INSTITUTE OF ENGINEERING

ADVANCED COLLEGE OF ENGINEERING AND MANAGEMENT

KALANKI, KATHMANDU

(AFFILIATED TO TRIBHUVAN UNIVERSITY)

LAB REPORT

SUBJECT: DIGITAL SIGNAL ANALYSIS AND PROCESSING (DSAP)

LAB NO: 04 and 05

SUBMITTED BY:

NAME: DIPESH DHUNGANA

ROLL NO: ACE077BCT035

DATE: July 11, 2024

SUBMITTED TO:

DEPARTMENT OF COMPUTER AND ELECTRONICS

TITLE: IIR (INFINITE IMPULSE RESPONSE) FILTER AND FIR (FINITE IMPULSE RESPONSE) FILTER.

OBJECTIVE:

- To design IIR (Infinite Impulse Response) filter.
- To design FIR (Finite Impulse Response) filter.

THEORY:

IIR (Infinite Impulse Response) filter

An Infinite Impulse Response (IIR) filter is a type of digital filter characterized by its recursive nature, where past output values are fed back into the filter along with current and past input values, resulting in an impulse response that theoretically lasts indefinitely. The transfer function of an IIR filter is the ratio of two polynomials, with the numerator representing the feedforward part and the denominator representing the feedback part. IIR filters are efficient in terms of computational resources, achieving sharper magnitude responses than Finite Impulse Response (FIR) filters with the same number of coefficients. However, they typically exhibit nonlinear phase characteristics, which can introduce phase distortion. Stability is a crucial aspect, requiring all poles to lie within the unit circle in the z-plane. Common design methods include Butterworth, Chebyshev, and Elliptic filters, each offering different trade-offs in terms of flatness and sharpness of response. IIR filters are widely used in applications like audio processing, data smoothing, and noise reduction, where sharp cutoff frequencies are needed and phase linearity is less critical.

Difference Equation: The general form of the difference equation for an IIR filter is:

$$y[n] = \sum_{k=0}^{M} b_k x[n-k] - \sum_{j=1}^{N} a_j y[n-j]$$

where,

y[n] is the current output
x[n] is the current input
b_k and a_k are the filter coefficients
M and N are the filter orders

Application of IIR filters

IIR filters are used in various applications, such as:

- Audio Processing: For equalization, noise reduction, and sound enhancement.
- Communications: For channel equalization, modulation, and demodulation.
- Control Systems: For system stability and response shaping.

Filter Designs

- Butterworth filters are known for their flat frequency response in the passband.
- Chebyshev filters have a steeper roll-off than Butterworth filters, with ripples in either the passband (Chebyshev Type I) or the stopband (Chebyshev Type II).
- Elliptic filters offer the steepest roll-off for a given filter order but have ripples in both the passband and the stopband.
- The stability of an IIR filter is determined by the location of its poles. An IIR filter is stable if all poles lie inside the unit circle in the z-plane.

FIR (Finite Impulse Response)

A Finite Impulse Response (FIR) filter is a type of digital filter characterized by its finite duration impulse response, which means that the filter's response to an impulse input will settle to zero in a finite number of steps. FIR filters are non-recursive, using only the current and past input values to produce the output, and do not rely on feedback from past outputs. The transfer function of an FIR filter is a polynomial in (z)^-1, with coefficients representing the filter taps. FIR filters are inherently stable because they do not use feedback, and they can be designed to have a linear phase response, which preserves the waveform shape of filtered signals and avoids phase distortion. Although FIR filters typically require more coefficients than IIR filters to achieve the same magnitude response, they are easier to design and implement, especially when a linear phase is required. Common design methods for FIR filters include the window method, the frequency sampling method, and the optimal equiripple method. FIR filters are widely used in applications where phase linearity is critical, such as in data communications, image processing, and audio signal processing.

LAB TASKS:

1. H(s)=(s+0.1)/((s+0.1) 2 +9)

   Impulse invariance

   i. Convert the analog filter into a digital IIR filter by means of impulse invariance method.
   ii. Plot the frequency response in s-domain and z-domain.
   iii. Plot the impulse response of the LTI model.
   iv. Plot the impulse response in z-domain.

2. Bilinear Transformation

   i. Convert the analog filter into a digital IIR filter by means of bilinear transformation method.
   ii. Plot the frequency response of the transformed filter by bilinear transformation
   iii. Plot the impulse response in z-domain.
   iv. Plot the step response of the LTI model.
   v. Step response of discrete time linear systems.

3. An IIR digital low pass filter is required to meet the following specifications: Pass band attenuation 4≤ dB, Stop band attenuation≤30 dB, Passband edge=400 Hz, Stopband edge=800Hz and Sample rate=2 KHz.

   a. Find its order.
   b. Plot its frequency response.
   c. Plot its poles and zeros.
   d. Plot its impulse response.
   e. Convert this filter into digital IIR filter by means of impulse invariance method.
   f. Plot the frequency response in z-domain.
   g. Plot the impulse response in z-domain
   h. Step response of the discrete time linear system

   (Butterworth filter)

4. An IRR digital lowpass filter required to meet following specifications:

   Passband attenuation <= 1dB, Stopband attenuation <= 15dB, Passband edge frequency = 0.2 π rad/sec, Stopband edge frequency = 0.3 π rad/sec.

