Ryerson University

Department of Electrical & Computer Engineering

BME 632 - Signals & Systems II

Lab Report - ECG

Lab Number: 3

Instructor: April Khademi

Section: 2

Due Date: March 22th, 2020

Student Name (ID): [redacted]

Signature: [redacted]

Student Name (ID): **Pass Fail** **Kushal Shah** **500843903**

Signature: KS

By signing above you attest that you have contributed to this written lab report and confirm that all work you have contributed to this lab report is your own work. Any suspicion of copying or plagiarism in this work will result in an investigation of Academic Misconduct and may result in a 0 on the work, an F in the course, or possibly more severe penalties, as well as a Disciplinary Notice on your academic record under the Student Code of Academic Conduct, which can be found online at: www.ryerson.ca/senate/current/pol60.pdf.

Before submitting your report, your TA asks questions about your report. If there is no consistency between your oral answer and your report, you will lose 50% of your total mark. For this part, Pass or fail will be circled next to your name accordingly.

Introduction:

As possibly the most important organ in the human body, it is vital for medical professionals to monitor the activity of the heart. The heart has the important task of pumping oxygenated blood into and throughout the body to provide the necessary nutrients, and to pump the deoxygenated blood to the lungs to expel the carbon dioxide. The pumping action of the heart is created by muscle contractions, which the heart is one huge muscle. To create these muscle contractions electrical signals must be sent to the heart and activate the sinoatrial and atrioventricular nodes. These electrical signals, like all other electrical signals studied in previous labs, can be acquired and measured. This signal is known as an electrocardiogram or ECG. It is commonly referred to as the most important vital signal of the human body according to medical practitioners. The signal itself can be seen as a repetitive waveform with the characteristic waveform shown if figure 1. The waveform is split into different waves such as R-wave and T-wave, and those waves can be paired to create segments or complexes (also shown in figure 1.). Each of these waves correlate to different cardiac events such as ventricular constriction and atrial relaxation. This waveform can also help monitor other medical data such as heart rate, or to find cardiac diseases such as heart attacks or atrial flutter. With all this said, it is very important for biomedical engineers to understand what these signals are and how to process them, for the purpose of making devices which can detect irregular heart functions.

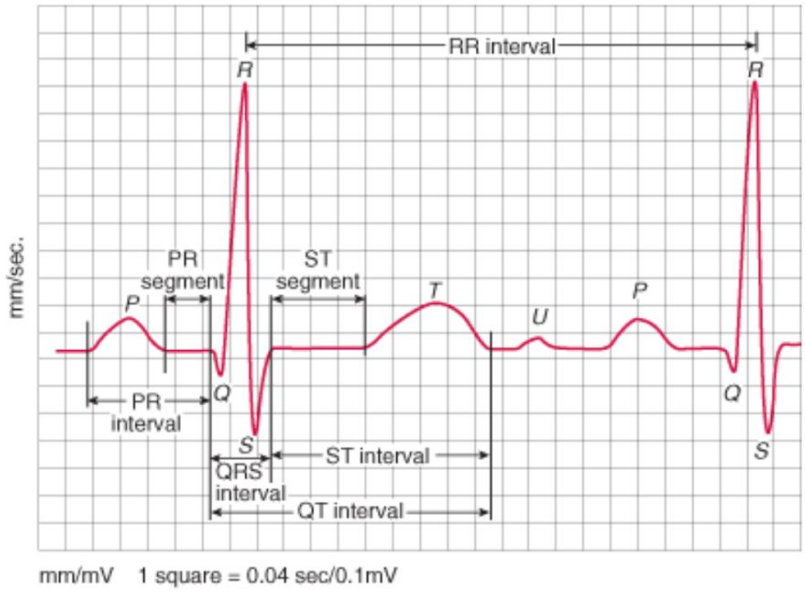

Figure 1. Characteristic waveform of an ECG signal [1].

Pre-Lab Questions:

1. An ecg is a biomedical signal which characterizes cardiac events by the electrical impulses sent to the heart. It itself is characterized by different waves which each of those correspond to different cardiac events.
2. Eithoven's triangle is a vector representation of an ecg signal. In a three lead configuration for an ECG acquisition, each lead is represented by a vector and the three vectors create a triangle, and creates the equation Lead 1 + Lead 3 = Lead 2. The magnitude and angles of the vectors change throughout the cardiac cycle, and the summation of all these vectors create the signal shown in figure 1.
3. FIR and IIR are categories of discrete time filters. IIR stands for Infinite Impulse response and can be unstable as they can infinitely continue while Finite Impulse Response filters are stable. FIR filters are used in applications where linear characteristics are desired while linear characteristics are not a concern when used in IIR applications. Most importantly, FIR filters are causal i.e. the output of the filter depends on the past and present values and require a memory, while IIR filters are non causal and don't require a memory.
   Difference equation for FIR is:
   $$\sum_{k=0}^{M} b_k x(n-k)$$
   And the difference equation for IIR is:
   $$-\sum_{k=1}^{N} a_k y(n-k) + \sum_{k=0}^{M} b_k x(n-k)$$

Experiment 1: NORMAL RESTING ECG

Exercise 1.1:

a)

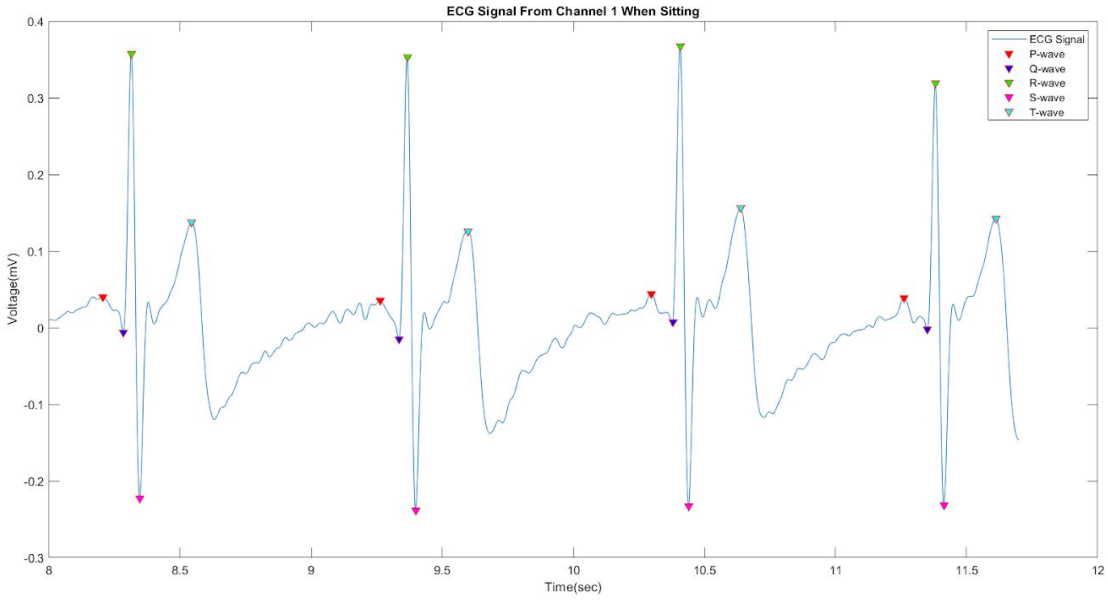

Figure 2. Plot of four full ECG waveforms with their respective PQRST-waves indicated.

b) Calculated Heart Rate = 78 bpm

% we will choose to use the R-wave because it is the most accurate part of the ECG wave to recognize and measure

```
sizeofbeats = length(locs_Rwave);
range1 = locs_Rwave(1);
range2 = locs_Rwave(sizeofbeats);
rangebpm = range2 - range1;
rangebpm = rangebpm/1000;

bps = sizeofbeats/rangebpm;

bpm = bps * 60;
```

The above code uses the number of R-waves and the length of the waveform to calculate heart rate. Heart rate is calculated by the number of beats that happened during a certain period of time. Therefore, I took the length of the signal as the time frame and the number of R-waves as the number of beats.

Exercise 1.2:

a)

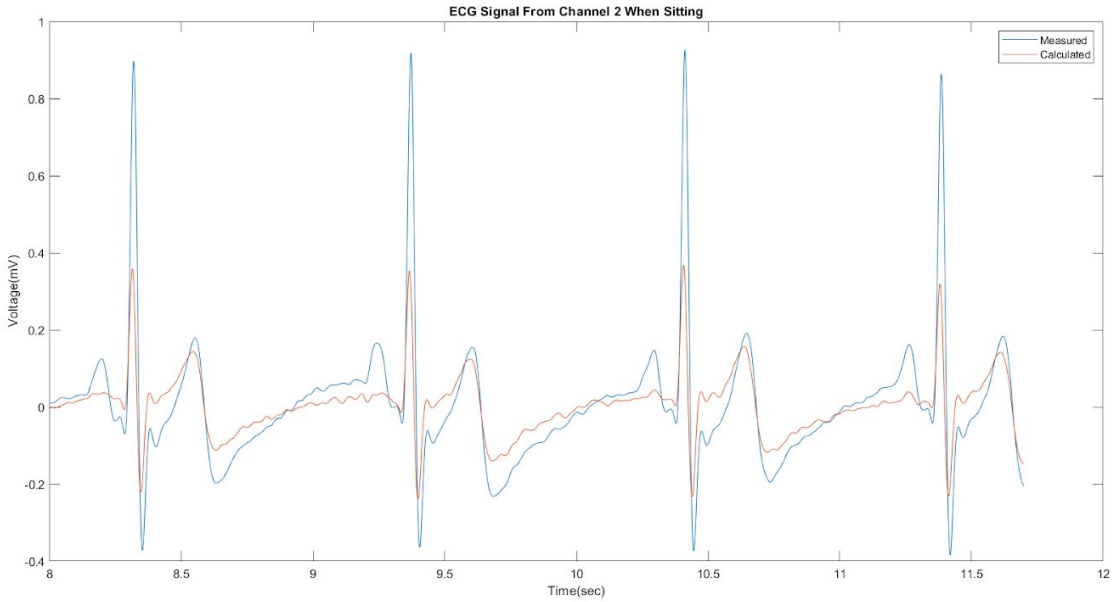

Figure 3. Plot of lead 2 waveforms which compares both the measured wave and the calculated wave using Einthoven's Law.

b)

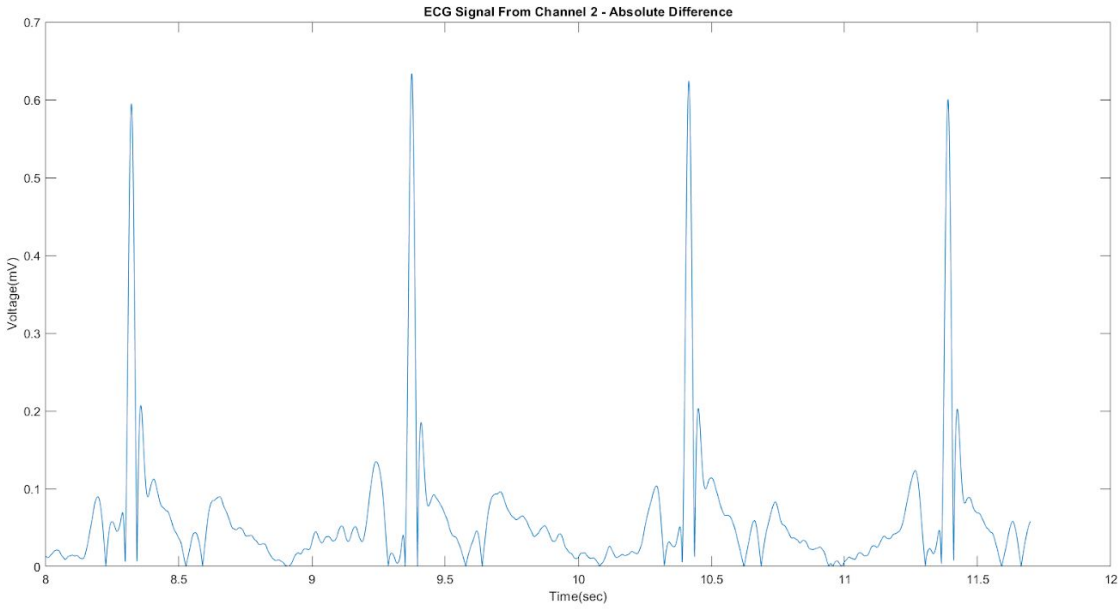

Figure 4. Plot of the absolute difference between the measured and calculated waves from part b.

c) MSE = 0.0031. This value tells us the error between the recorded and calculated lead 2 values. Since the MSE is very low, this means that we are very accurate with our calculations.

```
% c) from main file

[MSE] = meansqrError(sitECG2calc, sitECG2sec)

sitECG2mse = MSE;

%from function file
function [MSE] = meansqrError(signal1, signal2)
% calculating the MSE

signal1(isnan(signal1)) = 0;
signal2(isnan(signal2)) = 0;
```

```
MSE = immse(signal1,signal2);

end
```

Exercise 1.4:

a)

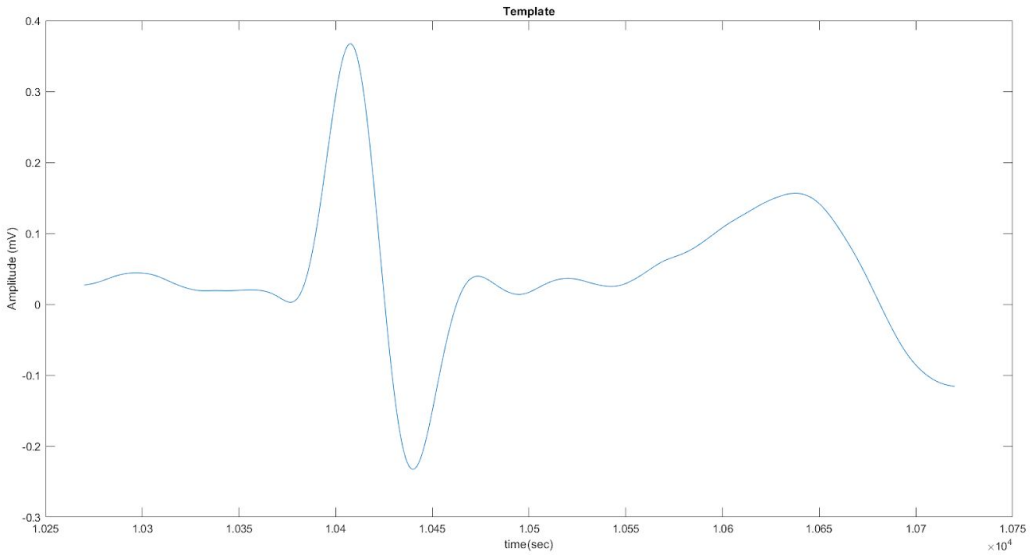

Figure 5. Template wave for cross correlation. From lead 1.

