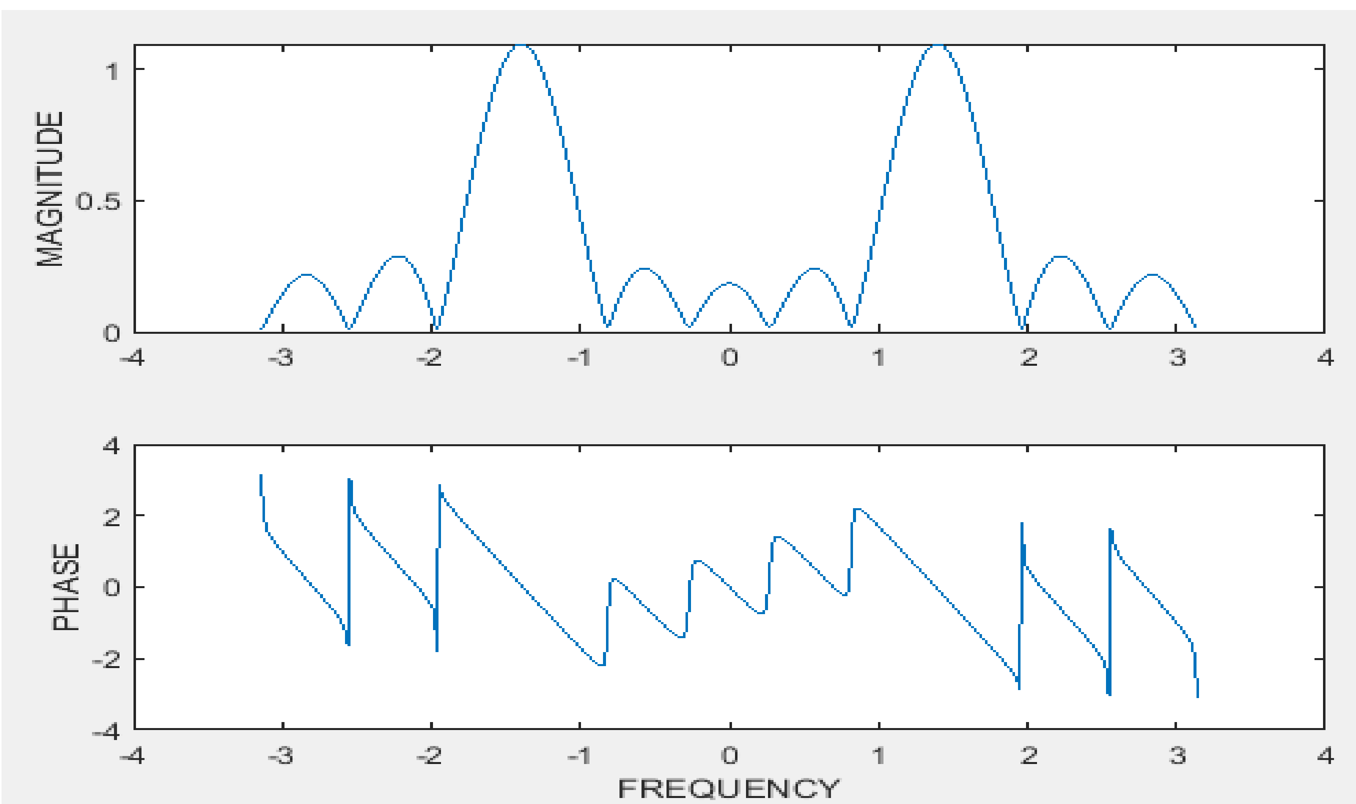
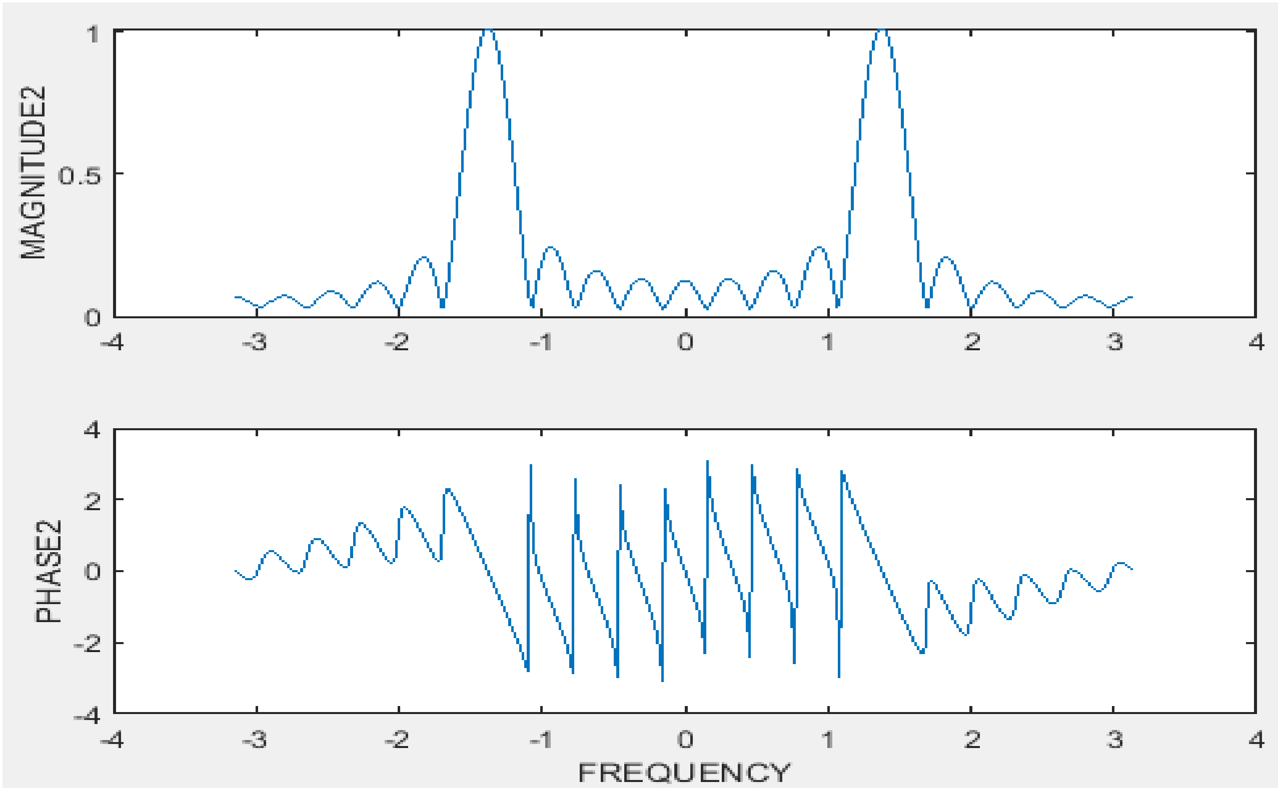
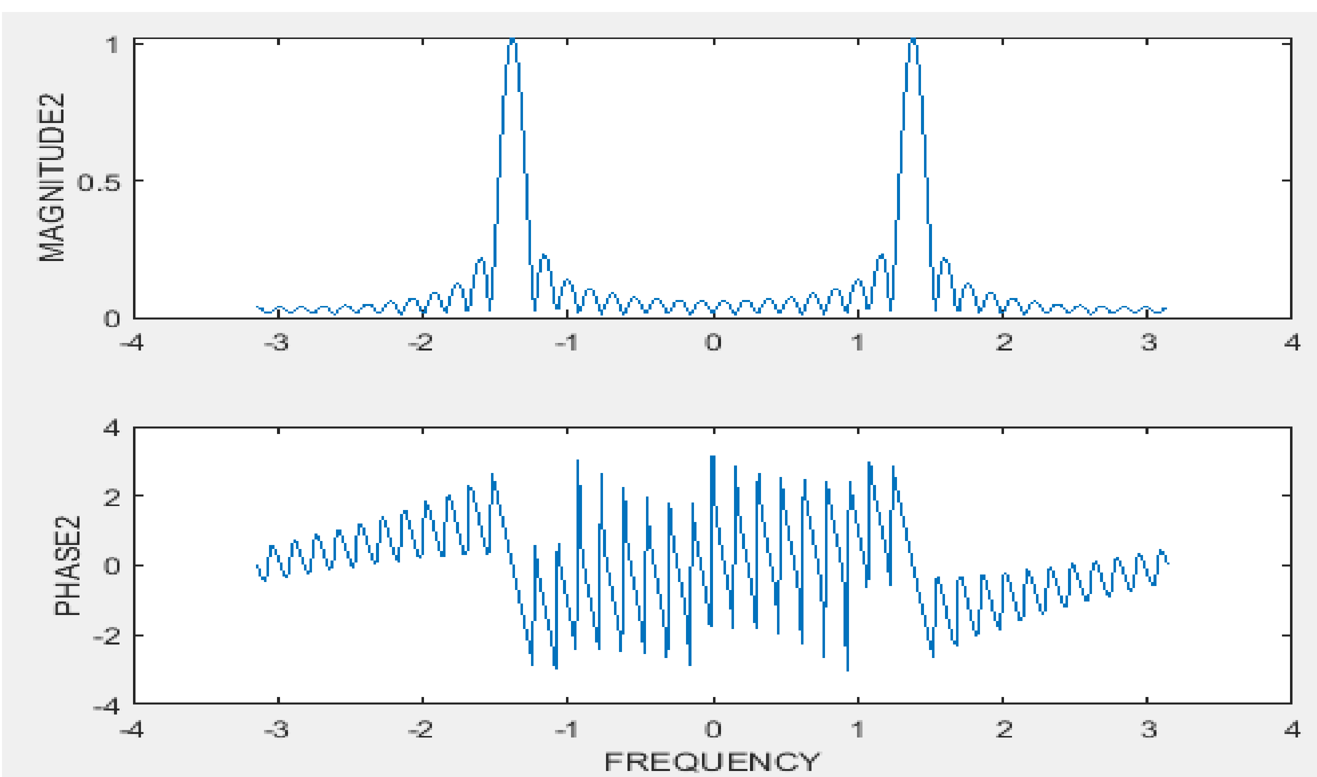
```
diary on
format compact
%Johnny Li
%EEL3135 Fall 2018
%Lab 6 Part 2

%Generate a bandpass filter that will pass a frequency component at W=0.44*
%pi, with L = 10.
%Length
L=10;
nn=0:L-1;
%Frequency
ww=0.44;
%Bandpass function
hh=(2/L)*cos(ww*pi*nn);
%Time interval
tt=-pi:pi/200:pi;

%Obtain frequency response
HH = freqz( hh, 1, tt );

%Plot
%Magnitude
subplot(2,1,1)
plot( tt, abs(HH) )
ylabel("MAGNITUDE")
%Phase
subplot(2,1,2)
plot( tt, angle(HH) )
ylabel("PHASE")
xlabel("FREQUENCY")
```



```
%The gain of the filter at w=0.3pi, w=0.44pi, and w=0.7pi.
w=0.3*pi;
q=abs(HH(find(abs(w-tt)<1E-4)))

q =

    0.2836

w=0.44*pi;
%Code given by TA.
q=abs(HH(find(abs(w-tt)<1E-4)))

q =

    1.0961

w=0.7*pi;
q=abs(HH(find(abs(w-tt)<1E-4))

q =

    0.2861


%2.1.2

%Determine the passband width.
%Code given by TA.
%Obtain pass band vector of L=10.
pbv= tt(find(abs(HH)>= sqrt(0.5)*(max(abs(HH)))));
%Subtract max and min to get range.
%pb=(pbvmax-pbvmin)/pi
pd=(1.6336-1.1624)/pi

pb =

   0.1500

%Obtain pass band vector of L=20.
pbv= tt(find(abs(HH)>= sqrt(0.5)*(max(abs(HH)))));
%Subtract max and min to get range.
pd=(1.5080-1.2566)/pi

pb =

    0.0800
```

