Applications of DSP

Project 2

UNIVERSITY OF CENTRAL FLORIDA
DEPARTMENT OF ELECTRICAL ENGINEERING
March 7, 2017
Authored by**: Rajib Dey**

Applications of DSP

Project 2

Question 1:

Part 1 a

Given,

I. $\boldsymbol{H(j\omega)}$ is a 4th order low pass Chebyshev function
II. Ap=0.5 dB
III. f1=2khz
IV. fs=10kHz

obtain $H_D(z)$.

Solution: Using low pass Chebyshev prototype that is available in MATLAB (cheb1ap) in the impulse invariant method we obtain the following output in the console window:

```
------------Answer to question 1-----------
1a)______ Hd(z)when f1=2KHz fs=10KHz______

ans =

  -9.095e-17 s^3 + 0.09261 s^2 + 0.2321 s + 0.04396
  -------------------------------------------------
   s^4 - 1.479 s^3 + 1.46 s^2 - 0.8131 s + 0.2221

Continuous-time transfer function.

poles(1)

z1 =

   1.0e+15 *

    1.0183
   -0.0000
   -0.0000


p1 =

   0.2323 + 0.7679i
   0.2323 - 0.7679i
   0.5071 + 0.2965i
   0.5071 - 0.2965i
```

Part 1 b:

Poles of $H_D(z)$

In Impulse invariant method, the transfer function in the analog domain is given by:

$$HA(s) = \sum_{i=1}^{N} \left(\frac{Ai}{s - pi}\right)$$

$$H_A(s) \leftrightarrow h_A(t)$$

Now, $h_D(t)=h_A(nT)$, where, T is the sampling period.

$$HD(z)=\sum_{i=1}^{N} Ai * \frac{z}{z-e^{PiT}}$$

We also know that, If our poles are inside the unit circle then our system is stable. From the console of the MATLAB we get The Magnitude of the poles:

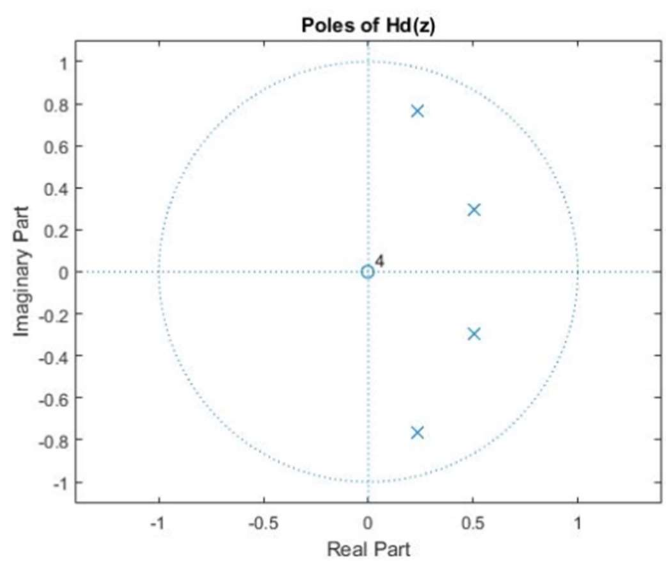
```
_________________1b_____________
the magnitude of each pole

p1_1 =

    0.8022


p1_2 =

    0.8022


p1_3 =

    0.5874


p1_4 =

    0.5874
```

Also we get the Angles of the poles:

```
the angle of each pole

A1_1 =

   73.1702


A1_2 =

  -73.1702


A1_3 =

   30.3081


A1_4 =

  -30.3081
```

Here, |Zp| max is 0.8022.

Pole Zero locations are mapped below :

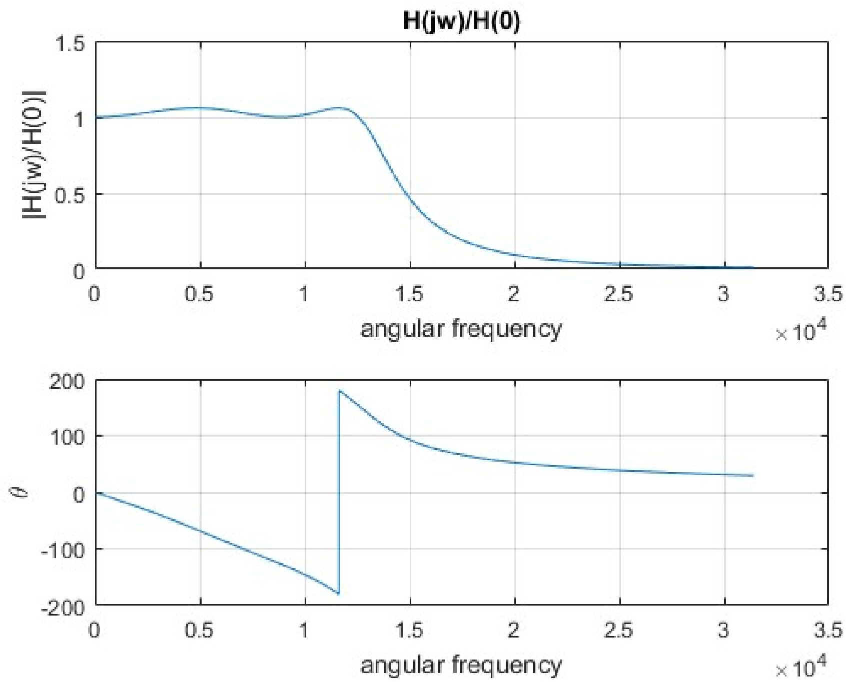


As all the poles are inside the unit circle, the system is stable.

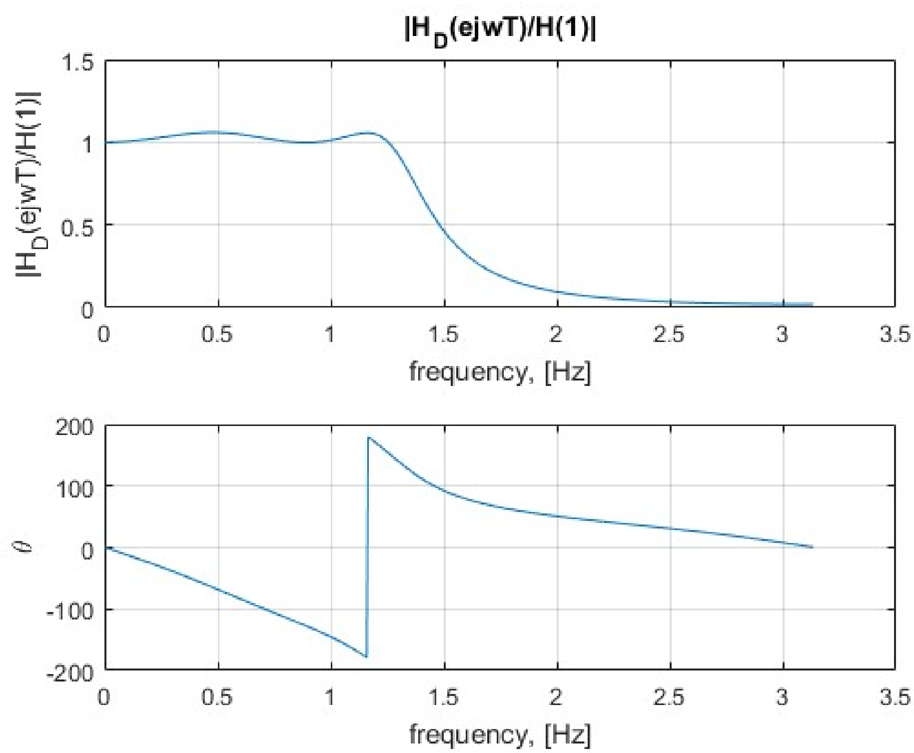
Part 1 c:

Here we need to plot The magnitude and phase of $\frac{H(jw)}{H(0)}$.

