University of Illinois **Spring 2011**

ECE 361: Lecture 6: The Discrete-Time Gaussian Channel

6.1. Sequential Transmission of Data

In the previous Lecture Notes, we considered the transmission of one bit over an additive white Gaussian noise channel and then generalized the idea to transmitting multiple bits at the same time via M-ary Amplitude Shift Keying (also called Pulse Amplitude Modulation or PAM). The decision as to which of the M messages was being sent was based on the observation $\mathbb{Y}$ which we modeled as a Gaussian random variable whose variance was a constant ($N_0/2$, the value of the two-sided power spectral density of the noise) while the mean of $\mathbb{Y}$ depended on which of the messages was being transmitted. We placed no restrictions on the *duration* of the signals being transmitted or received. In practice, of course, the signals are of finite duration, say T seconds, and the communication system transmits $\log_2 M$ bits of data with M-ASK every T seconds by using time-delayed versions of the same signals. We discuss this notion in more detail.

Suppose that we are using M-ASK with basic signal $\psi(t)$, and that $\psi(t) = 0$ for all $t \leq 0$ and also for all $t \geq T$. Choosing the sampling time T_0 to be T or more, we can be guaranteed that $\psi(T_0 - t) = 0$ for $t < 0$, that is, the matched filter (whose impulse response is $h(t) = \psi(T_0 - t)$) is a *causal* filter. Note also that $\psi * h$ is nonzero only for $t \in (T_0 - T, T_0 + T)$, that is, the response of the matched filter to $\psi(t)$ is of duration $2T$ lasting from $t = T_0 - T > 0$ to $T_0 + T$. Now consider the transmission of *two* messages over $2T$ seconds. Specifically, suppose that the signal $K\sqrt{\mathcal{E}}\psi(t)$ is transmitted during $(0, T]$ and the signal $L\sqrt{\mathcal{E}}\psi(t-T)$ during $(T, 2T]$, that is, the signal $K\sqrt{\mathcal{E}}\psi(t) + L\sqrt{\mathcal{E}}\psi(t-T)$ of duration $2T$ is transmitted. Our matched filter is matched to $\psi(t)$ for a sampling time of $t = T_0$ and thus produces a maximum response $K\sqrt{\mathcal{E}}$ at $t = T_0$. The response of the filter dies away and is 0 for $t \geq T_0 + T$. But what is the matched filter for signal $\hat{\psi}(t) = \psi(t-T)$ for sampling time $T_0 + T$? It is $\hat{\psi}(T_0 + T - t) = \psi(T_0 + T - t - T) = \psi(T_0 - t) = h(t)$! That is, the *same* matched filter works with both signals $\psi(t)$ and $\psi(t-T)$, being matched to $\psi(t)$ for a sampling time on T_0, and to $\psi(t-T)$ for a sampling time of $T_0 + T$. Furthermore, the response of the matched filter to $\psi(t-T)$ is nonzero only for $t \in (T_0, T_0 + 2T)$, which overlaps with the response to $\psi(t)$ but does not affect the value obtained by sampling at T_0. From linearity and superposition, we conclude that when the input $K\sqrt{\mathcal{E}}\psi(t) + L\sqrt{\mathcal{E}}\psi(t-T)$ is applied to the matched filter, the response is $K\sqrt{\mathcal{E}}$ at $t = T_0$ and $L\sqrt{\mathcal{E}}$ at $t = T_0 + T$.

All the above generalizes in a straightforward manner. For each integer i, the filter with impulse response $h(t) = \psi(T_0 - t)$ is matched to $\psi(t - iT)$ for a sampling time of $T_0 + iT$. Furthermore, the response of the matched filter to $\psi(t - iT)$ lasts from $(T_0 + iT) - T = T_0 + (i-1)T$ to $(T_0 + iT) + T = T_0 + (i+1)T$. Thus, the samples taken at $t = T_0 + (i-1)T$ and $t = T_0 + (i+1)T$ are not affected by this response. The signal

