Marwin B. Alejo 2020-20221 EE274_ProgEx07

Table of Contents

Also accessible through http://www.github.com/soymarwin/ee274/EE274_ProgEx07 for history tracking.

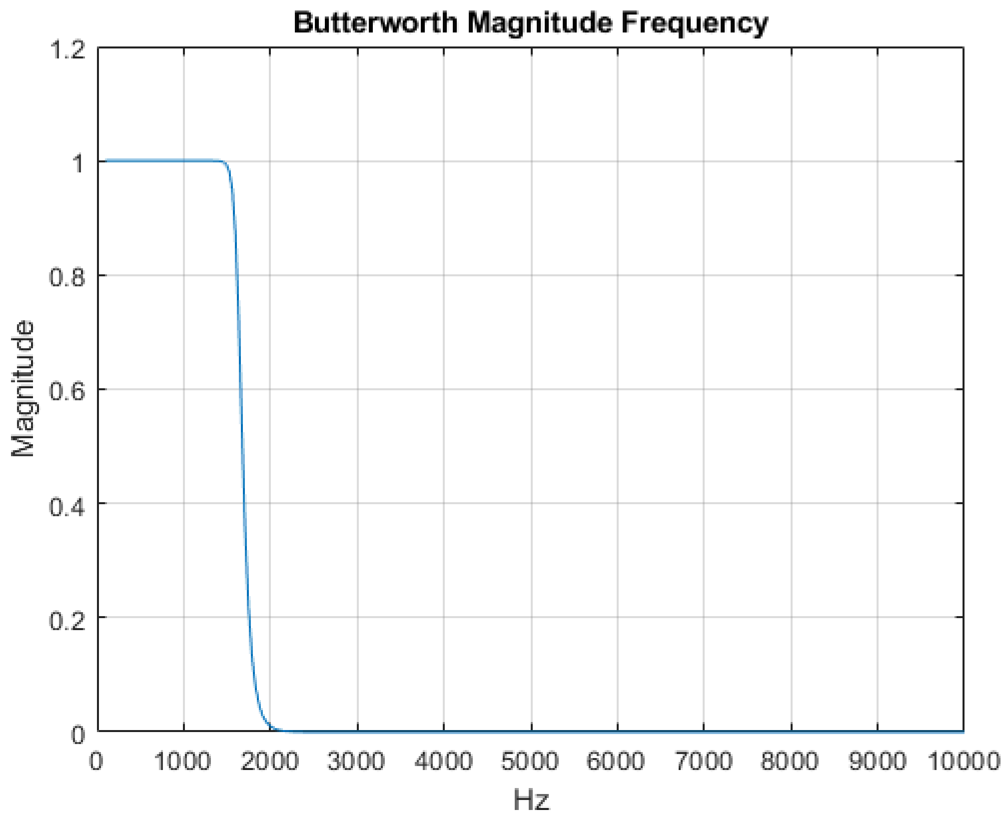
1. ANALOG FILTER DESIGN

a. Butterworth Filter

```
wp_1a=1500;
ws_1a=2000;
rp_1a=-20*log10(0.99);
rs_1a=-20*log10(0.01);
[n_1a,wc_1a]=buttord(wp_1a, ws_1a, rp_1a, rs_1a,'s');
[num_1a,den_1a]=butter(n_1a,wc_1a,'s');
[H_1a,w_1a]=freqs(num_1a,den_1a);
```

```
fprintf('The lowest-order Butterworth filter that satisifies the
 specification is %d. \n',n_1a);
figure();plot(w_1a,abs(H_1a));title('Butterworth Magnitude
 Frequency');grid on;xlabel('Hz');ylabel('Magnitude');
```

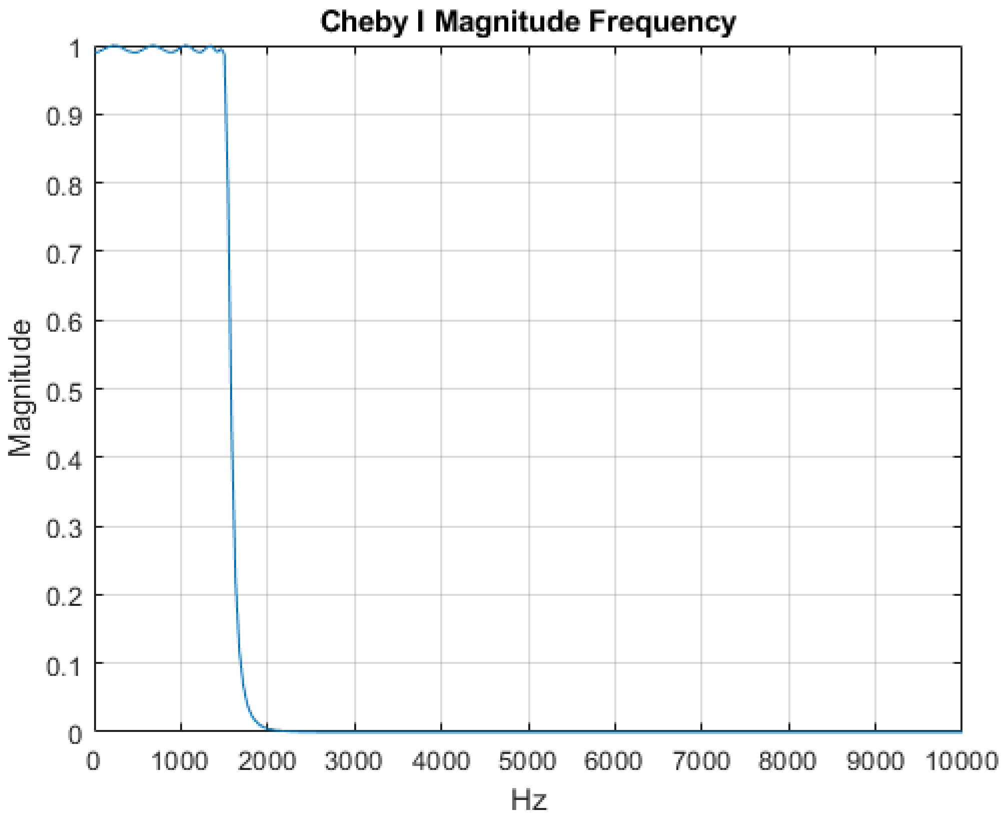
The lowest-order Butterworth filter that satisifies the specification is 23.



