Laboratory #3

Pulse Shaping for Band-limited Channels

SYSC 4600 Digital Communications

Department of Systems and Computer Engineering
Faculty of Engineering
Carleton University

Purpose and Objectives

As we have studied in this course, the shape of the transmitted pulse has no effect on the probability of error of the communication system, which depends only on the relative positions of the points in the signal constellation. The choice of the pulse shape is primarily to control the power spectral density of the transmitted signal, which is particularly important for communication systems that must operate in a limited bandwidth, such as all wireless communication systems. In this laboratory experiment we will investigate the practical limitations on the choice of the pulse shape, and how to adjust system parameters to facilitate transmission over channels with band-limited frequency response. It builds on Laboratory #2, and you are encouraged to refer to previous lab manuals while working on this lab.

Background

For IQ modulation schemes (e.g., PSK, PAM, QAM), the transmitted bandpass signal can be expressed mathematically as

$$v_c(t) = \Re\left\{v(t)\sqrt{2}e^{j2\pi f_c t}\right\}, \tag{1}$$

where $v(t)$ is the complex lowpass equivalent baseband signal, f_c is the carrier frequency, and $\Re\{z\}$ denotes the real part of z. The complex lowpass equivalent signal can be expressed using a pulse train as

$$v(t) = \sum_{n=0}^{N_v-1} v_n h_T(t - nT) \ , \tag{2}$$

where v_n is the n^{th} transmitted symbol, N_v is the number of transmitted symbols, T is the symbol period and $h_T(t)$ is the pulse shape. The transmitted symbols are points in the signal constellation for the modulation scheme that is being used. These are represented as complex numbers.

When the signal is transmitted it is corrupted by additive noise, so the received signal can be expressed as

$$r_c(t) = v_c(t) + w_c(t) \ , \tag{3}$$

where $w_c(t)$ is the additive white Gaussian noise (AWGN).

The demodulated signal at the receiver is given by

$$\begin{aligned} r_o(t) &= r_c(t)\sqrt{2}e^{-j2\pi f_c t} \\ &= [v_c(t) + w_c(t)]\sqrt{2}e^{-j2\pi f_c t} \\ &= v_c(t)\sqrt{2}e^{-j2\pi f_c t} + w_c(t)\sqrt{2}e^{-j2\pi f_c t} \\ &= v_c(t)\sqrt{2}e^{-j2\pi f_c t} + w_o(t) \ , \end{aligned} \tag{4}$$

where $w_o(t) = w_c(t)\sqrt{2}e^{-j2\pi f_c t}$ is the demodulated noise, which is modelled as complex additive white Gaussian noise (the real and imaginary parts are independent stationary Gaussian random processes with zero-mean and a flat double-sided noise power spectral density of $N_0/2$). Substituting Equation (1) for $v_c(t)$ gives in Eq. (4)

$$r_o(t) = \Re\left\{v(t)\sqrt{2}e^{j2\pi f_c t}\right\}\sqrt{2}e^{-j2\pi f_c t} + w_o(t) \ . \tag{5}$$

Since $\Re\{z\} = \frac{1}{2}[z+z^*]$, where the superscript * denotes complex conjugate, we can write this as

$$r_o(t) = \tfrac{1}{2}\left[v(t)\sqrt{2}e^{j2\pi f_c t} + v^*(t)\sqrt{2}e^{-j2\pi f_c t}\right]\sqrt{2}e^{-j2\pi f_c t} + w_o(t) \;, \tag{6}$$

which simplifies to

$$r_o(t) = v(t) + v^*(t)e^{-j2\pi(2f_c)t} + w_o(t) \;. \tag{7}$$

The first term in Eq. (7) contains the message data (the part we're most interested in), the second term is centered around a high frequency ($2f_c$) and will be filtered out by the matched filter so we can ignore it, and the third term is the additive noise. Substituting Eq. (2) for $v(t)$ in Eq. (7) and dropping the high-frequency term gives

$$r_o(t) = \sum_{n=0}^{N_v-1} v_n h_T(t - nT) + w_o(t) \;. \tag{8}$$

The demodulated signal is filtered and sampled every T seconds. In general, the filter is matched to the transmitted pulse shape and so has an impulse response of $h_R(t) = h_T(T_0 - t)$, where $h_T(t)$ is the transmitted pulse shape, and T_0 is the time at which the output of the matched filter should be sampled to maximize the signal-to-noise ratio. In previous labs we chose T_0 to be equal to T, but strictly speaking that is not a requirement. For now we will choose $T_0 = 0$, and will discuss the merits of different values later.

