EEE 491 - Biomedical Engineering

Final Project

Ahmed Mohamed Roshdi	18010212
Taha Ahmed Taha	19015866

1 Introduction

Electrocardiography (ECG) is a non-invasive medical test used to detect and diagnose cardiovascular diseases. It measures the electrical activity of the heart and produces a graph, called an electrocardiogram, that shows the heart's rhythm and any abnormalities in its function. ECG is a vital tool for medical professionals in diagnosing and monitoring heart conditions.

However, ECG signals are often contaminated with noise and interference, making it difficult to analyze them accurately. In this report, we will explore different signal processing techniques to enhance the quality of ECG signals and extract useful information from them. Specifically, we will focus on removing 50 Hz noise using a notch filter, optimizing the signal-to-noise ratio by limiting the bandwidth, and detecting the heart rate using autocorrelation. We will apply these techniques to both a noisy ECG signal and a processed signal and compare the results.

The main objective of this report is to demonstrate how signal processing techniques can improve the quality of ECG signals and facilitate the accurate detection of heart rate. By achieving this objective, we hope to contribute to the ongoing efforts to enhance the effectiveness and reliability of ECG as a diagnostic tool for cardiovascular diseases.

2 Procedure

2.1 Removing signals from muscle movement

Plot the signal:

```
% Section 1 — Plot the original signal and remove the muscles noise
% load the signal
EKG1 = [struct2cell(load('ecg.mat')){:}];
fs = 500; % sampling frequency
time = (0:length(EKG1)-1)/fs; % time vector
figure
plot(time, EKG1);
set(gca, "fontsize", 24) % increase the fontsize of the plot
xlabel("Time (s)");
ylabel("Voltage (V)");
```