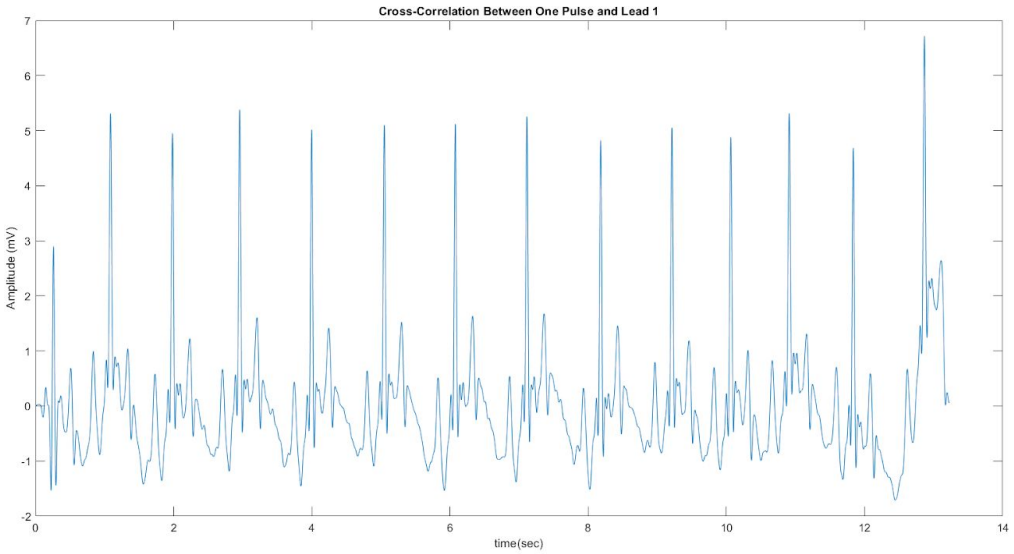

Figure 6. Plot of the cross correlation of lead 1 using the above template waveform.

B & c) The code for the automatic heart rate calculation can be seen in the appendix. The calculated heart rate = 66 bpm. When comparing this heart rate to the one calculated in exercise 1.1, I would say that this calculation is more accurate. This is because of two reasons. Firstly, the first calculation took into account the whole waveform, but this automated way only takes into account the time frame where the R-wave occurred and the time in between them. This would eliminate any time which is not necessary. Secondly, just by the numbers this automated method seems more accurate. The individual was resting and sitting for some time so their heart rate would be low, and the automated method provided us with a lower bpm.

Exercise 1.5:

a) To note: the templates were taken from R-wave to R-wave, therefore the waveforms shown start with R-wave on the left and they end with the P-wave on the right. Padding was required to properly sum the signals, therefore the R-waves might have multiple peaks due to them being shifted.

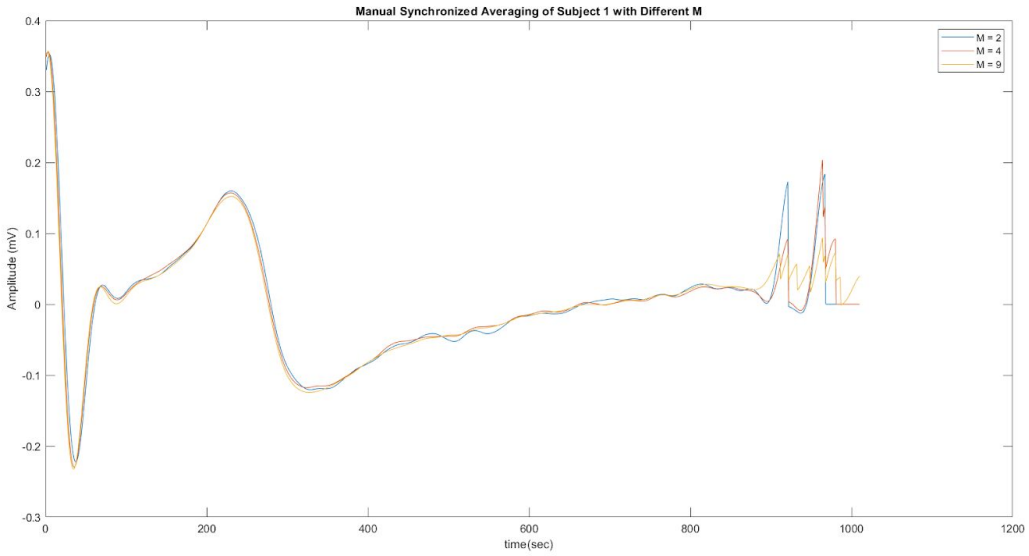

Figure 7. Plot of synchronized averaging for Subject 1 data with different M's. Manually found trigger points.

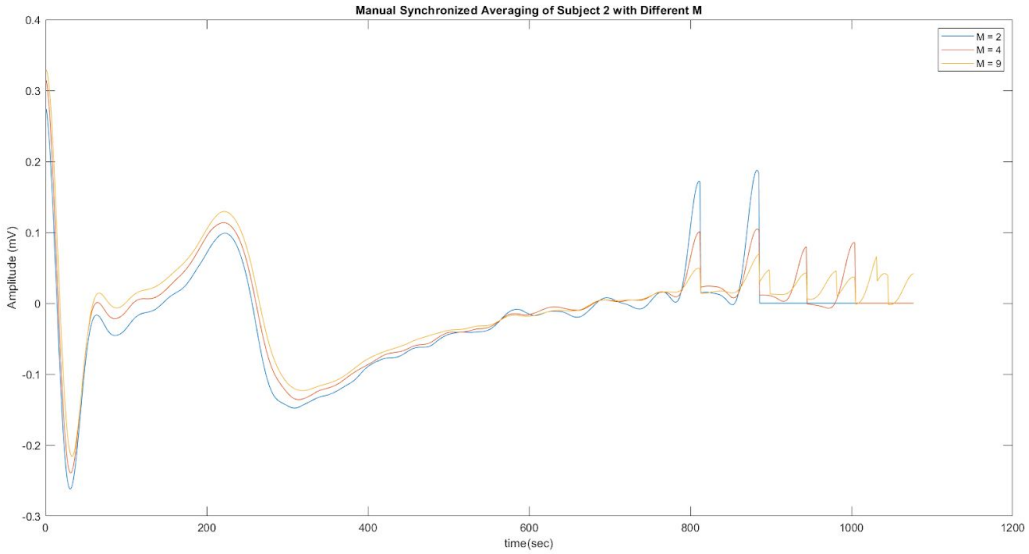

Figure 8. Plot of synchronized averaging for Subject 2 data with different M's. Manually found trigger points.

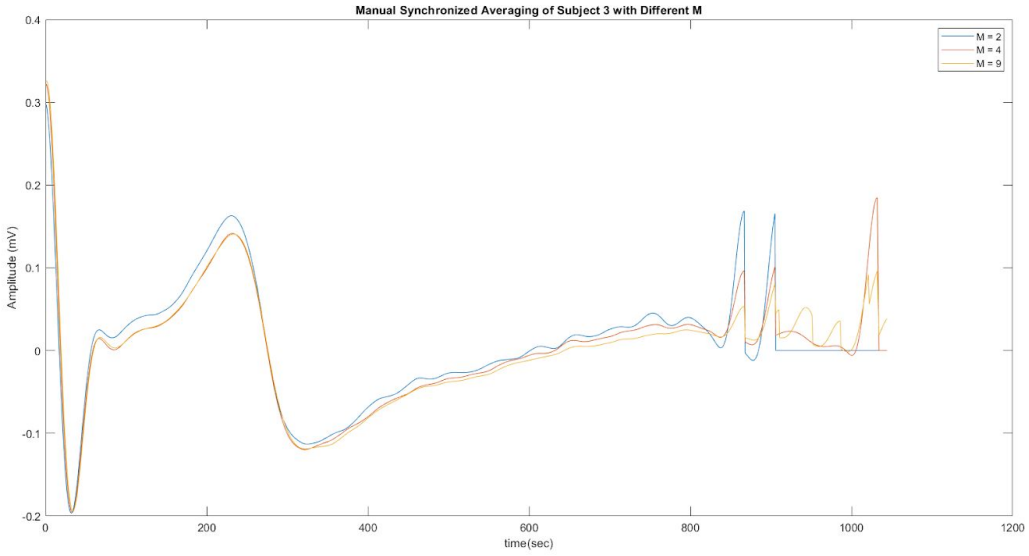

Figure 9. Plot of synchronized averaging for Subject 3 data with different M's. Manually found trigger points.

Firstly, we have to discuss the ending of each of these three plots. Due to the template waves, which were manually found, having different lengths padding was required at the ends to do the proper summation. This has caused these weird bumps near the end, which can be explained by the wave itself. Each section of the wave was chosen from R-wave to R-wave, therefore the

summation was of the middle portion of the waves. Therefore these waves at the end are due to the different positions of the P-wave. The beginning would be the same because we start at the R-wave, which we can see. Secondly, in each of the subjects, we can see that as we increase 'M' the lines become smoother. This is much more evident at the time between the T-wave and P-wave.

b) The manual stated to automate the trigger points, but we decided to automate the whole process. This was accomplished, however, knowing we wanted only M = 2,4,9, it does not take into account any integer value.

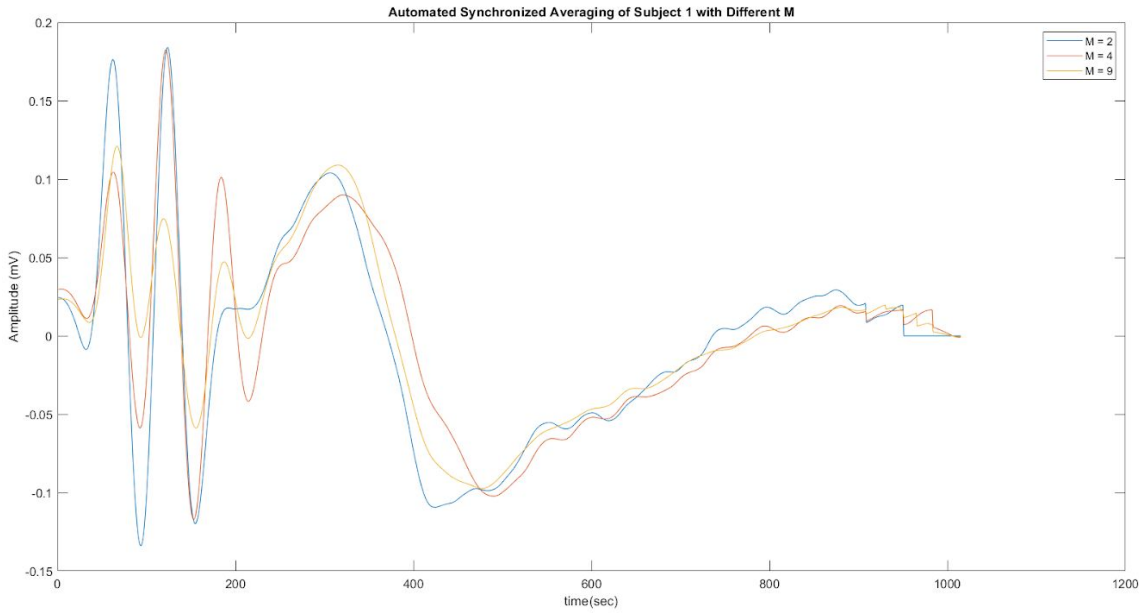

Figure 10. Plot of synchronized averaging for Subject 1 data with different M's. Automatically found trigger points.

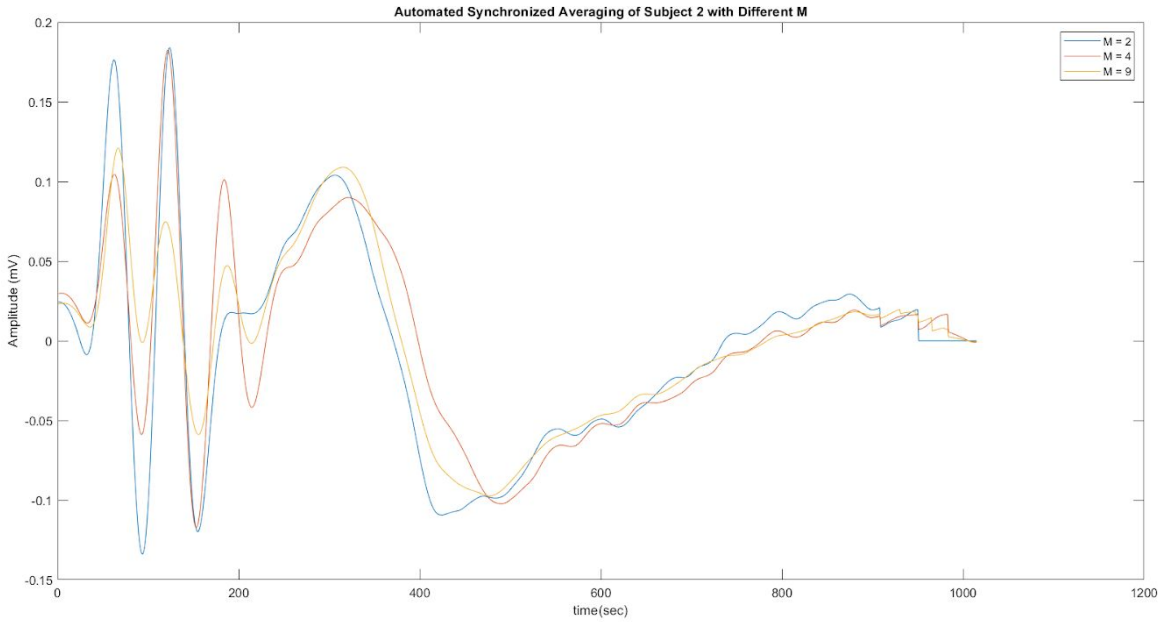

Figure 11. Plot of synchronized averaging for Subject 2 data with different M's. Automatically found trigger points.

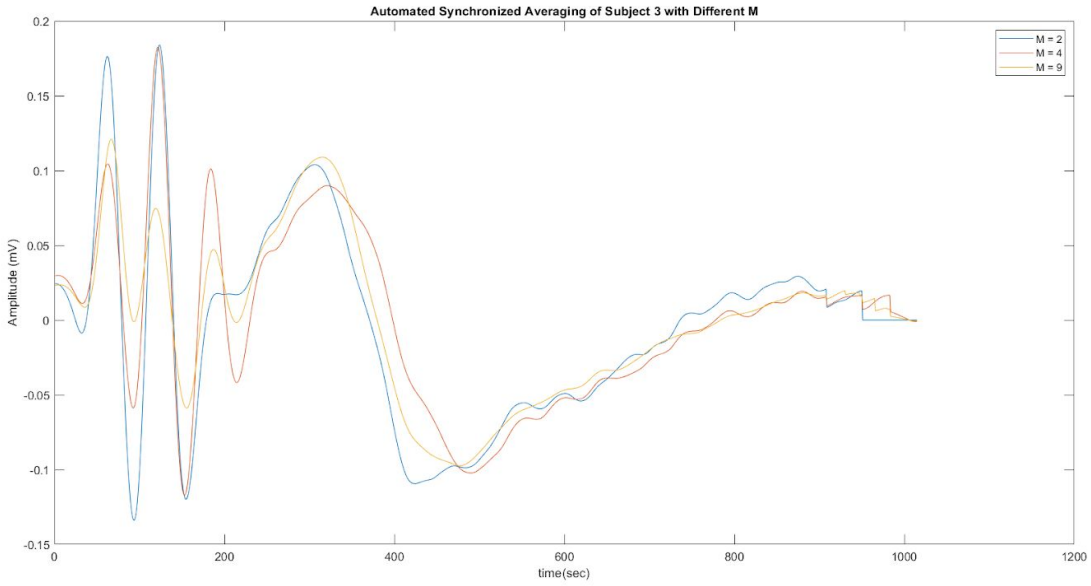

Figure 12. Plot of synchronized averaging for Subject 3 data with different M's. Automatically found trigger points.