The output of the matched filter is given by

$$r(t) = r_o(t) \circledast h_R(t) = \int_{-\infty}^{\infty} r_o(t-\tau)\, h_R(\tau)\, d\tau \tag{9}$$

where $\circledast$ denotes convolution. Substituting Eq. (8) into Eq. (9) gives

$$r(t) = \int_{-\infty}^{\infty} \left[\sum_{n=0}^{N_v-1} v_n h_T(t - \tau - nT) + w_o(t-\tau)\right] h_R(\tau)\, d\tau \;. \tag{10}$$

Rearranging the order of integration and summation gives

$$r(t) = \sum_{n=0}^{N_v-1} v_n \int_{-\infty}^{\infty} h_T(t - nT - \tau) h_R(\tau)\, d\tau + \int_{-\infty}^{\infty} w_o(t-\tau) h_R(\tau)\, d\tau \;. \tag{11}$$

By defining

$$h_{TR}(t) = h_T(t) \circledast h_R(t) = \int_{-\infty}^{\infty} h_T(t-\tau)\, h_R(\tau)\, d\tau \tag{12}$$

as the combined impulse response of the transmit and received filters, and

$$w(t) = \int_{-\infty}^{\infty} w_o(t-\tau) h_R(\tau)\, d\tau \tag{13}$$

as the filtered additive noise, we can write $r(t)$ as

$$r(t) = \sum_{n=0}^{N_v-1} v_n h_{TR}(t - nT) + w(t) \;. \tag{14}$$

With $T_0 = 0$ the output of the matched filter should be sampled at times $t = nT$, so the samples are

$$r_n = r(nT) = \sum_{m=0}^{N_v-1} v_m h_{TR}(nT - mT) + w_n \ . \tag{15}$$

where $w_n = w(nT)$. Ideally, each sample will depend on one and only one transmitted symbol. That is, r_n should depend only on v_n, and not on any of the other transmitted symbols. To fulfill this requirement, we need $h_{TR}([n-m]T)$ to be equal to zero for every $m \neq n$, so we need

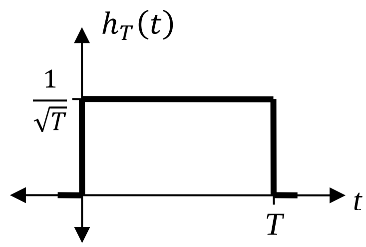
$$h_{TR}(nT) = \delta_n = \begin{cases} 1 & \text{if } n = 0 \\ 0 & \text{if } n \neq 0 \end{cases} \ . \tag{16}$$

where δ_n is the Kronecker delta. If this condition is met then Eq. (15) can be written as

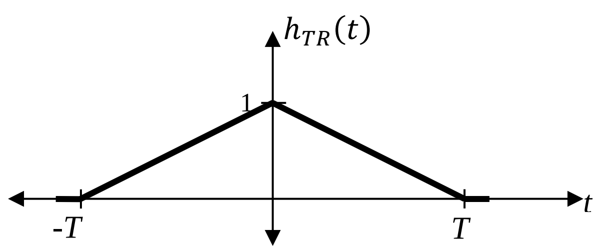
$$r_n = \sum_{m=0}^{N_v-1} v_m \delta_{n-m} + w_n = v_n + w_n \ , \tag{17}$$

Otherwise, r_n will depend not only on v_n but also on some of the other transmitted symbols. This is known as *intersymbol interference* (ISI). If ignored, ISI will severely reduce the performance of the communication system, and it is difficult to remove ISI if it occurs. As such, it is desirable to only use pulse shapes that do not cause ISI.

