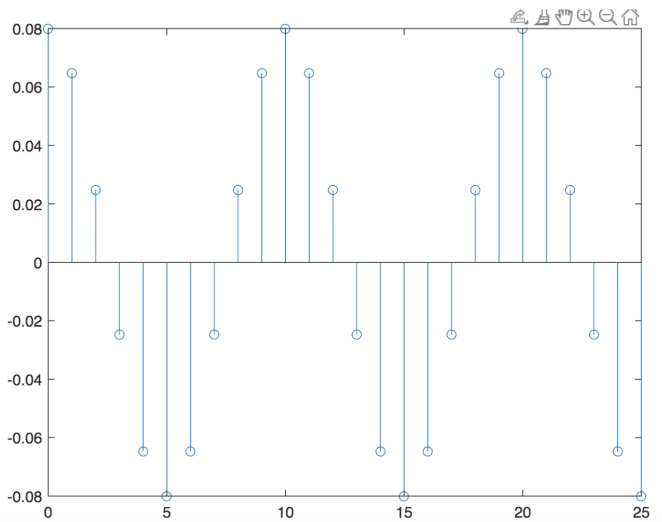
1.1) Implementing Octave Band Band filter

Question 1) Create a simple Bandpass Filter

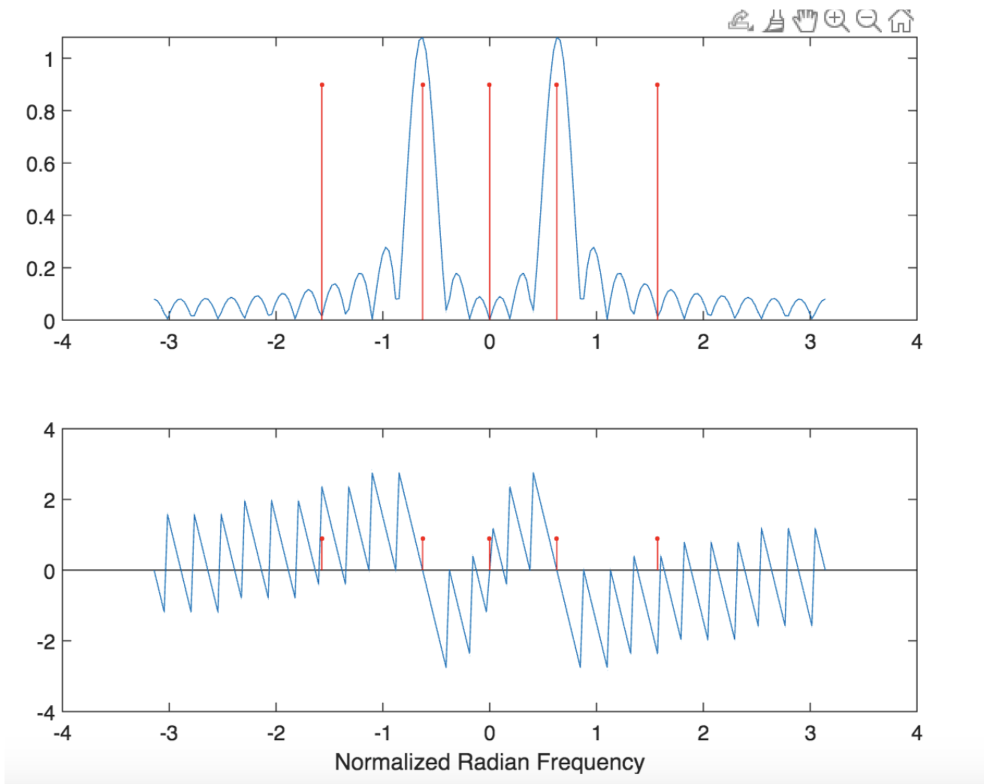
% 1 define the length of the filter
n = 0:1:25; hh1 = (2/25)*(cos(0.2*pi*n));
% Plot the impulse response
%stem(n,hh1);



The impulse response

Question 2:
% 2 compute the frequency response

ww = -pi:(pi/100):pi; %-- omega hat frequency axis
HH1 = freqz(hh1, 1, ww);
% Plot vertical markers at specific frequencies
subplot(2,1,1);
plot(ww, abs(HH1))
hold on;
stem(pi*[-0.5,-0.2,0,0.2,0.5],0.9*ones(1,5),'r.');
subplot(2,1,2);
plot(ww, angle(HH1))
hold on
stem(pi*[-0.5,-0.2,0,0.2,0.5],0.9*ones(1,5),'r.');
xlabel('Normalized Radian Frequency')



Question3:

```
% Define the bandpass filter
Hn = 1/(max(abs(HH1)))*abs(HH1);
indc = find(Hn>0.5);
w = ww(indc);w = w(w>0);
% calculate the magnitude and the phase of the
pass_width = max(w)-min(w)
```

Passband Width = 0.2513

Question 4:

