```
% Christopher Morales
% EE 4820  5/7/2022
```

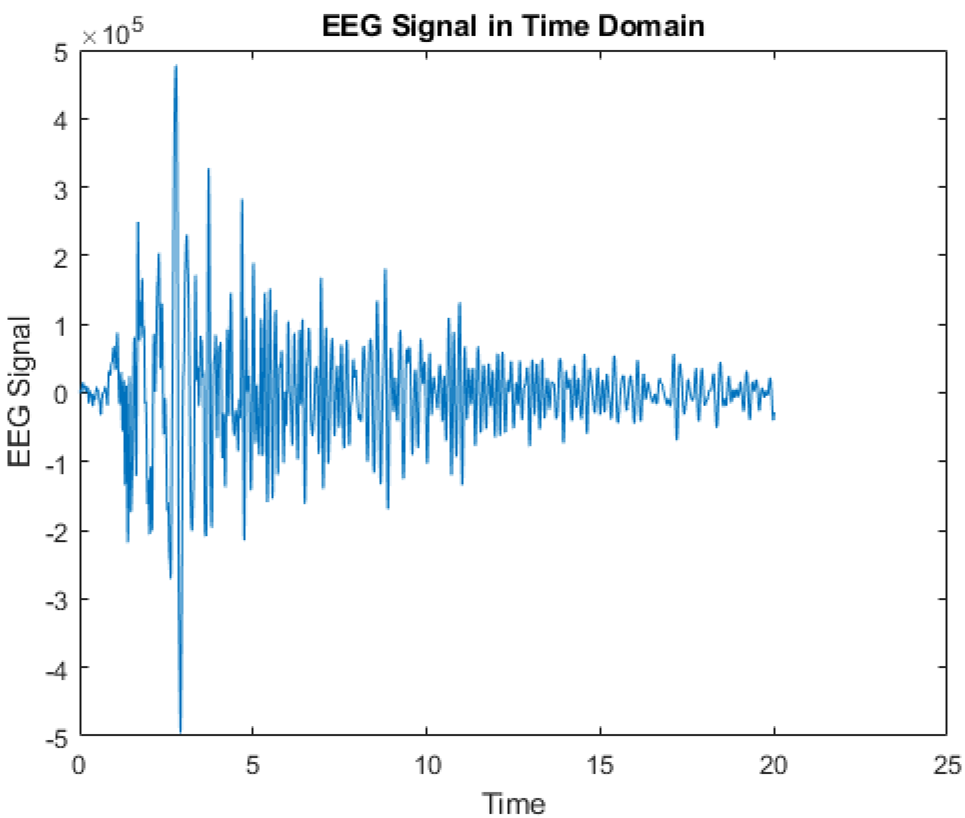
Problem 1

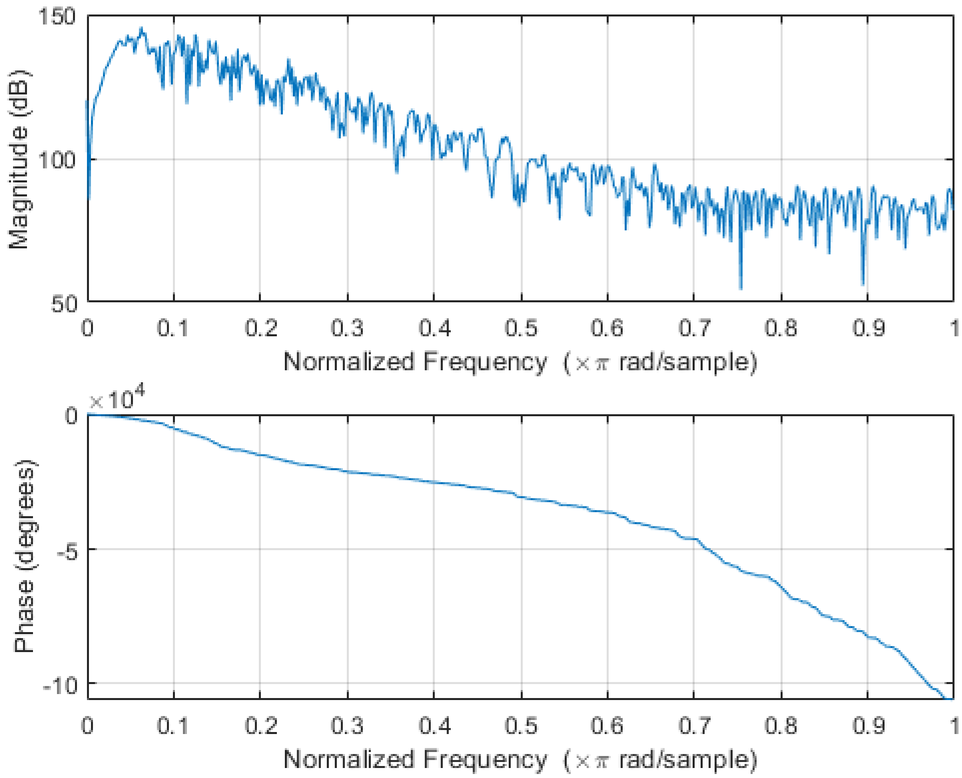
```
% Loading File and Content
load sig_2.mat;
whos

% Length of the data
L = length(data);

% Frequency Sampling Rate
fs = 100;

% Time Vector
t = [1:L]/fs;

% Creating a figure
figure(1);

% Plotting the EEG signal in Time Domain (visually determining the signal)
plot(t, data);
title('EEG Signal in Time Domain');
xlabel('Time');
ylabel('EEG Signal');

% By plotting the signal, we can determine there is noise
% from high to low frequencies and using butterworth filter.
% However, to use the butterworth filter we need to determine
% the frequency cutoff point where we could maximumize the
% filtration that has no ripples or long curvature slopes.

% Resolution
nfft = 2^10;

% Creating a figure
```

