ECE 361 - Fundamentals of Digital Communication

Lecture Notes

by Pramod Viswanath

January 13, 2017

Fundamentals of Digital Communication

Pramod Viswanath

January 19, 2015

Preface

Wireline and wireless communication devices (modems – DSL and cable and WiFi – and cellular radios) are the workhorses that drive the modern information age. Their central goal is to move around the basic digital currency of the information: *bits*. This movement is done on physical media, either via wires or via wireless. The key challenge is that the input-output relationship of the physical media is hard to pin down, so what one receives is not quite what one what sends. Broadly speaking, these notes are centered around a single theme:

reliably communicate bits over an unreliable physical medium.

The emphasis is on how to transfer bits between a *single* transmitter-receiver pair. The transfer involves a physical medium (wireline or wireless), whose input-output characteristics are not (cannot be) deterministically known. The curriculum has three broad parts:

- *Channel Model*: Even though the input-output relationship of the physical medium is not deterministically known, *statistical* quantities of this relationship, such as mean and correlation, can be physically measured and are typically constant over the time-scale of communication.

- *Transmission and Reception Strategies*: The statistical model of the physical medium is then brought into bearing in the engineering design of appropriate transmission and reception strategies (modulation and demodulation, in the language of this course).

- *Design Resources and Performance*: The resources available to the communication engineer are *power* and *bandwidth*. The final part of the course is to relate the statistical performance of the communication strategies to these two key resources.

These three parts are discussed in the course in the context of three specific physical media:

- *Additive white Gaussian noise* channel: The received signal is the transmit signal plus a statistically independent signal. This is a basic model that underlies the more complicated wireline and wireless channels.

- *Telephone* channel: The received signal is the transmit signal passed through a time-invariant, causal filter plus statistically independent noise. Voiceband v.90 modem and DSL technologies are used as examples.

- *Wireless* channel: The received signal is the transmit signal passed through a time-varying filter plus statistically independent noise. The GSM and CDMA standards are used as examples.

These lecture notes are aimed at introducing the fundamentals of digital communication to (junior) undergraduate students. The background expected includes a prior course in signals and systems and familiarity with statistical and probabilistic methods. The goals of the notes are to familiarize the students with the modeling, design and performance analysis of digital communication systems over two very common physical media: wireline and wireless channels. At the end of the course, the students should be able to:

- Reason out and arrive at appropriate *statistical* models for random physical phenomena. Specifically, this should be done for wireline and wireless channels. Further more, within wireless, terrestrial and deep space models are of interest.

- Translate the statistical modeling of the physical media into *design* strategies at the transmitter and the receiver. This should be done, specifically, for both the wireline and wireless channel models arrived at earlier.

- Understand the relation between the resources (bandwidth and power) and performance criteria (data rate and reliability). This should be specifically done for the wireline and wireless channels. The key difference between wireline and wireless channels in terms of which resource is worth what should be understood.

- Finally, be able to identify practical technologies around us that use the design ideas identified in the course.

A defining feature of these notes is that they succinctly relate the "story" of how bits get reliably transmitted from the source to the destination, via the physical medium. All the key "plots" involved in the story are developed organically and (mostly) from first principles – this means the material in these notes holistically incorporates topics typically covered in diverse areas of electrical engineering: communication theory, information theory, coding theory and signal processing. The material here gets the (undergraduate) student ready to explore more involved (graduate school) topics – ranging from theoretical developments underpinning modern wireless communication systems to more implementation-intensive projects involving reliable communication (examples: sensor networks, Internet of things).

Lecture 1: Discrete Nature of information

Introduction

The currency of today's information age is digital: *bits*. Digital communication is reliable transmission of this currency over an unreliable physical medium. The engineering appeal of representing very different types of information sources such as voice, images and video using a common currency is very clear: the storage and communication aspects are separated from the underlying information type. In some canonical instances this strategy is without loss of generality: i.e., in a strong mathematical sense, this strategy cannot be outperformed by any other joint representation and communication strategy (topics such as this are studied in a field of study known as information theory). The block diagram in Figure 1 shows a high level representation of a typical communication system. The *discrete message source* continuously outputs a stream of bits that represent the information we would like to transmit. Bits are abstract entities that need to be *mapped* to a physical quantity, such as an electromagnetic signal, to be transmitted over a physical medium.

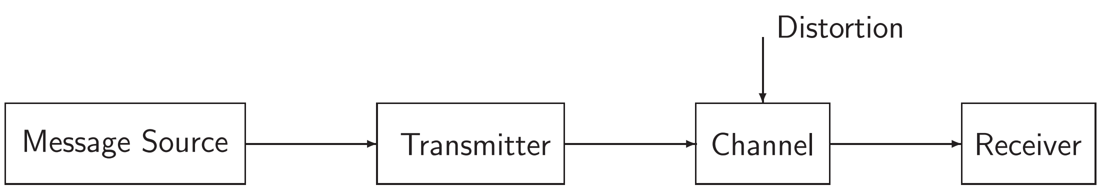

Figure 1: The basic block diagram of a communication system

The behavior of the physical medium is uncertain: what you get is *not a deterministic* function of what you send; this uncertainty is the essence of communication. While the behavior of the channel[1] over one experiment cannot be predicted, the *average* behavior, averaged over many experiments turns out to be well behaved in many physically interesting scenarios. The characterization of the average behavior, or in other words, the *statistical characterization* of the physical medium is crucial to understanding how to communicate the bits reliably to the receiver. A primary component of the communication engineer's tool-box is robust and reasonable statistical models of important physical channels such as the wireline channel and the wireless channel.

[1]Channel is a term we will use throughout these notes to denote the unreliable physical medium.

A Simple Noise Model

We will begin with a simple form of a physical medium where we only transmit and receive voltages (real numbers). The received voltage y, is the transmitted voltage x, plus "noise" w:

$$y = x + w. \tag{1}$$

The simplest model of the noise is that w is strictly within a certain range, say $\pm\sigma_{\text{th}}$. In other words, we receive a voltage that is within $\pm\sigma_{\text{th}}$ Volts from the voltage we transmitted.

A Simple Communication Scheme

Suppose we want to send a single bit across this channel. We can do this by transmitting a voltage v_0 to transmit an information content of the bit being "zero", and a voltage v_1 when transmitting an information content of the bit being "one". As long as

$$|v_0 - v_1| > 2\sigma_{\text{th}}, \tag{2}$$

we can be certain that our communication of the one bit of information is reliable over this channel. Physically, the voltage transmitted corresponds to some energy being spent: we can say that the energy spent in transmitting a voltage v Volts is (proportional to) v^2 Joules. In this context, a natural question to ask is the following: how many bits can we reliably communicate with an energy constraint of E Joules?

Some thought lets us come up with the following transmission scheme: we choose to transmit one of a collection of *discrete* voltage levels:

$$\left\{-\sqrt{E}, -\sqrt{E} + 2\sigma_{\text{th}}, \ldots, -\sqrt{E} + 2k\sigma_{\text{th}}, \ldots, +\sqrt{E}\right\}, \tag{3}$$

where we have assumed for simplicity that $\sqrt{E}$ is divisible by σ_{th}. So, we can communicate one of

$$1 + \frac{\sqrt{E}}{\sigma_{\text{th}}} \tag{4}$$

discrete voltage levels entirely *reliably* to the receiver. This corresponds to

$$\log_2\left(1 + \frac{\sqrt{E}}{\sigma_{\text{th}}}\right) \tag{5}$$

bits being reliably communicated to the receiver (why?). The diagram in Figure 2 demonstrates one possible mapping between the 4 sequences of 2 bits to the 4 discrete voltage levels being transmitted. Here $\sqrt{E} = 3\sigma_{\text{th}}$ and

$$v_k = -\sqrt{E} + 2k\sigma_{\text{th}}, \quad j = 0, 1, 2, 3. \tag{6}$$

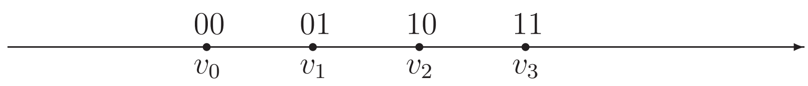

Figure 2: Mapping from bits to voltage levels.

Relation between Energy and Reliable Information Transmitted

For a given energy constraint E, the number of bits we can communicate reliably is, from (5),

$$\log_2\left(1 + \frac{\sqrt{E}}{\sigma_{\mathrm{th}}}\right). \tag{7}$$

A natural sort of question that the communication engineer is interested in is the following: if we want to send an additional bit reliably how much more energy do we need to expend? We can use the above expression to answer this question: the new energy energy $\tilde{E}$ required to send an extra bit of informamtion reliably has to satisfy:

$$\log_2\left(1 + \frac{\sqrt{\tilde{E}}}{\sigma_{\mathrm{th}}}\right) = 1 + \log_2\left(1 + \frac{\sqrt{E}}{\sigma_{\mathrm{th}}}\right), \tag{8}$$

$$1 + \frac{\sqrt{\tilde{E}}}{\sigma_{\mathrm{th}}} = 2\left(1 + \frac{\sqrt{E}}{\sigma_{\mathrm{th}}}\right), \tag{9}$$

$$\sqrt{\tilde{E}} = \sigma_{\mathrm{th}} + 2\sqrt{E}. \tag{10}$$

In other words, we need to more than *quadruple* the energy constraint to send just one extra bit of information reliably.

Another interesting thing to note is that the amount of reliable communication transmitted depends on the *ratio* between the transmit energy budget E and the energy of the noise σ_{th}^2. This ratio, E/σ_{th}^2, is called the *signal to noise* ratio and will feature prominently in the other additive noise models we will see.

Looking Forward

This simple example of a channel model gave us a feel for simple transmission and reception strategies. It also gave us an idea of how a physical resource such as energy is related to the amount of information we can communicate reliably. The deterministic channel model we have used here is rather simplistic; in particular, the choice of σ_{th} might have to be overly conservative if we have ensure that the additive noise has to lie in the range $\pm\sigma_{\text{th}}$ with full certainty. If we are willing to tolerate some error in reliable communication, we can set a lower range σ_{th} in our channel model and thus allowing for a higher rate of reliable communication of information. This is the topic of the next lecture.

Lecture 2: Statistical Channel Model

Introduction

We began our study of reliable communication last lecture with a very simple model of the additive noise channel. This is a reasonable model, except that one may have a very conservative value for the extremal noise fluctuations $\pm\sigma_{\text{th}}$. This will lead to a correspondingly poor performance (in terms of number of reliable bits communicated for a given energy constraint). In this lecture, we take a more nuanced look at the additive noise channel model. Our basic goal is to have a *statistical* model of the additive noise. This will allow us to talk about reliable communication with a desired level of reliability (as opposed to the "fully reliable" notion of the previous lecture).

Statistical models can be arrived at by plain experiments of how the additive noise looks like and taking the *histogram* as the statistical model. Based on what this model is and a desired reliability level, we could work out the appropriate value of σ_{th}. We could then directly use this choice of σ_{th} to our communication strategies from the previous lecture (transmit voltages as far apart from each other). While this already gives a significant benefit over and above the conservative estimates of the worst-case fluctuation σ_{th}, this may not be the optimal communication strategy (in terms of allowing largest number of bits for a given energy constraint and reliability level). We will see that depending on the exact shape of the histogram one can potentially do better. We will also see when the performance cannot be improved beyond this simple scheme for a wide range of histograms. Finally, we will see that most histograms that arise in nature are indeed of this type. Specifically, it turns out that most interesting noise models have the *same* statistical behavior with just two parameters that vary: the mean (first order statistics) and variance (second order statistics). So, we can design our communication schemes based on this *universal* statistical model and the performance only depends on two parameters: the mean and variance. This streamlines the communication design problem and allows the engineer to get to the heart of how the resources (power and bandwidth) can be used to get maximum performance (rate and reliability).

We start out with a set of properties that most additive noise channels tend to have. Next, we will translate these properties into an appropriate mathematical language. This will allow us to arrive at a robust *universal* statistical model for additive noise: it is *Gaussian* or *normally* distributed. We will see that our understanding of transmission and reception strategies using the deterministic model from the previous lecture extends naturally to one where the model is statistical.

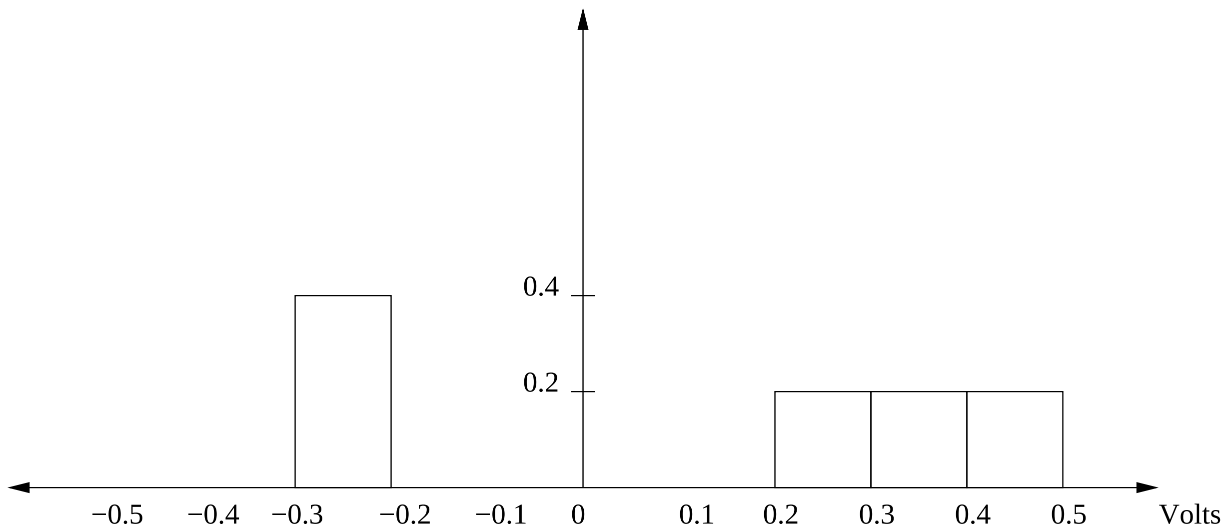

Figure 3: A exemplar histogram.

Histogram Models and Reliable Communication Strategies

Suppose we make detailed measurements of the noise values at the location where we expect communication to take place. Suppose we have made N separate measurements, where N is a large value (say, 10,000): $v_1, \ldots, v_N$. The *histogram* of the noise based on the measurements at a resolution level of Δ is simply a function from voltage levels to the real numbers: for every $a \in (m\Delta, (m+1)\,\Delta)$,

$$f_\Delta(a) \stackrel{\text{def}}{=} \frac{1}{N} \sum_{k=1}^{N} \mathbf{1}_{v_k \in (m\Delta,(m+1)\Delta)}, \tag{11}$$

where we have denoted the *indicator function*

$$\mathbf{1}_{\cdot} \stackrel{\text{def}}{=} \begin{cases} 1 & \text{if the parameter is true} \\ 0 & \text{else.} \end{cases} \tag{12}$$

One important property of the histogram function is that the area under the histogram curve is equal to unity. For example, with $N = 5$ and $v_1 = 0.2V, v_2 = -0.25V, v_3 = 0.45V, v_4 = -0.27V, v_5 = 0.37V$, the histogram at a resolution of $\Delta = 0.1V$ is depicted in Figure 3. In the limit of very large number of samples N and very small resolution Δ, the histogram function is called the *density* of the noise. Henceforth we will use the term density to denote the histogram created from the noise measurements. As any histogram, the density function is always non-negative and the area under it is unity. The density function of a noise that takes any voltage value in the range $[-0.5V, 0.5V]$ equally likely is depicted in Figure 4.

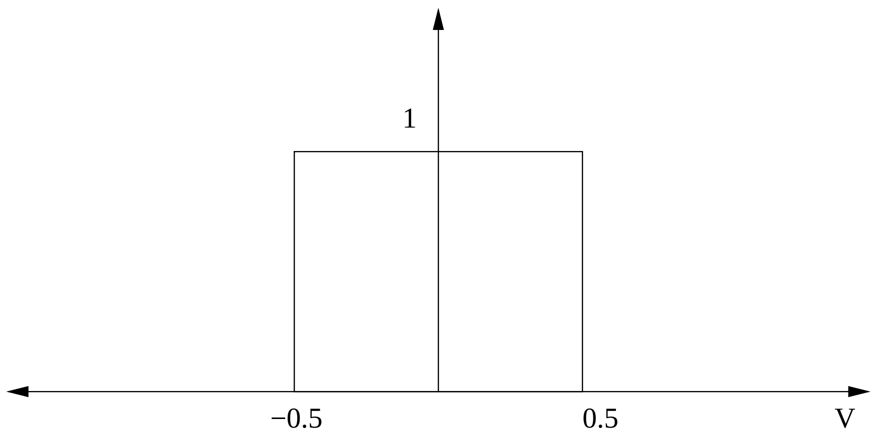

Figure 4: A uniform density function.

Now suppose we are willing to tolerate errors in communication a fraction η of the time. Then we can pick the smallest value of σ_{th} such that the area under the density function over the range $\pm\sigma_{\text{th}}$ is at least $1-\eta$. This ensures that the noise is within σ_{th} at least a fraction $1-\eta$ of the time. For the density function in Figure 4, a value of $\eta = 0.1$ means that $\sigma_{\text{th}} = 0.45V$; a pictorial depiction is available in Figure 5.

We can now pick the transmission and reception schemes as in Lecture 1 using this new value of $\sigma_{\text{th}} = 0.45V$. We are now guaranteed reliable communication at a level of tolerable unreliability $\eta = 0.1$. This corresponds to a saving in energy of a fraction

$$\frac{0.05V}{0.5V} = 10\%. \tag{13}$$

While this might seem modest, consider the density function in Figure 6, where σ_{th} in the usual sense of Lecture 1 would be $10V$. On the other hand with $\eta = 0.1$, the new value of σ_{th} is only $1.0V$. This corresponds to a savings in energy of a fraction

$$\frac{9V}{10V} = 90\%, \tag{14}$$

a remarkably large fraction!

In the transmission scheme of Lecture 1, we picked the different possible transmit voltage levels to be spaced by at least $2\sigma_{\text{th}}$. This seems reasonable since we know a bound for how much the noise can fluctuate. But we have more knowledge about how the noise fluctuates based on the density function. This provokes us to think along the following natural thought process:

> **Question**: Given the noise density function and energy and reliability constraints, is the scheme of keeping the different transmit voltages apart by $2\sigma_{\text{th}}$ the best one, in terms of giving maximum number of bits?

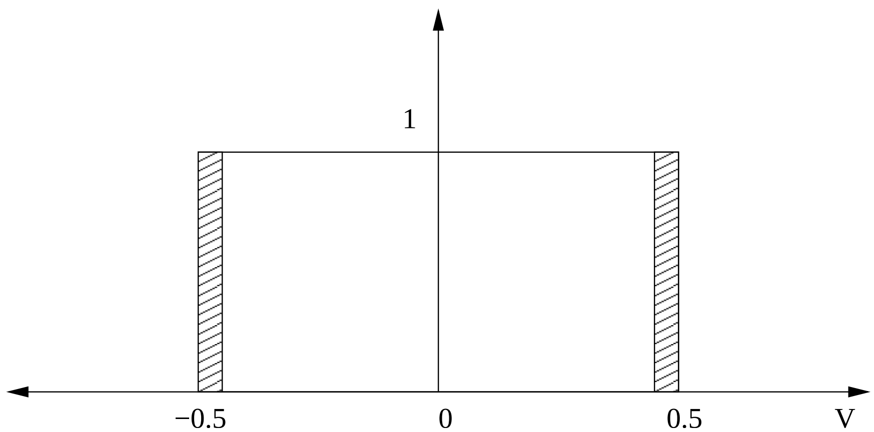

Figure 5: Choosing a threshold based on the reliability level.

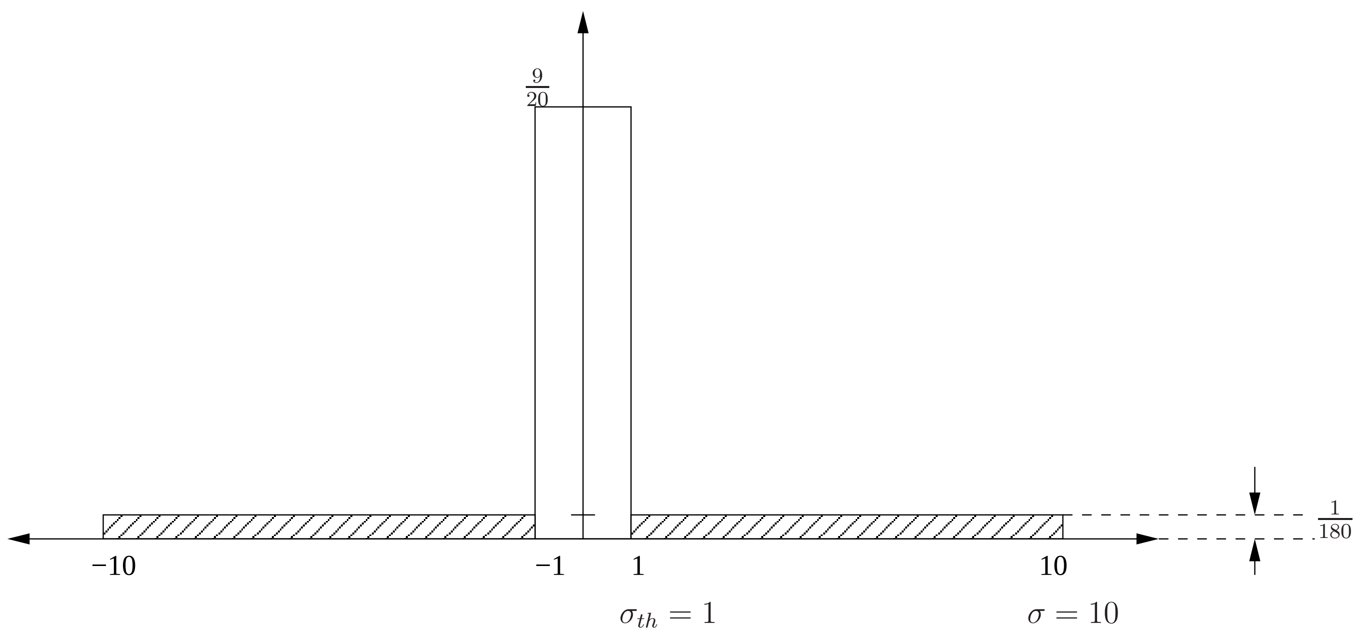

Figure 6: The threshold based on the reliability level can be significantly smaller than one based on worst-case.

It turns out that the general answer is *no*. But for a large class of density functions the natural approach of extracting the appropriate σ_{th} from the density function to use in the design of Lecture 1 *suffices*. Interestingly, it turns out that most density functions for *additive* noise have this property. In the rest of this lecture, we will study some canonical properties of the density of additive noise; we start with some simple physical properties.

Physical Properties of Additive Noise

An enumeration of some reasonable properties we may anticipate the additive forms of noise to take is the following.

1. The noise is the overall result of many additive "sub-noises". Typical sub-noises could be the result of thermal noise, device imperfections and measurement inaccuracies.

2. These sub-noises typically have little correlation with respect to each other. We suppose the stronger statement: they are statistically *independent* of each other.

3. No sub-noise is particularly dominant over the other. In other words, they all contribute about the same to the total noise.

4. Finally, there are many sources of sub-noises.

We will work to convert these physical properties into more precise mathematical statements shortly.

Representation of Additive Noise

Using some notation, we can write the total additive noise w as

$$w = n_1 + n_2 + \cdots + n_m, \tag{15}$$

the sum of m sub-noises $n_1, \ldots, n_m$. Furthermore, the sub-noises $n_1, \ldots, n_m$ are statistically independent of each other. Let us denote the densities of the sub-noises as $f_{n_1}(\cdot), \ldots, f_{n_m}(\cdot)$, respectively. An important result from basic probability is the following result:

> The density of the total noise w is the *convolution* of the densities of the sub-noises.

This result is best understood in the Laplace or Fourier domain. Specifically, the Laplace transform of a density function $f_w(\cdot)$ is defined as

$$F_w(s) = \int_{-\infty}^{\infty} e^{-sa} f_w(a),\ da \qquad \forall s \in \mathbb{C}. \tag{16}$$

Here $\mathbb{C}$ is the complex plane. In terms of the Laplace transforms of the densities of each of the sub-noises,

$$F_w(s) = \prod_{k=1}^{m} F_{n_k}(s), \quad \forall s \in \mathbb{C}. \tag{17}$$

We know what the density function of a noise is from an engineering stand point: it is simply the histogram of a lot of noise measurements at a fine enough resolution level. How does one understand the Laplace transform of the density function from an engineering and physical view point? We can do a Taylor series expansion around $s = 0$ to get a better view of the Laplace transform of a density function:

$$F_w(s) = F_w(0) + sF_w^{'}(0) + \frac{s^2}{2}F_w^{''}(0) + o(s^2), \tag{18}$$

where the function $o(s^2)$ denotes a function of s^2 that when divided by s^2 goes to zero as s approaches zero itself. The first term

$$F_w(0) = \int_{-\infty}^{\infty} f_w(a)\, da \tag{19}$$
$$= 1, \tag{20}$$

since the area under a density function is unity. The second term can be calculated as

$$\frac{d}{ds}F_w(s) = \int_{-\infty}^{\infty} -ae^{-sa} f_w(a)\, da, \tag{21}$$
$$F_w^{'}(0) = \int_{-\infty}^{\infty} a f_w(a)\, da \tag{22}$$
$$\stackrel{\text{def}}{=} \mathbb{E}\left[w\right]. \tag{23}$$

The quantity $\mathbb{E}[w]$ is the *mean* of the noise w and is a readily measured quantity: it is just the *average* of all the noise measurements. In the sequence above, we blithely interchanged the differentiation and integration signs; this will need to be mathematically justified, but we skip this step here for brevity.

Now for the third term:

$$\frac{d^2}{ds^2}F_w(s) = \int_{-\infty}^{\infty} a^2 e^{-sa} f_w(a)\, da, \tag{24}$$
$$F_w^{''}(0) = \int_{-\infty}^{\infty} a^2 f_w(a)\, da \tag{25}$$
$$= \mathbb{E}\left[w^2\right]. \tag{26}$$

Here the quantity $\mathbb{E}[w^2]$ is the *second moment* of the noise w and is a readily measured quantity: it is just the *average* of the square of the noise measurements. Again, we have interchanged the differentiation and integration signs in the calculation above.

In conclusion, the first few terms of the Taylor series expansion of the Laplace transform of the density of the additive noise w involves easily measured quantities: mean and second moment. Sometimes the second moment is also calculated via the *variance*:

$$\mathrm{Var}(w) \stackrel{\mathrm{def}}{=} \mathbb{E}\left[w^2\right] - \left(E\left[w\right]\right)^2. \tag{27}$$

These two quantities, the mean and variance, are also referred to simply as *first* and *second order statistics* of the measurements and are fairly easily empirically evaluated. Let us denote these two quantities by μ and σ^2_{th}, respectively henceforth. While we may not have access to the densities of the individual sub-noises, we can calculate their first and second order statistics by using the assumption that each of the sub-noises contributes the same level to the total noise. This means that, since

$$\mathbb{E}\left[w\right] = \sum_{k=1}^{m} \mathbb{E}\left[n_k\right], \tag{28}$$

we can say that

$$E\left[n_k\right] = \frac{\mu}{m}, \quad \forall k = 1 \dots m. \tag{29}$$

Similarly for statistically independent sub-noises $n_1, \dots, n_m$ we have

$$\mathrm{Var}\left(w\right) = \sum_{k=1}^{m} \mathrm{Var}\left(n_k\right), \tag{30}$$

we can say that

$$\mathrm{Var}\left(n_k\right) = \frac{\sigma^2}{m}, \quad \forall k = 1 \dots m, \tag{31}$$

$$\mathbb{E}\left[n_k^2\right] = \frac{\sigma^2}{m} + \frac{\mu^2}{m^2}. \tag{32}$$

Here we used Equation (27) in arriving at the second step.

Now we can use an approximation as in Equation (18), by ignoring the higher order terms, to write

$$F_{n_k}(s) \approx 1 - \frac{\mu s}{m} + \frac{\sigma^2 s}{2m} + \frac{\mu^2 s^2}{2m^2}, \quad \forall k = 1 \dots m. \tag{33}$$

Substituting this into Equation (17), we get

$$F_w(s) \approx \left(1 - \frac{\mu s}{m} + \frac{\sigma^2 s^2}{2m} + \frac{\mu^2 s^2}{2m^2}\right)^m, \quad \forall s \in \mathbb{C}. \tag{34}$$

We are interested in the density function of the noise w for large number of sub-noises, i.e., when m is large. From elementary calculus techniques, we know the limiting formula:

$$\lim_{m\to\infty} F_w(s) = e^{-\mu s + \frac{s^2\sigma^2}{2}}, \quad \forall s \in \mathbb{C}. \tag{35}$$

Remarkably, we have arrived at a *universal* formula for the density function that is parameterized by only two simply measured physical quantities: the first order and second order statistics (mean μ and variance σ^2, respectively). This calculation is known as the *central limit theorem*.

It turns out that the density function whose Laplace transform corresponds to the one in Equation (35) is

$$f_w(a) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(a-\mu)^2}{2\sigma^2}}, \quad \forall a \in \mathbb{R}. \tag{36}$$

This density function is called *Gaussian*, in honor of the first person who discovered it. It is also called the *normal* density since it also shows up in many real world situations which are entirely unrelated to additive noises (all the way from temperature measurements to weights of people to the eventual grades of the students in this course (hopefully!), are all "normally" behaved).

There are some important modern day data that are famously *not* normal: size of packets on the internet and the number of goods bought in an online store. I recommend the following books

> C. Anderson, *The Long Tail: Why the Future of Business is Selling Less of More*, Hyperion, 2006;

and

> Nassim Nicholas Taleb, *The Black Swan: The Impact of the Highly Improbable*, Random House, 2007,

that make for interesting reading (unrelated to the scope of this course). You can also get a broader feel for how such measurements are harnessed in making engineering and economic decisions.

Looking Forward

In the next lecture we will see how to use this particular structure of the density function in choosing our communication transmit and receive strategies.

Lecture 3: Histogram to Optimum Receiver

Introduction

In this lecture we focus our study on how to use the detailed statistical knowledge available in the histogram of the noise in reliably communicating at a desired level of reliability. Though our specific interest will be on the Gaussian statistics, it helps (for later lectures) to study the more general situation. For a fixed transmission strategy, we will derive the *optimum* receiver in terms of minimizing the unreliability of communication. Towards doing this, we formally define what unreliability means by carefully looking at the different sources of randomness and what statistical assumptions we make about them. We conclude with a fundamental relation between the variance of the noise σ^2, the transmit energy constraint E, and the reliability of communication.

Sources of Randomness

There are two sources of randomness from the perspective of the receiver: one intrinsic (the information – bits – itself is unknown) and the other extrinsic (the additive noise introduced by the channel). The receiver typically knows some statistical information about these sources of knowledge.

- *Statistics of the bit*: this is the fraction of bits that are 0. If there is some prior information on how likely the transmitted information bit is say, 1, then that could factor in the decision rule. In the extreme instance, if we somehow knew before the communication process that the information bit is 1 for sure, then we don't need to worry about the received voltage. We just decide at the receiver that the information bit is 1. Many a time, no such prior knowledge is available. In this case, we suppose that the information bit is *equally likely* to be 1 or 0.

- *Noise Statistics*: knowing whether the noise is more likely to be small or large will help the receiver make the decision. For instance, if the noise is more likely to be near zero than large, the receiver would likely pick the nearer of the two possible transmit voltages as compared to the received voltage (the so-called *nearest-neighbor* rule). One of the main conclusions at the end of Lecture 2 is that additive noise in the physical world is (far) more likely to be near its mean than away from it.

Figure 7 illustrates the action taken at the receiver.

Formal Definition of Reliable Communication

Consider a single bit to be communicated reliably. Figure 8 diagrammatically illustrates the familiar bits-to-voltages mapping at the transmitter.

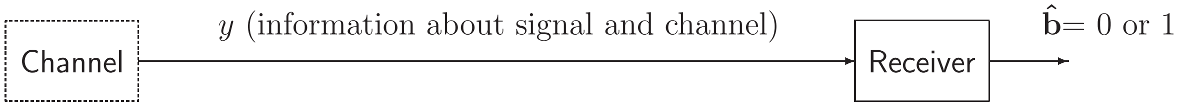

Figure 7: The basic block diagram of a receiver.

Figure 8: Mapping for sending 1 bit across an AGN channel.

The main job at the receiver is to decide on the information bit transmitted, denoted by say, $\hat{b}$, based on the received voltage y. The correct choice of the decision rule at the receiver is the one that maximizes the reliability of communication. Alternatively, we want to minimize the unreliability of communication. We will say an *error* occurs whenever communication is unreliable. In this case, the transmitted information is just one bit and there is only one way an error can occur. More generally, when we get around to sending multiple bits of information we will follow the convention that an error occurs *even if a single bit* is communicated erroneously. This convention is a natural byproduct of the nature of the digital world of information around us.

Actual sources of information (such as voice, images and video) have features that range the whole gamut from being very important to hardly any. For instance, if we consider digitizing voice with a 16-bit A/D converter the most significant bits (MSBs) are (almost by definition!) more important than the least significant bits (LSBs). Indeed, when communicating the 16-bit digital representation of the analog voice sample, we need to pay more attention to the reliability of the MSBs as compared to the LSBs.

On the other hand, the digital world around is organized very differently. Information collection is typically at a very different engineering level than information transmission: Information collection is done typically by microphones, cameras and smart phones. Information transmission is done typically over the ethernet or wireless. There are many layers of separation between the engineering devices that do these two operations. Specifically, there are, starting from information collection and moving down to information transmission:

- the *application* layer, that decides whether the digital format for the voice is `.wav` or `.mp3`;
- the *transport* layer, that decides whether the `TCP/IP` protocol is being used or a proprietary one used by cell phones and the corresponding impact on the digital representation of the analog voice sample;
- the *networking* and *physical* layers, that decide what format to finally package the digital voice data in.

So by the time the transmission of communication is initiated, the "analog" nature of the digital information (MSBs and LSBs) is entirely lost (or at least hidden underneath a whole lot of protocol layers). So, the communication problem is usually stated as trying to equally reliably send all the bits (whether they are MSBs, or LSBs, or formatting information corresponding to the different protocols involved). We will follow this tradition in this course by considering all the bits to be equally important.

We now have a formal definition of how reliable communication is. It is the *average* probability (averaged over the a priori probabilities with which the information bits take different values) with which all the bits are correctly received. We will next see the decision rule at the receiver that is optimal in the sense of allowing the most reliable communication.

The Optimal Decision Rule: MAP

To begin with, let us list all the information that the receiver has.

1. The a priori probabilities of the two values the information bit can take. We will normally consider these to be equal (to 0.5 each).
2. The received voltage y. While this is an analog value, i.e., any real number, in engineering practice we *quantize* it at the same time the waveform is converted into a discrete sequence of voltages. For instance if we are using a 16-bit ADC for the discretization, then the received voltage y can take one of 2^{16} possible values. We will start with this discretization model first.
3. The *encoding rule*. In other words we need to know how the information bit is mapped into voltages at the transmitter. For instance, this means that the mapping in illustrated in Figure 8 should be known to the receiver. This could be considered part of the protocol that both the transmitter and receiver subscribe to. In engineering practice, all widespread communication devices subscribe to a universally known *standard*.

Assuming L possible discrete received voltage levels, Figure 9 illustrates the two possible transmit voltages and the chance that they lead to the discrete shows a plot of possible transmitted and the chance with which they could lead to the L possible received voltages (here $L = 3$). The additive Gaussian noise channel model combined with the discretization of the received voltage level naturally leads to a statistical characterization of how likely a certain received voltage level is given a certain transmit voltage level. In Figure 9, we have written these probabilities in the most general form.

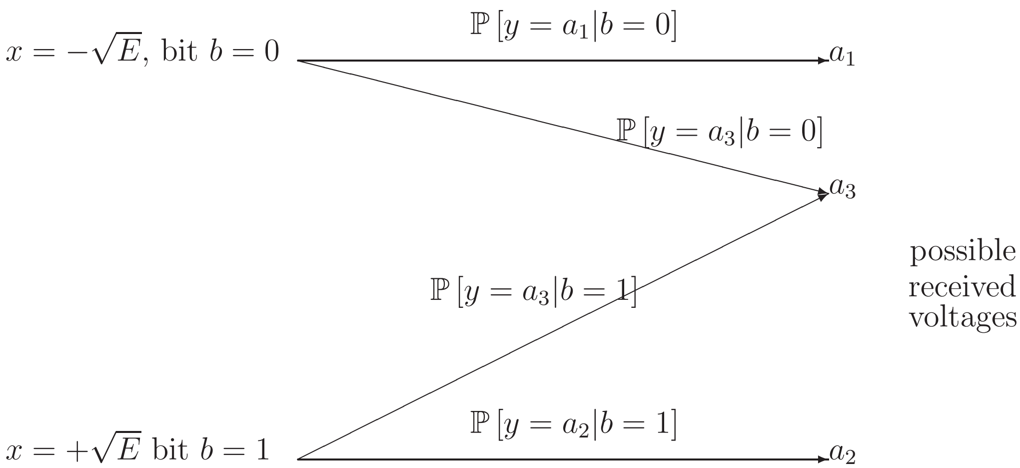

Figure 9: Sent and received voltage pairs along with their conditional probabilities

The probability that the information bit is i (either 1 or 0) *and* the received voltage is a (one of L possible values, denoted by $a_1, \ldots, a_j$) is simply

$$\mathbb{P}[b = i, y = a] = \mathbb{P}[b = i | y = a] \, \mathbb{P}[y = a], \tag{37}$$

where the unconditional probability that the received voltage is a,

$$\mathbb{P}[y = a] = \mathbb{P}[b = 0, y = a] + \mathbb{P}[b = 1, y = a], \tag{38}$$

does *not depend* on the actual value of the information bit b. The quantity $\mathbb{P}[b = i | y = a]$ in Equation (37) is known as the *a posteriori* probability of the information bit being equal to i. This captures the role of the communication process: the received voltage level alters our perception of what the information bit could possibly be.

The decision rule at the receiver then is to map every possible received voltage level to a particular estimate $\hat{b}(a)$ of what was sent. The reliability of communication conditioned on a specific receieved voltage level (say, a) is simply the a posteriori probability of the information bit b being equal to the estimate $\hat{b}$:

$$\mathbb{P}[\mathcal{C} | y = a] = \mathbb{P}\left[b = \hat{b}(a) | y = a\right]. \tag{39}$$

We want to maximize $\mathbb{P}[\mathcal{C}|y=a]$, so we should just choose $\hat{b}(a)$ to be that value (1 or 0) which has the larger *a posteriori* probability.

But how does one calculate this quantity at the receiver, using the three quantities that the receiver has access to (enumerated at the beginning of this lecture)? For any received voltage level a in the set $\{a_1, \ldots, a_L\}$, the a posteriori probability for the information bit b being equal to, say 1, can be written using the *Bayes* rule as:

$$\mathbb{P}[b=1|y=a] = \frac{\mathbb{P}[y=a|b=1]\,\mathbb{P}[b=1]}{\mathbb{P}[y=a]}. \tag{40}$$

Similarly the a posteriori probability for the information bit b being equal to 0, given that the received voltage is the same a, is

$$\mathbb{P}[b=0|y=a] = \frac{\mathbb{P}[y=a|b=0]\,\mathbb{P}[b=0]}{\mathbb{P}[y=a]}. \tag{41}$$

Since the denominator is common to both the two a posteriori probabilities and the decision rule is only based on the relative comparison, we only need the numerators to form the decision rule. The a priori probabilities $\mathbb{P}[b=1]$ and $\mathbb{P}[b=0]$ sum to unity and is part of the information the receiver has ahead of time. The *likelihoods*

$$\mathbb{P}[y=a|b=1] \quad \text{and} \quad \mathbb{P}[y=a|b=0] \tag{42}$$

is to be calculated based on the statistical knowledge of the channel noise. We will do this shortly for the Gaussian noise, but a couple of quick digressions are in order before we do that.

ML Decision Rule

As we discussed earlier, a common situation in communication is that that the a priori probabilities of the information bit are equal to each other. In this (typical) situation, the MAP rule simplifies even more. It now suffices to just compare the two likelihoods (the two quantities in Equation (42)). The decision rule is then to decide that $\hat{b}$ is 1 if

$$\mathbb{P}[y=a|b=1] > \mathbb{P}[y=a|b=0], \tag{43}$$

and 0 otherwise. This rule is called the *maximum likelihood* rule. Due to its typicality, this will be the decision rule we will use throughout this course at the receiver.

MAP and ML Rules for the AGN Channel

Given the universality of the Gaussian statistics for additive noise models, it is of immediate interest to calculate these rules for such a statistical channel model. The only potential hurdle is that the statistics are described for *analog* valued noise (and hence received voltage) levels. In our setup so far, we only considered a discrete set of voltage levels. We now have one of two options: either generalize the previous description to analog values (a whole continuous range of voltage levels than a finite number) or deduce the statistics of the discrete noise levels as induced by the Gaussian statistics on the analog noise level and the ADC. We take the former approach below.

The generalization required is only a matter of calculating the a posteriori probabilities conditioned on a whole continuous range of received voltage levels, than just a finite number. Following the earlier calculation in Equation (40), we see the main technical problem:

$$\mathbb{P}\left[y = a | b = 1\right] \quad \text{and} \quad \mathbb{P}\left[y = a\right] \tag{44}$$

are both zero: the chance that an analog noise level is exactly a value we want is simply zero. So we cannot use Bayes rule naively. Since we only need the ratio of these two quantities (cf. Equation (44)) in the MAP rule, we can use the L'Hopital's rule:

$$\frac{\mathbb{P}\left[y = a | b = 1\right]}{\mathbb{P}\left[y = a\right]} = \lim_{\epsilon \to 0} \frac{\mathbb{P}\left[y \in (a - \epsilon, a + \epsilon) | b = 1\right]}{\mathbb{P}\left[y \in (a - \epsilon, a + \epsilon)\right]} = \frac{f_y(a | b = 1)}{f_y(a)}. \tag{45}$$

Here $f_y(\cdot)$ is the PDF of the *analog* received voltage y and $f_y(\cdot | b = 1)$ is the PDF of the received voltage conditioned on the event that the information bit b is 1. So, the MAP rule when the received voltage is equal to a is:

decide $\hat{b} = 1$ if

$$\mathbb{P}\left[b = 1\right] f_y(a | b = 1) \geq \mathbb{P}\left[b = 0\right] f_y(a | b = 0) \tag{46}$$

and 0 otherwise.