If the communication system uses a simple normalized rectangular pulse, as shown below,



with a matched filter at the receiver, so $h_R(t) = h_T(-t)$ with $T_0 = 0$, then the combined impulse response of the transmit and receive filters is



Observe that $h_{TR}(nT)$ equals 1 if $n = 0$ and equals 0 for any other integer n, thus satisfying the condition of Eq. (16).

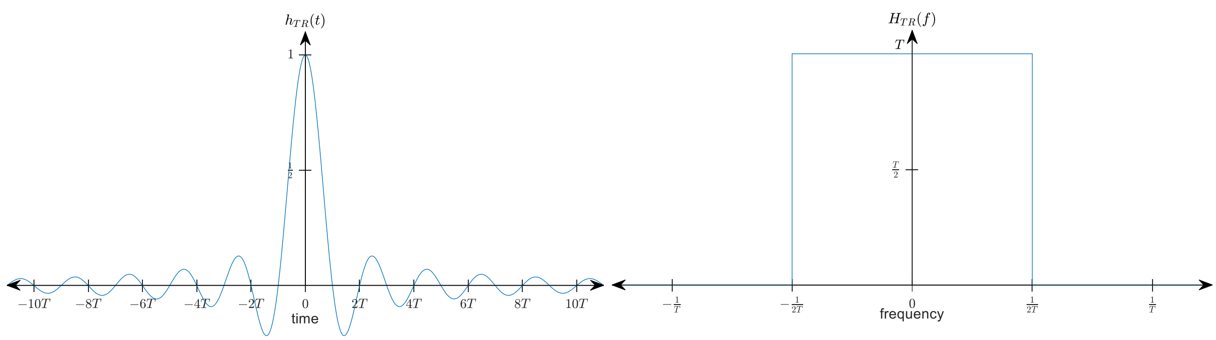
The condition of Eq. (16) is satisfied for all pulse shapes that have a duration that is less than or equal to the symbol period (that is, T seconds or less), such as all the pulse shapes studies in previous labs in this course. However, all pulse shapes that are limited in duration will theoretically require infinite bandwidth to transmit. In practice, because the power spectral density of the transmitted signal when these time-limited pulse shapes are used gets very small as at larger frequencies, they do have a practical range where the PSD is *essentially* non-zero (the 95% power bandwidth is one useful measure). However, this practical bandwidth is still very large.

By making the pulse shape last for longer than one symbol interval (i.e., longer than T seconds), this practical bandwidth can be reduced. When this is done, the pulses for different symbols will overlap in time. For example, at time $t = 0$ the transmitter will begin transmitting the first symbol, with a pulse shape that may last $10T$ seconds. At time $t = T$, the transmitter will begin transmitting the second symbol, even though the first symbol is still being transmitted. In this example there would be up to ten different symbols being transmitted at any given time. Clearly in such a situation, because the symbols overlap in time, intersymbol interference seems almost inevitable.

Fortunately, there are some pulse shapes that give a limited bandwidth while also avoiding ISI. One such example is the sinc pulse,

$$h_{TR}(t) = \frac{\sin \pi t/T}{\pi t/T} \tag{18}$$

which looks like the following in the time and frequency domain.



We can see that $h_{TR}(nT) = \delta_n$ so there is no ISI, and the spectrum of the signal is limited to frequencies below $1/(2T)$. This pulse shape is optimal in that it requires the smallest possible bandwidth without introducing ISI. Unfortunately, because $h_{TR}(t)$ decays slowly the pulse duration must be very long. However, the sinc pulse isn't the only choice we have.