   a) Find its order
   b) Plot frequency response

c) Plot its poles and zeros

d) Plot impulse response

e) Convert to IIR by impulse variance method

f) Plot frequency response in z-domain

g) Plot impulse response in z-domain

h) Step response of discrete time linear system

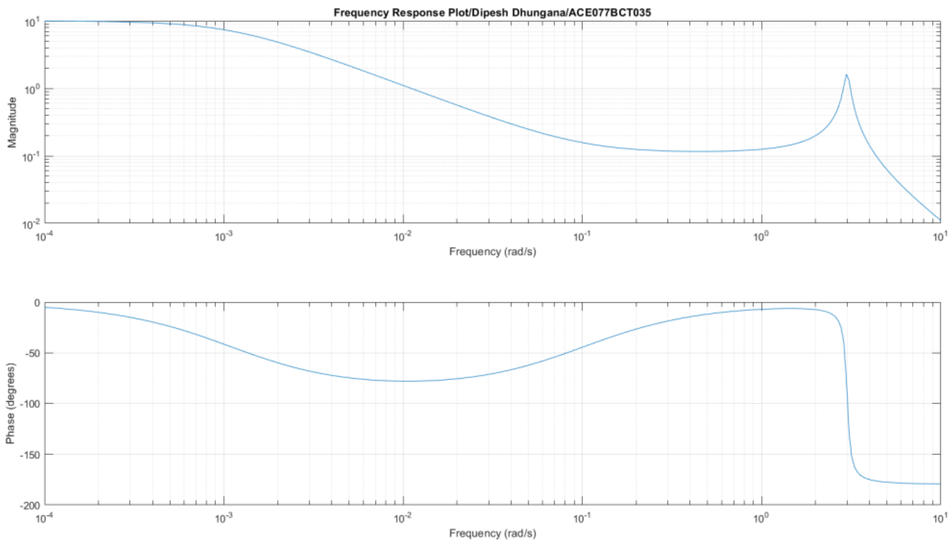
(Chebyshev I and II)

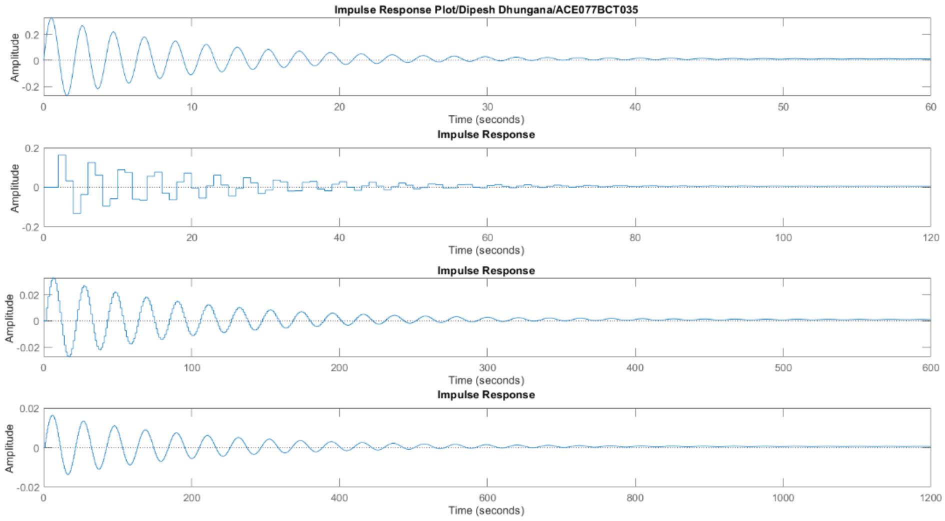
(Source code 1)

```
clc;

clear all;
close all;
b= [1 .1];
a= [1 .2 9 .01];
figure; freqs(b,a);
title('Frequency Response Plot/Dipesh Dhungana/ACE077BCT035');
figure;
subplot(4,1,1);
impulse(b,a);
title('Impulse Response Plot/Dipesh Dhungana/ACE077BCT035');
[bz,az]= impinvar(b,a,2);
subplot(4,1,2);
dimpulse(bz,az);
[bz,az]= impinvar(b,a,10);
subplot(4,1,3);
dimpulse(bz,az);
[bz,az]= impinvar(b,a,20);
subplot(4,1,4);
dimpulse(bz,az);
```

(Outputs)

