Design of IIR Digital Filters

Lab Exercises on January 23, 2022

Department of Electronics and Computer Engineering

Pulchowk Campus, Lalitpur

Ashlesh Pandey

PUL074BEX007

Contents

List of Figures

Listings

1 Objectives

- Familiarization with design of IIR digital filters from analog filter.
- Comparison of response for different filter approximations like butterworth, chebyshev I, chebyshev II and elliptic filters.

2 Background Theory

There are several methods that can be used to design digital filters having an infinite duration unit sample response. One of the popular methods is based on converting an analog filter into a digital filter. In this method we begin the design of digital filter in the analog domain and then convert the design into the digital domain. For this purpose, depending on the specifications of the required digital filter the various approximations like butterworth, chebyshev I, chebyshev II and elliptic filters are used.
Among the different approaches used in the design of digital IIR filters this lab experiment deals with impulse invariance method and bi-linear transformation.

2.1 Impulse Invariance Method

In impulse invariance method, the objective is to design an IIR filter having an unit sample response $h[n]$ that is the sampled version of the impulse response of the analog filter.

$$h[n] = h[nT] \qquad n = 0, 1, 2, \ldots\ldots, \text{where } T \text{ is the sampling interval}$$

2.2 Bi-Linear Transformation

In bi-linear transformation a conformal mapping from s-plane to z-plane is carried out with the relation given as,

$$s = \frac{2}{T}\left(\frac{1 - z^{-1}}{1 + z^{-1}}\right)$$

3 Functions Used

Function with syntax	**Description**
`[bz,az]=impinvar(b,a,fs)`	Creates a digital filter with coefficients bz and az, whose impulse response is equal to the impulse response of the analog filter with coefficients b and a, scaled by $1/f_s$, where f_s is the sample rate.
`[numd,dend]=bilinear(num,den,fs)`	Converts the s-domain transfer function specified by coefficients num and den to a discrete equivalent.
`[n,Wc]=buttord(Wp,Ws,Rp,Rs)`	Returns the lowest order n of the butterworth filter and scalar (or vector) of corresponding cutoff frequencies Wc.
`[n,Wp]=cheb1ord(Wp,Ws,Rp,Rs)`	Returns the lowest order n of the chebyshev I filter and scalar (or vector) of corresponding cutoff frequencies Wp.
`[n,Ws]=cheb2ord(Wp,Ws,Rp,Rs)`	Returns the lowest order n of the chebyshev II filter and scalar (or vector) of corresponding cutoff frequencies Ws.
`[n,Wp]=ellipord(Wp,Ws,Rp,Rs)`	Returns the lowest order n of the elliptic filter and scalar (or vector) of corresponding cutoff frequencies Wc.
`[b,a]=butter(n,Wc)`	Returns the transfer function coefficients of an nth-order lowpass digital butterworth filter with normalized cutoff frequency Wc.

`[b,a]=cheby1(n,Rp,Wp)`	Returns the transfer function coefficients of an nth-order lowpass digital chebyshev I filter with normalized pass band edge frequency Wp and Rp decibels of peak-to-peak pass band ripple.
`[b,a]=cheby2(n,Rs,Wp)`	Returns the transfer function coefficients of an nth-order lowpass digital chebyshev II filter with normalized stop band edge frequency Ws and Rs decibels of stop band attenuation down from the peak pass band value.
`[b,a]=ellip(n,Rp,Rs,Wp)`	Returns the transfer function coefficients of an nth-order lowpass digital elliptic filter with normalized pass band edge frequency Wp. The resulting filter has Rp decibels of peak-to-peak pass band ripple and Rs decibels of stop band attenuation down from the peak pass band value.
`impulse(b, a, Tfinal)`	Plots the impulse response of system with coefficients b and a from t = 0 to the final time t = Tfinal.
`dimpulse(bz, az)`	Plots the impulse response of system with coefficients bz and az.

4 Lab Exercises

Problem 1

a. Convert the analog filter $H_a(s) = \dfrac{s+0.1}{(s+0.1)^2+9}$ into a digital IIR filter by means of the impulse invariance method. Plot the frequency response (magnitude) of the designed filter taking sampling interval (T) of 0.1, 0.5 seconds. Compare the response of the filter designed to that of the analog one. Comment on the effect of T on the response.

b. Compare the unit sample response of the designed digital IIR filter with the impulse response of analog filter for T=0.1 and 0.5.

c. Convert the above analog filter in to a digital IIR filter by means of bilinear transformation and repeat all the procedures as specified in Problem 1.a.

```
function [b_digital, a_digital] = transformation_selector(b_analog, a_analog, fs, selected_transformation)
    switch selected_transformation
        case {'impulse invariance method', 'impinvar'}
            [b_digital, a_digital] = impinvar(b_analog, a_analog, fs);
        case {'bilinear transformation','bilinear'}
            [b_digital, a_digital] = bilinear(b_analog, a_analog, fs);
        otherwise
            error('The transformation you want is not found.')
    end
end
```

