LAB#05

A huge thanks to Ms. Mehri for her kind support thus far ☺

Shayan Vassef, UIN = 674579894, $N_{id} = 894$

1 FIR filter design for comparison of filter order with IIR filters

Q5.1)

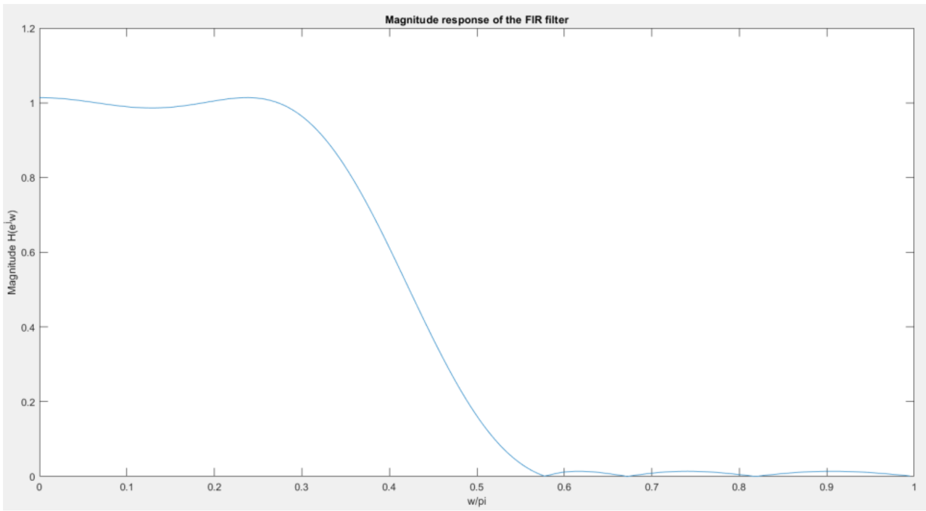

Fig 1. FIR filter magnitude response

MATLAB output:

Estimated filter order:

13

Maximum deviation for passband:

0.0140

Maximum deviation for stopband:

0.0166

MATLAB code:

```
clc
close all
dp=0.01;
ds=0.01;
NID = 894;
fp=0.375 - NID * 1e-4;
fs=0.475 + NID * 1e-4;
Fsampl=2;
[nfir_ord,fr_edge,des_mag,wt] = firpmord( [fp fs], [1 0], [dp ds], Fsampl);
hfir= firpm(nfir_ord,fr_edge,des_mag,wt);
[Hfir,w]=freqz(hfir,1,201);
figure;
plot(w/pi,abs(Hfir));
xlabel('w/pi')
ylabel('Magnitude H(e^jw)')
```

```
title('Magnitude response of the FIR filter')
disp('Estimated filter order:')
disp(nfir_ord)
%label axes in plot, put title, etc.
MAX_DEV = zeros(1,2);
for i = 1:2
    band_start = round(fr_edge(2*i - 1) * 201) + 1;
    band_end = round(fr_edge(2*i) * 201);
    MAX_DEV(i) = max(abs(abs(Hfir(band_start:band_end)) - des_mag(2*i-1)));
end

disp('Maximum deviation for passband:')
disp(MAX_DEV(1))
disp('Maximum deviation for stopband:')
disp(MAX_DEV(2))
```

2 Butterworth lowpass filter Design

Q5.2)

