MATLAB PROJECT 1: BAND STOP FILTER (ECE 535)

By Sai Kasyap Kamaraju

A simple discrete-time notch filter is defined by the frequency response:

$$H(e^{j\omega}) = \frac{(1 - e^{-j(\omega-\omega_0)})(1 - e^{-j(\omega+\omega_0)})}{((1 - re^{-j(\omega-\omega_0)})(1 - re^{-j(\omega+\omega_0)})}$$ where r and ω0 are constants

1. The difference equation of the notch filter is found out as shown below :

$$H(e^{j\omega}) = \frac{\left(1-e^{-j(\omega-\omega_0)}\right)\left(1-e^{-j(\omega+\omega_0)}\right)}{\left(1-re^{-j(\omega-\omega_0)}\right)\left(1-re^{-j(\omega+\omega_0)}\right)}$$

$$= \frac{1-e^{-j(\omega-\omega_0)} - e^{-j(\omega+\omega_0)} + e^{-2j\omega}}{1-r\left(e^{-j(\omega-\omega_0)}+e^{-j(\omega+\omega_0)}\right)+r^2e^{-2j\omega}}$$

$$= \frac{1-2e^{-j\omega}\cos\omega_0 + e^{-2j\omega}}{1-2re^{-j\omega}\cos\omega_0 + r^2e^{-2j\omega}} \quad \text{--(1)}$$

Converting above equation to z-Transform by substituting $z = e^{+j\omega}$

$$\Rightarrow H(z) = \frac{1-2\cos\omega_0 \cdot z^{-1} + z^{-2}}{1-2r\cos\omega_0 z^{-1} + r^2z^{-2}}$$

But $H(z) = Y(z)/X(z)$

$$\Rightarrow (1-2r\cos\omega_0 z^{-1}+r^2z^{-2})Y(z) = (1-2\cos\omega_0 z^{-1}+z^{-2})X(z)$$

Applying Inverse z-Transform to find out LCCDE

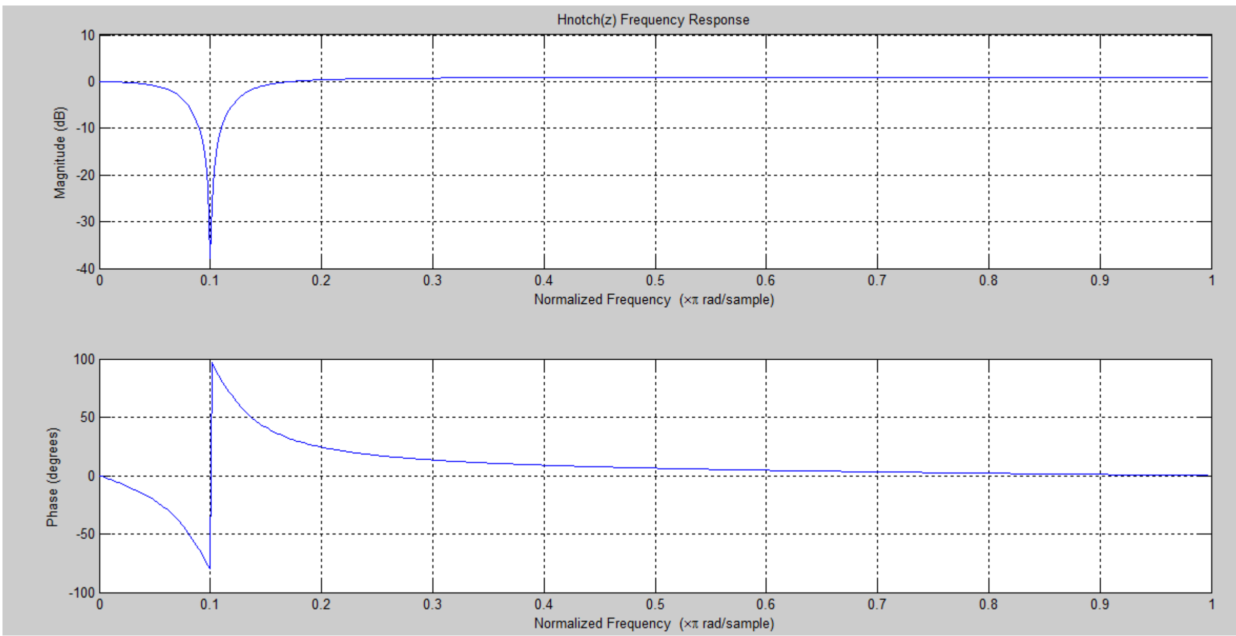
$$\boxed{y[n] - 2r\cos\omega_0 \cdot y[n-1] + r^2 y[n-2] = x[n] - 2\cos\omega_0 x[n-1] + x[n-2]}$$

$$\Rightarrow y[n] = x[n] - 2\cos\omega_0 x[n-1] + x[n-2] + 2r\cos\omega_0 y[n-1] - \cancel{+} r^2 y[n-2]$$

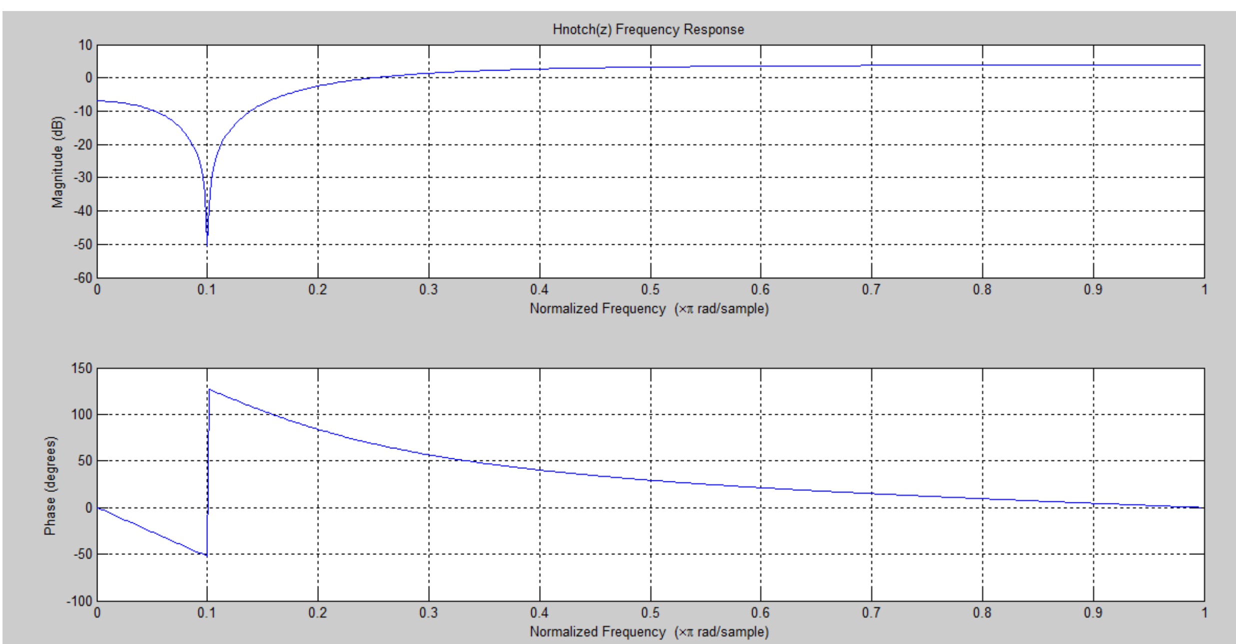
2. For the system to be stable and causal, the poles should be inside unit circle.
   This implies that the magnitude of r should be < 1.
   For the ROC, $|z| > r$.
   Hence the system is causal and stable for above conditions
3. For this question, as we need to use different 3 values of r and Wo; It is better to create a function in MATLAB.

