EECE 2150 – Circuits and Signals: Biomedical Applications

Lab 16 – Digital Signal Manipulation of the ECG Signal

Zachary Ruhlen – ruhlen.z@northeastern.edu

December 9, 2020

Introduction

The human nervous systems operates with a combination of ionic electric potentials and chemical potentials. The brain sends these signals to the heart, which responds with a heartbeat. The electric potential from these signals can be measured using what's called an electrocardiogram, or ECG for short. There are a few components to a nerve signal, including a large spike up followed by a small dip. Fig. 1 goes into detail on the signal components and the mechanism. The resting potential of a nerve is around -70 mV and its peak potential is around 30 mV. ECG signals generally contain frequencies from 0.1 Hz to 400 Hz.

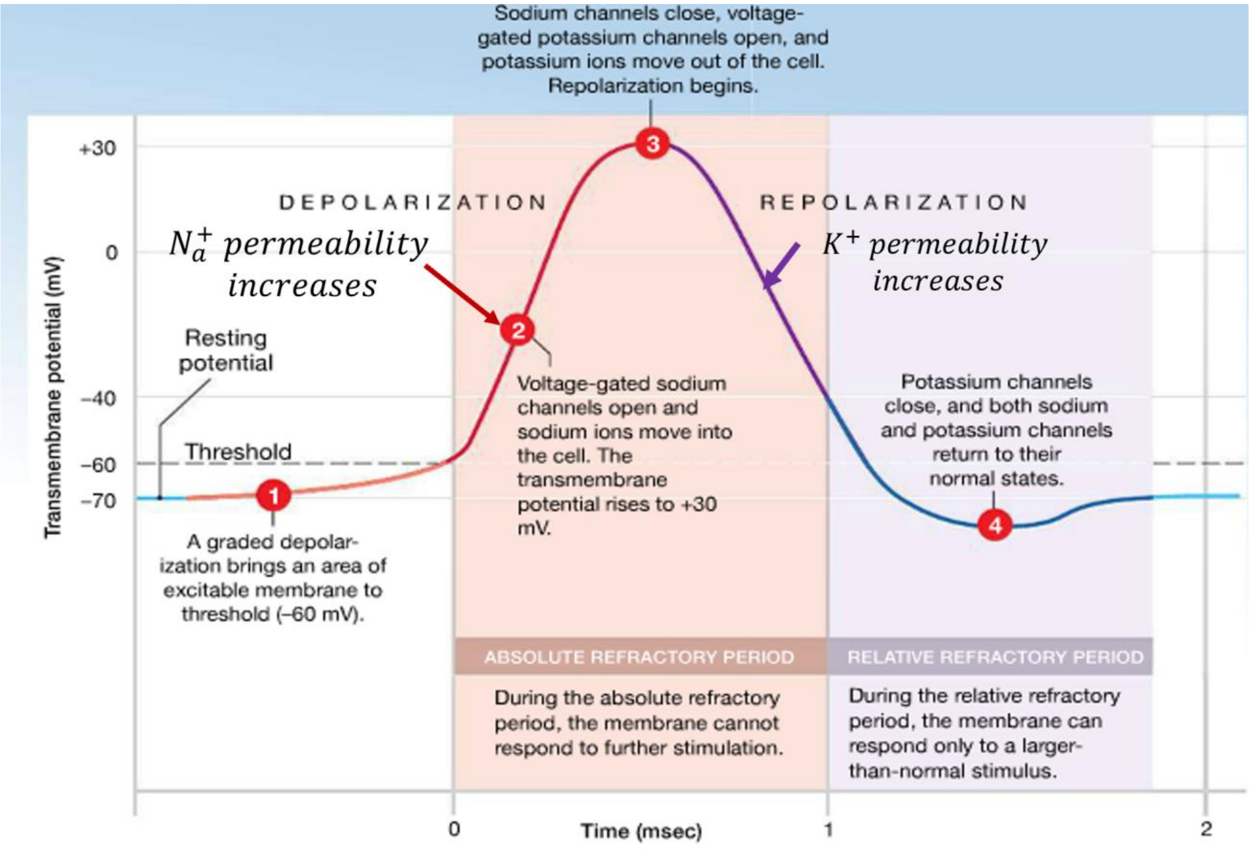
Sodium channels close, voltage-gated potassium channels open, and potassium ions move out of the cell. Repolarization begins.
+30
0
−40
−60
−70
Transmembrane potential (mV)
DEPOLARIZATION
REPOLARIZATION
N_a^+ permeability increases
K^+ permeability increases
Resting potential
Threshold
1
2
3
4
A graded depolarization brings an area of excitable membrane to threshold (−60 mV).
Voltage-gated sodium channels open and sodium ions move into the cell. The transmembrane potential rises to +30 mV.
Potassium channels close, and both sodium and potassium channels return to their normal states.
ABSOLUTE REFRACTORY PERIOD
During the absolute refractory period, the membrane cannot respond to further stimulation.
RELATIVE REFRACTORY PERIOD
During the relative refractory period, the membrane can respond only to a larger-than-normal stimulus.
0
1
2
Time (msec)