The Analog filter response is as given below:



The digital filter response is as given below:



Question 2:

Here, we repeat question 1 for f1=2kHz and fs=20KHz. Then we compare |Zp|max in (2) and (1) as well as the approximation accuracy of $\frac{Hde^{jwt}}{Hd(1)}$ relative to $\frac{H(jw)}{H(0)}$, |w|≤$\frac{ws}{2}$

Console output:

```
---------------------Answer to question 2--------------
2a______________Hd(z)when f1=2KHz fs=20KHz_____________

ans =

  -4.547e-17 s^3 + 0.007521 s^2 + 0.02436 s + 0.005167
  ----------------------------------------------------
     s^4 - 2.918 s^3 + 3.518 s^2 - 2.032 s + 0.4713

Continuous-time transfer function.

poles(2)

z3 =

   1.0e+14 *

    1.6538
   -0.0000
   -0.0000


p3 =

   0.7192 + 0.5338i
   0.7192 - 0.5338i
   0.7398 + 0.2004i
   0.7398 - 0.2004i
```

Part 2 b:

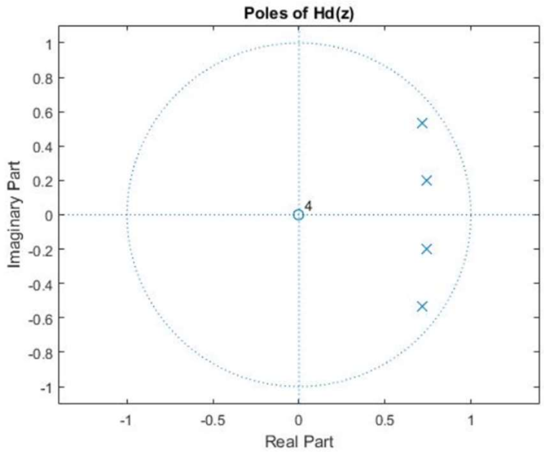
The Magnitude of the poles found from the console window:

```
__________2b_____________
the magnitude of each pole

p2_1 =

    0.8957


p2_2 =

    0.8957


p2_3 =

    0.7664


p2_4 =

    0.7664
```

The Angle of the poles found from the console window:

```
the angle of each pole

A2_1 =

   36.5851


A2_2 =

  -36.5851


A2_3 =

   15.1541


A2_4 =

  -15.1541
```

The pole-zero plot is as follows:



As all the poles are inside the unit circle and hence system is stable.

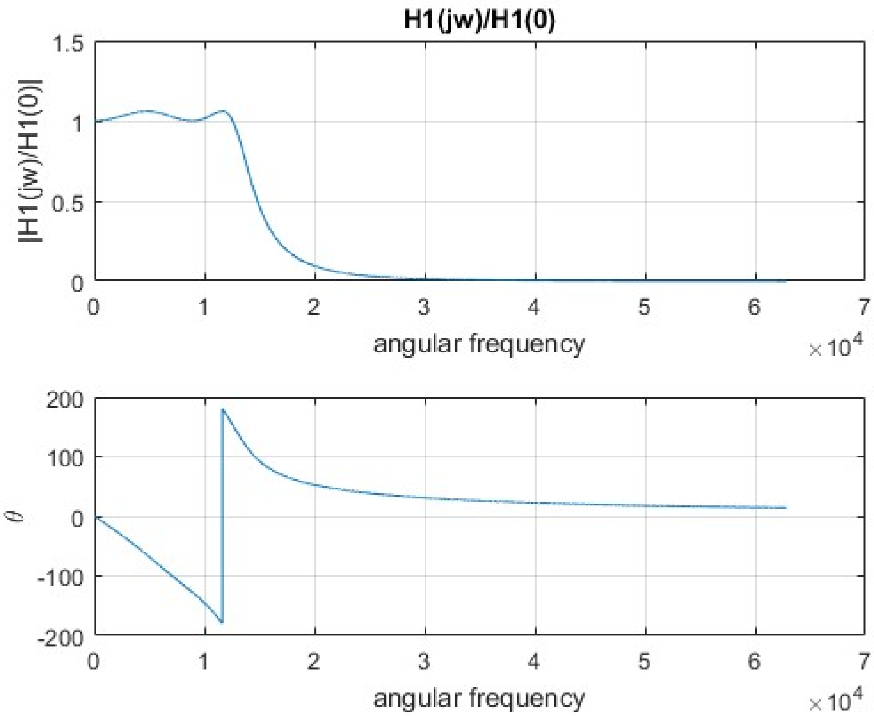
Part 2 c:

|Zp|max valus from the console window:

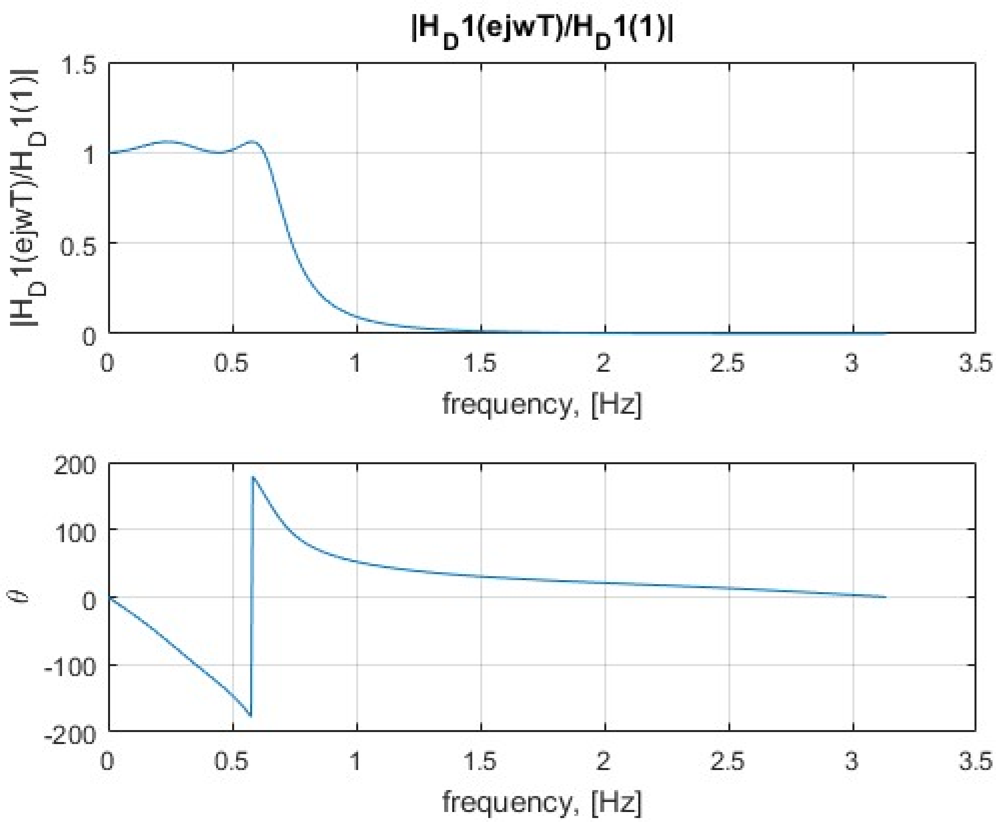
```
Zp1max =

    0.8022


Zp2max =

    0.8957
```

Now comparing the |Zp|max in both the questions, we see that we have P1= 0.8022 and P2=0.8957. So the |Zp|max in part 2 is larger.
Coming to the part where we have to compare the approximation accuracy of $H_D(e^{j\omega t})/H_D(1)$ relative to $H(j\omega)/H(0)$.

The Analog filter response is:



The Digital filter response is:



Question 3:

Here, we repeat question 1 with f1=3KHz and fs=15KHz.
The MATLAB console output showing the poles and zeros for this question is as follows:

```
--------------------Answer to question 3-----------------
3a____________ Hd(z)when f1=3KHz fs=15KHz _____________

ans =

  -6.063e-17 s^3 + 0.09261 s^2 + 0.2321 s + 0.04396
  -------------------------------------------------
   s^4 - 1.479 s^3 + 1.46 s^2 - 0.8131 s + 0.2221

Continuous-time transfer function.

poles(3)

z5 =

   1.0e+15 *

    1.5274
   -0.0000
   -0.0000


p5 =

   0.2323 + 0.7679i
   0.2323 - 0.7679i
   0.5071 + 0.2965i
   0.5071 - 0.2965i
```

The Magnitude of the poles found from the console window:

```
the magnitude of each pole

p3_1 =

    0.8022


p3_2 =

    0.8022


p3_3 =

    0.5874


p3_4 =

    0.5874
```

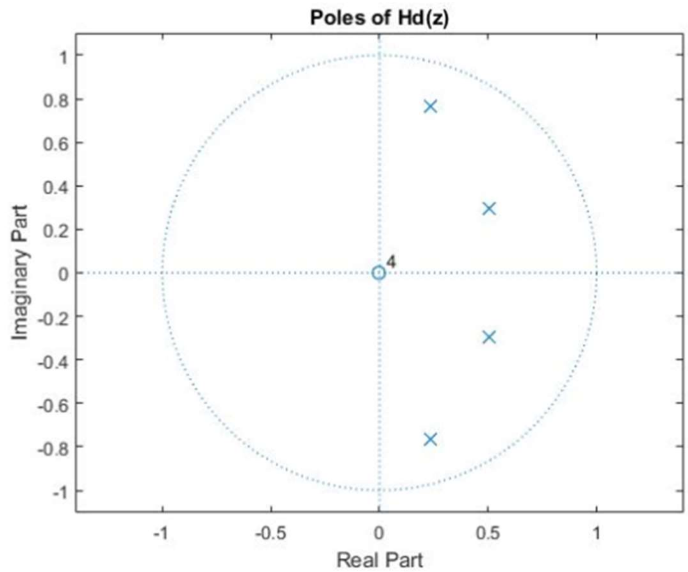
The Angle of the poles found from the console window:

```
the angle of each pole

A3_1 =

   73.1702


A3_2 =

  -73.1702


A3_3 =

   30.3081


A3_4 =

  -30.3081
```

|Zp|max valus from the console window:

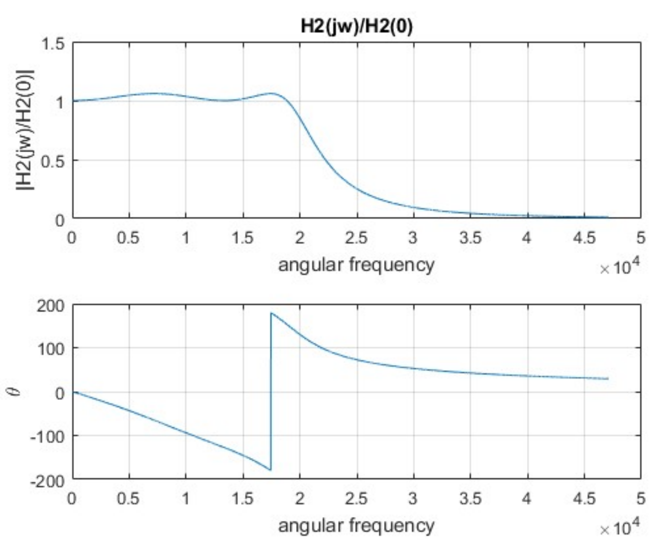
```
Zp1max =

    0.8022


Zp3max =

    0.8022
```

Comparing problem 1.b and this one, we get |Zp|max for 1.b = |Zp|max for 3. Because that the ratio of the passband frequency is equal to that of the sampling frequency, pole locations in question1 and question 3 are same.

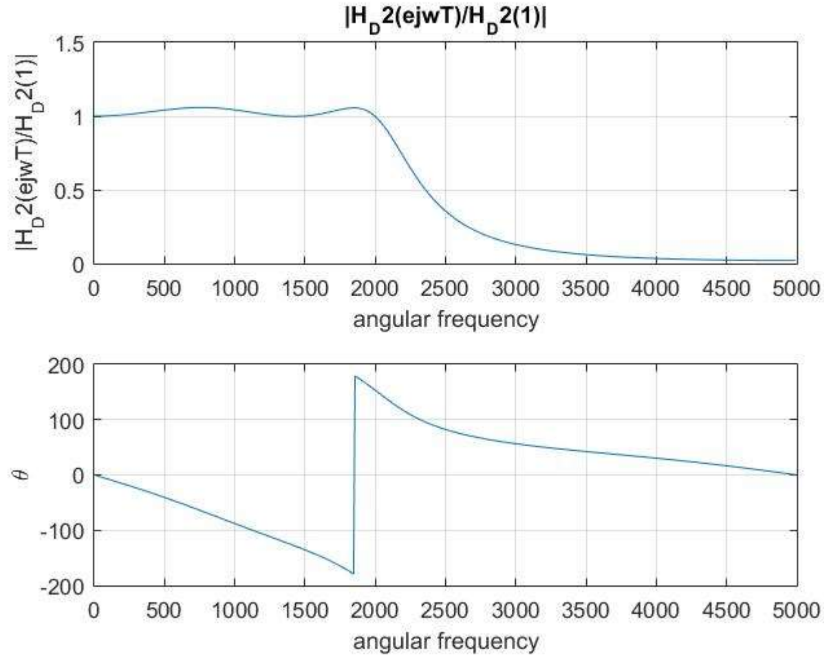


Since all the poles are inside the unit circle, the system is stable.

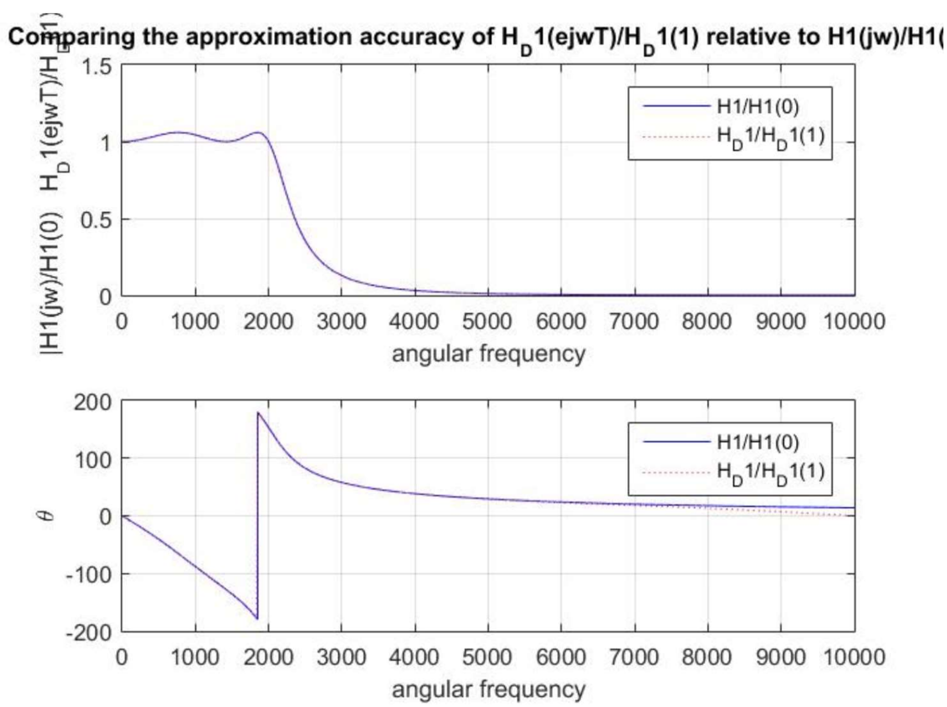
The Analog filter response is as follows:



The Digital filter response is as follows:



Comparing the approximation accuracy



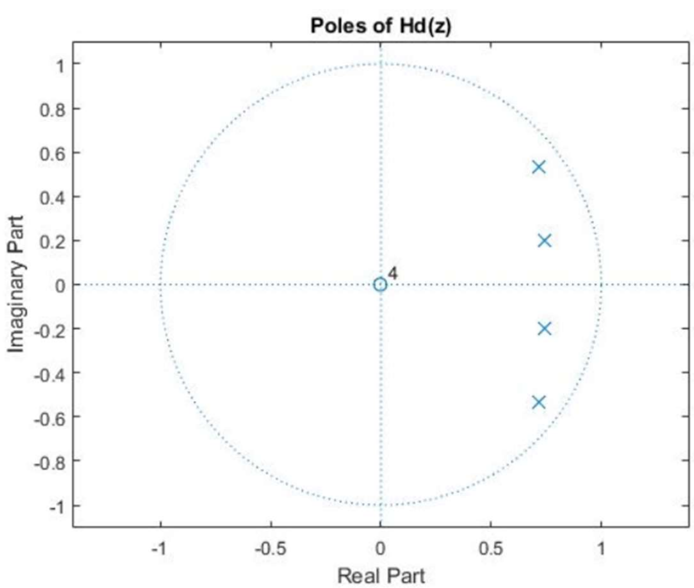
Question 4:

Here, we repeat question 2 with f1=4KHz and fs=40KHz.

The console output is as follows:

```
-------------------Answer to question 4------------
4a____________ Hd(z)for f1=4KHz fs=40KHz__________

ans =

  2.274e-17 s^3 + 0.007521 s^2 + 0.02436 s + 0.005167
  ---------------------------------------------------
    s^4 - 2.918 s^3 + 3.518 s^2 - 2.032 s + 0.4713

Continuous-time transfer function.

poles(4)

z7 =

   1.0e+14 *

   -3.3076
   -0.0000
   -0.0000


p7 =

   0.7192 + 0.5338i
   0.7192 - 0.5338i
   0.7398 + 0.2004i
   0.7398 - 0.2004i
```

The pole locations are following:



Since all the poles are inside the unit circle, the system is stable.

