# CSC 340 Final Software Development Project and Examination Due: Tuesday, 1 May 2018 DUE: BY 11:00 AM (FIRM)

Name \_\_Brenton Belanger\_\_\_

## Signal and basic DSP Problems:

1. Common signals (24 points total):  
   For this exercise you will be demonstrating your understanding of common periodic functions that are synthesized from the trigonometric sine function by modifying its amplitude, frequency or phase. Since you are creating these complicated functions from simple ones, you will know what to expect from transformed representations that reveal amplitude, frequency or phase content of a periodic function.
   1. Using S terms, with S=3, 10 and 50, from each of the series below and at least 512 regularly spaced samples of t, where t ε [0, 1), generate plots of graphs of the following functions:
      1. (3 points; 1 point for each function, with all three plotted on a single chart)
      2. (3 points; 1 point for each function, with all three plotted on a single chart, not the same chart as used for the previous question)
   2. Create and plot the power spectral density (PSD) estimates for the functions *f50* and *g50.* You need only display the positive frequencies in the PSD. (6 points)
   3. Observe that, as more terms (S=3, S=10, S=50,…, etc.) are used for approximating the functions fs and gs, the graphs appear to be approaching limit functions *fL* and *gL*.
      1. Provide a verbal description of the graphs of the two limit functions (3 points).

Function f takes on a “square wave” sound with horizontal limits approaching 0.8 and -0.8, and vertical limits at ever 256 samples. This oscillation shows that the sound is either on at the limits, or off between the limits. Function g takes on a “sawtooth wave” sound with limits at 0.8 and -0.8, with a vertical limit at every 256 samples. This style, however, is a pulse and decay, where the pulse is large and loud and decays on a cycle.

* + 1. Generate your own data to estimate the limit functions *fL* and *gL* at 512 points, and then overlay plots of their graphs onto your plot of the graphs of *f50* and *g50* (6 points)?
  1. Create and plot PSD estimates for the functions *fL* and *gL* (3 points).

1. Several in-class examples, and the functions described in Problem 1, employed ***sums*** of signals of the form *v(t) =* ***a*** *sin(2π* ***f*** *(t-****c****))*, where ***a*** is the *amplitude*, ***f*** is the *frequency*, ***c*** is the *phase* shift of the signal, and *t* ε [0, 1). Consider two sine functions v1(t) and v2(t), where both v1(t) and v2(t) have the same amplitude (***a***=1) and the same phase shift (***c***=0). Also assume that the frequency for v1(t) is ***f****1* = 13 and for v2(t) assume ***f****2* = 31.  
   ***Compare and contrast*** the power spectral density (PSD) estimates for two signals x(t) and y(t), where x(t) = v1(t) + v2(t), the ***sum*** of the two sine functions, and y(t) = v1(t) \* v2(t), the ***product*** of the same two sine functions. Generate 512 regularly spaced samples for both signals *x(t)* and *y(t)*, where *t* ε [0, 1), and then calculate the PSDs for x(t) and y(t)*.* (6 points total)
   1. With respect to the ***existence*** of peaks, how are the two PSDs ***similar*** (3 points)?

Both sets of peaks are equal in height within their respective functions. As well, there are four peaks per function.

* 1. With respect to the ***locations*** of the peaks, how do the two PSDs ***differ*** (3 points)?

The peaks in x(t) are closer together than the peaks in y(t). Also, the peaks in x(t) are closer to the extremes of the samples than y(t).

