To: Prof. Ross Snider

From: Jeff Meirhofer

Regarding: Lab #5 – FIR Filtering of Sinusoidal Waveforms

Date: February 19, 2009

**Summary:**

The purpose of this lab is to explore the properties of FIR filters. We will generate several signals and test the results of running them through different styles and configurations of FIR filters. The output responses we see will help us to understand the properties of FIR filters.

**FIR Filtering**

**3)** We will be using a FIR filter of the form . Our experiments will consist of input signals consisting of cosine functions.

**3.2)** A cosine with amplitude 7, phase π/3, and ω = 0.125 is constructed. It is then passed through a first-difference filter defined by .

**3.2 a)** The length of the filtered signal is 50, while the length of the original signal is 49. The output contains one extra point because there is a point at the beginning of the sequence that contains only the x[n] term and a point at the end of the sequence that contains only the x[n-1] term.

**3.2 b)** Both x[n] and y[n] are plotted in the same window, shown below.



Figure 1. A cosine before and after filtering.

**3.2 c)** Using ginput, we can see that the amplitude of x[n] is 7.07, and the phase is .32\*π. These are close enough to verify the graph.

**3.2 d)** We can see from looking at the graph that the output y[n] is indeed a scaled and shifted cosine wave at 0.125\*π radians/sec. The first sample is different from the others because the x[n-1] term is not represented in the input, it is just zero.

**3.2 e)** Again using ginput, we can determine that y[n] has an amplitude of 13.3, a phase of .75\*π, and a frequency of 0.125\*π rad/sec.

**3.2 f)** At 0.125\*π radians/sec, our filter has an amplitude ratio of 1.88, and the phase of the output leads the phase of the input by 0.43\*π.

**3.2 g)** To verify the correctness of our results, we now calculate what these ratios should be by hand. By allowing x[n] to be , we can test the magnitude and phase of h(ω). After evaluating with , we simplify to . When we check this function at our frequency (0.125\*π), we see that the amplitude is multiplied by 1.95 and the phase shifts -0.43\*π radians, just as seen before.

**3.3 a)** Now we multiply the original x[n] by two and send it through the filter again. The resultant graph is shown below, and we use ginput to once again make measurements from it. From these measurements, we see that the amplitude gain is 1.8447 and the phase difference is -.432\*π. This shows us that these ratios remain constant for a scaled input.



Figure 2. Scaled input with same FIR filter as before

**3.3 b)** Next, we will generate a new input signal, with an amplitude of 8 and a frequency of 0.25\*π. We will also run this signal through the same filter we have been using. The figure is shown below. The amplitude ratio is 4.07 and the phase difference is -0.4\*π, according to ginput. The frequency stays the same as the input signal.



Figure 3. Response from a different cosine signal.

**3.3 c)** Finally, we create a signal that is the sum of xa and xb, called xc. This signal is ran through the filter, and then compared to summing the two signals after they have both been run through the filter. The two graphs are identical. We can see that it does not matter if the signals are filtered and then added, or first added and then filtered. The graphs are shown in Figure 4.



Figure 4. Graph showing that the filtered sums are identical.

**3.4)** Now the input is time shifted before it is filtered. The signal is delayed by 3 units. The output can be shifted 3 units back the other way and it will match the original signal (output) exactly. A graph that compares the two is shown below.



Figure 5. Graph comparing time-shifted input

**3.5 a)** In this section, we will explore a system with two blocks. The first is a squarer, and the second is a FIR filter. The input signal (x[n]) is squared (then becomes w[n]), and then filtered by another first-difference filter with = {1 -1}.

**3.5 b)** All three of the waveforms are shown below.



Figure 6. All three waveforms of the cascaded system

**3.5 c)** Below are three sketches, one of each of the frequency spectrums of the three signals.

x[n] w[n] y[n]

**3.3 d)** When we examine the graph of w[n], we can see that a higher frequency component is introduced. Although the stem plot makes it hard to see, the cosine wave is going twice as fast as in the input.

