

Assessment (ECE2111)

Lab 04 Result Document



MONASH University

Name: Tan Jin Chun

Student ID: 32194471

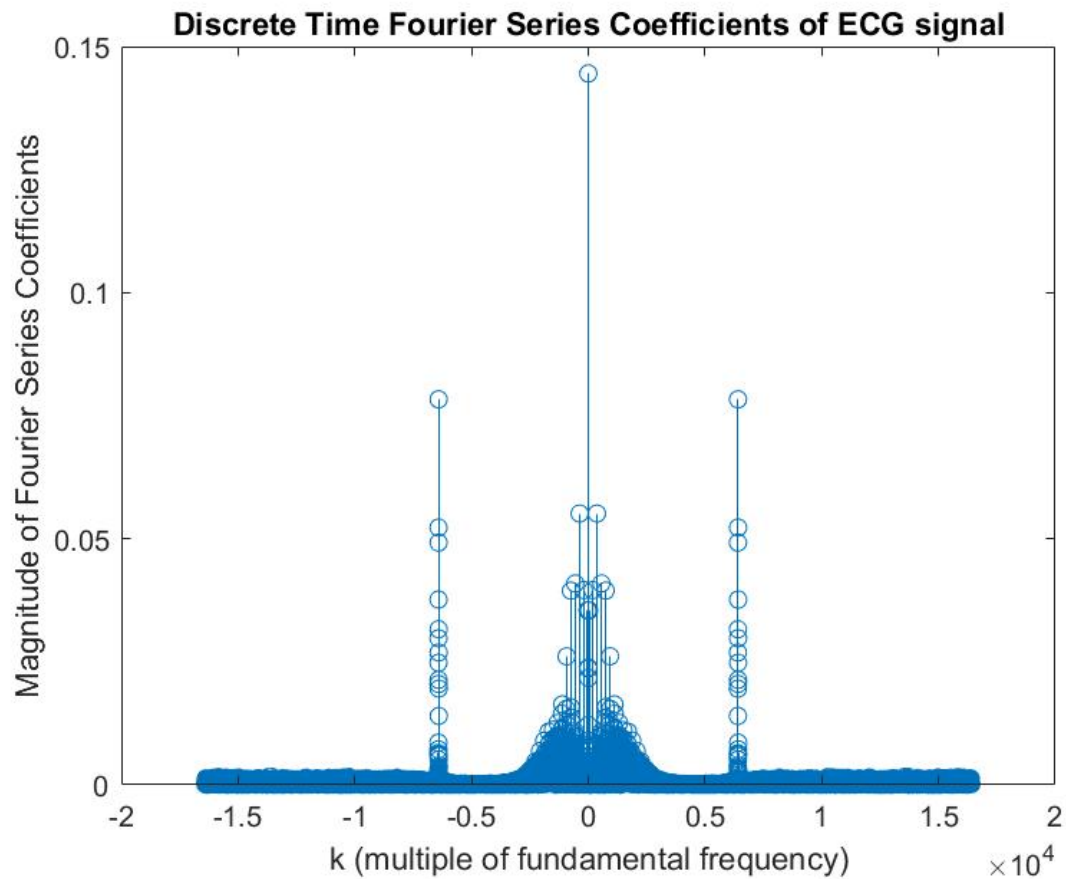
ECE2111 lab4 results document:

Name: Tan Jin Chun

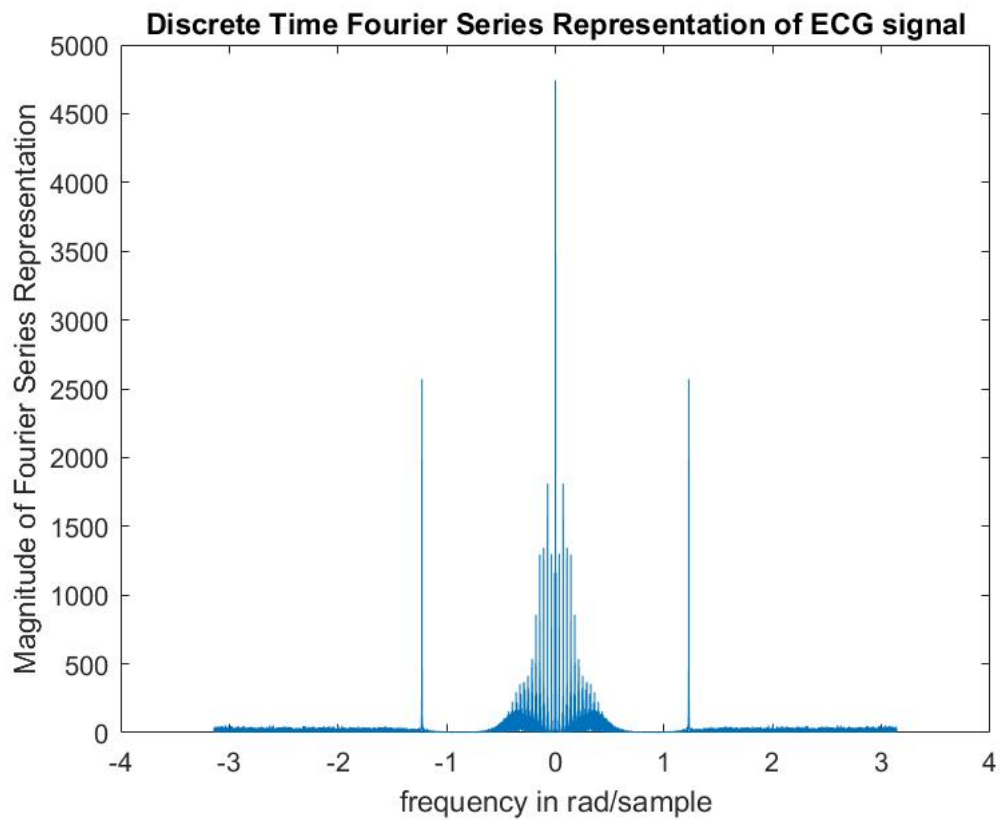
Names of all students you discussed this lab with: Chong Yen Juin, Huan Meng Hui, Ku Yew Siang, Loh Jia Quan

Section 1:

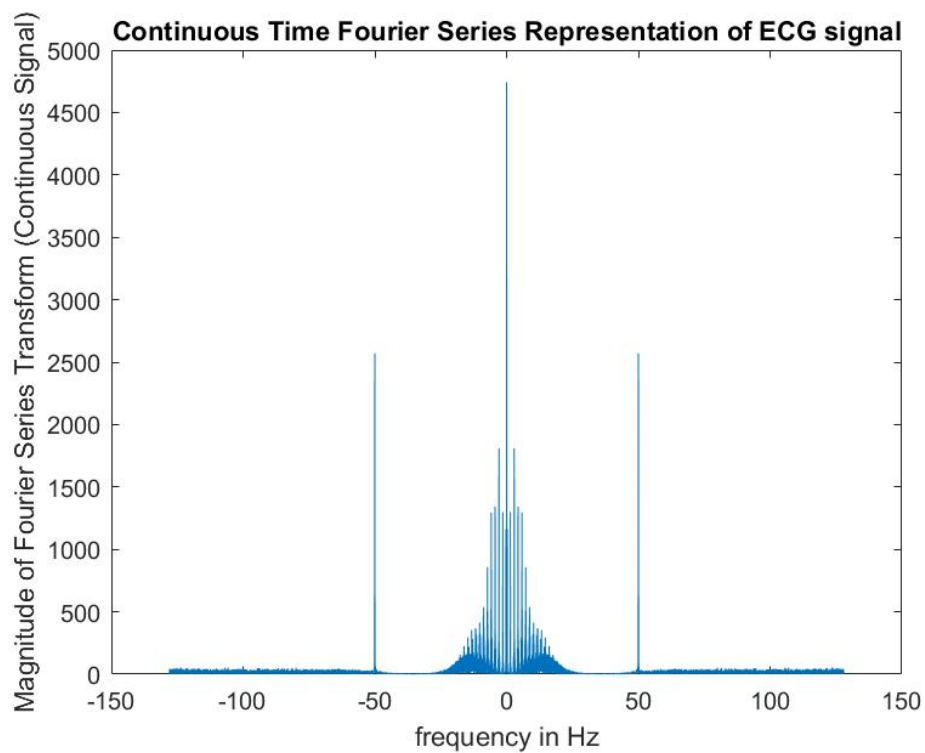
Question 2(a): Include your labeled plot below.



Question 2(b): Include your labeled plot below.



