Team #501

BE 312 Spring 2021 Lab 4, Signals

Sunday, April 18th, 2021

Connor Bittlingmaier, Mathew Fiel, Lauren McLaughlin-Kelly, Ricky Palacio

Abstract

In this lab, we used MATLAB and its associated signal processing and wavelet toolboxes to augment, analyze, and display raw electrocardiography data after processing. Raw ECG data was processed using a low pass filter function to reduce noise and discern specific phases of the P-QRS-T cycle. We then realigned this filtered data using a provided wavelet function in order to remove drift from the baseline, essentially 'flattening' our data and making peaks more consistent with each other. The filtered and re-aligned dataset was then analyzed to determine the values, corresponding array locations, width, and prominence of the peaks in the QRS complexes. This data was also used to determine the heartrate of the subject by taking the number of peaks evaluated and dividing it by the total time span of the dataset. During this lab, for all aforementioned steps and procedures, plots and subplots were generated where required to graphically show the effects of filtering, baseline drift correction, and peak identification within the raw and filtered datasets. This lab allowed us to create a filtered ECG signal that was capable of being analyzed based on its individual complexes.

Questions

General

1. All .m files, input and output files, and the README.txt file must be in the shared Box folder for your team. The answer to this question is “Yes” if you have done this. Full credit will be given if these all are in the folder.

Yes.

Noise Removal

2. Provide a time plot of your data before filtering (just the raw signal) and after filtering. Discuss the differences, if any.

The time plot of our data before filtering was very difficult to read due to the inability to locate specific data points and interpret them. There were so many data points between ~ 50 Hz to ~ 100 Hz, which would make the QRS complex, R-R interval, and the overall ECG data indiscernible.

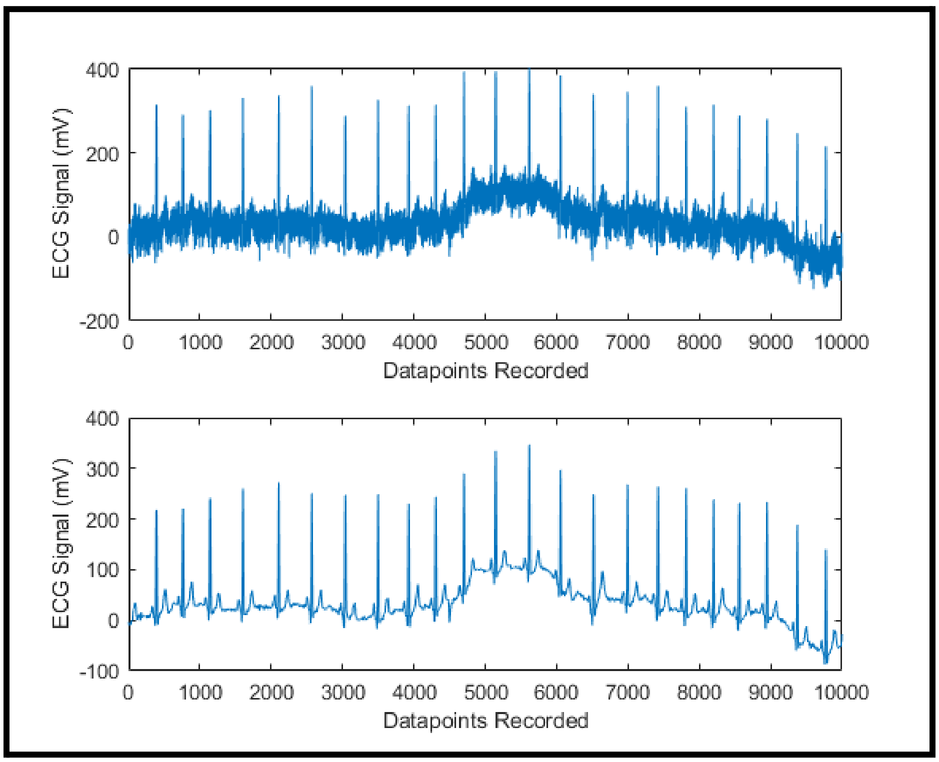

Figure 1: Subplot of the raw ECG data (top) and the raw data after being run through a low pass filter (bottom)

After the data was filtered, it was much easier to read and interpret its characteristics (i.e. QRS complex, R-R interval, etc.). Although we removed most of the noise, background noise is still present, which can be due to several factors such as the subject moving during data collection and environmental factors.

