Homework Set 2

Due by 12:00 noon on Wednesday, Oct. 11, 2006 in 16-352.

Here we will look at power spectra in MATLAB, and some issues associated with discrete-time Fourier analysis. You should save your MATLAB code in an m-file, which will make it easy to modify and reuse it, and submit your m-file along with your plots.

Remember, when in doubt about any of the MATLAB commands given below, use the `help` command to get info about how to use each one – the Windows installation of MATLAB will have this same info and more accessible via the Help menu.

1. Use MATLAB to generate a time vector $y(t)$, approximately one second long, with a sampling frequency f_{samp} of a few kHz (the syntax "`vector=start_value:increment:end_value;`" will be useful here — `increment` is the time between your samples or $1/f_{samp}$).
   Then create a sinusoid based on that time vector — choose a frequency f_{SIN} of a few hundred Hz:
   $$y = sin(\omega_{SIN} t) = sin(2\pi f_{SIN} t)$$
   Calculate the PSD of your sinusoid using the `pwelch` [1] command, and plot it on a linear scale and a logarithmic scale; use the `plot` and `loglog` commands, respectively. What is the significance of the highest and lowest frequencies that appear on the plot?

2. An important feature of `pwelch` is that it always correctly normalizes the total power of the PSD, but—depending on the parameters you use—you can get quite different PSD shapes for the same signal. For example, the default parameters give a lot of *spectral leakage*. To see a low-leakage spectrum, try running `pwelch` with the `nfft` and `window` parameters satisfying the following conditions:
   - the sinusoid frequency f_{SIN} is an integer multiple of the quantity f_{samp}/N_{fft}
   - `window` is the same length as `nfft`.

   Can you suggest what causes spectral leakage? Plot the low-leakage PSD on the same axes as your previous linear and log plots (use `hold` after plotting one waveform to freeze a figure, before plotting a second). Considering the relative magnitudes, how much does the leakage matter?

3. Recall that one of the consequences of Parseval's theorem is the following relationship between a time-domain signal $f(x)$ and its PSD $F(\omega)$:
   $$\left\langle f(x)^2 \right\rangle = \int_0^\infty F(\omega) d\omega.$$
   Verify that this is the case for the sine wave you've been using by computing its mean-square value in the time domain, and the integral of its PSD in the frequency domain (MATLAB's `var` and `sum` functions will be useful here). Remember that you're effectively calculating an area, and make sure that units match up: `pwelch` gives you the PSD in units^2/Hz, while the integral of your PSD needs to be equal to a mean-square value (units^2).

[1] The syntax is `[PSDvect, freqvect]=pwelch(signal,window,n_overlap,nfft,f_samp)` and only requires you to supply a `signal` vector and a number for `f_samp` to properly scale and calculate the frequencies for the PSD – i.e. `pwelch(signal,[],[],[],fsamp)` will get you a result. The result is stored in the two vectors before the = sign. Use MATLAB's `help` to find out what the other parameters do. For parameters that you leave out by entering "`[]`", MATLAB uses its defaults (also found in help).

4. Now use the `randn` command to generate a noise signal with the same length as your time vector. Calculate its PSD with the `pwelch` parameters of your choice, and plot it on a new log plot. To observe the benefits of averaging, generate a noise signal that is 10× longer in duration, calculate its PSD, using the same `pwelch` parameters, and plot its spectral density on the same plot.

5. Take the sinusoid from part (a), but with its amplitude reduced tenfold, and the first (short duration) noise signal from part (d), and add them together. Look at a section of the summed waveform in the time domain - can you find the sine wave at all? Now plot the PSD of the combined signal - can you find the sine wave peak in the noise? What can you do to get it to resolve more clearly? (Part (d) should provide a clue.) Plot your result on the same axes.

6. ***(BONUS)*** Finally, take the original sinusoid from (a), compute its Discrete Fourier Transform using the `fft` command, and plot its magnitude (an FFT is complex-valued) on a semi-log plot (see the `semilogy` command). Compare this to the PSD of the sine wave given by `pwelch`, and comment on the FFT's features: Why two peaks? What is the meaning of its x-axis values? Why are the values along the y-axis so large?