SIGNALS AND SYSTEMS LAB 9

UMAR HABIB
08471

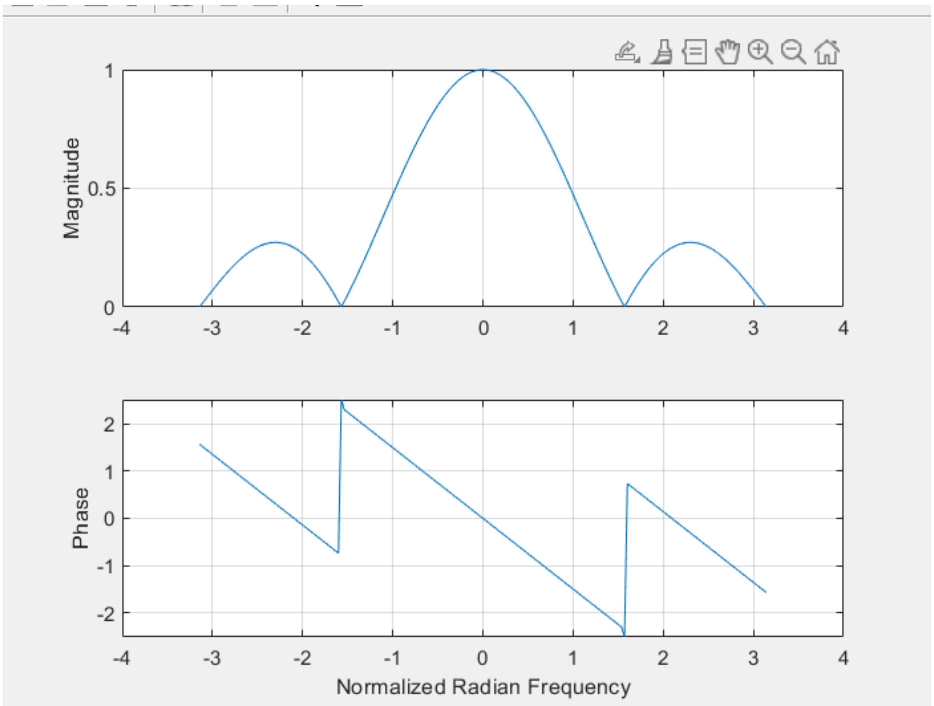
TASK : 01

(a)

CODE:

```
clc
clear all
close all
yn = [0.25,0.25,0.25,0.25];
t = -pi:(pi/100):pi;
H = freqz(yn,1,t);
subplot(2,1,1);
plot(t,abs(H)), grid on
subplot(2,1,2);
plot(t,angle(H)), grid on
xlabel('Normalized Radian Frequency')
```

SNAPSHOT:



(b)The filter is making the signal smoother by removing sudden frequency fast changes and normalizing the magnitude. It lowers the high freq parts and keeps the low freq parts

C)

$$e^{j(\omega n-\omega+\phi)} = e^{j(\omega n+\phi)-j\omega}$$

$$e^{j(\omega n-2\omega+\phi)} = e^{j(\omega n+\phi)-j2\omega}$$

Date ________

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

$$= \frac{1}{4}Ae^{j(\omega n+\phi)} + \frac{1}{4}Ae^{j(\omega(n-1)+\phi)} + \frac{1}{4}Ae^{j(\omega(n-2)+\phi)} + \frac{1}{4}Ae^{j(\omega(n-3)+\phi)}$$

$$= \frac{A}{4}\left[e^{j(\omega n+\phi)} + e^{j(\omega n+\phi)-j\omega} + e^{j(\omega n+\phi)-j2\omega} + e^{j(\omega n+\phi)-j3\omega}\right]$$

$$= \underbrace{Ae^{j(\omega n+\phi)}}_{x[n]}\underbrace{\left[\frac{1}{4} + \frac{1}{4}e^{-j\omega} + \frac{1}{4}e^{-2j\omega} + \frac{1}{4}e^{-3j\omega}\right]}_{H(j\omega)}$$

