University of Dhaka
DSP LAB EEE 3204

Lab 1: Generate and plot a sine wave

[file name: lab3.m]
Following parameters are given
sine wave duration = dur = 3 sec;
frequency = f = 300 Hz,
sampling frequency = fs = 8kHz.
Now generate and plot the sine wave.

MATLAB COMMANDS

```
>>close all; clear all;
>>dur=3;
>>f=300;
>>fs=8000;
>>n=0:dur*fs;
>>t=n/fs;
>>xn=sin(2*pi*f*t);

% plot two figures

>>plot(xn); title('Plot of a sine wave') % 1st figure
>>figure; plot(xn); % 2nd figure
>>xmin=0; xmax=200; ymin=-1.5; ymax=1.5;
>>axis([xmin xmax ymin ymax]); grid
>>title('Plot of a fraction of the sine wave for better visibilty!')
>>xlabel('Samples')
>>ylabel('Values')
```

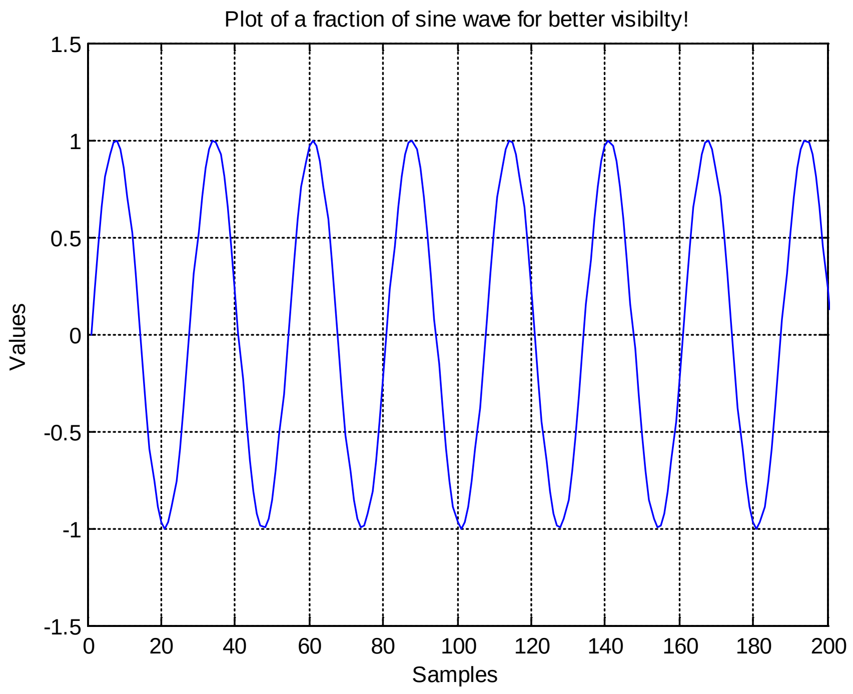


Figure: Plot of 1st 201 samples.

Lab 2: adding and shifting sequences.

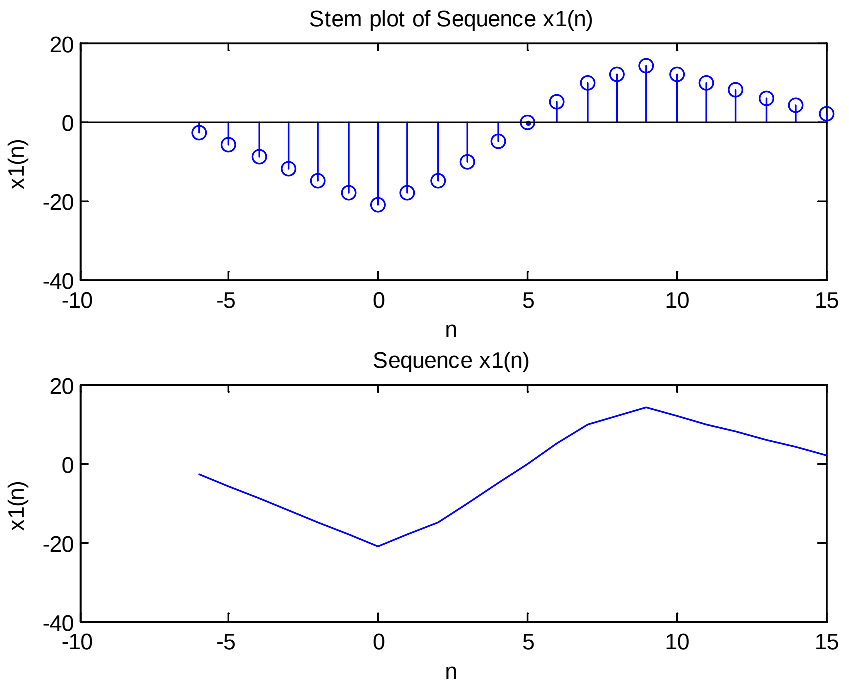
Implement following expression and plot the result using matlab.

