Digital Filter Design (EET 3134)

Lab 05: Design of IIR Filters using Impulse Invariance Method

(Submission by: 11th July 2018)

Branch: ECE			
Sl. No.	Name	Registration No.	Signature

Department of Electronics & Communication Engineering
Institute of Technical Education and Technology
(Faculty of Engineering)

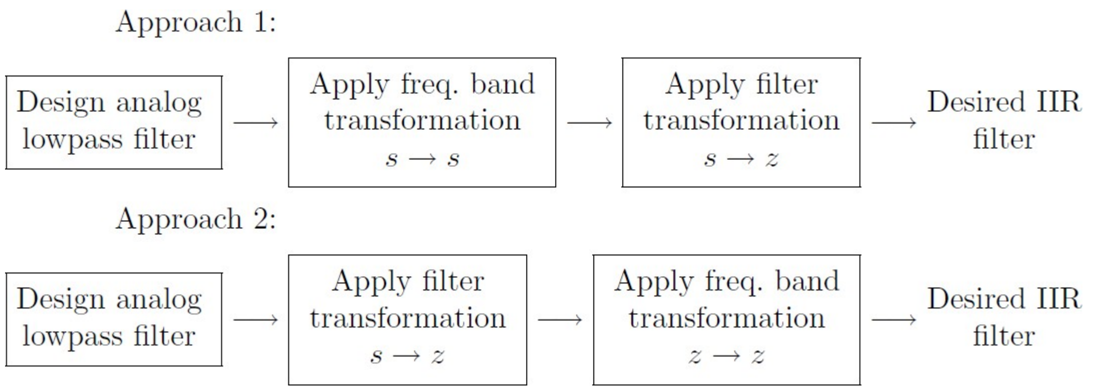
Siksha 'O' Anusandhan Deemed to be University, Bhubaneswar.

1 Introduction

IIR filters have infinite-duration impulse responses, hence they can be matched to analog filters, all of which generally have infinitely long impulse responses. Therefore the basic technique of IIR filter design transforms well-known analog filters into digital filters using complex-valued mappings. The advantage of this technique lies in the fact that both analog filter design (AFD) tables and the mappings are available extensively in the literature. This basic technique is called the A/D (analog-to-digital) filter transformation.

However, the AFD tables are available only for lowpass filters. We also want to design other frequency-selective filters (highpass, bandpass, bandstop, etc.). To do this, we need to apply frequency-band transformations to lowpass filters. These transformations are also complex-valued mappings, and they are also available in the literature.

There are two approaches to this basic technique of IIR filter design:



We will study the second approach as it involves the frequency-band transformation in the digital domain. Hence in this IIR filter design technique we will follow the following steps:

- Design analog lowpass filters.
- Study and apply filter transformations to obtain digital lowpass filters.
- Study and apply frequency-band transformations to obtain other digital filters from digital lowpass filters.

2 Prototype AnalogFilters

IIRfilterdesigntechniquesrelyonexistinganalogfilterstoobtaindigitalfilters.Wedesignate these analog filters as prototype filters. Four prototypes are widely used in practice.

- Butterworth lowpass
- Chebyshev lowpass (Type I)
- Chebyshev lowpass (Type II) and
- Elliptic lowpass.

All the above prototype filters require the following specifications

- A_p in dB
- A_s in dB
- Ω_p in Hz
- Ω_s in Hz

The prototype filters are represented in s domain and its transfer function is represented as $H_c(s)$

3 Transformation of continuous-time filters to discrete-time IIR filters

Several procedures for conversion of continuous-time filters (analog prototypes) to discrete-time filters have been developed over the years. However, we only discuss two methods: impulse-invariance transformation and bilinear transformation. The first has limited applicability, but it has educational value; the second has universal applicability and it is the most widely used method in IIR filter design software packages. Each transformation is equivalent to a mapping function $s = f(z)$ from the s-plane to the z-plane. Any useful mapping should satisfy three desirable conditions:

(a) A rational $H_c(s)$ should be mapped to a rational $H(z)$ (realizability)

- Rational $H_c(s) \rightarrow$ Rational $H(z)$.

(b) The imaginary axis of the s-plane is mapped on the unit circle of the z-plane.

- $\{s = j\Omega \mid -\infty < \Omega < \infty\} \rightarrow \{z = e^{j\omega} \mid -\pi < \omega < \pi\}$

(c) The left-half s-plane is mapped into the interior of the unit circle of the z-plane:

- $\{s \mid Re(s) < 0\} \rightarrow \{z||z| < 1\}$..

