SAI KASYAP KAMARAJU
G00845717

ECE 535 PROJECT -2 MATLAB

(10 Points) Use quan2N(N,b) to quantize b = pi+j*pi using N= 16, 4, and 1 bits. Explain the results you obtain.

Solution :

N value	Result	command
For N = 16	3.1416 + 3.1416i	quan2N(16,pi+j*pi)
N = 4,	3.0000 + 3.0000i	quan2N(4,pi+j*pi)
N =1,	4.0000 + 4.0000i	quan2N(1,pi+j*pi)

Explanation:
The value of pi is 3.141592….
We observe that only for the value of N = 16 is close enough to pi+j*pi.
The results indicate that the higher the value of N , the better the precision it has.
As the value of the N decreases, the accuracy of quantized filter also decreases
.

2. (20 Points) This function can be used to quantize filter coefficients in different ways. To illustrate, consider an LTI system with transfer function:

```
                 1-1.13z^-1
H(z)   =    ------------------------
              1-1.5z^ -1+0.9z^-2
```

Find the pole and zero locations of this filter using full precision.
Solution : Givien a transfer function .
The pole zero locations of the unquantized filter are found out in the following way in MATLAB :
Firstly the coefficients of unquantized filter are defined in the following way :

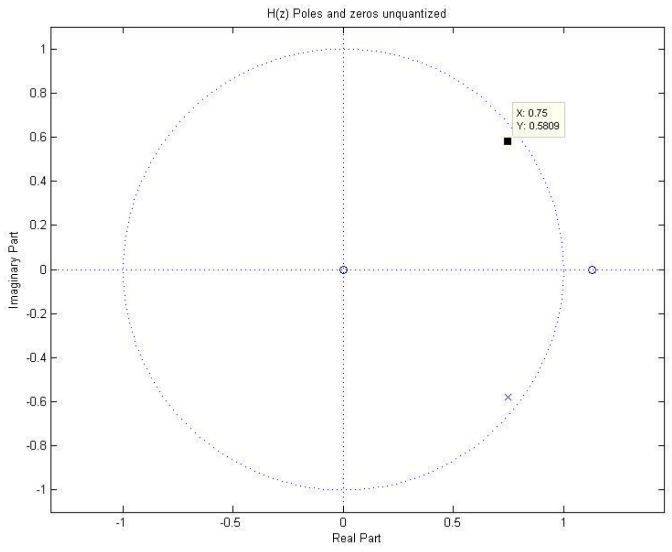
```
b = [1 -1.13]; % numerator
a =[1 -1.5 0.9];% denominator
```

Now the pole zero locations are plotted by using 'zplane' function in matlab, which is done as shown below.

```
zplane(b,a); title('H(z) Poles and zeros unquantized')
```

For full precision , the value of N in the quantizer function should be high. This can be done again in two ways as explained in the next question.

Result :



The values of zero and poles are as follows :
Zero : 1.1300 Poles : 0.7500 + 0.5809i
0.7500 - 0.5809i
In this question , the coefficients of the filter are quantized by either by individual quantization or group quantization;

Individual Quantization :
In this type of quantization , all coefficients are quantized individually and then all are the quantized coefficients are used as filter coefficients as shown below:

```
bq = quan2N(3,-1.13);
bq1 = [1 bq];
 aq1 = quan2N(3,-1.5);
 aq2 = quan2N(3,0.9);
 aq = [1 aq1 aq2];
```

Group Quantization : All coefficients in the filter are grouped and only one quantizer function is used as shown below :

```
b = [1 -1.13];
a = [1 -1.5 .9];
v = [b a];
v_q = quan2N(3,v);
b_q = v_q(1:2);
a_q = v_q(3:5);
```

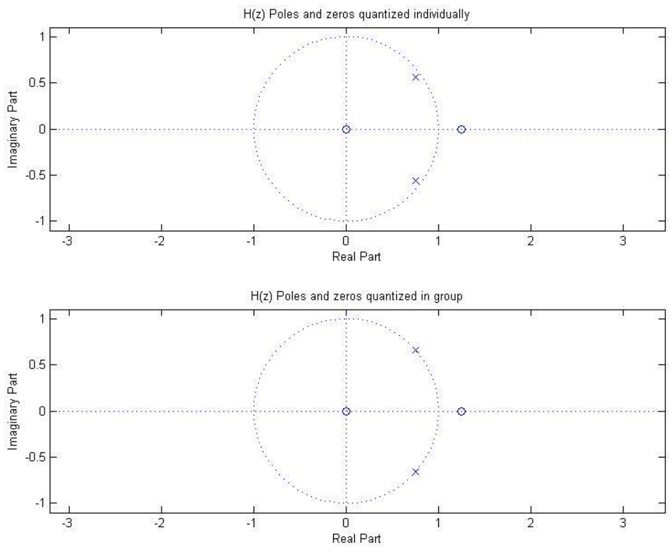
The poles and zeros are plotted again using zplane function in matlab.

SAI KASYAP KAMARAJU
G00845717

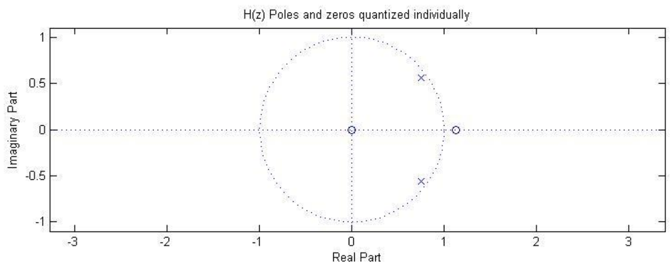
Observation :

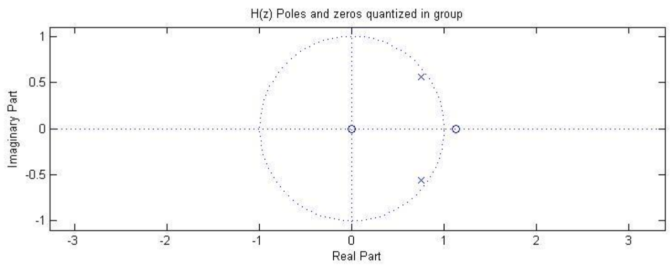
Value of N (bits)	Individual quantization zeros	Individual quantization poles	Group quantization zeros	Group quantization poles
3	1.2500	0.7500 + 0.5590i 0.7500 - 0.5590i	1.2500	0.7500 + 0.6614i 0.7500 - 0.6614i
4	1.1250	0.7500 + 0.5590i 0.7500 - 0.5590i	1.1250	0.7500 + 0.5590i 0.7500 - 0.5590i
8	1.1328	0.7500 + 0.5796i 0.7500 - 0.5796i	1.1328	0.7500 + 0.5796i 0.7500 - 0.5796i

For value N = 3, the pole zero plot is as shown:

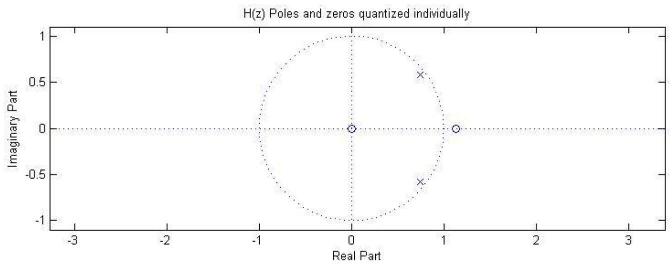


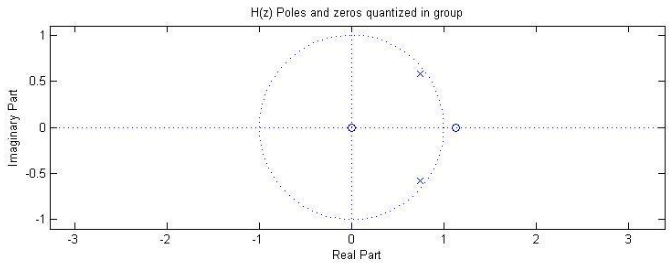
For the value N =4 , the pole –zero plot for individual and group quantization coefficients is as shown below:





For the value N =8, the pole –zero plots for individual and group quantization coefficients is as shown below:





Conclusion :
For N = 3 , the poles of group quantization coefficients are located on the unit circle, which makes the filter marginally stable.
And when it comes to individual quantization the poles are inside the unit circle(so the system is stable) and it looks to be more precise comparatively from the observation table.
For N =4 , Both the group and individual quantization coefficients pole zero plot is the same but it has less accurate pole & zeros.
For N= 8 , Again, Both pole-zero plots are the same and has high precision of quantization coefficients than N= 4. (from observation table)

Individual quantization is used to better predict actual filter performance since its precision is high even for low values of N.(lower bits) But need use more quantizers.