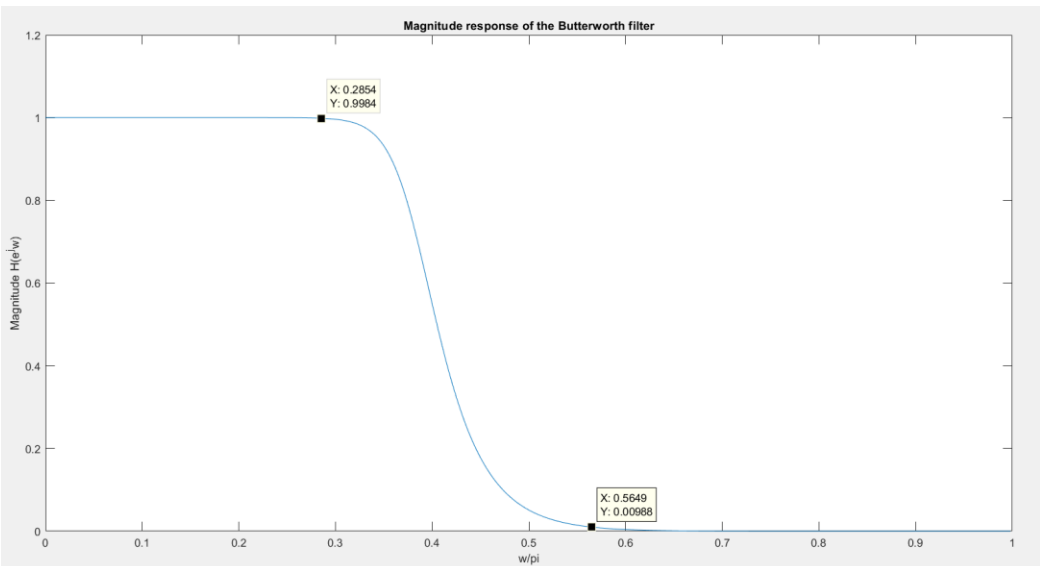

Fig 2. Butterworth filter magnitude response

As we know, the magnitude response of the Butterworth filter is monotonously decreasing, causing the maximum magnitude deviation in passband and stopband to happen at the given passband and stopband frequencies respectively.

Based on the specifications, the passband normalized frequency is 0.2856 and the stopband normalized frequency is 0.5644; based on the aforementioned point, we can approximate the maximum magnitude deviation for both bands:

$\Delta_{passband} = 1 - 0.9984 = 0.0016, \Delta_{stopband} = 0.00988 \cong 0.0099$

MATLAB output:

Estimated filter order:

8

maximum deviation for stopband happens at stopband normalized frequency that is equal to:

0.2856

maximum deviation for passband happens at passband normalized frequency that is equal to:

0.5644

Maximum deviation for passband:

0.0014

Maximum deviation for stopband:

0.0099

Based on the above outputs of MATLAB, the maximum magnitude deviation that was calculated using Figure 2. and MATLAB turned out to be the same.

3 Chebyshev Type I lowpass filter Design

3.1 Chebyshev Type I lowpass filter design with unquantized coefficients

Q5.3)

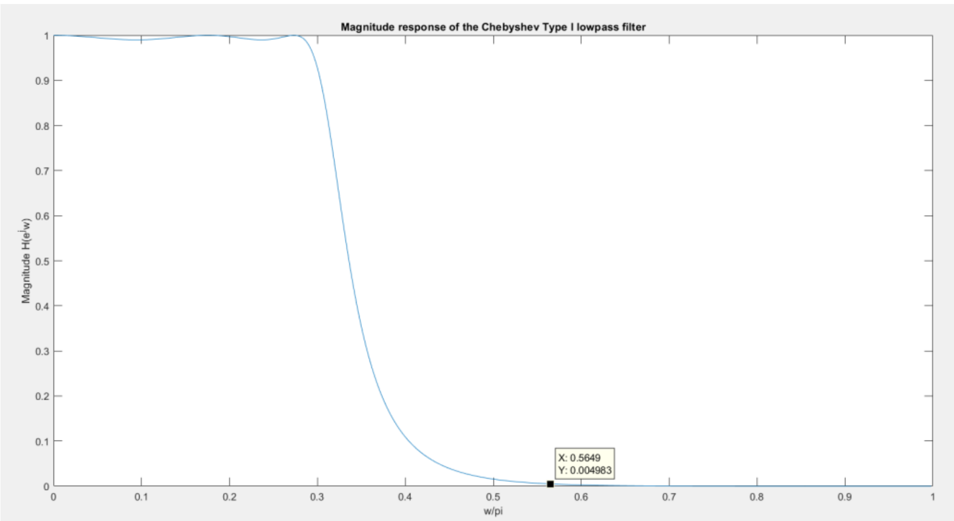

Fig 3. Chebyshev type 1 lowpass filter magnitude response

In the Chebyshev type I lowpass filter, the magnitude response is monotonously decreasing in the stopband, enabling us to derive the maximum magnitude deviation in the stopband using the magnitude response figure.

