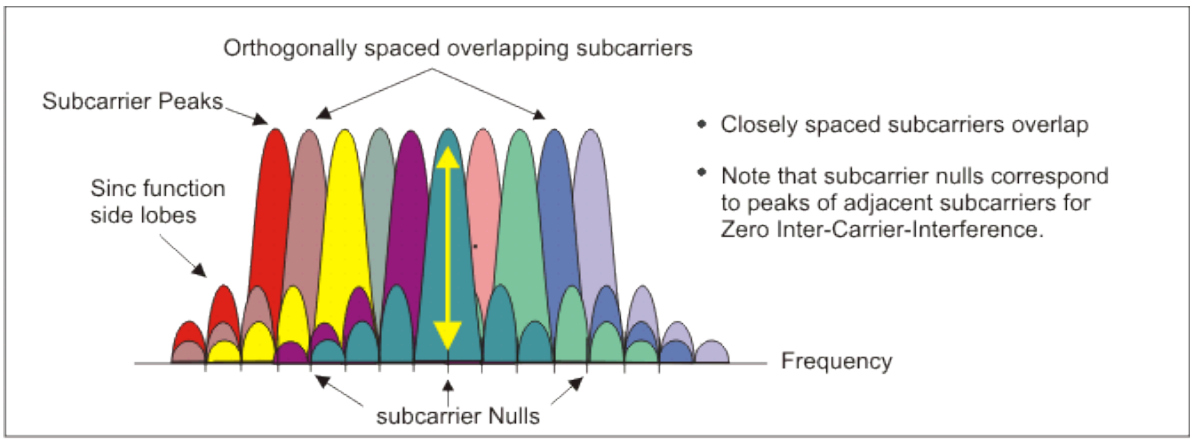


Figure 24: Graphical depiction of the power spectral density of an OFDM signal, from Keysight, "Concepts of Orthogonal Frequency Division Multiplexing (OFDM) and 802.11 WLAN", `https://cutt.ly/MkNXbl0`.

are orthonormal. Note we have $2M$ basis functions here in the same bandwidth as non-coherent M-ary FSK.

The signal on subcarrier k during symbol n for OFDM might be represented as:

$$x_k(t) = \sqrt{\frac{2}{T_s}} \left[a_{k,I}(n) \cos(\omega_0 t + 2\pi k \Delta f t) - a_{k,Q}(n) \sin(\omega_0 t + 2\pi k \Delta f t) \right]$$

On the kth channel, the signal could be described as some kind of QAM or PSK modulation. Regardless, over all channels, the modulation is called OFDM. The OFDM signal of the sum of all K subcarrier signals during symbol n might then be represented as

$$\begin{aligned} x(t) &= \sqrt{\frac{2}{T_s}} \mathbb{R} \left\{ \sum_{k=0}^{B-1} \left[a_{k,I}(n) + j a_{k,Q}(n) \right] e^{j(\omega_0 + 2\pi k \Delta f) t} \right\} \\ x(t) &= \sqrt{\frac{2}{T_s}} \mathbb{R} \left\{ e^{j\omega_0 t} \sum_{k=0}^{B-1} A_n[k] e^{j2\pi k t / T_s} \right\} \end{aligned} \tag{32}$$

where $A_n[k] = a_{k,I}(n) + j a_{k,Q}(n)$.

Does this look like an inverse discrete Fourier transform (DFT)? Recall from Lecture 4 we had

$$x(n) = \frac{1}{N} \sum_{k=0}^{N-1} X[k] e^{j \frac{2\pi}{N} nk}. \tag{33}$$

Renaming $X[k] \to A_n[k]$, $N \to B$ and $n \to t$, realizing that n/N is the same as t/T_s for a discrete-valued t that are the sample times $0 < t < T_s$,

$$x(t) = \frac{1}{B} \sum_{k=0}^{B-1} A_n[k] e^{j2\pi k t / T_s}. \tag{34}$$

If this derivation makes sense, than you can see why it might be possible to use an inverse DFT to generate the transmitted signal. The problem so far is that the inverse DFT (or the DFT for that

matter) is difficult to calculate in high speed digital hardware. Remember that we're hoping to transmit signals with 10s of MHz of bandwidth, and (32) is proposing many floating point multiplies and adds per sample.

FFT implementation: There is a particular implementation of the transmitter and receiver that use FFT/IFFT operations. This avoids having K independent transmitter chains and receiver chains. The FFT implementation (and the speed and ease of implementation of the FFT in hardware) is why OFDM is popular.

We transmit many more bits per symbol than possible in M-ary FSK. Since the carriers in OFDM are orthogonal and mostly at different frequencies, the signal is like FSK. But within each subcarrier, the cosine and sine bases make us think of QAM modulation. So what happens in OFDM is that, rather than transmitting on one of the K basis functions at a given time (like in FSK) we transmit QAM modulated information in parallel on all B subcarriers simultaneously.

For example, consider sending 16-square QAM on each subcarrier. We might draw the constellation diagram for any three of the basis functions at a time, and we'd get Figure 25.

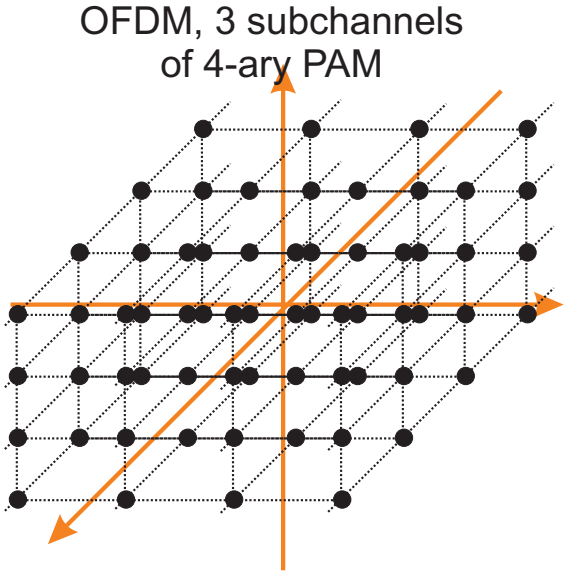


Figure 25: Constellation diagram for 3 of the basis functions in an OFDM signal which has 16-square QAM on each subcarrier.

There are two additional things that reduce the efficiency of OFDM:

1. There is ringing in OFDM in the frequency domain from the use of the rectangular pulse, but within the OFDM signal, the ringing doesn't cause a problem because the K basis functions are all mutually orthogonal. Outside of the subcarriers' bands (0 through $(B-1)\Delta f$), there are significant sidelobes as shown in Figure 24 because of the use of the rect function as the pulse shape, so extra bandwidth needs to be used to protect other out-of-band signals from interference from OFDM signals.

2. The FFT assumes that the signal is periodic. Its calculation is not equal to the DFT if the signal is aperiodic. To allow the FFT to be closer to the DFT, we repeat part of the end of the signal before its start, in the time domain. This is called the cyclic prefix. But since it does not contain additional information, it is essentially wasted time.

3. The receiver still needs a phase locked loop to match to the frequency of the transmitted signal. We typically help it by including *pilot tones* in a few subcarriers. Pilot signals are unmodulated subcarriers that are transmitted with the signal. Those pilot tones don't carry data, so they reduce the efficiency of OFDM to some extent.

Example: 802.11a
IEEE 802.11a uses OFDM with 52 subcarriers. Four of the subcarriers are reserved for pilot tones, so effectively 48 subcarriers are used for data. Each data subcarrier can be modulated in different ways. One example is to use 16 square QAM on each subcarrier (which is 4 bits per symbol per subcarrier). The symbol rate in 802.11a is 250k/sec. Thus the bit rate is

$$250 \times 10^3 \frac{\text{OFDM symbols}}{\text{sec}} 48 \frac{\text{subcarriers}}{\text{OFDM symbol}} 4 \frac{\text{coded bits}}{\text{subcarrier}} = 48 \frac{\text{Mbits}}{\text{sec}}$$

15 Probability in Digital Communications

Question: What is random about digital communications signals? Why do we need probabilistic analysis tools in the design of digital communications systems?
Solution: Here are some of my ideas, but of course there are others:

1. The data being sent
2. Additive noise
3. Interference from competing systems
4. Multipath fading
5. Doppler
6. Transmit signal imperfections
7. The timing of when a packet starts
8. Transmit power
9. Whether a receiver is awake or in sleep state at any time
10. Frequency offset

In general, random variables are (typically) unknown. The use of probability theory and analysis is to allow us to quantify what can be known about those random systems, and to use that in engineering design of those systems.

15.1 Distributions

Probability distributions give us a tool to quantify how likely particular a range of values is, or even a particular value is, of a random variable.

For two random variables X_1 and X_2 that are both in S, the range of possible values for the random variables,

- Joint CDF: $F_{X_1,X_2}(x_1,x_2) = P\left[\{X_1 \le x_1\} \cap \{X_2 \le x_2\}\right]$ It is the probability that both events happen simultaneously.
- Joint pmf: $p_{X_1,X_2}(x_1,x_2) = P\left[\{X_1 = x_1\} \cap \{X_2 = x_2\}\right]$ It is the probability that both events happen simultaneously.
- Joint pdf: $f_{X_1,X_2}(x_1,x_2) = \frac{\partial^2}{\partial x_1 \partial x_2} F_{X_1,X_2}(x_1,x_2)$

The (pdf / pmf) (integrates / sums) to one, and is non-negative. The CDF is non-negative and non-decreasing, with

$$\begin{aligned} \lim_{x_i \to -\infty} F_{X_1,X_2}(x_1,x_2) &= 0, \text{ and} \\ \lim_{x_1,x_2 \to +\infty} F_{X_1,X_2}(x_1,x_2) &= 1. \end{aligned} \tag{35}$$

To find the probability of an event, you integrate. For example, for event $B \in S$,

- Discrete case: $P[B] = \sum \sum_{(X_1,X_2)\in B} p_{X_1,X_2}(x_1,x_2)$
- Continuous Case: $P[B] = \int \int_{(x_1,x_2)\in B} f_{X_1,X_2}(x_1,x_2) dx_1 dx_2$

The marginal distributions are:

- Marginal pmf: $p_{X_2}(x_2) = \sum_{x_1 \in S_{X_1}} p_{X_1,X_2}(x_1,x_2)$
- Marginal pdf: $f_{X_2}(x_2) = \int_{x_1 \in S_{X_1}} f_{X_1,X_2}(x_1,x_2) dx_1$

Two random variables X_1 and X_2 are independent if and only if for all x_1 and x_2,

- $p_{X_1,X_2}(x_1,x_2) = p_{X_1}(x_1) p_{X_2}(x_2)$
- $f_{X_1,X_2}(x_1,x_2) = f_{X_2}(x_2) f_{X_1}(x_1)$

15.2 Random Vectors

Def'n: *Random Vector*
A random vector is a list of multiple random variables X_1, X_2, ..., X_n,

$$\mathbf{X} = [X_1, X_2, \ldots, X_n]^T$$

Here are models of random vectors:

1. The CDF of random vector $\mathbf{X}$ is $F_{\mathbf{X}}(\mathbf{x}) = F_{X_1,\ldots,X_n}(x_1,\ldots,x_n) = P[X_1 \le x_1, \ldots, X_n \le x_n]$.
2. The pmf of a discrete random vector $\mathbf{X}$ is $p_{\mathbf{X}}(\mathbf{x}) = p_{X_1,\ldots,X_n}(x_1,\ldots,x_n) = P[X_1 = x_1, \ldots, X_n = x_n]$.
3. The pdf of a continuous random vector $\mathbf{X}$ is $f_{\mathbf{X}}(\mathbf{x}) = f_{X_1,\ldots,X_n}(x_1,\ldots,x_n) = \frac{\partial^n}{\partial x_1 \cdots \partial x_n} F_{\mathbf{X}}(\mathbf{x})$.

15.3 Conditional Distributions

Given event $C \in S$ which has $P[C] > 0$, the joint probability conditioned on event C is

- Discrete case:
$$p_{X_1,X_2|C}(x_1,x_2) = \begin{cases} \frac{p_{X_1,X_2}(x_1,x_2)}{P[C]}, & (X_1,X_2) \in C \\ 0, & o.w. \end{cases}$$
- Continuous Case:
$$f_{X_1,X_2|C}(x_1,x_2) = \begin{cases} \frac{f_{X_1,X_2}(x_1,x_2)}{P[B]}, & (X_1,X_2) \in C \\ 0, & o.w. \end{cases}$$

Given random variables X_1 and X_2,

- Discrete case. The conditional pmf of X_1 given $X_2 = x_2$, where $p_{X_2}(x_2) > 0$, is
$$p_{X_1|X_2}(x_1|x_2) = p_{X_1,X_2}(x_1,x_2)/p_{X_2}(x_2)$$
- Continuous Case: The conditional pdf of X_1 given $X_2 = x_2$, where $f_{X_2}(x_2) > 0$, is
$$f_{X_1|X_2}(x_1|x_2) = f_{X_1,X_2}(x_1,x_2)/f_{X_2}(x_2)$$

Def'n: *Bayes' Law*
Bayes' Law is a reformulation of he definition of the marginal pdf. It is written either as:

$$f_{X_1,X_2}(x_1,x_2) = f_{X_2|X_1}(x_2|x_1)f_{X_1}(x_1)$$

or

$$f_{X_1|X_2}(x_1|x_2) = \frac{f_{X_2|X_1}(x_2|x_1)f_{X_1}(x_1)}{f_{X_2}(x_2)}$$

15.4 Simulation of Digital Communication Systems

A simulation of a digital communication system is often used to estimate a bit error rate. Each bit can either be demodulated without error, or with error. Thus the simulation of one bit is a Bernoulli trial. This trial E_i is in error ($E_1 = 1$) with probability p_e (the true bit error rate) and correct ($E_i = 0$) with probability $1 - p_e$. What type of random variable is E_i?

Simulations run many bits, say N bits through a model of the communication system, and count the number of bits that are in error. Let $G = \sum_{i=1}^{N} E_i$, and assume that $\{E_i\}$ are independent and identically distributed (i.i.d.).

1. What type of random variable is G?
2. What is the pmf of G?
3. What is the mean and variance of G?

Solution: E_i is called a Bernoulli random variable and G is called a Binomial random variable, with pmf

$$p_G(g) = \binom{N}{g} p_e^g (1-p_e)^{N-g}$$

The mean of G is the the same as the mean of the sum of $\{E_i\}$,

$$\begin{aligned} E_G[G] &= E_{\{E_i\}}\left[\sum_{i=1}^{N} E_i\right] = \sum_{i=1}^{N} E_{E_i}[E_i] \\ &= \sum_{i=1}^{N} [(1-p)\cdot 0 + p\cdot 1] = Np \end{aligned}$$

We can find the variance of G the same way:

$$\begin{aligned} \mathrm{Var}_G[G] &= \mathrm{Var}_{\{E_i\}}\left[\sum_{i=1}^{N} E_i\right] = \sum_{i=1}^{N} \mathrm{Var}_{E_i}[E_i] \\ &= \sum_{i=1}^{N} \left[(1-p)\cdot(0-p)^2 + p\cdot(1-p)^2\right] \\ &= \sum_{i=1}^{N} \left[(1-p)p^2 + (1-p)(p-p^2)\right] \\ &= Np(1-p) \end{aligned}$$

We may also be interested knowing how many bits to run in order to get an estimate of the bit error rate. For example, if we run a simulation and get zero bit errors, we won't have a very good idea of the bit error rate. Let T_1 be the time (number of bits) up to and including the first error.

1. What type of random variable is T_1?
2. What is the pmf of T_1?
3. What is the mean of T_1?

Solution: T_1 is a Geometric random variable with pmf

$$p_{T_1}(t) = (1-p_e)^{t-1} p_e$$

The mean of T_1 is

$$E[T_1] = \frac{1}{p_e}$$

Note the variance of T_1 is $\mathrm{Var}[T_1] = (1-p_e)/p_e^2$, so the standard deviation for very low p_e is almost the same as the expected value.

So, even if we run an experiment until the first bit error, our estimate of p_e will have relatively high variance.

15.5 Expectation

Def'n: *Expected Value (Joint)*
The expected value of a function $g(X_1, X_2)$ of random variables X_1 and X_2 is given by,

1. Discrete:
$$E\left[g(X_1, X_2)\right] = \sum_{X_1 \in S_{X_1}} \sum_{X_2 \in S_{X_2}} g(X_1, X_2) p_{X_1, X_2}(x_1, x_2) \tag{36}$$

2. Continuous:
$$E\left[g(X_1, X_2)\right] = \int_{X_1 \in S_{X_1}} \int_{X_2 \in S_{X_2}} g(X_1, X_2) f_{X_1, X_2}(x_1, x_2) \tag{37}$$

Typical functions $g(X_1, X_2)$ are:

- Mean of X_1 or X_2: $g(X_1, X_2) = X_1$ or $g(X_1, X_2) = X_2$ will result in the means μ_{X_1} and μ_{X_2}.
- Variance (or 2nd central moment) of X_1 or X_2: $g(X_1, X_2) = (X_1 - \mu_{X_1})^2$ or $g(X_1, X_2) = (X_2 - \mu_{X_2})^2$. Often denoted $\sigma^2_{X_1}$ and $\sigma^2_{X_2}$.
- Covariance of X_1 and X_2: $g(X_1, X_2) = (X_1 - \mu_{X_1})(X_2 - \mu_{X_2})$.
- Expected value of the product of X_1 and X_2, also called the 'correlation' of X_1 and X_2: $g(X_1, X_2) = X_1 X_2$.

15.6 Gaussian Random Variables

For a single Gaussian random variable X with mean μ_X and variance σ^2_X, we have the pdf,

$$f_X(x) = \frac{1}{\sqrt{2\pi\sigma^2_X}} e^{-\frac{(x-\mu_X)^2}{2\sigma^2_X}}$$

Consider Y to be Gaussian with mean 0 and variance 1. The distribution of Y is also called *standard normal* in the statistics community without any sense of shame for the redundancy of the two words. Regardless, we define a new symbol for the CDF of standard normal random variable Y: CDF of Y is denoted as $F_Y(y) = P\left[Y \leq y\right] = \Phi(y)$. So, for X, which has non-zero mean and non-unit-variance, we can write its CDF as

$$F_X(x) = P\left[X \leq x\right] = \Phi\left(\frac{x - \mu_X}{\sigma_X}\right)$$

You can prove this by showing that the event $X \leq x$ is the same as the event

$$\frac{X - \mu_X}{\sigma_X} \leq \frac{x - \mu_X}{\sigma_X}$$

Since the left-hand side is a unit-variance, zero mean Gaussian random variable, we can write the probability of this event using the unit-variance, zero mean Gaussian CDF.

15.6.1 Complementary CDF

The probability that a unit-variance, zero mean Gaussian random variable X exceeds some value x is one minus the CDF, that is, $1 - \Phi(x)$. This is so common in digital communications, it is given its own name, $Q(x)$,

$$Q(x) = P\left[X > x\right] = 1 - \Phi\left(x\right)$$

What is $Q(x)$ in integral form?

$$Q(x) = \int_x^\infty \frac{1}{\sqrt{2\pi}} e^{-w^2/2} dw$$

For an Gaussian random variable X with variance σ_X^2,

$$P\left[X > x\right] = Q\left(\frac{x - \mu_X}{\sigma_X}\right) = 1 - \Phi\left(\frac{x - \mu_X}{\sigma_X}\right)$$

15.6.2 Error Function

In math, in some texts, and in Matlab, the $Q(x)$ function is not used. Instead, there is a function called erf(x)

$$\text{erf}(x) \triangleq \frac{2}{\sqrt{\pi}} \int_0^x e^{-t^2} dt$$

Example: Relationship between $Q(\cdot)$ and erf$(\cdot)$
What is the functional relationship between $Q(\cdot)$ and erf$(\cdot)$?
Solution: Substituting $t = u/\sqrt{2}$ (and thus $dt = du/\sqrt{2}$),

$$\begin{aligned} \text{erf}(x) &\triangleq \frac{2}{\sqrt{2\pi}} \int_0^{\sqrt{2}x} e^{-u^2/2} du \\ &= 2 \int_0^{\sqrt{2}x} \frac{1}{\sqrt{2\pi}} e^{-u^2/2} du \\ &= 2\left(\Phi(\sqrt{2}x) - \frac{1}{2}\right) \end{aligned}$$

Equivalently, we can write $\Phi(\cdot)$ in terms of the erf$(\cdot)$ function,

$$\Phi(\sqrt{2}x) = \frac{1}{2}\text{erf}(x) + \frac{1}{2}$$

Finally let $y = \sqrt{2}x$, so that

$$\Phi(y) = \frac{1}{2}\text{erf}\left(\frac{y}{\sqrt{2}}\right) + \frac{1}{2}$$

Or in terms of $Q(\cdot)$,

$$Q(y) = 1 - \Phi(y) = \frac{1}{2} - \frac{1}{2}\text{erf}\left(\frac{y}{\sqrt{2}}\right) \tag{38}$$

You should go to Matlab and create a function Q(y) which implements:

```
function rval = Q(y)
rval = 0.5.*erfc(y./sqrt(2));
```

In Python there is an erfc function in both math and scipy.special packages.

Example: Probability of Error in Binary Example
As in the previous example, we have a model system in which the receiver sees $X_2 = X_1 + N$. Here, $X_1 \in \{0, 1\}$ with equal probabilities and N is independent of X_1 and zero-mean Gaussian with variance $1/4$. The receiver decides as follows:

- If $X_2 \leq 1/3$, then decide that the transmitter sent a '0'.
- If $X_2 > 1/3$, then decide that the transmitter sent a '1'.

1. Given that $X_1 = 1$, what is the probability that the receiver decides that a '0' was sent?
2. Given that $X_1 = 0$, what is the probability that the receiver decides that a '1' was sent?

Solution:

1. Given that $X_1 = 1$, since $X_2 = X_1 + N$, it is clear that X_2 is also a Gaussian random variable with mean 1 and variance $1/4$. Then the probability that the receiver decides '0' is the probability that $X_2 \leq 1/3$,

$$\begin{aligned} P\left[error|X_1 = 1\right] &= P\left[X_2 \leq 1/3\right] \\ &= P\left[\frac{X_2 - 1}{\sqrt{1/4}} \leq \frac{1/3 - 1}{\sqrt{1/4}}\right] \\ &= 1 - Q\left((-2/3)/(1/2)\right) \\ &= 1 - Q(-4/3) \end{aligned}$$

2. Given that $X_1 = 0$, the probability that the receiver decides '1' is the probability that $X_2 > 1/3$,

$$\begin{aligned} P\left[error|X_1 = 0\right] &= P\left[X_2 > 1/3\right] \\ &= P\left[\frac{X_2}{\sqrt{1/4}} > \frac{1/3}{\sqrt{1/4}}\right] \\ &= Q\left((1/3)/(1/2)\right) = Q(2/3) \end{aligned}$$

16 Detection Threshold Activity

To motivate detection theory, think of the following 1-D binary baseband PAM system. The transmitter sends either $s_0(t) = a_0 p(t)$ or $s_1(t) = a_1 p(t)$. The receiver correlates the received signal with $p(t)$ and measures $X = a_0 + W$ or $X = a_1 + W$ where W is additive Gaussian noise. Your receiver decides based on X what symbol was sent.

In the game version of this real-world problem, you will work in a team and compete against other teams. Your team must choose a threshold, where if $X <$ this threshold, your receiver will decide that $s_0(t)$ was sent; and if $X >$ the threshold, it will decide $s_1(t)$ was sent. In Matlab, I will

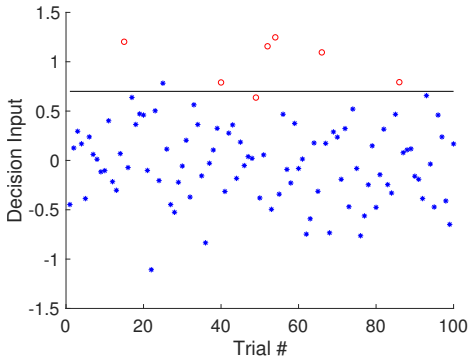


Figure 26: Activity example for threshold $= 0.7$, $a_0 = 0$, $a_1 = 1$, $\sigma_W = 0.4$, $P[H_0] = 0.9$, and 100 trials. There are a total of 2 errors.

generate random symbols and random noise, and calculate based on your threshold, and 100 trials, what your number of errors is. The team with the fewest errors wins.

Lecture 10

Today: (1) 1-D Detection

- Reading for these notes: Kay [7] Sections 3.3 & 3.6.
- Lecture videos for this lecture are at:
  `https://youtube.com/playlist?list=PLQuDEk4rPDOMYanAsrBZnIWOH2gQ4MA_-`

17 Bayesian 1-D Detection

When we say 'optimal detection' in the Bayesian detection framework, we mean that we want the smallest probability of symbol error. The probability of symbol error is denoted

$$P[\text{symbol error}]$$

By error, we mean that a different symbol was detected than the symbol that was sent. At the start of every detection problem, we list the events that could have occurred, *i.e.*, the symbols that could have been sent. We follow all detection and statistics textbooks and label these classes H_i.

Later we will use detection theory to describe why we use a matched filter. For now, we're studying a somewhat simpler problem, assuming that our receiver uses a matched filter, time synchronization block, and downsampling block. Our receivers need to make a decision after the downsampling. based on the voltages X_i measured at this point, for each waveform $\phi_i(t)$. Further for this lecture, we are studying the case when there is only one waveform, e.g., PAM. Thus we call this voltage X for simplicity. The voltage X should be close to one of M possible symbol values $a_0, \ldots, a_{M-1}$.

In summary we describe the decision as a list of *models* that describe what the conditional

distribution of X is given that the ith symbol is sent:

$$\begin{aligned} H_0: &\quad X = a_0 + W \\ H_1: &\quad X = a_1 + W \\ \cdots &\quad \cdots \\ H_{M-1}: &\quad X = a_{M-1} + W \end{aligned}$$

where W is Gaussian additive noise with mean 0 and variance σ^2. This must be a complete listing of events. That is, the events $H_0 \cup H_1 \cup \cdots \cup H_{M-1} = S$, where the $\cup$ means union, and S is the complete event space.

Let's just say for now that there are only two symbols, i.e., $M = 2$. We need to decide from X whether symbol 0 or symbol 1 was sent.

The hypotheses are:

$$\begin{aligned} H_0: &\quad X = a_0 + W \\ H_1: &\quad X = a_1 + W \end{aligned}$$

We use the law of total probability to say that

$$P[\text{error}] = P[\text{error} \cap H_0] + P[\text{error} \cap H_1] \tag{39}$$

Where the cap means 'and'. Then using Bayes' Law,

$$P[\text{error}] = P[\text{error}|H_0]\,P[H_0] + P[\text{error}|H_1]\,P[H_1]$$

17.1 Decision Region

We're making a decision based only on X. Over some set R_0 of values of X, we'll decide that H_0 happened (symbol 0 was sent). Over a different set R_1 of values, we'll decide H_1 occurred (that symbol 1 was sent). *We can't be indecisive*, so

- There is no overlap: $R_0 \cap R_1 = \emptyset$.
- There are no values of x disregarded: $R_0 \cup R_1 = S$.

17.2 Formula for Probability of Error

So the probability of error is

$$P[\text{error}] = P[X \in R_1|H_0]\,P[H_0] + P[X \in R_0|H_1]\,P[H_1] \tag{40}$$

The probability that X is in R_1 is one minus the probability that it is in R_0, since the two are complementary sets.

$$\begin{aligned} P[\text{error}] &= (1 - P[X \in R_0|H_0])P[H_0] + P[X \in R_0|H_1]\,P[H_1] \\ P[\text{error}] &= P[H_0] - P[X \in R_0|H_0]\,P[H_0] + P[X \in R_0|H_1]\,P[H_1] \end{aligned}$$

Now note that probabilities that $X \in R_0$ are integrals over the event (region) R_0.

$$\begin{aligned} P\left[\text{error}\right] &= P\left[H_0\right] - \int_{x \in R_0} f_{X|H_0}(x|H_0) P\left[H_0\right] dx \\ &\quad + \int_{x \in R_0} f_{X|H_1}(x|H_1) P\left[H_1\right] dx \\ &= P\left[H_0\right] + \\ &\quad \int_{x \in R_0} \left\{ f_{X|H_1}(x|H_1) P\left[H_1\right] - f_{X|H_0}(x|H_0) P\left[H_0\right] \right\} dx \end{aligned} \tag{41}$$

We've got a lot of things in the expression in (41), but the only thing we can change is the region R_0. Everything else is determined by the time we get to this point. So the question is, how do you pick R_0 to minimize (41)?

17.3 Selecting R_0 to Minimize Probability of Error

We can see what the integrand looks like. Figure 27(a) shows the conditional probability density functions. Figure 27(b) shows the joint densities (the conditional pdfs multiplied by the bit probabilities $P\left[H_0\right]$ and $P\left[H_1\right]$. Finally, Figure 27(c) shows the full integrand of (41), the difference between the joint densities.

We can pick R_0 however we want - we just say what region of x, and the integral in (41) will integrate over it. The objective is to minimize the probability of error. Which x's should we include in the region? Should we include x which has a positive value of the integrand? Or should we include the parts of x which have a negative value of the integrand?

Solution: Select R_0 to be all x such that the integrand is negative.

Then R_0 is the area in which

$$f_{X|H_0}(x|H_0) P\left[H_0\right] > f_{X|H_1}(x|H_1) P\left[H_1\right]$$

If $P\left[H_0\right] = P\left[H_1\right]$, then this is the region in which X is more probable given H_0 than given H_1.

Rearranging the terms,

$$\frac{f_{X|H_1}(x|H_1)}{f_{X|H_0}(x|H_0)} < \frac{P\left[H_0\right]}{P\left[H_1\right]} \tag{42}$$

The left hand side is called the likelihood ratio. The right hand side is a threshold. Whenever x indicates that the likelihood ratio is less than the threshold, then we'll decide H_0, *i.e.*, that $s_0(t)$ was sent. Otherwise, we'll decide H_1, *i.e.*, that $s_1(t)$ was sent.

Equation (42) is a very general result, applicable no matter what conditional distributions x has.

17.4 Log-Likelihood Ratio

For the Gaussian distribution, the math gets much easier if we take the log of both sides. Why can we do this?

Solution: 1. Both sides are positive, 2. The log() function is strictly increasing.

Now, the *log-likelihood ratio* is

$$\log \frac{f_{X|H_1}(x|H_1)}{f_{X|H_0}(x|H_0)} < \log \frac{P\left[H_0\right]}{P\left[H_1\right]}$$

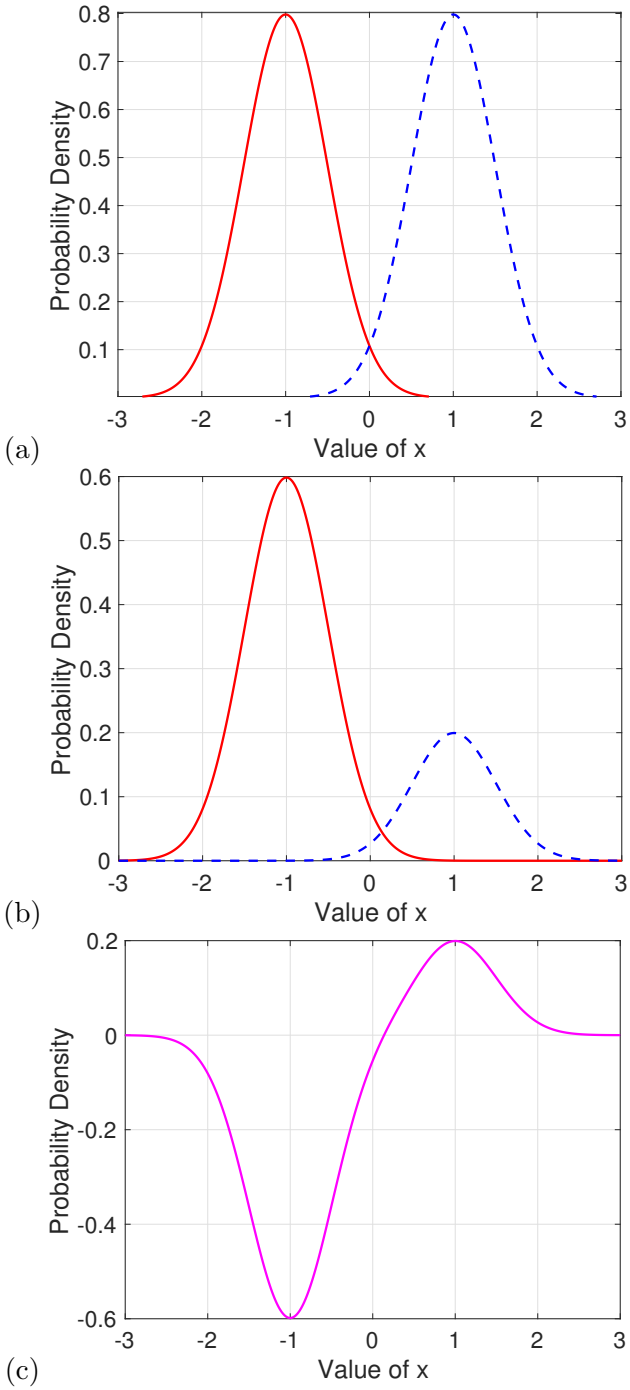


Figure 27: The (a) conditional p.d.f.s (likelihood functions) $f_{X|H_0}(x|H_0)$ and $f_{X|H_1}(x|H_1)$, (b) joint p.d.f.s $f_{X|H_0}(x|H_0)P[H_0]$ and $f_{X|H_1}(x|H_1)P[H_1]$, and (c) difference between the joint p.d.f.s, $f_{X|H_1}(x|H_1)P[H_1] - f_{X|H_0}(x|H_0)P[H_0]$, which is the integrand in (41).