Fig. 1 - Components of a nerve electrical signal

A previous lab used the circuit shown in Fig. 2 to collect an ECG sample 3.2 seconds long. The ECG sample is shown in Fig. 3.

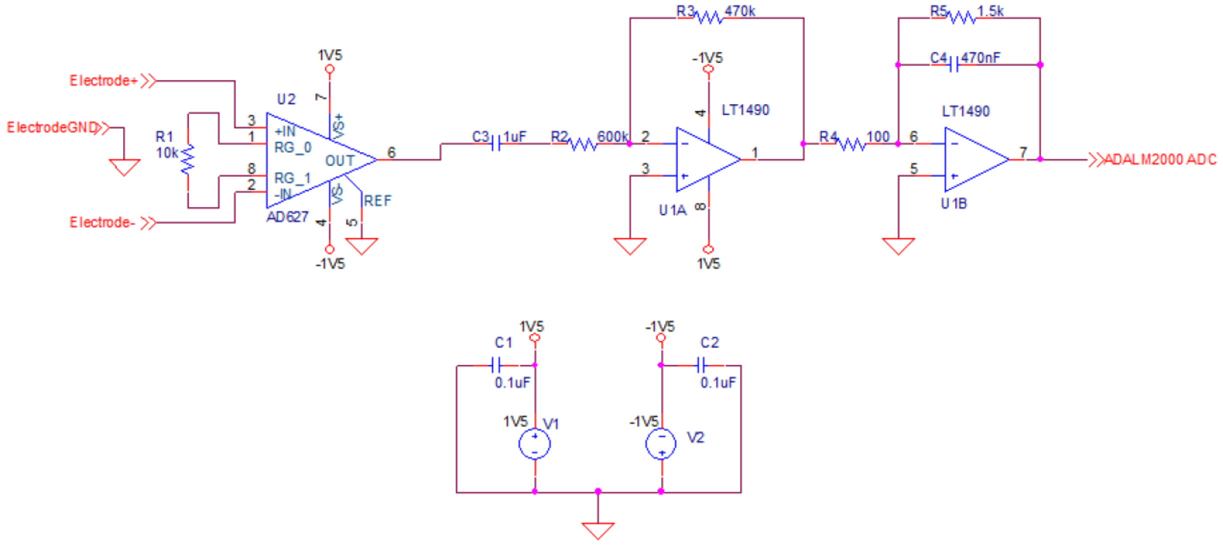

Fig. 2 - Circuit used to capture ECG

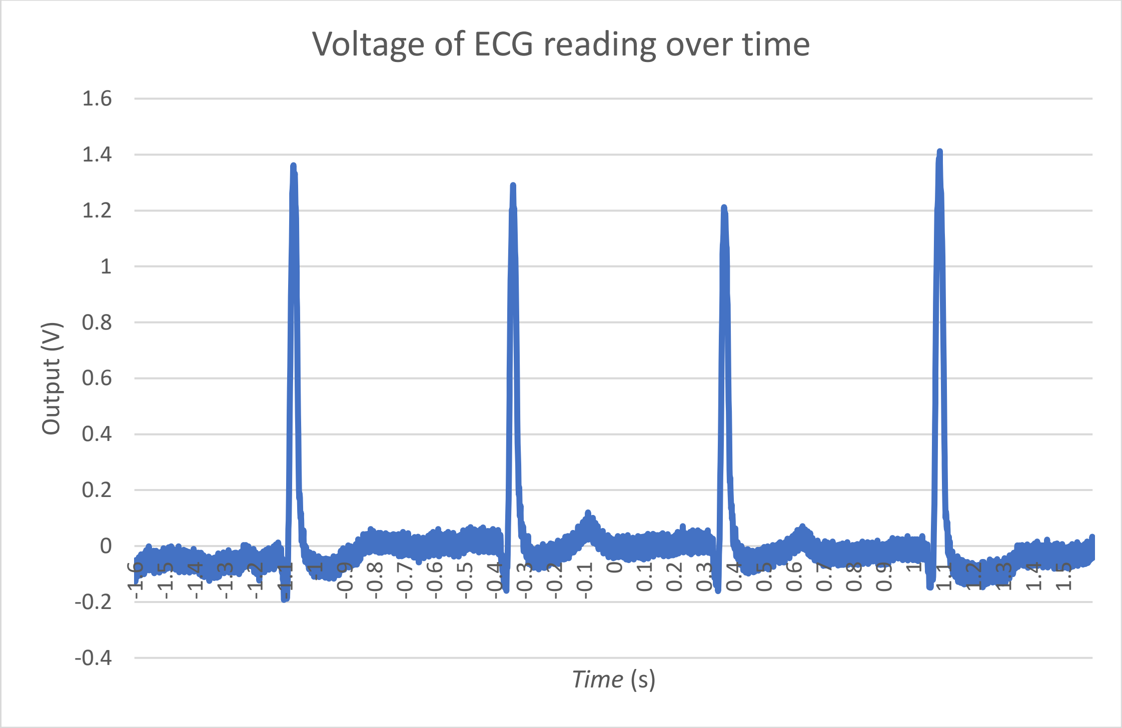

Fig. 3 - Captured ECG

The digital function firpmord() in MATLAB R2020a will be used to perform a Parks-McLellan filter on the trace to apply digital filtering at the same cutoff frequencies of the analog filters. It will also be used to “notch” out 60Hz noise from the power grid. These steps will produce cleaner ECG signals that are easier to read. MATLAB will also be used to calculate the heartrate from the ECG trace.

Results

Part 1

The first part of this lab applies the Parks-McLellan filter to the ECG sample. The firpmord() function is the first step in doing this. Break frequencies of 0.26 and 226 were provided to mirror the cutoff frequencies from the analog filters. The amplitude was provided as [1 0] to keep the maximum normalized amplitude of 1. The attenuation in the passband is 3dB and in the stopband 40 dB. The freqz() function was used to verify this design. This is shown in Fig. 4. Notice the ripples; this is why it's called an equiripple design. The phase jumps because of this rippling action. The sampling rate of the ECG is 1000 and the tolerances were calculated as in the documentation for the function. The filter function was used, passing in the return values from firpmord(), the original voltages, and an amplitude of 1, to complete the filtering. The result, compared to the original trace, is shown in Fig. 5.