3. What type of filtering did you use to remove noise? Why? Is there any noise remaining? Why or why not?

We used a low pass filter to remove most of the noise. We successfully removed most of the background noise that ranged from about ~ 50 Hz to ~100 Hz, which gave us a much clearer view of the ECG data. By using a low pass filter we removed all the high frequency noise, leaving us with just the low frequency data/ECG data. If we had left the noise, we wouldn't have been able to clearly interpret the data from the ECG, such as R-R interval, QRS complex, as well as the R height.

4. State the name of the Matlab program that does the noise removal.

The Matlab file *ECG_noise_removal_0501.m* calculates the noise removal in section one. The noise removal portion of the code terminates at line 25, where the time values from the raw data and the filtered ECG data are collated into the file 'ECG_noise_filtered_0501.txt'.

This portion of the lab uses the lowpass function to filter the ECG data from the ECG_rawdata_0501.txt file. This function filters the raw data within set parameters for sampling rate and pass frequency. This filtered data was then plotted against time, which is represented in the first column of ECG_rawdata_0501.txt. Collating time and filtered ECG into a new 10000x2 array saved as ECG_noise_filtered_0501.txt.

Baseline Drift Correction

5. You should have used the Wavelet filtering code provided to remove the baseline drift. Provide a plot of before correction and after correction. Discuss the differences, if any. Do you think the baseline drift has been removed? Why or why not?

We used the Wavelet filtering code which was provided to us, in order to attempt to remove the baseline drift. The plot before the correction had a baseline drift that was very prominent around the X-axis value of 5000. This baseline drift caused the data to create a hill-like wave which causes difficulty when trying to read and interpret the data.

We think the baseline drift has been removed because the plot after the correction was leveled out. Meaning we are able to see the data in a straight line, as opposed to a drifting line, making it much easier to read and interpret the data.

