

Lab #3: Design of a Bioinstrumentation Amplifier for ECG

Submitted: November 3rd, 2022

Name	Student #	Lab Contribution
Sophie Mansfield	400178392	Questions A2, A5, B3, B8, B11
Lauren Stephens	400270259	Questions B6, B9, B10
Jiaqi Du	400249700	Questions C1, C2, D3

Part A: Building the Circuit

Question A2. How can the second amplification stage be bypassed without breaking the circuit?

The amplifier is bypassed by shorting R_2 with a wire after the first high-pass filter to make the resistance 0, so that a gain of 1 (unity gain) is applied and the signal isn't amplified.

$$G = 1 + \frac{R_2}{R_3}$$

$$G = 1 + \frac{0}{R_3}$$

$$G = 1$$

Question A5. Plot the gain vs. frequency response for signals between DC and 300 Hz.

The gain values were recorded at different frequency values, shown below in Table 1 and plotted in Figure 1.

Table 1. Circuit output voltages and gain values calculated for different input frequencies

Frequency (Hz)	$V_{in}(mV\ p - p)$	$V_{out}(mV\ p - p)$	$G = V_{out}/V_{in}$
1	10	1900	190
10	10	2070	207
50	10	1050	105
60	10	780	78
100	10	1550	155
200	10	1070	107
300	10	500	50

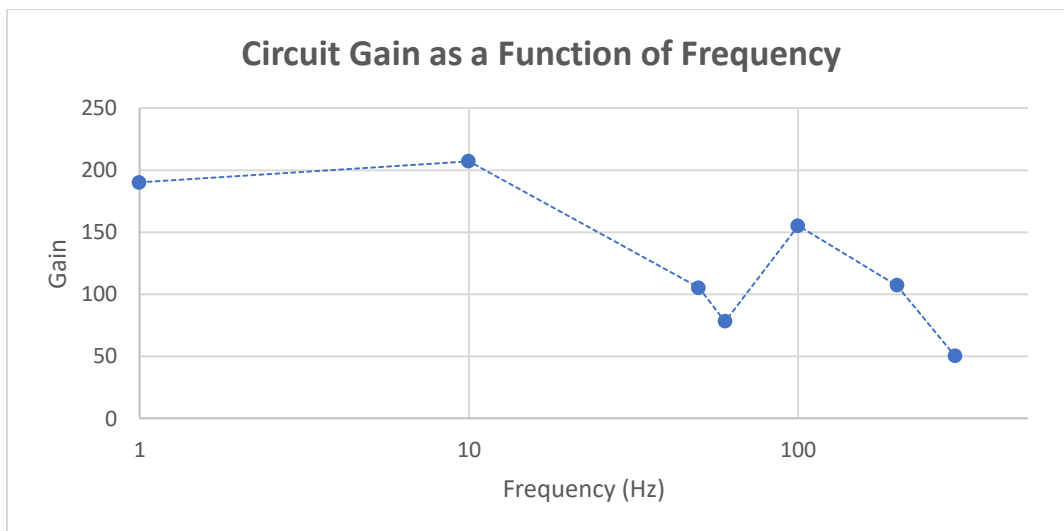


Figure 1. Circuit gain as a function of frequency (semi-log plot)

Part B: Acquire the ECG Signal

Question B3. The heart frequency rate for “normal sinus rhythm” was varied from 60, 80, and 120 BPM. The results were captured with the oscilloscope.

The oscilloscope results for capturing normal sinus rhythm at varying BPM via the patient simulator are shown in Figures 2-4.

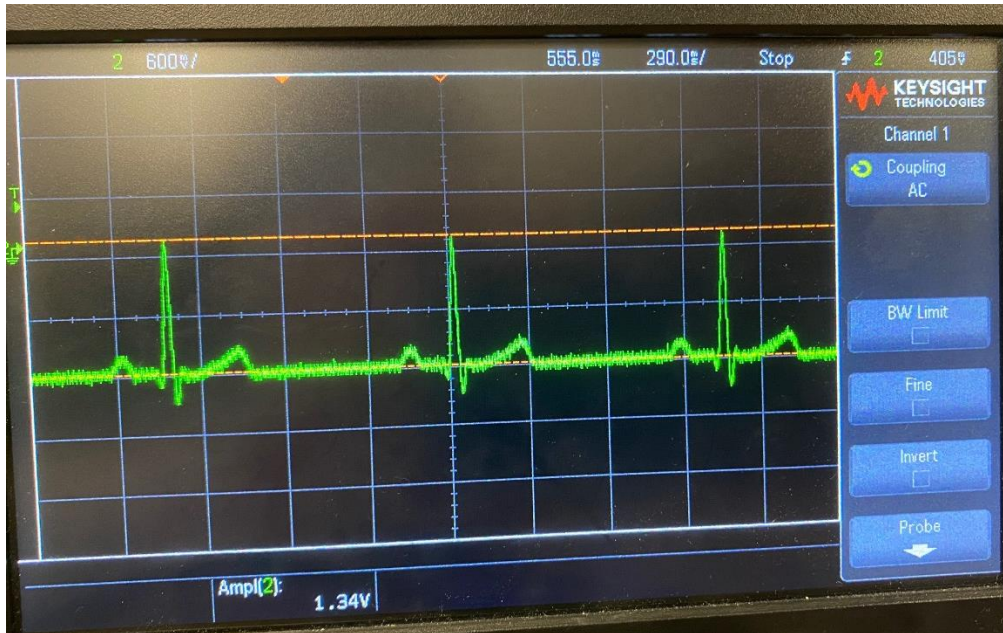


Figure 2. Oscilloscope response for a 60 BPM heart rate (normal sinus rhythm)