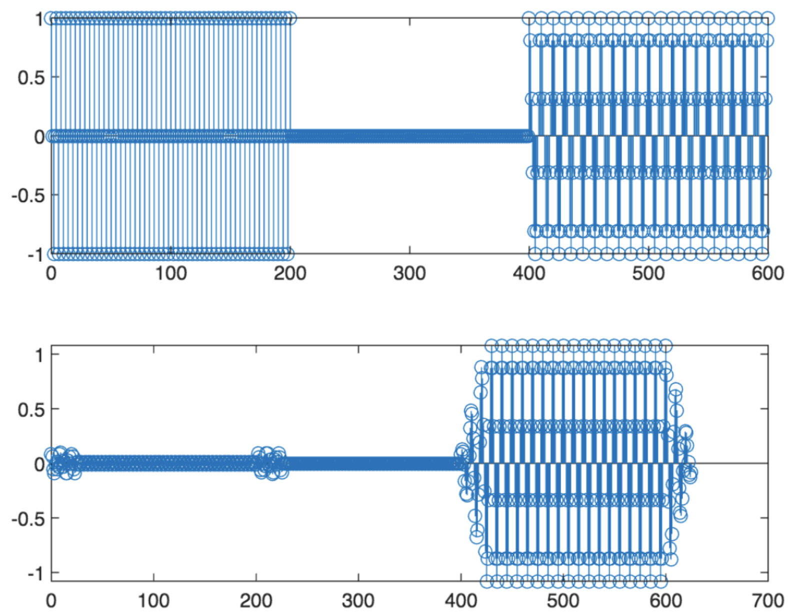
Increasing the length L decreases the bandwidth of the bandpass filter in the frequency domain.

Filters	L	Passband Width
hh2	20	0.3142
hh3	80	0.0628

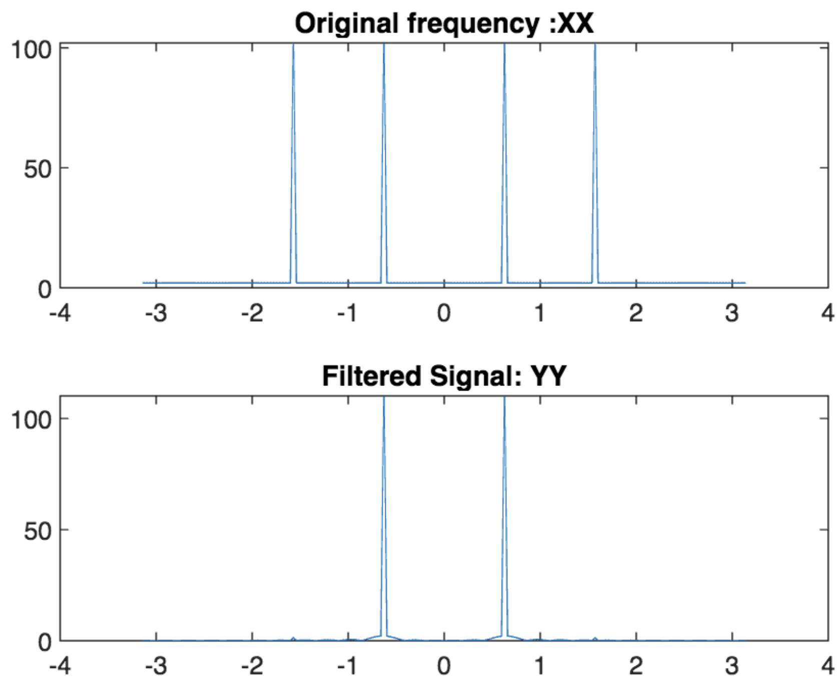
```
ww2 = -pi:(pi/100):pi; %-- omega hat frequency axis
ww3 = -pi:(pi/100):pi; %-- omega hat frequency axis
n2 = 0:1:20;
n3 = 0:1:80;
hh2 = (2/20)*(cos((pi*0.4)*n2));
hh3 = (2/80)*(cos((pi*0.4)*n3));
HH2 = freqz(hh2, 1, ww2);
HH3 = freqz(hh3, 1, ww3);
Hn2 = 1/(max(abs(HH2)))*abs(HH2);
indc = find(Hn2>0.5);
w2 = ww2(indc);w2 = w2(w2>0);
Hn3 = 1/(max(abs(HH3)))*abs(HH3);
indc = find(Hn3>0.5);
w3 = ww3(indc);w3 = w3(w3>0);
pass_width2 = max(w2)-min(w2)
pass_width3 = max(w3)-min(w3)
```

Question 5:

```
% Create the input signal
xx = zeros(1,601);
x1 = cos(0.5*pi*(0:1:200)); x2 = cos(0.2*pi*(400:1:600));
xx(1:1:201) = x1; xx(401:1:601) = x2;

% Convolve the two functions
subplot(2,1,1); stem(0:1:600,xx);
subplot(2,1,2); stem(0:1:625,yy);
```



Filtered Signal in the Frequency Domain:



The passband filter successfully passed the intended frequency in the mid range.

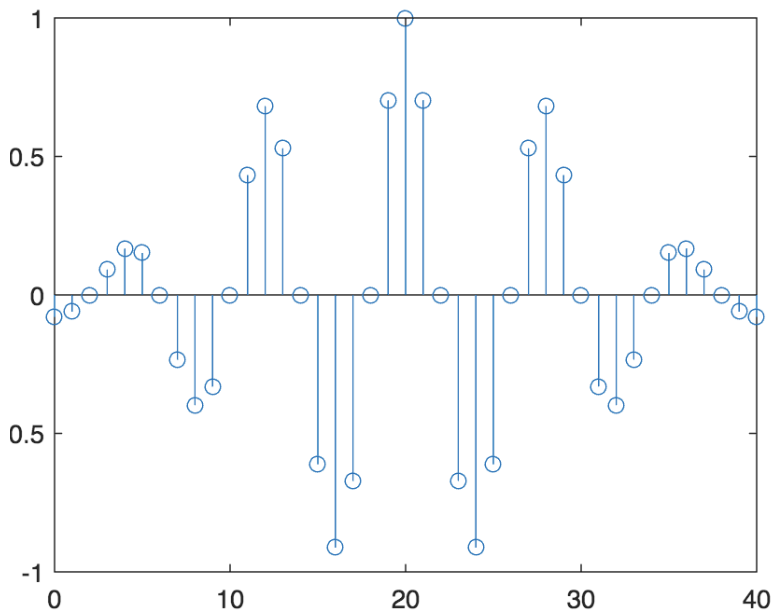
```
% Calculate the frequency response of the filtered signal
XX = freqz(xx, 1, ww);
YY = freqz(yy, 1, ww);
subplot(2,1,1); plot(ww, abs(XX)); title('Original frequency :XX')
subplot(2,1,2); plot(ww, abs(YY)); title('Filtered Signal: YY')
```