Listing 1: Matlab function to convert analog to digital IIR filter using user selection and plot impulse response

```
Ts = input('Enter sampling time: ');
selected_transformation = input('Enter transformation method: ', 's');
```

```
b_analog = [1 0.1];
a_analog = [1 0.2 9.01];
fs = 1 / Ts;
[b_digital, a_digital] = transformation_selector(b_analog, a_analog, fs, selected_transformation);
[Ha, Wa] = freqs(b_analog, a_analog, 512);
[Hz, Wz] = freqz(b_digital, a_digital, 512, fs);
figure(1)
l1 = tiledlayout(1, 1);
nexttile
plot(Wa/(2*pi), 20 * log10(abs(Ha)), 'LineWidth', 1.5)
hold on
plot(Wz, 20 * log10(abs(Hz)), 'LineWidth', 1.5)
hold off
xlabel('Frequency (Hz)')
ylabel('Magnitude (dB)')
title(l1, {'Plot for magnitude response comparison when conversion performed', sprintf('using %s for T=%.2f sec', selected_transformation, Ts), '(PUL074BEX007)'})
legend('Analog Filter', 'Digital Filter')
sys_analog = tf(b_analog, a_analog);
figure(2)
l2 = tiledlayout(1, 1);
[y1, t1] = impulse(sys_analog,100);
nexttile
plot(t1, y1, 'LineWidth', 1)
hold on
[y2, t2]=dimpulse(b_digital, a_digital,100);
stairs(y2, 'LineWidth', 1)
hold off
xlabel('Time (seconds)')
ylabel('Amplitude')
title(l2, {'Plot for impulse response comparison when conversion performed', sprintf('using %s for T=%.2f sec', selected_transformation, Ts), '(PUL074BEX007)'})
legend('Analog Filter', 'Digital Filter')
print('-f1', sprintf('../Figures/mag_res_%s_%d', selected_transformation, round(Ts)), '-depsc')
print('-f2', sprintf('../Figures/impulse_res_%s_%d', selected_transformation, round(Ts)), '-depsc')
```

Listing 2: Matlab script to plot magnitude response comparison for analog and digital filter

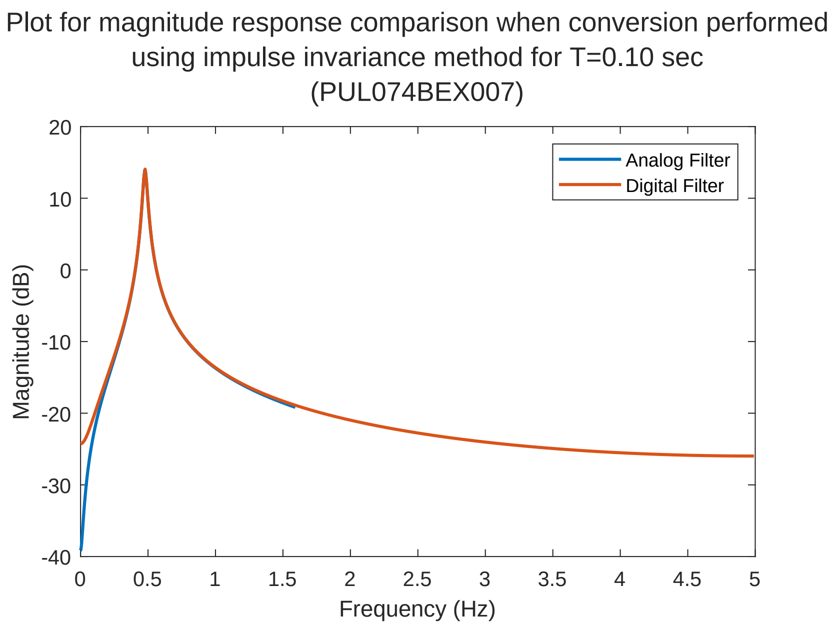

Figure 1: Plot for magnitude response comparison when conversion performed with impinvar and $T = 0.1$ second

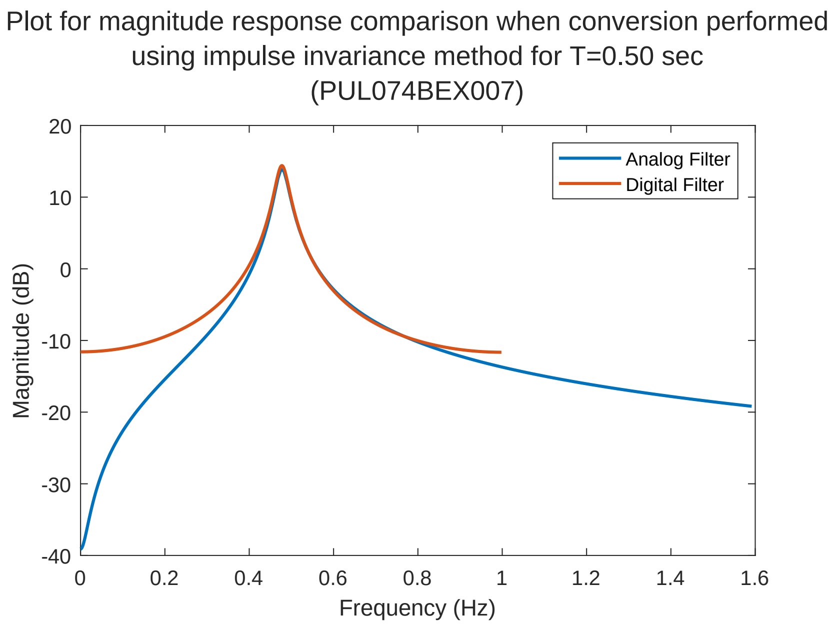

Figure 2: Plot for magnitude response comparison when conversion performed with impinvar and $T = 0.5$ second