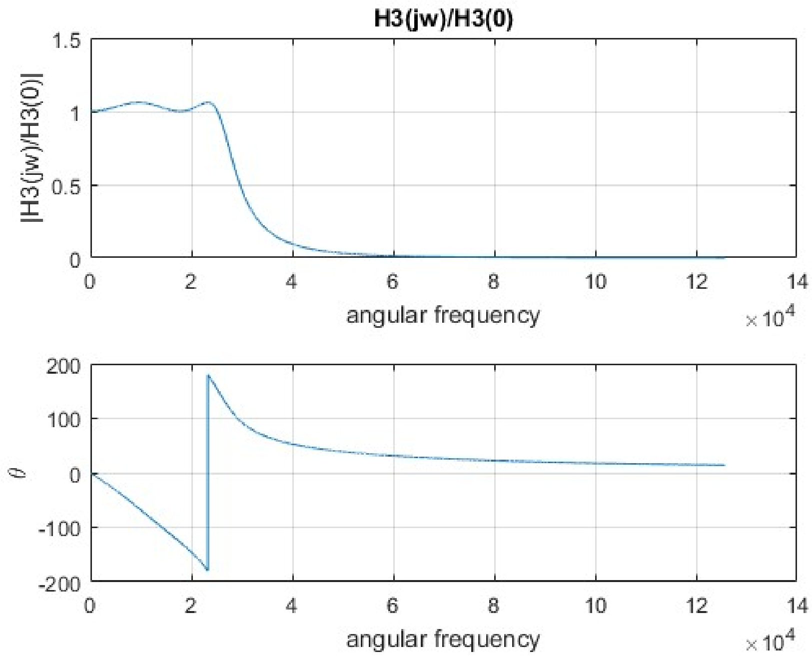
The Magnitude of the poles found from the console window:

```
the magnitude of each pole

p4_1 =

    0.8957


p4_2 =

    0.8957


p4_3 =

    0.7664


p4_4 =

    0.7664
```

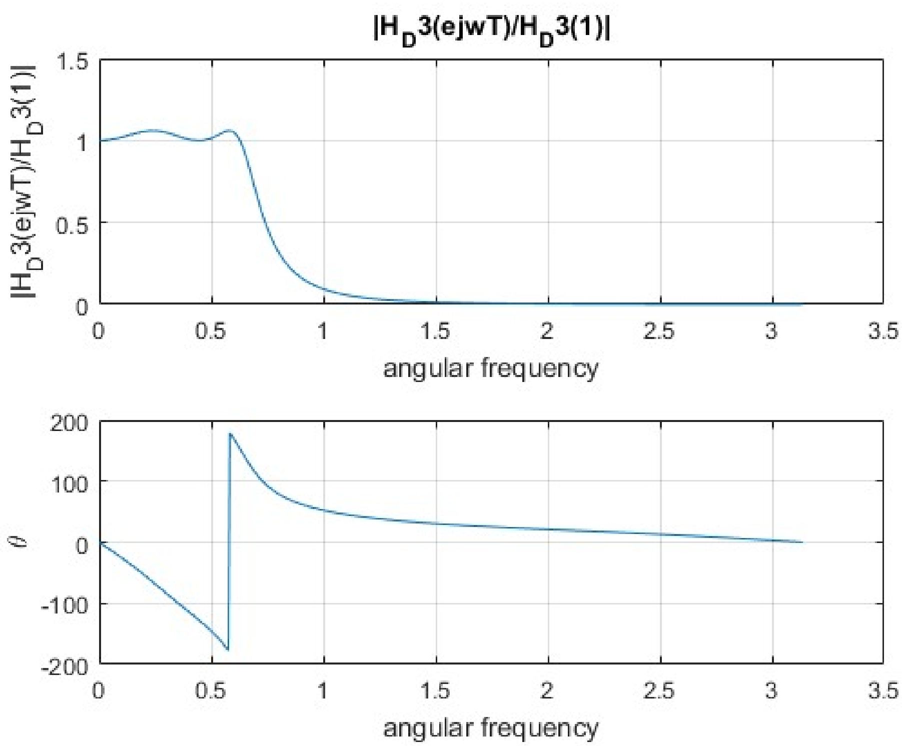
The Angle of the poles found from the console window:

```
the angle of each pole

A4_1 =

   36.5851


A4_2 =

  -36.5851


A4_3 =

   15.1541


A4_4 =

  -15.1541
```

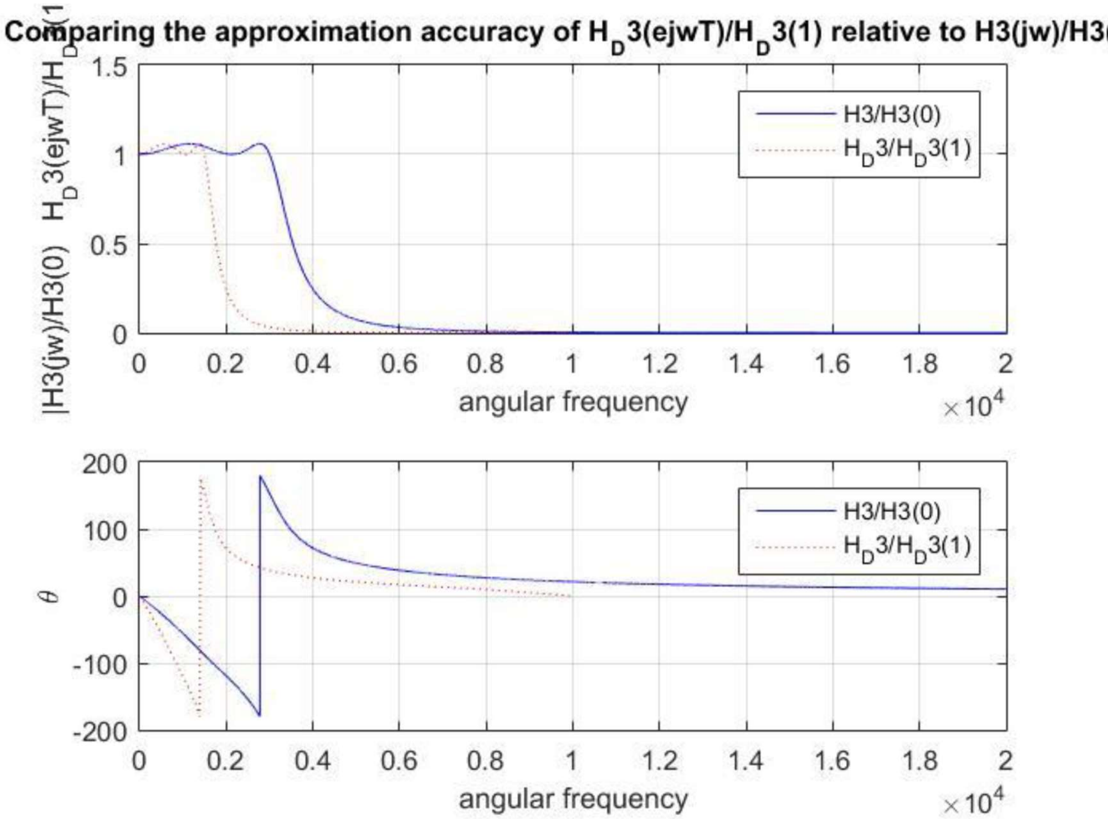
The Analog filter response is as follows:



The Digital filter response is as follows:



Comparing the approximation accuracy of $HDe^{j\omega t}/H_D(1)$ relative to $H(J\omega)/H(0)$:



<u>Question 5:</u>

Here, we repeat question 1 using bilinear transformation method. We compute the new value of f1 of HD(z) analytically.

Bilinear Transformation is used to map a transfer function from the analog to the digital domain. This maps from s-plane to z-plane by using the following transformation of s.

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

$$\omega c = 2\pi fs = 1.4 \times 10^{4} rad/sec$$

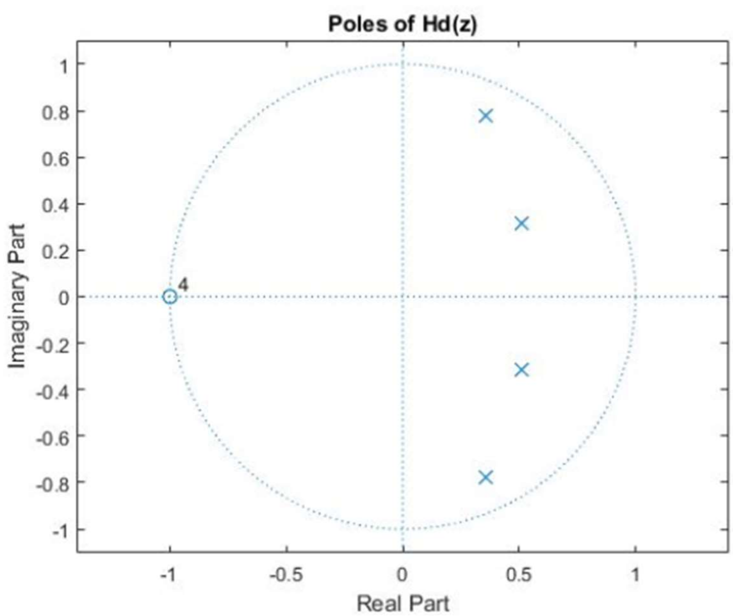
$$\Omega\, c = \frac{2}{T} tan^{-1}\left(\frac{\omega cT}{2}\right)$$
$$= 1.2 \times 10^{4\, rad/sec}$$

So f_1=1.9 kHz

From MATLAB Console:

```
--------------------Answer to question 5------------------
________ Hd(z) when fs=40KHz___________________________

ans =

  0.02033 s^4 + 0.08131 s^3 + 0.122 s^2 + 0.08131 s + 0.02033
  -----------------------------------------------------------
        s^4 - 1.735 s^3 + 1.822 s^2 - 1.009 s + 0.2661

Continuous-time transfer function.
```

The pole locations are following:



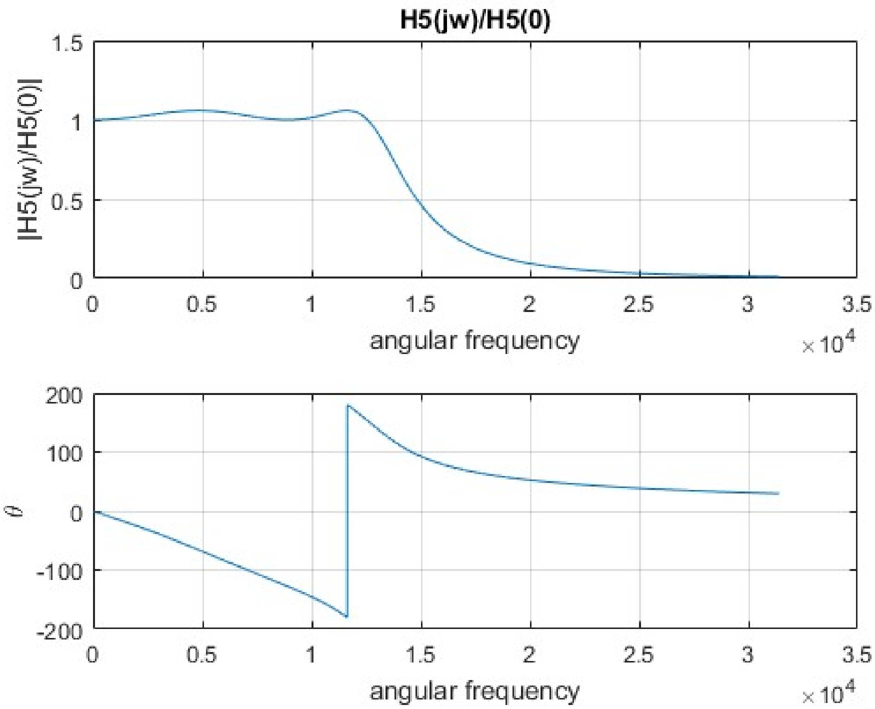
The Magnitude of the poles found from the console window:

```
the magnitude of each pole

p5_1 =

    0.8552


p5_2 =

    0.8552


p5_3 =

    0.6033


p5_4 =

    0.6033
```

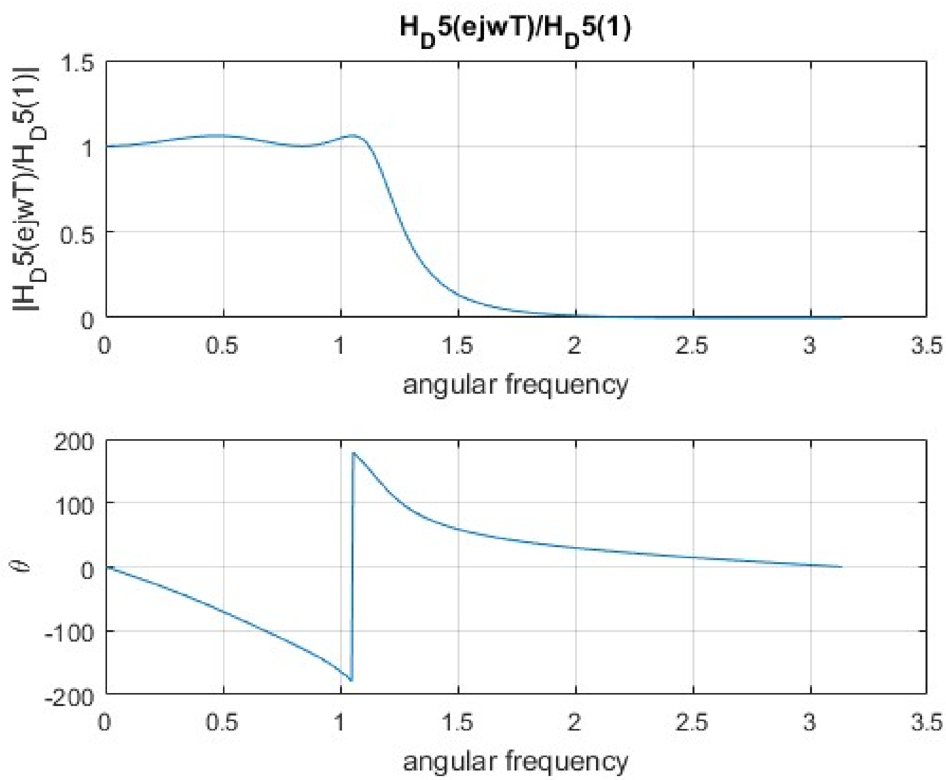
The Angle of the poles found from the console window:

```
the angle of each pole

A5_1 =

   65.5688


A5_2 =

  -65.5688


A5_3 =

   31.6162


A5_4 =

  -31.6162
```

The Analog filter response is as follows:



The Digital filter response is as follows:



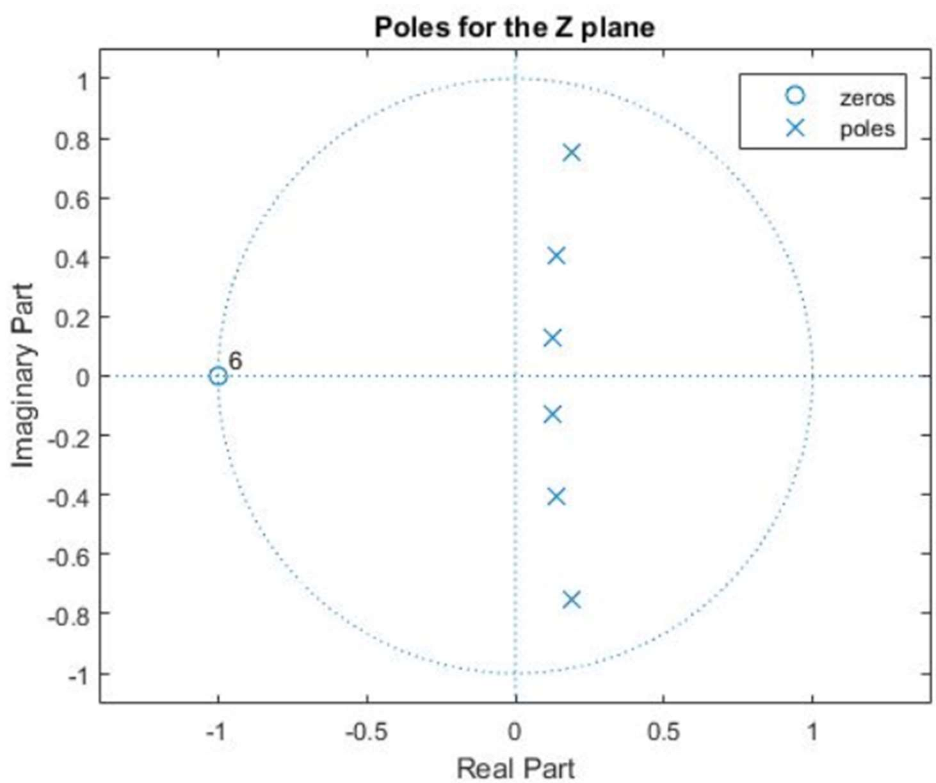
Question 6:

Here, We repeat question 1 for a 6th order HP Butterworth filter $\boldsymbol{H(j\omega)}$ where $\boldsymbol{\omega s}$=$\boldsymbol{4\omega}_{3dB}$. We compute 3dB frequency for digital filter and compare with graph.

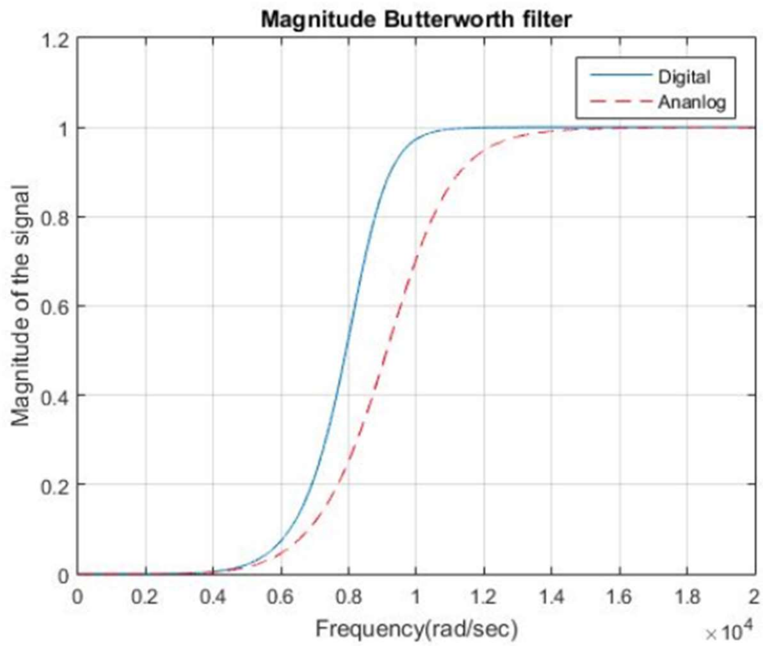
The console output is as follows:

```
H_bilinear =

(- 4.524e-7 + 1.524i)/((6366.0*(2.0*z - 2.0))/(z + 1.0) + (7071.0 + 7071.0i)) +
(- 2.285 - 1.319i)/((6366.0*(2.0*z - 2.0))/(z + 1.0) + (9659.0 - 2588.0i)) +
(- 4.524e-7 - 1.524i)/((6366.0*(2.0*z - 2.0))/(z + 1.0) + (7071.0 - 7071.0i)) +
(0.3536 - 0.2041i)/((6366.0*(2.0*z - 2.0))/(z + 1.0) + (2588.0 - 9659.0i)) +
(- 2.285 + 1.319i)/((6366.0*(2.0*z - 2.0))/(z + 1.0) + (9659.0 + 2588.0i)) +
(0.3536 + 0.2041i)/((6366.0*(2.0*z - 2.0))/(z + 1.0) + (2588.0 + 9659.0i))

cutoff frequency: 8477.00 rad/seconds
analytical cutoff frequency : 8476.89 rad/seconds
```