$$\sum_i K_i \sqrt{\mathcal{E}}\psi(t - iT)$$

thus can be used to transmit a *sequence* of signal levels that $(\ldots, K_{i-1}, K_i, K_{i+1}, \ldots)$. Since $h(t)$ is matched to $\psi(t - iT)$ for a sampling time of $T_0 + iT$, the sequence of signal sample values at the receiver is $(\ldots, K_{i-1}\sqrt{\mathcal{E}}, K_i\sqrt{\mathcal{E}}, K_{i+1}\sqrt{\mathcal{E}}, \ldots)$. Of course, each actual sample consists of the signal sample value plus noise, and the noise random variables are *independent* random variables. This last follows from a property of white Gaussian noise that we have not discussed before. Suppose that white Gaussian noise is the input to a filter with impulse response $h(t)$, and that we take noise samples from the filter output at times t_1 and t_2. Then these samples are jointly Gaussian random variables with *covariance* $\frac{N_0}{2}R_h(t_1 - t_2)$ where $R_h(\tau)$ is the *autocorrelation function* of $h(t)$. Note that for $t_1 = t_2$, the covariance becomes the variance which is thus $\frac{N_0}{2}R_h(0) = \frac{N_0}{2}\int_{-\infty}^{\infty} |h(t)|^2\,dt$. Now, in our problem, $h(t) = \psi(T_0 - t)$ is a unit-energy signal so that $R_h(0) = 1$ giving that the variance is $N_0/2$ as we have already seen, Furthermore, $h(t)$ is of duration T and so $R_h(\tau) = 0$ for $|\tau| \geq T$. In other words, two noise samples separated by T or more seconds are *uncorrelated* random variables, and since they are jointly Gaussian, they are *independent* Gaussian random variables. In the next section, we discuss a simple channel model that captures the essence of these ideas.

Let us summarize what we have thus far. Using M-ary ASK and pulses time-limited to a duration of T seconds or less, we can transmit data at the rate of $\log_2 M$ bits per T seconds with error probability

$$P(E) = 2\left(1 - \frac{1}{M}\right) Q\left(\sqrt{\frac{6\bar{\mathcal{E}}}{(M^2-1)N_0}}\right) = 2\left(1 - \frac{1}{M}\right) Q\left(\sqrt{\frac{2\mathcal{E}_{\max}}{(M-1)^2 N_0}}\right) = 2\left(1 - \frac{1}{M}\right) Q\left(\frac{\sqrt{\text{SNR}}}{M-1}\right)$$

where we have written SNR for $2\mathcal{E}_{\max}/N_0 = \mathcal{E}_{\max}/(N_0/2) = \mathcal{E}_{\max}/\sigma^2$ since $N_0/2$ is the noise variance. The smallest error probability $Q(\sqrt{2\mathcal{E}_{\max}/N_0}) = Q(\sqrt{SNR})$ (but also the smallest data rate $1/T$ bits per second) is obtained when $M = 2$. Unfortunately, this is often not small enough, and the only way to reduce the error probability further is to engineer the system so as to increase SNR which in almost all cases means increasing the received energy $\mathcal{E}_{\max}$ by hook or by crook. In most practical situations, such engineering is practically or commercially infeasible. But *why* do we need incredibly small bit error probabilities? Let us restrict ourselves to 2-ASK with bit error probability $p = Q(\sqrt{SNR})$. Very often, the user of the communication system is not concerned as much with bit error probability as with the probability of being able to transmit a *packet* – a block of data – with very high reliability, say 99.9%. Thus, typically one packet in a thousand is not received correctly, and has to be re-transmitted. If the packet has, say, 1500 bits, then the probability that all 1500 bits are received correctly is $(1-p)^{1500}$ which for $p = 10^{-4}$ is just a little over 86%. Indeed, to get 99.9% reliable transmission of the packet would require p to be less than 10^{-7} and thus require a much larger SNR. The solution to achieving highly reliable communication of large blocks of data (which of course means even more highly reliable communication of the individual bits) is to use a technique called *coding.* In contrast to sequential communication in which each successive transmitted pulse carries one individual bit (or one ($\log_2 M$)-bit byte) of data, coding uses n successive pulses to *jointly* transmit a total of k bits of data (where $k < n$). The difference between coded communication and sequential communication is that the receiver cannot make a decision on any individual data bit (or byte) upon receipt of just one pulse; the k data bits are *jointly encoded* into (that is, together determine) the amplitudes of *all* n pulses, and the decision as to which of the 2^k possible bit patterns was transmitted is made only after all n pulses have been received. There are several features of coding schemes that should be remembered.