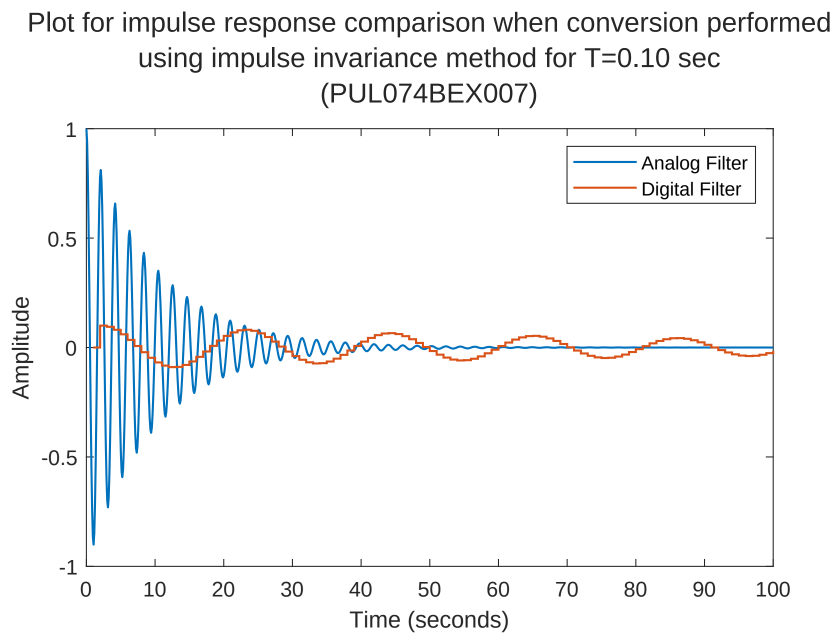

Figure 3: Plot for impulse response comparison when conversion performed with impinvar and $T = 0.1$ second

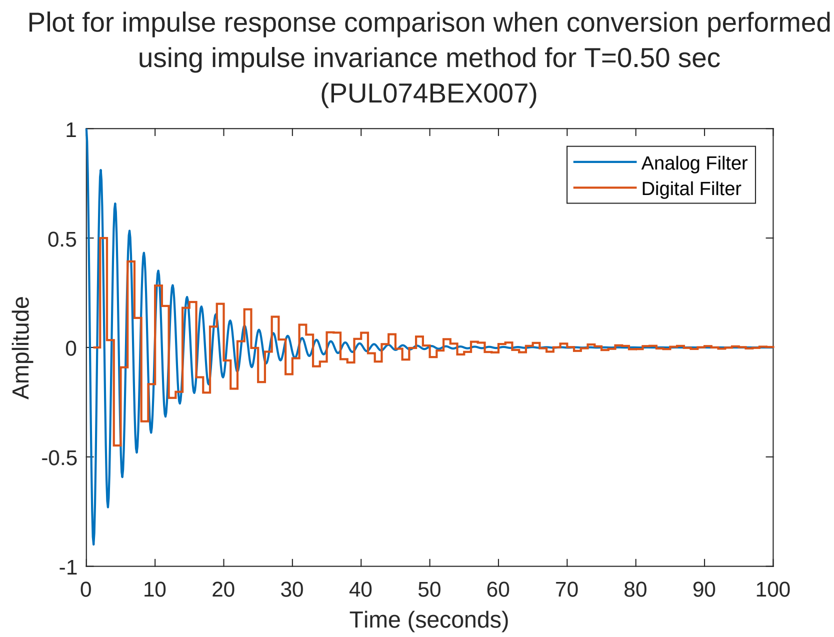

Figure 4: Plot for impulse response comparison when conversion performed with impinvar and $T = 0.5$ second