SAI KASYAP KAMARAJU
G00845717

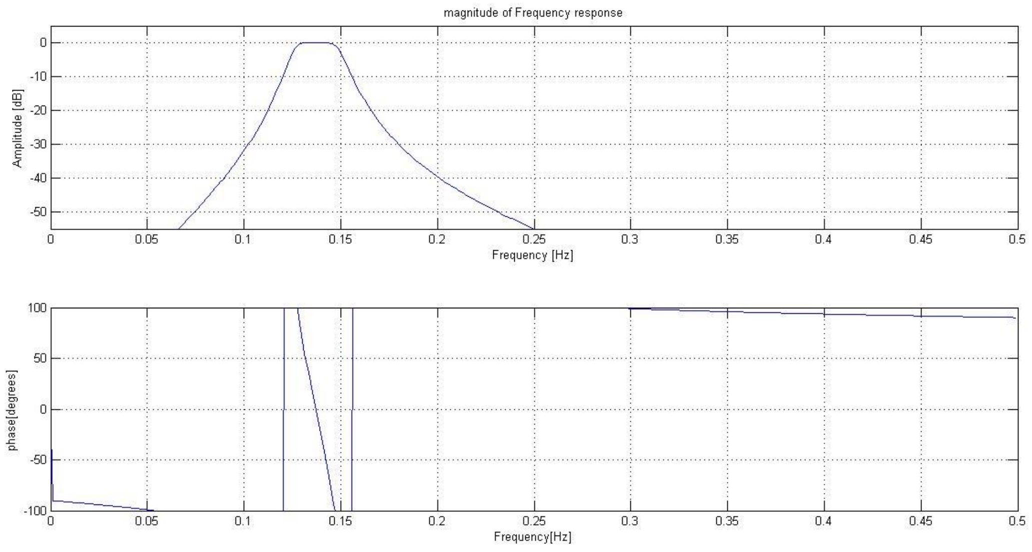
For Higher values of N, Both group and individual quantization have same precision rates for prediction.(for group , we can just use one quantizer)

Question 3 a :
A 6-pole band pass filter that is used to pass signals in the frequency range 0.25 Hz to 0.30Hz and fs =1 Hz.
[b_b a_b] = butter(3,[0.25 0.30],'bandpass');
The frequency Response of this filter is plotted by scaling the vertical axis from-55dB to +5dB and a frequency range(x-axis) from 0 to 0.5 Hz.
Solution : (i)
The frequency response of the filter is plotted by using the freqz command in the matlab as shown below:
[h,w] = freqz(b_b,a_b)

freqz command plots freq in rad/s , so we to convert rad/s to Hz by dividing by 2*pi.
For magnitude plot now use :
plot(w/(2*pi),20*log10(abs(h)))

Now the axis is scaled as specified from the range -55dB to +5dB and a frequency range from 0 to 0.5 Hz.
axis([0 0.5 -55 5])

The phase is plotted as shown below (Degrees vs Hz)
plot(w/(2*pi), 360/(2*pi)*angle(h))



(ii)Need to use group quantization to quantize the filter coefficients to N bits.
Which is done in the following way as shown below :

v = [b_b a_b];
v_q = quan2N(18,v);

```
b_q = v_q(1:7);
a_q = v_q(8:14);
[h,w] = freqz(b_q,a_q);
```

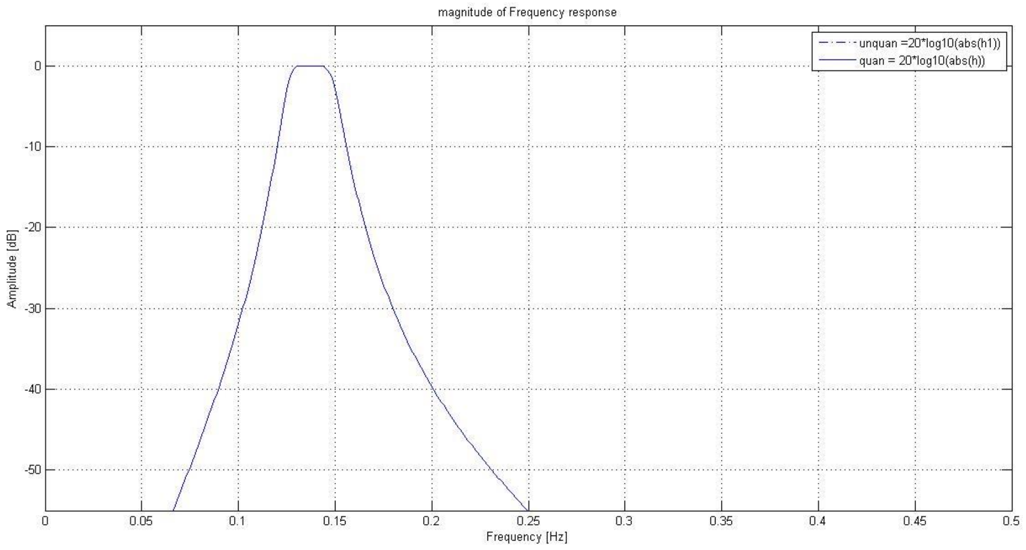
Now the frequency of quantized and unquantized coefficients are plotted using hold on in the following way :

```
% unquantized
figure()%magnitude
plot(w1/(2*pi),20*log10(abs(h1)))
hold on
%quantized filter freq response
plot(w/(2*pi),20*log10(abs(h)))
```

The axis scaling is done as the same way explained above.

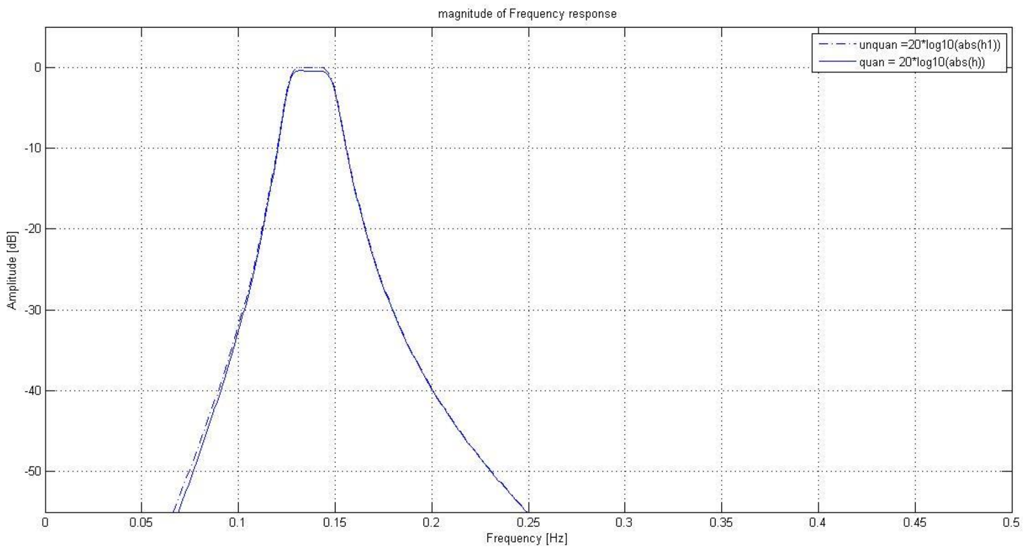
Observation :

For N = 32 to 19 ;
The plots are superimposed one on the other exactly



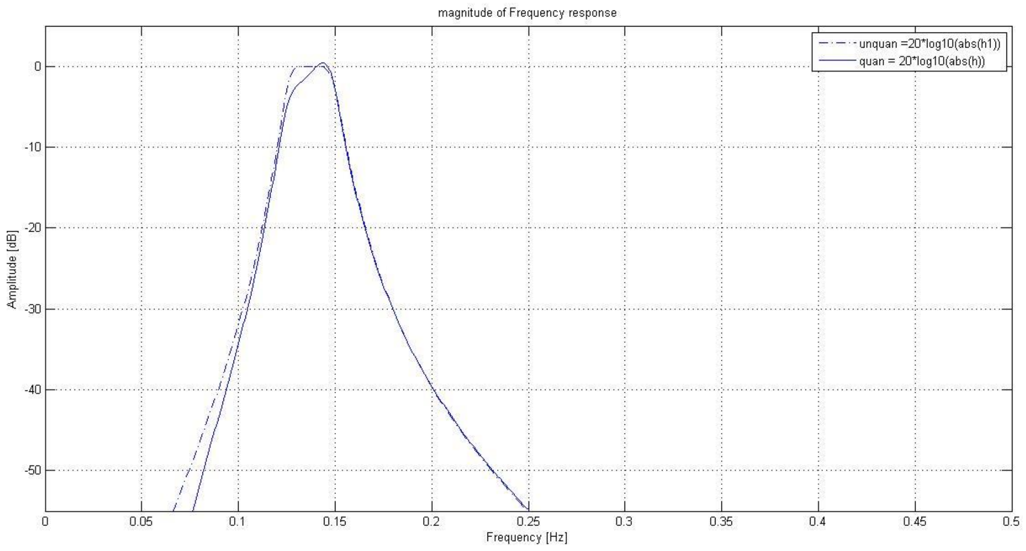
SAI KASYAP KAMARAJU
G00845717

For N = 18 ;