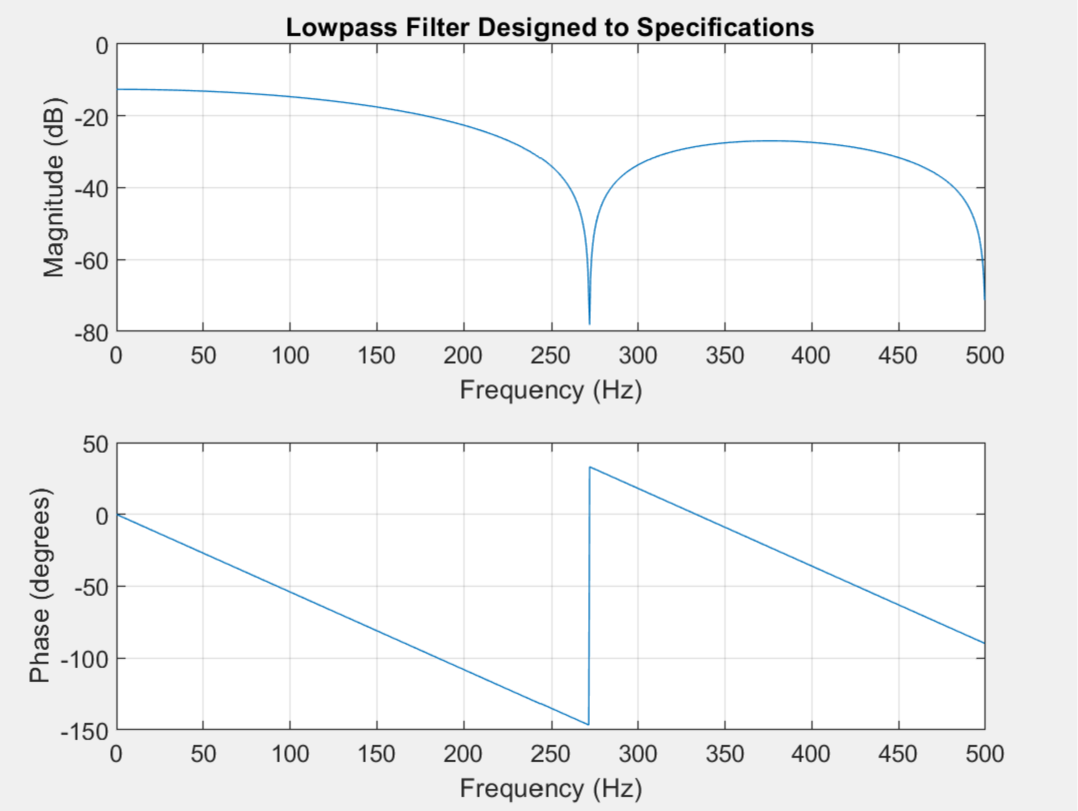

Fig. 4 - freqz() verification of the firpmord() filter

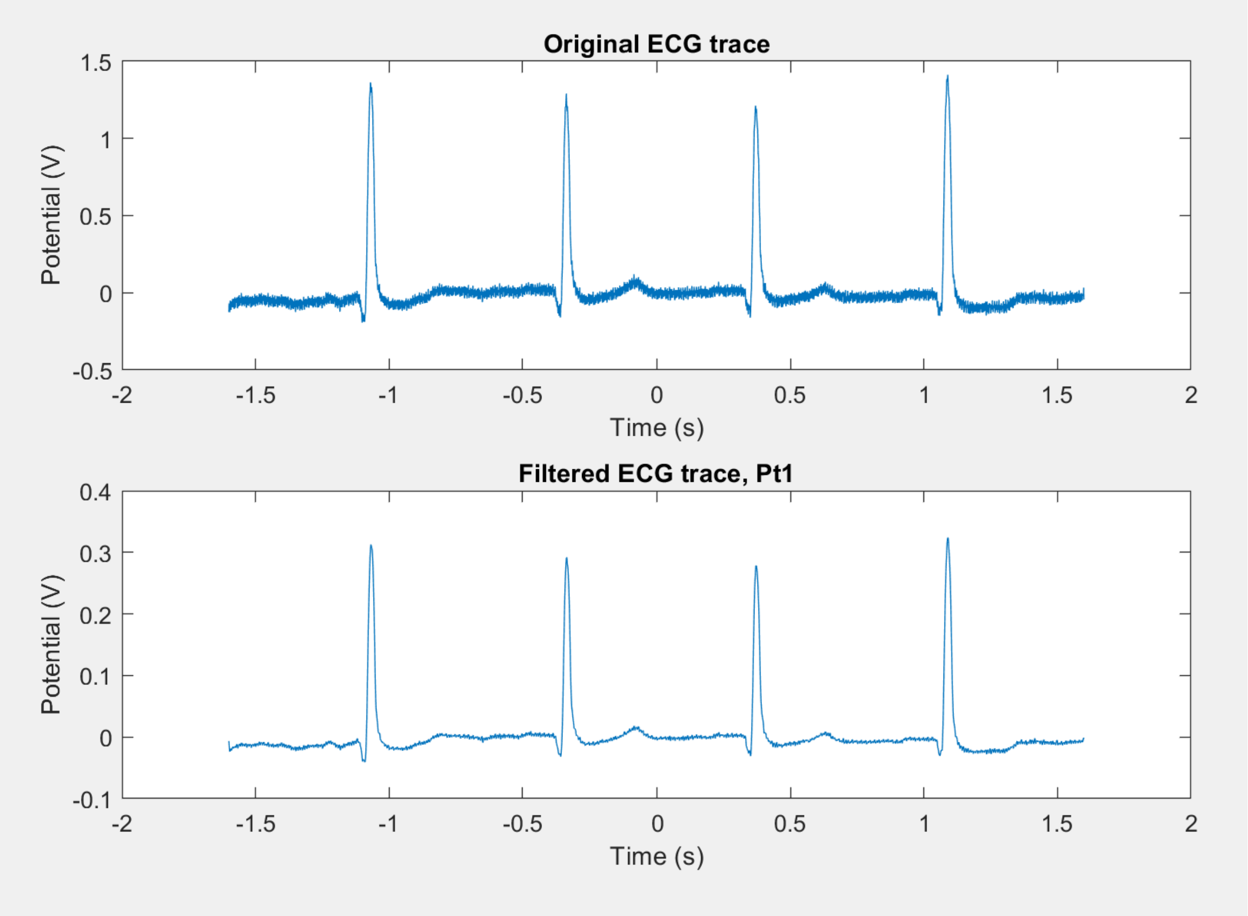

Fig. 5 - Top: Original ECG trace Bottom: ECG trace filtered, passband 0.26-226 Hz

Part 2

Next, the same function was used to create a notch at 60 Hz to remove power grid interference. A low pass filter with a passband of 0-59 Hz and stopband attenuation of 500 dB was created. A high pass filter with a passband of 61-500 Hz was created with a stopband attenuation of 500 dB. The original voltage curve was filtered through both of these and the results were added. Finally, the filter from Part 1 was reapplied. It was plotted alongside the other two traces. This is shown in Fig. 4. This did not make much of a difference, but the signal was not very noisy to begin with after the filtering from part 1.