Just like with the manual in part b, we can see that the curve does smoothen out. Interestingly we can see three waves where there should be only one wave, and that with the more waves averaged the amplitudes change significantly. If you take our actual data and look, there are

some discrepancies at the beginning which could be due to many factors. But, the point of this part is to see the output smoothing out after more waves, which we can see.

c) It is quite clear which method is better for synchronized averaging. The manual trigger identification method is superior. Even though the automatic method might be convenient, we can see a major problem compared to the manual method. The automatic method will take the whole signal without disregarding anything we deem necessary or too different compared to the rest of the data. With the manual method we can take out anything we think is noisy or just isn't a good representation of the data set. We can keep consistent waveforms and analyze the data we want. However, there is a downside to both methods. Since ECG signals are repetitive signals, we see the same waveform over and over. But as shown in this lab, each waveform can be different lengths, making us pad the signals to be the same length. In both the manual and automatic systems we can see that the padding causes problems at the end of the signals, but the automatic method seems to deal with it better than the manual.Therefore both methods have their flaws and advantages, but the manual method seems better.

Experiment 2: MOTION ARTIFACTS

Exercise 2.1:

a) In the manual it says to show 4 beats, but we chose to show the whole waveform and only colour the whole artifact to show the comparison with a resting signal.

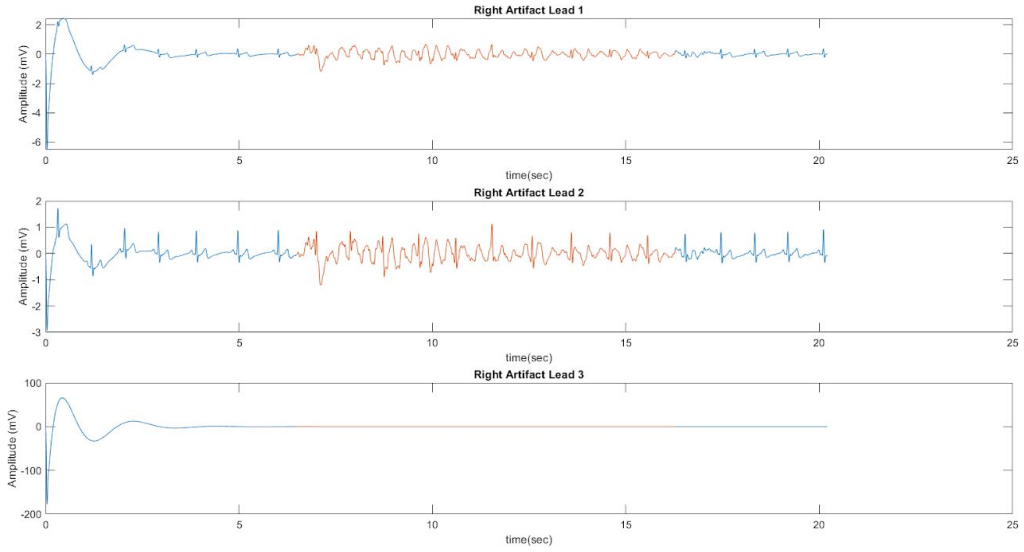

Figure 13. Plot of the highlighted areas (red) where there are the motion artifacts for the right arm.

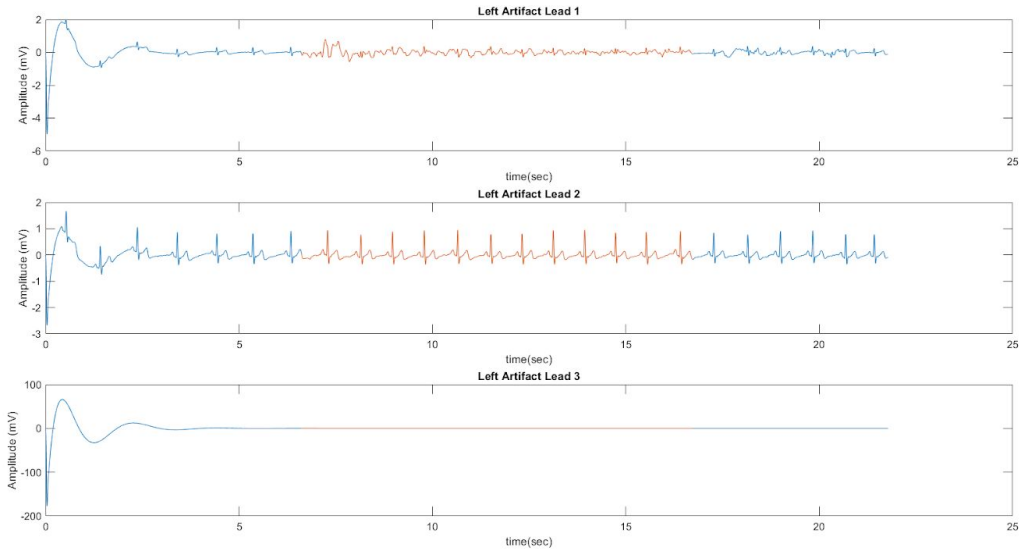

Figure 14. Plot of the highlighted areas(red) where there are the motion artifacts for the left arm.

b) The distortions in the signals are the electrical signals created by the individual when they waved their hand. The right and left signals have different numbers of leads that are active because of how the leads are connected to the BioRadio channels. The right arm was connected to both lead one and two, and the left lead was connected to the first and third channels. Also in our data acquisition we had a very large spike in channel 3 at the beginning of each dataset so the graph looks like it stabilizes after a while but when we take a closer look it does fluctuate.

c) To note, since we had to attach the leads to two different channels, this created a different outcome from what it should have been. Just by assessment of the above figures, you can see that the right artifact was picked up on two channels but the left artifact was only picked up on one channel. Therefore, when calculating the non-distorted waveform for the right hand it will be difficult. Since the channels only showed distorted signals, the calculated ideal non-distorted waveform will not be as perfect. This is clearly represented by the two figures below.

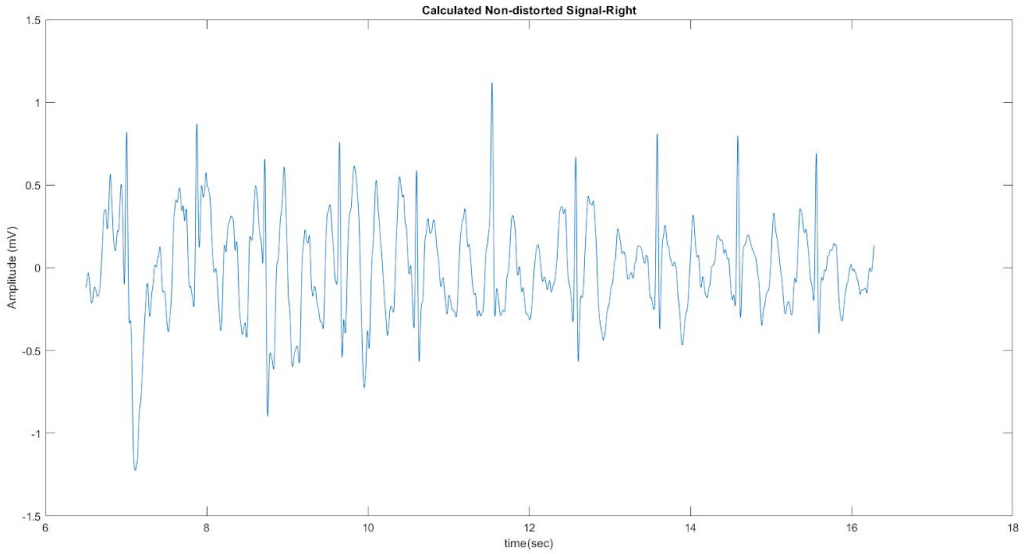

Figure 15. Plot of the calculated non-distorted waveform from right artifact.

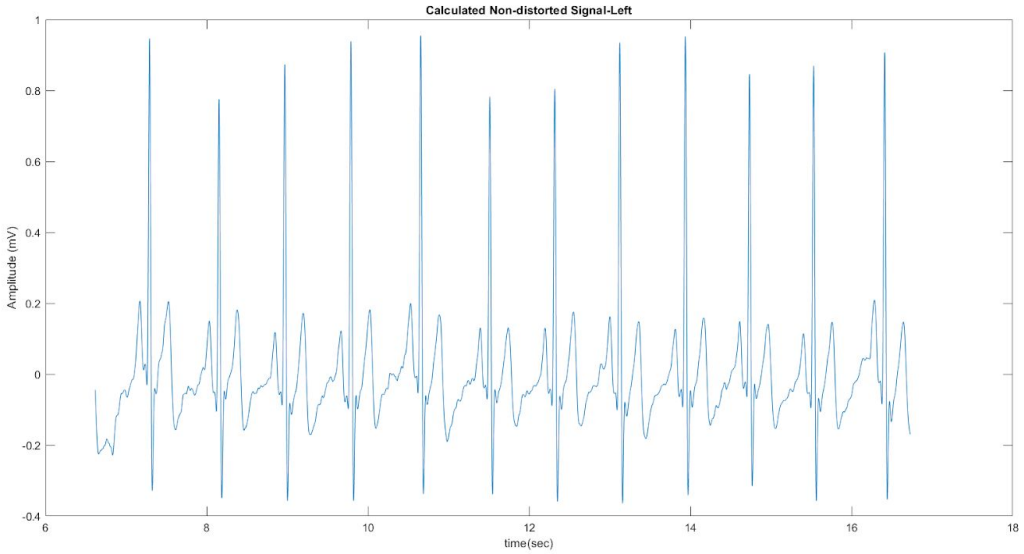

Figure 16. Plot of the calculated non-distorted waveform from left artifact.

PART C: ECG FREQUENCY DOMAIN ANALYSIS

Exercise 1.1:

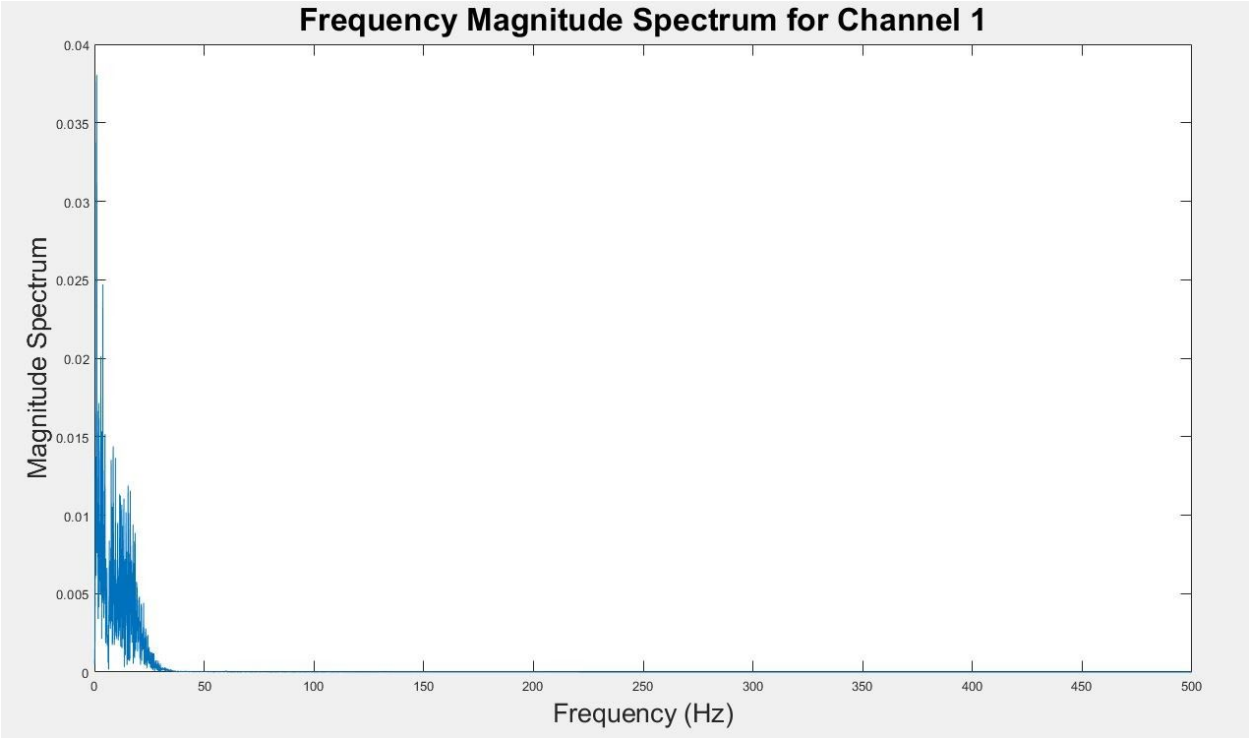

Figure 17. Frequency magnitude spectrum for channel 1

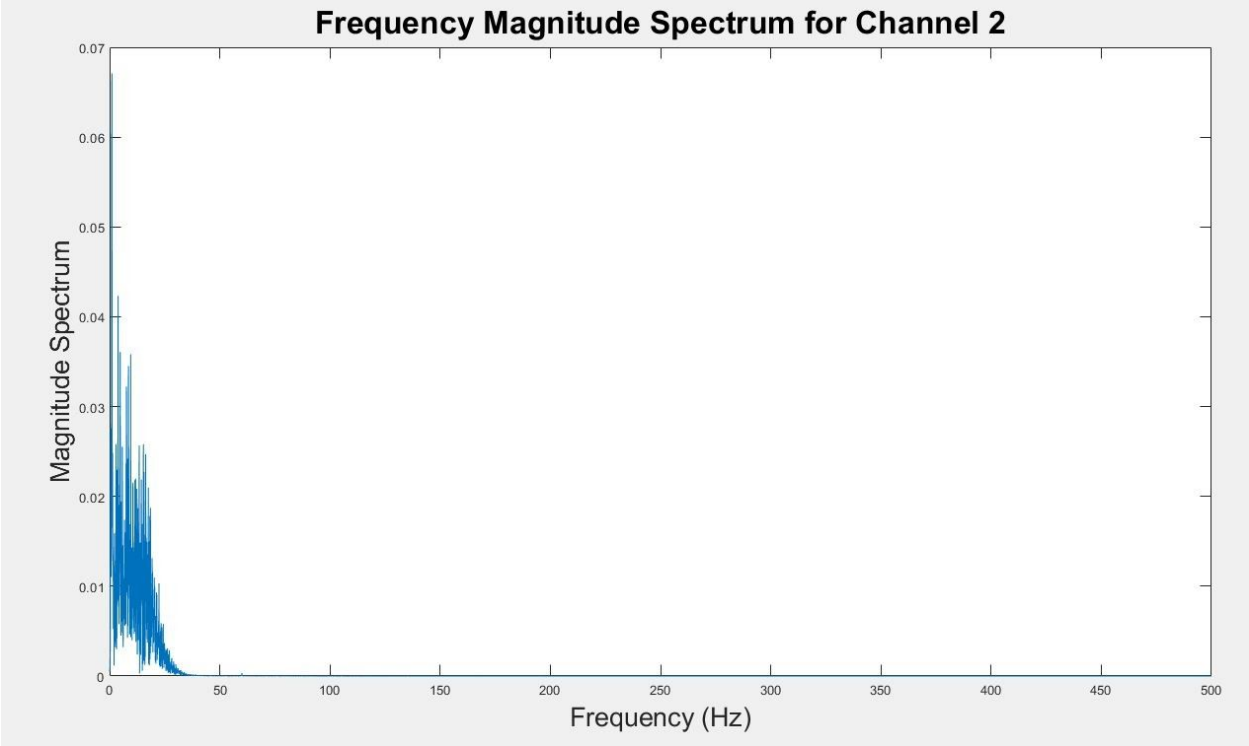

Figure 18. Frequency magnitude spectrum for channel 2

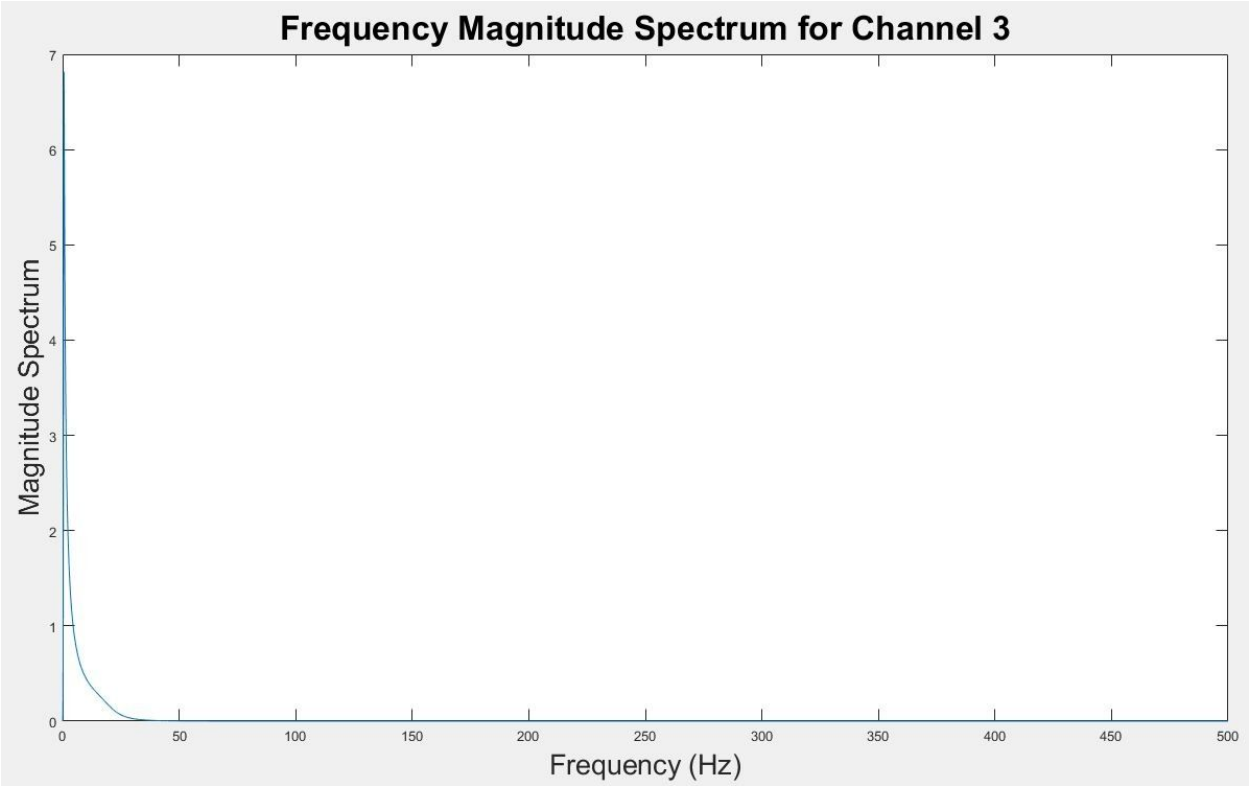

Figure 19. Frequency magnitude spectrum for channel 3

Description: Based on the visual analysis of the magnitude spectrum of figures 17 and 18, the noise seems to look like high frequency noise between 0-22 Hz seemed relevant data. The data after 22 Hz started to decrease in amplitude and this indicates the low frequency noise which was acquired during signal acquisition. The two types of noise in an ECG signal include the Electromyogram and low power interference for high frequency and baseline wandering for low frequency [4]. Thus, for 0 - 22 Hz, the noise can be classified as Electromyogram noise and baseline wander for frequency after 22 Hz. These noise artifacts that could have been produced by the magnetic field that was produced during the tangling or the position of the wires.

Exercise 1.2:

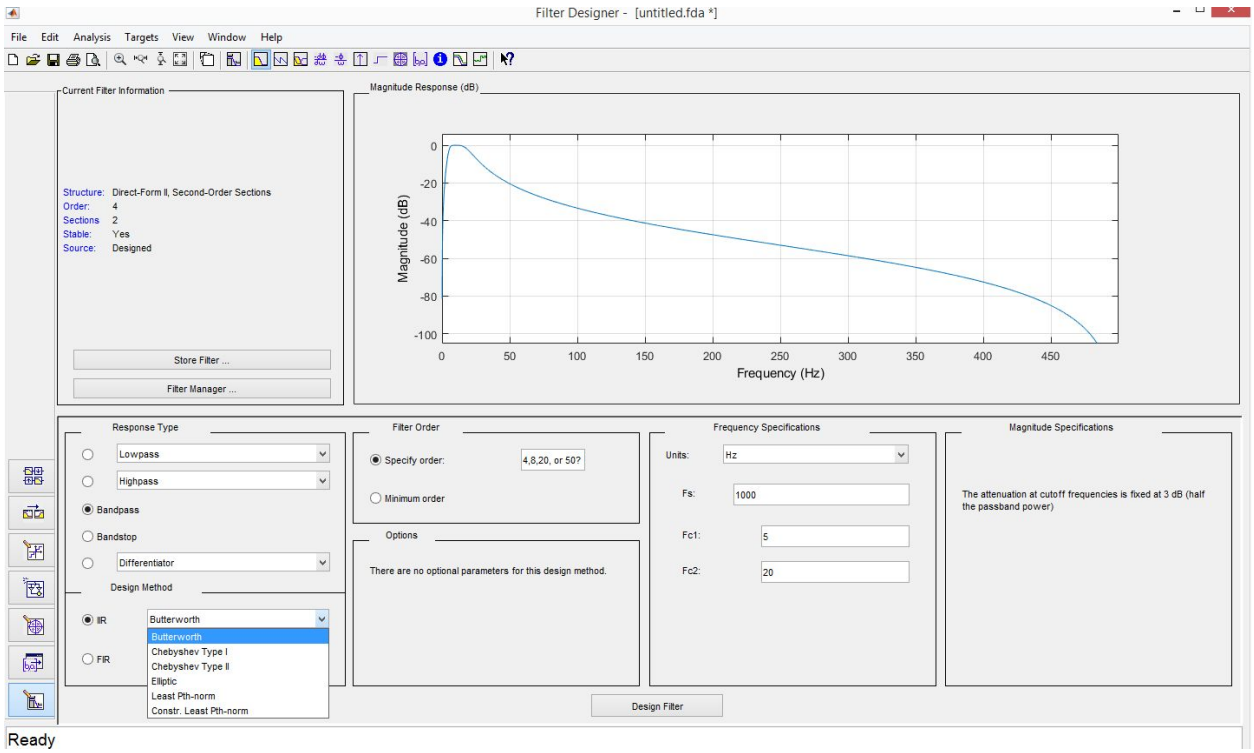

Figure 20: FDA tool used for designing filters of various types and orders. This lab focused on the IIR bandpass Butterworth and Chebyshev filters of orders 4, 8, 20, and 50. The above plot is of butterworth filter of order 4.

a)