Question 8:

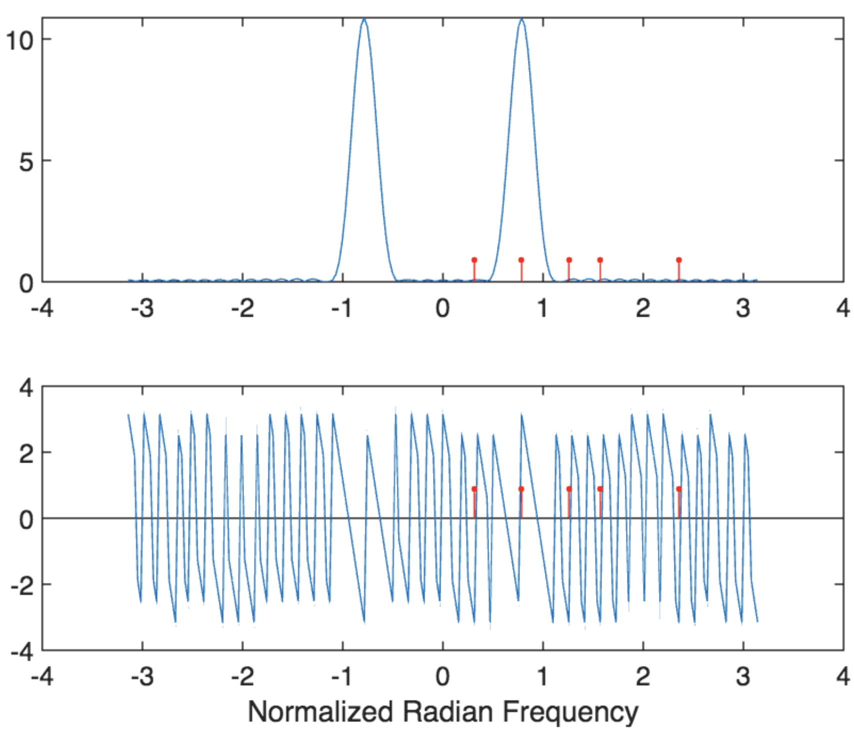
The transition period lasts 25 samples. It comes from the time it takes the filter to completely overlap with the input signal.

2.2) Improving Bandpass Filter Design

Question 1)



Frequency Response:



Frequency	Magnitude	Phase
0.1π	0.0084	3.1416
0.25π	10.88	3.1416
0.4π	0.0084	3.1416
0.5π	0.0084	-3.1416
0.75π	0.0084	-3.1416
0	0.0084	-3.1416

```
ww = -pi:(pi/100):pi; %-- omega hat frequency axis
H1 = freqz(h, 1, ww);
subplot(2,1,1);
plot(ww,abs(H1))
hold on; stem(pi*[0.1,0.25,0.4,0.5,0.75],0.9*ones(1,5),'r.');
subplot(2,1,2);
plot(ww, angle(H1))
hold on
hold on; stem(pi*[0.1,0.25,0.4,0.5,0.75],0.9*ones(1,5),'r.');
xlabel('Normalized Radian Frequency')
hold off;
```

Question 2)

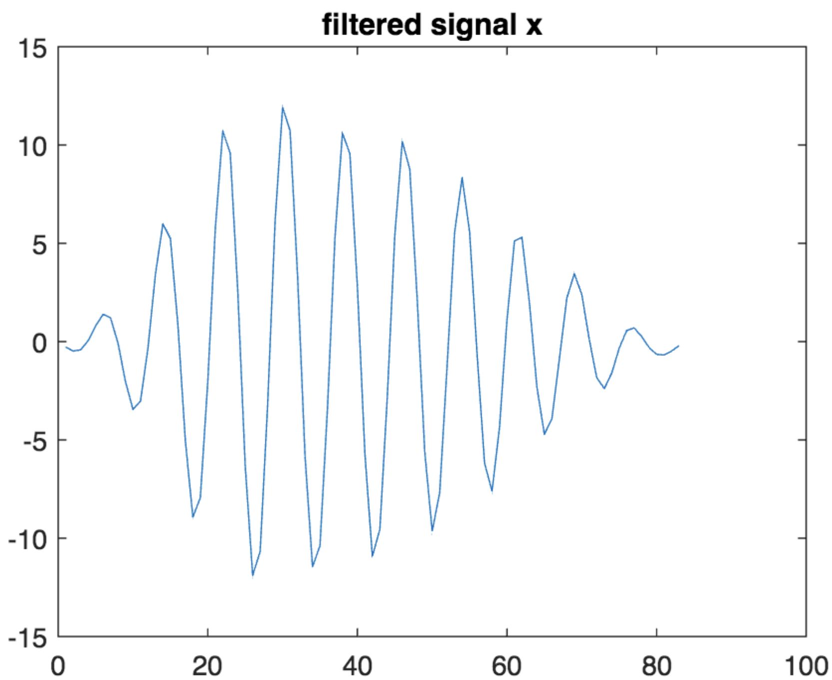
Increasing L decreases the passband frequency range.

