University of Massachusetts Dartmouth
Department of Electrical and Computer Engineering

ECE 320 DISCRETE-TIME LINEAR SYSTEMS

Fall 2013

MATLAB **Project 2**

DT Fourier Transform and Filtering

Issued: Wednesday, 23 October 2013 **Due:** Thursday, 14 November 2013

The ground rules for this project are the same as Project 1.

Use the solutions distributed for Project 1 as a model for your report for this project. Please make sure you include detailed minutes of whic group members met when, and what they did at the meeting. Also, make sure that any analytic solutions are included in the report, not as handwritten appendices.

Remember that non-participating group members may be fired from the group after one warning as discussed in the course information handout distributed at the first class meeting. Students fired from a group must either convince another group to hire them, or complete the entire project on their own.

Complete Tutorial 3.2 on `freqz` before doing the projects below.

Part 1: Buck, Daniel, & Singer, Project 5.1, parts (a)–(e).
Part 2: Buck, Daniel, & Singer, Project 6.3, parts (a)–(d).
Part 3: New project below.

In this project, you will use filters to remove noise and annoying tones from speech. Each group has different data for this part, so you will need to begin by getting your group's data file from the Resources section of the course web. Your are looking for a Matlab file named `groupx.mat`, where `x` is your group number.

(a) Load your group's data into Matlab. You should have a single variable `y`, which represents the DT signal $y[n]$. This signal is based on speech sampled at 10 kHz. Use the `soundsc(y,10000)` command to listen to it. If `soundsc` isn't working on your computer, save the file to a `.wav` file using the following command:
`>> wavwrite(y/max(abs(y)),10000,16,'noisy.wav')`
where `noisy.wav` is the filename where you wish to save it. You should mainly hear noise and a tone, with some speech far in the background. **Warning: If you are using headphones, make sure that the volume isn't too loud before playing it.**

Using what you learned from Project 5.1, plot the magnitude spectrum $|Y(e^{j\omega})|$ for the frequency range $-\pi \leq \omega < \pi$. Make plots on both a regular and dB scale for $|Y(e^{j\omega})|$. Using `text` or `gtext`, label each region of the spectrum indicating whether it contains noise, speech, or a tone. These plots will help you later in designing your filters to make the speech signal more intelligible.

(b) A common technique for removing an unwanted tone is to apply a notch filter. A notch filter allows most frequencies to pass through, but has a gain of zero at one frequency to remove that undesired frequency. The impulse response

$$h_{notch}[n] = \delta[n] - 2\cos(\omega_n)\delta[n-1] + \delta[n-2]$$

is one implentation of a notch filter. Find $H_{notch}(e^{j\omega})$ analytically, and specify which frequency or frequencies have $H_{notch}(e^{j\omega}) = 0$.

(c) Using the results from part (b) and the plots from part (a), define a vector `hnotch` for the impulse response of a filter which will remove the tone. It might take a few tries for you to get the notch frequency exactly right. You might need to use `zoom` or `axis` to home in on your plots from part (a). Plot the frequency response of `hnotch` on a dB scale. Does this match what you would predict from your analytic answer in the previous part?

(d) Use `filter` with your impulse response `hnotch` to remove the tone from the signal `y`. Define the output from this filtering to be `r`. Listen to the resulting signal $r[n]$ and plot its spectrum $R(e^{j\omega})$. Is the tone gone? If not, you may need to adjust the notch frequency you used to define `hnotch`.

(e) Based on your plots from part (a), describe an ideal filter to remove the noise corrupting the speech signal. Based on your plot from part (a), what cutoff frequency will you use for your filter? Use `fir1` to design a 101-point FIR filter to attenuate the noise. The command

```
>> hlpf = fir1(100,alpha);
```

will return a 101-point impulse response for a lowpass FIR filter with cutoff `alpha`π. For example, to make a filter with cutoff $\omega_c = \pi/2$, set `alpha` $= 0.5$. Plot the frequency response of your filter to confirm that it matches what you think it should be.

(f) Filter your tone-free signal `r` with your new impulse response `hlpf` and define the new output as `s`. Listen to the result. Is the speech easier to understand? Plot the spectrum $S(e^{j\omega})$. Is the noise completely removed?

(g) Using the FIR filters you found above, define a single impulse response `hcombo` for an FIR filter which will remove the noise and the tone in a single filtering operation. Plot the frequency response $H_{combo}(e^{j\omega})$. Does the frequency response indicate that it will have the desired effect? Filter the original signal `y` with `hcombo` and listen to verify that the new output `s2` has both the tone and noise removed.