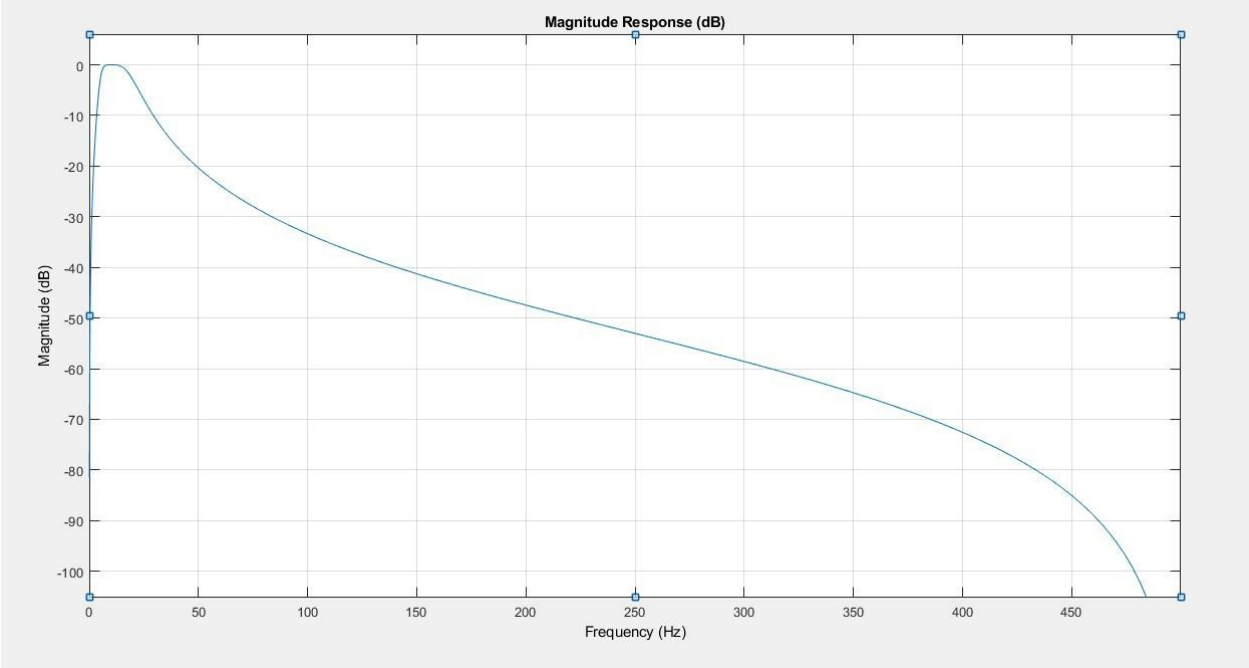

Figure 21. Frequency magnitudes of sit ECG using Butterworth filter of order 4

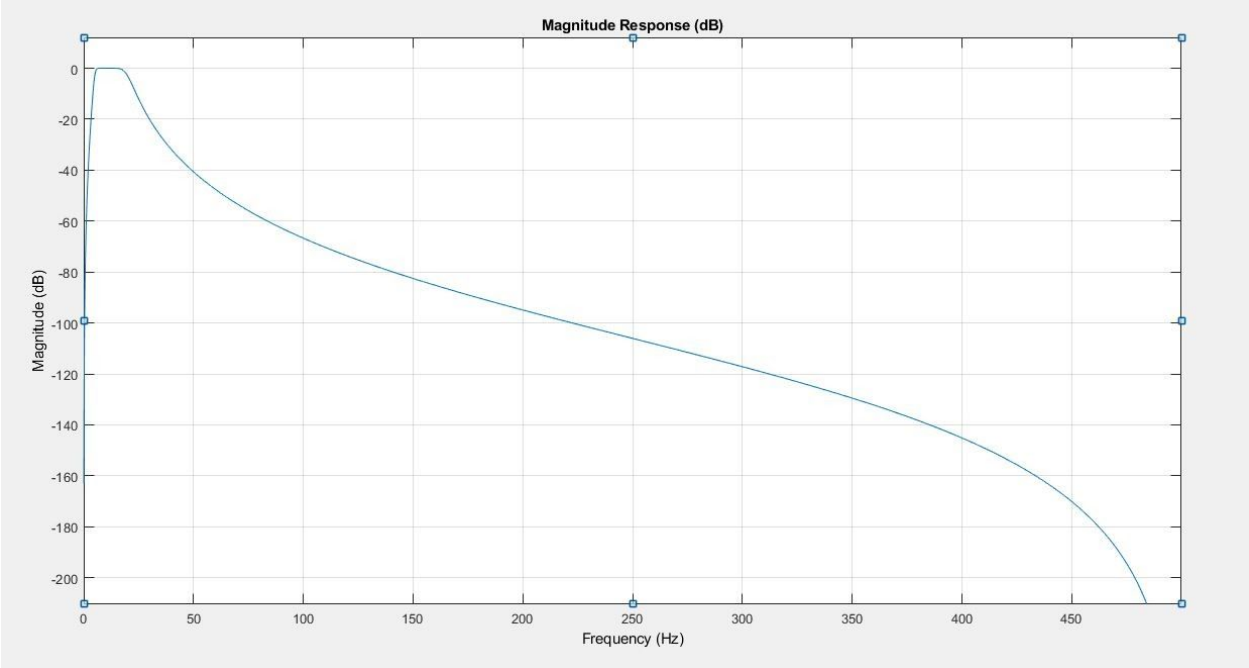

Figure 22. Frequency magnitudes of sit ECG using Butterworth filter of order 8

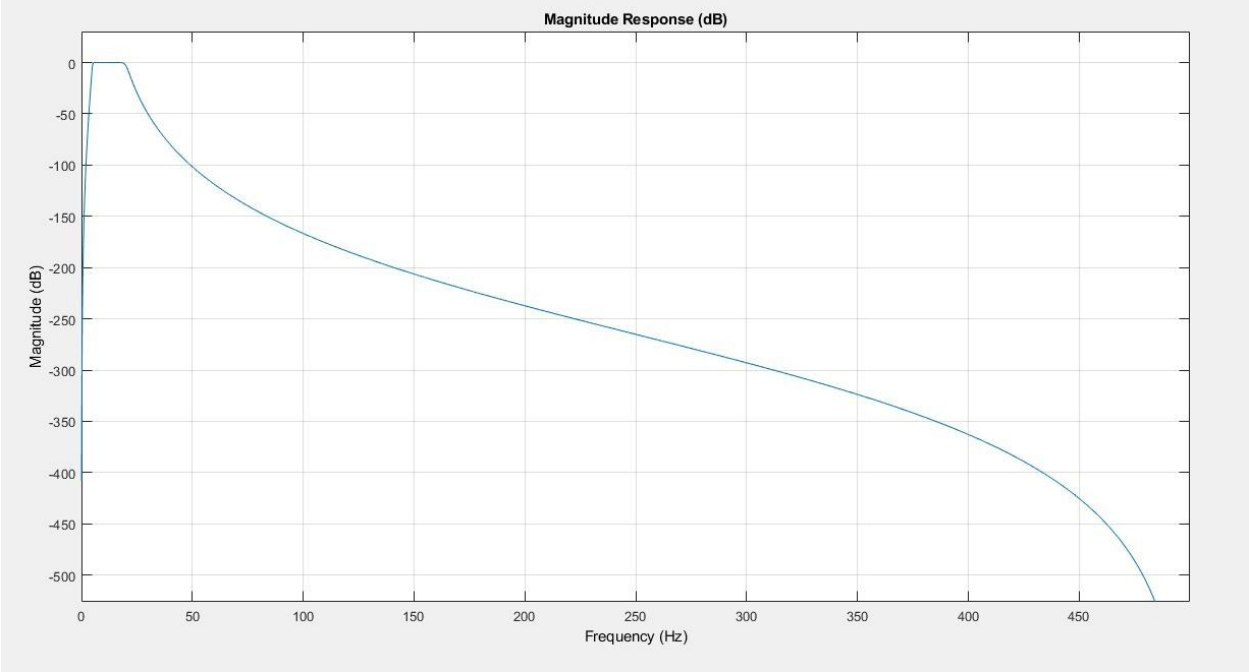

Figure 23. Frequency magnitudes of sit ECG using Butterworth filter of order 20

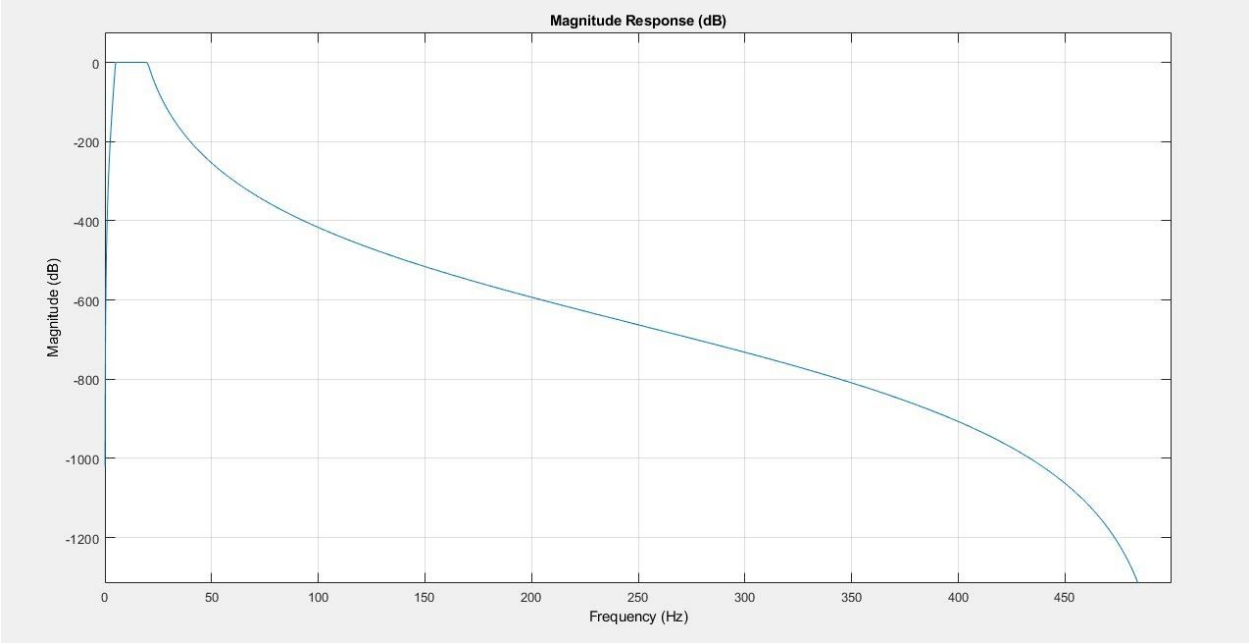

Figure 24. Frequency magnitudes of sit ECG using Butterworth filter of order 50

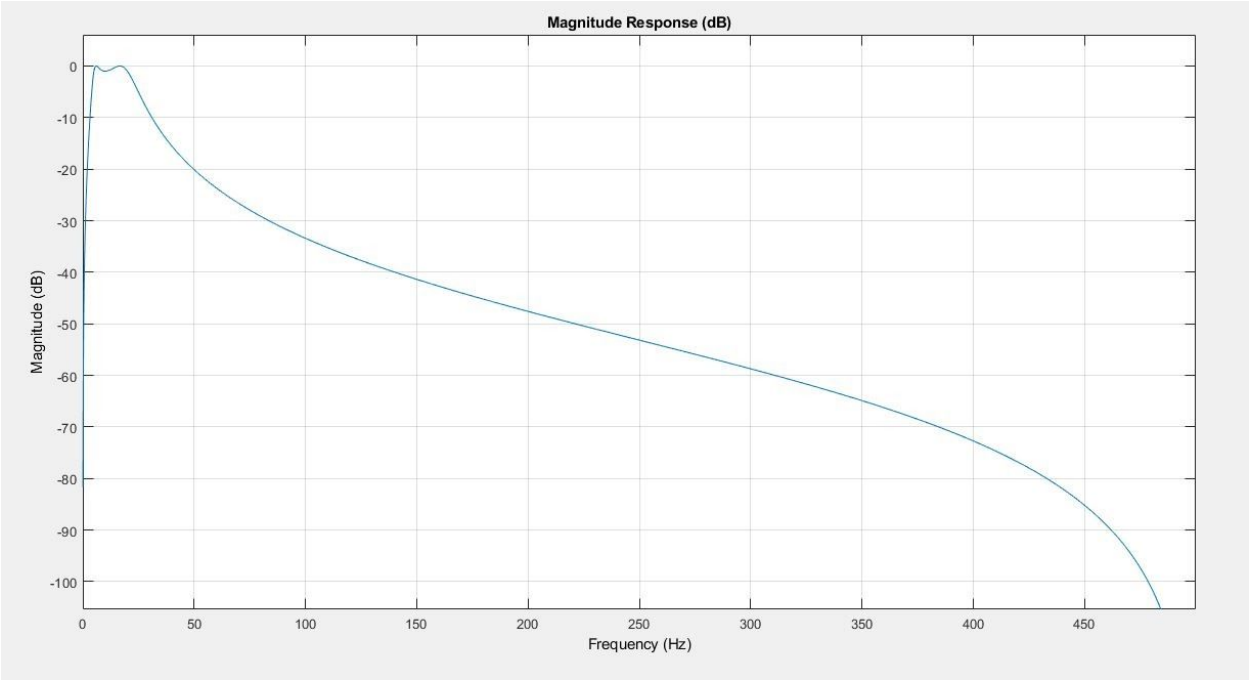

Figure 25. Frequency magnitudes of sit ECG using Chebyshev filter of order 4

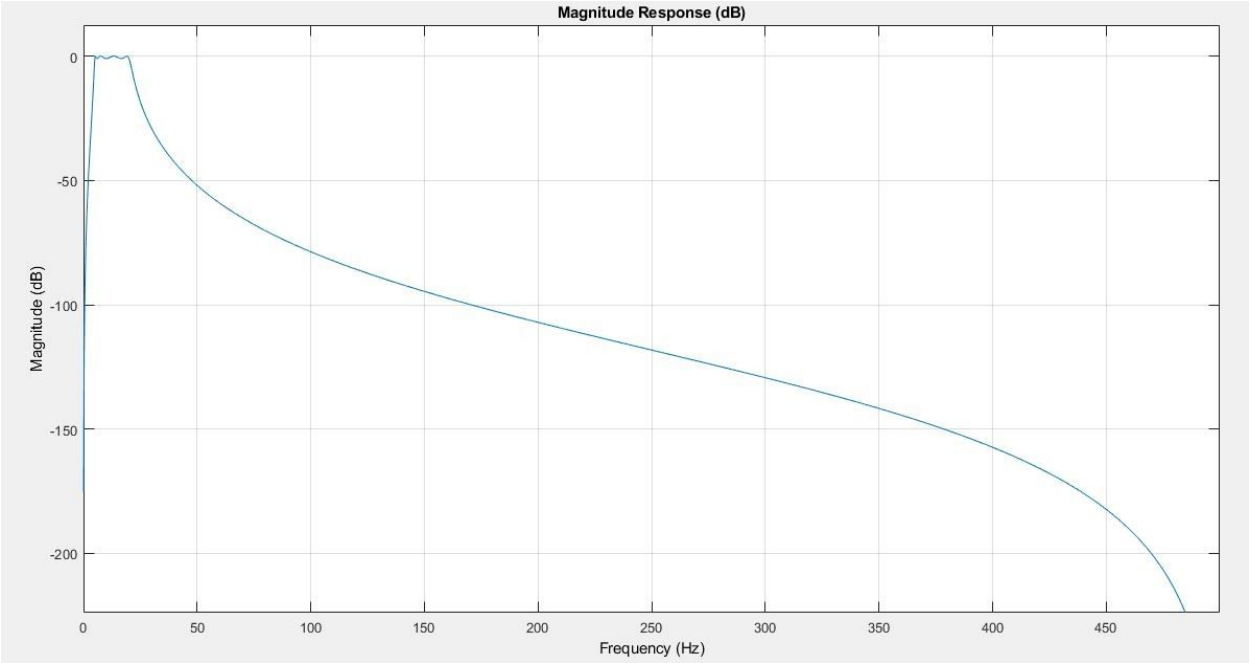

Figure 26. Frequency magnitudes of sit ECG using Chebyshev filter of order 8

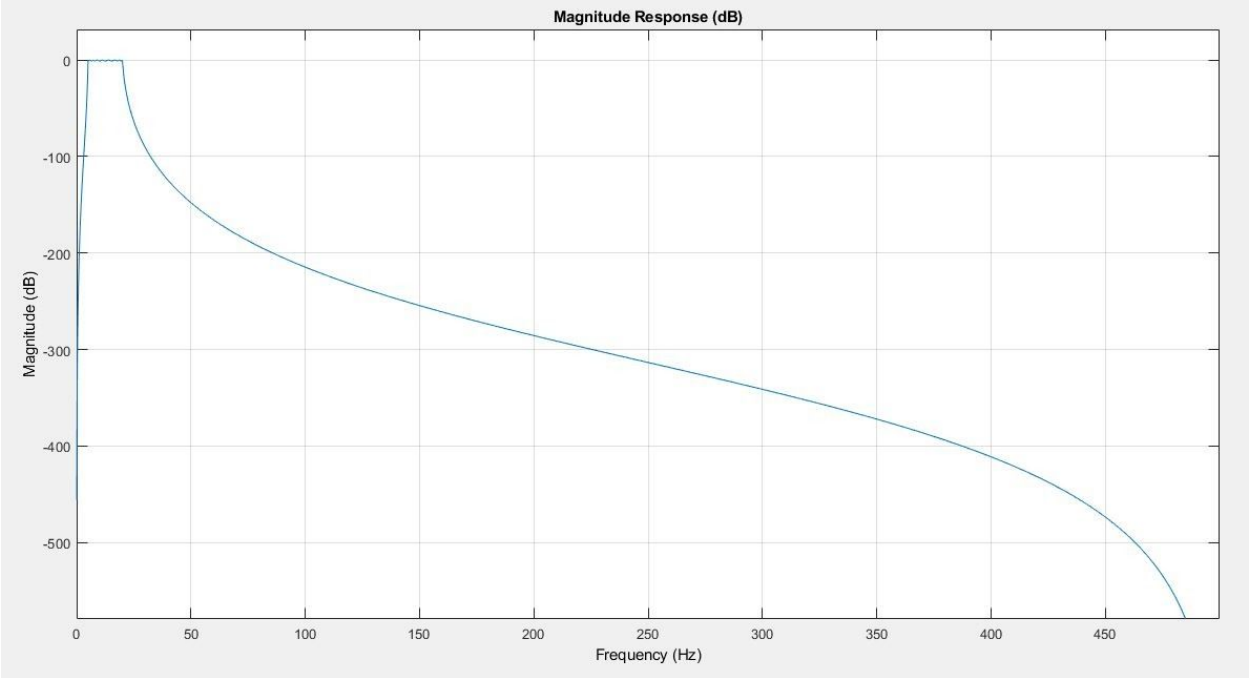

Figure 27. Frequency magnitudes of sit ECG using Chebyshev filter of order 20

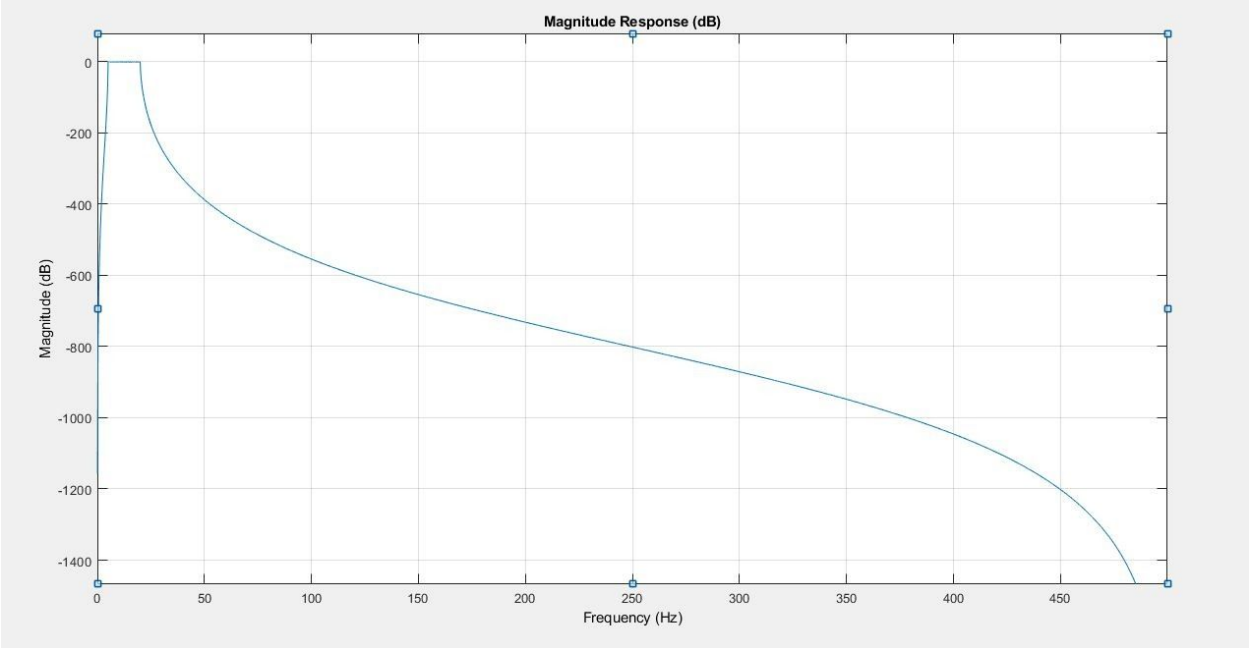

Figure 28. Frequency magnitudes of sit ECG using Chebyshev filter of order 50

Description: After analysing the frequency responses of the butterworth and chebyshev filters of different orders, both bandpass filters are comparable. The chebyshev frequency has steeper slopes for filters 4,8, and 20. The graphs are quite similar for both bandpass filters of order 50. For chebyshev filters, the steep slope and flat graph helps in flattening the signal and this inturn

reduces the low amplitude noises. Whereas, for butterworth filters the slope is not that steep and keeps changing mimcing an oscillation. When this filter is applied to a signal, the noise is not removed that much because of its nature of oscillating motion. Thus, it is not ideal for filtering. Therefore, the Chebyshev filter is better than the butterworth filter based on its frequency spectrum graph.