Case 1 : system is stable :

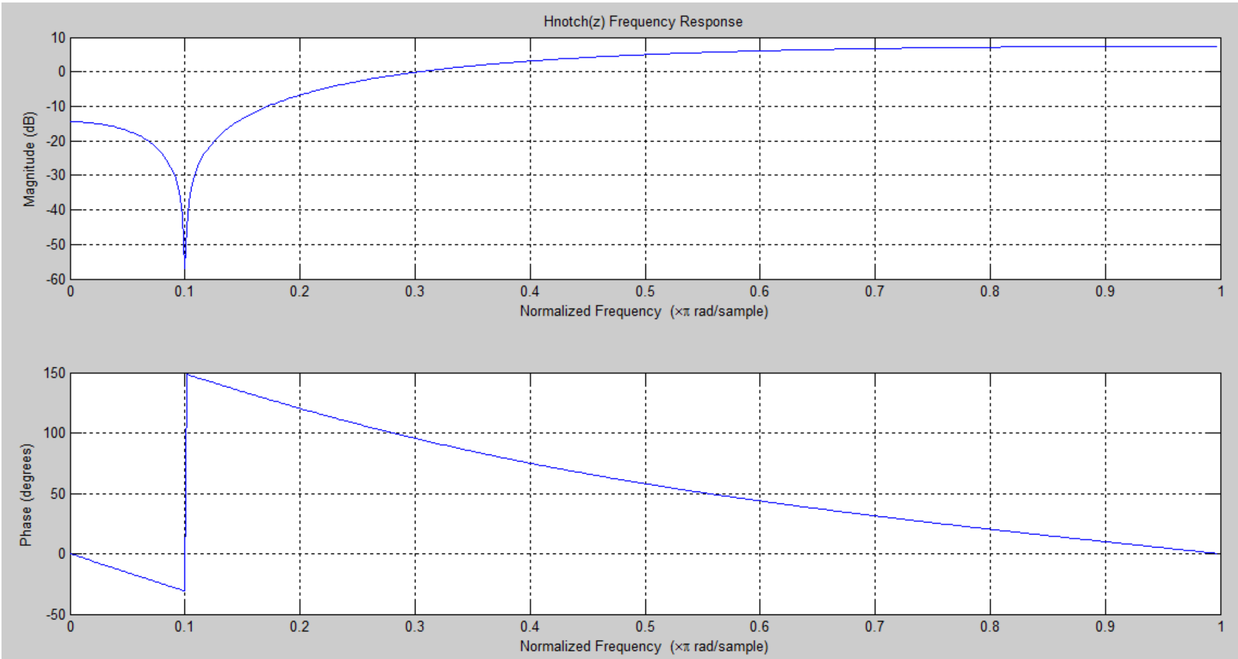
Case 1a : Where r = 0.9, Wo = 0.1*pi



For r =0.6 ,Wo = 0.1*pi;



For r = 0.3 , Wo = 0.1*pi;



Observation:

Magnitude plot:

Keeping Wo constant, Varying r: as r value is high the magnitude is close to 0 dB line, as r decreases the value of magnitude is decreasing in –ve db(0.9 – close to zero db line , 06 – close to -10 db line 0.3 – close to -15 b line)

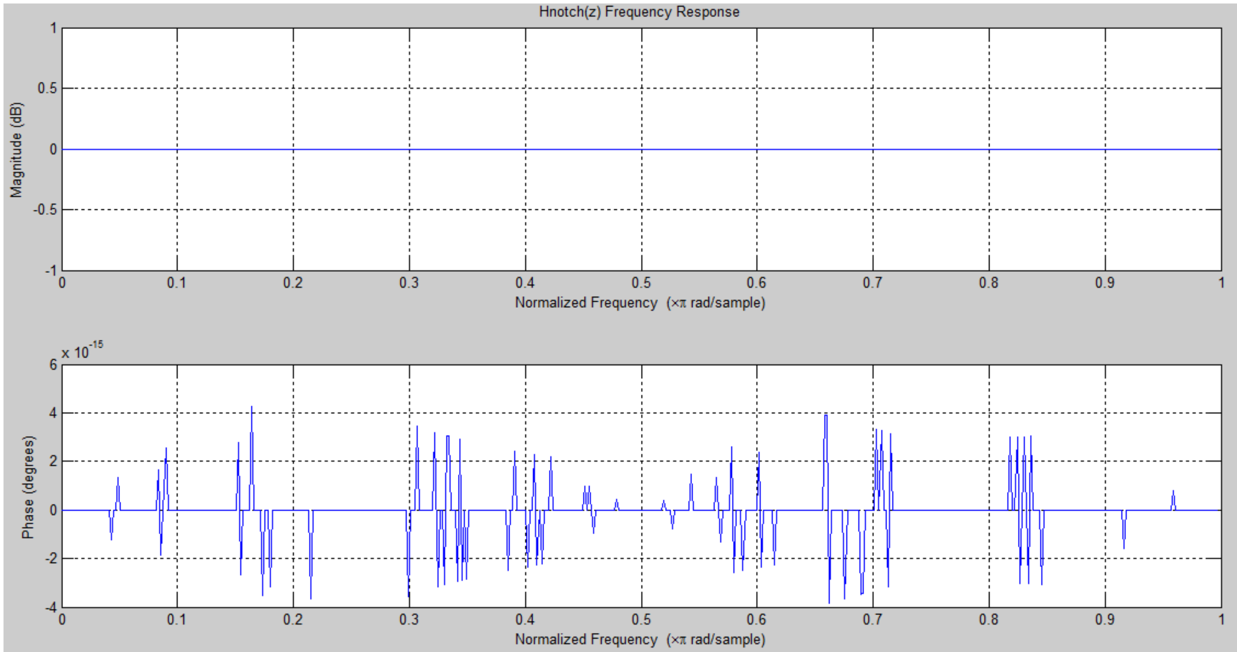
after the frequency of Wo magnitude plot attains steady state. The notch appears exactly at Wo value(minimum at Wo).

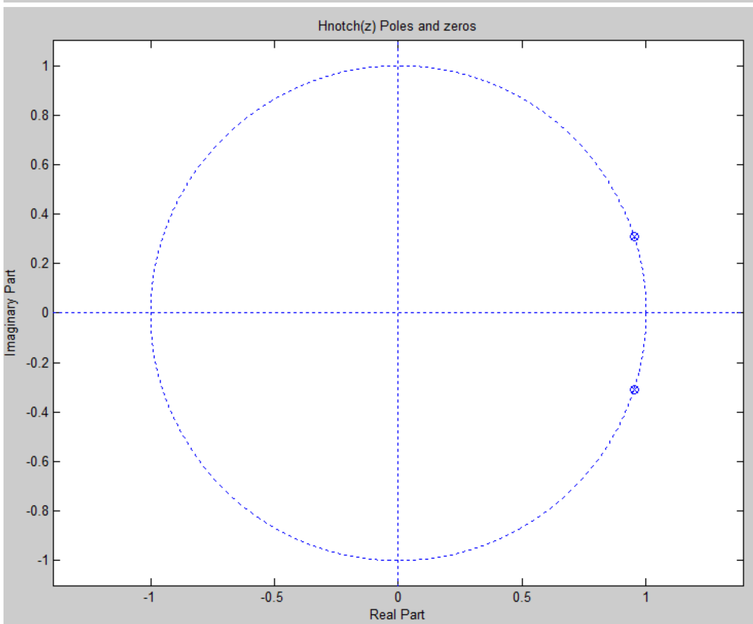
The magnitude of the steady state is given by -20log(r).

In the phase plot , there is change of phase (-ve to +ve) at the value of Wo.

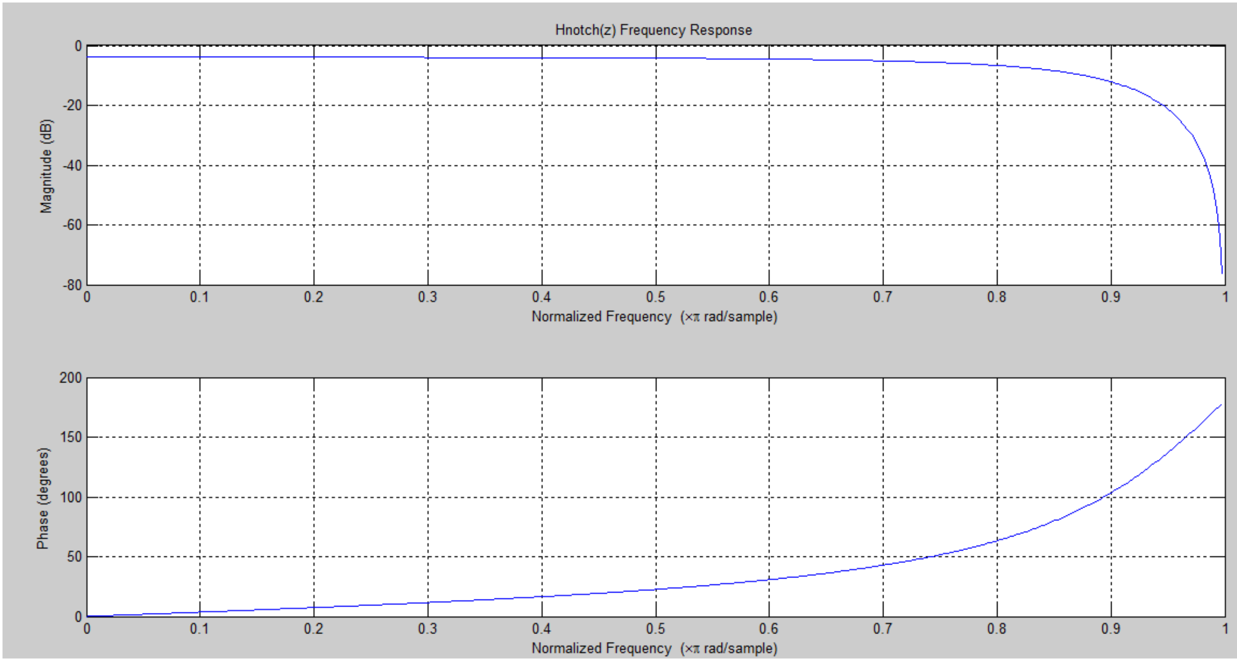
For the unstable and marginally stable systems