17.5 Case of $a_0 = 0$, $a_1 = 1$ in Gaussian noise

In this example, $W \sim \mathcal{N}(0, \sigma_w^2)$. In addition, assume for a minute that $a_0 = 0$ and $a_1 = 1$. What is:

1. The log of a Gaussian pdf?
2. The log-likelihood ratio?
3. The decision regions for x?

Solution: What is the log of a Gaussian pdf?

$$\begin{aligned} \log f_{X|H_0}(x|H_0) &= \log\left[\frac{1}{\sqrt{2\pi\sigma_w^2}} e^{-\frac{x^2}{2\sigma_w^2}}\right] \\ &= -\frac{1}{2}\log(2\pi\sigma_w^2) - \frac{x^2}{2\sigma_w^2} \end{aligned} \tag{43}$$

The $\log f_{X|H_1}(x|H_1)$ term will be the same but with $(x-1)^2$instead of x^2. Continuing with the log-likelihood ratio,

$$\begin{aligned} \log f_{X|H_1}(x|H_1) - \log f_{X|H_0}(x|H_0) &< \log\frac{P[H_0]}{P[H_1]} \\ \frac{x^2}{2\sigma_w^2} - \frac{(x-1)^2}{2\sigma_w^2} &< \log\frac{P[H_0]}{P[H_1]} \\ x^2 - (x-1)^2 &< 2\sigma_w^2 \log\frac{P[H_0]}{P[H_1]} \\ 2x - 1 &< 2\sigma_w^2 \log\frac{P[H_0]}{P[H_1]} \\ x &< \frac{1}{2} + \sigma_w^2 \log\frac{P[H_0]}{P[H_1]} \end{aligned}$$

In the end result, there is a simple test for x - if it is below the *decision threshold*, decide H_0. If it is above the decision threshold,

$$x > \frac{1}{2} + \sigma_w^2 \log\frac{P[H_0]}{P[H_1]}$$

decide H_1. Rather than writing both inequalities each time, we use the following notation:

$$x \underset{H_0}{\overset{H_1}{\gtrless}} \frac{1}{2} + \sigma_w^2 \log\frac{P[H_0]}{P[H_1]}$$

This completely describes the detector receiver.

For simplicity, we also write $x \underset{H_0}{\overset{H_1}{\gtrless}} \gamma$ where

$$\gamma = \frac{1}{2} + \sigma_w^2 \log\frac{P[H_0]}{P[H_1]} \tag{44}$$

17.6 General Case for Arbitrary Symbols

If, instead of $a_0 = 0$ and $a_1 = 1$, we had arbitrary values for them (the signal space representations of $s_0(t)$ and $s_1(t)$), we could have derived the result in the last section the same way. As long as $a_0 < a_1$, we'd still have $r \underset{H_0}{\overset{H_1}{\gtrless}} \gamma$, but now,

$$\gamma = \frac{a_0 + a_1}{2} + \frac{\sigma_w^2}{a_1 - a_0} \log \frac{P[H_0]}{P[H_1]} \tag{45}$$

17.7 Equi-probable Special Case

If symbols are equally likely, $P[H_1] = P[H_0]$, then $\frac{P[H_1]}{P[H_0]} = 1$ and the logarithm of the fraction is zero. So then

$$x \underset{H_0}{\overset{H_1}{\gtrless}} \frac{a_0 + a_1}{2}$$

The decision above says that if x is closer to a_0, decide that $s_0(t)$ was sent. And if x is closer to a_1, decide that $s_1(t)$ was sent. The boundary is exactly half-way in between the two signal space vectors.

This receiver is also called a maximum likelihood detector, because we only decide which likelihood function is higher (neither is scaled by the prior probabilities $P[H_0]$ or $P[H_1]$.

17.8 Examples

Example: When H_1 becomes less likely, which direction will the optimal threshold move, towards a_0 or towards a_1?

Solution: Towards a_1.

Example: Let $a_0 = -1$, $a_1 = 1$, $\sigma_w^2 = 0.1$, $P[H_1] = 0.4$, and $P[H_0] = 0.6$. What is the decision threshold for x?

Solution: From (45),

$$\gamma = 0 + \frac{0.1}{2} \log \frac{0.6}{0.4} = 0.05 \log 1.5 \approx 0.0203$$

Example: Can the decision threshold be higher than both a_0 and a_1 in this binary, one-dimensional modulation, receiver?

Solution: Yes, it can. You can make $\log \frac{P[H_0]}{P[H_1]}$ arbitrarily high, try it!

Given a_0, a_1, σ_w^2, $P[H_1]$, and $P[H_0]$, you should be able to calculate the optimal decision threshold γ.

Example: In this example, given all of the above constants and the optimal threshold γ, calculate the probability of error from (40).

Starting from

$$P\left[\text{error}\right] = P\left[x \in R_1 | H_0\right] P\left[H_0\right] + P\left[x \in R_0 | H_1\right] P\left[H_1\right]$$

we can use the decision regions in (44) to write

$$P\left[\text{error}\right] = P\left[x > \gamma | H_0\right] P\left[H_0\right] + P\left[x < \gamma | H_1\right] P\left[H_1\right]$$

What is the first probability, given that $r|H_0$ is Gaussian with mean a_0 and variance σ_w^2? What is the second probability, given that $x|H_1$ is Gaussian with mean a_1 and variance σ_w^2? What is then the overall probability of error?

Solution:

$$\begin{aligned} P\left[x > \gamma | H_0\right] &= \mathrm{Q}\left(\frac{\gamma - a_0}{\sigma_w}\right) \\ P\left[x < \gamma | H_1\right] &= 1 - \mathrm{Q}\left(\frac{\gamma - a_1}{\sigma_w}\right) = \mathrm{Q}\left(\frac{a_1 - \gamma}{\sigma_w}\right) \\ P\left[\text{error}\right] &= P\left[H_0\right] \mathrm{Q}\left(\frac{\gamma - a_0}{\sigma_w}\right) + P\left[H_1\right] \mathrm{Q}\left(\frac{a_1 - \gamma}{\sigma_w}\right) \end{aligned}$$

17.9 Review of Binary Detection

We did three things to prove some things about the optimal detector:

- We wrote the formula for the probability of error.
- We found the decision regions which minimized the probability of error.
- We used the log operator to show that for the Gaussian error case the decision regions are separated by a single threshold.
- We showed the formula for that threshold, both in the equi-probable symbol case, and in the general case.

Lecture 11

Today: (1) Random Processes for Noise (2) Gaussian Random Vectors

- Reading for these notes: Rice 4.4-4.5
- Lecture videos for this lecture are at:
  `https://youtube.com/playlist?list=PLQuDEk4rPDOMyCyTzrkNUPNp8yeh_tAj_`

18 Random Processes for Noise

In order to study how noise affects communications receivers, we need to recall some background in the analysis of random processes. A *random process* $X(t)$ is a random function of continuous time t. (A *random sequence* $X(n)$ is a sequence of random variables indexed by time index n.)

A note about my notes! Notation is a bit awkward here because when we talk about random variables, we name the random variable with a capital letter, like X_k or W_k. But we use the same letter in its lower case form, like x_k or w_k to refer to a specific value that the random variable takes (or could take). We also use the lower case letter when we need a 'local' variable for a definite integral. Previously, we may have talked about x_k, but if that is now random because of added noise, I will name it X_k to acknowledge that it is a random variable here.

The physics of thermal noise says that it has power spectral density that is approximately kT_e where $k = 1.3807 \times 10^{23}$ J/K (Joules per Kelvin) is Boltzmann's constant and $T_e = FT_0$ is called the "effective noise temperature", which is proportional to T_0, the temperature of what the antenna is pointing at, and multiplied by a unitless factor F of how much the noise power is amplified within the receiver. Both T_e and T_0 are temperatures in Kelvin. This constant kT_e is constant across the frequencies we use for communications system. At room temperature, the power spectral density of thermal noise goes to zero at frequencies above 10^{13} Hz, that is, 10 Terahertz or 10,000 GHz. This is above frequencies at which most communications signals are sent. The Rice book (4.5.2) has a good analysis of the physics of thermal noise. In short, the physics says that the PSD is flat in the bands we use, and that it is zero mean.

One of the most important and surprising results from random processes is that the autocorrelation function and the power spectral density are Fourier transform pairs, given certain conditions. This helps us figure out how noise affects receivers, as we show next.

18.1 Autocorrelation and Power Spectral Density

Def'n: *Mean Function*
The mean function of the random process $X(t)$ is

$$\mu_X(t) = E[X(t)]$$

Note the mean is taken over all possible realizations of $X(t)$. If you record one signal over all time t, you don't have anything to average to get the mean function $\mu_X(t)$.

Def'n: *Autocorrelation Function*
The autocorrelation function of a random process $X(t)$ is

$$R_X(t, \tau) = E[X(t)X(t - \tau)]$$

The autocorrelation of a random sequence $X(n)$ is

$$R_X(n, k) = E[X(n)X(n - k)]$$

Def'n: *Wide-sense stationary (WSS)*
A random process is wide-sense stationary (WSS) if

1. $\mu_X = \mu_X(t) = E[X(t)]$ is independent of t.
2. $R_X(t, \tau)$ depends only on the time difference τ and not on t. We then denote the autocorrelation function as $R_X(\tau)$.

A random process is wide-sense stationary (WSS) if

1. $\mu_X = \mu_X(n) = E[X(n)]$ is independent of n.
2. $R_X(n, k)$ depends only on k and not on n. We then denote the autocorrelation function as $R_X(k)$.

The power of a signal is given by $R_X(0)$.

Power Spectral Density The power spectral density $S_X(f)$ is a positive real-valued function equal to the density of power in a random process $X(t)$ near the frequency f. It has units of Watts / Hz. The (average) power between two frequencies f_1 and f_2 is the integral of $S_X(f)$ from $f_1 < f < f_2$.

We know from random processes: For a WSS random process $X(t)$ (and for a random sequence $X(n)$) that its power spectral density can be computed as,

$$\begin{aligned} S_X(f) &= \mathfrak{F}\{R_X(\tau)\} \\ S_X(e^{j\Omega}) &= \text{DTFT}\{R_X(k)\} \end{aligned}$$

Example: What is the autocorrelation function for thermal noise?

We know from physics that thermal noise has constant power spectral density, and that the random process is WSS. As Rice says "for historical reasons this constant value is designated $N_0/2$". In this case, what is the autocorrelation function $R_W(\tau)$?
Solution: We're given $S_W(f) = N_0/2$. The relationship is:

$$S_W(f) = \mathfrak{F}\{R_W(\tau)\}$$

So

$$R_W(\tau) = \mathfrak{F}^{-1}\{S_W(f)\} = \mathfrak{F}^{-1}\{N_0/2\}$$

But $N_0/2$ is just a constant. Looking at a Fourier transform table, the solution is

$$R_W(\tau) = \frac{N_0}{2}\delta(\tau).$$

18.2 Uncorrelated Noise

A random sequence $W(n)$ is an *uncorrelated noise* sequence if it is WSS and has autocorrelation function

$$R_W(k) = E[W(n)W(n-k)] = \sigma^2\delta(k)$$

This says that each element of the sequence $W(n)$ is uncorrelated with $W(m), m \neq n$.

A random process $W(t)$ is an *uncorrelated noise* random process if it is WSS and has autocorrelation function

$$R_W(\tau) = E\left[W(t)W(t-\tau)\right] = \sigma^2\delta(\tau)$$

Again, this says that the value of $W(t)$ is uncorrelated with $W(t'), t' \neq t$.

An uncorrelated noise process is generally a good thing, mathematically, as it makes optimal reception easier to implement.

Note that this noise random process is referred to in textbooks as 'white noise' because it has equal parts of every frequency (analogy to light). But black and gray are also constant in the frequency domain. Calling it 'white' is arbitrary, and calling positive things 'white' and negative things 'black' is a bad historical habit in English. Thus I use a more statistically precise name for this noise random process: *uncorrelated noise*.

Note that describing a random process / sequence as "uncorrelated" does NOT say that the distribution of its samples are Gaussian. In order to specify that, we call it *uncorrelated Gaussian noise*. We typically model thermal noise as added to the signal. So we call it additive uncorrelated Gaussian noise (AUGN).

18.3 Noise in Correlation Receiver

Previously, we had said that $r(t)$ was equal to the transmitted signal $s(t)$ plus noise:

$$r(t) = s(t) + w(t)$$

As implied, we consider $w(t)$ to be uncorrelated and Gaussian with zero mean and PSD $S_W(f) = N_0/2$, or equivalently, $R_W(\tau) = N_0/2\delta(\tau)$.

What is the output of the correlation receiver? Recall X_k is defined as $\langle r(t), \phi_k(t)\rangle$, or

$$\begin{aligned} X_k &= \int_{-\infty}^{\infty} r(t)\phi_k(t)dt \\ &= \int_{-\infty}^{\infty} [s_i(t) + w(t)]\phi_k(t)dt \\ &= a_{i,k} + \int_{-\infty}^{\infty} w(t)\phi_k(t)dt \\ &= a_{i,k} + W_k \end{aligned}$$

where we define

$$W_k = \langle w(t), \phi_k\rangle = \int_{-\infty}^{\infty} w(t)\phi_k(t)dt.$$

What can we know about W_k? What are the mean and covariance of $\{W_k\}$? For practice, prove that: 1) W_k are all zero mean; and 2) the correlation of W_k and W_m is zero unless $k = m$, and in that case, is equal to $N_0/2$.

Solution: First, W_k is zero mean:

$$E\left[W_k\right] = \int_{-\infty}^{\infty} E\left[w(t)\right]\phi_k(t)dt = 0$$

Next we can show that any two W_k and W_m for $k \neq m$ have zero correlation by calculating the autocorrelation $R_w(m,k)$.

$$\begin{aligned} R_W(m,k) &= E\left[W_k W_m\right] \\ &= \int_{t=-\infty}^{\infty}\int_{\tau=-\infty}^{\infty} E\left[w(t)w(\tau)\right]\phi_k(t)\phi_m(\tau)d\tau dt \\ &= \int_{t=-\infty}^{\infty}\int_{\tau=-\infty}^{\infty} \frac{N_0}{2}\delta(t-\tau)\phi_k(t)\phi_m(\tau)d\tau dt \\ &= \frac{N_0}{2}\int_{t=-\infty}^{\infty}\phi_k(t)\phi_m(t)dt \\ &= \frac{N_0}{2}\delta(k-m) = \begin{cases} \frac{N_0}{2}, & m=k \\ 0, & o.w. \end{cases} \end{aligned}$$

Is W_k Gaussian? Yes – an integral is a linear operation, and any linear function of a Gaussian random process is also Gaussian.

Are $\{W_k\}$ independent? Yes – for Gaussian random variables, a covariance of zero implies independent.

Since the noise components are independent, then X_k (the sum of $a_{i,k}$ and W_k) are also Gaussian and independent. Why? Because $a_{i,k}$ is a deterministic constant, and thus X_k are Gaussian with mean $a_{i,k}$. But that change in mean doesn't change the autocovariance function.

What is the pdf of $\mathbf{X} = [X_0, \ldots, X_K]^T$?

Solution:

$$f_{X_k}(x_k) = \frac{1}{\sqrt{2\pi(N_0/2)}} e^{-\frac{(x_k - a_{i,k})^2}{2(N_0/2)}}$$

And, since the $\{X_k\}$ are independent, the joint pdf of all of them is the product of the marginal pdfs:

$$\begin{aligned} f_{\mathbf{x}}(\mathbf{x}) &= \prod_{k=1}^{K} f_{X_k}(x_k) \\ &= \prod_{k=1}^{K} \frac{1}{\sqrt{2\pi(N_0/2)}} e^{-\frac{(x_k - a_{i,k})^2}{2(N_0/2)}} \\ &= \frac{1}{[2\pi(N_0/2)]^{K/2}} e^{-\frac{\sum_{k=1}^{K}(x_k - a_{i,k})^2}{2(N_0/2)}} \end{aligned}$$

18.4 Gaussian Random Vectors

Def'n: *Multivariate Gaussian R.V.*

An n-length R.V. $\mathbf{X}$ is multivariate Gaussian with mean $\mu_{\mathbf{X}}$, and covariance matrix $C_{\mathbf{X}}$ if it has the pdf,

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

where det() is the determinant of the covariance matrix, and $C_{\mathbf{X}}^{-1}$ is the inverse of the covariance matrix.

Any linear combination of jointly Gaussian random variables is another jointly Gaussian random variable. For example, if we have a matrix A and we let a new random vector $\mathbf{Y} = A\mathbf{X}$, then $\mathbf{Y}$ is also a Gaussian random vector with mean $A\mu_{\mathbf{X}}$ and covariance matrix $AC_{\mathbf{X}}A^T$.

If the elements of $\mathbf{X}$ were independent random variables, the pdf would be the product of the individual pdfs (as with any random vector) and in this case the pdf would be:

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

Section 4.3 spends some time with 2-D Gaussian random vectors, which is the dimension with which we spend most of our time in this class.

Lecture 12

Today: K-dim Bayesian Detection Theory with M Symbols

- Reading for these notes: Rice [11] 6.1, 6.2.
- Lecture videos for this lecture are at:
  `https://youtube.com/playlist?list=PLQuDEk4rPDOM_9ZK_wij6oGA1v5Mj_3YU`

19 M-ary Detection Theory with K-Dimensional Signals

We are going to start to talk about QAM, PSK, and FSK, modulations with $K \geq 2$ basis functions. We are also simultaneously going to extend our discussion from $M = 2$ symbols to $M \geq 2$ symbols in the constellation.

Our setup:

- Transmit: one of M possible symbols, $\mathbf{s}_0, \ldots, \mathbf{s}_{M-1}$. Recall these $\mathbf{s}_i$ vectors are length K, we write its elements as:
$$\mathbf{s}_i = \begin{bmatrix} a_{i,0} \\ \vdots \\ a_{i,K-1} \end{bmatrix}.$$
- Receive: the symbol vector plus noise, after the matched filter and downsampler:
$$\begin{aligned} H_0: &\quad \mathbf{X} = \mathbf{s}_0 + \mathbf{W} \\ \cdots &\quad \cdots \\ H_{M-1}: &\quad \mathbf{X} = \mathbf{s}_{M-1} + \mathbf{W} \end{aligned}$$
- Assume: $\mathbf{W}$ is multivariate Gaussian, each of K components W_k are independent with zero mean and variance $\sigma_W^2 = N_0/2$.
- Assume: Symbols are equally likely.
- Question: What are the optimal decision regions?

19.1 Optimal Detection with Multiple Hypotheses

The measurement $\mathbf{X}$ has a different joint probability density under each hypothesis:

$$\begin{aligned} H_0: &\quad f_{\mathbf{X}|H_0}(\mathbf{x}|H_0)P[H_0] \\ H_1: &\quad f_{\mathbf{X}|H_1}(\mathbf{x}|H_1)P[H_1] \\ \dots &\quad \dots \\ H_{M-1}: &\quad f_{\mathbf{X}|H_{M-1}}(\mathbf{x}|H_{M-1})P[H_{M-1}] \end{aligned}$$

Given that a particular $\mathbf{x}$ is measured, what is our method for deciding which hypothesis (about which symbol was sent) is true, in a way that minimizes the error? We can look to our previous lecture on binary decision. We minimized error by finding which joint probability of $\mathbf{x}$ and H_i for $i \in \{0,1\}$ was highest. A similar derivation to that one would show that the probability of error is minimized by finding the i which joint probability of $\mathbf{x}$ and H_i for $i \in \{0,\dots,M-1\}$ is highest. That is, we decide symbol i was sent if

$$f_{\mathbf{X}|H_i}(\mathbf{x}|H_i)P[H_i] > f_{\mathbf{X}|H_j}(\mathbf{x}|H_j)P[H_j] \text{ for all } j \neq i.$$

For this class, we'll usually consider the case of equally probable symbols. While symbols are sometimes not equally probable for $M=2$ binary detection, it is very rare in higher M communication systems because it is easy for communications systems designers to encode the data so that each symbol is equally likely to be transmitted. If $P[H_0] = \cdots = P[H_{M-1}]$ then we only need to find the i that makes the likelihood $f_{\mathbf{X}|H_i}(\mathbf{x}|H_i)$ maximum, that is, maximum likelihood detection:

$$\text{Symbol Decision} = \arg\max_i f_{\mathbf{X}|H_i}(\mathbf{x}|H_i) \tag{46}$$

Here we have multivariate Gaussian measurement. As we derived earlier, the elements of vector $\mathbf{X}$ are uncorrelated and each have the same variance σ_W^2. This means that

$$f_{\mathbf{X}|H_i}(\mathbf{x}|H_i) = \frac{1}{(2\pi\sigma_W^2)^{K/2}} \exp\left[-\sum_{i=0}^{K-1} \frac{(x_k - a_{i,k})^2}{2\sigma_W^2}\right].$$

Since $\sum_i (x_k - a_{i,k})^2$ can be written as the squared Euclidean distance between two vectors $\mathbf{x}$ and $\mathbf{s}_i$, we can simplify by writing:

$$f_{\mathbf{X}|H_i}(\mathbf{x}|H_i) = \frac{1}{(2\pi\sigma_W^2)^{K/2}} \exp\left[-\frac{\|\mathbf{x}-\mathbf{s}_i\|^2}{2\sigma_W^2}\right].$$

When we want to solve (46) we can simplify, as we did in the binary decision case: 1) using the natural log to remove the exp; 2) removing any additive terms and multiplying out any terms that are not a function of i, and 3) using the square root. Note that the log and the $\sqrt{\cdot}$ are monotonically increasing functions and thus don't change the output of the argmin.

$$\begin{aligned} \hat{i} &= \operatorname*{argmax}_i \left\{ \log \frac{1}{(2\pi\sigma_W^2)^{K/2}} - \frac{\|\mathbf{x}-\mathbf{s}_i\|^2}{2\sigma_W^2} \right\} \\ \hat{i} &= \operatorname*{argmax}_i -\frac{\|\mathbf{x}-\mathbf{s}_i\|^2}{2\sigma_W^2} \\ \hat{i} &= \operatorname*{argmin}_i \|\mathbf{x}-\mathbf{s}_i\|^2 \\ \hat{i} &= \operatorname*{argmin}_i \|\mathbf{x}-\mathbf{s}_i\| \end{aligned} \tag{47}$$

Again: The short story is that we just find the $\mathbf{s}_i$ in the signal space diagram which is closest to $\mathbf{x}$.

19.2 Pairwise Comparisons

When is $\mathbf{x}$ closer to $\mathbf{s}_i$ than to $\mathbf{s}_j$ for some other signal space point j? Solution: **The two decision regions are separated by a straight line** (Note: replace "line" with plane in 3-D, or subspace in K-D). To find this line:

1. Draw a line segment connecting $\mathbf{s}_i$ and $\mathbf{s}_j$.
2. Draw a point in the middle of that line segment.
3. Draw the perpendicular bisector of the line segment through that point.

Example: Derive a formula for the dividing line (dividing plane when $K > 2$) between $\mathbf{s}_i$ and $\mathbf{s}_j$ for $j \neq i$ when symbols are equally likely.

Solution: Try to find the locus of points $\mathbf{x}$ which satisfy the equality of distances between the two symbol points:

$$\|\mathbf{x} - \mathbf{s}_i\|^2 = \|\mathbf{x} - \mathbf{s}_j\|^2$$

You can do this by using the inner product to represent the magnitude squared operator:

$$(\mathbf{x} - \mathbf{s}_i)^T(\mathbf{x} - \mathbf{s}_i) = (\mathbf{x} - \mathbf{s}_j)^T(\mathbf{x} - \mathbf{s}_j)$$

Then use FOIL (multiply out), cancel, and reorganize to find a linear equation in terms of $\mathbf{x}$.

$$\begin{aligned}
\mathbf{x}^T\mathbf{x} - \mathbf{x}^T\mathbf{s}_i - \mathbf{s}_i^T\mathbf{x} + \mathbf{s}_i^T\mathbf{s}_i &= \mathbf{x}^T\mathbf{x} - \mathbf{x}^T\mathbf{s}_j - \mathbf{s}_j^T\mathbf{x} + \mathbf{s}_j^T\mathbf{s}_j \\
-2\mathbf{s}_i^T\mathbf{x} + \|\mathbf{s}_i\|^2 &= -2\mathbf{s}_j^T\mathbf{x} + \|\mathbf{s}_j\|^2 \\
2(\mathbf{s}_j - \mathbf{s}_i)^T\mathbf{x} &= \|\mathbf{s}_j\|^2 - \|\mathbf{s}_i\|^2 \\
(\mathbf{s}_j - \mathbf{s}_i)^T\mathbf{x} &= \frac{\|\mathbf{s}_j\|^2 - \|\mathbf{s}_i\|^2}{2}
\end{aligned}$$

That is, the inner product of $(\mathbf{s}_j - \mathbf{s}_i)^T$ and $\mathbf{x}$ should be equal to the constant, $\frac{\|\mathbf{s}_j\|^2 - \|\mathbf{s}_i\|^2}{2}$. To see an example of this, consider picking a case when the origin is halfway between the two symbols. This makes $\|\mathbf{s}_j\| = \|\mathbf{s}_i\|$. In this case, the condition says $(\mathbf{s}_j - \mathbf{s}_i)^T\mathbf{x} = 0$, in other words, the two vectors $(\mathbf{s}_j - \mathbf{s}_i)$ and $\mathbf{x}$ are perpendicular. That is, $\mathbf{x}$ must be at the origin, or on any line from the origin that is perpendicular to the line between the two symbols, $\mathbf{s}_j - \mathbf{s}_i$.

Also note that the coordinate system we use to draw the set of $\mathbf{x}$ that meets the criterion is arbitrary, as long as we do it in the same coordinate system as $\mathbf{s}_i$ and $\mathbf{s}_j$, that is, we're consistent.

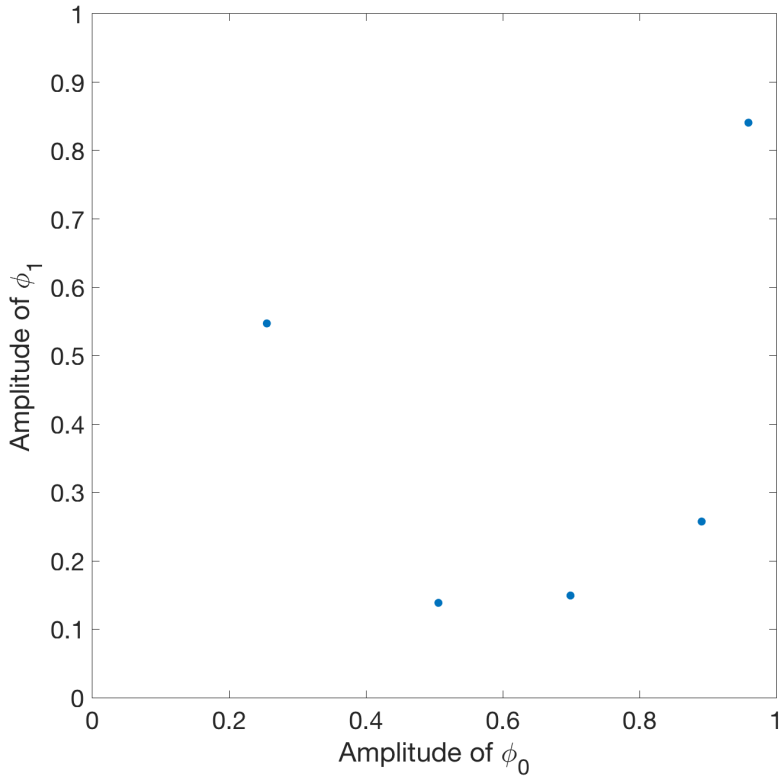
19.3 Decision Regions

Each pairwise comparison results in a linear division of space (a half-space). The combined decision region, the intersection of all of these of half-spaces, R_i, is the space which in which all conditions are satisfied. This intersection space is where $\mathbf{s}_i$ is the closest symbol out of all of the M symbols.

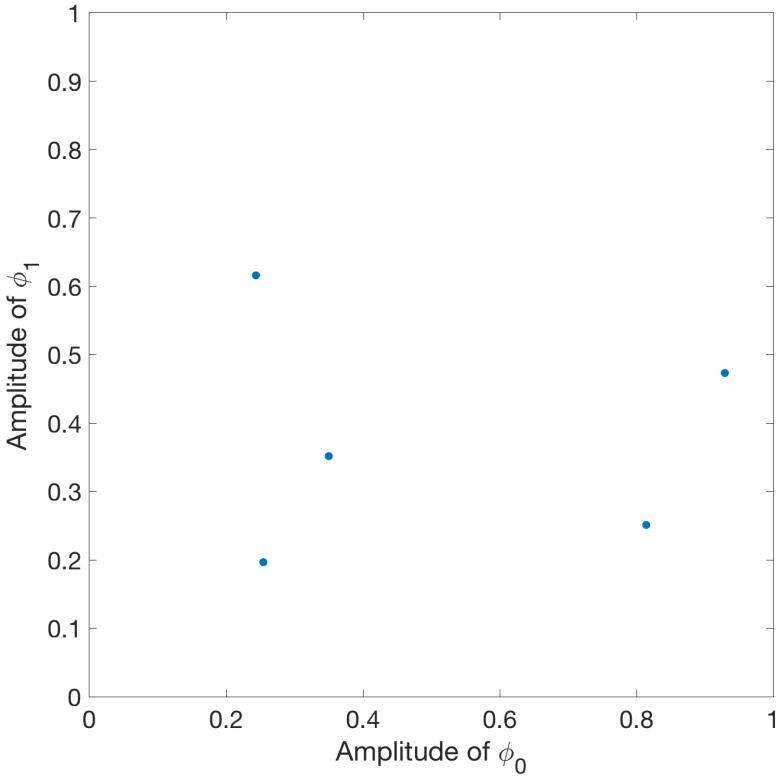
A diagram of all of the decision regions is called a Voronoi diagram.

Example: Optimal Decision Regions

We will do an activity in class to draw the decision regions for Bayesian detection for a 2D constellation diagram. See Figure 28 for two randomly generated constellations with $M = 5$.



(a)



(b)

Figure 28: Example signal space diagrams. Draw the optimal decision regions.

19.4 Symbol Distances

We mentioned, when talking about signal space diagrams, a distance between vectors,

$$d_{i,j} = \|\mathbf{s}_i - \mathbf{s}_j\| = \left[\sum_{k=1}^{M}(a_{i,k} - a_{j,k})^2\right]^{1/2}$$

In general we will start to use these distances quite often in the analysis of a modulation method.

Lecture 13

Today: (1) dB Notation; (2) Probability of Error in M-ary PAM

- Reading for these notes: Rice [11] Section 6.1
- Lecture videos for this lecture are at:
  `https://youtube.com/playlist?list=PLQuDEk4rPDONbbuyiFMTpmBR_yUNtFDpT`

20 Decibel Notation

We often use a decibel (dB) scale for power. If P_{lin} is the power in Watts, then the power in dBW is

$$P\,[\text{dBW}] = 10\log_{10}\frac{P_{lin}}{1\text{ W}}.$$

We convert from dB to linear by inverting the above formula,

$$P_{lin} = (1\text{ W})10^{P[\text{dBW}]/10}.$$

Decibels are more general - they can apply to other unitless quantities as well, such as a gain (loss) $L(f)$ through a filter $H(f)$,

$$L(f)\,[\text{dB}] = 10\log_{10}|H(f)|^2 \tag{48}$$

Note: Why is the capital B used? The 'Bel' refers to a Alexander Graham 'Bell' so it is capitalized. The Bel is defined as the $\log_{10}$ of the ratio, so following the SI convention, the decibel is ten times that. The standard is to use the decibel $10\log_{10}(\cdot)$ which is then abbreviated as dB, just like milliwatt is abbreviated mW in the SI system.

Note that (48) could also be written as:

$$L(f)\,[\text{dB}] = 20\log_{10}|H(f)| \tag{49}$$

Be careful with your use of 10 vs. 20 in the dB formula.

- Only use 20 as the multiplier if you are simultaneously converting linear to dB **and** from voltage to power; *i.e.*, taking the $\log_{10}$ of a voltage gain and expecting the result to be a dB power gain.

Our standard in communications systems design is to consider power gains and losses, not voltage gains and losses. So if we say, for example, the channel has a loss of 20 dB, this refers to a loss in power. In particular, the output of the channel has 100 times less power than the input to the channel.

Remember these three dB numbers:

- 3 dB: This means the number is double in linear terms.
- 10 dB: This means the number is ten times in linear terms.
- 1 dB: This means the number is a little over 25% more (multiply by 5/4) in linear terms.

With these three numbers, you can quickly convert losses or gains between linear and dB units without a calculator. Just convert any dB number into a sum of multiples of 10, 3, and 1.

Example: Convert dB to linear values:

1. 30 dBW
2. 33 dBm
3. -20 dB
4. 4 dB

Solution:

1. $(1\text{ W})10^{30[\text{dBW}]/10} = 10^3\text{W} = 1000\text{ W}$.
2. 33 dBm = 30 dBm + 3 dB = 1000 mW $\times 2 = 2$ W.
3. -20 dB $= 10^{-20[\text{dBW}]/10} = 10^{-2} = 0.01$.
4. 4 dB = 3 dB + 1 dB $\approx 2(1.25) = 2.5$.

Example: Convert linear values to dB:

1. 0.2 W
2. 40 mW

Solution:

1. $0.2\text{ W} = (0.1\text{W})(2) = (\,[\text{dBW}] - 10) + 3\,[\text{dB}] = -7\text{ dBW}$
2. $40\text{ mW} = (10\text{mW})(2)(2) = 10\,[\text{dBm}] + 3\,[\text{dB}] + 3\,[\text{dB}] = 16\,[\text{dBW}]$.

Example: Convert power relationships to dB:
Convert the expression to one which involves only dB terms.