b)

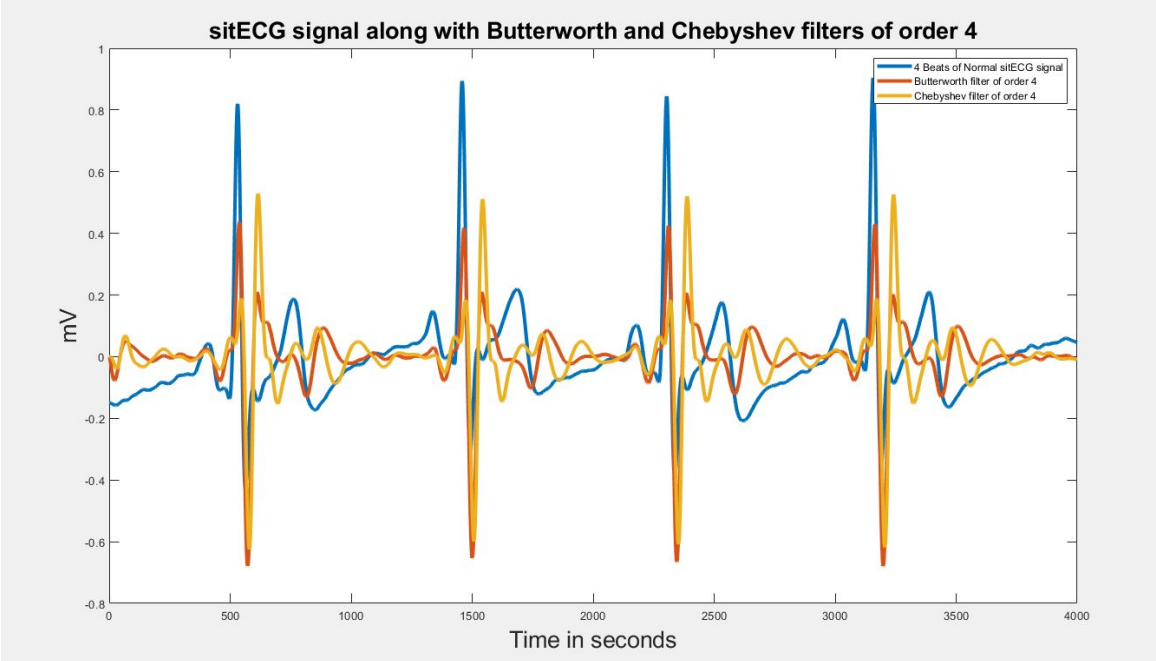

Figure 29: sitECG signal filtered with Butterworth and Chebyshev filters of order 4

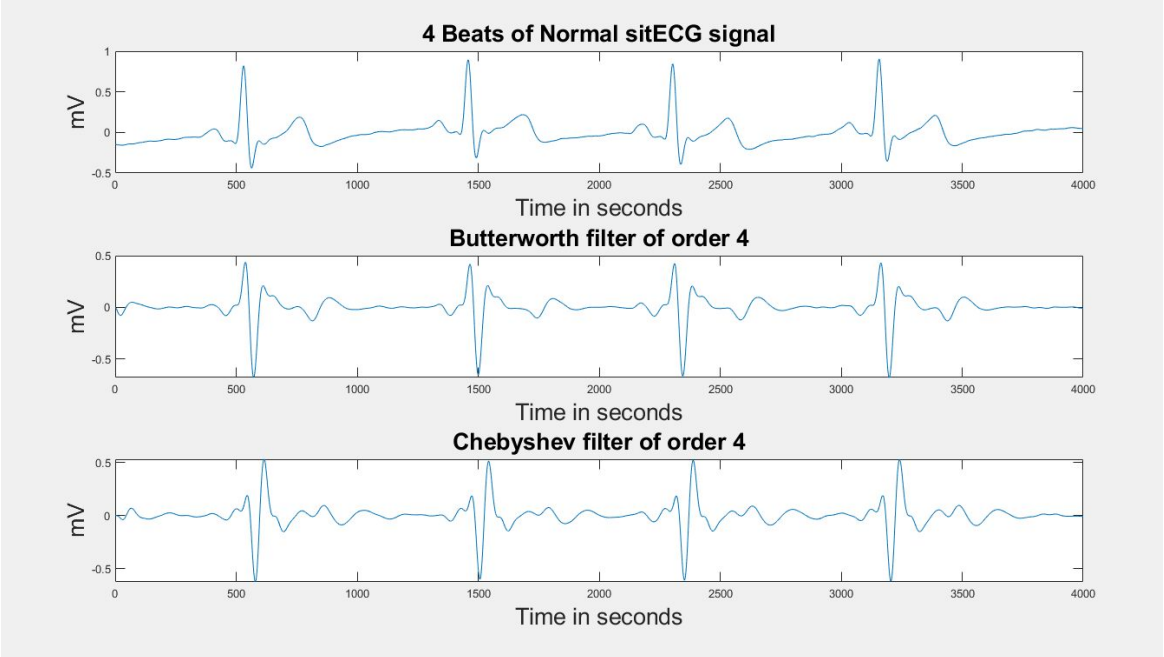

Figure 30: Another representation or subplots of sitECG signal filtered with Butterworth and Chebyshev filters of order 4.

The Chebyshev signal seems to be doing a better job of maintaining the integrity of the signal. It can be seen in figure 29 that when there is a bigger difference in between T and P waves when the original ecgSIT signal is filtered with Chebyshev filter of order 4 (yellow) vs Butterworth filter of order 4 (red). Since the Chebyshev filter maintains higher difference between those waves, it does a better job of maintaining the integrity of the signal.

c)

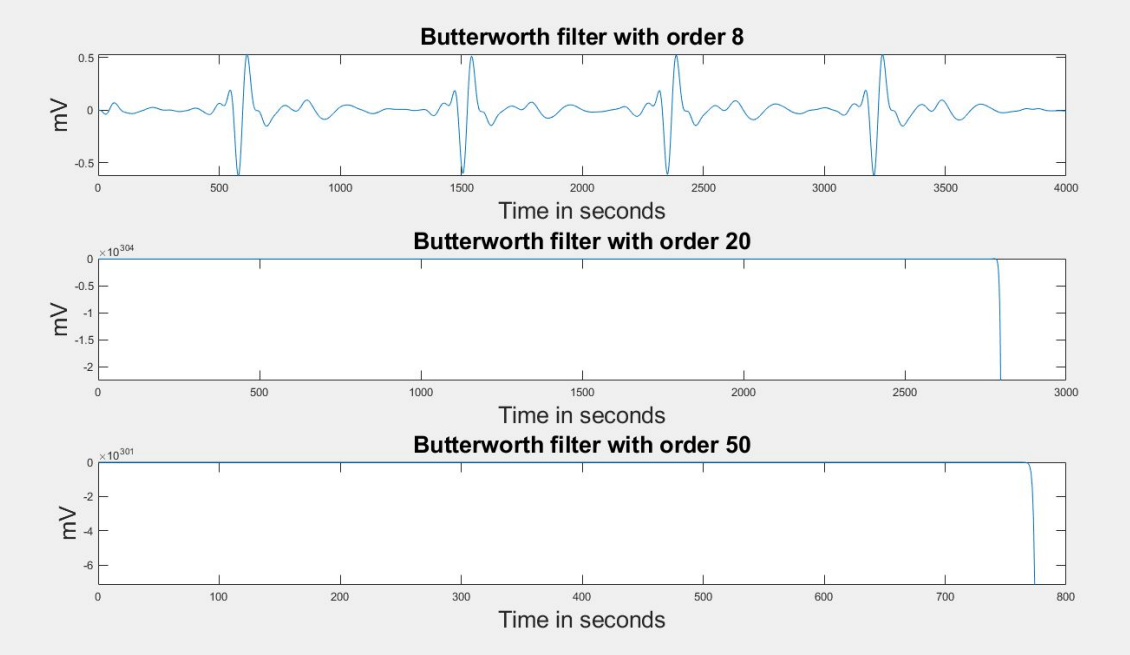

Figure 31: Frequency spectrum of butterworth signal of orders 8, 20, and 50 in frequency domain.

Description: It is evident from figure 31 that the butterworth filter of order 8 does a better job in filtering out the noise and maintaining signal integrity. Although the filtering technique is the same for butterworth filters, the change in order affects the behaviour of the filter partially also because of the change in coefficients when exported to the workspace. The filter with orders 20 and 50 are blank and therefore the signal will almost not be filtered. Therefore, butterworth filter with order 8 is doing an efficient if not the best job in filtering out the noise and maintaining integrity.

d)

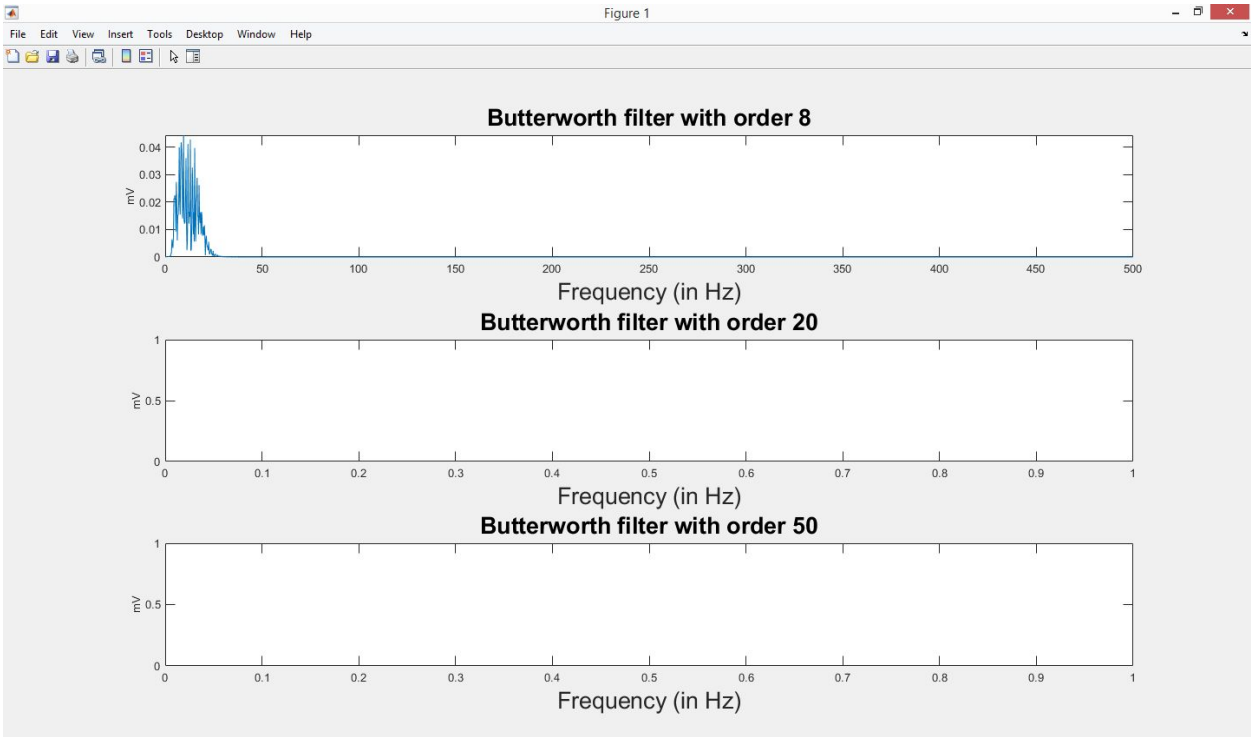

Figure 32: Frequency spectrum of butterworth signal of orders 8, 20, and 50 in frequency domain.

Description: Yes, the noise seems to be gone by the butterworth filter of order 8. Since the filters in time domain with orders 20 and 50 do not have proper filtering graphs as shown in Figure

31, the filters in frequency domain are empty as well, thus the butterworth filter with order offers the best noise reduction for both time and frequency domain systems.

Conclusion:

This lab had two major purposes, the first being to learn about ECG signals and the second was to learn about noise in biomedical signals. ECG signals, as previously stated, are probably the most important biomedical signal for a medical practitioner. It is a representation of the heart's movements due to electrical signals. However, like any other signal there will be the main signal, which is what we want, and this signal will have underlying noise. In this lab we explored different ways to deal with this noise. In the first part of the lab we explored cross correlation and synchronized averaging to smoothen the signal (aka remove noise). In the last part of the lab we explored different filters to reduce the noise. We specifically looked at theButterworth and Chebyshev Type 1 IIR bandpass filters. We experiment different order filters to see which one is the best at filtering out the noise of an ECG. From the above analysis and comparison of butterworth and chebyshev filters we came to a conclusion that chebyshev type 1 filter is better in removing noise from the signal. While both filters have different applications in noise filtering, this lab was more geared towards chebyshev filters. The nature of chebyshev filters with flat slope and steep slope makes it efficient in removing noise when it is applied to a signal. When compared to a butterworth signal its nature of changing slope makes it oscillative and thus when its is applied to the ecg signal its not as efficient as opposed to chebyshev filter, as evident in the above analysis and plots.

References:

[1] Khademi, April . "Lab Manual" BME 632 - Signals and System II, 17 March 2020, Ryerson University

[2] Khademi, April . "Lab Experiment" BME 632 - Signals and System II, 17 March 2020, Ryerson University

[3]R. Green and B.P. Lathi, *Linear Systems and Signals*: Oxford University Press,2018.