Condition (a) is needed to preserve the frequency characteristics of the continuous time filter. Condition (b) guarantees that a stable continuous-time filter is mapped into a stable discrete-time filter. Any mapping procedure must satisfy (c). The reason is that stable continuous-time systems have their poles on the left-half s-plane, whereas stable discrete-time systems have their poles inside the unit circle of the z-plane as shown in Fig.1. Clearly, different procedures give rise to different mapping functions, and, hence, the resulting discrete-time filters are different.

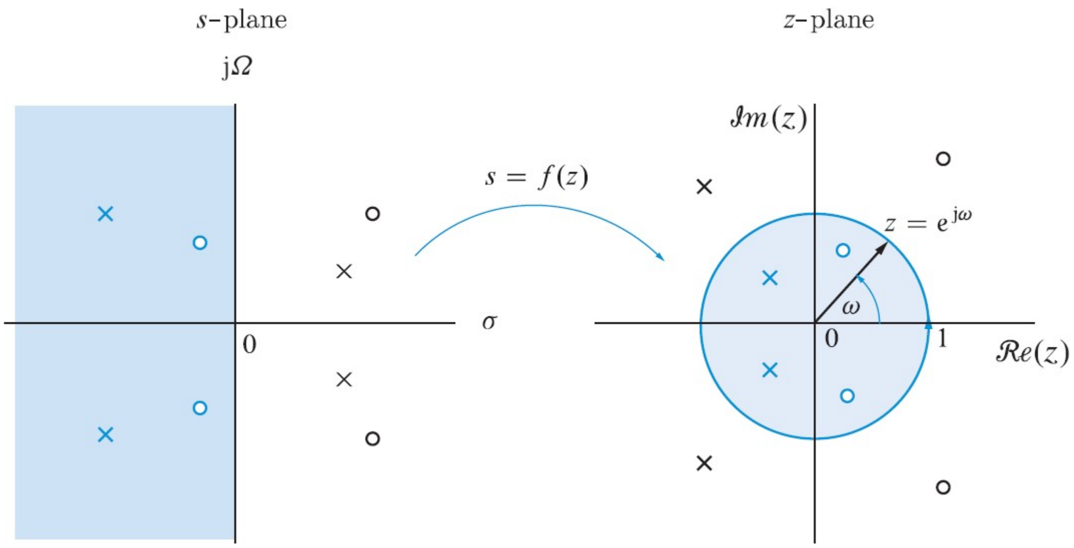


Figure 1: Two desirable requirements, (b) and (c), for functions of the form $s = f(z)$ that map the continuous-time s-plane to the discrete-time z-plane.

4 Impulse-invariance transformation

The most natural way to convert a continuous-time filter to a discrete-time filter is by sampling its impulse response

$$h(n) = h_c(nT) \tag{1}$$

where T is called the design sampling period.

The parameter T is chosen so that the shape of $h_c(t)$ is "captured" by the samples. Since this is a sampling operation, the analog and digital frequencies are related by

$$\omega = \Omega T \Rightarrow e^{j\omega} = e^{j\Omega T} \tag{2}$$

Since $z = e^{j\omega}$ on the unit circle and $s = j\Omega$ on the imaginary axis, we have the following transformation from the s-plane to the z-plane:

$$z = e^{sT} \tag{3}$$

The system functions $H(z)$ and $H_c(s)$ are related through the frequency domain aliasing formula

$$H(z) = \frac{1}{T}\sum_{k=-\infty}^{\infty} H_c\left(s - j\frac{2\pi}{T}k\right) \tag{4}$$

This transformation is known as impulse-invariance because it preserves the shape of the impulse response. The frequency response of the resulting discrete-time filter is related to the frequency response of the continuous time filter by

$$H\left(e^{j\omega}\right) = \sum_{k=-\infty}^{\infty} H_c\left(j\frac{\omega}{T} + j\frac{2\pi}{T}k\right) \tag{5}$$

Thus, in general, the impulse-invariance mapping causes aliasing, as illustrated in Fig.2. The fundamental difference between continuous-time and discrete-time filters is the periodicity of frequency-response for discrete-time systems, that is, $H\left(e^{j\omega}\right)$ is periodic whereas $H_c(j\Omega)$ is nonperiodic.