x1(n)=2x(n-5)-3x(n+4)
Let x(n)={1 2 **3** 4 5 6 7,6,5,4,3,2,1}; Here n=2 is the origin of x(n).

****
This lab requires 3 files in the same folder:
1. sigadd.m 2. sigshift.m and 3. lab4.m

```
%%% Step 1. Make a function sigadd:

function [y,n]=sigadd(x1,n1,x2,n2)
%The function sigadd implements y(n)=x1(n)+x2(n)
min_n=min(min(n1),min(n2));
max_n=max(max(n1),max(n2));
n=min_n:max_n;
y1=zeros(1,length(n));
y2=y1;
y1(find((n>=min(n1))&(n<=max(n1))==1))=x1;
y2(find((n>=min(n2))&(n<=max(n2))==1))=x2;
y=y1+y2


%% Save this function as sigadd.m in a folder where
%% other m files should exist.

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%%% Step 2. Make a function sigshift.

function [y,n]=sigshift(x,m,n0)
%implements shifting a sequence x(n).
%The result is y(n)=x(n-k);
%Here +k implements a delay by k samples.
n=m+n0;
y=x;

%% Save this function as sigshift.m in a folder where
%% other m files should exist.
%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%%% Step 3. Write following statements in command window to implement
x1(n)
>>clc
>>close all; clear all
>>x=[1:7,6:-1:1]; % implements x(n)
>>n=-2:10; % determine origin of x(n) (position of arrow)
>>[x11,n11]=sigshift(x,n,5); % shifts x(n) by k=5 i.e.,x(n) delays
>>[x12,n12]=sigshift(x,n,-4); % shifts x(n) by k=-4 i.e, x(n) advances.
>>[x1,n1]=sigadd(2*x11,n11,-3*x12,n12) % multiplay and add shifted
sequences
>>disp('You got the result x1(n)')
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%%% Step 4. Plot the final result


>>subplot(2,1,1);stem(n1,x1); % stem plot
>>xlabel('n'); ylabel('x1(n)');
>>title('Stem plot of Sequence x1(n)')

>>subplot(2,1,2);plot(n1,x1); % simple plot
>>xlabel('n'); ylabel('x1(n)'); title('Sequence x1(n)')
```



Lab 3: Determination of Convolution of 2 sequences

```
Determine the convolution y(n)=x(n)*h(n)
Given:
x(n) = {3,11,7,0,-1,4,2}; nx=[-3:3]; % n=3 is the Origin of x(n);
position of arrow
h(n) = {2,3,0,-5,2,1}; nh=[-1:4]; %  n=1 is the origin of h(n):
position of arrow
```

```
Step 1. Make a function conv_m. This function uses matlab built in
function conv

function [y,ny]=conv_m(x,nx,h,nh)
% Modified convolution function
%[y,ny]=conv_m(x,nx,h,nh)
```

```
%[y,ny]=convolution result
%[x,nx]=1st input signal; [h,nh]=2nd input signal
%
nyb=nx(1)+nh(1); nye=nx(length(x))+nh(length(h));
ny=nyb:nye;
y=conv(x,h);
```

Save this m file in a directory where all other files will stay.

Step 2. Write the following statements in matlab command window.

```
>>clc; close all; clear all
>>x=[3,11,7,0,-1,4,2]; nx=[-3:3];
>>h=[2,3,0,-5,2,1]; nh=[-1:4];
>>[y,ny]=conv_m(x,nx,h,nh)

>>disp('Now you got the result.')
```

Lab 6: Convolution of z-transforms

<missing>

Lab 7: Plotting poles and zeros onto z-plane (pole-zero map)

Given, 2 zeros z1= 0+1.0j and z2 = 0 - 1.0j.
3 poles: -1.0000+j0; 0.5000 + 0.5000i; 0.5000 - 0.5000i
Now plot zeros and poles on the z-plane using matlab.

```
MATLAB COMMANDS
>>clear all
>>close all
>>zeros=[j;-j];
>>poles=[-1;0.5+0.5j;0.5-0.5j];
>>zplane(zeros,poles); % zplane command draws the pole-zero map
>>title('My pole-zero map')
```

Lab8: Pole-zero plot of the digital filter given the difference equation of the filter.