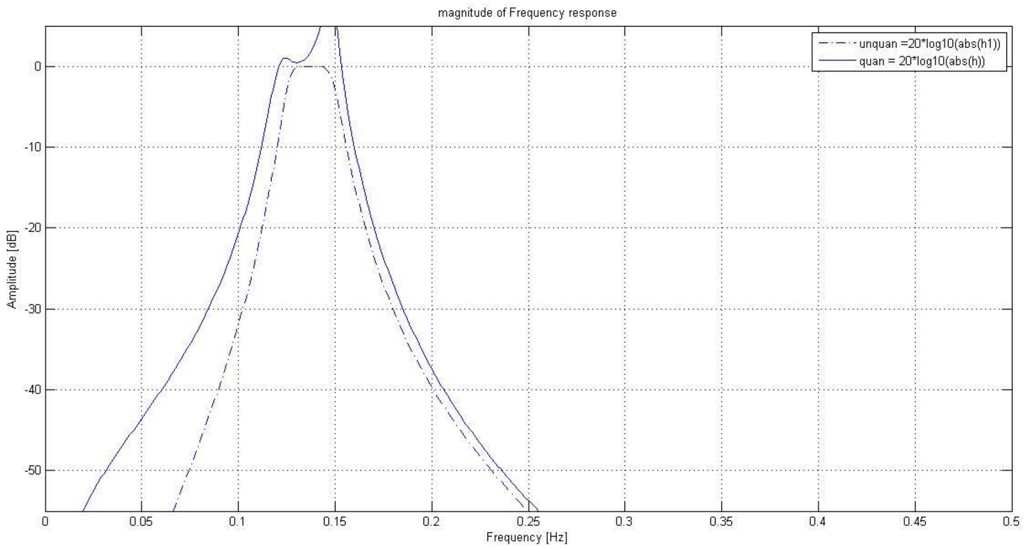


We can observe a slow level of performance degradation of the quantized filter from the unquantized one.

For N = 16 ;



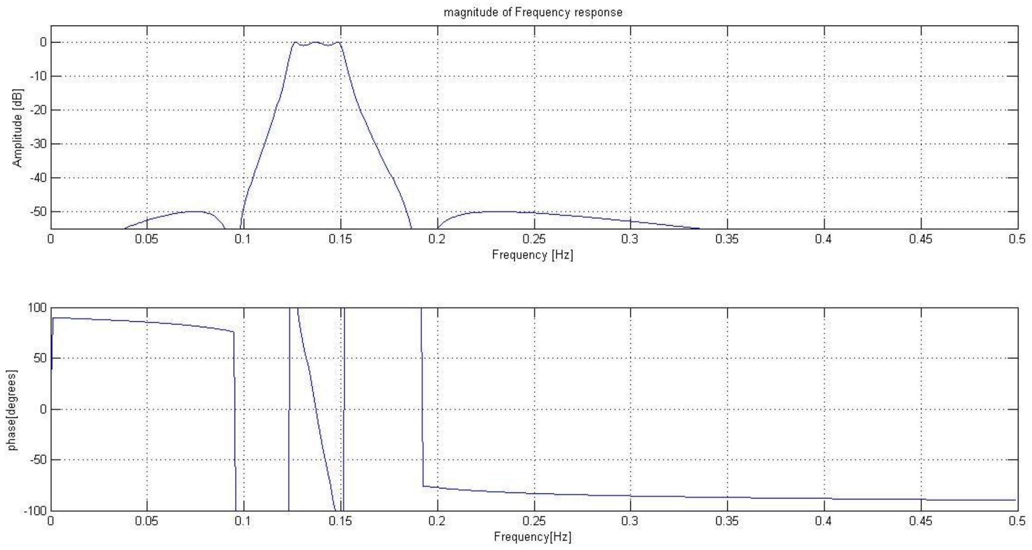
For N =13 ;



For N < = 12 bits we don't observe any quantized filter plot at all.

So minimum value of N for which the performance degradation occurs is 18.
N min = 18.

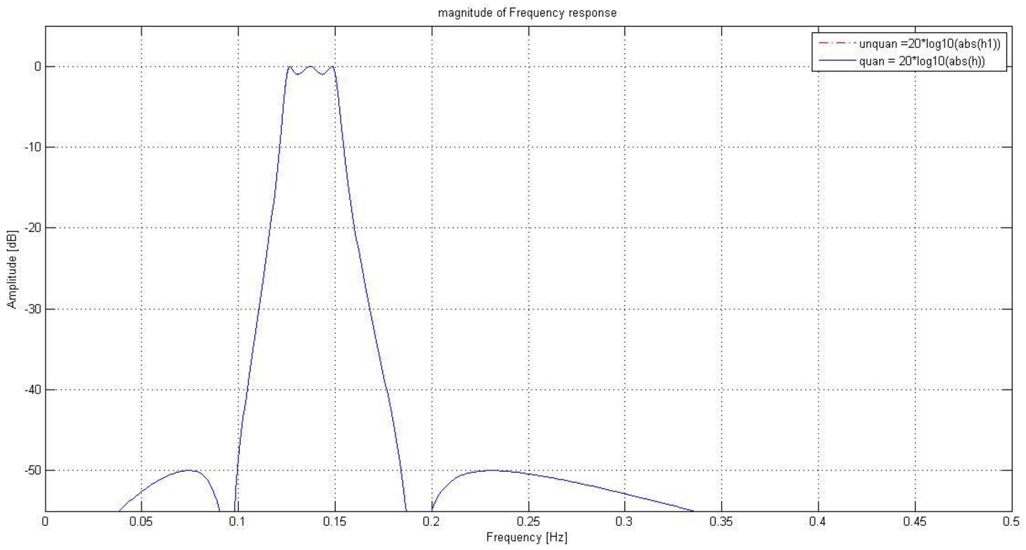
[b_e a_e] = ellip(3,1,50,[0.25 0.30],'bandpass');
This filter will have 1dB of pass band ripple and 50dB stop band rejection.
The frequency response of this filter is plotted in the same fashion as explained in part (a)
The frequency response of the filter is as shown below :



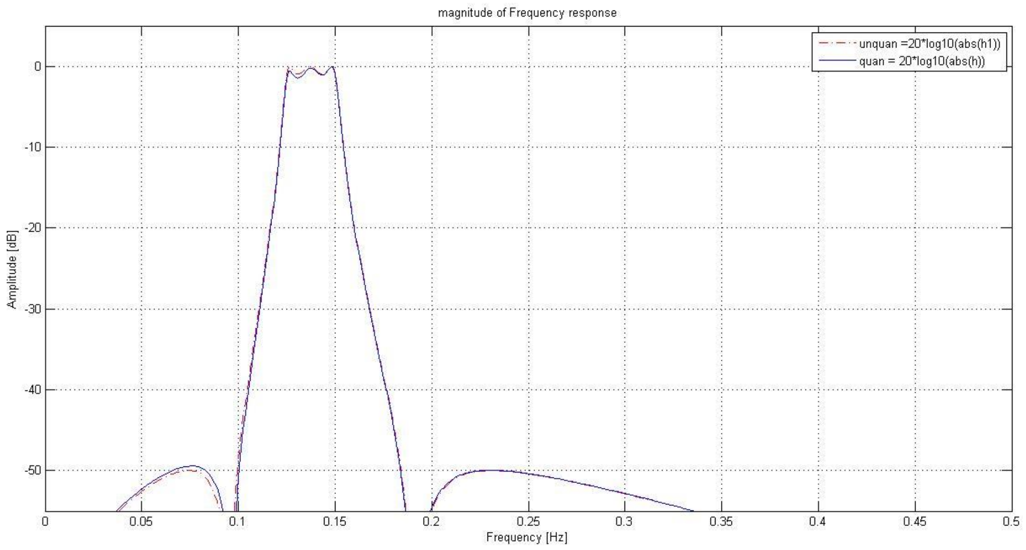
Need to use group quantization to quantize the filter coefficients to N bits which is done the in same fashion as explained above.

SAI KASYAP KAMARAJU
G00845717

Observation :
For N =32 to 20;
The graphs are superimposed on one another. So there is no performance degradation till N = 20.