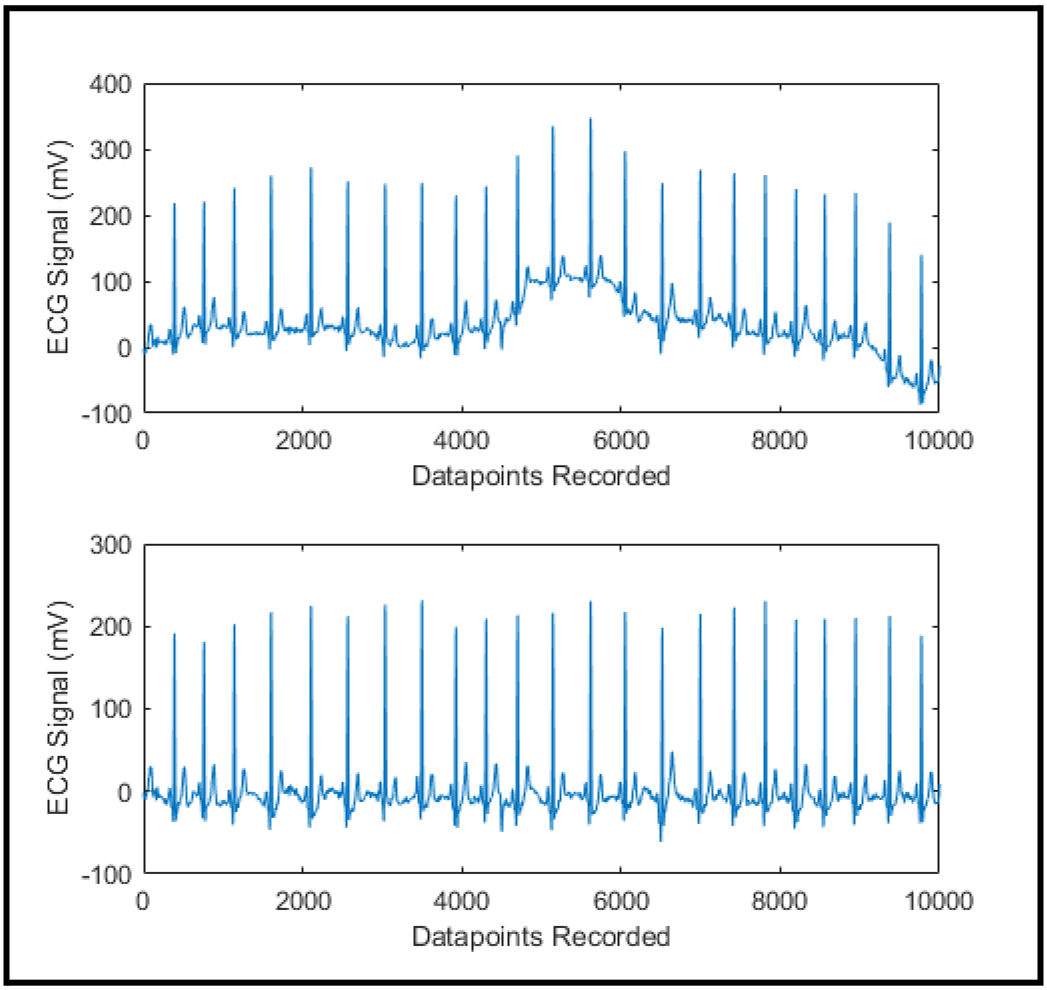

Figure 2: Subplot of the filtered ECG signal (top) and the same filtered ECG signal following baseline drift correction (bottom)

6. State the name of the Matlab program that does the baseline drift correction.

The Matlab file *ECG_baseline_correction_0501.m* calculates the baseline drift correction in section two. This portion of code was provided from a given wavelet file on UBLearns, which utilizes wavelets to correct the baseline drift of the filtered signal. This equalizes the function to a uniform baseline, essentially ‘flattening’ the signal.

R Peaks

7. Provide a summary of the value (height) and location (in time) of each R peak in your data. This should be a table. Use Matlab to put these values into a text file so you can easily make a table.

Table 1: R Peak Times and Voltages

	Time (seconds)	R Peak (mV)
1	0.776	191.391
2	1.520	181.121
3	2.280	202.287
4	3.206	217.160
5	4.208	224.797
6	5.130	212.342
7	6.072	226.276
8	6.996	231.479
9	7.850	199.426
10	8.608	208.933
11	9.398	213.578
12	10.286	216.396
13	11.234	230.477
14	12.100	217.643
15	13.032	198.376
16	13.988	215.086
17	14.840	222.875
18	15.622	230.507
19	16.394	208.208
20	17.118	208.688
21	17.892	209.864
22	18.746	212.770
23	19.546	188.627

8. State the name of the Matlab program that does the R peak determination.

The Matlab file *ECG_Rpeak_locator_0501.m* calculates the R peaks in section three. The function used was the findpeaks function with the filtered and baseline corrected ECG data. This function gave us the values of any peaks, the corresponding array location, the width of the peaks, and the prominence of the peaks. In this example, width gives values corresponding to array locations. Prominence is a value that represents how much a peak stands out from other peaks based on relative size and location.

Using these values, we deleted any peak with a max value below 18mV, width greater than 17, or prominence below 30. Additionally, we deleted any peaks that had a max value that was less than 1.7 times the width of the peak. These numbers were chosen by running the findpeaks function and manually analyzing the data. Using these cutoff values with a for loop, allows us to determine the R peaks, while ignoring any other peaks.

Another for loop was used to convert the location values from the findpeak function into the corresponding time values. Finally, time and R peak values were printed to the text file ECG_Rpeaks_0501.txt, with time being the first column and R peak values being the second column.

Heart Rate

9. Manually calculate the heart rate for your subject. Explain how you did this.

To calculate the heart rate for our subject the number of R peaks were visually counted from the figure produced with noise removed and baseline drift corrected.