- The data rate is reduced since we are able to transmit only k bits of data with n pulses where $n > k$. Instead of transmitting a minimum of one bit per pulse (one bit every T seconds if you prefer), we can average only $k/n < 1$ bit per pulse. The quantity k/n is called the *rate* R of the coding scheme.

- A coded data transmission scheme has additional delays built into it as compared to sequential transmission. The transmitter must first collect and save k data bits. Only after k data bits are provided to the transmitter can the transmitter proceed to determine the n pulse amplitudes. Thus, the *data source* is sending bits to the transmitter at a rate of k bits in nT seconds (that is, with a clock rate $= k/nT = R/T < 1/T$ Hz). The receiver must wait for all n pulses to be received. Thus, ignoring any delays in the channel and assuming the receiver makes an instantaneous decision, at least $2nT$ seconds elapse between the first data bit being available at the transmitter and the decision being made. If the receiver sends the data bits to their destination on a serial line, then it takes another nT seconds to read the data out of the receiver and into the destination.

- The reason that most communications engineers are willing to pay the price of a reduced data rate and an increased delay is that for almost any given SNR, the probability that the entire set of k bits is received without error can be made as close to 1 as desired by using a suitable coding scheme, and furthermore, the cost of such a more complicated scheme is often far less than the cost of sticking with sequential communication and increasing the received energy so as to achieve the low bit error probability needed for reliable block communication.

In order to discuss such matters more concisely, in the next section we consider a simplified model of communication systems that gets away once again from the matched filters etc. that we have discussed previously, and concentrates attention on the key parameters that determine the performance of communication systems.

6.2. The Discrete-Time Gaussian Channel

The discrete-time Gaussian channel is a simplified model of the communication systems and matched filters etc. that have been discussed previously. The input to the channel corresponds to the amplitude of the transmitted pulse (that is, the transmitted signal), and we denote it as $\mathbb{X}$ which we take to be a discrete random variable taking on a finite set of values.[1] The output of the channel is denoted $\mathbb{Y}$ and it corresponds to the output of the matched filter described previously. Thus, $\mathbb{Y} = \mathbb{X} + \mathbb{N}$ where $\mathbb{N} \sim \mathcal{N}(0, \sigma^2)$ is a Gaussian random variable that is independent of $\mathbb{X}$. Note that $\mathbb{Y}$ is a continuous random variable whose *conditional* pdf given that $\mathbb{X} = \alpha$ is a Gaussian pdf with mean α and variance σ^2. The law of total probability tells us that that the *unconditional* pdf of $\mathbb{Y}$ is a *weighted sum* of these Gaussian pdfs, that is, $\mathbb{Y}$ is not a Gaussian random variable. Fortunately, we will need to consider only the conditional pdfs of $\mathbb{Y}$ given the value of the channel input, and all these conditional pdfs are Gaussian pdfs.

Since the channel will be used multiple times, we use subscript i to denote the random variables occurring on the i-th use of the channel. We write the output of the channel as $\mathbb{Y}_i = \mathbb{X}_i + \mathbb{N}_i$ where the N_i's are taken to be *independent* $\mathcal{N}(0, \sigma^2)$ random variables for the reasons described in the previous section. The $\mathbb{N}_i$'s are also assumed to be independent of all the $\mathbb{X}_i$'s too: the noise does not depend on the transmitted signal.