[4] Peter, Sonia. “Noise Analysis and Different Denoising Techniques of ECG Signal - A Survey.” *Https://Www.iosrjournals.org/Iosr-Jece/Papers/ICETEM/Vol. 1 Issue 1/ECE 06-40-44.Pdf*

Appendix:

PART B

```
%MAIN
%% Opening File
sit = readtable('sitECG.csv');
lay = readtable('layingECG.csv');
sub1 = readtable('subjectECG1.csv');
sub2 = readtable('subjectECG2.csv');
sub3 = readtable('subjectECG3.csv');

%% Creating vectors

%extracting time values from the experiment data
sitECG1 = sit(:,2);
sitECG2 = sit(:,3);
sitECG3 = sit(:,4);
tsit = sit(:,1);

layECG1 = lay(:,2);
layECG2 = lay(:,3);
layECG3 = lay(:,4);
tlay = lay(:,1);

sub1ECG1 = sub1(:,2);
sub1ECG2 = sub1(:,3);
sub1ECG3 = sub1(:,4);
tsub1 = sub1(:,1);

sub2ECG1 = sub2(:,2);
sub2ECG2 = sub2(:,3);
sub2ECG3 = sub2(:,4);
tsub2 = sub2(:,1);

sub3ECG1 = sub3(:,2);
sub3ECG2 = sub3(:,3);
sub3ECG3 = sub3(:,4);
tsub3 = sub3(:,1);

%making the matrix useful for calculations
sitECG1 = sitECG1{:,:};
```

```
sitECG2 = sitECG2{:,:};
sitECG3 = sitECG3{:,:};
tsit = tsit{:,:};

layECG1 = layECG1{:,:};
layECG2 = layECG2{:,:};
layECG3 = layECG3{:,:};
tlay = tlay{:,:};

sub1ECG1 = sub1ECG1{:,:};
sub1ECG2 = sub1ECG2{:,:};
sub1ECG3 = sub1ECG3{:,:};
tsub1 = tsub1{:,:};

sub2ECG1 = sub2ECG1{:,:};
sub2ECG2 = sub2ECG2{:,:};
sub2ECG3 = sub2ECG3{:,:};
tsub2 = tsub2{:,:};

sub3ECG1 = sub3ECG1{:,:};
sub3ECG2 = sub3ECG2{:,:};
sub3ECG3 = sub3ECG3{:,:};
tsub3 = tsub3{:,:};

%% Changing time to samples

%conversion of time array to take into the account the sampling rate (1000Hz)

tsit = 0 : length(tsit) - 1;
tsit = tsit * 0.001;

tlay = 0 : length(tlay) - 1;
tlay = tlay * 0.001;

tsub1 = 0 : length(tsub1) - 1;
tsub1 = tsub1 * 0.001;

tsub2 = 0 : length(tsub2) - 1;
tsub2 = tsub2 * 0.001;

tsub3 = 0 : length(tsub3) - 1;
tsub3 = tsub3 * 0.001;
```

```
%% Experiemnt 1.1

% a)
tsec1 = 1:8000;
tsec2 = 11700:13226;
sitECG1sec = section(sitECG1,tsec1,tsec2);
[locs_Pwave, locs_Qwave, locs_Rwave, locs_Swave, locs_Twave] = pqrst(sitECG1sec);

figure(1)
plot(tsit,sitECG1sec);
hold on
plot((locs_Pwave/1000),sitECG1(locs_Pwave),'rv','MarkerFaceColor','r');
plot((locs_Qwave/1000),sitECG1(locs_Qwave),'rv','MarkerFaceColor','b');
plot((locs_Rwave/1000),sitECG1(locs_Rwave),'rv','MarkerFaceColor','g');
plot((locs_Swave/1000),sitECG1(locs_Swave),'rv','MarkerFaceColor','m');
plot((locs_Twave/1000),sitECG1(locs_Twave),'rv','MarkerFaceColor','c');
legend('ECG Signal', 'P-wave', 'Q-wave', 'R-wave', 'S-wave', 'T-wave')
title('ECG Signal From Channel 1 When Sitting')
xlabel('Time(sec)')
ylabel('Voltage(mV)')

%b)

% we will choose to use the R-wave because it is the most accurate part of the ECG wave to
recognize and measure

sizeofbeats = length(locs_Rwave);
range1 = locs_Rwave(1);
range2 = locs_Rwave(sizeofbeats);
rangebpm = range2 - range1;
rangebpm = rangebpm/1000;

bps = sizeofbeats/rangebpm;

bpm = bps * 60;

%% Experiment 1.2

% a)
tsec1 = 1:8000;
tsec2 = 11700:13226;
sitECG2sec = section(sitECG2,tsec1,tsec2);
sitECG3sec = section(sitECG3,tsec1,tsec2);
```

```
sitECG2calc = ECG2calculated(sitECG1sec, sitECG3sec);

figure(2)
plot(tsit,sitECG2sec)
hold on
plot(tsit,sitECG2calc)
legend('Measured', 'Calculated')
title('ECG Signal From Channel 2 When Sitting')
xlabel('Time(sec)')
ylabel('Voltage(mV)')

% b)

sitECG2diff  = absoluteDiff(sitECG2calc,sitECG2sec)

figure(3)
plot(tsit,sitECG2diff)
title('ECG Signal From Channel 2 - Absolute Difference')
xlabel('Time(sec)')
ylabel('Voltage(mV)')

% c)

[MSE] = meansqrError(sitECG2calc, sitECG2sec)

sitECG2mse = MSE;

%% Experiment 1.3 DON"T DO THIS PART
%% Experiment 1.4

% a)

%creating the sectioned off area of sitECG1 to use as template

tpulse = 10270:10720;
sectionedECG1 = sitECG1;
sectionedECG1 = sectionedECG1(tpulse);
tfull = 1:13226;

cc = xcorr(sectionedECG1,sitECG1);
cc = cc(tfull);

[~,l] = findpeaks(cc, 'MinPeakHeight', 4);
```

```
figure(4)
plot(tpulse,sectionedECG1)
title('Template')
xlabel('time(sec)')
ylabel('Amplitude (mV)')

figure(5)
plot(tsit,cc)
title('Cross-Correlation Between One Pulse and Lead 1')
xlabel('time(sec)')
ylabel('Amplitude (mV)')

% b) & c)

hr = heartRate(cc);

%% Experiment 1.5

% a)

figure(6)
subplot(311)
plot(tsub1((2000:13118)),sub1ECG1(2000:13118))
title('Subject 1')
xlabel('time(sec)')
ylabel('Amplitude (mV)')
subplot(312)
plot(tsub2((2800:11817)),sub2ECG1(2800:11817))
title('Subject 2')
xlabel('time(sec)')
ylabel('Amplitude (mV)')
subplot(313)
plot(tsub3((2000:15616)),sub3ECG1(2000:15616))
title('Subject 3')
xlabel('time(sec)')
ylabel('Amplitude (mV)')

%% Subject 1
% Getting each waveform from lead 1
sub1M1 = sub1ECG1(2920:3840);
sub1M2 = sub1ECG1(3840:4806);
sub1M3 = sub1ECG1(4806:5785);
```

```
sub1M4 = sub1ECG1(5785:6748);
sub1M5 = sub1ECG1(6748:7679);
sub1M6 = sub1ECG1(7679:8626);
sub1M7 = sub1ECG1(8626:9635); %longest wave
sub1M8 = sub1ECG1(9635:10620);
sub1M9 = sub1ECG1(10620:11530);
%padding them to the same length as the longest wave
sub1M1 = pad(sub1M1,sub1M7);
sub1M2 = pad(sub1M2,sub1M7);
sub1M3 = pad(sub1M3,sub1M7);
sub1M4 = pad(sub1M4,sub1M7);
sub1M5 = pad(sub1M5,sub1M7);
sub1M6 = pad(sub1M6,sub1M7);
sub1M8 = pad(sub1M8,sub1M7);
sub1M9 = pad(sub1M9,sub1M7);
%% Subject 2
% Getting each waveform from lead 1
sub2M1 = sub2ECG1(3112:3923);
sub2M2 = sub2ECG1(3923:4807);
sub2M3 = sub2ECG1(4807:5810);
sub2M4 = sub2ECG1(5810:6753);
sub2M5 = sub2ECG1(6753:7650);
sub2M6 = sub2ECG1(7650:8680);
sub2M7 = sub2ECG1(8680:9756); %longest wave
sub2M8 = sub2ECG1(9756:10800);
sub2M9 = sub2ECG1(10800:11780);
%padding them to the same length as the longest wave
sub2M1 = pad(sub2M1,sub2M7);
sub2M2 = pad(sub2M2,sub2M7);
sub2M3 = pad(sub2M3,sub2M7);
sub2M4 = pad(sub2M4,sub2M7);
sub2M5 = pad(sub2M5,sub2M7);
sub2M6 = pad(sub2M6,sub2M7);
sub2M8 = pad(sub2M8,sub2M7);
sub2M9 = pad(sub2M9,sub2M7);
%% Subject 3
% Getting each waveform from lead 1
sub3M1 = sub3ECG1(3790:4656);
sub3M2 = sub3ECG1(4656:5560);
sub3M3 = sub3ECG1(5560:6591);
sub3M4 = sub3ECG1(6591:7623);
sub3M5 = sub3ECG1(7623:8666); %longest wave
sub3M6 = sub3ECG1(8666:9651);
```

```
sub3M7 = sub3ECG1(9651:10560);
sub3M8 = sub3ECG1(10560:11510);
sub3M9 = sub3ECG1(11510:12530);
%padding them to the same length as the longest wave
sub3M1 = pad(sub3M1,sub3M5);
sub3M2 = pad(sub3M2,sub3M5);
sub3M3 = pad(sub3M3,sub3M5);
sub3M4 = pad(sub3M4,sub3M5);
sub3M6 = pad(sub3M6,sub3M5);
sub3M7 = pad(sub3M7,sub3M5);
sub3M8 = pad(sub3M8,sub3M5);
sub3M9 = pad(sub3M9,sub3M5);

%% M's
% M = 2

syncAvgsub1M2 = (sub1M1 + sub1M2)/2;
syncAvgsub2M2 = (sub2M1 + sub2M2)/2;
syncAvgsub3M2 = (sub3M1 + sub3M2)/2;

% M = 4

syncAvgsub1M4 = (sub1M1 + sub1M2 + sub1M3 + sub1M4)/4;
syncAvgsub2M4 = (sub2M1 + sub2M2 + sub2M3 + sub2M4)/4;
syncAvgsub3M4 = (sub3M1 + sub3M2 + sub3M3 + sub3M4)/4;

% M = 9

syncAvgsub1M9 = (sub1M1 + sub1M2 + sub1M3 + sub1M4 + sub1M5 + sub1M6 + sub1M7 +
sub1M8 + sub1M9)/9;
syncAvgsub2M9 = (sub2M1 + sub2M2 + sub2M3 + sub2M4 + sub2M5 + sub2M6 + sub2M7 +
sub2M8 + sub2M9)/9;
syncAvgsub3M9 = (sub3M1 + sub3M2 + sub3M3 + sub3M4 + sub3M5 + sub3M6 + sub3M7 +
sub3M8 + sub3M9)/9;

% Plots of M's

figure(7)
plot(1:length(sub1M7),syncAvgsub1M2)
hold on
plot(1:length(sub1M7),syncAvgsub1M4)
plot(1:length(sub1M7),syncAvgsub1M9)
title('Manual Synchronized Averaging of Subject 1 with Different M')
```

```
legend('M = 2', 'M = 4', 'M = 9')
xlabel('time(sec)')
ylabel('Amplitude (mV)')

figure(8)
plot(1:length(sub2M7),syncAvgsub2M2)
hold on
plot(1:length(sub2M7),syncAvgsub2M4)
plot(1:length(sub2M7),syncAvgsub2M9)
title('Manual Synchronized Averaging of Subject 2 with Different M')
legend('M = 2', 'M = 4', 'M = 9')
xlabel('time(sec)')
ylabel('Amplitude (mV)')

figure(9)
plot(1:length(sub3M5),syncAvgsub3M2)
hold on
plot(1:length(sub3M5),syncAvgsub3M4)
plot(1:length(sub3M5),syncAvgsub3M9)
title('Manual Synchronized Averaging of Subject 3 with Different M')
legend('M = 2', 'M = 4', 'M = 9')
xlabel('time(sec)')
ylabel('Amplitude (mV)')
%% Automated

tpulse1 = 6505:7505;
tpulse2 = 5612:6574;
tpulse3 = 8492:9495;

tfull1 = 1:length(sub1ECG1);
tfull2 = 1:length(sub2ECG1);
tfull3 = 1:length(sub3ECG1);

syncAvg1M2 = synchronizedAveraging(tpulse1,sub1ECG1,tfull1,2);
syncAvg1M4 = synchronizedAveraging(tpulse1,sub1ECG1,tfull1,4);
syncAvg1M9 = synchronizedAveraging(tpulse1,sub1ECG1,tfull1,9);

syncAvg2M2 = synchronizedAveraging(tpulse1,sub1ECG1,tfull1,2);
syncAvg2M4 = synchronizedAveraging(tpulse1,sub1ECG1,tfull1,4);
syncAvg2M9 = synchronizedAveraging(tpulse1,sub1ECG1,tfull1,9);

syncAvg3M2 = synchronizedAveraging(tpulse1,sub1ECG1,tfull1,2);
syncAvg3M4 = synchronizedAveraging(tpulse1,sub1ECG1,tfull1,4);
```

```
syncAvg3M9 = synchronizedAveraging(tpulse1,sub1ECG1,tfull1,9);

figure(10)
plot(1:length(syncAvg1M2),syncAvg1M2)
hold on
plot(1:length(syncAvg1M2),syncAvg1M4)
plot(1:length(syncAvg1M2),syncAvg1M9)
title('Automated Synchronized Averaging of Subject 1 with Different M')
legend('M = 2', 'M = 4', 'M = 9')
xlabel('time(sec)')
ylabel('Amplitude (mV)')

figure(11)
plot(1:length(syncAvg2M2),syncAvg2M2)
hold on
plot(1:length(syncAvg2M2),syncAvg2M4)
plot(1:length(syncAvg2M2),syncAvg2M9)
title('Automated Synchronized Averaging of Subject 2 with Different M')
legend('M = 2', 'M = 4', 'M = 9')
xlabel('time(sec)')
ylabel('Amplitude (mV)')

figure(12)
plot(1:length(syncAvg3M2),syncAvg3M2)
hold on
plot(1:length(syncAvg3M2),syncAvg3M4)
plot(1:length(syncAvg3M2),syncAvg3M9)
title('Automated Synchronized Averaging of Subject 3 with Different M')
legend('M = 2', 'M = 4', 'M = 9')
xlabel('time(sec)')
ylabel('Amplitude (mV)')

%% Experiment 2.1

% a)

artifactRight = readtable('artifactrightECG.csv');
artifactLeft = readtable('artifactleftECG.csv');

artR1 = artifactRight(:,2);
artR1 = artR1{:,:};
artR2 = artifactRight(:,3);
artR2 = artR2{:,:};
```

```
artR3 = artifactRight(:,4);
artR3 = artR3{:,:};
tartR = artifactRight(:,1);
tartR = tartR{:,:};

artL1 = artifactLeft(:,2);
artL1 = artL1{:,:};
artL2 = artifactLeft(:,3);
artL2 = artL2{:,:};
artL3 = artifactLeft(:,4);
artL3 = artL3{:,:};
tartL = artifactLeft(:,1);
tartL = tartL{:,:};

tartR = 0 : length(tartR) - 1;
tartR = tartR * 0.001;
tartL = 0 : length(tartL) - 1;
tartL = tartL * 0.001;

tsec3 = 1:6497;
tsec4 = 16280:20213;

artR1sec = section(artR1,tsec3,tsec4);
artR2sec = section(artR2,tsec3,tsec4);
artR3sec = section(artR3,tsec3,tsec4);

figure(16)
subplot(311)
plot(tartR,artR1)
hold on
plot(tartR,artR1sec)
title('Right Artifact Lead 1')
xlabel('time(sec)')
ylabel('Amplitude (mV)')
subplot(312)
plot(tartR,artR2)
hold on
plot(tartR,artR2sec)
title('Right Artifact Lead 2')
xlabel('time(sec)')
ylabel('Amplitude (mV)')
subplot(313)
plot(tartR,artR3)
```

```
hold on
plot(tartR,artR3sec)
title('Right Artifact Lead 3')
xlabel('time(sec)')
ylabel('Amplitude (mV)')

tsec5 = 1:6618;
tsec6 = 16730:21779;

artL1sec = section(artL1,tsec5,tsec6);
artL2sec = section(artL2,tsec5,tsec6);
artL3sec = section(artL3,tsec5,tsec6);

figure(17)
subplot(311)
plot(tartL,artL1)
hold on
plot(tartL,artL1sec)
title('Left Artifact Lead 1')
xlabel('time(sec)')
ylabel('Amplitude (mV)')
subplot(312)
plot(tartL,artL2)
hold on
plot(tartL,artL2sec)
title('Left Artifact Lead 2')
xlabel('time(sec)')
ylabel('Amplitude (mV)')
subplot(313)
plot(tartL,artL3)
hold on
plot(tartL,artL3sec)
title('Left Artifact Lead 3')
xlabel('time(sec)')
ylabel('Amplitude (mV)')

% c)

artR1calc = artR2sec - artR3sec;
artL1calc = artL2sec - artL3sec;

figure(18)
plot(tartR,artR1calc)
```

```
title('Calculated Non-distorted Signal-Right')
xlabel('time(sec)')
ylabel('Amplitude (mV)')

figure(19)
 plot(tartL,artL1calc)
title('Calculated Non-distorted Signal-Left')
xlabel('time(sec)')
ylabel('Amplitude (mV)')

%Functions

function [syncAvg] = synchronizedAveraging (tpulse, signal,tfull,M)
% to keep consistent the Ms will be 2,4,9
% cross-correltation to find peaks
sectionedSignal = signal(tpulse);
cc = xcorr(sectionedSignal,signal);
cc = cc(tfull);
[~,ccpeaks] = findpeaks(cc, 'MinPeakHeight', 4);

r1 = ccpeaks(1);
r2 = ccpeaks(2);
r3 = ccpeaks(3);
r4 = ccpeaks(4);
r5 = ccpeaks(5);
r6 = ccpeaks(6);
r7 = ccpeaks(7);
r8 = ccpeaks(8);
r9 = ccpeaks(9);
r10 = ccpeaks(10);


wave1 = r1:r2;
wave2 = r2:r3;
wave3 = r3:r4;
wave4 = r4:r5;
wave5 = r5:r6;
wave6 = r6:r7;
wave7 = r7:r8;
wave8 = r8:r9;
wave9 = r9:r10;

% finding the largest wave andindexing it
```

```
waveArray =
[length(wave1),length(wave2),length(wave3),length(wave4),length(wave5),length(wave6),length
(wave7),length(wave8),length(wave9)];
[~,maxIndex] = max(waveArray);

% making a maxWave variable to hold the largest wave according to the index
% found above

if maxIndex == 1
        maxWave = wave1;
elseif maxIndex == 2
        maxWave = wave2;
elseif maxIndex == 3
        maxWave = wave3;
elseif maxIndex == 4
        maxWave = wave4;
elseif maxIndex == 5
        maxWave = wave5;
elseif maxIndex == 6
        maxWave = wave6;
elseif maxIndex == 7
        maxWave = wave7;
elseif maxIndex == 8
        maxWave = wave8;
elseif maxIndex == 9
        maxWave = wave9;
end

% creating the signals

signal1 = signal(wave1);
signal2 = signal(wave2);
signal3 = signal(wave3);
signal4 = signal(wave4);
signal5 = signal(wave5);
signal6 = signal(wave6);
signal7 = signal(wave7);
signal8 = signal(wave8);
signal9 = signal(wave9);
signalMax = signal(maxWave);

% padding the signals so they are the same length
signal1 = pad(signal1,signalMax);
```

```
signal2 = pad(signal2,signalMax);
signal3 = pad(signal3,signalMax);
signal4 = pad(signal4,signalMax);
signal5 = pad(signal5,signalMax);
signal6 = pad(signal6,signalMax);
signal7 = pad(signal7,signalMax);
signal8 = pad(signal8,signalMax);
signal9 = pad(signal9,signalMax);

%averaging

if M == 2

        syncAvg = (signal1+signal2)/M;

elseif M == 4

        syncAvg = (signal1+signal2+signal3+signal4)/M;

elseif M ==9

        syncAvg =
(signal1+signal2+signal3+signal4+signal5+signal6+signal7+signal8+signal9)/M;

end

End


function [paddedSignal] = pad(signal1, signal2)
% padding signal1 to be the same length as signal2
% singal2 must be larger than signal1

pad = length(signal2) - length(signal1);
signal1 = padarray(signal1,pad,0,'post');
paddedSignal = signal1;

end


function  [hr] = heartRate(crossCor)

[~,hrpeaks] = findpeaks(crossCor, 'MinPeakHeight', 4);
```

```
hrpeaksL = length(hrpeaks);
hr = hrpeaksL./((hrpeaks(hrpeaksL)-hrpeaks(1))/1000);
hr = hr *60;

End

function [locs_Pwave, locs_Qwave, locs_Rwave, locs_Swave, locs_Twave] = pqrst(ECG)

ECGp = ECG;
ECGq = ECG;
ECGq = -ECGq;
ECGr = ECG;
ECGs = ECG;
ECGs = -ECGs;
ECGt = ECG;

maskP = ECG >= 0 & ECG <= 0.1;
maskQ = ECG >= -0.015 & ECG <= 0.0029;
maskR = ECG >= 0.2 & ECG <= 0.4;
maskS = ECG >= -0.25 & ECG <= -0.15;
maskT = ECG >= 0.1 & ECG <= 0.2;

ECGp(~maskP) = nan;
ECGq(~maskQ) = nan;
ECGr(~maskR) = nan;
ECGs(~maskS) = nan;
ECGt(~maskT) = nan;

[~,locs_Pwave] = findpeaks(ECGp, 'MinPeakHeight', 0, 'MinPeakDistance', 600)
[~,locs_Qwave] = findpeaks(ECGq, 'MinPeakHeight', -0.02, 'MinPeakDistance', 500)
[~,locs_Rwave] = findpeaks(ECGr, 'MinPeakHeight', 0.2, 'MinPeakDistance', 100)
[~,locs_Swave] = findpeaks(ECGs, 'MinPeakHeight', -0.3, 'MinPeakDistance', 100)
[~,locs_Twave] = findpeaks(ECGt, 'MinPeakHeight', 0.1, 'MinPeakDistance', 100)


end

function [ECG2calc] = ECG2calculated(ECG1, ECG3)
% Using Einthoven's Law to find expected value of ECG2

ECG2calc = ECG1 + ECG3;

End
```

```
function [absDiff] = absoluteDiff(ECGm,ECGc)
% finding the absolute difference of a measured and calculated value

absDiff = abs(ECGm-ECGc);

End

function [MSE] = meansqrError(signal1, signal2)
% calculating the MSE

signal1(isnan(signal1)) = 0;
signal2(isnan(signal2)) = 0;

MSE = immse(signal1,signal2);

End



function [sectioned] = section(ECG,t,t1)
% making the section wanted for the ECG signal


ECG(t) = nan;
ECG(t1) = nan;
sectioned = ECG;
end
```