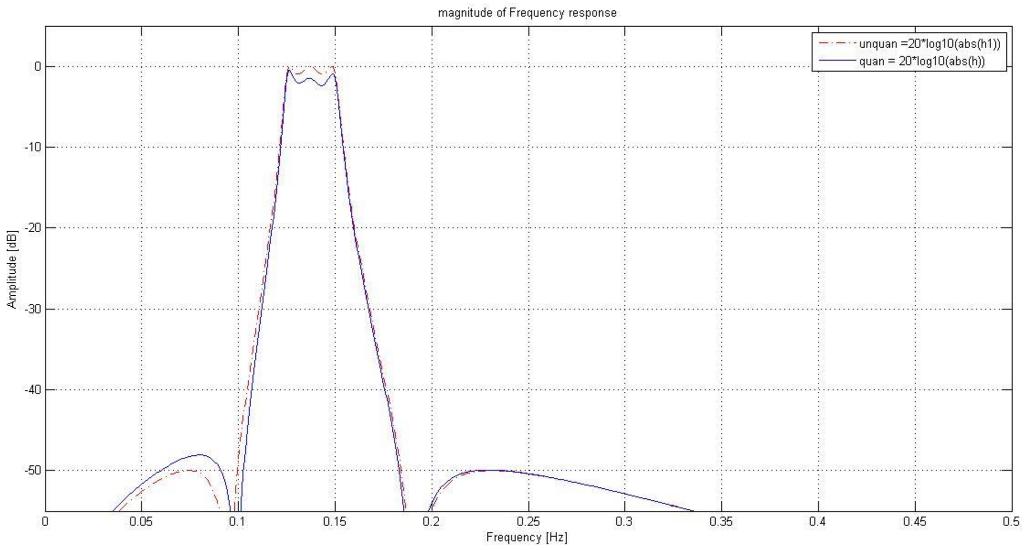


For N = 19;

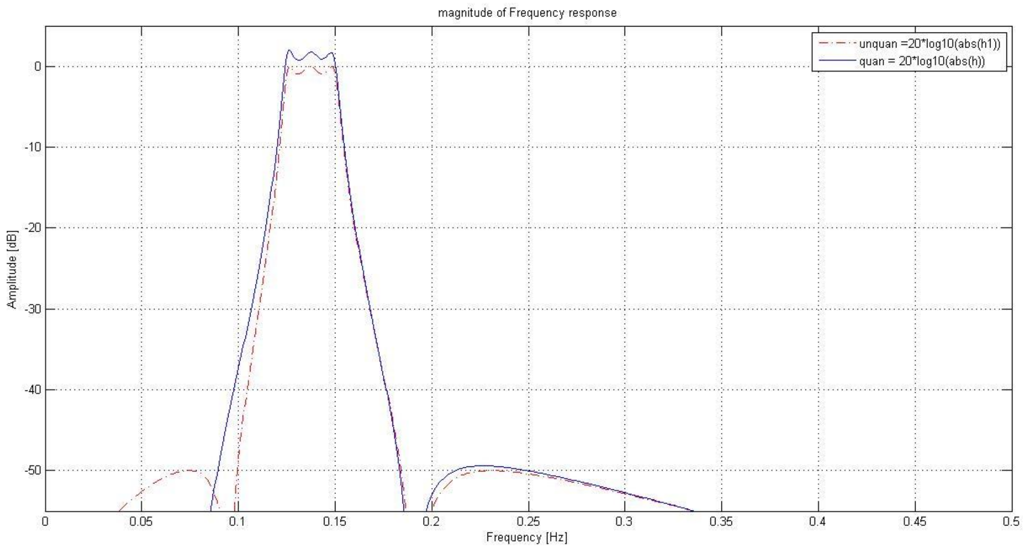


For N =19 , there is a performance of degradation of the filter which can be observed from the graph.
For N= 18;

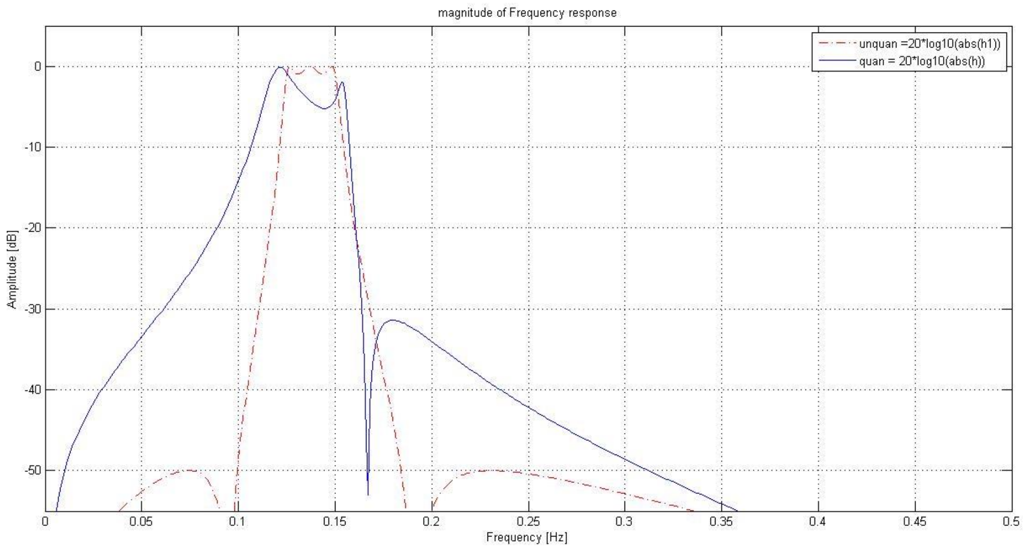
SAI KASYAP KAMARAJU
G00845717



For N =16;



N = 11;



For N <=10 , The quantized filter response is not observed.

So minimum value of N for which the performance degradation occurs is 19.

Comparing the graphs, the elliptic filters are have pass band and stop ripples in them Butterworth doesn't. (this is due to the specification we have given)
The band pass filter performance of the butter worth filter is better than elliptic filter.
As Elliptic filter looks more sensitive to the quantization of coefficients than the Butterworth filter.
N = 16 of butter worth has equivalent band pass performance to N =18 elliptic filter.

4. Given 6th-order Butterworth filter in part 3. has a transfer function that can be expressed as the product of three second-order systems:

$$H(z) = \frac{b_b(1) + b_b(2)z^{-1} + b_b(3)z^{-2} + \cdots\ b_b(6)z^{-6}}{1 + a_b(2)z^{-1} + a_b(3)z^{-2} + \cdots\ a_b(6)z^{-6}}$$

$$= \left(\frac{b_{11} + b_{12}z^{-1} + b_{13}z^{-2}}{1 + a_{12}z^{-1} + a_{13}z^{-2}}\right)\left(\frac{b_{21} + b_{22}z^{-1} + b_{23}z^{-2}}{1 + a_{22}z^{-1} + a_{23}z^{-2}}\right)\left(\frac{b_{31} + b_{32}z^{-1} + b_{33}z^{-2}}{1 + a_{32}z^{-1} + a_{33}z^{-2}}\right)$$

The above statement is true because when we multiply all the coefficients of the corresponding 2nd order filter we get back the direct form i.e 6th order coefficients.
Moreover , This can be expressed in bilinear transformation. There was no aliasing observed when these are cascaded together.
Hence due to the following reasons , the above statement is true.
finding the bij and aij coefficients for the 6th-order Butterworth filter in part 3
finding numerator and denominator roots using roots function :

SAI KASYAP KAMARAJU
G00845717

NUMERATOR ROOTS

```
b_r = roots(b_b)
b_r =
  -1.0000
  -1.0000 + 0.0000i
  -1.0000 - 0.0000i
   1.0000
   1.0000 + 0.0000i
   1.0000 - 0.0000i.
```

DENOMINATOR ROOTS :

```
a_r = roots(a_b)
 a_r =
0.5729 + 0.7695i
0.5729 - 0.7695i
0.6753 + 0.6876i
0.6753 - 0.6876i
0.6039 + 0.6995i
0.6039 - 0.6995i
```

(ii),(iii) and(iv)
Finding the polynomial coefficients of numerator and denominator using poly function in matlab as shown below :

```
%for numerator
q1 =poly([b1 b2])
q2= poly([b3 b4])
q3 = poly([b5 b6])
```

```
%denominator
p1 = poly([a1 a2])
p2 = poly([a3 a4])
p3 =poly([a5 a6])
```

The result is observed as shown below in the command window :

Numerator polynominal coeff

```
q1 =
1.0000           2.0000 - 0.0000i
1.0000 - 0.0000i
q2 =
  1.0000           -0.0000 + 0.0000i  -
1.0000 - 0.0000i
q3 =
   1.0000   -2.0000    1.0000
```

denominator polynominal coeff

```
p1 =
   1.0000   -1.1459    0.9204
p2 =
   1.0000   -1.3506    0.9289
p3 =
   1.0000   -1.2079    0.8541
```

All the 18 coefficients are found out in the following way after using poly function :
For Denominator where p1 ,p2 , p3 are polynomial functions of denominator:

```
a_11 =  p1(1);
a_12 = p1(2);
a_13 = p1(3);
a_21 = p2(1);
a_22 = p2(2);
a_23 =  p2(3);
a_31 = p3(1);
a_32 = p3(2);
a_33 = p3(3);
```

For Numerator where q1 ,q2 , q3 are polynomial functions of numerator:

```
b_11 = q1(1);
b_12 = q1(2);
```

```
b_13 = q1(3);
b_21 = q2(1);
b_22 = q2(2);
b_23 = q2(3);
b_31 = q3(1);
b_32 = q3(2);
b_33 = q3(3);
```