Filters	L	Passband Width
hh1	41	0.2513
hh2	21	0.5027
hh3	81	0.1257

```
ww1 = -pi:(pi/100):pi; %-- omega hat frequency axis
ww2 = -pi:(pi/100):pi; %-- omega hat frequency axis
ww3 = -pi:(pi/100):pi; %-- omega hat frequency axis
n1 = 0:1:41;
L1 = max(n1);
n2 = 0:1:21;
L2 = max(n2);
n3 = 0:1:81;
L3 = max(n3);
a0 = 0.54;
hh1 = (a0-(1-a0)*cos(2*pi*n1*1/(L1-1))).*cos(0.25*pi*(n1-(0.5*(L1-1))));
hh2 = (a0-(1-a0)*cos(2*pi*n2*1/(L2-1))).*cos(0.25*pi*(n2-(0.5*(L2-1))));
hh3 = (a0-(1-a0)*cos(2*pi*n3*1/(L3-1))).*cos(0.25*pi*(n3-(0.5*(L3-1))));
HH1 = freqz(hh1, 1, ww1);
HH2 = freqz(hh2, 1, ww2);
HH3 = freqz(hh3, 1, ww3);
Hn1 = 1/(max(abs(HH1)))*abs(HH1);
indc = find(Hn1>0.5);
w1 = ww1(indc);w1 = w1(w1>0);
Hn2 = 1/(max(abs(HH2)))*abs(HH2);
indc = find(Hn2>0.5);
```

```
w2 = ww2(indc);w2 = w2(w2>0);
Hn3 = 1/(max(abs(HH3)))*abs(HH3);
indc = find(Hn3>0.5);
w3 = ww3(indc);w3 = w3(w3>0);
pass_width1 = max(w1)-min(w1)
pass_width2 = max(w2)-min(w2)
pass_width3 = max(w3)-min(w3)
```

Question 3:



From the frequency response, the gain for the frequencies, 0.1π and 0.25π are respectively 0.0084 and 10.88. Similarly, the phases shifts are -π .

Y = 2*0.008 - 2*0.008*cos($0.1\pi n + \frac{\pi}{3}$) - 2*10.8*cos($0.25\pi n - \frac{\pi}{3}$)

Y = 0.016- 0.08*cos($0.1\pi n + \frac{\pi}{3}$) -10.8cos($0.25\pi n - \frac{\pi}{3}$)

Question 4:

In the filtered signal above, we only see 0.25π since the two other frequencies are removed by the passband filter.

2.2) Piano Octaves

Octaves	Key Range	Frequency Range(Hz)	Midpoint Frequency	Frequency Range in (w^) ($\frac{2\pi F}{Fs}$)
#2	$C_2 - B_2$	65.41 - 123.47 Hz	94.44 Hz	0.074173
#3	$C_3 - B_3$	130.81-246.94	188.875	0.14834
#4	$C_4 - B_4$	261.63 - 493.88	377.755	0.296689
#5	$C_5 - B_5$	523.25 - 987.77	755.51	0.593376
#6	$C_6 - B_6$	1046.50-1975.5	1511	1.18673

Bandpass Filter Bank Design:

Question 1)

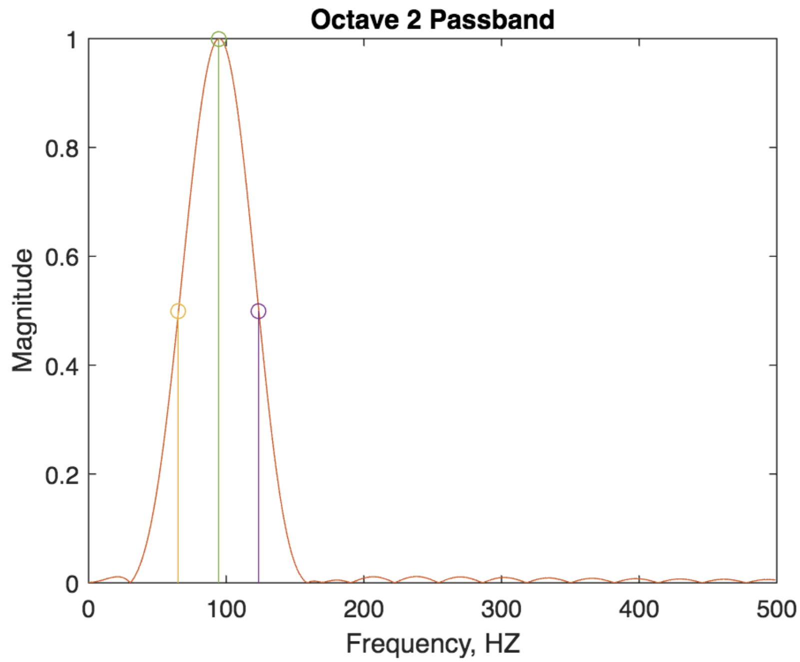
1) We find the maximum value of frequency response and use that to normalize the frequency response.

```
% Define the first filter for octave # 2 with a center frequency of 0.03849
n2 = 0:1:250;
L2  = max(n2);
h2 = (a0-(1-a0)*cos((2*pi*n2)*(1/(L2-1)))).*cos(0.074173*(n2-(L2-1)*0.5));
H2 = freqz(h2, 1, ww2,Fs);Hn2 = 1/(max(abs(H2)))*abs(H2);
indc = find(Hn2>=0.5);
w2 = ww2(indc);H2 = Hn2(indc);
```