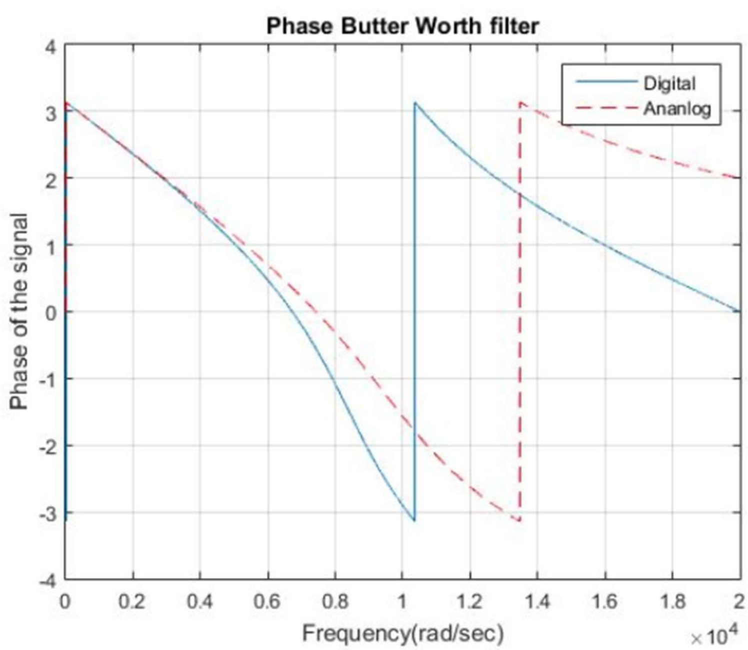
The pole zero plot is like the following:



Plot of the magnitude of the Butterworth filter:



Plot of the phase of the Butterworth filter:

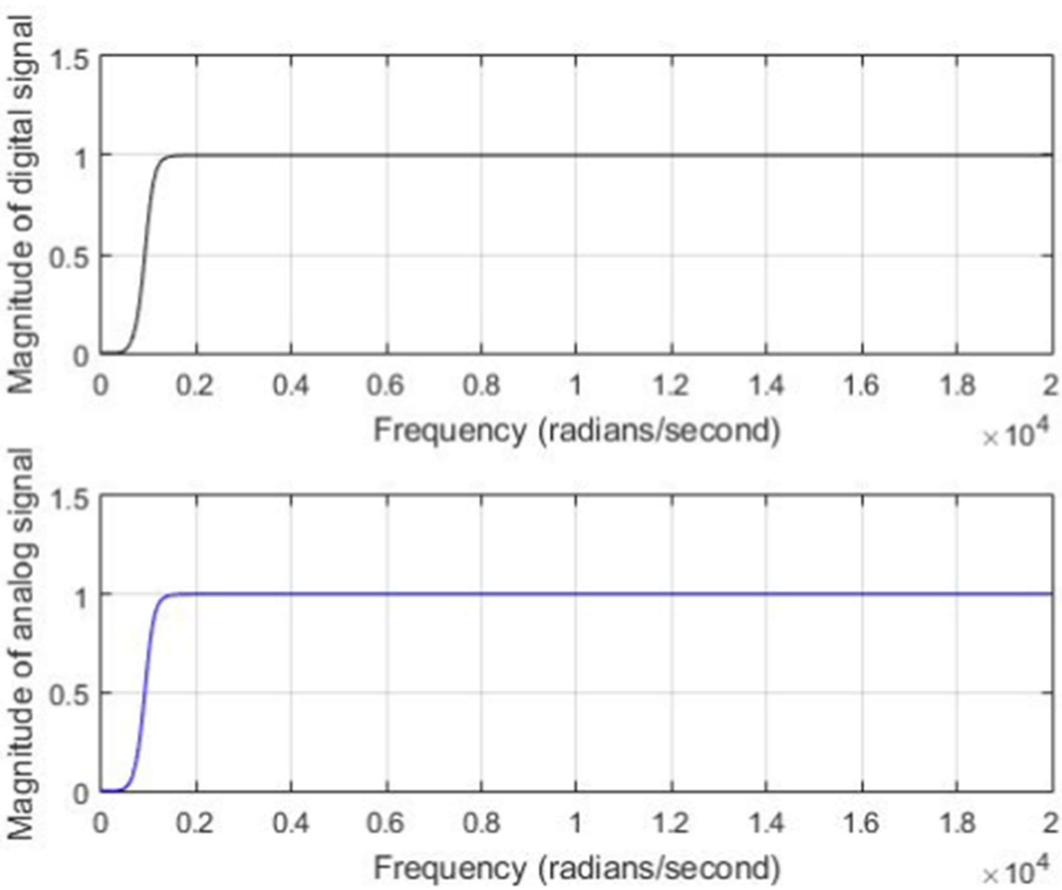


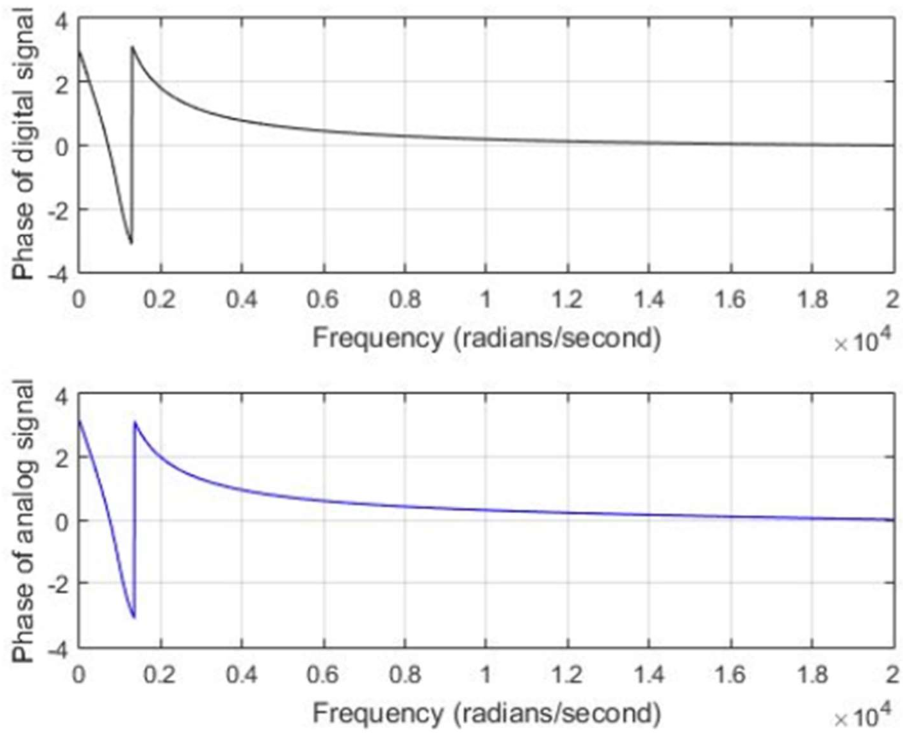
Question 7:

Here, we repeat 6 by prewarping the prototype to get the exact 3-dB frequency of the digital filter.

The continuous time filter frequency ωa=0 corresponds to the discrete time filter frequency ωd=0. The continuous time filter frequency ωa=±∞ corresponds to the discrete time frequency ωd=±(πT). This is called *frequency warping*. The continuous-time filter can be designed to compensate for this frequency warping by setting ωa=2Ttan$\omega d T$2 for every frequency specification that the designer has control over (such as corner frequency or center frequency). This is called *pre-warping* the filter design.

The MATLAB output is as follows:





Conclusion:

In this project we have worked on Chebyshev and Butterworth filters using MATLAB. Using Bilinear transformation method, we calculated the new value of f1. Then we pewarped to getttttt 3-db frequency of the digital filter.

Reference:

1. https://en.wikipedia.org/wiki/Butterworth_filter
2. https://www.mathworks.com/
3. https://en.wikipedia.org/wiki/Chebyshev_filter
4. https://en.wikipedia.org/wiki/Bilinear_transform
5. John G. Proakis, Dimitri K. Manolakis, Digital Signal Processing, Pearson, 4th edition

The following is the codes and plots of the entire project in MATLAB.