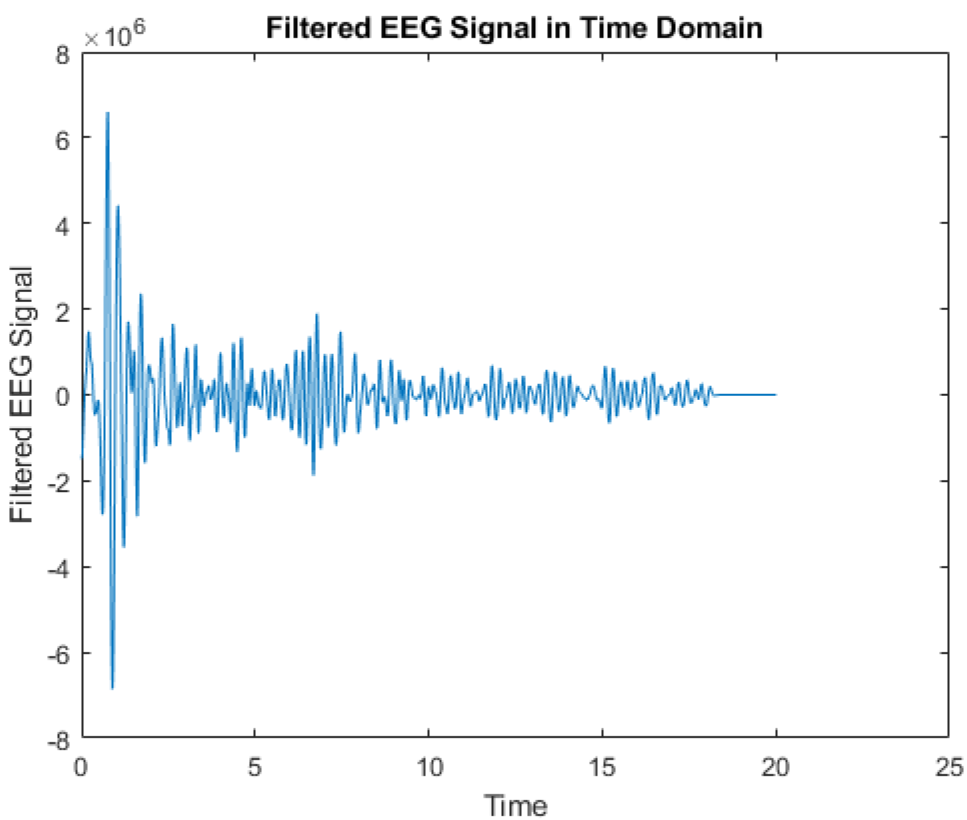
```
figure(2);

% Gives the magnitude and phase angle of the Frequency Response
freqz(data);


% By plotting the signal for f vs |H| in Decibel and f vs |H|
% there is no correlation but at 0. However, f vs |H|, we can
% see the spectrum of the signal is larger at high frequencies
% more but there are still quite large frequencies at the low
% spectrum. By looking the highest peak, the frequency cutoff
% is 3.07617 = 3 Hz

% Frequency Cutoff (Alpha band)
fc = [4 7];

% Frequency Contraint
Wn = fc./(fs/2);

% Filter Order, Nyquist order (stated by matlab)
N = 5;

% Returns transfer function coefficient (creating a tranfer function filter)
% 10 is the gain
% Wn is the vector (knows the high or low of the vector)
%[b, a] = cheby1(N, 10, Wn, 'stop');
[b, a] = butter(N, Wn, "bandpass");

% Creating a figure
figure(3);

% Frequency Response (Filtered )
h = freqz(b, a, [], fs);

% Apply the convolution (Filtered Transfer Function * Data)
y = conv(data, h, 'same');

% Plotting the Convolution Signal
plot(t,y);
title('Filtered EEG Signal in Time Domain');
xlabel('Time');
ylabel('Filtered EEG Signal');


% Now we are going to compared the convoluted signal with the original
% signal

% Creating a figure
figure(4);

% Original Signal
subplot(2,1,1);
plot(t, data);
```

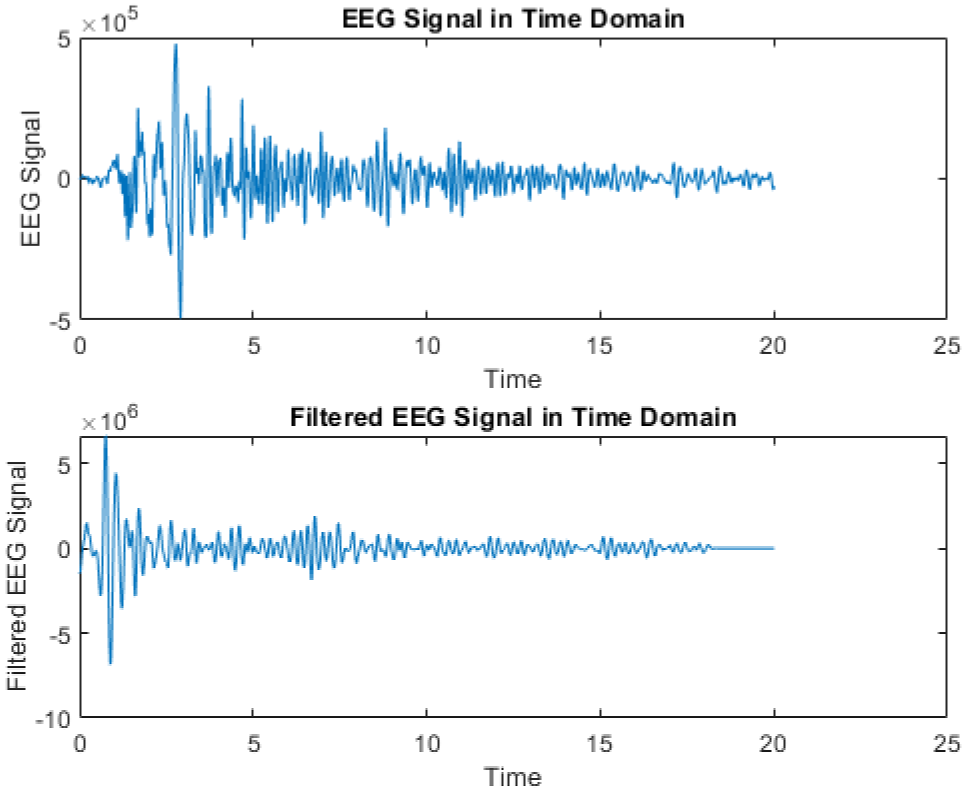
```
title('EEG Signal in Time Domain');
xlabel('Time');
ylabel('EEG Signal');

% Convoluted signal
subplot(2,1,2);
plot(t,y);
title('Filtered EEG Signal in Time Domain');
xlabel('Time');
ylabel('Filtered EEG Signal');
```

```
  Name      Size            Bytes  Class     Attributes

  data      1x2001          16008  double

Warning: Imaginary parts of complex X and/or Y arguments ignored.
Warning: Imaginary parts of complex X and/or Y arguments ignored.
```




