**Department of Electrical Engineering**

**EE-330 Digital Signal Processing**

**Lab #9 Frequency Response and Nulling Filters**

**Department of Electrical Engineering**

|  |  |
| --- | --- |
| **Faculty Member: Dr Ahmed Salman** | **Dated: 3 April 2109** |
|  |  |
| **Course/Section: Bee 8d** | **Semester: 6** |
|  |  |

**EE-330 Digital Signal Processing**

**Lab #9 Frequency Response and Nulling Filters**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  |  | **PLO4-CLO4** | | **PLO5-CLO5** | **PLO8-CLO6** | **PLO9-CLO7** |
| **Name** | **Reg. No** | **Viva / Quiz / Lab Performance** | **Analysis of data in Lab Report** | **Modern Tool Usage** | **Ethics and Safety** | **Individual and Team Work** |
|  |  | **5 Marks** | **5 Marks** | **5 Marks** | **5 Marks** | **5 Marks** |
| **Muhammad Anus Siddiqui** | **174308** |  |  |  |  |  |
| **Haris Rehman** | **182797** |  |  |  |  |  |
| **Husnain Afridi** | **192414** |  |  |  |  |  |
|  |  |  |  |  |  |  |

|  |  |
| --- | --- |
|  |  |

**Lab9: Frequency Response, Bandpass and Nulling Filters**

**Objectives**

The goal of this lab is to study the response of FIR filters to inputs such as complex exponentials and sinusoids. In the experiments of this lab, you will use firfilt(), or conv(), to implement filters and freqz() to obtain the filter’s frequency response. As a result, you should learn how to characterize a filter by knowing how it reacts to different frequency components in the input

* Introduction to bandpass filters
* Introduction to Nulling filters
* Cascade systems and their frequency response
* How to extract information from sinusoidal signals

# Frequency Response: Bandpass and Nulling Filters

## 9.1 Pre-Lab

This lab also introduces two practical filters: bandpass filters and nulling filters. Bandpass filters can be

used to detect and extract information from sinusoidal signals, e.g., tones in a touch-tone telephone dialer.

Nulling filters can be used to remove sinusoidal interference, e.g., jamming signals in a radar.

### Frequency Response of FIR Filters

The output or *response* of a filter for a complex sinusoid input, depends on the frequency Often a filter is described solely by how it affects different input frequencies—this is called the *frequency response*. For example, the frequency response of the two-point averaging filter 

can be found by using a general complex exponential as an input and observing the output or response.

In (3) there are two terms, the original input, and a term that is a function of This second term is the

frequency response and it is commonly denoted by which in this case is



Once the frequency response has been determined, the effect of the filter on any complex exponential may be determined by evaluating at the corresponding frequency. The output signal y[n], will be a complex exponential whose complex amplitude has a constant magnitude and phase. The phase describes the phase change of the complex sinusoid and the magnitude describes the gain applied to the complex sinusoid. The frequency response of a general FIR linear time-invariant system is

In the example above, M = 1, and b0 = 1 /2 and b1 = 1/2 .



### MATLAB Function for Frequency Response

MATLAB has a built-in function called freqz() for computing the frequency response of a discrete-time LTI system. The following MATLAB statements show how to use freqz to compute and plot both the magnitude (absolute value) and the phase of the frequency response of a two-point averaging system as a function of in the range 

bb = [0.5, 0.5]; %-- Filter Coefficients

ww = -pi:(pi/100):pi; %-- omega hat

HH = freqz(bb, 1, ww); %<--freekz.m is an alternative

subplot(2,1,1);

plot(ww, abs(HH))

subplot(2,1,2);

plot(ww, angle(HH))

xlabel(’Normalized Radian Frequency’)

For FIR filters, the second argument of freqz( , 1, ) must always be equal to 1. The frequency vector ww should cover an interval of length 2π for and its spacing must be fine enough to give a smooth curve for .Note: we will always use capital HH for the frequency response.

### Periodicity of the Frequency Response

The frequency responses of discrete-time filters are *always* periodic with period equal to 2π.

then considering two input sinusoids whose frequencies are And + 2π. 