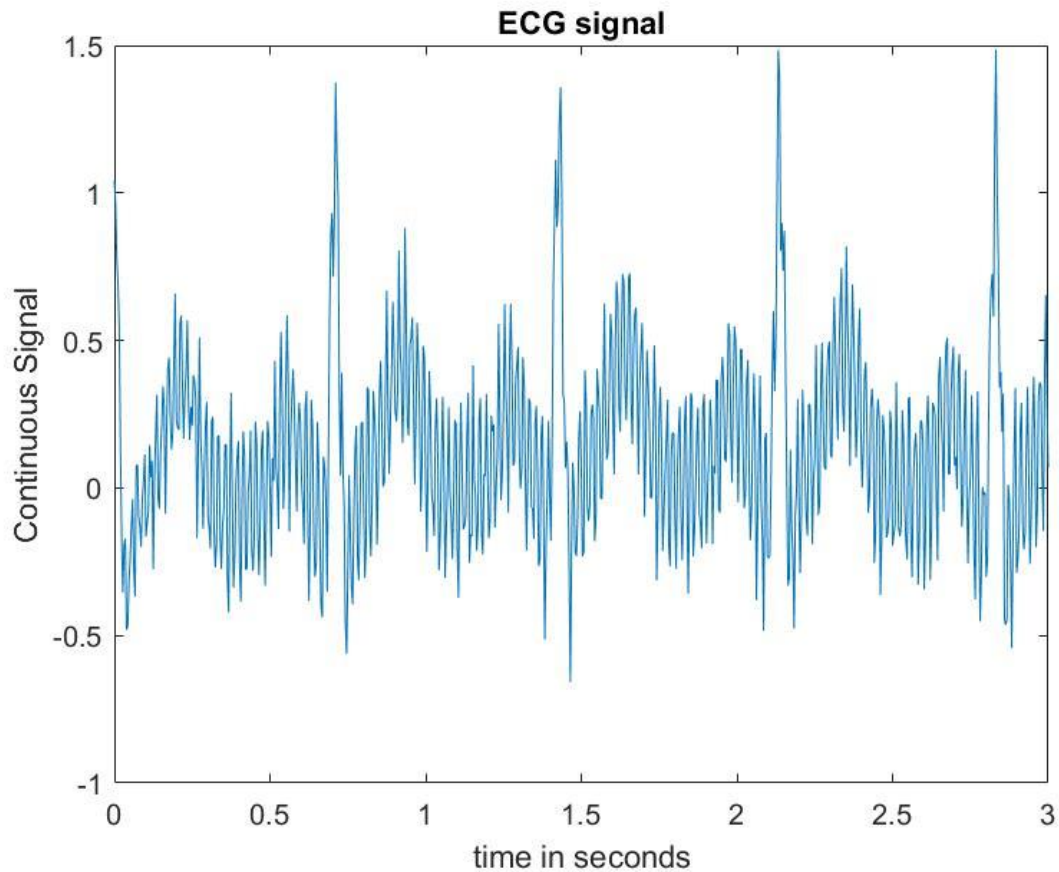
Question 2(c): Include your labeled plot below.



Question 3: Identify the power-line frequency in rad/sample and in Hz.

The power-line frequency in Hz is 50Hz and 0.390625π ($2 \cdot 50 \cdot \pi / 256$) rad/sample.

Question 4: Include a labeled plot of the signal in time domain over an interval of time that shows the features of the signal well.



Question 5: How have the low-frequency noise and power-line interference contaminated the ECG, based on the time-domain plot?

The low frequency noise and power-line interference has contaminated the ECG by distorting the signal in such a way that the ECG signal shows a large variation (oscillation) in the signal which would interfere with the analysis of the ECG signal. Multiple spikes could also be seen in the ECG signal.

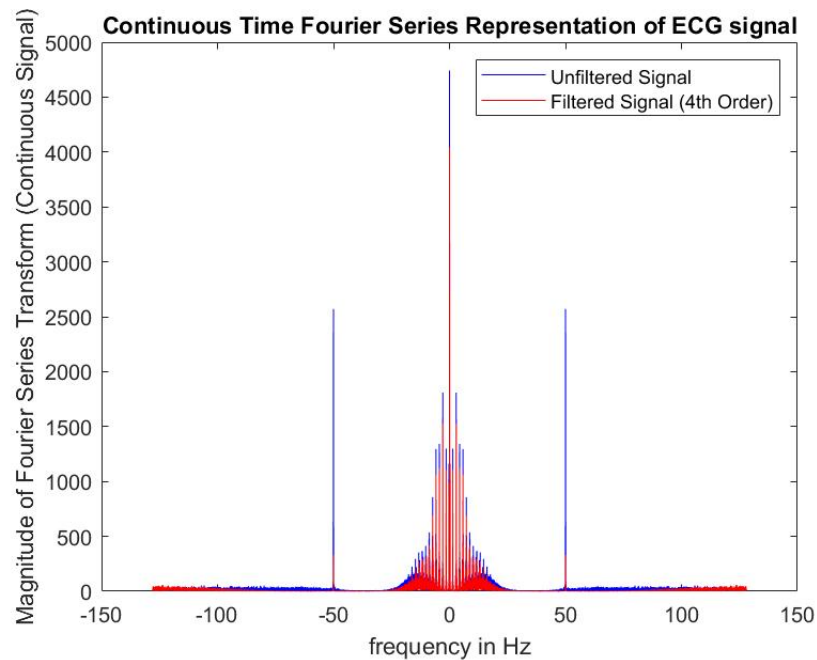
Section 2:

Question 2(b): On the same set of axes, plot the signal in the frequency domain before and after filtering. Include your labeled plot, below. Do this for each choice of parameters (cutoff frequencies and filter lengths).

Choice of parameters:

Cut-off frequencies: 49 – 51 Hz

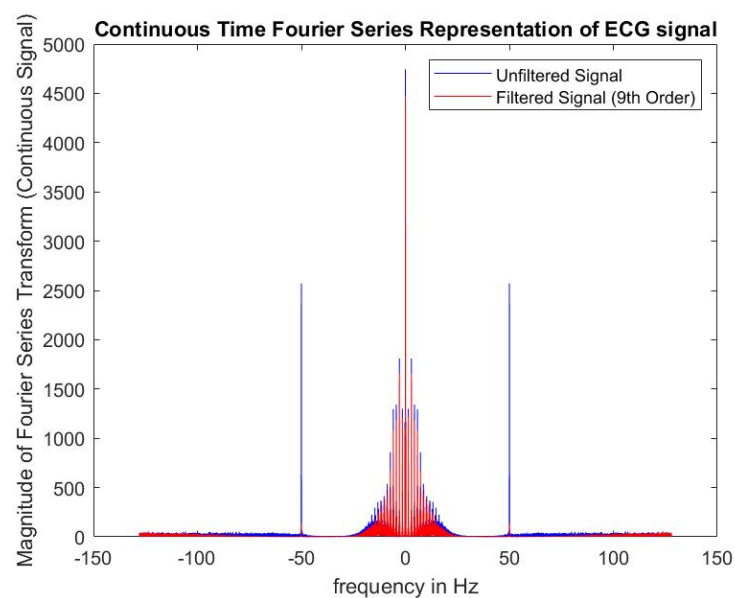
Order: 4 ($L = 5$).



Choice of parameters:

Cut-off frequencies: 49 – 51 Hz

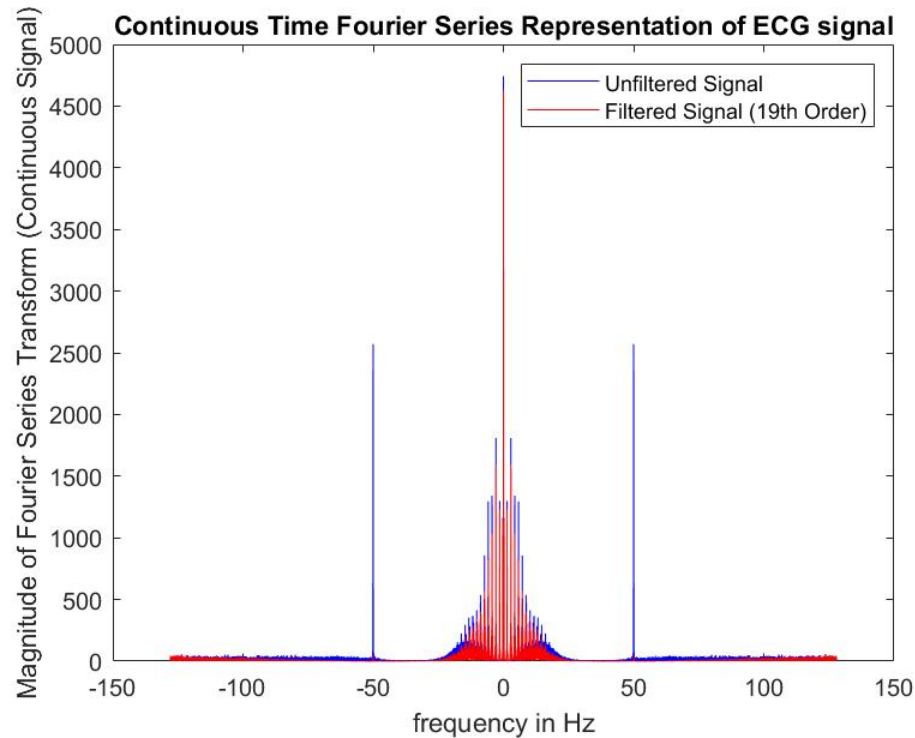
Order: 9 ($L = 10$).



