Taylor Rembos
EEL 3135
Lab 6 Part 1

Lab 6: Introduction to Frequency Response

1.1 Frequency Response of the Four-Point Averager

1.1 Frequency Response of the Four-Point Averager

(a) Using the Euler formulas, show by hand that the frequency response for the 4-point running average operator is given by

$$H(e^{j\hat{\omega}}) = ((2\cos(0.5\,\hat{\omega}) + 2\cos(1.5\,\hat{\omega}))/4) * e^{-j1.5\hat{\omega}}$$

(b) Implement this equation directly in MATLAB and plot the magnitude and phase response of this filter. For $\hat{\omega}$, use a vector that includes 400 samples between $-\pi$ and π. Since the frequency response is a complex-valued quantity, use abs()and angle()to extract the magnitude and phase of the frequency response for plotting.

(c) The function freqz() (as described in Section A.1) evaluates the frequency response for all frequencies in the vector ww by using the *summation formula* instead of the formula from part (a). Use freqz()in MATLAB to plot the magnitude and phase of $H(e^{j\hat{\omega}})$ versus $\hat{\omega}$. The plots from this section should look the same as the ones from part b.

Hint: The filter coefficient vector for the 4-point averager can be defined by bk = 1/4*ones(1,4);

(a)

Lab 6

1.1

(a) $H(e^{j\hat{\omega}}) = \frac{2\cos(0.5\hat{\omega}) + 2\cos(1.5\hat{\omega})}{4} e^{-j1.5\hat{\omega}}$

$y[n] = \frac{1}{4} \sum_{k=0}^{k=3} x[n-k]$

$= \frac{1}{4} [x[n] + x[n-1] + x[n-2] + x[n-3]]$

$= \frac{1}{4} X(e^{j\hat{\omega}}) [1 + e^{-j\hat{\omega}} + e^{-j2\hat{\omega}} + e^{-j3\hat{\omega}}]$

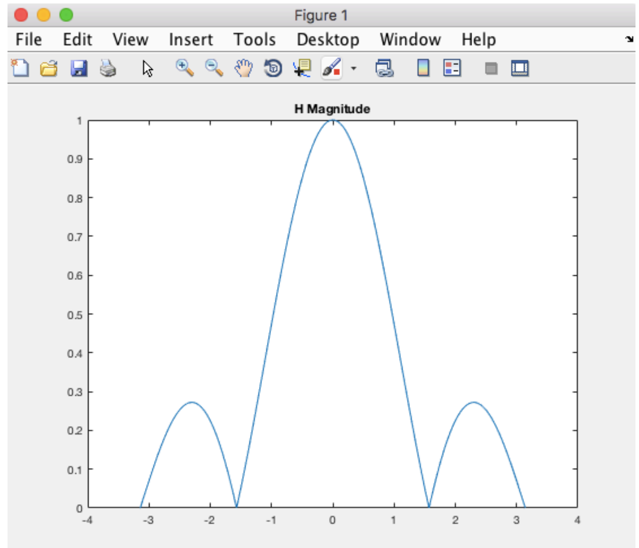
$\frac{Y(e^{j\hat{\omega}})}{X(e^{j\hat{\omega}})} = H(e^{j\hat{\omega}}) = \frac{1}{4} [1 + e^{-j\hat{\omega}} + e^{-j2\hat{\omega}} + e^{-j3\hat{\omega}}]$

$= \frac{1}{4} \left[2\left(\frac{e^{j0.5\hat{\omega}} + e^{-j0.5\hat{\omega}}}{2}\right) + 2\left(\frac{e^{j1.5\hat{\omega}} + e^{-j1.5\hat{\omega}}}{2}\right)\right]$