Additional features of the discrete-time Gaussian channel model are under the control of the communications engineer. If M-ASK is the modulation being considered, then each $\mathbb{X}_i$ is a discrete random variable uniformly distributed on the set $\{-(M-1)\sqrt{\mathcal{E}}, -(M-3)\sqrt{\mathcal{E}}, \ldots, (M-3)\sqrt{\mathcal{E}}, (M-1)\sqrt{\mathcal{E}}\}$. If, in addition, we are using a sequential communication scheme, then the $\mathbb{X}_i$ are also independent random variables because each is determined by a different $(\log_2 M)$-bit data byte. But what about the model for a coded communication scheme in which k bits are encoded into the amplitudes of n pulses? Suppose that pulses numbered 0 through $n-1$ are used to convey the first set of k bits. Then, since the amplitudes of all the n pulses are determined by all the k bits, it is no longer appropriate to assume that $\mathbb{X}_0, \mathbb{X}_1, \ldots, \mathbb{X}_{n-1}$ are independent random variables; they are indeed very much dependent. In fact, it is not even necessary to insist that $\mathbb{X}_0, \mathbb{X}_1, \ldots, \mathbb{X}_{n-1}$ all have the same pmfs, or even necessarily take on the same M values though it should be remembered that such great generality also comes at great expense. Not only must the receiver remember all the details, but it must adjust its thresholds etc. at each of the sampling instants so as to optimally detect which value is being transmitted.[2] More often than not, it is best to stick with the same pmf for all times, though every now and then we shall encounter the exception that proves the rule. Fortunately, the saving grace in all this is that, regardless of any assumptions about the independence or dependence of the $\mathbb{X}_i$, we have that *conditioned* on $(\mathbb{X}_0 = \alpha_0, \mathbb{X}_1 = \alpha_1, \ldots, \mathbb{X}_{n-1} = \alpha_{n-1})$, the random variables $\mathbb{Y}_i, 0 \leq i \leq n-1$ are *conditionally independent* Gaussian random variables with means $\alpha_i, 0 \leq i \leq n-1$ respectively, and with common variance σ^2.

Finally, we consider the inclusion of energy constraints in our model for the channel inputs $\mathbb{X}_i$. We have thus far considered a maximum energy constraint which we can include by insisting that the values of the $\mathbb{X}_i$'s be bounded: $\boxed{\max |\mathbb{X}_i|^2 \leq \mathcal{E}_{\max} \text{ for all } i.}$ In the context of Gaussian channel models, this is often referred to as a *peak power constraint* because the energy is being transmitted over some time interval (T seconds in Section 6.1), and thus the maximum power (remember that power = energy/time?) is limited. Notice that a peak power constraint implies an average power constraint since if $\max |\mathbb{X}_i|^2 = \mathcal{E}_{\max}$, then $\mathsf{E}[\mathbb{X}_i^2] \leq \mathcal{E}_{\max}$, with equality exactly when $\mathbb{X}_i$ takes on values $\pm\sqrt{\mathcal{E}_{\max}}$ with equal probability $\frac{1}{2}$. A different and weaker constraint is the *average power constraint* which is sometimes expressed as $\boxed{\mathsf{E}[\mathbb{X}_i^2] \leq \bar{\mathcal{E}} \text{ for all } i.}$ Notice that an average power constraint does not imply a peak power constraint. For example, a random variable $\mathbb{X}$ that has values $\pm\sqrt{\bar{\mathcal{E}}/2p}$ each with probability p, and value 0 with probability $1-2p$ satisfies $\mathsf{E}[\mathbb{X}^2] = \bar{\mathcal{E}}$ but $\max |\mathbb{X}|^2 = \bar{\mathcal{E}}/2p$ can be arbitrarily large depending on how close p is to 0. Now, the statement that $\mathsf{E}[\mathbb{X}_i^2] \leq \bar{\mathcal{E}}$ for all i is perfectly satisfactory when all the $\mathbb{X}_i$ have the same pmf, but since we are allowing the

[1] In the general theory, $\mathbb{X}$ can be a continuous random variable too, but we do not consider this general case.

[2] Note that if the pmfs of the $\mathbb{X}_i$'s are different, then they repeat periodically: for each integer ℓ, the joint pmf of $(\mathbb{X}_{\ell n}, \mathbb{X}_{\ell n+1}, \ldots, \mathbb{X}_{(\ell+1)n-1})$ is the same as the joint pmf of $(\mathbb{X}_0, \mathbb{X}_1, \ldots, \mathbb{X}_{n-1})$, and thus $\mathbb{X}_{\ell n+i}$ has the same pmf as $\mathbb{X}_i$.