The ML rule is simpler, as usual:

decide $\hat{b} = 1$ if

$$f_y(a | b = 1) \geq f_y(a | b = 0) \tag{47}$$

and 0 otherwise.

For the additive noise channel, it is a straightforward matter to calculate the conditional PDFs of the received voltage. Indeed

$$\begin{aligned} f_y(a | b = 1) &= f_y(a | x = +\sqrt{E}) && (48) \\ &= f_w(a - \sqrt{E} | x = +\sqrt{E}) && (49) \\ &= f_w(a - \sqrt{E}). && (50) \end{aligned}$$

In the first step we used the knowledge of the mapping between the information bit to transmit voltage levels (cf. Figure 8). The second step is simply using the fact that $w = y - x$. The third step used the statistical independent of the additive noise and the voltage transmitted. So, the MAP and ML rules for the additive noise channel are:

MAP: decide $\hat{b} = 1$ if

$$\mathbb{P}\left[b = 1\right] f_w(a + \sqrt{E}) \leq \mathbb{P}\left[b = 0\right] f_w(a - \sqrt{E}) \tag{51}$$

and 0 otherwise;

and

ML: decide $\hat{b} = 1$ if

$$f_w(a + \sqrt{E}) \leq f_w(a - \sqrt{E}) \tag{52}$$

and 0 otherwise.

We can simplify the rules even further given some more knowledge of the statistics of the noise. For example, suppose we know that the noise w is more likely to be small in magnitude than large (since the the mean was supposed to be zero, this means that the noise is more likely to be near the average value than farther away):

$$f_w(a) \geq f_w(b), \quad |a| \geq |b|. \tag{53}$$

This property is definitely true for the Gaussian statistics. Then the ML rule simplifies significantly: decide $\hat{b} = 0$ if

$$\begin{aligned} f_w(a + \sqrt{E}) &\geq f_w(a - \sqrt{E}) \\ (a + \sqrt{E})^2 - 0^2 &\leq (a - \sqrt{E})^2 - 0^2 \\ 4\sqrt{E}a &\leq 0 \\ a &\leq 0. \end{aligned}$$

In other words, the ML decision rule take the received voltage $y = a$ and estimates:

$a \leq 0 \Rightarrow 0$ was sent
Else, 1 was sent.

Figure 10 illustrates the ML decision rule when superposed on the "bits to voltage" mapping (cf. Figure 8). The decision rule picks that transmitted voltage level that is closer to the received voltage (closer in the usual sense of Euclidian distance). Hence, the maximum likelihood (ML) rule is also known as the minimum distance rule or the *nearest neighbor* rule.

In the rest of this lecture we look at two natural extensions of the material developed painstakingly so far:

1. an evaluation of the performance of the ML rule and the reliability to communication it affords. Our focus will be on understanding the relation between the energy constraint at the transmitter and the noise variance in deciding the reliability level.

2. move forward towards sending multiple bits at the same time instant. There is a natural generalization of the nearest-neighbor rule and the corresponding level of reliability to communication.

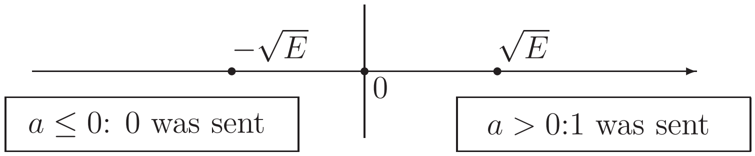

Figure 10: The ML rule superposed on Figure 8.

Reliability of Communication

The receiver makes an error if it decides that a 1 was sent when a 0 was sent, or vice versa. The *average* error is a weighted sum of the probabilities of these two types of error events, with the weights being equal to the a priori probabilities of the information bit:

$$\mathbb{P}[\mathcal{E}] = \mathbb{P}[\mathcal{E}|b=0]\,\mathbb{P}[b=0] + \mathbb{P}[\mathcal{E}|b=0]\,\mathbb{P}[b=1]. \tag{54}$$

We suppose the a priori probabilities are equal (to 0.5 each). Let us focus on one of the error events by supposing that the information bit was actually 0. Then with the nearest neighbor rule,

$$\begin{aligned} P[\mathcal{E}|b=0] &= P[\hat{b}=1|b=0] \\ &= P[y>0|b=0] \\ &= P[x+w>0|b=0] \\ &= P[w>\sqrt{E}] \\ &= Q\left(\frac{\sqrt{E}}{\sigma}\right). \end{aligned}$$

Due to the complete symmetry of the mapping from the bit values to the voltage levels and the decision rule, the probability of the other error event is also the same:

$$\begin{aligned} P[\mathcal{E}|b=1] &= P[\hat{b}=0|b=1] \\ &= P[y<0|b=1] \\ &= P[x+w<0|b=1] \\ &= P[w<-\sqrt{E}] \\ &\stackrel{\text{def}}{=} Q\left(\frac{\sqrt{E}}{\sigma}\right). \end{aligned}$$

The average probability of error is also equal to the same $Q\left(\frac{\sqrt{E}}{\sigma}\right)$.

SNR and Reliability of Communication

The first observation we make from the expression for the unreliability of communication is that it depends only on the *ratio*, of the transmit energy E and the noise variance σ^2: the error probability is

$$Q\left(\sqrt{\mathsf{SNR}}\right). \tag{55}$$

We have already seen this phenomenon before in Lecture 1, albeit in a deterministic setting. This ratio is called the *signal to noise ratio*, or simply SNR. Basically, the communication engineer can design for a certain reliability level by choosing an appropriate SNR setting. While the $Q(\cdot)$ function can be found in standard statistical tables, it is useful for the communication engineer to have a rule of thumb for how sensitive this SNR "knob" is in terms of the reliability each setting offers. For instance, it would be useful to know by how much the reliability increases if we double the SNR setting. To do this, it helps to use the following approximation:

$$Q(a) \approx \frac{1}{2}\exp\left(-\frac{a^2}{2}\right). \tag{56}$$

This approximation implies that the unreliability level

$$Q\left(\sqrt{\mathsf{SNR}}\right) \approx \frac{1}{2}e^{\frac{-\mathsf{SNR}}{2}}. \tag{57}$$

Equation (57) is saying something very interesting: it says that the SNR has an *exponential* effect on the probability of error. For instance, supposing we double the SNR setting the error probability

$$Q\left(2\sqrt{\mathsf{SNR}}\right) \approx \left(Q\left(\sqrt{\mathsf{SNR}}\right)\right)^2, \tag{58}$$

is a *square* of what it used to be before.

Transmitting Multiple Bits

Let us consider the same transmit energy constraint as before and see by how much the reliability is reduced when we transmit multiple bits in the same single time sample. As in Lecture 1, let us start with mapping the bits to voltage levels that are as far apart from each other: this is illustrated in Figure 11 for 2 bits (and hence 4 voltage levels).

The ML rule is the same nearest neighbor one: pick that transmit voltage level that is closest to the received voltage level. Figure 12 provides a short justification.

Reliability of Communication

A look at the bits-to-voltage mapping in Figure 11 suggests that the inner two voltage levels (♣ and ♠) are less reliable than the outer ones (♡ and ♢): the inner levels have neighbors

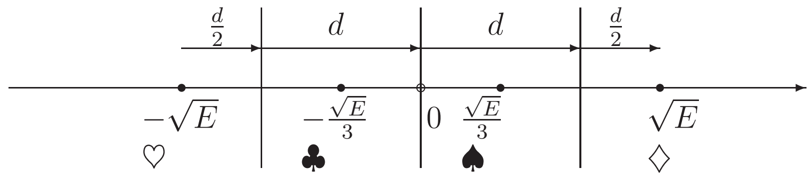

Figure 11: Sending 2 information bits across an AGN channel.

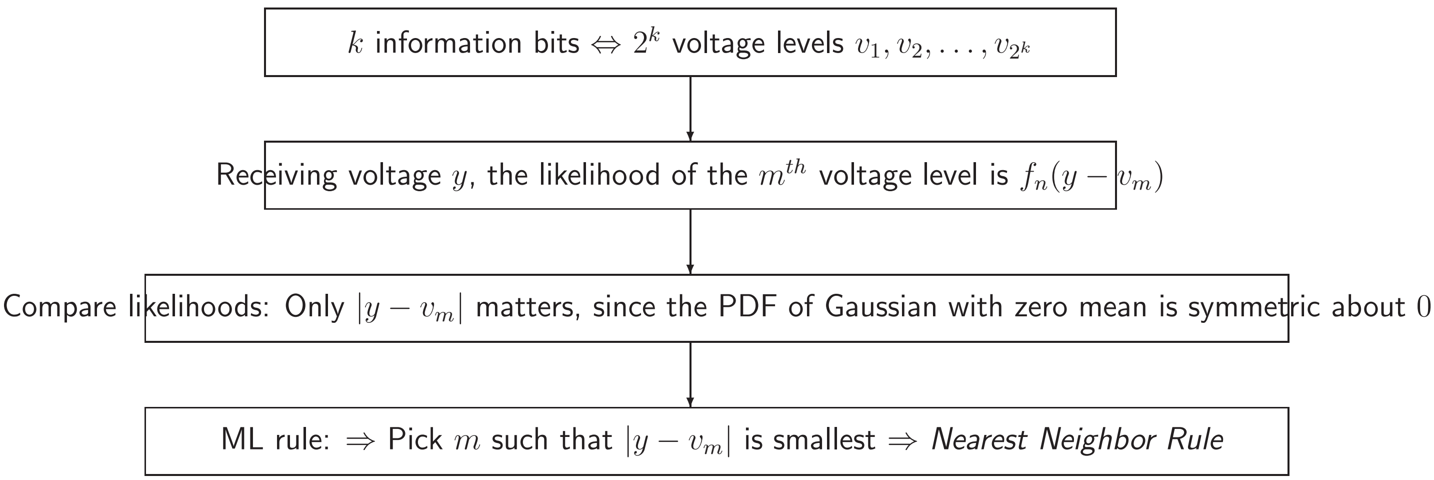

Figure 12: ML Rule for k information bits is the nearest neighbor rule.

on *both* sides while the outer ones have only one neighbor. We can calculate the probability of making an error with the ML rule given the transmission of an outer voltage level (say, the ♡) exactly as in the earlier part of this lecture:

$$\mathbb{P}\left[\mathcal{E}|\heartsuit\right] = \mathbb{P}\left[w > \frac{d}{2}\right] \tag{59}$$

$$= Q\left(\frac{\sqrt{E}}{3\sigma}\right). \tag{60}$$

On the other hand, the probability of making an error with the ML rule given the transmission of an inner voltage level (say, the ♠) is:

$$\mathbb{P}[\mathcal{E}|\spadesuit] = \mathbb{P}\left[\left\{w > \frac{d}{2}\right\} \cup \left\{w < \frac{-d}{2}\right\}\right] \tag{61}$$

$$= 2Q\left(\frac{\sqrt{E}}{3\sigma}\right). \tag{62}$$

Finally, the average probability of making an error (averaged over all the 4 different voltage levels) is

$$\mathbb{P}[\mathcal{E}] = 2 \times \frac{1}{4} \times Q\left(\frac{\sqrt{E}}{3\sigma}\right) + 2 \times \frac{1}{4} \times 2Q\left(\frac{\sqrt{E}}{3\sigma}\right) \tag{63}$$

$$= \frac{3}{2}Q\left(\frac{\sqrt{E}}{3\sigma}\right). \tag{64}$$

We have done this analysis for transmitting $k = 2$ information bits. The same analysis carries over to larger k. Indeed, the error probability is readily calculated to be, as an extension of Equation (64):

$$P_e = \left(2 - \frac{1}{2^{k-1}}\right) Q\left(\frac{d}{2\sigma}\right), \tag{65}$$

where the minimum distance between two of the equally spaced voltage levels is

$$d = \frac{2\sqrt{E}}{2^k - 1}. \tag{66}$$

As before the error probability is determined only by the SNR of the channel.

An Engineering Conclusion

At the end of Lecture 1, we noted a relationship between completely reliable communication of number of bits, transmit energy, bounds to the additive noise: the required energy to maintain the same reliability essentially quadrupled when we looked to transmit one extra bit. We have relaxed our definition of reliability, and replaced the noise bounds by a statistical quantity (an appropriate multiple of the standard deviation σ). But the essential relationship between transmit energy and the number of information bits you can reliably transmit (at any reliability level) is unchanged: a linear increase in the number of transmit bits requires an exponential increase in the required transmit energy while maintaining the same reliability level.

Looking Ahead

We notice from Equation (65) and (66) that the reliability goes down to zero as the number of information bits sent increases. This picture suggests that increasing the rate of communication invariably leads to a degradation of the reliability. We will see in the next few lectures that this is only an artifact of the specific communication scheme we have chosen and not a fundamental relation. Specifically, we will see that buffering the information bits and jointly communicating them over multiple time instants will improve the reliability significantly, as compared to sending the bits on a time-sample-by-time-sample basis. Sure, there is no free lunch: the cost paid here is the delay involved in buffering the information and then communicating them jointly.

Lecture 4: Sequential and Block Communication

Introduction

In the previous lecture, we have studied reliable communication of a bit (or a handful of bits) at a given time instant. In practice, there tend to be several hundreds of thousands of bits that need to be communicated reliably, but over multiple time instants. A natural scheme that communicates a whole bunch of information bits over multiple time instants comes is *sequential communication*:

> read the information bits serially, say k at a time, and transmit them sequentially at different time instants.

We will see that this scheme, while simple, has limitations. In particular, as time grows the reliability level approaches zero. To ameliorate this situation, we turn to *block communication*, where all the voltages at different times are picked *jointly* as a function of all the information bits. In this lecture we will see a simple block communication scheme, the so-called *repetition coding* strategy. We see that it promises arbitrarily reliable communication, but at the cost of arbitrarily small data rate and energy efficiency.

Channel Model

Corresponding to multiple transmissions, we have multiple received voltages. To figure out how to process these received voltages into a decision on the transmitted information bits, we first need a statistical model for how the additive noise varies *over time*. Our discussion in Lecture 2 lets us argue that the statistics of the additive noise at any time instant is Gaussian. Without much loss of generality, we can suppose the mean and the variance are unchanged over time. In practice, it is typically also a good assumption to suppose that the additive noise at one time instant has little relation statistically to that at the next time instant. In other words, the additive noises at different times are *statistically independent* from each other. Such a noise model is said to be *white*; this is as opposed to the "colored" noise where the value at one time instant sheds some information at another. The received voltage at time m

$$y[m] = x[m] + w[m], \tag{67}$$

is the sum of the transmitted voltage at time m and the noise at time m. Since the noise statistics are Gaussian, we will refer to this channel as the *additive white Gaussian noise* channel, or simply the AWGN channel.

Reliability of Sequential Communication

Now, we have argued earlier that information bits can be well modeled as statistically independent of one another. In sequential communication different bits are sent at different times.

This means that the transmitted voltages at different times are also statistically independent of one another. Since the additive noises at different times are statistically independent of one another, we conclude that the received voltages are also statistically independent of each other. In other words, sequential transmission over an AWGN channel naturally leads to sequential reception as well: at each time m, make a ML decision on the k bits transmitted over that time instant. This local ML rule is also the global one.

We know the reliability of communication at any given time instant: from Equation (**??**) of Lecture 3, the average error probability is

$$p \stackrel{\mathrm{def}}{=} \left(2 - \frac{1}{2^{k-1}}\right) Q\left(\frac{\sqrt{\mathsf{SNR}}}{2^k - 1}\right). \tag{68}$$

Here SNR is the ratio between the transmit energy E and the noise variance σ^2. Due to the statistical independence of these error events, the probability of communicating every bit reliably at *every* time instant for n consecutive time instants is

$$(1-p)^n. \tag{69}$$

Now we get a hint of what might go wrong with sequential communication: even though we might have ensured that the reliability is pretty high (p is very small) at any given time instant, the overall reliability of communicating correctly at *each and every* time instant is quite slim. To get a concrete feel, let us turn to an example: $k = 4$ and $n = 250$. Suppose we had set the error probability (p) at any time instant to be 10^{-4} (based on Equation (68), how much should the SNR be?). Then the reliability of the entire sequential communication process is, from Equation (69),

$$\left(1 - 10^{-4}\right)^{250} \approx 0.75. \tag{70}$$

In other words, there is a 1 in 4 chance that at least one of the 1000 bits is wrongly communicated. For most engineering applications, this is a unreasonably high level of unreliability (imagine if every 1 of your 4 downloads – blogs, music, whatever – didn't work!). The only way to compensate for this is to set the SNR so high that the overall reliability is large enough. Typical reliability levels (for wireline communication) is of the order of 10^{-10} or so. For large enough files this would mean astronomical SNR values.

Is this the price for reliable communication? Successful communication technologies (wires, wireless) around us provide a solid clue to the answer; surely, there must be a way out since we do manage to communicate reliably enough. The trouble is with the sequential nature of transmission and reception. The key lies in moving to *block communication*: buffering all the information bits and jointly decide what voltage levels we will transmit at the multiple time instants (this process is known as *coding*) and have the receiver make a deliberate decision on all the information bits together, after seeing the entire set of received voltages (this process is known as *decoding*). In this lecture, we will start out with a simple coding scheme: the so-called repetition coding strategy.

Repetition Coding

In this simple scheme, we transmit the *same* voltage level (say x) at *each* time instant, for say n consecutive time instants. The entire set of possible information bits is mapped to different voltage levels (in the context of the previous example, this corresponds to 2^{1000} levels!). The actual voltage level transmitted is based on the values of all the information bits. The received voltages are

$$y[m] = x + w[m], \quad m = 1 \dots n. \tag{71}$$

How should the receiver make its decision on what (common) voltage level was transmitted? In other words, what does the ML rule look like now? To answer this question, it helps to go to a more general coding scheme and see what the ML rule is. Once we have a general principle at hand, it is easy to specialize it to the current scheme of interest: repetition coding.

Vector ML Rule

Suppose we have a total of B bits to communicate over n time instants. (The rate of such a communication is said to be B/n.) Consider a *coding* scheme that maps these B bits to 2^B possible *vector* voltage levels: $\mathbf{v}_1, \dots, \mathbf{v}_{2^B}$. There are 2^B such levels since that is the number of possible realizations B bits can take. The image of the mapping is a vector because we need to know what voltage levels to transmit at different time instants. It is convenient to collect together the n different voltage levels to transmit at the n time instants as a vector. Now the ML rule involves, from Lecture 3, calculating the *likelihood* as a function of the n received voltage levels at the n time instants (it is useful to collect these together as well and represent them as an n-dimensional vector $\mathbf{y}$). We compare the likelihoods and pick the largest one and use it to decode the transmitted information bits.

Suppose the received voltage vector $\mathbf{y}$ is equal to $\mathbf{a}$, i.e., voltages $a(1), \dots, a(n)$ are received at the n time instants. Directly from Equation (50) of Lecture 3, we see that the likelihood for the k^{th} transmit voltage vector is (up to a scaling constant)

$$\begin{aligned} L_k &= f_{\mathbf{y}}(\mathbf{a}|\mathbf{x} = \mathbf{v}_k) && (72)\\ &= \Pi_{m=1}^{n} f_{y[m]}\left(a(m)|x[m] = \mathbf{v}_k(m)\right) && (73)\\ &= \Pi_{m=1}^{n} f_{w[m]}\left(a(m) - \mathbf{v}_k(m)\right). && (74) \end{aligned}$$

Here we used the statistical independence of the additive noises at the different time instants in arriving at Equation (73). Using the explicit PDF of the Gaussian noise statistics, we can simplify the likelihood even further:

$$L_k = \left(\frac{1}{(2\pi\sigma^2)^{\frac{n}{2}}}\right) \exp\left(-\frac{1}{2\sigma^2}\left(\sum_{m=1}^{n} |a(m) - \mathbf{v}_k(m)|^2\right)\right). \tag{75}$$

So, picking the index that has the largest likelihood L_k simply corresponds to picking the index that minimizes the *Euclidean* squared distances:

$$\|\mathbf{a} - \mathbf{v}_k\|^2 \stackrel{\text{def}}{=} \sum_{m=1}^{n} |a(m) - \mathbf{v}_k(m)|^2, \quad k = 1, \ldots, 2^B. \tag{76}$$

So, the ML rule continues to be a nearest-neighbor rule with an appropriately defined notion of distance: the Euclidean distance between the set of received voltages collected together as a vector and the set of transmitted voltages collected together as a vector.

ML Rule for Repetition Coding

We are now ready to study the ML rule with repetition coding at the transmitter. Figure 13 illustrates the constellation diagram and the nearest-neighbor ML rule for the special case of $B = n = 2$.

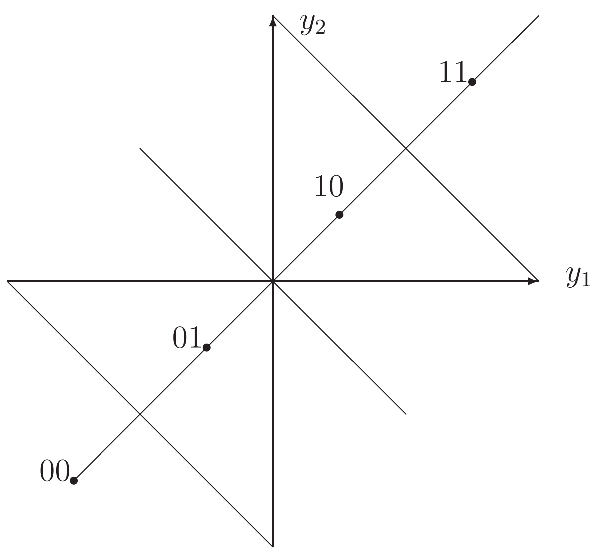

Figure 13: Constellation diagram and ML rule for repetition coding; here $B = n = 2$.

We can simplify the ML rule some more by taking a careful look at the structure of the transmit voltage vectors; they are all now of the form

$$\mathbf{v}_k = v_k \mathbf{1}, \tag{77}$$

where $\mathbf{1}$ is the vector of all ones and v_k is a single voltage level among the 2^B equally spaced voltages between $-\sqrt{E}$ and $\sqrt{E}$. Thus, the Euclidean distance between a received voltage vector $\mathbf{a}$ and $\mathbf{v}_k$ is

$$\begin{aligned}\|\mathbf{a}-\mathbf{v}_k\|^2 &= \sum_{m=1}^{n}(a(m)-v_k)^2 &(78)\\ &= nv_k^2+\left(\sum_{m=1}^{n}a(m)^2\right)+-2v_k\left(\sum_{m=1}^{n}a(m)\right). &(79)\end{aligned}$$

When comparing these distances, the constant term ($\sum_{m=1}^{n}a(m)^2$) cancels off. Thus it only suffices to pick the smallest of the terms

$$nv_k^2-2v_k\left(\sum_{m=1}^{n}a(m)\right),\quad k=1\dots 2^B. \tag{80}$$

A Sufficient Statistic

We observe from Equation (80) that the sum of the received voltages

$$\left(\sum_{m=1}^{n}a(m)\right) \tag{81}$$

is sufficient to evaluate the ML rule for repetition coding. In other words, even though we received n different voltage levels, only their sum is relevant to making a decision based on the ML rule. Such a quantity is called a *sufficient statistic*; we say that the sum of the received voltages is a sufficient statistic to derive the ML rule for repetition coding.

Reliability of Communication with Repetition Coding

The sufficient statistic, or the sum of received voltages, can be written as

$$\sum_{m=1}^{n}y[m]=nx+\sum_{m=1}^{n}w[m]. \tag{82}$$

Denoting the average of the received voltage by $\tilde{y}$, we see that

$$\begin{aligned}\tilde{y} &\stackrel{\text{def}}{=} \frac{1}{n}\sum_{m=1}^{n}y[m] &(83)\\ &= x+\tilde{w}, &(84)\end{aligned}$$

where

$$\tilde{w} \stackrel{\text{def}}{=} \frac{1}{n} \sum_{m=1}^{n} w[m] \tag{85}$$

is additive noise that is zero mean and Gaussian with variance $\frac{\sigma^2}{n}$. Here x is one of the 2^B equally spaced voltage levels between $-\sqrt{E}$ and $\sqrt{E}$. The error probability of detecting such an x is, directly from Equation (68) (replacing σ^2 by $\frac{\sigma^2}{n}$ and k by B),

$$p^{\text{rep}} \stackrel{\text{def}}{=} \left(2 - \frac{1}{2^{B-1}}\right) Q\left(\frac{\sqrt{n\mathsf{SNR}}}{2^B - 1}\right). \tag{86}$$

The rate of communication is

$$R^{\text{rep}} \stackrel{\text{def}}{=} \frac{B}{n} \tag{87}$$

bits per unit time instant. We can now make the following observations.

1. The error probability increases to 1 (as $n \to \infty$) for any non-zero rate R^{rep} (this means that B increases linearly with n).

2. The only way to drive the error probability to zero (very high reliability) is by having a rate that is going to zero. Specifically, if the number of information bits B is such that

$$\lim_{n\to\infty} \frac{B}{\log n} = 0, \tag{88}$$

then

$$\lim_{n\to\infty} \frac{\sqrt{n\mathsf{SNR}}}{2^B - 1} = \infty. \tag{89}$$

Substituting this in Equation (86) and using the fact that

$$\lim_{a\to\infty} Q(a) = 0, \tag{90}$$

we conclude that the unreliability of repetition coding becomes arbitrarily small:

$$\lim_{n\to\infty} p^{\text{rep}} \to 0. \tag{91}$$

3. The *energy per bit* of a scheme is the ratio of the energy consumed to the number of bits communicated:

$$\frac{nE}{B}, \tag{92}$$

in this case. The smaller this value, the more energy efficient the scheme is. We see that repetition block coding is not energy efficient at small reliability levels. In particular, if we desire arbitrarily reliable communication this comes at the high price of arbitrarily large energy per bit.

Looking Ahead

We have seen that the unreliability of communication becomes arbitrarily worse with sequential transmission schemes. To ameliorate this situation, we considered a simple block communication scheme: repetition coding. Now the reliability can be made arbitrarily good, but at the cost of diminishing data rate of communication and poor energy efficiency. In the next couple of lectures we will see that smarter coding techniques will resolve this deficiency as well: we can get arbitrarily reliable communication with non-zero communication rate and finite energy efficiency.

Lecture 5: Energy Efficient Communication

Introduction

So far, we have seen that block communication (using the simple repetition coding) can improve the reliability of communication significantly, over and above that possible with sequential communication. This is particularly true when we want to communicate a large amount of data. But this has come at a high cost: specifically we can get arbitrarily reliable communication, but

- the data rate (number of bits per unit time sample) goes to zero. Specifically, we know from Lecture 4 that the data rate is

$$\frac{B}{n} = o\left(\frac{\log_2 n}{n}\right), \tag{93}$$

  where we used the notation $o(n)$ to denote a function of n that has the property that

$$\lim_{n\to\infty} o(n) = 0. \tag{94}$$

  Simply put we can think of the data rate of reliable communication with repetition coding as approximately

$$\frac{\log_2 n}{n} \tag{95}$$

  which is very small for large n. For a large data packet (of size, say B), we need an amount of time approximately 2^B to communicate it reliably using repetition coding!

- it is very energy inefficient. Here, we have defined the energy efficiency is defined as the amount of energy (in Joules) consumed per bit that is reliably communicated. In the repetition coding scheme, using n time instants we are using a total energy nE. Further, we need 2^B time samples to send B bits reliably. So we use up energy proportional to 2^B and thus the energy efficiency is

$$\frac{2^B E}{B}. \tag{96}$$

  For large data size B, this goes to infinity: the repetition coding scheme is hardly energy efficient.

In summary,

> repetition coding significantly improved the reliability of communication over sequential communication, particularly for large data packets, but at the cost of zero data rate and zero energy efficiency.

This is in stark contrast to sequential communication that had *non-zero* data rate and energy efficiency: after all, we keep transmitting new information bits at every time sample (so the data rate is non-zero) and we only use a finite energy at any time sample (so the energy efficiency is also non-zero).

Question: Can we have the desirable features of sequential communication, non-zero data rate and energy efficiency, while ensuring that the reliability is very good? In other words, is there a free breakfast, lunch and dinner?

Well, the short answer is *yes*. The long answer is that the block communication scheme that does it is quite involved. It *actually* is "rocket-science" (almost!). We will spend several lectures trying to figure out what it takes to reliably communicate at non-zero data rates and non-zero energy efficiency. It is a remarkable success story that has drawn various aspects of electrical engineering: role of modeling, asking the right questions, mathematical abstraction, mathematical legerdemain, algorithmic advances and finally advances in circuits to handle the computational complexity of the algorithms involved.

We take a small step towards this story in this lecture by focusing on reliably communicating at non-zero energy efficiency (while not worrying about non-zero data rate).

Peak Energy vs Average Energy Constraints

Before we introduce our energy-efficient communication scheme, it helps to consider a slightly different type of energy constraint. So far, our constraint has been that our voltage level at any time sample is bounded in magnitude (i.e., it has to be within a limit of $\pm\sqrt{E}$ Volts in our previous notation). This comes from a constraint on the instantaneous energy at every time sample. In several communication scenarios, the constraint is on the *average* energy, averaged over many time instants. This is the constraint whenever an electronic device is rated in Watts (unit of power) measured in Joules/second. For instance, we could restrict the total power transmitted to P and this would mean that the sum of the square of the voltages transmitted in n time samples is no more than nP. In this lecture, we will focus on reliable block communication when there is a power constraint (or an average energy constraint), as opposed to instantaneous (or peak) energy constraint.

Suppose we want to transmit B bits reliably and energy efficiently. This means that we want to use a finite amount of energy per bit transmitted: so the total energy allowed to transmit bits is directly proportional to B. Let us denote the energy allowed per bit to be $\mathcal{E}_b$ (a finite value), so the total energy allowed is $B\mathcal{E}_b$.

Since the data rate is not the focus here, let us consider using a lot of time samples to transmit the bits (just as we needed to do in the repetition coding scheme). Specifically, suppose we use 2^B time instants (same order of time as in repetition coding). The data rate is surely very small for large B, but as we said, let us not worry about that now. Let us number the time samples from 0 through $2^B - 1$. Now every data packet with B bits can be made to correspond *exactly* to an integer between 0 and $2^B - 1$: one way to do this is think of the data packet (string of bits) as the binary expansion of an integer. That integer is surely somewhere between 0 and $2^B - 1$.

Now we see the possible reason for choosing the number of time samples *equal* to the number of different possible realizations (2^B) of B bits. It is to allow for the following *position* modulation scheme: look at the information packet of B bits and convert it to an integer between 0 and $2^B - 1$, say denoted by k. Now we transmit nothing (or zero in mathematical terms) at all *except* the k^{th} time samples when we use up all the energy we have: we do this by transmitting the voltage $+\sqrt{B\mathcal{E}_b}$.

This type of modulation is very different from the types of modulation schemes seen in earlier lectures: here the information is in the *position* of a large voltage rather than the specific *amplitude* of a voltage. As such, this type of communication is referred to as position modulation and is in contrast to the earlier amplitude modulation schemes. In terms of the transmit voltage vector (of length 2^B), it looks like a vector

$$\left(0, 0, \ldots, \sqrt{B\mathcal{E}_b}, 0, \ldots, 0\right) \tag{97}$$

where the only non-zero entry $B\mathcal{E}_b$ is at the k^{th} location in the vector. Figure illustrates the position modulation scheme for $B = 2$.

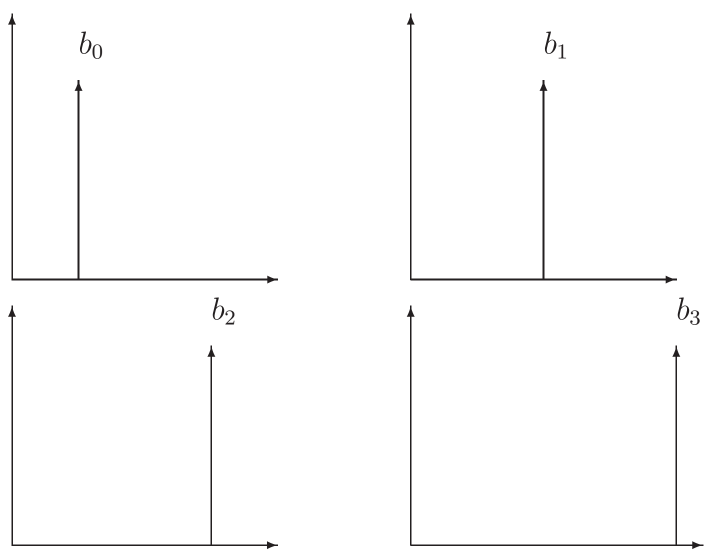

Figure 14: Position modulation: horizontal axis represents time samples and the vertical axis the voltages transmitted.

ML Rule

How do we expect the receiver to decide on which position the voltage might have been sent? Clearly, taking the average of the voltages (like we did earlier for repetition coding) is not going to help. A natural idea is that since Gaussian noise is more likely to be small (near its mean of zero) than large, we just pick that time when the received voltage is the largest. This very intuitive rule is indeed what the ML rule also is. We work this out below, for completeness and as a formal verification of our engineering intuition.

The receiver receives 2^B voltages

$$y[m] = x[m] + w[m], \quad m = 1, 2, \ldots, 2^B. \tag{98}$$

The ML rule involves calculating the likelihood of the voltages received for each possible position of where the non-zero voltage was transmitted. Suppose $k = 1$ is the index where the non-zero voltage is transmitted, i.e., the transmit voltage vector is

$$x[m] = \begin{cases} \sqrt{B\mathcal{E}_b} & m = 1 \\ 0 & \text{else.} \end{cases} \tag{99}$$

The likelihood of receiving the voltages $a_1, \ldots a_{2^B}$ is

$$\begin{aligned} L_1 &= f_{w_1}\left(a_1 - \sqrt{B\mathcal{E}_b}\right) f_{w_2}\left(a_2\right) \ldots f_{w_{2^B}}\left(a_{2^B}\right) && (100) \\ &= \left(\frac{1}{\sqrt{2\pi\sigma^2}}\right)^{2^B} \exp\left(-\frac{\left(a_1 - \sqrt{B\mathcal{E}_b}\right)^2 + \sum_{m=2}^{2^B} a_m^2}{2\sigma^2}\right) && (101) \\ &= \left(\frac{1}{\sqrt{2\pi\sigma^2}}\right)^{2^B} \exp\left(-\frac{\sum_{m=1}^{2^B} a_m^2}{2\sigma^2} - \frac{B\mathcal{E}_b}{2\sigma^2}\right) \exp\left(\frac{a_1\sqrt{B\mathcal{E}_b}}{\sigma^2}\right). && (102) \end{aligned}$$

By the symmetry of the modulation scheme in the transmit index k, we have (following Equation (102))

$$L_k = \left(\frac{1}{\sqrt{2\pi\sigma^2}}\right)^{2^B} \exp\left(-\frac{\sum_{m=1}^{2^B} a_m^2}{2\sigma^2} - \frac{B\mathcal{E}_b}{2\sigma^2}\right) \exp\left(\frac{a_k\sqrt{B\mathcal{E}_b}}{\sigma^2}\right). \tag{103}$$

The ML rule picks that index k which has the largest likelihood L_k. The first two terms in the formula (cf. Equation (103)) for L_k are *independent* of the index k. So, we can focus on just the third term. There we see that maximizing it is simply a matter of picking k such that a_k is the largest.

Reliability with Position Modulation

Suppose, again, that the index k where the non-zero voltage was transmitted is 1. Since there is complete symmetry of the modulation scheme with respect to k, we can just calculate the probability of error for this value of k and it will be the same unreliability level for all other choices. Now, the ML rule makes a mistake whenever at least one of the received voltages $a_2, \ldots, a_{2^B}$ is larger than a_1. Denoting the event

$$\text{Error Event}_{1j} = \{a_j > a_1\}, \quad j = 2 \ldots 2^B \tag{104}$$

we see that the error event when $k = 1$ is their *union*:

$$\text{Error} = \bigcup_{j=2}^{2^B} \text{Error Event}_{1j}. \tag{105}$$

It turns out that the probability of the error event is somewhat complicated to calculate directly. We can find an upper bound to it easily enough though, which itself will be easier to evaluate.

The probability of making an error is now upper bounded by the sum of the probabilities of the *pair-wise* events: indeed,

$$\mathbb{P}[\text{Error}] \leq \sum_{j=2}^{2^B} \mathbb{P}[\text{Error Event}_{1j}]. \tag{106}$$

The idea is that if we can bound the right-hand side of this equation by a small enough number, then the unreliability level of communication with position modulation itself is less than that small enough number. Such a way of bounding the error probability is known as the *union bound*.

How do we calculate the pair-wise error event probability $\mathbb{P}[\text{Error Event}_{1j}]$? We only need to focus on what happens at the time samples 1 and j. The received voltages at these two time samples look as follows:

$$y[1] = \sqrt{B\mathcal{E}_b} + w[1], \tag{107}$$

$$y[j] = w[j]. \tag{108}$$

Now the pair-wise error probability is

$$\mathbb{P}[\text{Error Event}_{1j}] = \mathbb{P}[y[j] > y[1]] \tag{109}$$

$$= \mathbb{P}\left[w[j] - w[1] > \sqrt{B\mathcal{E}_b}\right]. \tag{110}$$

Observe that the difference of two independent Gaussian noises (with the same mean and variance) also has Gaussian statistics but has *twice* the variance as the original ones. So, the difference $w[j] - w[1]$ is Gaussian with zero mean and variance $2\sigma^2$. Now we have a simple expression for the pair-wise error probability: continuing from Equation (110)

$$\mathbb{P}[\text{Error Event}_{1j}] = Q\left(\frac{\sqrt{B\mathcal{E}_b}}{\sqrt{2\sigma^2}}\right), \quad j = 2 \ldots 2^B. \tag{111}$$

We can substitute Equation (111) in Equation (106) to arrive at an upper bound to the unreliability level of communication with position modulation:

$$\mathbb{P}\left[\text{Error}\right] \leq \left(2^B - 1\right) Q\left(\frac{\sqrt{B\mathcal{E}_b}}{\sqrt{2\sigma^2}}\right). \tag{112}$$

Using the usual upper bound to the $Q(\cdot)$ function we can get a further upper bound to the error probability:

$$\begin{aligned}
\mathbb{P}\left[\text{Error}\right] &< \left(2^B - 1\right) \frac{1}{2} \exp\left(-\frac{B\mathcal{E}_b}{4\sigma^2}\right) &(113)\\
&< 2^B \exp\left(-\frac{B\mathcal{E}_b}{4\sigma^2}\right) &(114)\\
&= \exp\left(B \log_e 2 - \frac{B\mathcal{E}_b}{4\sigma^2}\right) &(115)\\
&= \exp\left(-B\left(\frac{\mathcal{E}_b}{4\sigma^2} - \log_e 2\right)\right). &(116)
\end{aligned}$$

So, is the unreliability small for large packet sizes B? The answer depends on how large the energy per bit $\mathcal{E}_b$ we invest in: if it is large enough:

$$\mathcal{E}_b > 4\sigma^2 \log_e 2, \tag{117}$$

then for large values of B the probability of error goes to zero.

Reprise

We have seen a very different type of block communication, position modulation, that is arbitrarily reliable *and* energy efficient. It came about by relaxing the instantaneous energy constraint to an *average* energy constraint. A few key questions arise naturally at this point.

1. In engineering practice, it may not be possible to transmit a large voltage (as the packet size B grows, the voltage magnitude also grows without bound). Indeed, most electronic devices come with both an average *and* peak power rating. If the peak power allowed is finite, the pulse modulation scheme described here will not work anymore. In this case, there are no known simple ways to get energy efficient reliable communication and we will address this issue in the lectures to come.

2. Is there something fundamental about the threshold for energy per bit given in Equation (117)? Perhaps there are other schemes that promise arbitrarily reliable communication and yet allow lower energy per bit than the threshold in Equation (117)?

(a) It turns out that the threshold in Equation (117) can be lowered by a factor of 2 by doing a more nuanced calculation of the error probability (as compared to the crude union bound used in Equation (106)).

(b) It turns out that the improved threshold of half of that in Equation (117) is truly fundamental:

> any communication scheme promising arbitrarily reliable communication over an AWGN channel has to expend energy per bit of *at least* $2\sigma^2 \log_e 2$.

In this sense $2\sigma^2 \log_e 2$ is a fundamental number for reliable communication over the AWGN channel. We will get more intuition on where this comes from shortly.

Apart from these aspects, position modulation is important on its own right.

- We will see position modulation shows up naturally in *deep space*communication (where data rate is much less an issue than energy efficiency). Deep space communication covers both satellite communication and earth's communication with remote interplanetary space missions.

- It is conjectured (based on experimental data) that the human nervous system communicates using position. The book

  > *Spikes: Exploring the Neural Code* by Fred Rieke, David Warland, Rob de Ruyter van Steveninck, and William Bialek, MIT Press, 1999,

  provides reading material in this direction.

Looking Ahead

We have delivered on one of the free food items promised earlier: reliable communication in an energy efficient manner. But this still entailed very small data rates. In the next lectures, we will see what it takes to do arbitrarily reliable and energy efficient communication with non-zero rates as well.

Lecture 6: Rate Efficient Reliable Communication

Introduction

We now move to rate efficient reliable communication (energy efficiency tends to come for free in this scenario). In this lecture we see that there are block communication schemes smarter than the naive repetition coding seen earlier that promise arbitrarily reliable communication while still having a non-zero data rate. We begin by setting the stage for studying rate efficient reliable communication by carefully dividing the transmitter strategy of mapping the information bits to transmit voltages into two distinct parts:

1. maps information bits into *coded* bits by adding more redundancy: the number of coded bits is larger than the number of information bits and the ratio is called the *coding rate*. This process is generally called *coding*.