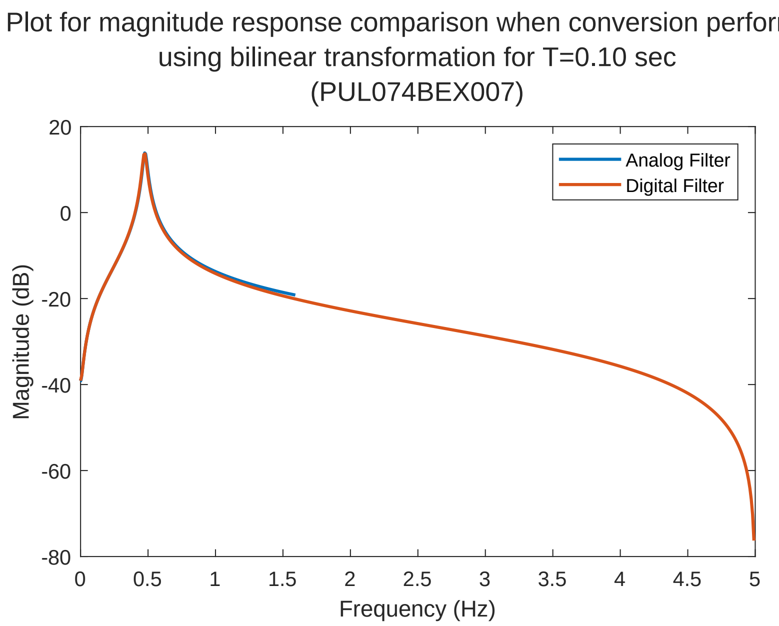

Figure 5: Plot for magnitude response comparison when conversion performed with bilinear and $T = 0.1$ second

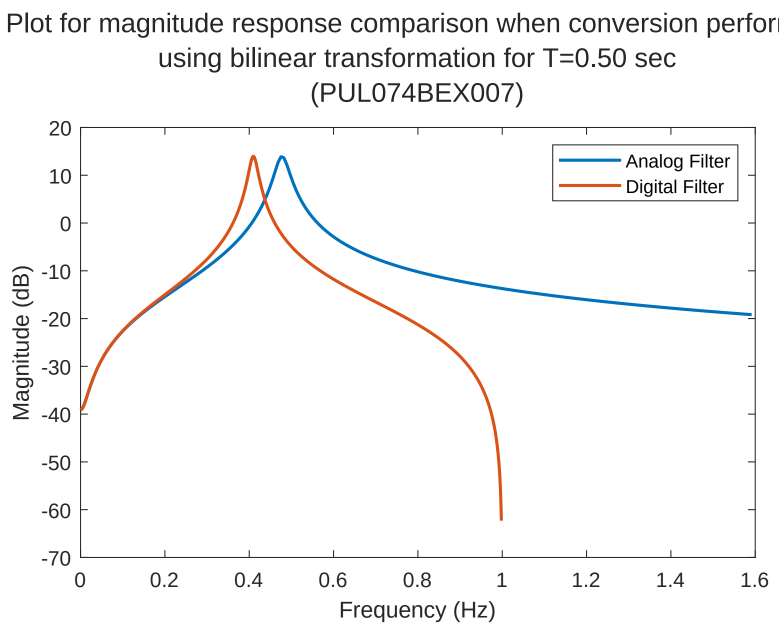

Figure 6: Plot for magnitude response comparison when conversion performed with bilinear and $T = 0.5$ second

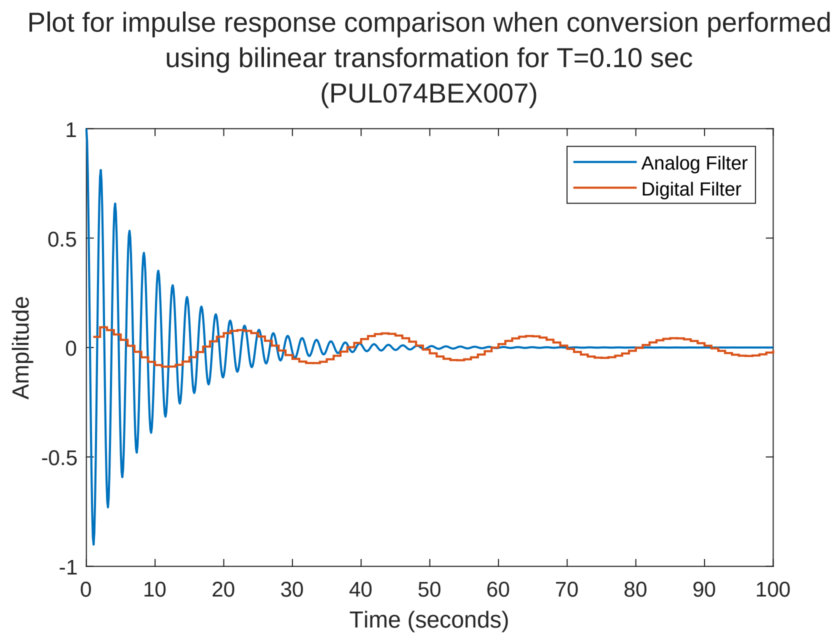

Figure 7: Plot for impulse response comparison when conversion performed with bilinear and $T = 0.1$ second