PART C

EXERCISE 1.1:

```
%% Opening File
sit = readtable('sitECG.csv');


%% Creating vectors

%extracting time values from the experiment data
sitECG1 = sit(:,2);
```

```
sitECG2 = sit(:,3);
sitECG3 = sit(:,4);
tsit = sit(:,1);


%making the matrix useful for calculations
sitECG1 = sitECG1{:,:};
sitECG2 = sitECG2{:,:};
sitECG3 = sitECG3{:,:};
tsit = tsit{:,:};


%conversion of time array to take into the account the sampling rate (1000Hz)

tsit = 0 : length(tsit) - 1;
tsit = tsit * 0.001;

L = length(sitECG1); % Length of signal
L2 = length(sitECG2);
L3 = length(sitECG3);

y = sitECG1;
y2 = sitECG2;
y3 = sitECG3;

NFFT = 2^nextpow2(L);
NFFT2 = 2^nextpow2(L3);
NFFT3 = 2^nextpow2(L3);


Y = fft(y,NFFT)/L;
Y2 = fft(y2,NFFT2)/L2;
Y3 = fft(y3,NFFT3)/L3;

%Linearly spaced vectors function
f = Fs/2*linspace(0,1,NFFT/2+1);
f2 = Fs/2*linspace(0,1,NFFT2/2+1);
f3 = Fs/2*linspace(0,1,NFFT3/2+1);

% Plot single-sided amplitude spectrum.

figure(1)
plot(f,2*abs(Y(1:NFFT/2+1)))
```

```
title('Frequency Magnitude Spectrum for Channel 1','FontSize', 24)
xlabel('Frequency (Hz)','FontSize', 20)
ylabel('Magnitude Spectrum','FontSize', 20)

figure(2)
plot(f2,2*abs(Y2(1:NFFT2/2+1)))
title('Frequency Magnitude Spectrum for Channel 2','FontSize', 24)
xlabel('Frequency (Hz)','FontSize', 20)
ylabel('Magnitude Spectrum','FontSize', 20)

figure(3)
plot(f3,2*abs(Y3(1:NFFT3/2+1)))
title('Frequency Magnitude Spectrum for Channel 3','FontSize', 24)
xlabel('Frequency (Hz)','FontSize', 20)
ylabel('Magnitude Spectrum','FontSize', 20)
```

Exercise 1.2:

B.

```
sit = readtable('sitECG.csv');
%extracting time values from the experiment data
sitECG1 = sit(:,2);
sitECG2 = sit(:,3);
sitECG3 = sit(:,4);
tsit = sit(:,1);


sitECG1 = sit.lead2(1000:5000); % Selects only four beats from the original signal

%making the matrix useful for calculations
%sitECG1 = sitECG1{:,:};
sitECG2 = sitECG2{:,:};
sitECG3 = sitECG3{:,:};
tsit = tsit{:,:};

tsit = 0:length(sitECG1)-1;


[b,a] = sos2tf(SOSb4, Gb4);
ecgbutter = filter(b,a,sitECG1);
```

```
[b,a] = sos2tf(SOS, G);
ecgcheby = filter(b,a,sitECG1);


figure(1)
subplot(3,1,1); plot(tsit, sitECG1); title('4 Beats of Normal sitECG signal','FontSize',
20);xlabel('Time in seconds','FontSize', 20);ylabel('mV','FontSize', 20);
subplot(3,1,2); plot(tsit, ecgbutter); title('Butterworth filter of order 4','FontSize', 20);xlabel('Time
in seconds','FontSize', 20); ylabel('mV','FontSize', 20);
subplot(3,1,3); plot(tsit, ecgcheby); title('Chebyshev filter of order 4','FontSize', 20);xlabel('Time
in seconds','FontSize', 20);ylabel('mV','FontSize', 20);


figure(2)
plot(tsit, sitECG1,'LineWidth',3);
hold on
plot(tsit, ecgbutter,'LineWidth',3);
hold on
plot(tsit, ecgcheby,'LineWidth',3);
 title('sitECG signal along with Butterworth and Chebyshev filters of order 4','FontSize', 20);
xlabel('Time in seconds','FontSize', 20);
ylabel('mV','FontSize', 20);
legend('4 Beats of Normal sitECG signal','Butterworth filter of order 4','Chebyshev filter of order
4');
```

C.

```
sit = readtable('sitECG.csv');
%extracting time values from the experiment data
sitECG1 = sit(:,2);
sitECG2 = sit(:,3);
sitECG3 = sit(:,4);
tsit = sit(:,1);


sitECG1 = sit.lead2(1000:5000); % Selects only four beats from the original signal

tsit = tsit{:,:};

tsit = 0:length(sitECG1)-1;
```

```
[b,a] = sos2tf(SOS, G);
ecgbutter = filter(b,a,sitECG1);

[b,a] = sos2tf(SOS2, G2);
ecgbutter2 = filter(b,a,sitECG1);

[b,a] = sos2tf(SOS3, G3);
ecgbutter3 = filter(b,a,sitECG1);


figure(1)
subplot(3,1,1); plot(tsit, ecgbutter); title('Butterworth filter with order 8','FontSize',
20);xlabel('Time in seconds','FontSize', 20);ylabel('mV','FontSize', 20);
subplot(3,1,2); plot(tsit, ecgbutter2); title('Butterworth filter with order 20','FontSize',
20);xlabel('Time in seconds','FontSize', 20); ylabel('mV','FontSize', 20);
subplot(3,1,3); plot(tsit, ecgbutter3); title('Butterworth filter with order 50','FontSize',
20);xlabel('Time in seconds','FontSize', 20);ylabel('mV','FontSize', 20);
```

D.

```
sit = readtable('sitECG.csv');

%extracting time values from the experiment data
sitECG1 = sit(:,2);
sitECG2 = sit(:,3);
sitECG3 = sit(:,4);
tsit = sit(:,1);
tsit = tsit{:,:};

% Selects only four beats from the original signal
sitECG1 = sit.lead2(1000:5000);
tsit = 0:length(sitECG1)-1;

%order 8
[b,a] = sos2tf(SOS, G);
ecgbutter = filter(b,a,sitECG1);

%order 20
[b,a] = sos2tf(SOS2, G2);
ecgbutter2 = filter(b,a,sitECG1);

%order 50
```

```
[b,a] = sos2tf(SOS3, G3);
ecgbutter3 = filter(b,a,sitECG1);

L = length(ecgbutter);
L2 = length(ecgbutter2);
L3 = length(ecgbutter3);

%sampling frequency
fs = 1000;

Y = fft(ecgbutter);
Y2 = fft(ecgbutter2);
Y3 = fft(ecgbutter3);

P2 = abs(Y/L); %ORDER 8
P22 = abs(Y2/L2); %ORDER 20
P23 = abs(Y3/L3); %ORDER 50

P1butter = P2(1:L/2+1);%ORDER 8
P1butter2 = P22(1:L2/2+1);%ORDER 20
P1butter3 = P23(1:L3/2+1);%ORDER 50

P1butter(2:end-1) = 2*P1butter(2:end-1);
P1butter2(2:end-1) = 2*P1butter2(2:end-1);
P1butter3(2:end-1) = 2*P1butter3(2:end-1);

fbutter = fs*(0:(L/2))/L;
fbutter2 = fs*(0:(L2/2))/L2;
fbutter3 = fs*(0:(L3/2))/L3;


figure(1)
subplot(3,1,1); plot(fbutter, P1butter); title('Butterworth filter with order 8','FontSize',
20);xlabel('Frequency (in Hz)','FontSize', 20);ylabel('mV','FontSize', 10);
subplot(3,1,2); plot(fbutter2, P1butter2); title('Butterworth filter with order 20','FontSize',
20);xlabel('Frequency (in Hz)','FontSize', 20); ylabel('mV','FontSize', 10);
subplot(3,1,3); plot(fbutter3, P1butter3); title('Butterworth filter with order 50','FontSize',
20);xlabel('Frequency (in Hz)','FontSize', 20);ylabel('mV','FontSize', 10);
```