2. map coded bits directly into transmit voltages. This is done *sequentially*: for instance, if only two transmit voltages are allowed ($\pm\sqrt{E}$) then every coded bit is sequentially mapped into one transmit voltage. If four possible transmit voltages are allowed ($\pm\sqrt{E}, \pm\frac{\sqrt{E}}{3}$), then every two consecutive coded bits are mapped into a single transmit voltage sequentially. This mapping is typically called *modulation* and can be viewed as a labeling of the discrete transmit voltages with a binary sequence.

The receiver could also be potentially broken down into two similar steps, but in this lecture we will continue to focus on the ML receiver which maps the received voltages *directly* into information bits. Focusing on a simple binary modulation scheme and the ML receiver, we see in this lecture that there are plenty of good coding schemes: in fact, we will see that *most* coding schemes promise arbitrarily reliable communication provided they are decoded using the corresponding ML receiver!

Transmitter Design: Coding and Modulation

We are working with an energy constraint of E, so the transmit voltage is restricted to be within $\pm\sqrt{E}$ at each time instant. For simplicity let us restrict that only two transmit voltages are possible: $+\sqrt{E}$ and $-\sqrt{E}$.[2]

If we are using T time instants to communicate, this means that the number of *coded* bits is T, one per each time instant. With a coding rate of R, the number of *information* bits (the size of the data packet) is $B = RT$. Surely, $R \leq 1$ in this case. The scenario of $R = 1$ exactly corresponds to the *sequential* communication scheme studied in Lecture 4. As we saw there, the reliability level approaches zero for large packet sizes. The point is that

[2]We will explore the implications of this restriction in a couple of lectures from now.

even though we have spaced the transmit voltages far enough apart (the spacing is $2\sqrt{E}$ in this case), the chance that at least one of the bits is decoded incorrectly approaches unity when there are a lot of bits. The idea of introducing redundancy between the number of information bits and coded bits (by choosing $R < 1$) is to ameliorate exactly this problem.

Linear Coding

As we have seen, coding is an operation that maps a sequence of bits (information bits, specifically) to a *longer* sequence of bits (the coded bits). While there are many types of such mappings, the simplest one is the *linear code*. This is best represented mathematically by a matrix C whose elements are drawn from $\{0, 1\}$:

$$\begin{bmatrix} \text{vector of} \\ \text{coded bits} \end{bmatrix} = C \begin{bmatrix} \text{vector of} \\ \text{information} \\ \text{bits} \end{bmatrix}. \tag{118}$$

Here the vector space operations are all done on the binary field $\{0, 1\}$: i.e., multiplication and addition in the usual modulo 2 fashion. The dimension of the matrix C is $T \times RT$ and it maps a vector of dimension $RT \times 1$ (the sequence of information bits) into a vector of dimension $T \times 1$ (the sequence of coded bits). The key problem is to pick the appropriate code C such that the unreliability with ML decoding at the receiver is arbitrarily small. It turns out that *almost all* matrices C actually have this property! We discuss this in detail next, but could be skipped in a first course on communication. It is important to know the punchline though; so, please read the last the last two sections, titled "Reliable Rate of Communication" and "Looking Ahead".

A Novel Approach

To study the construction of appropriate coding marries C, we will consider the set $\mathcal{C}$ of *all possible* binary matrices C: there are 2^{RT^2} number of them (each entry of the matrix can be 0 or 1 and there are RT^2 entries in the matrix). We will show that the average unreliability, averaged over all the matrices C:

$$\overline{\mathbb{P}[\mathcal{E}]} \stackrel{\text{def}}{=} \frac{1}{2^{RT^2}} \sum_{C \in \mathcal{C}} \mathbb{P}[\mathcal{E}|C], \tag{119}$$

is arbitrarily small for large packet sizes B (and hence large time T). In Equation (119) we have used the notation $\mathbb{P}[\mathcal{E}|C]$ to denote the unreliability of communication with the appropriate ML receiver over the AWGN channel *when using the code* C at the transmitter.[3]

[3]Keep in mind that the ML receiver will, of course, depend on the code C used at the transmitter.

If $\overline{\mathbb{P}[\mathcal{E}]}$ is arbitrarily small, then most code matrices C must have an error probability that is also arbitrarily small. In fact, only at most a polynominal (in RT) number of codes can have poor reliability.

Calculating Average Unreliability

This unreliability level is the average unreliability experienced, averaged over all possible information bit sequences:

$$\mathbb{P}[\mathcal{E}|C] = \frac{1}{2^{RT}} \sum_{k=1}^{2^{RT}} \mathbb{P}[\mathcal{E}|B_k, C], \tag{120}$$

where B_k is the k^{th} information packet B (there are $2^B = 2^{RT}$ possible information packets). The error event $\mathcal{E}$ occurs when the likelihood of the T received voltages is larger for some other packet B_j with $j \neq k$. The probability of this event will depend on the nature of the code C and is, in general, quite complicated to write down precisely. As in the previous lecture, we will use the union bound to get an upper bound on this unreliability level:

$$\mathbb{P}[\mathcal{E}|B_k, C] < \sum_{j \neq k, j=1}^{2^{RT}} \mathbb{P}_2[B_k, B_j|C], \tag{121}$$

where we have denoted $\mathbb{P}_2[B_k, B_j|C]$ as the probability of mistakenly concluding that B_j is the information packet when actually B_k was transmitted using the code C.

Substituting Equations (121) and (120) into Equation (119) we get

$$\overline{\mathbb{P}[\mathcal{E}]} < \frac{1}{2^{RT}} \sum_{k=1}^{2^{RT}} \sum_{j \neq k, j=1}^{2^{RT}} \left(\frac{1}{2^{RT^2}} \sum_{C \in \mathcal{C}} \mathbb{P}_2[B_k, B_j|C] \right). \tag{122}$$

The Key Calculation

We now come to the key step in this whole argument: *evaluating* the value of the expression

$$\frac{1}{2^{RT^2}} \sum_{C \in \mathcal{C}} \mathbb{P}_2[B_k, B_j|C]. \tag{123}$$

From our derivation in Lecture 4, we know that the error probability

$$\mathbb{P}_2[B_k, B_j|C] = Q\left(\frac{d}{2\sigma}\right) \tag{124}$$

where d is the Euclidean distance between the vectors of transmitted voltages corresponding to the information packets B_k, B_j using the code C. Since we have only a binary transmit voltage choice, the distance d simply depends on the number of time instants where the *coded bits* corresponding to the information packets B_k, B_j using the code C are *different*. Suppose the coded bits are different at ℓ of the possible T locations. Then the Euclidean distance squared

$$d^2 = \ell 4E. \tag{125}$$

The idea is that for each time where the coded bits are different, the square of the distance is $4E$ (since the corresponding transmit voltages are $\pm\sqrt{E}$ leading to a distance of $2\sqrt{E}$). So we can write the key expression in Equation (123) using Equations (124) and (125) as

$$\frac{1}{2^{RT^2}} \sum_{C \in \mathcal{C}} \mathbb{P}_2 \left[B_k, B_j | C \right] = \sum_{\ell=0}^{T} f_\ell^{(k,j)} Q \left(\frac{\sqrt{\ell E}}{\sigma} \right). \tag{126}$$

Here we have denoted $f_\ell^{(k,j)}$ as the fraction of the codes that lead to exactly ℓ mismatches in the T coded bits for the pair of information packets B_k and B_j.

While the value of ℓ depends on the code C and the pair of packets B_k, B_j, we can calculate $f_\ell^{(k,j)}$ readily. In fact, it is the *same* for all pair of information packets B_k, B_j. Towards this calculation, let us focus on the coded bit at any specific time instant, say the first one. Since we are considering all possible codes C, there is no special bias towards this bit being a 1 or a 0. In other words, the fraction of codes that result in this coded bit being 1 (or 0) is exactly 0.5. This is true for every information packet. Furthermore, there is no special bias for this coded bit to be the same or different for any pair of information packets (B_k and B_j in our notation). In other words, the fraction of codes for which the coded bit of interest (say, the first one out of the total T) is the same for any pair of information packets B_k, B_j is exactly 0.5. So, the fraction of codes that lead to exactly ℓ mismatches in the T coded bits for any pair of information packets B_k, B_j is

$$f_\ell^{(k,j)} = \left(\begin{array}{c} T \\ \ell \end{array} \right) \left(\frac{1}{2} \right)^T. \tag{127}$$

The fraction is the same for all pairs (k, j). To elaborate a bit more on how we arrived at the expression in Equation (127), we see that:

1. there are $\left(\begin{array}{c} T \\ \ell \end{array} \right)$ possible ways of choosing the ℓ mismatches in a sequence of T coded bits;

2. the term $\left(\frac{1}{2}\right)^T$ is simply a normalization term – the sum of all the fractions of codes

that lead to exactly ℓ mismatches, summed over all ℓ, should be unity:

$$\sum_{\ell=0}^{T} f_\ell^{(k,j)} = 1. \tag{128}$$

Using the binomial formula

$$\sum_{\ell=0}^{T} \binom{T}{\ell} = 2^T \tag{129}$$

we see that the normalization term has to be $\left(\frac{1}{2}\right)^T$.

Now substituting Equation (127) in Equation (126) we arrive at

$$\frac{1}{2^{RT^2}} \sum_{C \in \mathcal{C}} \mathbb{P}_2 \left[B_k, B_j | C\right] = \sum_{\ell=0}^{T} \binom{T}{\ell} \left(\frac{1}{2}\right)^T Q\left(\frac{\sqrt{\ell E}}{\sigma}\right). \tag{130}$$

We can use the upper bound on the $Q(\cdot)$ function

$$Q(a) \leq \frac{1}{2} e^{\frac{-a^2}{2}}, \quad a \geq 0, \tag{131}$$

in Equation (130) to write

$$\frac{1}{2^{RT^2}} \sum_{C \in \mathcal{C}} \mathbb{P}_2 \left[B_k, B_j | C\right] < \sum_{\ell=0}^{T} \binom{T}{\ell} \left(\frac{1}{2}\right)^T e^{-\frac{\ell E}{2\sigma^2}}. \tag{132}$$

Using the binominal formula

$$\sum_{\ell=0}^{T} \binom{T}{\ell} e^{-\frac{\ell E}{2\sigma^2}} = \left(1 + e^{-\frac{E}{2\sigma^2}}\right)^T \tag{133}$$

in Equation (132) we come to

$$\frac{1}{2^{RT^2}} \sum_{C \in \mathcal{C}} \mathbb{P}_2 \left[B_k, B_j | C\right] < \left(1 + e^{-\frac{E}{2\sigma^2}}\right)^T \left(\frac{1}{2}\right)^T \tag{134}$$

$$= 2^{T \log_2 \left(1 + e^{-\frac{E}{2\sigma^2}}\right)} \left(\frac{1}{2}\right)^T \tag{135}$$

$$= 2^{-TR^*} \tag{136}$$

where we have defined

$$R^* \stackrel{\text{def}}{=} \log_2 \left(\frac{2}{1 + e^{-\frac{E}{2\sigma^2}}}\right) \tag{137}$$

$$= 1 - \log_2 \left(1 + e^{-\frac{E}{2\sigma^2}}\right). \tag{138}$$

Reliable Rate of Communication

The denouement is near now. Substituting Equation (136) into Equation (122) we see the average unreliability, averaged over all possible linear codes C is no more than

$$\overline{\mathbb{P}\left[\mathcal{E}\right]} < \frac{1}{2^{RT}} \sum_{k=1}^{2^{RT}} \sum_{j\neq k, j=1}^{2^{RT}} 2^{-R^*T} \tag{139}$$

$$< 2^{RT} 2^{-R^*T}. \tag{140}$$

As long as the data rate R is strictly less than R^* (defined in Equation (138)), the average unreliability level becomes arbitrarily small for large T (and hence large information packet sizes). The term R^* is a simple function of the SNR (defined, as usual, to be the ratio of E to σ^2) and is strictly positive. It can be interpreted as the threshold below which arbitrary reliable communication data rates are possible. Further more, this performance is guaranteed by many linear codes C when coupled with the appropriate ML decoder at the receiver end.

Finally, any linear code C that guarantees reliable communication is also energy efficient. This is explored in a homework exercise.

Looking Ahead

In this lecture we focused primarily on the design of the transmitter so that we can communicate reliably while still being rate efficient. We supposed the availability of the corresponding ML decoder at the receiver end. The ML decoder involves finding that information packet yielding the largest likelihood of the T received voltages. We have to calculate the likelihood for each possible information packet, 2^{RT} of them: the number of such choices grows exponentially with the number of information bits RT. Supposing a constant computation complexity to calculate the likelihood the total computation complexity of the ML receiver grows exponentially with the size of the information packet. Even with moderate sized data packets (say 1000s of bits) this is just too expensive (in fact, widely considered to be *impossible*) to implement in digital circuitry. In the language of CS (computer science) we say that the ML decoder algorithm is NP (non-polynomial time).

To complete the story of rate efficient reliable communication, we turn to simplifying the receiver design. While the ML decoder is indeed the optimal receiver, there are suboptimal decoders that perform about just as well in practice. This is the focus of the next lecture.

Lecture 7: Reliable Communication with Erasures

Introduction

So far, we have seen that arbitrarily reliable communication is possible at non-zero rates provided the receiver is well designed. In this lecture we will take a closer look at simplifying (in a computational sense) the complexity of the receiver design. We break up the receiver into two steps:

1. *Demodulation*: Map the analog received voltage to a finite number of discrete levels. To be concrete, we focus on the following situation: let the transmit voltages be binary (just as in the previous lecture). Then we map the analog voltage into one of *three* possible levels. Two of them correspond to the two levels of the binary modulation at the transmitter. The third, called an *erasure*, models the scenario when the received analog voltage is not enough to make a decision one way or the other.

2. *Decoding*: The second step involves taking the erasures into careful account and recovering the original information bits.

Receiver Design in Two Steps

For concreteness, the discussion in this lecture is limited to *binary* modulation on the AWGN channel:

$$y[m] = x[m] + w[m], \quad m = 1 \ldots T; \tag{141}$$

i.e., the transmit voltage is restricted to be $\pm\sqrt{E}$. Further, the transmitter is assumed to be broken up into the two steps described in the previous lecture (sequential modulation and linear coding). For concreteness let us suppose that $-\sqrt{E}$ is transmitted when the corresponding coded bit is 0 and $\sqrt{E}$ is transmitted when the corresponding to when the coded bit is 1.

The ML receiver (from the previous lecture) took the T received voltages and mapped them directly to the information bits. While this is the optimal design, it is also prohibitively expensive from a computational view point. Consider the following simpler two-stage design of the receiver.

1. *Demodulation*: At each time m the received voltage $y[m]$ is mapped into one of three possible choices: Let us fix $c \in (0, 1)$.

   (a) If

   $$y[m] \leq -c\sqrt{E} \tag{142}$$

   then we map into a 0.

Figure 15: Demodulation Operation.

(b) If

$$y[m] > c\sqrt{E} \tag{143}$$

then we map into a 1.

(c) In the intermediate range, i.e.,

$$-c\sqrt{E} \leq y[m] \leq c\sqrt{E} \tag{144}$$

we map into an "e" (standing for an erasure).

The process is described in Figure fig:demod and is a very easy step computationally.

2. *Decoding*: We now take the T outputs of the demodulator (one ternary symbol in $\{0, 1, e\}$, at each time instant) and map them into the information bits.

In the following, we will study each of these two stages carefully and analyze the end-to-end performance.

Demodulation

The idea of making erasures is that when the received voltage is in the intermediate range, we are less sure of whether $+\sqrt{E}$ was transmitted or $-\sqrt{E}$ was transmitted. For instance, if we receive a voltage of 0, then the corresponding transmit voltage could equally likely be $\pm\sqrt{E}$. What is the probability of the demodulation event at any time m resulting in an erasure? It is simply equal to

$$p \stackrel{\text{def}}{=} \mathbb{P}\left[-c\sqrt{E} \leq y[m] \leq c\sqrt{E}\right] = Q\left((1-c)\sqrt{\mathsf{SNR}}\right) - Q\left((1+c)\sqrt{\mathsf{SNR}}\right). \tag{145}$$

Here we have denoted SNR to be the ratio of E to the variance of the additive Gaussian noise (σ^2).

Even with the unclear intermediate range marked by erasures, it is possible that a coded bit of 0 (corresponds to a transmit voltage of $-\sqrt{E}$) could result in a demodulated symbol of 1 (corresponds to a received voltage larger than $c\sqrt{E}$). The probability of this event

$$\mathbb{P}\left[w[m] > (1+c)\sqrt{E}\right] = Q\left((1+c)\sqrt{\mathsf{SNR}}\right), \tag{146}$$

gets smaller as c is made larger. We could decide to set c large enough so that this probability is made desirably small. In the following, we will suppose this is small enough that we can ignore it (by setting it to zero).[4]

Let us summarize the demodulation output events:

1. When the coded bit is a 0, the output of the demodulated step takes on one of two possible values:
$$\begin{array}{ll} 0 & \text{with probability } 1-p \quad \text{(cf. Equation (145))} \\ e & \text{with probability } p. \end{array} \tag{147}$$

2. Analogously, when the coded bit is a 1, the output of the demodulated step takes on one of two possible values:
$$\begin{array}{ll} 1 & \text{with probability } 1-p \\ e & \text{with probability } p. \end{array} \tag{148}$$

Decoding

We now have T demodulation outputs (each one of them being either 0, 1 or e). Whenever we get a 0 (1), we are confident that the corresponding code bit could only have been a 0 (1). This is based on our assumption that the chance that a coded bit of 0 will result in a demodulated symbol of 1 is very small; so small, that we have modeled it as zero. Thus the only time instants where the decoder has to do some work is in the erased locations. For T large, what fraction of the locations do we expect to be erasures? Since we are supposing that the coded bits are equally likely to be 0 or 1 and are statistically independent from time to time, the fraction of erasures is p. Indeed, by the *law of large numbers*,

$$\mathbb{P}\left[\left|\frac{\text{Number of erasures over time } T}{T} - p\right| > \delta\right] \to 0, \quad \text{as } T \to \infty. \tag{149}$$

So, we have approximately $(1-p)T$ of the coded bits recovered correctly at the output of the demodulation step.

The job of the decoder is to use these as inputs and figure out the original information bits. At this point it is useful to take a close look at the coding operation at the transmitter that mapped the information bits into the coded bits (cf. Equation (1) from Lecture 6):

$$\left[\begin{array}{c}\text{vector of} \\ \text{coded bits}\end{array}\right] = C \left[\begin{array}{c}\text{vector of} \\ \text{information} \\ \text{bits}\end{array}\right]. \tag{150}$$

[4]It is important to keep in mind that no matter how small this probability is, the chance that at least one of such an undesirable event occurs in the transmission of a large packet grows to one as the packet size grows large. This is the same observation we have made earlier in the context of the limitation of sequential communication.

The size of the coding matrix C is $T \times RT$ where $R < 1$ is the coding rate. Now at the decoder we have available a fraction $(1-p)T$ number of the coded bits with a fair degree of certainty.[5] Thus we can rewrite Equation (150) as

$$\begin{bmatrix} \text{vector of} \\ \text{demodulated} \\ \text{bits} \end{bmatrix} = \tilde{C} \begin{bmatrix} \text{vector of} \\ \text{information} \\ \text{bits} \end{bmatrix}. \tag{151}$$

Here the matrix $\tilde{C}$, of dimension $(1-p)T \times RT$, is a *sub-matrix* of the original linear coding matrix C: it is formed by choosing $(1-p)T$ of the T rows of C (exactly which rows are chosen depend on which of the demodulated outputs were *not* erasures). Now, to recover the information bits from the linear set of equations in Equation (151), we need at least as many equations $(1-p)T$ as variables (RT). Further we need at least RT of these equations to be linearly independent. Putting these conditions into mathematical language, we need:

- $R < 1 - p$.
- The matrix $\tilde{C}$ has *rank* at least RT.

The first condition is simply a constraint on the coding rate R. This is readily satisfied by choosing the data rate appropriately at the transmitter. The second condition says something about the linear coding matrix C. We need every subset of RT rows of the matrix C to be linearly independent.

Design of Good Linear Codes

How does one construct such linear codes? This has been the central focus of research for several decades and only recently could we say with some certainty that the final word has been said. The following is a quick summary of this fascinating research story.

1. Consider the random linear code (we studied this in the previous lecture as well): each entry of C is i.i.d. 0 or 1 with probability 0.5 each. It turns out that almost surely the random matrix C has the desired rank property. Thus it is easy to construct linear codes that work for the erasure channel (almost every random linear code works). This is a classical result:

   P. Elias, "Coding for two noisy channels," *Information Theory*, 3rd London Symposium, 1955, pp. 6176.

   The problem with this approach is the decoding complexity – which involves inverting a $(1-p)n \times (1-p)n$ matrix – is $O(n^3)$.

[5] Exactly which fraction of the coded bits are available is unclear; all is known is that a total of $(1-p)T$ coded bits are available.

2. **Reed-Solomon Codes**: These are *structured* linear codes that guarantee the decodability condition. They have the additional property that their decoding complexity is smaller: $O(n^2)$. Reed-Solomon codes are used in several data storage applications: for example, hard disks and CDs. One can learn about these codes from *any* textbook on coding theory. Unfortunately one cannot get nonzero data rates from these structured codes.

3. **Digital Fountain Codes**: These are a relatively new class of random linear codes that satisfy the decodability condition with very high probability. The distinguishing feature is that the matrix C is very sparse, i.e., most of its entries are 0. The key feature is a simple decoding algorithm that has complexity $O(n \log(n/\delta))$ with probability larger than $1 - \delta$. For a wide class of channels, a sparse linear code admits a simple decoding algorithm, called the *message passing algorithm*. This is a very recent development that has revolutionized coding theory, both theoretically and from a practical point of view. Here are a few references for you to learn more about this exciting area. Anytime there is a such a significant breakthrough, you can expect some entrepreneurial activity surrounding it. The interested student might want to check out `http://www.digitalfountain.com`.

Looking Ahead

We have looked at erasures as an intermediate step to simplify the receiver design. This does not however, by itself, allow arbitrarily reliable communication since the total error probability is dominated by the chance that at least one of the coded bits is demodulated in error (i.e., not as an erasure). To get to arbitrarily reliable communication, we need to model this *cross-over* event as well: coded bits getting demodulated erroneously. A detailed study of such a model and its analysis is a bit beyond the scope of this course. So, while we skip this step, understanding the fundamental limit on the reliable rate of communication after such an analysis is still relevant. This will give us insight into the fundamental tradeoff between the resource of energy and performance (rate and reliability). This is the focus of the next lecture.

Lecture 8: Capacity of the AWGN Channel

Introduction

In the last two lectures we have seen that it is possible to communicate rate efficiently and reliably. In this lecture we will see what the fundamental limit to the largest rate of such a reliable communication strategy is. This fundamental limit is called the *capacity*.

Examples

We can see what the fundamental limits to reliable communication are in the context of the scenarios in the last two lectures:

1. *AWGN channel*: With binary modulation and random linear coding at the transmitter and ML decoding at the receiver, we have seen that the largest rate of reliable communication is
$$R^* = 1 - \log_2\left(1 + e^{-\frac{\mathsf{SNR}}{2}}\right). \tag{152}$$
This is the capacity of the AWGN channel when the transmitter is restricted to do linear coding and binary modulation.

2. *Erasure channel*: We developed this model in Lecture 7 in the context of simplifying the receiver structure. But it is a very useful abstract model on its own right and widely used to model large packet networks (such as the Internet). The basic model is the following: the transmitter transmits one bit at a time (you could replace the word "bit" by "packet"). The receiver either receives the bit correctly or it is told that the bit got *erased*. There is only a single parameter in this channel and that is the rate of erasures (the chance that any single transmit bit will get erased before reaching the receiver): p.

   What is the largest data rate at which we can hope to communicate reliably? Well, since only a single bit is sent at any time, the data rate cannot be more than 1 bit per unit time. This is rather trivial and we can tighten our argument as follows: the *receiver* receives only a fraction $1-p$ of the total bits sent (the remaining p fraction of the total bits sent got erased). So, the data rate for reliable communication could not have been any more than the fraction of bits that the receiver got without erasures. We can thus conclude that the data rate is no more than $1-p$ bits per unit time. We can say something stronger: if the data rate is more than $1-p$ bits per unit time then the reliability of communication is arbitrary poor (the chance of not getting all the bits correctly at the receiver is very close to 100%).

   How should the transmitter do to ensure that we really can communicate at rates close to this upper limit? If the transmitter *knew in advance* where the erasures are going to be, then it could simply ignore this time instants and communicate the bits only over the remaining time instants. This, of course, would let reliable communication occur at data rates very close to $1-p$. But the position of the erasures are unlikely to be known in advance. We saw linear coding strategies in Lecture 7 that can still achieve reliable communication at rates very close to $1-p$ even in the absence of the knowledge of the erasure locations at the transmitter.

So we can conclude:

- Communication at data rate larger than $1-p$ bits per unit time entails arbitrarily poor reliability.
- Reliable communication can be achieved by an appropriate transmit-receive strategy as long as the data rate is less than $1-p$ bits per unit time.

The quantity $1-p$ represents a *sharp* and fundamental threshold for the data rate of reliable communication: no strategy exists when the data rate is larger than $1-p$ and there do exist strategies when the data rate is less than $1-p$. These two different aspects are summarized by the single sentence:

The *capacity* of the erasure channel is $1-p$ bits per unit time.

In the rest of this lecture we will see what the capacity of the AWGN channel is. Before we do this, we set the stage by highlighting a subtle difference between energy and power constraints on the transmitter.

Power Constraint

In our discussion so far, we have considered the energy constraint on the transmit voltages:

$$|x[m]| \leq \sqrt{E}, \quad \forall m. \tag{153}$$

This constraint is also called a *peak power* constraint. An alternative and weaker constraint is on the average power:

$$\sum_{m=1}^{N} |x[m]|^2 \leq NE. \tag{154}$$

The peak power constraint in Equation (153) implies the average power constraint in Equation (154), but not vice versa. In this lecture we will consider the weaker average transmit power constraint. Our focus is the usual AWGN channel

$$y[m] = x[m] + w[m], \quad m = 1, \ldots \tag{155}$$

where $w[m]$ is i.i.d. (independent and identically distributed) with statistics at any time being Gaussian (zero mean and variance σ^2). In the last two lectures we had restricted the transmit voltage to be one of only two possible voltages ($\pm\sqrt{E}$), now we allow any real voltage as long as the average power constraint in Equation (154) is met. We will denote the ratio

$$\mathsf{SNR} \stackrel{\text{def}}{=} \frac{\mathsf{E}}{\sigma^2} \tag{156}$$

as the signal to noise ratio of the channel.

Capacity

It turns out that the largest rate of arbitrarily reliable communication is

$$C_{\text{awgn}} \stackrel{\text{def}}{=} \frac{1}{2} \log_2 (1 + \mathsf{SNR}) \quad \text{ bits/channel use.} \tag{157}$$

This is the most important formula in communications: a sort of the equivalent of, the more famous, formula from physics:

$$E = mc^2. \tag{158}$$

That formula was derived by, as is very well known, by Albert Einstein. The communication equivalent (cf. Equation (157)) was derived by Claude Shannon in 1948. Again, the operational meaning of the capacity C_{awgn} is as before: for every rate below C_{awgn} there exist transmitter-receiver strategies that ensure arbitrarily reliable communication. Furthermore, for any rate larger than C_{awgn} communication is hopelessly unreliable.

We won't quite go into how Equation (157) was derived, but we will work to see how it is useful to communication engineers. We do this next. As a starting point, it is instructive to see how the capacity performs at low and high SNRs.

At high SNR, we can approximate $1 + \mathsf{SNR}$ by SNR and then

$$C_{\text{awgn}} \approx \frac{1}{2} \log_2 \mathsf{SNR} \quad \text{bits/channel use.} \tag{159}$$

We see that for every quadrupling of SNR the capacity increases by one bit. This is exactly the same behavior we have seen very early in this course, indeed way back in Lecture 1.

At low SNR, we have

$$C_{\text{awgn}} \approx \frac{1}{2} (\log_2 e)\, \mathsf{SNR} \quad \text{bits/channel use.} \tag{160}$$

In this situation a quadrupling of SNR also quadruples the capacity due to the linear relation between capacity and SNR.

Transmitter and Receiver Designs

How do the transmitter and receiver strategies that hope to work close to this fundamental limit look like?

- *Transmitter*: In our attempts to understand reliable communication at non-zero rates (in the last two lectures) we divided the transmitter strategy into two parts:
  - *coding*: mapping the information bits into coded bits; this is done at the block level. We focused specifically on *linear* coding.
  - *modulation*: mapping the coded bits into transmit voltages; this is done sequentially.

  It turns out that essentially the same steps continue to work even in attaining the fundamental reliable rate of communication in Equation (157). At low SNRs, binary modulation suffices. At high SNR, the modulation involves larger alphabets and is also done in a block manner, albeit the modulation block size is usually smaller than the coding block size.

- *Receiver*: In our study of the erasure channel in the previous lecture, we saw a fairly simple receiver structure. In this general setting, the receiver is more involved: the ML receiver is hopeless (computationally) to implement. Harnessing the understanding gleamed from the erasure channel codes, a class of suboptimal (compared to ML) receiver techniques that are simple to implement have been developed in the last decade. Specifically, these receivers *iterate* by alternating between demodulation and linear decoding, eventually converging to the true information bit transmitted. This study is somewhat out of the scope of this course. We will provide some reading material for those interested in this literature at a later point.

Looking Ahead

So far we have focused on the discrete time additive noise channel (cf. Equation (155)). We arrived at this model in Lecture 1 by using the engineering blocks of DAC (digital to analog conversion) and ADC (analog to digital conversion) at the transmitter and receiver, respectively. In the next lecture, we take a closer look at the DAC and ADC blocks in terms of how their design impacts the end-to-end communication process. Of specific interest will be what constrains the rate of discretization. Clearly, the larger this rate, the larger the capacity of the end-to-end communication. We will see that the *bandwidth* of the transmit signal plays an important role in constraining the number of channel uses per second. Further we will be able to discuss the relation between the largest rate of reliable communication and the key physical resources available to the engineer: power and bandwidth.

Lecture 9: Pulse Shaping and Sampling

Introduction

Information is digital in today's world but the physical world is still analog. Digital communication entails mapping digital information into electromagnetic energy (voltage waveforms) and transmitting over an appropriate physical medium (over a wire or wireless). At the receiver, we record the electromagnetic energy (voltage waveform again) and based on this knowledge, try to recover the original information bits. In the first lecture, we pointed out that for engineering convenience, the mapping between digital information and analog voltage waveforms is divided into two separate parts. At the transmitter:

- we first map digital information into a *discrete sequence* of voltage levels; this is the modulation or coding step.
- next, we interpolate between these voltage levels to produce an analog voltage waveform that is then transmitted; this is the DAC (digital to analog conversion) step.

At the receiver:

- we *sample* the received analog voltage waveform to produce a discrete sequence of voltage levels; this is the ADC (analog to digital conversion) step.
- next, we map the discrete sequence of sampled voltage levels to the information bits; this is the demodulation or decoding step.

These operations are depicted in Figure 16, in the context of transmission over the AWGN channel.

We have seen in the previous lectures, in substantial detail, the steps of modulation (coding) and demodulation (decoding). In this lecture, we will delve deeper into the DAC and ADC steps. At the end of this lecture, we will be able to derive a relationship between the sampling and interpolation rates (of the ADC and DAC, respectively) with an important physical parameter of an analog voltage waveform: *bandwidth*.

Digital to Analog Conversion (DAC)

How do we map a sequence of voltages, $\{x[m]\}$, to waveforms, $x(t)$? This mapping is known as DAC. There are a few natural conditions we would like such a mapping to meet:

1. Since the digital information is contained in the discrete sequence of voltages $\{x[m]\}$, we would like these voltages to be readily recovered from the voltage waveform $x(t)$. One way of achieving this is to set

$$x(mT) = x[m] \tag{161}$$

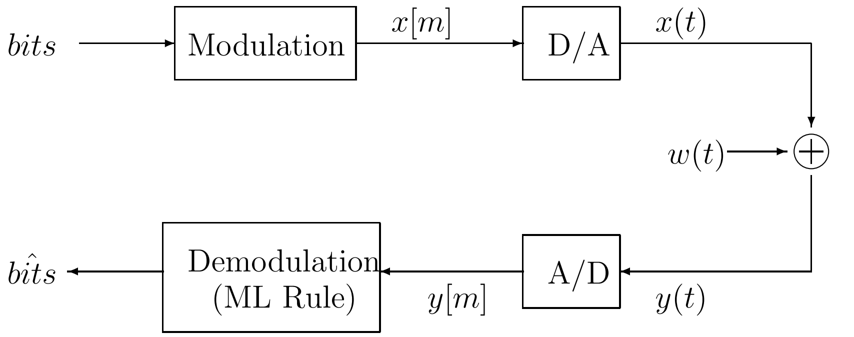

Figure 16: The basic transmit and receive operations in the context of communicating over the AWGN channel.

where T is the time period between voltage samples. This way, all the information we want to communicate is present in the transmitted waveform and readily extractable too.

2. We could potentially pick any waveform that satisfies Equation (161). In other words, we seek to *interpolate* between the uniformly spaced voltage sequence. Of course, this interpolation should be universal, i.e., it should work for *any* sequence of discrete voltage levels (this is because the voltage sequence varies based on the coding method (sequential vs block) and also the information bits themselves). While such interpolation could be done in any arbitrary fashion (as long as Equation (161) is obeyed), there are some natural considerations to keep in mind:

   (a) We would like the resulting waveform $x(t)$ to have the smallest possible bandwidth. As we will see shortly, physical media, both wireless and wireline, impose spectral restrictions on the waveform transmitted through them; the most common type of these restrictions is that the bandwidth be as small as possible.

   (b) With an eye towards ease of implementation, we would like to have a *systematic* way of interpolating the voltage sequences. It would also be useful from an implementation stand point if the waveform value at any time can be generated using the discrete voltage values in its *immediate neighborhood*.

A convenient way of taking care of these conditions while interpolating, is to use the *pulse shaping* filter:

$$x(t) = \sum_{m>0} x[m] g\left(t - mT\right) \qquad (162)$$

where T is once again the time period between samples, and $g(t)$ is our "pulse", also known as the "interpolation filter". Essentially, a DAC that uses this pulse shaping equation will

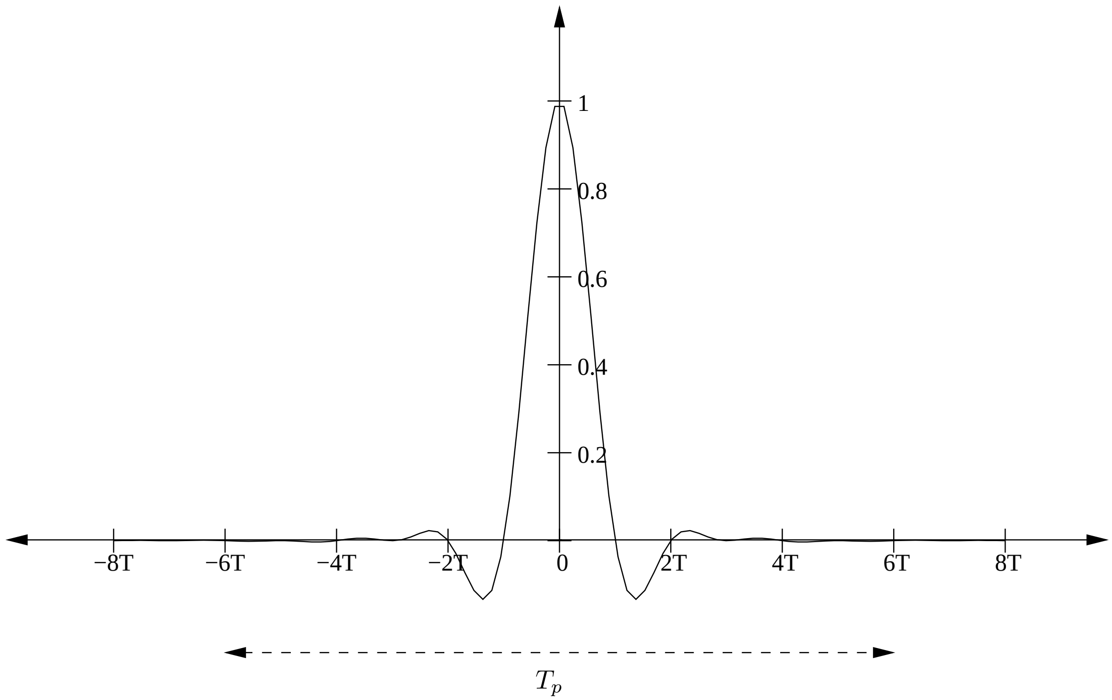

Figure 17: A pulse and its spread T_p.

overlay (or convolve) the pulse over the voltage impulse defined by $x[m]$, and add all the convolutions together. Thus, the DAC in a transmitter is also called a "pulse shaper". This ensures a systematic way of generating the voltage waveform, but how well the other two conditions enumerated above are met depends on the choice of the pulse $g(t)$:

1. From Equation (162), we see that the bandwidth of $x(t)$ is exactly the *same* as the bandwidth of $g(t)$. So, controlling the bandwidth of $x(t)$ is the same as appropriate design of the pulse $g(t)$.

2. How many neighboring discrete voltage values $x[m]$ affect the actual value of the voltage waveform $x(t)$ at any time t depends on the *spread* of the pulse: the larger the spread, the more the impact of the number of neighboring discrete voltage values in deciding the waveform voltage value. Figure 17 illustrates this point: the number of neighboring voltages that make an impact is approximately the ratio of the spread of the pulse T_p to the time period between the discrete voltage values T.

These two aspects are better appreciated in the concrete context of the three example pulses discussed below.

Exemplar Pulses

- Rectangular Pulse: The rectangular pulse (`rect` pulse for short), or Zero Order Hold (ZOH) would overlay a voltage sample with a waveform that looks like that in Figure 18. As is, the `rect` pulse is nonzero for negative values of t, making it depend on a future discrete voltage level. An alternative shifted version of the ZOH simply holds each voltage until the next time instant: thus exactly one discrete voltage level is all that is needed (the immediately previous one) to decide the voltage waveform value at any time t.

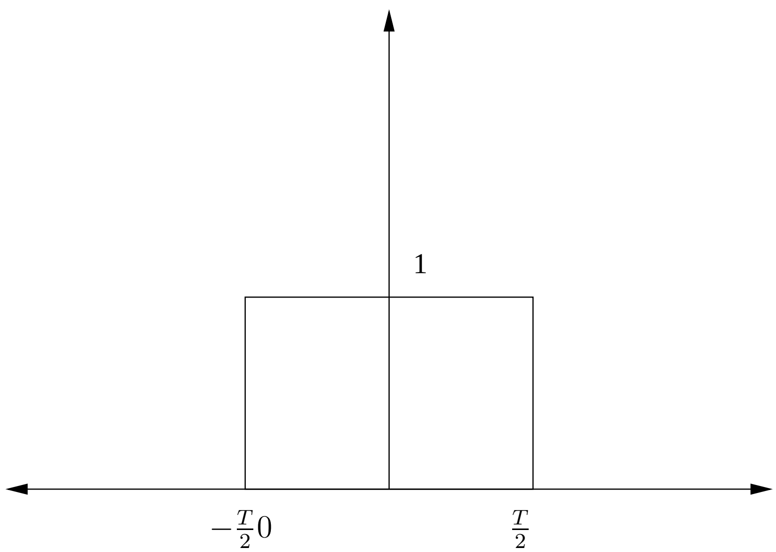

Figure 18: The rectangular pulse waveform $g(t) = \text{rect}\left(\frac{t}{T}\right)$

The greatest advantage of the ZOH is that it is very simple to implement with minimal memory requirements. On the flip side however, we know from our previous knowledge on waveforms and the Fourier transform that sharp edges in the time domain means large bandwidth in the frequency domain. Thus, this is the main disadvantage of the ZOH, since its Fourier transform is actually a `sinc` function, which has infinite bandwidth. As a result, the rectangular pulse is not a very practical interpolation filter to use, since it is not possible to keep the spectral content of the waveform within any bandwidth constraint (that a channel would impose).

- Sinc Pulse: From the rectangular pulse, we learn that we want to constrain the bandwidth of the interpolation filter that we choose. If that is the case, then a natural choice for our pulse should have a power spectrum with a rectangular shape similar to that in Figure 18. A pulse shape with this spectrum is the "ideal interpolation filter" in signal processing, and corresponds to $g(t) = \text{sinc}\left(\frac{t}{T}\right) \stackrel{\text{def}}{=} \frac{\sin\left(\frac{\pi t}{T}\right)}{\frac{\pi t}{T}}$ in the time domain (see Figure 19).

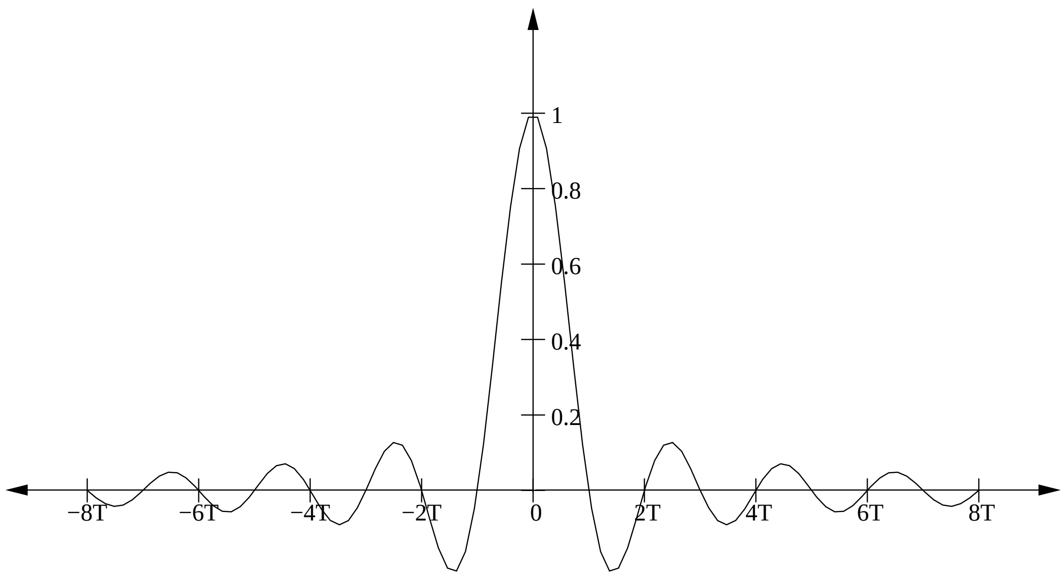

Figure 19: The sinc pulse waveform $g(t) = \text{sinc}\left(\frac{t}{T}\right)$.

