Class Activity #7
Application: The Optimal Digital Receiver

Digital Communications

The communication of digital data is widespread and only increasing. Be it the encoded voice signal transmitted from a cellular phone or an internet page downloaded over a DSL line, there are many variations on the same theme: the transmission and reception of 1's and 0's from a transmitter to a receiver. Due to various forms of interference (aka noise), the accurate communication of digital data is compromised. Fortunately, there are methods to improve the noise immunity of digital communications by applying many of the same analysis tools that we have learned in class thus far.

Representation of Binary Data

In a very basic sense, digital communications can be seen as the transmission of binary digits. To represent these digits, a pulse is used to represent each bit. For example, assume that the binary stream $[10110]$ is to be transmitted. One approach would be to send a positive pulse for '1' (like a 5V square pulse) and no pulse for '0' (this is the unipolar signaling class). While plausible, this approach is rarely used due to two problems: during the '0' pulse time, the line is "dead"; secondly, the square pulse used for '1' has very steep edges requiring a large amount of "bandwidth" to be accurately constructed (smooth signals require less frequency content than quickly changing signals – usable bandwidth is limited and allocated by the FCC). To correct for these problems, bipolar signaling is used (a positive pulse for '1' and a negative pulse for '0') and a smooth pulse shape is employed (such as a Gaussian-based pulse). The stream $[10110]$ is represented below in Figure 1 using bipolar signaling and a Gaussian pulse (rate of 1 Hz).

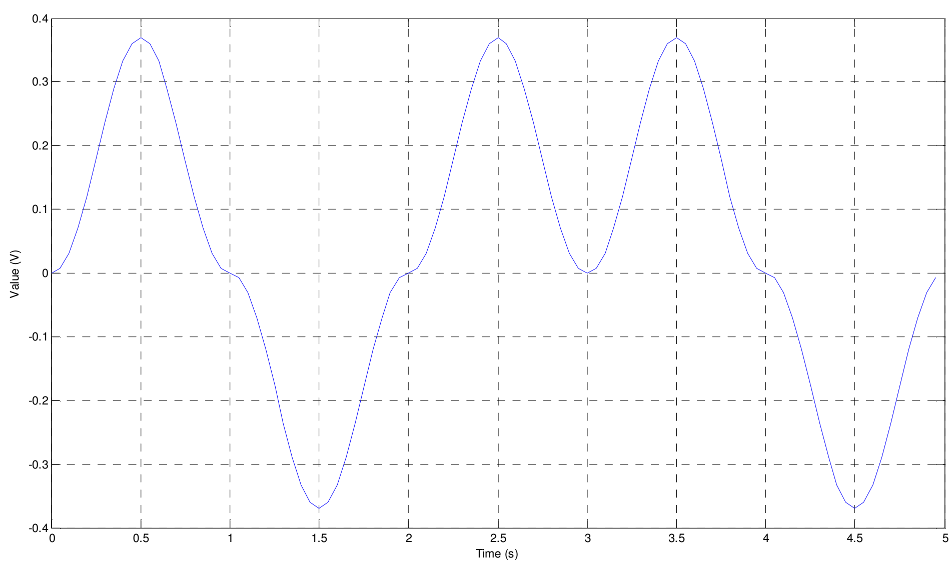

Figure 1: Bipolar Gaussian Pulse Stream

Baseband versus Bandpass

In many communication systems, the original information signal undergoes a process known as modulation. Modulation may package the signal using some encoding scheme while simultaneously translating the "baseband" signal to a "bandpass" signal. A baseband signal is the base information, and a low frequency-range signal. A common example occurs when listening to the radio; the baseband signal is the audio that you hear, i.e., the song or the DJ talking (or the commercial). To listen to that song, the radio is tuned to a specific frequency or channel. However, the transmitted radio signal is sent as a bandpass signal, i.e., the original baseband audio is now centered about some much higher frequency. The radio's job is to tune in the frequency range that you want to listen to and demodulate the bandpass signal back to baseband so you can listen.