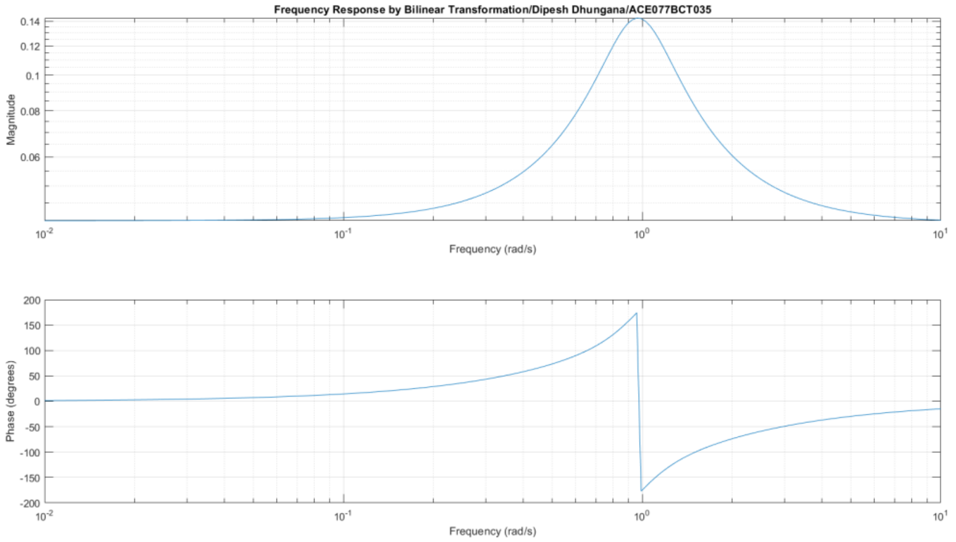


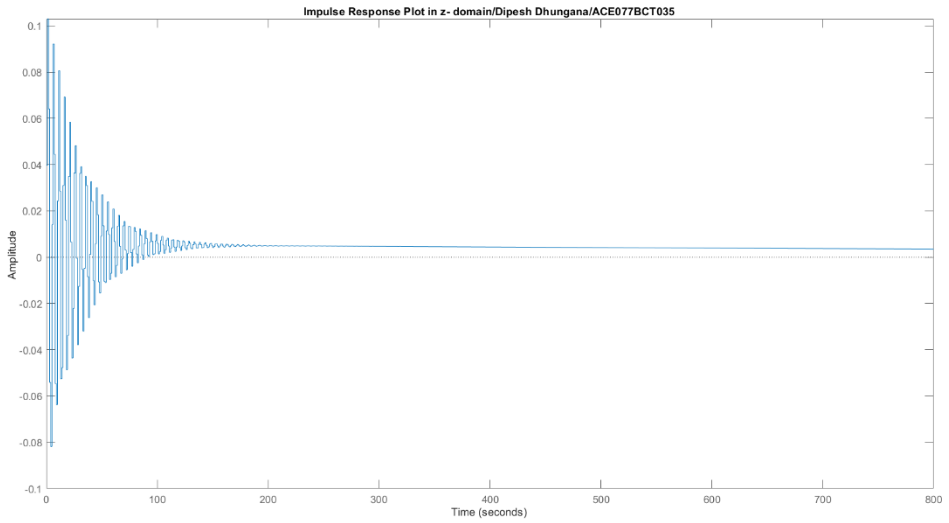


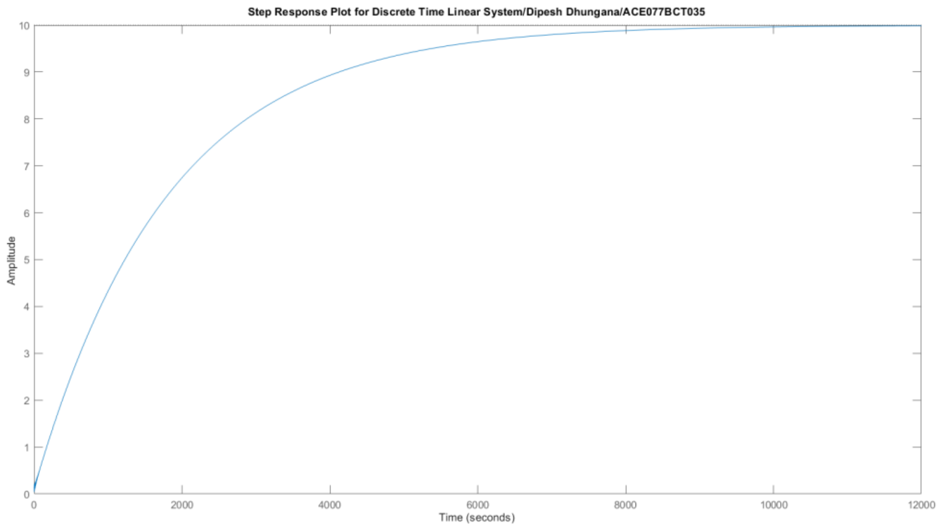
(Source code 2)

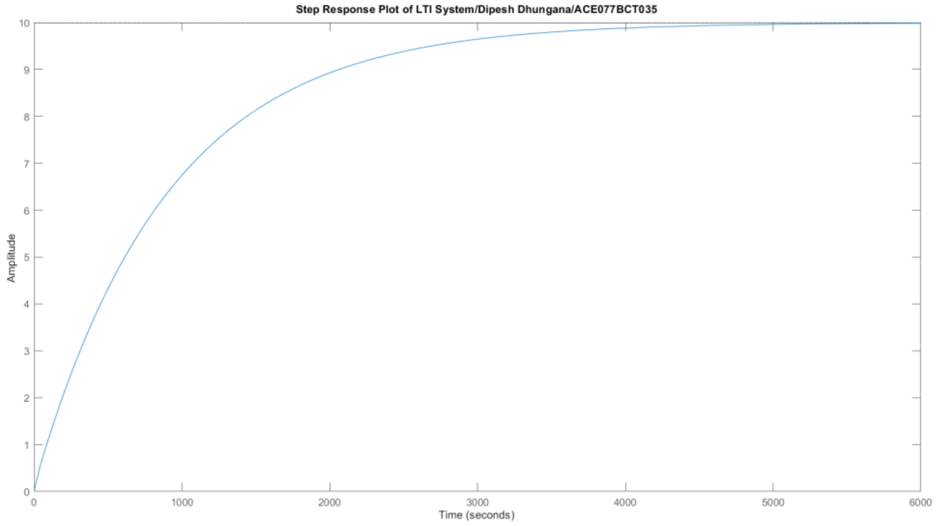
```
clc;
clear all;
close all;
b= [1 .1];
a= [1 .2 9 .01];
[bz,az]= bilinear(b,a,2);
freqs(bz,az);
title('Frequency Response by Bilinear Transformation/Dipesh
Dhungana/ACE077BCT035');
figure;
dimpulse(bz,az);
title('Impulse Response Plot in z- domain/Dipesh Dhungana/ACE077BCT035');
figure;
sys=tf(b,a);
step(sys);
title('Step Response Plot of LTI System/Dipesh Dhungana/ACE077BCT035');
figure;
dstep(bz,az);
title('Step Response Plot for Discrete Time Linear System/Dipesh
Dhungana/ACE077BCT035');
```

(Outputs)


Frequency Response by Bilinear Transformation/Dipesh Dhungana/ACE077BCT035
Magnitude
Frequency (rad/s)
Phase (degrees)
Frequency (rad/s)


Impulse Response Plot in z- domain/Dipesh Dhungana/ACE077BCT035
Amplitude
Time (seconds)

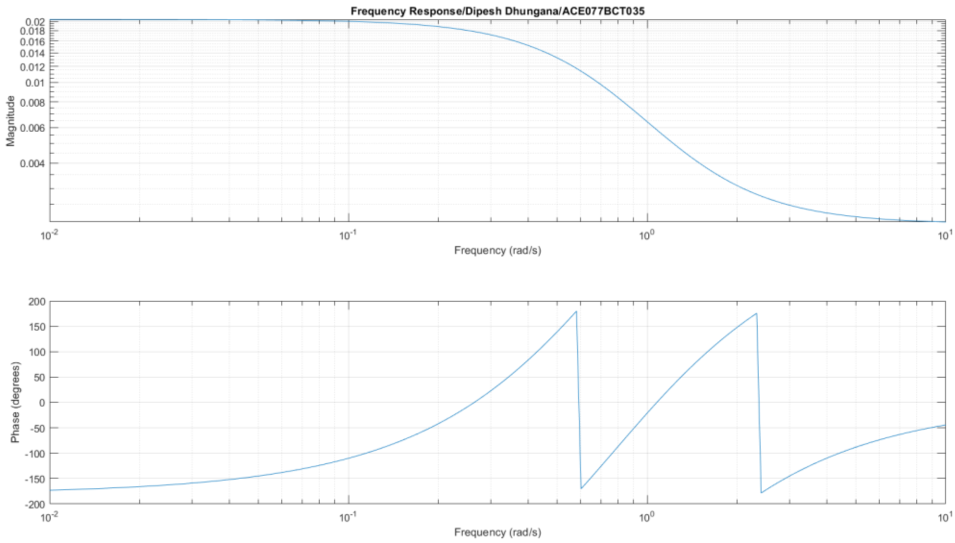

Step Response Plot for Discrete Time Linear System/Dipesh Dhungana/ACE077BCT035
Amplitude
Time (seconds)