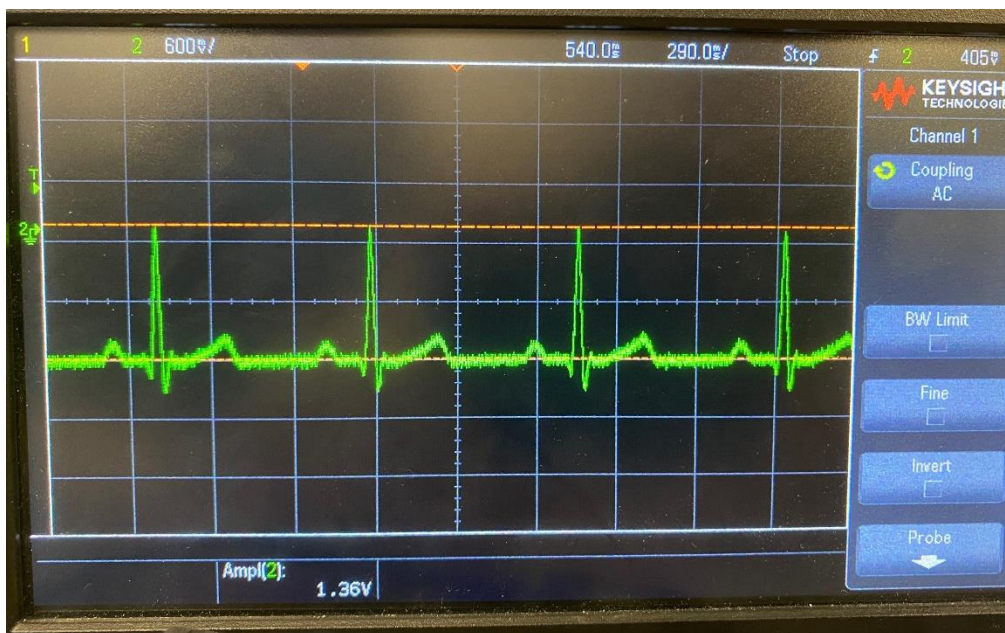


Figure 3. Oscilloscope response for a 80 BPM heart rate (normal sinus rhythm)

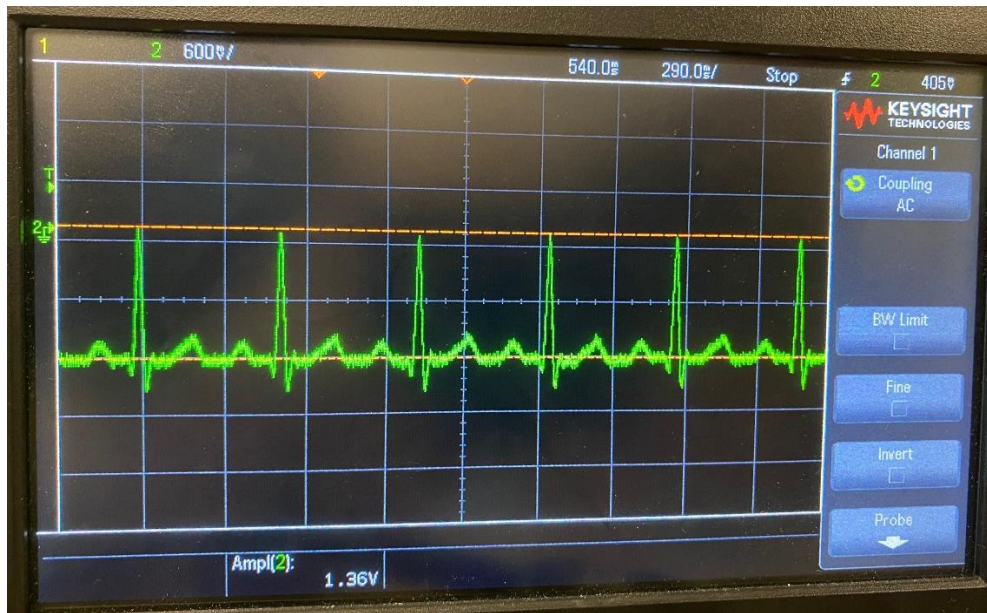


Figure 4. Oscilloscope response for a 120 BPM heart rate (normal sinus rhythm)

Question B6. Use MATLAB or other software to compute the heart rate from the data

The computed heart rate from the data was 84 beats per minute. This was calculated by finding the number of local minima using MATLAB's findpeaks function (shown in Appendix 1). Setting 'MinPeakHeight' to below a certain threshold (-0.5V) means only the lowest points on the ECG will be tallied. One potential problem is noise introducing multiple peaks underneath the threshold with each heartbeat (Figure 6). These peaks were removed by introducing a 'MinPeakDistance' of 100, which corresponds to 0.1 seconds. If a second peak is found within 0.1 seconds of another, it is discarded, since it can be assumed that this peak is noise and not another heartbeat. The total peaks found within 10 seconds is multiplied by 6 to get the beats per minutes. The ECG data was then plotted to verify the results.