Given,
$y(n) = b_0x(n)+b_1x(n-1)+b_2x(n-2)$ .. $-a_1(y(n-1)-a_2y(n-2)$ … eqn-1

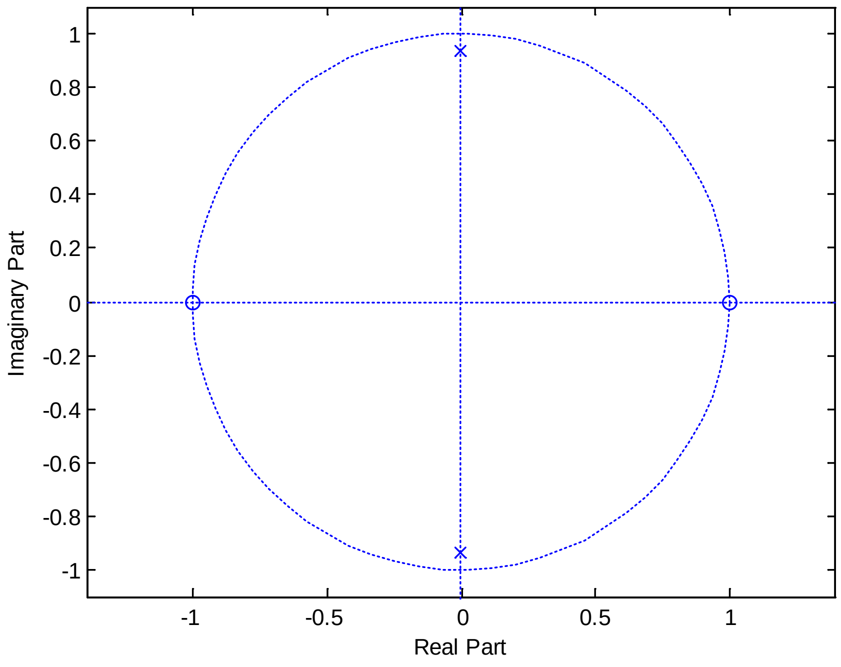
Then

$H(z) = \frac{b_0+b_1z^{-1}+b_2z^{-2}+\cdots}{1-a_1z^{-1}-a_2z^{-2}-\cdots}$ … eqn-2

Now inspecting above equation

$H(z) = \frac{1+0z^{-1}-1z^{-2}}{1-0z^{-1}-0.8779z^{-2}} = \frac{1-z^{-2}}{1-0.8779z^{-2}}$ … eqn-3 from above, $b_0=1$, $b_1=0$, $b_2=-1$ and $a_1=0$, $a_2=0.8779$

```
MATLAB COMMANDS
>>close all
>>clear all
>>b=[1 0 -1];
>>a=[1 0 0.8779];
>>zplane(b,a)
```



Lab 9: Plotting frequency responses (magnitude and phase response) of digital filters, given the {b} and {a} coefficients.

(Use the coefficient values of the previous lab)

```
MATLAB COMMANDS
>>Close all
>>Clear all
>>b=[1 0 -1];
>>a=[1 0 0.8779];
>>[H,W]=freqz(b,a,100);
>>magH=abs(H);
>>phaH=angle(H);
>>subplot(2,1,1),plot(W/pi,magH);grid
>>subplot(2,1,2),plot(W/pi,phaH);grid
```