Based on the above figure, the maximum magnitude deviation in the stopband happens at the stopband normalized frequency (0.5644), which is equal to 0.0049883 or 0.0050 approximately.

MATLAB output:

Estimated filter order:

5

maximum deviation for passband happens at passband normalized frequency that is equal to:

0.5644

Maximum deviation for passband:

0.0100

Maximum deviation for stopband:

0.0050

4 Elliptic lowpass filter Design

4.1 Elliptic filter lowpass design with unquantized coefficients

Q5.4)

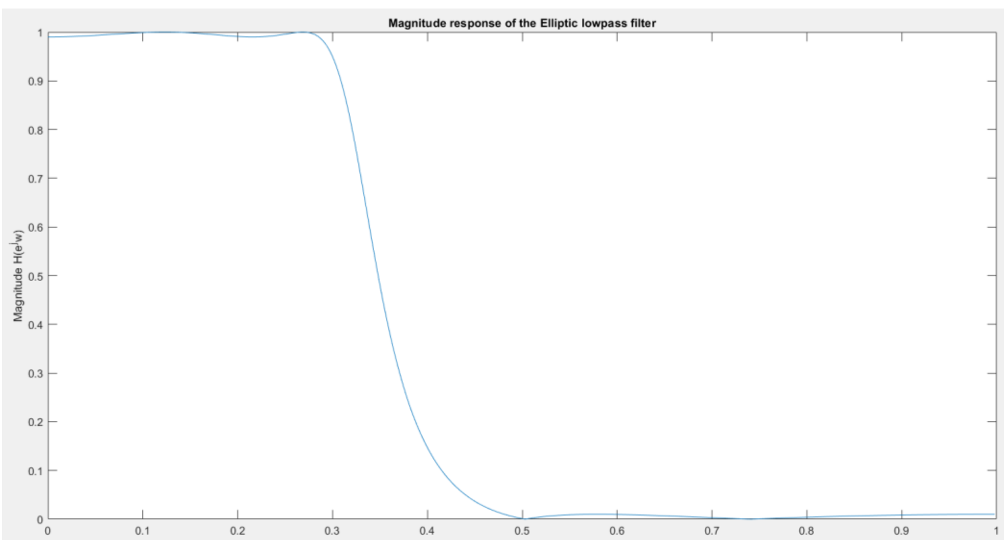

Fig 4. Elliptic lowpass filter magnitude response

Since the magnitude response of the elliptic filter is not monotonous in either passband or stopband, we can't derive the maximum magnitude deviation using the magnitude response figure.

MATLAB output:

Estimated filter order:

4

Maximum deviation for passband:

0.0100

Maximum deviation for stopband:

0.0100

Q5.5)

As the stopband and passband frequencies as well as the desired magnitude in each band were set similarly in each of the designed filters, we can say that a unique specifications were used as a whole. Based on the results of each part, we can summarize the estimated filter order obtained in each filter in the following table:

Table I. Summary of estimated filter orders for previously-designed lowpass filters

Designed lowpass filters	Estimated filter order
FIR filter	13
Butterworth	8
Chebyshev type I	5
Elliptic	4

In the case of the four different lowpass filters I've designed (FIR, Butterworth, Chebyshev type I, and Elliptic), the filter orders decreasing from FIR to Elliptic can be explained as follows:

The order of a filter is related to its complexity. In general, a higher filter order means a more complex filter with more filter coefficients, which can achieve a sharper frequency response and better performance in terms of passband and stopband characteristics. However, different filter design methods have different characteristics and trade-offs, which can result in variations in filter order.

1. **FIR (Finite Impulse Response) Filter:**
    - FIR filters have a fixed, finite number of filter coefficients, and the order directly corresponds to the number of coefficients.
    - The order is determined by the filter design method and the desired filter specifications.
    - Typically, FIR filters require a higher order to meet specific stopband attenuation and passband ripple requirements.
2. **Butterworth Filter:**

- Butterworth filters are designed to have a maximally flat frequency response in the passband.
- The filter order is determined by the desired cutoff frequency and the level of attenuation in the stopband.
- In the case of the question-specific design, it seems that a lower order (8) is sufficient to meet the given specifications.