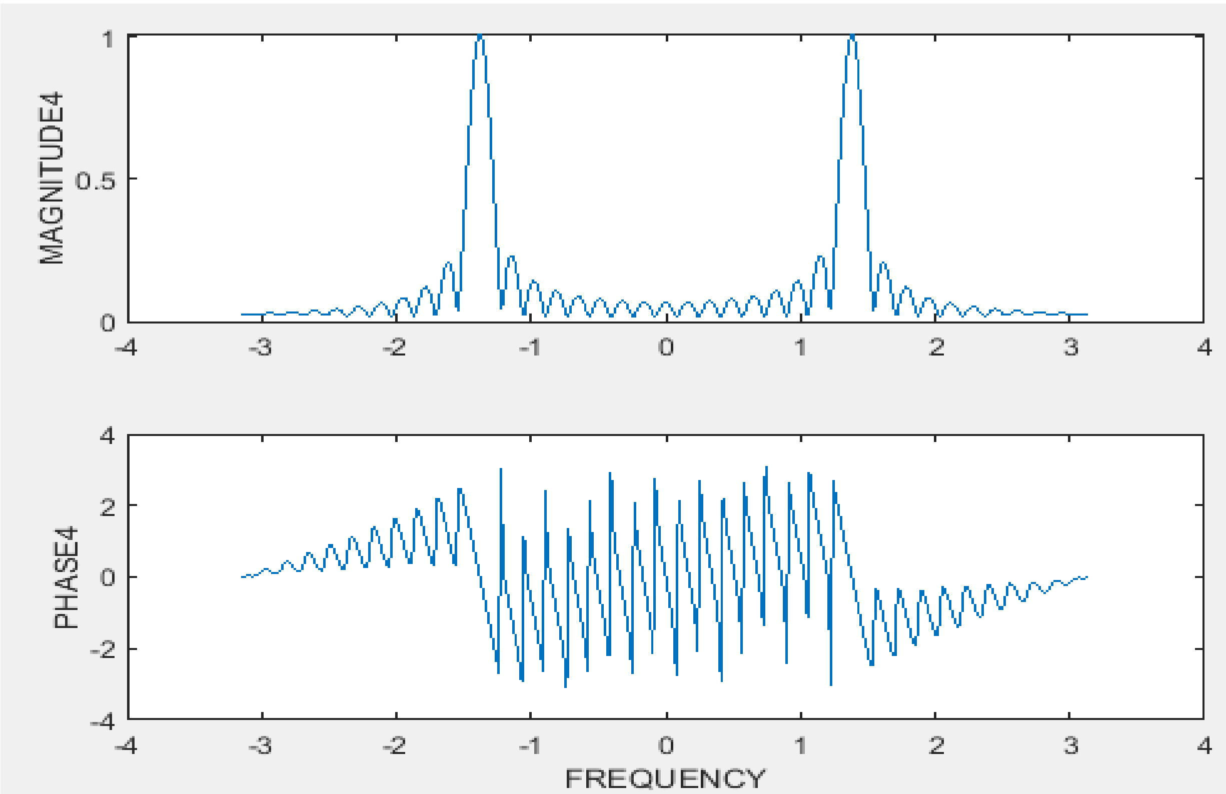



```
%Obtain pass band vector of L=40.
pbv= tt(find(abs(HH)>= sqrt(0.5)*(max(abs(HH)))));
%Subtract max and min to get range.
pd=(1.4451-1.3195)/pi

pb =

    0.0400
```