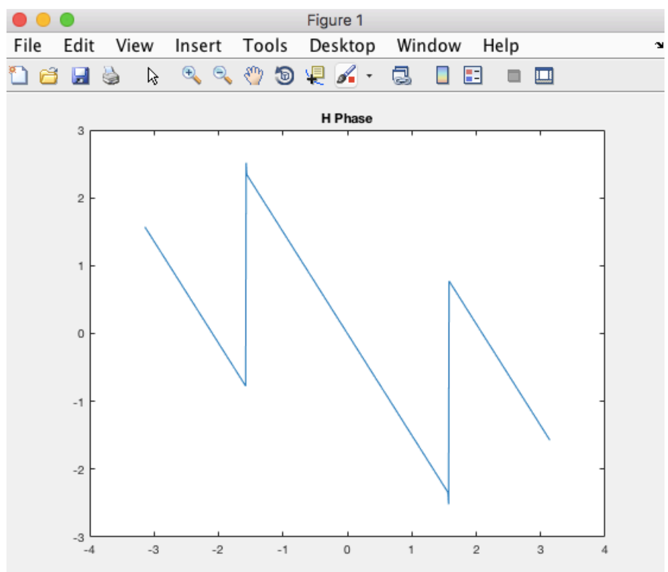
$= \frac{2\cos(0.5\hat{\omega}) + 2\cos(1.5\hat{\omega})}{4} e^{-j1.5\hat{\omega}}$

(b) & (c)

```
bk = 1/4*ones(1,4); % given
ww = -pi:(pi/400):pi; % 400 samples between -pi and pi
H = freqz(bk, 1, ww);
H_mag = abs(H);
H_phase = angle(H);

plot(ww,H_mag);
title('H Magnitude');
```




```
plot(ww,H_phase);
title('H Phase');
```



1.2 The MATLAB find()function

As part of signal processing functions, we often need to perform logical tests, such as with if statements. MATLAB's find() function allows logical tests to be done in a vectorizable form. It returns a vector of indices of all the vector elements for which the logical test is true. For example, in the following code, find() returns the indices of all the numbers that "round" to 3:

```
xx = 1.4:0.33:5;
jkl = find(round(xx)==3);
xx(jkl)
```

So, the xx(jkl)command returns the elements of the xx vector with indices found by the find(). Note that the argument of the find()function can be any logical expression. find()returns a list of indices where such logical condition is true.

Suppose that you have a frequency response:

```
ww = -pi:(pi/500):pi;
HH = freqz(1/4*ones(1,4), 1, ww);
```

