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

ECE 498MH Signal and Image Analysis

Lab 5

Fall 2014

Assigned: Thursday, October 23, 2014 Due: Thursday, October 30, 2014

Reading: *Fundamentals of Signal Processing* by Minh Do, Sections 1-3 and 1-4

Lab 5.1

(a) 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.

Your `runlab.m` function should call your modified versions of `frogStart` and `notchStart`, then 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.)

(b) 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!)

Create a function `spikeStart.m` that generates new figures (diferent from the figures in part (a)), then populates them with the spectra and waveforms of original and filtered data. The filter, in this case, should be a first-order IIR notch filter (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 also find it useful to try the Matlab script `filterStart.m` which 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.