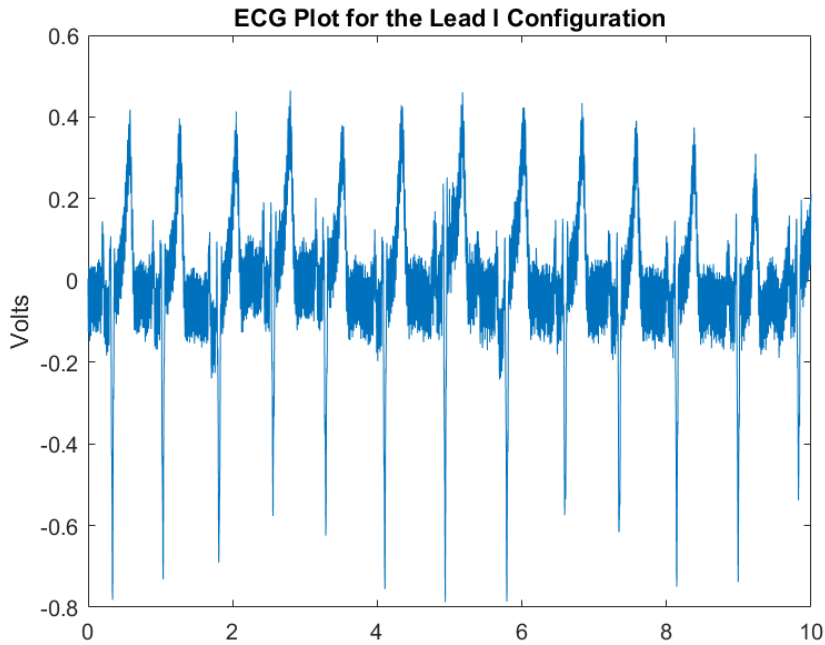


Figure 5. Output graph used to verify Lead I heart rate calculations.

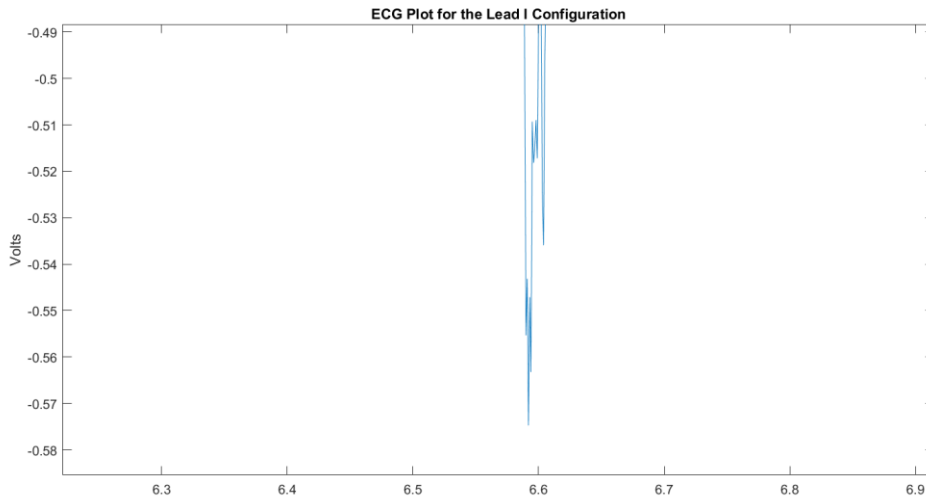


Figure 6. Zoomed in graph showing multiple peaks from noise

Question B8. Write MATLAB code that digitally filters the signals using a 6th order bandpass filter with passband 10-25Hz and display the results. Is it easier to compute the heart rate now (using the code written in B6)?

A 6th order bandpass filter was created in Matlab using `fdesign.bandpass` (shown in Appendix 2). This filter was specified as order 6 with minimum and maximum cutoff frequencies of 10Hz

and 25Hz, respectively, and a sampling rate of 1000Hz. The ECG traces, in this case, are much more visible, with less noise impacting the signal. The reduction of noise means the 'MinPeakDistance' term is not needed when finding the peaks, which simplifies the process. Changing the threshold to 0.2V produces the same heart rate (78 beats/min).

