To: Prof. Ross Snider

From: Jeff Meirhofer

Regarding: Lab #6 – Filtering Sampled Waveforms

Date: February 26, 2009

**Summary:**

The purpose of this lab is to explore how systems of FIR filters will affect signals. First, we will test some filters on a stair-step signal to observe the response. Then, we will explore how filtering a speech signal affects the signal. In both of these exercises, we will examine the result of cascading filters in a different order.

**Cascaded FIR Filtering**

**3.2)** A 5-point averager is created using the “firfilt” function in MATLAB. We are given a stair-step signal, “x1[n]”, to input into the filter. This signal, along with all of the other supplied signals, come from a file called “LAB06DAT.MAT”.

**3.2 a)** The output of the system, “v1”, is plotted and compared to the input. The figure is shown below. We can see that the 5-point averager acts to “smooth” the signal. It appears that the output is time shifted by about 2 units.



Figure 1. Input and output of 5-point averager

**3.2 b)** Now we will use “freqz” to show the frequency response of the filter. The plot is shown on the next page.



Figure 2. Frequency response of 5-point averager

It is clear that the filter is a low-pass filter. Low frequencies are passed and high frequencies are rejected. This makes sense because sharp transitions (high slope) in the time domain have frequency components that are very high. When we remove these high frequencies, the sharp edges become smoother.

**3.2 c)** The slope of the phase is -2. This makes sense because the output is time shifted by 2 units in the time domain.

**3.3)** Now we will implement a first-difference filter on the same input signal.

**3.3 a)** The input and output are compared in the figure on the following page.



Figure 3. Input and output of first-difference filter

**3.3 b)** The first-difference filter caused the output to have spikes at the points where the input was changing rapidly. It “roughed” the input signal. The output has peaks at points where the input has areas of high slope.

**3.3 c)** Now, we use “freqz” to show the frequency response of the filter. It is shown in the following figure.



Figure 4. Frequency response of first-difference filter

The shape of the magnitude curve is as expected. We can see that it passes high frequencies and rejects low frequencies. This is exactly what we see happening in Figure 3 also. The output only reacts when there are high frequencies present in the input. The First Difference is definitely a high-pass filter.

**3.4 a)** In this part, we will run the input through the 5-point averager and then run the result through the first difference filter.

**3.4 b)** The frequency response of the entire system is shown below.



Figure 5. Frequency response of cascaded system.

**3.5 a)** Now we will send the input through the same two filters, but in the opposite order.

**3.5 b)** The frequency response of this cascaded circuit is shown below.



Figure 6. Frequency response of 2nd cascaded system

As you can see, the frequency responses for the two cascaded systems are identical. This proves that the ordering of the filters does not matter.

**3.6)** Now we will use MATLAB to estimate the difference between the outputs of the two cascaded systems. We implement the function with MATLAB. This calculation computes the difference between y1 and y2 for each discrete point plotted. It then squares this difference, and adds all of the squared differences. The result we get is 2.3668e-029. Since this is a very, very small number (basically 0), we can tell that the outputs are equal and that the ordering of the filters does not matter. The two outputs can be compared in the graph below:



Figure 7. Comparison of outputs from both systems

**3.7)** Now we will experiment with a speech signal by running it through some supplied filters.

**3.7 a)** We start by running a supplied speech signal, x2, through a supplied filter with coefficients contained in h1. By looking at the input compared to the output, we can see that the output is smoother than the input. This is done using the “inout” function and shown in Figure 8.



Figure 8. Input and output of filter h1

**3.7 b)** The next figure shows the frequency response of the filter with coefficients given by h1.



Figure 9. Frequency response of filter h1.

We can clearly see that the filter passes low frequencies, and attenuates high frequencies. For this reason, it is called a low-pass filter.

**3.7 c)** Now we will make a stem plot of the filter coefficients. It is shown in Figure 10.



Figure 10. Stem plot of coefficients in h1.

We can see that the point of symmetry in the coefficients is at the 23rd point.

**3.7 d)** Now we will filter the signal with the second supplied filter, h2. A plot of the input versus the output is shown on the next page.