3. **Chebyshev Type I Filter:**
    - Chebyshev filters have equiripple characteristics in the passband and a faster roll-off in the stopband compared to Butterworth filters.
    - The filter order depends on the level of passband ripple and stopband attenuation specified.
    - For our design, a lower order (5) indicates that the filter meets the requirements with less complexity.
4. **Elliptic Filter:**
    - Elliptic filters are known for having the steepest roll-off in the transition band but may exhibit ripples in both the passband and stopband.
    - The filter order is determined by the desired specifications and the trade-off between passband ripple, stopband attenuation, and filter complexity.
    - In our case, an even lower order (4) suggests that the filter can achieve the specified characteristics with a relatively simple structure.

4.2 Elliptic lowpass filter Design with quantized coefficients

Q5.6)

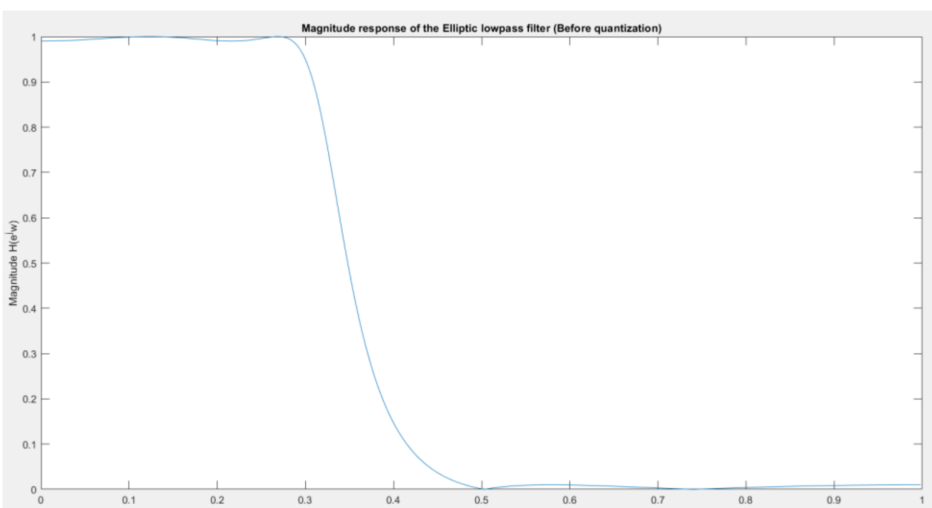

Fig 5. Elliptic lowpass filter magnitude response (Before quantization)

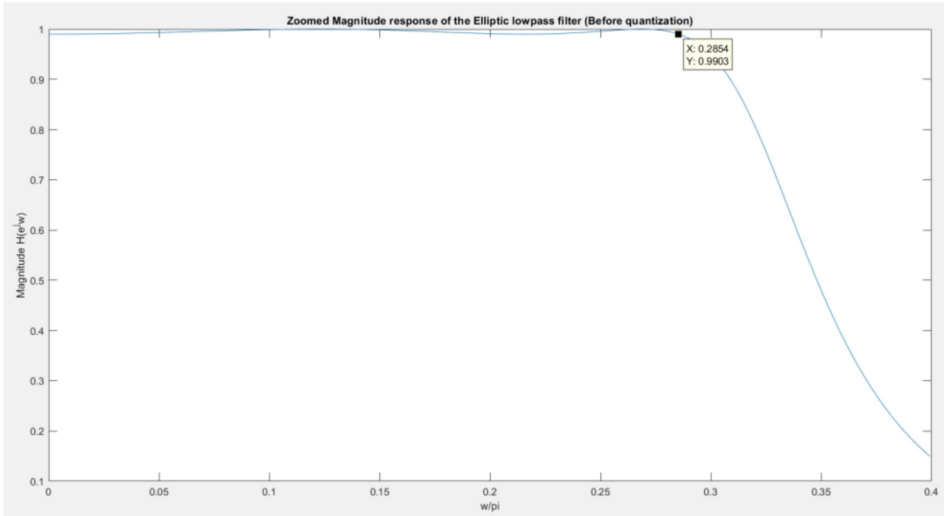

Fig 6. Zoomed Elliptic lowpass filter magnitude response (Before quantization)