All the 18 coefficients must be quantized for N =16;

Solution :
Firstly , the numerator and denominator coefficients are divided in to 3 array vectors each as shown :

```
%for numerator
c_1 = [b_11 b_12 b_13];
        c_2 = [b_21 b_22 b_23];
        c_3 = [b_31 b_32 b_33];
        num = conv(c_1,conv(c_2,c_3)); %convolving

%for denominator

d_1 = [a_11 a_12 a_13];
d_2 = [a_21 a_22 a_23];
d_3 = [a_31 a_32 a_33];
den = conv(d_1,conv(d_2,d_3));
```

Now they are group quantized in following way :

```
v_gb = [num den];
v_gb = quan2N(16,v_gb); %for N =16;
num_qb = v_gb(1:7);
den_qb = v_gb(8:14);
```

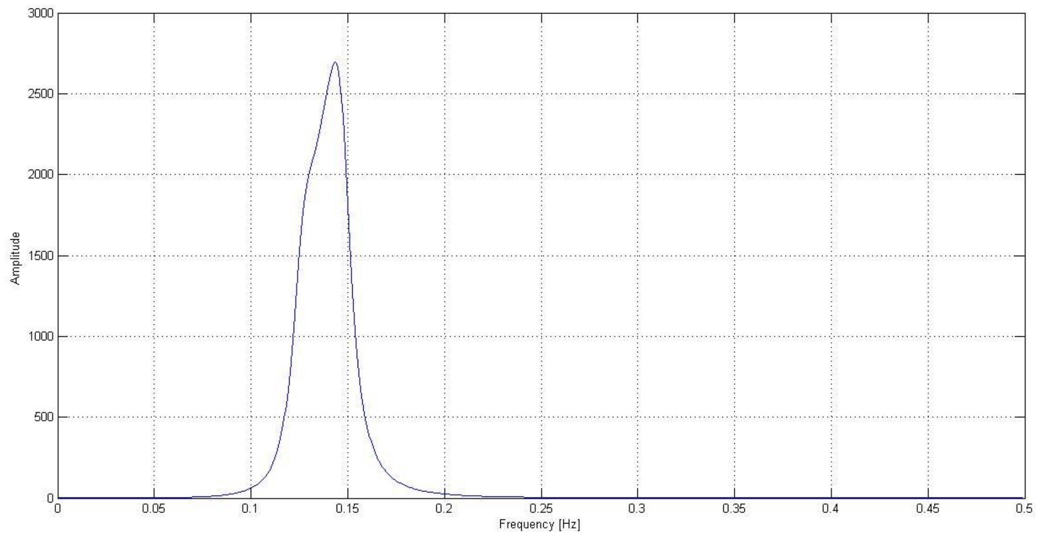
The frequency response of the group quantized filter for 16 bits is plotted as shown:

Code :

```
[H3,W3] = freqz(num_qb,den_qb);

plot(W3/(2*pi),(abs(H3)))
 xlabel('Frequency [Hz]'), ylabel('Amplitude ')
axis([0 0.5 0 3000]), grid
```

SAI KASYAP KAMARAJU
G00845717



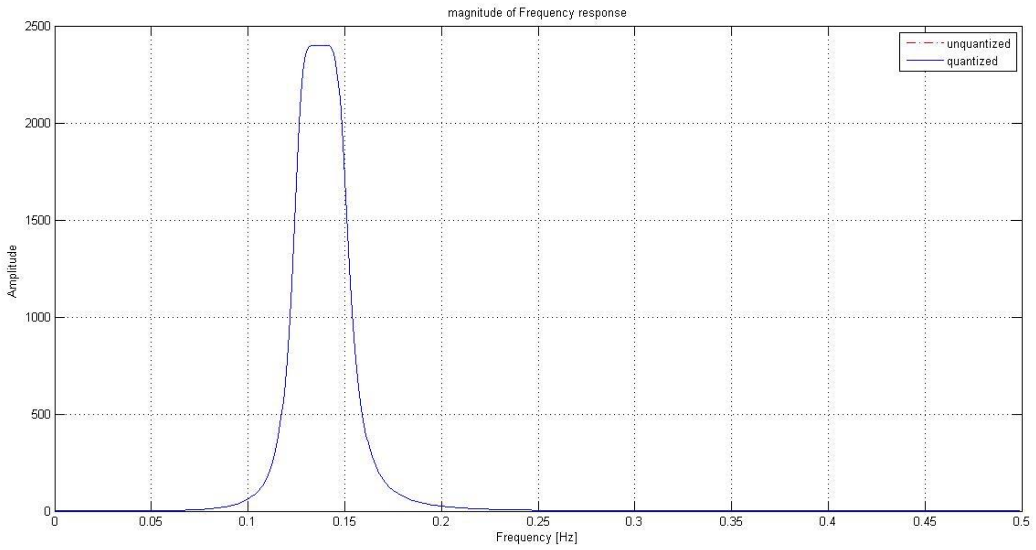
In this individual quantization of coefficients is done in the following way :
After getting all the 18 coefficients ;apply quantizers to filter each coefficient ;

```
%num coeff
qb_11 = quan2N(N,b_11);
qb_12 = quan2N(N,b_12);
qb_13 = quan2N(N,b_13);
qb_21 = quan2N(N,b_21);
qb_22 = quan2N(N,b_22);
qb_23 = quan2N(N,b_23);
qb_31 = quan2N(N,b_31);
qb_32 = quan2N(N,b_32);
qb_33 = quan2N(N,b_33);
%den coeff
qa_11 = quan2N(N,a_11);
qa_12 = quan2N(N,a_12);
qa_13 = quan2N(N,a_13);
qa_21 = quan2N(N,a_21);
qa_22 = quan2N(N,a_22);
qa_23 = quan2N(N,a_23);
qa_31 = quan2N(N,a_31);
qa_32 = quan2N(N,a_32);
qa_33 = quan2N(N,a_33);
```

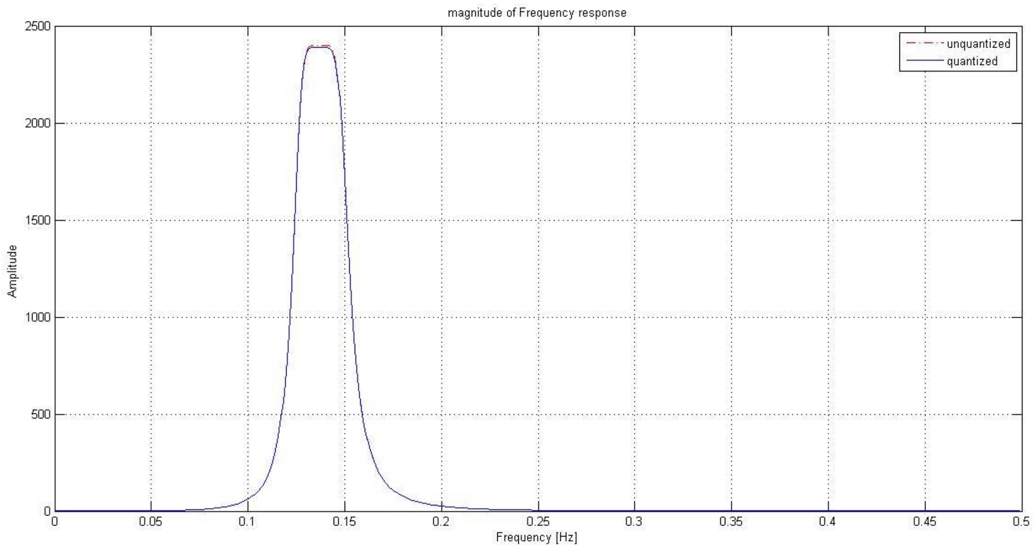
Now convolve the functions in the similar way as shown above.
Now the quantized and unquantized filters are compared from starting from the value of N = 16;

For N =16 to 12;
The quantized filter response and unquantized filter response super impose on each other exactly.