Consult Chapter 6 for a mathematical proof that the outputs from each of these signals will be identical

(basically because x1[n] is equal to x2[n].The implication of periodicity is that a plot of only needs to extend over the interval or any other interval of length 2π.

### Lab Task 1: Frequency Response of the Four-Point Averager

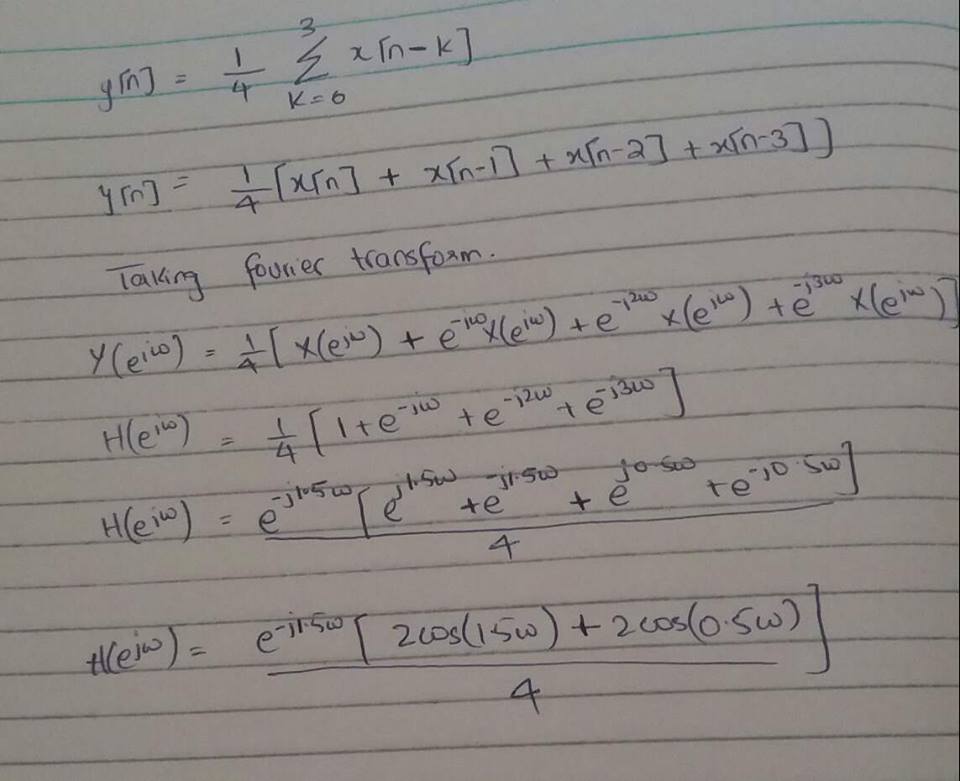
In class we examined filters that average input samples over a certain interval. These filters are called “running average” filters or “averages” and they have the following form for the L-point averager:



**(a)** Use Euler’s formula and complex number manipulations to show that the frequency response for the

4-point running average operator is given by:





**(b)** Implement (7) directly in MATLAB. Use a vector that includes 400 samples between –π and π for

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. Plotting the real and imaginary parts of

is not very informative.

**Matlab code:**

ww = -pi:(pi/200):pi;

H = ((2.\*cos(0.5.\*ww)+2.\*cos(1.5.\*ww))/4).\*exp(-i\*1.5.\*ww);

subplot(2,1,1)

plot(ww,abs(H))

xlabel('frequency')

ylabel('amplitude')

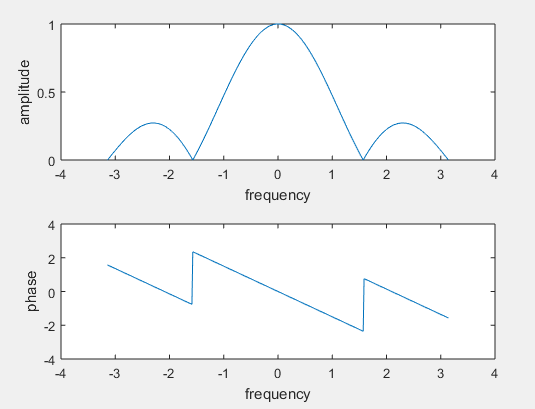
subplot(2,1,2)

plot(ww,angle(H))

xlabel('frequency')

ylabel('phase')