Now, considering the frequency response

$$H(j\omega) = \frac{e^{-j\omega} + e^{-2j\omega} + e^{0} + e^{-3j\omega}}{4}$$

$$= \frac{1}{4}\left[e^{0.5j\omega-1.5j\omega} + e^{-0.5j\omega-1.5j\omega} + e^{1.5j\omega-1.5j\omega} + e^{-1.5j\omega-1.5j\omega}\right]$$

$$= \frac{1}{2}e^{-1.5j\omega}\left[\frac{e^{0.5j\omega} + e^{-0.5j\omega} + e^{1.5j\omega} + e^{-1.5j\omega}}{2}\right]$$

By Euler's formula: $\frac{e^{j\theta} - e^{-j\theta}}{2} = \cos\theta$

$$H(j\omega) = \frac{e^{-1.5j\omega}}{2}\left[\cos(0.5\omega) + \cos(1.5\omega)\right]$$

∵ multiply and divide by 2

$$= \frac{2\cos(0.5\omega) + 2\cos(1.5\omega)}{4}e^{-1.5j\omega}$$

Proved

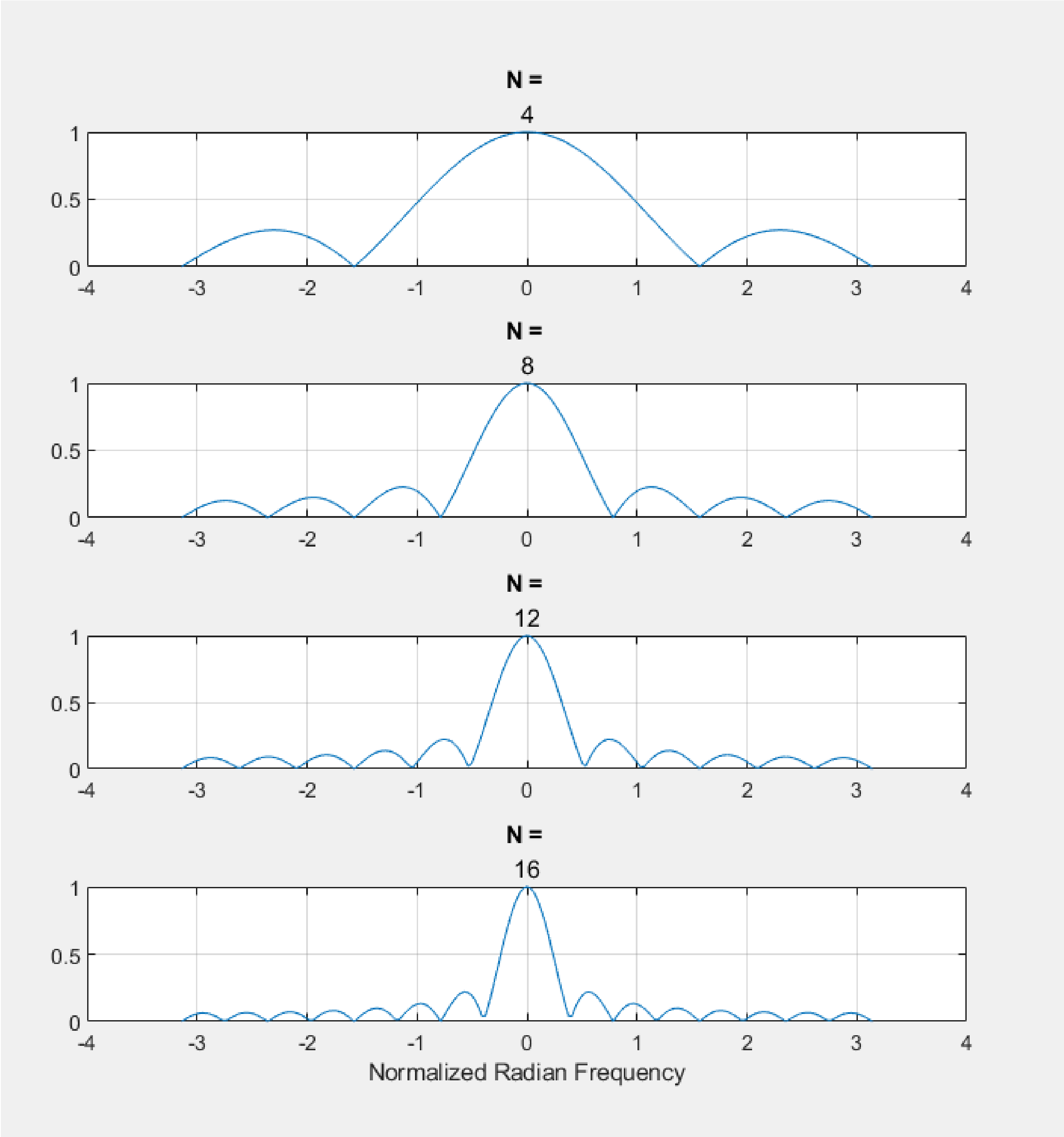
MAAZ

(d)

CODE:

```
figure
for i=4 :+4 : 16
   yn = ones(i,1);
   yn = yn.*(1/i);
   H = freqz(yn,1,t);
   subplot(4,1,i/4);
   plot(t,abs(H)), grid on
   title('N = ',i);
end
xlabel('Normalized Radian Frequency')
```

SNAPSHOT:



OBSERVATION:
As we increase N, the number of samples taken for averaging, the width of the filter, also known as passband width, decreases. While the amplitude and other factors remain the same.

TASK : 02

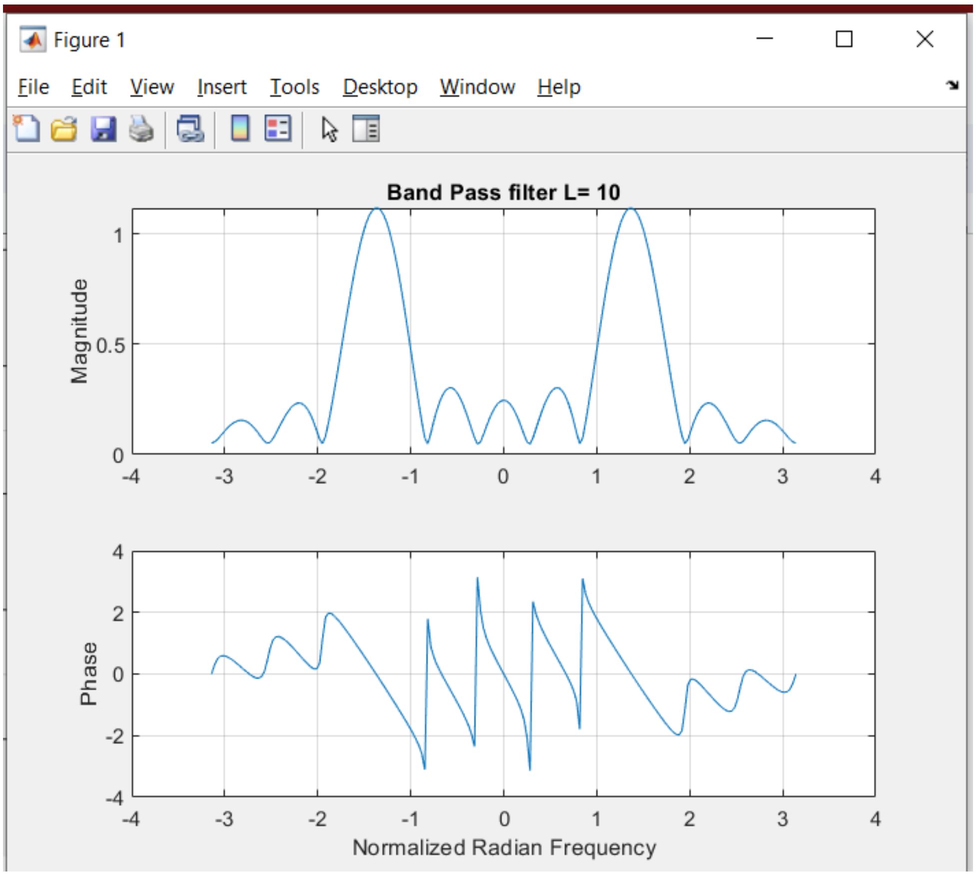
(a)