Magnitude (dB)
Normalized Frequency (×π rad/sample)
Phase (degrees)
×10^4
Normalized Frequency (×π rad/sample)


Filtered EEG Signal in Time Domain
×10^6
Filtered EEG Signal
Time



Problem 2

```
% Load File and Content
load ecg_lfn.dat;
whos

% Frequency Sample Rate
fs = 1000;

% Length of the ECG
L = length(ecg_lfn);

% Time Vector
t = [1:L]/fs;

% Creating a figure
figure(5);

% Plotting ECG Signal
plot(t, ecg_lfn);
title('ECG Singal in Time Domain');
xlabel('Time');
ylabel('ECG Signal');

% By looking at the ECG singal, we can high peaks of frequency and when
% zooming onto the ECG signal, we can see more static to the low
```

```
% frequencies
% To determine the frequency cutoff, we are going to plot the

% Filter Order, Nyquist order (stated by matlab)
n = 4;

% Creating a figure
figure(6);

% Frequency Response of the ECG Signal
freqz(ecg_lfn);
[h, f] = freqz(ecg_lfn, n, 'whole');

% Getting the Magnitude and Magnitude in Decibel Frequency Response
hmag = abs(h);
hdecibel = 20 * log10(abs(h));

% Creating a figure
figure(7);

% Plotting the Magnitude and Magnitude in Decibel to see the correlation
% and determine the frequency cutoff
plot(f, hmag, f, hdecibel);
axis([0 6.3 -25 60]);
title('Frequency Response');
xlabel('Frequency');
ylabel('|H| and |H| in Decibel');
legend('|H|','|H| in Decibel');

% By looking at the plot above, we can determine the frequency cutoff is
% about 6 Hz.

% Frequency Cutoff
fc = 6;

% Frequency Contraint
Wn = fc/(fs/2);

%%%%%%%%%%%%%%%%%%%%%%%%%%
```

Butterworth Filter %%%

```
%%%%%%%%%%%%%%%%%%%%%%%%%%

% Applying the Vutterworth Filter
[b1, a1] = butter(n, Wn, "high");

% Frequency Response (Filtered )
%h1 = freqz(b1, a1, [], fs);

% Applying the convolution * butterworth filter
%y1 = conv(ecg_lfn, h1, 'same');
```

```
y1 = filtfilt(b1,a1,ecg_lfn);

% Creating a figure
figure(8);

% Plotting the convoluted Butterworth Filter
plot(t, abs(y1));
title('Convoluted Butterworth Filter ECG Signal');
xlabel('Time');
ylabel('|ECG|');

%%%%%%%%%%%%%%%%%%%%%%%%
```

Chebychev Filter %%%

```
%%%%%%%%%%%%%%%%%%%%%%%%

% Returns the Transfer Function Coeffiecient
[b2, a2] = cheby1(n, 10, Wn, 'high');
title('Convoluted Chebychev Filter ECG Signal');
xlabel('Time');
ylabel('|ECG|');
% Frequency Response (Filtered )
%h2 = freqz(b2, a2, [], fs);

% Applying the convolution * Chebychev filter
%y2 = conv(ecg_lfn, h2, 'same');
y2 = filtfilt(b2,a2, ecg_lfn);

% Creating a figure
figure(9);

% Plotting the convoluted Chebychev Filter
plot(t, abs(y2));
title('Convoluted Chebychev Filter ECG Signal');
xlabel('Time');
ylabel('|ECG|');

%%%%%%%%%%%%%%%%%%%%%%%%%
```

Elliptical Filter %%%

```
%%%%%%%%%%%%%%%%%%%%%%%%%

% Elliptical Filter
[b3, a3] = ellip(n, 1, fc, Wn, 'high' );

% Frequency Response (Filtered )
%h3 = freqz(b3, a3, [], fs);

% Applying the convolution * Chebychev filter
```