The advantages of the `sinc` pulse are similar to the advantages of the rectangular pulse, except that they hold in the frequency domain rather than in the time domain. It completely restricts the frequency content within a compact box so that we can tightly meet any bandwidth constraint. But such a perfect cut-off in the frequency domain implies that the time domain sequence has to be of infinite length, which means that the pulse spread is very large.

- Raised Cosine Pulse: A solution to the problems of the rectangular and `sinc` filters is to choose a pulse that lies in between. This way, the pulse would not be infinitely long in either the time or frequency domain. An example of such a pulse is the *raised cosine* pulse, whose waveform is shown in Figure 20. Mathematically, the raised cosine pulse is given by

$$g_{rc}(t) = \text{sinc}\left(\frac{t}{T}\right) \frac{\cos\left(\frac{\pi\beta t}{T}\right)}{1 - \frac{4\beta^2 t^2}{T^2}}. \tag{163}$$

By varying the parameter β between 0 and 1, we can get from a `sinc` pulse ($\beta = 0$) to a much dampened version ($\beta = 1$).

As the illustration in Figure 20 demonstrates, the raised cosine has a smooth roll-off

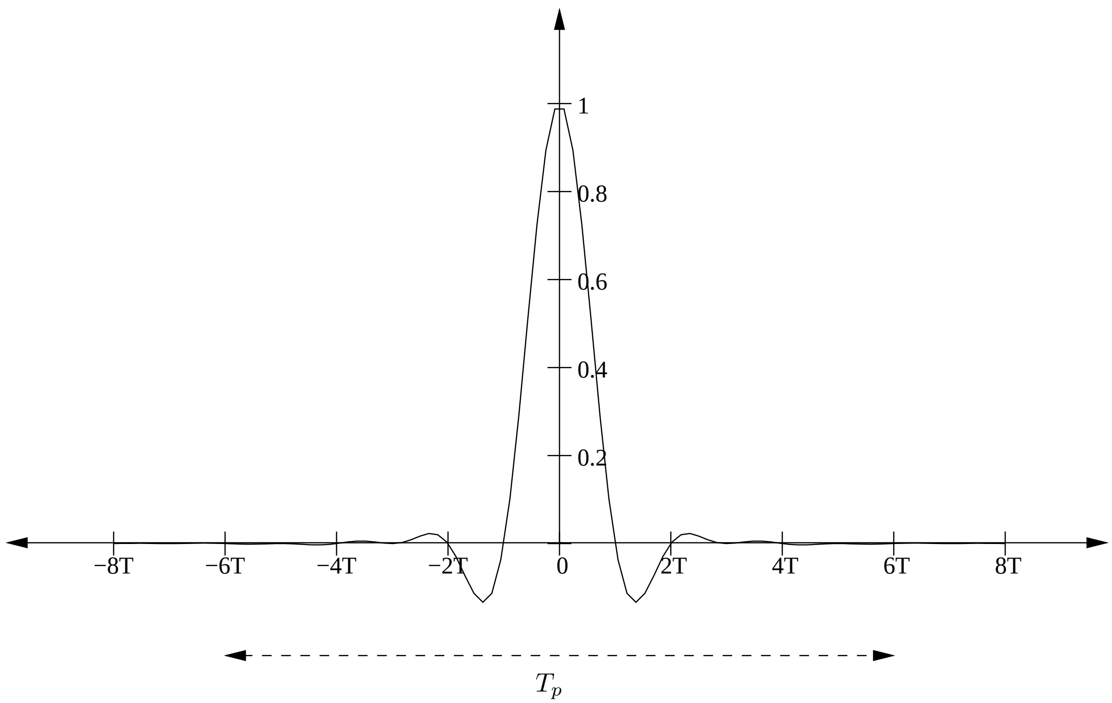

Figure 20: The raised cosine pulse waveform.

in the time domain, restricting the amount of bandwidth it uses. It also dampens to zero more quickly than the `sinc`, meaning that it is also more practical to use.

Bandwidth and Narrowest Spacing

What is the bandwidth of the transmit signal with the exemplar pulses studied above? There are several definitions of bandwidth, but let us agree on an approximate definition that is good for our purposes: bandwidth of a voltage waveform $x(t)$ is the measure of the smallest set of frequencies where most of the energy of the *Fourier* transform $X(f)$ is contained in. For instance: the Fourier transform of the pulse $\text{sinc}\left(\frac{t}{T}\right)$ is $T\text{rect}\,(fT)$ (cf. Homework exercise). We see that all the energy of the Fourier transform is contained entirely within the finite spread of frequencies $\left[-\frac{1}{2T}, \frac{1}{2T}\right]$. So, we say that the bandwidth is $1/T$. More generally, the bandwidth of the raised cosine pulse depends on the parameter β: it increases as β increases. Specifically, the bandwidth is equal to $1/T$ when $\beta = 0$, about $\frac{1.5}{T}$ when $\beta = 0.5$ and about $\frac{2}{T}$ when $\beta = 1$ (cf. Homework exercise).

In each of the cases studied above the bandwidth of the transmit signal is directly proportional to $1/T$, exactly the rate of discrete transmit voltage sequences. But our study of the special cases above was motivated by engineering purposes (simple methods of interpolation, etc). It is useful to know how much we are losing by taking the engineering considerations very seriously: this entails asking the following fundamental question:

> Supposing any interpolation strategy, what is the smallest bandwidth one can get as a function of the discrete transmit voltage rate $\frac{1}{T}$?

A fundamental result in communication theory says that the answer is approximately equal to $\frac{1}{T}$.[6] Concretely, this is exactly achieved by using the *sinc* pulse to shape the transmit waveform. In engineering practice, a raised cosine pulse (with an appropriate choice of β) is used. While that will entail a "loss" in bandwidth usage, we will simply use the term $1/T$ for the bandwidth of the transmit signal. This will not seriously affect any of the conceptual developments to follow (the numerical values of data rate might change).

Analog to Digital Conversion (ADC)

The received voltage waveform $y(t)$ has potentially larger bandwidth than $x(t)$ due to the addition of the noise waveform $w(t)$. However, the information content is available only within the bandwidth of the transmit waveform $x(t)$. So we can, without loss of any information, filter the received waveform and restrict its bandwidth to be the same as that of the transmit waveform $x(t)$. We would like to convert this waveform into a discrete sequence of voltage levels that can be further processed to decode the information bits. The natural thing to do is to *sample* the received waveform at the same rate $1/T$: this creates a discrete sequence of voltage levels

$$\begin{aligned} y[m] &\stackrel{\text{def}}{=} y(mT) & (164) \\ &= x(mT) + w(mT) & (165) \\ &= x[m] + w[m], & (166) \end{aligned}$$

that are the transmit voltage sequence corrupted by additive noise. This is the basic AWGN channel we have studied reliable communication over. In engineering practice, it is quite common to over sample at the receiver: for instance at the rate of $2/T$ rather than $1/T$ samples per second. This leads to a somewhat more involved model where the intermediate received samples are noisy versions of a weighted sum of transmit voltages. We will take a careful look at the analytical forms of this channel model in the next lecture when we begin studying wireline communication. Further, we will see the engineering rationale for over sampling at the receiver during our upcoming study of reliable communication over wireline channels.

[6] We say approximately because our definition of bandwidth involved the mathematically ambiguous word "most".

Looking Ahead

In the next lecture we will talk about the capacity of the *analog* (continuous time) AWGN channel. We do this by coupling the observations of this lecture with the formula for the capacity of the discrete-time AWGN channel derived earlier. The implications on system resources (power and bandwidth) are also explored next.

Lecture 10: Capacity of the Continuous time AWGN Channel

Introduction

In the penultimate lecture we saw the culmination of our study of reliable communication on the discrete time AWGN channel. We concluded that there is a threshold called capacity below which we are guaranteed arbitrarily reliable communication and above which all communication is hopelessly unreliable. But the real world is *analog* and in the last lecture we saw in detail the engineering way to connect the *continuous time* AWGN channel to the discrete time one. In this lecture we will connect these two story lines into a final statement: we will derive a formula for the capacity of the continuous time AWGN channel. This is the largest rate of reliable communication (as measured in bits/second) and depends only on the two key physical resources: bandwidth and power. We will see the utility of this formula by getting a feel for how the capacity changes as a function of the two physical resources.

The Continuous Time AWGN Channel

The channel is, naturally enough,

$$y(t) = x(t) + w(t), \quad t > 0. \tag{167}$$

The power constraint of $\bar{P}$ Watts on the transmit signal says that

$$\lim_{N\to\infty} \frac{1}{NT} \int_0^{NT} \left(x(t)\right)^2 \, dt \quad \le \quad \bar{P}. \tag{168}$$

The (two-sided) bandwidth constraint of W says that much of the energy in the transmit signal is contained within the spectral band $\left[-\frac{W}{2}, \frac{W}{2}\right]$.

We would like to connect this to the discrete time AWGN channel:

$$y[m] = x[m] + w[m], \quad m \ge 1. \tag{169}$$

This channel came with the discrete-time power constraint:

$$\lim_{N\to\infty} \frac{1}{N} \sum_{m=1}^{N} \left(x(m]\right)^2 \le P \quad \forall N. \tag{170}$$

We have already seen that there are W channel uses per second in the continuous time channel if we constrain the bandwidth of the analog transmit voltage waveform to W Hz. So, this fixes the sampling rate to be W and thus unit time in the discrete time channel corresponds to $\frac{1}{W}$ seconds.

To complete the connection we need to:

1. connect the two power constraints $\bar{P}$ and P;

2. find an appropriate model for the continuous time noise $w(t)$ and connect it to the variance of the additive noise $w[m]$.

We do these two steps next.

Connecting the Power Constraints

The continuous time transmit waveform is the related to the discrete sequence of transmit voltages through the DAC operation (cf. Lecture 9):

$$x(t) = \sum_{m>0} x[m]g(t - mT). \tag{171}$$

Now we have

$$\frac{1}{NT}\int_0^{NT} (x(t))^2 \; dt = \frac{1}{NT}\int_0^{NT} \left(\sum_{m>0} x[m]g(t-mT)\right)^2 \; dt \tag{172}$$

$$= \frac{1}{NT}\sum_{m_1,m_2>0} x[m_1]x[m_2]\left(\int_0^{NT} g(t-m_1T)g(t-m_2T)\right) \; dt \tag{173}$$

$$= \frac{1}{NT}\sum_{m_1=m_2=m>0} (x[m])^2 \int_0^{NT} (g(t-mT))^2 \; dt \tag{174}$$

$$+\frac{1}{NT}\sum_{m_1\neq m_2>0} x[m_1]x[m_2]\left(\int_0^{NT} g(t-m_1T)g(t-m_2T)\right) dt \tag{175}$$

Consider the first term of the RHS above:

$$\frac{1}{NT}\sum_{m_1=m_2=m>0} (x[m])^2 \int_0^{NT} (g(t-mT))^2 \; dt. \tag{176}$$

From Lecture 9 we know that the pulse $g(\cdot)$ has a finite spread of T_p; in our notation, the pulse is nonzero mostly over the range $[-t_0, T_p + t_0]$. Further, we have seen that typically T_p a few multiples of T. From this, we can make the following two observations:

1. the summation index m in Equation (176) spans from 1 to

$$N + \frac{T_p - t_0}{T} \approx N, \tag{177}$$

when N is large enough.

2. Next, the integral
$$\frac{1}{T}\int_{0}^{NT} (g(t-mT))^2 \; dt \tag{178}$$
is more or less constant for each m in the range from 1 to N (except perhaps for a few values at the boundary.

We can now combine these two observations to conclude that the term in Equation (176) is approximately the same as
$$\frac{c_p}{N}\sum_{m=1}^{N} (x(m])^2 , \tag{179}$$
which we see, by comparing with Equation (170), is directly proportional to the discrete time power consumed.

To complete the connection, we still need to account for the second term in the RHS of Equation (175):
$$\frac{1}{NT}\sum_{m_1 \neq m_2 > 0} x[m_1]x[m_2]\left(\int_{0}^{NT} g(t-m_1T)g(t-m_2T)\; dt\right). \tag{180}$$
Fortunately, for most pulses of interest this quantity is zero. Specifically, this statement is true for the three exemplar pulses of Lecture 9: the `sinc`, `rect` and `raised cosine` pulses. This is verified in a homework exercise. In practice, the term in Equation (180) is kept reasonably small and we can ignore its effect on the summation in Equation (175). Thus,
$$\frac{1}{NT}\int_{0}^{NT} (x(t))^2 \; dt \quad \approx \quad \frac{c_p}{N}\sum_{m=1}^{N} (x(m])^2 . \tag{181}$$
Combining this with Equations (168) and (170) we arrive at
$$\bar{P} = c_p P. \tag{182}$$
The constant c_p is a design choice that depends on the energy in the DAC pulse. For notational simplicity we will simply consider it to be unity. This allows us to map the continuous time power constant of $\bar{P}$ directly into the P, the discrete time power constraint.

Analog Noise Models and Bandwidth

Consider the continuous time noise $w(t)$ it's (random) Fourier transform. Since noise waveforms are *power* signals, i.e.,
$$\lim_{T\to\infty}\frac{1}{2T}\int_{-T}^{T} (w(t))^2 \; dt \quad = \quad \sigma^2 > 0 \tag{183}$$

the Fourier transform is not well defined. To avoid this problem we could consider the *time-restricted* noise waveform

$$w_T(t) \stackrel{\text{def}}{=} \begin{cases} w(t) & -T \le t \le T \\ 0 & \text{else} \end{cases} \tag{184}$$

which is an *energy* signal. We denote the Fourier transform of $w_T(t)$ by $W_T(f)$. The *average* variance of the Fourier transform of the time-restricted noise in the limit of no restriction is called the *power spectral density*:

$$\text{PSD}_w(f) \stackrel{\text{def}}{=} \lim_{T\to\infty} \frac{1}{2T} \text{Var}\left(W_T(f)\right). \tag{185}$$

Based on measurements of additive noise, a common model for the power spectral density is that it is constant, denoted by $\frac{N_0}{2}$, measured in Watts/Hz. Furthermore this model holds over a very wide range of frequencies of interest to communication engineers: practical measurement data suggests a value of about 10^{-14} Watts/Hz for N_0.

The corresponding statistical model of $w(t)$ is called *white Gaussian noise*. In modeling the discrete time noise (cf. Lecture 2) we used the term "white" to denote statistical independence of noise over different time samples. Here the term "white" is being used to denote statistical independence of the continuous time noise $w(t)$ over different frequencies.

This model immediately implies the following strategy: consider the received signal $y(t)$. The transmit signal is known to be bandlimited to $x(t)$ and the additive noise $w(t)$ is independent over different frequencies. So, without loss of generality:

we can *filter* the received waveform $y(t)$ so that it is bandlimited to W as well.

In practice, the received waveform is always filtered to contain it to within the same spectral band as that of the transmit waveform. Filtering the received waveform is tantamount to filtering the noise waveform alone (since the transmit waveform is anyway in the same band as that allowed by the filter). With a (double sided) bandwidth of W, the total area under the power spectral density of this filtered noise is

$$\int_{-\frac{W}{2}}^{\frac{W}{2}} \text{PSD}(f)\, df = \frac{N_0 W}{2}. \tag{186}$$

It turns out that the variance of the noise sample $w[m]$, at time sample m, is *exactly equal* to the expression in Equation (186)! This calculation is explored in a bonus homework exercise. So, we conclude that the variance of the noise sample increases in direct proportion to the bandwidth W.

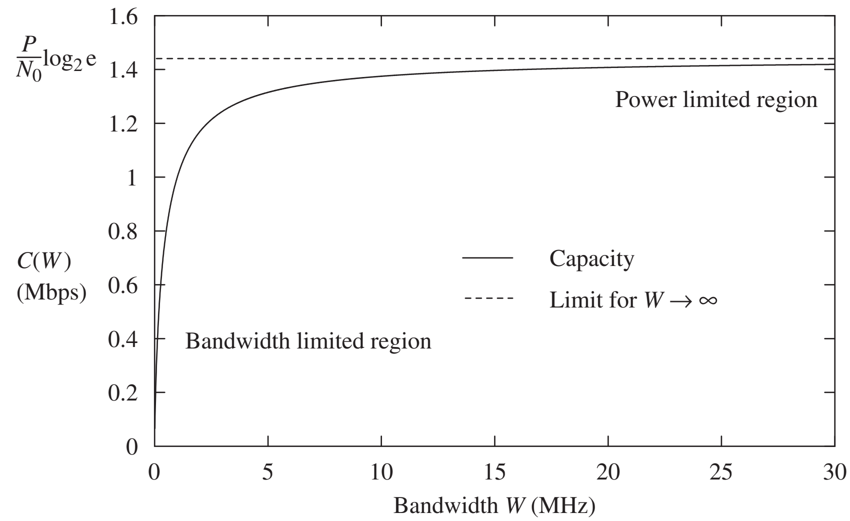

Figure 21: Capacity as a function of bandwidth for SNR per Hz $2P/N_0 = 10^6$.

Capacity

We can now put together these observations into our earlier discussion of the capacity of the discrete time AWGN channel,

$$\frac{1}{2}\log_2\left(1+\frac{P}{\sigma^2}\right) \quad \text{bits/unit time,} \tag{187}$$

to arrive at the capacity of the continuous time AWGN channel:

$$C = \frac{W}{2}\log_2\left(1+\frac{2\bar{P}}{N_0 W}\right) \quad \text{bits/s.} \tag{188}$$

Now we can see how the capacity depends on the bandwidth W. Surely the capacity can only increase as W increases (one can always ignore the extra bandwidth). One can directly show that the capacity is a *concave* function of the bandwidth W (this is explored in an exercise). Figure 21 plots the variation of the capacity as a function of bandwidth for an exemplar value of SNR per Hz.

Two important implications follow:

- When the bandwidth is small, the capacity is very sensitive to changes in bandwidth: this is because the SNR per Hz is quite large and then the capacity is pretty much linearly related to the bandwidth. This is called the *bandwidth limited* regime.

- When the bandwidth is large, the SNR per Hz is small and

$$\frac{W}{2}\log_2\left(1+\frac{2\bar{P}}{N_0W}\right) \approx \frac{W}{2}\left(\frac{2\bar{P}}{N_0W}\right)\log_2 e \tag{189}$$

$$= \frac{\bar{P}}{N_0}\log_2 e. \tag{190}$$

  In this regime, the capacity is proportional to the total power $\bar{P}$ received over the whole band. It is insensitive to the bandwidth and increasing the bandwidth has only a small impact on capacity. On the other hand, the capacity is now linear in the received power and increasing power does have a significant effect. This is called the *power limited* regime.

As W increases, the capacity increases monotonically and reaches the asymptotic limit

$$C_\infty = \frac{\bar{P}}{N_0}\log_2 e \quad \text{bits/s}. \tag{191}$$

This is the capacity of the AWGN channel with only a power constraint and no bandwidth constraint. It is important to see that the capacity is finite even though the bandwidth is not. The connection of this expression to energy efficient communication is explored in a homework exercise.

Looking Ahead

Starting next lecture, we shift gears and start looking at communication over wires. We start by looking at how a wireline channel affects voltage waveforms passing through it. We will be able to arrive at a discrete time wireline channel model by combining the effect of the wire on the voltage waveforms passing through it along with the DAC and ADC operations.

Lecture 11: Modeling the Wireline Channel: Intersymbol Interference

Introduction

We are now ready to begin communicating reliably over our first physical medium: the wireline channel. Wireline channels (telephone and cable lines) are readily modeled as linear time invariant (LTI) systems (their impulse response changes very slowly – usually across different seasons of the year). In this lecture, we will arrive at a simple discrete-time model of the wireline channel, taking into account both the sampling operation and the LTI waveform of the channel itself. The main feature of this model is that the previously transmitted symbols affect the current received symbol. This feature is called *inter-symbol interference* (ISI) and is the main new challenge that has to be dealt with in wireline channels, apart from the (by now familiar) additive Gaussian noise.

Wireline Media

A wire is a single, usually cylindrical, elongated strand of drawn metal. The primary metals used in the conducting wire are aluminium, copper, nickel and steel and various alloys therein. We are used to several wireline media in our day-to-day life. We enumerate below a few of the common ones, along with a brief description of their physical composition and, more importantly, their impulse response characteristics.

1. *Telephone wire*: This connects houses and local telephone exchange, typically using a pair of copper conducting wires. They were designed to carry human voice which are all well contained in under 10 kHz. Depending on the condition of the wire and the length (distance between the house and the local telephone exchange) the telephone wire can be thought of as a low pass filter with bandwidth of about 1 or 2 MHz.

2. *Ethernet wire*: This is typically a collection of *twisted pairs* of wires: a form of wiring in which two conductors are wound together for the purposes of canceling out electromagnetic interference from external sources and crosstalk from neighboring wires. Twisting wires decreases interference because the loop area between the wires (which determines the magnetic coupling into the signal) is reduced.

   The twist rate (usually defined in twists per meter) makes up part of the specification for a given type of cable. The greater the number of twists, the greater the attenuation of crosstalk. Further, the length of the wire decides how much the transmitted signal is attenuated; the longer the distance, the more the attenuation. There has been a standardization of these cable qualities and are typically denoted as "Cat`x`" cables: where `x` stands for the different versions. For instance, the state-of-the-art technology is the Cat6 cable (x $= 6$) with four pairs of copper wires twisted together. The maximum allowed length is about 90 meters and the impulse response can be thought of as a low pass filter with the bandwidth of about 250 MHz.

3. *Cable TV wire*: This is typically a *coaxial* cable, consisting of a round conducting wire, surrounded by an insulating spacer, surrounded by a cylindrical conducting sheath, usually surrounded by a final insulating layer. Very high bandwidths are possible: for instance, RG-6, used for cable TV, has a bandwidth of about 500 MHz.

A Discrete Time Channel Model

A wire, as we have seen is simply a low pass filter; so, its effect can be completely captured in terms of its *impulse response*: $h(\cdot)$. Two important features of this impulse response are:

1. *Causality*: physical systems are causal and so

$$h(\tau) = 0, \quad \tau < 0. \tag{192}$$

2. *Dispersion*: This is the length time (denoted by T_d) over which a good large fraction of the total energy of the impulse response is contained within:

$$\int_0^{T_d} |h(\tau)|^2 \, d\tau \quad \approx \quad \int_0^{\infty} |h(\tau)|^2 \, d\tau. \tag{193}$$

If we transmit a voltage waveform $x(t)$ over a wire with impulse response $h(\cdot)$, then the received waveform is the *convolution* between $x(\cdot)$ and $h(\cdot)$ plus an additive noise waveform:

$$\int_0^{T_d} h(\tau)x(t-\tau) \quad d\tau \quad + \quad w(t). \tag{194}$$

In writing this equation, we have used the two properties of the wire enumerated above. We also denoted the additive noise waveform by $w(t)$.

From Lecture 8, we know that the transmit waveform $x(t)$ is formed by *pulse shaping* a discrete voltage sequence. Further, the received voltage waveform is sampled to generate a discrete sequence of (received) voltages (cf. Figure 22). In other words, the transmit waveform

$$x(t) = \sum_{m>0} x[m] g(t - mT) \tag{195}$$

where $g(\cdot)$ is the pulse shaping waveform (such as the raised cosine pulse). The received waveform is sampled at regular intervals (also spaced T apart, the same spacing as in the

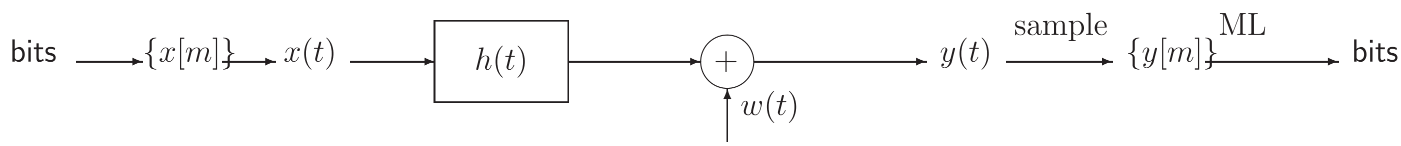

Figure 22: Illustration of the end-to-end digital communication process over a wireline channel.

transmit waveform generation):

$$y[k] \stackrel{\text{def}}{=} y(kT) \tag{196}$$

$$= \int_0^{T_d} h(\tau) \left(\sum_{m>0} x[m] g(kT - mT - \tau) \right) \quad d\tau \quad + \quad w(kT) \tag{197}$$

$$= \sum_{m>0} x[m] \left(\int_0^{T_d} h(\tau) g\left((k-m)\,T - \tau\right) \quad d\tau \right) + w[k]. \tag{198}$$

We used Equations (194) and (195) in arriving at Equation (197). Further, we have denoted the noise voltage $w(kT)$ by $w[k]$ in Equation (198).

Denoting

$$h_{k-m} \stackrel{\text{def}}{=} \int_0^{T_d} h(\tau) g\left((k-m)\,T - \tau\right) \quad d\tau, \tag{199}$$

we can rewrite Equation (198) as

$$y[k] = \sum_{m>0} x[m] h_{k-m} + w[k], \quad k \geq 1. \tag{200}$$

Observing Equation (199) more closely, we see the following:

- Since the pulse $g(\cdot)$ has a finite *spread* of T_p, i.e. it is almost zero except for a period of time T_p (cf. the discussion in Lecture 8):

$$g(t) \approx 0, \quad \forall t \notin (-t_0, T_p - t_0) \tag{201}$$

  for some t_0, we see that h_{k-m} is significantly non-zero for only a *finite* number of index values $(k-m)$.

- Specifically,

$$h_{k-m} \approx 0, \quad (k-m)T \notin [-t_0, T_p + T_d - t_0]. \tag{202}$$

Substituting $\ell = k - m$, we can say,

$$h_\ell \approx 0, \quad \forall \ell \notin \left\{ -\left\lceil \frac{t_0}{T} \right\rceil, \ldots, \left\lfloor \frac{T_p + T_d - t_0}{T} \right\rfloor \right\}. \tag{203}$$

Substituting this observation in Equation (200), we can write the discrete received voltage *sequence* as

$$y[k] = \sum_{\ell = -\lceil \frac{t_0}{T} \rceil}^{\lfloor \frac{T_p + T_d - t_0}{T} \rfloor} h_\ell x[k-\ell] + w[k]. \tag{204}$$

So, the combination of transmit and receive processing (DAC and ADC) and the wireline channel effects the input-output relationship between the discrete voltage sequence in a very simple manner:

> the received voltage sequence is a *discrete convolution* of the transmit voltage sequence with an impulse response that has a *finite* number of nonzero coefficients along with the ubiquitous additive noise voltage sequence.

Inter-Symbol Interference

Viewed in the language of discrete time signal processing, the transmit voltage sequence has been passed through an FIR (finite impulse response) filter plus a noise voltage sequence. The number of non-zero coefficients of the channel (the number of *taps* in the FIR channel) is

$$L \stackrel{\text{def}}{=} \left\lceil \frac{T_p + T_d}{T} \right\rceil. \tag{205}$$

It is convenient to view *shifted* version of the received signal, shifted to the left by $-\left\lceil \frac{t_0}{T} \right\rceil$,

$$\tilde{y}[k] \stackrel{\text{def}}{=} y\left[k + \left\lfloor \frac{t_0}{T} \right\rfloor\right] \tag{206}$$

$$= \sum_{\ell=0}^{L-1} h_\ell x[k-\ell] + w\left[k - \left\lceil \frac{t_0}{T} \right\rceil\right]. \tag{207}$$

This way, the FIR filter involved is *causal* and makes for easier notation when we begin studying how to communicate reliably over this channel. As usual, we model the additive noise voltage sequence as Gaussian distributed (zero mean and variance σ^2) and all statistically independent from each other: this is the familiar *additive white Gaussian noise* model from the earlier lectures.

Henceforth, we will not bother to keep track of the shifted index in the received voltage sequence so explicitly in the main text. We will just write the received voltage sequence as

$$y[m] = \sum_{\ell=0}^{L-1} h_\ell x[m-\ell] + w[m], \quad m \geq 1. \tag{208}$$

We can now see the main difference that the wireline channel makes as compared to the AWGN channel from before: the transmit voltages from a previous time sample also play a role in determining the received voltage at the present time sample. This phenomenon is known as *inter-symbol interference* or ISI for short: transmit voltages at different time samples (symbols) mix, or interfere, with each other to form the received voltages. The challenge of reliable wireline communication is to deal with the additive Gaussian noise in the presence of ISI.

Channel Coefficient Measurement

The channel coefficients $h_0, \ldots, h_{L-1}$ that make up the FIR filter depend on:

- sampling rate T;
- pulse shaping filter $g(\cdot)$;
- wireline channel impulse response $h(\cdot)$.

The first two quantities are chosen by the communication engineer (and hence known to both the transmitter and receiver). The final quantity depends on the physical characteristics of the wire involved. While the impulse response varies significantly from wire to wire, it is usually quite stable for a given wire (typically changing only over months along with the seasonal temperature changes). Of course, the time scale of practical reliable communication over the wire is far shorter. This means that the impulse response of the wire involved can be *learnt* at the very beginning part of the communication process and then used for reliable information transmission the rest of the time. In this section, we briefly describe this process of learning the channel coefficients.

Consider transmitting a voltage $+\sqrt{E}$ at the first time sample and nothing else after that. The received voltages of significance are:

$$y[\ell+1] = \sqrt{E}h_\ell + w[\ell+1], \quad \ell = 0, \ldots, L-1. \tag{209}$$

After these first L voltages, we only receive noise. Typical values of L are known ahead of time based on the wire involved (and the nature of the pulse shaping filter and sampling time). The precise number of taps L can also be learnt in this measurement phase; this is explored more in a homework exercise.

If $\sqrt{E} \gg \sigma$, then the received voltage $y[\ell+1]$ should provide a reasonably good estimation of h_ℓ. What is the appropriate scaling of $y[\ell+1]$ that yields a good estimate of h_ℓ? Suppose that we guess

$$\hat{h}_\ell \stackrel{\text{def}}{=} cy[\ell+1]. \tag{210}$$

Then the error in the estimate of h_ℓ is $(h_\ell - \hat{h}_\ell)$. A natural choice of the scaling constant c is to minimize the mean of the squared error:

$$\begin{aligned} \mathbb{E}\left[\left(h_\ell - \hat{h_\ell}\right)^2\right] &= \mathbb{E}\left[(cy[\ell+1] - h_\ell)^2\right] &(211)\\ &= \mathbb{E}\left[h_\ell^2\right]\left(1 - c\sqrt{E}\right)^2 + c^2\sigma^2, &(212)\end{aligned}$$

since the channel coefficient h_ℓ and the noise voltage $w[\ell+1]$ are reasonably modeled as statistically independent of each other. Continuing from Equation (212), the variance of the error in the channel coefficient estimate is

$$\mathbb{E}\left[h_\ell^2\right]\left(1 - c\sqrt{E}\right)^2 + c^2\sigma^2 = c^2\left(\mathbb{E}\left[h_\ell^2\right]E + \sigma^2\right) - 2c\mathbb{E}\left[h_\ell^2\right]\sqrt{E} + \mathbb{E}\left[h_\ell^2\right]. \tag{213}$$

This quadratic function in c is readily minimized at

$$c^* \stackrel{\text{def}}{=} \frac{\mathbb{E}\left[h_\ell^2\right]\sqrt{E}}{\mathbb{E}\left[h_\ell^2\right]E + \sigma^2}. \tag{214}$$

To actually implement this scaling, we need to know the average attenuation of the ℓ^{th} channel coefficient. Typical values of such average attenuation are usually known based on the wire involved (that the telephone, ethernet and cable wires are *standardized*, allows for such data to be universally available). The corresponding error in the channel estimate is now

$$\begin{aligned} \mathbb{E}\left[h_\ell^2\right]\left(1 - c^*\sqrt{E}\right)^2 + (c^*)^2\sigma^2 &= \mathbb{E}\left[h_\ell^2\right] - \frac{\left(\mathbb{E}\left[h_\ell^2\right]\right)^2 E}{\mathbb{E}\left[h_\ell^2\right]E + \sigma^2} &(215)\\ &= \frac{\mathbb{E}\left[h_\ell^2\right]}{1 + \mathbb{E}\left[h_\ell^2\right]\mathsf{SNR}}. &(216)\end{aligned}$$

Here we have denoted SNR as the ratio of E to σ^2. We can choose SNR to be large enough so that the variance of the error in the channel estimate is deemed small enough.

Looking Ahead

The wireline channel has now been simply modeled as an FIR filter acting on the transmit voltage sequence. Further, the coefficients of this filter are known before reliable data transmission is initiated (via the estimation process). We will see how to deal with the additive Gaussian noise in the presence of the ISI in the next several lectures.

Lecture 12: Intersymbol Interference Management: Low SNR Regime

Introduction

So far, we have seen that the wireline channel is modeled as an FIR (finite impulse response) filter acting on the transmit voltage sequence. The main difference between the AWGN channel and wireline channel is the presence of *inter-symbol interference* (ISI), i.e., transmit voltages of the previous symbols also mix along with the additive Gaussian noise with the voltage of the current symbol. The main challenge in wireline channel is to handle noise and ISI simultaneously in the quest to achieve rate efficient reliable communication.

Our approach to wireline channel will be to "simply" process the transmit and receive voltage sequence (at the transmitter and receiver, respectively) to mitigate ISI and harness the block codes developed for rate efficient reliable communication on the AWGN channel. In the next few lectures we study the *receiver centric* methods to deal with ISI. The transmitter will be more or less the same as that in the case of AWGN channel. In this lecture, we focus on the low SNR regime: in this scenario, the noise power dominates the total signal power. Thus noise dominates the ISI.

A First Approach

The received voltage is

$$y[m] = \sum_{l=0}^{L-1} h_l x[m-l] + w[m] \tag{217}$$

$$= h_0 x[m] + \sum_{l=1}^{L-1} h_l x[m-l] + w[m] \tag{218}$$

$$= S[m] + I[m] + N[m] \tag{219}$$

where

$$S[m] \stackrel{\text{def}}{=} h_0 x[m] \tag{220}$$

is the "signal";

$$I[m] \stackrel{\text{def}}{=} \sum_{l=1}^{L-1} h_l x[m-l] \tag{221}$$

is interference and

$$N[m] \stackrel{\text{def}}{=} w[m] \tag{222}$$

is noise. The implication of the low SNR regime is that

$$\mathbb{E}\left[(I[m])^2\right] \ll \mathbb{E}\left[(N[m])^2\right]. \tag{223}$$

In this regime, the natural approach is to ignore the interference completely and treat it as noise. In other words, the receiver could just do the nearest neighbor detection (even though it may not be optimal since the discrete interference definitely does not have Gaussian statistics). A natural question is:

What reliable rate of communication is feasible with this approach?

In an AWGN channel, SNR is the only parameter of interest: the capacity is a direct function of the SNR. If we are to use that intuition here, we could look at the SINR (signal to interference plus noise ratio) defined as:

$$\mathsf{SINR} \stackrel{\text{def}}{=} \frac{\mathbb{E}\left[(S[m])^2\right]}{\mathbb{E}\left[(I[m])^2\right] + \mathbb{E}\left[(N[m])^2\right]} \tag{224}$$

$$= \frac{h_0^2 \mathbb{E}\left[x[m]^2\right]}{\mathbb{E}\left[\left(\sum_{l=1}^{L-1} h_l x[m-l]\right)^2\right] + \sigma^2}. \tag{225}$$

Here σ^2 is equal to $\mathbb{E}(w[m]^2)$, the variance of the discrete-time noise. In deriving this expression, we implicitly used the statistical independence between transmitted voltages and the noise. Denoting

$$\mathbb{E}(x[m]^2) = E \tag{226}$$

to be the signal power (the student is encouraged to think of binary modulation for concreteness), the SINR can be written as

$$\mathsf{SINR} = \frac{h_0^2 E}{\left(\sum_{l=1}^{L-1} h_l^2\right) E + \sigma^2} \tag{227}$$

$$= \frac{h_0^2 \mathsf{SNR}}{1 + \left(\sum_{l=1}^{L-1} h_l^2\right) \mathsf{SNR}}. \tag{228}$$

If we continue the analogy with the AWGN channel, we expect the largest rate of communication when we treat the interference as noise to be

$$\frac{1}{2}\log_2(1 + \mathsf{SINR}) \quad \text{bits/channel use.} \tag{229}$$

How do we use our previous knowledge to conclude this result? Our previous result on the capacity (cf. Lecture 8a) depended on two basic assumptions:

- noise is independent of the signal.
- statistics of the noise is Gaussian.

Both these issues are in contention in this scenario when we have considered interference as part of the overall noise:

1. the overall noise (interference plus noise) is *dependent* on the signal; indeed, the interference is just previously transmitted signals!
2. the statistics of the overall noise is not Gaussian.

We can overcome the first issue by the following simple but *important* strategy. The overall information packet is broken up into sub-packets (that are all statistically independent of each other). The sub-packets are coded and modulated. The transmit voltages corresponding to each sub-packet are then *interleaved* at the transmitter. The receiver *deinterleaves* the received voltages. This operation is illustrated in Figure 23 where the number of sub-packets is exactly chosen to be the length of the ISI.

Now the consecutive transmitted symbols over a block of length L are statistically independent since they are generated based on statistically independent sub-packets. So, from the perspective of each of the transmit symbols, the interference is statistically independent of it.

However, interference is still *not* Gaussian. But it turns out that the Gaussian statistics is the "worst case" scenario for noise, i.e., by treating interference as noise with Gaussian statistics, the error probability of the nearest neighbor receiver (though not the optimal ML receiver anymore) for good AWGN channel codes remains the same. Look at it another way, and we can conclude that the nearest neighbour detector is quite robust. As a rule, we derive every statement we make from first principles in these lectures. However, this fact is a bit too deep for us to delve into right now (and will lead us astray from our main purpose: reliable communication over wireline channels). So, we provide the appropriate reference to this result which the interested student can read and mull over.

> A. Lapidoth, "Nearest neighbor decoding for additive non-Gaussian noise channels," *IEEE Transactions on Information Theory*, Vol. 42(5), pp. 1520-1529, 1996.

With the two roadblocks resolved, we have now justified the expression in Equation (229) as a rate of reliable communication that is possible to achieve over the wireline channel.[7] Since we are operating at low SNR, we can approximate the rate of reliable communication by this first order approach (cf. Equation (228)) by

$$C \approx \frac{1}{2}\left(\log_2 e\right) h_0^2 \mathsf{SNR}. \tag{230}$$

[7]Strictly speaking, the first L symbols of each of the L sub-packets do not see the same interference as rest of the symbols. However, the fraction of these L symbols to the total data is negligible.

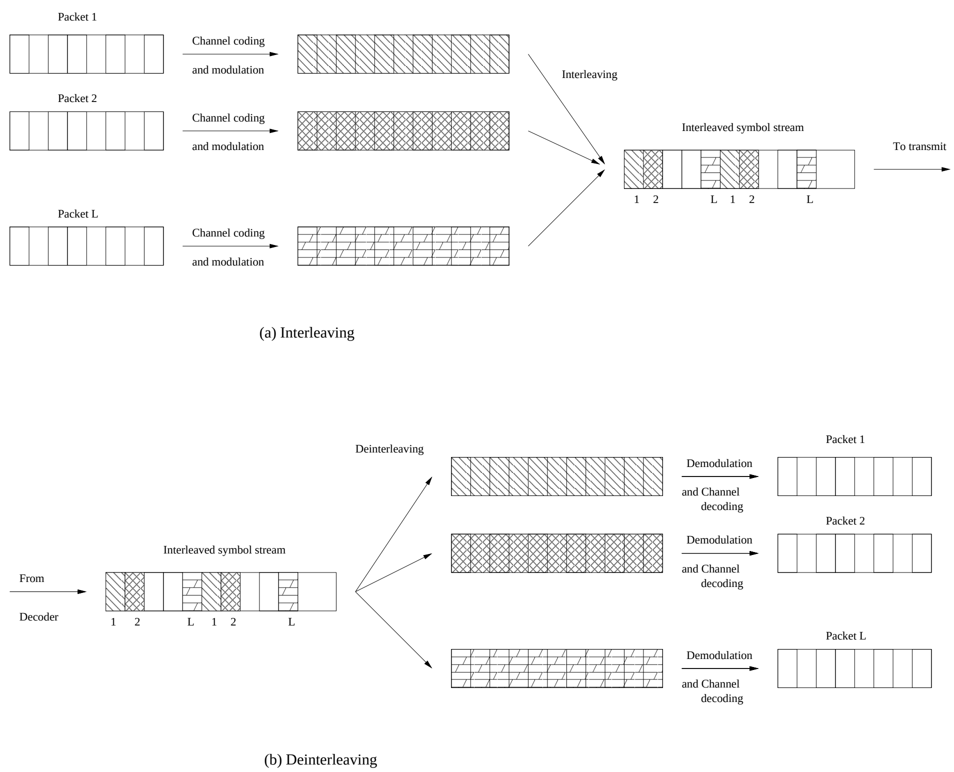

Figure 23: Interleaving and deinterleaving.