Figure 11. Input and output of filter h2

We can see that the output is much “rougher” than the input.

**3.7 e)** The frequency response is shown below.



Figure 12. Frequency response of filter h2

We can see that the filter passes all high frequencies, but attenuates low frequencies. For this reason, it is called a high-pass filter.

**3.7 f)** We can make a stem plot of the coefficients of h2, shown on the next page:



Figure 13. Stem plot of coefficients in h2

The coefficients have a point of symmetry at the 23rd point.

**3.7 g)** Next, we will listen to all three signals in succession. The first signal is the original. The second signal, filtered with h1, sounds muddy (less clear). The third signal, filtered with h2, sounds clearer than the original.

**3.7 h)** If we were to add the two filtered signals together, we would expect the result to sound like the original signal. Since one filtered signal contains only the low frequencies of the original, and the other contains the high frequencies, we would expect that adding them produces a signal similar to the original. When we listened to the result, our predictions were correct. If we added the frequency responses together, we would expect to see a filter that does not produce gain or attenuation at any frequency. This is confirmed by Figure 14.



Figure 14. Frequency response of h1 + h2.

**Appendix**

**3.1 – 3.6**

close all

clear all

w = [-pi:pi/100:pi];

%----------------------------------------------------------------------

%3.2

%----------------------------------------------------------------------

load lab06dat

b5 = 1/5\*ones(1,5);

v1 = firfilt(b5,x1);

figure(1)

subplot(2,1,1),plot(1:length(x1),x1),title('x1[n]'),axis([0 105 -15 5])

subplot(2,1,2),plot(1:length(v1),v1),title('v1[n]'),axis([0 105 -15 5])

% [m,n] = ginput(2)

%

% m(2)-m(1)

figure(2),freqz(b5,1, w)

%----------------------------------------------------------------------

%3.3

%----------------------------------------------------------------------

bfd = [1 -1];

v2 = firfilt(bfd, x1);

figure(3)

subplot(2,1,1),plot(1:length(x1),x1),title('x1[n]'),axis([0 105 -15 5])

subplot(2,1,2),plot(1:length(v2),v2),title('v2[n]'),axis([0 105 -5 5])

figure(4),freqz(bfd, 1, w)

%----------------------------------------------------------------------

%3.4

%----------------------------------------------------------------------

y1 = firfilt(bfd,v1);

figure(5),freqz(firfilt(b5,bfd),1, w)

%----------------------------------------------------------------------

%3.5

%----------------------------------------------------------------------

y2 = firfilt(b5,v2);

figure(6),freqz(firfilt(bfd,b5),1, w)

figure(7)

subplot(2,1,1),plot(1:length(y1),y1),title('y1[n]')

subplot(2,1,2),plot(1:length(y2),y2),title('y2[n]')

%----------------------------------------------------------------------

%3.6

%----------------------------------------------------------------------

sum((y1-y2).^2)

**3.7**

close all

clear all

load lab06dat

w = [-pi:pi/100:pi];

%----------------------------------------------------------------------

%a

%----------------------------------------------------------------------

y1 = firfilt(h1,x2);

figure(1),inout(x2,y1,3000,1000,3)

%----------------------------------------------------------------------

%b

%----------------------------------------------------------------------

figure(2),freqz(h1,1,w)

%----------------------------------------------------------------------

%c

%----------------------------------------------------------------------

figure(3),stem(h1)

% [m,n] = ginput(1);

% m

%----------------------------------------------------------------------

%d

%----------------------------------------------------------------------

y2 = firfilt(h2,x2);

figure(4),inout(x2,y2,3000,1000,3)

%----------------------------------------------------------------------

%e

%----------------------------------------------------------------------

figure(5),freqz(h2,1,w)

%----------------------------------------------------------------------

%f

%----------------------------------------------------------------------

figure(6),stem(1:length(h2),h2)

%----------------------------------------------------------------------

%g

%----------------------------------------------------------------------

% soundsc([x2; y1; y2],11025)

%----------------------------------------------------------------------

%h

%----------------------------------------------------------------------

% soundsc([x2; y1+y2],11025)

figure(7),freqz(h1+h2,1,w)