Music 421A
Winter 2011-2012
Homework #5
Spectral Peaks + Noise
Due in one week

Theory Problems

1. (10 pts) Suppose we are going to analyze a sinusoid whose frequency is $f_0 = 440$ Hz using a rectangular window. Assume the window length is $M = 255$ and the sampling rate is $f_s = 8192$ Hz.
   (a) (3 pts) What is the relative error (*i.e.,* $\Delta f/f_0 = |f_0 - \hat{f}|/f_0$) between the actual frequency (f_0) of the sinusoid and the peak frequency ($\hat{f}$) when there is no zero-padding?
   (b) (3 pts) Repeat part (a) with the zero-padding factor of 5.
   (c) (4 pts) Repeat part (a) with the zero-padding factor of 5 and parabolic interpolation using the peak and its two neighbors. You don't need to build algorithms for peak finding or parabolic interpolation for now. Just find the peak and its neighbors graphically.

2. (7 pts) **Least Squares Sinusoidal Parameter Estimation**
   (a) (5 pts) Use the *orthogonality principle* to construct an optimal least-squares estimator for the amplitude A and phase ϕ of a sinusoid having known frequency ω_0 in the presence of another sinusoid at some unknown frequency. That is, the measurements consist of N samples of the following signal
   $$x(n) = \mathcal{A}e^{j\omega_0 n} + s(n)$$
   where $\mathcal{A} = A\exp(j\phi)$ and $s(n)$ is an unknown *interferer* (sinusoid at some other unknown frequency ω_i, phase ϕ_i, and amplitude A_i).
   (b) (2 pts) Under what conditions is the estimate $\hat{\mathcal{A}}$ unbiased?

3. (2 pts) Let $e(n)$ denote a sample of white noise. Then its power spectral density $S_e(\omega)$ is constant from $-\pi$ to π. By the stretch/repeat DTFT theorem, the signal $x = \text{STRETCH}_2(e)$ also has a constant power spectral density, yet every other sample is constrained to be zero. Is $x(n)$ also white noise? Explain.

4. (8 pts) Find the (cyclic) unbiased autocorrelation of the following sequences in $\mathbf{R}^8$:
   $$\begin{array}{llll} (a) & [1,0,0,0,0,0,0,0] & (b) & [1,1,0,0,0,0,0,0] \\ (c) & [1,1,1,1,0,0,0,0] & (d) & [1,1,1,1,1,1,1,1] \end{array}$$

5. (5 pts) Spectrum analysis of a weighted moving average process (FIR filter). To smooth or average out a time sequence, we may use a *weighted moving average*. Given the discrete time triangular filter $h(t)$
   $$h(t) = \begin{cases} 4 - |t|, & |t| < 4 \\ 0, & |t| \geq 4 \end{cases},$$
   we need to investigate the frequency response so as to characterize how the filter will respond to a noisy input sequence.
   (a) (2 pts) Compute the acyclic biased autocorrelation of $h(t)$.
   (b) (3 pts) Compute the biased power spectral density of $h(t)$.

Lab Assignments

1. (20 pts) **Spectral Peak Estimation**

   (a) (10 pts) Construct a program to find the frequencies (in Hz) and the magnitudes (in linear amplitude) of the positive-frequency peaks of an input DFT of a given signal, using the code skeleton shown below. Be sure to convert matlab frequency index numbers to frequencies in Hz.

```
function [peaks,freqs]=findpeaks(Xwdb,maxPeaks,fs,win,N)
% peaks = a vector containing the peak magnitude estimates (linear) using
%         parabolic interpolation in order from largest to smallest peak.
% freqs = a vector containing the frequency estimates (Hz) corresponding
%         to the peaks defined above
% Xwdb  = DFT magnitude (in dB scale) vector of a windowed signal.
%         NOTE that it may contain
%         only low-frequency (length < N/2+1), positive-frequency
%         (length = N/2+1), or all (length = N) bins of the FFT.
% maxPeaks = the number of peaks we are looking for
% fs       = sampling frequency in Hz
% win      = window used to obtain Xwdb (assumed zero phase)
% N        = NFFT, the number of points used in the FFT creating Xwdb

%-- Find all peaks (magnitudes and indices) by comparing each point of ---%
%-- magnitude spectrum with its two neighbors ---%
allPeaks = [];
for i=2:length(Xwdb)-1
    ...
    ...
end

%-- Order from largest to smallest magnitude, keep only maxPeaks of them --%
peaks = ...

%-- Do parabolic interpolation in dB magnitude to find more accurate peak --%
%-- and frequency estimates --%
for i=1:maxPeaks
    idx=find(Xwdb==peaks(i));
    %parabolic interpolation
    a=Xwdb(idx-1);
    b=Xwdb(idx);
    c=Xwdb(idx+1);
    ...
    ...
end
%-- Return linear amplitude and frequency in Hz --%
% NOTE that we must use knowledge of the window to normalize amplitude here
% if we have a TD cosine of amplitude 0.6, this output should be 0.6
peaks = ...
freqs = ...
```