**Output:**



**(c)** In this part, use freqz.m in MATLAB to compute numerically (from the filter coefficients and plot its magnitude and phase versus . Write the appropriate MATLAB code to plot both the magnitude and phase of Follow the example in Section 8.1.2. The filter coefficient vector forthe 4-point averager is defined via:

**Matlab code:**

bb = 1/4\*ones(1,4);

bb = [0.25, 0.25, 0.25, 0.25];

ww = -pi:(pi/200):pi;

HH = freqz(bb, 1, ww);

subplot(2,1,1);

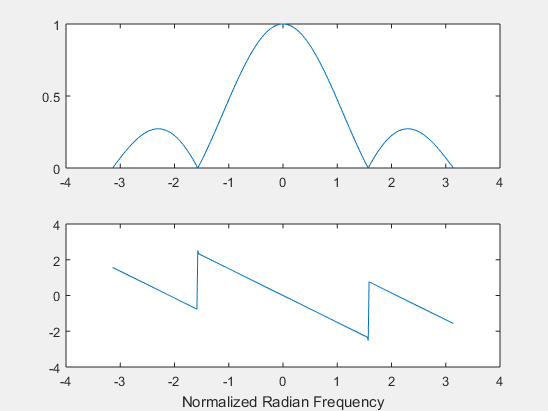
plot(ww, abs(HH))

subplot(2,1,2);

plot(ww, angle(HH))

xlabel('Normalized Radian Frequency');

**Matlab plot:**



Note: the function freqz(bb,1,ww) evaluates the frequency response for all frequencies in the

vector ww. It uses the summation in (5), not the formula in (7). The filter coefficients are defined in

the assignment to vector bb. How do your results compare with part (b)?

**The magnitude and phase plots are the same as part b.**

### The MATLAB FIND Function

Often signal processing functions are performed in order to extract information that can be used to make

a decision. The decision process inevitably requires logical tests, which might be done with if-then

constructs in MATLAB. However, MATLAB permits vectorization of such tests, and the find function is one way to do lots of tests at once. In the following example, find extracts all the numbers that “round” to 3:

xx = 1.4:0.33:5,

jkl = find(round(xx)==3),

xx(jkl)

The argument of the find function can be any logical expression. Notice that find returns a list of indices where the logical condition is true. See help on relop for information. Now, suppose that you have a frequency response:

ww = -pi:(pi/500):pi;

HH = freqz( 1/4\*ones(1,4), 1, ww );

Use the find command to determine the indices where HH is zero, and then use those indices to display the list of frequencies where HH is zero. 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) of HH is less than some rather small number, e.g., 1 × 10−6. Compare your answer to the frequency response that you plotted for the four-point average in Section 8.1.4.

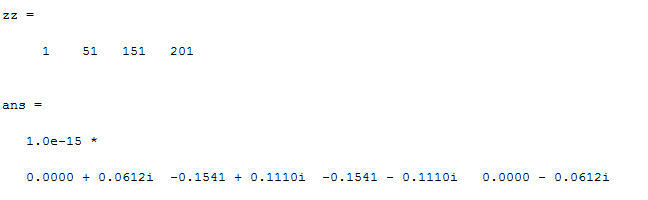
bb = [0.25, 0.25, 0.25, 0.25]; %-- Filter Coefficients

ww = -pi:(pi/500):pi; %-- omega hat

HH = freqz(bb, 1, ww); %<--freekz.m is an alternative

zero = find(abs(HH)<1\*10^-6);

x = ww(zero)';



### Cascading Two Systems

More complicated systems are often made up from simple building blocks. In Fig. 2, two FIR filters are

shown connected “in cascade.”



Assume that the system in Fig. 2 is described by the two equations



**(a)** Use freqz() in MATLAB to get the frequency responses for the case where α = 0.8 and M = 9.

Plot the magnitude and phase of the frequency response for Filter #1, and also for Filter #2. Which

one of these filters is a *lowpass filter*?