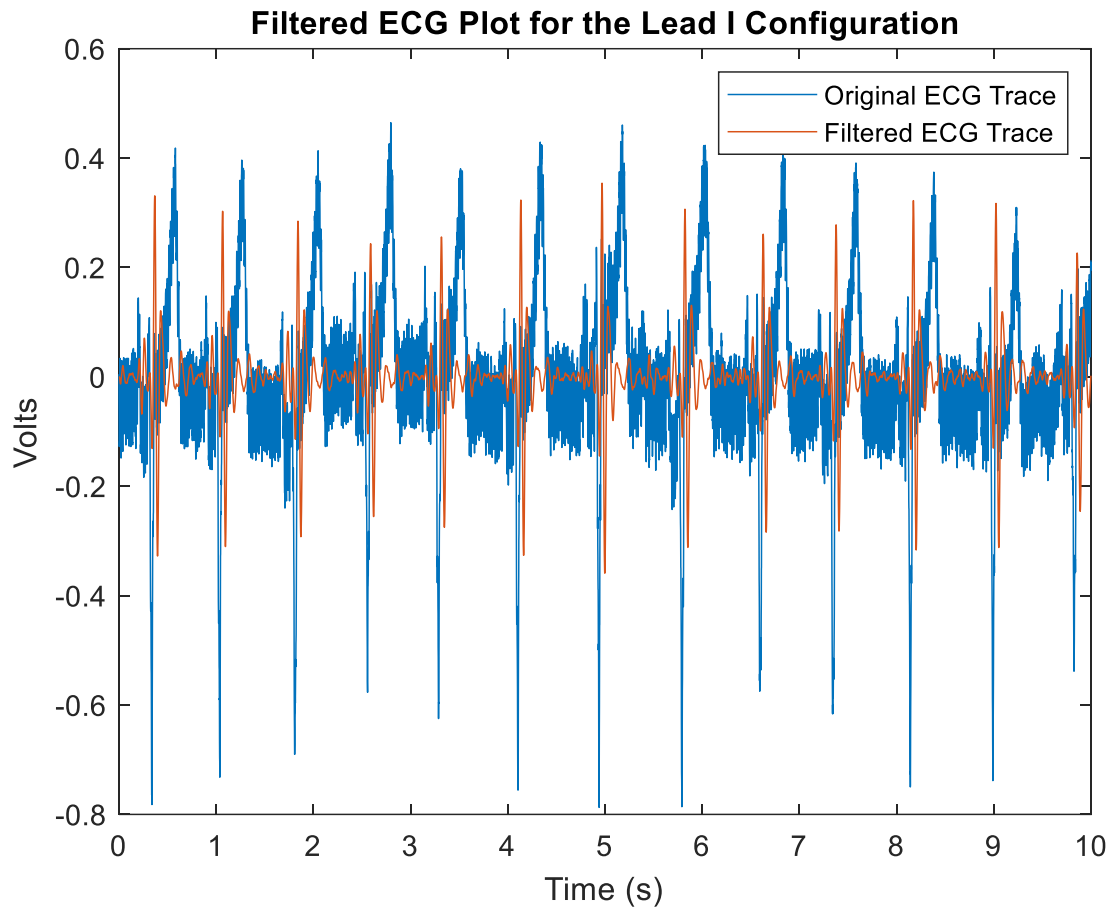


Figure 7. Output graph for data after filtering with a 6th order bandpass filter.

Question B9. Repeat B6 but with the input electrodes in the Lead II configuration

The computed heart rate from the data was 84 beats per minute. Due to the different lead configuration, this value was calculated by finding the number of local maxima above a certain threshold (2.5V) within 10 seconds and multiplying it by 6. The ECG data was then plotted to verify the results.

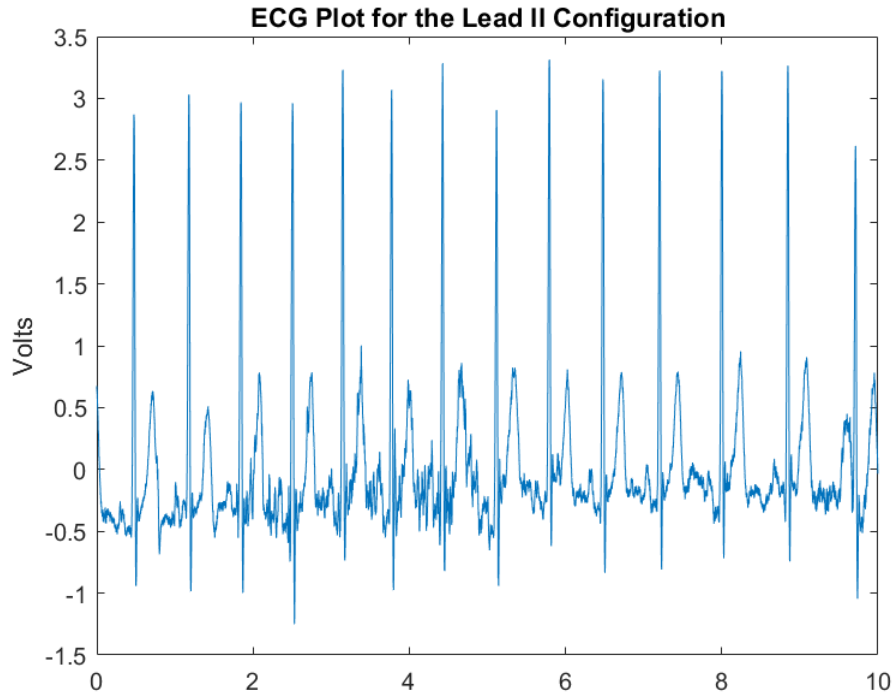


Figure 8. Output graph used to verify Lead II heart rate calculations.

Question B10. Repeat B6 but with the input electrodes in the Lead III configuration

The computed heart rate from the data was 78 beats per minute. Due to the different lead configuration, this value was calculated by finding the number of local maxima above a certain threshold (2.5V) within 10 seconds and multiplying it by 6. The ECG data was then plotted to verify the results.

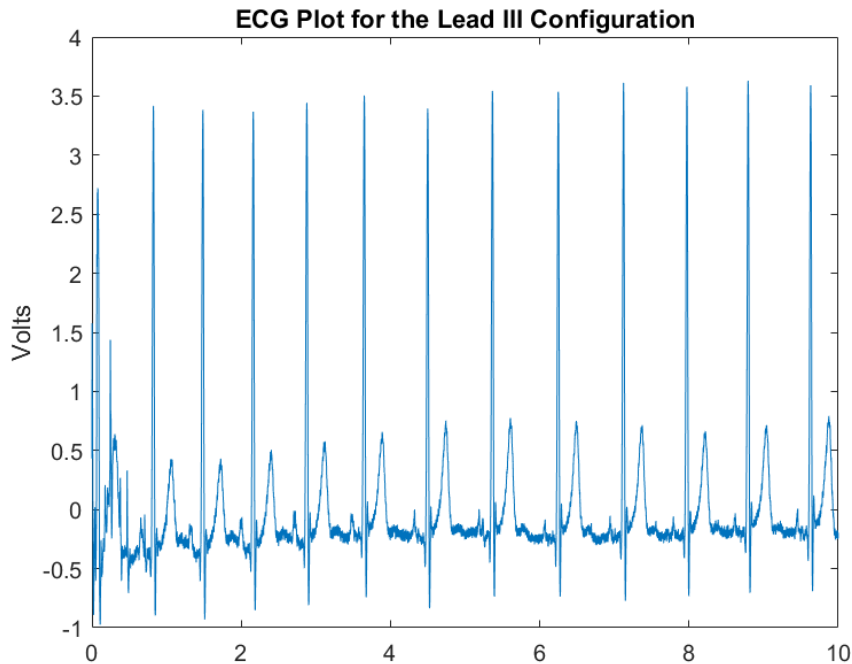


Figure 9. Output graph used to verify Lead II heart rate calculations.