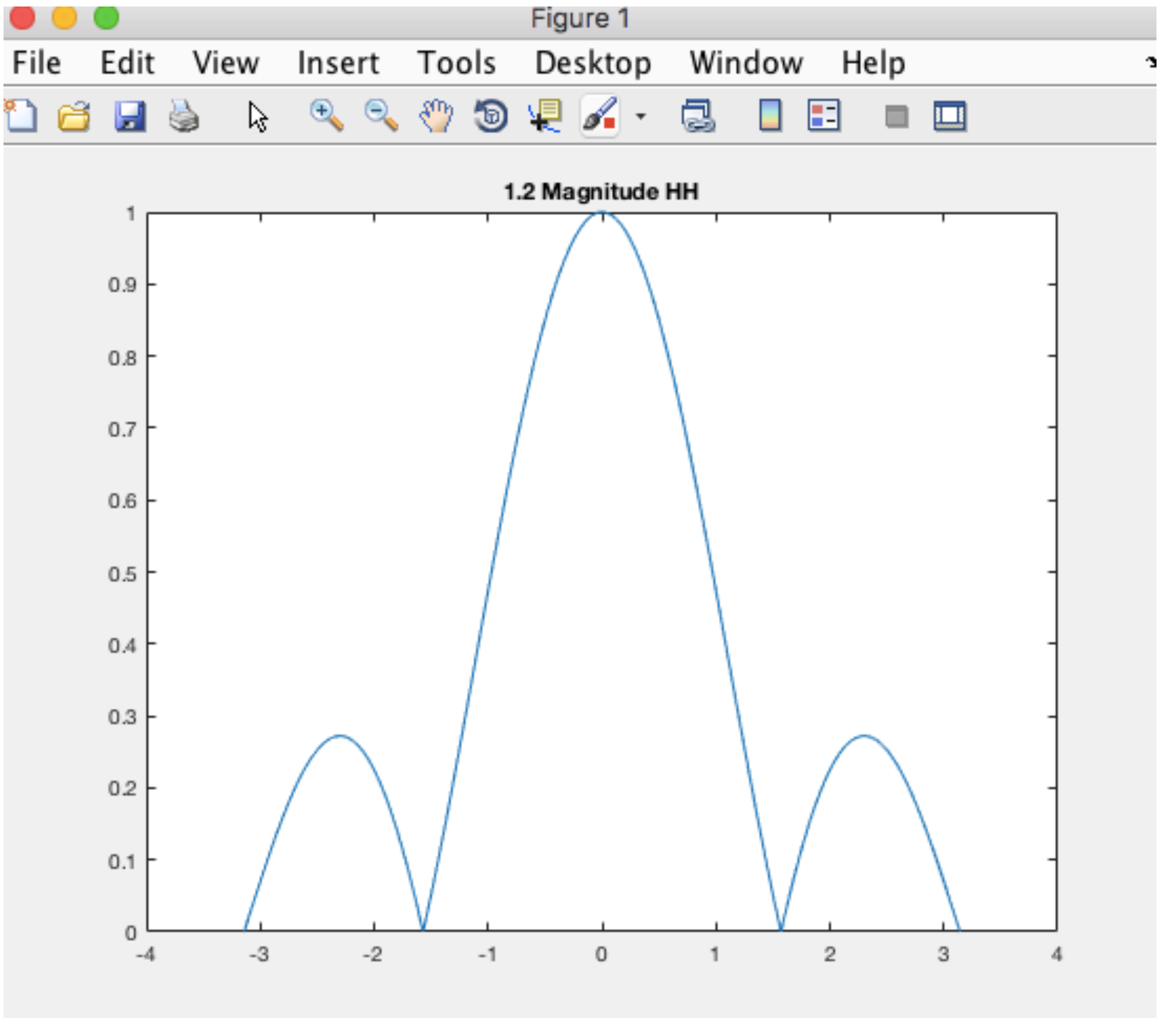
Using find, display the list of frequencies where HH is approximately zero. Note that since there might be round-off error in calculating HH, the logical test should probably be a test for those indices where the magnitude (absolute value in MATLAB) ofHH is less than some small number, e.g., 1×10^{-6}. Does this match the frequency response that you plotted for the 4-point average?

```
>> xx = 1.3:0.33:5;
>> jkl = find(round(xx)==3);
>> xx(jkl)

ans =

    2.6200    2.9500    3.2800


>> ww = -pi:(pi/500):pi;
>> HH = freqz(1/4*ones(1,4),1,ww);
>> mno = find((abs(HH))<=(1*10^-6))

mno =

           1         251         751        1001

>> mno*(pi/500)

ans =
```

0.0063   1.5771   4.7187   6.2895

```
>> plot(ww,abs(HH));
>> title('1.2 Magnitude HH');
```

The plots of the magnitudes match but the values do not match the frequency response values for the 4-point average.