**Matlab code:**

%Filter 1

M=9;

aa=0.8;

l=0:M;

bb=aa.^l;

ww=-pi:(pi/500):pi;

HH=freqz(bb,1,ww);

subplot(4,1,1);

plot(ww,abs(HH))

xlabel('frequency')

ylabel('amplitude')

subplot(4,1,2);

plot(ww,angle(HH))

xlabel('freqeuncy')

ylabel('Phase')

%Filter 2

bb=[1,-0.8];

ww=-pi:(pi/500):pi;

HH=freqz(bb,1,ww);

subplot(4,1,3);

plot(ww,abs(HH))

xlabel('frequency')

ylabel('amplitude')

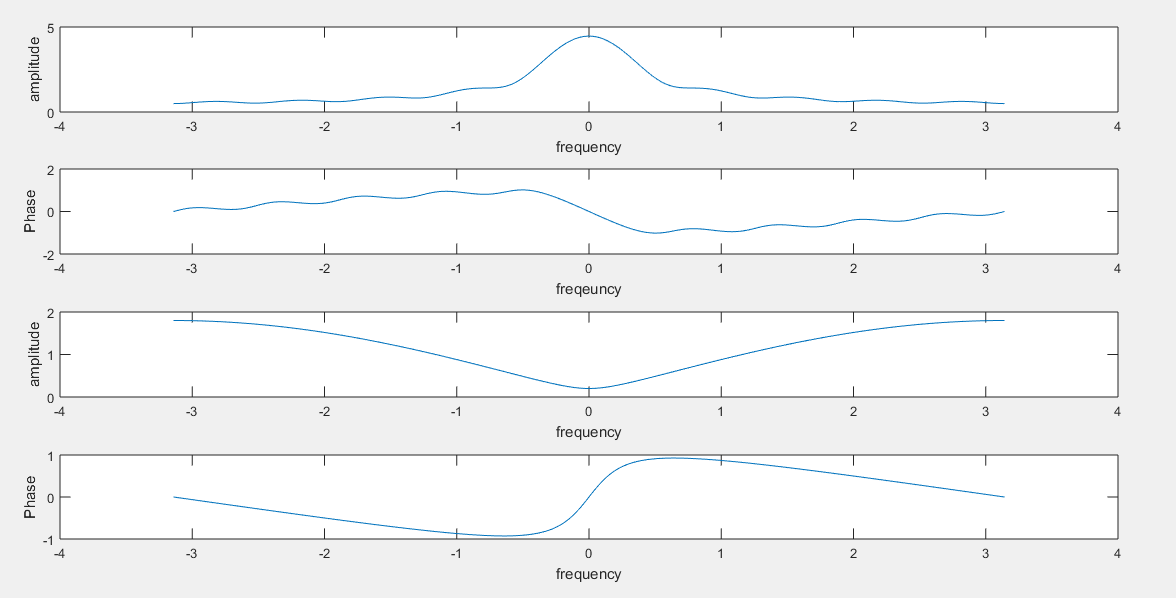
subplot(4,1,4);

plot(ww,angle(HH))

xlabel('frequency')

ylabel('Phase')

**Matlab plot:**



**The first one is the low pass filter**

**(b)** Plot the magnitude and phase of the frequency response of the overall cascaded system.

**Matlab code:**

M=9;

aa=0.8;

l=0:M;

bb=aa.^l; %Filter 1

ww=-pi:(pi/500):pi;

HH=freqz(bb,1,ww);

zz1\_abs=abs(HH);

zz1\_angle=angle(HH);

% the magnitude and phase of the frequency response for Filter 1

bb1=[1,-0.8]; % Filter 2

ww=-pi:(pi/500):pi;

HH1=freqz(bb1,1,ww);

zz2\_abs=abs(HH1);

zz2\_angle=angle(HH1);

% the magnitude and phase of the frequency response for Filter 2

ww=-pi:(pi/500):pi;

ma=zz1\_abs.\*zz2\_abs; % multiply the magnitude

subplot(2,1,1);