1. $P_{y,lin} = 100P_{x,lin}$

2. $P_{o,lin} = G_{connector,lin} L_{cable,lin}^{-d}$, where $P_{o,lin}$ is the received power in a fiber-optic link, where d is the cable length (typically in units of km), $G_{connector,lin}$ is the gain in any connectors, and $L_{cable,lin}$ is a loss in a 1 km cable.

3. $P_{r,lin} = P_{t,lin} \frac{G_{t,lin} G_{t,lin} \lambda^2}{(4\pi d)^2}$, where λ is the wavelength (m), d is the path length (m), and $G_{t,lin}$ and $G_{t,lin}$ are the linear gains in the antennas, $P_{t,lin}$ is the transmit power (W) and $P_{r,lin}$ is the received power (W). This is the Friis free space path loss formula.

These last two are what we will need in Section 6.4, when we discuss link budgets. The main idea is that we have a limited amount of power which will be available at the receiver.

21 M-ary PAM Probability of Error

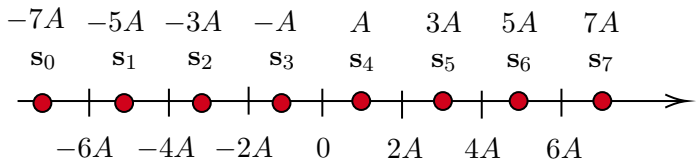


Figure 29: Signal space diagrams for 8-PAM, with optimal detection thresholds.

Consider M-PAM. Our goal in this section is to find a formula for the probability that our optimal receiver makes a symbol error when receiving.

21.1 Symbol Error

The probability that we don't get the symbol correct is the probability that x does not fall within the range between the thresholds in which it belongs. As we have argued before, for M-ary modulation, we can generally assume that each symbol is equally likely, and thus the threshold is halfway between two neighboring symbols. Here, each $\mathbf{s}_i = -7A + i(2A)$. Also the noise $\sigma_W^2 = N_0/2$, as described in our lecture on thermal noise.

Let's consider first that symbol $i = 1$ is transmitted. Recall $\mathbf{s}_1 = -5A$. The probability of error is the sum of the probability of deciding H_0, plus the probability of deciding H_i for $i = 2, \ldots, 7$. That is, that the x value is below $-6A$, or above $-4A$. Thus

$$\begin{aligned} P(\text{symbol error}|H_1) &= \mathrm{Q}\left(\frac{-5A-(-6A)}{\sqrt{N_0/2}}\right) + \mathrm{Q}\left(\frac{-4A-(-5A)}{\sqrt{N_0/2}}\right) \\ &= 2\mathrm{Q}\left(\frac{A}{\sqrt{N_0/2}}\right). \end{aligned} \tag{50}$$

Assuming neighboring symbols a_i are spaced by $2A$, the decision threshold is always A away from the symbol values. For the symbols i in the 'middle' (with two neighbors),

$$P(\text{symbol error}|H_i) = 2\mathrm{Q}\left(\frac{A}{\sqrt{N_0/2}}\right).$$

Note that there are $M-2$ of these symbols. For the symbols i on the 'sides',

$$P(\text{symbol error}|H_i) = \mathrm{Q}\left(\frac{A}{\sqrt{N_0/2}}\right).$$

Note that there are two of these symbols ($\mathbf{s}_0$ and $\mathbf{s}_{M-1}$). Since each symbol H_i is equally likely, the overall probability of symbol error is the average of the conditional probabilities of symbol error. That is:

$$\begin{aligned} P(\text{symbol error}) &= \frac{1}{M}\sum_{i=0}^{M-1} P(\text{symbol error}|H_i) \\ P(\text{symbol error}) &= \frac{1}{M}\left\{(M-2)2\mathrm{Q}\left(\frac{A}{\sqrt{N_0/2}}\right) + (2)\mathrm{Q}\left(\frac{A}{\sqrt{N_0/2}}\right)\right\}. \end{aligned}$$

Simplifying,

$$P(\text{symbol error}) = \frac{2(M-1)}{M}\mathrm{Q}\left(\frac{A}{\sqrt{N_0/2}}\right)$$

Symbol Error Rate and Average Bit Energy:

How does this relate to the average bit energy $\mathcal{E}_b$? We calculated in an earlier lecture the average symbol error for M-PAM as

$$\mathcal{E}_s = \frac{(M^2-1)}{3}A^2$$

But there are $\log_2 M$ bits per symbol. We're going to want to express energy per bit instead of average energy per symbol. Thus we use $\mathcal{E}_b$ to denote the average energy per bit. Thus

$$\mathcal{E}_b = \frac{1}{\log_2 M}\frac{(M^2-1)}{3}A^2, \tag{51}$$

which means that

$$A = \sqrt{\frac{3\log_2 M}{M^2-1}\mathcal{E}_b}$$

So

$$P(\text{symbol error}) = \frac{2(M-1)}{M}\mathrm{Q}\left(\sqrt{\frac{6\log_2 M}{M^2-1}\frac{\mathcal{E}_b}{N_0}}\right) \tag{52}$$

Equation (52) is plotted in Figure 30.

21.2 Bit Errors and Gray Encoding

For binary PAM, there are only two symbols, one will be assigned binary 0 and the other binary 1. When you make one symbol error (decide H_0 or H_1 in error) then it will cause one bit error.

For $M > 2$ PAM, bits and symbols are not synonymous. Instead, we carefully assign bit codes to symbols $0 \ldots M-1$ so that the most common receiver errors cause only one bit to be in error.

Example: Bit coding of $M = 4$ symbols

While the two options shown in Figure 31 both assign 2-bits to each symbol in unique ways, one will lead to a higher bit error rate than the other. Is one is better or worse?

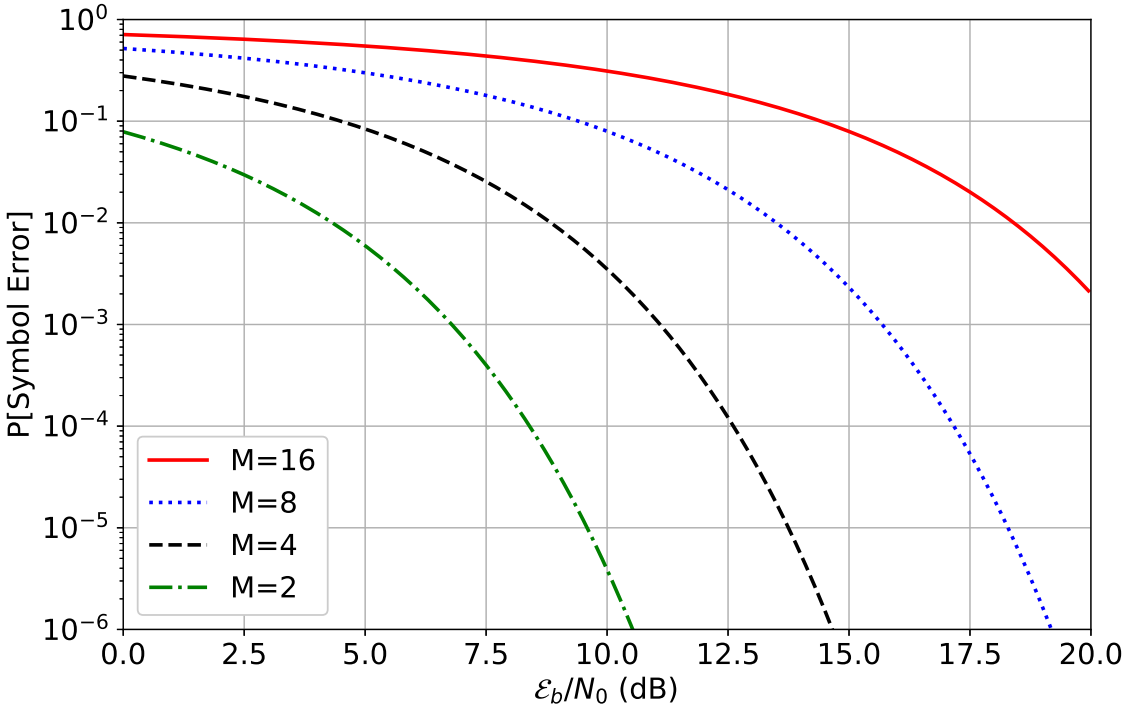


Figure 30: Probability of Symbol Error in M-ary PAM.

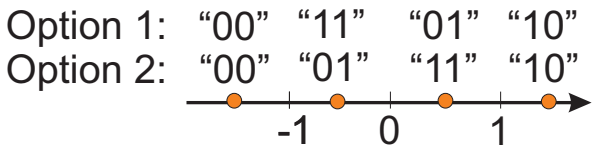


Figure 31: Two options for assigning bits to symbols in 4-PAM.

The key is to recall the model for noise. It will not shift a signal uniformly across symbols. It will tend to leave the received signal x close to the original signal a_i. The neighbors of a_i will be more likely than distant signal space vectors.

Thus Gray encoding will only change one bit across boundaries, as in Option 2 in Figure 31.

Example: Bit coding of $M = 8$ symbols
Assign three bits to each symbol such that any two nearest neighbors are different in only one bit (Gray encoding).
Solution: Here is one solution.

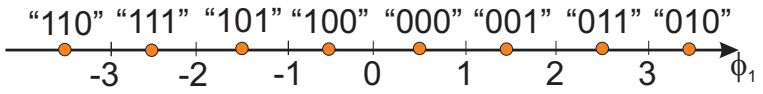


Figure 32: Gray encoding for 8-PAM.

21.2.1 Bit Error Probabilities

How many bit errors are caused by a symbol error in M-ary PAM?

- One. If Gray encoding is used, the errors will tend to be just one bit flipped, more than multiple bits flipped. At least at high $\mathcal{E}_b/N_0$,

$$P(\text{error}) \approx \frac{1}{\log_2 M} P(\text{symbol error}) \tag{53}$$

- Maybe more, up to $\log_2 M$ in the worst case. Then, we need to study further the probability that x will jump more than one decision region.

Generally, we study digital communications systems with high reliability, and high $\mathcal{E}_b/N_0$. Thus when we calculate bit error rate for M-ary PAM, we use (53). We will show that this approximation is very good, in almost all cases for M-PAM and most common QAM/PSK modulations. We will also discuss some particular examples in multi-dimensional signalling when this is not a great approximation.

Thus for M-PAM:

$$P(\text{bit error}) \approx \frac{2(M-1)}{M \log_2 M} \mathrm{Q}\left(\sqrt{\frac{6 \log_2 M}{M^2 - 1} \frac{\mathcal{E}_b}{N_0}}\right) \tag{54}$$

22 Square QAM Probability of Error

Consider now QAM modulation, in general, which is $K = 2$. Recall that our symbol decision is $\arg\max_i \|\mathbf{x} - \mathbf{s}_i\|$, and that this creates decision regions using a Voronoi diagram. This can, in general, be quite complicated because we'd have a probability of error that is an integral of a 2-D Gaussian pdf in the area outside of some polygon containing $\mathbf{s}_i$. Two dimensional integrals of a Gaussian pdf aren't something for which we can find an analytical solution. While we can solve numerically for such integrals, and people do, we tend to find easier approximations whenever possible.

Square QAM is an example where we can write the analytical probability of symbol error formula without any additional approximations. To do this, consider square M-QAM as two

orthogonal $\sqrt{M}$-PAM systems. In this case, the bits of each symbol are separated into first the bits corresponding to the in-phase component of the modulation, and next the bits corresponding to the quadrature component. We can see that using gray coding in each component, we can arrange it so that the first bits are completely dependent on the in-phase component. Because these $\log_2 \sqrt{M}$ bits are independent of the quadrature, the probability of bit error for these bits is the same as the the probability of bit error for $\sqrt{M}$-PAM. Similarly, the next $\log_2 \sqrt{M}$ bits are independent of the in-phase component, and thus the probability of bit error for these bits is the same as the the probability of bit error for $\sqrt{M}$-PAM. Overall, the probability of bit error for square M-QAM is equal to the probability of bit error for $\sqrt{M}$-PAM:

$$P(\text{bit error}) \approx \frac{4(\sqrt{M}-1)}{\sqrt{M}\log_2 M} \mathrm{Q}\left(\sqrt{\frac{3\log_2 M}{M-1}\frac{\mathcal{E}_b}{N_0}}\right). \tag{55}$$

The probability of symbol error is the probability of a symbol error *either* in the in-phase or quadrature components. In other words, it is 1− the probability that we don't make an error in the in-phase, and we don't make an error in the quadrature component. The two error events are independent so the error probabilities multiply each other. Thus it is $1-(1-P\left[\text{Symbol error in } \sqrt{M}\text{-PAM}\right])^2$, or

$$P(\text{symbol error}) = 1 - \left[1 - \frac{2(\sqrt{M}-1)}{\sqrt{M}} \mathrm{Q}\left(\sqrt{\frac{3\log_2 M}{M-1}\frac{\mathcal{E}_b}{N_0}}\right)\right]^2.$$

Lecture 14

Today: QAM/PSK Probability of Error: (1) Union Bound, (2) Nearest neighbor approximation

- Reading for these notes: Rice [11] Section 6.2.
- Lecture videos for this lecture are at:
  `https://youtube.com/playlist?list=PLQuDEk4rPDONt_BY-zoatsuigr38YgjKl`

23 QAM/PSK Probability of Error

23.1 Options for Probability of Error Expressions

Here are some choices you have to compute the probability of symbol error for particular modulations. In order of preference:

1. *Exact formula.* In a few cases, there is an exact expression for P [symbol error] in an AWGN environment, e.g., for square QAM, for PAM, for PSK.
2. *Union bound.* This is a provable upper bound on the probability of error. It is not an approximation, in that sense. It can be used for "worst case" analysis which is often very useful for the engineering design of systems.
3. *Nearest Neighbor Approximation.* This is a way to get a solution that is analytically easier to handle. Typically this approximation is good at high $\frac{\mathcal{E}_b}{N_0}$.

23.2 Exact Error Analysis

As we discussed last time, the exact probability of error formulas in K-dimensional modulations with arbitrary constellation diagrams can be very difficult to compute. This is because our decisions regions are more complex than one threshold test. They require an integration of a K-D Gaussian pdf across an area. We needed a Q function to get a tail probability for a 1-D Gaussian pdf. To find the probability of a subspace of K-D Gaussian pdf, we need not just a tail probability... but a K-D integral under some part of the K-dimensional pdf.

For example, consider M-ary PSK. Essentially, we must find calculate the probability of symbol

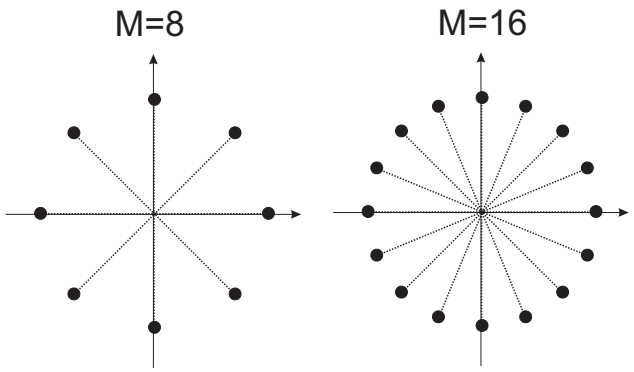


Figure 33: Signal space diagram for M-ary PSK for $M = 8$ and $M = 16$.

error as 1 minus the area in the sector within $\pm\frac{\pi}{M}$ of the correct angle ϕ_i. This is,

$$P(\text{symbol error}) = 1 - \int_{r\in R_i} \frac{1}{2\pi\sigma^2} e^{-\frac{\|\mathbf{r}-\boldsymbol{\alpha}_i\|^2}{2\sigma^2}} \tag{56}$$

This integral is a double integral, and we don't generally have any exact expression to use to express the result in general.

23.3 Probability of Error in QPSK

In QPSK, the probability of error is analytically tractable. Consider the QPSK constellation diagram, when Gray encoding is used. You have already calculated the decision regions for each symbol; now consider the decision region for the first bit.

The decision is made using *only* one dimension, of the received signal vector $\mathbf{x}$, specifically x_1. Similarly, the second bit decision is made using only x_2. Also, the noise contribution to each element is independent. The decisions are decoupled – x_2 has no impact on the decision about bit one, and x_1 has no impact on the decision on bit two. Since we know the bit error probability for each bit decision (it is the same as bipolar PAM) we can see that the bit error probability is also

$$P\left[\text{error}\right] = \text{Q}\left(\sqrt{\frac{2\mathcal{E}_b}{N_0}}\right) \tag{57}$$

This is an extraordinary result – the bit rate will double in QPSK, but in theory, the bit error rate does not increase. As we will show later, the bandwidth of QPSK is identical to that of BPSK.

23.4 Neighbors

For a particular symbol i in a constellation, we define the concept of *neighbors*. These are the symbols j which are necessary to draw the (Voronoi) decision region for symbol i. That is, the perpendicular bisector of the line between i and j is one of the boundaries of the decision region for i. We denote this set as $N(i)$.

The *nearest neighbors* is the set of neighbors which are at the minimum distance $d_{i,j}$ for all j in $N(i)$. There may be several neighbors j with exactly the same distance $d_{i,j}$, which is the reason the nearest neighbors is a set, but of course there is a minimum of one nearest neighbor of i. We denote this set as $NN(i)$.

Example: Listing Neighbor Symbols

Consider the constellation in Figure 34.

1. What are the neighbors of 1, $N(1)$?
2. What are the nearest neighbors of 1, $NN(1)$?

Solution: For (1), $N(1) = \{4, 5, 6, 9, 10\}$. Note that 3 could be a neighbor, since we've only drawn the Voronoi boundaries within the $[0, 1]^2$, and the line for $(10, 3)$ may intersect with the line from $(1, 10)$ far to the upper left of this figure. For (2) just by looking at the plot, $NN(1) = \{4\}$.

Generally, since we don't put symbols in random locations as was done for this plot, it is easier to define neighbors and nearest neighbors.

23.5 Probability of $j|i$ Error

What is the probability of deciding H_j when H_i is true? When the space is divided into two, that is, there are no other symbols, this probability of being closer to $\mathbf{s}_j$ than to $\mathbf{s}_i$ can be computed with a 1-D integral of a Gaussian pdf. Recall from probability that any linear combination of a multi-variate Gaussian vector is also Gaussian. If we rotate the axes so that one is parallel to the line between $\mathbf{s}_i$ and $\mathbf{s}_j$ (which we call the *new axis*, then this rotated vector is also multivariate Gaussian, and the distance along this new axis is Gaussian. Further because the standard deviation is identical in every dimension, the standard deviation of the value on this new axis is also the same, $\sigma_W = \sqrt{N_0/2}$. Let y be the value along the new axis.

Consider the probability that, given i was sent, that y is closer to j than to i and thus we decide H_j. Denote this event $E_{j|i}$. Since $\mathbf{s}_i$ and $\mathbf{s}_j$ are $d_{i,j} = \|\mathbf{s}_i - \mathbf{s}_j\|$ apart, the threshold is halfway between, or $d_{i,j}/2$ away from the $\mathbf{s}_i$. The probability is

$$P\left[E_{j|i}\right] = \mathrm{Q}\left(\frac{d_{i,j}/2}{\sqrt{N_0/2}}\right) = \mathrm{Q}\left(\sqrt{\frac{d_{i,j}^2}{4}}\sqrt{\frac{2}{N_0}}\right) = \mathrm{Q}\left(\sqrt{\frac{d_{i,j}^2}{2N_0}}\right) \tag{58}$$

We often want the *pairwise probability of error* in terms of $\frac{\mathcal{E}_b}{N_0}$ or $\mathcal{E}_s/N_0$. To make this more explicit, I am also showing the first step in how to get this expression:

$$P\left[E_{j|i}\right] = \mathrm{Q}\left(\sqrt{\frac{d_{i,j}^2}{2N_0}}\right) = \mathrm{Q}\left(\sqrt{\frac{d_{i,j}^2}{2\mathcal{E}_s}\frac{\mathcal{E}_s}{N_0}}\right) = \mathrm{Q}\left(\sqrt{\frac{d_{i,j}^2}{2\mathcal{E}_b}\frac{\mathcal{E}_b}{N_0}}\right) \tag{59}$$

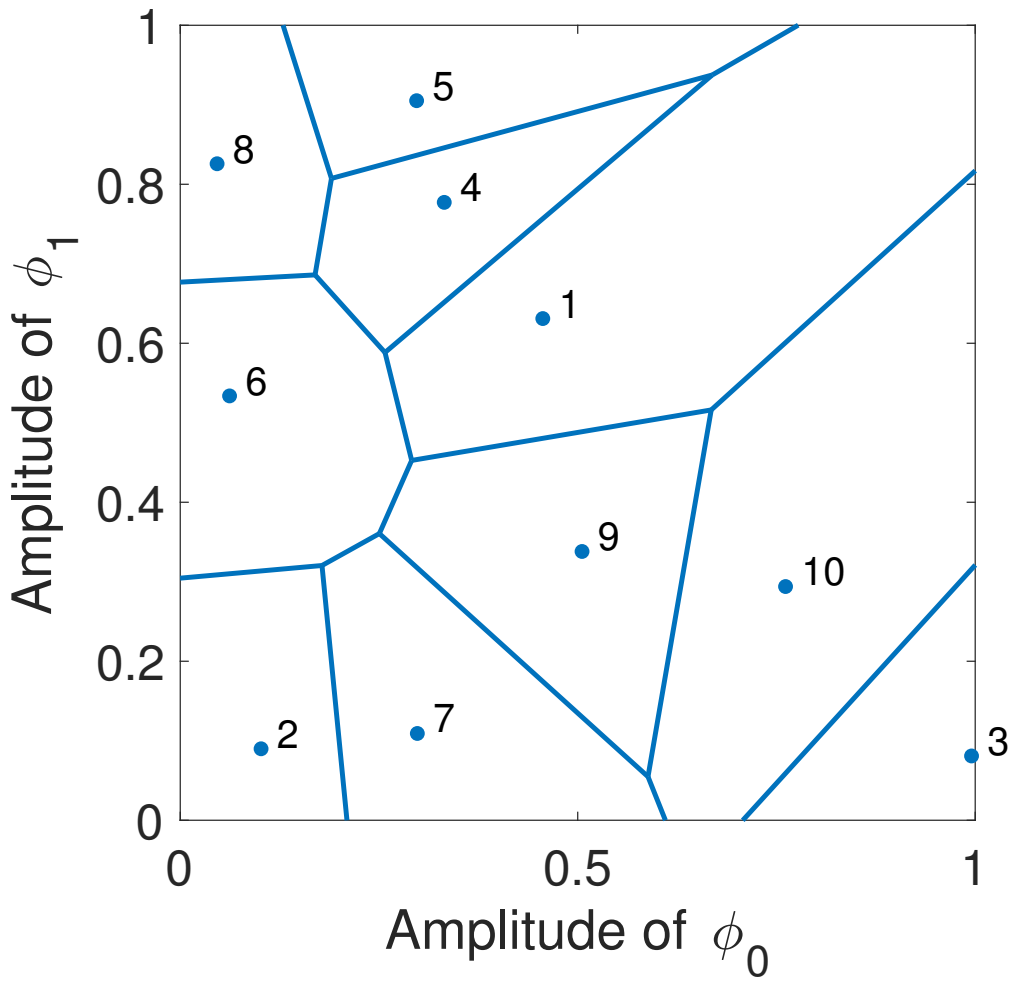


Figure 34: An example constellation diagram. Find the neighbors and nearest neighbors of each symbol.

If we use a constant A when describing the signal space vectors (as we usually do), then, since $\mathcal{E}_s$ will be proportional to A^2 and $d^2_{m,n}$ will be proportional to A^2, the factors A will cancel out of the expression.

A more detailed proof of (59) is detailed in Section 6.2 of the Rice book [11].

23.6 Union Bound

From 5510 or an equivalent class (or a Venn diagram) you may recall the probability formula, that for two events E and F that

$$P\left[E \cup F\right] = P\left[E\right] + P\left[F\right] - P\left[E \cap F\right]$$

You can prove this from the three axioms of probability. (This holds for *any* events E and F!) Then, using the above formula, and the first axiom of probability, we have that

$$P\left[E \cup F\right] \leq P\left[E\right] + P\left[F\right]. \tag{60}$$

Furthermore, from (60) it is straightforward to show that for *any* list of sets $E_1, E_2, \ldots E_n$ we have that

$$P\left[\bigcup_{i=1}^{n} E_i\right] \leq \sum_{i=1}^{n} P\left[E_i\right] \tag{61}$$

This is called the *union bound*, and it is very useful across communications. If you know one inequality, know this one. It is useful when the overlaps $E_i \cap E_j$ are small but difficult to calculate.

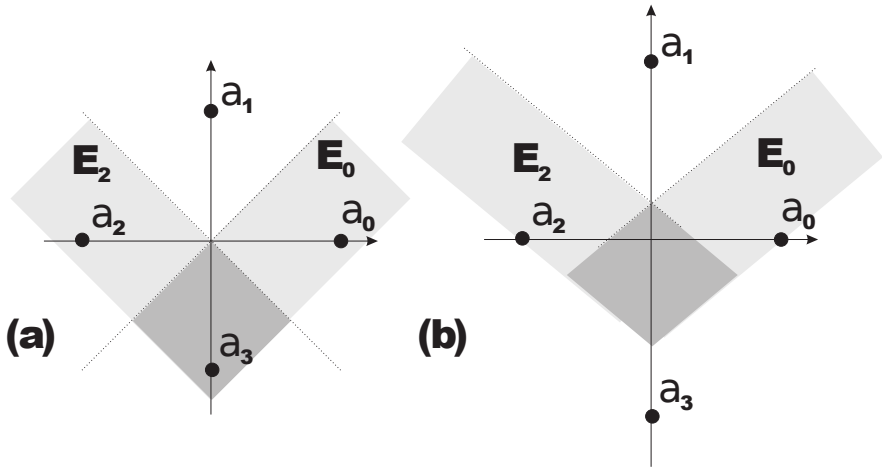


Figure 35: Union bound examples. (a) is QPSK with symbols of equal energy $\sqrt{\mathcal{E}_s}$. In (b) $\mathbf{s}_1 = -\mathbf{s}_3 = [0, \sqrt{3\mathcal{E}_s/2}]^T$ and $\mathbf{s}_2 = -\mathbf{s}_0 = [\sqrt{\mathcal{E}_s/2}, 0]^T$.

Example: QPSK

First, let's study the union bound for QPSK, as shown in Figure 35(a). Assume $s_1(t)$ is sent. We know that from our previous lecture on square QAM:

$$P(\text{bit error}) \approx \frac{4(\sqrt{M}-1)}{\sqrt{M}\log_2 M} \mathrm{Q}\left(\sqrt{\frac{3\log_2 M}{M-1}\frac{\mathcal{E}_b}{N_0}}\right). \tag{62}$$

For $M = 4$, this result uses $\sqrt{M} = 2$ and $\log_2 M = 2$, thus:

$$P(\text{bit error}) \approx \frac{4}{2(2)} \mathrm{Q}\left(\sqrt{\frac{3(2)}{3}\frac{\mathcal{E}_b}{N_0}}\right). \tag{63}$$

Or

$$P\left[\text{bit error}\right] = \mathrm{Q}\left(\sqrt{\frac{2\mathcal{E}_b}{N_0}}\right)$$

The probability of symbol error is one minus the probability that there were no error in either bit,

$$P\left[\text{symbol error}\right] = 1 - \left(1 - \mathrm{Q}\left(\sqrt{\frac{2\mathcal{E}_b}{N_0}}\right)\right)^2 \tag{64}$$

We can also write this as:

$$P\left[\text{symbol error}\right] = 2\mathrm{Q}\left(\sqrt{\frac{2\mathcal{E}_b}{N_0}}\right) - \left[2\mathrm{Q}\left(\sqrt{\frac{2\mathcal{E}_b}{N_0}}\right)\right]^2 \tag{65}$$

In contrast, let's calculate the union bound on the probability of error. There are two neighbors of each node i. Consider for example node $i = 1$. We can write

$$P\left[\text{symbol error}|H_1\right] = P\left[E_2 \cup E_0\right]$$

We ignored E_3 because it overlaps completely with $E_2 \cup E_0$. That is, $E_2 \cup E_0 \cup E_3 = E_2 \cup E_0$.

Then, we use the union bound.

$$P\left[\text{symbol error}|H_1\right] \leq P\left[E_2\right] + P\left[E_0\right]$$

These two probabilities are just the probability of error for a binary modulation, and both are identical, so

$$P\left[\text{symbol error}|H_1\right] \leq 2\mathrm{Q}\left(\sqrt{\frac{2\mathcal{E}_b}{N_0}}\right)$$

The overall probability of error is the average of $P\left[\text{symbol error}|H_i\right]$ for $i = 0, 1, 2, 3$; however these will all be identical due to the symmetry of QPSK. Thus $P\left[\text{symbol error}\right] = P\left[\text{symbol error}|H_1\right]$.

What is missing / What is the difference in this expression compared to (65)?

See Figure 36 to see the union bound probability of error plot, compared to the exact expression. Only at very low E_b/N_0 is there any noticeable difference!

23.7 General Application of Union Bound

In this class, our events are typically error events. Let $E_{j|i}$ represent the event that $\mathbf{x}$ is closer to $\mathbf{s}_j$ than to symbol $\mathbf{s}_i$ given that symbol i was actually sent. (Recall the half-spaces bordered by the perpendicular bisector of the line between $\mathbf{s}_j$ and $\mathbf{s}_i$ – it splits the symbol space into the space closer to $\mathbf{s}_i$ and the space closer to $\mathbf{s}_j$.) In this case, the union bound can be used to find the overall error given that i was sent:

$$P\left[\text{symbol error}|H_i\right] \leq \sum_{j \in N(i)} P\left[E_{j|i}\right] \tag{66}$$

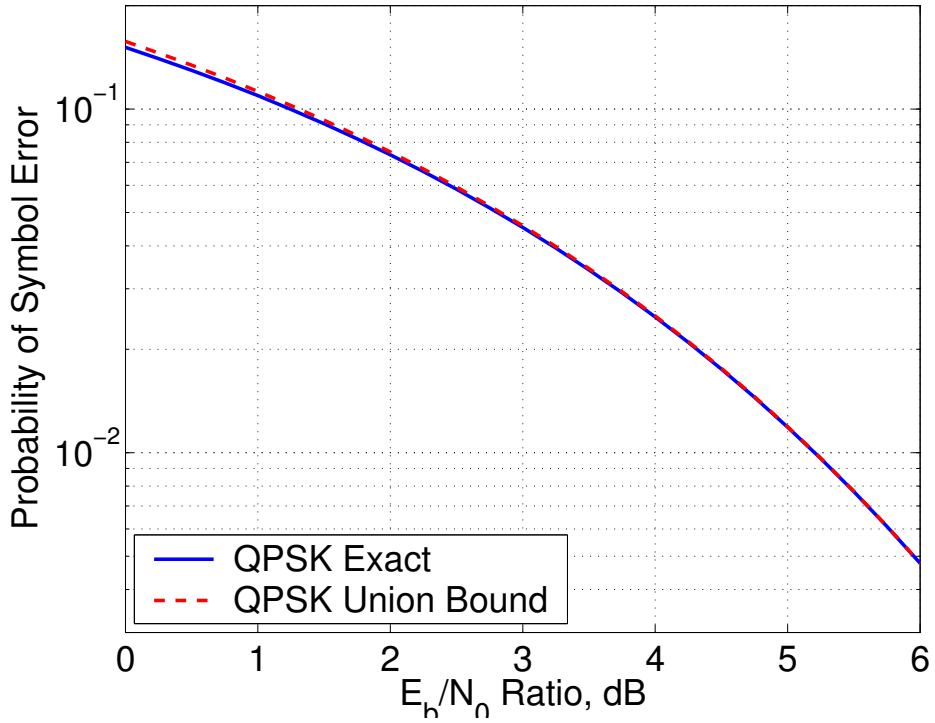


Figure 36: For QPSK, the exact probability of symbol error expression vs. the union bound.

These events $E_{j|i}$ for all $j \in N(i)$ cover all of the area outside of the decision region for i. That is, by combining their $P\left[E_{j|i}\right]$, we are greater than or equal to the probability of error given symbol i was sent.

The overall probability of error, averaged over all i that could be sent, is

$$P\left[\text{symbol error}\right] \leq \frac{1}{M} \sum_{i=0}^{M-1} \sum_{j \in N(i)} P\left[E_{j|i}\right] \tag{67}$$

Using the formula from (59),

$$P\left[\text{symbol error}\right] \leq \frac{1}{M} \sum_{i=0}^{M-1} \sum_{j \in N(i)} \mathrm{Q}\left(\sqrt{\frac{d_{i,j}^2}{2N_0}}\right) \tag{68}$$

The union bound gives a conservative estimate. This can be useful for quick initial study of a modulation type.

23.7.1 Rice book formula

This is the formula for the union bound given in Rice [11]:

$$P\left[\text{symbol error}\right] \quad \leq \quad \frac{1}{M} \sum_{m=0}^{M-1} \sum_{\substack{n=0 \\ n \neq m}}^{M-1} P\left[\text{decide } H_n | H_m\right]$$

Note the $\leq$ sign. This means the actual P [symbol error] will be *at most* this value. It is probably less than this value. This is a general formula and *is not necessarily the best* upper bound. What do we mean? If I draw any function that is always above the actual P [symbol error] formula, I have drawn an upper bound. But I could draw lots of functions that are upper bounds, some higher than others.

In particular, for some of the error events, decide $H_n|H_m$ may be redundant, and do not need to be included. We have talked about the concept of "neighboring" symbols and "nearest neighboring" symbols to symbol i, which we call respectively, $N(i)$ and $NN(i)$. The $N(i)$ are the ones that are necessary to include in the union bound.

Note that Rice [11] uses E_{avg} where I use $\mathcal{E}_s$ to denote average symbol energy. The Rice book uses E_b where I use $\mathcal{E}_b$ to denote average bit energy.