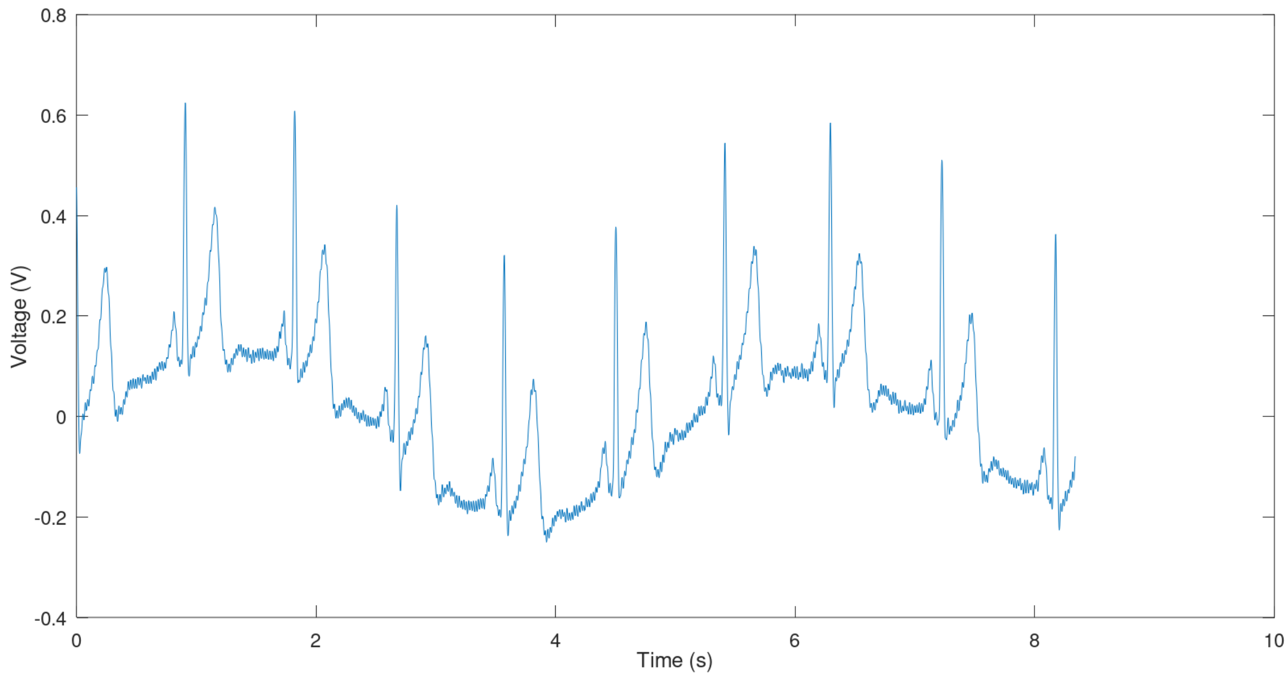


Figure 1: Original Signal

Zoom on one period:

```
t_start = 0.7; % start time for zoomed in plot
t_end = 1.5; % end time for zoomed in plot
idx_start = round(t_start*500); % index corresponding to start time
idx_end = round(t_end*500); % index corresponding to end time
figure
plot(time(idx_start:idx_end), EKG1(idx_start:idx_end), "linewidth", 2);
set(gca, "linewidth", 2, "fontsize", 24);
xlabel("Time (s)");
ylabel("Voltage (V)");
```

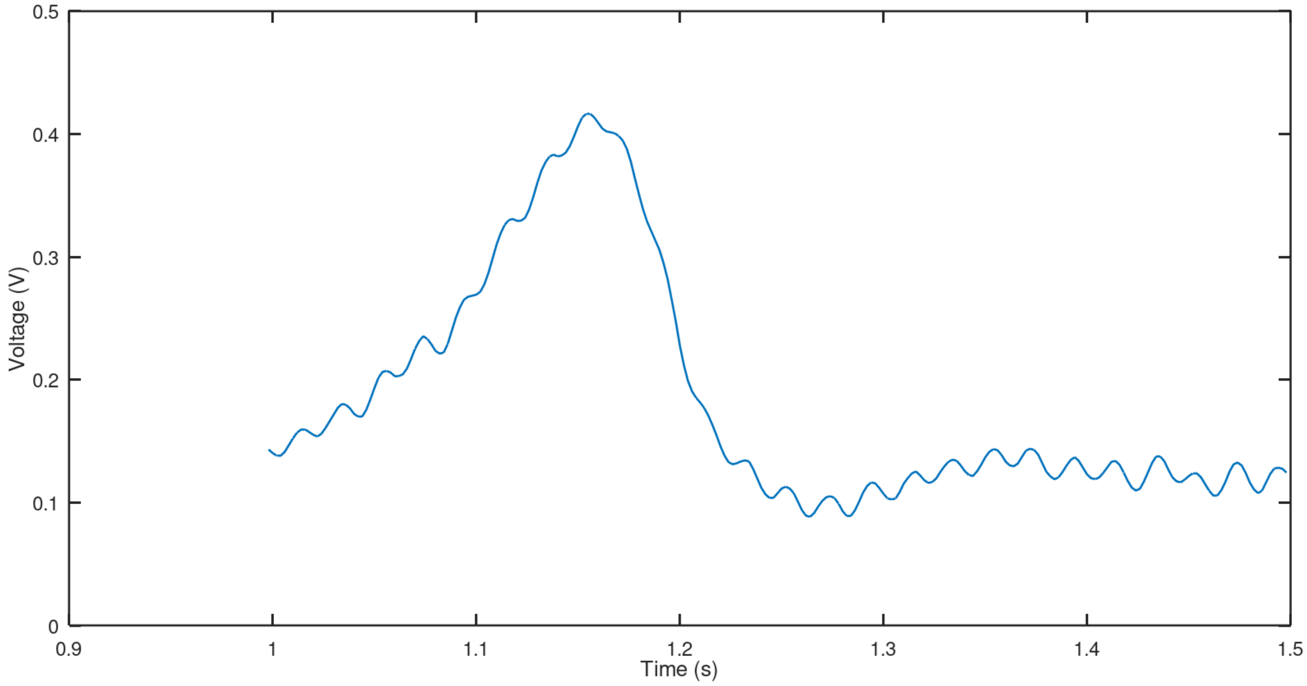


Figure 2: One interval of the signal, from `t=1.5` to `t=1.52`

Remove noise generated by muscles:

```
% remove the muscles noise
freq = (0:length(EKG1)-1)/length(EKG1)*500; % frequency vector
ecg_fft = fft(EKG1); % Fourier transform of signal
ecg_fft(freq < 0.5) = 0; % set frequencies below 0.5 Hz to zero
ecg1 = real(ifft(ecg_fft)); % inverse Fourier transform of filtered signal
figure
plot(time, EKG1, 'b', time, ecg1, 'r');
set(gca, "linewidth", 2, "fontsize", 24);
xlabel("Time (s)");
ylabel("Voltage (V)");
legend("Original signal", "Filtered signal");
```