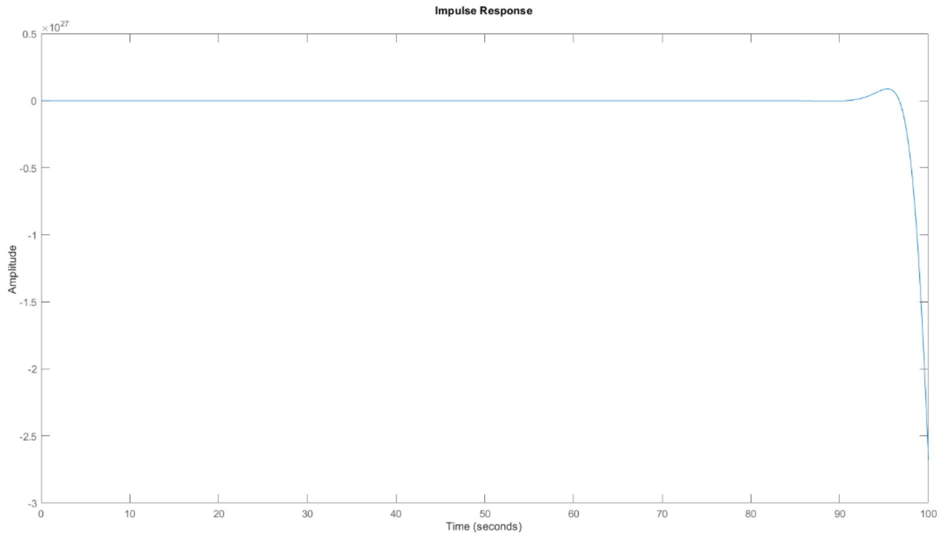


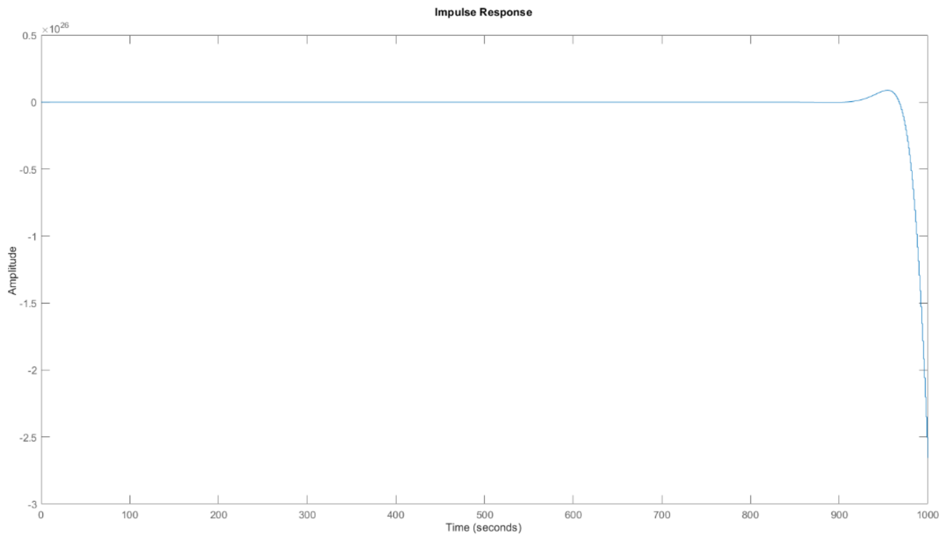
(Source code 3)

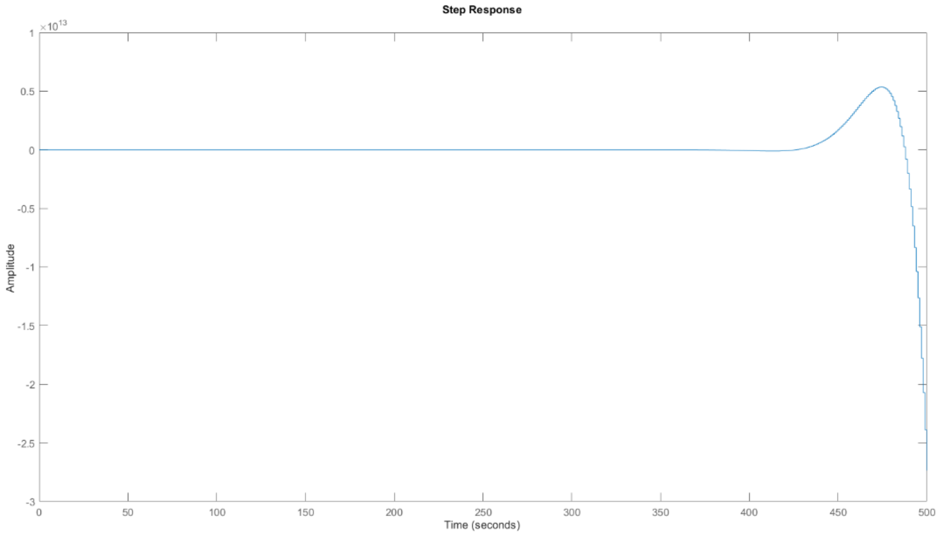
```
clc;
close all;
clear all;
rp=4;
rs=30;
wp=400/2000;
ws=800/2000;
[n,wn]=buttord(wp,ws,rp,rs);
[b,a]=butter(n,wn);
sys=tf(b,a);
figure;
freqs(b,a);
title('Frequency Response/Dipesh Dhungana/ACE077BCT035');
figure;
pzmap(sys),
title('Pole and Zero/Dipesh Dhungana/ACE077BCT035');
figure;
impulse(b,a),
[bz,az]=impinvar(b,a,10);
figure;
dimpulse(bz,az);
figure;
dstep(bz,az);
```