Choice of parameters:

Cut-off frequencies: 49 – 51 Hz

Order: 19 ($L = 20$). (Chosen Filter)



Question 2(c): How does filtering affect the signal in the frequency domain?

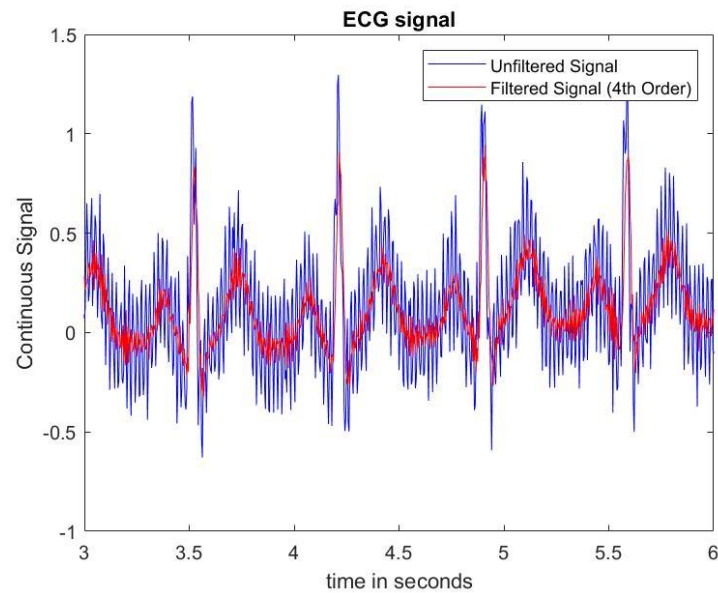
When the narrow band stop filter (notch filter) is applied, the signal in the frequency domain around the power line frequency can be seen filtered out. The frequency around the cut-off frequency can also be seen filtered out but it is not as substantial as the power line frequency.

Question 2(d): On the same set of axes, plot the signal in the time domain before and after filtering. Include your labeled plot, below. Restrict your plot to an interval of time that contains a few heart beats, so it is easy to compare the original and filtered signals. Do this for each choice of parameters (cutoff frequencies and filter lengths).

Choice of parameters:

Cut-off frequencies: 49 – 51 Hz

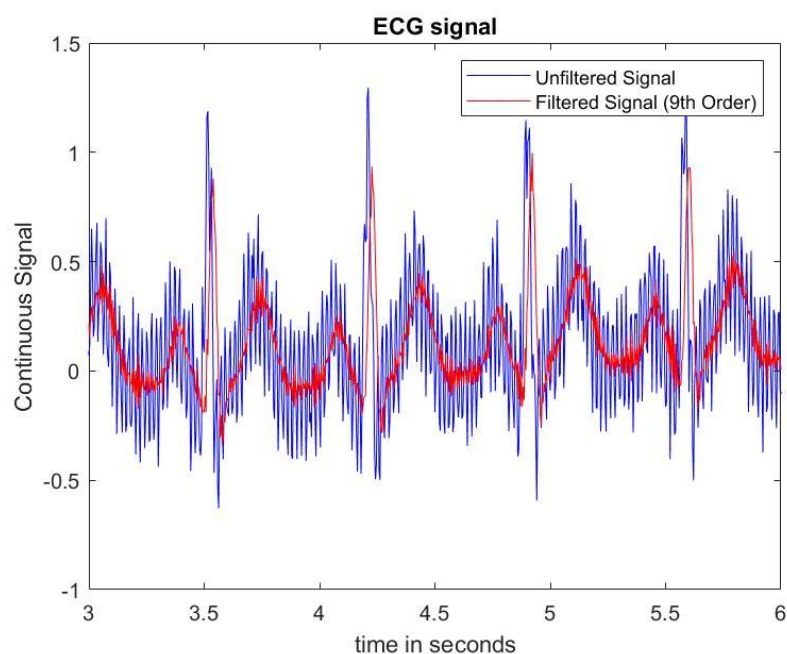
Order: 4 ($L = 5$).



Choice of parameters:

Cut-off frequencies: 49 – 51 Hz

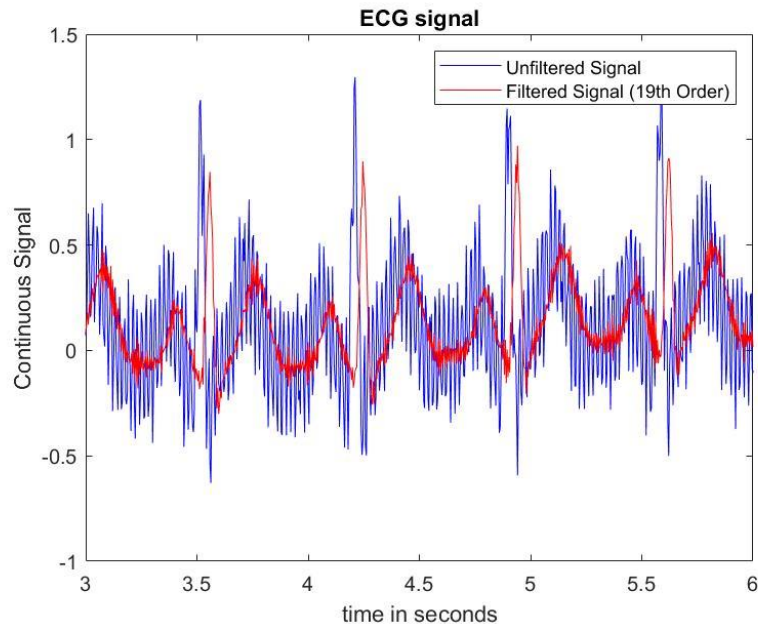
Order: 9 ($L = 10$).



Choice of parameters:

Cut-off frequencies: 49 – 51 Hz

Order: 19 ($L = 20$).



Question 2(e): How does filtering affect the signal in the time domain?

The signal can be seen smoothen out and the signal can be seen delayed by a few centi-seconds.

Question 2(f): Calculate the time shift of the output signal with respect to the input signal of each filter. Give your answer in seconds and in samples. (Hint: look at the time domain plots.) Do this for each choice of parameters (cutoff frequencies and filter lengths).

The time shift of the output signal with respect to the input signal of each filter can be calculated by first getting the total number of samples that got delayed. This can be obtained by using the following formula: $(L-1)/2$

For each of the different length of the filter, we can obtain

$L = 5$, it will delay by $(5-1)/2 = 2$ samples

$L = 10$, it will delay by $(10-1)/2 = 4.5$ samples ~ 5 samples

$L = 20$, it will delay by $(20-1)/2 = 9.5$ samples ~ 10 samples

Using the calculated total number of samples that got delayed, we can find the time delayed by multiplying the above numbers with the sampling rate which is 256 samples per second.

$L = 5$, 2 samples $\times 1/256 = 7.83\text{e-}3$ seconds

$L = 10$, 5 samples $\times 1/256 = 0.0195$ seconds

$L = 20$, 10 samples $\times 1/256 = 0.0391$ seconds

Question 3: How does the filter length affect the time delay that you observe? Why do you think this delay occurs?

The longer the filter length, the larger the time delay observed. The delay occurs because of the phase shift of the signal which results from the filtering of the signal.

