ECE 535 DIGITAL SIGNAL PROCESSING

Matlab Project 3

Spring 2003

Issued: Friday, April 18, 2003 **Due:** Tuesday, May 6, 2003

The purpose of this project is to give you some experience with two filter design methods: window design and frequency sampling. In addition, it will give you some practice calculating frequency responses using the `fft` function.

Your solutions to this assignment should include plots and Matlab code for each exercise along with any necessary explanations or answers to questions. The latter may be handwritten or typed, whichever you prefer.

1 Preliminaries

In these exercises you will use the `fft` function to compute frequency responses. Recall from class that an $N-$point FFT of a finite-length signal $x[n]$ contains N samples of the Fourier transform of $x[n]$, *i.e.*, N samples of $X(e^{j\omega})$.[1] These frequency samples are between 0 and 2π. In other words the frequency vector associated with the samples is $\omega = 0, \frac{2\pi}{N}, \frac{2\pi(2)}{N} \cdots \frac{2\pi(N-1)}{N}$.

Sometimes it is useful to plot the frequency response between $-\pi$ and $+\pi$, instead of 0 to 2π. Recall that the Fourier transform is periodic, thus the samples between π and 2π are identical to those between $-\pi$ and 0. Matlab provides the `fftshift` command to facilitate this type of plot. `fftshift` swaps the first half and the second half of the output of `fft`. Then all that remains to do is to define the appropriate set of sample frequencies to plot against. As an example, the Matlab code given below computes the K-point FFT of a signal stored in the vector `h`, uses `fftshift` to rearrange the frequency samples, and computes a vector of frequencies to plot against. It stores the FFT in the vector `Hw` and the frequency vector in `w`.

```
Hw=fft(h,K);                    % take the K-point fft
w=(2*pi/K)*[ 0:(K-1) ]';        % vector of frequencies:  0 to 2pi
mid=ceil(K/2)+1;                % find midpoint of freq. vector
w(mid:K)=w(mid:K)-2*pi;         % change [pi,2pi) to [-pi,0)
w=fftshift(w);                  % use fftshift to swap both w
Hw=fftshift(Hw);                %   and Hw vectors
```

The above code is *one* example of how to compute the frequency vector for a given value of K (which can be even or odd). Obviously, there are other ways to do the same thing. You are welcome to use this code or write your own for the frequency response calculations in this project. (Some of the exercises in 7.6 described below require plotting the frequency response between $-\pi$ and $+\pi$.) Whether you decide to use this code or write your own, I encourage you to spend a few minutes thinking about how the FFT samples the Fourier transform.

2 Frequency Sampling Filter Design

Do all of the exercises in Section 5.4 of *Computer Explorations in Signals and Systems* by Buck, Daniel, and Singer. Your writeup should include answers to all of the questions in the book and any other ob-

[1] Assuming that N is greater than or equal to the length of $x[n]$.

servations you make as you complete the exercises.

Additional instructions:

- For part c, include a plot of the phase response (`Hp`) that you define.

3 Discrete-Time Differentiation

Do all of the exercises in Section 7.6 of *Computer Explorations in Signals and Systems* by Buck, Daniel, and Singer. Your writeup should include answers to all of the questions in the book and any other observations you make as you complete the exercises.

Additional instructions:

- You may find it helpful to review Section 5.7 of your textbook (Oppenheim/Schafer/Buck) and our discussion of Type I-IV FIR filters prior to doing these exercises.