b. Chebyshev1 Filter

```
[n_1b,wc_1b]=cheb1ord(wp_1a, ws_1a, rp_1a, rs_1a,'s');
[num_1b,den_1b]=cheby1(n_1b,rp_1a,wc_1b,'s');
[H_1b,w_1b]=freqs(num_1b,den_1b);
fprintf('The lowest-order Chebyshev Type-I filter that satisifies the
 specification is %d. \n',n_1b);
figure(); plot(w_1b,abs(H_1b));title('Cheby I Magnitude
 Frequency');grid on;xlabel('Hz');ylabel('Magnitude');
```

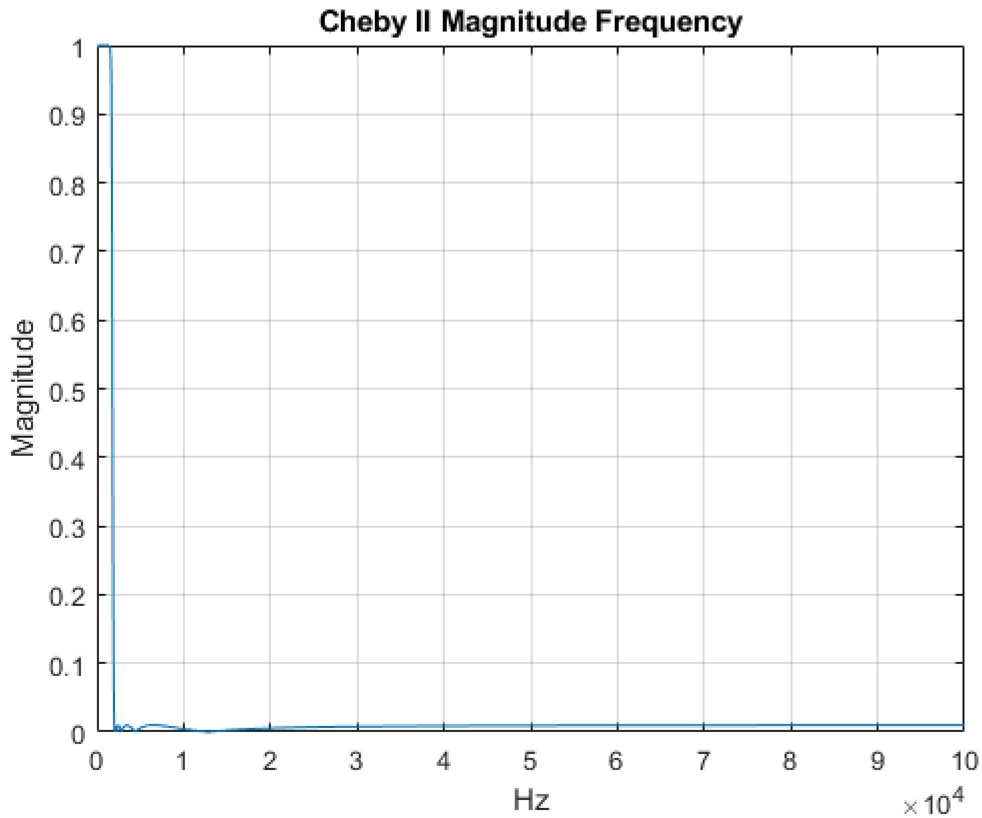
The lowest-order Chebyshev Type-I filter that satisifies the specification is 10.



c. Chebyshev2 Filter

```
[n_1c,wc_1c]=cheb2ord(wp_1a, ws_1a, rp_1a, rs_1a,'s');
[num_1c,den_1c]=cheby2(n_1b,rs_1a,ws_1a,'s');
[H_1c,w_1c]=freqs(num_1c,den_1c);
fprintf('The lowest-order Chebyshev Type-II filter that satisifies the
 specification is %d. \n',n_1b);
figure(); plot(w_1c,abs(H_1c));title('Cheby II Magnitude
 Frequency');grid on;xlabel('Hz');ylabel('Magnitude');
```

```
The lowest-order Chebyshev Type-II filter that satisifies the
 specification is 10.
```



d. Results Comparison

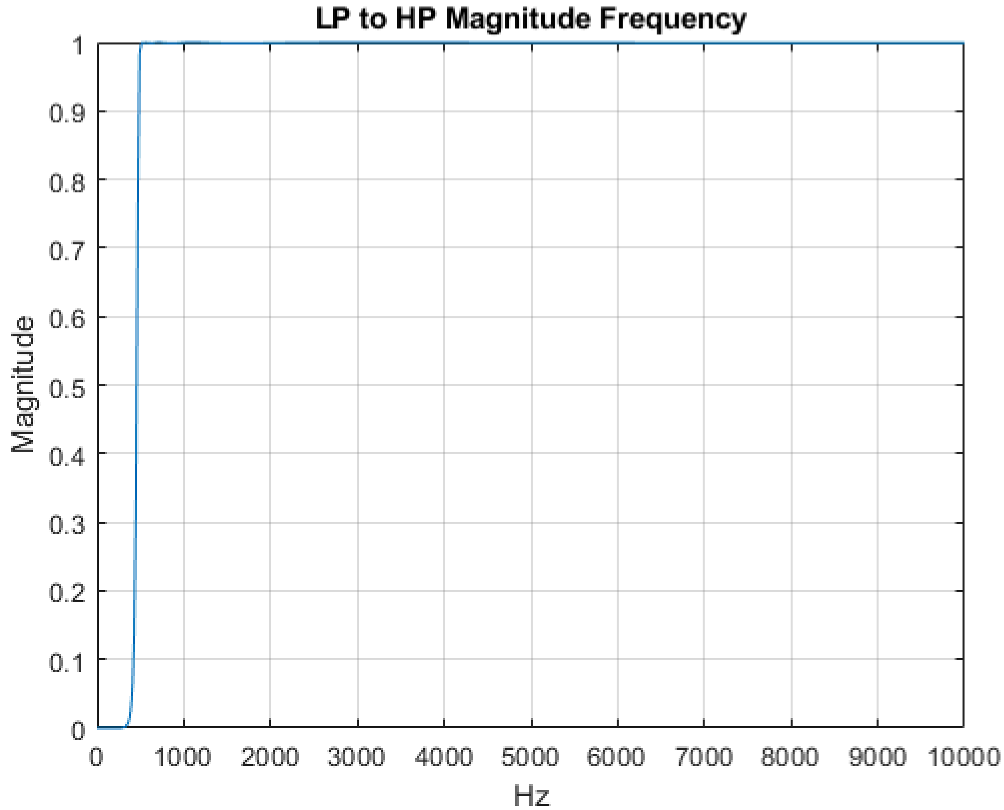
All the analog filters yield a lowpass filter except that each differ in terms of their order, transition width region, passband ripple, and stopband ripple. The Butterworth filter, in cosideration of the given specification, yields a lowpass filter of 23rd order and its passband and stopband ripple not visible by inspection. As with the Chebyshev Type-I filter, a 10th-order lowpass filter was created with its ripple band visible to the naked eye by inspection and not with its stopband ripples. In contrary to Chebyshev type-1, Chebyshev Type-2 yields the opposite of type-1. Althought a 10th-order lowpass filter was created, its ripple in its stopband is visible by inspection while its passband ripple isn't. Additionally, the transition width region of each generated lpf are the same.