The reason behind translating a baseband signal to a bandpass signal is twofold: first, to build a practical antenna and second, to allow multiple channels. As you may know from physics, the minimum length of an antenna is one-quarter the wavelength of the signal. Since wavelength is inversely proportional to frequency, very low frequencies make for impractically long antennas. In order to be transmittable, the signal needs to be at a high enough frequency such that a practical antenna can be built. Secondly, a radio that can only play one channel is limited. By modulating different baseband sources up to different bandpass ranges, multiple channels can share the same airwave channel. This is known as frequency-division multiplexing.

Our analysis in this lab will only consider the baseband signal. Some communication systems do transmit the baseband signal and do not frequency translate to a higher bandpass signal. Additionally, the results obtained are directly related to the results that would occur with more complex modulation. Typically, the baseband performance metric is used as the benchmark to compare to alternative approaches of modulation.

The Optimal Receiver

The optimal baseband digital receiver for a bandlimited AWGN channel (more later) is shown below in Figure 2 (sometimes known as the "matched filter" since the multiplying function *matches* the pulse template). Such a receiver maximizes the output SNR (the derivation of the matched filter can be found in most books that discuss digital communications).

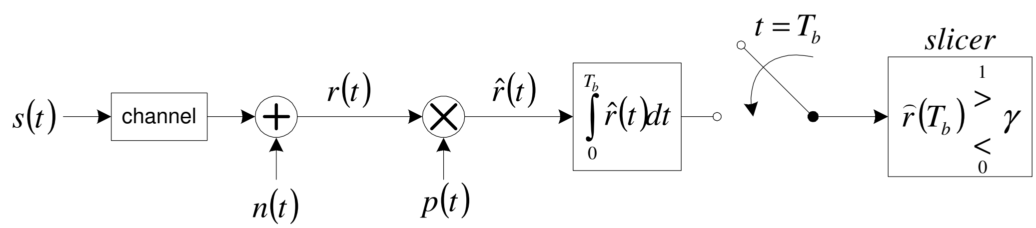

Figure 2: Matched Filter for a Baseband System

In Figure 2, $s(t)$ is the signal originating from the transmitter (in our case, a series of positive and negative pulses representing a stream of 1's and 0's). The signal $s(t)$ is then sent over the channel to the receiver, where the channel may be the atmosphere for a wireless transmission, or it might be a coaxial cable, etc. (for the purposes of this lab, assume that the channel is transparent and does not affect the signal). At the input to the receiver, it is assumed that the signal is corrupted by Additive White Gaussian Noise (AWGN). That is, the noise $n(t)$ *adds* to $s(t)$ as

$$r(t) = s(t) + n(t).$$

Furthermore, the noise distribution is *Gaussian*, or Normal. The designation *white* indicates that the noise is spectrally flat (like white light containing equal levels of colors of the rainbow); while this spectral property affects the design of the matched filter, it can be disregarded for current purposes.

The received signal $r(t)$ is then multiplied by the same pulse template $p(t)$ that was used to create $s(t)$. At this point, it is critical that the data stream be aligned to the pulse template; that is, each incoming pulse is one-by-one multiplied by $p(t)$ to generate $\hat{r}(t)$. The signal is then sent to a memory-less integrator (integration starts at the edge of each pulse with zero previous accumulation) which integrates for exactly one pulse period. The integrated signal is sampled at the end of the integration, resulting in the single value $\hat{r}(T_b)$. This value is then examined at the slicer (which includes the threshold γ): if $\hat{r}(T_b) > \gamma$, the assumed transmitted value is assigned to '1'; if $\hat{r}(T_b) < \gamma$, the assumed transmitted value is assigned '0' (cases of $\hat{r}(T_b)$ equal to the threshold can be assigned arbitrarily).

Noise and BER

The digital pulse stream is subject to noise interference from many sources. These sources include other transmissions (crosstalk), unrelated electrical activity, circuit noise (thermal noise), etc. The possible sources are numerous and varied such that the Central Limit Theorem applies and the noise interference has a Normal distribution. Furthermore, the noise will have zero mean (no reason to assume any DC bias). Note that the noise affects the '1' and the '0' pulse equally.

