# ECE 301 Computational Problems

You can use MATLAB or any other computational software. Please turn in the final plots and your code as a single pdf. Make sure to label the plots and add 1 paragraph (few sentences) of explanation per plot. Hints for each problem will be uploaded soon. You are welcome to discuss with the TA during the next two weeks during office hours. Solve the problem sequentially from the first to the last, that's the best way to proceed.

## 1) Graphical Convolution

The convolution of two continuous time functions f(t) and g(t) can be given by the convolution integral $\int_{-\infty}^{\infty} f(tau)g(t - tau)dtau$. Develop an algorithm in Matlab to graphically represent the convolution of two functions f(t) and g(t), where either function is a rectangular pulse of different lengths. The final deliverable must contain the following subplots:

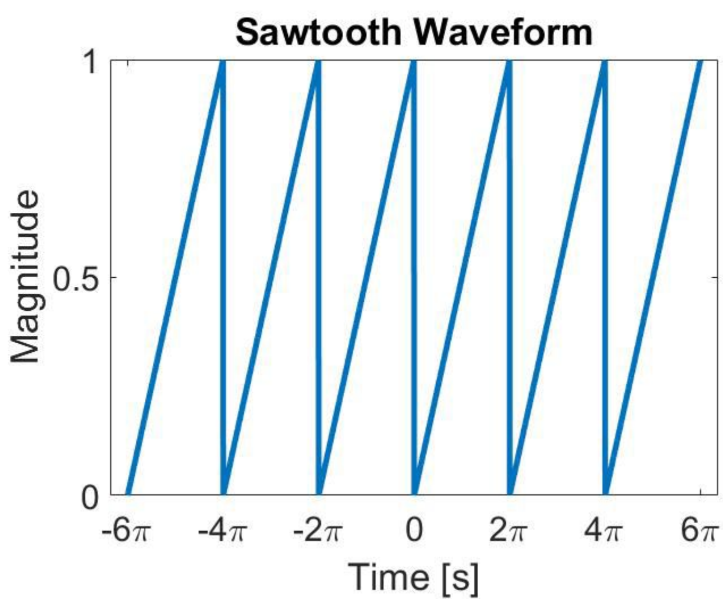
- The function f(t)
- Another subplot to graphically animate the function f(t) overlapping with g(t). Hints: Keeping g(t) static, use the **pause** and **drawnow** features in Matlab to move f(t) as a function of time.
- The final subplot must represent a graphical animation of the convolution product which is only defined in the overlapping regions. (refer to the above hint on moving the plot in time).

## 2) Fourier Series and Systems

The signal shown in figure #1 is periodic with $2\pi$. Your task is to represent the time domain signal using its Fourier series coefficients. You must plot this representation in Matlab keeping 3, 5, and 10 sinusoidal terms.

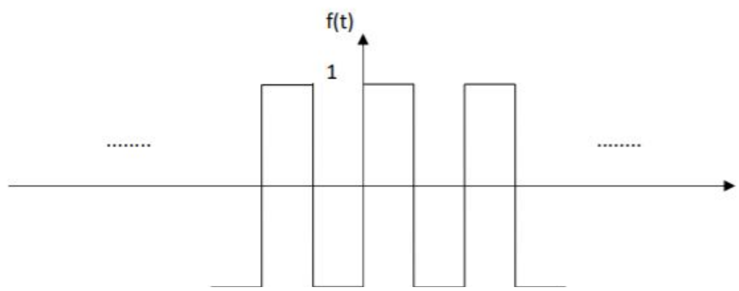
What you need to turn in:

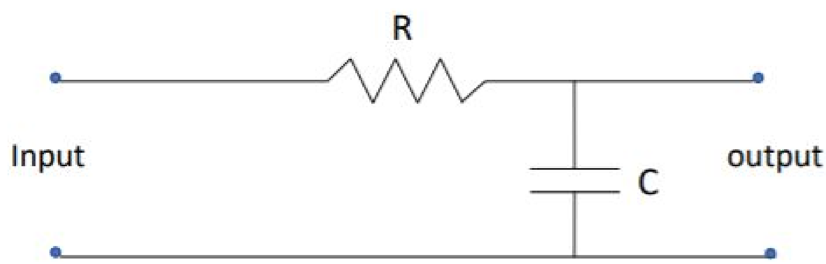
- An expression for the time domain signal in terms of sines and cosines.
- The three plots.



## 3) Fourier Series and Systems

The goal of this Matlab problem is to understand low pass filters and signal distortion. When a signal with high frequency components passes through a standard RC circuit (a low pass filter), the output waveform is distorted. Consider a waveform as shown below which passes through the RC circuit. Plot the output signal in the time domain.




