ECE 461: Digital Communication

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 promised 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

$$y[m] = x[m] + w[m],$$
 (1)

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 sequenctial 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 (29) of Lecture 3,

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

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

$$(1-p)^n. (3)$$

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 2, how much should the SNR be?). Then the reliability of the entire sequential communication process is, from Equation 3,

$$(1-10^{-4})^{250} \approx 0.75.$$
 (4)

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 (this is explored in a homework exercise).

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

$$y[m] = x + w[m], \quad m = 1 \dots n.$$
 (5)

How should the receiver make its decision on what (common) voltage level was transmitted? In other words, how 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: $\mathbf{v}_1, \ldots, \mathbf{v}_{2^B}$. 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 $a(1), \ldots, a(n)$ are received at the *n* time instants. Directly from Equation (11) of Lecture 3, we see that the likelihood for the k^{th} transmit voltage vector is (up to a scaling constant)

$$L_k = f_{\mathbf{y}}(\mathbf{a}|\mathbf{x} = \mathbf{v}_k) \tag{6}$$

$$= \prod_{m=1}^{n} f_{y_m} \left(a(m) | x_m = \mathbf{v}_k(m) \right)$$

$$\tag{7}$$

$$= \prod_{m=1}^{n} f_{w_m} \left(a(m) - \mathbf{v}_k(m) \right).$$
(8)

Here we used the statistical independence of the additive noises at the different time instants in arriving at Equation (7). Using the explicit PDF of the Gaussian noise statistics, we can simplify the likelihood even further:

$$L_k = \left(\frac{1}{(2\pi\sigma)^{\frac{n}{2}}}\right) \exp\left(-\frac{1}{2\sigma^2} \left(\sum_{m=1}^n |a(m) - \mathbf{v}_k(m)|^2\right)\right).$$
(9)

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

$$\|\mathbf{a} - \mathbf{v}_k\|^2 \stackrel{\text{def}}{=} \sum_{m=1}^n |a(m) - \mathbf{v}_k(m)|^2, \quad k = 1, \dots, 2^B.$$
(10)

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 1 illustrates the constellation diagram and the nearest-neighbor ML rule for the special case of B = n = 2.



Figure 1: 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

$$\mathbf{v}_k = v_k \mathbf{1},\tag{11}$$

where **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 **a** and **v**_k is

$$\|\mathbf{a} - \mathbf{v}_k\|^2 = \sum_{m=1}^n (a(m) - v_k)^2$$
(12)

$$= nv_k^2 + \left(\sum_{m=1}^n a(m)^2\right) + -2v_k \left(\sum_{m=1}^n a(m)\right).$$
(13)

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

$$nv_k^2 - 2v_k\left(\sum_{m=1}^n a(m)\right), \quad k = 1\dots 2^B.$$
 (14)

A Sufficient Statistic

We observe from Equation (14) that the sum of the received voltages

$$\left(\sum_{m=1}^{n} a(m)\right) \tag{15}$$

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

$$\sum_{m=1}^{n} y[m] = nx + \sum_{m=1}^{n} w[m].$$
(16)

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

$$\tilde{y} \stackrel{\text{def}}{=} \frac{1}{n} \sum_{m=1}^{n} y[m] \tag{17}$$

$$= x + \tilde{w}, \tag{18}$$

where

$$\tilde{w} \stackrel{\text{def}}{=} \frac{1}{n} \sum_{m=1}^{n} w[m] \tag{19}$$

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 (2) (replacing σ^2 by $\frac{\sigma^2}{n}$ and k by B),

$$p^{\text{rep}} \stackrel{\text{def}}{=} \left(2 - \frac{1}{2^B - 1}\right) Q\left(\frac{\sqrt{n\mathsf{SNR}}}{2^B - 1}\right). \tag{20}$$

The rate of communication is

$$R^{\rm rep} \stackrel{\rm def}{=} \frac{B}{n} \tag{21}$$

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

1. The error probability increases to 1 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

$$\lim_{n \to \infty} \frac{B}{\log n} = 0, \tag{22}$$

then

$$\lim_{n \to \infty} \frac{\sqrt{n\mathsf{SNR}}}{2^B - 1} = \infty.$$
(23)

Substituting this in Equation (20) and using the fact that

$$\lim_{a \to \infty} Q(a) = 0, \tag{24}$$

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

$$\lim_{n \to \infty} p^{\text{rep}} \to 0.$$
 (25)

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

$$\frac{nE}{B},\tag{26}$$

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 efficiency.

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.