CODE:

```
%% TASK 02(a)
clc
```

```
clear all
close all
W = 0.44*pi;
L=10;
n=0:L-1;
yn = (2/L).*(cos(W.*n));
t = -pi:(pi/100):pi;
H = freqz(yn,1,t);
figure
subplot(2,1,1);
plot(t,abs(H)), grid on
title('Band Pass filter L= 10')
ylabel('Magnitude')
subplot(2,1,2);
plot(t,angle(H)), grid on
xlabel('Normalized Radian Frequency')
ylabel('Phase')
```

SNAPSHOT:



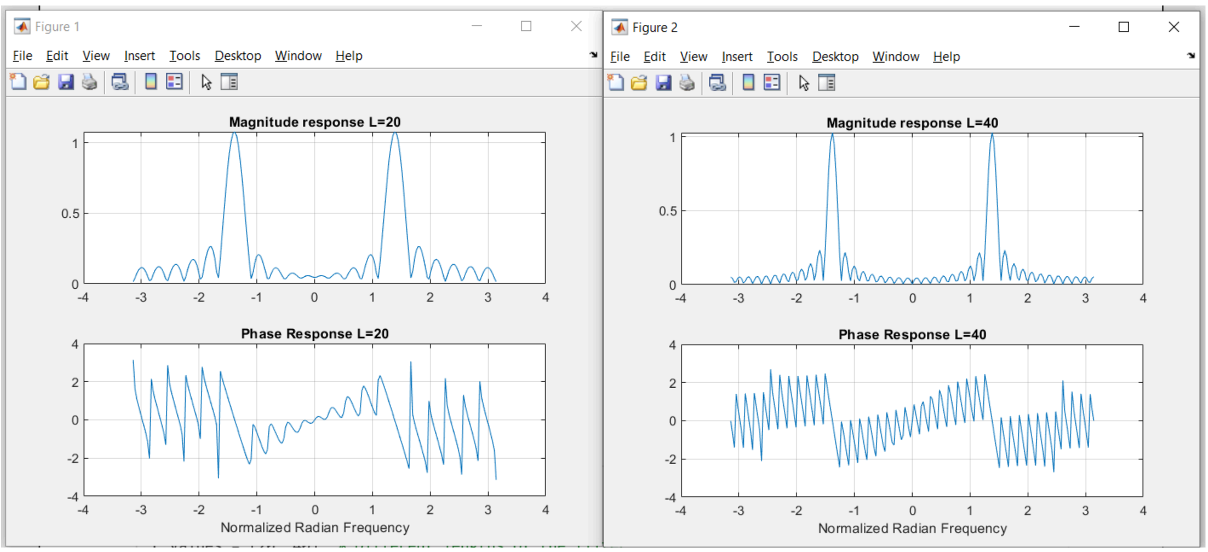
(b)

CODE:

```
clc
```

```
clear all
close all
W = 0.44*pi;
t = -pi:(pi/100):pi; %frequency vector
for L=20 :+20 : 40
   %plotting freq response for L=20 and 40
   n=0:L-1;
   yn = (2/L).*(cos(W.*n));
   H = freqz(yn,1,t);
   figure
   subplot(2,1,1);
   plot(t,abs(H)), grid on
   title(['Magnitude response L=' num2str(L)])
   subplot(2,1,2);
   plot(t,angle(H)), grid on
   title(['Phase Response L=' num2str(L)])
   xlabel('Normalized Radian Frequency')
   %determining width of passband filters
   Hmax = max(abs(H));
   pbwVector = find(abs(H)./Hmax > 0.707); %vector satisfying given
conditions
   L
   passBandWidth = t(pbwVector(end)) - t(pbwVector(1)) %passband width
end
```

SNAPSHOT:



```
L =

    20


passBandWidth =

    3.0159


L =

    40


passBandWidth =

    2.8903

fx >>
```

OBSERVATIONS:

- The band of allowable frequencies will decrease if we increase the value of L. Observing that the magnitude plot is compressed, we can conclude that the range of frequencies that are being allowed is now smaller. This is validated by the smaller value of passband width after increasing L