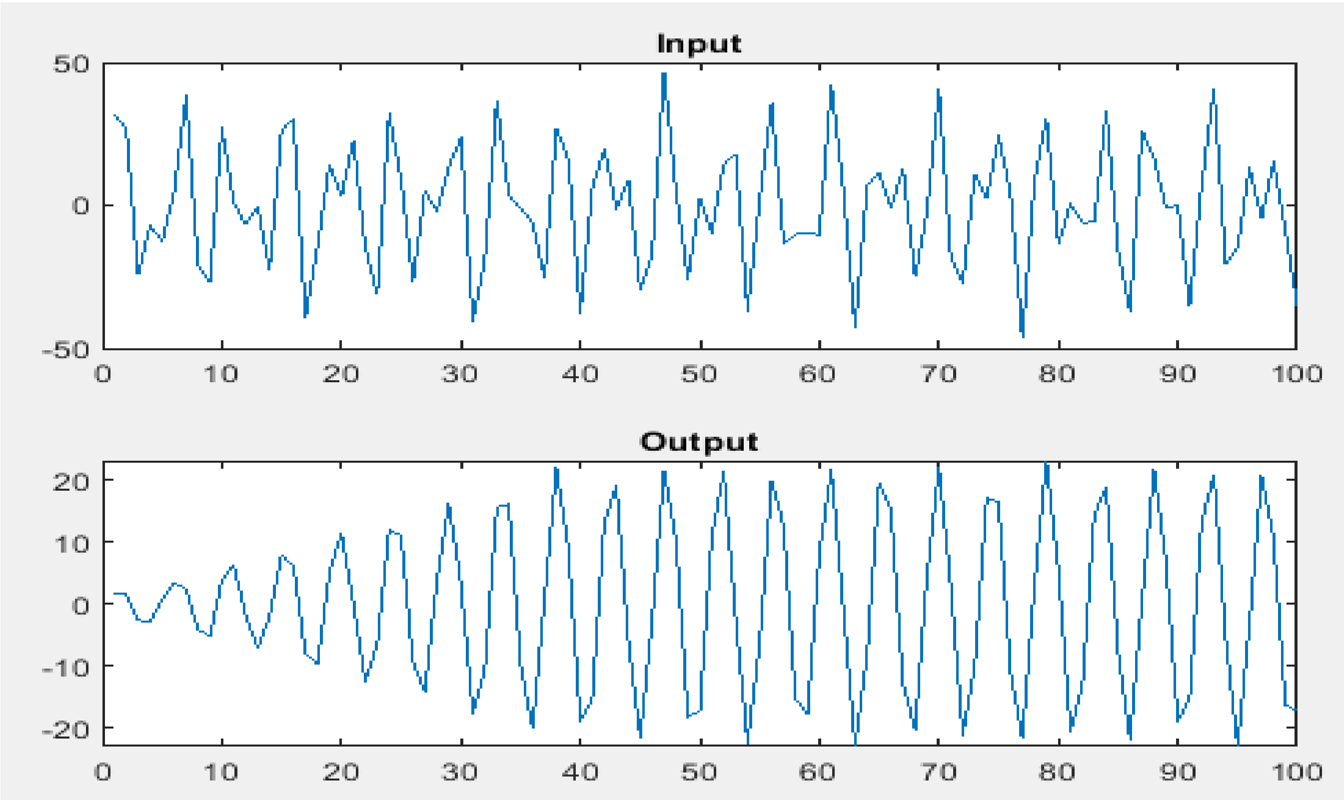
```
%Explain how the width of the passband is related to filter length L.
%The width of the passband is inversely proportional to the filter length
%of L, as doubling the filter length from 10 to 20 cuts the width of the
%passband in halve, from 0.15 to 0.08. Also, halving the filter length from
%40 to 20 doubles the width of the passband, from 0.04 to 0.08.

%2.1.3

%Comment on the selectivity of the L = 10 bandpass filter.
%Frequency response can pass one component at w=0.44pi while reducing or
%rejecting the others at w=0.3pi and w=0.7pi because a band pass aims to
%filter the mid-range value frequency and reduce the lower and higher bound
%ranges. In this band filter, since L = 10 where 2/10 =0.5, the accepted
%range is around 0.5+/-0.1, thus from 0.4pi to 0.6pi frequency.


%2.1.4

%Generate a bandpass filter that will pass the frequency component at
%w=0.44pi, but now make the filter length (L) long enough so that it will
%also greatly reduce frequency components at (or near) w=0.3pi and w=0.7pi.
%Length
L=38;
nn=0:L-1;
%Frequency
ww=0.44;
%Bandpass function
hh=(2/L)*cos(ww*pi*nn);
%Time interval
tt=-pi:pi/200:pi;

%Obtain frequency response
HH = freqz( hh, 1, tt );

%Plot
%Magnitude
subplot(2,1,1)
plot( tt, abs(HH) )
ylabel("MAGNITUDE2")
%Phase
subplot(2,1,2)
plot( tt, angle(HH) )
ylabel("PHASE2")
xlabel("FREQUENCY")

%Determine the smallest value of  so that the following conditions both
%hold: - Any frequency component satisfying abs(w)<=0.3pi will be reduced
%by a factor of 10 or more. - Any frequency component satisfying 0.7pi<=
abs(w)<=pi will be reduced by afactor of 10 or more.
%Find filter length L.
%Concatenate arrays horizontally
%Declare by TA, (range obtained from viewing plot of L=10)
st= horzcat(abs(HH(101:131)),abs(HH(178:201)));
pss= 0.1*max(abs(HH));
%Loop to increment L till the lower and uppper bounds are met.
while(max(st) > pss)
```

```
     max(st);
     %Upper bound
     st= horzcat(abs(HH(101:131)),abs(HH(178:201)));
     %Lower bound
     pss= 0.1*max(abs(HH));
     %Increment L
     L=L+1;
%L = 38.
End
```