2. ANALOG FILTER TRANSFORMATION

```
[z2,p2,k2]=cheb1ap(n_1b,0.01);
[num2,den2]=zp2tf(z2,p2,k2);
```

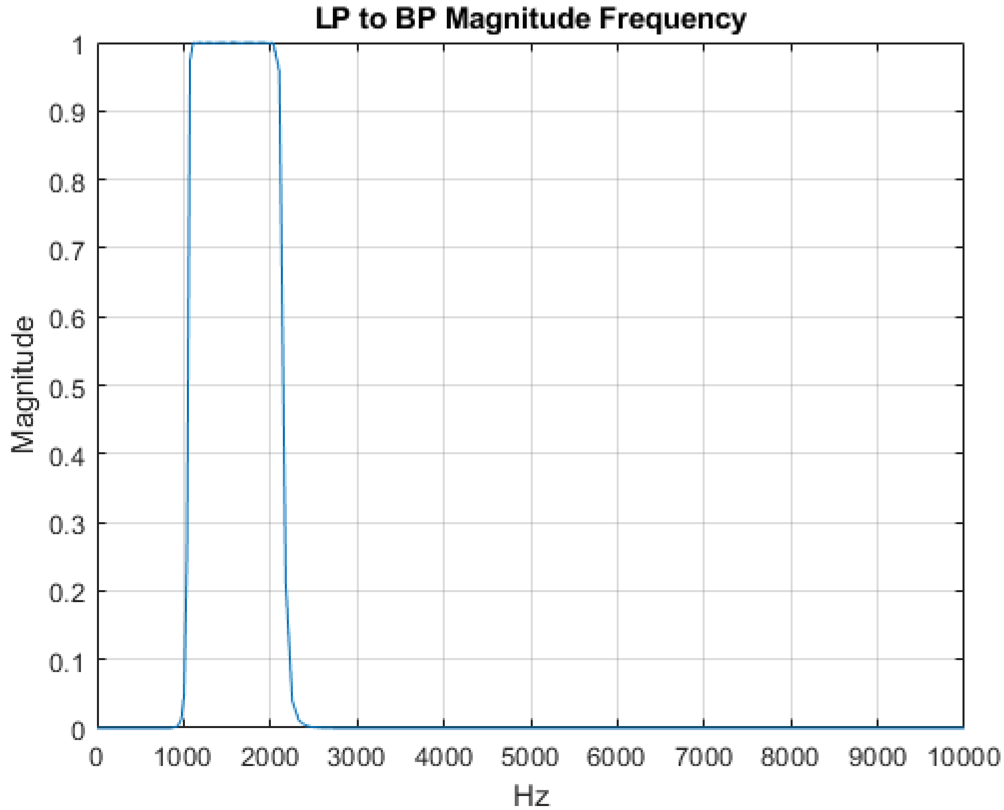
a. LP to HP

```
[num_2a,den_2a]=lp2hp(num2,den2,500);
[hh_2a,ah_2a]=freqs(num_2a,den_2a);
figure(); plot(ah_2a,abs(hh_2a)); grid on; title('LP to HP Magnitude
 Frequency');xlabel('Hz');ylabel('Magnitude');
```



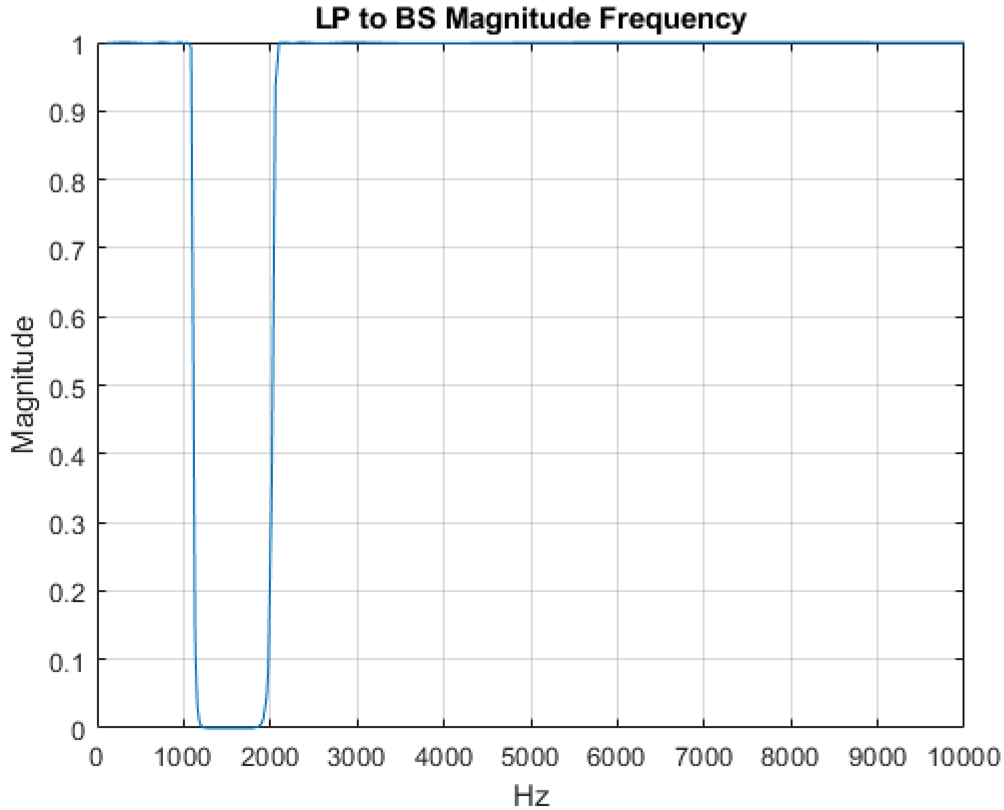
b. LP to BP

```
[num_2b,den_2b]=lp2bp(num2,den2,1500,1000);
[hh_2b,ah_2b]=freqs(num_2b,den_2b);
figure(); plot(ah_2b,abs(hh_2b)); grid on; title('LP to BP Magnitude
 Frequency');xlabel('Hz');ylabel('Magnitude');
```