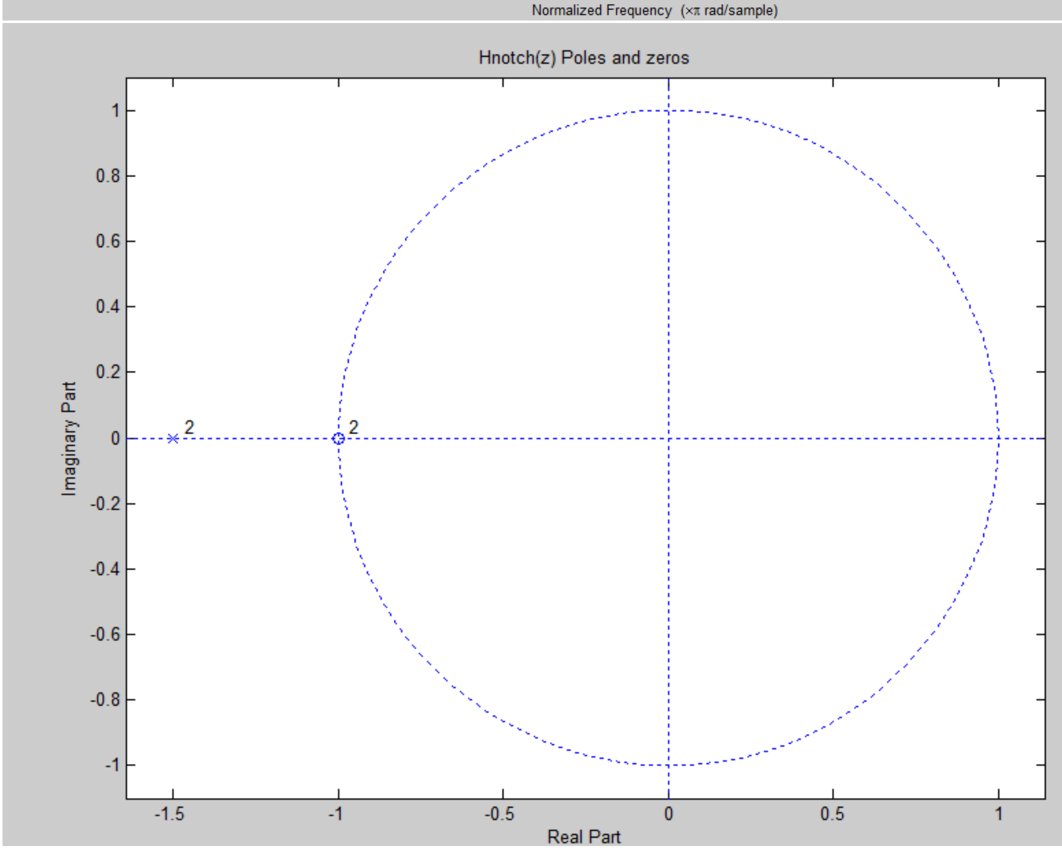
Case 2 : for r = 1, Wo = 0.1*pi marginally stable





Case 3 : for r= 1.5 , Wo = pi; unstable





Observation: For marginally systems the magnitude of H(z) is zero
For unstable systems, the magnitude tends towards negative infinity.

4. A analog signal consisting of an undesured 440Hz component. Sampling frequency = 44100 Hz.

a. According Nyqusit sampling theorem , to avoid aliasing ,

Fs >= 2* Fn (where Fs = sampling frequency, Fn = highest frequency contained in that signal)

This implies, Fn <= Fs/2/

So the highest frequency that can be contained in the signal is Fs/2.

=>Fn = 22050

Therefore, the highest frequency that can be contained in the analog signal = 22050 Hz

b. The value of Wo for the notch filter to eliminate the 440 Hz component :

The relationship between discrete and continuous freq is as follows

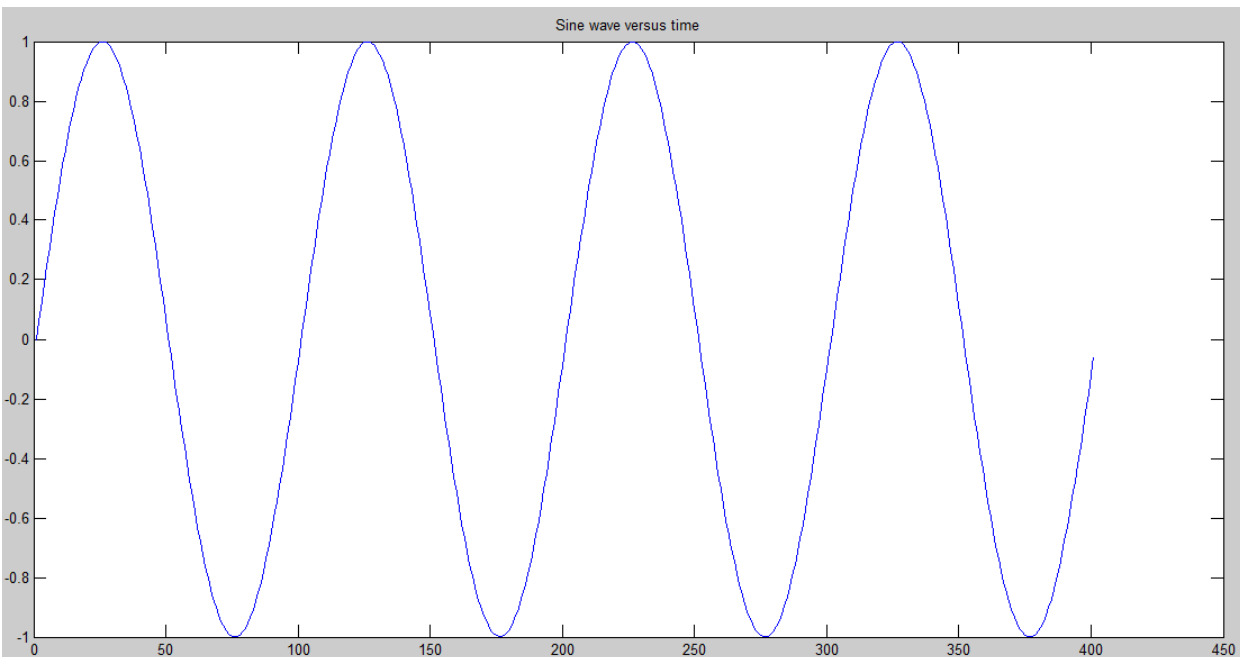
Fd = Fc*T

T=1 / Fs.

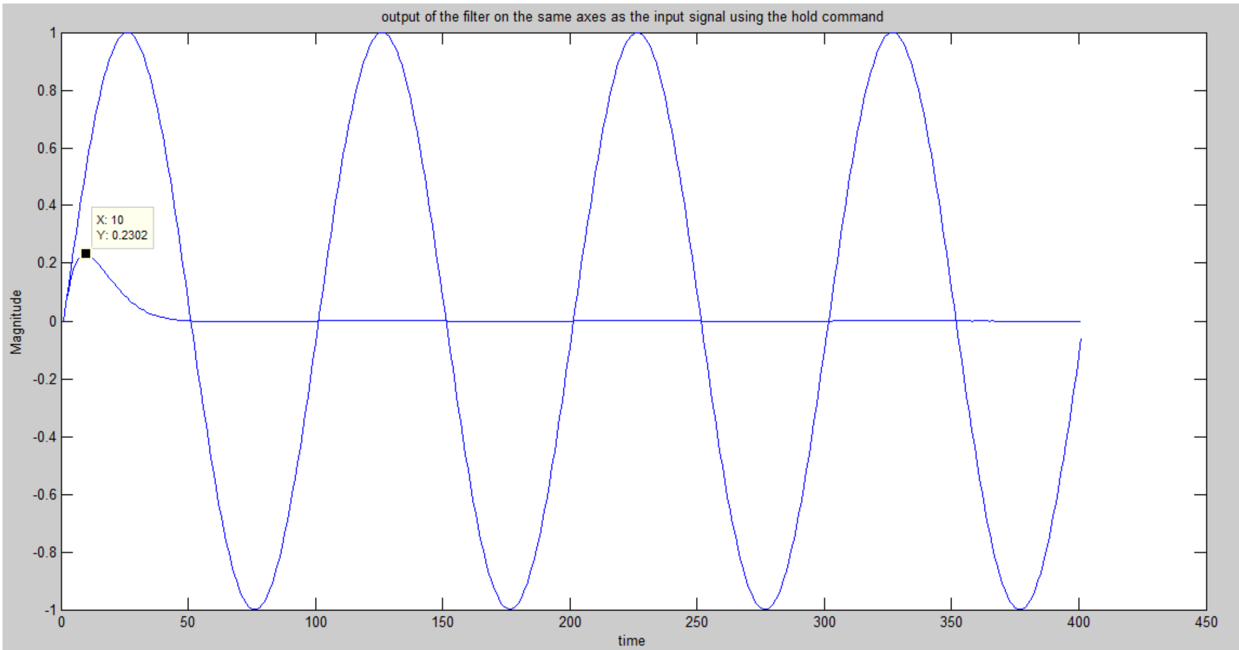
Fd = Fc/Fs.

