**Fourier Transforms, Convolution, and Filtering**

**In the document you upload to Dropbox please answer each question below. A bullet point indicates questions.**

1. Time-bandwidth exercise. Make 1 second of a pure sine wave at 200 Hertz. Use a sampling rate of 10000 samples per second.

* What is the maximum frequency sine wave that you could create at this sampling rate? This frequency is the Nyquist frequency at this sampling rate.
* If you Fourier transformed the entire 1 second sample what would be the frequency step f?

2. Now create a Gaussian envelope that we will use to modulate the sine wave to make a pulse. The equation of a Gaussian as implemented in MATLAB is:

>>env=exp(-(t-0.5).^2/(0.02)^2);

where, t, is the time array created in part 1. The 0.5 makes the Gaussian located in the center of the 1 second time interval. The 0.02 is the width of the Gaussian. Plot t versus env to make sure you see what it looks like. Now we are going to use this envelope function to modulate the pure sine wave creating the variable pulse by forming the product

>>pulse=env.\*y;

Plot t versus pulse. Listen to it if you have sound on your computer using the command

>>sound(pulse, 10000)

(Note that the 10000 tells the sound command to use the playback rate of 10000 samples per second. What happens if you put 20000? or 5000?)

3. Now we will use the elements developed in 1 and 2 to explore the idea of the time bandwidth product. As explained in class this concept is the basis of the uncertainty principle in quantum mechanics. In the classical wave sense we say f t ~ 1. It’s just approximate as the ~ implies.

* Change the width of the Gaussian to make a series of 5 pulses with decreasing width t (from 0.1 down to 0.001 or so). For each Dt determine the corresponding Df from the Fourier transform. Plot your Df versus Dt values. Does it approximate a straight line? Note you will have to decide how you define Df. Expalin your method in answering this part. Hand in your plot.

4. You can access the file SignalNoise.wav from D2L (in the “Files” module under Content). It contains a message, but it is obscured by narrow bandwidth noise. Your task is to filter the signal so that you can recover the message. To load the .wav file into MATLAB use the command

>>[y,fs]=wavread(‘SignalNoise.wav’);

…which loads the data into the variable y; fs is the sampling rate of the .wav file.

* What are the dominant loud frequencies that are obscuring the message?
* What does the message say?
* Explain how you recovered the information in the file by filtering.