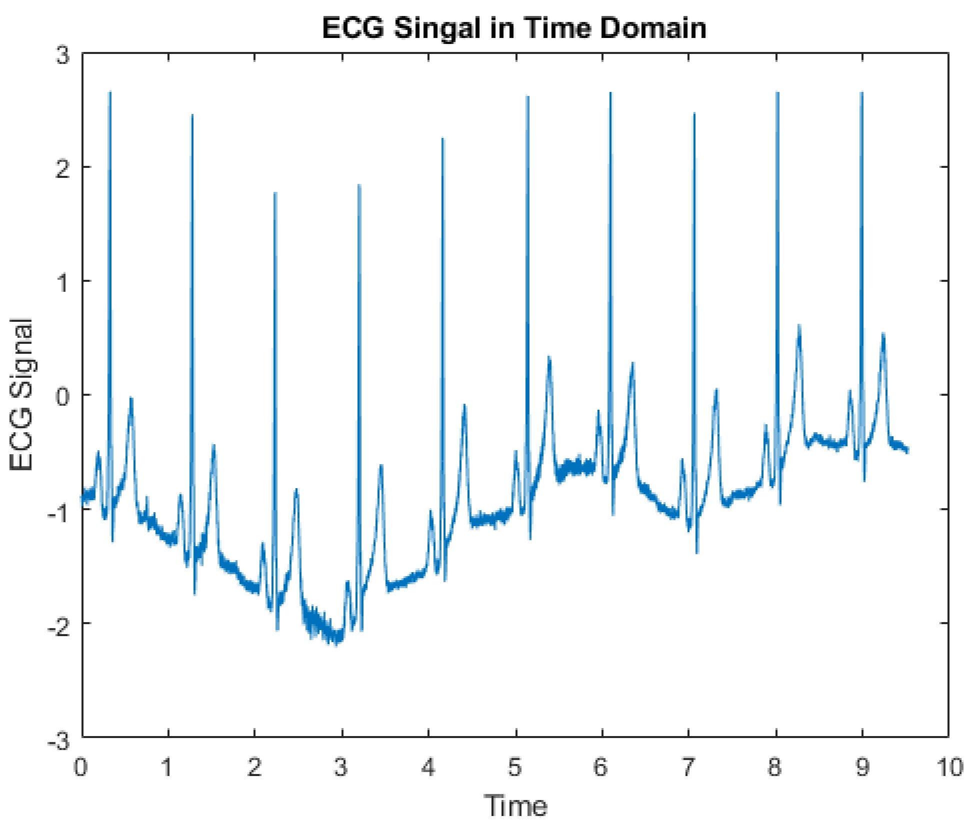
```
%y3 = conv(ecg_lfn, h3, 'same');
y3 = filtfilt(b3, a3, ecg_lfn);

% Creating a figure
figure(10);

% Plotting the convoluted Chebychev Filter
plot(t, abs(y3));
title('Convoluted Elliptical Filter ECG Signal');
xlabel('Time');
ylabel('|ECG|');

% Plotting the output signal
% H = Y / X
% Y = H * X
% X = Y * H


% Create a figure
figure(11);

% Comparing the orginal signal with the concoluted butterworth signal
plot(t, ecg_lfn, 'blue', t, abs(y1), 'magenta');
title('Input vs Output ECG Signal');
xlabel('Time');
ylabel('ECG');
legend( 'Fitlered ECG', 'Raw ECG data');

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

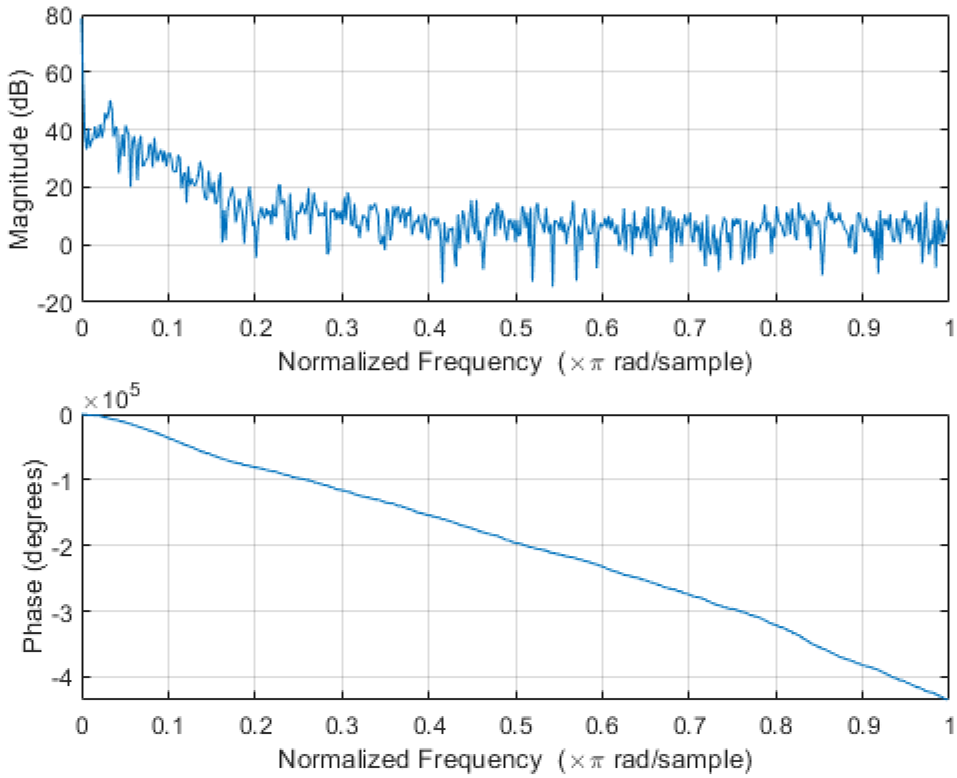
Comparing the Elliptical and Chebychev Filter %%%

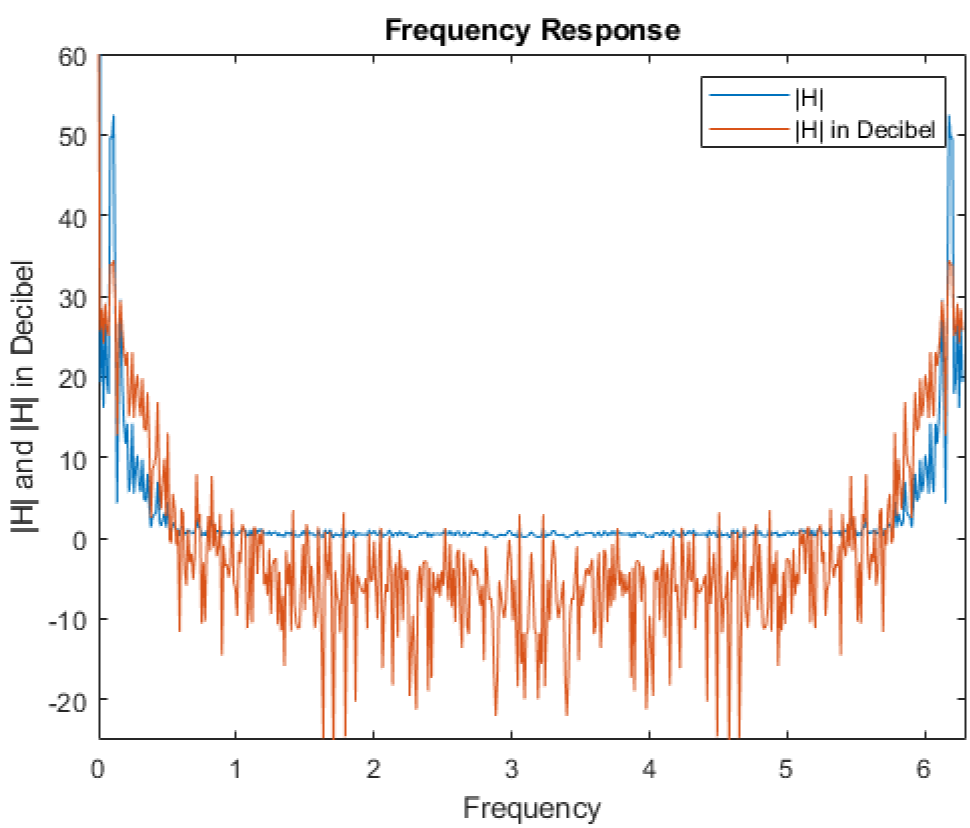
```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% Creating a figure
figure(12);