c. LP to BS

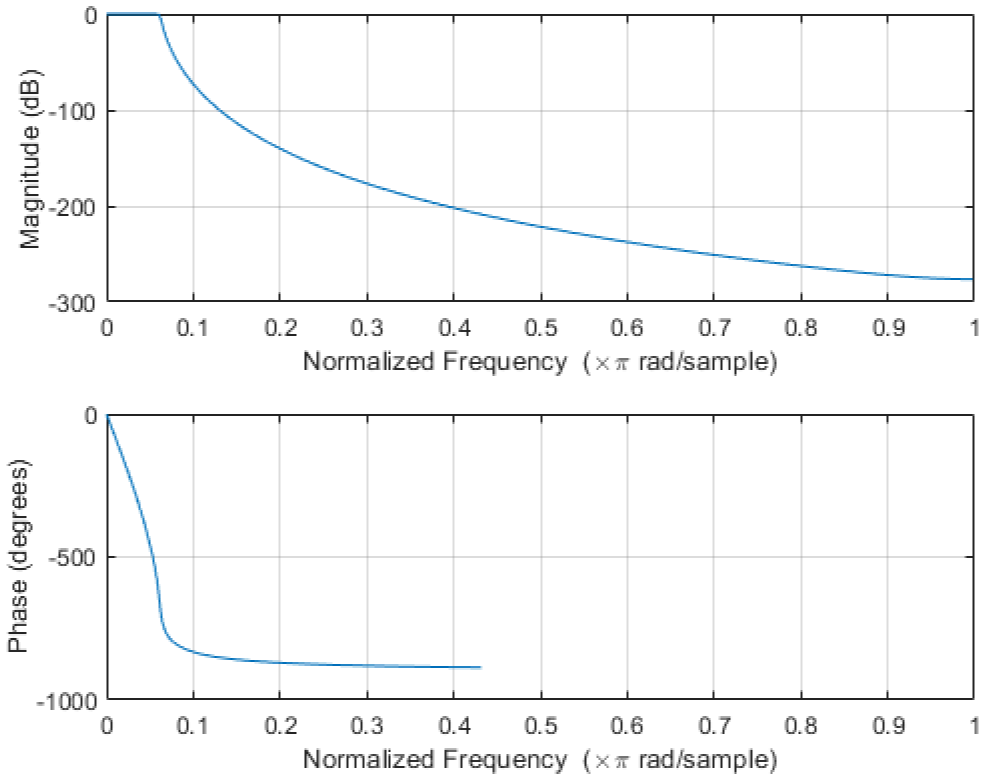
```
[num_2c,den_2c]=lp2bs(num2,den2,1500,1000);
[hh_2c,ah_2c]=freqs(num_2c,den_2c);
figure(); plot(ah_2c,abs(hh_2c)); grid on; title('LP to BS Magnitude
 Frequency');xlabel('Hz');ylabel('Magnitude');
```



3. ANALOG TO DIGITAL FILTER TRANSFORMATION

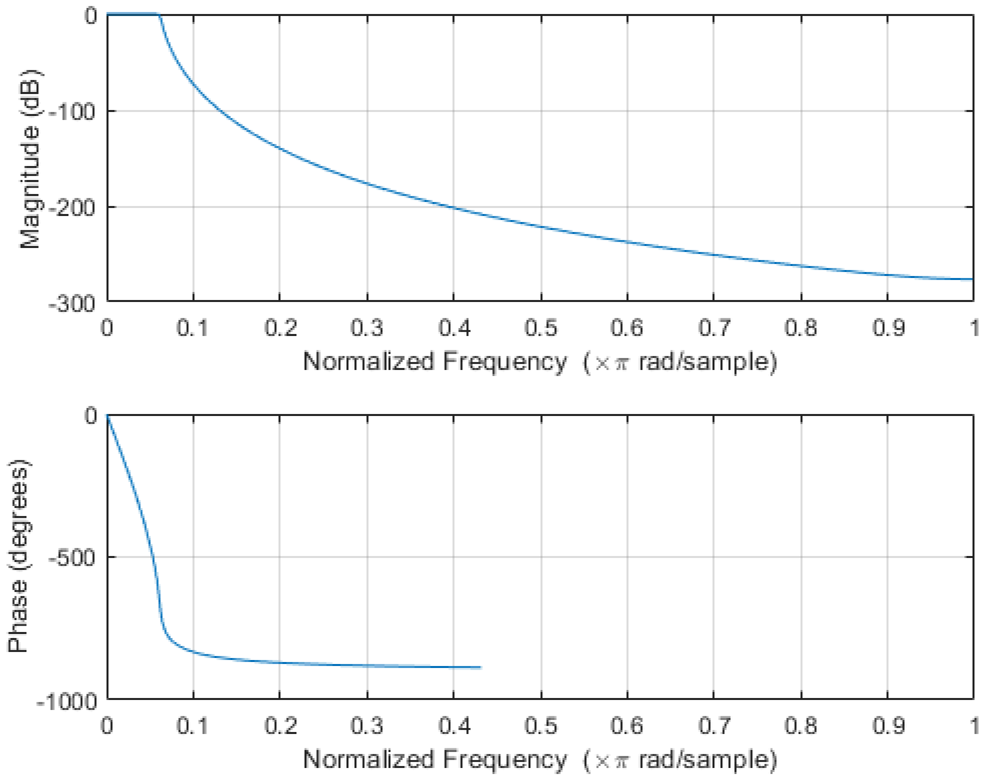
a. Impulse-Invariance

```
[num_3a,den_3a]=impinvar(num_1b,den_1b,8000);
figure(); freqz(num_3a,den_3a);
```