Matched Filter : A Refined Approach

Can we reliably communicate at better rates even while continuing to treat interference as noise? The key to answer this question is to recall that we are power limited at low SNRs (cf. Lecture 8a). So any boost in the received SNR will translate directly and linearly into a boost in the rate of reliable communication. How do we boost the received SNR of any symbol? A careful look at the received voltage (cf. Equation (217)) shows that we are receiving L different copies of *each* transmitted symbol. These copies are scaled differently and shifted in time (example: symbol $x[1]$ will appear in $y[1], \cdots, y[L]$). The scenario is similar to repetition coding except that now each copy of the transmitted symbol is scaled differently and sees different interference. However, since we are in low SNR regime, noise dominates the interference and we are justified in ignoring the interference. Consider the following "approximate" channel where we have just removed the interference altogether.

$$y[m+l] \approx h_l x[m] + w[m+l] \quad l = 0, 1, \cdots, L-1. \tag{231}$$

A natural desire is to combine these received voltages so as to maximize the SINR for $x[m]$. Since the noise dominates in the low SNR regime, maximizing SINR is same as maximizing SNR. To be concrete and explicit, let us restrict our strategies to the *linear combinations* of the received voltages:

$$\hat{y}[m] \stackrel{\text{def}}{=} \sum_{l=0}^{L-1} c_l y[m+l] \tag{232}$$

$$= \sum_{l=0}^{L-1} c_l h_l x[m] + \sum_{l=0}^{L-1} c_l w[m+l]. \tag{233}$$

The idea is to choose the vector $\mathbf{c}$, i.e., the coefficients $c_0, \ldots, c_{L-1}$ that maximizes the resulting SNR of the estimate:

$$\mathsf{SNR}(\mathbf{c}) \stackrel{\text{def}}{=} \frac{\left(\sum_{l=0}^{L-1} c_l h_l\right)^2 \mathbb{E}\left[x[m]\right]^2}{\left(\sum_{l=0}^{L-1} c_l^2\right) \sigma^2} \tag{234}$$

$$= \frac{\left(\mathbf{c}^T \mathbf{h}\right)^2}{\mathbf{c}^T \mathbf{c}} \mathsf{SNR}. \tag{235}$$

Here we have written $\mathbf{h}$ to be the vector $(h_0, \ldots, h_{L-1})^T$ of channel tap coefficients.

Key Optimization Problem

What is the solution to the optimization problem

$$\max_{\mathbf{c}} \quad \frac{\left(\mathbf{c}^T \mathbf{h}\right)^2}{\mathbf{c}^T \mathbf{c}}? \tag{236}$$

To get a feel for this problem, first observe that the *magnitude* of the vector of combining coefficients $\mathbf{c}$ has no role to play: the function being maximized in the problem in Equation (236) is the *same* for all arguments $a\mathbf{c}$, as long as the scaling coefficient $a \neq 0$. So, without loss of generality let us suppose that

$$\mathbf{c}^T\mathbf{c} = 1. \tag{237}$$

Now we have a slightly refined picture of the optimization problem in Equation (236). We aim to maximize the magnitude of the inner product between a fixed vector $\mathbf{h}$ and a unit magnitude vector $\mathbf{c}$. Figure 24 depicts the situation pictorially. The solution is fairly obvious there: we should align the unit magnitude vector $\mathbf{c}$ in the *direction* of $\mathbf{h}$. In other words, we should choose

$$\mathbf{c} = \frac{\mathbf{h}}{\|\mathbf{h}\|}. \tag{238}$$

Here we have denoted the Euclidean magnitude $\sqrt{\mathbf{h}^T\mathbf{h}}$ of the vector $\mathbf{h}$ by $\|\mathbf{h}\|$. To use more words to say the same thing: in two dimensions, $\mathbf{c}$ lies on the unit circle. If $\mathbf{c}$ makes an angle θ with $\mathbf{h}$, then

$$|\mathbf{h}^T\mathbf{c}| = \|\mathbf{h}\||cos\theta|. \tag{239}$$

This is maximized when $\theta = 0$.

This argument holds true for $n > 2$ as well, since we can project each of the vectors $\mathbf{h}$ and $\mathbf{c}$ to the same two dimensional plane without altering the inner product between them. A more verbose, but mathematically complete, proof of this claim is derived in the appendix. To conclude, the SNR in Equation (235) is optimized by choosing the coefficient as in Equation (238). Since the combining coefficients are *matched* to the channel coefficients, this operation is also called a *matched filter*.

Now let us revert back to the original wireline channel of Equation (217). This time we will account for the interference as well. Denoting $\hat{y}_{MF}[m]$ as the output of the matched filter at time index m, we have

$$\begin{aligned}
\hat{y}_{\mathrm{MF}}[m] &\stackrel{\mathrm{def}}{=} \sum_{l=0}^{L-1} h_l y[m+l] && (240)\\
&= \sum_{l=0}^{L-1} h_l \left(\sum_{k=0}^{L-1} h_k x[m+l-k] + w[m+l] \right) && (241)\\
&= \sum_{l=0}^{L-1}\sum_{k=0}^{L-1} h_l h_k x[m+l-k] + \sum_{l=0}^{L-1} h_l w[m+l] && (242)\\
&= \left(\sum_{l=0}^{L-1} h_l^2 \right) x[m] + \sum_{l=0}^{L-1}\sum_{k=0;k\neq l}^{L-1} h_l h_k x[m+l-k] + \sum_{l=0}^{L-1} h_l w[m+l]. && (243)
\end{aligned}$$

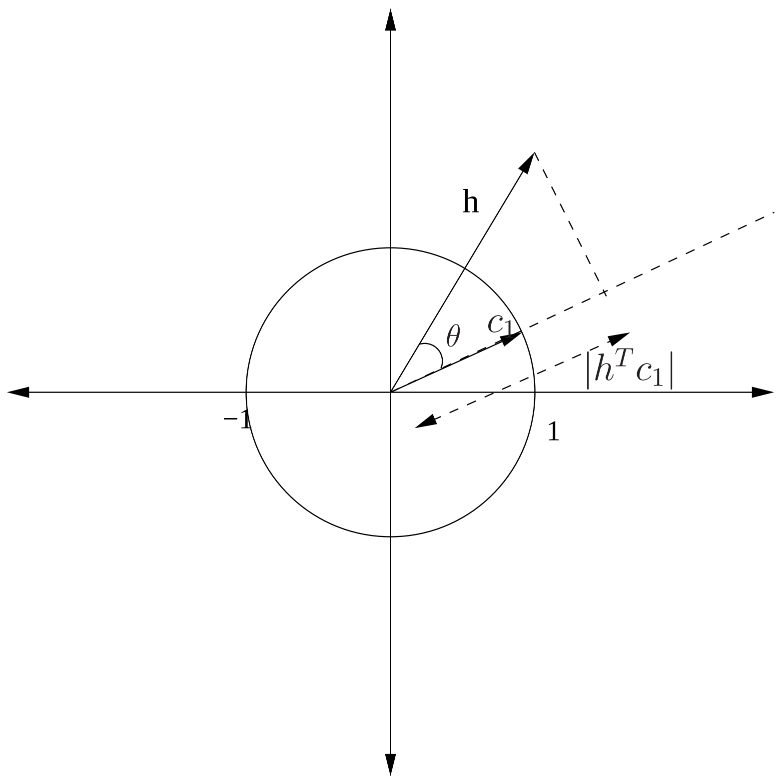

Figure 24: Pictorial explanation of Cauchy Schwarz inequality.

The first term in the last equation is the desired signal, second term is the interference and the third term is the noise. The corresponding SINR is somewhat messy, but can be worked out to be

$$\mathsf{SINR}_{\mathsf{MF}} = \frac{\left(\sum_{l=0}^{L-1} h_l^2\right)^2 \mathbb{E}(x[m]^2)}{\sum_{l=0}^{L-1} \sum_{k=0;k\neq l}^{L-1} h_l^2 h_k^2 \mathbb{E}(x[m+l-k]^2) + \left(\sum_{l=0}^{L-1} h_l^2\right)\sigma^2}. \tag{244}$$

Note that here we have used the facts that noise samples $w[m+l]$ are i.i.d. with variance σ^2 and that the interference terms are statistically independent and are also independent of noise.

We can further simplify the SINR expression.

$$\mathsf{SINR}_{\mathsf{MF}} = \frac{\left(\sum_{l=0}^{L-1} h_l^2\right)^2 E}{\left(\sum_{l=0}^{L-1} \sum_{k=0;k\neq l}^{L-1} h_l^2 h_k^2 E\right) + \left(\sum_{l=0}^{L-1} h_l^2\right)\sigma^2} \tag{245}$$

$$= \frac{\left(\sum_{l=0}^{L-1} h_l^2\right)\mathsf{SNR}}{\frac{\sum_{l=0}^{L-1} \sum_{k=0;k\neq l}^{L-1} h_l^2 h_k^2}{\sum_{l=0}^{L-1} h_l^2}\mathsf{SNR} + 1}. \tag{246}$$

The rate of reliable communication with the matched filter is now

$$\frac{1}{2}\log_2\left(1 + \mathsf{SINR}_{\mathsf{MF}}\right) \quad \text{bits/channel use.} \tag{247}$$

At low SNRs we can approximate this rate by

$$\frac{1}{2}\left(\log_2 e\right)\mathsf{SINR}_{\mathsf{MF}} \tag{248}$$

that is further approximated by

$$\frac{1}{2}\left(\log_2 e\right)\left(\sum_{\ell=0}^{L-1} h_\ell^2\right)\mathsf{SNR}. \tag{249}$$

Comparing this with the naive approach first espoused (cf. Equation (230)), we see that the rate has strictly improved: the larger the channel coefficients at later taps, the more the improvement.

Looking ahead

In the low SNR regime, interference is not much of an issue. The focus is hence on collecting as much of the signal energy as possible at the receiver; the matched filter is a natural outcome of this line of thought. In the next lecture, we look at an alternative regime: high SNR. This is the dominant mode of operation over wireline channels. Here treating interference as noise is disastrous. We will need different strategies.

Appendix: Cauchy Schwarz Inequality

We show that the solution to the optimization problem in Equation (236) is the one in Equation (238). We do this by deriving a fundamental inequality known as Cauchy-Schwarz inequality. We will now prove it formally. Fix a and define

$$\mathbf{x} \stackrel{\text{def}}{=} \mathbf{h} - a\mathbf{c}. \tag{250}$$

Now we have

$$\begin{aligned} 0 &\leq \mathbf{x}^T\mathbf{x} && (251)\\ &= (\mathbf{h} - a\mathbf{c})^T(\mathbf{h} - a\mathbf{c}) && (252)\\ &= \mathbf{h}^T\mathbf{h} - 2a\mathbf{h}^T\mathbf{c} + a^2\mathbf{c}^T\mathbf{c}. && (253)\end{aligned}$$

Since this inequality holds for all a, let us choose $a = \frac{\mathbf{h}^T\mathbf{c}}{\mathbf{c}^T\mathbf{c}}$. Substituting this choice, we arrive at

$$0 \leq \mathbf{h}^T\mathbf{h} - 2\frac{|\mathbf{h}^T\mathbf{c}|^2}{\mathbf{c}^T\mathbf{c}} + \frac{|\mathbf{h}^T\mathbf{c}|^2}{\mathbf{c}^T\mathbf{c}}. \tag{254}$$

$$\tag{255}$$

Hence

$$|\mathbf{h}^T\mathbf{c}|^2 \leq \mathbf{h}^T\mathbf{h}\mathbf{c}^T\mathbf{c} \tag{256}$$

and equality holds when $\mathbf{c} = a\mathbf{h}$.

Lecture 13: Intersymbol Interference Management: High SNR Regime

Introduction

We have seen that the key aspect of the wireline channel is the intersymbol interference (ISI), and our focus is to develop receiver techniques that will mitigate and potentially harness the effects of interference.

In Lecture 10, we focused on the low SNR regime. Here the noise power dominates the total signal power and hence the interference power. Essentially, we saw that in this regime, interference is not the issue and treating interference as noise is indeed a good strategy. The matched filter, which focusses on collecting the signal energy and ignores the interference, performs well in this regime.

The focus of this lecture is on high SNR regime. Now the total signal power, and hence the interference power, dominates the noise power. Hence, the main aspect of the receiver design is on how to handle interference. It makes no sense in this regime to naively ignore interference and treat it as noise. Matched filter will not perform well in this regime and we need to develop new techniques to deal with interference.

In this lecture, we will again adopt the receiver centric approach and will focus on simple linear processing schemes that will handle interference and convert the ISI channel into an effective AWGN channel. We can then use our analysis and codes developed for AWGN channel. In particular, we will study two schemes, viz., zero forcing equalizer (ZFE) and successive interference cancellation (SIC), that perform well in high SNR regime. Both of these schemes pay attention to the fact that interference has a structure and given the knowledge of present symbol, interference can be predicted. We will begin with a simple calculation to understand why matched filter fails in High SNR regime.

Performance of Matched Filter at High SNR

The received voltage of the wireline channel is

$$y[m] = \sum_{\ell=0}^{L-1} h_\ell x[m-\ell] + w[m], \quad m > 0. \tag{257}$$

Recall that the SINR at the output of the matched filter is

$$\mathsf{SINR}_{\mathsf{MF}} \;=\; \frac{\left(\sum_{l=0}^{L-1} h_l^2\right)\mathsf{SNR}}{\frac{\sum_{\ell=0}^{L-1}\sum_{k=0;k\neq\ell}^{L-1} h_\ell^2 h_k^2}{\sum_{\ell=0}^{L-1} h_\ell^2}\mathsf{SNR}+1}, \tag{258}$$

where

$$\mathsf{SNR} = \frac{\mathbb{E}\left[(\mathsf{x}[\mathsf{m}])^2\right]}{\sigma^2}. \tag{259}$$

Note that the SINR at the output of the matched filter depends on the operating SNR. Two different regimes are of keen interest:

- *Low* SNR: For small values of SNR, the SINR is close to linear in SNR. In other words, a doubling of the operating SNR also doubles the $\mathsf{SINR}_{\mathrm{MF}}$. Mathematically, this regime happens when the interference

$$\frac{\sum_{l=0}^{L-1}\sum_{k=0;k\neq l}^{L-1} h_l^2 h_k^2}{\sum_{l=0}^{L-1} h_l^2}\mathsf{SNR} \ll 1. \tag{260}$$

  In this regime, the interference is much smaller than the background noise and it makes fine sense to just ignore the interference. The channel is almost like an AWGN one and hence the linear relationship between SNR and $\mathsf{SINR}_{\mathrm{MF}}$.

- *High* SNR: For large values of SNR, the $\mathsf{SINR}_{\mathrm{MF}}$ is almost constant and hardly changes. Mathematically, this regime kicks in when the interference

$$\frac{\sum_{l=0}^{L-1}\sum_{k=0;k\neq l}^{L-1} h_l^2 h_k^2}{\sum_{l=0}^{L-1} h_l^2}\mathsf{SNR} \gg 1 \tag{261}$$

  and

$$\mathsf{SINR}_{\mathrm{MF}} \approx \frac{\left(\sum_{\mathsf{l}=0}^{\mathsf{L}-1} \mathsf{h}_\mathsf{l}^2\right)^2}{\sum_{\mathsf{l}=0}^{\mathsf{L}-1}\sum_{\mathsf{k}=0;\mathsf{k}\neq \mathsf{l}}^{\mathsf{L}-1} \mathsf{h}_\mathsf{l}^2 \mathsf{h}_\mathsf{k}^2}. \tag{262}$$

  In this regime, the interference is much larger than noise and we pay a steep price by just ignoring its presence. The interference level is directly proportional to the transmit signal energy and hence the $\mathsf{SINR}_{\mathrm{MF}}$ saturates to a fixed value and is insensitive to increase in SNR in this regime.

To ameliorate the deleterious presence of interference in the high SNR regime, it helps to try to eliminate the source of interference as much as possible. The zero forcing equalizer, the next topic, precisely does this: it entirely removes the interference (or forces the interference to zero, hence its name).

The Zero Forcing Equalizer

Let's consider communication over n channel uses on the L-tap wireline channel. The received voltage sequence of length $n+L-1$ is given by Equation (257). We can collect together all the n transmissions and receptions and represent them compactly in vector form. Let $\mathbf{x}$ and $\mathbf{y}$ be the vector of transmitted and received voltages respectively. Let $\mathbf{w}$ be the noise vector of i.i.d. Gaussian noises. The convolution of the channel vector $\mathbf{h}$ with the transmitted vector $\mathbf{x}$ can be written as a product of channel matrix $\mathbf{H}$ and vector $\mathbf{x}$. Rows of the channel matrix $\mathbf{H}$ are obtained from shifting $\mathbf{h}$ appropriately. To illustrate this, let's consider a concrete example of transmission of 4 symbols over a 3 tap channel, i.e., $n = 4$ and $L = 3$. We can write the Equation (257) in matrix notation as

$$\begin{bmatrix} y[1] \\ y[2] \\ y[3] \\ y[4] \\ y[5] \\ y[6] \end{bmatrix} = \begin{bmatrix} h_0 & 0 & 0 & 0 \\ h_1 & h_0 & 0 & 0 \\ h_2 & h_1 & h_0 & 0 \\ 0 & h_2 & h_1 & h_0 \\ 0 & 0 & h_2 & h_1 \\ 0 & 0 & 0 & h_2 \end{bmatrix} \begin{bmatrix} x[1] \\ x[2] \\ x[3] \\ x[4] \end{bmatrix} + \begin{bmatrix} w[1] \\ w[2] \\ w[3] \\ w[4] \\ w[5] \\ w[6] \end{bmatrix}. \tag{263}$$

Thus, we can write Equation (263) in short hand notation as,

$$\mathbf{y} = \mathbf{H}\mathbf{x} + \mathbf{w} \tag{264}$$

where $\mathbf{x}$ is a n dimensional vector, $\mathbf{y}$ and $\mathbf{w}$ are $n+L-1$ dimensional vectors. The channel matrix $\mathbf{H}$ is an $n+L-1 \times n$ dimensional matrix.

If there were no noise, then we have $n+L-1$ equations in n unknowns. This is an over specified system of equations (more equations than unknowns). Then we can choose any n independent equations to solve for $\mathbf{x}$. We can choose these n equations in many ways and all the choices are equivalent.

With noise $\mathbf{w}$ present in the channel, however, the *choice* of the independent equations in solving for $\mathbf{x}$ matters. Each of the choices will still be able to remove the interference completely, but leave behind noises of *different* variances (the noise left behind is just the linear combinations of $w[m]$s). The SNR of the resulting AGN channel now depends on our choice of the equations we use to obtain the interference-free observations.

The zero forcing equalizer (ZFE) with respect to one of the transmit voltages, say $x[1]$, is that linear combination of the observed voltages which satisfies two properties:

- it leads to an interference-free observation (but still corrupted by additive Gaussian noise) of $x[1]$;
- it has the largest output SNR in the resulting AWGN channel with respect to $x[1]$.

Now we see why the ZFE is much better than the matched filter at high SNRs: ZFE cancels the interference completely (at the cost of potentially changing the noise variance; but this change is independent of the operating transmit SNR). Thus, as we increase the signal power, the effective received SNR scales *linearly* with the signal power. This is in stark contrast to the performance of the matched filter which had a ceiling for received SNR no matter how much the transmit power is increased.

Example

To get a concrete feel, let us consider the ISI channel model by considering just a single interfering symbol, i.e., $L = 2$:

$$y[m] = h_0 x[m] + h_1 x[m-1] + w[m], \quad m \geq 1. \tag{265}$$

Let's also restrict the transmission to 3 symbols, i.e., $n = 3$. So we are considering

$$\begin{aligned} y[1] &= h_0 x[1] + w[1] && (266) \\ y[2] &= h_1 x[1] + h_0 x[2] + w[2] && (267) \\ y[3] &= h_1 x[2] + h_0 x[3] + w[3] && (268) \\ y[4] &= h_1 x[3] + w[4]. && (269) \end{aligned}$$

Again, let us consider sequential communication starting at $m = 1$. Since $x[0] = 0$, there is *no* ISI at the first time sample:

$$y[1] = h_0 x[1] + w[1]. \tag{270}$$

The first transmit voltage $x[1]$ also features in the voltage received at the second time sample:

$$y[2] = h_1 x[1] + h_0 x[2] + w[2], \tag{271}$$

but this time it is corrupted by interference ($h_0 x[2]$). At high operating SNRs, the interference dominates the additive Gaussian noise and $y[1]$ is far more valuable in decoding the information contained in $x[1]$ than $y[2]$. We could use $y[1]$ alone to decode the information bits at the first time sample: this is just a simple AWGN channel with SNR equal to

$$h_0^2 \mathsf{SNR}. \tag{272}$$

The important conclusion is that it increases linearly with the operating SNR level regardless of whether the SNR value is large or small. Indeed, we can expect this simple receiver to outperform the matched filter at large SNR operating levels.

Well, getting to $x[1]$ without any interference was easy, but how about $x[2]$? Now we could use the two linear equations

$$\begin{aligned} y[1] &= h_0 x[1] + w[1] & (273) \\ y[2] &= h_1 x[1] + h_0 x[2] + w[2], & (274) \end{aligned}$$

to "eliminate" the interfering term $x[1]$: consider the linear combination

$$h_0 y[2] - h_1 y[1] = h_0^2 x[2] + (h_0 w[2] - h_1 w[1]) . \tag{275}$$

This leads to a plain AWGN channel with respect to the transmit voltage $x[2]$. Since we do not have any interference from the previous symbol $x[1]$, the SNR of this AWGN channel is simply

$$\frac{h_0^4}{h_0^2 + h_1^2} \mathsf{SNR} \tag{276}$$

where SNR is, as usual, the ratio of the transmit energy to that of the additive Gaussian noise. Again, observe the linear relation to SNR regardless of whether it is high or low.

Now, the pattern is clear: to decode the information in $x[3]$ we consider the linear combination

$$h_0^2 y[3] - h_1 (h_0 y[2] - h_1 y[1]) . \tag{277}$$

Substituting from Equation (275) we see that

$$h_0^2 y[3] - h_1 (h_0 y[2] - h_1 y[1]) = h_0^3 x[3] + \left(h_0^2 w[3] + h_1^2 w[1] - h_0 h_1 w[2] \right), \tag{278}$$

an AWGN channel with respect to the transmit voltage $x[3]$. The SNR at the output of this AWGN channel is

$$\frac{h_0^6}{h_0^4 + h_1^4 + h_0^2 h_1^2} \mathsf{SNR}, \tag{279}$$

again depending linearly with respect to the operating SNR.

As we have already noted, it is not the only way to do this. Specifically, we could take *different* linear combinations of the received voltages that also lead to the desired zero interference characteristic. We could then pick that linear combination which yields the largest output SNR. To see this concretely, suppose we wanted to use the ISI channel only for three time instants $m = 1, 2, 3$. This means that we observe $x[3]$ without interference from $y[4]$:

$$y[4] = h_1 x[3] + w[4], \tag{280}$$

an AWGN channel with output SNR equal to

$$h_1^2 \mathsf{SNR}. \tag{281}$$

Depending on whether this SNR value is larger or smaller than the one in Equation (279), we could use either just $y[4]$ alone or $h_0^2 y[3] - h_1\left(h_0 y[2] - h_1 y[1]\right)$ to decode the information in $x[3]$; both of them are interference-free observations of $x[3]$. To be concrete, this comparison comes down to the following: if

$$\left(h_1^2 + h_0^2\right)\left(h_1^4 + h_0^4\right) \geq 2h_0^6, \tag{282}$$

then we would prefer to use $y[4]$ to decode the information in $x[3]$. Else, we prefer $h_0^2 y[3] - h_1 h_0 y[2] + h_1^2 y[1]$.

Analogously, we could have used the linear combination

$$h_1 y[3] - h_0 y[4] = h_1^2 x[2] + \left(h_1 w[3] - h_0 w[4]\right) \tag{283}$$

to decode the information in the transmit voltage $x[2]$. The corresponding output SNR is

$$\frac{h_1^4}{h_0^2 + h_1^2}\mathsf{SNR}. \tag{284}$$

This is an alternative to the linear combination used in Equation (275) and we see that the output SNRs of the two linear combinations are not the same (cf. Equation (276)). We would use the linear combination in Equation (275) if

$$h_0^2 \geq h_1^2 \tag{285}$$

and the linear combination in Equation (283) otherwise.

Finally, we could have used the linear combination

$$h_1^2 y[2] - h_0\left(h_0 y[4] - h_1 y[3]\right) = h_1^3 x[1] + \left(h_1^2 w[2] + h_0 h_1 w[3] - h_0^2 w[4]\right) \tag{286}$$

to decode the information in $x[1]$. The output SNR of this AWGN channel is

$$\frac{h_1^6}{h_0^4 + h_0^2 h_1^2 + h_1^4}\mathsf{SNR}. \tag{287}$$

Comparing this with the earlier choice of just using $y[1]$ to decode the information in $x[1]$, we see that we would prefer this simple choice (as compared to the linear combination in Equation (286)) provided

$$\frac{h_1^6}{h_0^4 + h_0^2 h_1^2 + h_1^4} \leq h_0^2. \tag{288}$$

Indeed, due to the similarity of the transmit voltages $x[1]$ and $x[3]$ (the first and last transmissions), this condition is also similar to that in Equation (282):

$$\left(h_1^2 + h_0^2\right)\left(h_1^4 + h_0^4\right) \geq 2h_1^6. \tag{289}$$

Successive Interference Cancelation

The ZFE is a linear processing technique to remove interference. If we allow *nonlinear* processing, there are alternative interference removal strategies. The most prominent of them is the so-called *successive interference cancelation* (SIC) strategy. It is different from the ZFE in two important ways:

1. it is a nonlinear operation on the received voltage symbols;

2. while ZFE works on removing *any* interference voltages, the SIC will only work when the interference voltages are the outputs of a good coding and modulation scheme. The SIC harnesses the fact that the transmit symbols are not any choice of voltages, but there is a clear structure and redundancy built into them.

We start out with a naive version. Once we fully understand it, we can build on top of the key idea. Observe that the very first transmitted symbol does not see any interference. If we can decode the first symbol reliably (probability of which is high in the high SNR regime), then we can subtract the interference caused by the first symbol from the successive received voltages. Thus, the second symbol, which had interference from the first symbol alone, now sees an AWGN channel. This symbol can be decoded reliably and interference that it has caused to the further symbols can be canceled. Thus, each symbol now sees only the additive noise.

Let's consider an example to see how successive interference cancelation (SIC) works. Consider L-tap wireline channel. The first few received voltages are

$$\begin{aligned} y[1] &= h_0 x[1] + w[1] && (290)\\ y[2] &= h_0 x[2] + h_1 x[1] + w[2] && (291)\\ y[3] &= h_0 x[3] + h_1 x[2] + h_2 x[1] + w[3] && (292) \end{aligned}$$

and so on.

Since, the first symbol does not see any ISI, we can simply decode it using ML receiver. Let $\hat{x}[1]$ be the decoded symbol. Note that $x[1]$ sees an AWGN channel with

$$\mathsf{SNR}_{\mathsf{effective}} = \mathsf{h}_0^2 \mathsf{SNR}. \tag{293}$$

Assuming that we have decoded $x[1]$ correctly, we subtract its effect from $y[2]$ before decoding $x[2]$.

$$\tilde{y}[2] = y[2] - h_1 \hat{x}[1] = h_0 x[2] + h_1 (x[1] - \hat{x}[1]) + w[2] \tag{294}$$

Thus, if $\hat{x}[1] = x[1]$, we have successfully removed the interference. $x[2]$ now sees an AWGN channel with

$$\mathsf{SNR}_{\mathsf{effective}} = \mathsf{h}_0^2 \mathsf{SNR}. \tag{295}$$

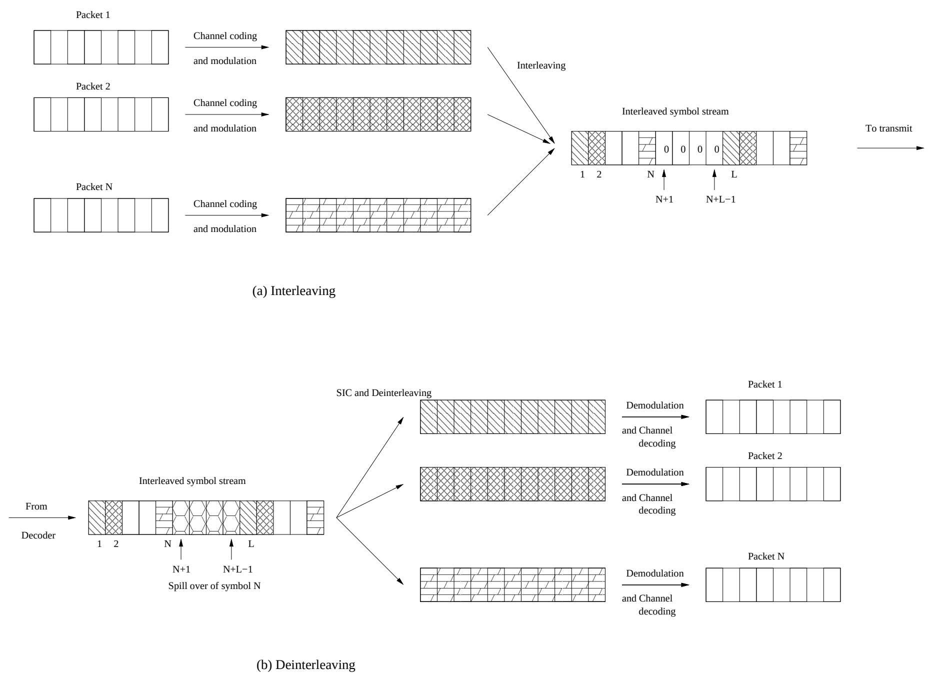

Figure 25: Interleaving and zero padding for SIC

We now use $\hat{x}[1]$ and $\hat{x}[2]$ to cancel the interference from $y[3]$. Thus, each of the symbol sees an interference-free channel with

$$\mathsf{SNR}_{\mathsf{SIC}} = \mathsf{h}_0^2\mathsf{SNR}. \tag{296}$$

Note that the effective SNR for SIC grows linearly with SNR.

However, this simple scheme has a caveat. Here we have assumed that we always decode the symbols correctly. If $\hat{x}[1] \neq x[1]$, then we are actually *adding* interference to $y[2]$ and thus, the probability of error in decoding $x[2]$ is now quite high. This is particularly true in the high SNR regime. Now given that $x[2]$ is decoded wrongly, an analogous argument suggests that $x[3]$ will also be decoded wrongly with a very high probability. Thus, the errors propagate in this scheme. So no matter how small $P(\varepsilon)$ is, as long as $P(\varepsilon) > 0$, we cannot achieve arbitrary reliability.

However, this problem can be easily overcome by using long block codes along with interleaving. We know that for an AWGN channel, we can achieve arbitrary reliability with long block codes. We will now split the data into N $(N > L)$ subpackets which are coded and modulated separately using capacity achieving codes. We then interleave one symbol from each of the N subpackets and pad these N symbols with $L-1$ zeros. Because of this zero padding, all the symbols from the first subpackets do not see any interference. We can then decode the entire first subpacket with arbitrary reliability. We then use the decoded subpacket to remove the interference from the received symbols of subsequent subpackets. Since each of the symbols of each subpacket is decoded with arbitrary reliability, the risk of error propagation is not there anymore. Note that this scheme has the overhead associated with zero padding. We pad $L-1$ zeros to each of the N symbols. Thus, the channel is used only for the $\frac{N}{N+L-1}$ fraction of time. But we can use $N \gg L-1$ and $\frac{N}{N+L-1} \approx 1$ as N grows. Figure 25 explains this scheme pictorially.

Note that SIC is different from the other equalizers we have studied so far. Note that matched filter and ZFE involve processing the received symbols *before* decoding. Decoding takes place only with the output of the equalizers. But SIC goes back and forth between decoding and processing received voltages. SIC can be coupled with the other equalizers to improve the performance. We will see the full power of the SIC receiver in the next lecture.

Looking Ahead

We have seen that the matched filter worked well at low values of SNR, while the zero forcing equalizer worked well at high values of SNR. In the next lecture, we will focus on the metric of the SNR at the output of the equalizer to derive the *optimal* filter: this is the so-called MMSE (minimum mean squared error) equalizer. It strictly outperforms both the matched filter and the ZFE and is commonly used in practical communication systems. It turns out that MMSE combined with SIC achieves the *capacity* (the fundamental limit on the rate of reliable communication) of the wireline channel.

Lecture 14: Interference Management at all SNRs: Minimum Mean Square Error (MMSE) Filter

Introduction

In the previous lectures, we have seen two linear processing techniques that deal with ISI: the matched filter (which worked well at low SNRs) and the zero forcing equalizer (which worked well at high SNRs). The matched filter harnessed the multiple delayed copies of each transmit symbol. The zero forcing equalizer worked to entirely null all the multiple delayed copies (and hence zero out the interference). In this lecture we study the *optimal* linear filter that balances both these effects: the need to harness the multiple delayed copies while still mitigating the effect of interference. As usual, our performance metric will be the SINR of the transmit symbol at the output of the filter. This filter, known as the minimum mean square error (MMSE) equalizer, in conjunction with the successive interference cancelation (SIC) technique, is routinely used in practical communication schemes over wireline channels (an example: the voiceband telephone line modem).

We have seen that the matched filter ignores the interference and combines the received voltage points to collect the signal energy. This strategy works well at low SNRs. In particular, the SINR at the output of matched filter is of the form

$$\mathsf{SINR}_{\mathsf{MF}} = \frac{\mathsf{aSNR}}{1 + \mathsf{bSNR}} \tag{297}$$

where a and b are the appropriate constants (that are independent of SNR).

The zero forcing equalizer works well in high SNR regime. It essentially ignores noise and removes ISI by solving an overspecified set of linear equations. The SINR at the output of ZFE is of the form

$$\mathsf{SINR}_{\mathsf{ZFE}} = \mathsf{cSNR} \tag{298}$$

where c is the appropriate constant.

Both these equalizers are simple linear processing that perform well in different SNR regimes. In this lecture, we study a linear processing strategy that combines the best of both the equalizers and strictly outperform them in both regimes. We will motivate it by studying an example.

Getting Started

To motivate the design of this filter, consider the simple 2-tap ISI channel:

$$y[m] = h_0 x[m] + h_1 x[m-1] + w[m], \quad m \geq 1. \tag{299}$$

Suppose we communicate information sequentially, say one bit at a time: so $x[m]$ is $\pm\sqrt{E}$ for each m and independent over different time samples. We want to filter only the first two received voltages $y[1], y[2]$ to estimate the first transmit voltage $x[1]$:

$$y[1] = h_0 x[1] + w[1] \quad (300)$$
$$y[2] = h_1 x[1] + (h_0 x[2] + w[2])\,. \quad (301)$$

The matched filter receiver would be

$$y^{\mathrm{MF}}[1] = h_0 y[1] + h_1 y[2], \quad (302)$$

while the ZFE would be

$$y^{\mathrm{ZFE}}[1] = y[1]. \quad (303)$$

We are interested in choosing a_1, a_2 such that using the filter

$$y^{\{a_1,a_2\}}[1] \stackrel{\mathrm{def}}{=} a_1 y[1] + a_2 y[2] \quad (304)$$
$$= (a_1 h_0 + a_2 h_1)\, x[1] + (a_1 w[1] + a_2 h_0 x[2] + a_2 w[2]) \quad (305)$$

has the largest SINR at its output in terms of the transmit voltage $x[1]$:

$$\mathsf{SINR}_{\{a_1,a_2\}} \stackrel{\mathrm{def}}{=} \frac{(a_1 h_0 + a_2 h_1)^2\, \mathsf{SNR}}{a_1^2 + a_2^2 + a_2^2 h_0^2 \mathsf{SNR}}. \quad (306)$$

Here we have denoted the ratio of E to σ^2 by SNR, as usual. There is an alternative nomenclature for the filter that has the largest SINR: the MMSE equalizer. Why the filter that has the largest SINR also has the "minimum mean squared error" is explored in a homework exercise.

To get a feel for the appropriate choice of the filter coefficients a_1 and a_2, consider the following somewhat abstract representation of Equations (300) and (301):

$$y[1] = h_0 x[1] + z[1] \quad (307)$$
$$y[2] = h_1 x[1] + z[2]. \quad (308)$$

Specifically, in terms of Equations (300) and (301), we have

$$z[1] = w[1] \quad (309)$$
$$z[2] = h_0 x[2] + w[2]. \quad (310)$$

Observe that the "interference plus noise" terms $z[1], z[2]$ are statistically uncorrelated (even statistically independent in this case). Let us denote the variance of $z[i]$ by $\sigma^2_{z[i]}$ for $i = 1, 2$. In the context of Equations (300) and (301) we have

$$\sigma^2_{z[1]} = \sigma^2 \quad (311)$$
$$\sigma^2_{z[2]} = h_0^2 E + \sigma^2. \quad (312)$$

Using the linear filter in Equation (304):

$$y^{\{a_1,a_2\}}[1] = (a_1h_0 + a_2h_1)\,x[1] + (a_1z[1] + a_2z[2])\,, \tag{313}$$

we see that the signal energy is

$$(a_1h_0 + a_2h_1)^2\,E. \tag{314}$$

On the other hand, interference plus noise energy is

$$a_1^2\sigma_{z[1]}^2 + a_2^2\sigma_{z[2]}^2, \tag{315}$$

where we used the property that $z[1], z[2]$ are uncorrelated. Thus the output SINR is

$$\mathsf{SINR}_{\{a_1,a_2\}} = \frac{(a_1h_0 + a_2h_1)^2\,E}{\left(a_1^2\sigma_{z[1]}^2 + a_2^2\sigma_{z[2]}^2\right)}. \tag{316}$$

The MMSE equalizer chooses a_1 and a_2 that will maximize SINR. Note that only the *ratio* of the linear coefficients a_1, a_2 decides the SINR. But how do the matched filter (MF) and ZFE choose these coefficients?

We know that the matched filter ignores the interference. Thus, it assumes $z[1]$ and $z[2]$ to have same variance σ^2. We have seen from Cauchy-Schwarz inequality that the coefficients that maximize the SNR are $a_1 = h_0$ and $a_2 = h_1$. We conclude:

> The matched filter would be indeed the MMSE equalizer, i.e., possess the largest output SINR among all linear filters, if the interference plus noise terms are uncorrelated and have the same variance.

On the other hand, ZFE ignores the noise and treats Equations (300) and (301) as "deterministic". Clearly, it chooses only $y[1]$ to decode $x[1]$ and hence $a_1 = 1$ and $a_2 = 0$.

Unlike the matched filter, MMSE equalizer maximizes SINR even when $\sigma_{z[1]}^2$ and $\sigma_{z[2]}^2$ are not necessarily the same. The key idea in deriving this filter is to *scale* the received voltages such that the scaled interference plus noise terms have the same variance. Consider

$$\tilde{y}[1] \stackrel{\text{def}}{=} y[1] = h_0x[1] + z[1] \tag{317}$$

$$\tilde{y}[2] \stackrel{\text{def}}{=} cy[2] = ch_1x[1] + cz[2]. \tag{318}$$

we choose c such that $z[1]$ and $cz[2]$ have the same variance, i.e.,

$$c^2(h_0^2E + \sigma^2) = \sigma^2 \tag{319}$$

$$c^2 = \frac{1}{1 + h_0^2\mathsf{SNR}}. \tag{320}$$

Now both the $z[1]$ and $cz[2]$ have variance σ^2. We know from Lecture 10 that the matched filter of the scaled received voltages will maximize the SINR. Hence,

$$y_{\text{MMSE}}[1] = h_0\tilde{y}[1] + ch_1\tilde{y}[2] \tag{321}$$

$$= h_0 y[1] + c^2 h_1 y[2]. \tag{322}$$

Thus, the optimal choice of the coefficients a_1 and a_2 are

$$a_1^* = h_0, \quad \text{and} \quad a_2^* = c^2 h_1 = \frac{h_1}{1 + h_0^2\mathsf{SNR}}. \tag{323}$$

It is instructive to check what happens to the MMSE filter coefficients at high and low SNRs:

- At low SNRs, when $\mathsf{SNR} \ll 1$, we see from Equation (323) that

  $$a_1 = h_0 \tag{324}$$

  $$a_2 \approx h_1. \tag{325}$$

  In other words, the MMSE filter is just the regular *matched filter* at low SNRs.

- At high SNRs, when $\mathsf{SNR} \gg 1$, we see from Equation (323) that

  $$a_1 = h_0 \tag{326}$$

  $$a_2 \approx 0. \tag{327}$$

  In other words, the MMSE filter is just the regular *zero forcing equalizer* at high SNRs.

This formally justifies our intuition (from Lectures 10 and 11) that the matched filter and zero forcing equalizers were performing quite well at low and high SNRs respectively.

From Equation (316), the SINR achieved by the MMSE equalizer is

$$\mathsf{SINR}_{\text{MMSE}} = \frac{\left(a_1^* h_0 + a_2^* h_1\right)^2 E}{((a_1^*)^2\sigma^2 + (a_2^*)^2(h_0^2 E + \sigma^2))} \tag{328}$$

$$\mathsf{SINR}_{\text{MMSE}} = \frac{(h_0^2 + c^2 h_1^2)^2\mathsf{SNR}}{(h_0^2 + c^2 h_1^2)} \tag{329}$$

$$= (h_0^2 + c^2 h_1^2)\mathsf{SNR} \tag{330}$$

$$= h_0^2\mathsf{SNR} + \frac{\mathsf{h}_1^2\mathsf{SNR}}{1 + \mathsf{h}_0^2\mathsf{SNR}}. \tag{331}$$

Note that the first term corresponds to the SINR that ZFE provides, whereas the second term can be seen as a contribution of the matched filter. Thus, MMSE combines both the equalizers and outperforms them at all SNRs.