% Plotting the two filters
plot(t, y2, t, y3, 'magenta');
xlabel('Time')
ylabel('ECG Filtered');
title('Cheby vs Elliptical Filter (Comparison)');
legend('ChebyChev Filter', 'Ellipical Filter');


% By looking at the two filters we can see the elliptical has a ripple
% effect. Like the signal is acting like a capacitor, the state of decay
% then the signal jumps back up. Also, it seems that the noise is still
% present on elliptical. While for Cheby, the signal is more stable and
% compare to the elliptical, it has less noise.
```

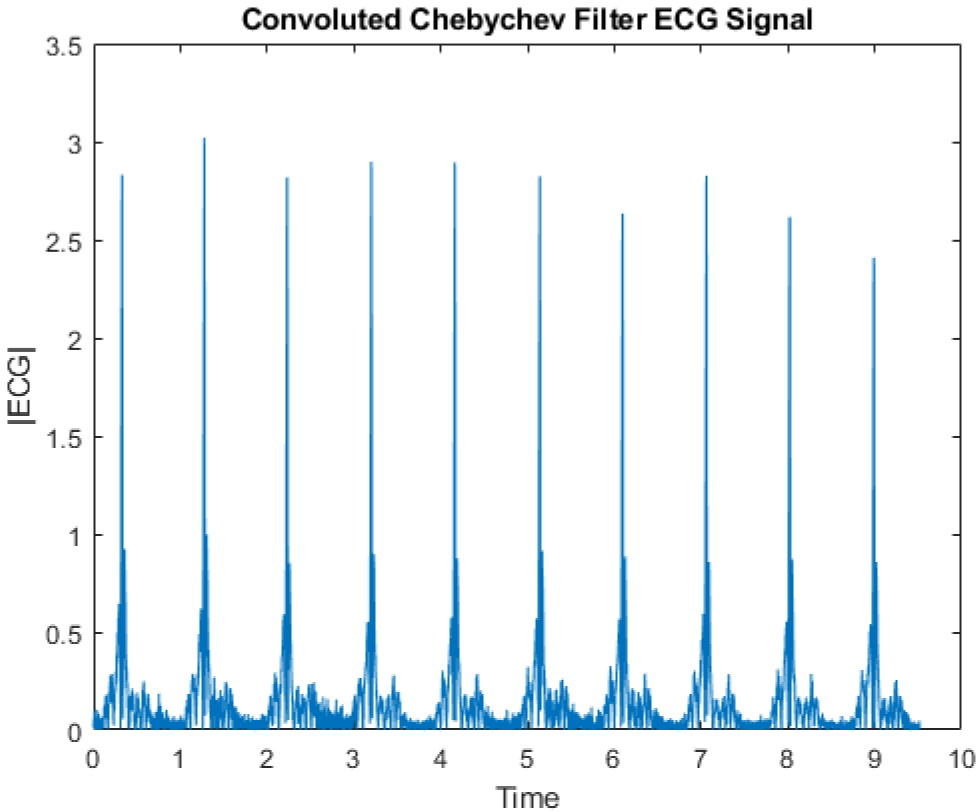
```
  Name               Size                  Bytes  Class      Attributes