$\mathbb{X}_i$ to have different pmfs, the average power constraint is really

$$\frac{1}{n}\sum_{i=0}^{n-1}\mathsf{E}[\mathbb{X}_i^2] \leq \bar{\mathcal{E}}$$

In words, the total energy received over n channel uses is no more than $n\bar{\mathcal{E}}$ but we don't care otherwise how the energy is divided up among the n pulses. Of course, both a peak power constraint and an average power constraint must be satisfied in a practical communication system. But, from purely a mathematical perspective, one can have an average power constraint but no peak power constraint, and so it is possible to transmit all $n\bar{\mathcal{E}}$ using $\mathbb{X}_0$ that takes on values $\pm\sqrt{n\bar{\mathcal{E}}}$ with equal probability $\frac{1}{2}$ while $\mathbb{X}_1 = \mathbb{X}_2 = \mathbb{X}_{n-1} = 0$ with probability 1. Thus, *shouting* out a bit value loudly in the very first transmission, and then shutting up for the next $n-1$ transmissions is perfectly allowable mathematical strategy; whether it would be at all feasible in practice is a different matter.

Remember in all this that n is a parameter that the system designer can choose, and that the distribution of the random variables $\mathbb{X}_0, \mathbb{X}_1, \ldots, \mathbb{X}_{n-1}$ is also something that the system designer can choose appropriately. The designer can choose to work under a peak power constraint, or just an average power constraint without any limitation on the peak power depending on the system that is being designed. All the channel itself does is add noise – an independent zero-mean Gaussian random variable with variance σ^2 to each transmission. What kinds of coding schemes might the designer use to make the bit error probability very small? This question will occupy us for some time and Lectures to come.

6.3. Repetition Coding

Suppose that we have a discrete-time Gaussian channel with a peak power constraint $\mathcal{E}$, that is $\max|\mathbb{X}_i|^2 \leq \mathcal{E}$ for all i. We can use 2-ASK, choosing $\mathbb{X}_i = \pm\sqrt{\mathcal{E}}$ with equal probability, to transmit one bit per channel use, and achieve bit error probability $Q(\sqrt{\mathcal{E}/\sigma^2}) \leq \frac{1}{2}\exp(-\mathcal{E}/2\sigma^2)$. Although the error probability decreases exponentially with $\mathcal{E}$, what if $\mathcal{E}$ is so small that the bit error probability is depressingly large? A very simple form of coding known as repetition coding is a possibility to consider.[3] Here, the *same* bit is transmitted n times. Thus,

$$(\mathbb{X}_0, \mathbb{X}_1, \ldots, \mathbb{X}_{n-1}) = \pm(\sqrt{\mathcal{E}}, \sqrt{\mathcal{E}}, \ldots, \sqrt{\mathcal{E}},)$$

and the receiver makes a decision based on n Gaussian random variables $\mathbb{Y}_0, \mathbb{Y}_1, \ldots, \mathbb{Y}_{n-1})$ all of which have the *same mean*, $\sqrt{\mathcal{E}}$ or $-\sqrt{\mathcal{E}}$, depending on whether a 0 or a 1 is being transmitted in these n channel uses. The decision-making is straightforward: remembering that all $\mathbb{Y}_i$'s have variance σ^2 and are conditionally independent given the transmitted bit, the likelihood ratio is the ratio of the joint pdfs and is given by

$$\lambda(\underline{v}) = \frac{f_1(\underline{v})}{f_0(\underline{v})} = \frac{\left(\frac{1}{\sqrt{2\pi}}\right)^n \exp\left[-\frac{1}{2\sigma^2}\sum_{i=0}^{n-1}(v_i+\sqrt{\mathcal{E}})^2\right]}{\left(\frac{1}{\sqrt{2\pi}}\right)^n \exp\left[-\frac{1}{2\sigma^2}\sum_{i=0}^{n-1}(v_i-\sqrt{\mathcal{E}})^2\right]} = \exp\left[-\frac{2\sqrt{\mathcal{E}}}{\sigma^2}\sum_{i=0}^{n-1}v_i\right].$$