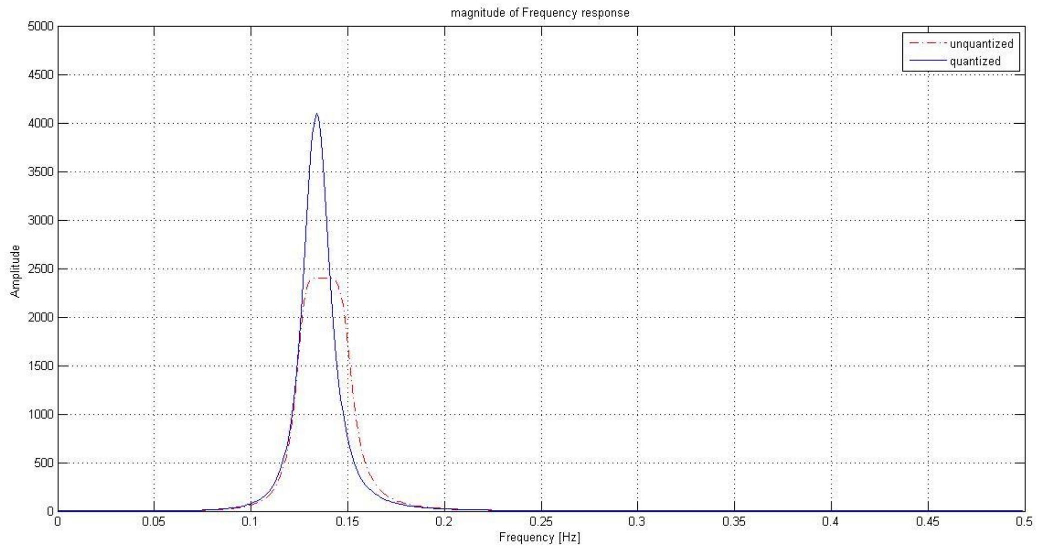
SAI KASYAP KAMARAJU
G00845717



For N =11;
There is a degradation of quantized filter response as shown in the figure below :



For N =3



For still more decrement in N the amplitude of the quantized filter tends to go infinity.

So Nmin =12 , for which the factored quantized filter has response equal to the unquantized filter.

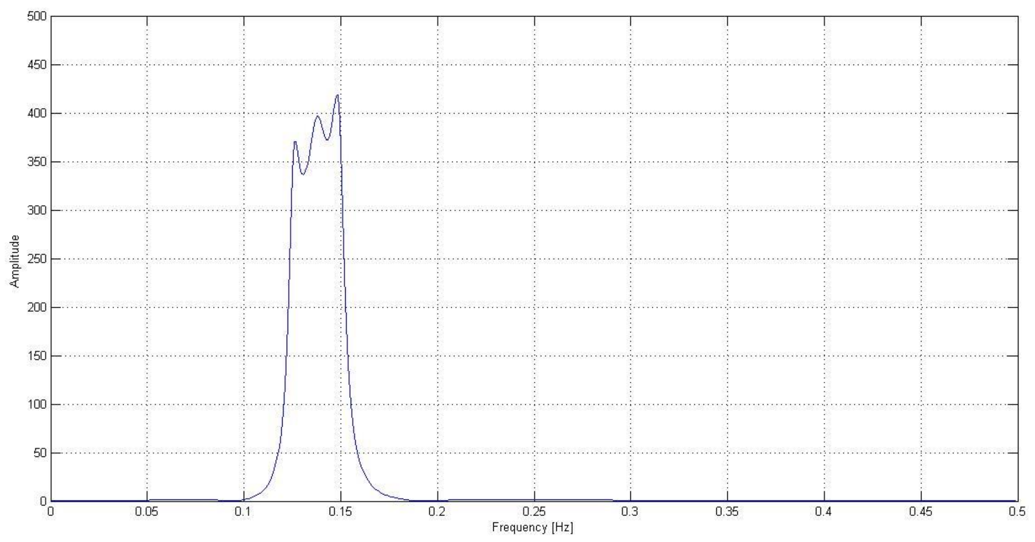
e)The amplitude seems to be relatively high when compared with the magnitude of frequency response of question 3.
We need less number of bits while cascading the 2nd order filter.
i.e
as less as 12 bits produces results equivalent to 19 bits for the original Butterworth filter.
This helps in reduced quantization levels.
The poles of original butterworth filter (direct form) looks tightly clustered so they are sensitive to coefficient changes.
Therefore , the direct or original butterworth filter suffers from quantization effects.
Where as ,in the cascaded form the poles are relatively spread.
So they are less sensitive to quantization effects.

5. [b_b a_b] = ellip(3,1,50,[0.25 0.30],'bandpass');

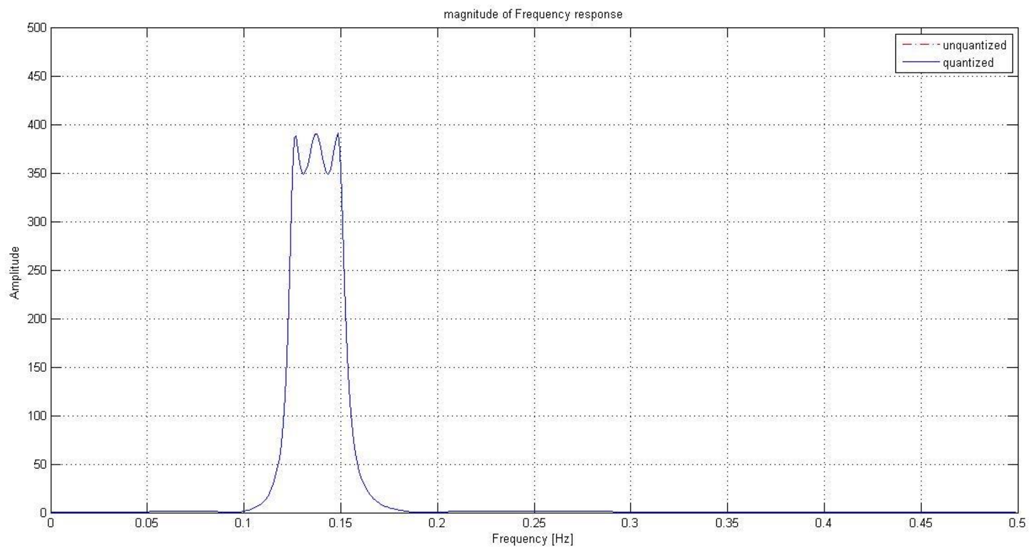
–→ b) It is true it can be cascaded (same explanation as mentioned in question 4)
The procedure is repetition of 4.
The graphs for d are plotted as shown below:

C) Group quantization for N =16 is done in the same fashion as mentioned in the above 4(c) part.

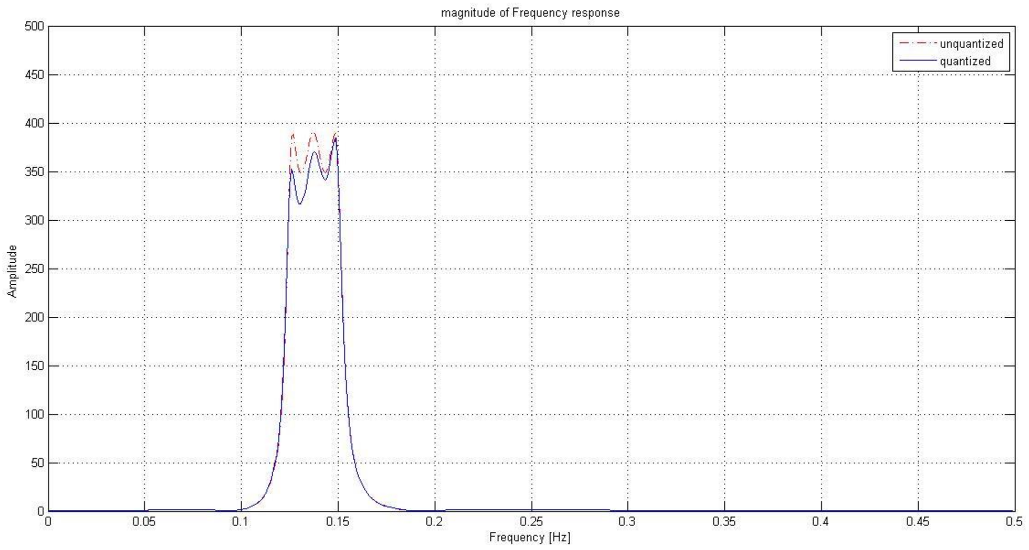
The frequency response for N =16 group quantized filter is as shown below :