MATLAB Code

```
% ------- Applications of DSP ----------------
% ------------Project 2 ----------------------
%------------Submitted by---------------------
%-------------Rajib Dey ----------------------


clc
close all
clear all

f1=2000;
fs=10000;
fsp=0:0.1:fs/2;
w=2*pi*fsp;

%%%%%%%%%%%%%%%%%%%%%% 1(a)
[z,p,k]=cheb1ap(4,0.5);% low pass filter prototype
[b,a] = zp2tf(z,p,k); %  % Convert to transfer function form
[b,a] = lp2lp(b,a,2*pi*f1);% analog low pass filter to low pass filter of
specified frequency
[bz,az]=impinvar(b,a,fs);% using impulse invarient method
disp('------------Answer to question 1-----------')
disp ('1a)______ Hd(z)when f1=2KHz fs=10KHz______')
tf(bz,az)

%%%%%%%%%%%%%%%%%%%%%%%1(b) poles of Hd(z)
disp ('poles(1)')
[z1,p1,k1] = tf2zp(bz,az)
disp ('_________________1b_____________')
disp ('the magnitude of each pole')
p1_1 = abs(p1(1))
p1_2 = abs(p1(2))
p1_3 = abs(p1(3))
p1_4 = abs(p1(4))
zplane(z,p)
disp ('the angle of each pole')
A1_1= angle(p1(1))*57.2958
A1_2=angle(p1(2))*57.2958
A1_3=angle(p1(3))*57.2958
A1_4=angle(p1(4))*57.2958

%1(c) magnitude and phase plot for |H(jw)/H(0)|

H = freqs(b,a,w);

figure(11);
subplot(2,1,1);
 plot(w, abs(H/H(1)));
grid on;
xlabel('angular frequency')
ylabel('|H(jw)/H(0)|')
title('H(jw)/H(0)')
```

```
subplot(2,1,2);
plot(w,angle(H/H(1))*180/pi);
xlabel('angular frequency')
ylabel('\theta')
grid on;

% magnitude and phase plot for |(H_D)/(H_D(1)))

[H_D,w1] = freqz(bz,az);

figure(12);
subplot(2,1,1);
plot(w1, abs((H_D)/(H_D(1))));
xlabel('frequency, [Hz]')
ylabel('|H_D(ejwT)/H(1)|')
title('|H_D(ejwT)/H(1)|')
grid on;
subplot(2,1,2);
plot(w1,angle((H_D)/(H_D(1)))*180/pi);
xlabel('frequency, [Hz]')
ylabel('\theta')
grid on;
%comparing the approximation accuracy of  H_D(ejwT)/H(0) and H(jw)/H(0)
figure(100);
subplot(2,1,1);
plot(w, abs(H/H(1)),'b-');
ylabel('|H(jw)/H(0)   H_D(ejwT)/H_D(1)|')
xlabel('angular frequency')
title('Comparing the approximation accuracy of H_D(ejwT)/H_D(1) relative to
H(jw)/H(0)')
hold on;
plot(w1, abs(H_D/H_D(1)),'r:');
legend('H/H(0)','H_D/H_D(1)')
grid on;

subplot(2,1,2);
plot(w,angle(H/H(1))*180/pi,'b-');
hold on;
plot(w1,angle(H_D/H_D(1))*180/pi,'r:');
ylabel('\theta')
xlabel('angular frequency')
legend('H/H(0)','H_D/H_D(1)')
grid on;
%%%%%%%%%%%%%%%% (2)for f1= 2000 fs1=20000

fs1=20000;
fsp1=0:1:fs1/2;
w1=2*pi*fsp1;

[z2,p2,k2]=cheb1ap(4,0.5);% low pass filter prototype
[b1,a1] = zp2tf(z2,p2,k2); %  % Convert to transfer function form
[b1,a1] = lp2lp(b1,a1,2*pi*f1);% analog low pass filter to low pass filter of
specified frequency
[bz1,az1]=impinvar(b1,a1,fs1);% using impulse invarient method
```

```
disp('--------------------Answer to question 2--------------')
disp ('2a_____________Hd(z)when f1=2KHz fs=20KHz____________')
tf(bz1,az1)
disp ('poles(2)')
[z3,p3,k3] = tf2zp(bz1,az1)
disp ('__________2b_____________')
disp ('the magnitude of each pole')
p2_1 = abs(p3(1))
p2_2 = abs(p3(2))
p2_3 = abs(p3(3))
p2_4 = abs(p3(4))
zplane(z,p)
disp ('the angle of each pole')
A2_1= angle(p3(1))*57.2958
A2_2=angle(p3(2))*57.2958
A2_3=angle(p3(3))*57.2958
A2_4=angle(p3(4))*57.2958

%%%comparing the Zpmax at (1) and (2)
Zp1max = max(abs(p1))
Zp2max = max(abs(p3))

%%comparing the approximation accuracy of  H_D(ejwT)/H(0) and H(jw)/H(0)

H1 = freqs(b1,a1,w1);
figure(21);
subplot(2,1,1);
 plot(w1, abs(H1/H1(1)));
grid on;
xlabel('angular frequency')
ylabel('|H1(jw)/H1(0)|')
title('H1(jw)/H1(0)')
subplot(2,1,2);
plot(w1,angle(H1/H1(1))*180/pi);
xlabel('angular frequency')
ylabel('\theta')
grid on;
[H_D1,w2] = freqz(bz1,az1);
figure(22);
subplot(2,1,1);
plot(w2, abs((H_D1)/(H_D1(1))));
xlabel('frequency, [Hz]')
ylabel('|H_D1(ejwT)/H_D1(1)|')
title('|H_D1(ejwT)/H_D1(1)|')
grid on;
subplot(2,1,2);
plot(w2,angle((H_D1)/(H_D1(1)))*180/pi);
xlabel('frequency, [Hz]')
ylabel('\theta')
grid on;



figure(3);
subplot(2,1,1);
plot(w1, abs(H1/H1(1)),'b-');
```

```
ylabel('|H1(jw)/H1(0)   H_D1(ejwT)/H_D(1)|')
xlabel('angular frequency')
title('Comparing the approximation accuracy of H_D1(ejwT)/H_D1(1) relative to
H1(jw)/H1(0)')
hold on;
plot(w2, abs(H_D1/H_D1(1)),'r:');
legend('H1/H1(0)','H_D1/H_D1(1)')
grid on;

subplot(2,1,2);
plot(w1,angle(H1/H1(1))*180/pi,'b-');
hold on;
plot(w2,angle(H_D1/H_D1(1))*180/pi,'r:');
ylabel('\theta')
xlabel('angular frequency')
legend('H1/H1(0)','H_D1/H_D1(1)')
grid on;

%%%%%%%%%%%%%%%%%%(3) for f2=3khz and fs2=15khz

f2=3000;
W2=2*pi*f2;
fs2=15000;

fsp2=0:1:fs2/2;
w3=2*pi*fsp2;

[z4,p4,k4]=cheb1ap(4,0.5);% low pass filter prototype
[b2,a2] = zp2tf(z4,p4,k4); %  % Convert to transfer function form
[b2,a2] = lp2lp(b2,a2,2*pi*f2);% analog low pass filter to low pass filter of
specified frequency
[bz2,az2]=impinvar(b2,a2,fs2);% using impulse invarient method

 %%%%%%%%%%%%%%%%%%%%%%%%%%%% poles of Hd3(z)
disp('-------------------Answer to question 3----------------')
disp ('3a____________ Hd(z)when f1=3KHz fs=15KHz _____________')
tf(bz2,az2)
disp ('poles(3)')
[z5,p5,k5] = tf2zp(bz2,az2)
disp ('the magnitude of each pole')
p3_1 = abs(p5(1))
p3_2 = abs(p5(2))
p3_3 = abs(p5(3))
p3_4 = abs(p5(4))
zplane(z,p)
disp ('the angle of each pole')
A3_1= angle(p5(1))*57.2958
A3_2=angle(p5(2))*57.2958
A3_3=angle(p5(3))*57.2958
A3_4=angle(p5(4))*57.2958
%%%%%%%%%%%%%%%%%comparing the Zpmax at (1) and (3)
Zp1max = max(abs(p1))
Zp3max = max(abs(p5))

H2 = freqs(b2,a2,w3);
```

```
figure(31);
subplot(2,1,1);
plot(w3, abs(H2/H2(1)));
grid on;
xlabel('angular frequency')
ylabel('|H2(jw)/H2(0)|')
title('H2(jw)/H2(0)')
subplot(2,1,2);
plot(w3,angle(H2/H2(1))*180/pi);
xlabel('angular frequency')