The likelihood ratio exceeds 1 if and only if $\sum\mathbb{Y}_i < 0$. Thus, the receiver decides that a 0 was transmitted if $\sum\mathbb{Y}_i > 0$, and that a 1 was transmitted if $\sum\mathbb{Y}_i < 0$. Since $\sum\mathbb{Y}_i$ is a Gaussian random variable with mean $n\sqrt{\mathcal{E}}$ and variance $n\sigma^2$, the error probability is $Q(n\sqrt{\mathcal{E}}/\sqrt{n\sigma^2}) = Q(\sqrt{n\mathcal{E}/\sigma^2}) \ll Q(\sqrt{\mathcal{E}/\sigma^2})$. Thus, repetition coding effectively *combines* all the energy transmitted during the n channel uses and makes a decision that is very much more likely to be correct. Since the error probability decreases exponentially with increasing n, it can be made as small as desired, no matter how small the peak power constraint is. Unfortunately, this reduction in error probability comes at a heavy price: we are transmitting only one bit in n channel uses, and thus the good error probability is accompanied by very low rate. Asymtotically, we have that $P(E) = Q(\sqrt{n\mathcal{E}/\sigma^2}) \to 0$ and $R = n^{-1} \to 0$ as $n \to \infty$.

What if we used 2^B-ASK and transmitted the same B-bit symbol n times? Well, the $\mathbb{X}_i$ would take on 2^B equally spaced values in the interval $[-\sqrt{\mathcal{E}}, +\sqrt{\mathcal{E}}]$, and the $\mathbb{Y}_i$'s would all have the same mean, etc. The sum

[3] Repetition coding is a technique that is of ancient, if not divine, origin, *cf.* Matthew 5:37 in the New Testament (King James Authorized Translation of the Bible).

$\sum \mathbb{Y}_i$ would be compared to thresholds that are larger by a factor of n to make the decision, and the answer would be that

$$P(E) = 2\left(1 - \frac{1}{2^B}\right) Q\left(\sqrt{\frac{n\mathcal{E}}{(2^B-1)^2\sigma^2}}\right) = 2\left(1 - \frac{1}{2^B}\right) Q\left(\frac{\sqrt{n\mathcal{E}/\sigma^2}}{2^B-1}\right)$$

which is also decreasing exponentially as n increases, while the data rate is now B/n bits per channel use. So, as we are increasing n to reduce the error probability, what if we increased B also so that the data rate is not going to 0 with increasing n? Unfortunately, this does not work! Looking at the argument of $Q(\cdot)$ in the above equation, it should be obvious that if B increases linearly with n, then the argument of $Q(\cdot)$ approaches 0 and so $P(E)$ approaches 1. In fact, only by allowing B to increase *more slowly than logarithmically* as a function of n can we hope to make the argument of $Q(\cdot)$ increase, and thus make $P(E)$ arbitrarily small. But, if B is increased less than logarithmically with n, the data rate B/n bits per channel use is obviously reducing rapidly towards 0.

6.4. A suboptimum receiver for binary repetition coding

The optimum receiver for binary repetition coding takes n sample values at n successive time instants, and makes a decision as to what bit (or symbol) was sent based on the *sum* of the samples. Thus, it is actually not necessary to *store* all the earlier samples while waiting for the last one: it suffices to store the *running sum* $\sum_{i=0}^{j-1} \mathbb{Y}_i$ while waiting for $\mathbb{Y}_j$, and then simply add the new value to the running sum. An even simpler, but definitely suboptimum, receiver makes a *decision*, 0 or 1, on each $\mathbb{Y}_i$ and stores just the *running count* of how many 1 decisions have been made. After all n samples have been processed, the final decision is in favor of a 1 if the count is greater than $n/2$ and in favor of a 0 if the running count is smaller than $n/2$. If matters are arranged right, the final decision might be as simple as looking at the most significant bit of the counter!

Needless to say, the error probability of the suboptimum receiver is worse than that of the optimum receiver, though of course the implementation is much simpler and might be preferable where a premium is placed on other factors such as cost, weight, power consumption etc. of the receiver. For a given value of $\mathcal{E}$, the suboptimum receiver needs more repetitions than would be needed by an optimum receiver to achieve the same $P(E)$. Alternatively, for a fixed n, the optimum receiver can achieve the same error probability with a smaller received energy (and hence a smaller transmitter energy), and all of these factors (which do not show up in the mathematical model discussed above) need to be considered in the design of actual communication systems.