Wd = 2*pi*Fc/Fs
 = 2*pi*440/44100
 ~ 0.02*pi.

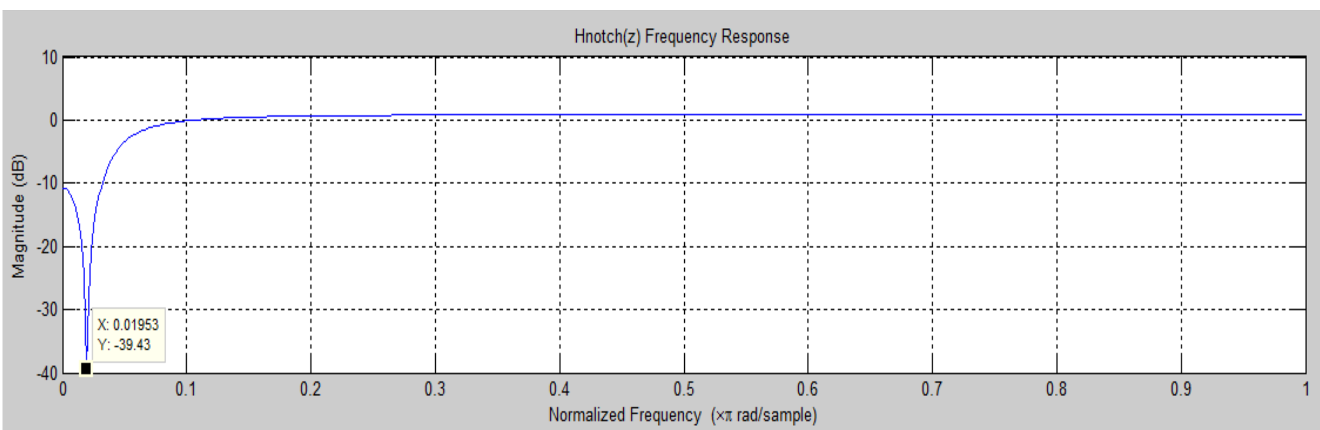
c.



d.



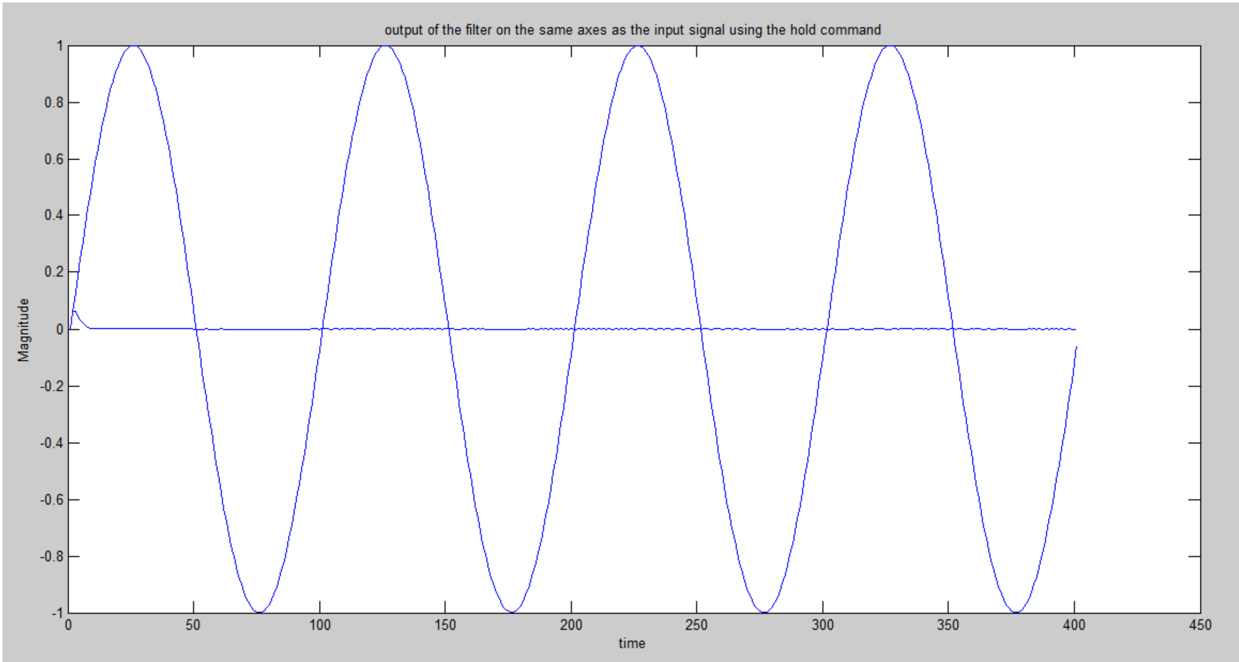
Yes , notch filter eliminates the 440 Hz freq.



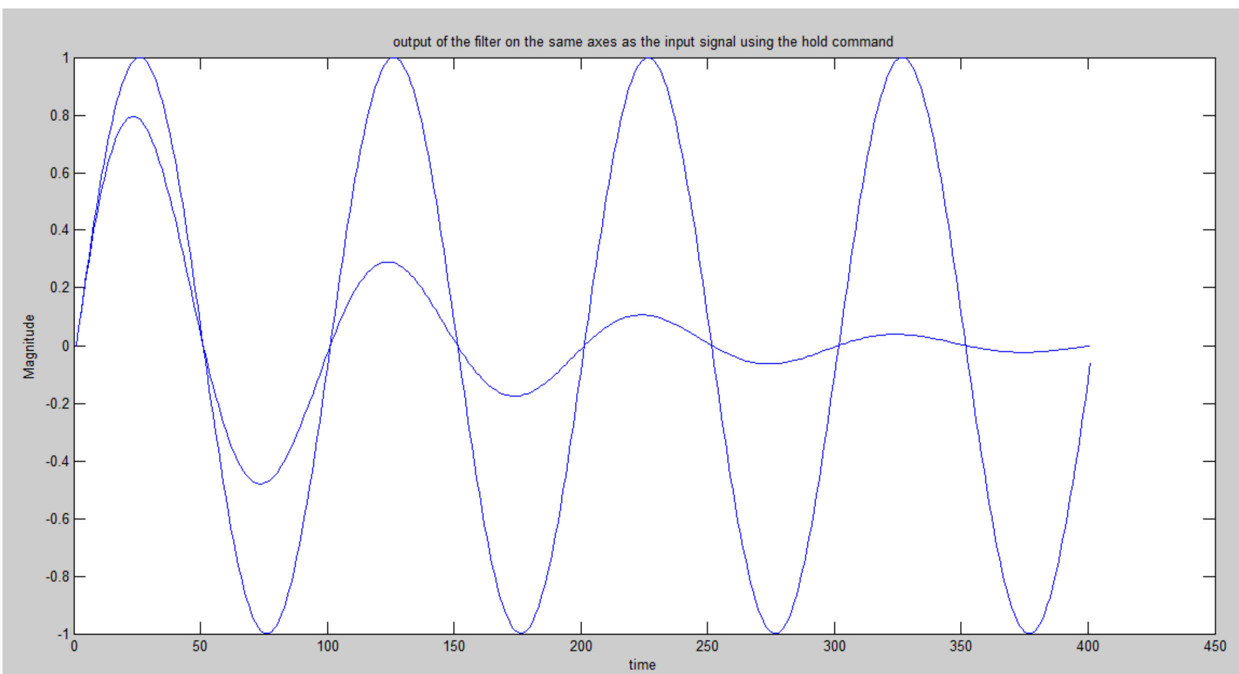
X= 0.01953 pi rad/ samples i.e 440 Hz.

e.)

for r = 0.5



for r = 0.99



The transient response depends on the value of r. As the value of r increases, the transient response increase. For smaller values of r we have less or no transient response.

(4e) Analytical Form of the Response

$$\sin(\omega_0)\, u[n] \overset{Z\text{-transform}}{\Longleftrightarrow} \frac{\sin(\omega_0) z^{-1}}{1 - 2\cos(\omega_0) z^{-1} - z^{-2}}$$

$$e^{-j\omega} = z^{-1} \quad \Updownarrow \text{F.T.}$$

$$\frac{\sin(\omega_0) e^{-j\omega}}{1 - 2\cos(\omega_0) e^{-j\omega} - e^{-2j\omega}}$$

$$Y(e^{j\omega}) = H(e^{-j\omega}) \cdot X(e^{-j\omega})$$

$$= \frac{1 - 2\cos\omega_0 e^{-j\omega} - e^{-2j\omega}}{1 - 2r\cos\omega_0 e^{-j\omega} - r^2 e^{-2j\omega}} \times \frac{\sin(\omega_0) e^{-j\omega}}{1 - 2\cos(\omega_0) e^{-j\omega} - e^{-2j\omega}}$$

$$= \frac{\left(1 - 2\cos\omega_0 e^{-j\omega} - e^{-2j\omega}\right)\left(\sin\omega_0\right) e^{-j\omega}}{\left(1 - 2r\cos\omega_0 e^{-j\omega} - r^2 e^{-2j\omega}\right)\left(1 - 2\cos(\omega_0) e^{-j\omega} - e^{-2j\omega}\right)}$$

The transient response depends on the value of the 'r' as shown in Analytical form

So the transient increase, the value of 'r' increase (denominator decrease) Amplitude increase.