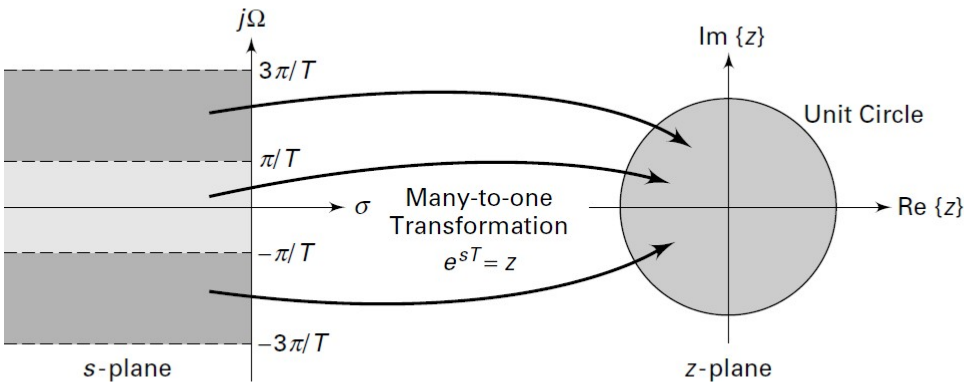


Figure 2: Complex-plane mapping in impulse-invariance transformation.

5 Design Procedure

Given the digital lowpass filter specifications ω_p, ω_s, A_p, and A_s, we want to determine $H(z)$ by first designing an equivalent analog filter and then mapping it into the desired digital filter. The steps required for this procedure are

I. Choose T and determine the analog frequencies

$$\Omega_p = \frac{\omega_p}{T}, \quad \Omega_s = \frac{\omega_s}{T} \tag{6}$$

II. Design an analog filter $H_c(s)$ using the specifications Ω_p, Ω_s, A_p, and A_s. This can be done using any one of the four (Butterworth, Chebyshev - I, Chebyshev - II, or elliptic) prototypes.

III. Using partial fraction expansion, expand $H_c(s)$ into

$$H_c(s) = \sum_{k=1}^{N} \frac{R_k}{s - p_k} \tag{7}$$

IV. Now transform analog poles $\{p_k\}$ into digital poles $\{e^{p_k T}\}$ to obtain the digital filter:

$$H(z) = \sum_{k=1}^{N} \frac{R_k}{1 - e^{p_k T} z^{-1}} \tag{8}$$

V. Check whether the filter satisfies the specification.

6 Prelab

Prior to beginning this lab, you must have carefully read over this lab in order to plan how to implement the tasks assigned. Make sure you are aware of all the items you need to include in your lab report.

7 Lab Assignments

7.1 Assignment 01

Write a MATLAB function to design IIR filter design using impulse-invariance method.The function should have the following syntax.

```
function [b,a] = imp_invr(c,d,T)
% Impulse Invariance Transformation from Analog to Digital Filter
% ---------------------------------------------------------------
% [b,a] = imp_invr(c,d,T)
% b = Numerator polynomial in z^(-1) of the digital filter
% a = Denominator polynomial in z^(-1) of the digital filter
% c = Numerator polynomial in s of the analog filter
% d = Denominator polynomial in s of the analog filter
% T = Sampling (transformation) parameter
```

Solution

```
function[b,a]=imp_invr(cs,ds,t)        %impulse invariance
[z,p,k]=residue(cs,ds);
p=exp(p*t);                            %Expression to convert from s->z
[b,a]=residuez(z,p,k);
b=real(b');
a=real(a');
end
```

Results-

```
[b,a]=imp_invr([1,1,1], [2,2,2], 1)

b =
   0.5000
  -0.3929
   0.1839

a =
   1.0000
  -0.7859
   0.3679
```

7.2 Assignment 02

Design a lowpass digital filter using a Butterworth prototype to satisfy

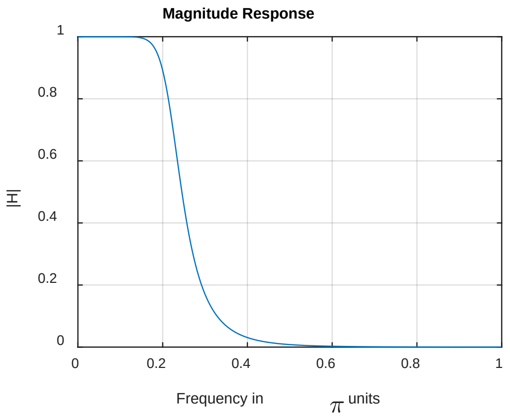
$$\omega_p = 0.2\pi,\ \omega_s = 0.3\pi,\ A_p = 1dB, \text{ and } A_s = 15dB$$