```

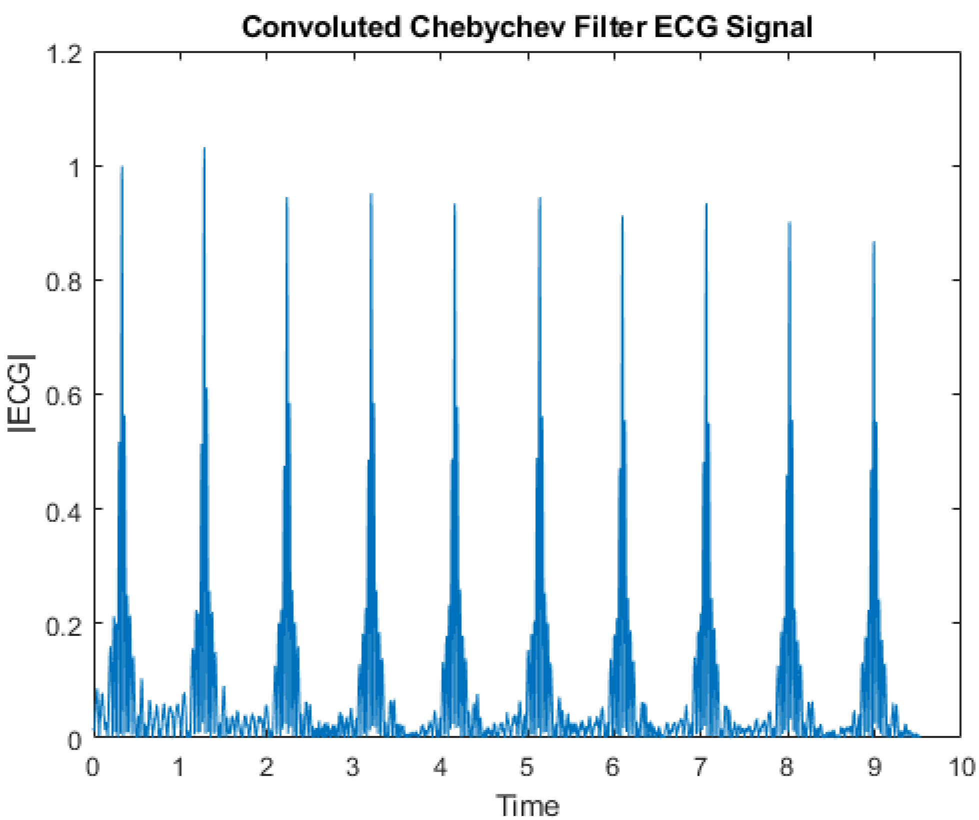
```
L              1x1           8  double
N              1x1           8  double
Wn             1x2          16  double
a              1x11         88  double
b              1x11         88  double
data           1x2001    16008  double
ecg_lfn     9519x1       76152  double
fc             1x2          16  double
fs             1x1           8  double
h            512x1        8192  double    complex
nfft           1x1           8  double
t              1x2001    16008  double
y              1x2001    32016  double    complex
```

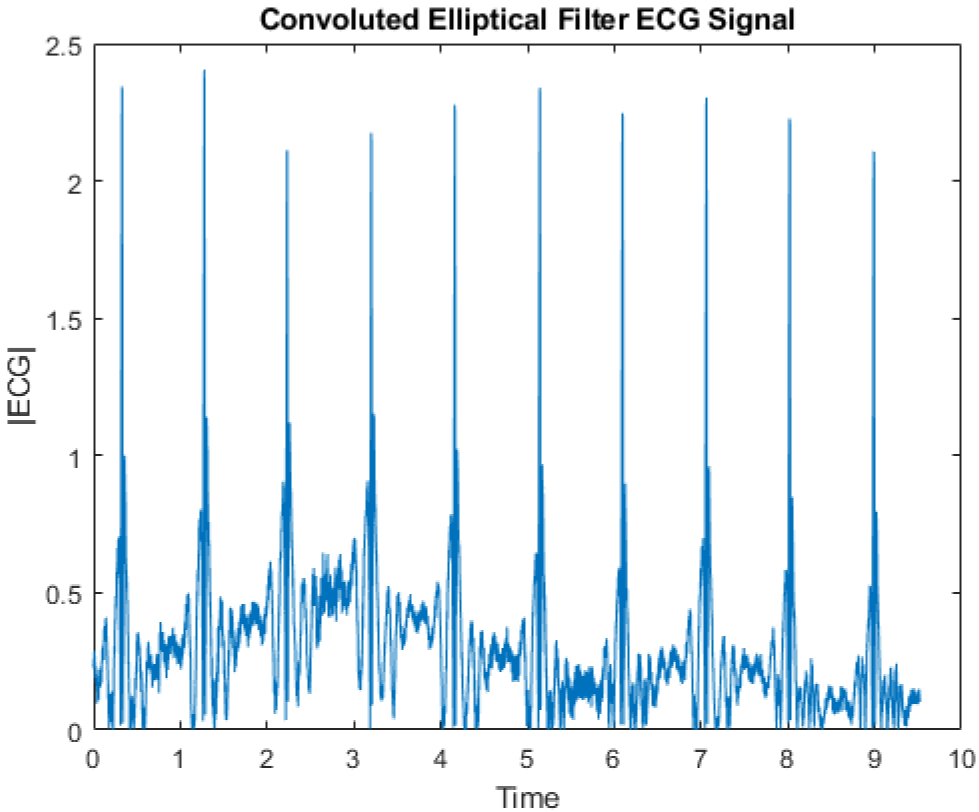


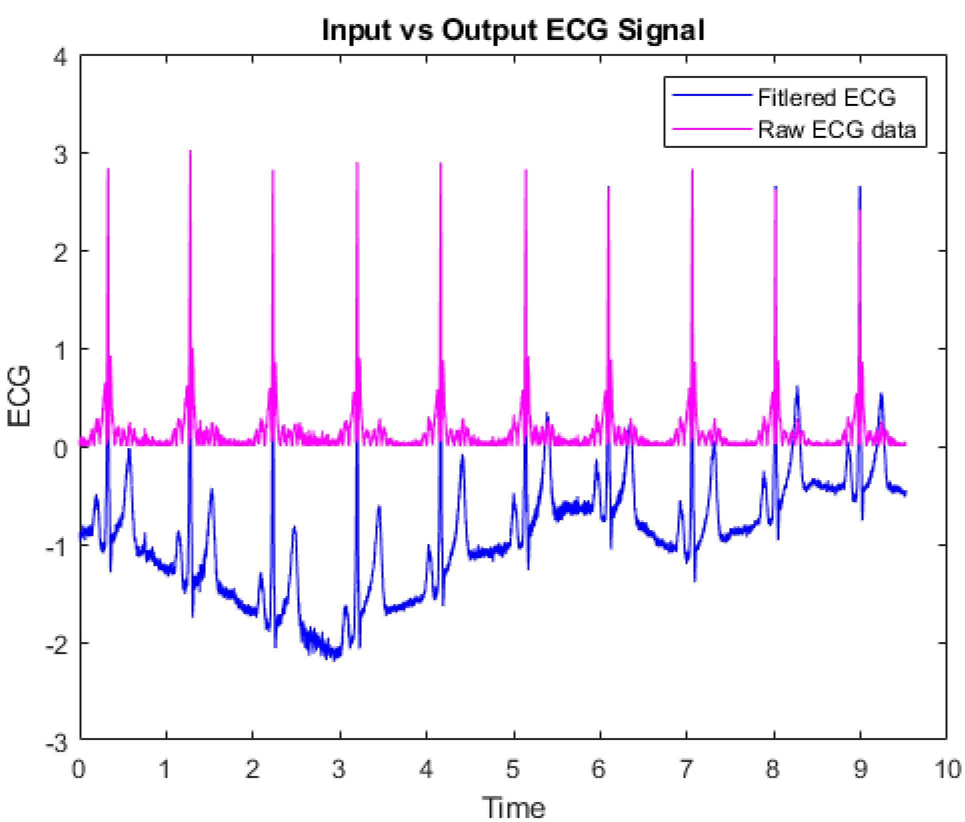