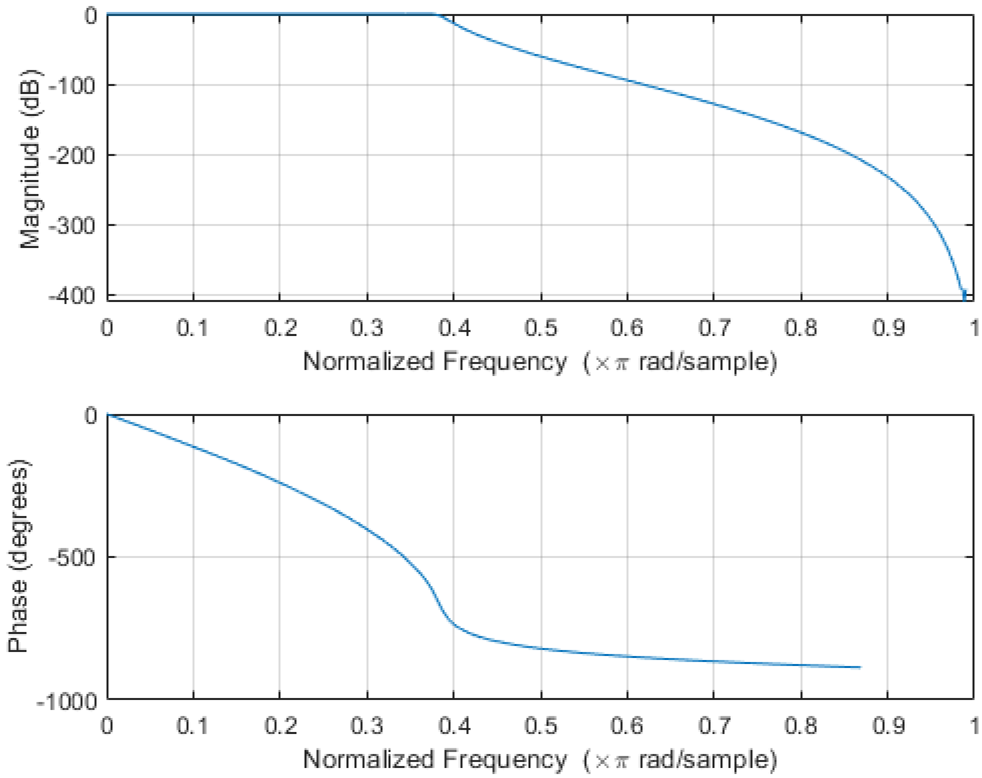
b. Bilinear

```
[num_3b,den_3b]=bilinear(num_1b,den_1b,8000);
figure(); freqz(num_3a,den_3a);
```



c. Digital Chebyshev Type-I

```
[n_3c,wc_3c]=cheb1ord(wp_1a/4000, ws_1a/4000, rp_1a, rs_1a);
[num_3c,den_3c]=cheby1(n_1b,rp_1a,wp_1a/4000);
figure();freqz(num_3c,den_3c);
```



Observation:

The magnitude and phase frequency yielded by A2D techiniques (impinvar() and bilinear() are not the same with their digital counterpart. Although, the phase responses of A2D filters are similar to the phase reposne of the digital filter by a factor of ~4*8000Hz. Hence, we can say that MATLAB, by default, considers the cut-off frequency in designing an IIR filter from analog filters. Also, LPF is the default IIR filter in MATLAB.

4. DIGITAL IIR DESIGN

a. Lowest-order DIIR

```
wp_4a1=8000/20000; ws_4a1=16000/20000; rp_4a1=0.2; rs_4a1=60;
```

a.1. Butterworth

```
[n_4a1,wn_4a1]=buttord(wp_4a1,ws_4a1,rp_4a1,rs_4a1);
fprintf('The lowest-order Butterworth filter that satisifies the
 specification is %d. \n',n_4a1);
```

The lowest-order Butterworth filter that satisifies the specification is 6.

a.2. Chebyshev Type-I

```
[n_4a2,wn_4a2]=cheb1ord(wp_4a1,ws_4a1,rp_4a1,rs_4a1);
fprintf('The lowest-order Cheby Type 1 filter that satisifies the
 specification is %d. \n',n_4a2);
```

The lowest-order Cheby Type 1 filter that satisifies the specification is 5.

a.3. Chebyshev Type-II

```
[n_4a3,wn_4a3]=cheb2ord(wp_4a1,ws_4a1,rp_4a1,rs_4a1);
fprintf('The lowest-order Cheby Type 2 filter that satisifies the
 specification is %d. \n',n_4a3);
```

The lowest-order Cheby Type 2 filter that satisifies the specification is 5.

a.4. Elliptic

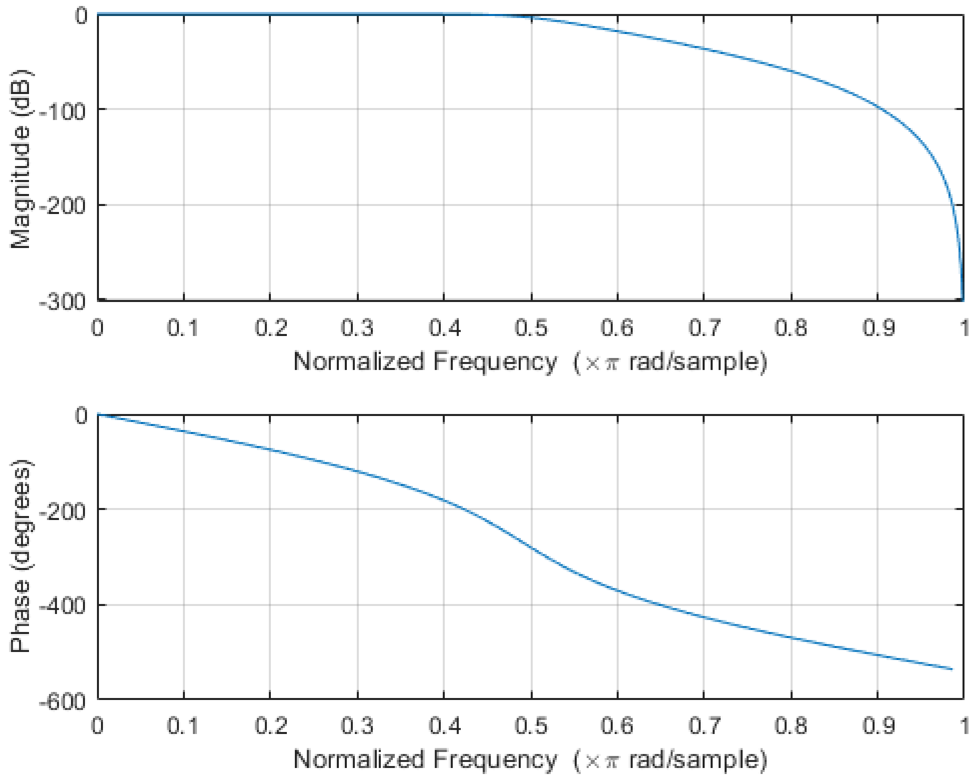
```
[n_4a4,wn_4a4]=elliord(wp_4a1,ws_4a1,rp_4a1,rs_4a1);
fprintf('The lowest-order Elliptic filter that satisifies the
 specification is %d. \n',n_4a4);
```

The lowest-order Elliptic filter that satisifies the specification is 4.

