UNIVERSITY OF ILLINOIS AT URBANA-CHAMPAIGN
Department of Electrical and Computer Engineering

ECE 498MH Signal and Image Analysis

Homework 5

Fall 2013

Assigned: Friday, October 11, 2013 Due: Friday, October 18, 2013

Reading: SPF Chapter 7 and Sections 8.1-8.5

Problem 5.1

Your friend Leo is designing a flying dragon. In order to make it fly more smoothly, he proposes to smooth all of the motor commands by passing them through the following difference equation:

$$y[n] = x[n] + 0.5x[n-1] + 0.25x[n-2] + 0.5y[n-1] - 0.25y[n-2]$$

where $x[n]$ is the original motor command, and $y[n]$ is the smoothed motor command.

(a) Does this difference equation implement an IIR or FIR filter?

(b) Compute samples of the impulse response $h[n]$ for $n = -1, 0, 1, 2, 3$ by hand, by applying the input $x[n] = \delta[n]$.

(c) Find the transfer function, $H(z) = Y(z)/X(z)$.

(d) Find and plot the poles and zeros.

(e) Find the frequency response, $H_d(\omega)$.

(f) Sketch the magnitude of the frequency response.

(g) It is a lowpass, highpass, bandpass, or bandstop filter?

Problem 5.2

Consider the FIR filter $h[n] = 0.25\delta[n+2] + 0.5\delta[n+1] + \sqrt{3}\delta[n] + 0.5\delta[n-1] + 0.25\delta[n-2]$.

(a) Calculate the frequency response, $H_d(\omega)$, of this filter. Note that, because it is symmetric in the time domain, the frequency response of this filter can be written as the sum of three cosines, and that is a useful way to write it.

(b) Calculate and sketch the magnitude and phase of this filter (hint: find $|H_d(\omega)|$ and $\angle H_d(\omega)$ for a few magic angles like $\omega = 0, \frac{\pi}{4}, \frac{\pi}{2}, \frac{3\pi}{4}, \pi$. Assume that the values at other frequencies are smoothly interpolated between those values).

(c) Is it a lowpass, highpass, bandpass, or bandstop filter?

Matlab Exercises

Problem 5.3

The signal file `chorus.dat` contains five seconds of a frog chorus digitally recorded with a sampling rate of 20 kHz at the Cibolo Nature Center in Boerne, Texas on the evening of March 21, 2007. The recorded signal has been corrupted by a (synthetic) sinusoidal interference. Your goal is to remove this interference using a notch filter so that you can hear the natural sounds. Download the data and the Matlab script `frogStart.m` and the Matlab function `notchStart.m`; these scripts need only a few changes and additions to complete the assignment, so you may want to start with them. Using your understanding of spectral analysis, determine the appropriate frequency for the notch to remove the interference. Use a pole magnitude of 0.98. Alter the notchStart.m function to compute the coefficients of your notch filter and to implement it via the appropriate difference equation. Places where you may need to edit the code are marked with XXX.

After the script runs, you can use the commands soundsc(chorus,20000) and soundsc(y,20000) to hear the results. (Note the short transient response from the filter at the beginning of the output; you may enjoy experimenting with different pole magnitudes to study the tradeoffs and to find the best value.) Print and turn in your figures with your results.

Problem 5.4

The signal file `spikeburst.dat` contains two seconds of raw data from a single-channel neural recording (with a sampling rate of 40 kHz) from the trigeminal nucleus in a rat's brain as it responds to whisker stimulation. This data is provided courtesy of Dr. Aniket Kaloti and Professor Mitra Hartmann from Northwestern University, and they should be acknowledged in any use of the data. Load and plot the data; you will observe huge low-frequency fluctuations in the signal around the stimulation time. (Those big bumps aren't spikes, but unwanted electrical signals or artifacts!) Apply a first-order IIR notch filter to the data (notch at DC = zero frequency) to filter out the low-frquency fluctuations to leave the spike bursts. Experiment with different pole magnitudes to make the best visual tradeoff in terms of reducing the low-frequency baseline fluctuations while preserving the spike burst. You may find it easiest to modify your code from Part 1 (I recommend saving both for future use!) Print and turn in your before-and-after plots.

The Matlab script `filterStart.m` loads the neural spike data, generates several IIR filters, and filters the data. Run the script, and plot the outputs. Comment on differences between bandpass filters and notch filters.