Magnitude (dB)
Normalized Frequency (×π rad/sample)
×10^5
Phase (degrees)
Normalized Frequency (×π rad/sample)

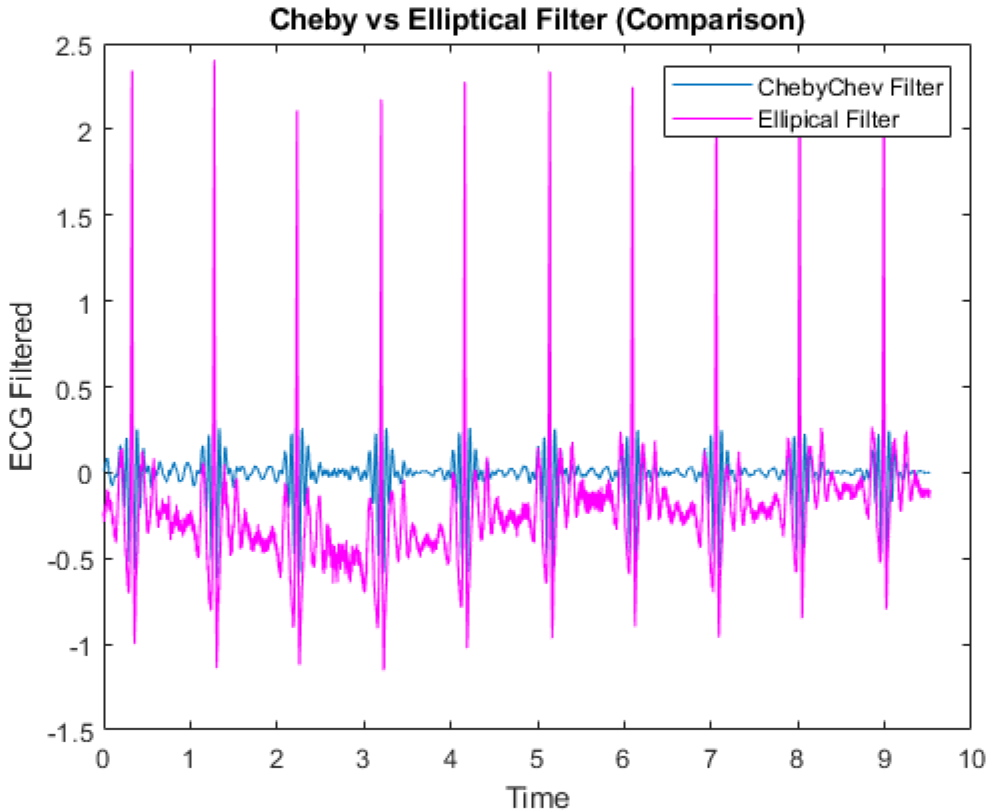

Frequency Response
|H|
|H| in Decibel
|H| and |H| in Decibel
Frequency


Convoluted Chebychev Filter ECG Signal
|ECG|
Time
3.5
3
2.5
2
1.5
1
0.5
0
0
1
2
3
4
5
6
7
8
9
10


Convoluted Chebychev Filter ECG Signal
|ECG|
Time
1.2
1
0.8
0.6
0.4
0.2
0
0
1
2
3
4
5
6
7
8
9
10


Convoluted Elliptical Filter ECG Signal
|ECG|
Time
0
0.5
1
1.5
2
2.5
0
1
2
3
4
5
6
7
8
9
10


Input vs Output ECG Signal
Fitlered ECG
Raw ECG data
ECG
Time
-3
-2
-1
0
1
2
3
4
0
1
2
3
4
5
6
7
8
9
10



Problem 3