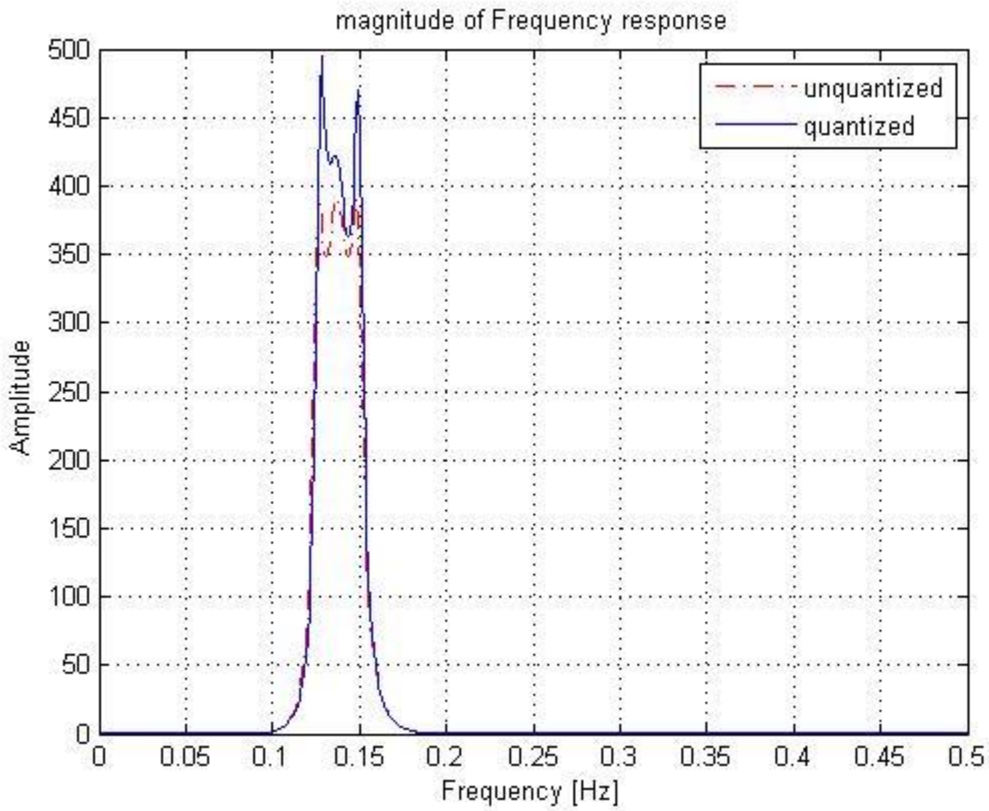
D)It is true it can be cascaded (same explanation as mentioned in question 4)
The procedure is once again the same as in question 4.
The graphs for different values of N are plotted as shown below :
For N =16 to 9 ;
The quantized filter response and unquantized filter response super impose on each other exactly.



For N = 8 ; We get a slightly fractured quantized filter response as shown below :



As N decreases more , we observe that the amplitude of the quantized filter keeps on increasing.



The amplitude seems to be relatively high when compared with the magnitude of frequency response of question 3b.
We need less number of bits while cascading the 2nd order filter.
i.e
as less as 19 bits produces results equivalent to 8 bits for the original elliptic filter.
This helps in reduced quantization levels.

SAI KASYAP KAMARAJU
G00845717

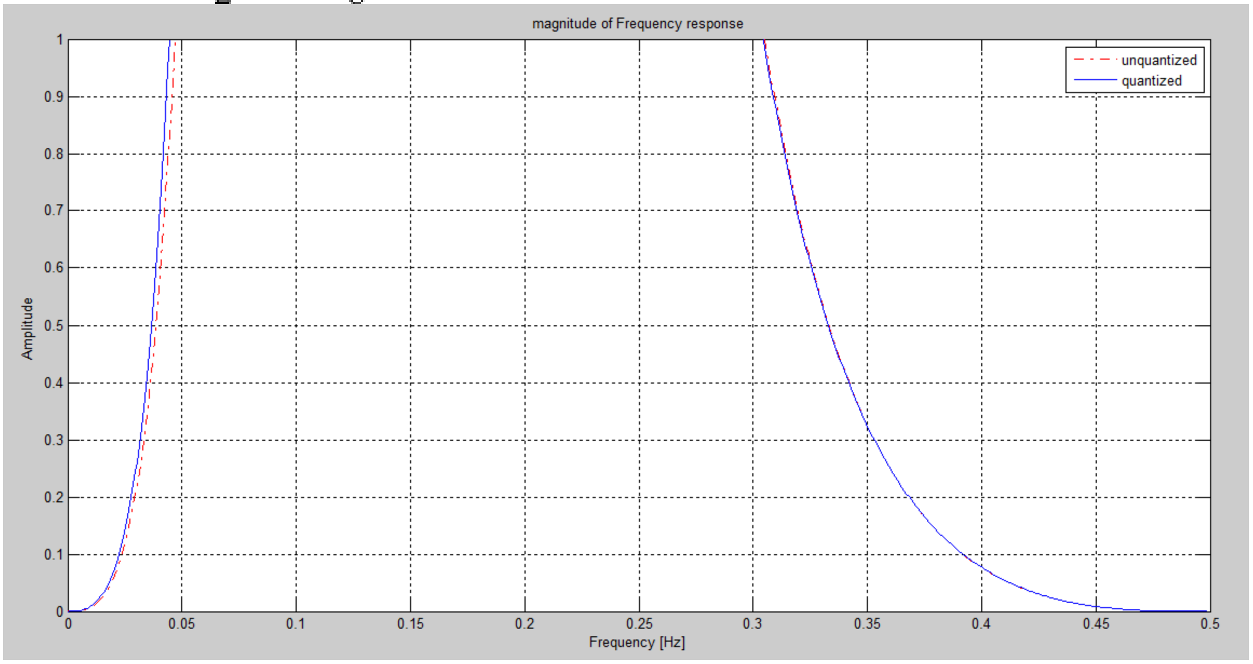
The poles of original elliptic filter (direct form) looks tightly clustered so they are sensitive to coefficient changes.
Therefore, the direct or original elliptic filter suffers from quantization effects.
Whereas, in the cascaded form the poles are relatively spread.
So they are less sensitive to quantization effects.

The passband and stopband ripples are eliminated in the case of cascaded elliptic filter but they are observed in original one.

The bandwidth of Butterworth filter is more than the elliptical filter.
The gain of the butterworth filter is more than the elliptic filter.
For band pass filters , Butterworth looks more ideal than the elliptic filters.
Butterworth filters look more precise in terms of upper and lower cutoff frequencies than elliptic filters.
For butterworth filters u require reduced number of quantization level than elliptic filters.

If we take gain as one for butterworth filter,maximum flatness is observed.

$$G(\omega) = 1 - \frac{1}{2}\omega^{2n} + \frac{3}{8}\omega^{4n} + \ldots$$



Butterworth filter has high frequency roll off :

$$\lim_{\omega \to \infty} \frac{d\log(G)}{d\log(\omega)} = -n.$$

Elliptic filters are sharp. So more sensitive than butterworth filters.

$$Q = -\frac{|s_{pm}|}{2\mathrm{Re}(s_{pm})} = -\frac{1}{2\cos(\arg(s_{pm}))}$$

(Q factor of a pole; elliptic)

SAI KASYAP KAMARAJU
G00845717

APPENDIX MATLAB CODE DSP PROJECT 2

1.

```
%load  the file quan2N
%  for N =4
quan2N(4,pi+j*pi)

%  for N =1

quan2N(1,pi+j*pi)

%for N =16

quan2N(16,pi+j*pi)
```