5. a. load('River_Short.mat')

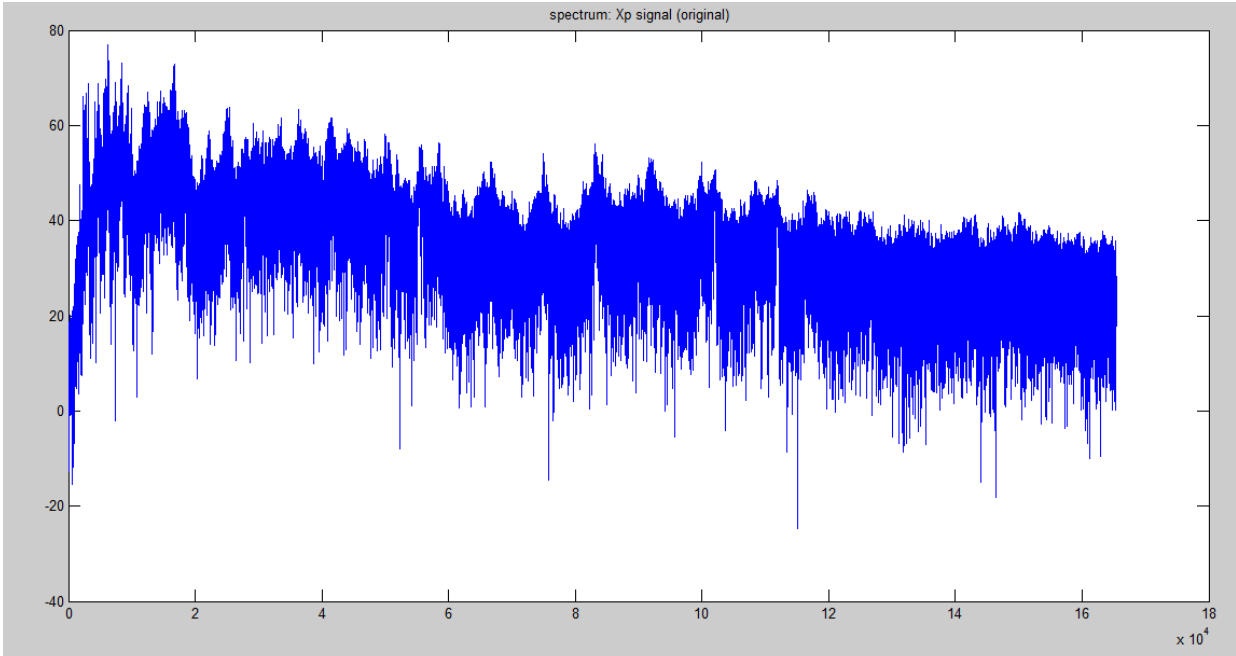
```
soundsc(xp,Fs)
```

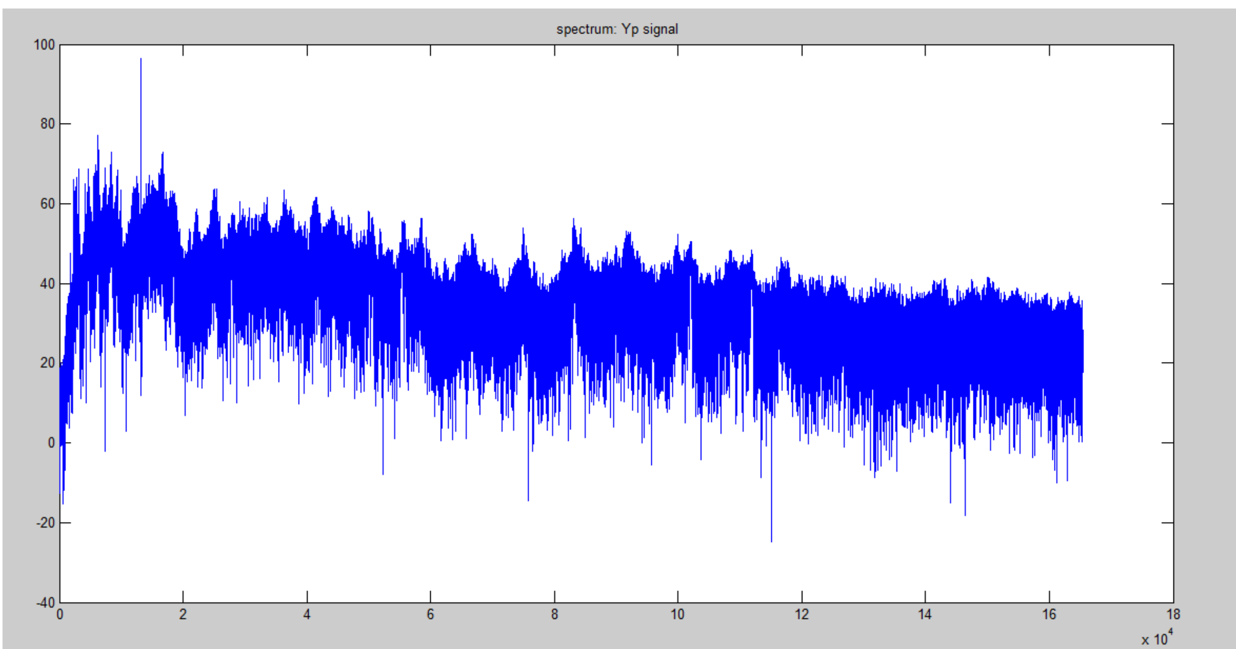
to play the original song.

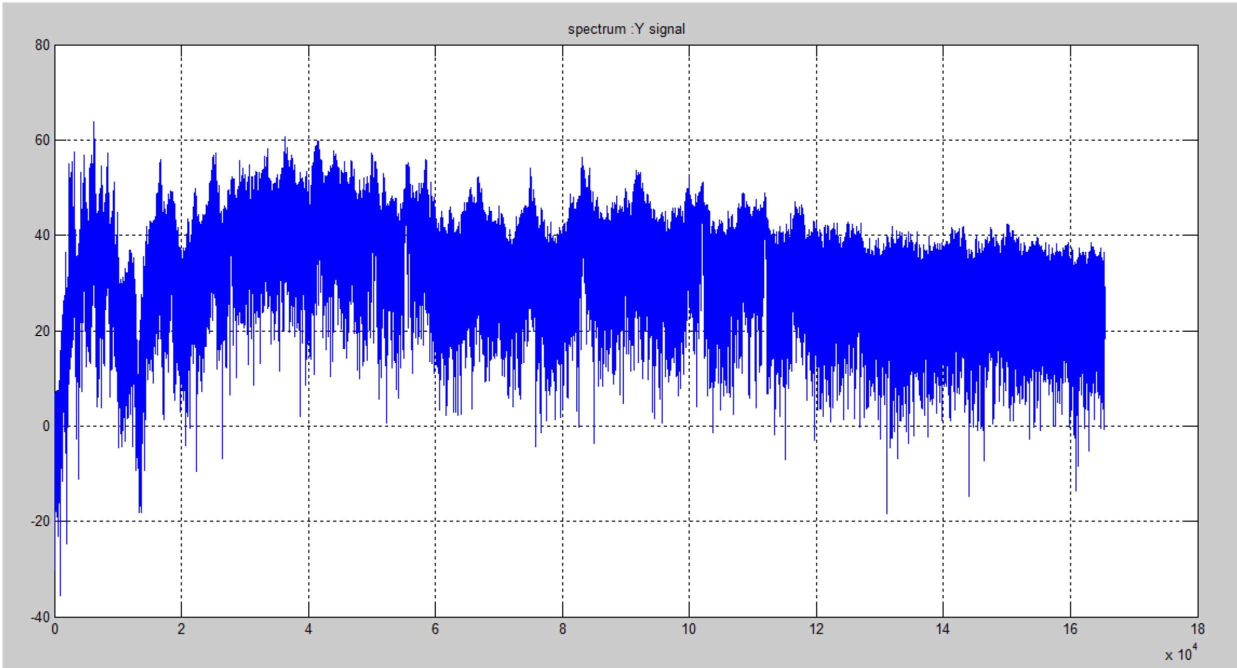
b. For r = 0.9, Wo = 2*pi*440/fs;

Xp – original signal , y - filtered signal after single notch.