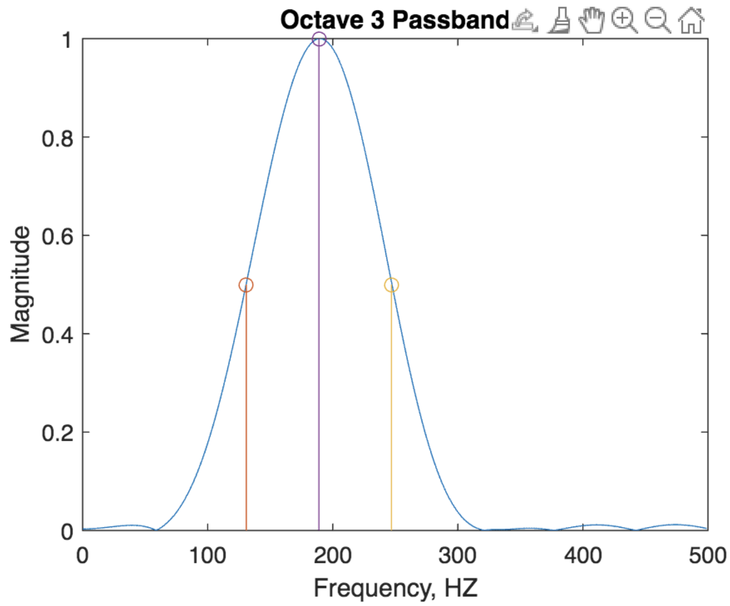
Question 2:

Filter	L
Octave2	250
Octave3	125
Octave4	62
Octave5	31
Octave6	16

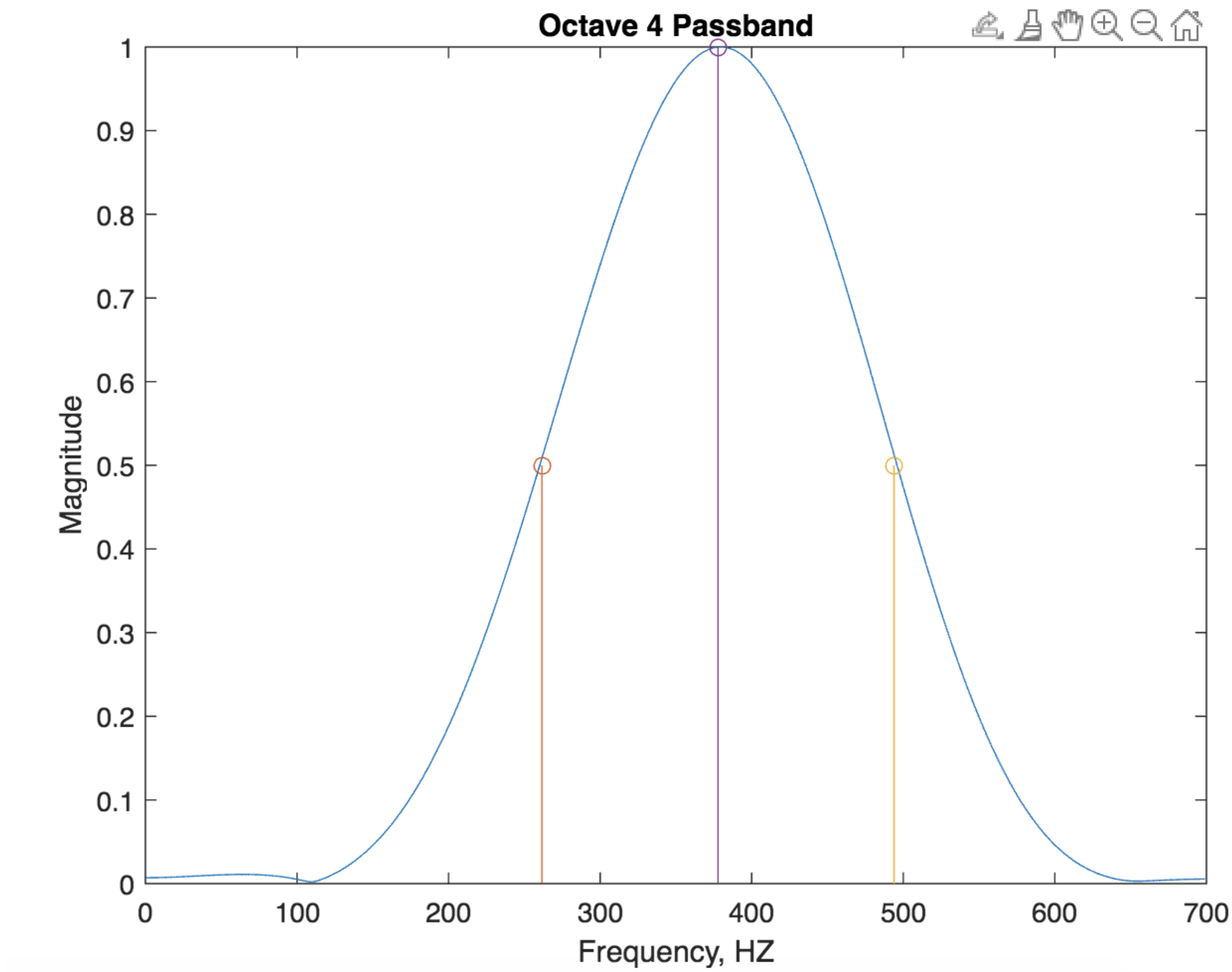
Question 3:




```
% Define the first filter for octave # 2 with a center frequency of 0.03849
% Bandpass Filter Bank Design
ww2 = 0:1:4000; %-- frequency axis in Hz
Fs = 8000;
a0 = 0.54;
n2 = 0:1:250;
L2  = max(n2);
h2 = (a0-(1-a0)*cos((2*pi*n2)*(1/(L2-1)))).*cos(0.074173*(n2-(L2-1)*0.5));
H2 = freqz(h2, 1, ww2,Fs);Hn2 = 1/(max(abs(H2)))*abs(H2);
%indc = find(Hn2>=0.5);
%w2 = ww2(indc);H2 = Hn2(indc);
plot(ww2(1:500),Hn2(1:500)); title('Octave 2 Passband')
xlabel('Frequency, HZ'); ylabel('Magnitude');
hold on
stem(65, 0.5);
stem(123.47, 0.5);
stem(94.44, 1);
hold off;
```