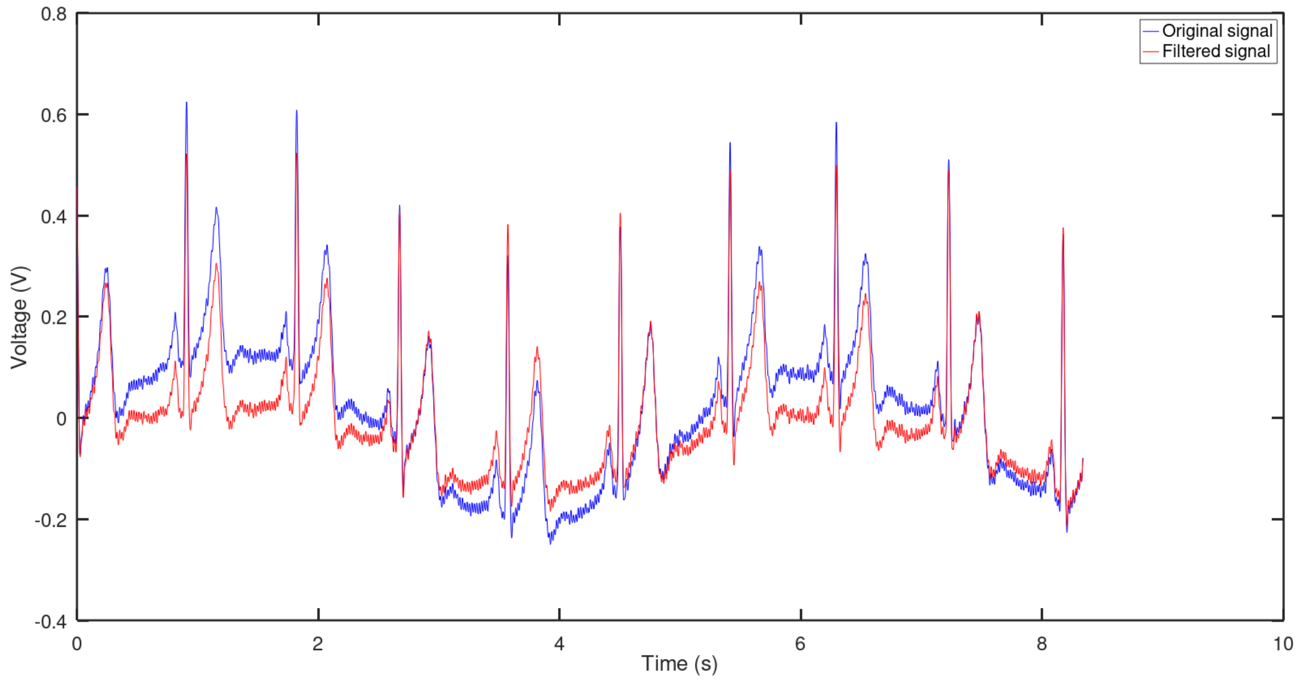


Figure 3: Filtered signal vs original signal

Make sure that your filter has a real impulse response by making an inverse Fourier transform of the transfer function.

```
filter_fft = ones(size(EKG1)); % initialize filter to all ones
filter_fft(freq < 0.5) = 0; % set frequencies below 0.5 Hz to zero
filter_ifft = ifft(ifftshift(filter_fft)); % inverse Fourier transform of filter
t = (-length(filter_ifft)/2:length(filter_ifft)/2-1)/fs; % create time vector
figure
plot(t, real(fftshift(filter_ifft)), "linewidth", 2); % center filter in time
    domain before plotting
set(gca, "linewidth", 2, "fontsize", 24);
xlabel("Time (s)");
ylabel("Amplitude");
```