1. The purpose of this exercise is for you to explore and demonstrate the effect, if any, that changes in *phase* have upon the PSD. (6 points total)
   1. Suppose that 256 consecutive samples of a signal contain a brief pulse, represented by a single 1 and 255 zeroes. Describe how the PSD estimate is affected by varying the phase of the pulse, that is, by varying the time at which the pulse, represented by the single 1, occurs among the 256 samples (2 points)?
   2. Suppose that ***another*** signal consists of a pure sinusoidal tone, *h(t) = sin( 20 t)* on the interval [0, 1).
      1. How is the discrete Fourier transform affected by introducing phase shift 1>***c***>0, so that *h(t) = sin(20t-****c****))* (2 points)?
      2. How is the PSD estimate affected by the phase shift (2 points)? Try several values for ***c*** and then report the effects, if any.
2. For this problem, use either *f50*or *g50* (***but not both***) from problem 1a. Thus, this problem description assumes that exactly the first 50 terms of a summation were used in forming the function. Construct ideal filters and apply them to the signal to as follows (12 points total):
   1. Use a low-pass (LP) filter that ***passes*** only the lowest seven frequencies into a filtered signal.
   2. Use a high-pass (HP) filter to generate a filtered signal in which the lowest seven frequencies have been ***suppressed*** and the upper 43 frequencies have been ***passed***.
   3. Use a band-pass (BP) filter to generate a filtered signal in which only the 5th through the 8th frequencies have been ***passed***.
   4. Use a notch filter to generate a filtered signal in which only the 5th through the 8th frequencies have been***suppressed***.
   5. Plot the original signal with each of the four filtered signals (8 points). How could you check your plot to know that it is correct without comparing it to the work of others?
   6. Provide a verbal description of how the original signal is affected by each of the four filtering processes (4 points):
      1. LP?
      2. HP?
      3. BP?
      4. Notch?
3. For this exercise, you will be using DSP to decode DTMF tones (a.k.a., a practical example from which you benefit many times every day). (10 points/5 points each)  
   In the directory P3data, you will find two text files, each containing integer representations of 4096 samples of a DTMF tone. The sampling frequency, ***f***s, was 44.1 kHz, i.e., there were 44,100 samples per second. Use the data in the files to determine what DTMF tone was used to generate the tone in each file. Recall that the center frequency of bin ***k***, is given by ***f***k = (***k***\****f***s)/***N***, and bins are counted beginning with the initial entry being bin zero, i.e., ***k*** = 0.
   1. toneA1 corresponds to DTMF key = \_\_\_\_\_\_\_\_\_ (Be sure to include your evidence and explanation thereof or there will be no credit.)
   2. toneB1 corresponds to DTMF key = \_\_\_\_\_\_\_\_\_ (Be sure to include your evidence and explanation thereof or there will be no credit.)
4. Correlation and convolution (20 points total)  
   In this section, you will use your digital signal processing tools to perform two common tasks: one, determining the range to a target by a process that is similar to what an echo-locating dolphin does physiologically; and two, smoothing a noisy signal:
   1. Assume that an acoustic pulse is transmitted in sea water and the velocity of sound in sea water in the area is approximately 1500 m/s (depending upon a myriad of factors such as depth, temperature, and salinity). Estimate the range to the primary reflector given the pulse and return signals given in “rangeTestSpring2018.txt” file. Assume that the receiver is turned off for 2 milliseconds after the pulse is transmitted, 50 kHz sampling rate, and the return signal measurements were taken from the first 1024-sample window after the receiver begins listening again. How far away is the primary reflector (10 points)?
   2. Use FFT convolution to smooth the returned signal with a 6-point filter (p=6). On a single chart, centered on the pulse, plot the first 256 consecutive samples of the original signal and the smoothed signal (10 points).
5. Two-dimensional FFT (22 points total)  
   In this task you will be using your digital signal processing tools to locate a simple target in an uncluttered image. Specifically, you will be applying a two-dimensional fast Fourier transform to correlate a two-dimensional pulse (represented by a C-shaped, block figure occupying part of a rectangular region that is 30x120, WxH) with the data in a large (512x512) image.  
     
   You may use the Picture class, linked to the course web site, or the Python PIL to create or render images for this problem.
   1. Set-up (6 points)
      1. Generate and display ***a test signal*** ***(return)*** corresponding to a monochrome image, a 512x512 array of pixel values as follows:
         1. Set all values to zero;
         2. Beginning at row 180 and column 220, change the entries to create a region R in the array having 110 columns (width) and 140 rows (height) whose values are 255;
         3. As illustrated below, overwrite a portion of the rectangle R by creating a 30 wide by 90 tall sub-rectangle whose pixel values are all zero (hence, its pixels are black).
         4. On a monochrome display, the resulting image should be similar to the image above with black, corresponding to pixel values of 0. There is a “C”-shaped region that occupies most of a 110Wx140H rectangular region that is white. White pixels correspond to pixel values where the components r=g=b=gray-level=255. Also, there is a 30Wx90H, black rectangular region centered top-to-bottom on the right side of R.
      2. Similarly, generate and display a 30Wx120H, white, test pulse, in the top left corner of the image as illustrated below. The horizontal bars of the “C”-shaped block letter should be 15 white pixels tall and the vertical bar should be 15 pixels wide. Pad the remainder of a 512x512 array with black pixels.
   2. Find and display the two-dimensional correlation of the signal and the test pulse (8 points).
   3. Create and render a display of the correlation magnitude, painting those locations whose correlation is within 10% of maximum with red (8 points).   
      Note: For both parts b. and c., you will likely need to scale (linear of logarithmic) the correlation magnitudes. Consider using a logarithmic scaling followed by a linear scaling to the range [0, 255] for the pixel values. Also, when exploring a logarithmic scaling, remember that the correlation magnitudes may include values that are outside the domain of the log function (non-positive values); accordingly, you may (or will) need to translate the correlation values into the domain of the log function.