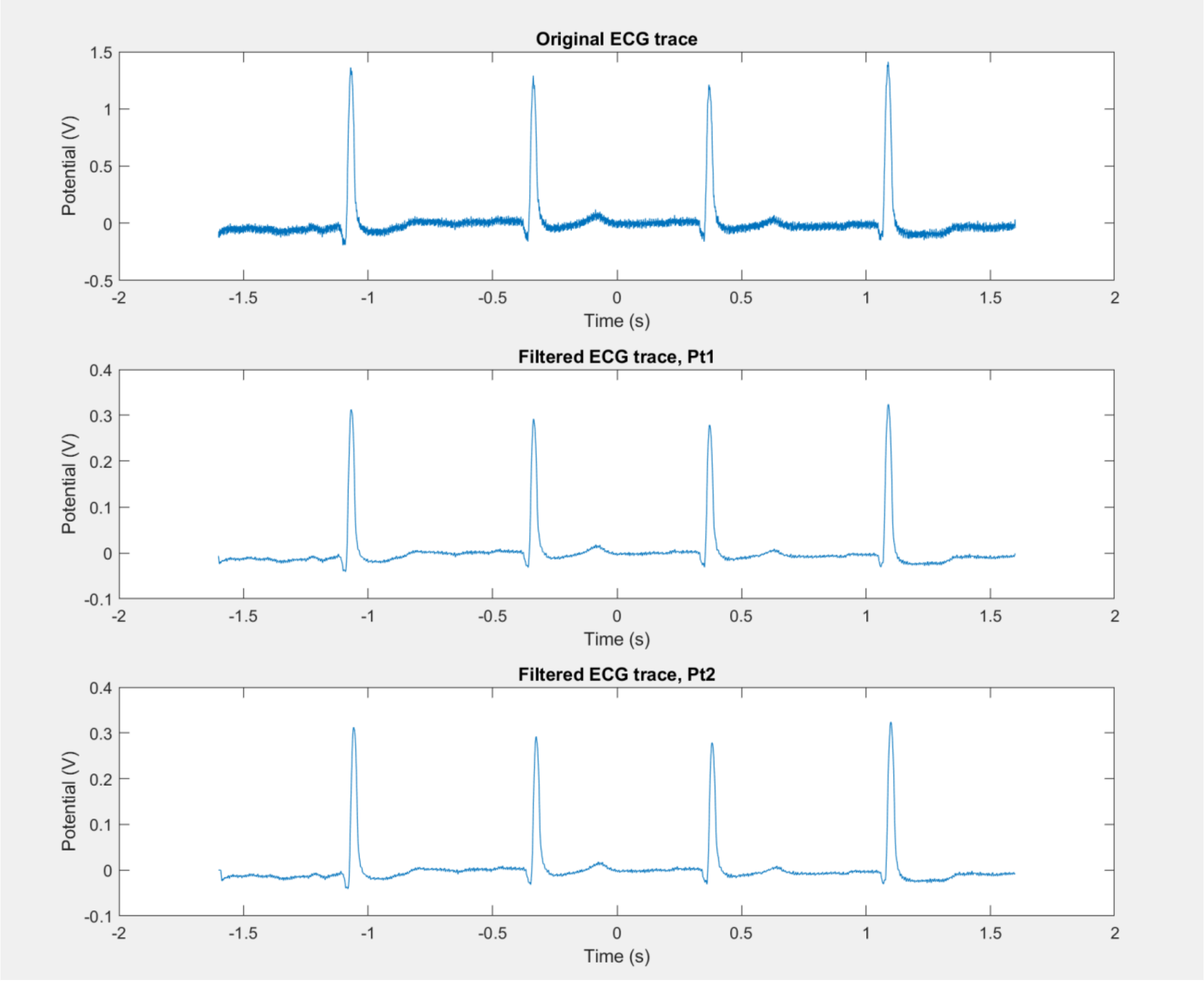

Fig. 6 - Top: Original Middle: Passband 0.26-226Hz Bottom: Notch of 60Hz, passband 0.26-226

Part 3

Finally, MATLAB was used to calculate the heartrate from the ECG signal. Heartrate can be calculated with the following formula: $HR = \frac{60\frac{seconds}{minute}}{t\ seconds} * \#beats$. $t = \#samples/F_s$, where F_s is the sampling rate in samples/sec. A textbox with the calculation is shown in Fig. 7 alongside the other three plots. It is important to note that accuracy using this method is low and a longer ECG trace would result in a more accurate calculation.