A General Derivation

While our derivation of the MMSE filter was very concrete, it was for a rather special scenario where the interference plus noise terms were uncorrelated and all we had to do was just normalize the variances of the noise plus interference terms. More generally, they would be correlated – for example, consider the 3-tap ISI channel:

$$y[m] = h_0 x[m] + h_1 x[m-1] + h_2 x[m-2] + w[m] \tag{332}$$

where we want to estimate $x[1]$ from $y[1], y[2], y[3]$:

$$\begin{aligned} y[1] &= h_0 x[1] + w[1] && (333)\\ y[2] &= h_1 x[1] + (h_0 x[2] + w[2]) && (334)\\ y[3] &= h_2 x[1] + (h_0 x[3] + h_1 x[2] + w[3])\,. && (335) \end{aligned}$$

It helps to write the received voltages in a *vector* form:

$$\begin{bmatrix} y[1] \\ y[2] \\ y[3] \end{bmatrix} = \begin{bmatrix} h_0 \\ h_1 \\ h_2 \end{bmatrix} x[1] + \begin{bmatrix} z[1] \\ z[2] \\ z[3] \end{bmatrix}, \tag{336}$$

where $z[1], z[2], z[3]$ are the interference plus noise terms. If $z[1], z[2], z[3]$ are all uncorrelated, then we could have derived the MMSE filter by:

- scaling the received voltages so that the variances of $z[1], z[2], z[3]$ are all the same;
- then matched filtering the scaled received voltages.

But in this case, we see that though $z[1]$ is independent of $z[2]$ and $z[3]$, $z[2]$ and $z[3]$ are correlated since the transmit voltage $x[2]$ appears in both of them. How does one derive the MMSE filter for such a situation?

It would help if we can "whiten" $y[2]$ and $y[3]$, i.e., ensure that the interference plus noise terms in them are uncorrelated. We should exploit the correlation between $z[2]$ and $z[3]$. Ideally we would like to get, by appropriate linear processing of y_2 and y_3,

$$\hat{y}[3] = \alpha x[1] + \hat{z}[3] \tag{337}$$

such that $z[1], z[2]$ and $\hat{z}[3]$ are uncorrelated. Here α is the coefficient that naturally arises when this process is completed (and, as we will see below, is uniquely defined).

The correlation between $z[2]$ and $z[3]$ is

$$\begin{aligned} \mathbb{E}\left[z[2]z[3]\right] &= \mathbb{E}\left[h_0 h_1 (x[2])^2\right] && (338)\\ &= h_0 h_1 E. && (339) \end{aligned}$$

Let's obtain $\hat{y}[3]$ by linear processing between $y[2]$ and $y[3]$.

$$\begin{aligned} \hat{y}[3] &= y[3] - ay[2] && (340)\\ &= (h_2 - ah_1)x[1] + (z[3] - az[2]). && (341) \end{aligned}$$

Let $\hat{z}[3] \stackrel{\text{def}}{=} (z[3] - az[2])$. We want to choose a such that $z[2]$ and $\hat{z}[3]$ are uncorrelated, i.e., we need

$$\begin{aligned} \mathbb{E}\left[z[2]\hat{z}[3]\right] &= 0 && (342)\\ \mathbb{E}\left[z[2]z[3]\right] &= a\mathbb{E}\left[(z[2])^2\right] && (343)\\ h_0h_1E &= a(h_0^2E + \sigma^2) && (344)\\ a &= \frac{h_0h_1E}{h_0^2E + \sigma^2} = \frac{h_0h_1\mathsf{SNR}}{h_0^2\mathsf{SNR} + 1}. && (345) \end{aligned}$$

With this choice of a, we have that

$$\mathbb{E}\left[z[2]\hat{z}[3]\right] = 0. \quad (346)$$

Further, the variance of $\hat{z}[3]$ is

$$\mathbb{E}\left[(\hat{z}[3])^2\right] = \mathbb{E}\left[(z[3] - az[2])^2\right] \quad (347)$$

Substituting for a and $\mathbb{E}\left[z[2]z[3]\right]$, we get

$$\begin{aligned} \mathbb{E}\left[(\hat{z}[3])^2\right] &= (h_0^2 + h_1^2)E + \sigma^2 - \frac{h_0^2h_1^2E^2}{(h_0^2E + \sigma^2)} && (348)\\ &= (h_0^2E + \sigma^2) + \frac{h_1^2E\sigma^2}{(h_0^2E + \sigma^2)}. && (349) \end{aligned}$$

We now have three noisy observations of $x[1]$, namely, $y[1], y[2]$ and $\hat{y}[3]$. These are corrupted by the noises $z[1], z[2]$ and $\hat{z}[3]$ that are uncorrelated. We now just need to normalize their variances and then choose the coefficients of $x[1]$ to be the optimal choice of a_1, a_2 and a_3 (matched filtering).

Thus, we have

$$\begin{aligned} \tilde{y}[1] = {} & y[1] = h_0x[1] + z[1] && (350)\\ \tilde{y}[2] = {} & c_2y[2] = c_2h_1x[1] + c_2z[2] && (351)\\ \tilde{y}[3] = {} & c_3\hat{y}[3] = c_3(h_2 - ah_1)x[1] + c_3\hat{z}[3] && (352) \end{aligned}$$

where c_2 and c_3 are chosen to normalize the variances. As before

$$\begin{aligned} c_2^2(h_0^2E + \sigma^2) &= \sigma^2 && (353)\\ c_2^2 &= \frac{1}{1 + h_0^2\mathsf{SNR}}. && (354) \end{aligned}$$

Here c_3 is such that

$$c_3^2 \mathbb{E}\left[(\hat{z}[3])^2\right] = \sigma^2 \quad (355)$$

$$c_3^2 = \frac{1 + h_0^2 \mathsf{SNR}}{(1 + h_0^2 \mathsf{SNR})^2 + \mathsf{h}_1^2 \mathsf{SNR}}. \quad (356)$$

MMSE equalizer is then

$$y_{\text{MMSE}}[1] = h_0 \tilde{y}[1] + c_2 h_1 \tilde{y}[2] + c_3 (h_2 - a h_1) \tilde{y}[3]. \quad (357)$$

More generally, when we have an arbitrary number of taps L (not simply less than 4) and when we are interested in a general transmit voltage $x[m]$ (not just for $m = 1$), there is a straightforward procedure to derive the MMSE filter without resorting to numerical optimization. This procedure is best explained using tools from linear algebra and is explored in some of your reference books.

Looking Ahead

So far, we have let the receiver handle much of the burden of mitigating ISI. In the next lecture, we will see how the transmitter can also work to address the ISI issue. Specifically, the interference is from symbols sent in the past and, surely, the transmitter is already aware of what interference it has caused. If the transmitter is savvy about what it transmits, then the ISI can be controlled to be within acceptable limits. Such transmitter strategies (that go beyond the sequential and block communication schemes we have seen in the context of the AWGN channel) are known as *precoding*. We will see this next.

Lecture 15: Transmitter-Centric ISI Management: Precoding

Introduction

The ISI channel with L taps,

$$y[m] = \sum_{\ell=0}^{L-1} h_\ell x[m-\ell] + w[m] \quad (358)$$

can alternatively be represented as

$$y[m] = h_0 x[m] + I[m] + w[m], \quad (359)$$

where

$$I[m] = \sum_{\ell=1}^{L-1} h_\ell x[m-\ell] \quad (360)$$

is the ISI. So far, we have looked at how the receiver can work to overcome the impact of ISI; all this while, the transmitter was continuing with sequential communication. In this lecture, we will take a different line of attack towards mitigating ISI: we do this by enlisting the transmitter as well in our efforts. The key point is that the ISI caused at time m, denoted by $I[m]$, is known *noncausally* at the transmitter:

- the channel tap coefficients $h_1, \dots, h_{L-1}$ are constant and known to the transmitter ahead of time;
- the previously transmitted symbols $x[m-1], \dots, x[m-L+1]$ are surely known to the transmitter.

Thus, the transmitter is in a position to *adapt* its present transmitted symbol $x[m]$ as a *function* of what ISI is being caused. Such an adaption is known in the literature as *precoding* and in this lecture we will investigate the most basic form of precoding.

A Simple Precoding Strategy

Since the transmitter *knows* what the ISI is going to be, the simplest idea is to just cancel it off: if we wanted to send a voltage $d[m]$ on an AWGN channel (with no ISI) we could transmit

$$x[m] = d[m] - \frac{I[m]}{h_0}. \quad (361)$$

This way the received voltage is simply

$$y[m] = h_0 d[m] + w[m], \quad (362)$$

a regular AWGN channel with no-ISI! This is an easy way to get rid of ISI, particularly when compared to the travails of the receiver-centric schemes we have seen in the last couple of lectures. But there is a catch, and that is we need more energy at the transmitter to cancel ISI than that present in $d[m]$.

For concreteness, let us suppose sequential communication is used to generate the voltages $d[m]$. Denoting the average energy used in the data symbol $d[m]$ by E, the average transmit energy used by the simple precoding scheme (cf. Equation (361)) at time sample m is

$$\mathbb{E}\left[(x[m])^2\right] = E + \frac{\mathbb{E}\left[(I[m])^2\right]}{h_0^2}, \tag{363}$$

where we supposed, as usual, that the mean of the voltage $d[m]$ is zero. Now, the average energy in the ISI at time m is (from Equation (360))

$$\mathbb{E}\left[(I[m])^2\right] = \sum_{\ell=1}^{L-1} h_\ell^2 \mathbb{E}\left[(x[m-\ell])^2\right] \tag{364}$$

where we used the fact, again, that the mean of the voltage $d[m]$ is zero. Substituting from Equation (363), we now get

$$\mathbb{E}\left[(I[m])^2\right] = \sum_{\ell=1}^{L-1} h_\ell^2 \left(E + \frac{\mathbb{E}\left[(I[m-\ell])^2\right]}{h_0^2}\right). \tag{365}$$

To get a quick feel for the solution to this *recursive* definition, suppose that $L = 2$. Then

$$\mathbb{E}\left[(I[m])^2\right] = Eh_1^2 + \frac{h_1^2}{h_0^2}\mathbb{E}\left[(I[m-1])^2\right] \tag{366}$$

$$= \sum_{k=1}^{m-1} \frac{h_1^{2k}}{h_0^{2k-2}} E, \tag{367}$$

since the ISI at the very first time sample, $I[1]$ is zero. Substituting this back into Equation (363) we arrive at

$$\mathbb{E}\left[(x[m])^2\right] = E + \sum_{k=1}^{m-1} \frac{h_1^{2k}}{h_0^{2k}} E. \tag{368}$$

Now we see that if $h_1^2 \geq h_0^2$ then the average transmit energy of the simple precoding scheme grows to infinity over time. Otherwise, the average transmit energy grows to

$$E + \frac{E\frac{h_1^2}{h_0^2}}{1 - \frac{h_1^2}{h_0^2}}, \tag{369}$$

strictly larger than the energy E used by the sequential precoding scheme. This calculation is for the special case of $L = 2$; a homework exercise will explore the solution for general number of taps L.

The main conclusion is that the average transmit energy required to execute the simple precoding scheme grows with time; how much it grows to (a finite amount or even an infinite value) depends on the channel encountered and thus cannot be predicted beforehand. This is quite a problem for the communication engineer since transmit power requirements are decided ahead of time while the filter tap coefficients change based on the wireline channel encountered. It would be good if we could eliminate the ISI for *all* wireline channels encountered using a *constant* increase in the transmit power. The key idea to achieve this goal occurred independently to two graduate students (working towards their PhD in two different continents) in the early 1970s: Tomlinson and Harashima. We will see their elegant improvement to the simple precoding scheme next.

Tomlinson-Harashima (TH) Precoding

To simplify the exposition, consider sequential communication of one bit at a time used to generate the voltages $d[m]$. Specifically, the voltage $d[m]$ is $\pm\sqrt{E}$ depending on the m^{th} information bit. The basic constellation diagram associated with this communication is illustrated in Figure 26. The TH precoding idea is based on *extending* this constellation

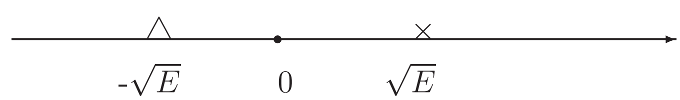

Figure 26: The constellation diagram associated with sequential communication of one bit at a time.

diagram, on both sides of the voltage axis; this is illustrated in Figure 27. The idea is that each of the crosses represents the same information bit (say, '1') and each of the triangles represents the same information bit (say, '0').

To describe the TH precoding scheme, it will help the exposition to suppose that h_0, the first tap coefficient, is 1. There is no loss of generality in this supposition since the receiver can normalize its received voltage $y[m]$ by h_0. Now suppose we want to communicate the information bit '1' at time m. In the simple precoding scheme we would transmit $x[m]$ such that when added to $I[m]$, the resulting sum is the cross situated at $\sqrt{E}$. Since there is only

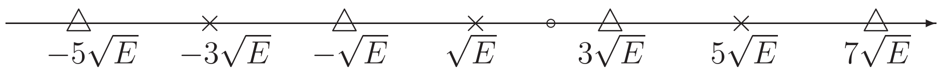

Figure 27: Extended constellation diagram.

a single cross, this would entail a large transmit energy whenever the ISI $I[m]$ is far away from $\sqrt{E}$. This operation is illustrated in Figure 28.

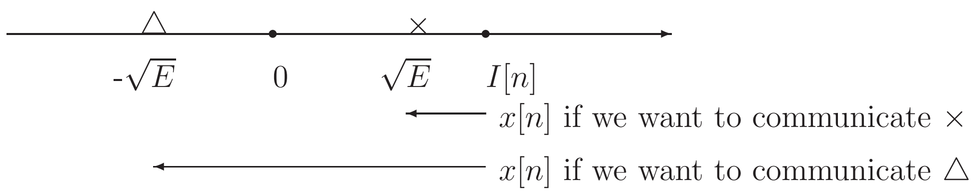

Figure 28: Simple precoding.

On the other hand, with the extended constellation diagram we could transmit $x[m]$ such that when added to the ISI $I[m]$, the resulting sum is *any one* of the many crosses available; specifically we could always pick that cross which is closest to $I[m]$ resulting in bounded transmit energy. This operation is illustrated in Figure 29; we see that the transmit energy is no more than $4E$ for all values of $I[m]$.

The received voltage at time m is

$$y[m] = d[m] + w[m] \tag{370}$$

where $d[m]$ is one of the crosses in Figure 27. Since the additive noise $w[m]$ is Gaussian, it makes sense to follow the nearest neighbor rule. We decode the information bit transmitted depending on whether a cross or a triangle is closer to the received voltage level. The probability error is

$$\mathbb{P}[\mathcal{E}] \leq 2Q\left(\frac{\sqrt{E}}{\sigma}\right). \tag{371}$$

This is not an exact calculation since there is some chance that the additive noise $w[m]$ is larger than the spacing between nearest points in the extended constellation diagram

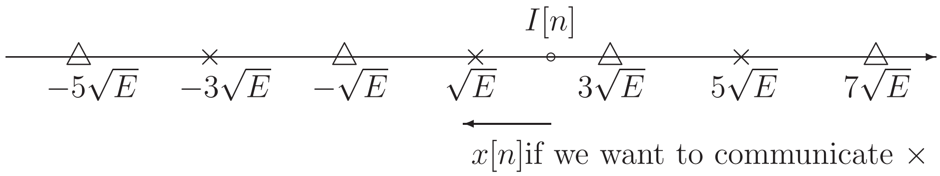

Figure 29: Tomlinson-Harashima precoding

of Figure 27, but the received voltage is still closer to a cross (if one of the crosses was transmitted).

We have focused on eliminating ISI associated with a sequential communication scheme. However, we have seen in earlier lectures that block communication schemes provide much better performances over the AWGN channel. The extension of the TH precoding scheme is natural; it takes place in higher dimensional spaces since the constellation diagrams themselves are described now in higher dimensional spaces.

Looking Ahead

The view espoused by the precoding technique here is the same one we had in our discussion of the zero forcing equalizer: interference is a nuisance and we work to get rid of it. But we developed a more nuanced and balanced view when we discussed the MMSE equalizer. A natural question is to ask for the transmitter-centric version of that balanced approach. We will see (in the next lecture) an important transmitter-centric technique designed from this perspective: it is called OFDM (orthogonal frequency division modulation) and is the basic technology behind many communication standards around us (examples: DSL – digital subscriber line – and Wi-Fi).

Lecture 16: Transmitter-Centric ISI Harnessing: Orthogonal Frequency Division Modulation (OFDM)

Introduction

In the previous lecture, we took a first order transmitter-centric approach to dealing with ISI: the focus was on eliminating the effects of ISI. In this lecture we take a more balanced view: harnessing the benefits of ISI instead of just treating it as interference, continuing our transmitter-centric approach. Our goal is to get the full benefit of the fact that multiple delayed copies of the transmit symbols appear at the receiver while still employing a receiver no more complicated than the one used over an AWGN channel. While this seems to be a tall order, we will see a remarkable scheme that achieves exactly this. In a sense, it is a natural culmination of the various ISI mitigation techniques we have seen in the course of the past several lectures. The scheme that converts the frequency selective ISI channel into a plain AWGN channel is known as *orthogonal frequency division modulation* (OFDM) and is the main focus of this lecture.

An Ideal Situation

Consider the frequency selective model that we have been working with as a good approximation of the wireline channel:

$$y[m] = \sum_{\ell=0}^{L-1} h_\ell x[m-\ell] + w[m], \quad m \geq 1. \tag{372}$$

Since we know how to communicate reliably over an AWGN channel (cf. Lecture 7), it would be ideal (not to mention, easy) if the channel with ISI can *somehow* (by appropriate transmitter and reciever operations) be converted into an AWGN one: say,

$$y[m] = \hat{h}x[m] + w[m], \quad m \geq 1. \tag{373}$$

In such a case, we could simply and readily use the transmitter and receiver techniques developed already for the AWGN channel (available "off-the-shelf", so to say).

While this is asking for a bit too much, we will see that we can get somewhat close: indeed, we will convert the ISI channel in Equation (372) into a *collection* of AWGN channels, each of different noise energy level:

$$\hat{y}[N_c k + n] = \hat{h}_n \hat{x}[N_c k + n] + \hat{w}[N_c k + n], \quad k \geq 0, \quad n = 0 \ldots N_c - 1. \tag{374}$$

The idea is that the time index m is replaced by $N_c k + n$. The inputs are voltages $\hat{x}$. The additive noise $\hat{w}[\cdot]$ is white Gaussian (zero mean and variance σ^2). We observe that there are N_c different AWGN channels, one for each $n = 0, \ldots, N_c - 1$. We can make two further observations:

- each of the N_c AWGN channels has a *different* operating SNR: the n^{th} channel has an SNR equal to $\hat{h}_n^2\mathsf{SNR}$ where SNR is, as usual, the ratio of the transmit energy to the noise energy;

- each of the N_c AWGN channels is available for use only a *fraction* $\frac{1}{N_c}$ of the time.

Such a collection of non-interfering AWGN channels is called a *parallel* AWGN channel. The individual AWGN channels within the collection are known as *sub-channels*. Our understanding of efficient reliable communication over the AWGN channel suggests a natural strategy to communicate over the parallel AWGN channel as well: we can split the information bits so that we communicate over each sub-channel *separately*. The only choice remaining is how to split the total power budget amongst the sub-carriers, say power P_n to the n^{th} sub-carrier, so that

$$\sum_{n=0}^{N_c-1} P_n = P, \tag{375}$$

where P is the total power budget for reliable communication. With a transmit power constraint of P_n, the overall SNR of the n^{th} sub-channel is

$$\frac{P_n \hat{h}_n^2}{\sigma^2}. \tag{376}$$

Thus, with an appropriate coding and decoding mechanism reliable communication is possible (cf. Lecture 7) on the n^{th} AWGN sub-channel at rate

$$R_n = \frac{1}{2N_c} \log_2\left(1 + \frac{P_n \hat{h}_n^2}{\sigma^2}\right), \tag{377}$$

measured in bits/symbol. The factor of $1/N_c$ in the rate appears because each of the sub-channels is available for use only a fraction $1/N_c$ of the time. The total rate of reliable communication is

$$\sum_{n=0}^{N_c-1} R_n = \frac{1}{2N_c} \sum_{n=0}^{N_c-1} \log_2\left(1 + \frac{P_n \hat{h}_n^2}{\sigma^2}\right). \tag{378}$$

We can now split the power to *maximize* the rate of reliable communication over the parallel AWGN channel:

$$\max_{P_n \geq 0, \sum_{n=0}^{N_c-1} P_n = P} \frac{1}{2N_c} \sum_{n=0}^{N_c-1} \log_2\left(1 + \frac{P_n \hat{h}_n^2}{\sigma^2}\right). \tag{379}$$

The optimal power split can be derived explicitly and is explored in a homework exercise. The main property of this optimal power split is that the larger the "quality" of a sub-channel, the more the power that is allocated to it, and hence the larger the corresponding data rate of reliable communication.

In the rest of this lecture we will see a "high level view" of how to go from the ISI channel (cf. Equation (372) to the parallel AWGN channel (cf. Equation (374)).

Convolution in Matrix Notation

Let's consider communicating over a block of N_c time instants over a L-tap wireline channel. Recall that the received voltage at the output of the wireline channel at time m is

$$y[m] = \sum_{l=0}^{L-1} h_l x[m-l] + w[m] \qquad m \geq 1. \tag{380}$$

Here, $x[m]$ and $w[m]$ are the transmitted and the noise voltages at time instant m, respectively. The wireline channel without the additive noise is a simple linear (finite impulse response) filter. It *convolves* the input voltage vector with the channel vector. Adding the noise vector gives the received voltage vector. Since convolution is, at its essence, a *linear* transformation, we may employ the matrix notation to depict the input-output relation over a wireline channel:

$$\begin{bmatrix} y[1] \\ y[2] \\ y[3] \\ \vdots \\ \vdots \\ y[N_c] \end{bmatrix} = \begin{bmatrix} h_0 & 0 & 0 & 0 & \cdots & 0 \\ h_1 & h_0 & 0 & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & \cdots & h_{L-1} & h_{L-2} & \cdot & h_0 \end{bmatrix} \begin{bmatrix} x[1] \\ x[2] \\ \vdots \\ x[N_c] \end{bmatrix} + \begin{bmatrix} w[1] \\ w[2] \\ w[3] \\ \vdots \\ \vdots \\ w[N_c] \end{bmatrix}. \tag{381}$$

We can write Equation (381) in short hand notation as

$$\mathbf{y} = \mathbf{H}\mathbf{x} + \mathbf{w} \tag{382}$$

where $\mathbf{x}$, $\mathbf{y}$ and $\mathbf{w}$ are the input, output and noise vectors (all of dimension $N_c \times 1$), respectively. The index k here refers to different N_c dimensional blocks of input voltage vectors $\mathbf{x}$ that could be sent over the wireline channel. The channel matrix $\mathbf{H}$ is a $N_c \times N_c$ dimensional matrix. The matrix $\mathbf{H}$ has an important feature: the diagonal elements are all identical. Such a matrix is said to be *Toeplitz*.

Communicating over a Diagonal Matrix

The channel expressed in Equation (382) is a *vector* version of the AWGN channel we are very familiar with. If the linear transformation $\mathbf{H}$ was diagonal, then we would have a *parallel* AWGN channel. Communicating over such a channel is relatively straightforward: we can code over each "sub-channel" reliably using coding and decoding methods developed for AWGN channels; this has been explained in detail in the previous lecture. However, the $\mathbf{H}$ that results from the convolution operation is diagonal exactly when there is just one tap (i.e., $L = 1$). So, we cannot really expect $\mathbf{H}$ to be diagonal at all. Since we know how to deal with a diagonal matrix, a question that naturally arises is:

> Can we perform some linear transformation on $\mathbf{H}$ so that the resulting matrix is diagonal?

If so, then the linear transformations on $\mathbf{H}$ can be absorbed into linear processing at the transmitter and the receiver. Concretely, let $\mathbf{Q}_1$ denote the transformation done at the transmitter: the "data" vector $\tilde{\mathbf{x}}$ and the transmit voltage vector $\mathbf{x}$ are related by

$$\mathbf{x} = \mathbf{Q}_1\tilde{\mathbf{x}}. \tag{383}$$

Analogously, let matrix $\mathbf{Q}_2$ denote the transformation on the received voltage vector $\mathbf{y}$:

$$\tilde{\mathbf{y}} = \mathbf{Q}_2\mathbf{y} \tag{384}$$

We can then write the *effective* channel between the data vector $\tilde{\mathbf{x}}$ and vector $\tilde{\mathbf{y}}$ as:

$$\tilde{\mathbf{y}} = \tilde{\mathbf{H}}\tilde{\mathbf{x}} + \tilde{\mathbf{w}} \tag{385}$$

where

$$\tilde{\mathbf{H}} = \mathbf{Q}_2\mathbf{H}\mathbf{Q}_1 \tag{386}$$

$$\tilde{\mathbf{w}} = \mathbf{Q}_2\mathbf{w}. \tag{387}$$

We want to choose $\mathbf{Q}_1$ and $\mathbf{Q}_2$ such that the matrix $\tilde{\mathbf{H}}$, the effective channel between $\tilde{\mathbf{x}}$ and $\tilde{\mathbf{y}}$, is diagonal. But we would like to do this while still not changing the statistics of the vector noise $\tilde{\mathbf{w}}$ (i.e., the entries of $\tilde{\mathbf{w}}$ are also i.i.d. Gaussian random variables). This way, we would have arrived at a parallel AWGN channel. To summarize: our goal is to find the matrices $\mathbf{Q}_1$ and $\mathbf{Q}_2$ such that

1. $\mathbf{Q}_2\mathbf{H}\mathbf{Q}_1$ is diagonal;
2. $\tilde{\mathbf{w}}$ is an i.i.d. Gaussian random vector.

Linear Transformation of a Matrix Into a Diagonal Form

Linear algebra is a mathematical field of study that answers questions of the type above. It would be a big detour for us to wander along this path, but here is the classical text book that would satiate all but the most curious graduate student:

R. A. Horn and C. R. Johnson, *Matrix Analysis*, Cambridge University Press, 1990.

The second condition, that $\mathbf{w}$ and $\tilde{\mathbf{w}}$ both be i.i.d. Gaussian is a relatively mild one. In fact, this is readily satisfied by all matrices $\mathbf{Q}_2$ with the property that

$$\mathbf{Q}_2^T \mathbf{Q}_2 = \mathbf{I}. \tag{388}$$

Here we have denoted the identity matrix by $\mathbf{I}$. A simple calculation verifying this claim for $N_c = 2$ is available in the appendix. The first condition, requiring $\tilde{\mathbf{H}}$ to be diagonal is more involved. In fact, the most fundamental result in all of linear algebra, the so-called *singular value decomposition* (SVD) theorem, answers this question in the affirmative: all matrices $\mathbf{H}$ can be linearly transformed into diagonal matrices $\tilde{\mathbf{H}}$. The choice of $\mathbf{Q}_1, \mathbf{Q}_2$ that achieve this transformation, in general, depend closely on the original matrix $\mathbf{H}$. However, we are dealing with a Toeplitz matrix $\mathbf{H}$ and there is some chance to have specific structure in the appropriate choice of $\mathbf{Q}_1$ and $\mathbf{Q}_2$. Indeed, there is strong structure when N_c is large. This aspect is explored in the theory of Toeplitz forms, a topic well beyond the boundaries of mathematics we are using. Then again, it is hard to say when one of you might need this knowledge. So, here is the classical reference on the topic:

U. Grenander, *Toeplitz Forms and Their Applications*, Chelsea Publication Company, 2nd edition, 1984.

The structure in $\mathbf{Q}_1, \mathbf{Q}_2$ for large enough N_c is a bit hard to describe. With a bit of modification, however, there is a very clean solution. This is the OFDM solution, the focus of the next lecture.

Looking ahead

We took a fundamental block-communication view of the ISI channel. We saw that it is possible to convert the channel into a *parallel* AWGN channel. Reliable communication over the parallel AWGN channel is simple: we send separate packets over the separate AWGN sub-channels. We have the flexibility of allocating powers (and appropriate data rates) in our communication over the sub-channels. A good strategy would be to allocate powers in proportion to the sub-channel quality. In the next lecture we will see a concrete and simple algorithmic method to convert the ISI channel into a parallel AWGN channel.

Appendix

Set $N_c = 2$. We are interested in properties of the 2×2 matrix $\mathbf{Q}_2$ such that both $\mathbf{w}$ and $\tilde{\mathbf{w}}$ are i.i.d. Gaussian:

$$\begin{bmatrix} \tilde{w}_1 \\ \tilde{w}_2 \end{bmatrix} = \mathbf{Q}_2 \begin{bmatrix} w_1 \\ w_2 \end{bmatrix} \tag{389}$$

$$\begin{bmatrix} \tilde{w}_1 \\ \tilde{w}_2 \end{bmatrix} = \begin{bmatrix} a_1 & b_1 \\ a_2 & b_2 \end{bmatrix} \begin{bmatrix} w_1 \\ w_2 \end{bmatrix}. \tag{390}$$

We know that w_1 and w_2 are i.i.d. Gaussian random variables with mean 0 and variance σ^2. We want $\tilde{w}_1$ and $\tilde{w}_2$ to also be i.i.d., i.e., they should necessarily have same variance and should be uncorrelated. (Note that this is also the sufficient condition as $\tilde{w}_i$ are Gaussian.)

$$\mathrm{Var}(\tilde{w}_1) = (a_1^2 + b_1^2)\sigma^2 \tag{391}$$

$$\mathrm{Var}(\tilde{w}_2) = (a_2^2 + b_2^2)\sigma^2 \tag{392}$$

$$\Rightarrow (a_1^2 + b_1^2) = (a_2^2 + b_2^2) = 1 \quad \text{(say)}. \tag{393}$$

The correlation $\mathbb{E}(\tilde{w}_1\tilde{w}_2)$ is

$$\mathbb{E}(\tilde{w}_1\tilde{w}_2) = \mathbb{E}[(a_1 w_1 + b_1 w_2)(a_2 w_1 + b_2 w_2)] \tag{394}$$

$$= (a_1 a_2 + b_1 b_2)\sigma^2. \tag{395}$$

For $\tilde{w}_1$ and $\tilde{w}_2$ to be uncorrelated, we want

$$a_1 a_2 + b_1 b_2 = 0 \tag{396}$$

From Equation (393), we can choose the vectors $[a_1, b_1]$ and $[a_2, b_2]$ as

$$\mathbf{Q}_2 = \begin{bmatrix} \cos\theta & \sin\theta \\ \cos\phi & \sin\phi \end{bmatrix}. \tag{397}$$

From Equation (396), we want

$$\cos\theta\cos\phi + \sin\theta\sin\phi = 0 \tag{398}$$

$$\cos(\theta - \phi) = 0 \tag{399}$$

$$\theta = \phi + \frac{\pi}{2} \tag{400}$$

Hence

$$\mathbf{Q}_2 = \begin{bmatrix} \cos\theta & \sin\theta \\ \sin\theta & -\cos\theta \end{bmatrix} \tag{401}$$

which has the property that $\mathbf{Q}_2^T \mathbf{Q}_2 = \mathbf{I}$ for all values of θ, as can be directly verified.

Lecture 17: Orthogonal Frequency Division Modulation (OFDM) and Capacity of the Wireline Channel

Introduction

In this lecture we will see in detail the OFDM method to convert the wireline channel into a parallel AWGN channel. We also see that this achieves the *capacity* of the wireline channel – in other words, the largest possible data rate of communication is achieved by the OFDM method.

OFDM

Consider the frequency selective model that we have been working with as a good approximation of the wireline channel:

$$y[m] = \sum_{\ell=0}^{L-1} h_\ell x[m-\ell] + w[m], \quad m \geq 1. \tag{402}$$

We will convert the ISI channel in Equation (402) into a *collection* of AWGN channels, each of different noise energy level:

$$\hat{y}[N_\mathrm{c}k+n] = \hat{h}_n \hat{x}[N_\mathrm{c}k+n] + \hat{w}[N_\mathrm{c}k+n], \quad k \geq 0, \quad n = 0 \ldots N_\mathrm{c}-1. \tag{403}$$

We will be able to make this transition by some very simple signal processing techniques. Interestingly, these signal processing techniques are *universally* applicable to every wireline channel, i.e., they do not depend on the exact values of channel coefficients $h_0, \ldots, h_{L-1}$. This makes OFDM a very *robust* communication scheme over the frequency-selective channel.

Cyclic Prefix

Suppose we have mapped our information bits into N_c voltages. We will revisit the issue of how these voltages were created from the information bits at a slightly later point in this lecture. For now, we write them as a vector:

$$\mathbf{d} = [d[0], d[1], \ldots, d[N_\mathrm{c}-1]]^t.$$

We use these N_c voltages to create an $N_\mathrm{c}+L-1$ block of *transmit* voltages as:

$$\mathbf{x} = [d[N_\mathrm{c}-L+1], d[N_\mathrm{c}-L+2], \ldots, d[N_\mathrm{c}-1], d[0], d[1], \ldots, d[N_\mathrm{c}-1]]^t, \tag{404}$$

i.e., we add a *prefix* of length $L-1$ consisting of data symbols rotated cyclically (Figure 30). The first $L-1$ transmitted symbols contain the "data" symbols $d[N_\mathrm{c}-(L-1)], \ldots, d[N_\mathrm{c}-1]$.

The next $N_{\rm c}$ transmitted voltages or symbols contain the "data" symbols $d[0], d[1], \ldots, d[N_{\rm c}-1]$. In particular, for a 2-tap frequency-selective channel we have the following result of cyclic precoding:

$$\begin{aligned} x[1] &= d[N_{\rm c}-1] \\ x[2] &= d[0] \\ x[3] &= d[1] \\ &\vdots \\ x[N_{\rm c}+1] &= d[N_{\rm c}-1] \end{aligned}$$

With this input to the channel (Equation (402)), consider the output

$$y[m] = \sum_{\ell=0}^{L-1} h_\ell x[m-\ell] + w[m], \qquad m = 1, \ldots, N_{\rm c} + 2(L-1).$$

The first $L-1$ elements of the transmitted vector $\mathbf{x}$ were constructed from circularly wrapped elements of the vector $\mathbf{d}$, which are included in the last $N_{\rm c}-1$ elements of $\mathbf{x}$. The receiver hence ignores the first $L-1$ received symbols $y[1], \ldots, y[L-1]$. The ISI extends over the first $L-1$ symbols and the receiver ignores it by considering only the output over the time interval $m \in [L, N_{\rm c}+L-1]$. Let us take a careful look at how the N receive voltages (received at times L through N_c+L-1) depend on the transmit voltages $d[0], \ldots, d[N_{\rm c}-1]$:

$$y[m] = \sum_{\ell=0}^{L-1} h_\ell d[(m-L-\ell) \text{ modulo } N_{\rm c}] + w[m]. \tag{405}$$

See Figure (30).

Denoting the received voltage vector of length $N_{\rm c}$ by

$$\mathbf{y} = [y[L], \ldots, y[N_{\rm c}+L-1]]^t,$$

and the channel by a vector of length $N_{\rm c}$

$$\mathbf{h} = [h_0, h_1, \ldots, h_{L-1}, 0, \ldots, 0]^t, \tag{406}$$

Equation (405) can be written as

$$\mathbf{y} = \mathbf{h} \otimes \mathbf{d} + \mathbf{w}. \tag{407}$$

Here we denoted

$$\mathbf{w} = [w[L], \ldots, w[N_{\rm c}+L-1]]^t, \tag{408}$$

as a vector of i.i.d. $\mathcal{N} \sim (0, \sigma^2)$ random variables. The notation of $\otimes$ to denote the *cyclic convolution* in (407) is standard in signal processing literature. The point of all this manipulation will become clear when we review a key property of cyclic convolution next.

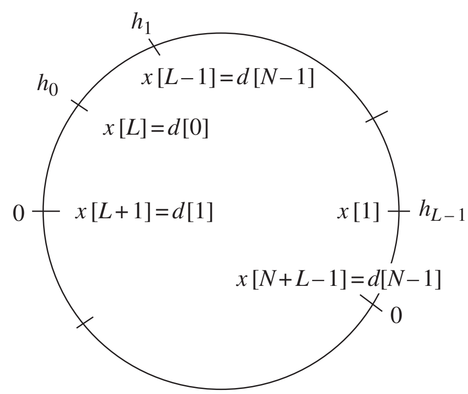

Figure 30: Convolution between the channel (**h**) and the input (**x**) formed from the data symbols (**d**) by adding a cyclic prefix. The output is obtained by multiplying the corresponding values of **x** and **h** on the circle, and outputs at different times are obtained by rotating the x-values with respect to the h-values. The current configuration yields the output $y[L]$.

Discrete Fourier Transform

The discrete Fourier transform (DFT) of a vector (such as **d**) is also another vector of the same length (though the entries are in general, complex numbers). The different components of the discrete Fourier transform of the vector **d**, denoted by DFT(**d**), are defined as follows:

$$\tilde{d}_n := \frac{1}{\sqrt{N_\mathrm{c}}} \sum_{m=0}^{N_\mathrm{c}-1} d[m] \exp\left(\frac{-\mathrm{j}2\pi nm}{N_\mathrm{c}}\right), \qquad n = 0, \ldots, N_\mathrm{c} - 1. \tag{409}$$

Even though the voltages $d[\cdot]$ are real, the DFT output $\tilde{d}_n$ are complex. Nevertheless, there is *conjugate symmetry*:

$$\tilde{d}_n = \tilde{d}^*_{N_\mathrm{c}-1-n}, \quad n = 0, \ldots, N_\mathrm{c} - 1. \tag{410}$$

DFTs and circular convolution are crucially related through the following equation (perhaps the most important of all in discrete time digital signal processing):

$$\mathrm{DFT}(\mathbf{h} \otimes \mathbf{d})_n = \sqrt{N_\mathrm{c}} \mathrm{DFT}(\mathbf{h})_n \cdot \mathrm{DFT}(\mathbf{d})_n, \qquad n = 0, \ldots, N_\mathrm{c} - 1. \tag{411}$$

The vector $[\tilde{h}_0, \ldots, \tilde{h}_{N_\mathrm{c}-1}]^t$ is defined as the DFT of the L-tap channel **h**, multiplied by $\sqrt{N_\mathrm{c}}$,

$$\tilde{h}_n = \sum_{\ell=0}^{L-1} h_\ell \exp\left(\frac{-\mathrm{j}2\pi n\ell}{N_\mathrm{c}}\right). \tag{412}$$

Thus we can rewrite (407) as

$$\tilde{y}_n = \tilde{h}_n \tilde{d}_n + \tilde{w}_n, \qquad n = 0, \ldots, N_\mathrm{c} - 1. \tag{413}$$

Here we have denoted $\tilde{w}_0, \ldots, \tilde{w}_{N_c-1}$ as the N_c-point DFT of the noise vector $w[1], \ldots, w[N_c]$. Observe the following.

- Even though the received voltages $y[\cdot]$ are real, the voltages at the output of the DFT $\tilde{y}_n$ are complex. Thus it might seem odd that we started out with N real numbers and ended up with $2N$ real numbers. But there is a redundancy in the DFT output $\tilde{y}[\cdot]$. Specifically,
$$\tilde{y}_n = \tilde{y}^*_{N_c-1-n}. \tag{414}$$
In other words, the real parts of $\tilde{y}_n$ and $\tilde{y}_{N_c-1-n}$ are the same. Further, the imaginary parts of $\tilde{y}_n$ and $\tilde{y}_{N_c-1-n}$ are negative of each other.

- Even though the noise voltages $w[\cdot]$ are real, the noise voltages at the output of the DFT $\tilde{w}_n$ are complex. Just as before, there is a redundancy in the noise voltages:
$$\tilde{w}_n = \tilde{w}^*_{N_c-1-n}, \quad n = 0, \ldots, N_c - 1. \tag{415}$$
We know that the noise voltages $w[m]$ were white Gaussian. What are the statistics of the DFT outputs? It turns out that they are *also* white Gaussian: the real and imaginary parts of $\tilde{w}_0, \ldots, \tilde{w}_{\frac{N_c}{2}-1}$ are all independent and identically distributed as Gaussian with zero mean and variance σ^2 (here we supposed for notational simplicity that N_c is even, so $\frac{N_c}{2}$ is an integer).

- Even though the channel coefficients h_ℓ are real (and zero for $\ell = L, \ldots, N_c - 1$), the values at the output $\tilde{h}_n$ of the DFT are complex (and, in general, non-zero for all values $n = 0, \ldots, N_c - 1$). Again, observe that
$$\tilde{h}_n = \tilde{h}[N_c - 1 - n]^*, \quad , n = 0, \ldots, N_c - 1. \tag{416}$$

The result that we get from this precoding is the following: the DFT of the received vector $\mathbf{y}$ and the DFT of our initial "data" vector $\mathbf{d}$ have the relationship that the received and transmitted vectors have in an AWGN channel with no ISI (given a suitable definition for the noise vector and it's DFT as given earlier). This seems to suggest that if we put the actual data that we want to transmit on the DFT, and take the DFT of what we receive, then we can perform something similar to traditional AWGN style decoding. Note that this scheme uses $L-1$ extra time instants. This yields the block diagram in Figure (32). A careful and detailed derivation of this step is carried out next. At the end of that calculation, we will have also shown how we arrive at the parallel AWGN channel (cf. Equation (403)).

Packaging the Data

In this section we will see how to connect the AWGN modulation techniques with the OFDM transmission scheme. Suppose we start out with N_c real voltages $\hat{x}[0], \ldots, \hat{x}[N_c-1]$. These are

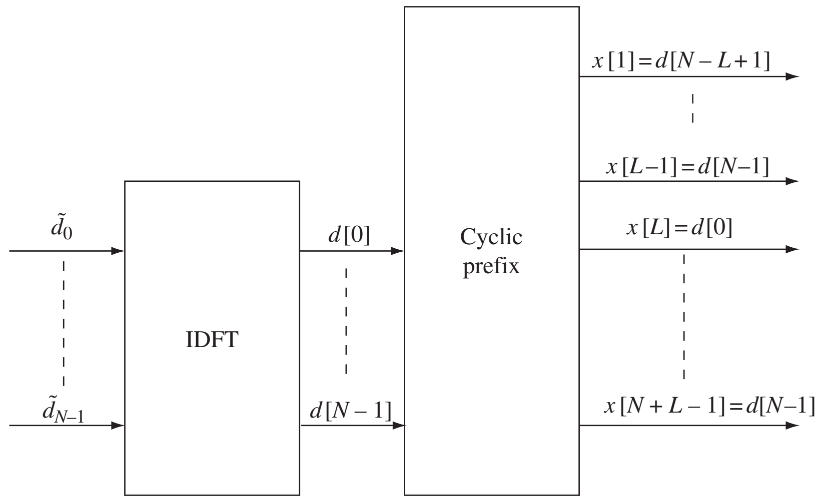

Figure 31: The cyclic prefix operation.