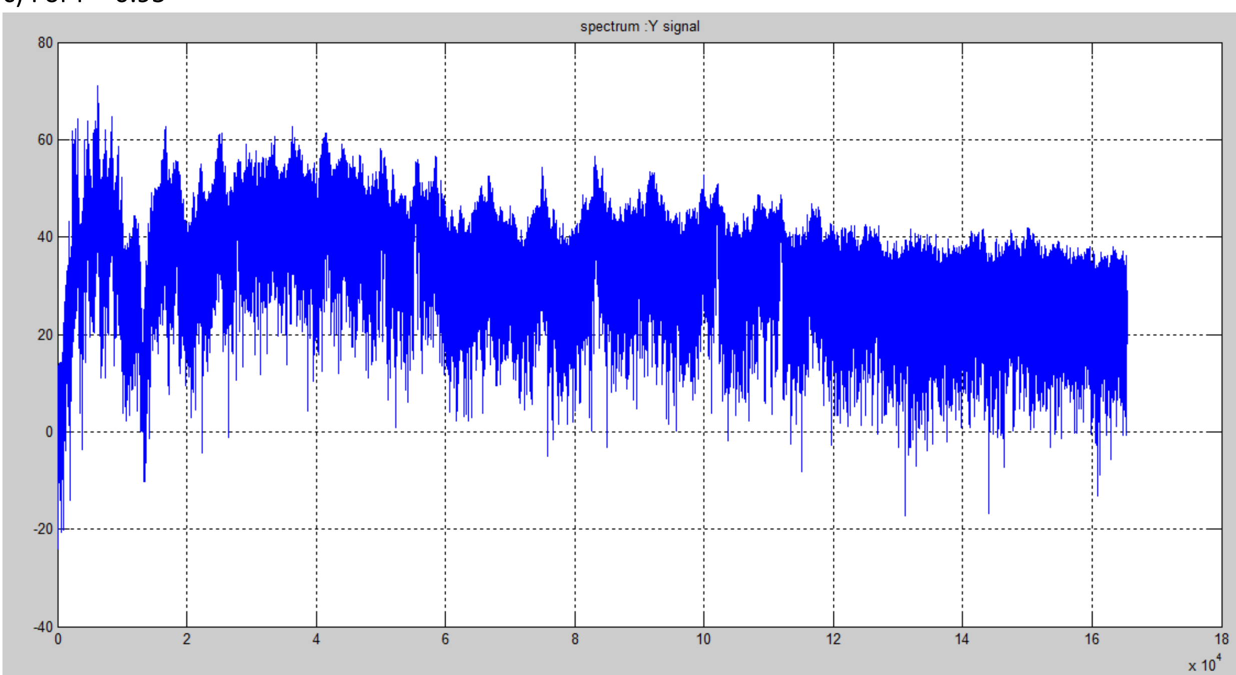
Yes the filter eliminates 440Hz frequency


spectrum: Xp signal (original)
80
60
40
20
0
-20
-40
0
2
4
6
8
10
12
14
16
18
x 10^4

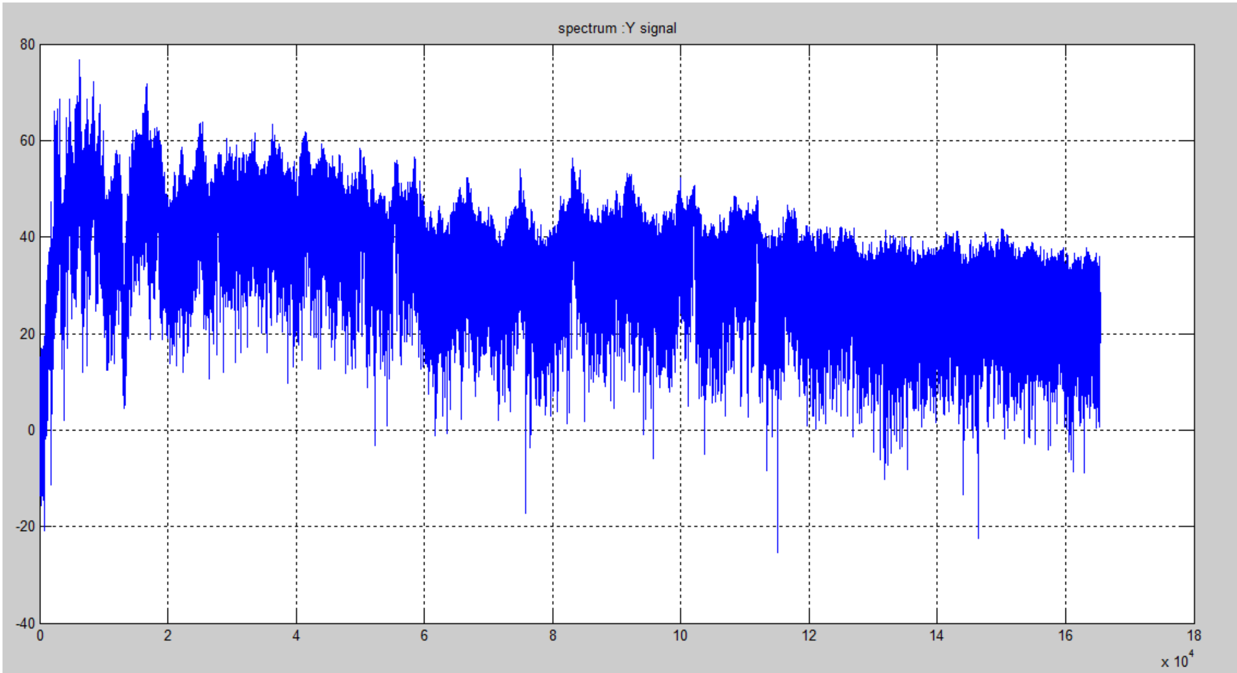

spectrum: Yp signal
100
80
60
40
20
0
-20
-40
0
2
4
6
8
10
12
14
16
18
x 10^4


spectrum :Y signal
80
60
40
20
0
-20
-40
0
2
4
6
8
10
12
14
16
18
x 10^4

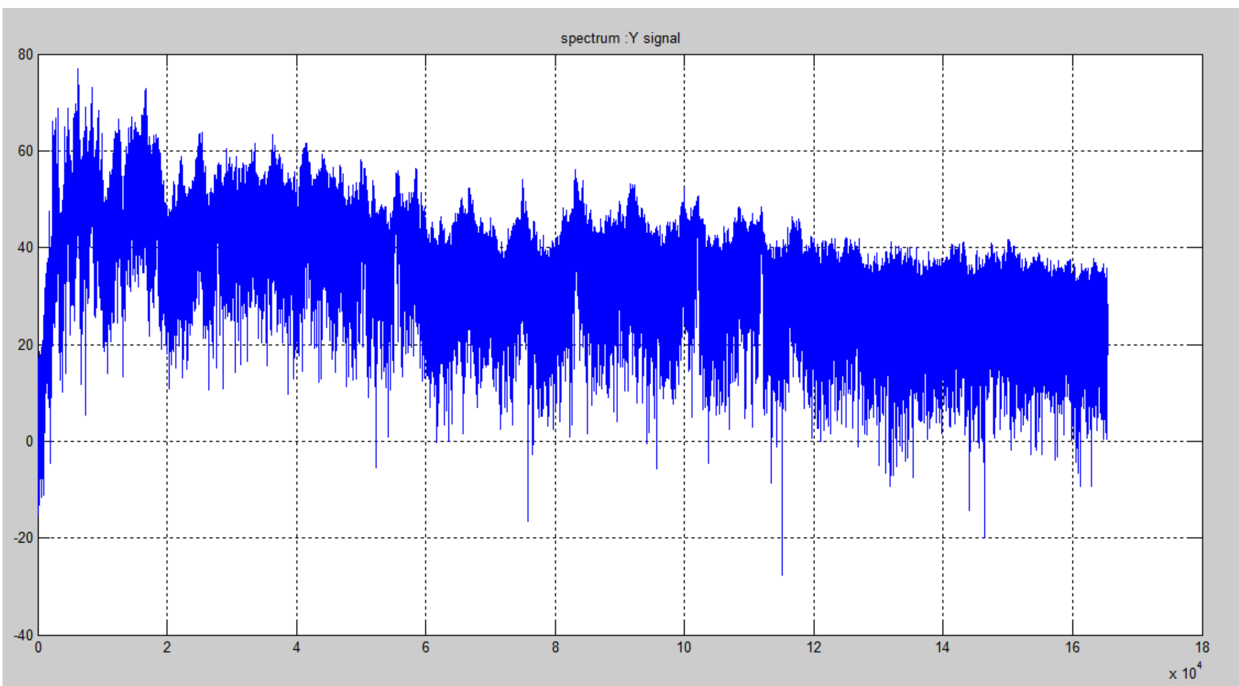
c) For r = 0.95


spectrum :Y signal
80
60
40
20
0
-20
-40
0
2
4
6
8
10
12
14
16
18
x 10^4

For r = 0.99



For r = 0.999



Observation: The amplitude (pitch) of the signal (here in the x-axis initially in the domain <2) increases gradually as we increase the value of 'r' .