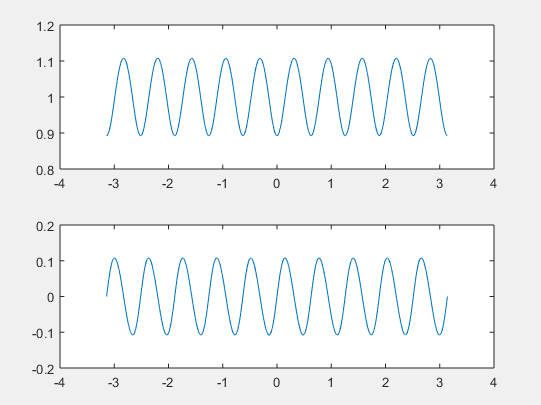
plot(ww,ma)

ph=zz1\_angle+zz2\_angle; % add the phase

subplot(2,1,2);

plot(ww,ph)

**Matlab plot:**



**(c)** Explain how the individual frequency responses in part(a) are combined to get the overall frequency

response in part(b). Comment on the magnitude combinations as well as the phase combinations.

The magnitudes of filter 1 and filter 2 are multiplied while the phases are added linearly. However, when calculating from previous figures( multiply the magnitude and add the phase point by point), it is easy to find that it is not an ideal filter system.

## Lab Tasks

### Lab Task 2

#### Nulling Filters for Rejection

Nulling filters are filters that completely eliminate some frequency component. If the frequency is = 0 or = π, then a two-point FIR filter will do the nulling. The simplest possible general nulling filter can have as few as three coefficients. If is the desired nulling frequency, then the following length-3 FIR filter

will have a zero in its frequency response at = . For example, a filter designed to completely eliminate signals of the form would have the following coefficients because we would pick the desired nulling frequency to be = 0.5π. b0 = 1, b1 = −2 cos(0.5π) = 0, b2 = 1.

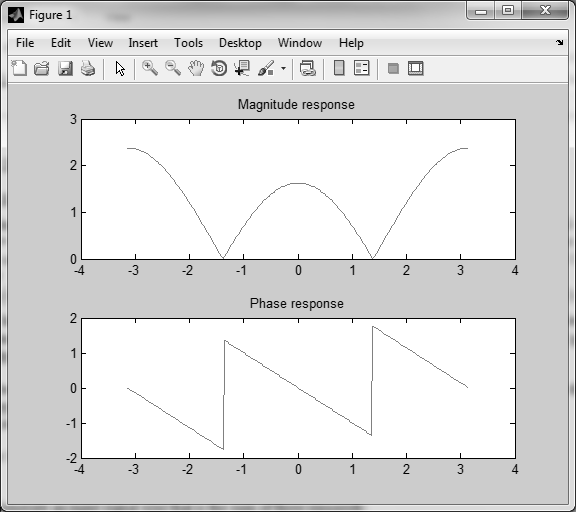
**(a)** Design a filtering system that consists of the *cascade of two FIR nulling filters* that will eliminate the

following input frequencies: = 0.44π, and = 0.7π. For this part, derive the filter coefficients of

both nulling filters.

nulling filter 1 coefficients: b0=1,   b1=-2\*cos(0.44\*pi)=0,      b2=1

nulling filter 2 coefficients: b0=1,   b1=-2\*cos(0.7\*pi)=0,        b2=1



**Filter 1**

bb = [1, -0.375, 1]; %-- Filter Coefficients

ww = -pi:(pi/100):pi; %-- omega hat

HH1 = freqz(bb, 1, ww); %<--freekz.m is an alternative

subplot(2,1,1);