Question B11. *Examine the PQRST components in your ECG traces. How do they differ for the 3 leads? Why?*

For the lead I configuration (Figure 5), the only distinguishable component of the unfiltered ECG trace is the R peak of the QRS complex. Lead II produces the most distinguishable trace because the amplitudes of the components are equal to the sum of the amplitudes of the components in the Lead I and Lead III configurations (according to Einthoven's Law). This explains why the R peaks (for example) are higher in Figure 8 than in Figures 5 and 9. The same is true for the P and T wave amplitudes. When electrical activity is travelling toward an electrode, it is displayed as an upward deflection, while electrical activity travelling away from an electrode is shown as a downward deflection. Thus, the highest signals observed in the Lead I configuration are negative; the heart is being observed "from the left (at 0°)." Contrastingly, the "exploring electrode" for Lead II and III configurations is that on the left leg; lead II observes the heart from an angle of 60° while lead III observes the heart from a 120° angle. Thus, the maximal positive deflected recorded in lead II occurs when the depolarization wave travels parallel to the axis between the right arm and left leg. The same occurs in lead III but for the axis between the left arm and left leg. For lead I, this axis is between the right and left arms.

Part C: Check the Performance of the System

Question C1. Find the magnitude of the DC offset in your input signal in B7. Where is it coming from since you have a high-pass filter which should block all DC?

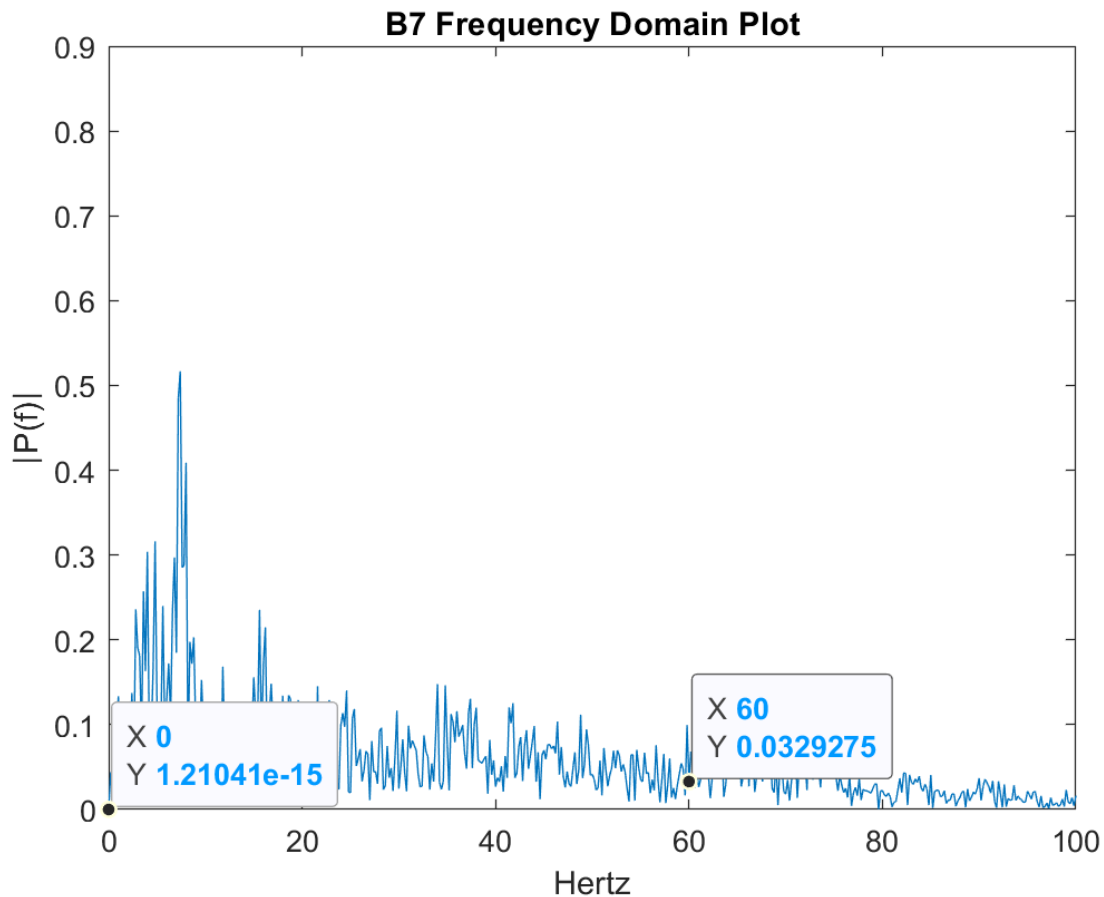


Figure 10. Frequency plot of noisy signal

The magnitude of the DC offset, from calculating the mean of the signal in the time domain, is $5.932143665177137e-16$, but the frequency domain plot claims that it's $1.21041e-15$, which are both very low values, although it isn't zero, which is what is expected when using a high pass filter. Since the high pass filter isn't perfect, current will still pass through the resistor instead of the capacitor, leading to a DC offset. However, the offset is negligible compared to the strength of the other frequencies, which means most of the signal still passed through the capacitor and was filtered out.