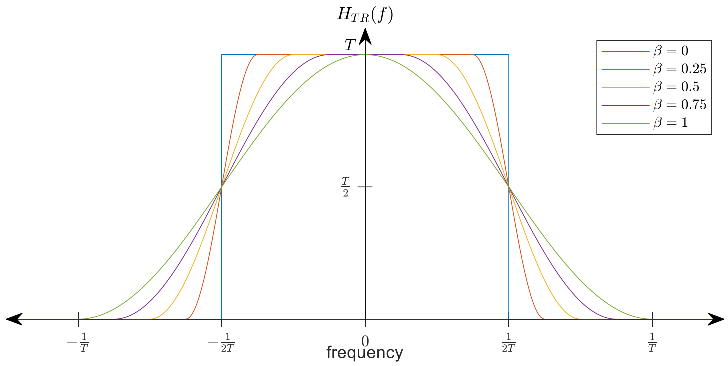
According to *Nyquist's ISI criterion*, any pulse shape with the property that

$$\frac{1}{T}\sum_{k=-\infty}^{\infty} H_{TR}\left(f - \frac{k}{T}\right) = 1 \quad \forall\, f \tag{19}$$

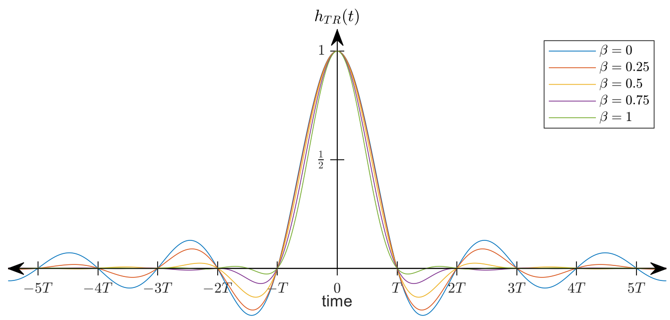
will also satisfy the condition of Eq. (16) and will therefore not introduce any ISI. The raised cosine pulses are a family of pulses that are band-limited and do not introduce ISI. They are defined by

$$H_{TR}(f) = \begin{cases} T & \text{if } |f| \le \dfrac{1-\beta}{2T} \\ \dfrac{T}{2}\left[1 + \cos\left(\dfrac{\pi T}{\beta}\left[|f| - \dfrac{1-\beta}{2T}\right]\right)\right] & \text{if } \dfrac{1-\beta}{2T} < |f| \le \dfrac{1+\beta}{2T} \\ 0 & \text{if } |f| > \dfrac{1+\beta}{2T} \end{cases} \tag{20}$$

where β, with $0 \leq \beta \leq 1$, is a parameter, known as the *roll-off factor*, that controls the bandwidth. Some examples, for different values of β, are shown below.



Note that the sinc pulse is a special case of the raised cosine pulse, when $\beta = 0$. In the time domain the raised cosine pulses appear as shown below.



The zero-crossings at multiples of T are evident for all β, ensuring no ISI.

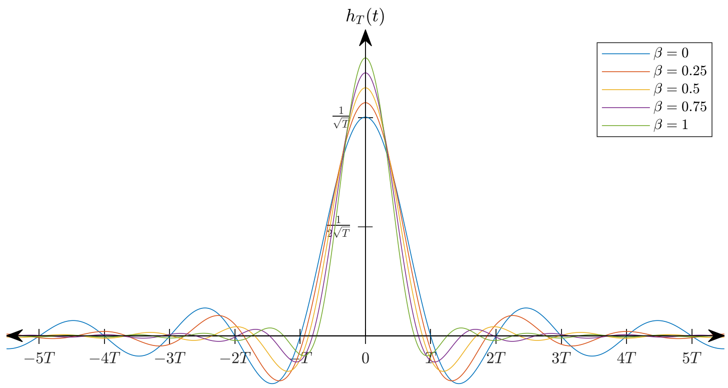
Although the properties of $h_{TR}(t)$ are important for preventing ISI, communication systems use $h_T(t)$ at the transmitter, and $h_R(t)$ at the receiver, and $h_{TR}(t)$ is the convolution of $h_T(t)$ and $h_R(t)$. To realize a desired combined pulse shape of $h_{TR}(t)$, the transmitter should use a pulse shape defined by

$$H_T(f) = \sqrt{H_{TR}(f)} \tag{21}$$

and the receiver should use a matched filter, with $h_R(t) = h_T(-t)$. That is, the transmitter should use a root raised cosine pulse shape, which is given in the time domain as

$$h_T(t) = \frac{1}{\sqrt{T}} \frac{4\beta t/T \cos[\pi(1+\beta)t/T] + \sin[\pi(1-\beta)t/T]}{(\pi\, t/T)[1-(4\beta t/T)^2]} \;, \tag{22}$$