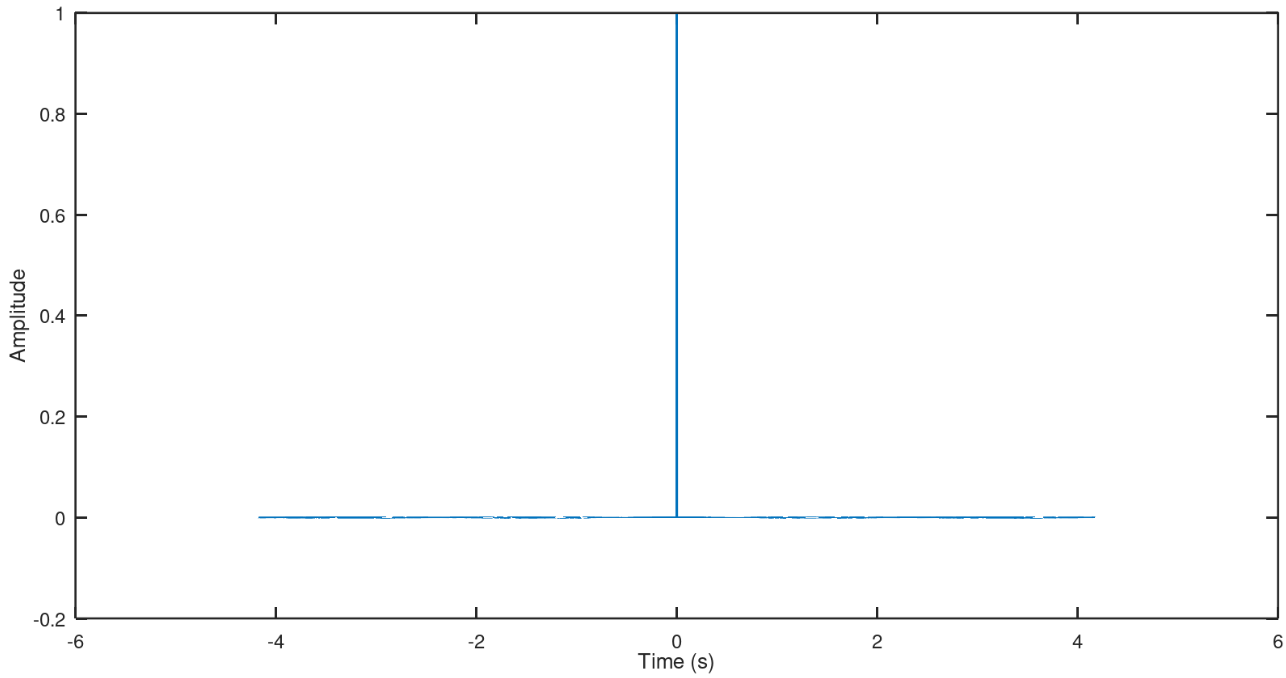


Figure 4: real impulse response

2.2 Removing 50 Hz interference:

Design a notch filter with a zero at 50 Hz. We will design it with butterworth filter

```
% Section 2 - remove the muscles noise with a notch filter
pkg load signal % I am using octave   \_(   )_/
fs = 500; % sampling frequency
f0 = 50; % notch frequency
bw = 1/(fs/2); % normalized notch bandwidth
Q = f0/bw; % quality factor
wo = f0/(fs/2); % normalized notch frequency
[b,a] = butter(2, [wo-bw/2, wo+bw/2], 'stop'); % design notch filter coefficients
ecg2 = filter(b, a, ecg1); % apply notch filter using filtfilt
figure
plot(time, ecg1, 'b', time, ecg2, 'r');
set(gca, "linewidth", 2, "fontsize", 24);
xlabel("Time (s)");
ylabel("Voltage (V)");
legend("Original signal", "Filtered signal");
```

