LAB#04

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

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

1 Design of FIR filters for common app

1.1 Linear-Phase FIR filter design: Low pass filter

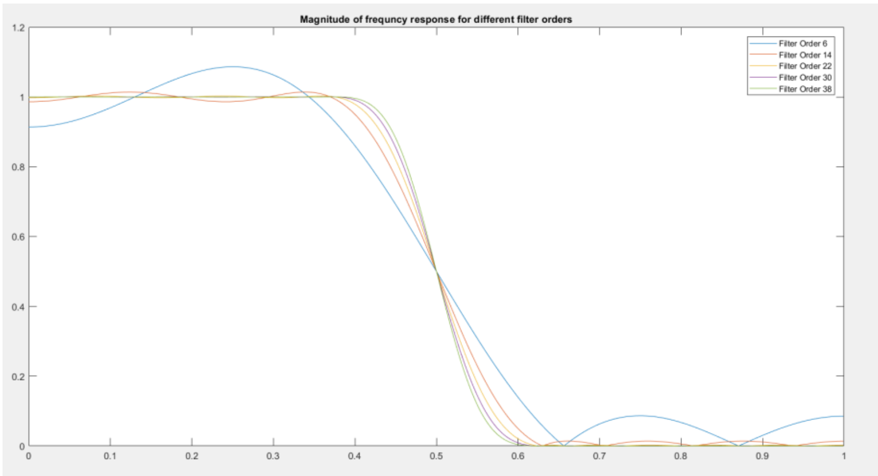

Fig 1. Magnitude of frequency response for given filter orders

Q4.1)

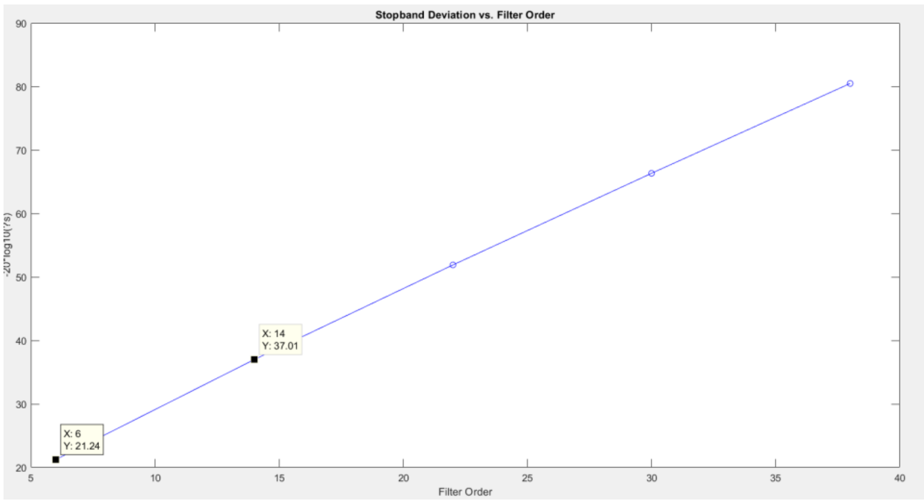

Fig2. $-20\log\Delta_s$ vs. filter order

As can be seen, the relation is linear, meaning that we can derive the filter order using specific Δ_s. First, let's find the straight line's characteristic:

$$say, -20\log\Delta_s = y_s, filter\ order = x_s, (y_s - 37.01) = \frac{37.01 - 21.24}{14 - 6}(x - 14)$$

$$y_s - 37.01 = 1.9712\,(x_s - 14) \rightarrow\ y_s = 1.9712\,x_s + 9.4160$$

Q4.2)

$$if, -20\log\Delta_s = 50 = y_s \rightarrow x_s = \frac{y_s - 9.4160}{1.9712} = 20.5858$$

We could say that the filter estimated order is 21. I also derived the estimated order using **firpmord** function in MATLAB and results are the same:

```
% Estimate filter order for -20*log10(?s) = 50
desired_deviation = 10^(-50/20); % Desired -20*log10(?s) value
estimated_order = firpmord([omega1/pi, omega2/pi], [1, 0], [desired_deviation, desired_deviation])
```

estimated_order =

21

As for the verification, I used the **firpm** command, derived the estimated frequency response using the **freqz** command, and finally, calculated the estimated deviation:

```
% Verify the estimated filter order using firpm
hLP_verified = firpm(estimated_order, [0, omega1 / pi, omega2 / pi, 1], [1, 1, 0, 0]);
[HLP_verified, w] = freqz(hLP_verified, 1, 1024);
estimated_deviation = - 20 * log10(max(abs(HLP_verified(round(omega2 / (pi/1024) )+1 :end))))
```

estimated_deviation =

47.4536

Q4.3)