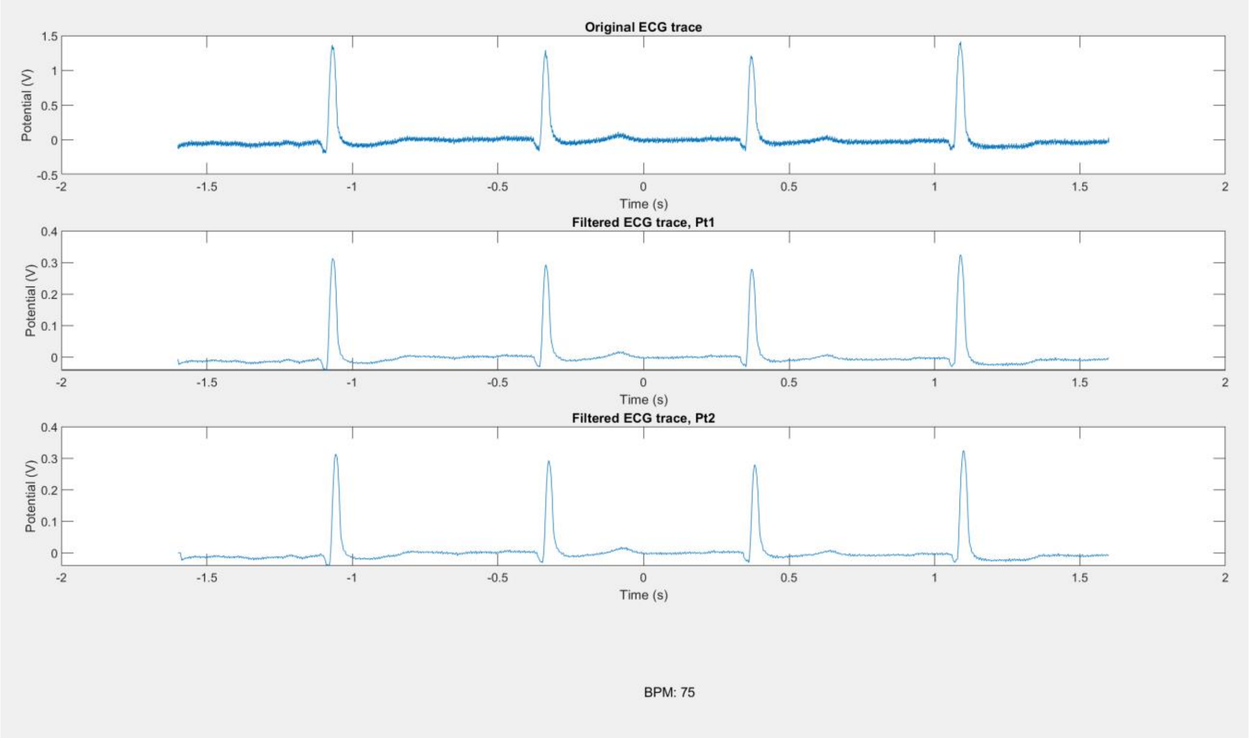

Fig. 7 - Top: Original Middle Top: Passband 0.26-226Hz Middle Bottom: Notch of 60Hz, passband 0.26-226 Bottom: BPM calculated from ECG trace

Conclusion

This lab was successful in applying digital filtering to the original ECG trace. All four components of a nerve signal are very clear in the filtered traces and a reasonable heartrate was calculated at 75 bpm.

Appendix – MATLAB Code

```
clear;
clf;
%%load values from csv
T=readtable('ecg.csv');
time = T.time;
v=T.V2;

%%plot original ecg curve
subplot(4, 1, 1)
plot(time, v)
xlabel('Time (s)')
ylabel('Potential (V)')
title('Original ECG trace')

%%filter high and low frequencies and plot
rp = 3;           % Passband ripple in dB
rs = 40;          % Stopband ripple in dB
fs = 1000;        % Sampling frequency
f = [0.26 226];    % Cutoff frequencies
a = [1 0];        % Desired amplitudes
dev = [(10^(rp/20)-1)/(10^(rp/20)+1) 10^(-rs/20)];
[n,fo,ao,w] = firpmord(f,a,dev,fs);
b = firpm(n,fo,ao,w);
vfilt=filter(b,1,v);

subplot(4, 1, 2)
plot(time, vfilt)
xlabel('Time (s)')
ylabel('Potential (V)')
title('Filtered ECG trace, Pt1')

%%notch 60hz
%below 60
rp = 3;           % Passband ripple in dB
rs = 500;          % Stopband ripple in dB
fs = 1000;        % Sampling frequency
f = [0 59];    % Cutoff frequencies