ylabel('\theta')
grid on;

[H_D2,w4] = freqz(bz2,az2);


figure(32);
subplot(2,1,1);
plot(w4, abs((H_D2)/(H_D2(1))));
xlabel('angular frequency')
ylabel('|H_D2(ejwT)/H_D2(1)|')
title('|H_D2(ejwT)/H_D2(1)|')
grid on;
subplot(2,1,2);
plot(w4,angle((H_D2)/(H_D2(1)))*180/pi);
xlabel('angular frequency')
ylabel('\theta')
grid on;


%%%%%%%%%%%%%%%%%%%Code for 4

f3=4000;
W3=2*pi*f3;
fs3=40000;
df=1;
fsp3=0:df:fs3/2;
w4=2*pi*fsp3;

[z6,p6,k6]=cheb1ap(4,0.5);% low pass filter prototype
[b3,a3] = zp2tf(z6,p6,k6); %  % Convert to transfer function form
[b3,a3] = lp2lp(b3,a3,2*pi*f3);% analog low pass filter to low pass filter of
specified frequency
[bz3,az3]=impinvar(b3,a3,fs3);% using impulse invarient method

 % poles of Hd4(z)
disp('-------------------Answer to question 4------------')
disp ('4a____________ Hd(z)for f1=4KHz fs=40KHz__________')
tf(bz3,az3)

disp ('poles(4)')
[z7,p7,k7] = tf2zp(bz3,az3)
disp ('the magnitude of each pole')
p4_1 = abs(p7(1))
```

```
p4_2 = abs(p7(2))
p4_3 = abs(p7(3))
p4_4 = abs(p7(4))
zplane(z,p)
disp ('the angle of each pole')
A4_1= angle(p7(1))*57.2958
A4_2=angle(p7(2))*57.2958
A4_3=angle(p7(3))*57.2958
A4_4=angle(p7(4))*57.2958
%%%comparing the Zpmax at (2) and (4)
Zp2max = max(abs(p3))
Zp4max = max(abs(p7))

H3 = freqs(b3,a3,w4);

figure(41);
subplot(2,1,1);
plot(w4, abs(H3/H3(1)));
grid on;
xlabel('angular frequency')
ylabel('|H3(jw)/H3(0)|')
title('H3(jw)/H3(0)')
subplot(2,1,2);
plot(w4,angle(H3/H3(1))*180/pi);
xlabel('angular frequency')
ylabel('\theta')
grid on;
[H_D3,w5] = freqz(bz3,az3);
figure(42);
subplot(2,1,1);
plot(w5, abs((H_D3)/(H_D3(1))));
xlabel('angular frequency')
ylabel('|H_D3(ejwT)/H_D3(1)|')
title('|H_D3(ejwT)/H_D3(1)|')
grid on;
subplot(2,1,2);
plot(w5,angle((H_D3)/(H_D3(1)))*180/pi);
xlabel('angular frequency')
ylabel('\theta')
grid on;
figure(43)
subplot(2,1,1);
plot(w4/2/pi, abs(H3/H3(1)),'b-');
ylabel('|H3(jw)/H3(0)   H_D3(ejwT)/H_D3(1)|')
xlabel('frequency, [Hz]')
title('Comparing the approximation accuracy of H_D3(ejwT)/H_D3(1) relative to
H3(jw)/H3(0)')
hold on;
plot(w5/2/pi*fs1, abs(H_D3/H_D3(1)),'r:');
legend('H3/H3(0)','H_D3/H_D3(1)')
grid on;

subplot(2,1,2);
plot(w4/pi/2,angle(H3/H3(1))*180/pi,'b-');
hold on;
plot(w5/pi*fs1/2,angle(H_D3/H_D3(1))*180/pi,'r:');
```

```
ylabel('\theta')
xlabel('frequency, [Hz]')
legend('H3/H3(0)','H_D3/H_D3(1)')
grid on;

%%%%%%%%%%%%%%%%(5) using bilinear transformation

%%%%%%%%%%%%%%%%% (a)
[z,p,k] = cheb1ap(4,0.5)  ;
[A,B,C,D] = zp2ss(z,p,k);
[A1,B1,C1,D1] = lp2lp(A,B,C,D,2*pi*f1);
[b4,a4] = ss2tf(A1,B1,C1,D1);
[bz4,az4] = bilinear(b,a,fs);

%(b)
disp('--------------------Answer to question 5------------------')
disp ('________ Hd(z) when fs=40KHz___________________________')
tf(bz4,az4)
disp ('poles(5)')
[z8,p8,k8] = tf2zp(bz4,az4);
disp ('the magnitude of each pole')
p5_1 = abs(p8(1))
p5_2 = abs(p8(2))
p5_3 = abs(p8(3))
p5_4 = abs(p8(4))
zplane(z,p)
disp ('the angle of each pole')
A5_1= angle(p8(1))*57.2958
A5_2=angle(p8(2))*57.2958
A5_3=angle(p8(3))*57.2958
A5_4=angle(p8(4))*57.2958
%(c)

H4 = freqs(b,a,w);

figure(51);
subplot(2,1,1);
plot(w, abs(H4/H4(1)));
grid on;
xlabel('angular frequency')
ylabel('|H5(jw)/H5(0)|')
title('H5(jw)/H5(0)')
subplot(2,1,2);
plot(w,angle(H4/H4(1))*180/pi);
xlabel('angular frequency')
ylabel('\theta')
grid on;

[H_D4,w6] = freqz(bz4,az4);

figure(52);
subplot(2,1,1);
plot(w6, abs(H_D4/H_D4(1)));
xlabel('angular frequency')
ylabel('|H_D5(ejwT)/H_D5(1)|')
```

```
title('H_D5(ejwT)/H_D5(1)')
grid on;
subplot(2,1,2);
plot(w6,angle(H_D4/H_D4(1))*180/pi);
xlabel('angular frequency')
ylabel('\theta')
grid on;
f1_5 = atan(f1*2*pi/fs/2)*2*fs/2/pi % new value of cuttoff frequency
%% Code for (6)

W3db=10e4;
ws=4*W3db;
fs4=(ws/2/pi);
df=1;
fsp4=0:df:fs4/2;
w4=2*pi*fsp4;

num=[0 0 0 0 0 0 1];
den=[1 3.8637033 7.4641016 9.1416202 7.4641016 3.8637033 1];
[b5,a5] = lp2hp(num,den,W3db);
[bz5,az5]=bilinear(b5,a5,fs4);

H5 = freqs(b5,a5,w4);

figure(61);
subplot(2,1,1);
plot(w4/2/pi, abs(H5));
xlabel('frequency, [Hz]')
ylabel('|H5(jw)|')
title('normalized |H5(jw)/H5(3184)|')
grid on;
subplot(2,1,2);
plot(w4/2/pi,angle((H5)*180/pi));
xlabel('frequency, [Hz]')
ylabel('\theta')
grid on;

[H_D5,w7] = freqz(bz5,az5);

figure(62);
subplot(2,1,1);
plot(w7/pi/2*fs4, abs(H_D5/H_D5(512)));
xlabel('frequency, [Hz]')
ylabel('|H_D5(ejwT)|')
title('|H_D5(ejwT)/H_D5(512)|')
grid on;
subplot(2,1,2);
plot(w7/pi/2*fs4,angle((H_D5/H_D5(512))*180/pi));
xlabel('frequency, [Hz]')
ylabel('\theta')
grid on;

disp ('poles(6)')
[z9,p9,k9] = tf2zp(bz5,az5);
disp ('the magnitude of each pole')
```

```
p6_1 = abs(p9(1))
p6_2 = abs(p9(2))
p6_3 = abs(p9(3))
p6_4 = abs(p9(4))
zplane(z,p)
% new value of cutoff for digital filter calculated analytically
w3db=atan(W3db/fs4/2)*2*fs4

%(7) Prewarping
 T=1;
 ws=4*10e4;
 Ts=1/fs4;

num=[0 0 0 0 0 0 1];
den=[1 3.8637033 7.4641016 9.1416202 7.4641016 3.8637033 1];
[b5,a5] = lp2hp(num,den,W3db);

w8=2*fs4*tan(W3db/2/fs4);
fp=(w8/2/pi);

[bz6,az6]=bilinear(b5,a5,fs4,fp);

H5 = freqs(b5,a5,w4);

figure(71);
subplot(2,1,1);
plot(w4/2/pi, abs(H5));
xlabel('frequency, [Hz]')
ylabel('|H6(jw)|')
title('|H6(jw)/H6(3184)|')
grid on;
subplot(2,1,2);
plot(w4/2/pi,angle((H5)*180/pi));
xlabel('frequency, [Hz]')
ylabel('\theta')
grid on;

[H_D6,w8] = freqz(bz6,az6);

figure(72);
subplot(2,1,1);
plot(w8/pi*fs4/2, abs(H_D6/H_D6(512)));
xlabel('frequency, [Hz]')
ylabel('|H_D6(ejwT)|')
title('|H_D6(ejwT)/H_D6(512)|')
grid on;
subplot(2,1,2);
plot(w8/pi*fs4/2,angle((H_D6/H_D6(512))*180/pi));
xlabel('frequency, [Hz]')
ylabel('\theta')
grid on;
```