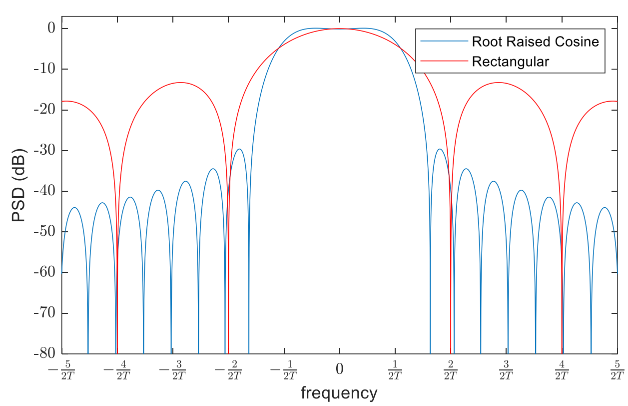
which is plotted in the figure below.



Observe that $h_T(t)$ does not by itself satisfy Eq. (16), since the zero-crossings are not at multiples of T. This is not important, since only the combination of $h_T(t)$ and $h_R(t)$, namely $h_{TR}(t)$, needs to satisfy Eq. (16).

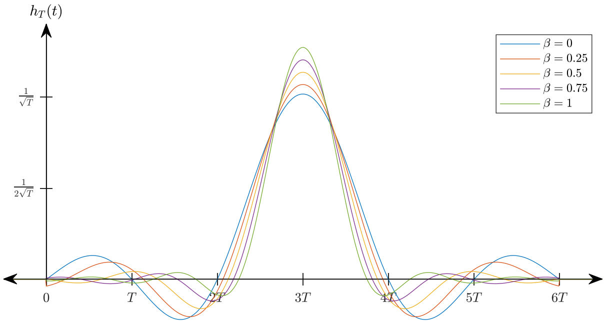
The root raised cosine pulse shapes are very appealing because they allow for communication over band-limited channels without introducing ISI. However, these theoretical benefits are not possible in practice. The problem is that the pulse shapes have infinite duration (so, if we wanted to use them in the future, we should have already started transmitting them now). In practice, a truncated version of the pulse shape, time-limited to last for only a few symbol periods, is used instead.

Truncating the pulse shape leads to two problems. One problem is that, because the truncated pulse shape is not ideal, it causes ISI to appear at the output of the matched filter. If the ISI is too severe, the probability of error of the communication system will be unacceptably high. Another problem is that the signal is no longer perfectly band-limited. Sidelobes will appear in the power spectral density of the transmitted signal, as shown in the figure below. This is known as *spectral regrowth*.



This figure is for $\beta = 0.5$, when the root raised cosine pulse is truncated to $L = 4$ symbol periods (that is, the pulse duration is $4T$ seconds long). Without this truncation, the PSD would be limited to a total double-sided bandwidth of $1.5/T$, whereas with truncation the main lobe is slightly wider, and there are sidelobes beyond that. However, the sidelobe are very low (containing very little power), so they won't cause much adjacent channel interference, and if they are cut off by a band-limited channel, the pulse shape will not be distorted very much. The sidelobes of the truncated root raised cosine pulse are much lower than the massive sidelobes that arise from the use of a rectangular pulse, as shown in the above figure. Increasing the pulse length, L, will decrease the height of the sidelobes, but this adds more complexity of the system so it is desirable to keep L small. Decreasing β will make the main lobe narrower but will increase the height of the sidelobes.