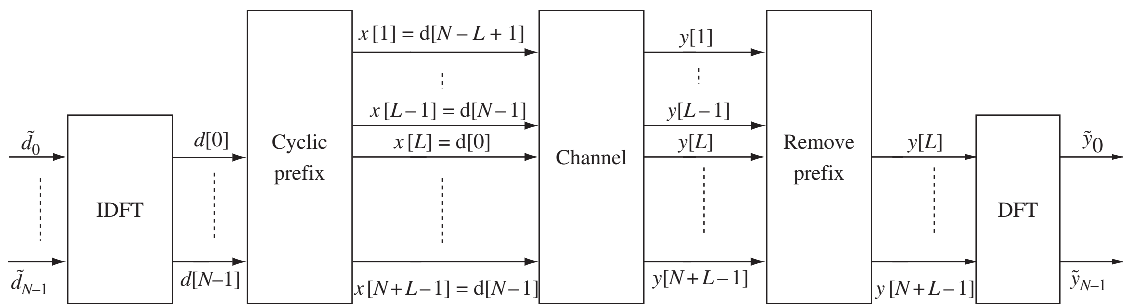

Figure 32: The OFDM transmission and reception schemes.

the transmit voltages on the N_c sub-channels using the separate communication architecture (they are generated by efficient coding techniques – such as LDPC codes – for the AWGN channel). Let us suppose that N_c is an even number. This will simplify our notations. We generate *half* of the data vector $\tilde{\mathbf{d}}$ as follows:

$$\Re\left[\tilde{d}_n\right] \stackrel{\text{def}}{=} \hat{x}[2n] \tag{417}$$

$$\Im\left[\tilde{d}_n\right] \stackrel{\text{def}}{=} \hat{x}[2n+1], \quad n = 0, \ldots, \frac{N_c}{2} - 1. \tag{418}$$

The second half is simply conjugate symmetric of the first part (so as to respect Equation (410)):

$$\tilde{d}_n = \tilde{d}^*_{N_c-1-n}, \quad n = \frac{N_c}{2}, \ldots, N_c - 1. \tag{419}$$

Since $\tilde{\mathbf{d}}$ is conjugate symmetric by construction, the inverse discrete Fourier transform

(IDFT) vector $\mathbf{d}$ is composed only of real numbers. The cyclic prefix is then added on and the transmit voltages $x[m]$ are generated. Observe that we need an extra $L-1$ time instants to send over the N_c voltages $\hat{x}[0], \ldots, \hat{x}[N_c - 1]$.

Unpacking the Data

At the output of the DFT of the received voltage vector $\mathbf{y}$ we have the complex vector $\tilde{\mathbf{y}}$. Taking complex conjugate operation on both sides of Equation (411):

$$\begin{aligned} \tilde{y}_n^* &= \tilde{h}_n^* \tilde{d}_n^* + \tilde{w}_n^* & (420)\\ &= \tilde{h}_{N_c-1-n} \tilde{d}_{N_c-1-n} + \tilde{w}_{N_c-1-n} & (421)\\ &= \tilde{y}_{N_c-1-n}. & (422) \end{aligned}$$

Here we used Equations (410),(416), (415) to verify Equation (414). This means that *half* the DFT outputs are *redundant* and can be discarded. Using the first half, we arrive at the following N_c received voltages $\hat{y}[0], \ldots \hat{y}[N_c - 1]$:

$$\begin{aligned} \hat{y}[2n] &\stackrel{\text{def}}{=} \Re\left[\frac{\tilde{h}_n^*}{|\tilde{h}_n|}\tilde{y}_n\right] & (423)\\ &= |\tilde{h}_n|\hat{x}[2n] + \hat{w}[2n] & (424)\\ \hat{y}[2n+1] &\stackrel{\text{def}}{=} \Im\left[\frac{\tilde{h}_n^*}{|\tilde{h}_n|}\tilde{y}_n\right] & (425)\\ &= |\tilde{h}_n|\hat{x}[2n+1] + \hat{w}[2n+1], \quad n = 0, \ldots, \frac{N_c}{2} - 1. & (426) \end{aligned}$$

Here $\hat{w}[\cdot]$ is also white Gaussian with zero mean and variance σ^2 (explored in a homework exercise). Putting together Equations (424) and (426), we can write

$$\hat{y}[n] = \hat{h}_n \hat{x}[n] + \hat{w}[n], \quad n = 0, \ldots, N_c - 1 \tag{427}$$

where we have written

$$\hat{h}_n \stackrel{\text{def}}{=} \begin{cases} |\tilde{h}_{\frac{n}{2}}| & n \text{ even} \\ |\tilde{h}_{\frac{n-1}{2}}| & n \text{ odd.} \end{cases} \tag{428}$$

We can repeat the OFDM operation over the next block of N_c symbols (taking up an extra $L-1$ time instants, as before) and since the wireline channel stays the same, we have the end-to-end relation (as in Equation (427)):

$$y[N_c + n] = \hat{h}_n \hat{x}[N_c + n] + \hat{w}[N_c + n], \quad n = 0, \ldots, N_c - 1. \tag{429}$$

By repeating the OFDM operation over multiple N_c blocks, we have thus created the parallel AWGN channel promised in Equation (403). This packaging and unpacking of data as appended to the basic OFDM scheme (cf. Figure 32) is depicted in Figures 33 and 34.

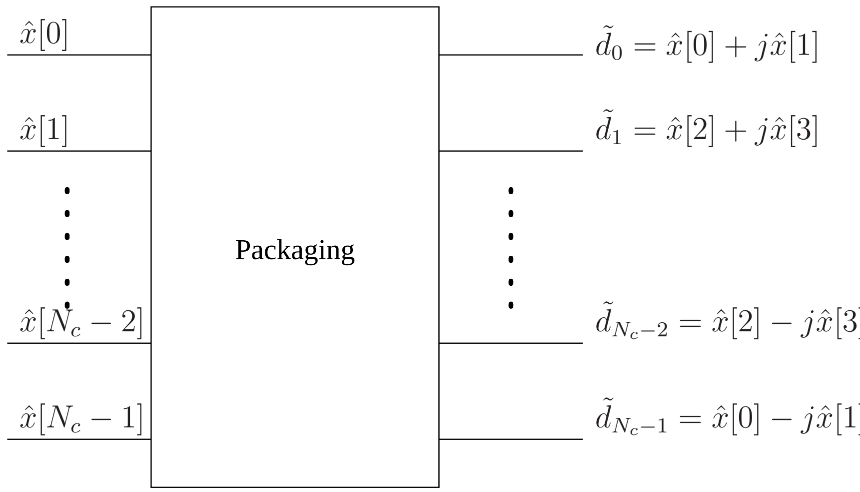

Figure 33: The packaging at the transmitter maps AWGN coded voltages of the sub-channels to the transmit voltages on the ISI channel.

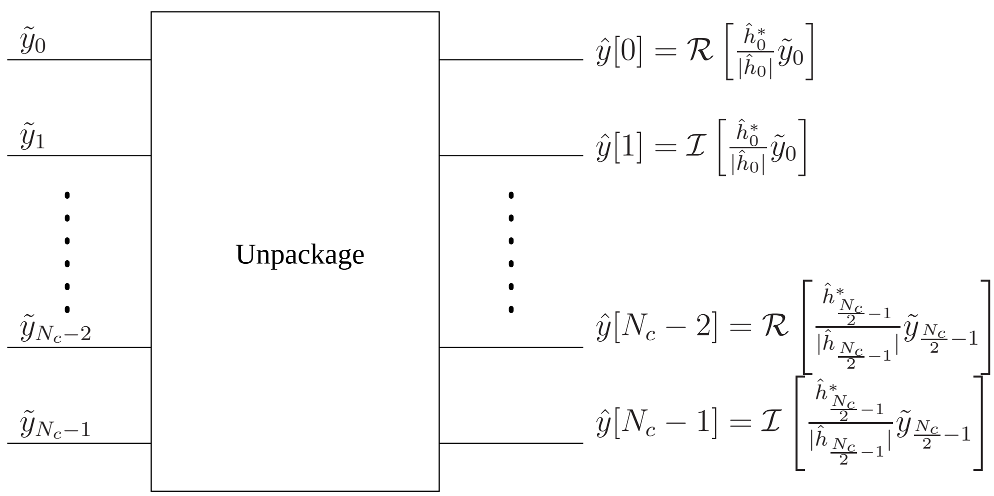

Figure 34: The unpacking at the receiver maps the DFT outputs into the outputs suitable for decoding the coded voltages of the sub-channels.

Prefix: Converting a Topelitz Matrix into a Circulant Matrix

The OFDM scheme uses $L-1$ extra time samples to transmit the block of N_c symbols. The resulting linear transformation between the input and output voltage vectors can now be written as:

$$\mathbf{y} = \mathbf{H}_\mathrm{c}\mathbf{x} + \mathbf{w}, \tag{430}$$

where $\mathbf{H}_\mathrm{c}$ is the *circulant* version of the original Toeplitz matrix $\mathbf{H}$. For example, with $L = 3, N_c = 5$, the Toeplitz and its corresponding circulant version are:

$$\mathbf{H} = \begin{bmatrix} h_0 & 0 & 0 & 0 & 0 \\ h_1 & h_0 & 0 & 0 & 0 \\ h_2 & h_1 & h_0 & 0 & 0 \\ 0 & h_2 & h_1 & h_0 & 0 \\ 0 & 0 & h_2 & h_1 & h_0 \end{bmatrix}, \quad \mathbf{H}_c = \begin{bmatrix} h_0 & 0 & 0 & h_2 & h_1 \\ h_1 & h_0 & 0 & 0 & h_2 \\ h_2 & h_1 & h_0 & 0 & 0 \\ 0 & h_2 & h_1 & h_0 & 0 \\ 0 & 0 & h_2 & h_1 & h_0 \end{bmatrix}. \tag{431}$$

Now the important fact:

All circulant matrices are *universally* diagonalized by the same $\mathbf{Q}_1, \mathbf{Q}_2$.

Furthermore, if we allow for linear transformations with complex entries, $\mathbf{Q}_1, \mathbf{Q}_2$ are the IDFT and DFT matrices! Here the DFT matrix is defined as usual: the (k, l) entry is

$$d_{kl} = \frac{1}{\sqrt{N_c}} e^{-j2\pi \frac{kl}{N_c}} \quad k, l = 0, 1, \cdots, N_c - 1. \tag{432}$$

Similarly, the (k, l) entry of IDFT matrix is given by

$$\frac{1}{\sqrt{N_c}} e^{j2\pi \frac{kl}{N_c}} \quad k, l = 0, 1, \cdots, N_c - 1. \tag{433}$$

A practical implication is that neither transmitter nor the receiver needs to know the channel matrix in order to convert the ISI channel into a parallel AWGN channel.

Capacity of the Wireline Channel

The cyclic prefix added a penalty of $L-1$ time samples over a block of length N_c. By choosing the block length very large, we can amortize the cost (in data rate) of the cyclic prefix overhead. So, the cyclic prefix operation can be thought of one without loss of much generality. Further, the IDFT and DFT operations (at the transmitter and receiver, respectively) are invertible matrix operations. So, they can be done without loss of generality as well (in the sense they can always be undone, if need be). So, the OFDM way of converting the wireline channel into a parallel AWGN channel entails no loss of reliable data rate of communication. We conclude:

The capacity of the wireline channel is the same as that of the corresponding parallel AWGN channel for large values of N_c.

Now, we know the capacity and the corresponding reliable communication techniques that achieve it in the context of the basic AWGN channel. A natural strategy would be to exploit this deep understanding in this slightly different take on the basic version of the AWGN channel. One such strategy to transmit a packet over the parallel AWGN channel would be to divide the data into the subpackets, one for each subchannel and then code and transmit each subpacket over the corresponding subchannel separately. However, since the channel coefficients of the subchannels are different, if we divide the total power equally, they will have different SNRs and hence different capacities. Intuitively, we should allot more power to the channel with the larger coefficient. It turns out that coding separately over each subchannel is indeed the best strategy, i.e., if the powers are allocated properly, this strategy will achieve the capacity of the parallel channel with N_c subchannels.

Let P_k be the power used on the subchannel k. The rate we can achieve over the k^{th} subchannel is

$$R_k = \frac{1}{N_c + L - 1} \frac{1}{2} \log_2 \left(1 + \frac{\hat{h}_k^2 P_k}{\sigma^2} \right) \quad \text{bits/channel use} \quad k = 0, 1, \cdots, N_c - 1 \qquad (434)$$

where $\tilde{h}_k$ is the coefficient of the k^{th} subchannel. We normalize the rate by $N_c + L - 1$ as we use each subchannel once in each block of $N_C + L - 1$ channel uses. The overhead of $L - 1$ is due to the cyclic prefix used in OFDM. We have the total power constraint that the total power used over all subchannels should not exceed P, i.e.,

$$\sum_{k=0}^{N_c-1} P_k \leq P \qquad (435)$$

The total rate is the sum of the individual capacities of the subchannels. Thus, the capacity is achieved by the solution of the following optimization problem.

$$\begin{aligned} &\max_{P_0, P_1, \cdot, P_{N_c-1}} \tfrac{1}{2(N_c+L-1)} \textstyle\sum_{k=0}^{N_c-1} \log_2 \left(1 + \frac{P_k \tilde{h}_k^2}{\sigma^2} \right) \\ &\text{subject to} \quad \textstyle\sum_{k=0}^{N_c-1} P_k \leq P \end{aligned} \qquad (436)$$

The solution of this optimization problem is

$$P_k^* = \left(\frac{1}{\lambda} - \frac{\sigma^2}{|\tilde{h}_k|^2} \right)^+ \qquad (437)$$

where the function $(x)^+ \stackrel{\text{def}}{=} \max(x, 0)$ and λ is chosen so that it satisfies

$$\sum_{k=0}^{N_c-1} P_k^* = P. \qquad (438)$$

Thus, the capacity of the parallel channel is

$$C_{N_c} = \frac{1}{2(N_c + L - 1)} \sum_{k=0}^{N_c - 1} \log_2 \left(1 + \frac{\tilde{h}_k^2}{\sigma^2} \left(\frac{1}{\lambda} - \frac{\sigma^2}{\tilde{h}_k^2}\right)^+\right). \tag{439}$$

How do the parallel channel coefficients $\tilde{h}_k$, the DFT (Discrete Fourier Transform) of the tap coefficients h_l of the original wireline channel,

$$\tilde{h}_k = \sum_{l=0}^{L-1} h_l e^{-j2\pi \frac{kl}{N_c}} \quad k = 0, 1, \cdots N_c - 1, \tag{440}$$

change with the block size N_c? To understand this, let us first consider the *discrete time Fourier transform* $H(f)$ of the channel: it is given by

$$H(f) = \sum_{l=0}^{L-1} h_l e^{-j2\pi \frac{lf}{W}} \quad f \in [0, W]. \tag{441}$$

Thus the channel coefficient $\tilde{h}_k$ is $H(f)$ evaluated at $f = \frac{kW}{N_c}$. Now we see that as the number of subcarriers N_c grows, the frequency width $\frac{W}{N_c}$ of each subcarrier goes to zero and they represent finer and finer sampling of the continuous spectrum. Further, the overhead $L - 1$ becomes negligible as compared to N_c. Thus, as N_c goes to infinity, the capacity achieved by the OFDM channel reaches the capacity of the wireline channel (cf. Equation (439) and the substitution $\sigma^2 = N_0/2$):

$$C = \int_0^W \frac{1}{2} \log_2 \left(1 + \frac{2P^*(f)|H(f)|^2}{N_0}\right) \mathrm{d}f \quad \text{bits/s.} \tag{442}$$

Here we have denoted the optimal power allocation function

$$P^*(f) = \left(\frac{1}{\lambda} - \frac{N_0}{2|H(f)|^2}\right)^+, \tag{443}$$

where the constant λ satisfies

$$\int_0^W P^*(f) \mathrm{d}f = P. \tag{444}$$

Looking Ahead

This brings us to an end to our study of reliable communication over the wireline channel. In the next several lectures we turn to reliable wireless communication. We will see that though wireless communication has its own unique challenges, our understanding of the wireline channel forms a basic building block. We will also see that OFDM will come to play an important role in reliable wireless communication.

Lecture 18: Passband Wireless Communication

Introduction

Beginning with this lecture, we will study wireless communication. The focus of these lectures will be on point to point communication. Communication on a wireless channel is inherently different from that on a wireline channel. The main difference is that unlike wireline channel, wireless is a shared medium. The medium is considered as a federal resource and is federally regulated. The entire spectrum is split into many licensed and unlicensed bands. An example of the the point to point communication in the licensed band is the cellular phone communication, whereas wi-fi, cordless phones and blue tooth are some of the examples of communication in the unlicensed band.

The transmission over a wireless channel is restricted to a range of frequencies $\left[f_c - \frac{W}{2}, f_c + \frac{W}{2}\right]$ around the central carrier frequency f_c. Typically

$$f_c \gg W; \tag{445}$$

for example, $W \approx 1$ MHz and $f_c \approx 1$ GHz for cellular communication. On the other hand, the wireline channel is quite a contrast: the carrier frequency $f_c = 0$. Further more, the same wireless system (say cell phones) use different carrier frequencies in different cities. Clearly, it is not practical to tailor the communication strategies to the different carrier frequencies. It would be a lot better if we could design the system for a fixed carrier frequency and then have a simple mechanism to translate this design to suit the actual carrier frequency of operation.

This is indeed possible, and the fixed carrier frequency might as well be zero. This has the added advantage that it lets us borrow our earlier understanding of communication strategies on the wireline channel. In other words, the plan is to design for "baseband" even though we are looking to communicate in passband. The focus of this lecture is on this conversion. Finally we also see how the effects of the wireless channel (which is in passband, after all) translate to the baseband: in other words, we derive a "baseband equivalent" of the passband wireless channel.

Baseband Representation of Passband Signals

Let's begin with a baseband signal $x_b(t)$ (of double sided bandwidth W) that we want to transmit over the wireless channel in a band centered around f_c. From our discussion in the

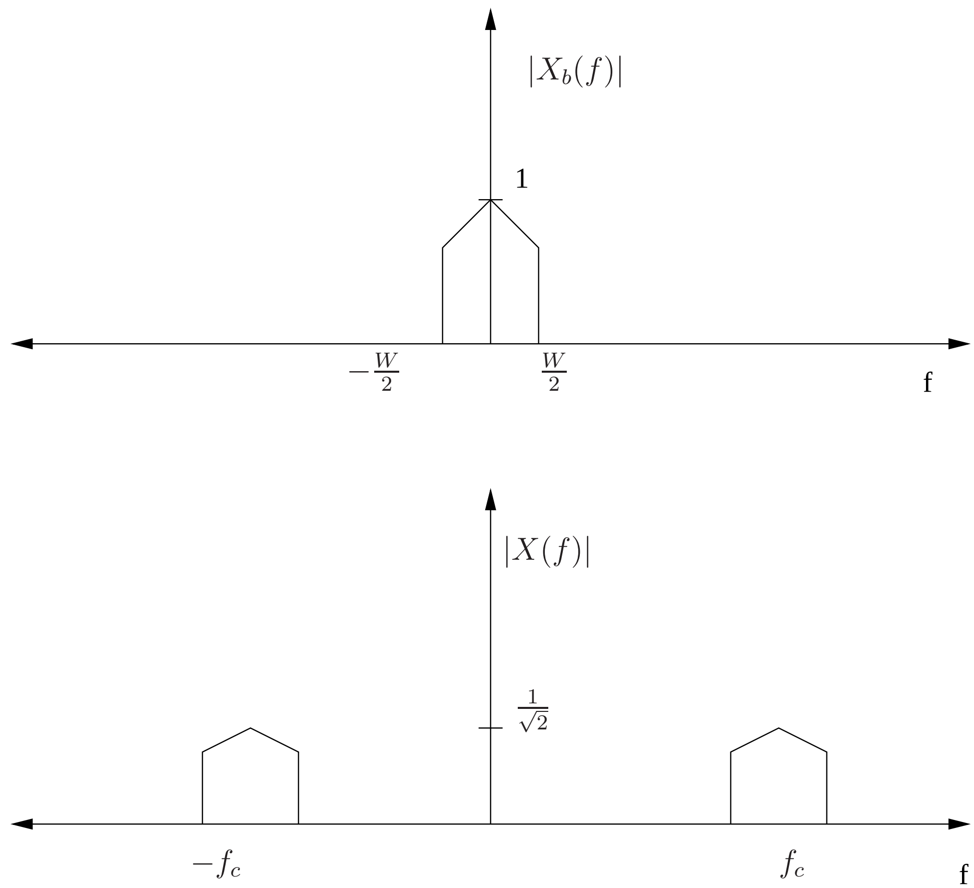

Figure 35: Magnitude spectrum of the real baseband signal and its passband signal.

wireline channel, $x_b(t)$ would be the signal at the output of the DAC at the transmitter. We can "up convert" this signal by multiplying it by $\cos 2\pi f_c t$:

$$x(t) = \sqrt{2} \cos 2\pi f_c t \tag{446}$$

is a passband signal, i.e., it's spectrum is centered around f_c and $-f_c$. Figure 35 shows this transformation diagrammatically. We scale the carrier by $\sqrt{2}$ as $\cos 2\pi f_c t$ has power $\frac{1}{2}$. Thus, by scaling, we are keeping the power in $x_b(t)$ and $x(t)$ same.

To get back the baseband signal, we multiply $x(t)$ again by $\sqrt{2}\cos 2\pi f_c t$ and then pass the signal through a low pass filter with bandwidth W:

$$x(t)\sqrt{2}\cos 2\pi f_c t = 2\cos^2(2\pi f_c t) x_b(t) \tag{447}$$

$$= (1 + \cos 4\pi f_c t) x_b(t) \tag{448}$$

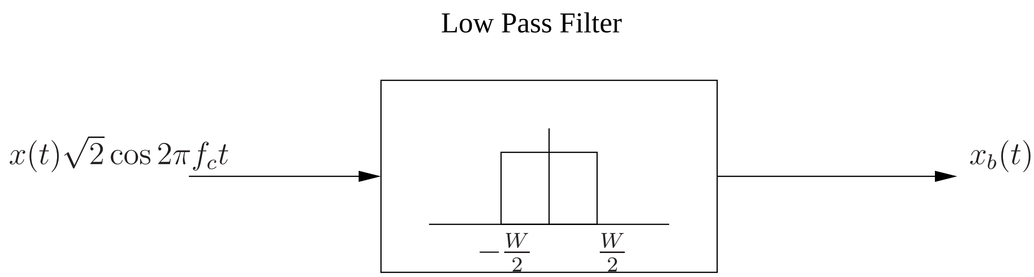

Figure 36: Down-conversion at the receiver.

The low pass filter will discard the signal $x_b(t)\cos 4\pi f_c t$ as it is a passband signal (centered around $2f_c$). Figure 36 shows this transformation diagrammatically.

One can see that if we multiply $x(t)$ by $\sqrt{2}\sin 2\pi f_c t$ instead of $\sqrt{2}\cos 2\pi f_c t$, we get $x_b(t)\sin 4\pi f_c t$ which would be eliminated completely by the low pass filter. There will be a similar outcome had we modulated the baseband signal on $\sqrt{2}\sin 2\pi f_c t$ and tried to recover it by using $\sqrt{2}\cos 2\pi f_c t$. This observation suggests the following engineering idea:

> We should upconvert two baseband signals, one using the cosine and the other using the sine and add them to make up the passband transmit signal. The receiver can recover each of the baseband signals by down converting by mixing with the cosine and sine waveforms in conjunction with a low pass filter.

Thus, we transmit *two* baseband signals in the same frequency band to create one passbnd signal. This is really possible because the total double sided bandwidth of a passband signal is $2W$ instead of just W in a baseband signal. To conclude: the passband signal $x(t)$ is created as

$$x(t) = x_{b_1}(t)\sqrt{2}\cos 2\pi f_c t - x_{b_2}(t)\sqrt{2}\sin 2\pi f_c t \tag{449}$$

The baseband signals $x_{b_1}(t)$ and $x_{b_2}(t)$ are obtained at the receiver by multiplying $x(t)$ by $\sqrt{2}\cos 2\pi f_c t$ and $\sqrt{2}\sin 2\pi f_c t$ separately and then passing both the outputs through low pass filters. We are modulating the *amplitude* of the carrier using the baseband signal and as such, this scheme is called *amplitude modulation*. When we use both the sin and cos parts of the carrier in the modulation process, the scheme is called *Quadrature Amplitude Modulation* (QAM). It is interesting to see if we can modulate a carrier using more than two independent baseband signals and yet recover each perfectly at the receiver. However this is not the case: from a basic trigonometric equality,

$$\cos(2\pi f_c t + \theta) = \cos\theta\cos 2\pi f_c t - \sin\theta\sin 2\pi f_c t. \tag{450}$$

We see that any modulation phase θ is uniquely determined by the amplitudes of $\cos 2\pi f_c t$ and $\sin 2\pi f_c t$. Thus we conclude that a single passband signal is exactly represented by two baseband signals. The baseband signal $x_b(t)$ is now defined in terms of the pair

$$(x_{b_1}(t), x_{b_2}(t)). \tag{451}$$

In the literature, this pair is denoted as

$$(x_b^I(t), x_b^Q(t)), \tag{452}$$

where I stands for "in phase" signal and Q stands for "quadrature phase" signal. To make the notation compact we can think of $x_b(t)$ as a *complex* signal defined as follows:

$$x_b(t) \stackrel{\text{def}}{=} x_b^I(t) + j x_b^Q(t). \tag{453}$$

This allows us to represent passband signals by a single *complex* baseband signal. It also makes much of the material from our study of baseband communication (over the wireline channel) to be readily used in our passband scenario.

The Passband AWGN Channel

As in the wireline channel, the wireless channel is also well represented by a linear system. The main difference is that this channel is *time varying*, as opposed to the time-invariant nature of the wireline channel. For now, let us suppose that the wireless channel is time invariant. This will allow us to focus on the passband nature of the signals and channels first. We will return to the time variation at a later point in the lectures. Denoting by $h(t)$, the impulse response of the (time-invariant) wireless channel, the received passband signal is

$$y(t) = h(t) * x(t) + w(t). \tag{454}$$

Here $w(t)$ is passband noise. We can use the complex baseband representation derived earlier (in the context of the transmit passband signal $x(t)$) for the passband received signal $y(t)$ and the passband noise $w(t)$ (denoted, naturally, by $y_b(t)$ and $w_b(t)$ respectively). As can be expected, the passband signal equation in Equation (454) turns into the following baseband signal equation:

$$y_b(t) = h_b(t) * x_b(t) + w_b(t). \tag{455}$$

The key question is:

> How is the "baseband equivalent" $h_b(t)$ related to the passband channel $h(t)$?

The answer to this question will let us work directly with the baseband channel in Equation (455) which is almost the same as the one for the wireline channel (except that all signals are complex instead of real).

To understand the relation between $h(t)$ and $h_b(t)$, let us consider a few examples.

1. We start with a very simple scenario:

$$h(t) = \delta(t). \tag{456}$$

Now

$$y(t) = x(t) + w(t) \tag{457}$$

and hence

$$y_b(t) = x_b(t) + w_b(t). \tag{458}$$

We conclude for this case that

$$h_b(t) = h(t) = \delta(t). \tag{459}$$

2. Now we move to a slightly more complicated scenario:

$$h(t) = \delta(t - t_0). \tag{460}$$

Now

$$y(t) = x(t-t_0) = x_b^I(t-t_0)\sqrt{2}\cos 2\pi f_c(t-t_0) - x_b^Q(t-t_0)\sqrt{2}\sin 2\pi f_c(t-t_0). \tag{461}$$

The baseband signal $y_b(t)$ is composed of:

$$\begin{aligned}
y_b^I(t) &= \text{low pass filter output of } \left(y(t)\sqrt{2}\cos 2\pi f_c t\right) && (462)\\
&= \text{low pass filter output of } \Big((2\cos 2\pi f_c(t-t_0)\cos 2\pi f_c t)x_b^I(t-t_0) \\
&\quad -(2\sin 2\pi f_c(t-t_0)\cos 2\pi f_c t)x_b^Q(t-t_0)\Big) && (463)\\
&= \text{low pass filter output of } \Big((\cos 2\pi f_c(2t-t_0) + \cos 2\pi f_c t_0)x_b^I(t-t_0) \\
&\quad -(\sin 2\pi f_c(2t-t_0) - \sin 2\pi f_c t_0)x_b^Q(t-t_0)\Big) && (464)\\
&= x_b^I(t-t_0)\cos 2\pi f_c t_0 + x_b^Q(t-t_0)\sin 2\pi f_c t_0 && (465)\\
&= \Re\left\{x_b(t-t_0)e^{-j2\pi f_c t_0}\right\}. && (466)
\end{aligned}$$

Similarly, we obtain $y_b^Q(t)$ as

$$\begin{aligned}
y_b^Q(t) &= \text{low pass filter output of } \left(-y_b(t)\sqrt{2}\sin 2\pi f_c t\right) && (467)\\
&= \Im\left\{x_b(t-t_0)e^{-j2\pi f_c t_0}\right\}. && (468)
\end{aligned}$$

Thus,

$$y_b(t) = x_b(t - t_0)e^{-j2\pi f_c t_0}. \tag{469}$$

We conclude that

$$\begin{aligned} h_b(t) &= e^{-j2\pi f_c t_0}\delta(t - t_0) & (470)\\ &= e^{-j2\pi f_c t_0}h(t). & (471) \end{aligned}$$

We notice that the baseband signal also gets delayed by the same amount as the passband signal. The key change is in the phase of the baseband signal (remember, it is a complex signal) depends on both the carrier frequency and time shift.

We are now ready to generalize: a general wireless channel is typically a sum of attenuated and delayed copies of the transmit signal. Specifically:

- the attenuation is due to the energy lost at the transmit and receive antennas, as well as any absorption in the medium;
- the delay is due to the time taken to travel the distance between the transmitter and receiver;
- the multiple copies are due to the fact that there are many physical paths between transmitter and receiver that are of the same order of attenuation.

In mathematical notation,

$$h(t) = \sum_i a_i \delta(t - \tau_i). \tag{472}$$

The complex baseband equivalent of the channel is, as a natural generalization of the examples we did earlier,

$$h_b(t) = \sum_i a_l e^{-j2\pi f_c \tau_i}\delta(t - \tau_i). \tag{473}$$

Looking Ahead

In this lecture we saw that passband communication can be represented readily by complex baseband communication. This representation is both in terms of the notation and mathematics, but also in terms of concrete engineering: we only need to upconvert the complex baseband signal at the transmitter and downconvert at the receiver. In the next lecture, we will add the usual DAC and ADC blocks and arrive at the discrete time complex baseband representation of passband communication. Finally, we will also be able to arrive at appropriate statistical models of the wireless channel.

Lecture 19: The Discrete Time Complex Baseband Wireless Channel

Introduction

In the previous lecture we saw that even though the wireless communication is done via passband signals, most of the processing at the transmitter and the receiver happens on the (complex) baseband equivalent signal of the real passband signal. We saw how the baseband to passband conversion is done at the transmitter. We also studied simple examples of the wireless channel and related it to the equivalent channel in the baseband. The focus of this lecture is to develop a robust model for the wireless channel. We want the model to capture the essence of the wireless medium and yet be generic enough to be applicable in all kinds of surroundings.

A Simple model

Figure 37 shows the processing at the transmitter. We modulate two data streams to generate the sequence of complex baseband voltage points $x_b[m]$. The real and imaginary parts of $x_b[m]$ pass through the D/A converter to give baseband signal $x_b(t)$. Real and imaginary parts of $x_b(t)$ then modulates cos and sin parts of the carrier to generate the passband signal $x(t)$. The passband signal $x(t)$ is transmitted in the air and the signal $y(t)$ received.

Given all the details of the reflectors and absorbers in the surroundings, one can possibly use Maxwell's equations to determine the propagation of the electromagnetic signals and get $y(t)$ as an exact function of $x(t)$. However, such a detailed model is neither required nor is desired. The transmitter and receiver antennas are typically separated by several wavelengths apart and far field approximations of the signal propagation are good enough. Secondly, we do not want the model to be very specific to certain surrounding. We want the model to be applicable to most of the surroundings and still be meaningful.

We can model the electromagnetic signal as rays. As the rays travel in the air, they get attenuated. There is a nonzero propagation delay that each ray experiences. Further, the rays gets reflected by different reflectors before reaching the receiver. Thus, the signal arrives at the receiver via multiple paths, each of which sees different delay and attenuation. There is also an additive noise present at the receiver.

Hence, we can have a simple model for the received signal $y(t)$ as

$$y(t) = \sum_i a_i x(t - \tau_i) + w(t), \tag{474}$$

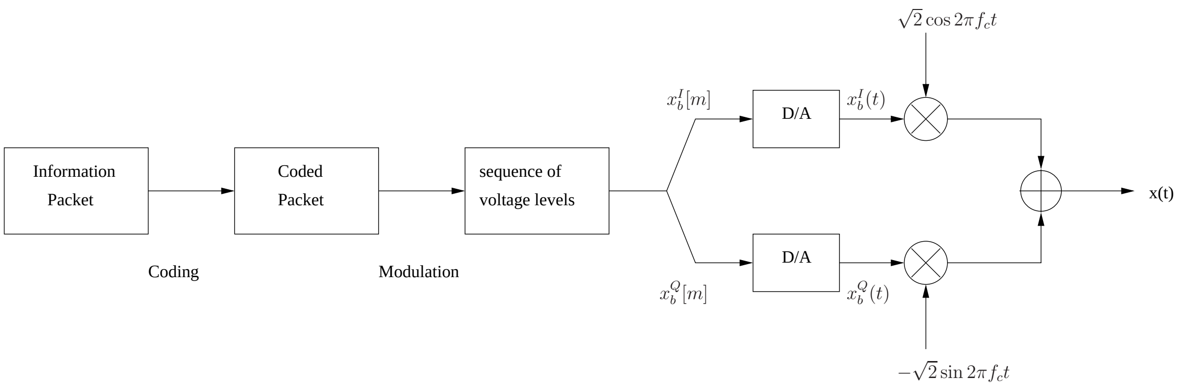

Figure 37: Diagrammatic representation of transmitter.

where a_i is the attenuation of the i^{th} path and τ_i is the delay it experiences and $w(t)$ denotes the additive noise.

The delay τ_i is directly related to the distance traveled by the path i. If d_i is the distance traveled by the path i, then the delay is

$$\tau_i = \frac{d_i}{c} \tag{475}$$

where c is the speed of light in air. The typical distances traveled by the direct and reflected paths in the wireless scenario ranges from of the order of 10 m (in case of Wi-Fi) to 1000 m (in case of cellular phones). As $c = 3 \cdot 10^8$ m/s, this implies that the delay values can range from 33 ns to 3.3 μs. The delay τ depends on the path length and is same for all the frequencies in the signal.

Another variable in Equation (474) is the attenuation a_i. In free space the attenuation is inversely proportional to the distance traveled by the path i, i.e., $a_i \propto \frac{1}{d_i}$. In the terrestrial communication, the attenuation depneds on the richness of the environment with respect to the scatterers. Depending on the environment, it can vary from $a_i \propto \frac{1}{d_i^2}$ to $a_i \propto e^{-d_i}$.

Scatterers can have different absorption coefficients for the different frequencies and the attenuation can depend on the frequency. However, we are communicating in a narrow band (in KHz) arround a high frequency carrier (in GHz). Thus, the variation within the band of interest are insignificant.

However, the most important aspect of the wireless communication is that the transmitter, the receiver and the surrounding are not stationary during the communication. Hence the number of path arriving at the receiver and the distance they travel (and hence the delay and the attenuation they experience) change with time. All these parameters are then

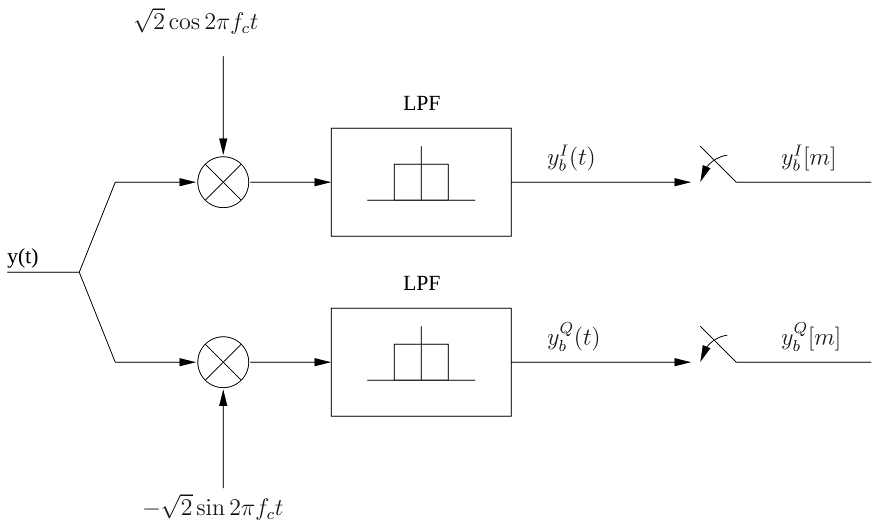

Figure 38: diagramatic representation of receiver

functions of time. Hence, Equation (474) should be modified to incorporate this factor.

$$y(t) = \sum_i a_i(t)x(t - \tau_i(t)) + w(t). \tag{476}$$

At the receiver $y(t)$ is down-converted to the baseband signal $y_b(t)$. Its real and imaginary parts are then sampled at the sampling rate W samples per second. Figure 38 depicts these operations.

Discrete Time Channel model

Since the communication is in the discrete time instances, we want a model for the channel between $x_b[m]$ and $y_b[m]$. Let us try to obtain the discrete time baseband channel model from Equation (474). We keep in mind that the delays and the attenuation of the paths are time varying, though we will not explicitly write them as functions of time in the following discussion. Equation (474) can be written as

$$y(t) = h(t) * x(t) + w(t) \tag{477}$$

where, $h(t)$ is the impulse response of the channel and is given by

$$h(t) = \sum_i a_i \delta(t - \tau_i). \tag{478}$$

From the previous lecture, we know the impulse response of the baseband channel $h_b(t)$ is given by

$$h_b(t) = \sum_i a_i e^{-j2\pi f_c \tau_i} \delta(t - \tau_i) \tag{479}$$

and the baseband received signal is

$$y_b(t) = h_b(t) * x_b(t) + w_b(t). \tag{480}$$

We know that $y_b[m]$ is obtained by sampling $y_b(t)$ with sampling interval $T = \frac{1}{W}$s.

$$\begin{aligned} y_b[m] &= x_b(mT) + w_b(mT) && (481) \\ &= \sum_i x_b(mT - \tau_i) a_i e^{-j2\pi f_c \tau_i} + w_b(mT) && (482) \end{aligned}$$

Recall that $x_b(t)$ is obtained from $x_b[n]$ by passing it through the pulse shaping filter. Assuming the ideal pulse shaping filter $\text{sinc}(\frac{t}{T})$, $x_b(t)$ is

$$x_b(t) = \sum_n x_b[n] \text{sinc}\left(\frac{t - nT}{T}\right). \tag{483}$$

Substituting in Equation (482), we get

$$\begin{aligned} y_b[m] &= \sum_n \sum_i x_b[n] \text{sinc}\left(m - n - \frac{\tau_i}{T}\right) a_i e^{-j2\pi f_c \tau_i} + w_b[m] && (484) \\ &= \sum_n x_b[n] \left(\sum_i a_i e^{-j2\pi f_c \tau_i} \text{sinc}\left(m - n - \frac{\tau_i}{T}\right)\right) + w_b[m]. && (485) \end{aligned}$$

Substituting $\ell := m - n$, we get

$$\begin{aligned} y_b[m] &= \sum_\ell x_b[m - \ell] \left(\sum_i a_i e^{-j2\pi f_c \tau_i} \text{sinc}\left(\ell - \frac{\tau_i}{T}\right)\right) + w_b[m] && (486) \\ &= \sum_{\ell=0}^{L-1} h_\ell x_b[m - \ell] + w_b[m], && (487) \\ & && (488) \end{aligned}$$

where the tap coefficient h_ℓ is defined as

$$h_\ell \stackrel{\text{def}}{=} \sum_i a_i e^{-j2\pi f_c \tau_i} \text{sinc}\left(\ell - \frac{\tau_i}{T}.\right) \tag{489}$$

We recall that these are exactly the same calculations as for obtaining the tap coefficients for the wireline channel in Lecture 9. From Lecture 9, we recall that if T_p is the pulse width and T_d is the total delay spread, then the number of taps L are

$$L = \lfloor \frac{T_p + T_d}{T} \rfloor \tag{490}$$

where the delay spread T_d is the difference between the delays between the shortest and the longest path.

$$T_d \stackrel{\text{def}}{=} \max_{i \neq j} |\tau_i - \tau_j|. \tag{491}$$

Note that Equation (487) also has the complex noise sample $w_b[m]$. It is the sampled baseband noise $w_b(t)$. We model the discrete noises $w_b[m]$, $m \geq 1$, as i.i.d. complex Gaussian random variables. Further we model both the real and imaginary parts of the complex noise as i.i.d. (real) Gaussian random variables with mean 0 and variance $\frac{\sigma^2}{2}$.

Equation (487) is exactly the same as that of wireline channel equation. However, the wireline channel and wireless channel are not the same. We recall that the delays and attenuations of the paths are time varying and hence the tap coefficients are also time varying. We also note unlike in the wireline channel, both h_l and $x_b[m]$ are complex numbers.

The fact that the tap coefficients h_l and even the number of taps L are time varying is a distinguishing feature of the wireless channel. In wireline channel the taps do not change and hence they can be learned. But now we cannot learn them once and use that knowledge for rest of the communication. Further, the variations in the tap coefficients can be huge. It seems intuitive that the shortest paths will add up in the first tap and since these paths are not attenuated much, h_0 should always be a good tap. It turns out that this intuition is misleading. To see this, let's consider Equation (489). Note that the paths whose delays are separated by at most T seconds. For the tap h_0, $\tau_i \leq T$ and we can approximate $\text{sinc}\left(-\frac{\tau_i}{T}\right) \approx 1$.