2.

```
% question 2 a

b = [1 -1.13];
a =[1 -1.5 0.9];
zplane(b,a); title('H(z) Poles and zeros unquantized')
figure
fvtool(b,a);title('H(z)using fvtool function')
figure
freqz(b,a); title('H(z) Frequency Response')

%Question 2b
%For N = 3 bits
%individual quantization vs group Quantization
%Individual
N=3;
bq = quan2N(N,-1.13);
bq1 = [1 bq];
 aq1 = quan2N(N,-1.5);
 aq2 = quan2N(N,0.9);
 aq = [1 aq1 aq2];
 % Ploting pole zeros
 figure
 subplot(2,1,1)
 zplane(bq1,aq); title('H(z) Poles and zeros quantized individually')
 [z,p,k] = tf2zp(bq1,aq)

 %For Group quantization

b = [1 -1.13];
a = [1 -1.5 .9];
v = [b a];
v_q = quan2N(N,v);
```

```
b_q = v_q(1:2);
a_q = v_q(3:5);
% Ploting pole zeros
subplot(2,1,2)
 zplane(b_q,a_q); title('H(z) Poles and zeros quantized in group')
  [z1,p1,k1] = tf2zp(b_q,a_q)
```

```
% %question 3
%question a(i)

[b_b a_b] = butter(3,[0.25 0.30],'bandpass');

[h,w] = freqz(b_b,a_b);
%magnitude
figure()
%unquantized filter freq response
subplot(2,1,1)
plot(w/(2*pi),20*log10(abs(h)))
title('magnitude of Frequency response');
xlabel('Frequency [Hz]'), ylabel('Amplitude [dB]')
axis([0 0.5 -55 5]), grid
%phase
subplot(2,1,2)
plot(w/(2*pi), 360/(2*pi)*angle(h))
xlabel('Frequency[Hz]'), ylabel('phase[degrees]')
axis([0 0.5 -100 100]), grid
%question 3 a(ii)
[b_b a_b] = butter(3,[0.25 0.30],'bandpass');

v = [b_b a_b];
v_q = quan2N(12,v);
b_q = v_q(1:7);
a_q = v_q(8:14);


[h,w] = freqz(b_q,a_q);% quantized
[h1,w1] = freqz(b_b,a_b);
% unquantized
figure()
```

```
plot(w1/(2*pi),20*log10(abs(h1)),'-.b')
%magnitude
hold on
%quantized filter freq response
plot(w/(2*pi),20*log10(abs(h)))
%unquantized filter freq response
title('magnitude of Frequency response');
xlabel('Frequency [Hz]'), ylabel('Amplitude [dB]')
 axis([0 0.5 -55 5]), grid
legend('unquan =20*log10(abs(h1))','quan = 20*log10(abs(h))');

%question 3 b (i)

[b_e a_e] = ellip(3,1,50,[0.25 0.30],'bandpass');

[h,w] = freqz(b_e,a_e);
%magnitude
figure()
%unquantized filter freq response
subplot(2,1,1)
plot(w/(2*pi),20*log10(abs(h)))
title('magnitude of Frequency response');
xlabel('Frequency [Hz]'), ylabel('Amplitude [dB]')
axis([0 0.5 -55 5]), grid
%phase
subplot(2,1,2)
plot(w/(2*pi), 360/(2*pi)*angle(h))
xlabel('Frequency[Hz]'), ylabel('phase[degrees]')
axis([0 0.5 -100 100]), grid
%question 3 b (ii)

[b_e a_e] = ellip(3,1,50,[0.25 0.30],'bandpass');
v = [b_e a_e];
v_q = quan2N(11,v);
b_q = v_q(1:7);
a_q = v_q(8:14);

[h,w] = freqz(b_q,a_q);
[h1,w1] = freqz(b_e,a_e);
figure()

%unquantized filter freq response
plot(w1/(2*pi),20*log10(abs(h1)),'-.r')
%magnitude
hold on
%quantized filter freq response
plot(w/(2*pi),20*log10(abs(h)))
title('magnitude of Frequency response');
```

```
xlabel('Frequency [Hz]'), ylabel('Amplitude [dB]')
 axis([0 0.5 -55 5]), grid
legend('unquan =20*log10(abs(h1))','quan = 20*log10(abs(h))');


%Question 4
[b_b a_b] = butter(3,[0.25 0.30],'bandpass');
% question a i
b_b
a_b
% question a ii
b_r = roots(b_b);

%roots
b1 = b_r(1);
b2 = b_r(2);
b3 = b_r(3);
b4 = b_r(4);
b5 = b_r(5);
b6 = b_r(6);

% iii
q1 =poly([b1 b2]);
q2= poly([b3 b4]);
q3 = poly([b5 b6]);



b_11 = q1(1);
b_12 = q1(2);
b_13 = q1(3);
b_21 = q2(1);
b_22 = q2(2);
b_23 = q2(3);
b_31 = q3(1);
b_32 = q3(2);
b_33 = q3(3);
%  Question iv
a_r = roots(a_b);
%roots
a1 = a_r(1);
a2 = a_r(2);
a3 = a_r(3);
a4 = a_r(4);
a5 = a_r(5);
a6 = a_r(6);

p1 = poly([a1 a2]);
```

```
p2 = poly([a3 a4]);
p3 =poly([a5 a6]);


a_11 =  p1(1);
a_12 = p1(2);
a_13 = p1(3);
a_21 = p2(1);
a_22 = p2(2);
a_23 =  p2(3);
a_31 = p3(1);
a_32 = p3(2);
a_33 = p3(3);

c_1 = [b_11 b_12 b_13];
c_2 = [b_21 b_22 b_23];
c_3 = [b_31 b_32 b_33];

num = conv(c_1,conv(c_2,c_3));
% denominator coeff


d_1 = [a_11 a_12 a_13];
d_2 = [a_21 a_22 a_23];
d_3 = [a_31 a_32 a_33];

den = conv(d_1,conv(d_2,d_3));

% Group Quantization
v_gb = [num den];
v_gb = quan2N(16,v_gb);
num_qb = v_gb(1:7);
den_qb = v_gb(8:14);

% individual quantization

% Giving value of N here
N = 3;
%num coeff
qb_11 = quan2N(N,b_11);
qb_12 = quan2N(N,b_12);
qb_13 = quan2N(N,b_13);
qb_21 = quan2N(N,b_21);
qb_22 = quan2N(N,b_22);
qb_23 = quan2N(N,b_23);
qb_31 = quan2N(N,b_31);
qb_32 = quan2N(N,b_32);
qb_33 = quan2N(N,b_33);
```

```
%den coeff
qa_11 = quan2N(N,a_11);
qa_12 = quan2N(N,a_12);
qa_13 = quan2N(N,a_13);
qa_21 = quan2N(N,a_21);
qa_22 = quan2N(N,a_22);
qa_23 = quan2N(N,a_23);
qa_31 = quan2N(N,a_31);
qa_32 = quan2N(N,a_32);
qa_33 = quan2N(N,a_33);


C_1 = [qb_11 qb_12 qb_13];
C_2 = [qb_21 qb_22 qb_23];
C_3 = [qb_31 qb_32 qb_33];

numer = conv(C_1,conv(C_2,C_3));

D_1 = [qa_11 qa_12 qa_13];
D_2 = [qa_21 qa_22 qa_23];
D_3 = [qa_31 qa_32 qa_33];

denom = conv(D_1,conv(D_2,D_3));

[H,W] = freqz(numer,denom);
[H1,W1] = freqz(num,den);

figure()
%unquantized filter freq response
plot(W1/(2*pi),(abs(H1)),'-.r')
%magnitude
 hold on
% %quantized filter freq response
 plot(W/(2*pi),(abs(H)))
%
 title('magnitude of Frequency response');
 xlabel('Frequency [Hz]'), ylabel('Amplitude ')
axis([0 0.5 0 5000]), grid
legend('unquantized','quantized');


%question 5
[b_b a_b] = ellip(3,1,50,[0.25 0.30],'bandpass');
% question a i
b_b;
a_b;
% question a ii
b_r = roots(b_b);
```

SAI KASYAP KAMARAJU
G00845717

```
%roots
b1 = b_r(1);
b2 = b_r(2);
b3 = b_r(3);
b4 = b_r(4);
b5 = b_r(5);
b6 = b_r(6);

% iii
q1 =poly([b1 b2]);
q2= poly([b3 b4]);
q3 = poly([b5 b6]) ;



b_11 = q1(1);
b_12 = q1(2);
b_13 = q1(3);
b_21 = q2(1);
b_22 = q2(2);
b_23 = q2(3);
b_31 = q3(1);
b_32 = q3(2);
b_33 = q3(3);
%  Question iv
a_r = roots(a_b);
%roots
a1 = a_r(1);
a2 = a_r(2);
a3 = a_r(3);
a4 = a_r(4);
a5 = a_r(5);
a6 = a_r(6);

p1 = poly([a1 a2]);
p2 = poly([a3 a4]);
p3 =poly([a5 a6]);


a_11 =  p1(1);
a_12 = p1(2);
a_13 = p1(3);
a_21 = p2(1);
a_22 = p2(2);
a_23 =  p2(3);
a_31 = p3(1);
a_32 = p3(2);
```

```
a_33 = p3(3);

c_1 = [b_11 b_12 b_13];
c_2 = [b_21 b_22 b_23];
c_3 = [b_31 b_32 b_33];

num = conv(c_1,conv(c_2,c_3));
% denominator coeff


d_1 = [a_11 a_12 a_13];
d_2 = [a_21 a_22 a_23];
d_3 = [a_31 a_32 a_33];

den = conv(d_1,conv(d_2,d_3));

% Group Quantization
v_gb = [num den];
v_gb = quan2N(16,v_gb);
num_qb = v_gb(1:7);
den_qb = v_gb(8:14);

% individual quantization

% Giving value of N here
N = 6 ;
%num coeff
qb_11 = quan2N(N,b_11);
qb_12 = quan2N(N,b_12);
qb_13 = quan2N(N,b_13);
qb_21 = quan2N(N,b_21);
qb_22 = quan2N(N,b_22);
qb_23 = quan2N(N,b_23);
qb_31 = quan2N(N,b_31);
qb_32 = quan2N(N,b_32);
qb_33 = quan2N(N,b_33);
%den coeff
qa_11 = quan2N(N,a_11);
qa_12 = quan2N(N,a_12);
qa_13 = quan2N(N,a_13);
qa_21 = quan2N(N,a_21);
qa_22 = quan2N(N,a_22);
qa_23 = quan2N(N,a_23);
qa_31 = quan2N(N,a_31);
qa_32 = quan2N(N,a_32);
qa_33 = quan2N(N,a_33);
C_1 = [qb_11 qb_12 qb_13];
C_2 = [qb_21 qb_22 qb_23];
```

```
C_3 = [qb_31 qb_32 qb_33];

numer = conv(C_1,conv(C_2,C_3));

D_1 = [qa_11 qa_12 qa_13];
D_2 = [qa_21 qa_22 qa_23];
D_3 = [qa_31 qa_32 qa_33];

denom = conv(D_1,conv(D_2,D_3));

[H,W] = freqz(numer,denom);
[H1,W1] = freqz(num,den);
figure()
%unquantized filter freq response
plot(W1/(2*pi),(abs(H1)),'-.r')
%magnitude
 hold on
% %quantized filter freq response
 plot(W/(2*pi),(abs(H)))
%
 title('magnitude of Frequency response');
 xlabel('Frequency [Hz]'), ylabel('Amplitude ')
axis([0 0.5 0 500]), grid
legend('unquantized','quantized');
```