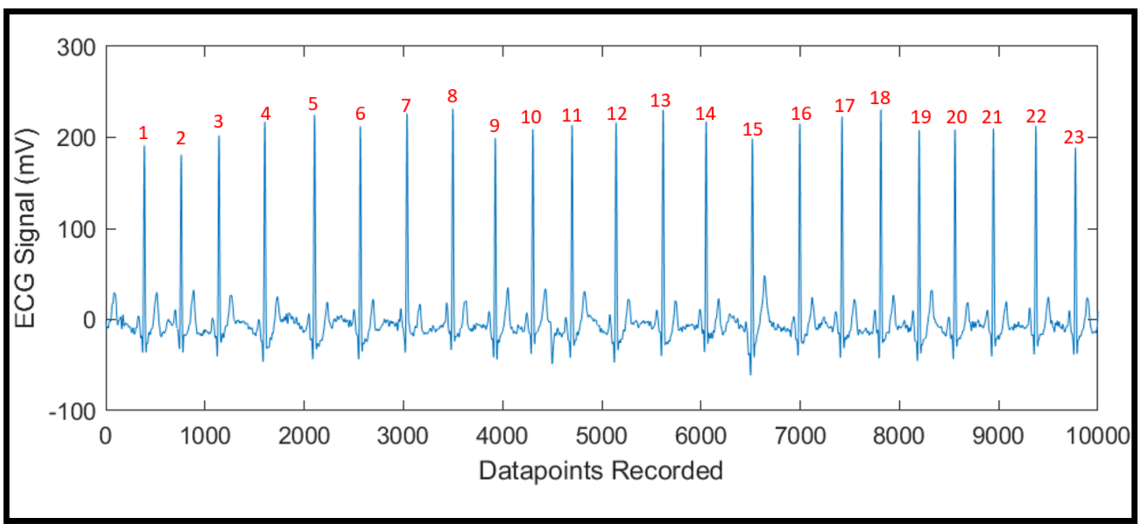

Figure 3: Filtered and Baseline Drift Corrected ECG Signal plotted against 10000 total data points with R-peaks numbered 1 to 23

$$BPM = (23\, beats\ /\ 20\, seconds) * 60\, seconds = 69 bpm$$

The 23 beats counted were divided by the total time of the data set, that being 20 seconds. This value was then multiplied by 60 seconds to get a value of 69 beats per minute.

10. Use Matlab to automatically calculate the mean heart rate for your subject. Have your code put the result in a text file called "MeanHeartrate_GGGG.txt". Discuss how you calculated this value.

The Matlab code uses the length function with R-peaks to calculate the total number of heartbeats recorded. Then the time of the first recorded heartbeat is subtracted from the time of the last recorded heartbeat. Since the start time corresponds to the first heartbeat, we subtract one from the total number of beats given by the length function. Total heartbeats minus 1, divided by the calculated time difference, gives the heartbeats per second. Multiplying this by 60 gives the heartbeats per minute. This is done in the Matlab file and gives a value of 70.325 bpm which is saved to the text file called MeanHeartrate_0501.txt

11. Calculate the variance or standard deviation of time for each R-to-R peak. Discuss this value.

Using Excel, we calculated the time between each R peak. We did this by subtracting the time of R Peak 1 from the time of R Peak 2 and so on for the 23 recorded R peaks. Then we used Excel to calculate the average and standard deviation of the time between each R peak. The average was 0.853 seconds and the standard deviation was 0.0826 seconds. This is a fairly small standard deviation, so we can expect the time between R peaks to be relatively consistent around 0.853 seconds.

12. State the name of the Matlab program that performs the calculations for Heart Rate.

The Matlab file *ECG_heart_rate_0501.m* calculates the heart rate.

General

13. Use your processed ECG signal to discuss any relevant, distinguishing features. Think about this in terms of information around Figure 2 in the Experiment Instructions.

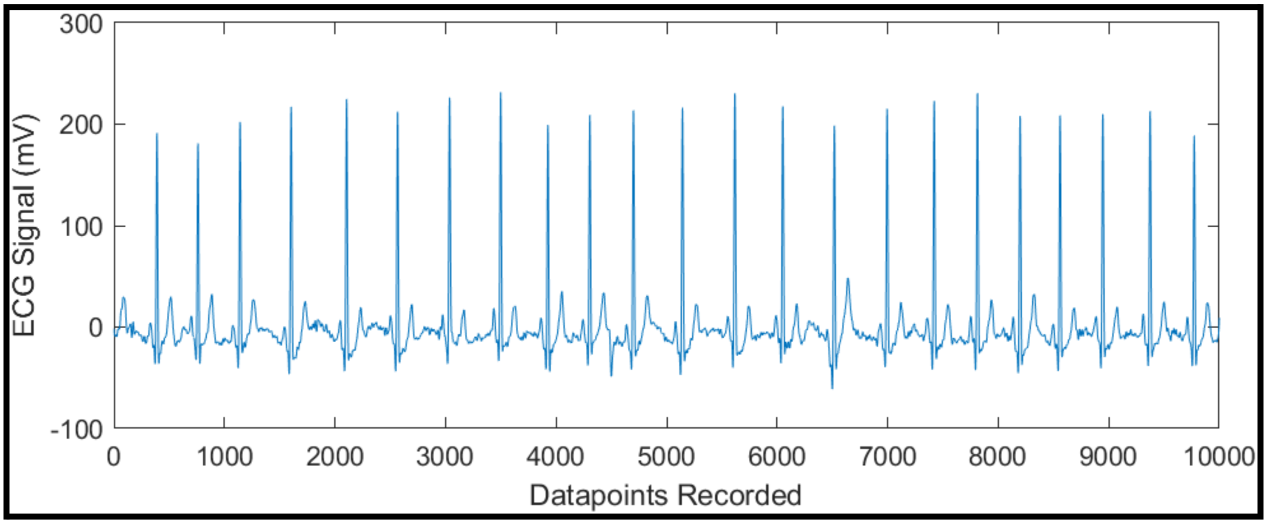

Figure 4: Filtered and Baseline Drift Corrected ECG Signal plotted against 10000 total data points