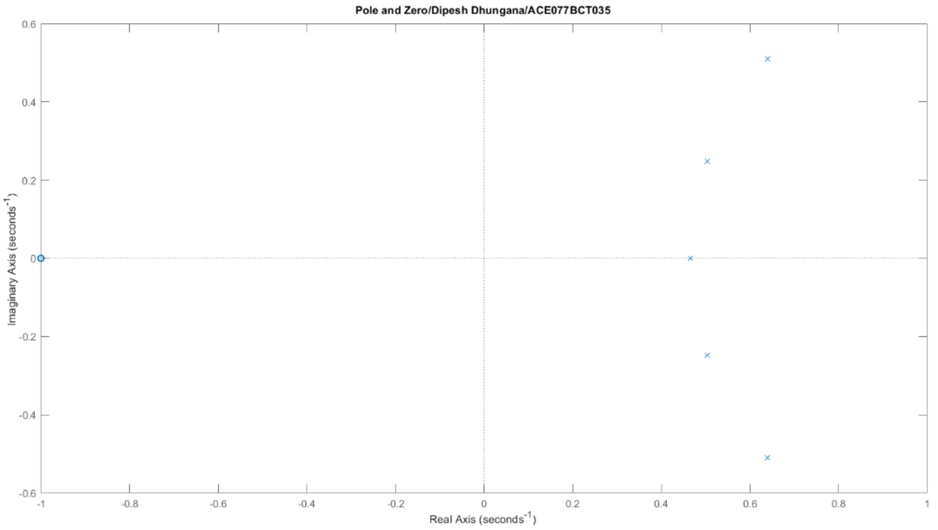
(Outputs)


Frequency Response/Dipesh Dhungana/ACE077BCT035
Magnitude
Frequency (rad/s)
Phase (degrees)
Frequency (rad/s)


Impulse Response
Amplitude
Time (seconds)


Impulse Response
Amplitude
Time (seconds)



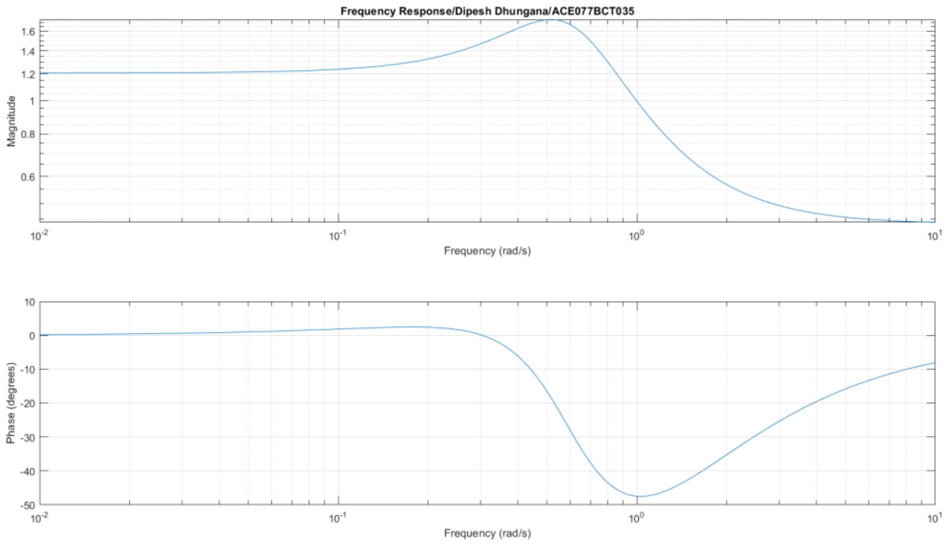


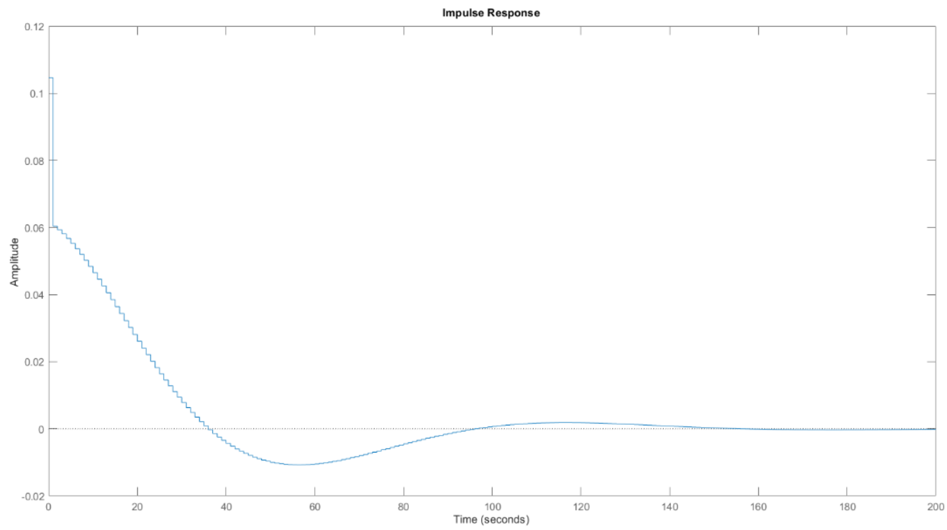
(Source code 4)