1.3 LTI Frequency Response Demo

dltidemo

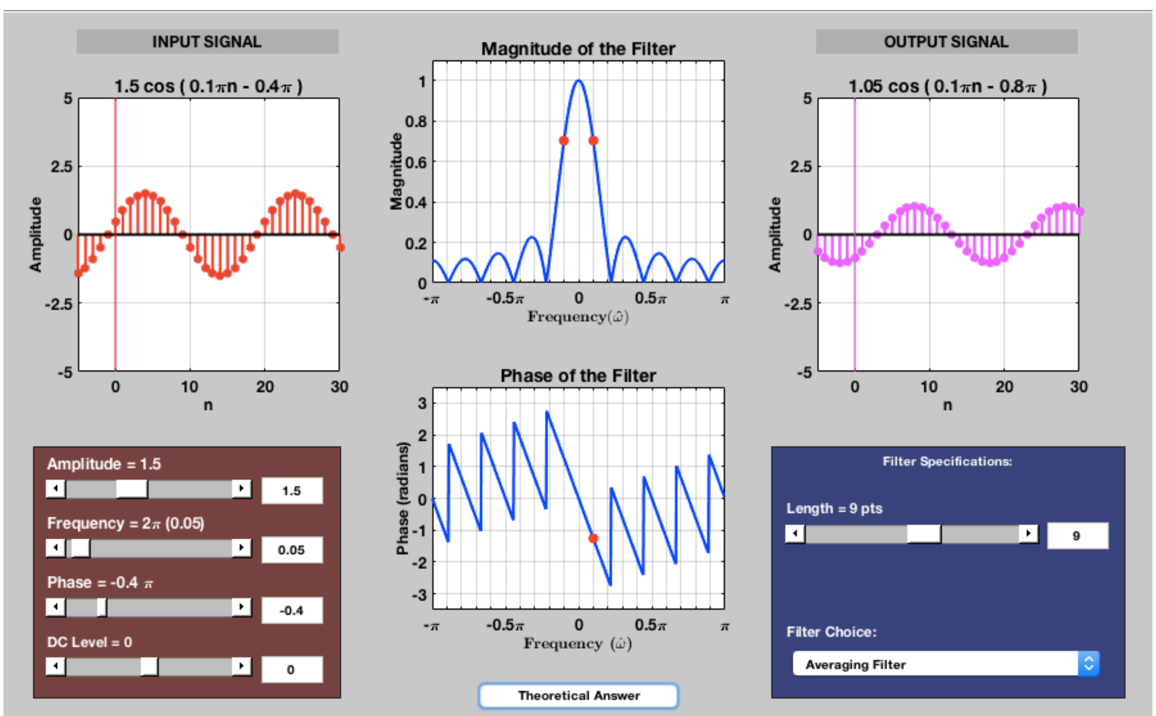
The dltidemo shown above illustrates the "sinusoid-IN gives sinusoid-OUT" property of discrete-time LTI systems. In this demo, you can change the amplitude, phase and frequency of an input sinusoid, $x[n]$, and you can also change the digital filter that processes the signal. Then, the GUI will show the output signal, $y[n]$.

(a) Set the input to $x[n] = 1.5\cos(0.1\pi(n-4))$, set the digital filter to be a 9-point averager and determine the formula for the output signal, written in the form $y[n] = A\cos(\hat{\omega}_0(n - n_d))$

(b) Using n_d for $y[n]$ and the fact that the input signal had a peak at $n = 4$, determine the amount of delay through the filter. In other words, how much has the peak of the cosine wave shifted?