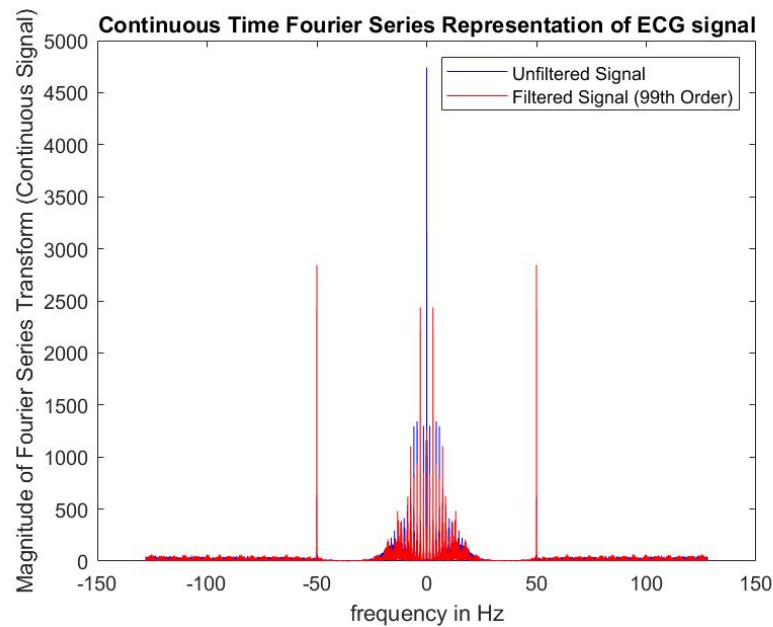
Section 3:

Question 2(b): Plot the signal in the frequency domain before and after filtering (on the same set of axes). Include your labeled plot, below. Do this for each choice of filter length.

Choice of parameters:

Cut-off frequencies: 1 Hz

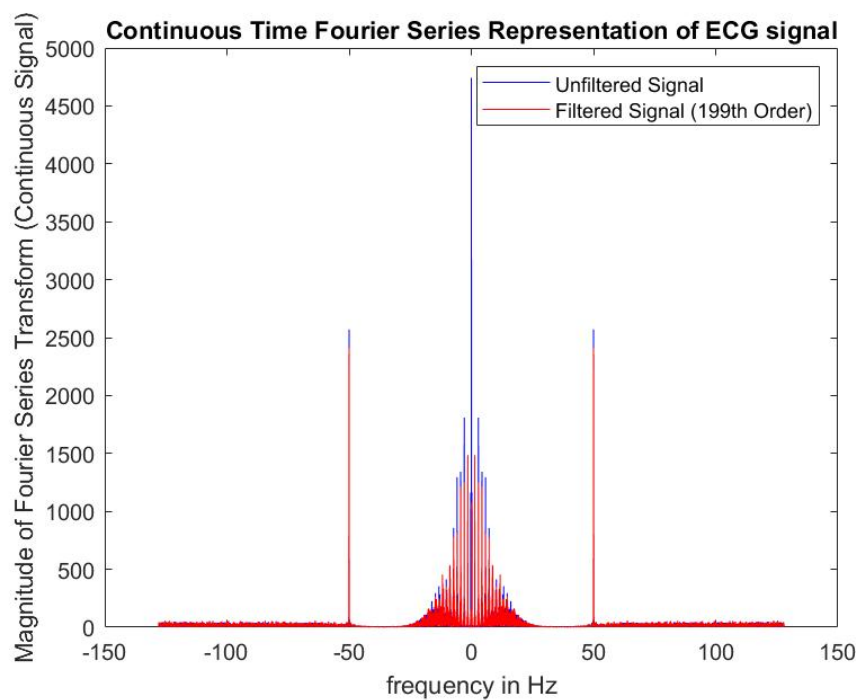
Order: 99 ($L = 100$).



Choice of parameters:

Cut-off frequencies: 1 Hz

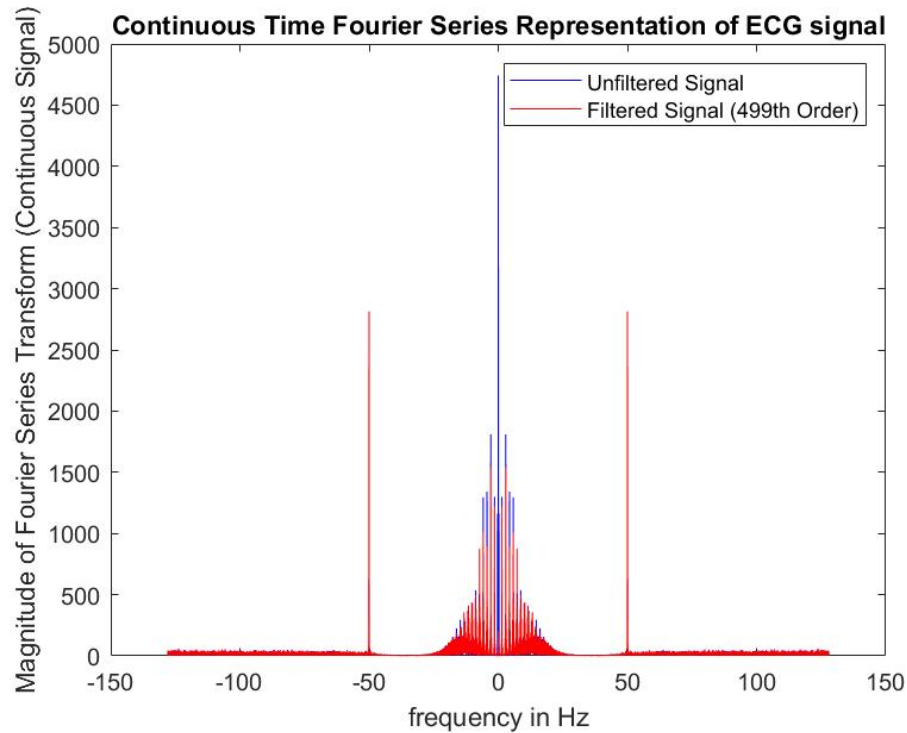
Order: 199 ($L = 200$).



Choice of parameters:

Cut-off frequencies: 1 Hz

Order: 499 (L = 500) (Chosen Filter)



Question 2(c): How does filtering affect the signal in the frequency domain?

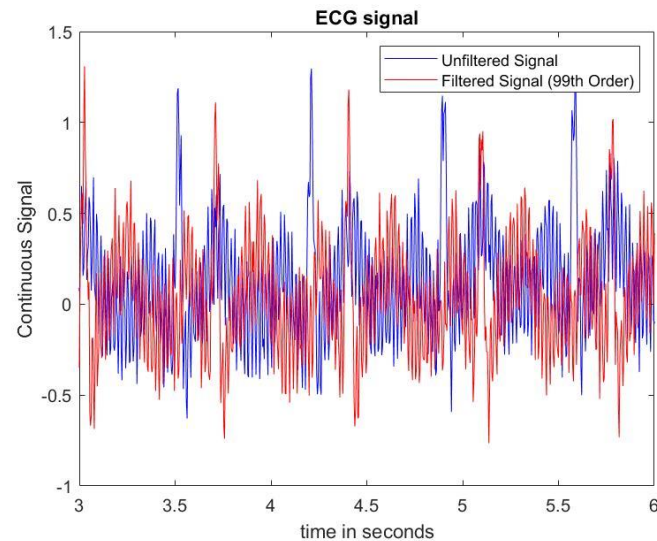
The signal with frequency below 1 Hz will be filtered out.

Question 2(d): Plot the signal in the time domain (over a short segment that displays its features well) before and after filtering (on the same set of axes). Do this for each choice of filter length.

Choice of parameters:

Cut-off frequencies: 1 Hz

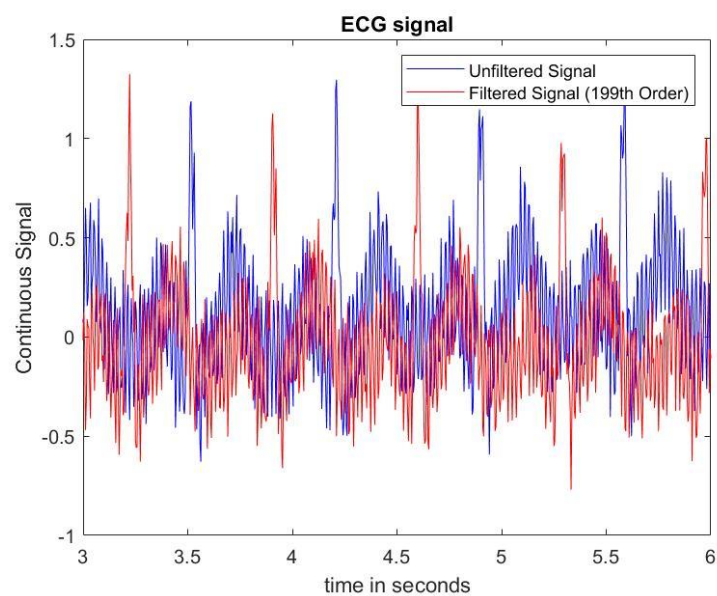
Order: 99 ($L = 100$).