Example: 4-QAM with two amplitude levels
This is shown (poorly) in Figure 35(b). The amplitudes of the top and bottom symbols are $\sqrt{3}$ times the amplitude of the symbols on the right and left. (They are positioned to keep the distance between points in the signal space equal to $\sqrt{2\mathcal{E}_s/N_0}$.) I am calling this "2-amplitude 4-QAM" (I made it up).

What is the union bound on the probability of symbol error, given H_1?
Solution: Given symbol 1, the probability is the same as above. Defining E_2 and E_0 as above, these two distances between symbol $\mathbf{s}_1$ and $\mathbf{s}_2$ or $\mathbf{s}_0$ are the same: $\sqrt{2\mathcal{E}_s/N_0}$. Thus the formula for the union bound is the same.

What is the union bound on the probability of symbol error, given H_2? **Solution:** Now, it is

$$P\,[\text{symbol error}|H_2] \leq 3\text{Q}\left(\sqrt{\frac{2\mathcal{E}_b}{N_0}}\right)$$

So, overall, the union bound on probability of symbol error is

$$P\,[\text{symbol error}] \leq \frac{3\cdot 2+2\cdot 2}{4}\text{Q}\left(\sqrt{\frac{2\mathcal{E}_b}{N_0}}\right) = 2.5\text{Q}\left(\sqrt{\frac{2\mathcal{E}_b}{N_0}}\right).$$

How about average energy? For QPSK, the symbol energies are all equal. Thus $\mathcal{E}_{av} = \mathcal{E}_s$. For the two-amplitude 4-QAM modulation,

$$\mathcal{E}_{av} = \mathcal{E}_s\frac{2(0.5)+2(1.5)}{4} = \mathcal{E}_s$$

Thus there is no advantage to the two-amplitude QAM modulation in terms of average energy.

23.8 Nearest-Neighbor Approximate Probability of Error

As it turns out, the probability of error is often well approximated by the terms in the union bound with the smallest $d_{i,j}$. This is because higher $d_{i,j}$ means a higher argument in the Q-function, which in turn means a lower value of the Q-function. A little extra distance means a much lower value of the Q-function. So approximately,

$$P\,[\text{symbol error}] \approx \frac{N_{min}}{M}\text{Q}\left(\sqrt{\frac{d_{min}^2}{2N_0}}\right)$$

where

$$d_{min} = \min_{j \neq i} d_{i,j}$$

and N_{min} is the number of *ordered pairs* of symbols which are separated by distance d_{min}. Be sure to double count each pair, otherwise this formula won't work!

Example: 2-Amplitude 4-QAM
What is the nearest neighbor approximation for 2-Amplitude 4-QAM?
Solution: The minimum distance $d_{min} = 2A = \sqrt{2\mathcal{E}_s}$. Starting from symbol 0, I count 3, 2, 3, and 2 such distances. This is a total of $N_{min} = 10$. Thus

$$P\left[\text{symbol error}\right] \approx \frac{10}{4} \mathrm{Q}\left(\sqrt{\frac{2\mathcal{E}_b}{N_0}}\right)$$

The same as the union bound.

Lecture 15

Today: Probability of Error: (1) QAM / PSK examples, (2) FSK, (3) Differential PSK

- Reading for these notes: Rice [11] Sections 6.2, 7.7, and Proakis-Salehi [10] pages 423-427 (Section 7.6.6).
- Lecture videos for this lecture are at:
  `https://youtube.com/playlist?list=PLQuDEk4rPDOMdP9UqZtOGY2uOJSyNEBEU`

23.9 QAM/PSK Probability of Error Examples

Example: Probability of Error in M-PSK

Find the probability of error in M-ary PSK using the nearest neighbor approximation. Is this the same as the union bound?

Solution: Here the distance between two neighboring symbols can be calculated by seeing the origin and the two symbol points as forming an isosceles triangle with top angle $2\pi/M$, and equal sides having length A. Thus the length of the base is $d_{min} = 2A \sin \frac{2\pi}{2M}$. The squared distance is

$$d_{min}^2 = 4A^2 \sin^2(\pi/M)$$

The average energy per symbol is simply A^2 because all of the symbol points are A from the origin. Thus the average energy per bit is $\mathcal{E}_b = A^2 / \log_2 M$. Thus $A^2 = \mathcal{E}_b \log_2 M$. So:

$$d_{min}^2 = 4\mathcal{E}_b (\log_2 M) \sin^2(\pi/M)$$

Using the expression for the nearest neighbor approximation:

$$\begin{aligned} P\left[\text{symbol error}\right] &\approx \frac{N_{min}}{M}\mathrm{Q}\left(\sqrt{\frac{d_{min}^2}{2N_0}}\right) \\ &\approx \frac{2M}{M}\mathrm{Q}\left(\sqrt{4(\log_2 M)\sin^2(\pi/M)\frac{\mathcal{E}_b}{2N_0}}\right) \\ &\approx 2\mathrm{Q}\left(\sqrt{2(\log_2 M)\sin^2(\pi/M)\frac{\mathcal{E}_b}{N_0}}\right) \end{aligned}$$

This is the same as the union bound because each symbol only has two neighbors (symbols that contribute to the decision boundary).

The Rice book [11], Section 6.2, page 319-321, derives an approximate formula for the probability of error in M-PSK for $M > 4$. (Recall QPSK has an exact solution.) For $M > 4$, the book shows how to use a polar transformation and approximate the probability of error integral. The solution is exactly the same, in the end, as the nearest neighbor approximation / union bound expression in (69).

Example: Additional QAM / PSK Constellations

Solve for the probability of symbol error in the completely made up signal space diagrams in Figure 37. You should calculate:

- An exact expression if one should happen to be available,
- The union bound,
- The nearest-neighbor approximation.

Solve for the probability of symbol error first, and next the probability of bit error. Figure 37 is in terms of amplitude A, but all probability of error expressions should be written in terms of $\mathcal{E}_b/N_0$.

Example: 2 by 4 grid QAM
Solve for the probability of symbol error in the signal space diagrams in Rice Figure 5.3.4 (a), copied in these notes as Figure 38 (left). You should calculate:

- An exact expression,
- The union bound,
- The nearest-neighbor approximation.

Solution: (a) An exact expression is possible because the decision is separable. That is, the x_0 will decide two of the three bits, while x_1 will decide the third bit. Assume that nearest neighbors are separated by 2A. For the third bit, the probability of error is that of bipolar PAM, that is, $\mathrm{Q}\left(\sqrt{\frac{2A^2}{N_0}}\right)$. For the first two bits, the probability of bit error is approximately (assuming Gray coding),

$$\frac{1}{\log_2 M}\frac{2(M-1)}{M}\mathrm{Q}\left(\sqrt{\frac{2A^2}{N_0}}\right) = \frac{3}{4}\mathrm{Q}\left(\sqrt{\frac{2A^2}{N_0}}\right)$$

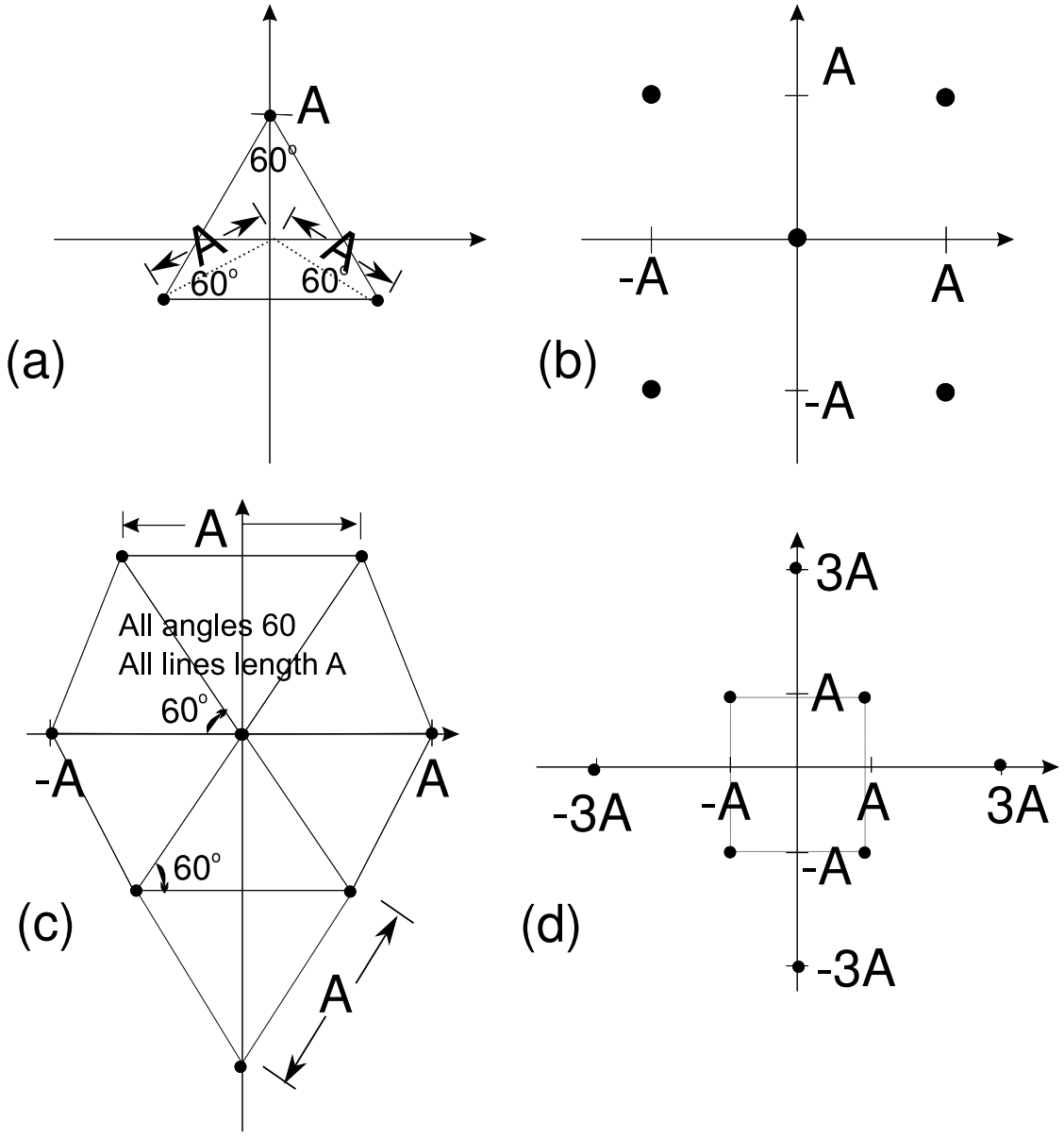


Figure 37: Constellation diagram for some example (made up) modulations.

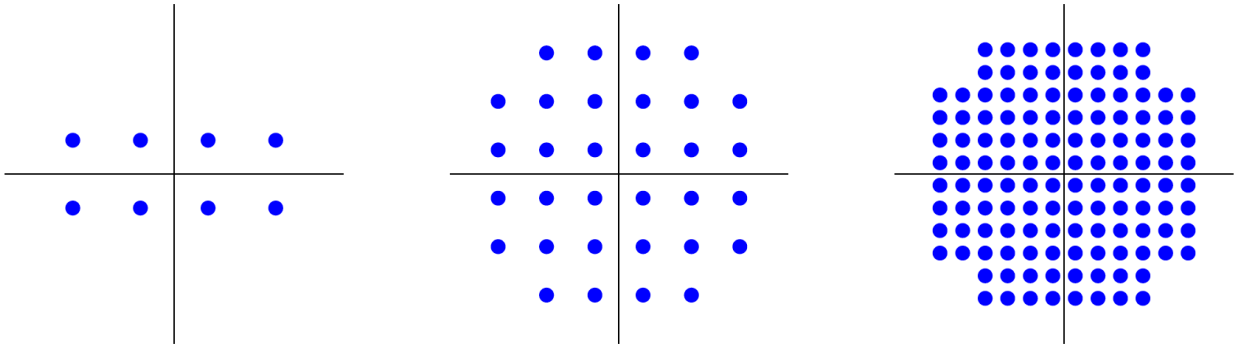


Figure 38: Constellation diagrams for Cross M-QAM for (left) $M = 8$, (center) $M = 32$, and (right) $M = 64$, from Rice Figure 5.3.4 [11].

The overall average probability of error will be a weighted average of these two – 2/3 times the probability of bit error in the first two bits and 1/3 times the probability of bit error in the third bit,

$$P\left[\text{bit error}\right] \approx \left[\frac{2}{3}\frac{3}{4} + \frac{1}{3}\right] \mathrm{Q}\left(\sqrt{\frac{2A^2}{N_0}}\right) = \frac{5}{6}\mathrm{Q}\left(\sqrt{\frac{2A^2}{N_0}}\right)$$

The average bit energy is

$$\begin{aligned} \mathcal{E}_b &= \frac{1}{\log_2 8}\mathcal{E}_s & (69) \\ &= \frac{1}{\log_2 8}\frac{1}{8}\left[4(2A^2) + 4 * (A^2 + 9A^2)\right] = 2A^2 & (70) \end{aligned}$$

Thus

$$P\left[\text{bit error}\right] = \frac{5}{6}\mathrm{Q}\left(\sqrt{\frac{\mathcal{E}_b}{N_0}}\right)$$

The union bound for (a) and the nearest neighbor approximation are the same. All nearest neighbors are separated by distance 2A. Four nodes have two neighbors, and four nodes have three neighbors. So $N_{min} = 20$, and $d_{min} = 2A$. So,

$$P\left[\text{symbol error}\right] \leq \frac{20}{8}\mathrm{Q}\left(\sqrt{\frac{4A^2}{2N_0}}\right) = 2.5\mathrm{Q}\left(\sqrt{\frac{\mathcal{E}_b}{N_0}}\right)$$

Note that this results in an approximate bit error rate of $\frac{5}{6}\mathrm{Q}\left(\sqrt{\frac{\mathcal{E}_b}{N_0}}\right)$, the same expression as above.

Example: Cross $M = 32$ QAM
Solve for the probability of symbol error in the signal space diagrams in Rice Figure 5.3.4 (b), copied to these notes as Figure 38 (center). You should calculate:

- The union bound, and
- The nearest-neighbor approximation.

Solution: The union bound requires us to count neighbors. Assume all nearest neighbors are separated by distance 2A.

1. Sixteen nodes (the center 4x4 grid) have four neighbors at distance $2A$,
2. Eight nodes have three neighbors at distance $2A$,
3. Eight nodes have two neighbors at distance $2A$ and one neighbor at distance $2\sqrt{2}A$.

Also, the average symbol energy is $\frac{1}{32}$ times twice the sum of the squared x-coordinates because of the symmetry of the constellation.

$$\mathcal{E}_s = \frac{2(12A^2 + 12(9A^2) + 8(25A^2))}{32} = 20A^2$$

Since $\log_2 M = 5$, we have $\mathcal{E}_b = \frac{\mathcal{E}_s}{5} = 4A^2$.

$$\begin{aligned} P\left[\text{symbol error}\right] &\leq \frac{1}{32}\left\{(16(4) + 8(3) + 8(2))\mathrm{Q}\left(\sqrt{\frac{4A^2}{2N_0}}\right) + 8\mathrm{Q}\left(\sqrt{\frac{8A^2}{2N_0}}\right)\right\} \\ &\leq \frac{13}{4}\mathrm{Q}\left(\sqrt{\frac{\mathcal{E}_b}{2N_0}}\right) + \frac{1}{4}\mathrm{Q}\left(\sqrt{\frac{\mathcal{E}_b}{N_0}}\right) \end{aligned} \tag{71}$$

The nearest neighbor approximation would simply remove the second term and replace the $\leq$ with an $\approx$.

24 FSK Probability of Error

24.1 Probability of Error for Coherent Binary FSK

First, let's look at coherent detection of binary FSK.

1. What is the detection threshold line separating the two decision regions?
2. What is the distance between points in the Binary FSK signal space?

What is the probability of error for coherent binary FSK? It is the same as bipolar PAM, but the symbols are spaced differently (more closely) as a function of $\mathcal{E}_b$. We had that

$$P\left[\text{error}\right]_{2-ary} = \mathrm{Q}\left(\sqrt{\frac{d_{0,1}^2}{2N_0}}\right)$$

Now, the spacing between symbols has reduced by a factor of $\sqrt{2}/2$ compared to bipolar PAM, to $d_{0,1} = \sqrt{2\mathcal{E}_b}$. So

$$P\left[\text{error}\right]_{2-Co-FSK} = \mathrm{Q}\left(\sqrt{\frac{\mathcal{E}_b}{N_0}}\right)$$

For the same probability of bit error, binary FSK is about 1.5 dB better than OOK (requires 1.5 dB less energy per bit), but 1.5 dB worse than bipolar PAM (requires 1.5 dB more energy per bit).

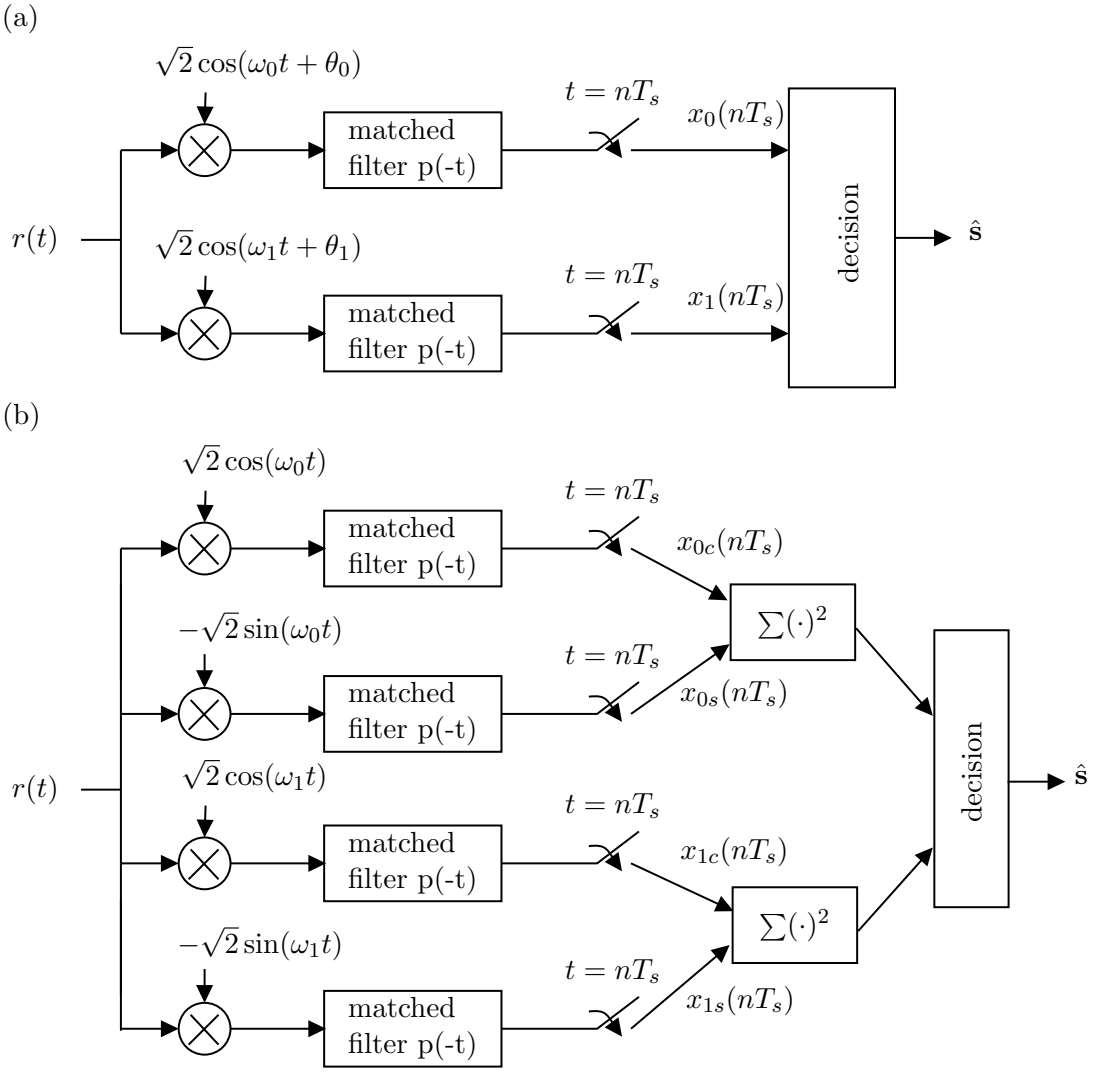


Figure 39: Block diagram of receivers for binary FSK. (a) A coherent RX synchronizes to the phases θ_0, θ_1 of the cosines at each frequency ω_0, ω_1, correlates with the two possible waveforms, and then decides which of the two symbols is closest. (b) A non-coherent RX does not try to estimate the phases, and instead, correlates with a cosine and a sine at each frequency ω_0 and ω_1, computes the energy in band k as $x_{kc}^2 + x_{ks}^2$, and finds which one has the highest energy.

24.2 Probability of Error for Noncoherent Binary FSK

The energy detector uses the energy in each frequency and selects the frequency with maximum energy.

This energy is denoted x_k^2 for frequency $k \in \{0, 1\}$ and is

$$x_k^2 = x_{kc}^2 + x_{ks}^2$$

This energy measure is a statistic which measures how much energy was in the signal at frequency f_k. The 'envelope' is a term used for the square root of the energy, so x_k is termed the envelope.

Question: What will x_k^2 equal when the noise is very small?

As it turns out, given the non-coherent receiver and x_{kc} and x_{ks}, the envelope x_k is an optimum (sufficient) statistic to use to decide between $s_0 \ldots s_{M-1}$.

What do they do to prove this in Proakis & Salehi? They prove it for *binary* non-coherent FSK. It takes quite a bit to do this proof; one needs to have some practice in transformations of random variables. A sketch:

1. Define the received vector $\mathbf{x}$ as a 4 length vector of the correlation of $r(t)$ with the sin and cos at each frequency f_0, f_1.

2. They formulate the prior probabilities $f_{\mathbf{x}|H_i}(\mathbf{x}|H_i)$. Note that this depends on θ_k, which is assumed to be uniform between 0 and 2π, and independent of the noise.

$$\begin{aligned} f_{\mathbf{x}|H_i}(\mathbf{x}|H_i) &= \int_0^{2\pi} f_{\mathbf{x},\theta_k|H_i}(\mathbf{x},\theta|H_i)d\theta \\ &= \int_0^{2\pi} f_{\mathbf{x}|\theta_k,H_i}(\mathbf{x}|\theta,H_i) f_{\theta_k|H_i}(\theta|H_i)d\theta \end{aligned} \tag{72}$$

   Note that $f_{\mathbf{x}|\theta_k,H_0}(\mathbf{x}|\theta,H_0)$ is a 2-D Gaussian random vector with i.i.d. components.

3. They formulate the joint probabilities $f_{\mathbf{x}\cap H_0}(\mathbf{x}\cap H_0)$ and $f_{\mathbf{x}\cap H_1}(\mathbf{x}\cap H_1)$.

4. Where the joint probability $f_{\mathbf{x}\cap H_0}(\mathbf{x}\cap H_0)$ is greater than $f_{\mathbf{x}|H_1}(\mathbf{x}|H_1)$, the receiver decides H_0. Otherwise, it decides H_1.

5. The decisions in this last step, after manipulation of the pdfs, are shown to reduce to this decision (given that $P[H_0] = P[H_1]$):

$$\sqrt{x_{0c}^2 + x_{0s}^2} \underset{H_1}{\overset{H_0}{\gtrless}} \sqrt{x_{1c}^2 + x_{1s}^2}$$

The "envelope detector" can equally well be called the "energy detector", and it often is. The full proof of the probability of error is in Proakis & Salehi, Section 7.6.9, page 430 (which is posted on Canvas). The expression for probability of error in binary non-coherent FSK is given by,

$$P[\text{error}]_{2-NC-FSK} = \frac{1}{2}\exp\left[-\frac{\mathcal{E}_b}{2N_0}\right] \tag{73}$$

The expressions for probability of error in binary FSK (both coherent and non-coherent) are important, and you should make note of them. You will use them to be able to design communication systems that use FSK.

25 Differential Coding for BPSK

Def'n: *Coherent Reception*
The reception of a signal when its carrier phase is explicitly determined and used for demodulation.

For coherent reception of PSK, will always need some kind of phase synchronization in BPSK. Typically, this means transmitting a training sequence.

For non-coherent reception of PSK, we use differential encoding (at the transmitter) and decoding (at the receiver).

25.1 DPSK Transmitter

Now, consider the bit sequence $\{b_n\}$, where b_n is the nth bit that we want to send. The sequence b_n is a sequence of 0's and 1's. How do we decide which phase to send? Prior to this, we've said, send $\mathbf{s}_0$ if $b_n = 0$, and send $\mathbf{s}_1$ if $b_n = 1$.

Instead of setting k for $\mathbf{s}_k$ only as a function of b_n, in differential encoding, we also include k_{n-1}. Now,

$$k_n = \begin{cases} k_{n-1}, & b_n = 0 \\ 1 - k_{n-1}, & b_n = 1 \end{cases}$$

Note that $1 - k_{n-1}$ is the complement or negation of k_{n-1} – if $k_{n-1} = 1$ then $1 - k_{n-1} = 0$; if $k_{n-1} = 0$ then $1 - k_{n-1} = 1$. Basically, for differential BPSK, a switch in the angle of the signal space vector from 0^o to 180^o or vice versa indicates a bit 1; while staying at the same angle indicates a bit 0.

Note that the TX and RX have to agree on the "zero" phase. Typically $k_0 = 0$. This becomes an *extra* bit sent with the data bits.

Example: Differential encoding

Let $\mathbf{b} = [1, 0, 1, 0, 1, 1, 1, 0, 0]$. Assume $b_0 = 0$. What symbols $\mathbf{k} = [k_0, \ldots, k_9]^T$ will be sent?
Solution:

$$\mathbf{k} = [0, 1, 1, 0, 0, 1, 0, 1, 1, 1]^T$$

These values of k_n correspond to a symbol stream with phases:

$$\angle\mathbf{s} = [0, \pi, \pi, 0, 0, \pi, 0, \pi, \pi, \pi]^T$$

25.2 DPSK Receiver

Now, at the receiver, we find b_n by comparing the phase of x_n to the phase of x_{n-1}. What our receiver does, is to measure the angle difference is small (close to zero) or large (bigger than $\pi/2$)

$$\cos(\angle x_n - \angle x_{n-1})$$

If this statistic is less than zero, decide $b_n = 1$, and if it is greater than zero, decide $b_n = 0$.

Example: Differential decoding

1. Assuming no phase shift in the above encoding example, show that the receiver will decode the original bitstream with differential decoding. **Solution:** Starting with the 2nd element of $\angle\mathbf{s}$ above,
$$\hat{b}_n = [1, 0, 1, 0, 1, 1, 1, 0, 0]^T.$$

2. Now, assume that all bits are shifted π radians and we receive
$$\angle\mathbf{x}' = [\pi, 0, 0, \pi, \pi, 0, \pi, 0, 0, 0].$$
What will be decoded at the receiver? **Solution:**
$$\hat{b}_n = [1, 0, 1, 0, 1, 1, 1, 0, 0].$$

Rotating *all* symbols by π radians does not cause any bit error.

25.3 Probability of Bit Error for DPSK

The probability of bit error in DPSK is slightly worse than that for BPSK:
$$P\left[\text{error}\right] = \frac{1}{2}\exp\left(-\frac{\mathcal{E}_b}{N_0}\right)$$
For a constant probability of error, DPSK requires about 1 dB more $\frac{\mathcal{E}_b}{N_0}$ than BPSK, which has probability of bit error $\mathrm{Q}\left(\sqrt{\frac{2\mathcal{E}_b}{N_0}}\right)$. Both are plotted in Figure 40.

Lecture 16

Today: (1) $M > 2$ FSK Prob. Error, (2) Modulation Comparison

- Lecture videos for this lecture are at:
`https://youtube.com/playlist?list=PLQuDEk4rPDOPQY_U9RlV2_7SwFv8GL6iZ`

26 M-ary FSK Probability of Error

26.1 M-ary Non-Coherent FSK

For M-ary non-coherent FSK, the derivation in the Proakis & Salehi book, section 7.6.9, provides an exact expression for the probability of error in M-ary FSK. The result is that
$$P\left[\text{symbol error}\right] = \sum_{n=1}^{M-1}(-1)^{n+1}\binom{M-1}{n}\frac{1}{n+1}e^{-\log_2 M\frac{n}{n+1}\frac{\mathcal{E}_b}{N_0}},$$
and
$$P\left[\text{error}\right]_{M-nc-FSK} = \frac{M/2}{M-1}P\left[\text{symbol error}\right].$$

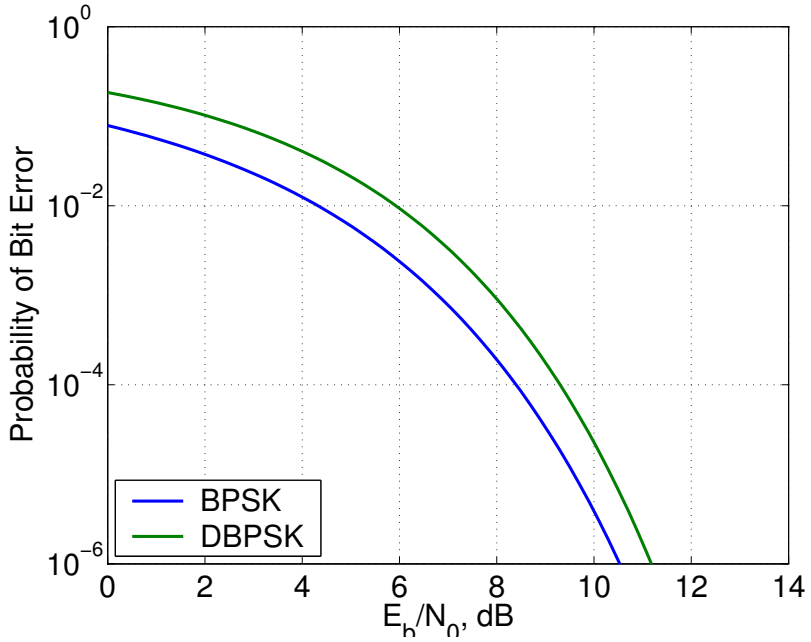


Figure 40: Comparison of probability of bit error for BPSK and Differential BPSK.

See Figure 41.

Proof Summary: Our non-coherent receiver finds the energy in each frequency. These energy values no longer have a Gaussian distribution (due to the squaring of the amplitudes in the energy calculation). They instead are either Rician (for the transmitted frequency) or Rayleigh distributed (for the "other" $M-1$ frequencies). The probability that the correct frequency is selected is the probability that the Rician random variable is larger than all of the other random variables measured at the other frequencies.

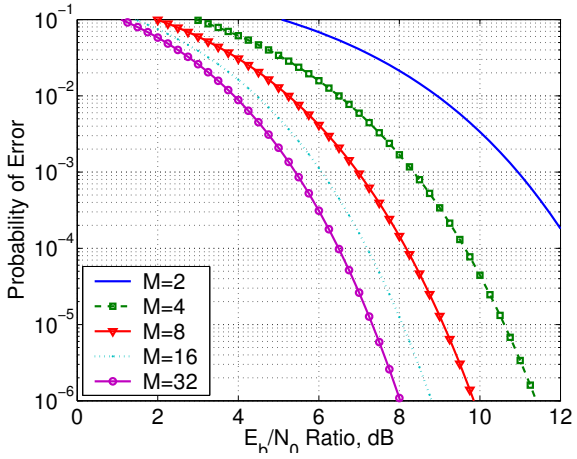


Figure 41: Probability of bit error for non-coherent reception of M-ary FSK.

Example: Probability of Error for Non-coherent $M = 2$ case

Use the above expressions to find the P [symbol error] and P [error] for binary non-coherent FSK.

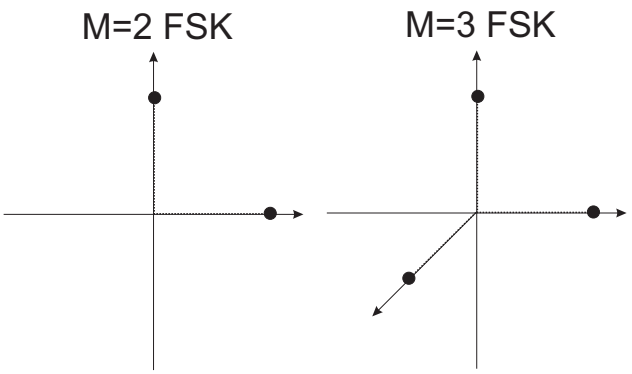


Figure 42: Constellation for $M = 2$, $M = 3$ FSK.

Solution:

$$P\left[\text{symbol error}\right] = P\left[\text{bit error}\right] = \frac{1}{2}e^{-\frac{1}{2}\frac{\mathcal{E}_b}{N_0}}$$

26.2 M-ary FSK Coherent Receiver

For M-ary FSK with $M > 2$, we don't have an exact expression for the probability of symbol error. Instead, we use the union bound. How many neighbors does each symbol have? You can the constellation visually when $M = 2$ or $M = 3$ (in Figure 42). For $M > 3$, consider plotting any three dimensions of the constellation, and one of the three axes is symbol 0. You can see that both other symbols contribute a plane that contributes to the decision region of the symbol 0. This is true regardless of which other symbols were chosen. Thus each symbol has $M - 1$ neighbors!