Question C2. Is there any 60Hz in your recorded ECG in B7? Where is it coming from? Hint: do FFT

There is a 60Hz frequency recorded in B7, at a strength of 0.0329. The 60Hz is caused by a powerline noise, which would have a much higher strength if not for the 60Hz notch filter. Just

like the high pass filter mentioned in question C1, the notch filter is not perfect either, which causes unfiltered signal to leak through. The notch filter was successful in attenuating the 60Hz powerline noise, but not completely eliminating it. In this case, since the strength of the powerline noise is much higher than the strength of the DC offset, the unfiltered leaked through signal is also at a higher strength than the DC offset.

Part D: Recording and Analyzing Fetal/Maternal ECG

Question D3 (BONUS). Write a MATLAB program to attempt to separate the signal into a maternal ECG and a fetal ECG and determine the heart rate of both.

The heart rate was separated using the *findpeaks* function in Matlab. As shown in the figure below, the fetal heartrate signal is usually higher due to electrode placement, so it is assumed that the higher peaks denotes the fetal heartrate, while the lower peaks denotes the maternal heartrate.

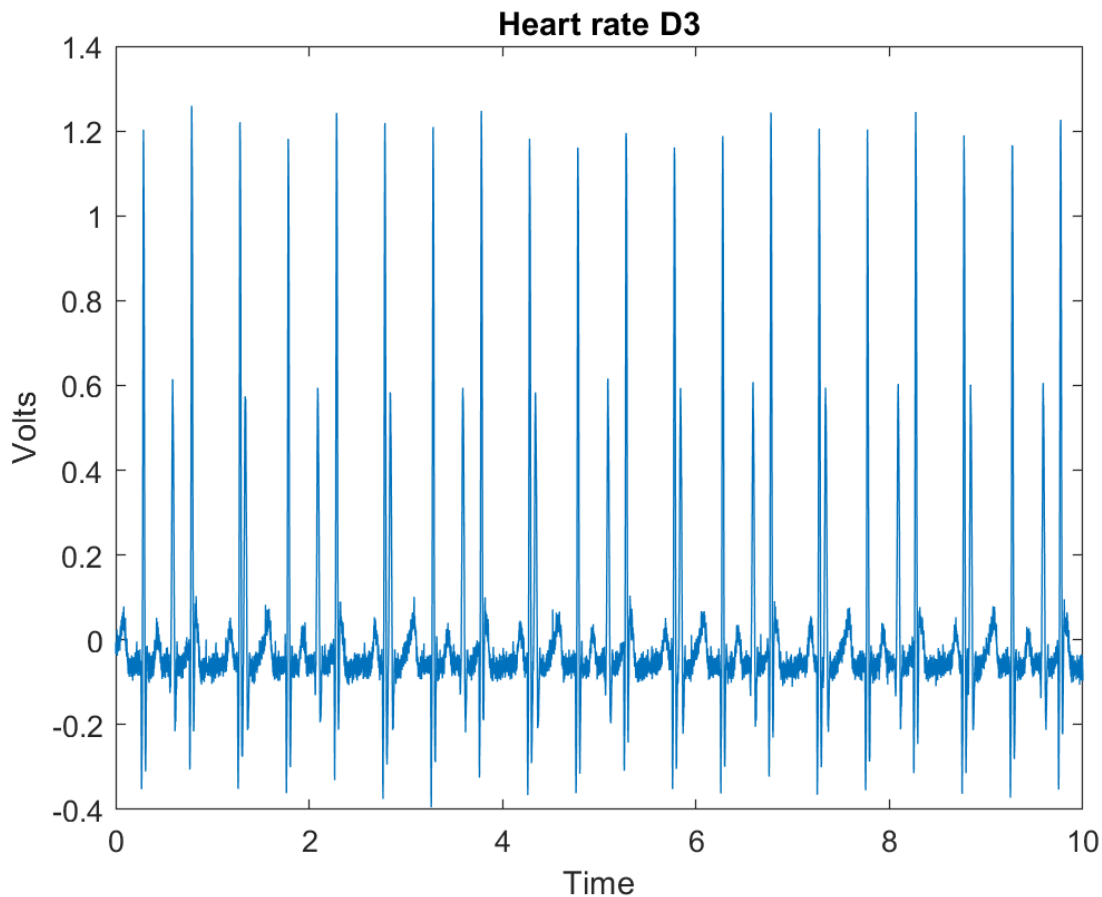


Figure 11. The time domain of the heart rates. Maternal ECG noise can be seen with lower peaks.

Using the *findpeaks* function with a minimum peak height of 1, the fetal heartrate can be separated:

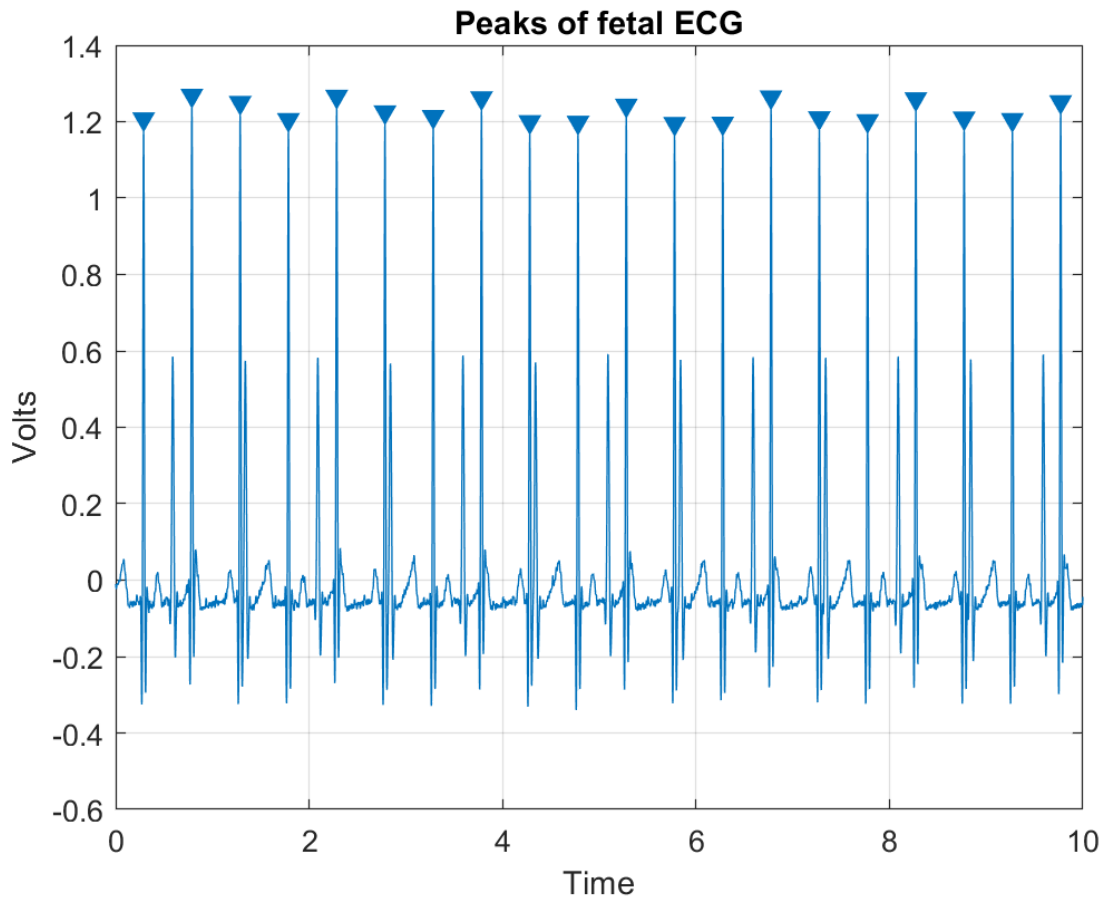


Figure 12. The peaks of the fetal heartbeat, which are stored in the variable “fECG” in matlab code, counted as 20 peaks.

The peak’s coordinates were stored and counted as 20, and since this occurred over 10 seconds, multiplying the count by 10 resulted in a fetal heartrate of 120bpm.

For maternal heartbeat, there was no method to isolate the peaks between a y limit of 0.5 and 1, so the peaks of both maternal and fetal heartbeat were found, which was then subtracted by the peaks of the fetal heartbeats found earlier in the question.

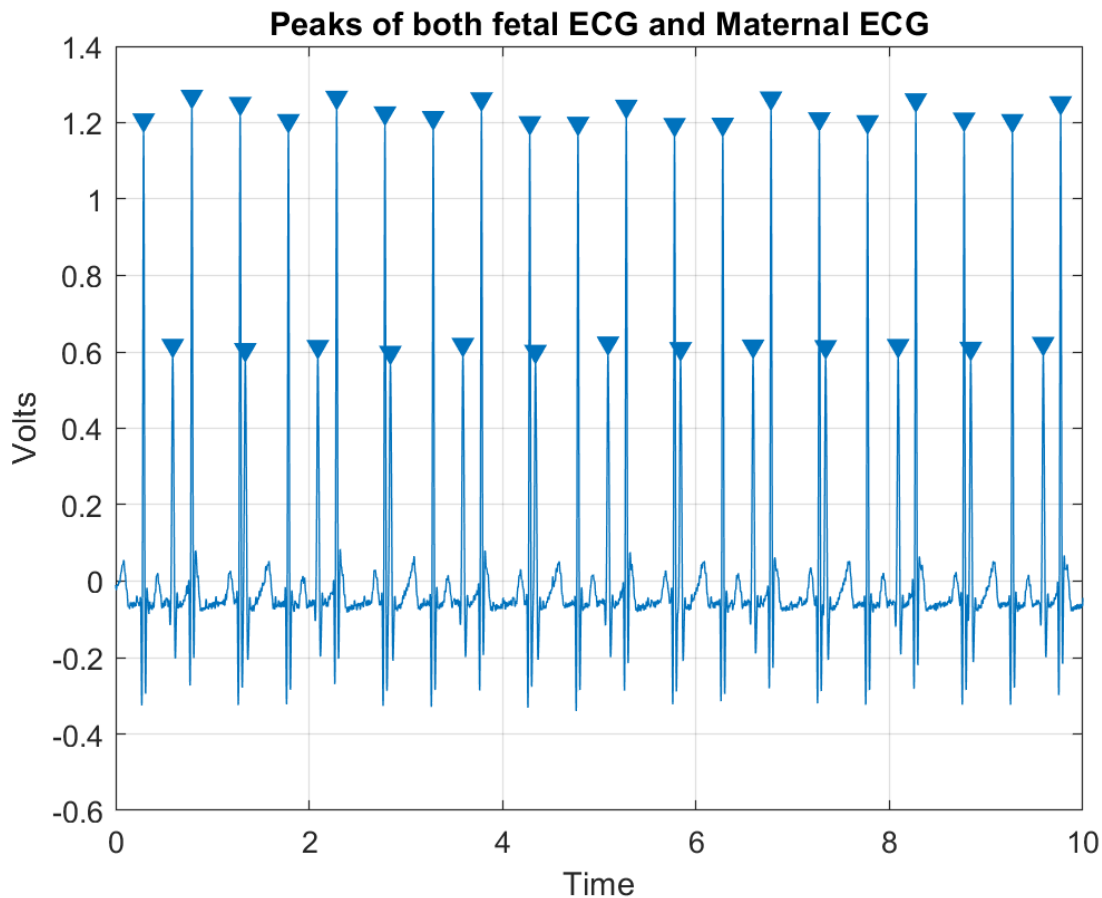


Figure 13. The peaks of both maternal and fetal heartbeat.