```
% Load File and Content
load rat4_10282011_110956-1.mat;
whos

% Frequency Sample Rate
fs = 3000;

% Getting data only from channel 2
channel2 = s(:,2);

% Length of channel 2
L = length(s(:,2));

% Time Vector
t = [1:L]/fs;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

Different time Length of Windows %%%

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% 80 ms window
```

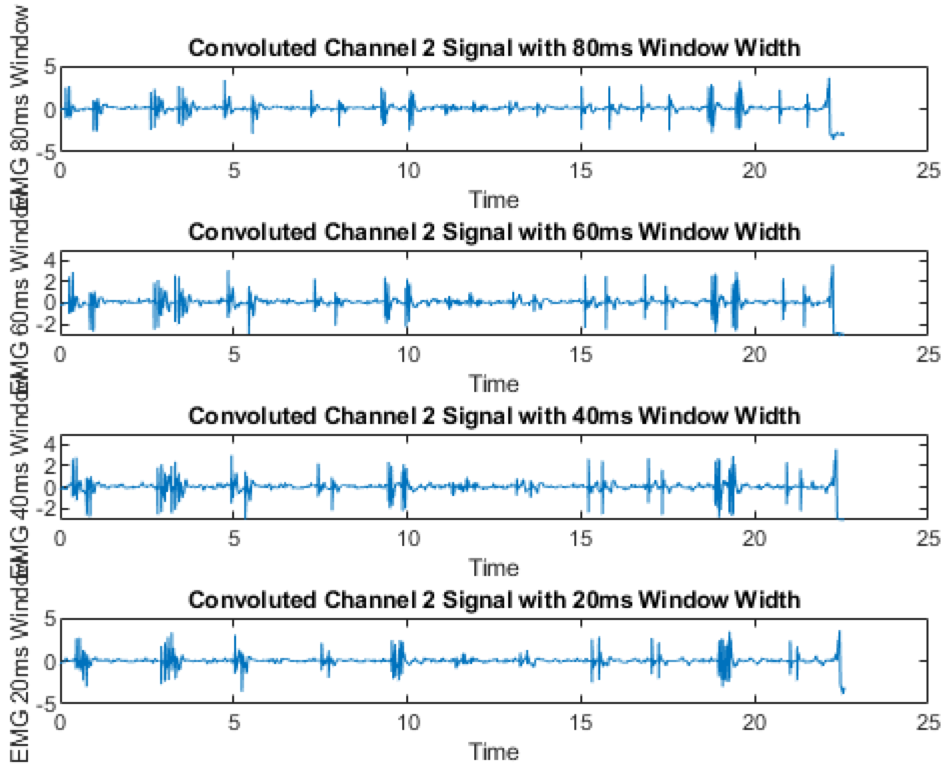
```
Lwin = fs * 0.8;
Lwin = ones(1, Lwin);

% 60 ms window
Lwin2 = fs * 0.6;
Lwin2 = ones(1, Lwin2);

% 40 ms window
Lwin3 = fs * 0.4;
Lwin3 = ones(1, Lwin3);

% 20 ms window
Lwin4 = fs * 0.2;
Lwin4 = ones(1, Lwin4);

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

Convolution of Channel 2 and time width of the window %%%

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% Convolution for 80 ms window
avg = conv(channel2, Lwin, 'same');

% Convolution for 60 ms window
avg2 = conv(channel2, Lwin2, 'same');

% Convolution for 40 ms window
avg3 = conv(channel2, Lwin3, 'same');

% Convolution for 20 ms window
avg4 = conv(channel2, Lwin4, 'same');

% Creating a figure
figure(13);

% Plotting the signals
subplot(4,1,1);
plot(t, avg);
title('Convoluted Channel 2 Signal with 80ms Window Width');
xlabel('Time');
ylabel('EMG 80ms Window');

subplot(4,1,2);
plot(t, avg2);
title('Convoluted Channel 2 Signal with 60ms Window Width');
xlabel('Time');
ylabel('EMG 60ms Window');

subplot(4,1,3);
plot(t, avg3);
title('Convoluted Channel 2 Signal with 40ms Window Width');
```

```
xlabel('Time');
ylabel('EMG 40ms Window');

subplot(4,1,4);
plot(t, avg4);
title('Convoluted Channel 2 Signal with 20ms Window Width');
xlabel('Time');
ylabel('EMG 20ms Window');

% By comparing the time window, I noticed that the EMG signal time starts
% to shift as I lower the time window. Telling me that, as i lower the
% window time, there is some sort of time latency in echange a general EMG
% signal. While increasing the time window, the EMG signal is more define.
% When I mean define, when comparing, 80ms vs 20ms, looking at the range
%  time domain 18 - 20 ms. We can see that the 20ms has has one spike
%  signal where 80 ms has two. So we are sacrificing data for an more avg
%  data
```

```
  Name                Size                Bytes  Class     Attributes

  L                   1x1                     8  double              
  N                   1x1                     8  double              
  Wn                  1x1                     8  double              
  a                   1x11                   88  double              
  a1                  1x5                    40  double              
  a2                  1x5                    40  double              
  a3                  1x5                    40  double              
  b                   1x11                   88  double              
  b1                  1x5                    40  double              
  b2                  1x5                    40  double              
  b3                  1x5                    40  double              
  data                1x2001              16008  double              
  ecg_lfn          9519x1                 76152  double              
  f                 512x1                  4096  double              
  fc                  1x1                     8  double              
  fs                  1x1                     8  double              
  h                 512x1                  8192  double    complex   
  hdecibel          512x1                  4096  double              
  hmag              512x1                  4096  double              
  n                   1x1                     8  double              
  nfft                1x1                     8  double              
  pos             11289x4                361248  double              
  s               67735x2               1083760  double              
  t                   1x9519              76152  double              
  time            11289x1                 90312  double              
  y                   1x2001              32016  double    complex   
  y1               9519x1                 76152  double              
  y2               9519x1                 76152  double              
  y3               9519x1                 76152  double              
```



Published with MATLAB® R2021b