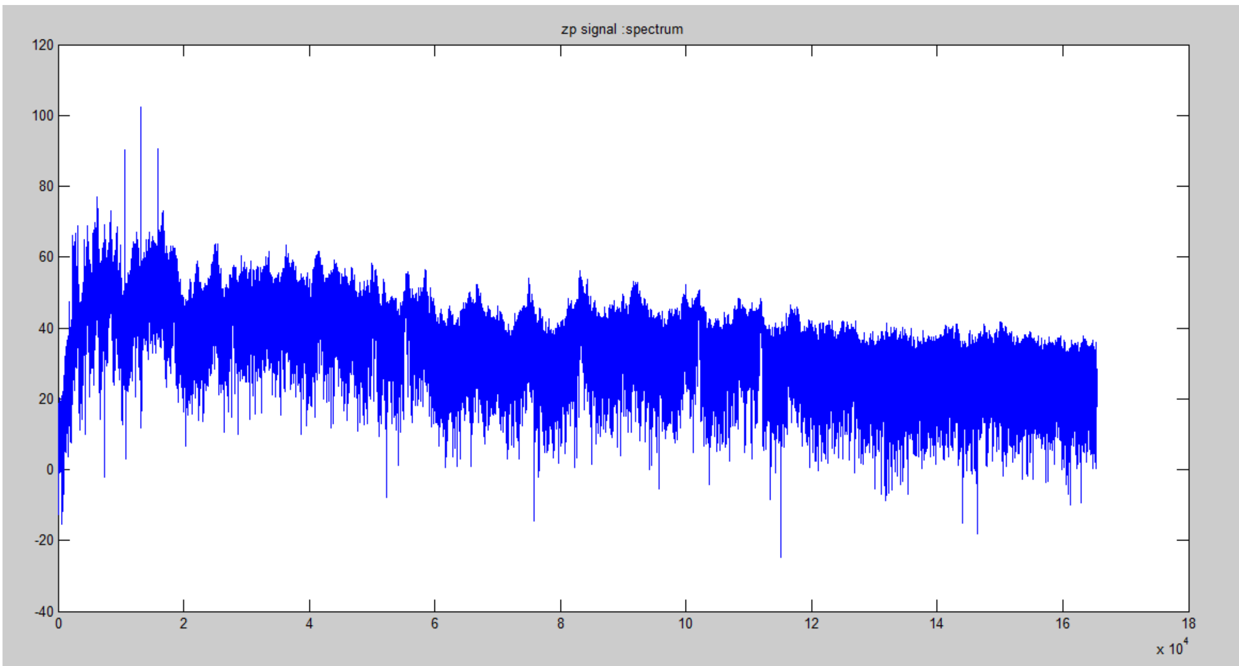
d) Triple Notch filter :

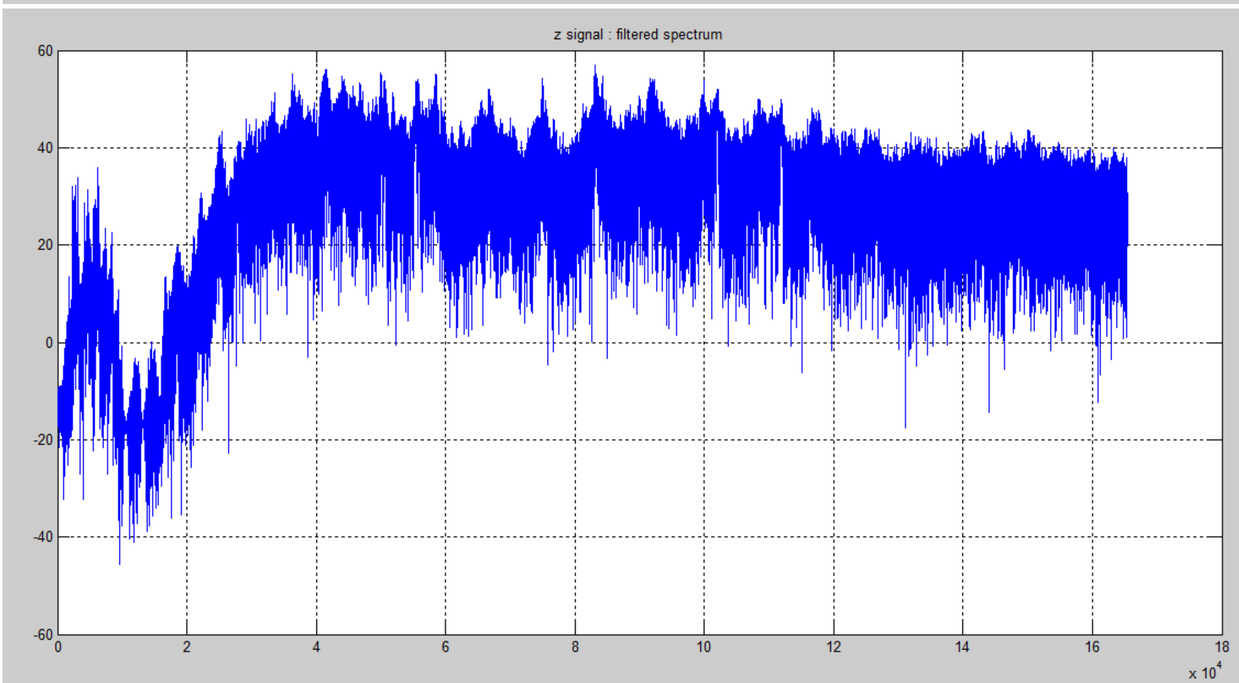
A Triple notch filter is a cascaded connection of 3 filters.

So , The approach in the code is done by using 3 filter functions implemented one after the other.

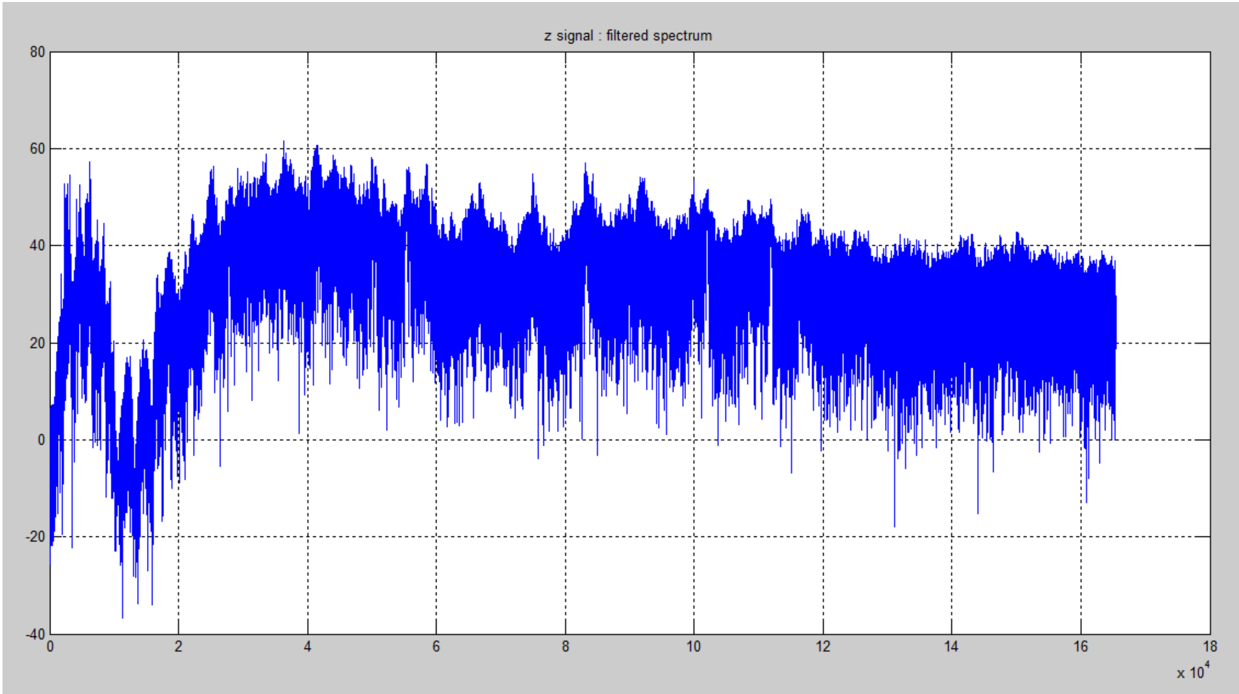
The conventional way is multiplying all the zeros and poles of the system (It becomes 6^{th} order) and use a filter function to get the filtered output.

Here Z is the final filtered signal.