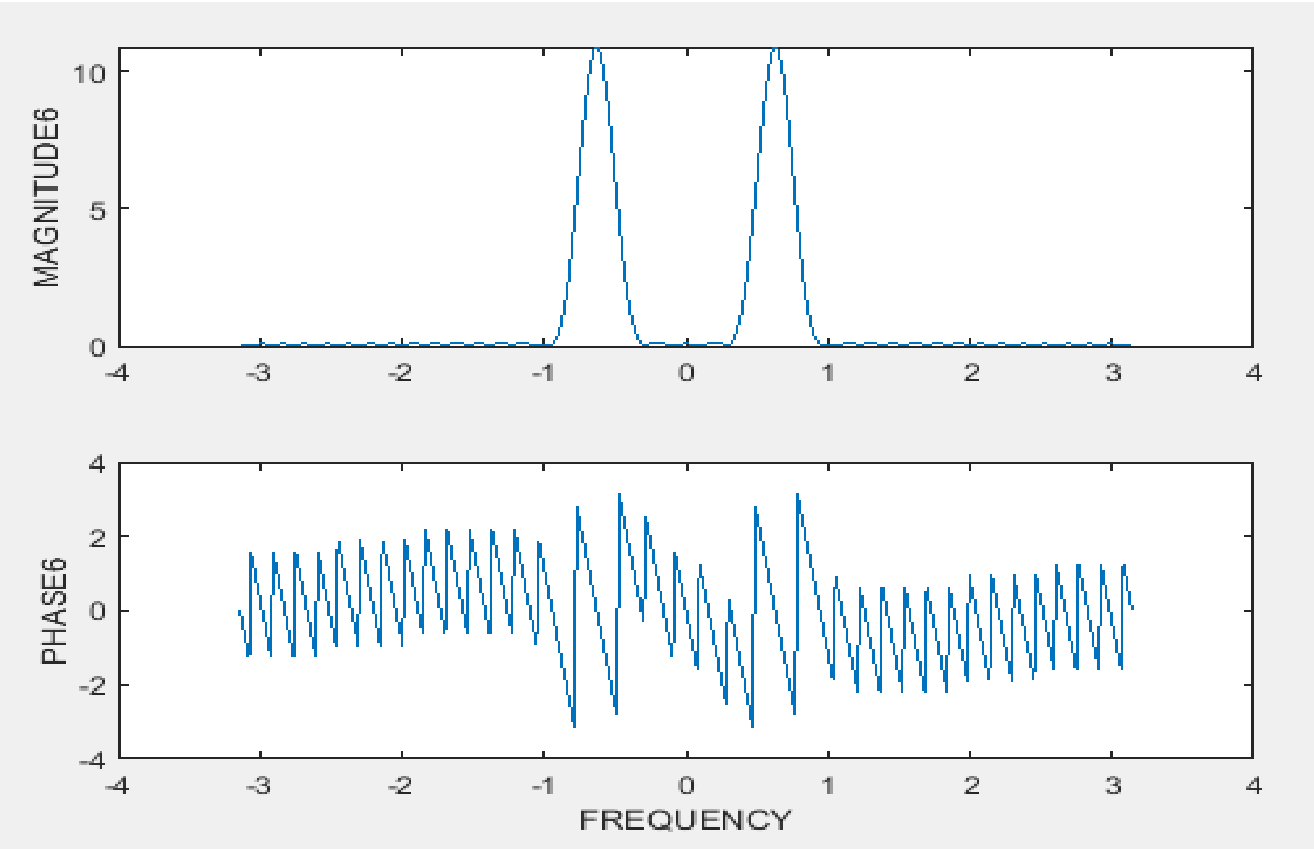

```
%2.1.5

%Make a plot of 100 points (over time) of the input and output signals of
%"sum of 3 sinusoids".
%Length
L=38;
nn=0:L-1;
%Frequency
ww=0.44;
%Bandpass function
hh=(2/L)*cos(ww*pi*nn);

%Time inveral of "sum of 3 sinusoids"
mm=0:149;
%Function of "sum of 3 sinusoids"
xx= 5*cos(0.3*pi*mm)+ 22*cos(0.44*pi*mm-(pi/3)) + 22*cos(0.7*pi*mm-(pi/4));
```

```
%Convolution
yy= firfilt(hh,xx);
```




```
%Explain how the filter has reduced or removed two of the three sinusoidal
%components.
%The filter has reduced or removed two of the three sinusoidal components
%as seen in the beginning of the plot by averaging the two signals.


%2.1.6


%Explain how H(e^(jw)) can be used to determine the relative size of each
%sinusoidal component in the output signal.
%By multiplying the input with the frequency response of the filter,
%the relative size of each sinusoidal component in the output signal can be
%determine as the output is just a proportion from the input due to
%property of multiplication. Since the frequency response is calculate, it is
known to reduce the boundary values.


%2.2.1


%Generate a Hamming bandpass filter that will pass a frequency component at
%w=0.2pi. Make the filter length L=41.
%Length
L=41;
```

```
nn=0:L-1;
%Frequency
ww=0.2;
%Better Bandpass function
hh=(0.54-0.46*cos((2*pi*nn)/(L-1))).*cos(ww*(nn-((L-1)/2)));
%Time interval
tt=-pi:pi/200:pi;

%Obtain frequency response
HH = freqz( hh, 1, tt );

%Plot
%Magnitude
subplot(2,1,1)
plot( tt, abs(HH) )
ylabel("MAGNITUDE6")
%Phase
subplot(2,1,2)
plot( tt, angle(HH) )
ylabel("PHASE6")
xlabel("FREQUENCY")
```