b. Implementation of DIIR

b.1. Butterworth

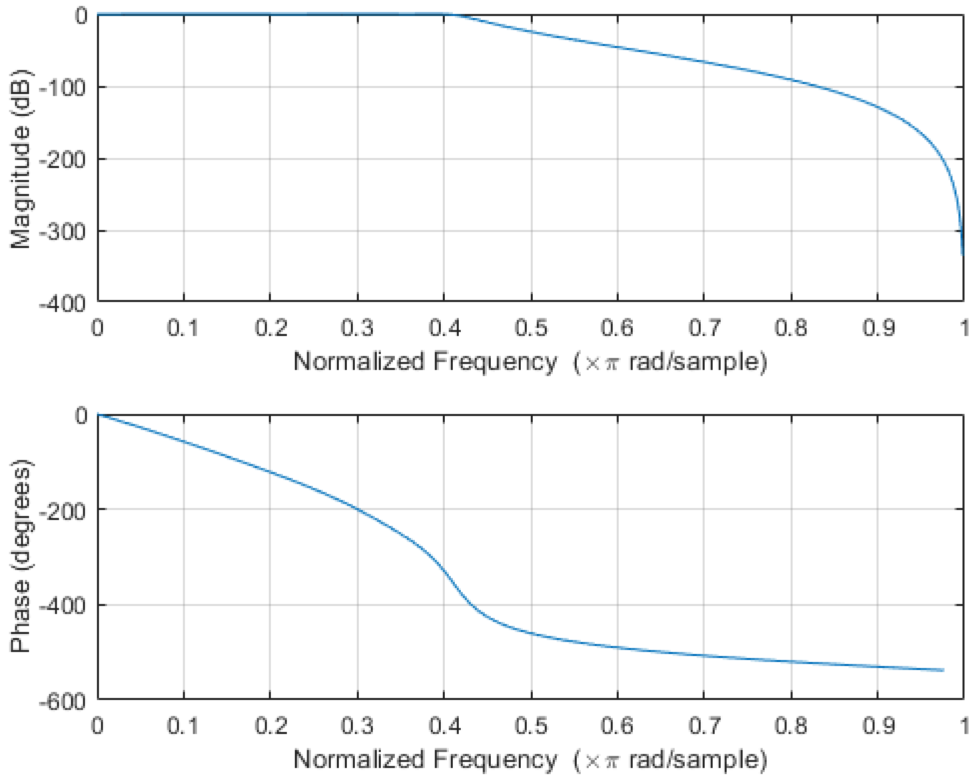
```
[b_4b1,a_4b1]=butter(n_4a1,wn_4a1);
figure(); freqz(b_4b1,a_4b1);
```



% Based from the output of the codes above, the specifications are satisfied.

b.2. Chebyshev Type-1

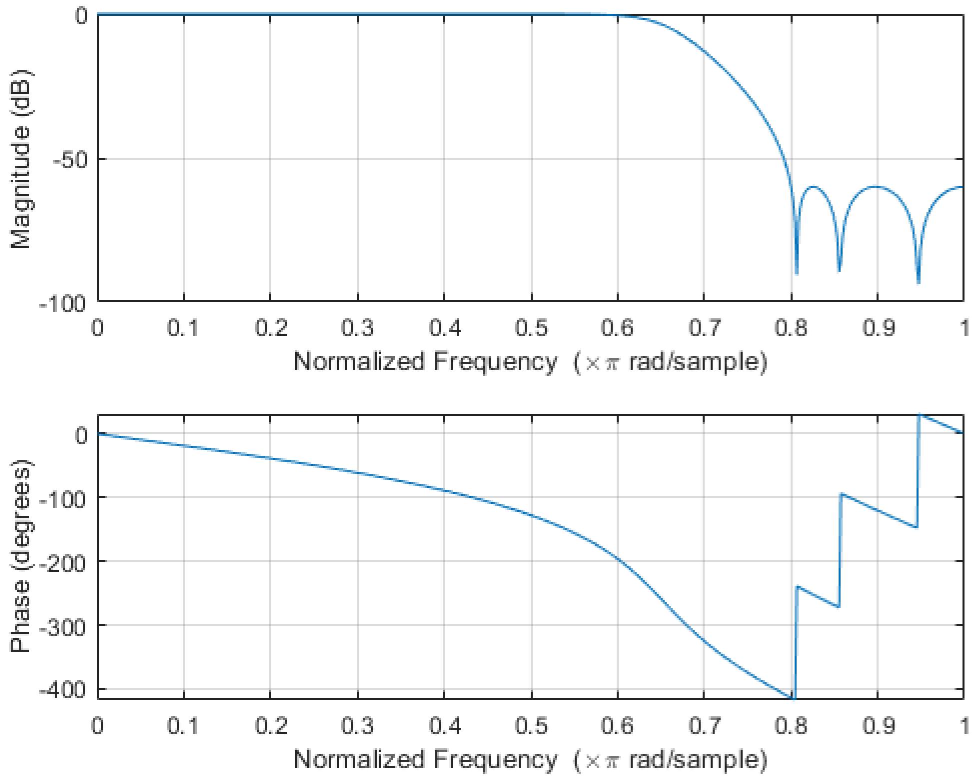
```
[b_4b2,a_4b2]=cheby1(n_4a1,rp_4a1,wn_4a2);
figure(); freqz(b_4b2,a_4b2);
```



% Based from the output of the codes above, the specifications are satisfied.