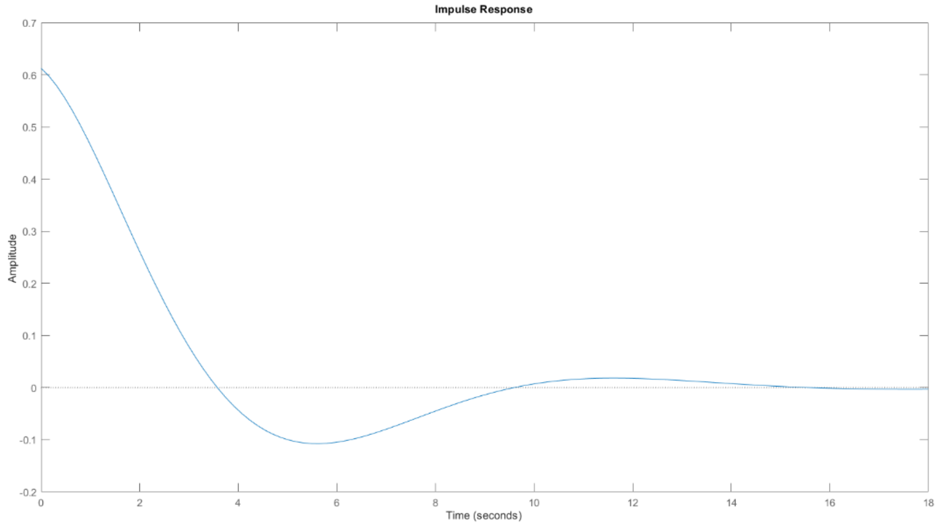
```
clc;
close all;
clear all;
rp=1;
rs=15;
wp=0.2*pi;
ws=0.3*pi;
[n,wn]=cheb1ord(wp,ws,rp,rs);
[b,a]=cheby1(n,rp,wn);
sys=tf(b,a);
figure;
freqs(b,a);
title('Frequency Response/Dipesh Dhungana/ACE077BCT035');
figure;
pzmap(sys),
title('Pole and Zero/Dipesh Dhungana/ACE077BCT035');
figure;
impulse(b,a),
```

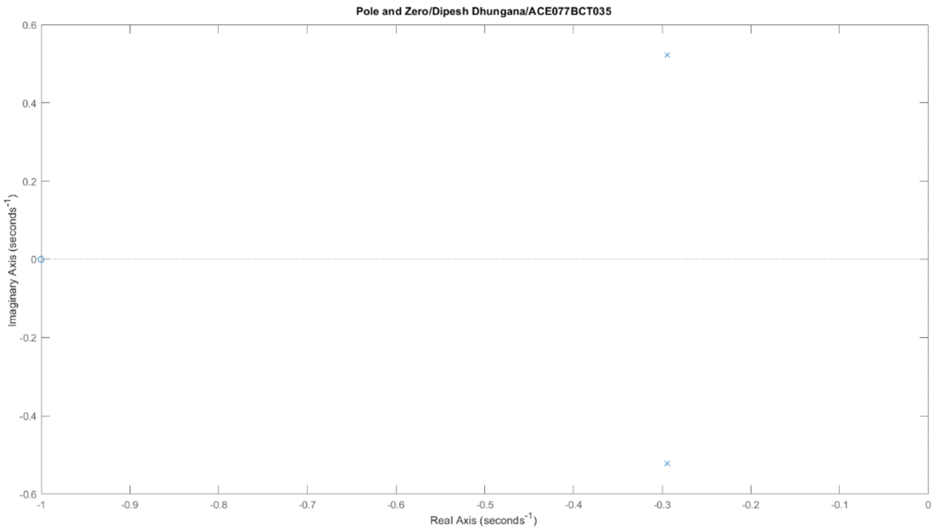
```
[bz,az]=impinvar(b,a,10);
figure;
dimpulse(bz,az);
figure;
dstep(bz,az);
```

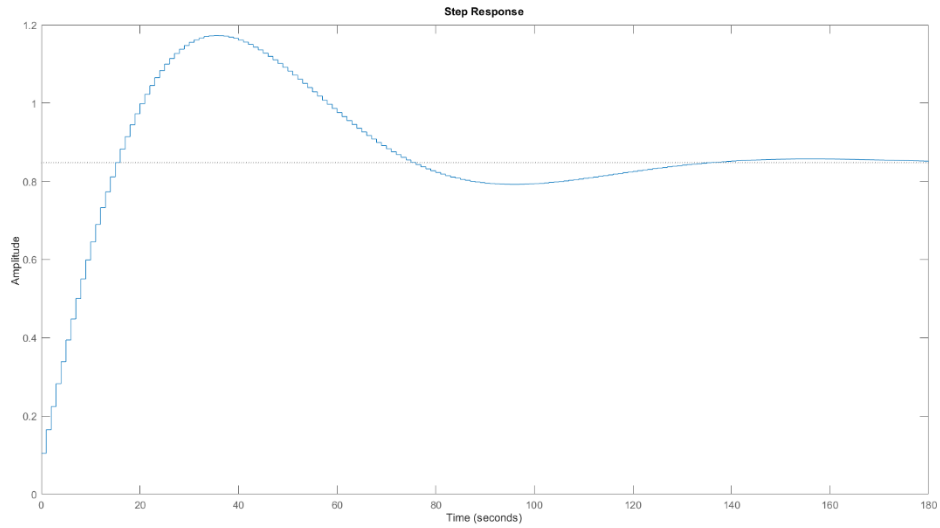
(Outputs)






Impulse Response
Amplitude
Time (seconds)

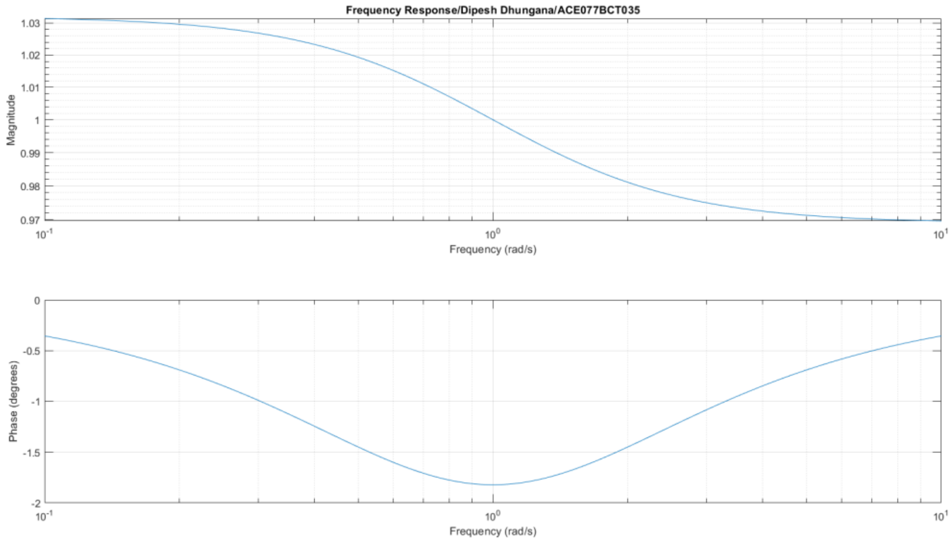

Pole and Zero/Dipesh Dhungana/ACE077BCT035
Imaginary Axis (seconds^{-1})
Real Axis (seconds^{-1})