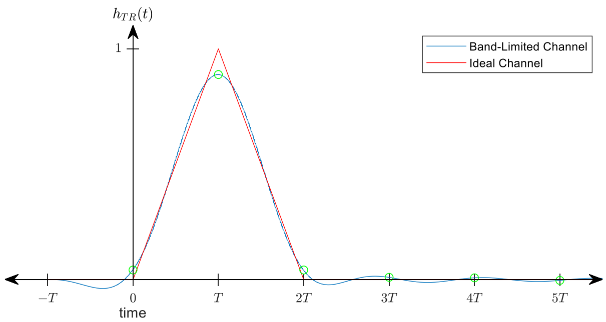
Because the truncated pulse runs from time $-LT/2$ to $LT/2$, it is non-causal, so in practice the pulse is delayed by $LT/2$ to make it causal. The actual transmitted pulse shapes then look like the ones shown below (shown for $L = 6$).



The matched filter at the receiver should have an impulse response of $h_R(t) = h_T(LT - t)$, and the output should be sampled at time $T_0 = LT$.

Band-limited Channels

In most real-life situations, the channel bandwidth is not unlimited. Physical properties of the channel, or governmental regulations, usually require that most of the transmitted signal power be focused in a narrow range of frequencies, and signal components outside of this range are blocked by the communication channel. This has the effect of distorting the pulse shape, thereby introducing additional ISI. The introduced ISI is more severe if a poor pulse shape is used relative to the bandwidth of the channel. To illustrate this effect, suppose a rectangular pulse with a duration of T seconds is used for communication over a channel that is band-limited to $0.75/T$ Hz. The received combined pulse shape, $h_{TR}(t)$, looks as shown below (in blue), instead of having the ideal triangular shape (shown in red). The values of $h_{TR}(t)$ at the sampling instants show that ISI will occur.



On the other hand, because sidelobes of the PSD of the root raised cosine are so small, if they are cut off by the channel the pulse shape is not affected very much and very little additional ISI is introduced.

Laboratory

Part I: Pulse Shaping

Step 1: Before you begin, make sure you have completed Lab #2. You will use the simulator you wrote for 4-QAM. Make sure it's working properly and that your simulator is giving results that closely match the theoretical probability of error when $N_a = 1000$ and $N_f = 1000$.

Note: If you were unable to complete Lab #2 then you can use the program start.m that is provided on cuLearn as your starting simulator.

Adjust your simulator parameters to send only one message word ($N_f = 1$) that contains $N_a = 20$ bits (i.e., ten 4-QAM message symbols), and disable the additive noise.

Use $T = 0.01$ seconds, $\eta = 64$ samples per symbol, and $f_c = 800$ Hz.

Make sure your simulator gives no bit errors when there is no noise. Confirm that the samples at the input to the decision device are equal to points in the 4-QAM signal constellation (i.e., they should all be $\pm 1 \pm j$).

Step 2: Replace the rectangular pulse with a root raised cosine pulse, with a roll-off factor of $\beta = 0.5$ and a truncated pulse duration of $L = 128$ symbol periods. You can use the function provided in root_raised_cosine.m on cuLearn (just download the file and put it in the same folder as your MATLAB script). You would use the function with something like

```
hT = root_raised_cosine(beta, L, T, eta);
```

where `beta` is the roll-off factor, `L` is the pulse duration (in symbol periods), `T` is the symbol period (0.01 seconds) and `eta` is the oversampling ratio (η).

Because the pulse duration is now LT seconds, instead of T seconds, you'll need to change when you sample the output of the matched filter. The samples should be

$$r_n = r(LT + nT) \ .$$

Make sure you are collecting at the correct sampling times. Confirm that the samples at the input to the decision device are very close to points in the 4-QAM signal constellation (i.e., $\pm 1 \pm j$). Note that they won't be exactly the same, but they should be within $\pm 10^{-6}$ if you're sampling at the correct times (Hint: the first sample should probably be element $L\eta$ in your MATLAB vector). Make sure your simulator gives no bit errors when there is no noise.