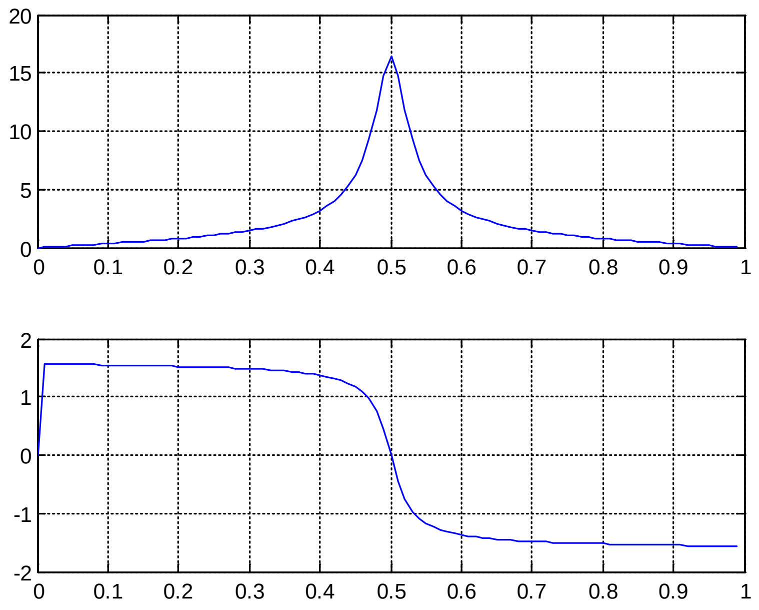
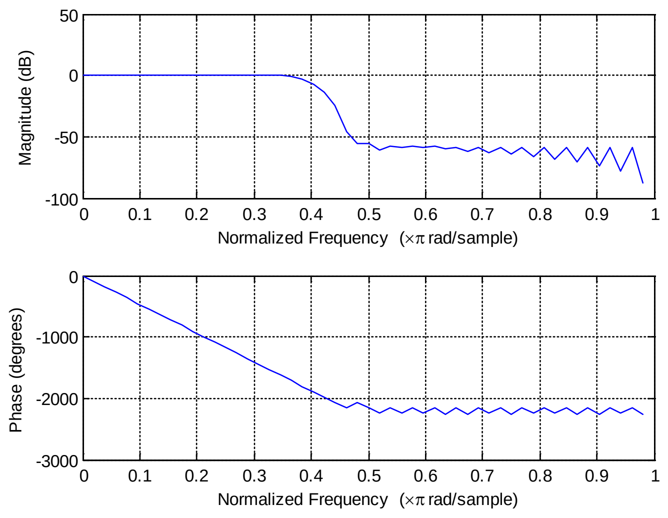


Figure: Top: Frequency response Bottom: Phase response

Lab 10: Design the FIR filter using matlab window function FIR filter design using the window method.

```
b = fir1(order,cutoff) designs an N'th order lowpass FIR digital filter
and returns the filter coefficients in length N+1 vector B.
The cut-off frequency must be between 0 < Wn < 1.0, with 1.0
corresponding to half the sample rate.

b=fir1(order,cutoff,'high') -HP
b=fir1(order,cutoff,'bandpass'), here cutoff=[w1,w2], also see help
fir1
```



MATLAB COMMANDS

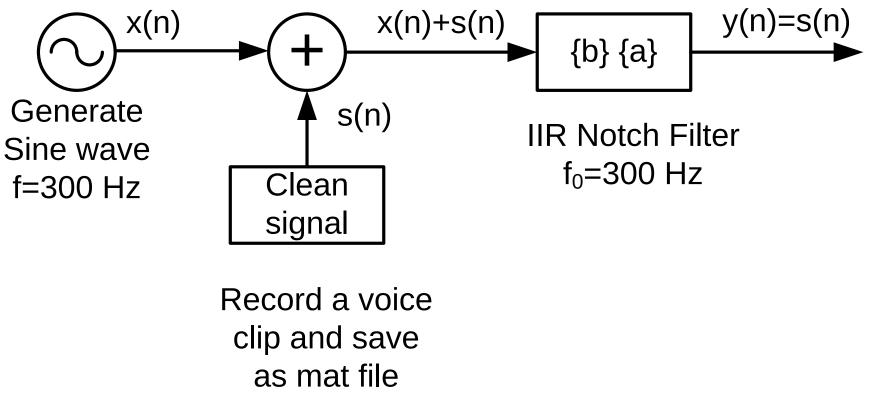
```
>>close all
>>clear all
>>order=52;
>>cutoff=0.4;
>>b=fir1(order, cutoff);
>>freqz(b,1,order)
```

LAB 11: Design of an IIR Notch Filter and apply it to remove an unwanted sinusoid from a sound clip

A disturbing sine signal is added purposefully with a clean sound signal. The signal now becomes noisy. Then filter the noisy signal with the notch filter. Get the clean signal after filtering it using the designed notch filter.