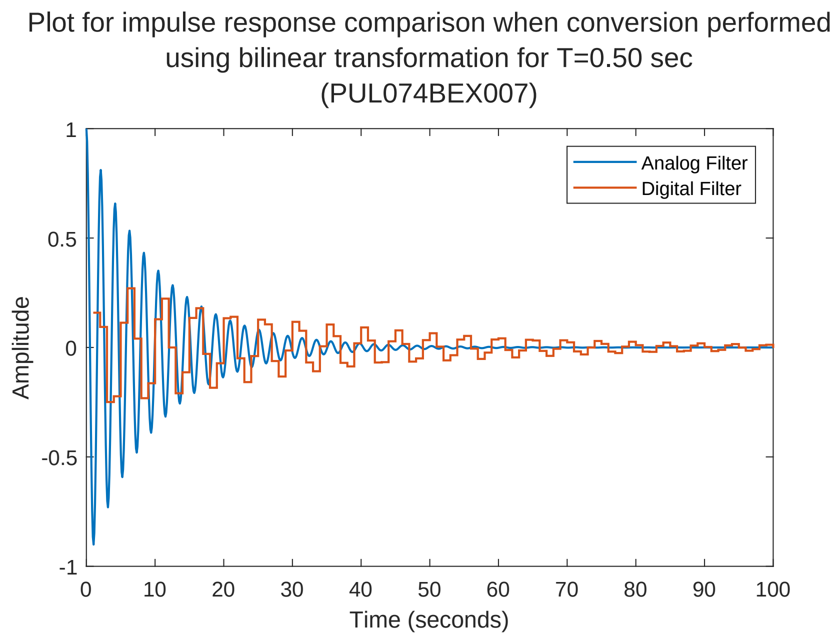

Figure 8: Plot for impulse response comparison when conversion performed with bilinear and $T = 0.5$ second

From the observations in Figure 1 and Figure 2 it is visible that the converted digital filter has somewhat of a similar magnitude response when compared with analog filter. However, the digital filter undergoes spectrum aliasing. This is due to the fact that s-plane to z-plane mapping is many-to-one, i.e., all poles in s-plane between $\left[\frac{(2k-1)\pi}{T}, \frac{(2k+1)\pi}{T}\right]$ where $k = 0, 1, 2, \ldots\ldots$ map into the entire z-plane, which leads to infinite number of poles mapped onto the same location thus producing aliasing effect. Due to this fact the impulse invariance method is not preferred while designing IIR filter other than lowpass nature. When the sampling time (T) is increased it generally results in a frequency response that is more spaces out hence decreasing the chances of aliasing. However, this is not the case with impulse invariance method. Hence the increase in sampling time has no effect on the reduction of aliasing that happens.

Similarly, from the observations in Figure 5 and Figure 6, there is no visible aliasing effect. This is due to the fact that s-plane to z-plane mapping is one-to-one. Due to this fact the bilinear transformation has no restriction on the type of filter that can be transformed. The only known disadvantage of the bilinear transformation is the frequency warping, which is the effect where there exists a non-linear relationship between the continuous time filter frequency and discrete time filter frequency. To compensate for this phenomena, a technique called pre-warping is applied.

Problem 2

An IIR digital low pass filter is required to meet the following specifications:

```
Pass band ripple (or peak to peak ripple):≤ 0.5 dB
Pass band edge: 1.2 kHz
Stop band attenuation:≥ 40 dB
Stop band edge: 2.0 kHz
Sample rate: 8.0 kHz
```

Use the MATLAB Signal Processing Toolbox functions to determine, the required filterorder, the cutoff frequency, the numerator and the denominator coefficients for the

digital Butterworth, digital Chebyshev and digital Elliptic filters. Also plot their frequency responses. Describe the nature of each response.

```matlab
function [N, wc, b_digital, a_digital] = filter_selector(wp, ws, Rp, Rs, selected_filter)
    switch selected_filter
        case {'butterworth', 'butter'}
            [N, wc] = buttord(wp, ws, Rp, Rs);
            [b_digital, a_digital] = butter(N, wc);
        case {'chebyshev I', 'cheby1'}
            [N, wc] = cheb1ord(wp, ws, Rp, Rs);
            [b_digital, a_digital] = cheby1(N, Rp, wc);
        case {'chebyshev II', 'cheby2'}
            [N, wc] = cheb2ord(wp, ws, Rp, Rs);
            [b_digital, a_digital] = cheby2(N, Rs, wc);
        case {'elliptic', 'ellip'}
            [N, wc] = ellipord(wp, ws, Rp, Rs);
            [b_digital, a_digital] = ellip(N, Rp, Rs, wc);
        otherwise
            error('The filter you want is not found.')
    end
end
```