The distance between neighbors is always $d = \sqrt{2}A$, and the average energy per symbol is $\mathcal{E}_s = A^2$, which means that $\mathcal{E}_b = A^2/\log_2 M$. Thus:

$$P\left[\text{symbol error}\right] \leq (M-1)\mathrm{Q}\left(\sqrt{\log_2 M \frac{\mathcal{E}_b}{N_0}}\right) \tag{74}$$

Note however that one cannot do Gray encoding on the bits assigned to the M symbols. For symbol i, all $M-1$ other symbols are neighbors to it, and they are all equally distant from i. Thus when a symbol error is made, it is equally likely to be to symbol $j \neq i$. What is the average number of bit errors made when a symbol error is made?

The Proakis & Salehi book provides a derivation for the result, which is:

$$P\left[\text{bit error}\right] = \frac{M/2}{M-1}P\left[\text{symbol error}\right] \leq \frac{M}{2}\mathrm{Q}\left(\sqrt{(\log_2 M)\frac{\mathcal{E}_b}{N_0}}\right). \tag{75}$$

Here is a short argument about why we get the result we see in the Proakis & Salehi handout: If I randomly pick any symbol, it will have $(\log_2 M)/2$ bit errors per symbol. However, this includes the correct symbol. Since we need to exclude the correct symbol, we need to multiply by a factor of $M/(M-1)$, that is $\frac{M\log_2 M}{(M-1)2}$ bit errors per symbol. Next, because this is the number of bit errors per symbol, we divide by $\log_2 M$ to get the number of bit errors per bit. That is, $\frac{M/2}{M-1}$.

Name	$P\left[\textbf{symbol error}\right]$	$P\left[\textbf{bit error}\right]$
BPSK	$= \mathrm{Q}\left(\sqrt{\frac{2\mathcal{E}_b}{N_0}}\right)$	same
OOK	$= \mathrm{Q}\left(\sqrt{\frac{\mathcal{E}_b}{N_0}}\right)$	same
DPSK	$= \frac{1}{2}\exp\left(-\frac{\mathcal{E}_b}{N_0}\right)$	same
M-PAM	$= \frac{2(M-1)}{M}\mathrm{Q}\left(\sqrt{\frac{6\log_2 M}{M^2-1}\frac{\mathcal{E}_b}{N_0}}\right)$	$\approx \frac{1}{\log_2 M}P\left[\text{symbol error}\right]$
QPSK		$= \mathrm{Q}\left(\sqrt{\frac{2\mathcal{E}_b}{N_0}}\right)$
M-PSK	$\leq 2\mathrm{Q}\left(\sqrt{2(\log_2 M)\sin^2(\pi/M)\frac{\mathcal{E}_b}{N_0}}\right)$	$\approx \frac{1}{\log_2 M}P\left[\text{symbol error}\right]$
Square M-QAM		$\approx \frac{4}{\log_2 M}\frac{(\sqrt{M}-1)}{\sqrt{M}}\mathrm{Q}\left(\sqrt{\frac{3\log_2 M}{M-1}\frac{\mathcal{E}_b}{N_0}}\right)$
2-non-co-FSK	$= \frac{1}{2}\exp\left[-\frac{\mathcal{E}_b}{2N_0}\right]$	same
M-non-co-FSK	$= \sum_{n=1}^{M-1}\binom{M-1}{n}\frac{(-1)^{n+1}}{n+1}\exp\left[-\frac{n\log_2 M}{n+1}\frac{\mathcal{E}_b}{N_0}\right]$	$= \frac{M/2}{M-1}P\left[\text{symbol error}\right]$
2-co-FSK	$= \mathrm{Q}\left(\sqrt{\frac{\mathcal{E}_b}{N_0}}\right)$	same
M-co-FSK	$\leq (M-1)\mathrm{Q}\left(\sqrt{\log_2 M\frac{\mathcal{E}_b}{N_0}}\right)$	$= \frac{M/2}{M-1}P\left[\text{symbol error}\right]$

Table 2: Summary of probability of bit and symbol error formulas for several modulations.

27 Fidelity Comparison: $P\left[\textbf{error}\right]$ vs. $\frac{\mathcal{E}_b}{N_0}$

Main modulations which we have evaluated probability of error vs. $\frac{\mathcal{E}_b}{N_0}$:

1. M-ary PAM, including Binary PAM or BPSK, OOK, DPSK.
2. M-ary PSK, including QPSK.
3. Square QAM
4. Non-square QAM constellations
5. FSK, M-ary FSK

In this part of the lecture we will break up into groups and derive: (1) the probability of error and (2) probability of symbol error formulas for these types of modulations.

See also Rice Section 6.3 [11].

27.1 Bandwidth Efficiency Comparison

We've talked about measuring data rate in bits per second. We've also talked about Hertz, *i.e.*, the quantity of spectrum our signal will use. Typically, we can scale a system, to increase the bit rate by decreasing the symbol period, and correspondingly increase the bandwidth. These two have a linear proportional relationship.

Def'n: *Bandwidth efficiency*
The bandwidth efficiency, typically denoted η, is the ratio of bits per second to bandwidth:

$$\eta = R_b/B_T$$

Bandwidth efficiency depends on the definition of "bandwidth". Since it is usually used for comparative purposes, we just make sure we use the same definition of bandwidth throughout a comparison.

η	$\alpha = 0$	$\alpha = 0.5$	$\alpha = 1$
BPSK	1.0	0.67	0.5
QPSK	2.0	1.33	1.5
16-QAM	4.0	2.67	2.0
64-QAM	6.0	4.0	3.0

Table 3: Bandwidth efficiency of PSK and QAM modulation methods using raised cosine filtering as a function of α.

The key figure of merit: bits per second / Hertz, *i.e.*, bps/Hz.

27.1.1 PSK, PAM and QAM

In these three modulation methods, the bandwidth is largely determined by the pulse shape. For root raised cosine filtering, the null-null bandwidth is $1 + \alpha$ times the bandwidth of the case when we use pure sinc pulses. The transmission bandwidth (for a bandpass signal) is

$$B_T = \frac{1+\alpha}{T_s}$$

Since T_s is seconds per symbol, we divide by $\log_2 M$ bits per symbol to get $T_b = T_s/\log_2 M$ seconds per bit, or

$$B_T = \frac{(1+\alpha)R_b}{\log_2 M}$$

where $R_b = 1/T_b$ is the bit rate.

Bandwidth efficiency is then

$$\eta = R_b/B_T = \frac{\log_2 M}{1+\alpha}$$

See Table 3 for some numerical examples.

27.1.2 FSK

We've said that the bandwidth of FSK is,

$$B_T = (M-1)\Delta f + 2B$$

where B is the one-sided bandwidth of the digital baseband signal. For the null-to-null bandwidth of raised-cosine pulse shaping, $2B = (1+\alpha)/T_s$. So,

$$B_T = (M-1)\Delta f + (1+\alpha)/T_s = \frac{R_b}{\log_2 M}\{(M-1)\Delta f T_s + (1+\alpha)\}$$

since $R_b = 1/T_s$ for

$$\eta = R_b/B_T = \frac{\log_2 M}{(M-1)\Delta f T_s + (1+\alpha)}$$

If $\Delta f = 1/T_s$ (required for non-coherent reception),

$$\eta = \frac{\log_2 M}{M+\alpha}$$

27.2 Bandwidth Efficiency vs. $\frac{\mathcal{E}_b}{N_0}$

For each modulation format, we have quantities of interest:

- Bandwidth efficiency, and
- Energy per bit ($\frac{\mathcal{E}_b}{N_0}$) requirements to achieve a given probability of error.

Example: Bandwidth efficiency vs. $\frac{\mathcal{E}_b}{N_0}$ for $M = 8$ PSK
What is the required $\frac{\mathcal{E}_b}{N_0}$ for 8-PSK to achieve a probability of bit error of 10^{-6}? What is the bandwidth efficiency of 8-PSK when using 50% excess bandwidth?
Solution: Given in Rice Figure 6.3.5 (Figure 6.13) to be about 14 dB, and 2 [11].

We can plot these (required $\frac{\mathcal{E}_b}{N_0}$, bandwidth efficiency) pairs. See Rice Figure 6.3.6 [11].

Lecture 17

Today: (1) Received Power Models, (2) Noise Energy, (3) Link Budgeting

- Reading for these notes: Rice [11] Section 6.4.
- Lecture videos for this lecture are at:
`https://youtube.com/playlist?list=PLQuDEk4rPDOOy1W2LDaUQGQIvEa8CUXiL`

28 Noise and Received Power Models

This section starts to make the connection between the energy per bit which we use in probability of error formulas to the transmit power and distance between the transmitter and receiver in a given communication system. Given that we want our wireless communication system to operate in some application or for some link, how can we calculate how much power will be received, typically? We do this here for both wireless and wired channels. Wireless channels we divide into free space channels, and obstructed (non-free-space channels). My research and experience is in the design of wireless systems, so I must warn you that I provide much more detail for wireless than for wired communication systems.

28.1 Free Space

'Free space' is the idealization in which nothing exists except for the transmitter and receiver, and can really only be used for deep space communications. In addition, this formula serves as a starting point for other radio propagation formulas. In free space, the received power is calculated from the Friis formula,

$$C = P_R = P_T G_T G_R \left(\frac{\lambda}{4\pi R_0}\right)^2 \left(\frac{R_0}{R}\right)^2 \tag{76}$$

where

- G_T and G_R are the antenna gains at the transmitter and receiver, respectively.
- P_T is the transmit power.

- λ is the wavelength at signal frequency. For narrowband signals, the wavelength is nearly constant across the bandwidth, so we just use the center frequency f_c. Note that $\lambda = c/f_c$ where $c = 3 \times 10^8$ meters per second is the speed of light.
- R_0 is a reference distance, say 1 meter or 10 meters. You may notice that the R_0^2 variables cancel, so why have I put them in? It is useful for understanding the relationships and for keeping track of the units. 1) One can measure the received power at a reference distance R_0 during testing, and then replace the $P_T G_T G_R \left(\frac{\lambda}{4\pi R_0}\right)^2$ part with the measurement; 2) Each fraction has units that cancel; and 3) We consider the $\left(\frac{R_0}{R}\right)^2$ part as the *path loss*, i.e., loss due to the actual path length (distance between the transmitter and receiver).

In (76), everything is in linear terms. Typically communication systems engineers use decibels to express these numbers. As a reminder and we write $P_R\,[\text{dBm}]$ or $G_T\,[\text{dB}]$, and their values are given by:

$$\begin{aligned} P_R\,[\text{dBm}] &= 10\log_{10}\frac{P_R}{1\text{ mW}} \\ G_T\,[\text{dB}] &= 10\log_{10} G_T \end{aligned} \tag{77}$$

The Friis formula of (76), given in dB, is

$$C\,[\text{dBm}] = G_T\,[\text{dB}] + G_R\,[\text{dB}] + P_T\,[\text{dBm}] + 20\log_{10}\frac{\lambda}{4\pi R_0} - 20\log_{10}\frac{R}{R_0} \tag{78}$$

This says the received power, C, is linearly proportional to the log of the distance R.

The received power at the reference distance R_0 is called $P_0\,[\text{dBm}]$ and is given by:

$$P_0\,[\text{dBm}] = G_T\,[\text{dB}] + G_R\,[\text{dB}] + P_T\,[\text{dBm}] + 20\log_{10}\frac{\lambda}{4\pi R_0} \tag{79}$$

Again, $P_0\,[\text{dBm}]$ is something you might measure with a well-calibrated receiver and an antenna. If you know (or estimate) $P_0\,[\text{dBm}]$, then the relationship of received power (dBm) with distance R is simply:

$$C\,[\text{dBm}] = P_0\,[\text{dBm}] - 20\log_{10}\frac{R}{R_0}. \tag{80}$$

There are also typically other losses in a transmitter / receiver system; losses in a cable, other imperfections, etc. In the Rice book, these are labelled as L[dB]. You must subtract the dB loss from the left-hand side of (78):

$$C\,[\text{dBm}] = P_0\,[\text{dBm}] - 20\log_{10}\frac{R}{R_0} - L\,[\text{dB}]\,. \tag{81}$$

Note that you need to be careful (even [11] is sometimes ambiguous) that if you call it a dB "loss", it is value that you subtract from (80). Some people will call it a gain $G\,[\text{dB}] = -L\,[\text{dB}]$, but you if you call it a "gain" than add $G\,[\text{dB}]$ to (80).

28.2 Non-free-space Channels

We don't study radio propagation path loss formulas in this course. But, a very short summary is that radio propagation on Earth is different than the Friis formula suggests. Lots of other formulas exist that approximate the received power as a function of distance, antenna heights, type of environment, etc. The main problem is that walls, trees, buildings, and terrain decrease the power that would be received as compared to the Friis model. This is called "shadowing" in analogy to how objects shadow light.

The most common model for real world shadowing effects is the path loss exponent model, in which received power (dBm) is linear with the log of distance,

$$C\,[\text{dBm}] = P_0\,[\text{dBm}] - 10n_p \log_{10} \frac{R}{R_0}, \tag{82}$$

for some constant n_p. Here, $P_0\,[\text{dBm}]$ is typically the same as we described earlier, that is, the path loss that the Friis model would predict at some short distance R_0 from the antenna. Usually R_0 is a very short distance, like 1 m or 10 m. Effectively, because of shadowing caused by buildings, trees, etc., the average loss may increase more quickly than $1/R^2$, instead, it may be more like $1/R^{n_p}$.

Equivalently, in linear terms:

$$C = P_R = P_T G_T G_R \left(\frac{\lambda}{4\pi R_0}\right)^2 \left(\frac{R_0}{R}\right)^{n_p}$$

This is a model that comes with lots of evidence from empirical measurement studies. In dense urban areas like Manhattan, people observe a much higher n_p than in rural areas. In mountainous areas, we typically observe a higher n_p than in flat areas. When the antennas are both close to the ground, the n_p is higher than when the antennas are held tens of meters away from the ground. Finally, in buildings, we observe a higher n_p when the walls are made of bricks or concrete, than when the walls are drywall and wood frame. The value of n_p may range from 2 to 4.5, with some values going lower than 2 or higher than 4.5. In real life, you would want to measure n_p in the type of environment you want to have your link operate in, or use values from the literature on measurement-based path loss models.

However, note that (82) is only the *average* received power. In reality, any one particular measurement of received power a distance R from the transmitter may be off by 0-30 dB from the average, due to differences in the particular path that the signal travels, due to small-scale fading, and for other reasons.

28.3 Wired Channels

Typically wired channels are lossy as well, but the loss is modeled as linear in the length of the cable. For example,

$$C\,[\text{dBm}] = P_T\,[\text{dBm}] - R(L_{1m}\,[\text{dB}])$$

where P_T is the transmit power and R is the cable length in meters, and L_{1m} is the loss per meter.

28.4 Noise Energy

The noise energy N_0 can be calculated as:

$$N_0 = kT_{eq}$$

where $k = 1.38 \times 10^{-23}$ J/K is Boltzmann's constant and T_{eq} is called the equivalent noise temperature in Kelvin. This is a topic covered in another course, and so you are not responsible for that material here. But in short, the equivalent temperature is a function of the receiver design, and T_0, the temperature of the environment at which the antenna is receiving from. T_{eq} is always going to be higher than T_0. Basically, all receiver circuits add noise to the received signal. With proper design (and more expensive and power-hungry components), T_{eq} can be kept low.

29 System Design

This section makes connections between the modulation performance formulas we've already discussed to the received power models just mentioned, in order to analyze a system design.

For example, we may want to figure out what modulation, bit rate, and transmit power to use for a deep space communication system. Or for a new long-range IoT system. Or for a fiber-optic communications system. You may find that you need several missing (or forgotten) connections to other topics in order to design a real-world communication system. This lecture and the next lecture are designed to present these connections.

As a digital communication system designer, your mission (if you choose to accept it) is to achieve:

1. High data rate
2. High fidelity (low bit error rate)
3. Low transmit power
4. Low bandwidth
5. Low transmitter/receiver complexity
6. Long range

But this is truly a mission impossible, because you can't have everything at the same time. So the system design depends on what the desired system really needs and what are the acceptable trade-offs. Typically some subset of requirements are given; for example, given the bandwidth limits, the received signal power and noise power, and bit error rate limit, what data rate can be achieved? Using which modulation?

30 Using the Relationship Flow Chart

For a typical system design question, we are given some constraints (at the given limits) and asked to determine other system parameters. For these types of problems I find it helpful to use the flow chart in Figure 43 to keep track of my given constraints, the many functional relationships that exist in wireless communication system design, and thus how to get to the parameter of interest.

In this lecture, we discuss this procedure, and define each of the variables we see in Figure 43, and to what they are related. This lecture is about system design, and it cuts across electrical and computer engineering classes; in particular circuits, radio propagation, antennas, optics, and the material from this class. You are expected to apply what you have learned in other classes (or learn the functional relationships that we describe here).

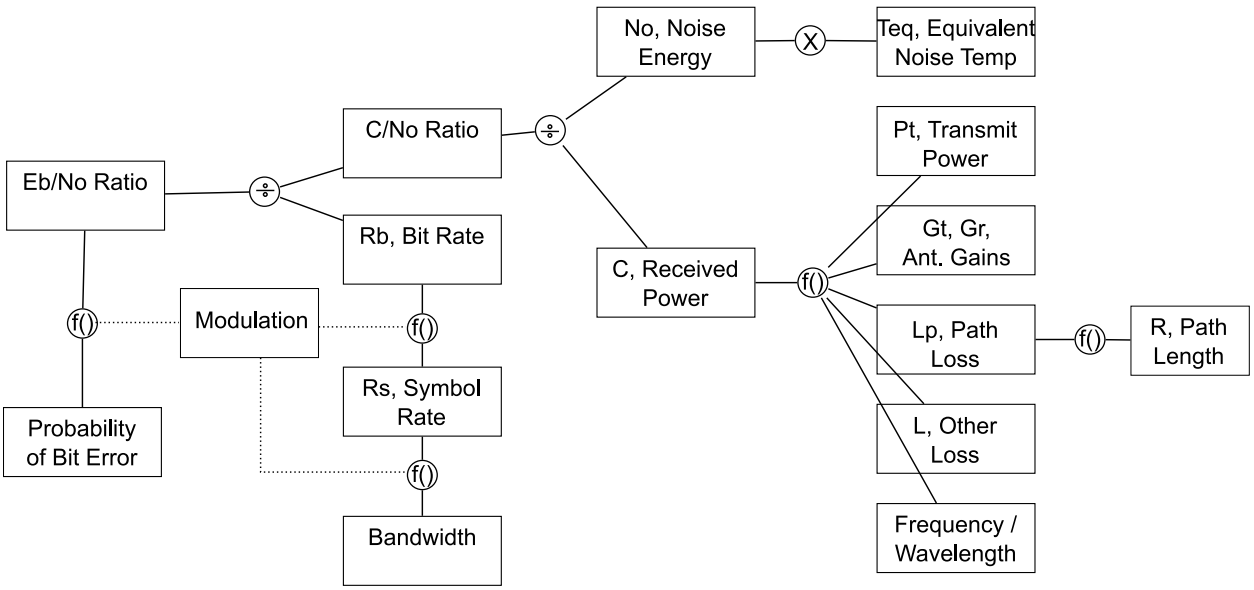


Figure 43: Relationships between important system design variables (rectangular boxes). Functional relationships between variables are given as circles, with division ($\div$), multiplication ($\times$), or another more complicated function $f()$. For example, C/N_0 is shown to have a divide by relationship with C and N_0. The effect of the choice of modulation impacts several functional relationships, *e.g.*, the relationship between probability of bit error and $\frac{\mathcal{E}_b}{N_0}$, which is drawn as a dotted line.

30.1 Link Budgets Given C/N_0

The received power is denoted C, it has units of Watts. What is C/N_0? It is received power divided by noise energy. It is a nebulous quantity, but it summarizes what we need to know about the signal and the noise for the purposes of system design.

- We (and other books) often describe both the received power as P_R, but in Rice [11] it is typically denoted C.
- We know the probability of bit error is typically written as a function of $\frac{\mathcal{E}_b}{N_0}$. The noise energy is N_0. The bit energy is $\mathcal{E}_b$. We can write $\mathcal{E}_b = CT_b$, since energy is power $\times$ time. To separate the effect of T_b, we often denote:

$$\frac{\mathcal{E}_b}{N_0} = \frac{C}{N_0} T_b = \frac{C/N_0}{R_b}$$

  where $R_b = 1/T_b$ is the bit rate. In other words, $C/N_0 = \frac{\mathcal{E}_b}{N_0} R_b$ What are the units of C/N_0? *Answer: Hz, 1/s.*
- Note that people often report C/N_0 in dB Hz, which is

$$10 \log_{10} \frac{C}{N_0}$$

- Be careful of Bytes (B) per second vs bits (b) per second. Commonly, computer science people use Bps (kBps or MBps) when describing data rate. For example, if it takes 5 seconds

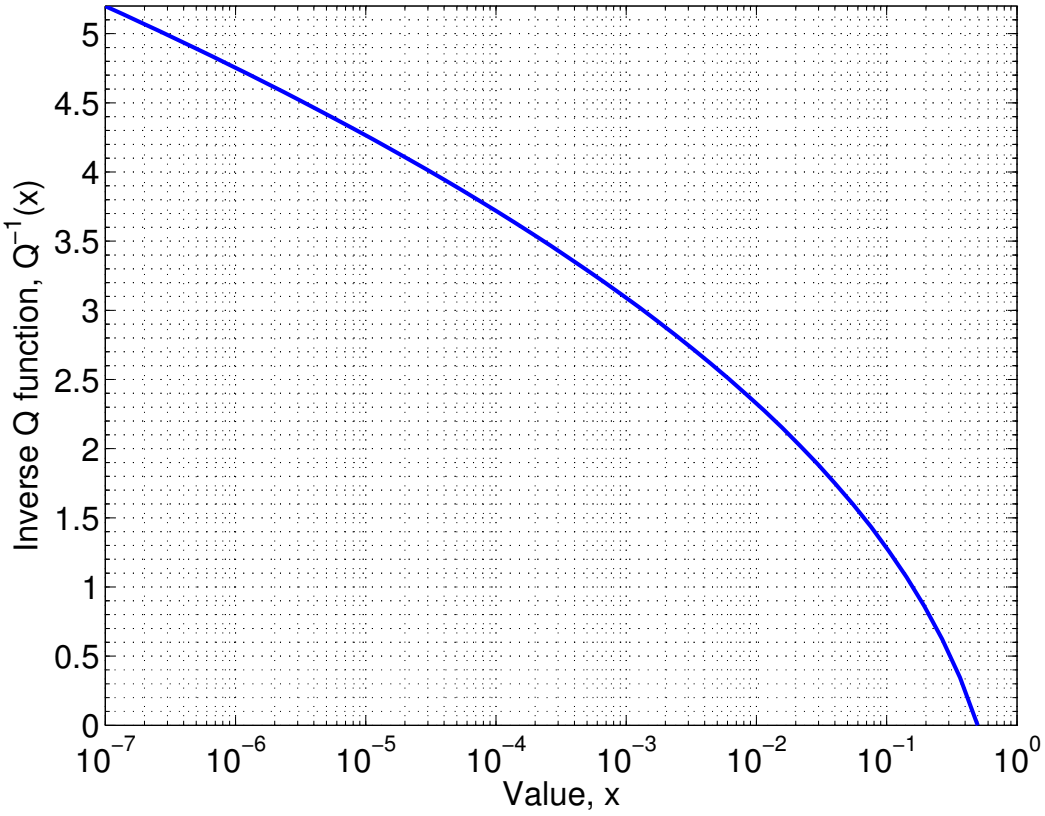
to transfer a 1MB file, then software often reports that the data rate is $1/5 = 0.2$ MBps or 200 kBps. But the bit rate is $8/5$ Mbps or 1.6×10^6 bps. This number is $8\times$ larger so it is the one used by your ISP when selling you your internet service!

Given C/N_0, we can now relate bit error rate, modulation, bit rate, and bandwidth.

Note: We typically use $\mathrm{Q}(\cdot)$ and $\mathrm{Q}^{-1}(\cdot)$ to relate BER and $\frac{\mathcal{E}_b}{N_0}$ in each direction. While you have Matlab, this is easy to calculate. If you can program it into your calculator, great. Otherwise, it's really not a big deal to pull it off of a chart or table. For your convenience, the following tables/plots of $\mathrm{Q}^{-1}(x)$ will appear on Exam 2. I am not picky about getting lots of correct decimal places.

TABLE OF THE $\mathrm{Q}^{-1}(\cdot)$ FUNCTION:

$\mathrm{Q}^{-1}(1\times10^{-6})=4.7534$	$\mathrm{Q}^{-1}(1\times10^{-4})=3.719$	$\mathrm{Q}^{-1}(1\times10^{-2})=2.3263$
$\mathrm{Q}^{-1}(1.5\times10^{-6})=4.6708$	$\mathrm{Q}^{-1}(1.5\times10^{-4})=3.6153$	$\mathrm{Q}^{-1}(1.5\times10^{-2})=2.1701$
$\mathrm{Q}^{-1}(2\times10^{-6})=4.6114$	$\mathrm{Q}^{-1}(2\times10^{-4})=3.5401$	$\mathrm{Q}^{-1}(2\times10^{-2})=2.0537$
$\mathrm{Q}^{-1}(3\times10^{-6})=4.5264$	$\mathrm{Q}^{-1}(3\times10^{-4})=3.4316$	$\mathrm{Q}^{-1}(3\times10^{-2})=1.8808$
$\mathrm{Q}^{-1}(4\times10^{-6})=4.4652$	$\mathrm{Q}^{-1}(4\times10^{-4})=3.3528$	$\mathrm{Q}^{-1}(4\times10^{-2})=1.7507$
$\mathrm{Q}^{-1}(5\times10^{-6})=4.4172$	$\mathrm{Q}^{-1}(5\times10^{-4})=3.2905$	$\mathrm{Q}^{-1}(5\times10^{-2})=1.6449$
$\mathrm{Q}^{-1}(6\times10^{-6})=4.3776$	$\mathrm{Q}^{-1}(6\times10^{-4})=3.2389$	$\mathrm{Q}^{-1}(6\times10^{-2})=1.5548$
$\mathrm{Q}^{-1}(7\times10^{-6})=4.3439$	$\mathrm{Q}^{-1}(7\times10^{-4})=3.1947$	$\mathrm{Q}^{-1}(7\times10^{-2})=1.4758$
$\mathrm{Q}^{-1}(8\times10^{-6})=4.3145$	$\mathrm{Q}^{-1}(8\times10^{-4})=3.1559$	$\mathrm{Q}^{-1}(8\times10^{-2})=1.4051$
$\mathrm{Q}^{-1}(9\times10^{-6})=4.2884$	$\mathrm{Q}^{-1}(9\times10^{-4})=3.1214$	$\mathrm{Q}^{-1}(9\times10^{-2})=1.3408$
$\mathrm{Q}^{-1}(1\times10^{-5})=4.2649$	$\mathrm{Q}^{-1}(1\times10^{-3})=3.0902$	$\mathrm{Q}^{-1}(1\times10^{-1})=1.2816$
$\mathrm{Q}^{-1}(1.5\times10^{-5})=4.1735$	$\mathrm{Q}^{-1}(1.5\times10^{-3})=2.9677$	$\mathrm{Q}^{-1}(1.5\times10^{-1})=1.0364$
$\mathrm{Q}^{-1}(2\times10^{-5})=4.1075$	$\mathrm{Q}^{-1}(2\times10^{-3})=2.8782$	$\mathrm{Q}^{-1}(2\times10^{-1})=0.84162$
$\mathrm{Q}^{-1}(3\times10^{-5})=4.0128$	$\mathrm{Q}^{-1}(3\times10^{-3})=2.7478$	$\mathrm{Q}^{-1}(3\times10^{-1})=0.5244$
$\mathrm{Q}^{-1}(4\times10^{-5})=3.9444$	$\mathrm{Q}^{-1}(4\times10^{-3})=2.6521$	$\mathrm{Q}^{-1}(4\times10^{-1})=0.25335$
$\mathrm{Q}^{-1}(5\times10^{-5})=3.8906$	$\mathrm{Q}^{-1}(5\times10^{-3})=2.5758$	$\mathrm{Q}^{-1}(5\times10^{-1})=0$
$\mathrm{Q}^{-1}(6\times10^{-5})=3.8461$	$\mathrm{Q}^{-1}(6\times10^{-3})=2.5121$	
$\mathrm{Q}^{-1}(7\times10^{-5})=3.8082$	$\mathrm{Q}^{-1}(7\times10^{-3})=2.4573$	
$\mathrm{Q}^{-1}(8\times10^{-5})=3.775$	$\mathrm{Q}^{-1}(8\times10^{-3})=2.4089$	
$\mathrm{Q}^{-1}(9\times10^{-5})=3.7455$	$\mathrm{Q}^{-1}(9\times10^{-3})=2.3656$	



30.2 Examples

Example: Rice 6.36 [11]

Consider a "point-to-point" microwave link. (Such links provide the internet backbone for cellular base stations and internet providers where fiber optic cables can't be installed, for example in rural areas.) Both antenna gains are 20 dB and the transmit antenna power is 10 W. The modulation is 51.84 Mbits/sec 256 square QAM with a carrier frequency of 4 GHz. Atmospheric losses are 2 dB and other incidental losses are 2 dB. A pigeon in the line-of-sight path causes an additional 2 dB loss. The receiver has an equivalent noise temperature of 400 K and an implementation loss of 1 dB. How far away can the two towers be if the bit error rate is not to exceed 10^{-8}? Include the pigeon.

Neal's hint: Use the dB version of Friis formula and subtract these mentioned dB losses: atmospheric losses, incidental losses, implementation loss, and the pigeon.

Solution: Starting with the modulation, $M = 256$ square QAM ($\log_2 M = 8$, $\sqrt{M} = 16$), to achieve $P\,[\text{error}] = 10^{-8}$,

$$\begin{aligned}
10^{-8} &= \frac{4}{\log_2 m}\frac{(\sqrt{M}-1)}{\sqrt{M}}\mathrm{Q}\left(\sqrt{\frac{3\log_2 M}{M-1}\frac{\mathcal{E}_b}{N_0}}\right) \\
10^{-8} &= \frac{4}{8}\frac{15}{16}\mathrm{Q}\left(\sqrt{\frac{3(8)}{255}\frac{\mathcal{E}_b}{N_0}}\right) \\
10^{-8} &= \frac{15}{32}\mathrm{Q}\left(\sqrt{\frac{24}{255}\frac{\mathcal{E}_b}{N_0}}\right). \\
\frac{\mathcal{E}_b}{N_0} &= \frac{255}{24}\left[\mathrm{Q}^{-1}\left(\frac{32}{15}10^{-8}\right)\right]^2 = 319.0
\end{aligned}$$

The noise power $N_0 = kT_{eq} = 1.38 \times 10^{-23}(\text{J/K})400(\text{K}) = 5.52 \times 10^{-21}$ J. So $\mathcal{E}_b = 319.0 \times 5.52 \times 10^{-21} = 1.76 \times 10^{-18}$ J. Since $\mathcal{E}_b = C/R_b$ and the bit rate $R_b = 51.84 \times 10^6$ bits/sec, $C = (51.84 \times 10^6)J(1.76 \times 10^{-18})1/sec = 9.13 \times 10^{-11}$ W, or -100.4 dBW.

Switching to finding an expression for C, the wavelength is $\lambda = 3\times10^8\text{m/s}/4\times10^9 1/\text{s} = 0.075m$, so:

$$\begin{aligned}
C\,[\text{dBW}] &= G_T\,[\text{dB}] + G_R\,[\text{dB}] + P_T\,[\text{dBW}] + 20\log_{10}\frac{\lambda}{4\pi} - 20\log_{10} R - 2\text{dB} - 2\text{dB} - 2\text{dB} - 1\text{dB} \\
&= 20\text{dB} + 20\text{dB} + 10\text{dBW} + 20\log_{10}\frac{0.075m}{4\pi} - 20\log_{10} R - 7dB \\
&= -1.48\text{dBW} - 20\log_{10} R \qquad (83)
\end{aligned}$$

Plugging in $C\,[\text{dBm}] = -100.4$ dBW $= -1.48\text{dBW} - 20\log_{10} R$ and solving for R, we find $R = 88.3$ km. Thus microwave towers should be placed at most 88.3 km (about 55 miles) apart.

Lecture 18

Today: (1) Link Budgeting

- Reading for these notes: Rice [11] Section 6.4.

- Lecture videos for this lecture are at:
  `https://youtube.com/playlist?list=PLQuDEk4rPDOMCwImUNlenSdc2nMTlsMwS`

31 Link Budgeting

The idea of this section is that system designs are not always straightforward solutions of the equations we have. We may have constraints that are difficult to simultaneously meet, for example, on bandwidth and power. To meet all constraints, we may need to be more restrictive than the constraint on one parameter in order to exactly meet the constraint on another.

31.1 Bandwidth and Energy Limited Channels

Assume the C/N_0, the maximum bandwidth, and the maximum BER are all given. Sometimes power is the limiting factor in determining the maximum achievable bit rate. Such links (or channels) are called *power limited* channels. Sometimes bandwidth is the limiting factor in determining the maximum achievable bit rate. In this case, the link (or channel) is called a *bandwidth limited* channel. You just need to try to solve the problem and see which one limits your system.

Here is a step-by-step version of what you might need do in this case:

Method A: Start with power-limited assumption:

1. Use the probability of error constraint to determine the $\frac{\mathcal{E}_b}{N_0}$ constraint, given the appropriate probability of error formula for the modulation.
2. Given the C/N_0 constraint and the $\frac{\mathcal{E}_b}{N_0}$ constraint, find the maximum bit rate. Note that $R_b = 1/T_b = \frac{C/N_0}{\frac{\mathcal{E}_b}{N_0}}$, but be sure to express both in linear units.
3. Given a maximum bit rate, calculate the maximum symbol rate $R_s = \frac{R_b}{\log_2 M}$ and then compute the required bandwidth using the appropriate bandwidth formula.
4. Compare the bandwidth at maximum R_s to the bandwidth constraint: If BW at R_s is too high, then the system is bandwidth limited; reduce your bit rate to conform to the BW constraint. Otherwise, your system is power limited, and your R_b is achievable.