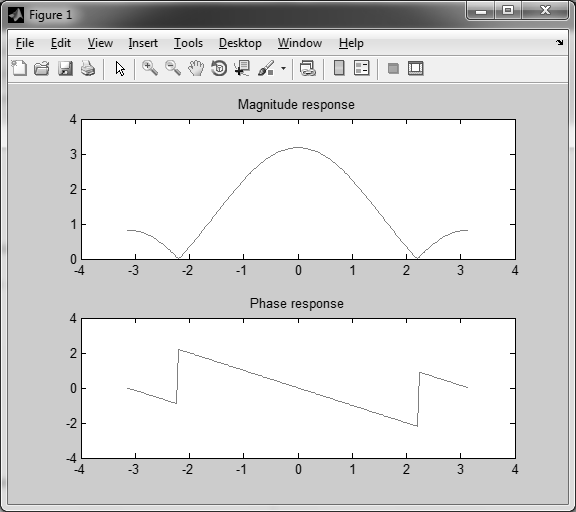
plot(ww, abs(HH1))

title('Magnitude response')

subplot(2,1,2);

plot(ww, angle(HH1))

title('Phase response')



**Filter 2**

bb = [1, 1.176, 1];

ww = -pi:(pi/100):pi; %-- omega hat

HH2 = freqz(bb, 1, ww); %<--freekz.m is an alternative

subplot(2,1,1);

plot(ww, abs(HH2))

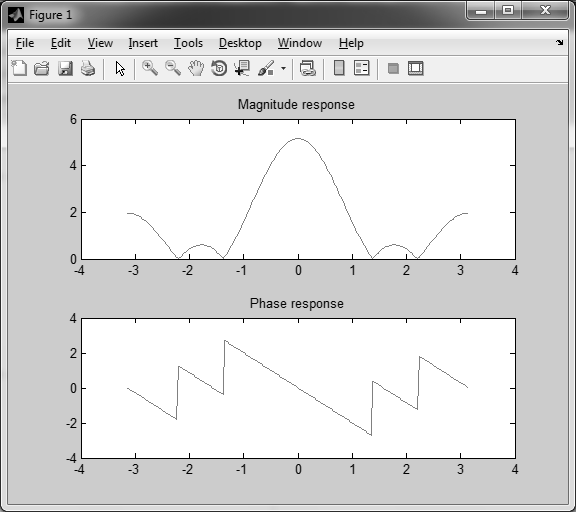
title('Magnitude response')

subplot(2,1,2);

plot(ww, angle(HH2))

title('Phase response')

**In cascade**



HH3 = HH1.\*HH2;

subplot(2,1,1);

plot(ww, abs(HH3))

title('Magnitude response')

subplot(2,1,2);

plot(ww, angle(HH3))

title('Phase response')

**(b)** Generate an input signal x[n] that is the sum of three sinusoids:



Make the input signal 150 samples long over the range 0 < n < 149.

**Matlab code:**

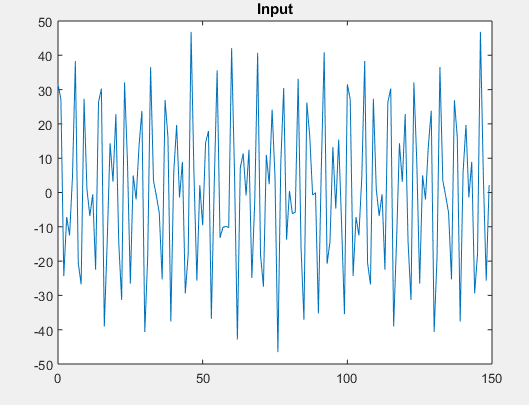
n=0:149;

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)

title('Input')

**Matlab plot:**



**(c)** Use filter (or conv) to filter the sum of three sinusoids signal x[n] through the filters designed

in part (a). Show the MATLAB code that you wrote to implement the cascade of two FIR filters.

**Matlab code:**

bb1 = [1, -0.375, 1]; %-- Filter Coefficients

ww = -pi:(pi/500):pi; %-- omega hat

HH1 = freqz(bb1, 1, ww); %<--freekz.m is an alternative

bb2 = [1, 1.176, 1];

ww = -pi:(pi/500):pi; %-- omega hat

HH2 = freqz(bb2, 1, ww); %<--freekz.m is an alternative

n=0:150;

x = 5\*cos(0.3\*pi\*n)+ 22\*cos(0.44\*pi\*n - (pi/3))+22\*cos(0.7\*pi\*n-(pi/4));

subplot(3,1,1)

plot(n,x)

title('Input')

yy1 = conv(bb1,x)%passing through filter 1

subplot(3,1,2)

plot(yy1)

title('Output after 1st filter')