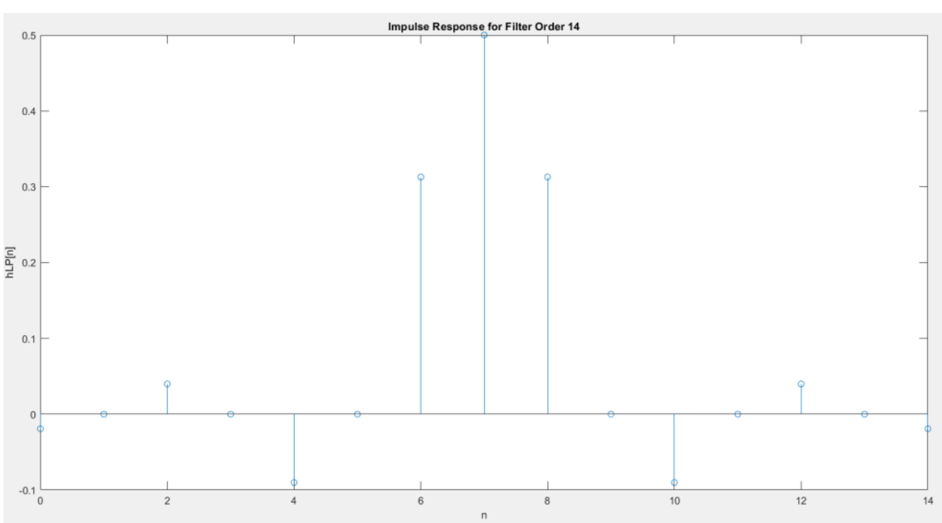

Fig 3. Stem plot of impulse response for filter order of 14

I wrote down a code to check if there exist zero or values smaller than epsilon in the impulse response, and if so, check that whether the existing zeros are symmetric around the central point or not:

```
% Check for zero values in the impulse response
zeros_indices = find(abs(hLP_order14) < 1e-4);
if isempty(zeros_indices)
   disp('No zero values in the impulse response.');
else
   disp('Zero values found in the impulse response.');
end

% Check for symmetry around the middle non-zero sample
middle_index = ceil(length(hLP_order14) / 2);
if all((zeros_indices(1:length(zeros_indices)/2)+ 8) == zeros_indices(length(zeros_indices)/2 + 1:end))
   disp('Zero indexes are symmetric around the middle non-zero sample.');
else
   disp('Zero indexes are symmetric around the middle non-zero sample.');
end
```

The resulting output:

Zero values found in the impulse response.

Zero indexes are symmetric around the middle non-zero sample.

1.2 Linear-Phase FIR filter design: highpass filter

Q4.4)

The stopband magnitude deviation was derived using the following code:

```
stopband_deviation_hp = max(abs(HHP(1:round(omega1 / (pi/1024) ) + 1)));
disp(['Q4.4: Stopband magnitude deviation for highpass filter: ', num2str(stopband_deviation_hp)]);
```