Listing 3: Matlab function to select particular filter type based on user input

```matlab
selected_filter = input('Enter the filter type: ', 's');
Rp = 0.5;
Rs = 40;
fs = 8000;
fn = fs / 2;
wp = 1200 / fn;
ws = 2000 / fn;
[N, wc, b_digital, a_digital] = filter_selector(wp, ws, Rp, Rs, selected_filter);
[H_digital, w_digital] = freqz(b_digital, a_digital);
mag = abs(H_digital);
magdB = 20 * log10(mag);
phase = angle(H_digital) * 180 / pi;
fprintf('Order of %s filter=%d\n', selected_filter,N)
disp('Numerator coeff=')
disp(b_digital)
disp('Denominator coeff=')
disp(a_digital)
l = tiledlayout(2, 2);
```

```matlab
nexttile([1 1])
plot(w_digital / pi, magdB, 'LineWidth', 1.5)
grid on
xline(wc, 'g--', sprintf('\\omega_c=%.2f', wc), 'LineWidth', 1, 'LabelVerticalAlignment', 'bottom', 'LabelHorizontalAlignment', 'left')
yline(max(magdB) - 3, 'r--', sprintf('K_c=%.2f', max(magdB) - 3), 'LineWidth', 1, 'LabelVerticalAlignment', 'bottom', 'LabelHorizontalAlignment', 'right')
ylabel('Magnitude (dB)')
title('Magnitude plot', 'FontWeight', 'normal')
nexttile([2 1])
plot(w_digital / pi, mag, 'LineWidth', 1.5)
grid on
ylabel('Gain')
title('Linearized form', 'FontWeight', 'normal')
nexttile([1 1])
plot(w_digital / pi, phase, 'LineWidth', 1.5)
grid on
ylabel('Phase (degree)')
title('Phase plot', 'FontWeight', 'normal')
xlabel(l, 'Normalized Frequency (\times \pi rad/sample)')
title(l, sprintf('Frequency response of %s filter\n(PUL074BEX007)', selected_filter));
print(sprintf('../Figures/%s', selected_filter), '-depsc')
```

Listing 4: Matlab script to plot frequency response, display cutoff frequency and order of selected filter

```
>> lab_5_b
Enter the filter type: butterworth
Order of butterworth filter=9
Numerator coeff=
    0.0004    0.0032    0.0129    0.0302    0.0453
    0.0453    0.0302    0.0129    0.0032    0.0004

Denominator coeff=
    1.0000   -2.7996    4.4582   -4.5412    3.2404
   -1.6330    0.5780   -0.1370    0.0197   -0.0013
```

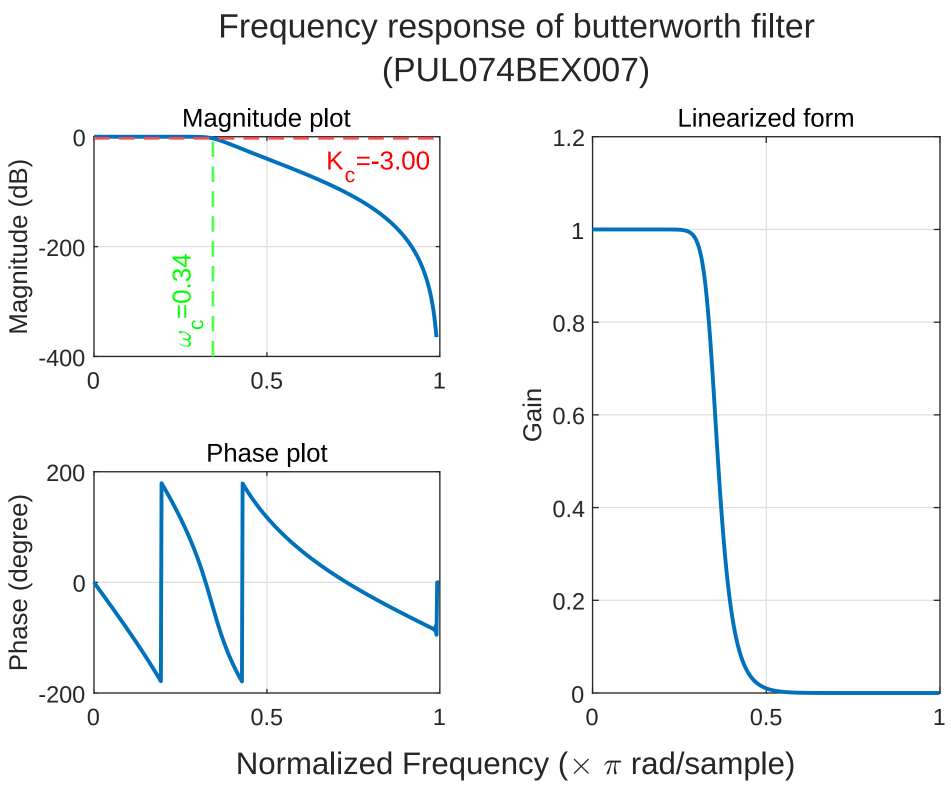

Figure 9: Plot for frequency response of butterworth filter

From the observations in Figure 9, the nature of the filter is lowpass. There is no ripple in the passband or the stopband of the filter, i.e. it is maximally flat. The cut-off frequency is

0.34 ($\times\pi$ rad/sample), which coincides with the plot at -3 dB. The order of the filter is 9.

```
>> lab_5_b
Enter the filter type: chebyshev I
Order of chebyshev I filter=5
Numerator coeff=
    0.0026    0.0132    0.0264    0.0264    0.0132    0.0026

Denominator coeff=
    1.0000   -2.9775    4.2932   -3.5124    1.6145   -0.3334
```