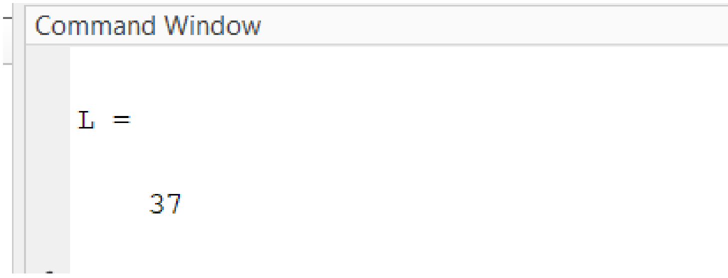
(c)

CODE:

```
clc
clear all
close all
Wlow = 0.3*pi; %lowest frequency
Whigh = 0.7*pi; %highest frequency
w = 0.44*pi ;
t = -pi:(pi/100):pi; %frequency vector
L=1; %initially
n = 0:L-1;
yn = (2/L).*(cos(w.*n)); %frequency response
H= freqz(yn,1,t);
Hmax = max(abs(H));
vect = find( abs(t)<= Wlow | (abs(t) >= Whigh & abs(w) <= pi) ); %vector of
indexes satisfying the given condition for allowed frequencies
%increasing value of L and checking the given condition until it is met
while true
   if( all(abs(10*H(vect))<=Hmax) ) %stop when the condition is met
       break
   end
   %repeating all above steps for the next value of N
   L=L+1;
   n = 0:L-1;
   yn = (2/L).*(cos(w.*n)); %frequency response
```

```
    H = freqz(yn,1,t);
    Hmax = max(abs(H));
    vect = find( abs(t)<= Wlow | (abs(t) >= Whigh & abs(w) <= pi) );
end
L
```

SNAPSHOT:



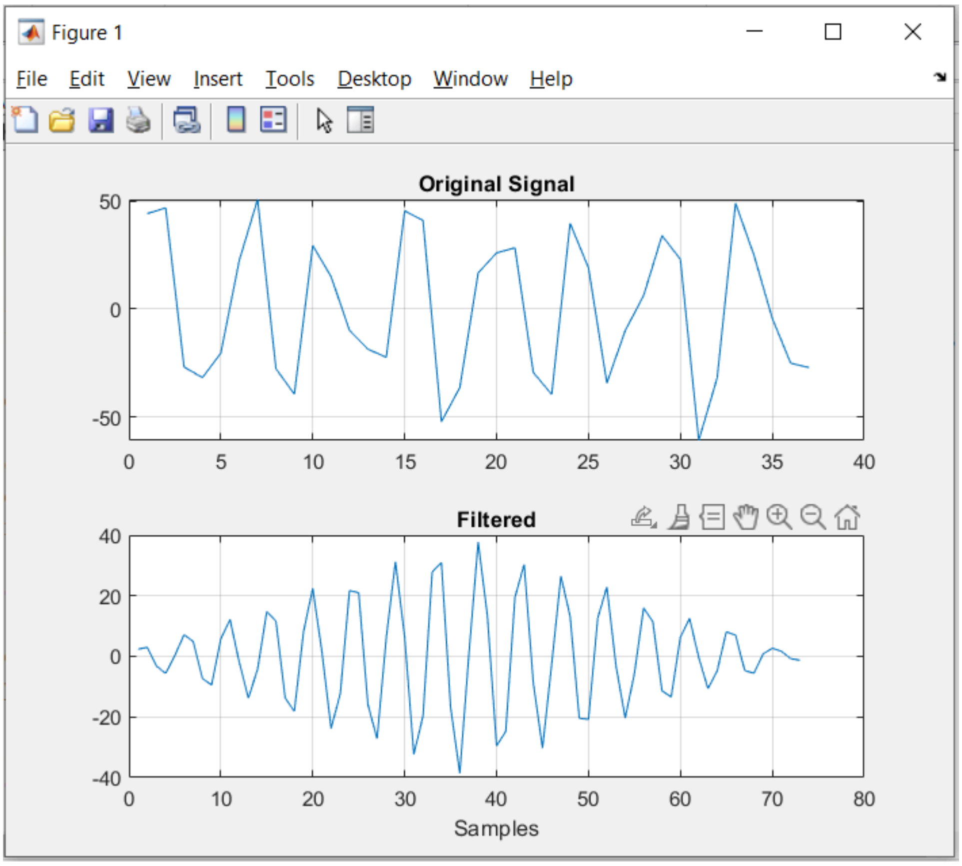
(d)