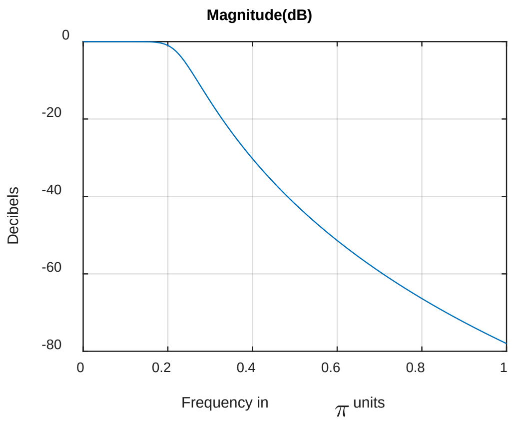
Solution

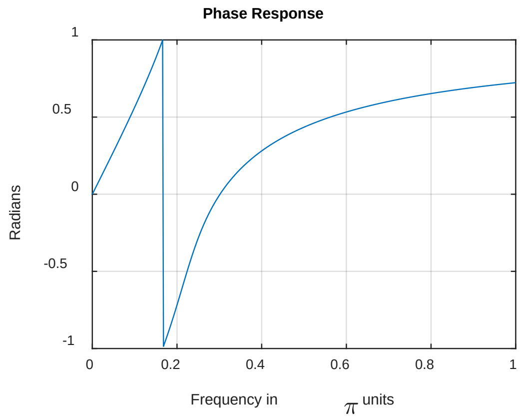
CODE

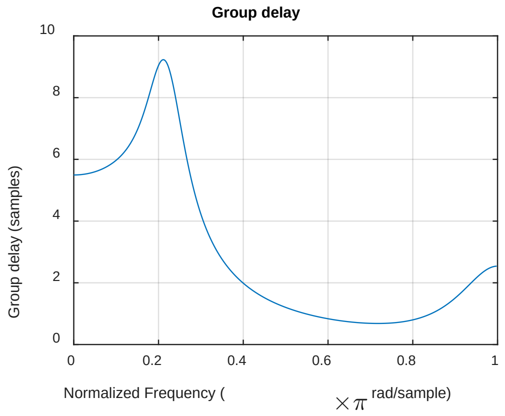
```
clc;
close all;
clear all;
% Parameter initialization
wp=0.2*pi;          %passband frequency
ws=0.3*pi;          %stopband frequency
Ap=1;               %passband ripple in dB
As=15;              %stopband ripple in dB
% Analog prototype specification
t=1;
Wp=wp/t;            % analog passband freq
Ws=ws/t;            % analog stopband freq
% order and cutoff freq calculation
N=ceil(log10((10^(Ap/10)-1)/(10^(As/10)-1))/(2*log10(Wp/Ws)))
Wc=Wp/((10^(Ap/10)-1)^(1/(2*N)))
% Analog filter design
[z,p,k]=buttap(N);   %BUTTERWORTH PROTOTYPE FILTER
p=p*Wc;
k=k*(Wc^N);
B=real(poly(z));
b0=k;
cs=k*B;
ds=real(poly(p));
% transformation from s->z
Fs=1/t;
[b,a]=imp_invr(cs,ds,t)
% plots
% Calculation of Frequency Response:
w = [0:1:500]*pi/500; H = freqs(cs,ds,w);
mag = abs(H); db = 20*log10((mag+eps)/max(mag)); pha = angle(H);
figure;set(gcf, 'Position', [0, 0, 700, 700])
subplot(322);plot(w/pi,db);grid on;title('Magnitude(dB)');xlabel('Frequency in \pi units');ylabel('Decibels');
subplot(321);plot(w/pi,mag);grid on;title('Magnitude Response');xlabel('Frequency in \pi units');ylabel('|H|');
subplot(323);plot(w/pi,-pha/pi);grid on;title('Phase Response');xlabel('Frequency in \pi units');ylabel('Radians');
% Calculation of Impulse response:
subplot(325);impulse(cs,ds);grid on
subplot(324);grpdelay(b,a);grid on;title('Group delay')
```

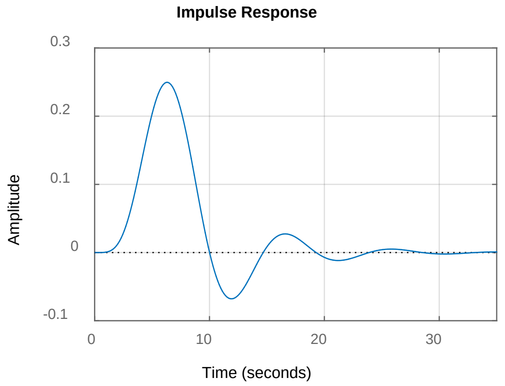
PLOT


Magnitude Response
1
0.8
0.6
0.4
0.2
0
|H|
0
0.2
0.4
0.6
0.8
1
Frequency in π units


Magnitude(dB)
0
-20
-40
-60
-80
Decibels
0
0.2
0.4
0.6
0.8
1
Frequency in π units


Phase Response
1
0.5
0
-0.5
-1
Radians
0
0.2
0.4
0.6
0.8
1
Frequency in π units


Group delay
10
8
6
4
2
0
Group delay (samples)
0
0.2
0.4
0.6
0.8
1
Normalized Frequency (×π rad/sample)