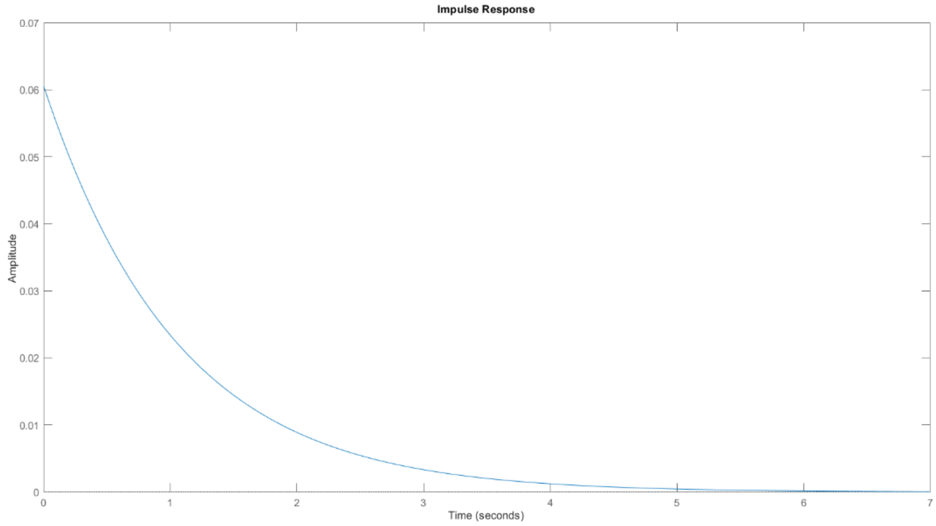

Step Response
Amplitude
Time (seconds)

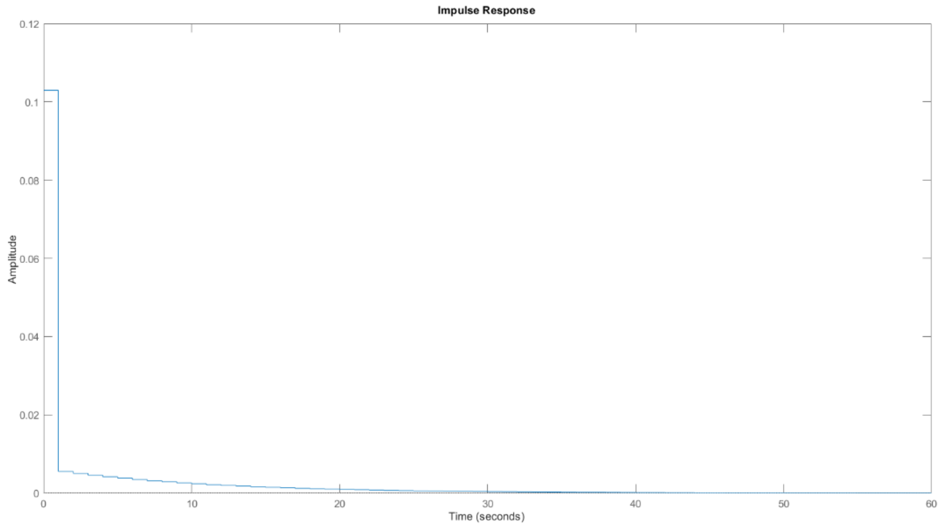
(Source code 5)

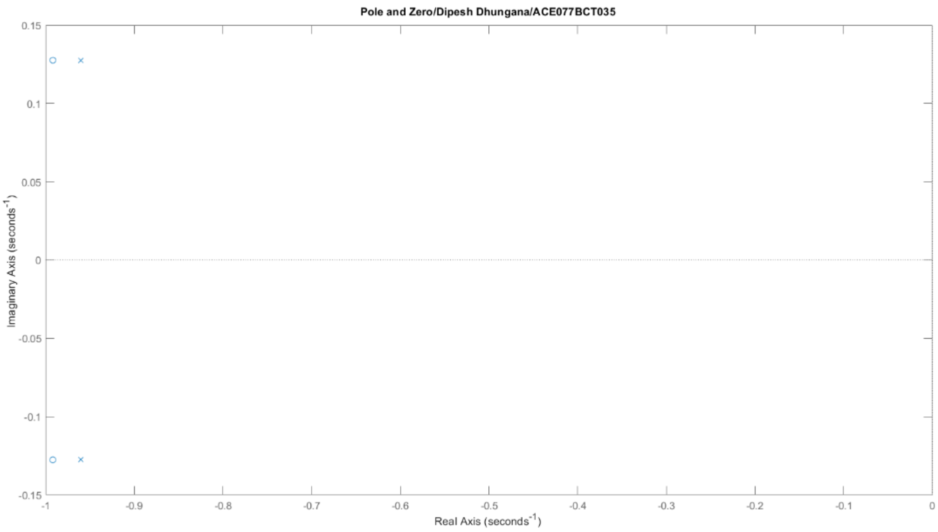
```
clc;
close all;
clear all;
rp=1;
rs=15;
wp=0.2*pi;
ws=0.3*pi;
[n,wn]=cheb2ord(wp,ws,rp,rs);
[b,a]=cheby2(n,rp,wn);
sys=tf(b,a);
figure;
freqs(b,a);
title('Frequency Response/Dipesh Dhungana/ACE077BCT035');
figure;
pzmap(sys),
title('Pole and Zero/Dipesh Dhungana/ACE077BCT035');
figure;
impulse(b,a),
[bz,az]=impinvar(b,a,10);
figure;
dimpulse(bz,az);
figure;
dstep(bz,az);
```