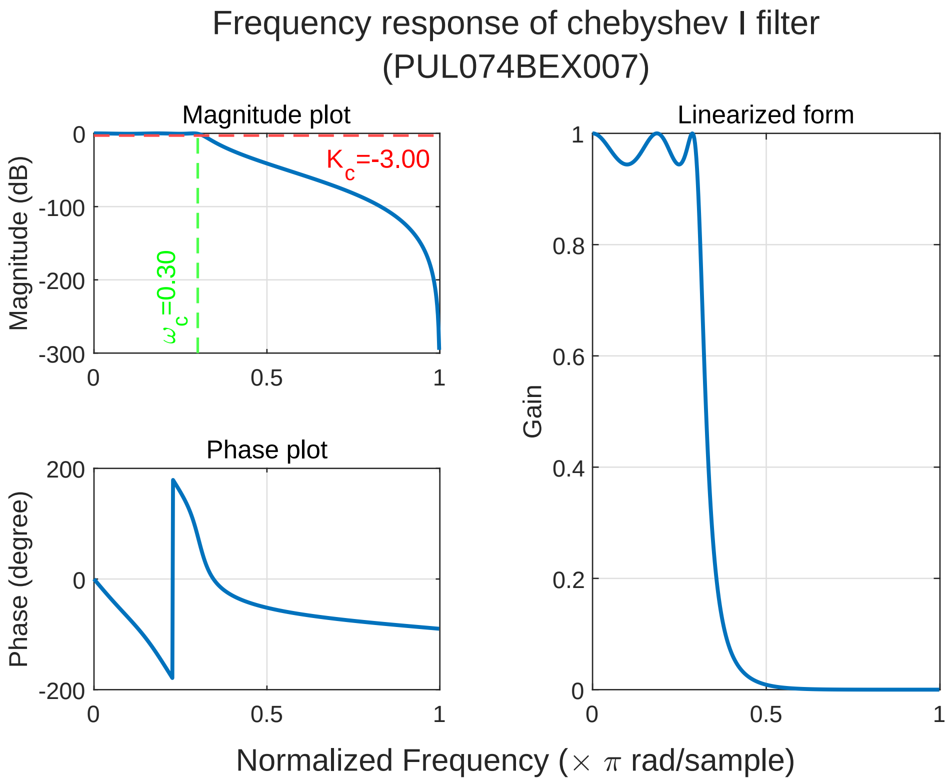

Figure 10: Plot for frequency response of chebyshev I filter

From the observations in Figure 10, the nature of the filter is lowpass. There is visible ripple in the passband but no ripple is seen in the stopband of the filter. The passband ripple is ≤ 0.5

dB. The cut-off frequency is 0.30 ($\times\pi$ rad/sample), which is actually the passband edge in normalized form. The order of the filter is 5.

```
>> lab_5_b
Enter the filter type: chebyshev II
Order of chebyshev II filter=5
Numerator coeff=
    0.0524    0.1086    0.1661    0.1661    0.1086    0.0524


Denominator coeff=
    1.0000   -1.2056    1.2136   -0.4984    0.1583   -0.0136
```