Method B: Start with a bandwidth-limited assumption:

1. Use the bandwidth constraint and the appropriate bandwidth formula to find the maximum symbol rate R_s and then the maximum bit rate R_b.
2. Find the $\frac{\mathcal{E}_b}{N_0}$ at the given bit rate by computing $\frac{\mathcal{E}_b}{N_0} = \frac{C/N_0}{R_b}$. (Again, make sure that everything is in linear units.)
3. Find the probability of error at that $\frac{\mathcal{E}_b}{N_0}$, using the appropriate probability of error formula.
4. If the computed $P\,[\text{error}]$ is greater than the BER constraint, then your system is power limited. Use the previous method to find the maximum bit rate. Otherwise, your system is bandwidth-limited, and you have found the correct maximum bit rate.

Example: Rice 6.33 [11]

Consider a bandpass communications link with a bandwidth of 1.5 MHz and with an available $C/N_0 = 82$ dB Hz. The maximum bit error rate is 10^{-6}.

1. If the modulation is 16-PSK using the SRRC pulse shape with $\alpha = 0.5$, what is the maximum achievable bit rate on the link? Is this a power limited or bandwidth limited channel?

2. If the modulation is square 16-QAM using the SRRC pulse shape with $\alpha = 0.5$, what is the maximum achievable bit rate on this link? Is this a power limited or bandwidth limited channel?

Solution:

1. Try Method A. For $M = 16$ PSK, we can find $\frac{\mathcal{E}_b}{N_0}$ for the maximum BER:

$$\begin{aligned} 10^{-6} &= P\left[\text{error}\right] = \frac{2}{\log_2 M}\mathrm{Q}\left(\sqrt{2(\log_2 M)\sin^2(\pi/M)\frac{\mathcal{E}_b}{N_0}}\right) \\ 10^{-6} &= \frac{2}{4}\mathrm{Q}\left(\sqrt{2(4)\sin^2(\pi/16)\frac{\mathcal{E}_b}{N_0}}\right) \\ \frac{\mathcal{E}_b}{N_0} &= \frac{1}{8\sin^2(\pi/16)}\left[\mathrm{Q}^{-1}\left(2\times 10^{-6}\right)\right]^2 \\ \frac{\mathcal{E}_b}{N_0} &= 69.84 \end{aligned} \tag{84}$$

   Converting C/N_0 to linear, $C/N_0 = 10^{82/10} = 1.585 \times 10^8$. Solving for R_b,

$$R_b = \frac{C/N_0}{\frac{\mathcal{E}_b}{N_0}} = \frac{1.585\times 10^8}{69.84} = 2.27{\times}10^6 = 2.27 \text{ Mbits/s}$$

   and thus $R_s = R_b/\log_2 M = 2.27{\times}10^6/4 = 5.67{\times}10^5$ Msymbols/s. The required bandwidth for this system is

$$B_T = \frac{(1+\alpha)R_b}{\log_2 M} = 1.5(2.27{\times}10^6)/4 = 851 \text{ kHz}$$

   This is clearly lower than the maximum bandwidth of 1.5 MHz. So, the system is power limited, and can operate with bit rate 2.27 Mbits/s. (If B_T had come out > 1.5 MHz, we would have needed to reduce R_b to meet the bandwidth limit.)

2. Try Method A. For $M = 16$ (square) QAM, we can find $\frac{\mathcal{E}_b}{N_0}$ for the maximum BER:

$$\begin{aligned} 10^{-6} &= P\left[\text{error}\right] = \frac{4}{\log_2 M}\frac{(\sqrt{M}-1)}{\sqrt{M}}\mathrm{Q}\left(\sqrt{\frac{3\log_2 M}{M-1}\frac{\mathcal{E}_b}{N_0}}\right) \\ 10^{-6} &= \frac{4}{4}\frac{(4-1)}{4}\mathrm{Q}\left(\sqrt{\frac{3(4)}{15}\frac{\mathcal{E}_b}{N_0}}\right) \\ \frac{\mathcal{E}_b}{N_0} &= \frac{15}{12}\left[\mathrm{Q}^{-1}\left((4/3)\times 10^{-6}\right)\right]^2 \\ \frac{\mathcal{E}_b}{N_0} &= 27.55 \end{aligned} \tag{85}$$

Solving for R_b,

$$R_b = \frac{C/N_0}{\frac{\mathcal{E}_b}{N_0}} = \frac{1.585 \times 10^8}{27.55} = 5.75{\times}10^6 = 5.75 \text{ Mbits/s}$$

The required bandwidth for this bit rate is:

$$B_T = \frac{(1+\alpha)R_b}{\log_2 M} = 1.5(5.75{\times}10^6)/4 = 2.16 \text{ MHz}$$

This is greater than the maximum bandwidth of 1.5 MHz, so we must reduce the bit rate to

$$R_b = \frac{B_T \log_2 M}{1+\alpha} = 1.5 \text{ MHz}\frac{4}{1.5} = 4 \text{ MHz}$$

In summary, we have a bandwidth-limited system with a bit rate of 4 MHz.

31.2 Link Budget Spreadsheet

I've shared a Google sheet that I use to calculate probability of error quickly for many different modulations:

`https://bit.ly/LinkBudgetSheet`

On one sheet, you may enter the TX/RX parameters, starting from the right side of the relationship diagram (Figure 1 in Lecture 16), and calculating the bandwidth and probability of bit error. On the second sheet, you start with the probability of bit error and C/N_0 ratio and it calculates the bit rate and bandwidth.

The sheet is linked to from the schedule table on Canvas. Please save a copy for yourself, modify parameters, and see that the changes go in the direction that you expect. You may use this to check your answer while doing homework, but please do the calculations yourself so that you have practice.

If you're interested in how I made the sheet work, please inspect the code I inserted for the $Q()$ function and $Q^{-1}()$ function by going to Tools:Script Editor; this is handy whenever your function is not a standard spreadsheet function.

Lecture 19

Today: (1) Source Coding & Entropy, (2) Joint / Conditional Entropy, (3) Entropy Rate, (4) Source Coding Theorem

- Reading for these notes: Claude E. Shannon, "A Mathematical Theory of Communication", *The Bell System Technical Journal*, 1948. Read Part I.
- Lecture videos for this lecture are at:
  `https://youtube.com/playlist?list=PLQuDEk4rPDOPI4CVPOm4W53JOGVvQkSk-`

32 Source Coding

We talk in this course quite a bit about "bits", the fundamental measure of the information we're designing a communication system to send. The purpose of today's lecture is to describe what a *bit rate* really means when we know what kind of data will be sent (a.k.a. our "source"). The short story is, when we do know about the data being sent, that the data rate we need to send is *at least* the *entropy rate* of the data, which is a quantity that can be determined from the statistics of the data.

This topic is a semester-long course at the graduate level in itself; but the basic ideas can be presented pretty quickly. Claude Shannon presented an introduction to source coding in Part 1 of his 1948 paper, "A mathematical theory of communication" [13], which introduced the concept.

We have done a lot of counting of bits as our primary measure of communication systems. Our information source is measured in bits, or in bits per second. Modulation schemes' bandwidth efficiency is measured in bits per Hertz, and energy efficiency is energy per bit over noise PSD. Lots of what a communication engineer does is measured in bits!

But how do we measure the bits of a source (*e.g.*, audio, video, email, SMS, ...)? Information can often be represented in many different ways. Images and sound can be encoded in different ways. Text files can be presented in different ways.

Here are two misconceptions:

1. The file size tells you how much information is contained within the file.
2. The number of bits is the $\log_2$ of the number of different values the data could possibly send.

For example, consider a digital black & white image (not grayscale, in this example, truly black or white).

1. You could store it as the value for each pixels. Each pixel has two possibilities (possible values), thus we could encode it in $\log_2 2 = 1$ bit per pixel.
2. You could simply send the coordinates of the pixels of one of the colors (*e.g.*all black pixels).

How many bits would be used in these two representations? What would make you decide which one is more efficient?

From this example, two equivalent representations could require a different number of bits. This is the idea behind *source compression*. For example, .zip or .tar files represent the exact same information that was contained in the original files, but with fewer bits.

What if we had a fixed number of bits to send any image, and we used the sparse B&W image coding scheme (2.) above? Sometimes, the number of bits in the compressed image would exceed what we had allocated. This would introduce errors into the image.

Two types of compression algorithms:

- Lossless: *e.g.*, Zip or compress.
- Lossy: *e.g.*, JPEG, MP3, MP4

Note: Both "zip" and the linux "compress" commands use the Lempel-Ziv algorithm for source compression.

So what is the intrinsic measure of bits of text, an image, audio, or video?

32.1 Entropy

Entropy is a measure of the randomness of a random variable (r.v.). *Randomness* and *information*, in non-technical language, are just two perspectives on the same thing:

- If you are told the value of a r.v. that doesn't vary that much, that telling conveys very little information to you.
- If you are told the value of a very "random" r.v., that telling conveys quite a bit of information to you.

Our technical definition of entropy of a discrete random variable is as follows.

Def'n: *Entropy*
Let X be a discrete random variable with pmf $p_X(x_i) = P[X = x]$. Here, there is a finite or countably infinite set S_X, and $x \in S_X$. We will shorten the notation by using p_i as follows:

$$p_i = p_X(x_i) = P[X = x_i]$$

where $\{x_1, x_2, \ldots\}$ is an ordering of the possible values in S_X. Then the entropy of X, in units of bits, is defined as,

$$H[X] = -\sum_i p_i \log_2 p_i \tag{86}$$

Notes:

- $H[X]$ is an operator on a random variable, not a function of a random variable. It returns a (deterministic) number, not another random variable. This it is like $E[X]$, another operator on a random variable.
- Entropy of a discrete random variable X is calculated using the probability values of the pmf of X, p_i. Nothing else is needed.
- The sum will be from $i = 1 \ldots N$ when $|S_X| = N < \infty$.
- Use that $0 \log 0 = 0$. This is true in the limit of $x \log x$ as $x \to 0^+$.
- All "log" functions are log-base-2 in information theory unless otherwise noted. Keep this in mind when reading a book on information theory. The "reason" the units are bits is because of the base-2 of the log. Actually, when theorists use $\log_e$ or the natural log, they express information in "nats", short for "natural" digits.

Example: Binary r.v.
A binary (Bernoulli) r.v. has pmf,

$$p_X(x) = \begin{cases} s, & x = 1 \\ 1 - s, & x = 0 \\ 0, & o.w. \end{cases}$$

What is the entropy $H[X]$ as a function of s?
Solution: Entropy is given by (86) and is:

$$H[X] = -s \log_2 s - (1 - s) \log_2 (1 - s)$$

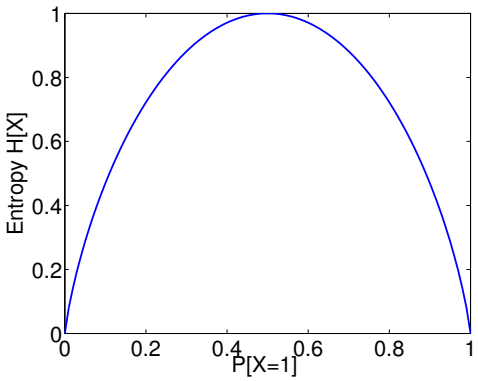


Figure 44: Entropy of a binary r.v.

The solution is plotted in Figure 44.

Example: Non-uniform source with five messages
Some signals are more often close to zero (*e.g.*, audio). Model the r.v. X to have pmf,

$$p_X(x) = \begin{cases} 1/16, & x = 2 \\ 1/4, & x = 1 \\ 1/2, & x = 0 \\ 1/8, & x = -1 \\ 1/16, & x = -2 \\ 0, & o.w. \end{cases}$$

What is its entropy $H[X]$?
Solution:

$$\begin{aligned} H[X] &= \frac{1}{2}\log_2 2 + \frac{1}{4}\log_2 4 + \frac{1}{8}\log_2 8 + 2\frac{1}{16}\log_2 16 \\ &= \frac{15}{8} \text{ bits} \end{aligned} \tag{87}$$

How could you encode X to have an average of 15/8 bits per value of X?
Solution: Generally we want to use fewer bits when the value is more likely:

- "2": encode as 1110
- "1": encode as 10
- "0": encode as 0
- "-1": encode as 110
- "-2": encode as 1111

For example, if we observe 1101110000 we would know the true values would be $-1, 2, 0, 0, 0$. On average the encoding would take

$$\text{bits per } X = \frac{1}{2}1 + \frac{1}{4}2 + \frac{1}{8}3 + 2\frac{1}{16}4 = \frac{15}{8} \text{ bits}$$

Other questions:

1. Do you need to know what the symbol set S_X is?
2. Would multiplying X by 2 change its entropy?
3. Would an arbitrary one-to-one function change the entropy of X?

32.2 Joint Entropy

Def'n: *Joint Entropy*
The joint entropy of two random variables X_1, X_2 with event sets S_{X_1} and S_{X_2} is defined as

$$H[X_1, X_2] = -\sum_{x_1 \in S_{X_1}} \sum_{x_2 \in S_{X_2}} p_{X_1,X_2}(x_1, x_2) \log_2 p_{X_1,X_2}(x_1, x_2) \tag{88}$$

For N joint random variables, $X_1, \ldots, X_N$, entropy is

$$H[X_1, \ldots, X_N] = -\sum_{x_1 \in S_{X_1}} \cdots \sum_{x_N \in S_{X_N}} p_{X_1,\ldots,X_N}(x_1, \ldots, x_N) \log_2 p_{X_1,\ldots,X_N}(x_1, \ldots, x_N)$$

What is the entropy for N i.i.d. random variables? You can show that

$$H[X_1, \ldots, X_N] = -N \sum_{x_1 \in S_{X_1}} p_{X_1}(x_1) \log_2 p_{X_1}(x_1) = NH(X_1)$$

The entropy of N i.i.d. random variables has N times the entropy of any one of them. In addition, the entropy of any N independent (but possibly with different distributions) r.v.s is just the sum of the entropy of each individual r.v.

When r.v.s are not independent, the joint entropy of N r.v.s is less than N times the entropy of one of them. Intuitively, if you know some of them, because of the dependence or correlation, the rest that you don't know become less informative. For example, the B&W image, since pixels are correlated in space, the joint r.v. of several neighboring pixels will have less entropy than the sum of the individual pixel entropies.

32.3 Conditional Entropy

How much additional entropy is in the joint random variables X_1, X_2 compared just to one of them? This is often an important question because it answers the question, "How much additional information do I get from both, compared to just one of them?". We call this difference the conditional entropy, $H[X_2|X_1]$:

$$H[X_2|X_1] = H[X_2, X_1] - H[X_1]. \tag{89}$$

What is an equation for $H[X_2|X_1]$ as a function of the joint probabilities $p_{X_1,X_2}(x_1, x_2)$ and the conditional probabilities $p_{X_2|X_1}(x_2|x_1)$.

Solution: Plugging in (86) for $H[X_2, X_1]$ and $H[X_1]$,

$$\begin{aligned}
H[X_2|X_1] &= -\sum_{x_1\in S_{X_1}}\sum_{x_2\in S_{X_2}} p_{X_1,X_2}(x_1,x_2)\log_2 p_{X_1,X_2}(x_1,x_2) \\
&\quad + \sum_{x_1\in S_{X_1}} p_{X_1}(x_1)\log_2 p_{X_1}(x_1) \\
&= -\sum_{x_1\in S_{X_1}}\sum_{x_2\in S_{X_2}} p_{X_1,X_2}(x_1,x_2)\log_2 p_{X_1,X_2}(x_1,x_2) \\
&\quad + \sum_{x_1\in S_{X_1}}\left[\sum_{x_2\in S_{X_2}} p_{X_1,X_2}(x_1,x_2)\right]\log_2 p_{X_1}(x_1) \\
&= -\sum_{x_1\in S_{X_1}}\sum_{x_2\in S_{X_2}} p_{X_1,X_2}(x_1,x_2)\left(\log_2 p_{X_1,X_2}(x_1,x_2) - \log_2 p_{X_1}(x_1)\right) \\
&= -\sum_{x_1\in S_{X_1}}\sum_{x_2\in S_{X_2}} p_{X_1,X_2}(x_1,x_2)\log_2 \frac{p_{X_1,X_2}(x_1,x_2)}{p_{X_1}(x_1)} \\
&= -\sum_{x_1\in S_{X_1}}\sum_{x_2\in S_{X_2}} p_{X_1,X_2}(x_1,x_2)\log_2 p_{X_2|X_1}(x_2|x_1) \qquad (90)
\end{aligned}$$

Note the asymmetry – there is the joint probability multiplied by the log of the conditional probability. This is not like either the joint or the marginal entropy.

We could also have multi-variate conditional entropy,

$$\begin{aligned}
H[X_N|X_{N-1},\ldots,X_1] &= -\sum_{x_{N-1}\in S_{X_{N-1}}}\cdots\sum_{x_1\in S_{X_1}} p_{X_1,\ldots,X_N}(x_1,x_N) \\
&\qquad \cdot\log_2 p_{X_N|X_{N-1},\ldots,X_1}(x_N|x_{N-1},\ldots,x_1,)
\end{aligned}$$

which is the additional entropy (or information) contained in the Nth random variable, given the values of the $N-1$ previous random variables.

32.4 Entropy Rate

Typically, we're interested in discrete-time random processes, in which we have random variables $X_1, X_2, \ldots$. Since there are infinitely many of them, the joint entropy of all of them may go to infinity as $N \to \infty$. For this case, we are more interested in the rate. How many additional bits, in the limit, are needed for the average r.v. as $N \to \infty$?

Def'n: *Entropy Rate*
The entropy rate of a stationary discrete-time random process, in units of bits per random variable (a.k.a. source output), is defined as

$$H = \lim_{N\to\infty} H[X_N|X_{N-1},\ldots,X_1].$$

It can be shown that entropy rate can equivalently be written as

$$H = \lim_{N\to\infty} \frac{1}{N} H[X_1, X_2, \ldots, X_N].$$

Example: Entropy of English text
Let X_i be the ith letter or space in a common English sentence. What is the sample space S_{X_i}? Is X_i uniform on that space?

What is $H[X_i]$? Solution: I had Matlab (my code is at `https://github.com/npatwari/letter-entropy`) read in the text of Shakespeare's *Romeo and Juliet* [12]. See Figure 45(a). For this pmf, I calculated an entropy of $H = 4.1199$. The Proakis & Salehi book [10] mentions that this value for a single character in general English text is about 4.3.

What is $H[X_i, X_{i+1}]$? Solution: Again, using Matlab on Shakespeare's *Romeo and Juliet*, I calculated the entropy of the joint pmf of each two-letter combination. This gives me the two-dimensional pmf shown in Figure 45(b). I calculate an entropy of 7.46, which is $2 \cdot 3.73$. For the three-letter combinations, the joint entropy was $10.04 = 3 \cdot 3.35$. For four-letter combinations, the joint entropy was $11.98 = 4 \cdot 2.99$.

You can see that the average entropy rate (in bits per letter) is decreasing quickly.

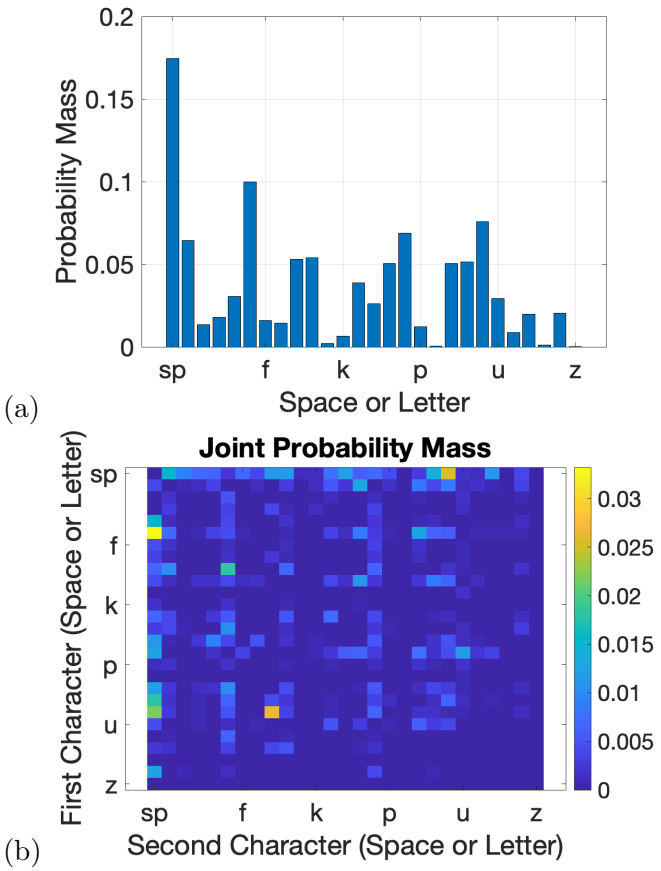


Figure 45: PMF of (a) single letters and (b) two-letter combinations (including spaces) in Shakespeare's *Romeo and Juliet*.

What is the entropy rate, H? Solution: For $N = 10$, we have $H = 1.3$ bits/letter [10, Section 6.2].

32.5 Source Coding Theorem

The key connection between this mathematical definition of entropy and the bit rate that we've been talking about all semester is given by the source coding theorem. It is one of the two fundamental theorems of information theory, and was introduced by Claude Shannon in 1948.

Theorem: A source with entropy rate H can be encoded with arbitrarily small error probability, at any rate R (bits / source output) as long as $R > H$. Conversely, if $R < H$, the error probability will be bounded away from zero, independent of the complexity of the encoder and the decoder employed.
Proof: Proof: Using *typical sequences*. See Shannon's original 1948 paper [13, Part I.9].

Notes:

- Here, an 'error' occurs when your compressed version of the data is not exactly the same as the original. Example: B&W images.
- R has units of bits per source output. A source output for audio, for example, would be one sample. If we had audio at 44,000 samples/second, then $R(44 \times 10^3)$ would give us the source bits/second.
- Theorem fact: Information measure (entropy) times source output (sample) rate gives us a minimum bit rate .
- What is the minimum possible rate to encode English text (if you remove all punctuation and use only lowercase letters)?
- The theorem does not tell us how to do it – just that it can be done if you are allowed infinite latency (N).
- The theorem does not tell us how well source encoding can be done if N is not infinite. That is, for a finite source, the rate may need to be higher.

Lecture 20

Today: (1) Channel Capacity, (2) Error Correction Coding

- Reading for these notes: Todd K. Moon, *Error correction coding: mathematical methods and algorithms*, Wiley-Interscience, 1st ed., 2005, Chapter 3 Sections 3.1-3.4.
- Lecture videos for this lecture are at:
`https://youtube.com/playlist?list=PLQuDEk4rPDONYIs0JuEyYbjkGHrba0R0r`

33 Channel Coding

33.1 Review of Source Coding

In the channel coding lecture, we defined entropy,

$$H[X] = -\sum_i p_i \log_2 p_i$$

and entropy rate,

$$H = \lim_{N\to\infty} \frac{1}{N} H[X_1, X_2, \ldots, X_N].$$

We showed that entropy can be used to quantify information. Given our information source X or $\{X_i\}$, the value of $H[X]$ or H gives us a measure of how many bits we need at a minimum to encode the source data without loss.

The major result was the Shannon's source coding theorem, which says that a source with entropy rate H can be encoded with arbitrarily small error probability, at any rate R (bits / source output) as long as $R > H$. Any lower rate than H would guarantee loss of information.

33.2 When we add noise

Now, we turn to the noisy channel. This discussion of entropy also allows us to consider the maximum data rate which can be carried without error on a bandlimited channel, which is affected by additive uncorrelated Gaussian noise.

Ralph V. L. Hartley (born Nov. 30, 1888) was a researcher for the Western Electric Company, involved in radio telephony, and published a paper in *The Bell System Technical Journal* on "Transmission of Information" [5].

Hartley was particularly influenced by Nyquist's sampling theorem. When transmitting a sequence of rectangular pulses, each of duration T_s, Nyquist determined that the pulse rate was limited to two times the available channel bandwidth B,

$$\frac{1}{T_s} \leq 2B.$$

He was considering digital transmission in pulse-amplitude modulated systems. The pulse rate was limited to $2B$, as described by Nyquist. But, depending on how pulse amplitudes were chosen, each pulse could represent more or less information.

Hartley assumed that the maximum amplitude available to the transmitter was A and the minimum amplitude was 0 (since early receivers were modified AM envelope detectors, and did not deal well with negative amplitudes). Then, Hartley made the assumption that the communication system could discern between pulse amplitudes, if they were at separated by at least a voltage spacing of A_δ. Given that a PAM system operates from 0 to A in increments of A_δ, as shown in Figure 46, the number of different pulse amplitudes (symbols) is

$$M = 1 + \frac{A}{A_\delta}.$$

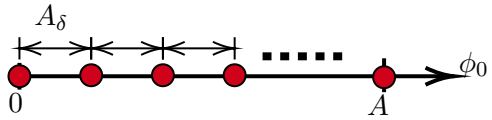


Figure 46: Hartley assigned symbols (•) in a 1-D non-negative PAM system with maximum amplitude A and distance between neighboring symbols of A_δ.

Next, Hartley used the 'bit' measure to quantify the data which could be encoded using M amplitude levels,

$$\log_2 M = \log_2 \left(1 + \frac{A}{A_\delta}\right)$$

Finally, Hartley quantified the data rate using Nyquist's relationship to determine the maximum rate R_{max}, in bits per second, possible from the digital communication system,

$$R_{max} = 2B \log_2 \left(1 + \frac{A}{A_\delta}\right)$$

33.3 C. E. Shannon

What was left unanswered by Hartley's capacity formula was the relationship between noise and the minimum amplitude separation between symbols. Engineers would have to be conservative when setting A_δ to ensure a low probability of error. Furthermore, the capacity formula was for a particular type of PAM system, and did not say anything fundamental about the relationship between capacity and bandwidth for arbitrary modulation.

33.3.1 Noisy Channel

Shannon did take into account an additive uncorrelated Gaussian noise channel, and used statistics to develop a universal bound for capacity, regardless of modulation type [13]. In this channel model, the ith symbol sample at the receiver (after the matched filter, assuming perfect synchronization) is y_i,

$$y_i = x_i + z_i$$

where x_i is the transmitted signal and z_i is the noise in the channel. The noise term z_i is assumed to be i.i.d. Gaussian with variance $E_N = N_0/2$.

33.3.2 Introduction of Latency

Shannon's key insight was to exchange latency (time delay) for reduced probability of error. In fact, his capacity bound considers n-dimensional signaling. So the received vector is $\mathbf{y} = [y_1, \ldots, y_n]$, of length n. These might be truly an n-dimensional signal (*e.g.*, FSK or OFDM), or they might use multiple symbols over time (recall that symbols at delays that are multiples of T_s are orthogonal). In either case, Shannon uses all n dimensions in the constellation – the detector must use all n elements of the $\mathbf{y}$ vector to make a decision. In the multiple symbols over time, this late decision will decide all values of $\mathbf{x} = [x_1, \ldots, x_n]$ simultaneously. Further, Shannon's proof considers the limiting case as $n \to \infty$.

This asymptotic limit as $n \to \infty$ allows for a proof using the statistical convergence of a sequence of random variables. In particular, we need a law called *the law of large numbers*. This law says that the following event,

$$\frac{1}{n} \sum_{i=1}^{n} (y_i - x_i)^2 \leq E_N$$

happens with probability one, as $n \to \infty$. In other words, as $n \to \infty$, the measured value $\mathbf{y}$ will be located within an n-dimensional sphere (hypersphere) of radius $\sqrt{nE_N}$ with center $\mathbf{x}$.

33.3.3 Introduction of Power Limitation

Shannon also formulated the problem as a energy-limited case, in which the *average* symbol energy in the desired signal x_i was limited to E. That is,

$$\frac{1}{n} \sum_{i=1}^{n} x_i^2 \leq E$$

This combination of signal energy limitation and noise energy results in the fact that we can use the same law of large numbers to show that, in probability,

$$\lim_{n\to\infty}\frac{1}{n}\sum_{i=1}^{n}y_i^2=\lim_{n\to\infty}\left[\frac{1}{n}\sum_{i=1}^{n}x_i^2+\frac{1}{n}\sum_{i=1}^{n}z_i^2\right]\leq E+E_N$$

As a result

$$\|\mathbf{y}\|\leq\sqrt{n(E+E_N)}$$

This result says that the vector $\mathbf{y}$, with probability one as $n\to\infty$, is contained within a hypersphere of radius $\sqrt{n(E+E_N)}$ centered at the origin.

33.4 Combining Two Results

The two results, together, show how we many different symbols we could have uniquely distinguished, within a period of n sample times. Hartley asked how many symbol amplitudes could be fit into $[0,A]$ such that they are all separated by A_δ. Shannon's formulation asks us how many multidimensional amplitudes $\mathbf{x}_i$ can be fit into a hypersphere of radius $\sqrt{n(E+E_N)}$ centered at the origin, such that hyperspheres of radius $\sqrt{nE_N}$ do not overlap. This is shown in Figure 47.

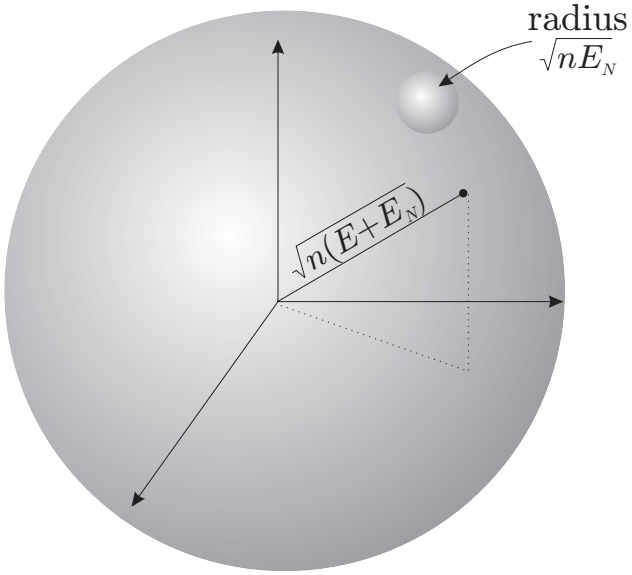


Figure 47: Shannon's capacity formulation simplifies to the geometrical question of: how many hyperspheres of a smaller radius $\sqrt{nE_N}$ fit into a hypersphere of radius $\sqrt{n(E+E_N)}$?

Keep in mind that the number M is the number of different symbols in the constellation diagram, that is, the number of different messages that could have been sent in n pulses.

Again, the problem has reduced to: how many hyperspheres of a smaller radius $\sqrt{nE_N}$ fit into a hypersphere of radius $\sqrt{n(E+E_N)}$? We don't have an exact answer, really – we just find M by dividing the volume of the large hypersphere by the volume of the smaller hypersphere and saying that M *couldn't* be any bigger than that. Using that approach,

$$M\leq\left(1+\frac{E}{E_N}\right)^{n/2} \tag{91}$$

33.4.1 Returning to Hartley

Adjusting Hartley's formula, if we could send M messages now in n pulses (rather than 1 pulse) we would adjust capacity to be:

$$R_{max} = \frac{2B}{n} \log_2 M$$

Using the M from (91) above,

$$R_{max} \leq \frac{2B}{n} \frac{n}{2} \log_2 \left(1 + \frac{E}{E_N}\right) = B \log_2 \left(1 + \frac{E}{E_N}\right)$$

33.4.2 Final Results

Since energy is power multiplied by time, $E = PT_s = \frac{P}{2B}$ where P is the maximum signal power and B is the bandwidth, and $E_N = N_0/2$, we have the Shannon-Hartley Theorem,

$$R_{max} \leq B \log_2 \left(1 + \frac{P}{N_0 B}\right). \tag{92}$$

This result says that **a communication system can operate at bit rate** R_{max} (in a bandlimited channel with width B given power limit E and noise value N_0), **with arbitrarily low probability of error.**

Note that R_{max} is often called C for "capacity", but as we already used C for received power, I'm using R_{max}.

Shannon also proved that any system which operates at a bit rate higher than the capacity, that is, $R_b > R_{max}$, will *certainly* incur a positive bit error rate. Any reliable communication system should thus operate at $R_b < R_{max}$, where R_b is the operating bit rate.

Note that the ratio $\frac{P}{N_0 B}$ is the signal power divided by the noise power, or signal to noise ratio (SNR). Thus the capacity bound is also written $R_{max} \leq B \log_2(1 + SNR)$.

33.5 Efficiency Bound

Another way to write the maximum signal power P is to multiply it by the bit period and use it as the maximum energy per bit, *i.e.*, $\mathcal{E}_b = PT_b$. That is, the energy per bit is the maximum power multiplied by the bit duration. Thus from (92),

$$R_{max} \leq B \log_2 \left(1 + \frac{\mathcal{E}_b/T_b}{N_0 B}\right)$$

or since $R_b = 1/T_b$,

$$R_{max} \leq B \log_2 \left(1 + \frac{R_b}{B} \frac{\mathcal{E}_b}{N_0}\right)$$

Here, R_{max} is just a capacity limit. Be know that our bit rate $R_b \leq R_{max}$, so

$$\frac{R_b}{B} \leq \log_2 \left(1 + \frac{R_b}{B} \frac{\mathcal{E}_b}{N_0}\right)$$

Defining $\eta = \frac{R_b}{B}$ (the spectral efficiency),

$$\eta \leq \log_2 \left(1 + \eta \frac{\mathcal{E}_b}{N_0}\right)$$

This expression can't analytically be solved for η. However, you can look at it as a bound on the bandwidth efficiency as a function of the $\frac{\mathcal{E}_b}{N_0}$ ratio. This relationship is shown in Figure 48. Figure 49 is the plot on a log-y axis with some of the modulation types discussed this semester.