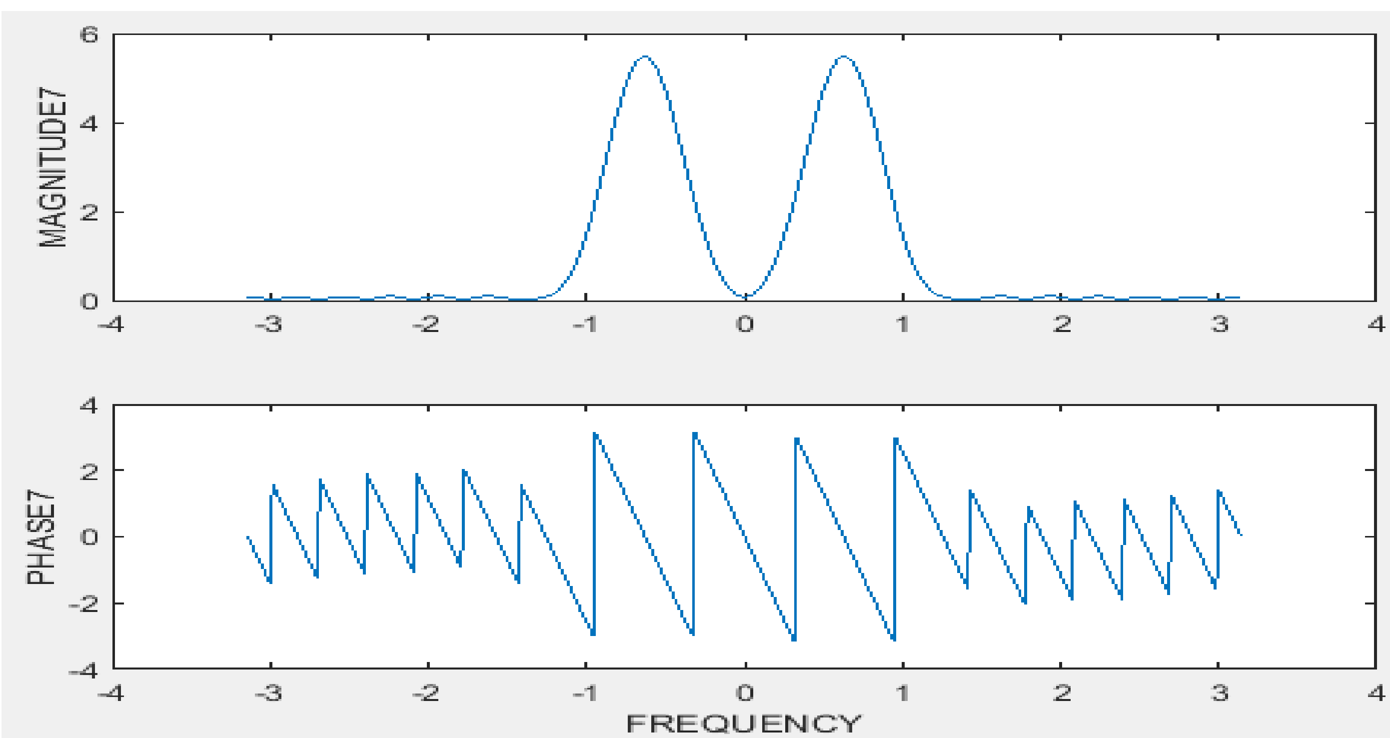



```
%Magnitude Values
M=[mag(1); mag(ceil(0.1*pi)); mag(ceil(0.25*pi)); mag(ceil(0.4*pi));
mag(ceil(0.5*pi)); mag(ceil(0.75*pi));];
%Phase Values
```

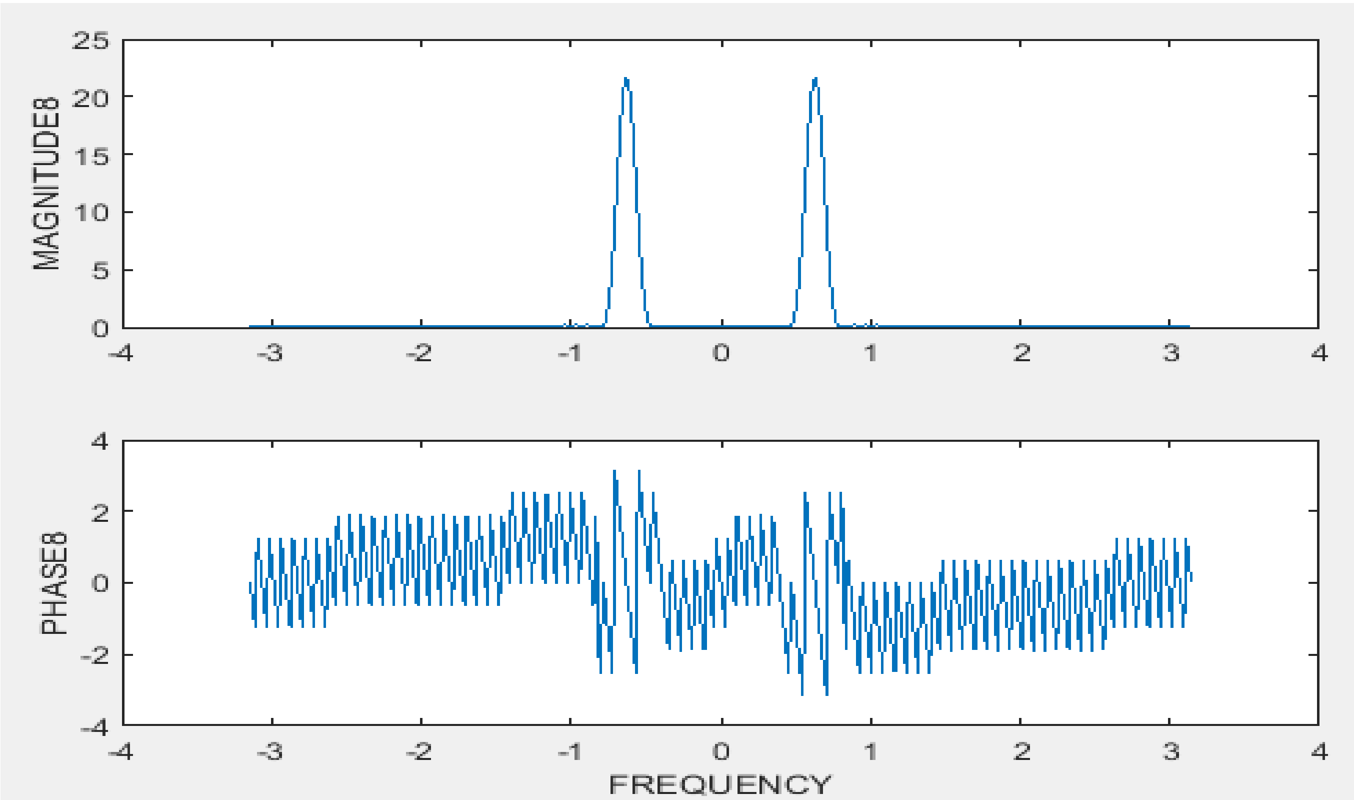
```
P=[pha(1); pha(ceil(0.1*pi)); pha(ceil(0.25*pi)); pha(ceil(0.4*pi));
pha(ceil(0.5*pi)); pha(ceil(0.75*pi));];
%Name Values
N={'w=0'; 'w=0.1*pi'; 'w=0.25*pi'; 'w=0.4*pi'; 'w=0.5*pi'; 'w=0.75*pi';};

%Make table
T=table(M,P,'RowNames', N);
%Display table
disp(T);
```