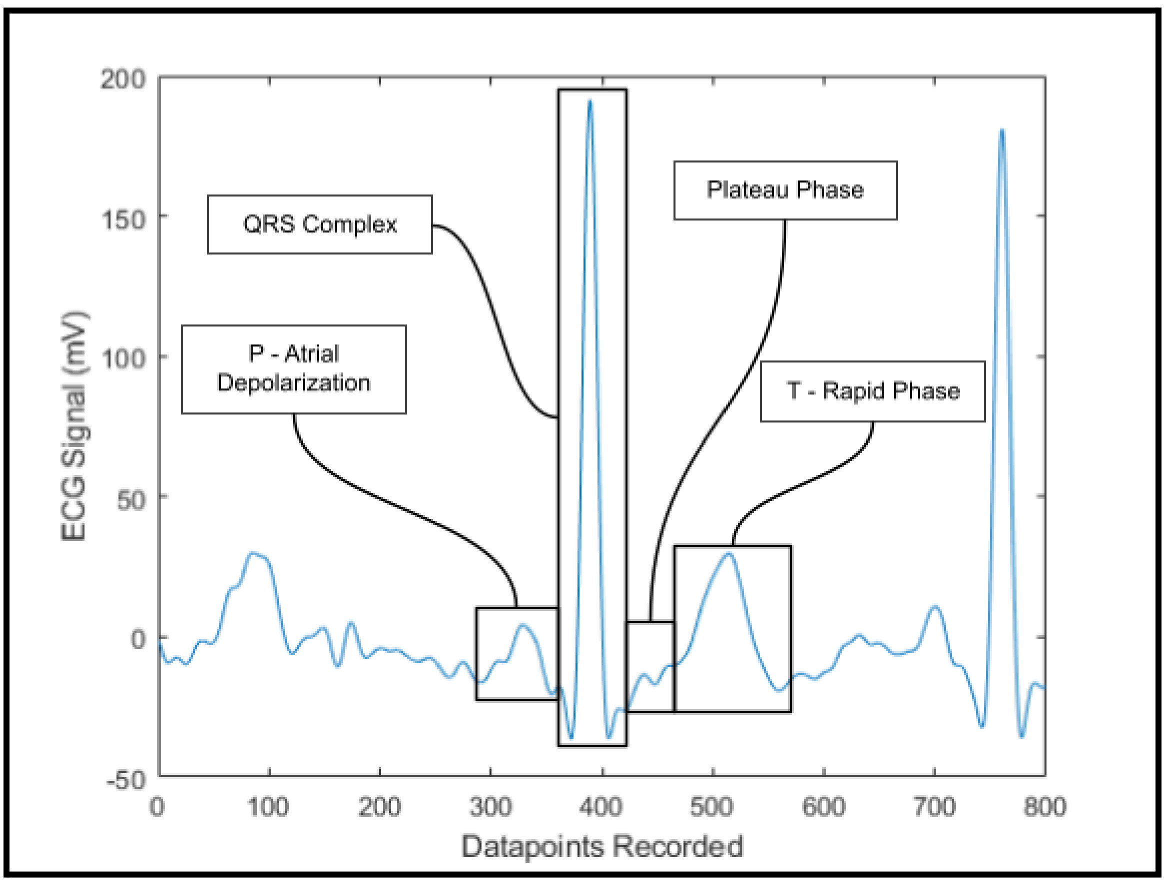

Figure 5: Sample of Figure 1's Filtered and Baseline Drift Corrected ECG Signal up to datapoint 2000

Figures 2 and 3 show the QRS cycles following the raw data being filtered and after having its baseline drift corrected. From this, we are able to distinguish the individual components of the QRS cycle, as labeled on one cycle in Figure 3.