b.3. Chebyshev Type-2

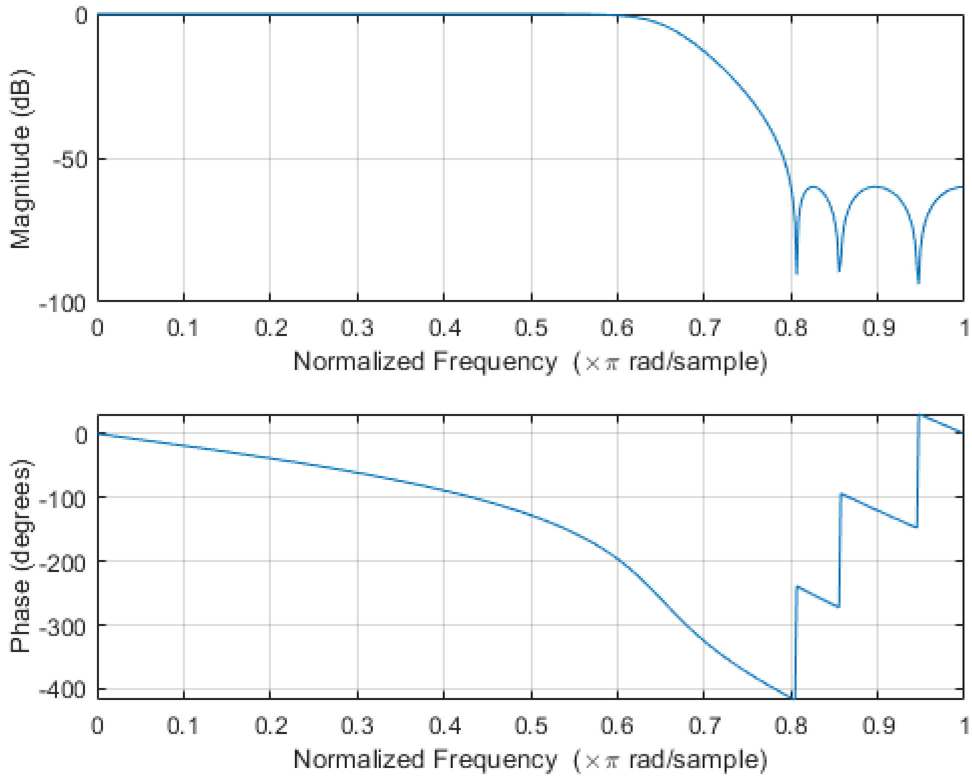
```
[b_4b3,a_4b3]=cheby2(n_4a1,rs_4a1,ws_4a1);
figure(); freqz(b_4b3,a_4b3);
```



% Based from the output of the codes above, the specifications are satisfied.

b.4. Elliptic

```
[b_4b4,a_4b4]=ellip(n_4a1,rp_4a1,rs_4a1,wp_4a1);
figure(); freqz(b_4b3,a_4b3);
```



% Based from the output of the codes above, the specifications are not satisfied.

5. APPLICATION

NOTE: See attached sheet for the documented answers of this section.

a. FIR application

Open firApplication.m firApplication

b. IIR Application

Open iirApplication.m iirApplication

Published with MATLAB® R2020b

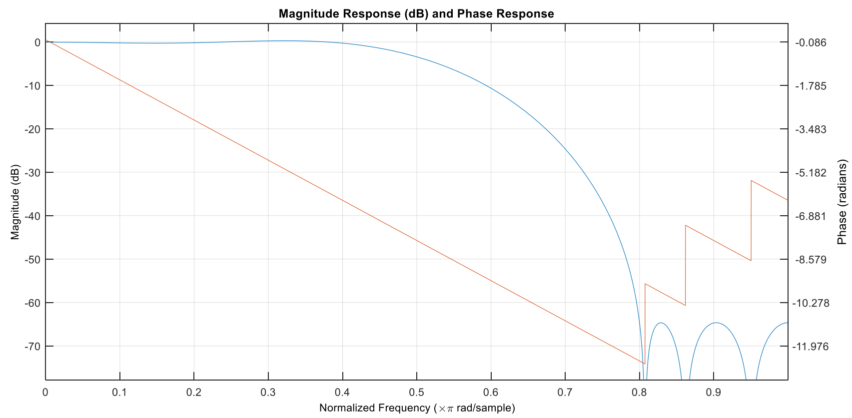
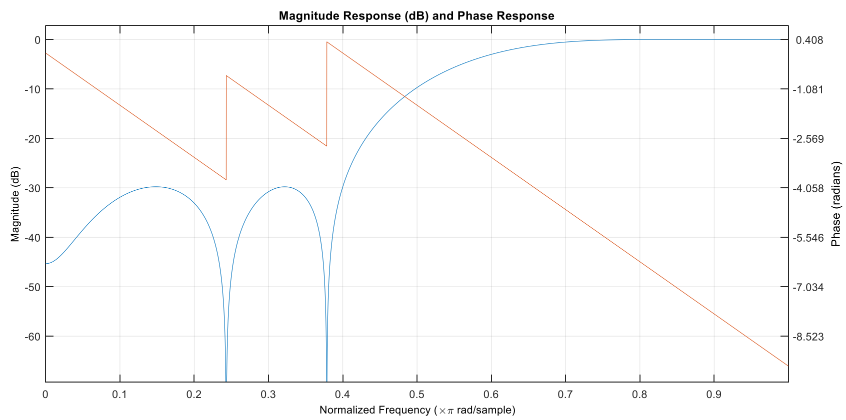
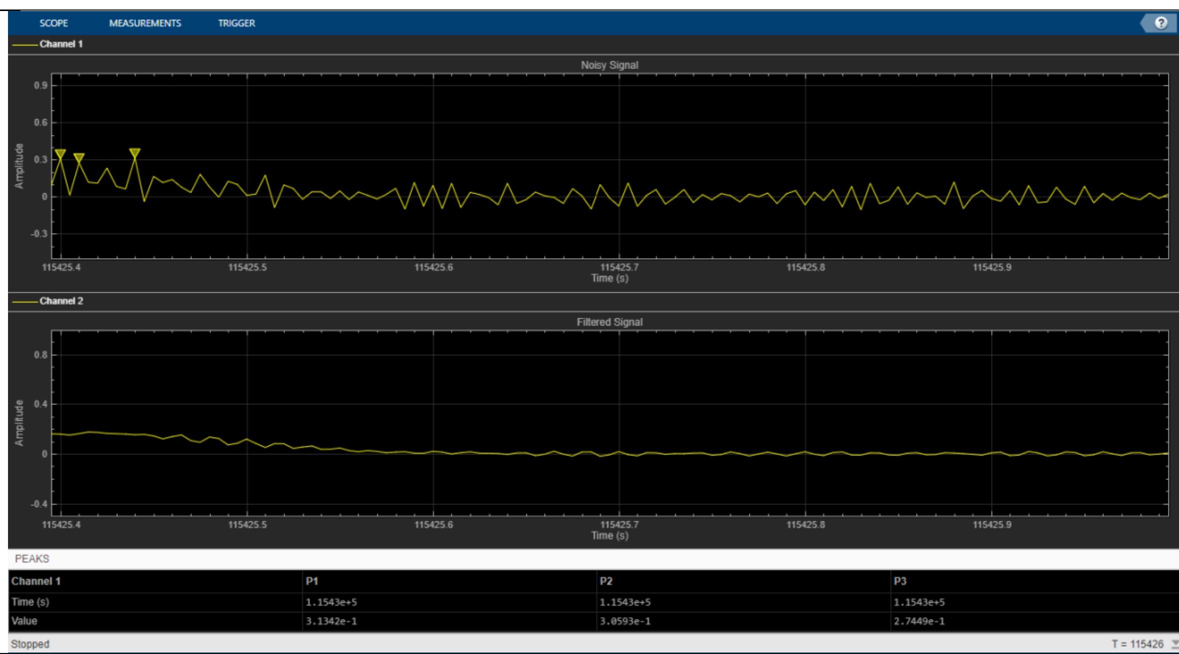
5. Application (For graduates only)