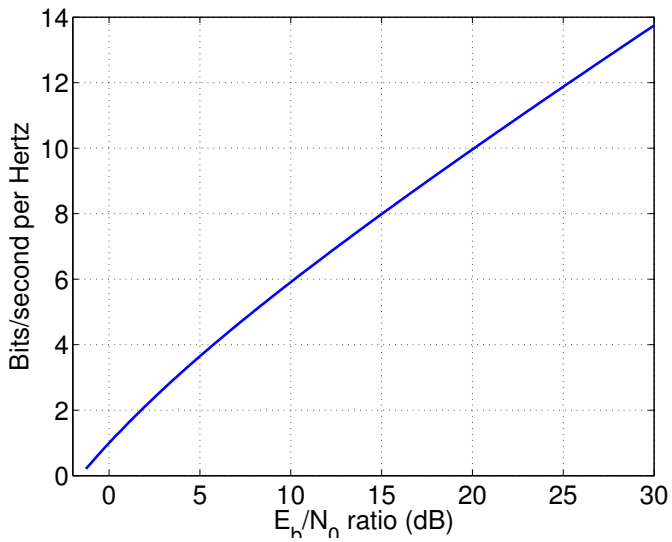


Figure 48: From the Shannon-Hartley theorem, bound on bandwidth efficiency, η.

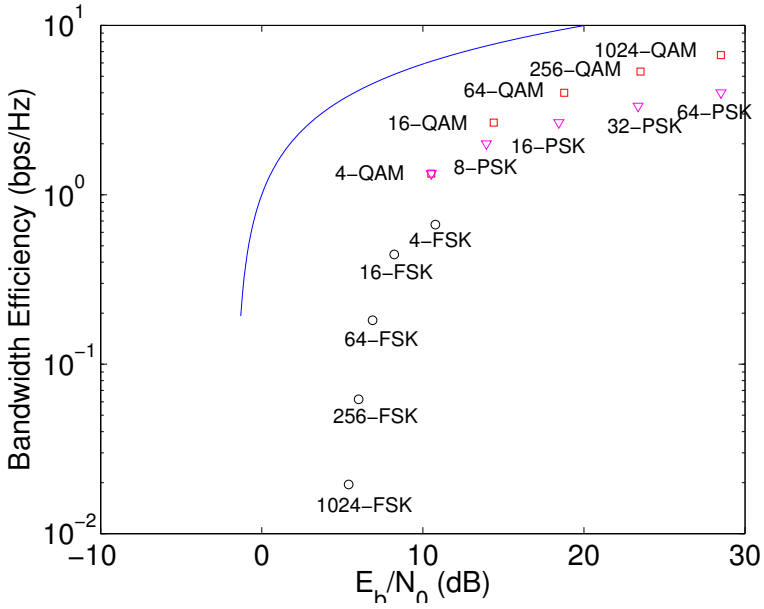


Figure 49: From the Shannon-Hartley theorem bound with achieved bandwidth efficiencies of M-QAM, M-PSK, and M-FSK.

34 Forward Error Correction Coding

As a result of Shannon's capacity theorem, you can see that the modulations we've covered to this point do not get very close to the bandwidth efficiency limit. In fact, the modulations we've covered are within 6-10 dB in $\frac{\mathcal{E}_b}{N_0}$, or a factor of 2-10 in terms of bandwidth efficiency. Communications engineers, over the past 50 years, have addressed this gap using advances in forward error correction coding (FEC). Starting from simple coding schemes which improved provided 1-3 dB of gain in the $\frac{\mathcal{E}_b}{N_0}$, the most recent coding methods (LDPC, turbo codes) can allow one to nearly achieve Shannon's bound.

Additional Resource: You might be interested in Prof. Jeff Frolik's MUSE channel coding video, the source of some of these lecture notes. It is available at:

- `http://www.uvm.edu/~muse/CTA.html`

Def'n: *Forward error correction coding or channel coding*
Adding redundancy to our data at the transmitter with the purpose of detecting and correcting errors at the receiver.

The transmitter takes in data bits and puts out coded bits. Our notation is that for each k data bits input to the FEC operator, the FEC operation will produce $n > k$ coded bits out.

You might complain that FEC appears to do the exact opposite of source coding. While source coding removed redundancy from the source data, FEC adds redundancy. The key is that FEC adds the redundancy in a structured way that enables it to correct errors at the receiver.

34.1 Block vs. Convolutional Coding

Def'n: (k, n) *Block Code*
A (k, n) block code inputs k-bits which are accumulated (via serial-to-parallel conversion) in a k-length vector $\mathbf{d}$. Block encoding multiplies $\mathbf{d}$ by a $k \times n$ generator matrix, G, to output a n-length bit vector $\mathbf{c}$. Block decoding then multiplies the received vector $\mathbf{r}$ by the syndrome matrix S to determine if any errors occurred and determine which (if any) bits were in error out of the n sent.

The syndrome is just a rearrangement of the transpose of the generator matrix, as shown by example below.

In contrast, a convolutional code is a "running" code. For encoding, bits are input into what is effectively a binary filter, the output bits are dependent on the current and past bits.

Compare the advantages and disadvantages:

- Block codes: Advantages: Better for data that is not coming in large streams (bursty data sources, <1000 bits), *e.g.*, wireless sensor networks. Simple linear block codes are not the best in terms of improving efficiency / removing errors. Low density parity check (LDPC) codes are a type of block code that can be used to nearly achieve the Shannon capacity limit (*capacity approaching*), and used today in DVB and 802.11n (WiFi n), and 5G.

- Convolutional codes: Advantages: Best for very large data streams. More energy efficient than block codes when you have large streams of data. Convolutional codes are used in: deep space communication (Voyager program), satellite and terrestrial digital video broadcasting. Disadvantages: Computational complexity increases exponentially in the length of the code. Andrew Viterbi (founder of Qualcomm) is credited with the optimal decoder, called the

Viterbi algorithm. Turbo codes are a type of convolutional code that can be used to nearly achieve the Shannon capacity limit, and used today in cellular (3G, 4G) protocols and in deep-space communications.

34.2 Block Code Implementation

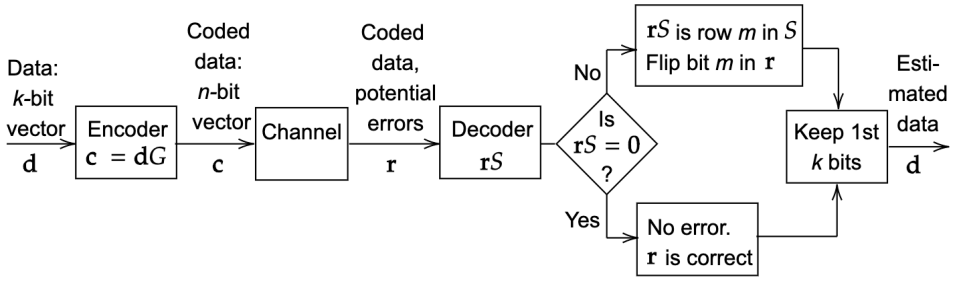


Figure 50: The source groups bits into blocks length k, and inputs them to the encoding block above. The channel may introduce error(s). The decoder multiplies each block with a syndrome, and if the result is a vector of all zeros, there is no error. If not, it looks up the product in the syndrome matrix and finds it in row m, and then flips the mth bit in $\mathbf{r}$. The final $n-k$ bits are dropped and the remaining k are the received decoded bits. All products are modulo-2.

Let the input be denoted $\mathbf{d}$, a k-bit vector. Let the output be $\mathbf{c}$, a n-bit vector. Let G be the generator matrix. Then

$$\mathbf{c} = \mathbf{d}G$$

Thus the G matrix has size $k \times n$. This operation is done modulo-2. That is, multiply all of the pairs, sum them, and then take the mod 2 of the result. That is, if the sum of the products is even, the answer is 0, if the sum is odd, the answer is 1.

Def'n: *Systematic*
The first k bits of the n bits output, are the same as the k bits in $\mathbf{d}$.

Example: (6, 3) systematic block code which can correct one bit error
Let G be given by:

$$G = \begin{bmatrix} 1 & 0 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 1 & 1 & 0 \end{bmatrix}$$

Encode the data bits $\mathbf{d} = [1, 1, 1]$.

Solution: $\mathbf{c} = [1, 1, 1, 0, 0, 0]$

Example: Reception
You receive $\mathbf{r} = [1, 1, 1, 0, 0, 1]$, that is, what you received has an error in the last bit compared to $\mathbf{c}$ (the coded bits that were sent through the channel). What was is the block decoder's estimate of the transmitted data?

Solution: At the receiver, multiply by the syndrome

$$S = \begin{bmatrix} 1 & 0 & 1 \\ 0 & 1 & 1 \\ 1 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Compute: $\mathbf{r}S = [0, 0, 1]$.

Look at all of the rows of the syndrome. The row number of the syndrome S that matches the output $\mathbf{r}S$, is the same as the number of the bit that was in error. If $\mathbf{r}S$ is all zeros, that indicates that there were no errors. Since the sixth bit was in error, instead of $[1, 1, 1, 0, 0, 1]$, we know the correct coded bits were $[1, 1, 1, 0, 0, 0]$.

Finally, because it is a systematic code, we know the first three bits are the data bits. The receiver will just drop the last three bits.

Example: (7, 4) Block Code

1. Encode $\mathbf{d} = [0, 1, 1, 0]$ with the $(7, 4)$ block code with generator,

$$G = \begin{bmatrix} 1 & 0 & 0 & 0 & 1 & 1 & 1 \\ 0 & 1 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 & 0 \end{bmatrix}$$

2. If $\mathbf{r} = [0, 1, 1, 0, 1, 1, 1]$ is received, and S is given as below, what would the receiver determine to be the demodulated bits?

$$S = \begin{bmatrix} 1 & 1 & 1 \\ 0 & 1 & 1 \\ 1 & 0 & 1 \\ 1 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

3. If $\mathbf{r} = [0, 0, 0, 1, 1, 1, 0]$ is received, what would the receiver determine to be the demodulated bits?

4. If $\mathbf{r} = [1, 0, 0, 1, 1, 1, 0]$ is received, what would the receiver determine to be the demodulated bits?

5. If $\mathbf{r} = [1, 1, 0, 1, 1, 1, 0]$ is received, what would the receiver determine to be the demodulated bits?

Solution: (1) I get $\mathbf{c} = [0, 1, 1, 0, 1, 1, 0]$. (2) Then, multiplying $[0, 1, 1, 0, 1, 1, 1]S$, I get $[0, 0, 1]$, which is the same as the 7th row, which says that the last row was incorrectly received, and so the 7th bit was incorrect. Thus the correct four bits sent were $[0, 1, 1, 0]$. (3) I get $\mathbf{r}S = [0, 0, 0]$

which means no bits were received in error, so the four data bits sent were $[0, 0, 0, 1]$. (4) I get $\mathbf{r}S = [1, 1, 1]$ which means that the first bit was received in error, so the four data bits sent were $[0, 0, 0, 1]$. (5) I get $\mathbf{r}S = [1, 0, 0]$ which means that the receiver thinks the fifth bit was received in error, so the receiver would guess the four data bits were $[1, 1, 0, 1]$.

34.3 Performance and Costs

Using a (7,4) block code, we can correct a single error in the seven bits. But we need to increase the number of bits to send, which then requires more energy. So when using channel coding, we reduce the transmit power such that the total energy is identical to transmitting the four uncoded bits. This allows an 'equal-energy' comparison of uncoded and coded transmissions. Still, the probability of bit error goes down for equal $\frac{\mathcal{E}_b}{N_0}$(dB). Equivalently, we can achieve the same bit error rate at 1 dB lower $\frac{\mathcal{E}_b}{N_0}$. This value, 1 dB, is the *coding gain*. In our link budgets, coding goes in the Gains column, added in with the antenna gains.

However, coding requires sending additional bits. So, in a coded system, there is always a ratio of data bits to coded bits, r, called the *code rate*. In the (7,4) block code it is $r =$ 4 data bits /7 coded bits. For a fixed bandwidth, this reduces the achievable data rate by r. For a fixed data rate, it increases the bandwidth by a factor of $1/r$.

34.4 For more information

Looking forward to other material not covered here: A full (graduate) course in error correction coding would teach you more algorithms to use to code and decode block and convolutional codes. Such algorithms and coding methods form the basis for the best codes we have today, such as c codes, polar codes, and turbo codes.

Lecture 21

Today: (1) Multipath Fading (2) Multiple Antennas: RX & TX Diversity, and MIMO

- Reading for these notes: Daniel Halperin, Wenjun Hu, Anmol Sheth, and David Wetherall. "802.11 with multiple antennas for dummies." *ACM SIGCOMM Computer Communication Review* 40, no. 1 (2010): 19-25.
- Lecture videos for this lecture are at:
  `https://youtube.com/playlist?list=PLQuDEk4rPDONxi5JRx8YBlpeCqxdHQcQ2`

35 Multipath Fading

To introduce the topic, I need to discuss multipath fading. Multipath is the phenomenon that multiple waves, each arriving from different directions, arrives at the receiver antenna, and the voltage measured at the receiver is a phasor sum of the complex amplitudes of these waves.

Let's say there are L multipath components, numbered 0 through $L-1$. Component l has amplitude V_l and phase θ_l. Then the total voltage at the receiver antenna V_{TOT} will be:

$$V_{TOT} = \sum_{l=0}^{L-1} V_l e^{j\phi_l} \tag{93}$$

The received power is then proportional to $|V_{TOT}|^2$. Recall that there is a time delay between when a signal is transmitted and when the signal is received, and that time delay is a function of how far the signal travels. We covered the Fourier transform property that a time delay of τ_l in the time domain corresponds to a phase change of

$$-j2\pi f\tau_l. \tag{94}$$

Each multipath travels a different distance, thus its τ_l is different. Plus, multipath component l arrives from a different angle with respect to the receive antenna.

The phases $\{\phi_l\}_l$ of these multipath components change as the antenna takes different positions. The delay τ_l is changing at a different rate while the receiver moves. To be more precise, the delay τ_l changes over time proportional to the receiver's speed, and also proportional to the cosine of the angle θ_l between its angle of arrival and the angle at which the receiver is moving, as drawn in Figure 51. For example in Figure 51 the length of path 0, τ_0, is increasing at a faster rate than the length of path 1, τ_1, because $|\cos(\theta_0)| > |\cos(\theta_1)|$. Because θ_l is different for each multipath, the complex value $e^{j\phi_l}$ rotates around the origin at a different rate (and perhaps in a different direction). The impact on the sum in (93) is that the total voltage is this time-varying sum of complex-valued multipath components, each rotating in phase with a different rate. Please see the lecture video to watch an example of this sum over time. The sum appears random, with a changing magnitude (distance from the origin) and power (squared distance from the origin). If you've seen a "Spirogragh" (the kids art toy), you're familiar with the sum of two vectors with different amplitudes V_l adding with phases changing at different rates.

The point about this to notice is that the power of the received signal varies as a function of the position of the antenna. Sometimes, the complex amplitudes have similar phases and add constructively. Other times, these complex amplitudes are nearly opposite in phase, and when added, are at or near the origin. The effect of multipath fading on the received power ($|V_{TOT}|^2$) is that it varies from a few dB gain to a 30 dB loss as the antenna moves on the order of a quarter of a wavelength. If there are many multipath with nearly equal amplitudes, the central limit theorem implies that the sum will be complex Gaussian with independent real and imaginary amplitudes. In this case, the fading is termed Rayleigh, after the pdf of the magnitude $|V_{TOT}|$ in this case. There is a chance, no matter what fade margin is chosen, that the actual received power will be even lower than that margin below the average received power. Figure 52 shows the probability that the actual received power will be lower than the fade margin below the average received power, as a function of the fade margin in dB. For example, one could set a fade margin of 10 dB, but there is a 9.5% chance that the signal experiences a fade worse than 10 dB; one could set a 30 dB fade margin, and there is a 0.1% chance that the signal experiences a fade worse than 10 dB in a Rayleigh fading multipath channel. This is a very severe problem for mobile communications since link budgets are tight (as you have seen) and there is not generally 30 dB to spare to make the link

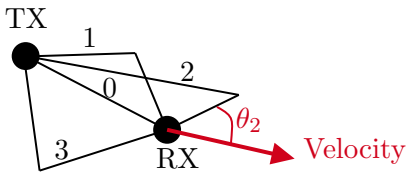


Figure 51: Example channel with $L = 4$ multipath components between the transmitter and receiver. The receiver is moving at an angle of θ_l with respect to multipath component l.

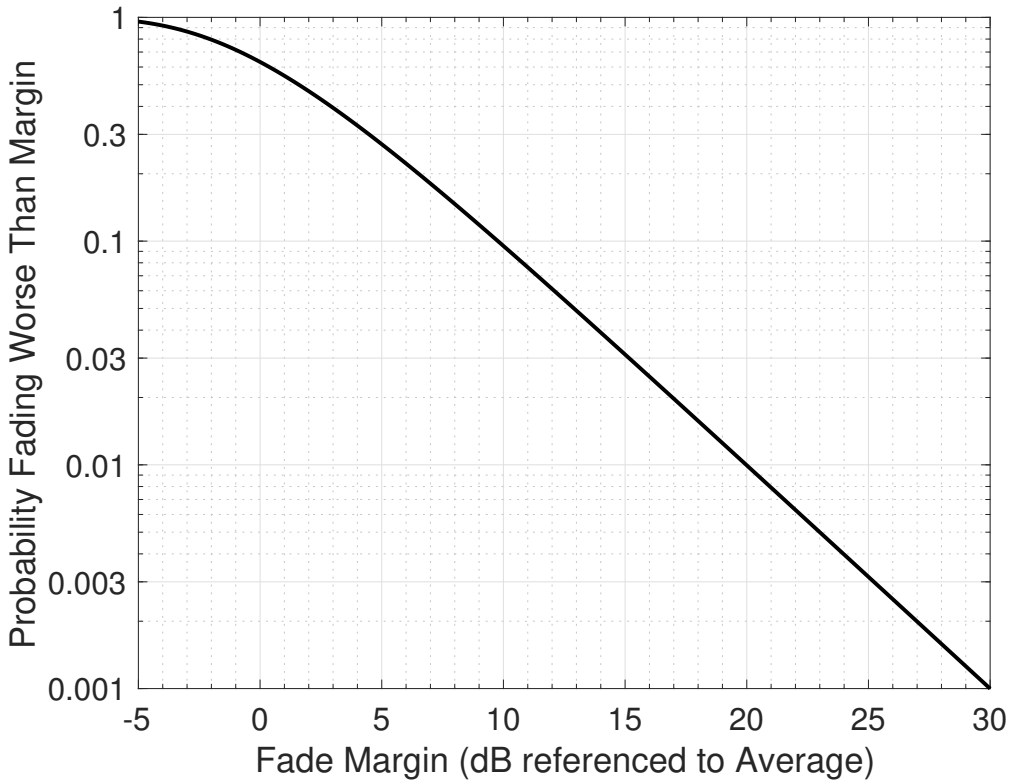


Figure 52: In Rayleigh fading (i.e., when V_{TOT} is complex circular Gaussian), a fade margin in dB is set as a loss in the link budget. This plot shows the probability the received power will be lower than the fade margin.

reliable even when fading is at its worst.

Note also that if the frequency f in (94) changes, it has a similar impact to a moving antenna. That is, it causes different multipath l to change phase at different rates, for the same change in frequency, because τ_l are different for each l. Thus multipath fading is also called *frequency selective*.

Finally note the difference between the antenna "moving" and the antenna "taking different positions". The change in phase comes from the antenna being at different positions, not because it has a non-zero velocity, by itself. Obviously having a speed moves the antenna to different positions over time. But the same differences would be observed by multiple stationary antennas at different positions. Using multiple antennas for one transceiver is a key method to deal with fading, as we discuss in the next section.

36 MIMO

Multiple-input multiple output (MIMO) is a particular type of space and/or polarization diversity in which both the transmitter and receiver may use multiple antennas in order to take advantage of the multipath channel.

The use of multiple antennas has been a standard technique in wireless communication to deal with multipath fading. The idea, up until 20 years ago, was as shown in Figure 53(a), where one

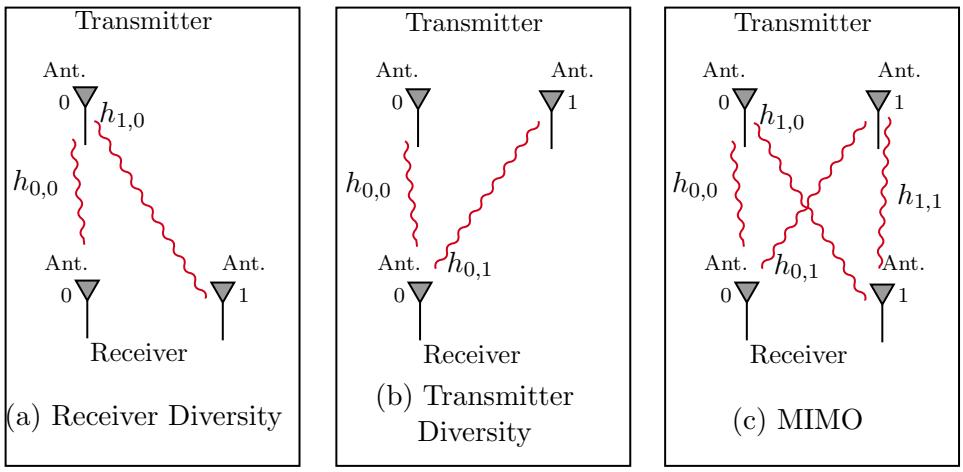


Figure 53: Transmit and receive space diversity schemes: (a) traditional space diversity with receiver combining, called single input multiple output (SIMO); (b) transmit diversity, which may use Alamouti's scheme, called multiple input single output (MISO); (c) 2×2 multiple input multiple output (MIMO).

transmit antenna sent power to multiple receive antennas. The receiver would use *combining* to, for example, pick the receive antenna with the highest power (*selection combining*), or multiply each received voltage with an optimal complex amplitude and add them together (*maximal ratio combining*, or MRC), as introduced in the next section.

The use of multiple antennas is called *space or polarization diversity*. As the two antennas are separated in space (and can be differently polarized), their received powers are different realizations of random variables, and the receiver essentially can benefit from getting multiple "chances" at a good result, just like asking people with diverse opinions can increase your chances that someone gives you a good answer.

36.1 Maximal Ratio Combining

We're going to introduce MIMO by starting with maximal ratio combining (MRC). Let's assume that the transmitter sends symbol (complex) voltage s out of its single antenna. (We're dropping the "(t)" from the signals to simplify the notation.) Assume the channel from TX antenna 0 to RX antenna 0 experiences total channel power gain $|h_{0,0}|^2$ (or voltage gain $h_{0,0}$), and the channel from TX antenna 0 to RX antenna 1 experiences total channel power gain $|h_{1,0}|^2$ (or voltage gain $h_{1,0}$). That is, the two received voltages are

$$\begin{aligned} x_0 &= h_{0,0}s + w_0 \\ x_1 &= h_{1,0}s + w_1 \end{aligned} \tag{95}$$

In MRC, the received signals x_0 and x_1 are multiplied by higher values if the SNR is higher, and lower value, and then summed. Multiplying received signals by a constant doesn't help (it amplifies the noise as much as the signal) so it really matters to multiply them by different numbers. Here, those numbers turn out to be $h_{0,0}^*$ and $h_{1,0}^*$, that is, the complex conjugate of the channel gains. In other words,

$$r_{MRC} = h_{0,0}^* x_0 + h_{1,0}^* x_1$$

That is,

$$r_{MRC} = \left[|h_{0,0}|^2 + |h_{1,0}|^2\right] s + h_{0,0}^* w_0 + h_{1,0}^* w_1$$

In the case when we had only one receive antenna, we would have received either x_1 or x_0. In comparison, the noise terms are multiplied by $h_{0,0}^*$ or $h_{1,0}^*$, but the signal s is multiplied by the sum of $|h_{0,0}|^2 + |h_{1,0}|^2$. If one h_i channel fades, we don't lose the entire signal s.

MRC assumes that the receiver is able to measure the complex channel gains $h_{0,0}$ and $h_{1,0}$. In the receiver diversity case, this could mean comparing the signal amplitude on the two antennas at the same time.

However, MRC requires exactly one transmit antenna, it is not a method to have multiple transmit antennas.

36.2 Alamouti code

MIMO started gaining steam in 1998, from two different results, one from Bell Labs, where they had built an experimental MIMO system they called V-BLAST [15], and a simple transmit diversity scheme from S. M. Alamouti now called the Alamouti scheme [1]. The Alamouti scheme is a simple way to achieve a performance similar to MRC using two transmit antennas, and a single receiver, like the system shown in Figure 53(b). The advantage is that in some cases, the transmitter is more able to have multiple antennas, while the receiver is more limited in size (for example, cellular communications on the downlink).

Alamouti presented a simple transmit diversity scheme that sends two symbols simultaneously, but takes two symbol periods to do so, and over the two transmit antennas. Denote these two symbols s_0 and s_1. The idea is, first transmit s_0 out of antenna 0 and s_1 out of antenna 1. At the receiver, assuming the channels are $h_{0,0}$ and $h_{0,1}$, will be

$$x_0 = s_0 h_{0,0} + s_1 h_{0,1} \tag{96}$$

Then, during the subsequent symbol period, send $-s_1^*$ out of antenna 0 and s_0^* out of antenna 1, where the superscript $*$ is used to denote complex conjugate. During the second symbol period the receiver will see

$$x_1 = -s_1^* h_{0,0} + s_0^* h_{0,1} \tag{97}$$

Note this assumes the channel gains were the same during the second symbol period as during the first.

The "magic" happens when we combine x_0 and x_1 in the following way to come up with estimates of s_0 and s_1. We form:

$$\begin{aligned} \tilde{s}_0 &= h_{0,0}^* x_0 + h_{0,1} x_1^* \\ \tilde{s}_1 &= h_{0,1}^* x_0 - h_{0,0} x_1^* \end{aligned}$$

Plugging in for x_0 and x_1 as given in (96) and (97), respectively,

$$\begin{aligned} \tilde{s}_0 &= h_{0,0}^*(s_0 h_{0,0} + s_1 h_{0,1}) + h_{0,1}(-s_1 h_{0,0}^* + s_0 h_{0,1}^*) \\ \tilde{s}_1 &= h_{0,1}^*(s_0 h_{0,0} + s_1 h_{0,1}) - h_{0,0}(-s_1 h_{0,0}^* + s_0 h_{0,1}^*) \end{aligned}$$

Simplifying,

$$\begin{aligned} \tilde{s}_0 &= |h_{0,0}|^2 s_0 + s_1 h_{0,0}^* h_{0,1} - s_1 h_{0,0}^* h_{0,1} + |h_{0,1}|^2 s_0 \\ \tilde{s}_1 &= |h_{0,1}|^2 s_1 + s_0 h_{0,0} h_{0,1}^* - s_0 h_{0,0} h_{0,1}^* + |h_{0,0}|^2 s_1 \end{aligned}$$

The middle terms cancel out in each case, so finally,

$$\begin{aligned}\tilde{s}_0 &= (|h_{0,0}|^2 + |h_{0,1}|^2)s_0 \\ \tilde{s}_1 &= (|h_{0,1}|^2 + |h_{0,0}|^2)s_1\end{aligned}$$

In short, in two symbol periods, we've managed to convey two symbols of information. Each symbol arrives with approximately the same signal amplitude that we would have had in the receiver antenna diversity MRC case.

Notes:

1. This is a two-by one code, that is, it works for two transmit antennas and one receive antenna. This code has been generalized for $N_r \times N_t$ MIMO systems, and called "space-time block codes", by Tarokh et. al. [14]. These can send more symbols in less time – in k symbol periods, you can send more than k symbols.

2. If you transmit out of two antennas, you would in general need twice as much power as the receiver diversity case, which had one transmit antenna. If instead we compare the two when using the same total transmit power, *i.e.*, cut the power in half to each of two antennas in the transmitter diversity case. The performance is thus 3 dB worse than the receiver MRC diversity case when constrained by total transmit power.

3. The Alamouti and space-time block codes are not optimal. Space-time coding is the name of the general area of encoding information the multiple channels. One better-performing scheme is called space-time trellis coding. But the decoding complexity of space-time trellis codes increases exponentially as a function of the spectral efficiency [6, p377] [14].

36.3 MIMO Channel Representation

In general for MIMO, we have multiple (N_t) transmitters and multiple (N_r) receivers. We refer to the system as a (N_r, N_t) or $N_r \times N_t$ MIMO system. Figure 53(c) shows the channels for a $(2, 2)$ MIMO system. For the channel between transmitter k and receiver i, we denote the "channel voltage gain" as $h_{i,k}$. This gain is a complex number, with real and imaginary parts. Recall that the phase of a multipath component changes with distance, frequency, due to reflections, etc. The channel power gain would be $|h_{i,k}|^2$. The received voltage signal at i, just from transmitter k, is $s_k h_{i,k}$, where s_k is what was transmitted from antenna k.

To keep all these numbers organized, we use vectors and matrices. The transmitted signal from antennas $1, \ldots, N_t$ is denoted $\mathbf{s}$,

$$\mathbf{s} = [s_1, \ldots, s_{N_t}]^T$$

and the channel gain matrix H is given as

$$H = \begin{bmatrix} h_{1,1} & h_{1,2} & \cdots & h_{1,N_t} \\ h_{2,1} & h_{2,2} & \cdots & h_{2,N_t} \\ \vdots & \vdots & \ddots & \vdots \\ h_{N_r,1} & h_{N_r,2} & \cdots & h_{N_r,N_t} \end{bmatrix} \tag{98}$$

Where there are N_r rows each corresponding to the channels measured at each receiver; and N_t columns each corresponding to the channels from each transmitter.

The received signal at receiver i is a linear combination of the s_k for $k = 1, \ldots, N_t$ terms plus noise:

$$x_i = \sum_{k=1}^{N_t} h_{i,k} s_k + w_i$$

where w_i the additive noise term, and $i = 1, \ldots, N_r$. In matrix form, we can rewrite this as:

$$\mathbf{x} = H\mathbf{s} + \mathbf{w}$$

where $\mathbf{x} = [x_1, \ldots, x_{N_r}]^T$ is the received vector and $\mathbf{w} = [w_1, \ldots, w_{N_r}]^T$ is the noise vector.

36.4 Direct-Mapped MIMO

In Section 4.1 of your reading [4], the authors describe the direct-mapped MIMO protocol in 802.11n involving a receiver estimating the channel $\hat{H}$ using some training fields (known signals) at the start of the packet. Assuming that $N_t = N_r$, the estimated channel is then inverted, $\hat{H}^{-1}$ and then multiplied by $\mathbf{x}$ to get an estimate of the transmitted signals $\mathbf{s}$. That is,

$$\hat{H}^{-1}\mathbf{x} = \hat{H}^{-1}H\mathbf{s} + \hat{H}^{-1}\mathbf{w} \approx \mathbf{s} + \hat{H}^{-1}\mathbf{w} \tag{99}$$

Assuming that the channel H is estimated well enough, and it is invertible, then $\hat{H}^{-1}H \approx I$ and the latter approximation holds.

For example let's consider the 2x2 MIMO case. In this case, the inverse of H is

$$H^{-1} = \frac{1}{h_{1,1}h_{2,2} - h_{1,2}h_{2,1}} \begin{bmatrix} h_{2,2} & -h_{1,2} \\ -h_{2,1} & h_{1,1} \end{bmatrix}. \tag{100}$$

When this matrix inverse is multiplied by the noise $\mathbf{w}$, we can get into trouble whenever the determinant of H in the denominator, $(h_{1,1}h_{2,2} - h_{1,2}h_{2,1})$, has magnitude very close to zero, because it will then magnify the noise considerably. Note that the determinant of H is the product of the eigenvalues, and any eigenvalue of 0 indicates rank deficiency and that the matrix cannot be inverted. However, (99) is called the *zero-forcing* solution for a MIMO receiver; simply estimate the channel, invert it, and multiply that inverse matrix by the received signal. Doing so gives an estimate of the two streams originally sent.

In general for $N_t = N_r \geq 2$ MIMO, we can have trouble with the zero-forcing solution when the channel matrix is rank-deficient, i.e., one of its eigenvalues is very close to zero. An alternate method called the minimum mean squared error (MMSE) solution is also presented in [4] to address this case.

An even higher data rate method is called *precoded MIMO*, in which the transmitter knows the channel matrix H and uses it to send data not on the the N_t antennas, but on the $\min(N_t, N_r)$ orthogonal eigenvectors of H. These eigenvector channels are the linear combinations of the N_t antennas that result in the receiver measuring independently transmitted signal streams on each of its N_r antennas. Refer to [4] for more details. The problem with precoded MIMO is that the transmitter must know H, which then would either require the receiver to send its measurement of H to the transmitter; or to operate both directions of a link on the same frequency channel (e.g., via time division duplexing) so that the transmitter can measure H when the channel is used in the reverse direction.

36.5 Capacity of MIMO Systems

We said in the lecture on Shannon channel capacity that there is a theoretical limit to the bps per Hz we can achieve on a channel. Using multiple antennas at the TX and RX increases this theoretical limit. We said that the limit on bandwidth efficiency is given as,

$$\frac{R_{max}}{B} = \log_2(1+\rho) \tag{101}$$

where R_{max} is the maximum possible bit rate which can be achieved on the channel for given signal to noise ratio ρ and bandwidth B.

In a $N_t \times N_r$ MIMO system with channel matrix H as given in (98), with $N_t \geq N_r$, the new Shannon limit on bps per Hz is [6],

$$\frac{R_{max}}{B} = E\left[\log_2\left\{\det\left(\mathrm{I}_{N_r} + \rho\frac{1}{N_t}HH^\dagger\right)\right\}\right] \tag{102}$$

where $H^\dagger$ is the conjugate transpose of H (I'm copying the notation of the Haykin Moher book), and ρ is the average signal to noise ratio. Here, we assume that each channel is Rayleigh, that is each channel voltage gain $h_{i,k}$ is complex Gaussian, and all channel gains are independent from each other. This is why we need an expected value – the matrix H is filled with random variables.