We can see in Figure 3 all components of the generic cardiac cycle. The P, QRS, and T cycles are clear and distinct from each other. While there are slight variances between the amplitudes of certain cardiac cycle components across the whole dataset, these appear to be within normal fluctuations seen when recording ECG data. Similarly, while plateaus between the T and P phase are not perfect, they are able to be distinguished from the T and P complexes at their start and end.

14. Discuss any problems encountered using the signal processing to filter, correct, or calculate your values in previous questions.

One issue we encountered was with using the lowpass function in Matlab. Depending on the sampling frequency used, the maximum value of the R peaks would change. However, this may simply be a byproduct of the effect that varied sampling frequencies have on the resolution of a processed signal.

Extra Credit

16. Write Matlab code to plot the ECG signal with R peaks labeled on the plots. The output should be a well-formatted signal vs time plot that has the peaks labeled. You must include the results and the Matlab code for full credit.

The matlab file that labels the R peaks is *ECG_Rpeak_label_0501.m* and is also copied here.

```
%% Labeled R Peaks
figure(4);
plot(ecgwavelet);
hold on;
PeakValues = Rwaves(:,1);
PeakValues = round(PeakValues,2);
values = string(PeakValues);
for i=1:1:length(Rwaves(:,1))
    plot(Rwaves(i,2),Rwaves(i,1),'o','MarkerSize',7);
    text(Rwaves(i,2),Rwaves(i,1),values(i));
    hold on;
end
xlabel('Datapoints Recorded');
ylabel('ECG Signal (mV)');
```

The labeled R peaks can be seen in the following generated figure

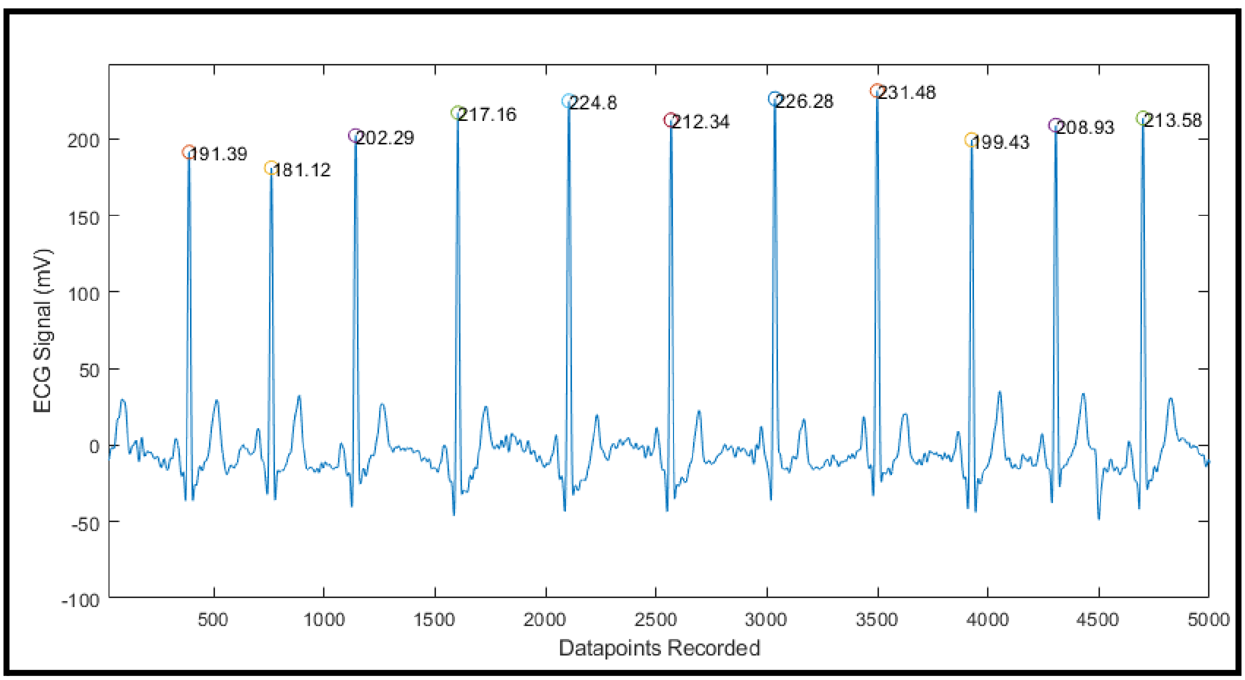

Figure 6: Filtered and baseline drift corrected signal with peaks identified and labeled with amplitude values. This shows only the first half of the data.