Stopband magnitude deviation for highpass filter: 0.01405

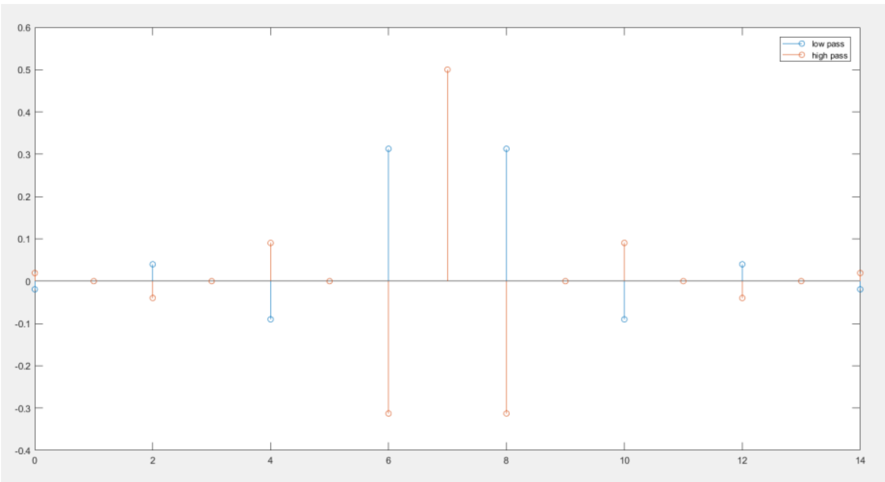

Fig 4. Comparison of impulse responses for low pass and high pass filters

Based on the above figure, for <u>even indexes</u> (indexes in MATLAB start from 1), the impulse responses for both lowpass and highpass filters have similar values, and for <u>odd indexes</u>, they are symmetrical around the x axis.

Q4.5)

As the impulse responses for both of the low pass and high pass filters are symmetrical around the central point, we can derive the frequency response based on the following formula:

$$\textit{impulse duration is odd } (M): H(e^{j\omega}) = e^{-\frac{j\omega(M-1)}{2}}\left(h\left(\frac{M-1}{2}\right) + 2\sum_{n=0}^{\frac{M-3}{2}} h(n)\cos\left(\left(\frac{M-1}{2} - n\right)\omega\right)\right)$$

I start with calculating the frequency response for the lowpass filter, and based on the relation, find the frequency response for the highpass as well:

$$\begin{aligned} HLP\left(e^{j\omega}\right) = e^{-7j\omega} * (&hlp[7] + 2 * hlp[0] * \cos(7\,\omega) + 2 * hlp[1] * \cos(6\,\omega) + 2 * hlp[2] \\ &* \cos(5\,\omega) + 2 * hlp[3] * \cos(4\,\omega) + 2 * hlp[4] * \cos(3\,\omega) + 2 * hlp[5] \\ &* \cos(2\,\omega) + 2 * hlp[6] * \cos(\omega)) \end{aligned}$$

$\rightarrow HHP(e^{j\omega}) = e^{-7j\omega} * (hlp[7] - 2 * hlp[0] * \cos(7\,\omega) + 2 * hlp[1] * \cos(6\,\omega) - 2 * hlp[2] * \cos(5\,\omega) + 2 * hlp[3] * \cos(4\,\omega) - 2 * hlp[4] * \cos(3\,\omega) + 2 * hlp[5] * \cos(2\,\omega) - 2 * hlp[6] * \cos(\omega))$

$\rightarrow HLP(e^{j\omega}) + HHP(e^{j\omega}) = e^{-7j\omega}(2 * hlp[7] + 4 * hlp[1] * \cos(6\,\omega) + 4 * hlp[3] * \cos(4\,\omega) + 4 * hlp[5] * \cos(2\,\omega)) \xRightarrow{hlp[1]=hlp[3]=hlp[5]=0} = \boldsymbol{e^{-7j\omega}}$

$\rightarrow |HLP(e^{j\omega}) + HHP(e^{j\omega})| = 1, \varphi\left(HLP(e^{j\omega}) + HHP(e^{j\omega})\right) = -7\omega$

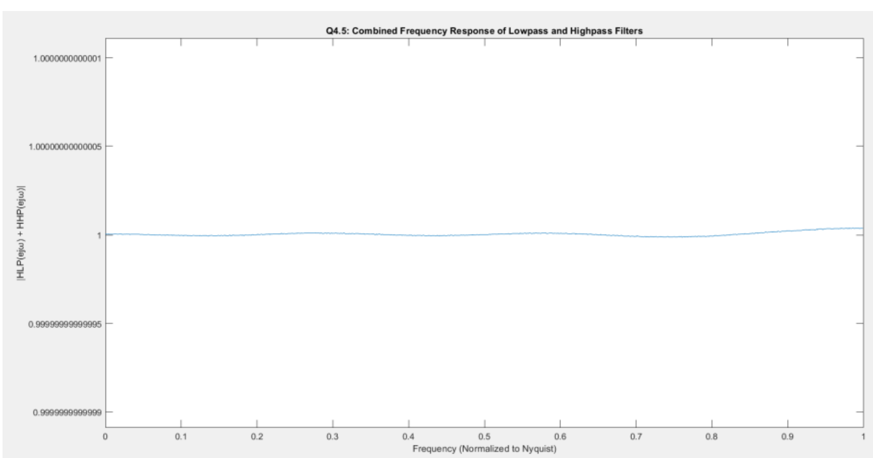

Fig 5. The magnitude of sum of lowpass and highpass's impulse responses

Q4.6)