Step 3: Test the simulator in the presence of noise. Increase the message word length to $N_a = 1000$ bits and simulate $N_f = 1000$ message words. Use $\beta = 0.5$, but reduce L to 4. For E_b/N_0 in the range of 0 dB to 10 dB with a step size of 1 dB, plot the BER vs. E_b/N_0 in a graph, along with the theoretical BER for when the channel is not band-limited. The theoretical and experimental bit error rates should be very similar.

Next, rerun your simulation with $\beta = 0.1$ and $L = 4$. There should now be a big difference between the theoretical and experimental results. This is because the truncated version of the root raised cosine pulse is not very accurate when both β and L are small, so some ISI is introduced, degrading the performance.

Rerun your simulation with $\beta = 0.1$ and $L = 16$. The theoretical and experimental results should now match closely. The smaller the choice for β, the larger that L needs to be to prevent

ISI. But increasing L adds complexity (and therefore cost) to the system. Did you notice your simulator running more slowly?

Save a backup of your MATLAB program, because you'll need it in Part II.

Step 4: Using the techniques you learned in Lab #1, calculate and plot the power spectral density of the transmitted bandpass signal, $v_c(t)$. Use the following parameters: $N_a = 1000$ bits, $T = 0.01$ seconds, $\eta = 64$ samples per symbol, and a carrier frequency of $f_c = 800$ Hz. Use a root raised cosine filter with parameters $\beta = 0.5$ and $L = 4$. Use $N_f = 1000$ frames to get smooth results.

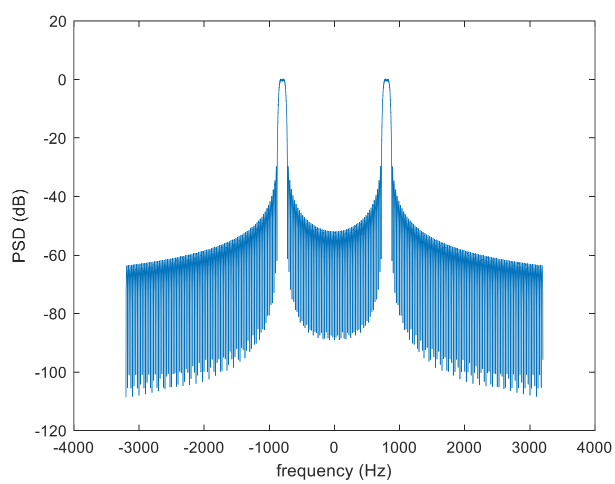
Note that in Lab #1 you were asked to find the PSD of the baseband signal, $v(t)$, whereas here you are to find the PSD of the bandpass signal. You can use the following MATLAB code to calculate and plot the PSD (assuming $N_f = 1$):

```
Ns = length(vct);                     % Length of bandpass signal
Vcf = fftshift(fft(vct));             % Calculate FFT
PSD = Vcf.*conj(Vcf) * Ts / Ns .* sinc((-Ns/2:Ns/2-1)/Ns).^2

f = (-Ns/2:Ns/2-1) / (Ns*Ts);         % Frequencies
plot(f, 10*log10(PSD));
```

where `vct` is the MATLAB vector containing the samples of the transmitted bandpass signal, $v_c(t)$, and `Ts` is the sampling period (T/η). You'll need to modify this slightly, as you did in Lab #1, to average the PSD over $N_f = 1000$ frames.

Your graph should look similar to the one shown below.



As expected, you can see that the bulk of the signal power is centered around the carrier frequency of 800 Hz (with a replica at -800 Hz). Now use the `axis` command to zoom in on the area around the carrier frequency:

```
axis([650 950 -80 10]);
```

Roughly estimate the null-to-null bandwidth. It should be slightly larger than $(1 + \beta)/T$.

Try changing the parameters a bit to see their effects on the PSD. For $\beta = 0.5$, try different values of L (e.g., with $L \in \{2,4,8,32\}$, if possible plotting them all in the same graph). Observe that the sidelobes decrease with larger L, and the null-to-null bandwidth comes closer to $(1+\beta)/T$.