The total peaks were found to be 33, subtracted from the fetal heartbeat of 20 to result in a maternal heartbeat of 13, which was multiplied by 6 to be 78 bpm. Therefore, the fetal heartrate is 120bpm, and the maternal heartrate is 78 bpm, based on the data shown above.

Appendix 1: MATLAB Code for Computing Heart Rate

```
% B6
clc;
clear all;
close all;

B6 = load('B6.mat');
B6_cell = struct2cell(B6);
B6_array = cell2mat(B6_cell);
time = (0:1/1000:10-1/1000)';

pks = findpeaks(-B6_array, 'MinPeakHeight',0.5, 'MinPeakDistance',100)
num_pks = length(pks)
heart_rate = num_pks * 6
```

```
plot(time, B6_array);
title('ECG Plot for the Lead I Configuration')
ylabel('Volts');

%%
% B9
clc;
clear all;
close all;

B9 = load('B9.mat');
B9_cell = struct2cell(B9);
B9_array = cell2mat(B9_cell);

pks = findpeaks(B9_array, 'MinPeakHeight',2.5, 'MinPeakDistance',100)
num_pks = length(pks)
heart_rate = num_pks * 6

time = (0:1/1000:10-1/1000)';
plot(time, B9_array);
title('ECG Plot for the Lead II Configuration')
ylabel('Volts');

%%
% B10
clc;
clear all;
close all;

B10 = load('B10.mat');
B10_cell = struct2cell(B10);
B10_array = cell2mat(B10_cell);

pks = findpeaks(B10_array, 'MinPeakHeight',2.5, 'MinPeakDistance',100)
num_pks = length(pks)
heart_rate = num_pks * 6

time = (0:1/1000:10-1/1000)';
plot(time, B10_array);
title('ECG Plot for the Lead III Configuration')
ylabel('Volts');
```

Appendix 2: MATLAB Code for Bandpass Filter

```
%IBEHS 4F03 Lab 3 (B8)

clc
clear all
close all

ECG = importdata("B6.mat");
```

```

N = 6;      % Order
Fc1 = 10;   % First
Fc2 = 25;   % Second
Fs = 1000;  % Sampling Frequency

h = fdesign.bandpass('n,Fc1,Fc2', N, Fc1, Fc2, Fs);
Hd = design(h, 'butter');
ECG_filtered = filter(Hd,ECG);

time = (0:1/1000:10-1/1000)';

plot(time, ECG, time, ECG_filtered)
legend('Original ECG Trace', 'Filtered ECG Trace')
title('Filtered ECG Plot for the Lead I Configuration')
ylabel('Volts');
xlabel('Time (s)');

pks = findpeaks(-ECG_filtered, 'MinPeakHeight',0.2)
num_pks = length(pks)
heart_rate = num_pks * 6

```

Appendix 3: MATLAB Code for C1-2

```

clc;
clear all;
close all;
load('B7.mat');
avg = mean(data);
%plot frequency domain using fourier
%c7 = timetable2table(C7);
Fs = 1000; %sampling freq
T = 1/Fs; %sampling time
StopTime = 5; %time in seconds
t = (0:T:StopTime-T);
%L = 5000; %sampling length
L = size(t,2);
x = fft(data); %fourier transform the signal
f = Fs*(0:(L/2))/L; %create x-axis
%dF = Fs/L;
%f = -Fs/2:dF:FsWithoutZero;
p2 = abs(x)/L; %calculate the 2 sided spectrum
p1 = p2(1:L/2+1); %calculate the 1 sided spectrum
p1(2:end-1)= 2*p1(2:end-1);
%fl = linspace(0,500,L);
plot(f,p1);
xlabel('Hertz');
ylabel('|P(f)|');
ylim([0,0.9]);
xlim([0,100]);
title('B7 Frequency Domain Plot');

```

Appendix 4: MATLAB Code for D3 (Bonus)

```
clc;
clear;
close all;
%plot time domain of C7
d = linspace(0,10,10000);
load('D3.mat');
space = 0.001;
figure;
plot(d,data);
title('Heart rate D3')
ylabel('Volts');
xlabel('Time');
d = d';
d2 = array2table(d);
data2 = array2table(data);
datat = table2timetable(data2,'SampleRate',1000);
D3_pass = lowpass(datat,30);
figure;
plot(d,D3_pass.data);
title('50hz low pass')
ylabel('Volts');
xlabel('Time');
%determining BPM of both
figure;
findpeaks(D3_pass.data,d,'MinPeakHeight',1.14);
[pks,locs,w,p] = findpeaks(D3_pass.data,'MinPeakHeight',1.14);
title('Peaks of fetal ECG');
ylabel('Volts');
xlabel('Time');
fECG = length(pks);
fetalbpm = fECG*6;
figure;
findpeaks(D3_pass.data,d,'MinPeakHeight',0.5);
ylabel('Volts');
xlabel('Time');
title('Peaks of both fetal ECG and Maternal ECG')
[pks1,locs1,w1,p1] = findpeaks(D3_pass.data,'MinPeakHeight',0.5);
mECG = length(pks1)- length(pks);
maternalbpm = mECG*6;
```