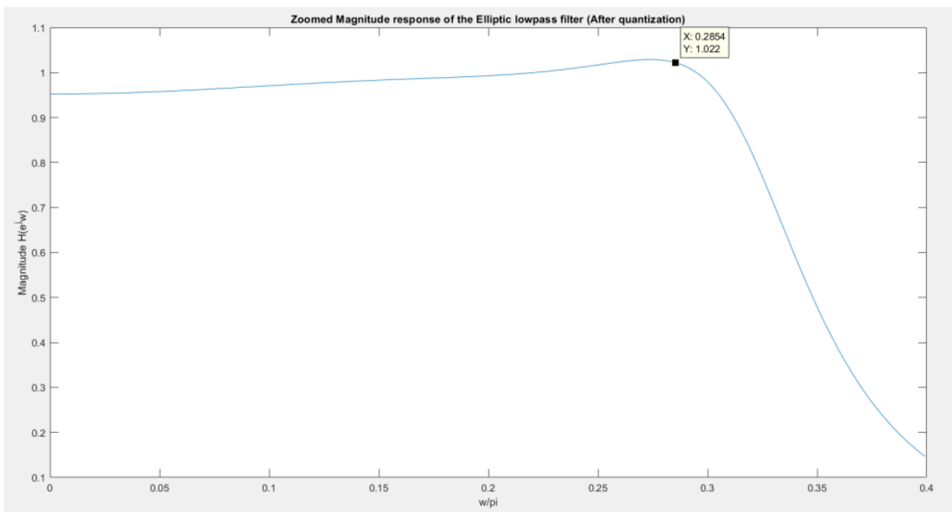

Fig 7. Zoomed Elliptic lowpass filter magnitude response (After quantization – 6 bits)

As can be seen in the two above figures, the magnitude response’s value before quantization in passband frequency is 0.9903 whereas this value changed to 1.022 which indicates that quantization does affect the “specs” at passband frequency, not to mention the stopband frequency. Besides, the deviation from the desired magnitude in the passband after quantization is much more pronounced compared to the deviations that exist in the original magnitude response. This also affects the maximum magnitude deviation that happens near $\omega = 0$ due to the positive slope stretching out from 0 to the normalized bandpass frequency.

Q5.7)

Based on the below figure, the magnitude response after quantizing the coefficients with 10 bits shows much more resemblance to the one before any quantization is made. This makes sense as increasing the quantization's bits reduces the error between the original coefficient and the manipulated one, causing the magnitude response to be more accurate in terms of specifications or whatever. The deviation from the desired magnitude, likewise, becomes negligible and more similar to the original magnitude response.

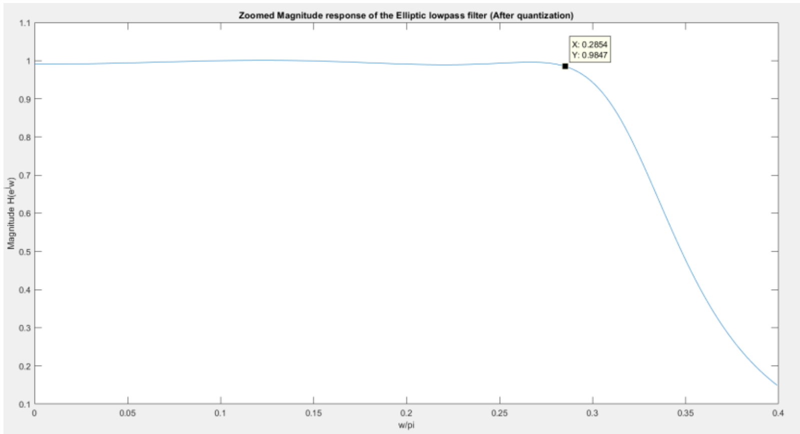

Fig 8. Zoomed Elliptic lowpass filter magnitude response (After quantization – 10 bits)