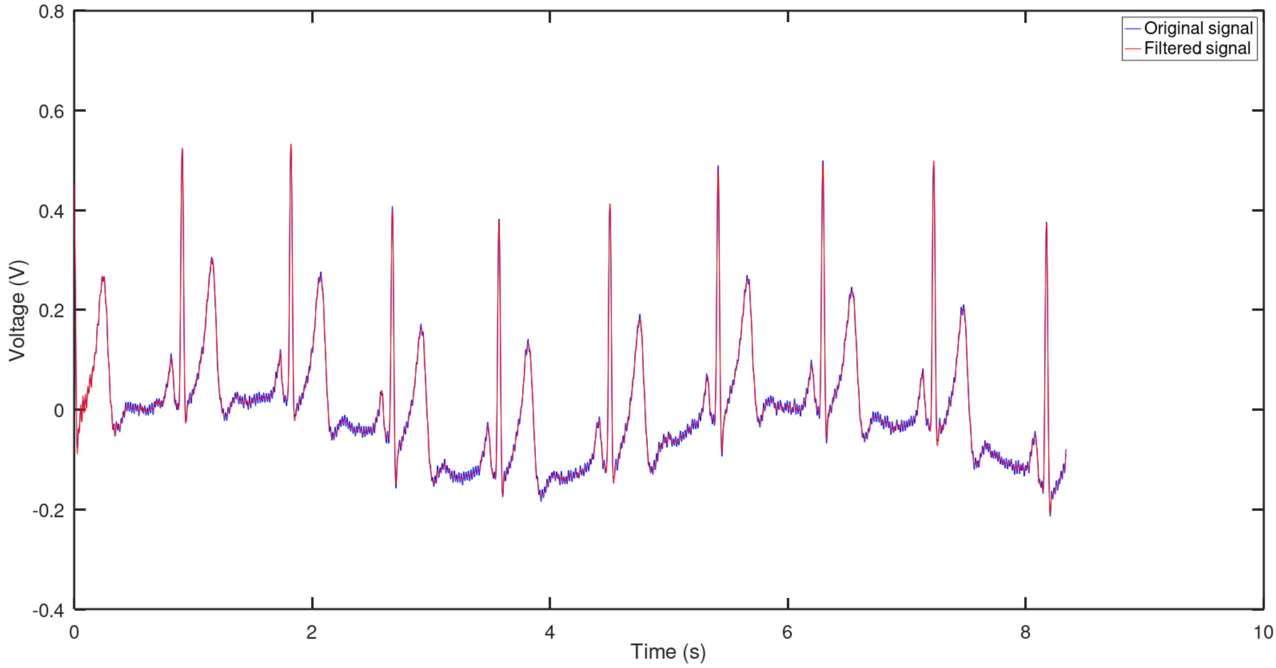


Figure 5: Apply notch filter to remove 50Hz noize

Zoom in with your mouse to see the smoothed signal

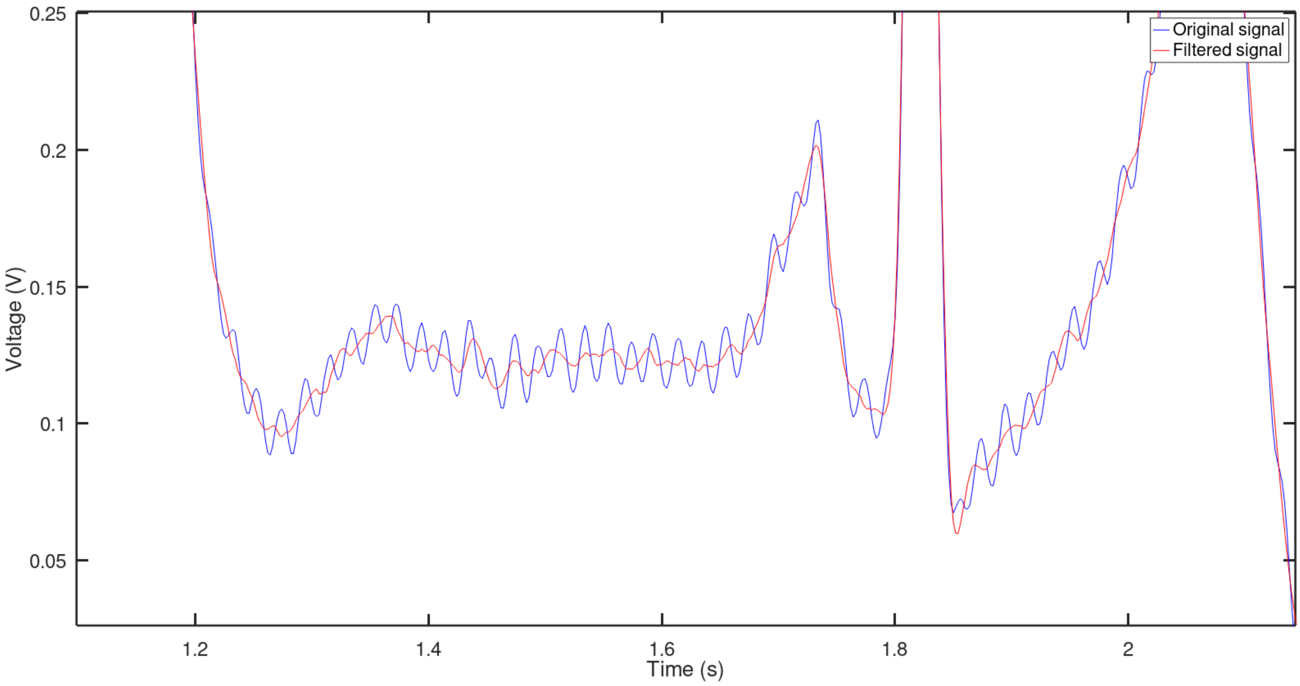


Figure 6: A little zoom

2.3 Increasing The SNR

```
% Section 3 — Increasing the signal—to—noise ratio (Low pass filter):
% Try different cut—off frequencies and see how the signal changes
fs = 500; % sampling frequency in Hz
order = 2; % filter order
fc = [5 20 100]; % cut—off frequencies in Hz
```

```
colors = ['r', 'g', 'y'];
figure;
hold on;
plot(time, ecg2, 'b', "linewidth", 2); % plot original signal