Impulse Response
0.3
0.2
0.1
0
-0.1
Amplitude
0
10
20
30
Time (seconds)

7.3 Assignment 03

Design a lowpass digital filter using a Chebyshev - I prototype to satisfy

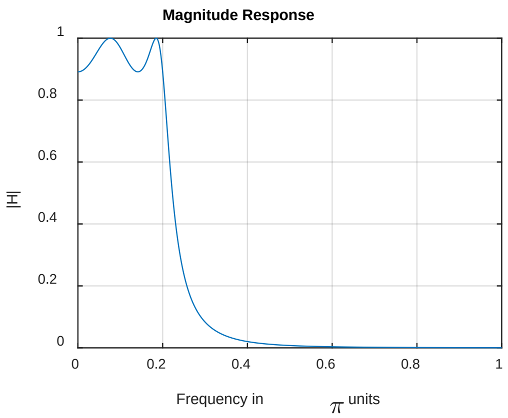
$$\omega_p = 0.2\pi,\ \omega_s = 0.3\pi,\ A_p = 1dB, \text{ and } A_s = 15dB$$

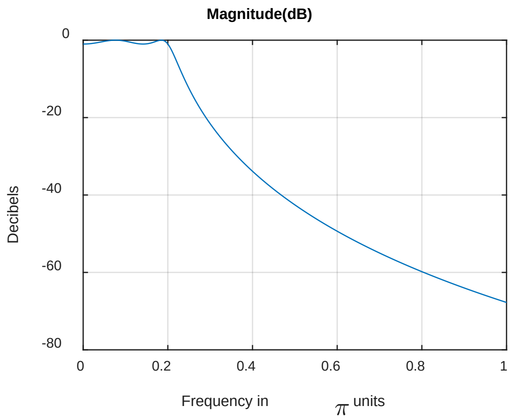
Solution

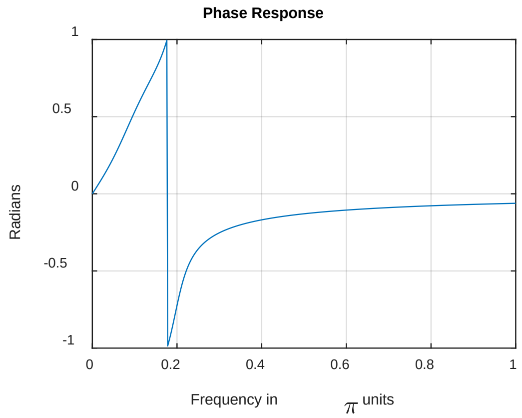
CODE

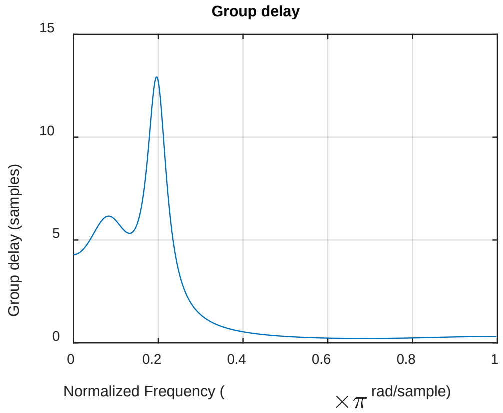
```
clc;clear all;close all;
% Parameter initialization
wp=0.2*pi
ws=0.3*pi
As=15
Ap=1
% Analog prototype specification
T=1;
Wp=wp/T;
Ws=ws/T;
Wc=Wp
Wr=Ws/Wp
E=sqrt(10^(0.1*Ap)-1)
A=10^(As/20)
g=sqrt((A^2-1)/E^2)
% order
N=ceil((log10(g+sqrt(g^2-1)))/(log10(Wr+sqrt(Wr^2-1))))
% Analog filter design
[z,p,k]=cheb1ap(N,Ap)
ds1=real(poly(p))
dNn=ds1(N+1)
p=p*Wc
ds=real(poly(p))
dNu=ds(N+1)
k=k*(dNu/dNn)
B=real(poly(z))
cs=k*B
[b,a] = imp_invr(cs,ds,T);
% plots
% Calculation of Frequency Response:
w = [0:1:500]*pi/500; H = freqs(cs,ds,w);
mag = abs(H); db = 20*log10((mag+eps)/max(mag)); pha = angle(H);figure;
figure;set(gcf, 'Position', [0, 0, 700, 900])
subplot(322);plot(w/pi,db);grid on;title('Magnitude(dB)');xlabel('Frequency in \pi
units');ylabel('Decibels');
subplot(321);plot(w/pi,mag);grid on;title('Magnitude Response');xlabel('Frequency in \pi
units');ylabel('|H|');
subplot(323);plot(w/pi,-pha/pi);grid on;title('Phase Response');xlabel('Frequency in \pi
units');ylabel('Radians');
% Calculation of Impulse response:
subplot(325);impulse(cs,ds);grid on
subplot(324);grpdelay(b,a);grid on;title('Group delay')
```