But note that the phase term $e^{-j2\pi f_c \tau_i}$ can vary a lot. The paths that have a phase lag π will have

$$f_c(\tau_1 - \tau_2) = \frac{1}{2} \tag{492}$$

$$\tau_1 - \tau_2 = \frac{1}{2f_c}. \tag{493}$$

With $f_c = 1$ GHz, $\tau_1 - \tau_2 = 0.5ns$. This corresponds to the difference in their path lengths to be of 15 cm. Thus, there could be many paths adding constructively and destructively and we could have a low operating SNR even when the transmitter and receiver are right next to each other.

We can now see that the key primary difference between the wire line and the wireless channels is in the magnitudes of the channel filter coefficients: in a wireline channel they are usually in a prespecified (standardized) range. In the wireless channel, however:

1. the channel coefficients can have a widely varying magnitude. Since there are multiple paths from the transmitter to the receiver, the overall channel at the receiver can still have a very small magnitude even though each of the individual paths are very strong.

2. the channel coefficients change in time as well. If the change is slow enough, relative to the sampling rate, then the overhead in learning them dynamically at the receiver is not much.

Even if the wireless channel can be tracked dynamically by the receiver, the communication engineer does not have any idea *a priori* what value to expect. This is important to know since the resource decisions (power and bandwidth) and certain traffic characteristics (data rate and latency) are fixed a priori by the communication engineer. Thus there is a need to develop *statistical* knowledge of the channel filter coefficients which the communication engineer can then use to make the judicial resource allocations to fit the desired performance metrics. This is the focus of the rest of this lecture.

Statistical Modeling of The Wireless Channel

At the outset, we get a feel for what statistical model to use by studying the reasons why the wireless channel varies with time. The principal reason is mobility, but it helps to separate the role as seen by different components that make up the overall discrete time baseband channel model.

1. The arrival phase of the constituent paths making up a single filter coefficient (i.e., these paths all arrive approximately within a single sample period) may change. The arrival phase of any path changes by π radians when the distance of the path changes by half a wavelength. If the relative velocity between the transmitter and receiver for path i is v_i, then the time to change the arrival phase by π radians is

$$\frac{c}{2 f_c v_i} \quad \text{seconds.} \tag{494}$$

   Substituting for the velocity of light c (as 3×10^8 m/s) and sample carrier frequency f_c values (as 10^9 Hz) and gasoline powered velocity v_i (as 60 mph) we get the time of phase reversal to be about 5 ms.

2. Another possibility is that a new path enters the aggregation of paths already making up a given channel filter tap. This can happen if the path travels a distance less (or more) than previously by an order of the sampling period. Since sampling period T is inversely related to the bandwidth W of communication and the bandwidth is typically three orders or so less than the carrier frequency (say, $W = 10^6$ Hz), this event occurs over a time scale that is three orders of magnitude larger than that of phase change

(so this change would occur at about 5 seconds, as opposed to the earlier calculation of 5 ms). As such, this is a less typical way (than the previous one) in which channel can change.

3. It could happen that the path magnitudes change with time. But this requires the distance between the transmitter and receiver to change by a factor of two or so. With gasoline powered velocities and outdoor distances of 1 mile or so, we need several seconds for this event to occur. Again, this is not the typical way channel would change.

In conclusion:

> The time scale of channel change is called *coherence interval* and is dominated by the effect of the phase of arrival of different paths that make up the individual channel filter tap coefficients.

How we expect the channel to change now depends on how many paths aggregate within a single sample period to form a single channel filter tap coefficient.

We review two popular scenarios below and arrive at the appropriate statistical model for each.

- **Rayleigh Fading Model**: When there are many paths of about the same energy in each of the sampling periods, we can use the central limit theorem (just as in Lecture 2) to arrive at a Gaussian approximation to the channel filter coefficients. Since the channel coefficient is a complex number, we need to arrive at a statistical model for both the real and imaginary parts. A common model is to suppose that both the real and imaginary parts are statistically independent and identically distributed to be Gaussian (typically with zero mean). The variance is proportional to the energy attenuation expected between the transmitter and receiver: it typically depends on the distance between the transmitter and receiver and on some gross topographical properties (such as indoors vs outdoors).

- **Rician Fading Model**: Sometimes one of the paths may be strongly dominant over the rest of the paths that aggregate to form a single channel filter coefficient. The dominant path could be a line of sight path between the transmitter and receiver while the weaker paths correspond to the ones that bounce off the objects in the immediate neighborhood. Now we can statistically model the channel filter coefficient as a Rayleigh fading with a non-zero mean. The stronger the dominant path relative to the aggregation of the weaker paths, the larger the ratio of the mean to the standard deviation of the Rayleigh fading model.

Looking Ahead

Starting next lecture we turn to using the statistical knowledge of the channel to communicate reliably over the wireless channel.

Lecture 20: Sequential Communication over a Slow Fading Wireless Channel

Introduction

Consider the simple *slow* fading wireless channel we arrived at in the previous lecture (the subscript b is dropped for notational convenience):

$$y[m] = hx[m] + w[m], \quad m = 1, \ldots, N. \tag{495}$$

Here NT is the time scale (the product of N, the number of time samples communicated over, and T, the sampling period) of communication involved. The channel coefficient h is well modeled as independent of m if NT is much smaller than the coherence time T_c of the channel. This is a very common occurrence in many practical wireless communication systems and we begin our study of reliable communication over the wireless channel. We will start with sequential communication, much as we started out with the AWGN channel (cf. Lecture 3). The goal of this lecture is to be able to compare and contrast the simple slow fading channel in Equation (495) with the familiar AWGN channel. In particular, we will calculate the unreliability of communicating a single bit as a function of the SNR. The main conclusion is the observation of how poorly the unreliability decays with increasing SNR. This is especially stark when compared to the performance over the AWGN channel.

Comparison with AWGN Channel Model

The channel model in Equation (495) is quite similar to that of the AWGN channel model. But there are differences as well, with some aspects being more important than others.

1. The transmit and receive symbols are complex numbers (pair of voltages) as opposed to the real numbers in the AWGN channel. This is a relatively minor point and poses hardly any trouble to our calculations and analysis (as seen further in this lecture).

2. The channel "quality" h can be learnt by the receiver (through the transmission of pilot symbols, cf. Lecture 9), but is not known a priori to the communication engineer. This is a very important difference, as we will see further in this lecture. The important point is that the calculation of the unreliability level is now to be calculated in terms of the knowledge the communication engineer has about h: the *statistical* characterization of h.

In this lecture we make the following suppositions about the channel quality h:

1. the channel quality h is learnt very reliably at the receiver. Since we have several, namely N samples to communicate over, we could spend the first few samples in

transmitting knows voltages (pilots) thus allowing the receiver to have a very good estimate of h. We will suppose that the receiver knows h exactly, and not bother to model the fact that there will be some error in the estimate of the channel as opposed to the true value.

2. a statistical characterization of h is available to the communication engineer: we study the relation between unreliability level and SNR in the context of a simple statistical model: *Rayleigh* fading.

Sequential Communication

Suppose we transmit a single information bit every time instant, using the modulation symbols $\pm\sqrt{E}$. We decode each bit separately over time as well. Focusing on a specific time m, we can write the received complex symbol as

$$\Re[y[m]] = \Re[h]\, x[m] + \Re[w[m]], \quad (496)$$
$$\Im[y[m]] = \Im[h]\, x[m] + \Im[w[m]]. \quad (497)$$

This follows since $x[m]$ is real (and restricted to be $\pm\sqrt{E}$). Since the receiver known $\Re[h]$ and $\Im[h]$, a sufficient statistic of the transmit symbol $x[m]$ is (cf. Lecture 5):

$$\tilde{y} \stackrel{\text{def}}{=} \Re[h]\,\Re[y[m]] + \Im[h]\,\Im[y[m]] \quad (498)$$
$$= \Re[h^* y[m]] \quad (499)$$
$$= |h|^2 x[m] + \tilde{w}. \quad (500)$$

Here we have denoted h^* as the *complex conjugate* of h: the real part of h and h^* are identical, but the imaginary part of h^* is the negative of that of h. Further,

$$\tilde{w} \stackrel{\text{def}}{=} \Re[h^* w[m]] = \Re[h]\,\Re[w[m]] + \Im[h]\,\Im[w[m]] \quad (501)$$

is real valued and has Gaussian statistics: zero mean and variance $\frac{1}{2}|h|^2\sigma^2$. Now the ML receiver to detect $x[m]$ from $\tilde{y}$ is very clear: Equation (500) shows that the relation is simply that of an AWGN channel and we arrive at the nearest neighbor rule:

$$\text{decide } x[m] = +\sqrt{E} \text{ if } \tilde{y} > 0, \quad (502)$$

and vice versa if $\tilde{y} \leq 0$. The corresponding error probability is

$$Q\left(\sqrt{2|h|^2 \mathsf{SNR}}\right), \quad (503)$$

where we have written $\mathsf{SNR} = \mathsf{E}/\sigma^2$ as usual.

Average Error Probability

At this point, it is important to observe the nature of the unreliability level calculated in Equation (503) depends on the actual value of the channel quality (h) experienced during communication. But the communication engineer who has to decide how to set the operating value of SNR does not have access to the actual value of h. Only a statistical characterization of h is available to the engineer. One natural way to use the *dynamic* unreliability level calculated in Equation (503) to calculate a quantity useful to the communication engineer is to take the statistical average of the dynamic unreliability levels in Equation (503):[8]

$$P_e \stackrel{\text{def}}{=} \mathbb{E}\left[Q\left(\sqrt{2|h|^2\mathsf{SNR}}\right)\right], \tag{504}$$

where the average is with respect to the statistical characterization of h available to the communication engineer.

The average of the dynamic unreliability level in Equation (504) can be calculated based on the statistics of $|h|^2$. When h has the Rayleigh statistics we mean that the real and imaginary parts of h are i.i.d. Gaussian (zero mean and variance $\frac{A}{2}$ each). Here the quantity A stands for the *attenuation* (ratio of received energy to transmit energy in the signal) caused by the channel. It turns out that the statistics of $|h|^2$ are very simple – the density turns out to be exponential (most communication books will give you this result – just look for Rayleigh distribution in the index):

$$f_{|h|^2}(a) = \frac{1}{A}e^{-\frac{a}{A}}, \quad a \geq 0. \tag{505}$$

We can use the exponential statistics of $|h|^2$ to explicitly evaluate the average probability over the Rayleigh slow fading wireless channel:

$$P_e = \int_0^\infty Q\left(\sqrt{2a\mathsf{SNR}}\right)\frac{1}{A}e^{-\frac{a}{A}}\,\mathrm{d}a \tag{506}$$

$$= \frac{1}{2}\left(1 - \sqrt{\frac{A\mathsf{SNR}}{1 + A\mathsf{SNR}}}\right). \tag{507}$$

The expression in Equation (507) is evaluated using the formula for "integrating-by-parts"; you can also take look at Equation 3.19 of the book *Fundamentals of Wireless Communication* by Tse and Viswanath. It is interesting to note that while the dynamic unreliability level did not have a concrete closed form expression (we could only write it as a Q function), the average of the dynamic unreliability level indeed has a simple formula.

[8]Note that this is not the only way to connect the dynamic unreliability level to one that depends only on the statistical characterization of the channel quality. Another approach would be to consider the smallest unreliability level that at least a fixed fraction, say 90%, of the channel qualities meet.

Average Unreliability vs SNR

Now we, as communication engineers, are in a position to decide what value of SNR to operate at given the need for a certain reliability level of communication. To do this, it will help to simplify the expression in Equation (507): using the Taylor series expansion

$$\sqrt{1+x} \approx 1 + \frac{x}{2}, \quad x \approx 0, \tag{508}$$

we can approximate the expression in Equation (507) as

$$P_e \approx \frac{1}{4A\mathsf{SNR}}, \quad A\mathsf{SNR} \gg 1. \tag{509}$$

Now we see the benefits of increasing SNR: for every doubling of SNR, the unreliability level only *halves*. This is in stark contrast to the behavior of the AWGN channel where the unreliability level *squared* for every doubling of SNR (cf. Lecture 3). To get a feel for how bad things are, let us consider a numerical example. If we desire a bit error probability of no more than 10^{-10} and have an attenuation of $A = 0.01$, then we are looking at a required SNR of $25 \cdot 10^{10}$! This corresponds to astronomical transmit powers – clearly physically impossible to meet.

Looking Ahead

We have seen that the performance of sequential communication is entirely unacceptable at physically attainable SNR levels. This is serious motivation to look for better strategies. One possibility is, continuing along the line of thought early in this course, to study block communication schemes. Block communication schemes improved the reliability level while maintaining non-zero communication rates. The goal of block communication was primarily to better exploit the statistical nature of the additive noise by mapping information packet directly to the transmit symbol vector (of high dimension). As such, it was a way to deal with additive noise. In the wireless channel, we have an additional source of noise: the channel quality h itself is random and it shows up in a *multiplicative* fashion. So, it is not entirely clear if block communication which was aimed at ameliorating the effects of additive noise would work too well in dealing with multiplicative noise. We get a better feel for this aspect in the next lecture where we see that block communication cannot improve the performance significantly beyond the expression in Equation (509).

Lecture 21: Typical Error Event in a Slow Fading Wireless Channel

Introduction

We studied sequential communication over a slowly varying wireless channel in the previous lecture. The key feature of our channel was that it so slowly changing in time that it is practically time-invariant over the course of communication. Mathematically, our model is:

$$y[m] = hx[m] + w[m], \quad m = 1, \ldots, N. \tag{510}$$

We defined the probability of error with sequential communication over this channel as the average of the dynamic probability of error. With Rayleigh fading (i.e., $|h|^2$ has exponential density with mean A) we had the following relationship between SNR and unreliability of a single bit:

$$P_e = \frac{1}{2}\left(1 - \sqrt{\frac{A\mathsf{SNR}}{1 + A\mathsf{SNR}}}\right) \tag{511}$$

$$\approx \frac{1}{4A\mathsf{SNR}}, \quad A\mathsf{SNR} \gg 1. \tag{512}$$

The main conclusion is that the probability of error decays only linearly with increase in SNR. This is in stark contrast to the situation in the AWGN channel (or a wireline channel) where we had an exponential decay. This, of course, requires huge improvements in the P_e vs SNR interdependence, for any communication over the wireless channel to be acceptable. As is clear from the Equation (512), there are two factors that cause errors over the wireless channel: the first is the additive noise w[m] (just as it was for the wireline channel) and the second (novel) factor h is multiplicative in nature. In this lecture we focus on isolating the effects of each of these factors in an attempt to figure out which one is more crucial. This insight will be used in later lectures to improve the poor relation between unreliability level and SNR that is exhibitited in Equation (512).

Sequential Communication

Consider sequential communication of a single bit x from the set $\pm\sqrt{E}$. We use the nearest neighbor rule on a real sufficient statistic

$$\tilde{y}[m] = \Re\left[\frac{h^*}{|h|} y[m]\right] \tag{513}$$

$$= |h| x[m] + \tilde{w}[m]. \tag{514}$$

Here $\tilde{w}[m]$ is real and Gaussian, with zero mean and variance $\frac{\sigma^2}{2}$. For example, the nearest neighbor rule at a time m (index is suppressed here) could be

$$\text{decide } x = +\sqrt{E} \text{ if } \tilde{y} > 0. \tag{515}$$

Now suppose we send $x = -\sqrt{E}$. Then an error occurs exactly when

$$\tilde{w} > |h|\sqrt{E}. \tag{516}$$

Observe that the error occurs due to the combined effect of $|h|$ and $\tilde{w}$. Thus there are two factors at play here when an error occurs. At a rough level we see that the smaller the $|h|$, the more likely the chance of an error. Let us consider only such events when $|h|$ is smaller than a threshold, denoted below by a_{th}:

$$|h|^2 < \frac{\sigma^2}{E} a_{\text{th}}. \tag{517}$$

One guess would be that such events, denoted henceforth as *outage*, would contribute significantly to errors (for appropriately chosen value of a_{th}).

A Suboptimal Receiver

Now consider a receiver that operates in a simple manner thus:

- in the event of an outage (i.e., the event in Equation (517) is satisfied), the receiver just gives up and refuses to make a decision based on the aposteriori probabilities. Thus the unreliability level of decoding the single bit correctly (based on the apriori probabilities) is 0.5.

- when not in outage (i.e., the value of $|h|$ is larger than the desired threshold), the receiver is the regular ML receiver. In this case, the unreliability level $Q\left(\sqrt{2|h|^2\mathsf{SNR}}\right)$ (cf. Equation (10) in Lecture 19) is at least as small as $Q\left(\sqrt{2a_{\text{th}}}\right)$. If a_{th} is chosen larger enough (say, 18) then the unreliability level is very small (say, 10^{-10}).

This simple receiver based on a modification of the optimal ML receiver is suboptimal in general. The exact probability of making an error with this receiver is:

$$\begin{aligned} P_e &= \mathbb{P}\left[\text{Outage}\right] \cdot 0.5 + \mathbb{P}\left[\text{NoOutage}\right] \cdot Q\left(\sqrt{2a_{\text{th}}}\right) && (518) \\ &\approx \mathbb{P}\left[\text{Outage}\right] \cdot 0.5. && (519) \end{aligned}$$

This is because we have supposed that the threshold a_{th} is large enough that errors when outage does not occur are very unlikely. How likely is the outage event? Of course, this would depend on the specific statistics of $|h|^2$. For the Rayleigh fading model, the statistics of $|h|^2$ is exponential (with mean A) and

$$\mathbb{P}\left[|h|^2 < \frac{a_{\mathrm{th}}}{\mathsf{SNR}}\right] = \int_0^{\frac{a_{\mathrm{th}}}{\mathsf{SNR}}} \frac{1}{A} e^{-\frac{a}{A}}\, \mathrm{d}a \tag{520}$$

$$= 1 - e^{-\frac{a_{\mathrm{th}}}{A\mathsf{SNR}}} \tag{521}$$

$$\approx \frac{a_{\mathrm{th}}}{A\mathsf{SNR}}. \tag{522}$$

With $a_{\mathrm{th}} = 18$ we see that the probability of outage is approximately

$$\mathbb{P}\left[\mathrm{Outage}\right] \approx \frac{18}{A\mathsf{SNR}}. \tag{523}$$

Based on the calculations so far, a few observations are in order now.

- The error probability of the receiver with the simple modification based on the outage event is dominated by the probability of the outage event. This is more or less by design, since we defined the outage event as one where errors are most likely.

- The probability of outage (cf. Equation (523)) while larger than the expression for the true error probability (cf. Equation (512)), has the *same* qualitative behavior with SNR: i.e., it decays inversely with SNR. This is a remarkable happenstance and allows us to conclude that the outage is the *typical* error event. Essentially we have seen that the fading channel magnitude being small is the dominant way errors occur.

We are now in a position to better evaluate our options to improve the behavior of unreliability level with SNR: we just need to reduce the chance of being in outage. Indeed, we will take this approach in the next several lectures. Before we get to this, it is worth taking a more fundamental look at our approach so far. We have made the observations regarding outage being the typical error event based on the assumption of sequential communication. We know from our earlier discussion (lectures 4 through 8) that block communication yields significantly better performance than that of sequential communication. It is worth redoing our outage analysis in the context of the fundamentally superior block coding context. This is the next topic.

Outage and Block Communication

In an earlier lecture we have derived the *capacity* of the (real) AWGN channel. What is the capacity of the *complex* based band AWGN channel:

$$y[m] = hx[m] + w[m], \tag{524}$$

i.e., the slow fading wireless channel if the complex channel gain h were known to the communication engineer who is designing the encoder and decoder rules? Here we have the usual power constraint (slightly modified to suit the complex transmit symbol):

$$\sum_{m=1}^{N} |x[m]|^2 \leq NP. \tag{525}$$

There is a simple way to convert the complex base band AWGN channel into the more familiar (real) AWGN channel: consider the invertible transformation

$$\begin{aligned} \tilde{y}[m] &= \frac{h^*}{|h|} y[m] & (526) \\ &= |h| x[m] + \tilde{w}[m]. & (527) \end{aligned}$$

Here $\tilde{w}[m]$ has the same statistics as that of w (this is similar to the statement made in our discussion on OFDM): i.e., it is a complex random variable with both real and imaginary parts independent and Gaussian with zero mean and variance $\frac{\sigma^2}{2}$. The complex AWGN channel in Equation (527) can be expanded out as

$$\begin{aligned} \Re\left[\tilde{y}[m]\right] &= |h| \Re\left[x[m]\right] + \Re\left[\tilde{w}[m]\right], & (528) \\ \Im\left[\tilde{y}[m]\right] &= |h| \Im\left[x[m]\right] + \Im\left[\tilde{w}[m]\right]. & (529) \end{aligned}$$

This is now a regular (real) AWGN channel where we get to transmit and receive two symbols for every time instant. The power constraint in Equation (525) means that the power per symbol in this (real) AWGN channel is $P/2$. The capacity of such a channel is simply *twice* that possible over the (real) AWGN channel with additive noise variance $\sigma^2/2$ and the largest rate of reliable communication over such a channel would be:

$$C = \log_2 \left(1 + \frac{|h|^2 P}{\sigma^2} \right) \text{bits/sample.} \tag{530}$$

Physically, one can understand the transmission of two symbols per time sample as representing the two signals transmitted in quadrature over the passband channel.

Here again, the capacity of the channel has the same interpretation as that for the (real) AWGN channel: i.e., arbitrarily reliable communication is possible at data rates below capacity and communication is arbitrarily unreliable at data rates above capacity. This is the situation in the complex AWGN case. But in the slow fading wireless channel, the communication engineer is not privy to the exact value of h and hence the corresponding *received* SNR of the channel. This is tantamount to fixing the data rate to R bits/sample ahead of time and using good AWGN encoding and decoding procedures. Depending on the dynamic channel quality we have one of two possible outcomes:

1. If $R < C$, then communication is arbitrarily reliable.

2. If $R > C$, then communication is arbitrarily unreliable.

The overall probability of error with block coding therefore is:

$$\begin{aligned} P_e &= \mathbb{P}\left[R > C\right] \cdot 1 + \mathbb{P}\left[R < C\right] \cdot 0 \quad (531) \\ &= \mathbb{P}\left[R > \log_2\left(1 + |h|^2 SNR\right)\right]. \quad (532) \end{aligned}$$

The goal of block communication was to avoid the deleterious effects of additive noise w and, indeed, we see that the error probability now depends only on the channel magnitude h. We can explicitly calculate the probability of error with rate efficient block communication for the Rayleigh fading model: In this case

$$\begin{aligned} P_e &= 1 - e^{-\frac{2^R - 1}{A\mathsf{SNR}}} \quad (533) \\ &\approx \frac{2^R - 1}{A\mathsf{SNR}}. \quad (534) \end{aligned}$$

Once again, a few observations are in order.

1. Even in the context of rate efficient block communication, the typical error event is the same as that defined earlier: outage. The only change is that the threshold a_{th} changed (from a somewhat arbitrary setting of 18 to $\left(2^R - 2\right)/\mathsf{SNR}$).

2. Comparing Equation (534) with Equation (523), we see that even with rate efficient block communication, the behavior of unreliability with SNR is the same as before: it decays linearly with SNR.

So now we can safely conclude that outage . Hence, the whole analysis done in this lecture leads to the final conclusion: the only way to improve error performance over the slow fading wireless channel is to somehow reduce the chance of an outage event!

Looking Ahead

We conclude that outage is a *fundamentally typical* error event. To improve the overall unreliability level (for a given SNR) the only way is to reduce the chance of outage (for the same SNR). We investigate several different ways to do this starting the next lecture.

Lecture 22: Time Diversity

Introduction

We have seen that the communication (even with coding) over a slow fading flat wireless channel has very poor reliability. This is because there is a significant probability that the channel is in outage and this event dominates the total error event (which also include the effect of additive noise). The key to improving the reliability of reception (and this is really required for wireless systems to work as desired in practice) is to reduce the chance that the channel is in outage. This is done by communicating over "different" channels and the name of the game is to harness the *diversity* available in the wireless channel. There are three main types: temporal, frequency and antennas. We will see each one over separate lectures, starting with time diversity.

Time Diversity Channel

The temporal diversity occurs in a wireless channel due to mobility. The idea is to code and communicate over the order of time over which the channel changes (called the *coherence* time). This means that we need enough delay tolerance by the data being communicated. A simple single-tap model that captures time diversity is the following:

$$y_\ell = h_\ell x_\ell + w_\ell, \quad \ell = 1, \dots, L. \tag{535}$$

Here the index ℓ represents a single time sample over *different* coherence time intervals. By interleaving across different coherence time intervals, we can get to the time diversity channel in Equation (535). A good statistical model for the channel coefficients $h_1, \dots, h_L$ is that they are all independent and identically distributed. Further more, in a environment with lots of multipath we can suppose that they are complex Gaussian random variables (the so-called Rayleigh fading).

A Single Bit Over a Time Diversity Channel

For ease of notation, we start out with $L = 2$ and just a single bit to transmit. The simplest way to do this is to *repeat* the same symbol: i.e., we set

$$x_1 = x_2 = x \pm \sqrt{E}. \tag{536}$$

At the receiver, we have *four* (real) voltages:

$$\Re[y_1] = \Re[h_1]x + \Re[w_1], \tag{537}$$

$$\Im[y_1] = \Im[h_1]x + \Im[w_1], \tag{538}$$

$$\Re[y_2] = \Re[h_2]x + \Re[w_2], \tag{539}$$

$$\Im[y_2] = \Im[h_2]x + \Im[w_2]. \tag{540}$$

As usual, we suppose *coherent* reception, i.e., the receiver has full knowledge of the exact channel coefficients h_1, h_2. By now, it is quite clear that the receiver might as well take the appropriate weighted linear combination to generate a single real voltage and then make the decision with respect to the single transmit voltage x. This is the matched filter operation:

$$y^{\mathrm{MF}} = \Re\left[y_1\right]\Re\left[h_1\right] + \Im\left[y_1\right]\Im\left[h_1\right] + \Re\left[y_2\right]\Re\left[h_2\right] + \Im\left[y_2\right]\Im\left[h_2\right]. \tag{541}$$

Using the complex number notation,

$$y^{\mathrm{MF}} = \Re\left[h_1^* y_1\right] + \Re\left[h_2^* y_2\right], \tag{542}$$

$$= \left(|h_1|^2 + |h_2|^2\right) x + \tilde{w}. \tag{543}$$

Here $\tilde{w}$ is a real Gaussian random variable because it is the sum of four independent and identically distributed Gaussians, but they get scaled by the real and imaginary parts of h_1 and h_2. Therefore $\tilde{w}$ is zero mean and has variance:

$$\mathrm{Var}(\tilde{w}) = \frac{\sigma^2}{2}\left(|h_1|^2 + |h_2|^2\right). \tag{544}$$

The decision rule is now simply the nearest neighbor rule:

$$\text{decide } x = +\sqrt{E} \text{ if } y^{\mathrm{MF}} > 0, \tag{545}$$

and zero, otherwise.

Probability of Error

The *dynamic* error probability is now readily calculated (since the channel – cf. Equation (543) – is now of an AWGN type):

$$P_e^{\mathrm{dynamic}} = Q\left(\frac{\left(|h_1|^2 + |h_2|^2\right)\sqrt{E}}{\frac{\sigma}{\sqrt{2}}\left(\sqrt{|h_1|^2 + |h_2|^2}\right)}\right). \tag{546}$$

From the transmitter's perspective, it makes sense to calculate the *average* error probability, averaged over the statistics of the two channels h_1, h_2:

$$P_e = \mathbb{E}\left[Q\left(\sqrt{2\mathsf{SNR}\left(|\mathsf{h}_1|^2 + |\mathsf{h}_2|^2\right)}\right)\right]. \tag{547}$$

For the statistical model we suggested earlier (independent Rayleigh fading), this expression can be evaluated exactly:

$$P_e = \left(\frac{1-\mu}{2}\right)^2 (1+\mu) \tag{548}$$

where

$$\mu \stackrel{\text{def}}{=} \sqrt{\frac{A\mathsf{SNR}}{1 + A\mathsf{SNR}}}. \tag{549}$$

Much like we did in earlier lectures, we can try to simplify Equation (548) in order to better see the relationship between unreliability level and SNR. Specifically, at high SNR (i.e., $SNR \gg 1$), we have

$$\begin{aligned}(1-\mu) &\approx \frac{1}{2A\mathsf{SNR}} && (550)\\ (1+\mu) &\approx 2. && (551)\end{aligned}$$

Substituting this approximations into Equation (548), we conclude that

$$P_e = \frac{1}{8A^2\mathsf{SNR}^2}. \tag{552}$$

Diversity Gain

Equation (552) that captures the approximate relationship between transmit SNR and unreliability level is instructive: we see that a doubling in SNR reduces the unreliability by a *quarter* factor. This is significant improvement when compared to the earlier case of no diversity: in that case, a doubling of SNR only decreased P_e by a factor of half. The exponent of SNR in Equation (552) is referred to as '*diversity gain*, and in this case it is equal to 2.

Generalization

We have worked with $L = 2$ for simplicity. For a general L, we have the natural extension to repetition coding at the transmitter and matched filter reception (with nearest neighbor decoding) at the receiver. Equation (543) generalizes to

$$y^{\text{MF}} = \left(\sum_{\ell=1}^{L} |h_\ell|^2\right) x + \tilde{w}, \tag{553}$$

where $\tilde{w}$ is real Gaussian with zero mean and variance (cf. Equation (544))

$$\left(\sum_{\ell=1}^{L} |h_\ell|^2\right) \frac{\sigma^2}{2}. \tag{554}$$

Thus, Equation (547) generalizes to

$$P_e = \mathbb{E}\left[Q\left(\sqrt{2SNR\left(\sum_{\ell=1}^{L} |h_\ell|^2\right)}\right)\right]. \tag{555}$$

ECE 361: Fundamentals of Digital Communications

Lecture 23: Frequency Diversity

Introduction

In many instances, temporal diversity is either not available (in stationary scenarios) or cannot be efficiently harnessed due to the strict delay constraints of the data being communicated. In these instances, and as an added source of diversity, looking to the frequency domain is a natural option.

Frequency Diversity Channel

Frequency diversity occurs in channels where the multipaths are spread out far enough, relative to the sampling period, so that multiple copies of the same transmit symbol are received over *different* received samples. Basically, we want a multitap ISI channel response: the L-tap wireless channel

$$y[m] = \sum_{\ell=0}^{L-1} h_\ell x[m-\ell] + w[m], \quad m \geq 1 \tag{1}$$

is said to have L-fold frequency diversity. The diversity option comes about because the different channel taps $h_0, \ldots, h_{L-1}$ are the result of different multipath combinations and are appropriately modeled as statistically independent. Thus the same transmit symbol (say $x[m]$) gets received multiple times, each over a statistically independent channel, (in this case, at times $m, m+1, \ldots, m+L-1$).

A Single Bit Over a Frequency Diversity Channel

Suppose we have just a single bit to transmit. The simplest way to do this is to transmit the appropriate symbol $x = \pm\sqrt{E}$ and stay silent. : i.e., we set

$$x[1] = \pm\sqrt{E}, \tag{2}$$

$$x[m] = 0 \quad m > 1. \tag{3}$$

At the receiver, we have L (complex) voltages that all contain the same transmit symbol immersed in multiplicative and additive noises:

$$y[\ell+1] = h_\ell x + w[\ell+1], \quad \ell = 0, \ldots, L-1. \tag{4}$$

As before, we suppose coherent reception, i.e., the receiver has full knowledge (due to accurate channel tracking) of the channel coefficients $h_0, \ldots, h_{L-1}$. We see that the situation is entirely similar to that of the time diversity channel with repetition coding (as seen in the previous lecture). Then, the appropriate strategy at the receiver (as seen several times, including in the previous lecture on time diversity) is to match filter:

$$y^{\mathrm{MF}} \stackrel{\mathrm{def}}{=} \Re\left[\sum_{\ell=0}^{L+1} h_\ell^* y[\ell+1]\right] \tag{5}$$

$$= \left(\sum_{\ell=0}^{L-1} |h_\ell|^2\right) x + \tilde{w}. \tag{6}$$

Diversity Gain

Here $\tilde{w}$ is real Gaussian with zero mean and variance

$$\left(\sum_{\ell=1}^{L} |h_\ell|^2\right) \frac{\sigma^2}{2}. \tag{7}$$

Thus, the average error probability is generalizes to

$$P_e = \mathbb{E}\left[Q\left(\sqrt{2\mathsf{SNR}\left(\sum_{\ell=0}^{L-1} |h_\ell|^2\right)}\right)\right]. \tag{8}$$

Again, there is an exact expression for the unreliability level when the channel coefficients are independent Rayleigh distributed (just as in the time diversity case):

$$\left(\frac{(1-\mu)}{2}\right)^L \sum_{k=0}^{L-1} \begin{pmatrix} L-1+k \\ k \end{pmatrix} \left(\frac{1+\mu}{2}\right)^k, \tag{9}$$

where μ is as in the previous lecture. Finally, we can look for a high SNR approximation; as earlier, we have:

$$P_e \approx \begin{pmatrix} 2L-1 \\ L \end{pmatrix} \frac{1}{(4A\mathsf{SNR})^L}. \tag{10}$$

The diversity gain is now L: doubling of SNR reduces the unreliability by a factor of

$$\frac{1}{2^L}. \tag{11}$$

OFDM

The approach so far has been very simple: by being silent over $L-1$ successive symbols for every single symbol transmission we have converted the frequency diversity channel into a time diversity one with repetition coding. While this allowed for easy analysis (we could readily borrow from our earlier calculations) and easy harnessing of frequency diversity (so reliability really improved) it has a glaring drawback: we are only transmitting once every L symbols and this is a serious reduction in data rate. If we choose not to stay silent, then we have to deal with ISI while still trying to harness frequency diversity. A natural way to do this is to use OFDM. This way, we convert (at an OFDM block level) to the channel:

$$\tilde{y}_n = \tilde{h}_n \tilde{x}_n + \tilde{w}_n, \quad n = 0, \ldots, N_c - 1. \tag{12}$$

This looks very much like the time diversity channel, except that the index n represents the "tones" or "sub-channels" of OFDM.

It is interesting to ask what diversity gain is possible from this channel. Well, since the channel really is the one we started out with (cf. Equation (1)), no more than L-order frequency diversity gain should be possible (there are only L independent channels!). However, a cursory look at Equation (12) suggests that there might be N_c order diversity gain, one for each sub-channel. Clearly this cannot be correct since N_c is typically much larger than L in OFDM.

The catch is that the OFDM channel coefficients $\tilde{h}_0, \ldots, \tilde{h}_{N_c-1}$ are *not* statistically independent. They are the discrete Fourier transform output of the input vector

$$[h_0, \ldots, h_{L-1}, 0, 0, \ldots, 0]. \tag{13}$$

Thus there are only about L "truly" independent OFDM channel coefficients and the diversity gain is restricted. Roughly speaking, once in every N_c/L tones fades independently. Since the "bandwidth" of each tone is about $\frac{1}{W}$ (where W is the total bandwidth of communication) we can say that the random discrete time frequency response $H(f)$ is independent over frequencies that are apart by about W/L. This separation of frequency over which the channel response is statistically independent from one another is known as the *coherence bandwidth* (analogous to the coherence time we have seen earlier).

The coherence bandwidth restricts the total amount of frequency diversity available to harness. Physically, this is restricted by the delay spread and communication bandwidth (which decides the sampling rate). Most practical wireless systems harness frequency diversity by a combination of having wide enough bandwidth of communication or a narrowband of communication but one that is hopped over different carrier frequencies.

Looking Ahead

In the next lecture we will see yet another form of diversity: spatial. It is harnessed by having antennas, at both transmitter and receiver.

ECE 361: Fundamentals of Digital Communications

Lecture 24: Antenna Diversity

Introduction

Time diversity comes about from mobility, which may not always be present (at least over the time scale of the application data being communicated). Frequency diversity comes about from good delay spread (relative to the bandwidth of communication) and may not always be available. Fortunately, there is another mode of diversity, *spatial*, that can be harnessed even under stationary scenarios with narrow delay spread. Antennas, transmit and/or receive, are a natural element to capture spatial diversity. Antenna diversity is a "form of time diversity" because instead of a single antenna moving in time we have multiple antennas capturing the same mobility by being in different points in space. This means that antenna *and* time diversity do not really occur independent of each other (while time and frequency diversity occur quite independent of each other).

Antennas come in two forms: receive and transmit. Both forms provide diversity gain, but the nature of harnessing them are quite different. Getting a better understanding of how this is done is the focus of this lecture. We will also get to see the differences between transmit and receive antennas. To be able to focus on the new aspect antennas bring into the picture, our channel models feature *no* time and frequency diversity.

Receive Antenna Channel

The single transmit and multiple receive antenna channel is the following (with no time and frequency diversity):

$$y_\ell[m] = h_\ell x[m] + w_\ell[m], \quad \ell = 1, \ldots, L, \quad m \geq 1. \tag{1}$$

Here L is the number of received antennas. Coherent reception is supposed, as usual. We observe that the channel in Equation (1) is identical to the transmit diversity one *with* repetition coding. So, the best strategy for the receiver is to match filter the L voltages:

$$y^{\text{MF}}[m] \stackrel{\text{def}}{=} \Re\left[\sum_{\ell=1}^{L} h_\ell^* y_\ell[m]\right], \tag{2}$$

$$= \left(\sum_{\ell-1}^{L} |h_\ell|^2\right) x[m] + \tilde{w}[m]. \tag{3}$$

As earlier, $\tilde{w}[m]$ is real noise voltage with Gaussian statistics: zero mean and variance equal to

$$\left(\sum_{\ell-1}^{L} |h_\ell|^2\right) \frac{\sigma^2}{2}. \tag{4}$$

Even though the scenario looks eerily like time diversity with repetition coding (indeed, the calculations are rather identical), there are two key differences:

- the "repetition" was done by nature in the spatial dimension. So, no time was wasted in achieving the "repetition". This means that there is no corresponding loss in data rate.

- since there was no actual repetition at the transmitter, there is none of the corresponding loss of transmit power. In other words, we receive more energy than we did in the single antenna setting. Of course, this model makes sense only when we have a small number of antennas and the energy received per antenna is a small fraction of the total transmit energy. Indeed, this model will break down if we have a huge number of antennas: surely, we cannot receive more signal energy than transmitted.

If the antennas are spaced reasonably apart (say about a wavelength distance from each other), then a rich multipath environment will result in statistically independent channel coefficients $h_1, \ldots, h_L$. This means that we have the L fold diversity gain as earlier. It is important to bear in mind that this diversity gain is in *addition* to the L fold *power* gain that was obtained due to the L receive antennas (these resulted in an L fold increase in receive energy, as compared with the single antenna situation).

To summarize:

> receive antennas are just an unalloyed good. They provide both power and diversity gains.

Transmit Antenna Channel

The multiple transmit, single receive antenna channel (with no time and frequency diversity) is modeled as:

$$y[m] = \sum_{\ell=1}^{L} h_\ell x_\ell[m] + w[m], \quad m \geq 1. \tag{5}$$

In other words, L distinct transmit (complex) voltages result in a single receive (complex) voltage. Again, if the antennas are spaced far enough apart (say a distance of one wavelength) in a good multipath environment then an appropriate statistical model would be to consider $h_1, \ldots, h_L$ to be statistically independent (and complex Gaussian distributed).

How does one harness the transmit diversity gain? Naive repetition coding, transmitting the same symbol over each antenna, is *not* going to work. To see this, suppose

$$x_\ell[m] = \frac{x[m]}{\sqrt{L}}, \quad \ell = 1, \ldots, L. \tag{6}$$

Here we have normalized by L to have the same total transmit power as if we had a single transmit antenna. Then the received signal is

$$y[m] = \left(\sum_{\ell=1}^{L} \frac{h_\ell}{\sqrt{L}} \right) x[m] + w[m]. \tag{7}$$

The "effective" channel

$$\left(\sum_{\ell=1}^{L} \frac{h_\ell}{\sqrt{L}} \right) \tag{8}$$

is also Rayleigh fading: a sum of independent complex Gaussian random variables is still a complex Gaussian random variable!

But a small twist will serve to fix the problem: we can repeat over the antennas, but use *only one at a time*. Consider the following strategy:

$$x_\ell[m] = \begin{cases} x & \text{if } \ell = m \\ 0 & \text{else.} \end{cases} \tag{9}$$

Then the first L received symbols are

$$y[\ell] = h_\ell x + w[\ell], \quad \ell = 1, \ldots, L. \tag{10}$$

This scenario is now identical to that of the receive antenna channel (cf. Equation (1)). So, the diversity gain is readily harnessed (by matched filtering at the receiver). But two important differences are worth highlighting:

- there is no power gain from transmit antennas (this is because we actually transmitted L times more energy than in the receive antenna scenario);
- there is a loss in data rate by a factor of $1/L$ (this is because we actually used the transmit one antenna at a time).

Modern communication techniques offer ways of using all the transmit antennas simultaneously while still allowing the harnessing of transmit diversity. This is the area of *space time coding*. While this is a bit beyond the scope of this lecture, we can point out a simple scheme that harnesses transmit diversity without sacrificing the data rate entailed by using one antenna at a time. Consider the following transmission scheme:

$$x_\ell[m] = x_{\ell-1}[m-1], \quad \ell > 1, \quad m > 2. \tag{11}$$

With this transmission technique, the received signal is

$$y[m] = \sum_{\ell=1}^{L} h_\ell x[m - \ell - 1] + w[m]. \tag{12}$$

In other words we have converted the transmit antenna channel into one with frequency diversity. Now a technique such as OFDM will allow the harnessing of the available diversity gain. This technique, known as *delay diversity*, was one of the early space time coding techniques.

Looking Ahead

All wireless systems harness one or more forms of diversity. This is the key to improving the reliability of wireless communication. In the next (and final) lecture, we will see the key new challenges (beyond those that arise in reliable point to point wireless communication) in building a wireless *system* where there are many users communication at the same time. We study these challenges in the context of the cellular wireless system.