	M	P
w=0.1*pi	6.8953	9.6876e-15
w=0.25*pi	0.045171	-3.1416
w=0.4*pi	0.039021	3.1416
w=0.5*pi	0.041863	-3.1416
w=0.75*pi	0.045555	-3.1416

```
%2.2.2

%Determine the passband width using the 50% level to define the pass band.
%Obtain pass band vector of L=41.
pbv= tt(find(abs(HH)>= sqrt(0.5)*(max(abs(HH)))));
%Subtract max and min to get range.
pd=(0.7226-0.5341)/pi

pb =

    0.0600


%Plots of BPFs for L = 21.
```



```
%Subtract max and min to get range.
pd=(0.8168-0.4398)/pi

pb =

    0.1200

%Plots of BPFs for L = 81.
```




```
%Subtract max and min to get range.
pd=(0.6597-0.5969)/pi

pb =

    0.0300


%Explain how the width of the passband is related to filter length L.
%The width of the passband is inversely proportional to the filter length
%of L, as doubling the filter length from 21 to 41 cuts the width of the
%passband in halve, from 0.12 to 0.06. Also, halving the filter length from
%81 to 41 doubles the width of the passband, from 0.03 to 0.06.


%2.2.3

%Determine (by hand) the formula for the output signal.
```

%Comment on the relative amplitudes of the three signal components in the
%output signal, observing whether or not they were in the passband or
%topband of the filter.

Johnny Li
ECE 3135
P3

Lab 6 Part 2

2.2.3) $x[n] = 2 + 2\cos(0.1\pi n + \pi/3) + \cos(0.25\pi n - \pi/3)$

$$y[n] = 2|H(e^{j0})| + 2|H(e^{j0.1\pi})|\cos(0.1\pi n + \tfrac{\pi}{3} + \angle H(e^{j0.1\pi})) + |H(e^{j0.25\pi})|\cos(0.25\pi n - \pi/3 + \angle H(e^{j0.25\pi}))$$

$\hat{\omega}$	$\lvert H(e^{j\omega})\rvert$	$\angle H(e^{j\omega})$
0	0.08	0
0.1π	0.08	0
0.25π	4.52	π

$$y[n] = (2*0.08) + (2*0.08)\cos(0.1\pi n + \tfrac{\pi}{3} + 0) + 4.52\cos(0.25\pi n - \pi/3 + \pi)$$

$$= \underbrace{0.16 + 0.16\cos(0.1\pi n + \tfrac{\pi}{3})}_{\text{Pass band - extermely small amplitude} < 1.} + \underbrace{4.52\cos(0.25\pi n + \tfrac{2\pi}{3})}_{\text{Topband - large amplitude} > 1}$$

%2.2.4

%Use the frequency response (and passband width) of the length-41 bandpass
%filter to explain how the filter can pass the components at
%w=+/-0.25pi, while reducing or rejecting others.
%The frequency response of the length-41 contain the frequency of 0.25pi in
%the peaks while the passband contains 0.1pi,0.4pi,0.5pi and 0.75pi which
%gets reduced.

%2.3

%Calculate the same parameters in terms of normalized radiant frequency w,
%and submit a table consists of those values.
%Done by hand.

2.3)

Octave	2	3	4	5	6
Lower key	16	28	40	52	64
Lower Hz	65.4	130.8	261.6	523.25	1046.5
Higher key	27	39	51	63	75
Higher Hz	123.5	246.9	493.9	987.8	1978.5
Center	94.4	188.85	379.74	755.51	1551

Calculate by: $\hat{w} = \frac{f 2\pi}{f_s} = \frac{f 2\pi}{8000} = \frac{f\pi}{4000}$

Lower Hz	0.0164π	0.0327π	0.0654π	0.1308π	0.2616π
Higher Hz	0.0309π	0.0617π	0.1235π	0.2470π	0.4946π
Center	0.0236π	0.0472π	0.0948π	0.1889π	0.3878π

%2.4.1

%Devise a strategy for picking the constant B so that the maximum value of
%the frequency response will be equal to one. Write the one or two lines of
%code that will do this scaling operation in general.
B=1/max(abs(HH));

%2.4.2

%Design five separate bandpass filters for the filter bank system.
%Determine the length L that is required to get the correct bandwidth.
%Done by hand.