To get more intuition about the bandwidth efficiency limit, consider that the term $HH^\dagger$ is a Hermitian $N_r \times N_r$ matrix with eigendecomposition $HH^\dagger = U\Lambda U^\dagger$ where U is the matrix of eigenvectors of $HH^\dagger$ and Λ is a diagonal matrix of eigenvalues λ_i for $i = 1, \ldots, N_r$. In this case, we can rewrite (102) as,

$$\frac{R_{max}}{B} = \sum_{i=1}^{N_r} E\left[\log_2\left(1+\rho\frac{\lambda_i}{N_t}\right)\right] \tag{103}$$

Compared to (101), Equation (103) is a sum of several Shannon capacities – each with effective SNR $\rho\frac{\lambda_i}{N_t}$. Recall this was the formula for $N_t \geq N_r$. For $N_r \geq N_t$, the formula is

$$\frac{R_{max}}{B} = \sum_{i=1}^{N_t} E\left[\log_2\left(1+\rho\frac{\lambda_i}{N_r}\right)\right] \tag{104}$$

These $\min(N_t, N_r)$ "channels" are called the "eigen-channels" of a MIMO system.

In summary, we have created $\min(N_t, N_r)$ eigen-channels. Results have shown that the total capacity increases approximately with $\min(N_t, N_r)$. MIMO is so attractive for current and future communication systems because it multiplies the achievable bit rate by this factor of $\min(N_t, N_r)$, without requiring additional bandwidth or signal energy.

Lecture 22

Today: (1) Wi-Fi Protocols, (2) Symbol Synchronization, (3) Interpolation Filtering

- Reading for these notes: Rice [11] Sections 8.4 and 8.5.
- There are no lecture videos for this class session.

37 Wi-Fi Protocols as Examples

As an example wireless protocol we will consider the 802.11 protocols advertised as *Wi-Fi*. These are standardized by the IEEE, within its networking standards committee (802) for wireless local area networking (subcommittee #11). The protocols are all time-division duplex, operating on the same frequency channel for both directions between two devices (commonly referred to as uplink, or to the access point, and downlink, or to the mobile device), but not at the same time.

37.1 DS-SS

The first (popular) protocol was called 802.11b, meant to operate in the 2.4 GHz ISM band in the US (2.400 to 2.483 GHz). Prior to this protocol, most wireless local area networks (WLANs) operated below 1 GHz, for example, in the 902-928 MHz band. The higher frequency band was a poor choice first because of its higher attenuation through walls, relatively higher cost of transceivers, and the microwave oven problem: microwave ovens had been allocated this same band. Microwave ovens were thought to impose a nearly insurmountable robustness challenge, and the FCC couldn't sell the band to any paying customer, so they essentially were forced to "give it away" for free as a very large unregulated band. To deal with the interference produced by microwave ovens, and other interfering ISM band transmissions, they specified that devices would have to use "spread spectrum" modulation methods. Essentially, a transmitter must spread its signal across a very wide bandwidth in order to "average out" the effects of narrowband interfering signals.

IEEE 802.11b met this requirement in a couple of different ways. The first way is called *direct sequence spread spectrum* (DS-SS). In DS-SS, the pulse shape is artificially set to have a high bandwidth. The 802.11b standard used what is called a *Barker code* as its pulse shape. The pulse shape $p(t)$ for the Barker code pulse shape is shown in Fig. 54 [9]. The Barker code pulse shape is essentially the function you'd get by modulating (via binary bipolar PAM) the bits 10110111000 using a SRRC pulse. These eleven "fake bits" used to generate the Barker code are called *chips*. The Barker code pulse shape is wide in bandwidth; the symbol rate is 1 MSymbol/sec and the bandwidth of the signal is 11 times this because it is just like the bandwidth of a 11 Msymbol/sec BPSK signal with a SRRC pulse shape (with a large α).

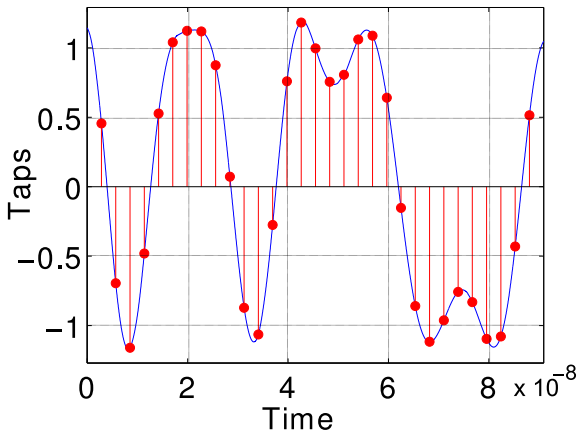


Figure 54: Samples of the Barker code used in IEEE 802.11b [9].

The advantage of the Barker code is that it is nearly orthogonal to itself at non-zero delays. If a correlator is running, it will see a sharp correlation peak when it is lined up perfectly with

the incoming Barker code, and will be nearly zero when it is not. This can help with symbol synchronization.

The original two modulations for 802.11b are differential BPSK or differential QPSK. We covered D-BPSK, but not D-QPSK. D-QPSK is $M = 4$ thus 2 bits per symbol, for a total of 2 Mbps, while D-BPSK has 1 Mbps. If the link quality was high enough, the link would use D-QPSK.

37.2 OFDM

The next major advancement was for 802.11g. While 802.11a technically was released about the same time as 802.11b, the "a" standard specified using the 5.8 GHz band. The combination of the computational costs of OFDM and the higher costs of 5.8 GHz components made for little commercial use initially. IEEE 802.11g used the 2.4 GHz band for the same OFDM modulation, and thus was less expensive than "a". OFDM, as we've described, uses many, many orthogonal sine and cosine waveforms, each at a different frequency. Each sine and cosine pair at one frequency is called a subcarrier, and there are 52 subcarriers. Four are sent un-modulated, that is with no data, just the sine or cosine wave itself, so that the receiver can perform synchronization.

The other 48 subcarriers in the OFDM signal are modulated with one of: BPSK, QPSK, 16 square QAM, or 64 square QAM. For each there is a choice of two convolutional codes, the first which sends out two coded bits for each information bit input, that is, a rate 1/2 code; or 2) a code that sends out four coded bits for each three information bits input, that is, a rate 3/4 code. You can see that the rate 3/4 code is more efficient, but it also corrects fewer errors. In all, there are 8 modulation/coding schemes (MCS). These are chosen adaptively, and communicated between devices so that the receiver knows what detector to use on each subcarrier.

The symbol rate in 802.11g is 4 μs. However, 0.8 μs of that symbol duration does not provide information; it simply repeats the final 0.8 μs at the beginning of the symbol. It is called the *cyclic prefix* and is used to make the symbol appear periodic. Because of this periodicity, the IFFT and FFT can be successfully used to generate the temporal signal to transmit, and to separate the 52 subcarriers at the receiver. Thus 3.2 μs of the symbol contains information. However, each symbol contains 48 subcarriers, each with up to 6 bits (when using $M = 64$ QAM, for a total of 288 bits per symbol. The maximum bit rate for 802.11g is this 288 coded bits per 4 μs, or at most 288(3/4) information bits per 4 μs, or 54 Mbps.

37.3 MIMO

The next big development was the addition of multiple input multiple output (MIMO) methods to the standard. IEEE 802.11n was the first major standardization of MIMO technology, which had just really been invented 9-10 years prior [1]. By using N_t antennas at the transmitter, and N_r at the receiver, the idea of MIMO is to achieve $N = \min(N_t, N_r)$ "separate" spatial streams. The 802.11n standard specified a maximum of $N = 4$ at both end of the link, that is, $N_t = N_r = 4$ is the maximum. The capacity is, theoretically, able to be multiplied by 4 at most.

The next generations of Wi-Fi use more antennas (up to 8 in 802.11ax), higher bandwidths (80 MHz), higher M (up to 1024-QAM in 802.11ax) and mm-wave bands (60 GHz bands in 802.11ay). The 60 GHz band is also an unlicensed band, and considerable bandwidth is available. The band has been avoided for long range applications because it has about 10 dB loss / km due to oxygen (O_2) absorption. There is 14 GHz of bandwidth available in this band. The 802.11ay standard allows allocating 8 GHz for a single link. One can achieve very high data rate with such a large bandwidth, even with modulations with low bandwidth efficiency.

37.4 Shortcomings

One of the problems with Wi-Fi is range. There is no rate lower than 1 Mbps, thus it is designed for relatively short range links. Recall that, for a given probability of bit error, you need a certain $\frac{\mathcal{E}_b}{N_0}$. The noise power spectral density N_0 is fixed; and $\mathcal{E}_b$ is the received power times the bit period, $T_b P_r$. If you want to extend the range, the P_r will go down, and you will be forced to increase the T_b, or equivalently, reduce the bit rate R_b. Thus there is a tradeoff between range and data rate. For many Internet-of-things (IoT) applications, the required data rate is very low compared to 1 Mbps. For example, an air quality sensor might need to send a 50 bytes of air quality data per hour, or about 0.1 bps, seven orders of magnitude less than Wi-Fi's minimum rate.

My lab has done some work to address the range limitation of Wi-Fi. We created a protocol that one can run on top of Wi-Fi with a standard Wi-Fi transceiver, with a change in software, that enables a new very low rate but much longer range data communications. It's called "on-off noise power communication" (ONPC), and effectively uses regular Wi-Fi packets as a means to increase the interference in the channel [8]. Even when those packets can't be demodulated because the transmitter is too far away, they increase the noise+interference power measured by a receiver. Our protocol transmits and receives temporal patterns in the Wi-Fi noise level and uses them to send bits.

Another problem with Wi-Fi for IoT is that the transceiver is relatively high in power consumption. OFDM requires a linear amplifier, and relatively high transmit power. Thus a transmission requires considerable power consumption. Further computational complexity, multiple antennas, and the fact that the receiver spends considerable time constantly listening to the channel, means that the receiver consumes considerable energy as well. For example, a 802.11n receiver would consume more than 1 W while in idle receive mode [3]. Prior to 802.11ah and ax, receivers were not permitted long sleep periods. The addition of the *target wake time* (TWT) in later protocol versions provides a means for Wi-Fi endpoints (devices besides the access point) to sleep for a designated period of time. Battery-powered low rate devices may prefer a modulation capable of more energy efficient operation.

38 Time Synchronization

The short introduction to this section is that so far in the class, we have assumed that, after the matched filter, you can simply downsample, and one of the samples will be at the correct symbol time, that is, a multiple of T_s plus the time the first symbol starts arriving at the receiver. This is generally not true, and today's digital receivers operate in an *sample first, ask questions later* kind of manner. That is, there is no analog synchronization loop that makes sure that one of the samples each symbol is at exactly the correct time. Instead, a receiver does the determination in digital – that is, it calculates what time the symbol should be sampled for best performance. The receiver has a sample each multiple of T, the sampling period, and this calculated optimal time is not generally one of these exact sampling times. So, the receiver then interpolates to calculate what the sample value would have been at that calculated optimal time. This interpolation is possible because of Nyquist's sampling theorem. It is made computationally simple using what are called Farrow filters. This lecture is on how to perform this computationally efficient interpolation. I've found these Farrow filters to be particularly useful in my research, and I hope to show that they're not particularly complicated.

38.1 Channel Delay Model

A radio channel adds a delay from the transmitter to the receiver, due to the fact that the two are separated in space and there is a finite speed of light (one foot per nanosecond). In addition to that delay, the transmitter and receiver typically use asynchronous clocks that have some delay with respect to each other.

At the transmitter, we send a new symbol at each multiple delay of T_s, the symbol period. If we denote $a_m(k)$ as the amplitude of waveform m in the kth symbol that is transmitted, then

$$s(t) = \sum_k \sum_m a_m(k)\phi_m(t - kT_s), \tag{105}$$

where $\{\phi_m(t)\}$ are our orthogonal waveforms, and at each time kT_s we pick a new symbol to send.

At the receiver, the transmitted signal arrives with a (generally unknown) delay τ, and can be written (assuming carrier synchronization) as:

$$x(t) = \sum_k \sum_m a_m(k)\phi_m(t - kT_s - \tau). \tag{106}$$

As a start, let's compare receiver implementations for a continuous-time and a discrete-time receiver. Figure 55 has an analog timing synchronization loop which controls the sampler (the ADC). The symbol timing PLL in Figure 55 gets data from the signals $x(t)$ and $y(t)$ in that figure and uses it to determine, over several samples, what a good time delay τ is.

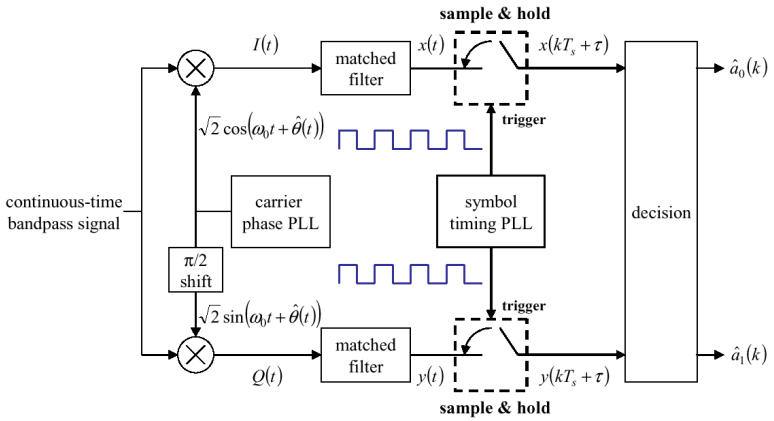


Figure 55: Block diagram for a continuous-time receiver, including analog timing synchronization (from: [11] Figure 8.3.1).

The input signal is downconverted and then run through matched filters, which correlate the signal with $\phi_n(t - t_k)$ for each n, and for some delay t_k. For the correlation with n,

$$\begin{aligned} r_n(k) &= \langle x(t), \phi_n(t)\rangle \\ r_n(k) &= \sum_k \sum_m \mathbf{s}_m(k)\langle \phi_n(t - t_k), \phi_n(t - kT_s - \tau)\rangle \end{aligned} \tag{107}$$

Note that if $t_k = kT_s + \tau$, then the correlation $\langle \phi_n(t - t_k), \phi_n(t - kT_s - \tau)\rangle$ is highest and closest to 1. This t_k is the correct timing delay at each correlator for the kth symbol. But, these are generally unknown to the receiver until timing synchronization is complete.

Figure 56 shows a receiver with an ADC immediately after the downconverter. Here, note the ADC has nothing controlling it. Instead, after the matched filter, an interpolator corrects the sampling time problems using discrete-time processing. This interpolation is the subject of this section.

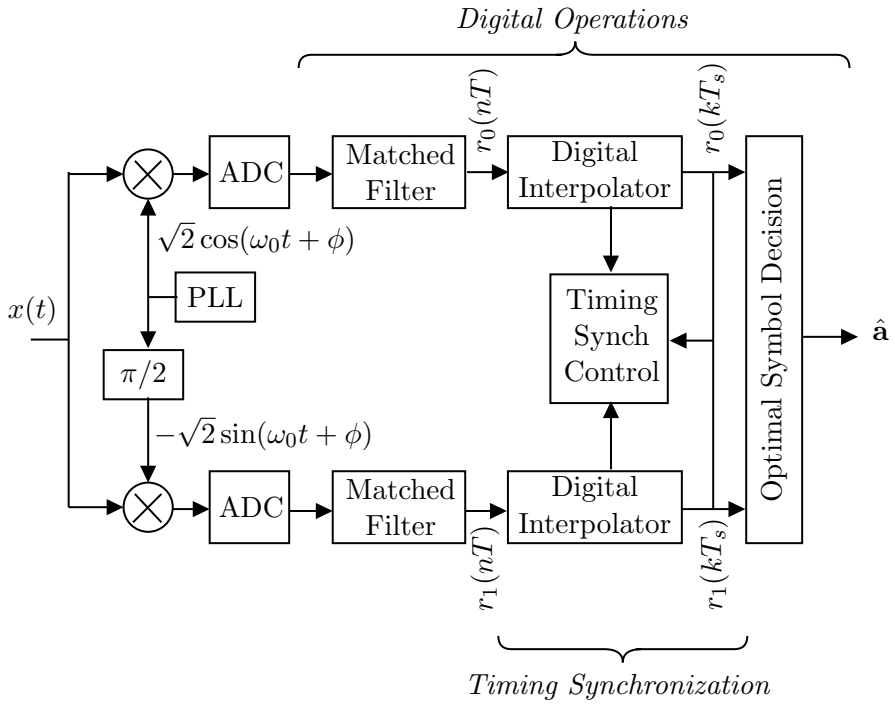


Figure 56: Block diagram for a digital receiver for QAM/PSK, including discrete-time timing synchronization.

Implementations are increasingly digital. A 'software radio' follows the idea that as much of the radio is done in digital, after the signal has been sampled. The idea is to "bring the ADC to the antenna" for purposes of reconfigurability, and reducing part counts and costs.

Another implementation is like Figure 56 but instead of the interpolator, the timing synch control block is fed back to the ADC. But again, this requires a DAC and feedback to the analog part of the receiver, which is not preferred. Also, because of the processing delay, this digital and analog feedback loop can be problematic.

First, we'll talk about interpolation, and then, we'll consider the control loop.

39 Interpolation

In this lecture, we emphasize digital timing synchronization using an interpolation filter. For example, consider Figure 57. In this figure, a BPSK receiver samples the matched filter output at a rate of twice per symbol, unsynchronized with respect to the symbol clock, resulting in samples $r(nT)$.

Some example sampling clocks, compared to the actual symbol clock, are shown in Figure 58. These are shown in degrees of severity of correction for the receiver. When we say 'synchronized

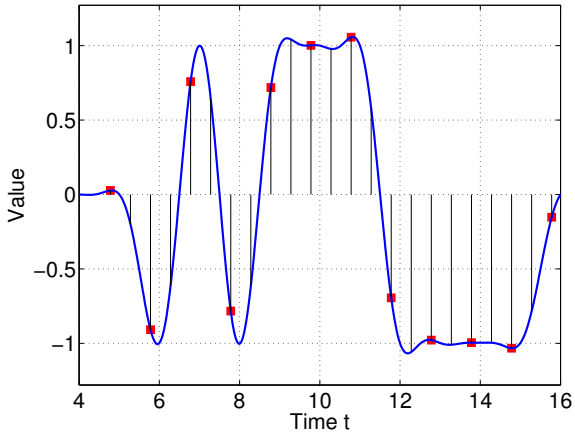


Figure 57: Samples of the matched filter output (BPSK, RRC with $\alpha = 1$) taken at twice the correct symbol rate (vertical lines), but with a timing error. If down-sampling (by 2) results in the symbol samples $r_0(k)$ given by red squares, then sampling sometimes reduces the magnitude of the desired signal.

in rate', we mean within an integer multiple, since the sampling clock must operate at (at least) double the symbol rate.

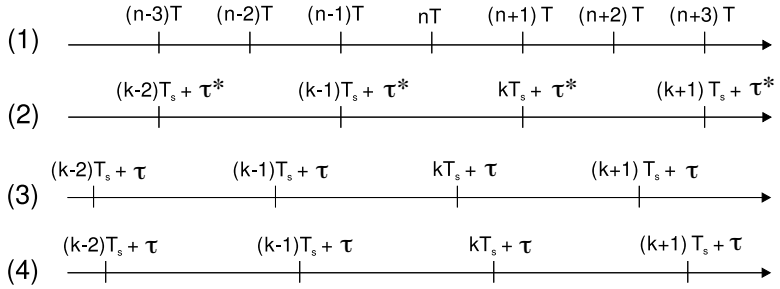


Figure 58: (1) Sampling clock and (2-4) possible actual symbol clocks. Symbol clock may be (2) synchronized in rate and phase, (3) synchronized in rate but not in phase, (4) synchronized neither in rate nor phase; with the sample clock.

In general, our receivers always deal with type (4) sampling clock error as drawn in Figure 58. That is, the sampling clock has neither the same exact rate nor the same phase as the actual symbol clock.

Def'n: *Incommensurate*
Two clocks with sampling period T and symbol period T_s are *incommensurate* if the ratio T/T_s is irrational. In contrast, two clocks are commensurate if the ratio T/T_s can be written as n/m where n, m are integers.

For example, $T/T_s = 1/2$, the two clocks are commensurate and we sample exactly twice per symbol period. As another example, if $T/T_s = 2/5$, we sample exactly 2.5 times per symbol, and every 5 samples the delay until the next correct symbol sample will repeat. Since clocks are generally incommensurate, we cannot count on them ever repeating exactly.

The situation shown in Figure 57 is case (3), where $T/T_s = 1/2$ (the clocks are commensurate),

but the sampling clock does not have the correct phase (τ is not equal to an integer multiple of T).

39.1 Sampling Time Notation

In general, for both cases (3) and (4) in Figure 58, the correct sampling times should be $kT_s + \tau^*$, but no samples were taken at those instants. Instead, $kT_s + \tau$ is always $\mu(k)T$ after the most recent sampling instant, where $\mu(k)$ is called the *fractional interval*. We can write that

$$kT_s + \tau = [m(k) + \mu(k)]T \tag{108}$$

where $m(k)$ is an integer, the highest integer such that $nT < kT_s + \tau$, and $0 \leq \mu(k) < 1$. In other words,

$$m(k) = \left\lfloor \frac{kT_s + \tau}{T} \right\rfloor$$

where $\lfloor x \rfloor$ is the greatest integer less than function (the Matlab `floor` function). This means that $\mu(k)$ is given by

$$\mu(k) = \frac{kT_s + \tau}{T} - m(k)$$

Example: Calculation Example
Let $T_s/T = 3.1$ and $\tau/T = 1.8$. Calculate $(m(k), \mu(k))$ for $k = 1, 2, 3$.
Solution:

$$\begin{aligned} m(1) &= \lfloor 3.1 + 1.8 \rfloor = 4; & \mu(1) &= 0.9 \\ m(2) &= \lfloor 2(3.1) + 1.8 \rfloor = 8; & \mu(1) &= 0 \\ m(3) &= \lfloor 3(3.1) + 1.8 \rfloor = 11; & \mu(1) &= 0.1 \end{aligned}$$

Thus your interpolation will be done: in between samples 4 and 5; at sample 8; and in between samples 11 and 12.

39.2 Seeing Interpolation as Filtering

Consider the output of the matched filter, $r(t)$ as given in (106). The analog output of the matched filter could be represented as a function of its samples $r(nT)$,

$$r(t) = \sum_n r(nT) h_I(t - nT) \tag{109}$$

where

$$h_I(t) = \frac{\sin(\pi t/T)}{\pi t/T}.$$

Why is this so? What are the conditions necessary for this representation to be accurate?

Answer: Read again the Nyquist sampling theorem from Lecture 4. As I said then, the Nyquist sampling theorem is an interpolation method.

If we wanted the signal at the correct sampling times, we could have it – we just need to calculate $r(t)$ at another set of times (not nT).

Call the correct symbol sampling times as $t = kT_s + \tau$ for integer k, where T_s is the actual symbol period used by the transmitter. Plugging these times in for t in (109), we have that

$$r(kT_s + \tau) = \sum_n r(nT) h_I(kT_s + \tau - nT)$$

Now, using the $(m(k), \mu(k))$ notation, since $kT_s + \tau = [m(k) + \mu(k)]T$,

$$r([m(k) + \mu(k)]T) = \sum_n r(nT) h_I([m(k) - n + \mu(k)]T).$$

Re-indexing with $i = m(k) - n$,

$$r([m(k) + \mu(k)]T) = \sum_i r([m(k) - i]T) h_I([i + \mu(k)]T). \tag{110}$$

This is a filter on samples of $r(nT)$, where the filter coefficients are dependent on $\mu(k)$.

Note: Good illustrations are given in [11] Section 8.4.2.

39.3 Approximate Interpolation Filters

Clearly, (110) is a filter. The desired sample at $[m(k)+\mu(k)]T$ is calculated by adding the weighted contribution from the signal at each sampling time. The problem is that in general, this requires an infinite sum over i from $-\infty$ to ∞, because the sinc function has infinite support.

Instead, we use polynomial approximations for $h_I(t)$:

- The easiest one we're all familiar with is linear interpolation (a first-order polynomial), in which we draw a straight line between the two nearest sampled values to approximate the values of the continuous signal between the samples. This isn't so great of an approximation.
- A second-order polynomial (*i.e.*, a parabola) is actually a very good approximation. Given three points, one can determine a parabola that fits those three points exactly.
- However, the three point fit does *not* result in a linear-phase filter. (To see this, note in the time domain that two samples are on one side of the interpolation point, and one on the other. This is temporal asymmetry.) Instead, we can use four points to fit a second-order polynomial, and get a linear-phase filter.
- Finally, we could use a cubic interpolation filter. Four points determine a 3rd order polynomial, and result in a linear filter.

To see results for different order polynomial filters, see the "Piecewise Polynomial Interpolation" sub-section of 8.4.2 in the Rice book.

39.4 Implementations

Note: These polynomial filters are called *Farrow filters* and are named after Cecil W. Farrow, of AT&T Bell Labs, who has the US patent (1989) for the "Continuously variable digital delay circuit". These Farrow filters started to be used in the 90's and are now very common due to the dominance of digital processing in receivers.

From (110), we can see that the filter coefficients are a function of $\mu(k)$, the fractional interval. Thus we could re-write (110) as

$$r([m(k) + \mu(k)]T) = \sum_i r([m(k) - i]T) h_I(i; \mu(k)). \tag{111}$$

That is, the filter is $h_I(i)$ but its values are a function of $\mu(k)$. The filter coefficients are a polynomial function of $\mu(k)$, that is, they are a weighted sum of $\mu(k)^0, \mu(k)^1, \mu(k)^2, \ldots, \mu(k)^p$ for a pth order polynomial filter.

Example: First order polynomial interpolation
For example, consider the linear interpolator.

$$r([m(k) + \mu(k)]T) = \sum_{i=-1}^{0} r([m(k) - i]T) h_I(i; \mu(k))$$

What are the filter elements h_I for a linear interpolation filter?
Solution:

$$r([m(k) + \mu(k)]T) = \mu(k) r([m(k) + 1]T) + [1 - \mu(k)] r(m(k)T)$$

Here we have used $h_I(-1; \mu(k)) = \mu(k)$ and $h_I(0; \mu(k)) = 1 - \mu(k)$.

Essentially, given $\mu(k)$, we form a weighted average of the two nearest samples. As $\mu(k) \to 1$, we should take the $r([m(k) + 1]T)$ sample exactly. As $\mu(k) \to 0$, we should take the $r(m(k)T)$ sample exactly.

39.4.1 Higher order polynomial interpolation filters

In general,

$$h_I(i; \mu(k)) = \sum_{l=0}^{p} b_l(i) \mu(k)^l$$

A full table of $b_l(i)$ is given in Table 8.1 of [11].

Note that the i indices seem backwards.

For the 2nd order Farrow filter, there is an extra degree of freedom – you can select parameter α to be in the range $0 < \alpha < 1$. It has been shown by simulation that $\alpha = 0.43$ is best, but people tend to use $\alpha = 0.5$ because it is only slightly worse, and division by two is extremely easy in digital filters.

Example: 2nd order Farrow filter
What is the Farrow filter for $\alpha = 0.5$ which interpolates exactly half-way between sample points?
Solution: From the problem statement, $\mu = 0.5$. Since $\mu^2 = 0.25, \mu = 0.5, \mu^0 = 1$, we can calculate that

$$\begin{aligned}
h_I(-2; 0.5) &= \alpha\mu^2 - \alpha\mu = 0.125 - 0.25 = -0.125 \\
h_I(-1; 0.5) &= -\alpha\mu^2 + (1 + \alpha)\mu = -0.125 + 0.75 = 0.625 \\
h_I(0; 0.5) &= -\alpha\mu^2 + (\alpha - 1)\mu + 1 = -0.125 - 0.25 + 1 = 0.625 \\
h_I(1; 0.5) &= \alpha\mu^2 - \alpha\mu = 0.125 - 0.25 = -0.125
\end{aligned} \tag{112}$$

Does this make sense? Do the weights add up to 1? Does it make sense to subtract a fraction of the two more distant samples?

Example: Matlab implementation of Farrow Filter

My implementation is called `plotFractionalInterpolationEg.m` and is posted on Canvas (along with four associated functions used in the script). Note I use T_sa in Matlab to denote the sample period because I felt "T" was too ambiguous. A plot of the result is shown in Figure 59.

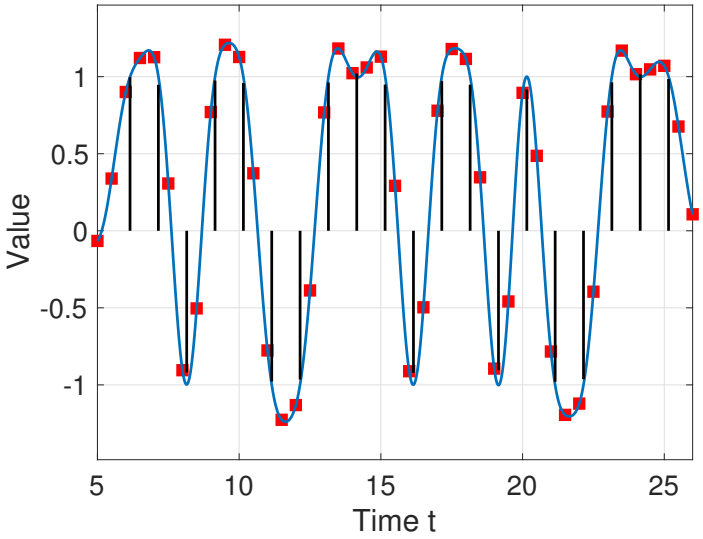


Figure 59: A BPSK signal is created with SRRC pulses and $\alpha = 0.75$ (blue solid line). Samples (red squares) are at $T/T_s = 1/4$. The optimal symbol sampling times are given by the black vertical lines. The interpolated value, using a cubic Farrow filter, is given by the height of the vertical black line.

39.5 Review

Timing synchronization is necessary to know when to sample the matched filter output. We want to sample at times $[n + \mu]T$, where n is the integer part and μ is the fractional offset. Often, we leave out the T and simply talk about the index n or fractional delay μ.

Implementations may be continuous time, discrete time, or a mix. We focus on the discrete time solutions.

- Problem: After matched filter, the samples may be at incorrect times, and in modern discrete-time implementations, there is generally no analog feedback to the ADC to correct when it is sampling.
- Solution: From samples taken at or above the Nyquist rate, you can interpolate between samples to find the desired sample.

However this solution leads to new problems:

- Problem: True interpolation requires significant calculation – the sinc filter has infinite impulse response.

- Solution: Approximate the sinc function with a 2nd or 3rd order polynomial interpolation filter, it works nearly as well.

- Problem: How does the receiver know when the correct symbol sampling time should be?

- Solution: It uses a 'timing locked' loop, analogous to a phased locked loop, which converges to the correct symbol sampling time. The Rice book covers this in its chapter on symbol synchronization [11]; but we don't cover this further in this course.

References

[1] S. M. Alamouti. A simple transmit diversity technique for wireless communications. *IEEE J. Select Areas in Communications*, 16(8):1451–1458, Oct. 1998.

[2] L. W. Couch. *Digital and Analog Communication Systems*. Pearson, 7th edition, 2007.

[3] D. Halperin, B. Greenstein, A. Sheth, and D. Wetherall. Demystifying 802.11n power consumption. In *Proc. 2010 Intl. Conf. on Power-Aware Computing and Systems*, 2010.

[4] D. Halperin, W. Hu, A. Sheth, and D. Wetherall. 802.11 with multiple antennas for dummies. *ACM SIGCOMM Computer Communication Review*, 40(1):19–25, 2010.

[5] R. V. Hartley. Transmission of information. *The Bell System Technical Journal*, 7(3):535–563, 1928.

[6] S. O. Haykin and M. Moher. *Modern Wireless Communications*. Pearson, 1st edition, 2005.

[7] S. Kay. *Fundamentals of Statistical Signal Processing, Volume II: Detection Theory*. Pearson, 1st edition, 1998.

[8] P. Lundrigan, N. Patwari, and S. K. Kasera. On-off noise power communication. In *Proc. of the ACM Intl. Conf. on Mobile Computing and Networking (MobiCom 2019)*, Oct. 2019. `https://span.engineering.wustl.edu/pub/lundrigan2019onoff.pdf`.

[9] D. Maas, M. H. Firooz, J. Zhang, N. Patwari, and S. K. Kasera. Channel sounding for the masses: Low complexity GNU 802.11b channel impulse response estimation. *IEEE Transactions on Wireless Communications*, 11(1):1–8, 2012.

[10] J. G. Proakis and M. Salehi. *Communication systems engineering*. Prentice Hall, 2nd edition, 2002.

[11] M. Rice. *Digital Communications: a Discrete-Time Approach*. Pearson Prentice Hall, 2009.

[12] W. Shakespeare. *Romeo and Juliet*. 1597.

[13] C. E. Shannon. A mathematical theory of communication. *The Bell System Technical Journal*, 27(3):379–423, 1948.

[14] V. Tarokh, H. Jafarkhani, and A. R. Calderbank. Space-time block codes from orthogonal designs. *IEEE Transactions on Information Theory*, 45(5):1456–1467, 1999.

[15] P. W. Wolniansky, G. J. Foschini, G. D. Golden, and R. A. Valenzuela. V-BLAST: An architecture for realizing very high data rates over the rich-scattering wireless channel. In *1998 URSI International Symposium on Signals, Systems, and Electronics*, pages 295–300, 1998.