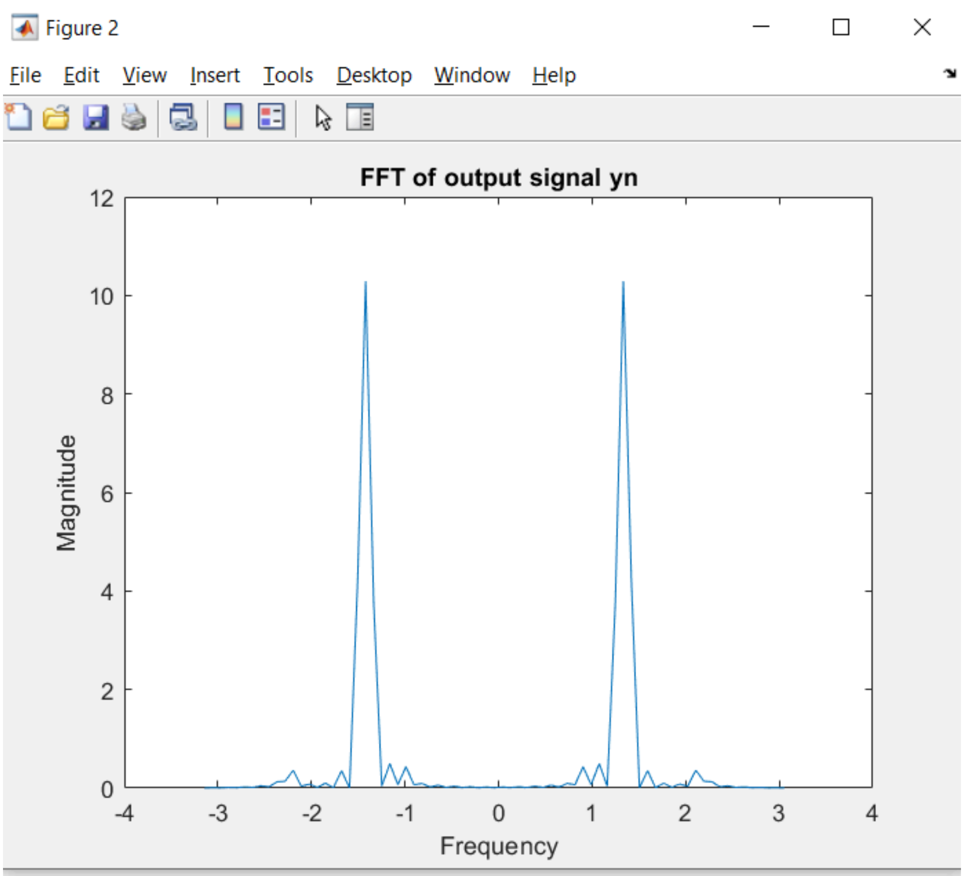
CODE:

```
w= 0.44*pi;
L=37;
n=0:L-1;
t = -pi:(pi/100):pi; %frequency vector
h = (2/L).*(cos(w.*n)); %frequency response
xn = 10*cos(0.3*pi*n) + 40*cos(0.44*pi*n - pi/3) + 20*cos(0.7*pi*n - pi/4);
%input signal
yn= conv(xn,h); %output
figure
subplot 211
plot(xn)
title('Original Signal')
grid on
subplot 212
plot(yn)
title('Filtered')
xlabel('Samples')
grid on

%FFT
fs = 2*pi; % sampling frequency
t = 0:(1/fs):(10-1/fs); % time vector
S= yn; %output acquired in step 1
n = length(S);
X = fft(S) ;
Y = fftshift(X);
fshift = (-n/2:n/2-1)*(fs/n); % zero-centered frequency range
figure
plot(fshift,abs(Y)/n)
title('FFT of output signal yn')
ylabel('Magnitude')
xlabel('Frequency')
```

SNAPSHOT:





OBSERVATIONS:

- The passed frequencies are centered at |0.44*pi| as we intended. We also have obtained the low(0.3pi) and high(0.7) limit frequencies reduced by a factor of 10 or more. The value of L is controlling the passband width, giving a distinction between allowed and discarded frequencies .