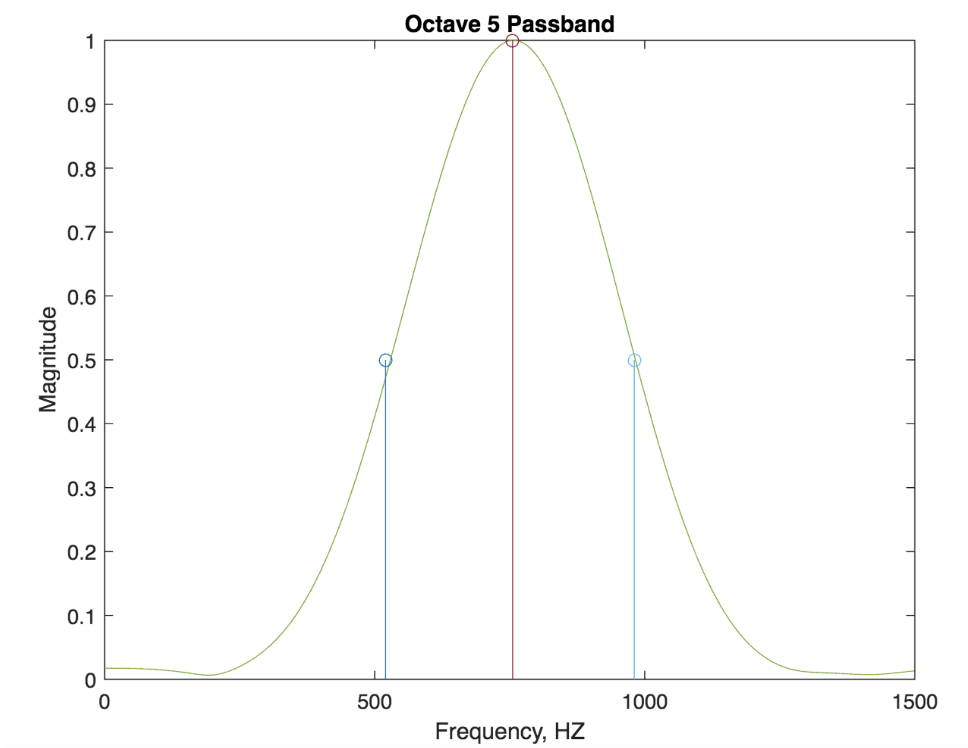

```
ww3 = 0:1:4000; %-- frequency axis in Hz
Fs = 8000;
a0 = 0.54;
% Define the second filter for octave 3
n3 = 0:1:250/2;
L3  = max(n3);
h3 = (a0-(1-a0)*cos((2*pi*n3)*(1/(L3-1)))).*cos(0.14834*(n3-(L3-1)*0.5));
H3 = freqz(h3, 1, ww3,Fs);Hn3 = 1/(max(abs(H3)))*abs(H3);
plot(ww3(1:500),Hn3(1:500)); title('Octave 3 Passband')
xlabel('Frequency, HZ'); ylabel('Magnitude');
hold on
stem(130.81, 0.5);
stem(246.94, 0.5);
stem(188.875, 1);
```




```
ww3 = 0:1:4000; %-- frequency axis in Hz
Fs = 8000;
a0 = 0.54;
% Define the second filter for octave 4
n4 = 0:1:62;
L4  = max(n4);
h4 = (a0-(1-a0)*cos((2*pi*n4)*(1/(L4-1)))).*cos(0.296689*(n4-(L4-1)*0.5));
H4 = freqz(h4, 1, ww3,Fs);Hn4 = 1/(max(abs(H4)))*abs(H4);
plot(ww3(1:700),Hn4(1:700)); title('Octave 4 Passband')
xlabel('Frequency, HZ'); ylabel('Magnitude');
hold on
```

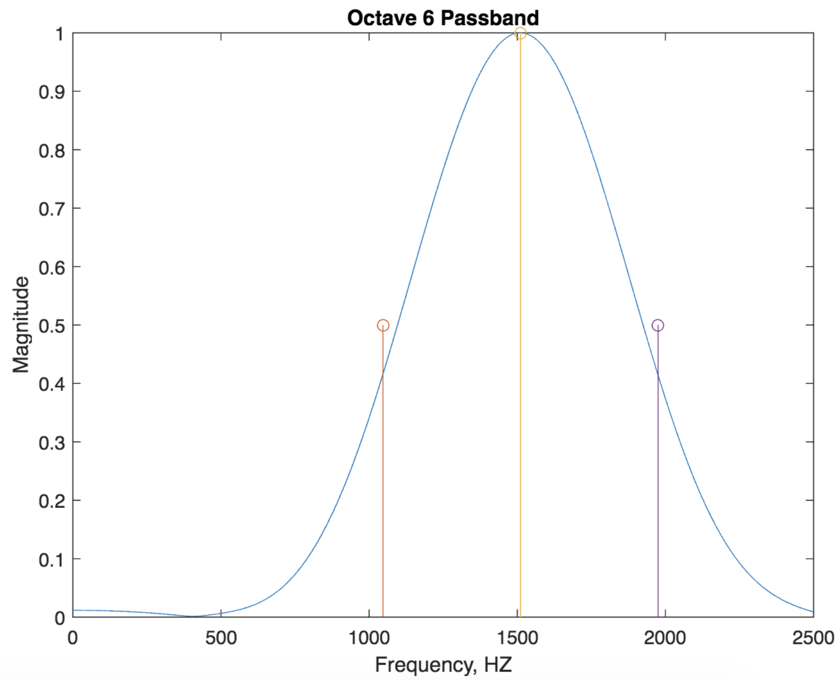
```
stem(261.63 , 0.5);
stem(493.88, 0.5);
stem(377.755, 1);
```