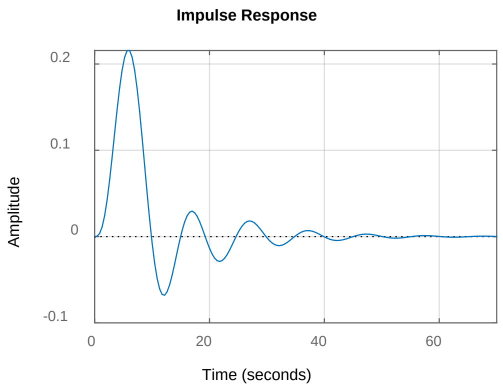
PLOT


Magnitude Response
|H|
1
0.8
0.6
0.4
0.2
0
0
0.2
0.4
0.6
0.8
1
Frequency in π units


Magnitude(dB)
Decibels
0
-20
-40
-60
-80
0
0.2
0.4
0.6
0.8
1
Frequency in π units


Phase Response
Radians
1
0.5
0
-0.5
-1
0
0.2
0.4
0.6
0.8
1
Frequency in π units


Group delay
Group delay (samples)
15
10
5
0
0
0.2
0.4
0.6
0.8
1
Normalized Frequency (×π rad/sample)


Impulse Response
Amplitude
0.2
0.1
0
-0.1
0
20
40
60
Time (seconds)

7.4 Assignment 04

Design a lowpass digital filter using a Chebyshev - II prototype to satisfy

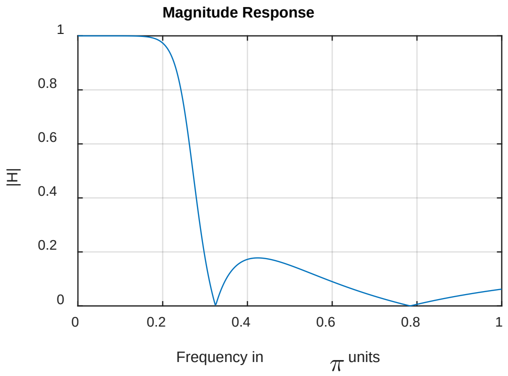
$$\omega_p = 0.2\pi,\ \omega_s = 0.3\pi,\ A_p = 1dB, \text{ and } A_s = 15dB$$

