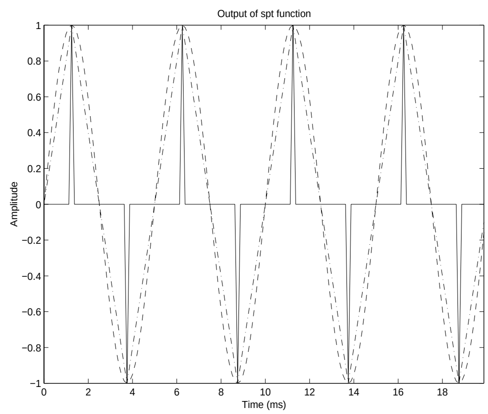
EE 381K **DIGITAL SIGNAL PROCESSING** Spring 1998

COMPUTER HOMEWORK ONE

Due: Monday, February 23, 1998

Note: Please make use of the MATLAB tutorial when doing this homework! You must **turn in** all your MATLAB code, including functions; **failure to do so will result in a zero grade**. Try to be reasonable with the amount of paper you use; don't use a cover sheet, and, when asked to plot several graphs on one page, do so.



1. Write a MATLAB function called `spt` that accepts a time vector as input and returns three vectors of the same length. The first vector returned is the sine of the input. The second vector is a triangle wave with peaks at the peaks of the sine wave, and of the same sign. The third vector is a pulse wave equal to +1 at the sine wave peaks, –1 at the sine wave troughs, and zero everywhere else. So, with an input of `0:pi/16:127*pi/16`, the outputs would be as shown at the left. Note that the sine wave may not equal exactly ±1 at its peaks, and therefore searching for ±1 to construct the pulse wave will not work for all time vectors.

2. Assuming a sampling frequency of 8000 Hz, **generate** a time vector of the correct length to contain *exactly* four cycles of a 200 Hz signal, as in the plot above. Use your function `stp` to create four cycles of a sine wave, triangle wave, and pulse wave, all with a fundamental frequency of 200 Hz. Plot the three signals against time on the same axes, and label the plot. Your plot should appear exactly as above. **Turn in a hard copy.**

3. Compute the frequency content of your three signals using `freqz` over 80 points. Use `help freqz` to find out how to use it correctly. Note that `a=1` here. Using `subplot`, plot the magnitude (absolute value) spectra of the three signals in three separate plots. Label the frequency axes correctly (the frequency is returned by `freqz`). **Turn in a hard copy**. If your axes crash into one another on the printout, use `set(gcf,'paperposition',[1 1 6 9])` before printing to make the plot larger. **Explain** the appearance of the three plots.

4. Now **create** a filter `f` using the command `f=remez(9, [0 400 800 4000]/4000, [1 1 0 0]);`. (Don't worry about why this works for the moment.) Using `freqz`, **compute** the frequency response magnitude of this filter over 256 points from 0 to 4000 Hz, and **plot it**. **Describe** the filter, and **turn in** a hard copy of the frequency response.

5. Filter the three signals you created in **2** with this filter using `conv`. Look at the output for the sine wave on the screen. You will see the sine wave is essentially intact, but has acquired 'tails' at each end. **Extract** exactly three full cycles of the filtered sine wave from this vector. (Note: If you name the extracted signal `s`, then `plot([s s])` will provide a simple check of whether you have extracted exactly three cycles. You should see six full cycles of the sine wave, with no obvious join.) **Extract** three full cycles of the other filtered waves in the same way, and **turn in a plot** of the three signals in three separate, labeled `subplot`s.

6. Compute the frequency content of these three filtered signals using `freqz` over 60 points, and plot them as you plotted the pre-filtered frequency responses in **3**. With the help of these plots, **explain** the appearance of the time domain plots from **5**. **Explain** how the filter has turned the triangle wave into something resembling a sine wave. **Speculate** on the result of multiple applications of the filter to the pulse wave.