a = [1 0];        % Desired amplitudes
dev = [(10^(rp/20)-1)/(10^(rp/20)+1) 10^(-rs/20)];
[n,fo,ao,w] = firpmord(f,a,dev,fs);
b = firpm(n,fo,ao,w);
vfiltbelow60=filter(b,1,v);

rp = 3;           % Passband ripple in dB
rs = 500;          % Stopband ripple in dB
fs = 1000;        % Sampling frequency
f = [61 500];    % Cutoff frequencies
a = [1 0];        % Desired amplitudes
dev = [(10^(rp/20)-1)/(10^(rp/20)+1) 10^(-rs/20)];
[n,fo,ao,w] = firpmord(f,a,dev,fs);
b = firpm(n,fo,ao,w);
vfiltabove60=filter(b,1,v);

vfilt=vfiltbelow60+vfiltabove60;
```

```
rp = 3;           % Passband ripple in dB
rs = 40;          % Stopband ripple in dB
fs = 1000;        % Sampling frequency
f = [0.26 226];    % Cutoff frequencies
a = [1 0];        % Desired amplitudes
dev = [(10^(rp/20)-1)/(10^(rp/20)+1) 10^(-rs/20)];
[n,fo,ao,w] = firpmord(f,a,dev,fs);
b = firpm(n,fo,ao,w);
vfilt2=filter(b,1,vfilt);

subplot(4, 1, 3)
plot(time, vfilt2)
xlabel('Time (s)')
ylabel('Potential (V)')
title('Filtered ECG trace, Pt2')

%%calculate hr
beats=findpeaks(v,fs,'MinPeakProminence',1); %beats (prominence of 1 V)
time=(size(v)/fs);
time=time(1);
samplespermin=60/time;
bpm=size(beats)*samplespermin;
bpm=bpm(1);
subplot(4, 1, 4);
textout=strcat("BPM: ", num2str(bpm));
text(0.5,0.5,textout); axis off
```