Take the time period of the signal to be unity and R=10k ohms, C= 5 microfarad.

Hints:

1. Note the signal is periodic. Find the Fourier Series Coefficients of $x(t)$.
2. Find the frequency response of the circuit. This was derived in class. You should plot the amplitude and phase in Matlab to understand the low pass filtering action better.
3. Remember, every frequency component passes through the low pass filter (linear system) independently so first obtain the output Fourier coefficient in the frequency domain. You can simply sum up all these output

frequency components with the correct weighting factors multiplied by the correct basis signals to obtain the output signal. Plot this expression to see the distortion in the output signal.

**4) Fourier Series and Systems**

The goal of this Matlab assignment is to get you acquainted with Gaussian optical pulses. In optical fiber inter-continental communications widely used today, the data bits are represented by pulses of light. The frequency of the light is chosen at which optical fibers made of glass have the lowest absorption loss. However, the optical fibers expand the pulse because of a "spectral filtering" effect. This causes bit errors. Here, you will analyze just the spectrum of a Gaussian pulse which is produced by a laser. In future assignments, you will analyze the spectral filtering effect in detail.

Find the Fourier transform of the following Gaussian signal: $e^{-at^2}$ . Plot the frequency domain spectral representation of this signal. Choose three cases a=1,a=100,a=1000. Note as the pulse compresses in the time domain its frequency spectrum increases correspondingly.

Hint: You can solve the problem analytically as well but it is recommended that you use Matlab to perform the Fourier transform numerically. The Gaussian integral formula is $\int_{-\infty}^{\infty} e^{-ax^2+bx}dx = \sqrt{\frac{pi}{a}}e^{\frac{b^2}{4a}}$ .

Playing with the above expression and comparing with the formula for the Fourier transform, you will see that the Gaussian time domain pulse has a Gaussian spectrum in the frequency domain as well. Another interesting result is that if you convolve two Gaussian functions in the time domain, you retrieve a Gaussian function in the time domain. In the frequency domain, convolution simply amounts to multiplying two

Gaussian functions after which you get another Gaussian function in the frequency domain.

## 5) Sampling - Wagon Wheel Effect

The wagon wheel/stroboscopic effect is an optical illusion in which a spoked wheel appears to rotate differently from its true rotation. The wheel can appear to rotate more slowly than the true rotation, it can appear stationary, or it can appear to rotate in the opposite direction from the true rotation. Common examples include helicopter rotors and airplane propellers. These forms of the effect are known as stroboscopic effects and they arise from temporal aliasing.
Write a Matlab algorithm to animate a rotating spot inside a circle with a certain signal frequency. As an observer with a camera, choose a sampling frequency to mimic the behavior of the spot for the following conditions:

- Nyquist Frequency >> Signal Frequency
- Nyquist Frequency > Signal Frequency
- Nyquist Frequency = Signal Frequency
- Nyquist Frequency < Signal Frequency
- Nyquist Frequency = 0.5 * Signal Frequency
- Nyquist Frequency < 0.5 * Signal Frequency

Plot and explain in detail, your observations for the aforementioned conditions

**6) AM and FM**

Imagine that you have an antenna whose bandwidth is limited but you happen to pick up some noise. The raw audio sounds like gibberish. Fortunately, you took the DTFT and got what is shown in figure #2. You deduce that there are 3 channels centered at 5-,12-,and 19-kHz, shown in blue, green, and red, respectively. You can determine the bandwidth of each by the script shown in figure #3. Your task is to apply what you have learned in ECE 30100 and demodulate the individual channels so you can hear them. The code that caused this spectrum is shown below in figure #3.

What you need to turn in:

- Rough, qualitative outline, in the form of sketches, showing what you are doing in the frequency/time domain as well as how you did it in matlab.
- What song was on channel 2, centered at 12 kHz?
- Explain the purpose of the channels being separated by 1-kHz
- What is the point of convolving the signal with a Low-Pass Filter/Band-Pass Filter (LPF/BPF)?

Getting Started:
You should create a directory in matlab that includes the 'RFspectrum.mat' file as well as the 'ece301conv.m' function.