Q5.8)

First, here is a summary of different magnitude Responses when using different bits for quantization:

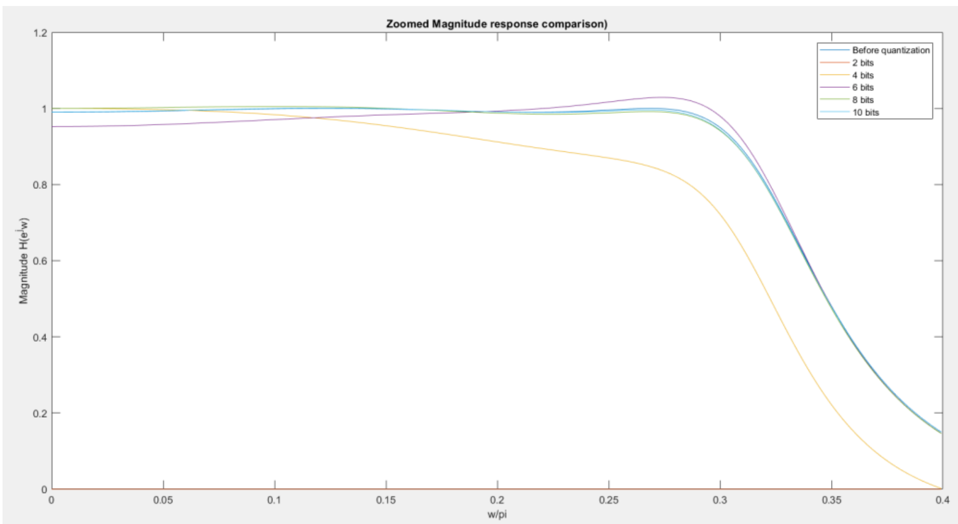

Fig 9. Summary of different magnitude responses with different bits for quantization

As discussed earlier, the bigger the number of bits becomes, the more similar the quantized-magnitude response looks to the original response.

Here is the MATLAB output regarding the maximum radius and the corresponding peak contribution for 5 different quantization bits:

r_max before quantization is:

0.8494

peak contribution before quantization

6.6420

'rq_max after quantization by 2 bits is 8.272709e-01:'

'peak contribution after quantization by 2 bits is 5.789413e+00:'

'rq_max after quantization by 4 bits is 8.607921e-01:'

'peak contribution after quantization by 4 bits is 7.183502e+00:'

'rq_max after quantization by 6 bits is 8.524571e-01:'

'peak contribution after quantization by 6 bits is 6.777688e+00:'

'rq_max after quantization by 8 bits is 8.502358e-01:'

'peak contribution after quantization by 8 bits is 6.677163e+00:'

'rq_max after quantization by 10 bits is 8.484523e-01:'

'peak contribution after quantization by 10 bits is 6.598581e+00:'

To have a better visualization, I depicted a figure, where the ylabel represents the peak contribution and the xlabel represents the number of bits used for quantization. Intuitively, 0 bits represent the peak contribution for the original impulse response (Before quantization). Based on the figure below, when the number of bits is small (e.g. 2 or 4), the peak contribution is way smaller or bigger than the original peak contribution, meaning that the closest pole to the unit circle has an inverse relation with the number of bits used for quantization. That is, the closest pole to the unit circle (the one with the largest radius) is more sensitive to the quantization bits when their values are small and less sensitive when their values become large (e.g. 8 or 10 bits). That being said, the sensitivity of peak contribution to quantization bits is similar to the sensitivity of magnitude response to the quantization bits. The bigger the bits become, the less error can be seen between the quantized- (magnitude response/peak contribution) and the original (magnitude response/peak contribution). As an another conclusion, poles near to unit circle are more sensitive to the quantization effect.