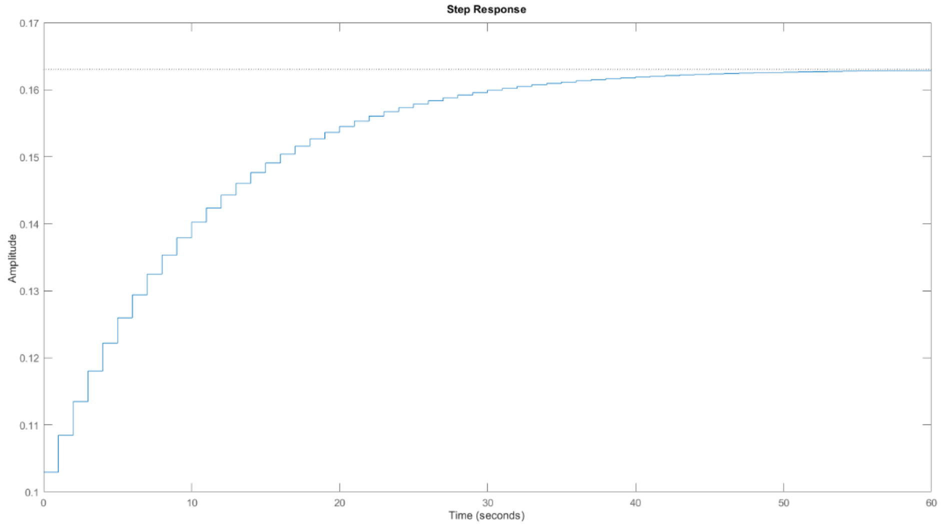
(Outputs)




Impulse Response
Amplitude
Time (seconds)
0.07
0.06
0.05
0.04
0.03
0.02
0.01
0
1
2
3
4
5
6
7


Impulse Response
Amplitude
Time (seconds)
0.12
0.1
0.08
0.06
0.04
0.02
0
10
20
30
40
50
60

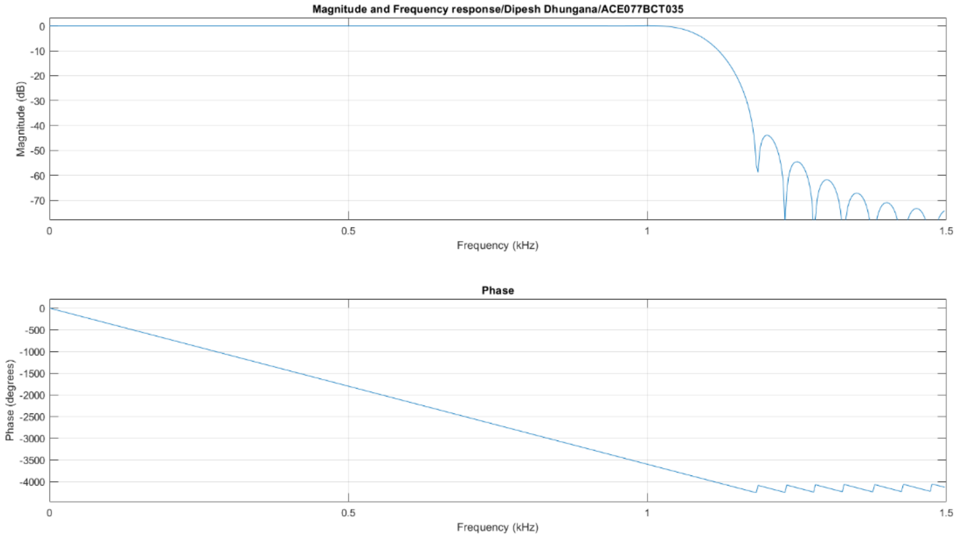

Pole and Zero/Dipesh Dhungana/ACE077BCT035
Imaginary Axis (seconds-1)
Real Axis (seconds-1)
0.15
0.1
0.05
0
-0.05
-0.1
-0.15
-1
-0.9
-0.8
-0.7
-0.6
-0.5
-0.4
-0.3
-0.2
-0.1
0



(Source Code 6)

```
clc;
close all;
clear all;
fp= 1000;%passband freq;
fs=1200;%stopband freq;
rs=45%;%stopband attenu;
rp=2; %passband attenu;
f=3000;%sampling freq;
num=-20*log(sqrt(rp*rs))-13;
dem=14.6*(fp-fs)/f;
n=ceil(num/dem)
n=abs(n);
wp=2*fp/f;
ws=2*fs/f;
wn=(wp+ws)/2;
if(rem(n,2)==0)
    m=n+1;
else
    m=n;
    n=n-1;
end
w=hann(m);
b=fir1(n,wn,'low',w);
freqz(b,1,500,3000);
title('Magnitude and Frequency response/Dipesh Dhungana/ACE077BCT035')
```

(Outputs)



Magnitude and Frequency response/Dipesh Dhungana/ACE077BCT035
Magnitude (dB)
Frequency (kHz)
Phase
Phase (degrees)
Frequency (kHz)

DISCUSSION AND CONCLUSION:

In our Digital Signal and Processing (DSAP) lab, we worked with two types of filters: Infinite Impulse Response (IIR) and Finite Impulse Response (FIR). IIR are typically used when efficiency is crucial and the stability of the filter can be carefully controlled. IIR filters used feedback, meaning that past outputs were fed back into the filter to help determine future outputs. This made IIR filters very efficient because they could achieve a desired response with fewer coefficients compared to FIR filters. However, this feedback can also make IIR filters less stable and more sensitive to numerical errors. FIR filters, on the other hand, did not use feedback. However, FIR filters usually required more coefficients than IIR filters to achieve the same performance, making them less efficient in terms of computational resources. During the lab, we observed that FIR filters were simpler to design and implement, and their stability made them easier to work with. IIR filters, while more complex, provided a more efficient solution for certain types of signal processing tasks due to their ability to achieve the desired response with fewer coefficients.