Octave 5




```
ww3 = 0:1:4000; %-- frequency axis in Hz
Fs = 8000;
a0 = 0.54;
n5 = 0:1:32;
L5  = max(n5);
h5 = (a0-(1-a0)*cos((2*pi*n5)*(1/(L5-1)))).*cos(0.593376*(n5-(L5-1)*0.5));
H5 = freqz(h5, 1, ww3,Fs);Hn5 = 1/(max(abs(H5)))*abs(H5);
plot(ww3(1:1500),Hn5(1:1500)); title('Octave 5 Passband')
xlabel('Frequency, HZ'); ylabel('Magnitude');
hold on
```

```
stem(980.77, 0.5);
stem(755.51, 1);
stem(520.25 , 0.5);
```



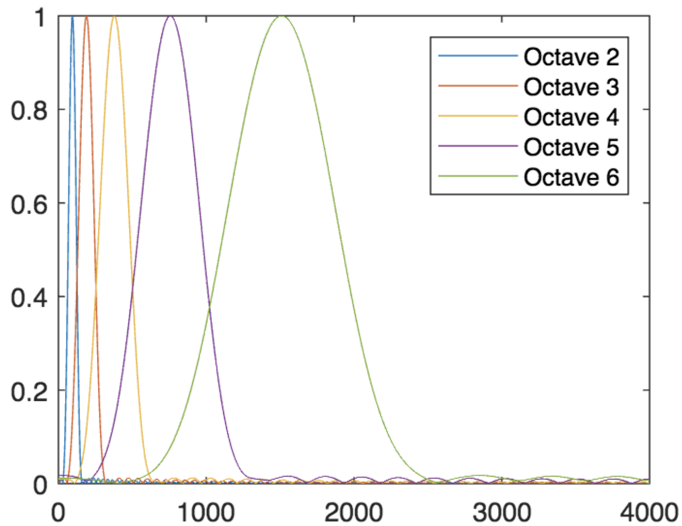

```
ww3 = 0:1:4000; %-- frequency axis in Hz

Fs = 8000;

a0 = 0.54;

% Define the second filter for octave 4

n4 = 0:1:62;

L4  = max(n4);

h4 = (a0-(1-a0)*cos((2*pi*n4)*(1/(L4-1)))).*cos(0.296689*(n4-(L4-1)*0.5));

H4 = freqz(h4, 1, ww3,Fs);Hn4 = 1/(max(abs(H4)))*abs(H4);

plot(ww3(1:700),Hn4(1:700)); title('Octave 4 Passband')

xlabel('Frequency, HZ'); ylabel('Magnitude');

hold on
```

stem(261.63 , 0.5);

stem(493.88, 0.5);

stem(377.755, 1);