For $L = 16$, try different values of β (e.g., with $\beta \in \{0,0.2,0.5,0.8,1\}$). Observe that the null-to-null bandwidth changes with β, and the sidelobes decrease with larger β.

For $L = 4$ and $\beta = 0.5$, try different values of T (with $T \in \{0.02, 0.01, 0.007\}$). As expected, the bandwidth of the signal increases with smaller values of T. That is, the faster we want to transmit information, the more bandwidth we need. If the bandwidth of the channel is limited, the challenge is to try to communicate as quickly as possible, without increasing the probability of error too much.

Part II: Band-limited Channels

In this section we will study the effects on the BER when the channel is band-limited (that is, the channel cuts off all frequency components outside of a certain range of frequencies. In particular, we will consider a channel where only signal components in the range of frequencies between 725 Hz and 875 Hz will pass through the channel (i.e., the channel bandwidth is limited to 150 Hz, centered around 800 Hz).

Experiments:

Step 1: Go back to your simulator from Lab #2 for 4-QAM with a rectangular pulse shape (or use the simulator in start.m in cuLearn). Make sure it's working properly and that your simulator is giving results that closely match the theoretical probability of error when $N_a = 1000$ and $N_f = 1000$.

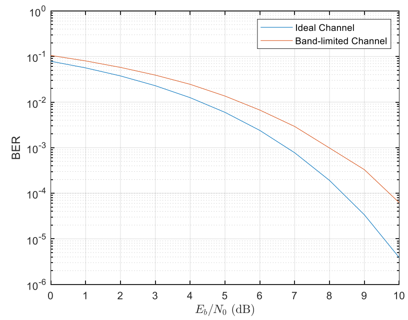
Use $T = 0.01$ seconds, $\eta = 64$ samples per symbol, and $f_c = 800$ Hz.

Step 2: Modify your simulator to use a band-limited channel. You should download and use the function provided in band_limited_channel.m on cuLearn. You can use this in your AWGN bandpass channel model as:

```
rct = band_limited_channel(vct, Ts);
rct = rct + sqrt(1/Ts*No/2)*randn(1, length(rct));
```

where `vct` is the MATLAB vector containing the samples of the transmitted bandpass signal, $v_c(t)$, `Ts` is the sampling period ($T_s = T/\eta$), `No` is the single-sided noise power spectral density, and `rct` will contain the samples of the received bandpass signal, $r_c(t)$.

For E_b/N_0 in the range of 0 dB to 10 dB with a step size of 1 dB, plot the BER vs. E_b/N_0 in a graph. Your graph should be similar to the one below. The "Ideal Channel" results are for a channel with infinite bandwidth (only AWGN). We can see that the performance in the band-limited channel is noticeably worse, because of the ISI that is introduced by the non-ideal channel.



By slowing down the transmission rate (i.e., by increasing T) the bandwidth of the signal will reduce, so the signal will fit better into the channel band. This will result in less ISI, and better performance. Try running your simulator with $T = 0.0333$. Observe that the performance is much closer to the ideal case, but note that the system is only sending 60 bits per second instead to 200 bits per second. The transmission rate is given by $(\log_2 M)/T$, where M is the constellation size ($M = 4$ for 4-QAM).

Step 3: Go back to the version of your solution that you saved in Step 3 of Part I, which plots the BER vs E_b/N_0 when a root raised cosine pulse shape is used. Make sure it's still working. Then make the same change to that program as you made in Step 2 of Part II. That is, use the band-limited channel.

Run your simulator with $T = 0.01$, $\beta = 0.5$, and $L = 4$. Observe that the performance is nearly identical to the ideal case, while still having a throughput of 200 bits per second. By using a pulse shape that has lower sidelobes, less ISI is introduced by the band-limited channel, so good performance is still possible.

Step 4: Bonus practice, but you will have to do something similar for the final project. Set $\beta = 0.25$ and $L = 16$. Set your simulator to only run at $E_b/N_0 = 7$ dB. Try to make T as small as possible (i.e., make the transmission rate as high as possible) while ensuring that the BER is less than 10^{-3}. What is the throughput (in bits per second) with this value of T?