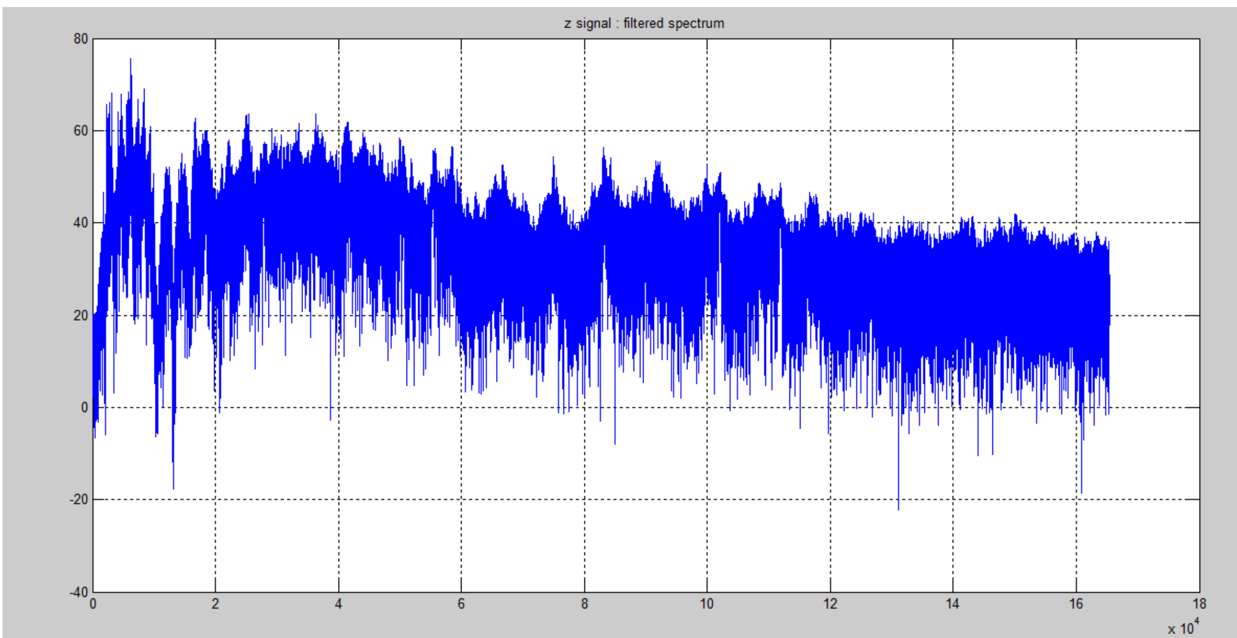




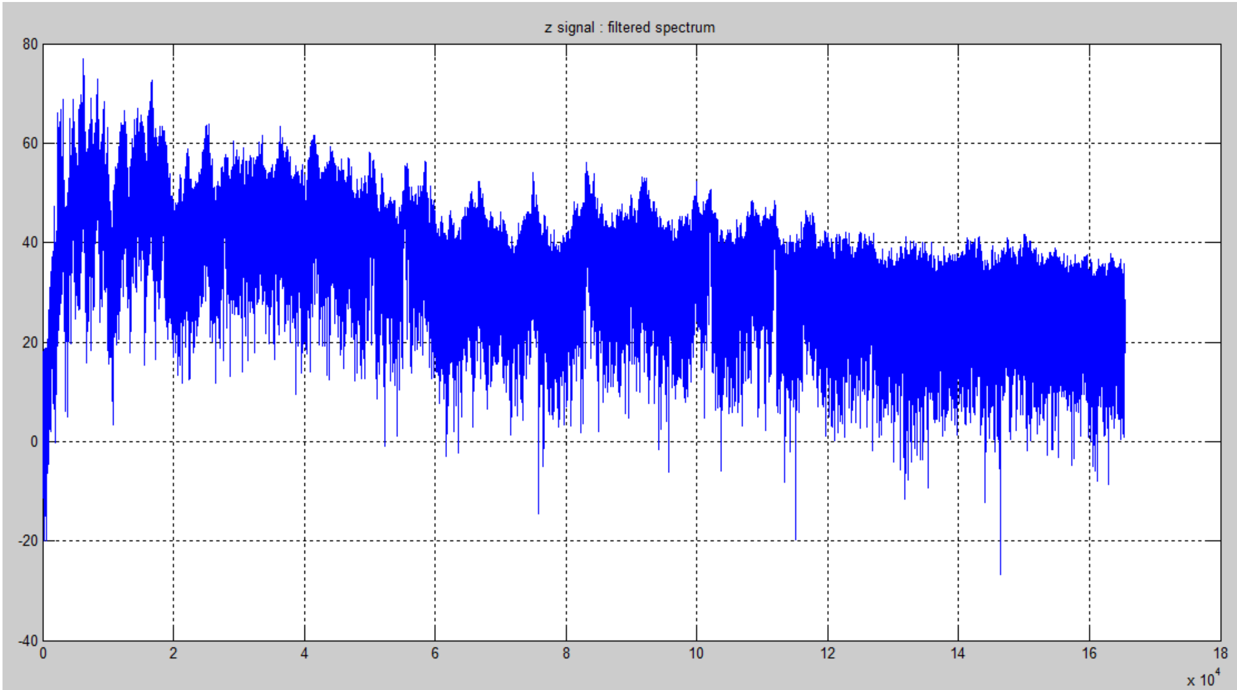
e) for r = 0.95



for r =0.99



For r =0.999



Observation:
As the value of ‘r’ increases, the initial pitch of the filtered signal increases which increases the noise also.

e) The potential problem in triple notch filter (I felt) is increase of r causes high pitch and noise (little) and if the value of r is very low , you cannot hear the filtered signal properly (for r =0.2). So the design of ‘r’ is critical for the triple notch filter.

Other method :
We can use comb filters when the required notch is periodic.

Appendix

MATLAB code

QUESTION 2 AND 3 :

```
function [] = test1(r,Wo)
% Question 2 and 3 :
% Simple Notch Filter Example

%Wo = 2*pi*F/Fs;

%numerator ----> a
a1 = [1 -exp(j*Wo)]; a2 = [1 -exp(-j*Wo)];
a = conv(a1,a2);
% Denominator ---> b
b1 = [1 -r*exp(j*Wo)]; b2 = [1 -r*exp(-j*Wo)];
b = conv(b1,b2);

figure
zplane(a,b); title('Hnotch(z) Poles and zeros')
figure
freqz(a,b); title('Hnotch(z) Frequency Response')
end
```

FOR QUESTION 4 C

```
% questionn 4 c
% Generation of Sine Wave

Fs = 44100;
t =0:1/Fs:400/Fs;
f= 440;

x = sin(2*pi*f*t);
% using plot function to plot the signal
figure;
plot(x);
title('Sine wave versus time');
xlabel('time');
ylabel('Magnitude');
```

QUESTION 4 D

```
% questionn 4 d
% Generation of Sine Wave
r =0.5;

Fs = 44100;
t =0:1/Fs:400/Fs;
f= 440;
% Wo value
Wo = 2*pi*f/Fs;

x = sin(2*pi*f*t);

%using filter function

d = [1 -2*cos(Wo) 1];
c = [1 -2*r*cos(Wo) r^2];

y = filter(d,c,x);
figure;
plot(y);
hold on;
plot(x);
plot(y);
xlabel('time');
ylabel('Magnitude');
title('output of the filter on the same axes as the input signal using the
hold command')
 % Plot the X values vs. the Y values
```

QUESTION 5 B and C

```
%r=0.9
%r=0.95
%r=0.99
r=0.999
Wo = 2*pi*440/fs;
a = [1 -2*cos(Wo) 1];
b = [1 -2*r*cos(Wo) r^2];
%Hearing Original signal
%soundsc(xp,fs);
%Hearing yp signal
%soundsc(yp,fs);
%filtering out the sound using notch filter
y = filter(a,b,yp);
%hearing the signal after filtering the noise in it
soundsc(y,fs)
%figure;
%plot(yp);
%figure;
%plot(y);

N = length(xp);
% Spectrum of org signal Xp
figure;
Xp = fft(xp,N);
plot(20*log10(abs(Xp(1:N/8))))
title(' spectrum: Xp signal (original)');
figure;
Yp = fft(yp,N);
plot(20*log10(abs(Yp(1:N/8))))
title(' spectrum: Yp signal');
%he spectrum of the signal  using FFT
figure;
Y = fft(y,N);
plot(20*log10(abs(Y(1:N/8))))
title(' spectrum :Y signal')
grid
```

Question 5 D and E :

```
%question 5 d and e
% defining values of r :
%r=0.9;
%r=0.95
%r=0.99
%r=0.999
r= 0.2
%intiating angular frerquency values for triple notch
Wo = 2*pi*440/fs;
Wi = 2*pi*(440+88)/fs;
Wii = 2*pi*(440-88)/fs;

%soundsc(yp,fs)
a = [1 -2*cos(Wo) 1];
b = [1 -2*r*cos(Wo) r^2];
% filtering out the sound using notch filter
z1 = filter(a,b,zp);
c = [1 -2*cos(Wi) 1];
d = [1 -2*r*cos(Wi) r^2];
z2 = filter(c,d,z1);
%soundsc(zp,fs)
e = [1 -2*cos(Wii) 1];
f = [1 -2*r*cos(Wii) r^2];
z = filter(e,f,z2);
% hearing the filitered signal
soundsc(z,fs)
N = length(xp);
Zp = fft(zp,N);
%spectrum of Zp signal
figure;
plot(20*log10(abs(Zp(1:N/8))))
title('zp signal :spectrum');
figure;
%spectrum of the signal after triple notch
Z = fft(z,N);
plot(20*log10(abs(Z(1:N/8))))
title('z signal : filtered spectrum')
grid
```