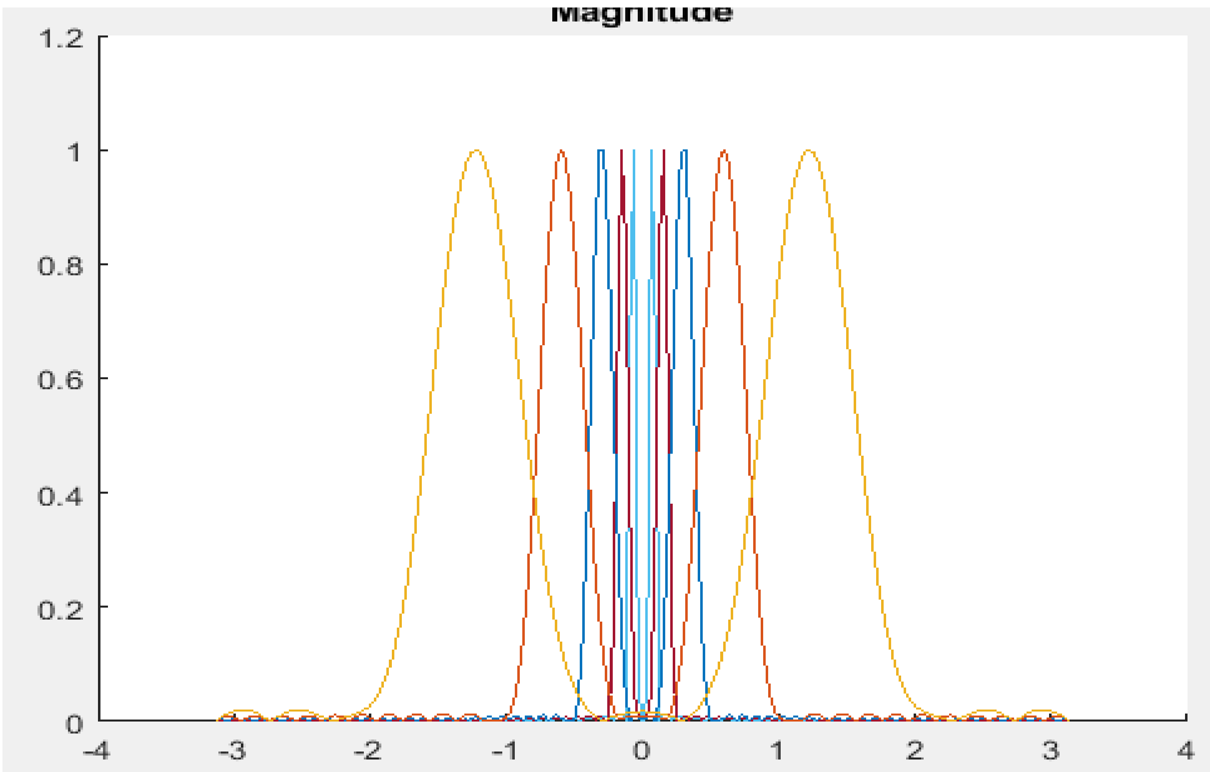
2.4.2) End − start = Band

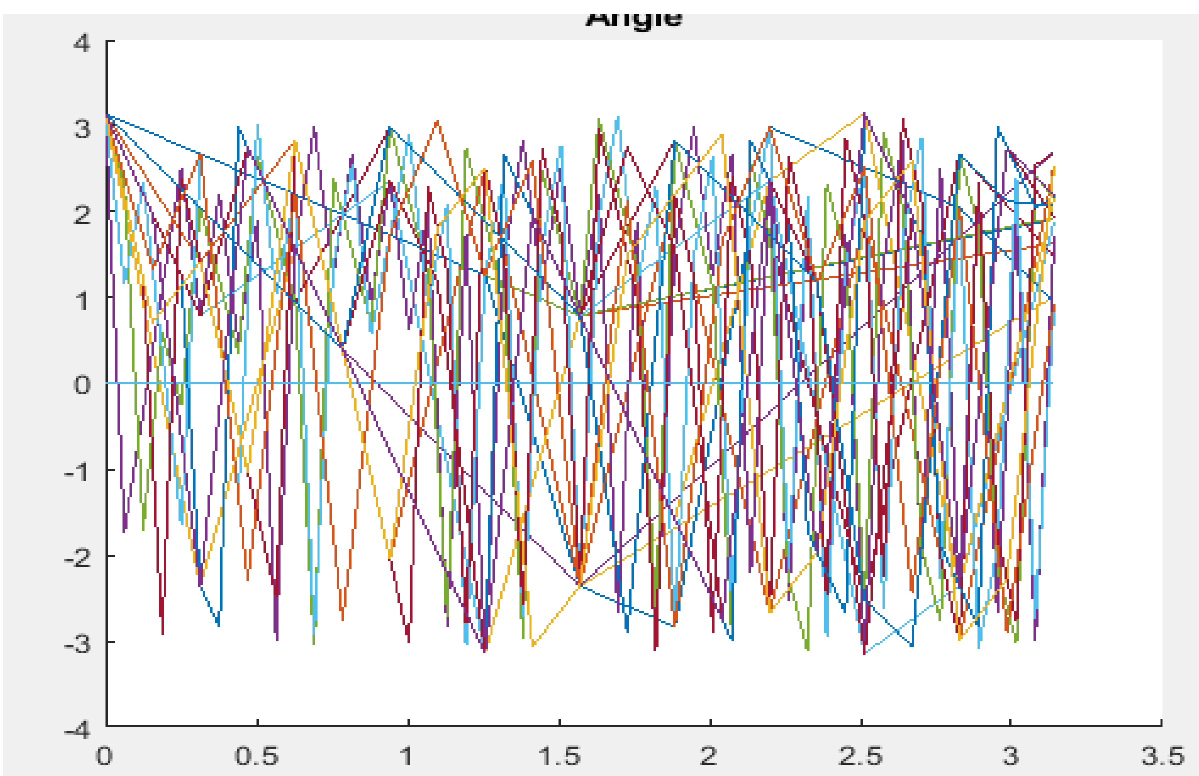
Octave	Start Radians	End Radians	Bandwidth Radian
2	0.051	0.096	0.045
3	0.102	0.193	0.091
4	0.205	0.387	0.182
5	0.410	0.775	0.364
6	0.821	1.551	0.729

Octave	Start Hz	End Hz	Bandwith Hz
2	65.4	123.5	58.1
3	130.8	246.9	116.1
4	261.6	493.9	232.3
5	523.3	987.8	464.6
6	1046.5	1978.5	929.0

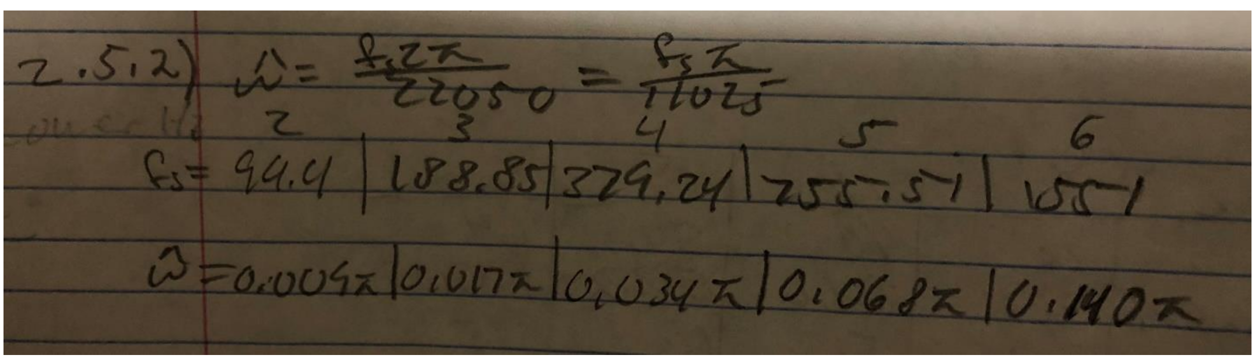
```
%This filter design process will be trial-and-error, length L.
Octave  Length
2       268
3       137
4       69
5       31
6       16

%2.4.3

%Plot the magnitude of the frequency responses all together on one plot.
type BetterBand

function HH1 = BetterBand(L,ww)
%BETTERBAND Generate a Hamming bandpass filter that will pass a frequency.
%Function for 2.4.3.

%Length range
nn=0:L-1;
%Better Bandpass function
hh=(0.54-0.46*cos((2*pi*nn)/(L-1))).*cos(ww*(nn-((L-1)/2)));
%Time interval
tt=0:pi/8000:pi;

%Obtain frequency response
HH = freqz( hh, 1, tt );

%Constant B
B=1/max(abs(HH));
%Multiple frequency response
hh1=B*hh;

%Obtain new frequency response
HH1 = freqz( hh1, 1, tt );

end

%Create signals
h1=BetterBand(268,0.069);
h2=BetterBand(137,0.137);
h3=BetterBand(69,0.274);
h4=BetterBand(31,0.549);
h5=BetterBand(16,1.097);

%Color frequency
hold on
plot(0.069,1,'red');
plot(0.137,1,'black');
plot(0.274,1,'blue');
plot(0.549,1,'green');
plot(1.097,1,'yellow');
hold off

%Range
tt=0:pi/8000:pi;
```

```
%Plot
hold on
plot(tt,abs(h1));
plot(tt,abs(h2));
plot(tt,abs(h3));
plot(tt,abs(h4));
plot(tt,abs(h5));
hold off
```