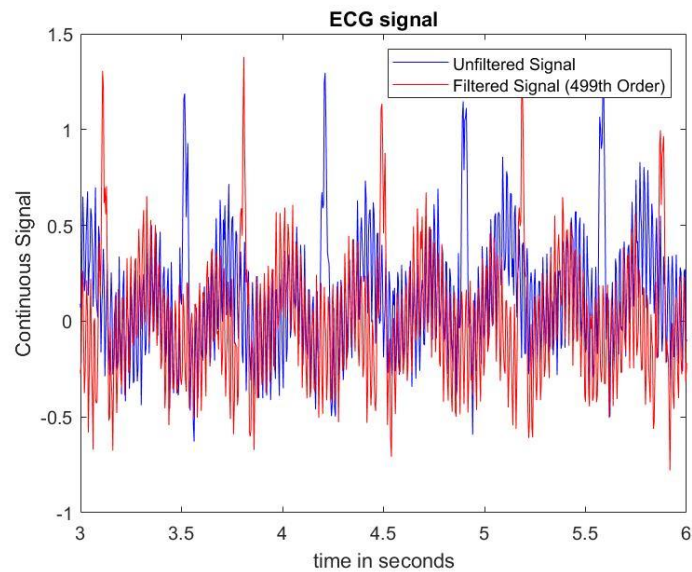
Choice of parameters:

Cut-off frequencies: 1 Hz

Order: 199 ($L = 200$).



Choice of parameters:
Cut-off frequencies: 1 Hz
Order: 499 ($L = 500$).



Question 2(e): How does filtering affect the signal in the time domain?

The signal in the time domain will be delayed after passing through the high-pass filter. The overall filtered signal has a lower amplitude.

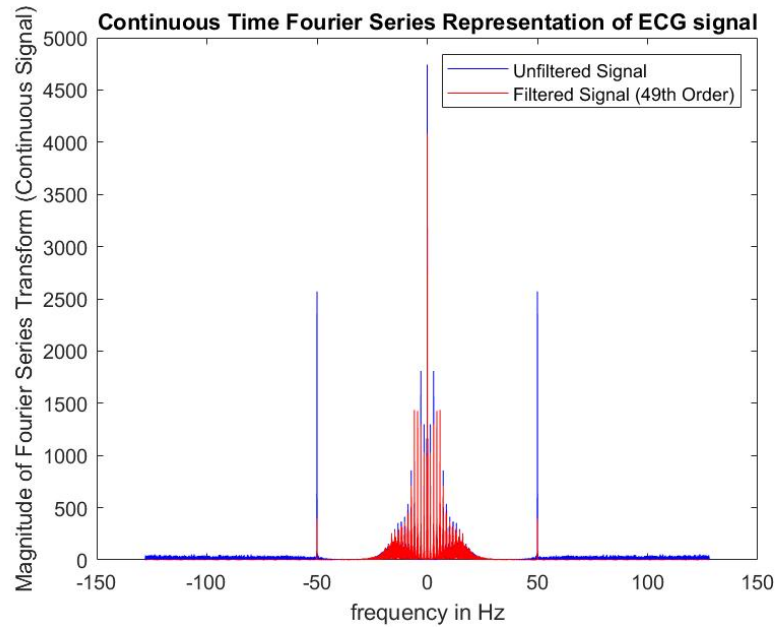
Section 4:

Question 2(b): Plot the signal in the frequency domain before and after filtering (on the same set of axes). Include your labeled plot, below. Do this for each choice of filter length.

Choice of parameters:

Cut-off frequencies: 40 Hz

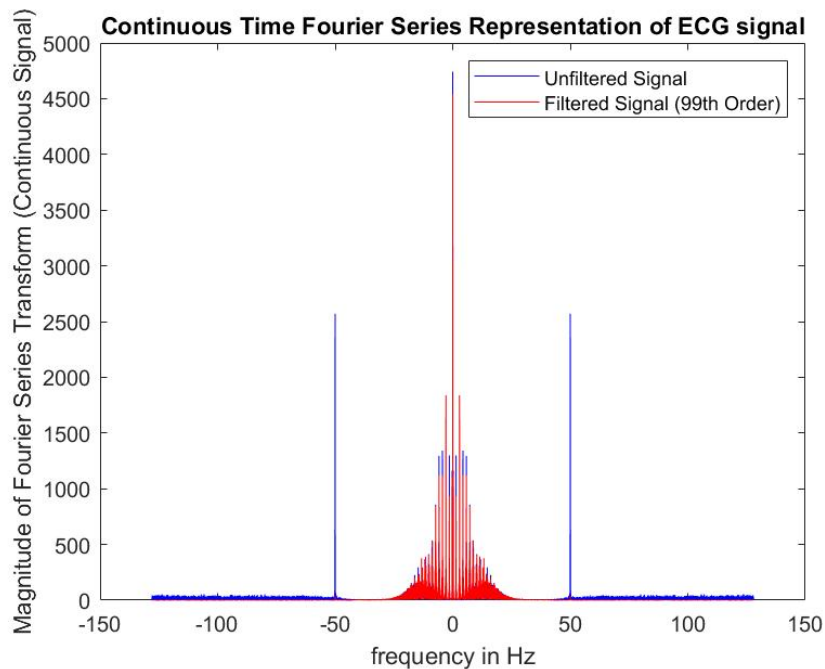
Order: 49 ($L = 50$).



Choice of parameters:

Cut-off frequencies: 40 Hz

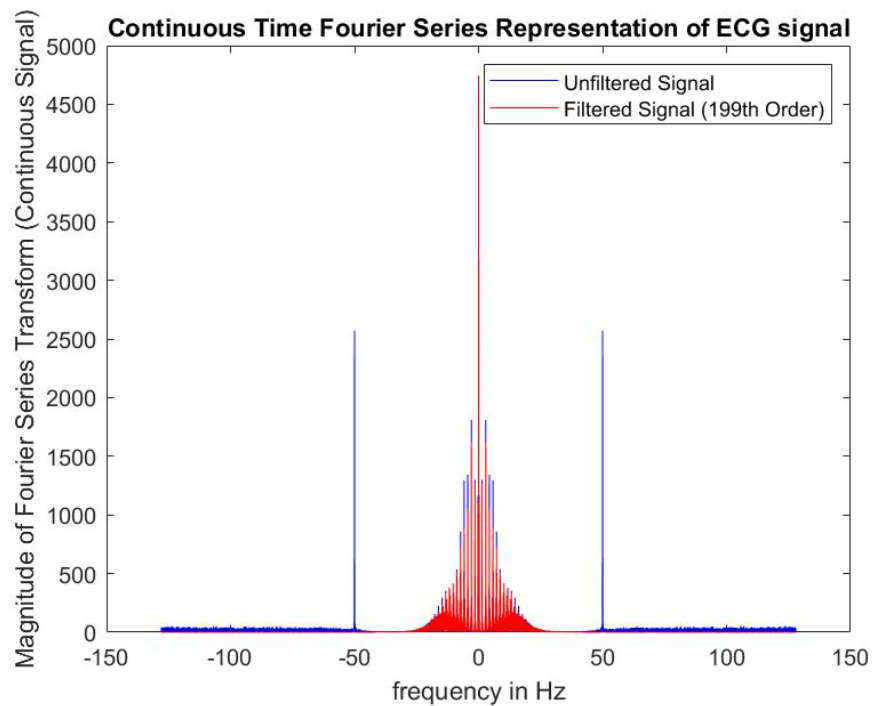
Order: 99 ($L = 100$).



Choice of parameters:

Cut-off frequencies: 1 Hz

Order: 199 (L = 200). (Chosen Filter)



Question 2(c): How does filtering affect the signal in the frequency domain?

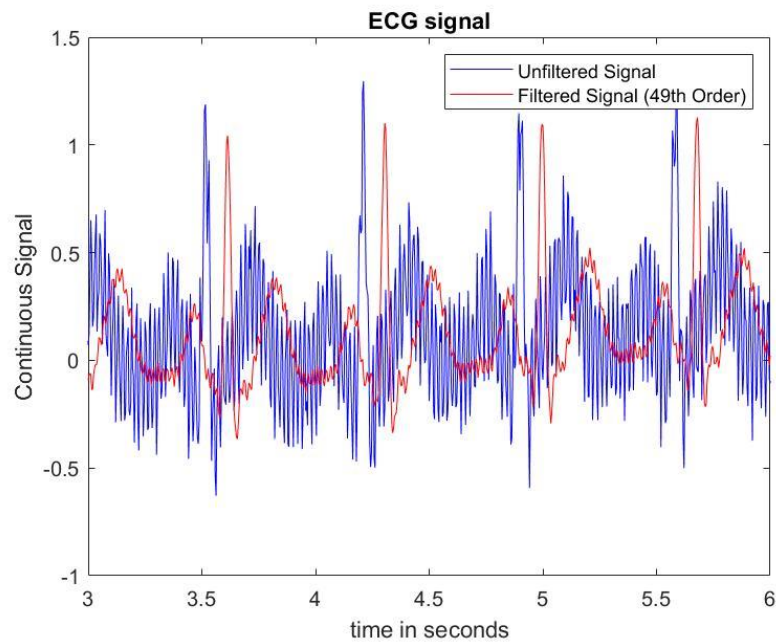
The signal with frequency above 40 Hz will be filtered out.