All the normalized filters in one plot,

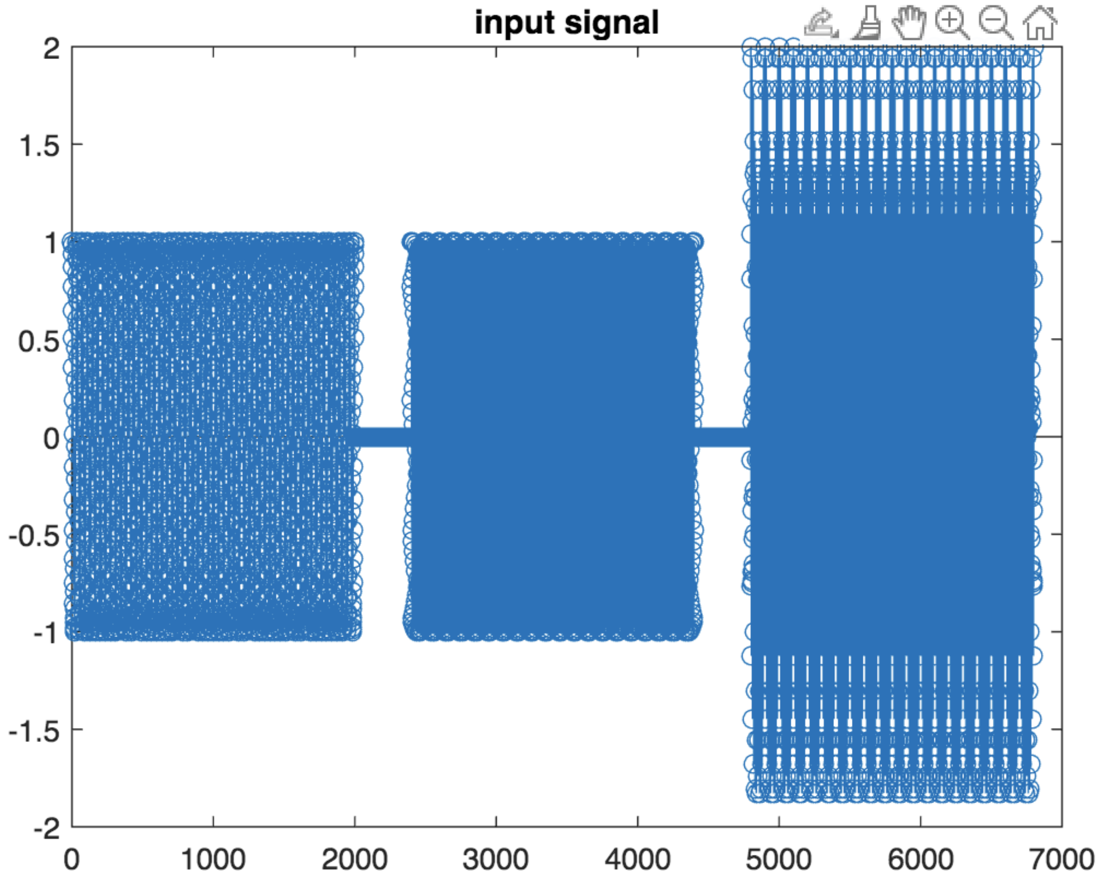


The lower octave filters overlap significantly in the transition region and close to the passband,

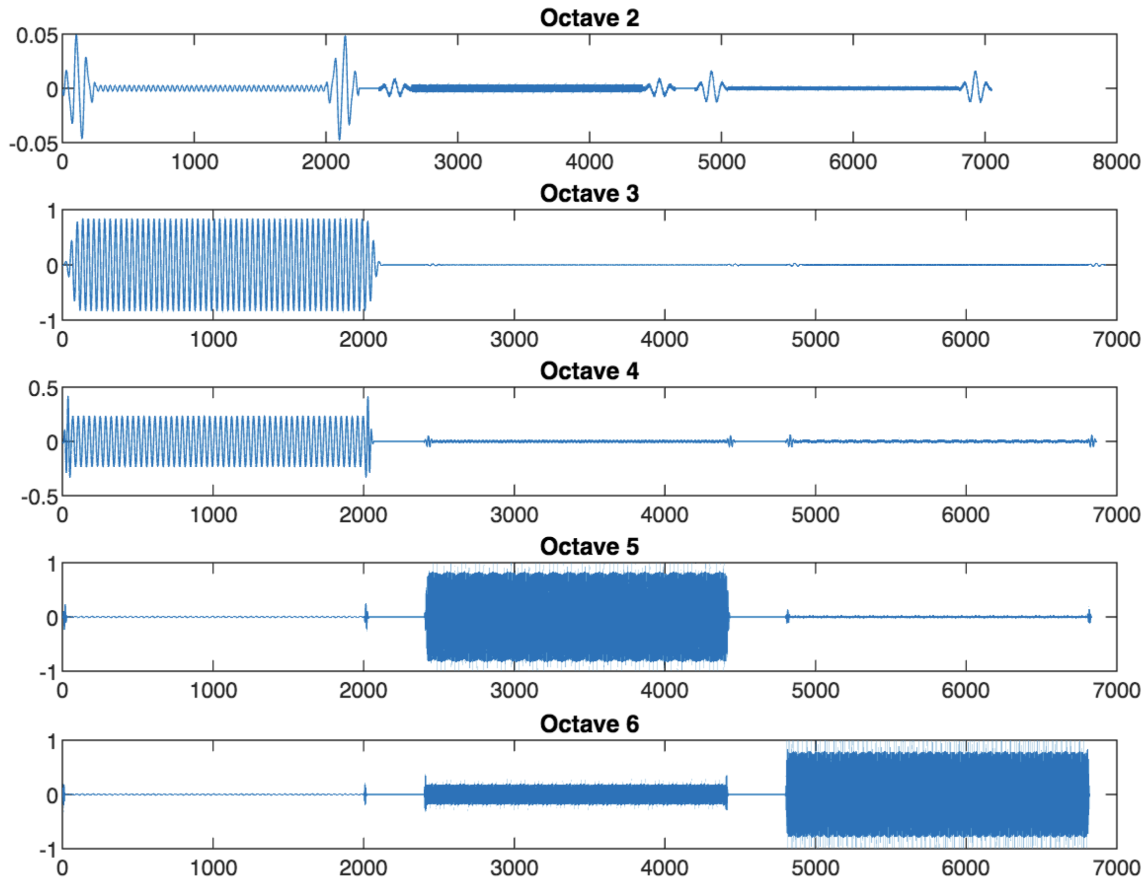
However, this is not an issue for higher octaves.

Piano Octave Decoding:

Testing:



After applying the filter, we get the following filtered signals. All the expected frequencies are passed by the filter banks.



Question 3:

Octave filter 3 has a frequency passband between 130.81-246.94 Hz and should pass the 220 Hz cosine. This matches the relationship we observe in the second subplot. Similarly, octave filter 4 passes the expected frequency of 440Hz. Octave filter 5 passes 800 Hz and octave 6 the 1760Hz.

Question 4:

The transient period decreases for the higher octaves as the length L decreases. We observe a high transient period of 250/8000, 0.03125s for Octave 2 and a short transient period of 0.0038 for octave 6.

Testing:

Sout1:

```
>> Sout1(1:15)

ans =

     6     6     6     6     3     3     5     5     3     4     4     4     4     3     3

>> |
```

Sout2:

```
>> Sout2(1:15)

ans =

     0     0     0     0     6     6     0     0     4     5     5     5     5     4     6
```

True notes:

```
true_notes =

    66    53    55    53    41    22    21    66    34    31     0    72    59    62    39
     0     0    46    71     0     0     0     0    37    27    63     0     0     0    63
```

The results generally agree with the true notes. The filter system has an accuracy of 40% . The inaccuracies happen when a note is near the boundaries of two adjacent octaves and more importantly between abrupt changes between notes. The system would work better if the notes in the original signal were ordered.