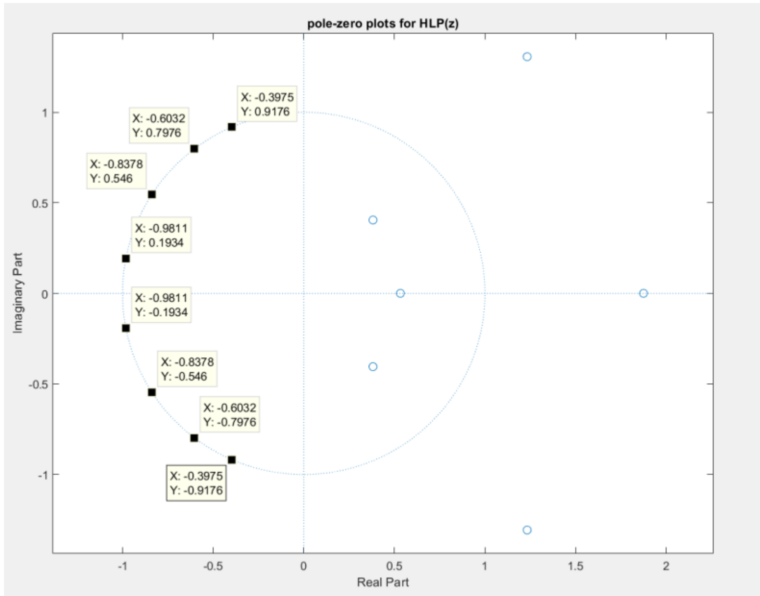

Fig 6. poles-zeros representation for low pass filter

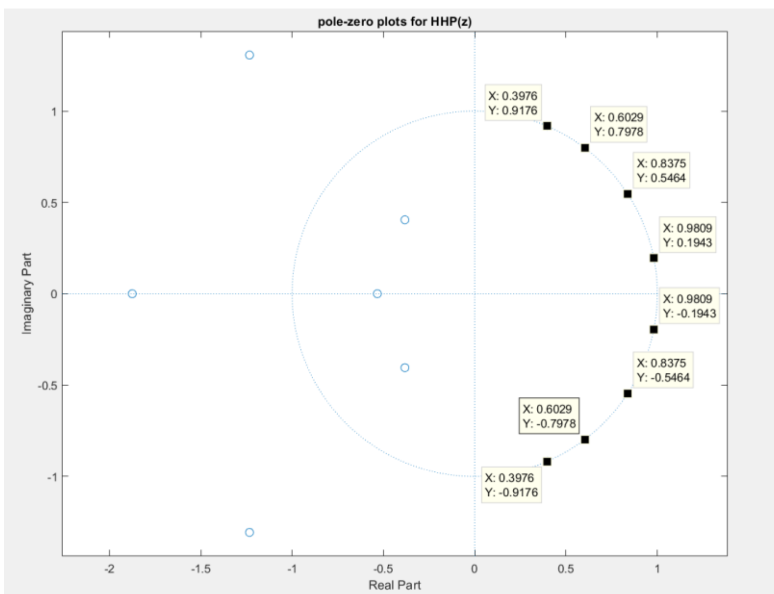

Fig 7. poles-zeros representation for high pass filter

As can be seen in the above figures, the <u>x values</u> for the zeros in lowpass and highpass filters are symmetrical to the y axis, and their <u>y values</u> are identical.

Q4.7)

In the following, I tried to find data values in the frequency response, where their values are below 1e-4. Consequently, two points were found and their corresponding frequencies were returned:

```
zero_frequency_indices = find(abs(HHP) < 1e-4); % Find where HHP is close to zero

% Convert the indices to frequency values
possible_omega0 = zero_frequency_indices * (pi / 1024);
disp('Q4.7: Possible values of ?0 where HHP(z) = 0:');
disp(possible_omega0);
```