set(gca, "linewidth", 2, "fontsize", 24);
xlabel("Time (s)");
ylabel("Voltage (V)");
legend("Original signal");

% plot filtered signals
for i = 1:length(fc)
    [b,a] = butter(order, fc(i)/(fs/2), 'low');
    ecg_filt = filter(b, a, ecg2);
    plot(time, ecg_filt, colors(i));
    set(gca, "linewidth", 2, "fontsize", 24);
end

title("Filtered Signals with Different Cut-off Frequencies");
legend("Original signal", "5", "10", "100");
hold off;
```

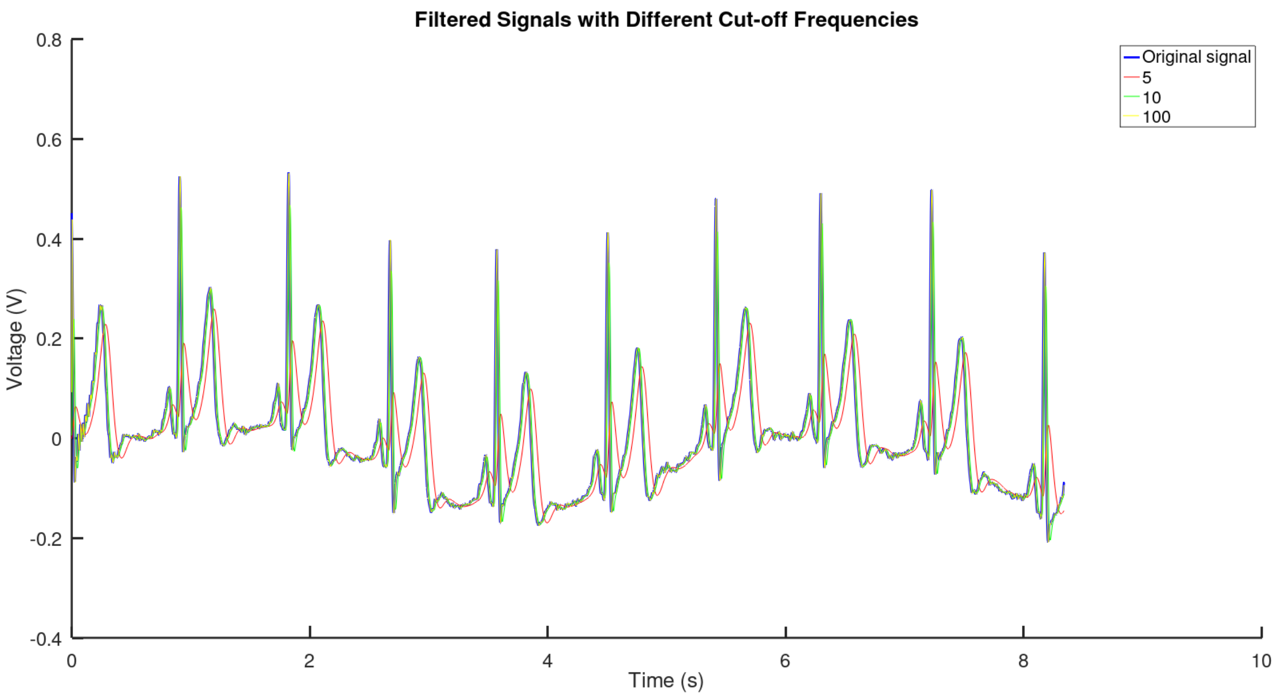


Figure 7: Try to increase the SNR value by applying low pass filter with different cutoff frequencies