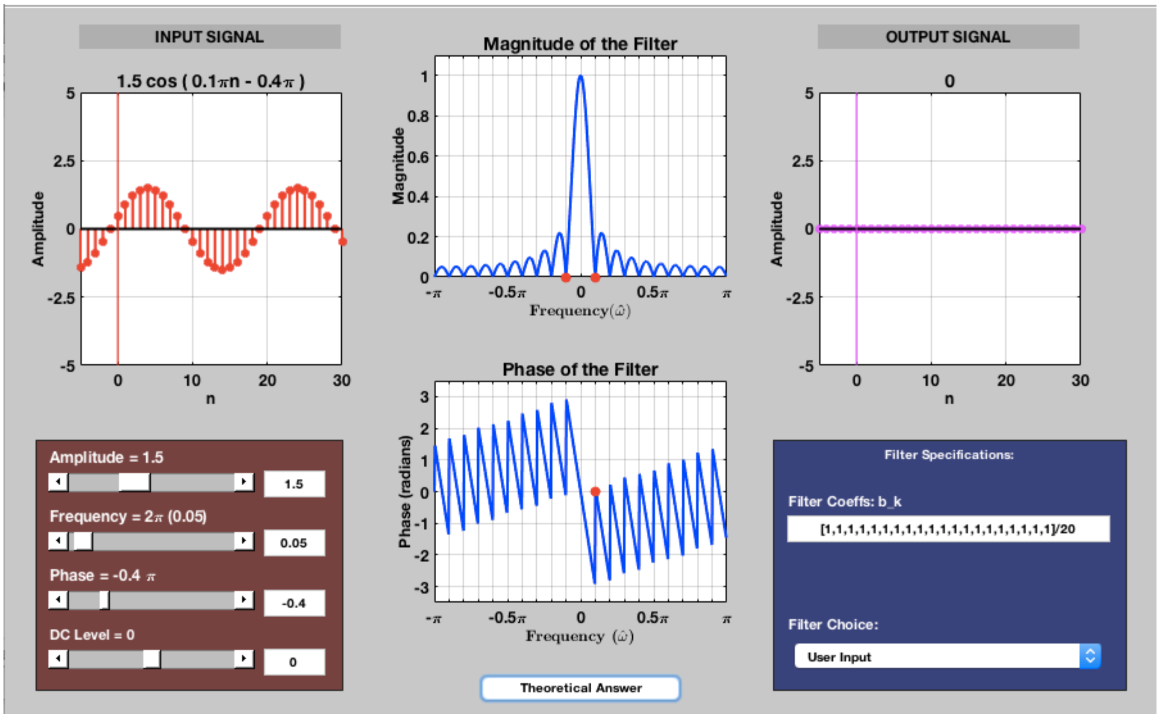
(c) Determine the length of the averaging filter needed to make the output zero, i.e., $y[n] = 0$. Using the GUI, show that you have the correct filter to zero the output. Hint: If the length is more than 15, you will have to enter the "Filter Specifications".
(d) When the output is zero, the filter acts as a *Nulling Filter,* because it eliminates the input at $\hat{\omega} = 0.1\pi$. Which other frequencies $\hat{\omega}$ are also nulled?

(a)



(b) t = -phi/w, phi = -0.4, w = 0.1, therefore, the amount of delay t = 0.4/0.1 = 4 seconds.

(c) The length of the averaging filter needed to make the output zero is 20.



(d) All values of w with a multiple of x when w = 0.1pi+2pi*x will also be nulled.

1.4 Nulling Filter:

(a) Design a filtering system that consists of the **cascade** of two FIR nulling filters that will eliminate the following input frequencies: $\hat{\omega}_n = 0.44\pi$, and $\hat{\omega}_n = 0.7\pi$. For this part, derive the filter coefficients of both nulling filters. Submit necessary code, and plots of the magnitude and phase responses of both filters, along with the cascaded system.

(b) Generate an input signal $x[n]$ that is the sum of three sinusoids: $x[n] = 5\cos(0.3\pi n) + 22\cos(0.44\pi n - \frac{\pi}{3}) + 22\cos(0.7\pi n - \frac{\pi}{4})$ Make the input signal 150 samples long over the range $0 \le n \le 149$. Submit your code to generate this signal.

(c) Using firfilt() or conv(), filter the sum-of-three-sinusoids signal $x[n]$ through the filters designed in part (a). Submit the MATLAB code that you wrote to implement the cascade of two FIR filters and the command to filter the signal.

(d) Make a plot of the **first 50 points** of the output signal. Determine (by hand) the exact mathematical formula (magnitude, phase and frequency) for the output signal for $n \ge 5$.

(e) Plot the mathematical formula determined in (d) with MATLAB and compare it to the plot obtained in (d) to show that it matches the filter output from firfilt()over the range $5 \le n \le 50$.

Explain why the output signal is different for the first few points. How many "start-up" points are found? How is this number related to the lengths of the filters designed in part (a)?