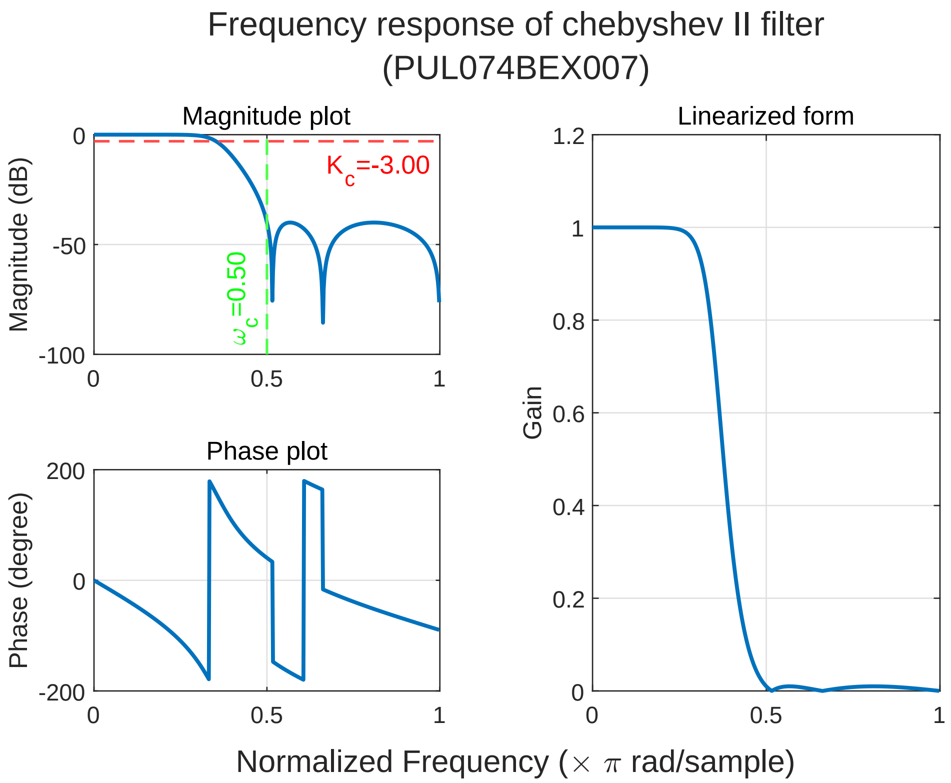

Figure 11: Plot for frequency response of chebyshev II filter

From the observations in Figure 11, the nature of the filter is lowpass. There is visible ripple in the stopband but no ripple is seen in the passband of the filter. The stopband attenuation is ≥ 40 dB. The cut-off frequency is 0.50 ($\times\pi$ rad/sample), which is actually the stopband edge in normalized form. The order of the filter is 5.

```
>> lab_5_b
Enter the filter type: elliptic
Order of elliptic filter=4
Numerator coeff=
    0.0389    0.0363    0.0665    0.0363    0.0389


Denominator coeff=
    1.0000   -2.1444    2.3658   -1.3250    0.3332
```

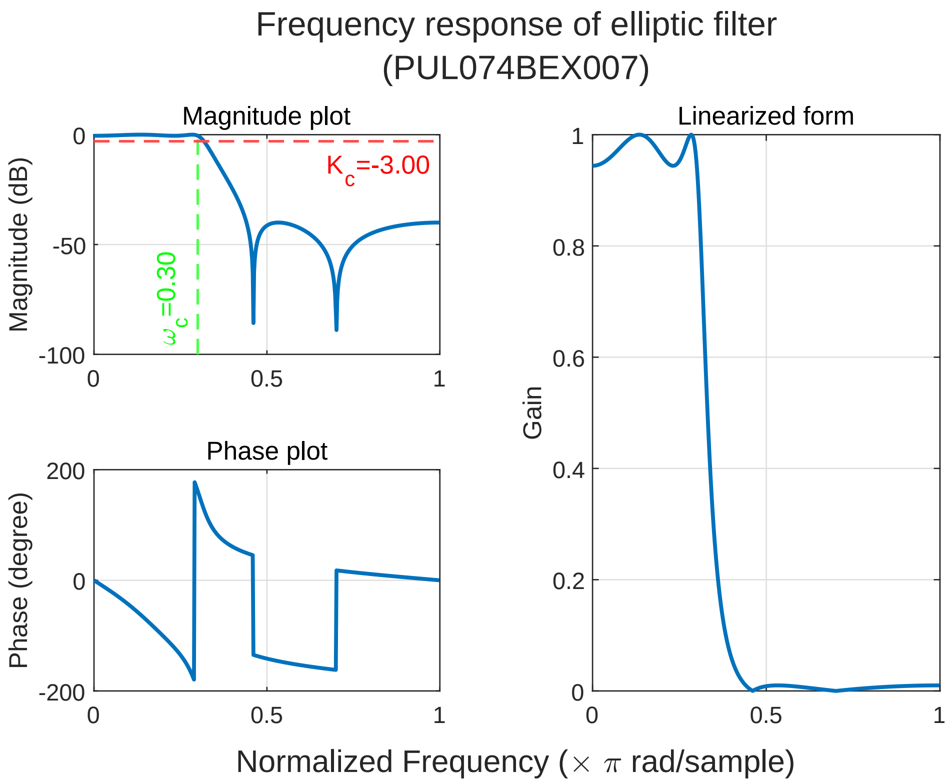

Figure 12: Plot for frequency response of elliptic filter

From the observations in Figure 12, the nature of the filter is lowpass. There is visible ripple in both the passband and the stopband of the filter. The passband ripple is ≤ 0.5 dB and the stopband attenuation is ≥ 40 dB. The cut-off frequency is 0.30 ($\times\pi$ rad/sample), which is actually the passband edge in normalized form. The order of the filter is 4.

Filter Type	**Passband**	**Stopband**	**Order**
Butterworth	Flat	Flat	9
Chebyshev I	Equiripple	Flat	5
Chebyshev II	Flat	Equiripple	5
Elliptic	Equiripple	Equiripple	4

From the summarized comparison, it is clear that there is a tradeoff between monotonic response and the order of the filter. For the maximally flat butterworth filter, the order is 9, which on contrast is 4 for the elliptic filter which has equiripple in both the passband and the stopband.

5 Discussion and Conclusion

In this lab experiment we dealt with the design of IIR filters. Firstly, conversion of analog filters using two methods, viz. impulse invariance method and bilinear transformation was performed. During this, the aliasing effect was noted for impulse invariance method which is why it should not be preferred for designing IIR filter other than lowpass nature. However, for bilinear transformation such effect wasn't seen, so it has no restriction on the type of filter that can be designed. Similarly, the other problem dealt with designing and comparing filter approximation techniques for given IIR digital lowpass specifications. The comparison showed that there is a tradeoff between the monotonic nature and order of filter.
Hence, the objectives of the lab experiment were fulfilled.