Assume that a '1' pulse is sent. Then, the mean of the received signal is

$$E[\hat{r}(T_b)] = E\left[\left(\int_0^{T_b} p^2(t) + p(t)n(t)\right)dt\right] = E\left[\int_0^{T_b} p^2(t)dt\right] + \int_0^{T_b} p(t)E[n(t)]dt = \int_0^{T_b} p^2(t)dt + 0 = \varepsilon_p,$$

where ε_p is defined as $\varepsilon_p = \int_0^{T_b} p^2(t)dt$. The expectation derivation above relies on the properties of expectation: expectation is linear (so the expectation of a sum of terms is the sum of the expectation of the terms); the expectation of a constant is that constant (for ε_p); and, again, expectation is linear (the expectation operator and the integral can be exchanged since both are linear operators), with the expected value of the noise zero ($\mu_n = 0$). Note that if a '0' is transmitted instead of a '1', the $E[\hat{r}(T_b)] = -\varepsilon_p$ (the transmitted pulse would be $-p(t)$).

The quantity $\hat{r}(T_b)$ can be thought of as a random variable, with its own PDF. Clearly, this random variable is Normally distributed since the noise giving this quantity its randomness is itself Normal. There are two PDFs, one if a '1' is sent and another if a '0' is sent. Since the noise that affects each of these scenarios is the same, the variance of each of these PDFs is the same, with only the means different (and opposite of each other).

The noise that appears at the slicer is related to the original noise as

$$\hat{n}(T_b) = \int_0^{T_b} n(t)p(t)dt .$$

Analyzing the variance of this noise is currently beyond the scope of this course. Suffice it say that since all of the transformations to the noise are linear, the distribution of the output noise remains Gaussian. Thus:

$$\hat{r}(T_b) \sim N(\varepsilon_p, \sigma_{\hat{n}}^2) \text{ if a '1' was sent;}$$

$$\hat{r}(T_b) \sim N(-\varepsilon_p, \sigma_{\hat{n}}^2) \text{ if a '0' was sent.}$$

That is, in either case of a '1' or '0' sent, the PDF is Gaussian and has the same shape, with equal and opposite means. A representative plot of these density functions are given below in Figure 3, with $\varepsilon_p = 1$ and $\sigma_{\hat{n}}^2 = 0.25$.

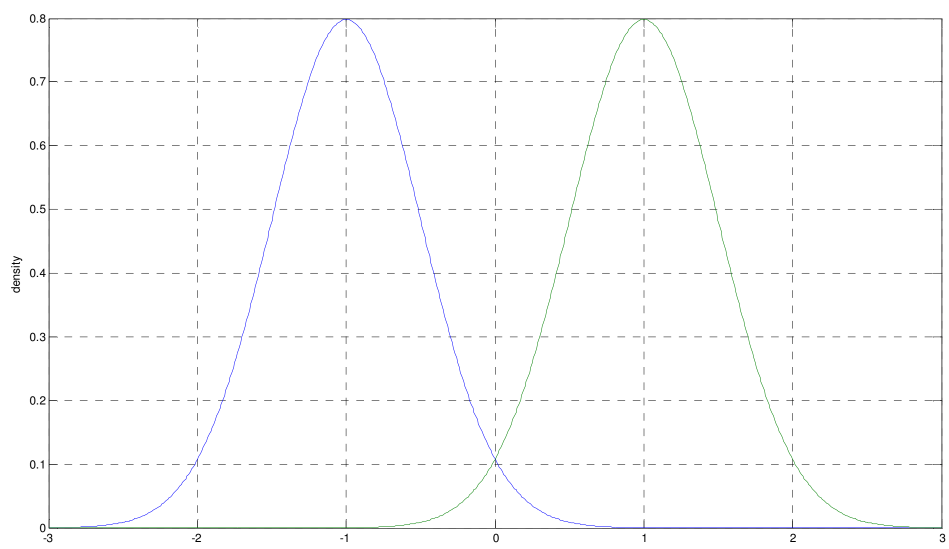

Figure 3: PDF's at Slicer for '0' and '1' Transmitted

In the case of equal probabilities of bits (i.e., $P(1) = P(0) = 0.5$), the optimum choice for the threshold is where the two PDF curves meet; that is,

$$\gamma_{opt} = f_{\hat{r}(T_b)|1}(x|1) = f_{\hat{r}(T_b)|0}(x|0) = \frac{\varepsilon_p - (-\varepsilon_p)}{2} .$$