Ref. Digital Sig Processing .. -by EC Ifeachor.

This code written by - Prof. Asadul Huq. 28-04-10.

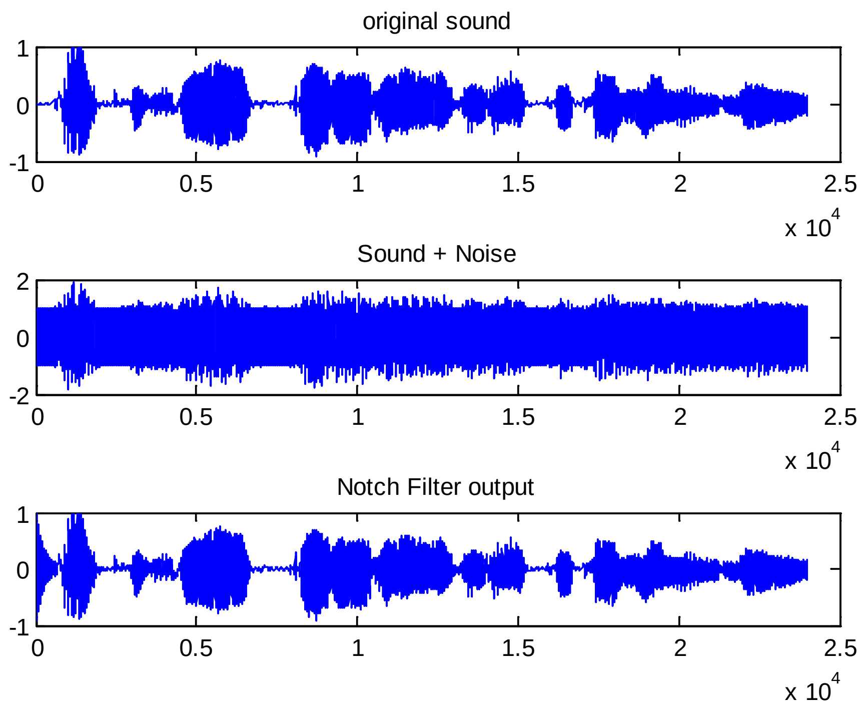


Notch filter application:
File: <notch_app.m>

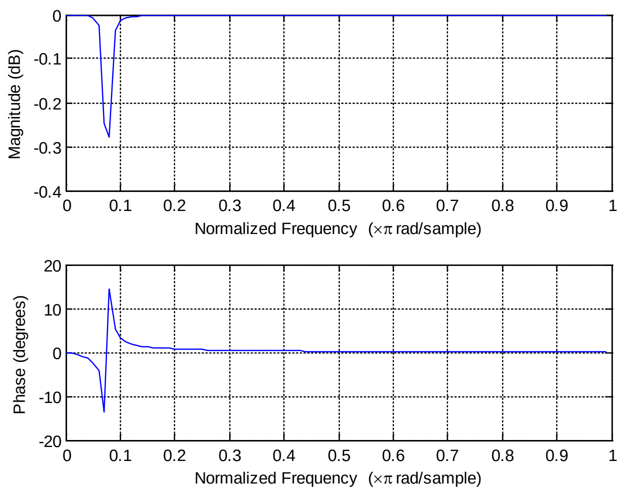
MATLAB COMMANDS

```
>>clear all; close all
>>dur=3;%disturbing sine wave time duration in sec
>>f=300;% sine wave frequency
>>fs=8000;% filter sampling frequency
>>n=1:dur*fs; %no of samples
>>t=n/fs;% no. of samples converted into time in sec
>>xn=sin(2*pi*f*t);% gen the sine wave with freq f and duration dur
%plot(xn)
%sound(xn,fs)%listen to disturbing sine wave
>>load pakhi;
>>xplusn=sounddata+xn.'; %orginal sound(sounddata) mixed with un wanted
sine xn
```

```
>>w0=f/(fs/2);%w0=notch frequency
>>bw=w0/30;
>>[b,a]=iirnotch(w0,bw);% notch filter function is used to get IIR
filter coefficients
>>y=filter(b,a,xplusn);%filter the noisy signal using b and a
%sound(y,fs)%listen to filtered sound
```



The frequency and Phase response follows


Magnitude (dB)
Phase (degrees)
Normalized Frequency (×π rad/sample)