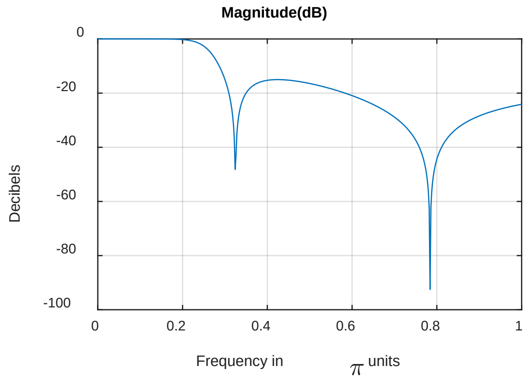
Solution

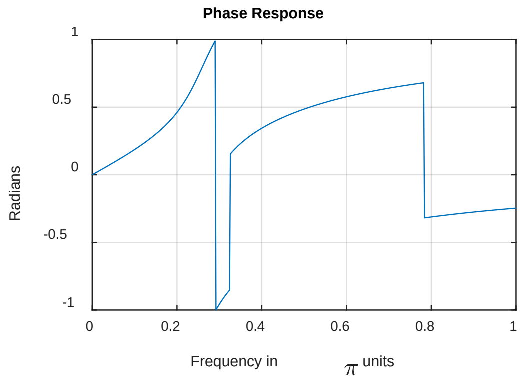
CODE

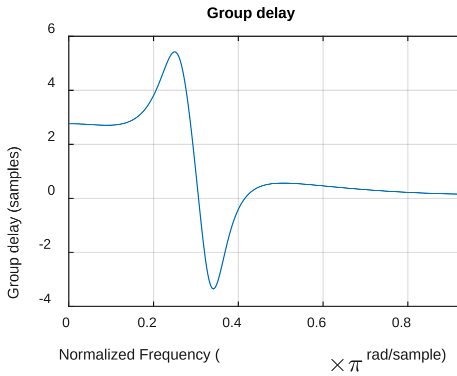
```
clc;clear all;close all;
% Parameter initialization
wp=0.2*pi;
ws=0.3*pi;
As=15;
Ap=1;
T=1;
Wp=wp/T;
Ws=ws/T;
Wc=Wp;
% Analog prototype specification
Wr=Ws/Wp;
E=sqrt(10^(0.1*Ap)-1)
A=10^(As/20)
g=sqrt((A^2-1)/E^2)
N=ceil((log10(g+sqrt(g^2-1)))/(log10(Wr+sqrt(Wr^2-1))))
% Analog filter design
[z,p,k]=cheb2ap(N,As)
ds = real(poly(p)); aNn = ds(N+1);
p = p*Ws; ds = real(poly(p)); aNu = ds(N+1);
cs = real(poly(z)); M = length(cs); bNn = cs(M);
z = z*Ws; cs = real(poly(z)); bNu = cs(M);
k = k*(aNu*bNn)/(aNn*bNu);
b0 = k; cs = k*cs;
[b,a] = imp_invr(cs,ds,T);
% plots
% Calculation of Frequency Response:
w = [0:1:500]*pi/500; H = freqs(cs,ds,w);
mag = abs(H); db = 20*log10((mag+eps)/max(mag)); pha = angle(H);figure;
figure;set(gcf, 'Position', [0, 0, 700, 700])
subplot(322);plot(w/pi,db);grid on;title('Magnitude(dB)');xlabel('Frequency in \pi
units');ylabel('Decibels');
subplot(321);plot(w/pi,mag);grid on;title('Magnitude Response');xlabel('Frequency in \pi
units');ylabel('|H|');
subplot(323);plot(w/pi,-pha/pi);grid on;title('Phase Response');xlabel('Frequency in \pi
units');ylabel('Radians');
% Calculation of Impulse response:
subplot(325);impulse(cs,ds);grid on
subplot(324);grpdelay(b,a);grid on;title('Group delay')
```

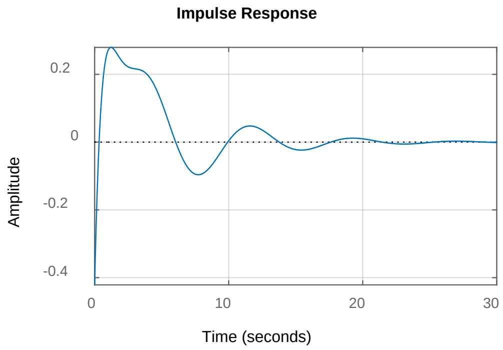
PLOT


Magnitude Response
1
0.8
0.6
0.4
0.2
0
|H|
0
0.2
0.4
0.6
0.8
1
Frequency in π units


Magnitude(dB)
0
-20
-40
-60
-80
-100
Decibels
0
0.2
0.4
0.6
0.8
1
Frequency in π units


Phase Response
1
0.5
0
-0.5
-1
Radians
0
0.2
0.4
0.6
0.8
1
Frequency in π units


Group delay
6
4
2
0
-2
-4
Group delay (samples)
0
0.2
0.4
0.6
0.8
1
Normalized Frequency ($\times\pi$ rad/sample)


Impulse Response
0.2
0
-0.2
-0.4
Amplitude
0
10
20
30
Time (seconds)

7.5 Assignment 05

Design a lowpass digital filter using a elliptic prototype to satisfy

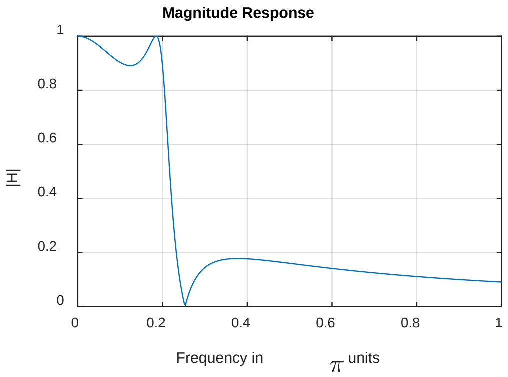
$$\omega_p = 0.2\pi,\ \omega_s = 0.3\pi,\ A_p = 1dB, \text{ and } A_s = 15dB$$