Question 2(d): Plot the signal in the time domain (over a short segment that displays its features well) before and after filtering (on the same set of axes). Do this for each choice of filter length.

Choice of parameters:

Cut-off frequencies: 1 Hz

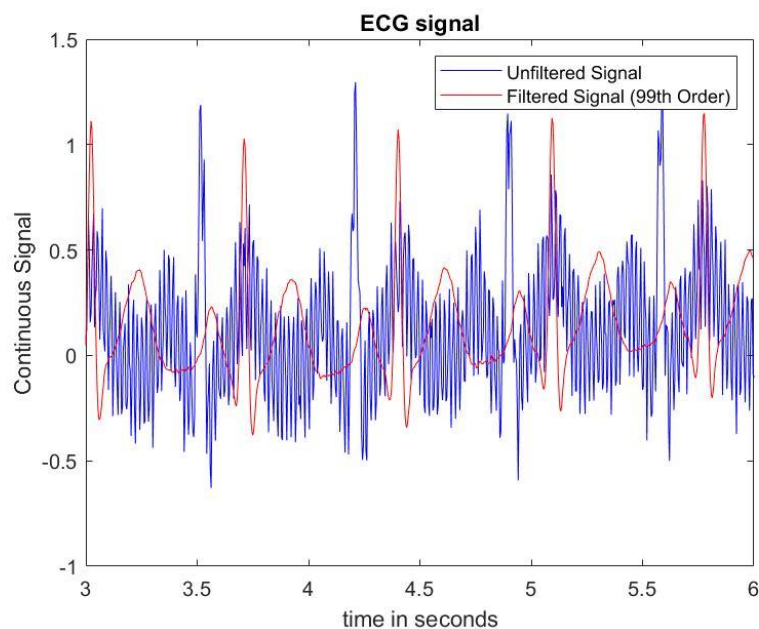
Order: 49 ($L = 50$).



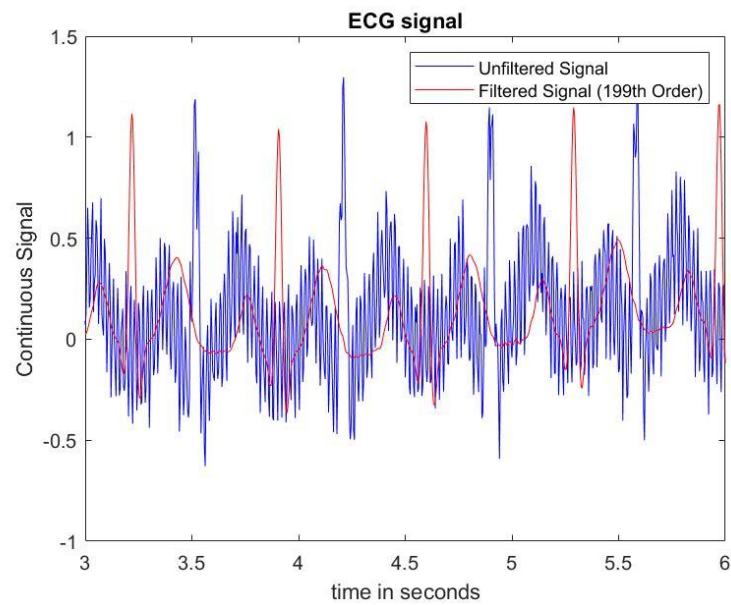
Choice of parameters:

Cut-off frequencies: 1 Hz

Order: 99 ($L = 100$).



Choice of parameters:
Cut-off frequencies: 1 Hz
Order: 199 (L = 200).

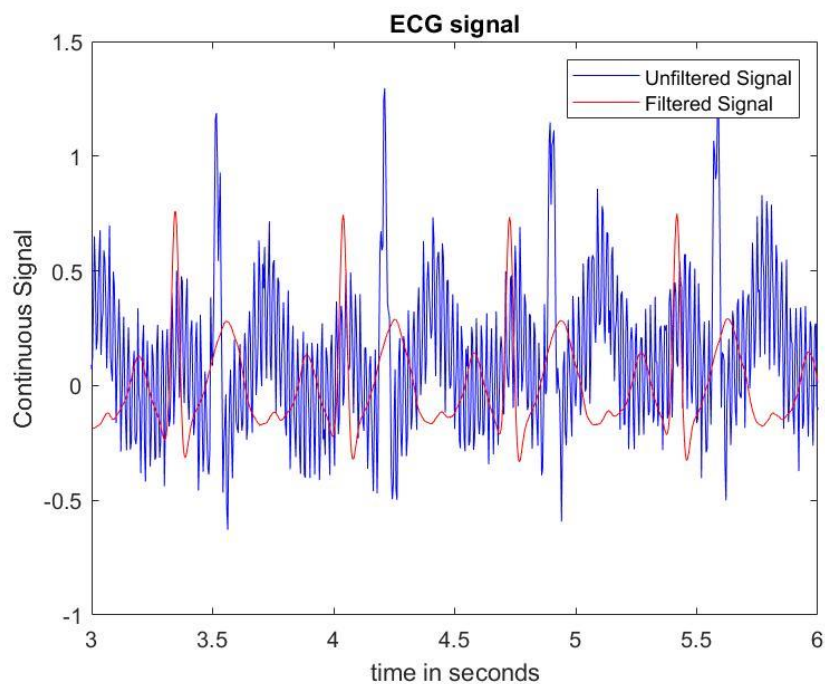
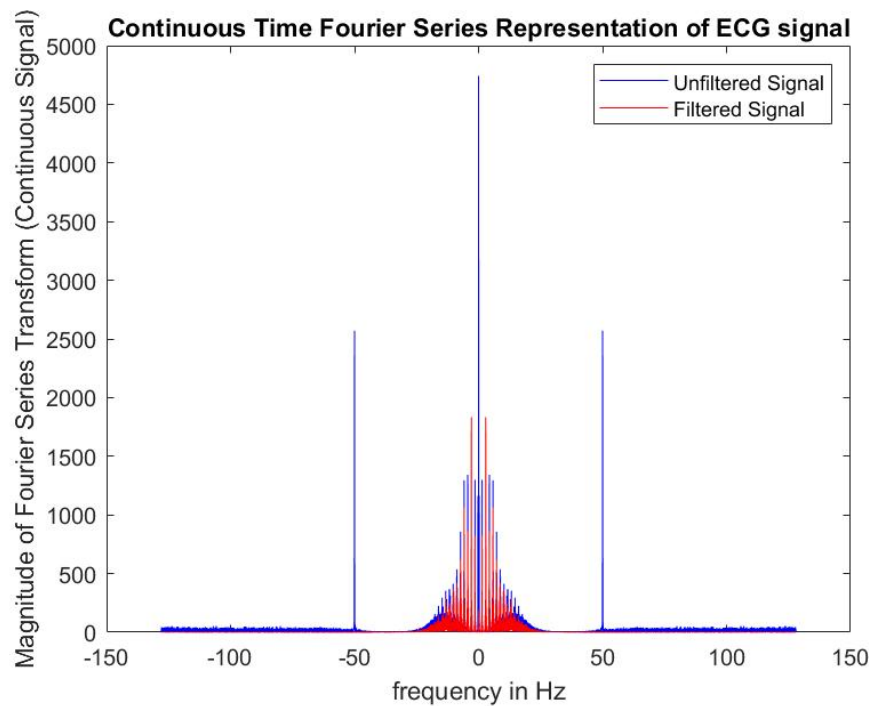