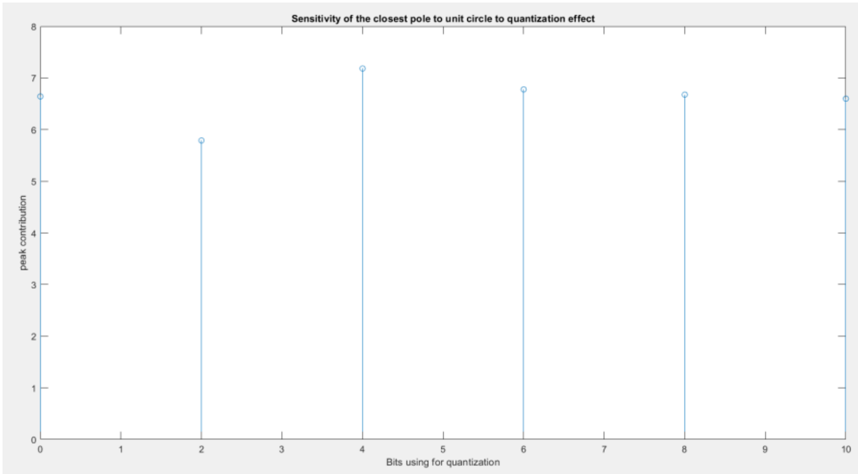

Fig 10. Peak contribution vs. Quantization bits

Written MATLAB code for this part:

```
clc
close all
dp=0.01;
ds=0.01;
NID = 894;
fp=0.375 - NID * 1e-4;
fs=0.475 + NID * 1e-4;
Fsampl=2;
fr_edge = [0 2*fp/Fsampl, 2*fs/Fsampl 1];
des_mag = [1, 0];
[nell_ord, wn] = ellipord(2*fp/Fsampl, 2*fs/Fsampl, -20*log10(1-dp), -20*log10(ds));% Filter order & parameter
wn are computed.
[b_ell,a_ell] = ellip(nell_ord, -20*log10(1-dp), -20*log10(ds), wn);% Filter
[Hell,w]=freqz(b_ell,a_ell,501);% Frequency response is computed.
figure;
plot(w/pi,abs(Hell));
xlabel('w/pi')
ylabel('Magnitude H(e^jw)')
title('Magnitude response of the Elliptic lowpass filter (Before quantization)')
disp('Estimated filter order:')
disp(nell_ord)
figure;
plot(w(1:201)/pi,abs(Hell(1:201)))
xlabel('w/pi')
ylabel('Magnitude H(e^jw)')
title('Zoomed Magnitude response of the Elliptic lowpass filter (Before quantization)')
% new
disp('r_max before quantization is:')
disp(max(abs(roots(a_ell))))
```

```
disp('peak contribution before quantization')
disp(1/(1-max(abs(roots(a_ell)))))
contribution = zeros(1,6);
contribution(1) = 1/(1-max(abs(roots(a_ell))));
c = 0;
legends = cell(size(6));
figure;
plot(w(1:201)/pi,abs(Hell(1:201)))
legends{1} = sprintf('Before quantization');
hold on
for i = [2,4,6,8,10]
    c = c + 1;
    bq_ell=quant(b_ell, 1/(2^i));
    aq_ell=quant(a_ell, 1/(2^i));
    [Hqell,~]=freqz(bq_ell,aq_ell,501);
    plot(w(1:201)/pi,abs(Hqell(1:201)))
    xlabel('w/pi')
    ylabel('Magnitude H(e^jw)')
    title('Zoomed Magnitude response comparison)')
    legends{c+1} = sprintf('%d bits', i);
    %label axes in plot, put title, etc.
    sprintf('rq_max after quantization by %d bits is %d:', i, max(abs(roots(aq_ell))))
    %disp(max(abs(roots(aq_ell))))
    sprintf('peak contribution after quantization by %d bits is %d:', i, 1/(1-max(abs(roots(aq_ell)))))
    contribution(c+1) = 1/(1-max(abs(roots(aq_ell))));
%     disp('peak contribution after quantization')
%     disp(1/(1-max(abs(roots(aq_ell)))))
end
legend(legends)
hold off
figure;
stem([0,2,4,6,8,10],contribution)
xlabel('Bits using for quantization')
ylabel('peak contribution')
title('Sensitivity of the closest pole to unit circle to quantization effect')
```