Possible values of ω0 where HHP(z) = 0:
0.1994
0.9265

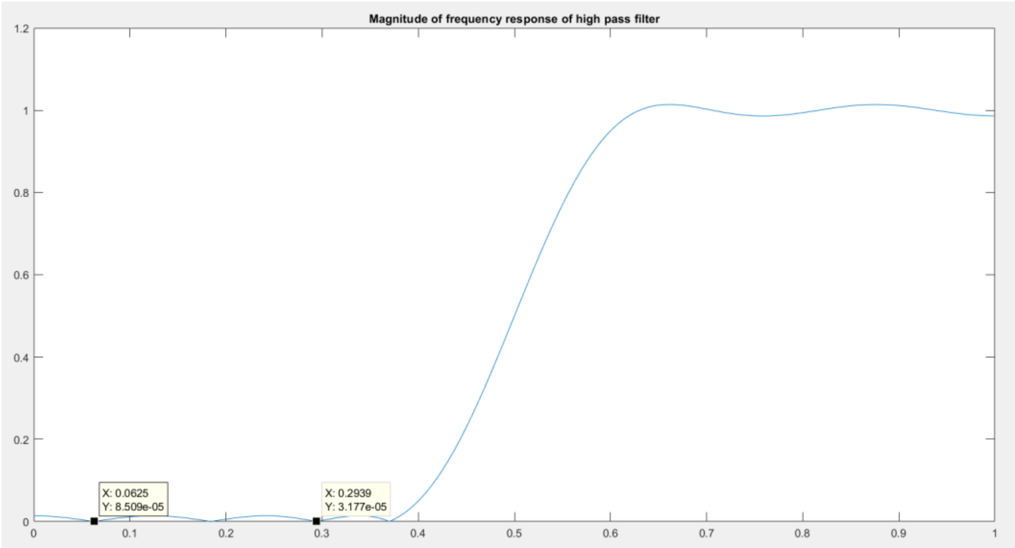

Fig 8. Magnitude response of the high pass filter

2 Design of multiband filters with more general responses

2.1 Linear-Phase FIR filter design: Three-band constant magnitude filter

Q4.8)

Filter Specifications for h3B

Band Edges (Normalized to Nyquist):

0 0.3288 0.4788 0.6788 0.8288 1.0000

Desired Magnitudes:

2 2 0 0 1 1

Deviation Weights:

2 1 3

The first minimum filter order for which the peak absolute magnitude deviation for band 1 is greater than 0.1 will achieve at **14**.

Q4.9)

Peak Absolute Magnitude Deviations:

0.1040

0.2106

0.0693

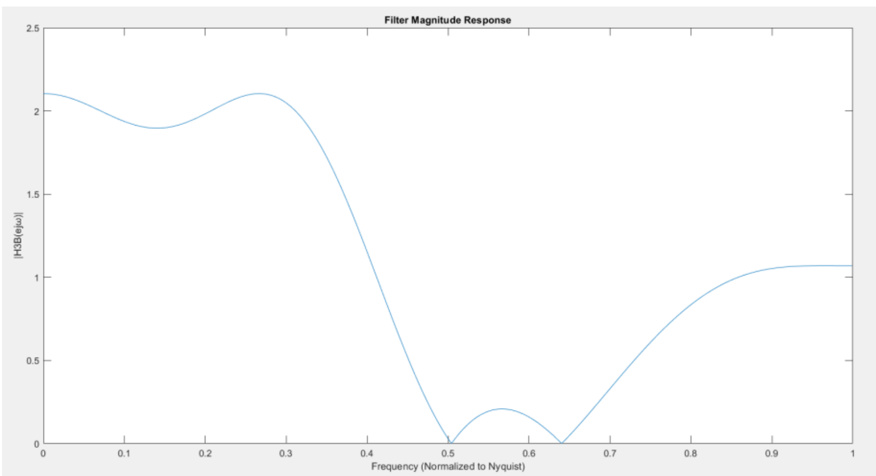

Fig 9. Magnitude response of multiband filter

Q4.10)

Adding the same magnitude to the magnitude response will cause no effect on the peak absolute magnitude deviation. That is, in each band, the magnitude is changed to the same degree, leaving no effect on the peak absolute magnitude deviation. This will only happen if the modified array of magnitudes has the following relation with the original one:

$$Modified = [a_1', a_2', a_3'], Original = [a_1, a_2, a_3], then, Modified = \beta + [a_1, a_2, a_3] : \begin{cases} a_1' = a_1 + \beta \\ a_2' = a_2 + \beta \\ a_3' = a_3 + \beta \end{cases}$$