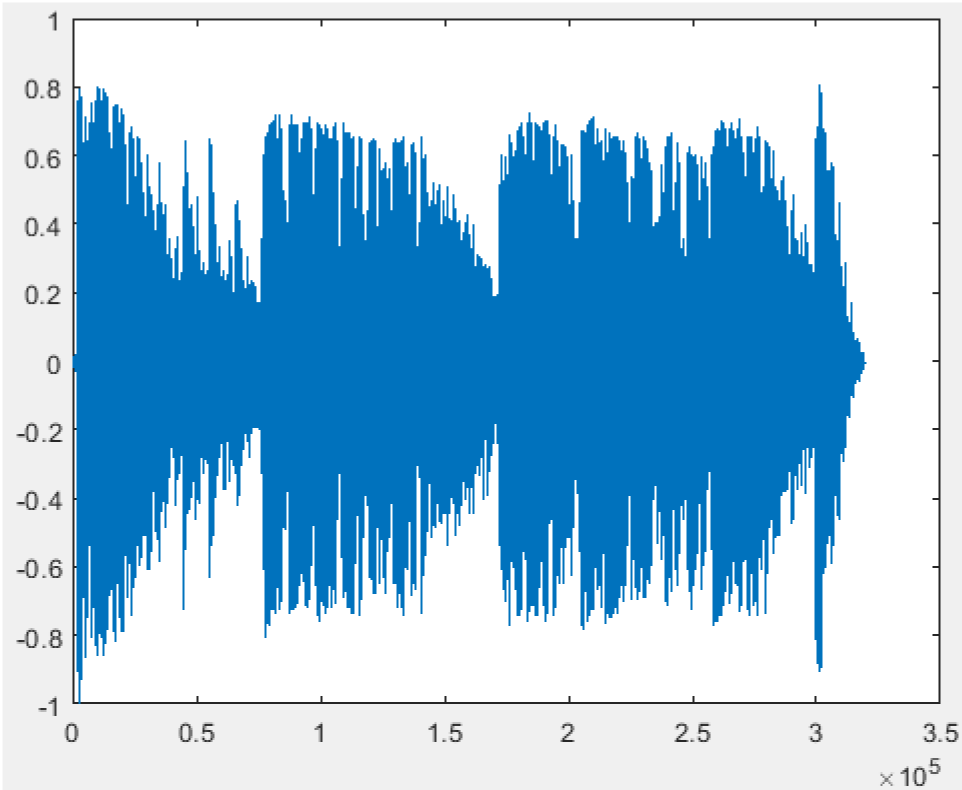
```
%2.4.4

%Comment on the selectivity of the bandpass filters.
%Frequency response can pass one component while reducing or rejecting the
%others because a band pass aims to filter the mid-range value frequency
%and reduce the lower and higher bound ranges.

%Are the filter's passbands narrow enough so that only one octave lies in
%the passband and the others are in the stopband?
%Yes, the filter's passbands is narrow enough so that only one octave lies
%in the passband and the others are in the stopband.

%2.5.1

%From x-file.wav, obtain the vector and sampling frequency of the input.
[y,Fs] = audioread('xfile.wav');
Fs

Fs =

       22050

%2.5.2

%Use fs obtained in previous section to re-calculate the normalized center
%frequencies.
%Done by hand.
```




```
%The lengths of the bandpass filters
L = [256, 128, 64, 32, 16];
%The center frequencies for the given octaves
freq = [.074, .148, .299, .593, 1.218];

%Length range
nn=0:L-1;
%Better Bandpass function
hh=(0.54-0.46*cos((2*pi*nn)/(L-1))).*cos(ww*(nn-((L-1)/2)));
%Time interval
tt=0:pi/8000:pi;

%Obtain frequency response
HH = freqz( hh, 1, tt );
```

```
%Constant B
B=1/max(abs(HH));
%Multiple frequency response
hh1=B*hh;

%Obtain new frequency response
HH1 = freqz( hh1, 1, tt );

%2.5.3

%Hence, modify your script so that all five filters have the same length as
%the longest filter.
type better

L=256;
%Length range
nn=0:L-1;
%Better Bandpass function
hh=(0.54-0.46*cos((2*pi*nn)/(L-1))).*cos(ww*(nn-((L-1)/2)));
%Time interval
tt=0:pi/8000:pi;

%Obtain frequency response
HH = freqz( hh, 1, tt );

%Constant B
B=1/max(abs(HH));
%Multiple frequency response
hh1=B*hh;

%Obtain new frequency response
HH1 = freqz( hh1, 1, tt );

%2.5.4

%Turn your script into a function that takes in the input vector xx, scale
%vector eq, sampling frequency fs and return output vector y.
function y = BetterBand(xx,eq,fs)
%BETTERBAND Generate a Hamming bandpass filter that will pass a frequency.
%Function for 2.5.4.

%The center frequencies for the given octaves
freq = [.07414, .14832, .29785, .59338, 1.2182];

%Storage Vector
ww=zeros(1,5);
HH=zeros(1,5);

%Create a vector of center frequency based on the inputted sampling
%frequency fs.
for i=1:5
ww=freq(i)*2*pi/fs;
end
```

```
for j=1:5
%Create five octave filters of the same length as the longest filter.
L=256;
%Length range
nn=0:L-1;
%Better Bandpass function
hh=(0.54-0.46*cos((2*pi*nn)/(L-1))).*cos(ww(j)*(nn-((L-1)/2)));
%Time interval
tt=0:pi/8000:pi;

%Obtain frequency response
HH = freqz( hh, 1, tt );

%Constant B
B=1/max(abs(HH));
%Multiple frequency response
hh1=B*hh;

%Obtain new frequency response
HH(j) = freqz( hh1, 1, tt );
end

%Scale each filter with a factor of 10^(eq(i)/20).
for t=1:5
HH(t)=10^(eq(i)/20)*HH(t);
end

%Sum up all five filter coefficients.
x=HH(1)+HH(2)+HH(3)+HH(4)+HH(5);

%Use convolution to obtain y.
yy= firfilt(x(1),x(2));
yy= firfilt(yy,x(3));
yy= firfilt(yy,x(4));
yy= firfilt(yy,x(5));

end

%2.5.5

%Plot magnitude response of your overall filter from -pi to pi.
[y,Fs] = audioread('xfile.wav');
eq = [20, 50,70, 0, 0];
test(y,eq,Fs);
```


1
0.8
0.6
0.4
0.2
0
-0.2
-0.4
-0.6
-0.8
-1
0
0.5
1
1.5
2
2.5
3
3.5
$\times 10^5$