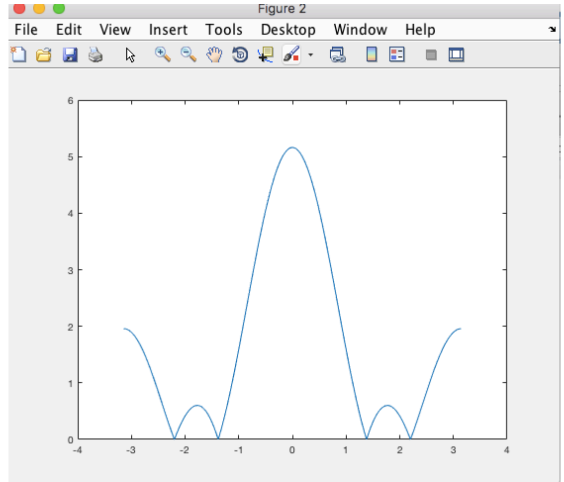
Hint: consider the length of a single FIR filter that is equivalent to the cascade of two length-3 FIRs.

(a)

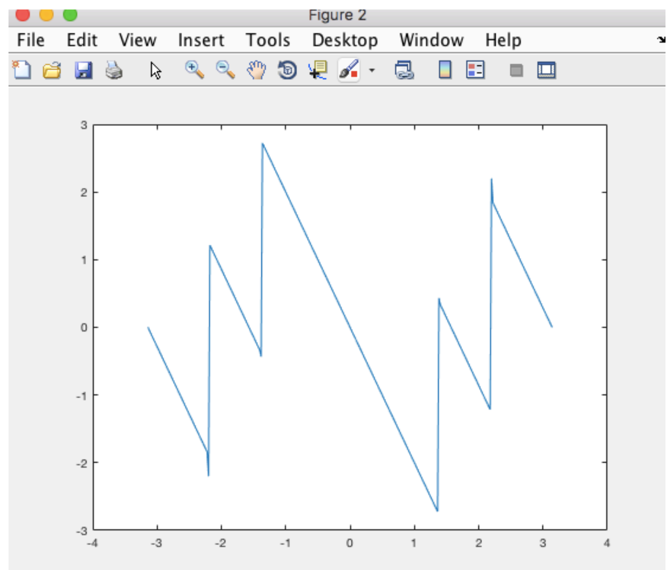
```
>> h1 = [1, -2*cos(0.44*pi), 1]

h1 =

    1.0000   -0.3748    1.0000

>> h2 = [1, -2*cos(0.7*pi), 1]

h2 =

    1.0000    1.1756    1.0000

>> f = conv(h1,h2)

f =
```

1.0000   0.8008   1.5594   0.8008   1.0000

```
>> ww = -pi:(pi/150):pi;
>> HH = freqz(f,1,ww);
>> plot(ww,abs(HH));
```

