EE-3221 LABORATORY

Notch Filters and Interference Removal

IMPORTANT: This is an INDIVIDUAL lab assignment. Each student must complete and submit their work independently.

Goal – Design a multistage digital filter to remove interfering tones from a signal.

Materials

- Laptop computer with MATLAB.
- Earbuds or similar are recommended. The speakers on your MSOE laptop likely reproduce frequencies down to 300 Hz acceptably, are noticeably quiet at 250 Hz, and have virtually no audible output at 200 Hz. Many headphones and earbuds can acceptably reproduce low frequencies down to 30 Hz, so headphones are preferred for this lab.

Overview:
Design a filter consisting of cascaded notch filters to improve an audio signal that is corrupted with interfering tones. Perform a DFT on the given signal to determine the spectral location of the interfering tones.

Download the audio files corresponding to your MSOE user name. The files can be found on the course website. The commands for loading and listening to audio in MATLAB are:

```
[x, fs] = audioread('filename.wav');
sound(x, fs)
```

Each audio file contains ***three*** interfering tones. The interfering tones are below 3000 Hz.

Recall that digital frequenices (Ω, with units radians/sample) can be related to continuous-time frequencies (f, with units Hz) via the following expression

$$\Omega = 2\pi f/fs$$

where fs is the sampling frequency. The audio files are sampled at 44100Hz.

You may use MATLAB's built-in FFT function for computing the DFT. Remember that the spectral spacing (resolution) between DFT points for an M point DFT is 2π/M. You will need enough resolution to accurately identify the frequencies of the interfering tones.

Once you have found a tone to notch out, record the value for the digital frequency, Ω_o, and construct a filter using the general form for the second order notch filter

$$\boldsymbol{H(z) = \frac{G(z - e^{j\Omega_0})(z - e^{-j\Omega_0})}{(z - re^{j\Omega_0})(z - re^{-j\Omega_0})} = \frac{G(1 - 2cos(\Omega_0)z^{-1} + z^{-2})}{1 - 2rcos(\Omega_0)z^{-1} + r^2z^{-2}}}$$

where G is a gain factor used to normalize the filter and r controls the width of the notch. It is safe to ignore the effect of G. Use "freqz" to see the frequency response of your filter. It is suggested to set the scale to hertz (consult the MATLAB documentation if needed). Notice that the passband gain of the filter is very close to 1 (0 dB). You will need to experiment with various values of r to ensure that the notch is

sufficiently narrow. For this lab, keep your notch filter widths (measured between the -3dB points of the notch) to no wider than 10Hz. You may use a trial and error process to determine r.

Apply the filter using:

```
x1 = filter(b,a,x);
```

Compute the DFT of this filtered signal and inspect the result, comparing to the DFT of the original audio signal.

Repeat this process to remove the remaining two tones.

Save the final audio with

```
audiowrite('filename_clean.wav', y, fs)
```

where y is the final output sequence signal.

Submit:

1. Provide an Executive Summary. A good executive summary tells the reader:
    a. what was done
    b. why it was done
    c. how it was done
    d. what were the key results
    e. what do the results mean
2. Submit a magnitude spectrum plot of the DFT for the original signal. Clearly label all axes.
3. Submit a table listing the interfering tones, both in radians/sample (Ω) and Hz (f), present in the original signal.
4. Submit the magnitude response plots for your notch filters, along with the corresponding filter coefficients used to remove these tones. Briefly discuss the design, including choice of r
5. Submit a magnitude plot of the DFT for the clean signal. Comment on the results.
6. Submit your complete MATLAB code
7. Attach the clean audio file in .wav format to your email submittal.