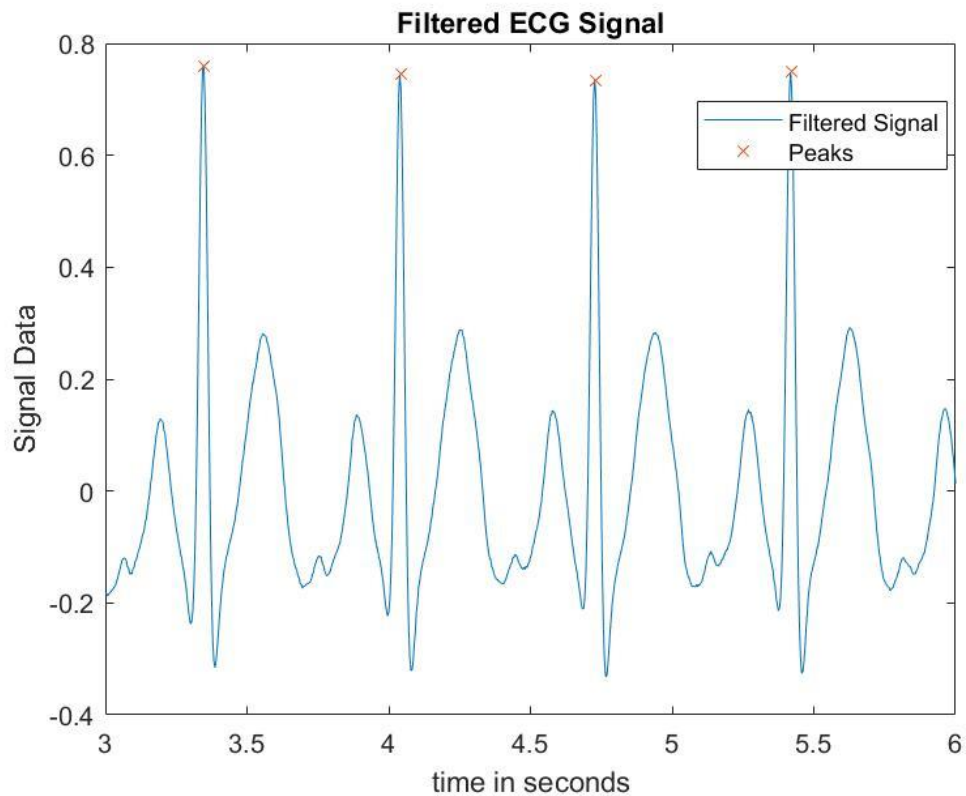
Question 2(e): How does filtering affect the signal in the time domain?

The low-pass filter will filter out the signal, resulting in a smoother signal as the most of the noise lies in the frequency range of above 40 Hz.

Section 5:

Question 4: Plot the filtered ECG in the time-domain over a short segment that shows the features of the ECG. Mark the location of identified R-peaks. Include your labeled plot, below.





Question 7: What average heart rate do you find? How did you compute it?

The average heart rate that I found is 87.11. The average heart rate is computed by first getting the location of the peaks by using the `findpeak` command. Next, we take the index and use this index to convert it by time by dividing it with the sampling rate. Then, we can calculate the BPM by using the formula: $60 / \text{RR-Interval}$. Do this for each of the RR-interval in second. From here, we get the mean of the BPM vectors and we will get our answer.

Code for section 1:

Code from section 1.1:

Paste your script in here.

```
% Written by Tan Jin Chun
% Last Modified : 3/9/2021
% Lab04T01

clear all;close all;clc

%% Question 1
% Loading the signals
% Sampled at 256 samples/second
load ECG.mat;

%% Question 2
% Question 2(a)
% Getting the length of the signal
N = length(ECG);

% Computing the discrete time Fourier Series Coefficient
% Use the fft() function
X = fft(ECG);
X1 = (1/N)*fft(ECG);

% Figure X1
% Plotting the magnitude of the fourier coefficient
figure;

% Plotting the magnitude
stem(-floor(N/2):(N-1-floor(N/2)), fftshift(abs(X1)));

% Labelling the graph
title("Discrete Time Fourier Series Coefficients of ECG signal");
xlabel("k (multiple of fundamental frequency)");
ylabel("Magnitude of Fourier Series Coefficients");

% Question 2(b)
% Plotting the representation of the discrete time Fourier transform
omega = (-floor(N/2):(N-1-floor(N/2)))*(2*pi/N);

% New figure
figure;
plot(omega, fftshift(abs(X)));
title("Discrete Time Fourier Series Representation of ECG signal");
xlabel("frequency in rad/sample");
ylabel("Magnitude of Fourier Series Representation");

% Question 2(c)
% Updated Omega value
Fs = 256;
omega = (-floor(N/2):(N-1-floor(N/2)))*(Fs/N);

% New figure
```

```

figure;
plot(omega, fftshift(abs(X)));
title("Continuous Time Fourier Series Representation of ECG
signal");
xlabel("frequency in Hz");
ylabel("Magnitude of Fourier Series Transform (Continuous Signal)");

%% Question 3
% The power line frequency is -50,0 and 50

%% Question 4
% Plotting the signal in the time domain that shows the signal well
% Getting the time domain
t = (0:length(ECG)-1) / Fs;

% Plotting the signal
figure;
plot(t,ECG);

% Setting the xlimit from 0 to 3 seconds
xlim([0 3]);

% Setting the labelling
xlabel("time in seconds");
ylabel("Continuous Signal");
title("ECG signal");

```

Code for section 2

Code from section 2.2:

Paste your script in here.

% Written by Tan Jin Chun

% Last Modified : 3/9/2021

% Lab04T02

clear all;close all;clc

% Question 1

% Loading the data

load ECG.mat;

% Designing low-pass filter to remove any high frequency noise above 40Hz

% Initialising the variable

% Lengths of the filters

L1 = 5;

L2 = 10;

L3 = 20;

Fs = 256;

% Getting the order of the signal

O1 = L1-1;

O2 = L2-1;

O3 = L3-1;

Order = [O1 O2 O3];

% Designing the low pass filters

% Using 49Hz and 51Hz as we want to filter out the Power line Frequency

% which is 50 Hz

Hz = [49 51];

% Omega in terms of pi

w = 2 * Hz / Fs;

% Using the firpm function to design the notch filter

% Using a for loop to plot the six graphs

for i = 1:length(Order)

h = firpm(Order(i), [0, w, 1], [1,0,0,1]);

%% Question 2

% Question 2(a)

% String for the legend

str = sprintf("Filtered Signal (%dth Order)", Order(i));

% Applying the filter to my signal using the conv function

y1 = conv(ECG, h);

% Question 2(b)

% Plotting the signal in the frequency domain before filtering

N = length(ECG);

X = fft(ECG);

```

% Updated Omega value
omega = (-floor(N/2):(N-1-floor(N/2)))*(Fs/N);

% New figure
figure;
plot(omega, fftshift(abs(X)), 'b');
title("Continuous Time Fourier Series Representation of ECG
signal");
xlabel("frequency in Hz");
ylabel("Magnitude of Fourier Series Transform (Continuous
Signal)");

% Plotting the signal in the frequency domain after filtering
N1 = length(y1);
X1 = fft(y1);
omega1 = (-floor(N1/2):(N1-1-floor(N1/2)))*(Fs/N1);
hold on
plot(omega1, fftshift(abs(X1)), 'r');
legend("Unfiltered Signal", str);
hold off

%% Question 2(d)
% Plotting the signal in the time domain that shows the signal
well
% Getting the time domain
t = (0:length(ECG)-1) / Fs;

% Plotting the signal
figure;
plot(t, ECG, 'b');

% Setting the xlimit from 0 to 3 seconds
xlim([3 6]);

% Setting the labelling
xlabel("time in seconds");
ylabel("Continuous Signal");
title("ECG signal");

% Plotting the filtered signal in the time domain
t1 = (0:length(y1)-1) / Fs;
hold on
plot(t1, y1, 'r');
legend("Unfiltered Signal", str);
hold off

end

```

Code for section 3

Code from section 3.1:

Paste your script in here.

```
% Written by Tan Jin Chun
% Last Modified : 3/9/2021
% Lab04T03
```

```
clear all;close all;clc
```

```
% Question 1
% Loading the data
load ECG.mat;
```

```
% Designing low-pass filter to remove any high frequency noise above 40Hz
```

```
% Initialising the variable
% Lengths of the filters
```

```
L1 = 100;
L2 = 200;
L3 = 500;
Fs = 256;
```

```
% Getting the order of the signal
```

```
O1 = L1-1;
O2 = L2-1;
O3 = L3-1;
Order = [O1 O2 O3];
```

```
% Designing the high-pass filters
```

```
Hz = [0.6 1];
```

```
% Omega in terms of pi
```

```
w = 2 * Hz / Fs;
```

```
% Designing my high pass filter using the firpm function
```

```
% Just change the order to O1, O2 or O3 to tinker around with diff number
```

```
% of filter length
```

```
% Using a for loop to plot all 6 plots
```

```
for i = 1:length(Order)
    h = firpm(Order(i), [0,w,1], [0,0,1,1]);
```

```
%% Question 2
```

```
    % Question 2(a)
```

```
    % String for the legend
```

```
    str = sprintf("Filtered Signal (%dth Order)", Order(i));
```

```
    % Applying the filter to my signal using the conv function
```

```
    y1 = conv(ECG, h);
```

```
    % Question 2(b)
```

```
    % Plotting the signal in the frequency domain before filtering
```

```
    N = length(ECG);
```

```
    X = fft(ECG);
```



```

% Updated Omega value
omega = (-floor(N/2):(N-1-floor(N/2)))*(Fs/N);

% New figure
figure;
plot(omega, fftshift(abs(X)), 'b');
title("Continuous Time Fourier Series Representation of ECG
signal");
xlabel("frequency in Hz");
ylabel("Magnitude of Fourier Series Transform (Continuous
Signal)");

% Plotting the signal in the frequency domain after filtering
N1 = length(y1);
X1 = fft(y1);
omega1 = (-floor(N1/2):(N1-1-floor(N1/2)))*(Fs/N1);
hold on
plot(omega1, fftshift(abs(X1)), 'r');
legend('Unfiltered Signal', str);
hold off

%% Question 2(c)

%% Question 2(d)
% Plotting the signal in the time domain that shows the signal
well
% Getting the time domain
t = (0:length(ECG)-1) / Fs;

% Plotting the signal
figure;
plot(t, ECG, 'b');

% Setting the xlimit from 0 to 3 seconds
xlim([3 6]);

% Setting the labelling
xlabel("time in seconds");
ylabel("Continuous Signal");
title("ECG signal");

% PLOtting the filtered signal
t1 = (0:length(y1)-1) / Fs;
hold on
plot(t1, y1, 'r');
legend("Unfiltered Signal", str);
hold off

end

```

Code for section 4

Code from section 4.1:

Paste your script in here.

```
% Written by Tan Jin Chun
% Last Modified : 2/9/2021
% Lab04T04
```

```
clear all;close all;clc
```

```
% Question 1
% Loading in the signal
load ECG.mat;
```

```
% Designing low-pass filter to remove any high frequency noise above 40Hz
```

```
% Initialising the variable
% Lengths of the filters
```

```
L1 = 50;
```

```
L2 = 100;
```

```
L3 = 200;
```

```
Fs = 256;
```

```
% Getting the order of the signal
```

```
O1 = L1-1;
```

```
O2 = L2-1;
```

```
O3 = L3-1;
```

```
Order = [O1 O2 O3];
```

```
% Designing the low-pass filters
```

```
Hz = [40 43];
```

```
% Omega in terms of pi
```

```
w = 2 * Hz / Fs;
```

```
% Question 2(a)
```

```
% Designing my high pass filter using the firpm function
```

```
% Using a for loop to plot all 6 plots
```

```
for i = 1:length(Order)
```

```
    h = firpm(Order(i), [0, w, 1], [1,1,0,0]);
```

```
    %% Question 2
```

```
    % String for the legend
```

```
    str = sprintf("Filtered Signal (%dth Order)", Order(i));
```

```
    % Applying the filter using the conv function
```

```
    y2 = conv(ECG, h);
```

```
    % Question 2(b)
```

```
    % Plotting the signal in the frequency domain before and after filtering
```

```
    % Updated Omega value
```

```
    % Initialising the variable
```

```
    % Getting the length of the signal
```

```
    N = length(ECG);
```

```

% Computing the discrete time Fourier Series Coefficient
% Use the fft() function
X = fft(ECG);

% Getting the value of omega
omega = (-floor(N/2):(N-1-floor(N/2)))*(Fs/N);

% New figure
figure;
plot(omega, fftshift(abs(X)), 'b');
title("Continuous Time Fourier Series Representation of ECG
signal");
xlabel("frequency in Hz");
ylabel("Magnitude of Fourier Series Transform (Continuous
Signal)");

% Initialising the variable for filtered signal
N2 = length(y2);
X2 = fft(y2);

omega2 = (-floor(N2/2):(N2-1-floor(N2/2)))*(Fs/N2);

% Plotting the filtered signal
hold on
plot(omega2, fftshift(abs(X2)), 'r');
legend("Unfiltered Signal", str);
hold off

%% Question 2(c)

%% Question 2(d)
% Plotting the signal in the time domain that shows the signal
well
% Getting the time domain
t = (0:length(ECG)-1) / Fs;

% Plotting the signal
figure;
plot(t, ECG, 'b');

% Setting the xlimit from 0 to 3 seconds
xlim([3 6]);

% Setting the labelling
xlabel("time in seconds");
ylabel("Continuous Signal");
title("ECG signal");

% Plotting the filtered signal in the time domain
t2 = (0:length(y2)-1) / Fs;
hold on
plot(t2, y2, 'r');
legend("Unfiltered Signal", str);

```

```
        hold off
end

%% Question 2(e)
```

Code for section 5

Code from section 5.1:

Paste your script in here.

% Written by Tan Jin Chun

% Last Modified : 2/9/2021

% Lab04T05

clear all;close all;clc

% Loading in the ECG.mat signal

load ECG.mat;

% Question 1

% First System (Bandstop Filter)

% Sampling Rate

Fs = 256;

% Omega in terms of pi

w1 = 2 * [49 51] / Fs;

w2 = 2 * [0.6 1] / Fs;

w3 = 2 * [40 43] / Fs;

% Using the firpm function to design the notch filter

h1 = firpm(19, [0,w1,1], [1,0,0,1]);

% Second System (high-pass filter)

h2 = firpm(499, [0,w2,1], [0,0,1,1]);

% Third System (low-pass filter)

h3 = firpm(199, [0,w3,1], [1,1,0,0]);

% Using the conv function for the series interconnection of the three

% system

y1 = conv(ECG, h1);

y2 = conv(y1, h2);

y3 = conv(y2, h3);

% Plotting the frequency domain graph

% Question 2(b)

% Plotting the signal in the frequency domain before filtering

N = length(ECG);

X = fft(ECG);

% Updated Omega value

omega = (-floor(N/2):(N-1-floor(N/2)))*(Fs/N);

% New figure

figure;

plot(omega, fftshift(abs(X)), 'b');

title("Continuous Time Fourier Series Representation of ECG
signal");

xlabel("frequency in Hz");

ylabel("Magnitude of Fourier Series Transform (Continuous Signal)");

```

% Plotting the signal in the frequency domain after filtering
N1 = length(y3);
X1 = fft(y3);
omega1 = (-floor(N1/2):(N1-1-floor(N1/2)))*(Fs/N1);
hold on
plot(omega1, fftshift(abs(X1)), 'r');
legend("Unfiltered Signal", "Filtered Signal");
hold off

% Plotting the time domain graph
% Getting the time domain
t = (0:length(ECG)-1) / Fs;

% Plotting the signal
figure;
plot(t, ECG, 'b');

% Setting the xlimit from 0 to 3 seconds
xlim([3 6]);

% Setting the labelling
xlabel("time in seconds");
ylabel("Continuous Signal");
title("ECG signal");

% Plotting the filtered signal
t3 = (0:length(y3)-1) / Fs;
hold on
plot(t3, y3, 'r');
legend("Unfiltered Signal", "Filtered Signal")
hold off

% Question 2 and Question 3
% Using the function findpeaks and finding the appropriate threshold
and
% Selecting the appropriate threshold to select only the peaks that
% correspond to R-peaks
threshold = 0.5;
[pks, locs] = findpeaks(y3, 'MinPeakHeight', threshold);

% Question 4
max_loc = max(t)*Fs;
min_loc = min(t)*Fs;
plot_loc = locs(locs <= max_loc & locs >= min_loc);
plot_pks = pks(locs <= max_loc & locs >= min_loc);

% Plotting the figure
figure;
plot(t3, y3);
xlabel("time in seconds");
ylabel("Signal Data");
title("Filtered ECG Signal");
hold on
plot(plot_loc/Fs, plot_pks, 'x');
legend("Filtered Signal", "Peaks");

```

```
hold off
xlim([3 6]);

% Question 5
% Calculating the sequence of time intervals between successive R-
peaks
% called (R-R peaks)
HBS_interval = diff(locs./Fs);

% Question 6
% BPM is calculated using 60/rr-interval
BPMS_interval = 60./HBS_interval;
BPM_second = mean(BPMS_interval);
fprintf("The average heartbeat per minute (BPM) is %.2f\n",
BPM_second);
```