A. FIR Filter

<table>
<tr><td>Code</td><td>

```
sig_x=ecg(400).';
sig_y=sgolayfilt(sig_x,0,5);
[sig_m,sig_n]=size(sig_y);
sig_fs=200;
sig_ts=timescope('SampleRate',sig_fs,...
    'TimeSpanSource','Property',...
    'TimeSpan',0.6,...
    'ShowGrid',true,...
    'NumInputPorts',2,...
    'LayoutDimensions',[2 1]);
sig_ts.ActiveDisplay=1;
sig_ts.YLimits=[-0.5 1];
sig_ts.Title = 'Noisy Signal';
sig_ts.ActiveDisplay=2;
sig_ts.YLimits=[-0.5 1];
sig_ts.Title='Filtered Signal';

% lowpass
lp_fp=40; lp_fs=80; lp_ap=0.05; lp_sp=0.0009;
lp_F=[0 lp_fp lp_fs sig_fs/2]/(sig_fs/2);
lp_A=[1 1 0 0];
lp_D=[lp_ap lp_sp];
lp_b=firgr('minorder',lp_F,lp_A,lp_D);
LP=dsp.FIRFilter('Numerator',lp_b);
lpord=filtord(lp_b,1);

% highpass
hp_fs=40; hp_fp=80; hp_sp=0.05; hp_ap=0.0009;
hp_F=[0 hp_fs hp_fp sig_fs/2]/(sig_fs/2);
hp_A=[0 0 1 1];
hp_D=[hp_sp hp_ap];
hp_b=firgr('minord',hp_F,hp_A,hp_D);
HP=dsp.FIRFilter('Numerator',hp_b);
hpord=filtord(lp_b,1);

fprintf('The order of the design lowpass and highpass FIR filters are %d
and %d.\n',lpord,hpord);

tic;
while toc < 50
    sig_x=.1*randn(sig_m,sig_n);
    highFreqNoise=HP(sig_x);
    noisySignal=sig_y+highFreqNoise;
    filteredSignal=LP(noisySignal);
    sig_ts(noisySignal,filteredSignal);
end

release(sig_ts);
```

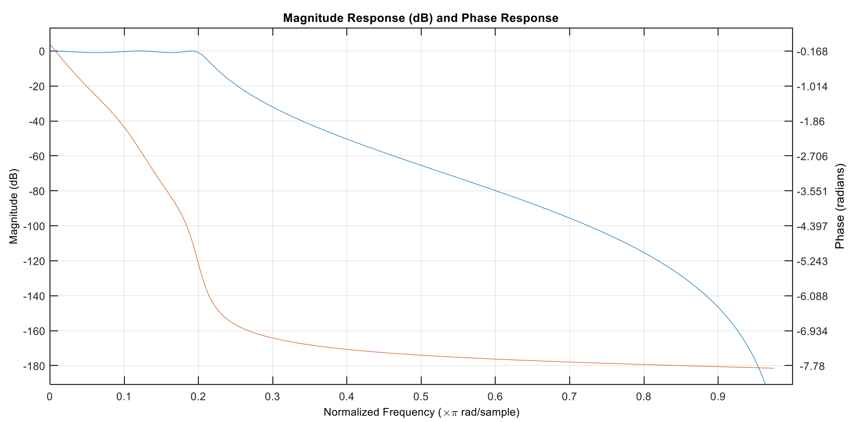
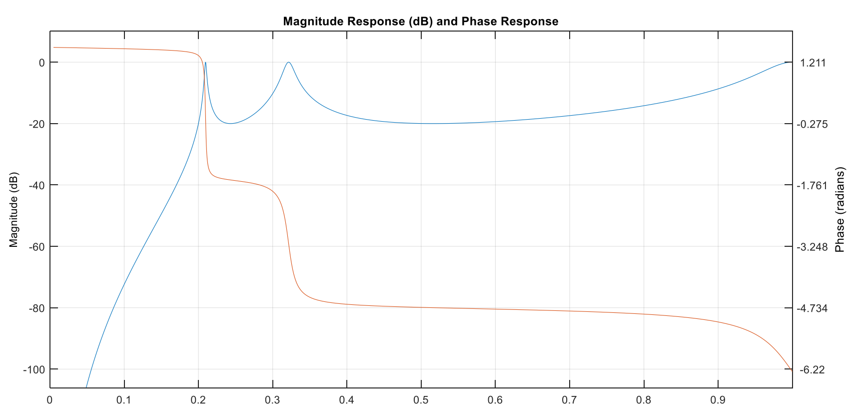
</td></tr>
<tr><td>Minimum Order</td><td>10</td></tr>
</table>

Lowpass and Highpass Filter Frequency Response	 
Noisy and Filtered Signal	

B. IIR Filter – Chebyshev Type-I

Code	`sig_x=ecg(400).';`
	`sig_y=sgolayfilt(sig_x,0,5);`
	`[sig_m,sig_n]=size(sig_y);`
	`sig_fs=200;`
	`sig_ts=timescope('SampleRate',sig_fs,...`
	`    'TimeSpanSource','Property',...`
	`    'TimeSpan',0.6,...`
	`    'ShowGrid',true,...`

```
    'NumInputPorts',2,...
    'LayoutDimensions',[2 1]);
sig_ts.ActiveDisplay=1;
sig_ts.YLimits=[-0.5 1];
sig_ts.Title = 'Noisy Signal';
sig_ts.ActiveDisplay=2;
sig_ts.YLimits=[-0.5 1];
sig_ts.Title='Filtered Signal';

% lowpass
[n,fc]=cheb1ord(40/200,80/200,-20*log10(0.9),-20*log10(0.01));
[b,a]=cheby1(n,-20*log10(0.9),fc);
LP=dsp.IIRFilter('Numerator',b,'Denominator',a);

% highpass
[b,a]=cheby1(n,-20*log10(0.1),fc, 'high');
HP=dsp.IIRFilter('Numerator',b,'Denominator',a);

fprintf('The order of the design IIR filter is %d.\n',n);

tic;
while toc < 30
    sig_x=.1*randn(sig_m,sig_n);
    highFreqNoise=HP(sig_x);
    noisySignal=sig_y+highFreqNoise;
    filteredSignal=LP(noisySignal);
    sig_ts(noisySignal,filteredSignal);
end

release(sig_ts);
```

Minimum Order	5
Lowpass and Highpass Filter Frequency Response	 

Noisy and Filtered Signal