As this works here, we just need to reset the magnitude array in the firpm command with replacing it with the modified one, the result is as follows:

```
desired_magnitudes_modified = [3.0, 3.0, 1.0, 1.0, 2.0, 2.0];
h3B_modified = firpm(filter_order, band_edges / pi, desired_magnitudes_modified, deviation_weights);
%
% Calculate the frequency response of the modified filter
[H3B_modified, w] = freqz(h3B_modified, 1, 1024);

% Plot the magnitude response of the modified filter
figure;
plot(w/pi, abs(H3B_modified));
xlabel('Frequency (Normalized to Nyquist)');
ylabel('|H3B_modified(ej?)|');
title('Modified Filter Magnitude Response for [3.0, 1.0, 2.0] Magnitudes');
```

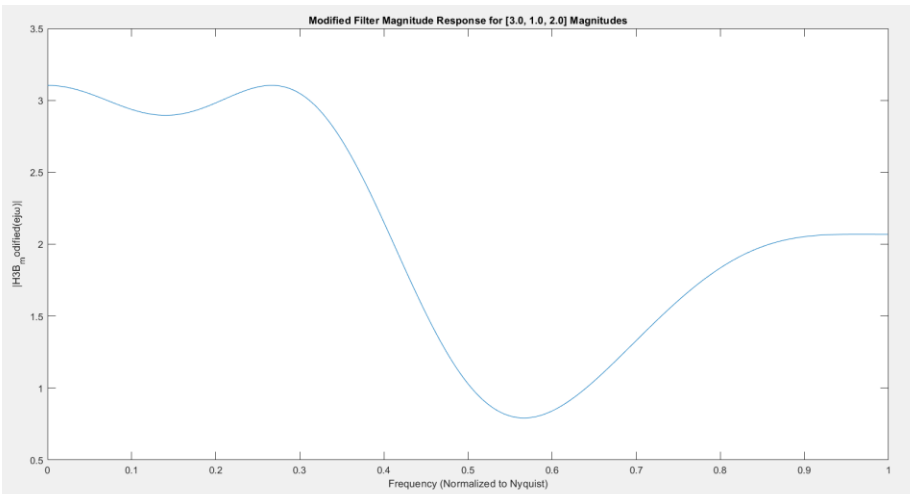

Fig 9. Modified magnitude response of multiband filter

2.2 Linear-Phase FIR filter design: Multiband filter with non-constant desired magnitude

Q4.11)

Filter Specifications for hPL

Band Edges (Normalized to Nyquist):

0 0.3288 0.4788 0.6288 0.8288 1.0000

Desired Magnitudes:

3 1 0 0 1 3

Deviation Weights:

1 2 1

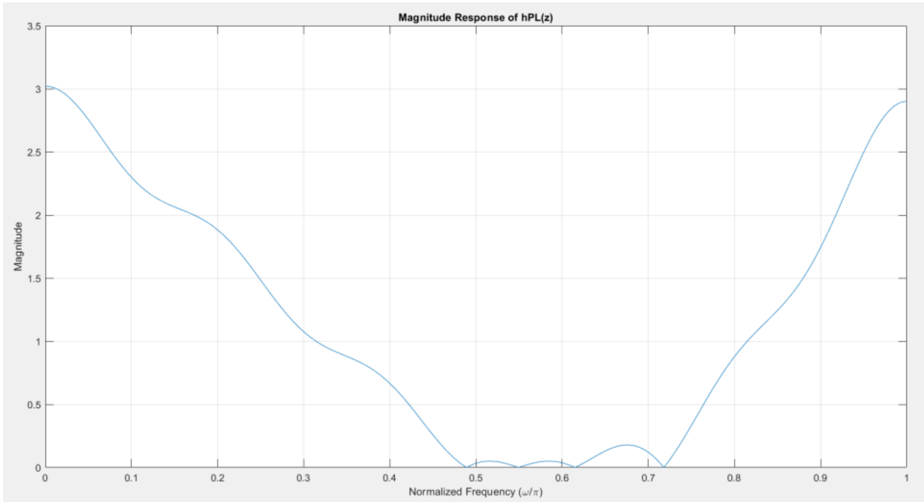

Fig 10. Magnitude response of multiband filter with non-constant desired magnitudes