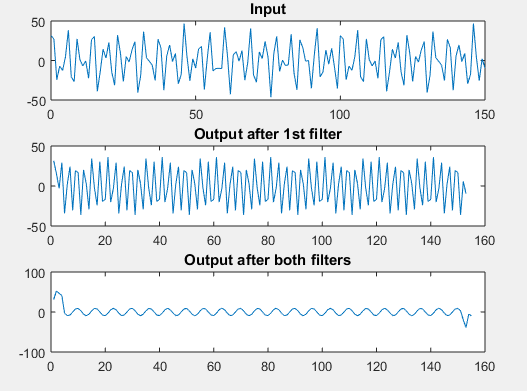
yy2 = conv(bb2,yy1) %passing through filter 2

subplot(3,1,3)

plot(yy2)

title('Output after both filters')

**Matlab plot:**



**(d)** Make a plot of the output signal—show the first 40 points. Determine (by hand) the exact mathematical formula (magnitude, phase and frequency) for the output signal for n ≥ 5.

**Matlab code:**

first=0;

last=39;

bb=[1 -2\*cos(0.44\*pi) 1]

xx=5\*cos(0.3\*pi.\*nn)+22\*cos(0.44\*pi.\*nn-pi/3)+22\*cos(0.7\*pi.\*nn-pi/4);

nn=first:last;

ww=conv(bb,xx);

bb2=[1 -2\*cos(0.7\*pi) 1];

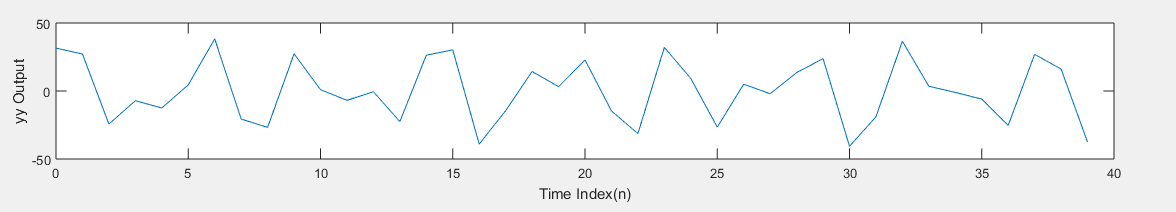
yy=conv(bb2,ww);