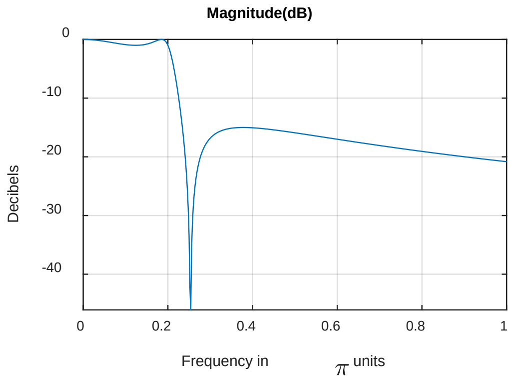
Solution

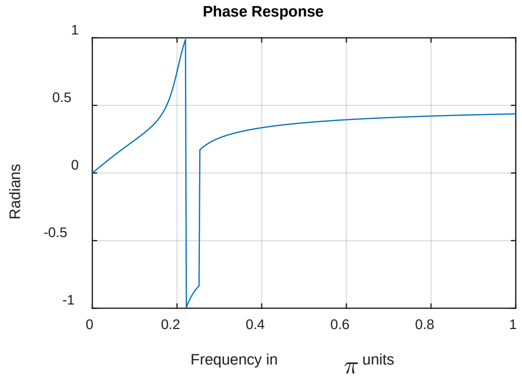
CODE

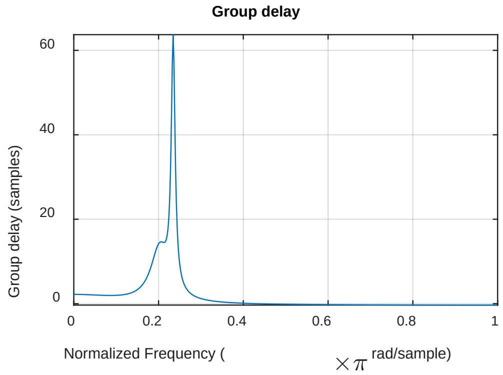
```
clc;clear all;close all;
% Parameter initialization
wp=0.2*pi;
ws=0.3*pi;
As=15;
Ap=1;
T=1;
Wp=wp/T;
Ws=ws/T;
Wc=Wp;
% Analog prototype specification
k=Wp/Ws;
E=sqrt(10^(0.1*Ap)-1);
A=10^(As/20);
k1=E/(sqrt(A^2-1));
N=ceil((ellipke(k)*ellipke(sqrt(1-k1^2)))/(ellipke(k1)*ellipke(sqrt(1-k^2))))
% Analog filter design
[z,p,k]=ellipap(N,Ap,As)
ds=real(poly(p));
dNn=ds(N+1);
p=p*Wc;
ds=real(poly(p));
dNu=ds(N+1);
cs=real(poly(z));
cNn=cs(end)
z=z*Wp;
cs=real((poly(z)));
cNu=cs(end)
k=k*(dNu/dNn)*(cNn/cNu);
cs=k*cs;
[b,a] = imp_invr(cs,ds,T);
%plots
% Calculation of Frequency Response:
w = [0:1:500]*pi/500; H = freqs(cs,ds,w);
mag = abs(H); db = 20*log10((mag+eps)/max(mag)); pha = angle(H);
figure;set(gcf, 'Position', [0, 0, 700, 700])
subplot(322);plot(w/pi,db);grid on;title('Magnitude(dB)');xlabel('Frequency in \pi units');ylabel('Decibels');
subplot(321);plot(w/pi,mag);grid on;title('Magnitude Response');xlabel('Frequency in \pi units');ylabel('|H|');
subplot(323);plot(w/pi,-pha/pi);grid on;title('Phase Response');xlabel('Frequency in \pi units');ylabel('Radians');
% Calculation of Impulse response:
subplot(325);impulse(cs,ds);grid on
subplot(324);grpdelay(b,a);grid on;title('Group delay')
```

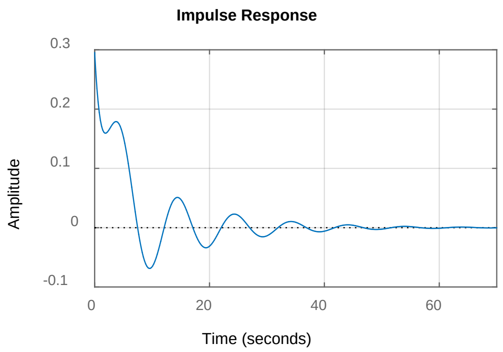
PLOT


Magnitude Response
|H|
1
0.8
0.6
0.4
0.2
0
0
0.2
0.4
0.6
0.8
1
Frequency in π units


Magnitude(dB)
Decibels
0
-10
-20
-30
-40
0
0.2
0.4
0.6
0.8
1
Frequency in π units


Phase Response
Radians
1
0.5
0
-0.5
-1
0
0.2
0.4
0.6
0.8
1
Frequency in π units


Group delay
Group delay (samples)
60
40
20
0
0
0.2
0.4
0.6
0.8
1
Normalized Frequency ($\times\pi$ rad/sample)


Impulse Response
Amplitude
0.3
0.2
0.1
0
-0.1
0
20
40
60
Time (seconds)

7.6 Assignment 06

Compare the performances of all the filters designed in this assignment set. Write suitable discussion and draw a proper conclusion from the same.

Solution