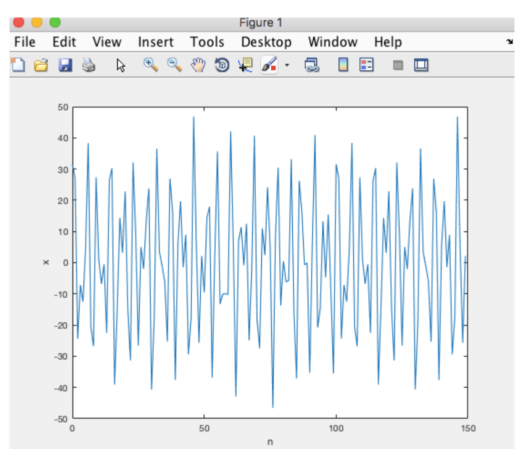



```
>> plot(ww,angle(HH));
```

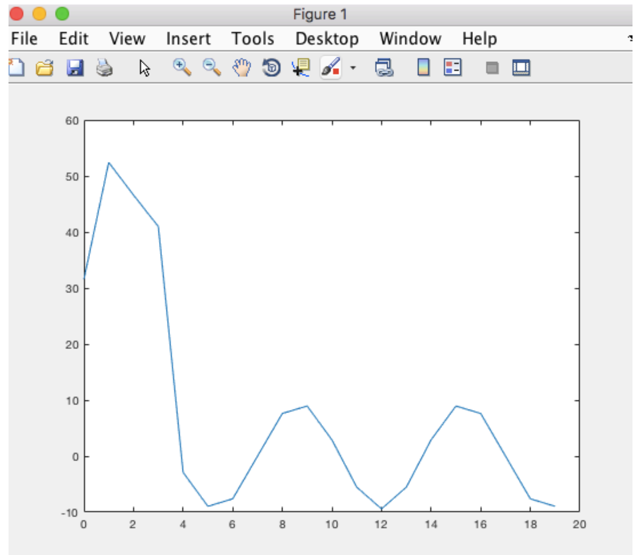


(b)

```
>> ww = -pi:(pi/150):pi;
>> n = 0:149;
>> x = 5*cos(0.3*pi*n)+22*cos(0.44*pi*n-(pi/3))+22*cos(0.7*pi*n-(pi/4));
>> plot(n,x);
>> xlabel('n');
>> ylabel('x');
```

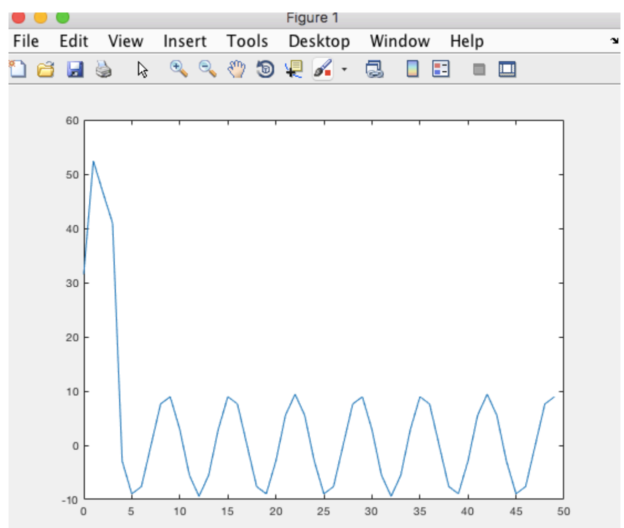


(c)
>> n = 0:149;
>> x = 5*cos(0.3*pi*n)+22*cos(0.44*pi*n-(pi/3))+22*cos(0.7*pi*n-(pi/4));
>> firfilt(f,x);
>> y = firfilt(f,x);
>> plot([0:19],y[1:20]);

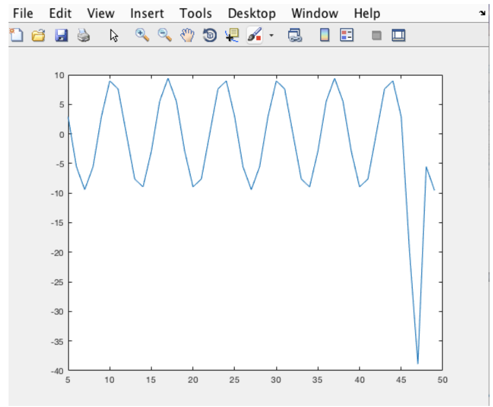


(d)

>> plot([0:49],y(1:50));



(e) >> ww = -pi:(pi/150):pi;
>> n = 5:50;
>> x = 5*cos(0.3*pi*n)+22*cos(0.44*pi*n-(pi/3))+22*cos(0.7*pi*n-(pi/4));
>> firfilt(f,x);
>> y = firfilt(f,x);
>> plot([5:49],y(6:50));



The plot is the inverse of that obtained in part (d). There are 48 start up points found which is 12 times the length of the 4-length filter in part (a).