This choice minimizes the probability of error *e*, which can be expressed as

$$P(e) = P(e|1)P(1) + P(e|0)P(0),$$

or

$$P(e) = P(1)\int_{-\infty}^{\gamma} f_{\tilde{r}(T_b)|1}(x|1)dx + P(0)\int_{\gamma}^{\infty} f_{\tilde{r}(T_b)|0}(x|0)dx.$$

For the situation of equal '1's and '0's, and PDF curves as shown in Figure 3, $\gamma_{opt} = 0$. Under these conditions and using the matched filter, the BER (Bit Error Rate or probability of error) is expressed as

$$P(e) = \Phi_c\left(\sqrt{\frac{\varepsilon_p}{\sigma_n^2}}\right),$$

where $\Phi_c(\cdot)$ is the complimentary Q-function (i.e., $1 - \Phi(\cdot)$; note that $\Phi_c(\cdot)$ is available on MATLAB as `qfunc`). Here, σ_n^2 is the variance of $n(t)$ *prior* to the matched filter. It is useful to think of $\varepsilon_p / \sigma_n^2$ as the SNR. Note that as the $SNR \to 0$, the $P(e) \to 0.5$, implying that a guess is just as good as processing the signal for extremely noisy situations. Conversely, $P(e) \to 0$ as the $SNR \to \infty$, as expected.

Assignment:

Note: as an option, you may use MATLAB Simulink to complete this lab.

1. Generate a set of at least 5000 randomly chosen bits with $P(1) = P(0) = 0.5$. Use a pulse template to create the transmitted data stream $s(t)$ (you may use a pulse template of your choosing or the Gaussian pulse available on Blackboard – the supplied pulse has $\varepsilon_p = 1$).
   - Suggestion: use a function to generate the binary digits and pulse stream.
   - Plot an arbitrary segment of 4 or 5 bits to make sure your pulse stream looks correct.
2. Compute the theoretical BER for our situation with $P(1) = P(0) = 0.5$, bipolar signaling, $\gamma_{opt} = 0$ and the matched filter (that is, evaluate $P(e) = \Phi_c(\sqrt{SNR})$, with $SNR = \varepsilon_p / \sigma_n^2$. Let the SNR range from $-8\,dB \to 10\,dB$.
3. Run the matched filter for the same SNR range as in step (2) above (to scale the noise, we have that the $SNR = \varepsilon_p / \sigma_n^2$, find σ_n and scale properly to generate $n(t)$). To integrate, use the MATLAB function `trapz` (for trapezoidal integration – no need to correct for unit spacing since any correction factor scales the noise and signal identically). Apply the slicer with $\gamma_{opt} = 0$ to estimate the received digits. Compare the detected digits with the original digits and count the number of errors to find the BER.

- Suggestion: create a function to apply the matched filter and extract the received digits and another function to compare the original data with the extracted data to determine the BER.
- Plot a segment of your noise-corrupted data stream (4-5 bits) for the case of the lowest SNR tested. Comment on the level of noise corruption.
- Plot both your theoretical BER and your empirically determined BER on the same plot using a logarithmic scale for the y-axis. Comment on how well these two curves agree.

4. Consider the discussion on how the threshold γ was determined. Qualitatively discuss why this choice of $\gamma_{opt} = 0$ yields the lowest probability of error. Discuss how the threshold should change if a '1' is more likely that a '0' and vice-versa. Rerun your simulation (for the same SNR range as in step (3)) with a slightly lower and higher γ. Report on the change in BER. Comment.
5. The matched filter uses a pulse template matched to the transmitted pulse. In your matched filter routine, try a pulse template that does not match your transmitted pulse (i.e., if you used the Gaussian pulse, try a simple flat-top box at the receiver). In this case, $\varepsilon_p = \int_0^{T_b} p(t)\hat{p}(t)dt$, where $\hat{p}(t)$ is the incorrect pulse template at the receiver. With this new value for ε_p, rerun your simulation for the same SNR range. Report on the change in SNR. Comment.

Turn in (due by start of the next lab):

- Plot of clean data stream (#1), noisy data stream (#3) and comments.
- Plot of theoretical BER curve versus empirically determined BER curve (on same plot using logarithmic y-axis). Comment on results and agreement.
- Results of BER test with non-optimal γ. Discussion.
- Results of BER test with non-optimal receiver pulse template. Discussion.
- Overall script to run all parts of the lab.
- MATLAB scripts for any functions.