**3.3 e)** As w[n] passes through the first-difference filter, the output is simply the difference between each point and the point before it. This makes y[n] positive when w[n] is rising, and negative when w[n] is falling.

**3.3 f)** Finally, the second block in the system is replaced with a FIR filter with When we plot the output, we can see that the only frequency components that remain are the DC ones. The components of 0.25\*π are attenuated by the filter. The way this is done can be shown by once again calculating h() of the filter by hand. We end up with . You can easily see that when ω = 0.25\*π, the amplitude is 0. This can also be seen from the graph of y[n] in Figure 7. Also, the spectrum of y[n] is sketched below the figure.



Figure 7. y2[n], showing that only DC components are left

y2[n]

**Appendix**

**3.2-3.4**

close all

clear all

L = 50;

n = (0:L-1);

w = .125\*pi;

x1 = 7\*cos(w\*n + pi/3);

bb = [5 -5];

yy = firfilt( bb, x1);

yy = yy( 1:L );

figure(1)

subplot(2,1,1)

stem(n , yy), title('Plot of y(n)')

subplot(2,1,2)

stem(n , x1), title('Plot of x(n)')

[Xn1,XA1] = ginput(1);

[Yn1,YA1] = ginput(1);

disp('x[n]')

disp('Phase')

-Xn1\*w

disp('Amplitude')

XA1

disp('y[n]')

disp('Phase')

-Yn1\*w

disp('Amplitude')

YA1

disp('Ratio of Amplitudes')

YA1/XA1

disp('Difference in Phase')

-Yn1\*w+Xn1\*w

xa = 2\*x1;

ya = firfilt(bb,xa);

ya = ya(1:L);

figure(3)

subplot(2,1,1),stem(n,ya),title('ya[n]')

subplot(2,1,2),stem(n,xa),title('xa[n]')

[x1,y1] = ginput(1);

[x2,y2] = ginput(1);

disp('3.3 - Part (a)')

disp('Ratio of Amplitudes')

y2/y1

disp('Difference in Phase')

-x2\*w+x1\*w

%b

xb = 8\*cos(.25\*pi\*n);

yb = firfilt(bb,xb);

yb = yb(1:L);

figure(4)

subplot(2,1,1),stem(n,yb),title('yb[n]')

subplot(2,1,2),stem(n,xb),title('xb[n]')

[x1,y1] = ginput(1);

[x2,y2] = ginput(1);

disp('3.3 - Part (b)')

disp('Ratio of Amplitudes')

y2/y1

disp('Difference in Phase')

-x2\*.25\*pi+x1\*.25\*pi

%c

xc = xa+xb;

yc = firfilt(bb,xc);

yc = yc(1:L);

figure(5)

subplot(2,1,1),stem(n,yc),title('yc[n]')

subplot(2,1,2),stem(n,ya+yb),title('ya[n]+yb[n]')

%3.4

xs = 7\*cos(w\*(n-3)+pi/3);

ys = firfilt(bb,xs);

ys = ys(1:L);

figure(6)

subplot(2,1,1),stem(n,ys),axis([0 50 -15 20]),title('ys[n]')

subplot(2,1,2),stem(n,yy),title('y1[n]')

**3.5**

close all

clear all

L = 50;

n = (0:L-1);

w = .125\*pi;

xx = 7\*cos(w\*n + pi/3);

ww = xx.\*xx;

bb = [1 -1];

yy = firfilt( bb, ww);

yy = yy( 1:L );

subplot(3,1,1), stem(n, xx), title('x[n]');

subplot(3,1,2), stem(n, ww), title('w[n]');

subplot(3,1,3), stem(n, yy), title('y[n]');

bb = [1 -2\*cos(.25\*pi) 1];

y2 = firfilt(bb,ww);

y2 = y2(1:L);

figure(2)

stem(n,y2)