   (b) (4 pts) Test your `findpeaks.m` function on a length-255, 400-Hz cosine having amplitude 1 and no phase offset, i.e.,

$$x(n) = \cos(2\pi \cdot 400nT), \quad n = -127, \ldots, 127.$$

   (Implicitly, a rectangular window of the same length is used here). Use your zero-phase zero-pad window function to extend the signal to length 2048 before peak detection. Let the sampling rate be $f_s = 8000$ Hz. Show your plot of the magnitude spectrum with the found peak clearly marked. Also plot the phase spectrum. The spectrum should be real because we zero-phase windowed a cosine, which is even (*i.e.*, the FFT of a real, even sequence is real and even). Thus, the phase should be $\pm\pi$. (Matlab may produce some "imaginary dust" in the spectrum due to round off error.) Here is some code to get you started:

```
subplot(211);
plot(abs(fft(zpzpwin(cos(2*pi*400/8000*(-127:127)'),boxcar(255),2048))));
subplot(212);
plot(angle(fft(zpzpwin(cos(2*pi*400/8000*(-127:127)'),boxcar(255),2048))));
```

(c) (3 pts) Download `s1.wav`[1] and find the amplitudes and frequencies of each sinusoidal component in the signal. You need not zero-phase window the signal, nor should you zero-pad it. Just use a rectangular window whose length is the same of the signal. To avoid getting NaN in your spectrum due to the log of zero, use something like `Xwdb = 20*log10(abs(Xlinear)+eps);` .

(d) (3 pts) Again, with `s1.wav` but now using the Hamming window of length equal to that of $x(n)$ (255 samples) and with the same amount of zero-padding (to length 2048), find the frequency and amplitude estimates. Do you expect the estimates to be better in the case of Hamming window here? Why or why not?

2. (14 pts) **Simple Pitch Estimation:** In this problem, you will be estimating the pitch of an oboe sound, using the `findpeaks` function written previously. This problem is also concerned with the *resolvability* of the spectral peaks under different windows.

   (a) Download the sound file

   (b) Download `oboe.ff.C4B4.wav`[2] and similarly the Matlab files
   `oboeanal.m`[3] and `dbn.m`[4] into your directory. The posted version of `oboeanal.m` should be compatible with your your `findpeaks()` function prototype.

   (c) (4 pts) Run the program `oboeanal.m`. Make sure you have your `findpeaks` function working already. Verify that the pitch estimate agrees with the plot. Explain what peak is used to estimate the pitch.

   (d) (4 pts) Comment on the resolvability of peaks for the three windows.

   (e) (6 pts) Modify the code on line 54, changing K to $(K-1)$ and then $(K+1)$ in order to vary the length of the signal under the windows. (NOTE: this line of code is after the window name has already been chosen, and currently contains only a comment line. Do NOT change K before this comment line. Note that some of the plots may then extend above 0 dB. Do not worry; you can zoom in on them anyway.) Comment on the resulting resolvability. Which case is more resolvable and amenable to spectral peak analysis?

[1] `http://ccrma.stanford.edu/~jos/sasp/hw/s1.wav`
[2] `http://ccrma.stanford.edu/~jos/sasp/hw/oboe.ff.C4B4.wav`
[3] `http://ccrma.stanford.edu/~jos/sasp/hw/oboeanal.m`
[4] `http://ccrma.stanford.edu/~jos/sasp/hw/dbn.m`