Useful Functions

- soundsc(signal, sampling frequency)
    - This plays the audio les though the computer's speakers
    - Ex. soundsc(x,44100)
    - Sampling frequency (Fs) is 44100 for this project

- ece301conv(x1,x2)
    - This convolves two signals

- cos(2*pi*f.*t) / sin(2*pi*f.*t)
    - t = (((0-4)*Fs+0.5):((duration-4)*Fs-0.5))/Fs
    - Where Fs is the sampling frequency (44100)
    - Duration = 8 (audio les are each 8 seconds long)

- X = load('RFspectrum.mat');
    - This loads the structure as 'X'
    - X = X.H; (this grabs the audio field)

- sinc(x)
    - Refer to figure #3 to see code and comment

Hints:

- Draw out what happens in the frequency domain. You'll need to bring the channels down to the base band while ensuring the other two channels do not interfere with your signal, hence the LPF/BPF.

- You may find useful the 'ece301conv.m' file included in the zip folder. This takes as input 2 signals you would like to convolve.

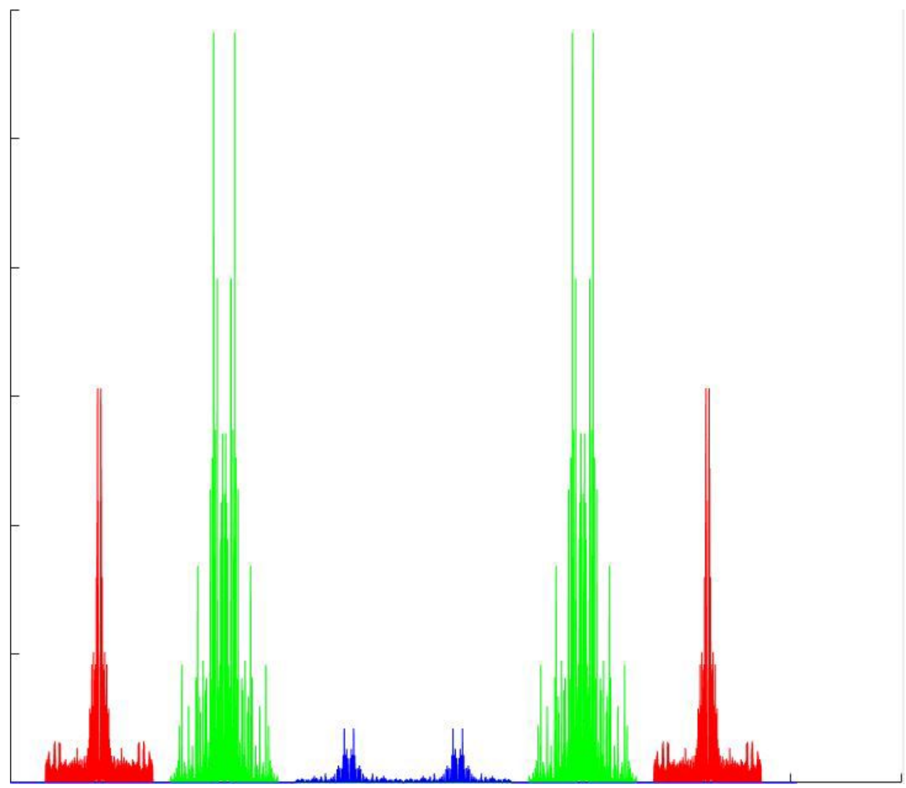


Figure 2: The radio spectrum with 3 channels spaced 1-kHz apart.

```
[y,Fs]=audioread('[redacted]mp3'); %import the audio file, Fs is the sampling frequency
[y2,Fs]=audioread('[redacted].mp3');
[y3,Fs]=audioread('[redacted].mp3');

duration = 8;
y = y(1:duration*Fs,1); %grabs the first 8 seconds
y2 = y2(1:duration*Fs,1);
y3 = y3(1:duration*Fs,1);

t = (((0-4)*Fs+0.5):((duration-4)*Fs-0.5))/Fs; %creates same-size time array as audio files

y = y'; y2 = y2'; y3 = y3'; %transposes matrix

LPF = 2*3E3*sinc(2*3E3*t); %creates low-pass filter (LPF) s.t. the audio files span 0-3kHz

y = ece301conv(LPF,y); %convolves the audio signals in time with th LPF
y2 = ece301conv(LPF,y2);
y3 = ece301conv(LPF,y3);

x = cos(2*pi*5E3*t); %creates cosine function that can upconvert the audio files appropriately
x2 = cos(2*pi*12E3*t);
x3 = cos(2*pi*19E3*t);

y = y.*x; %upconverts the signal
y2 = y2.*x2;
y3 = y3.*x3;

H = y + y2 + y3; %sums them up s.t. they are all in the same spectrum
```

Figure 3: The script that created the .mat file