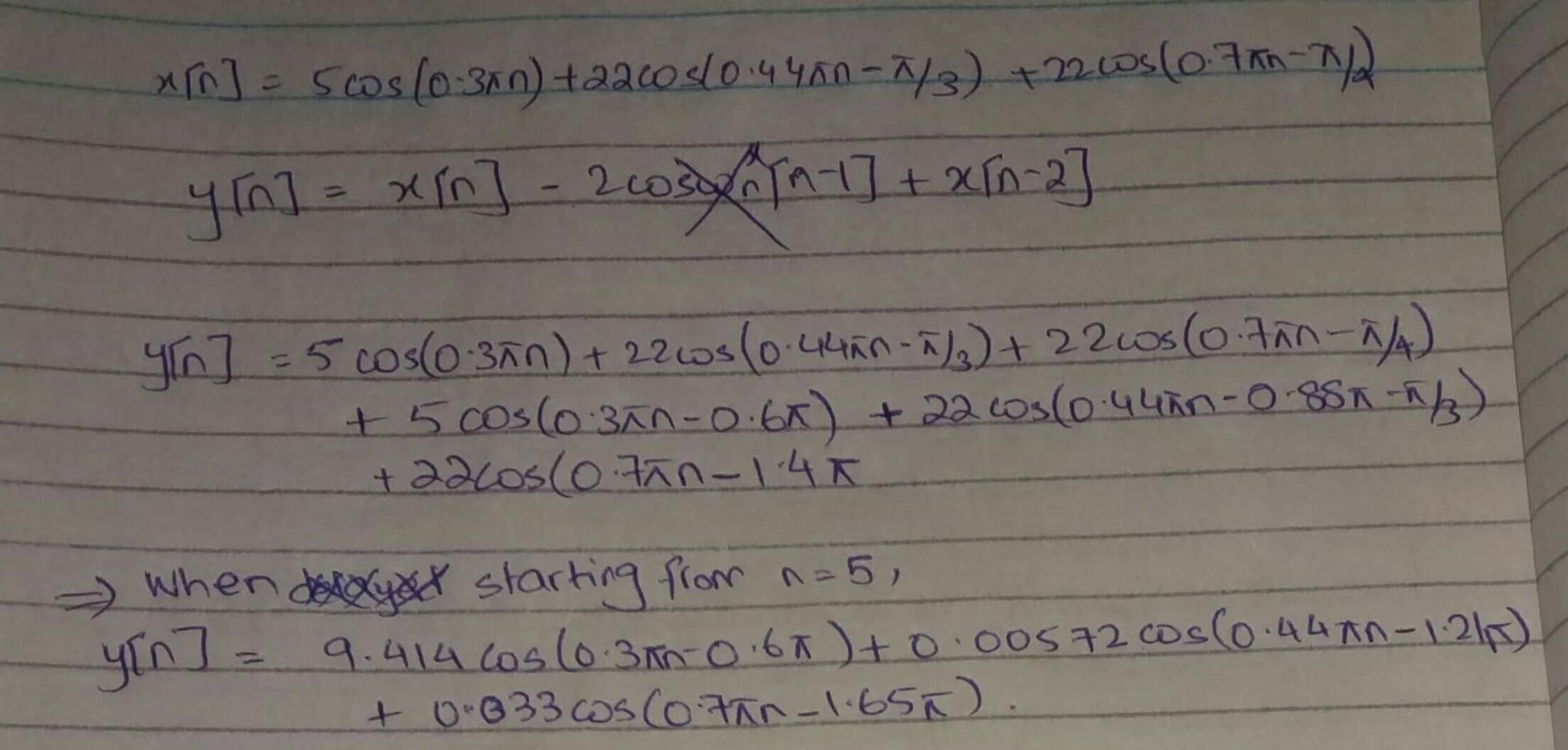
plot(nn,xx(nn+1))

xlabel('Time Index(n)')

ylabel('yy Output')

**Matlab output:**





**(e)** Plot the mathematical formula determined in (d) with MATLAB to show that it matches the filter output from filter over the range 5 ≤n ≤ 40.

**Matlab code:**

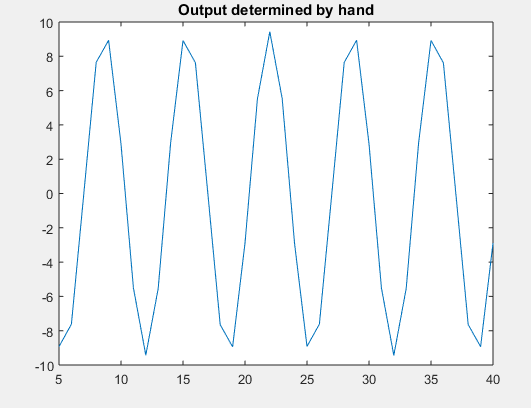
n=5:40

yy3=9.412\*cos(0.3\*pi.\*n-0.6\*pi)+0.00572\*cos(0.44\*pi.\*n-1.21\*pi)+0.033\*cos(0.7\*pi.\*n-1.65\*pi);

plot(n,yy3)

title('Output determined by hand')

**Matlab output:**



**(f)** Explain why the output signal is different for the first few points. How many “start-up” points are

found, and 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.

Input is not filtered properly until the filter and input completely overlap for convolution which does not happen for the first few points hence they’re different. The length of the second filter is 5 hence the first 5 points are not filtered entirely. So there are 5 start up points.

**Conclusion:**

In this lab we learned to design various (low pass, high pass, null type) FIR filters using the freqz() command in MATLAB. The command requires numerator and denominator coefficients of the filter transfer function. We also learned how to implement two cascaded filters and the response of the cascaded filters in comparison with the individual filters.