Zoom in with your mouse to see the smoothed signal

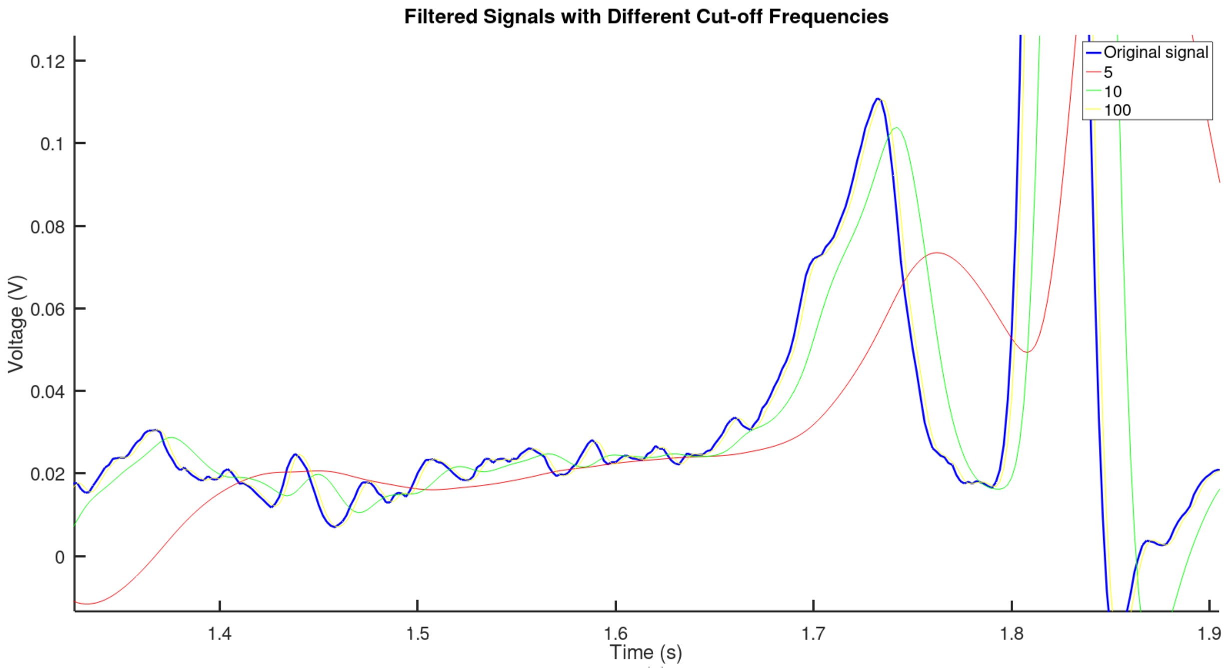


Figure 8: A little zoom

2.4 Finding the heart rate using autocorrelation:

```
% Section 4 — Finding the heart rate using autocorrelation:
% The autocorrelation sequence of a periodic signal has the same cyclic
    characteristics as the signal itself.
% Thus, autocorrelation can help verify the presence of cycles and determine their
    durations.
[b, a] = butter(5, 40/(fs/2), 'low');
ecg3 = filter(b, a, ecg2);
% plot the filtered signal
figure
plot(time, ecg3, 'b');
set(gca, "linewidth", 2, "fontsize", 24);
xlabel("Time (s)");
ylabel("Voltage (V)");
legend("Filtered signal");
% get the autocorrelation of the filtered signal
[autocor,lags] = xcorr(ecg3,'coeff');
% plot the autocorrelation
figure
plot(lags/fs,autocor, "linewidth", 2)
xlabel('Lag')
ylabel('Autocorrelation')
% find the peaks of the autocorrelation
[peaks,locs] = findpeaks(autocor,'MinPeakheight',0.4, "DoubleSided");
```

```

% find the average time between peaks
period = mean(diff(locs)) / fs;
% find the heart rate by dividing 60 seconds by the average period
heart_rate = 60 / period
% plot the peaks on top of the autocorrelation
hold on
pks = plot(lags(locs)/fs,peaks,'or', "linewidth", 2);
set(gca, "linewidth", 2, "fontsize", 24);
title(sprintf('Heart rate: %.1f bpm', heart_rate));
hold off
legend(pks, 'peaks')
```

Looking at the console, we can find that `heart_rate = 65.789`

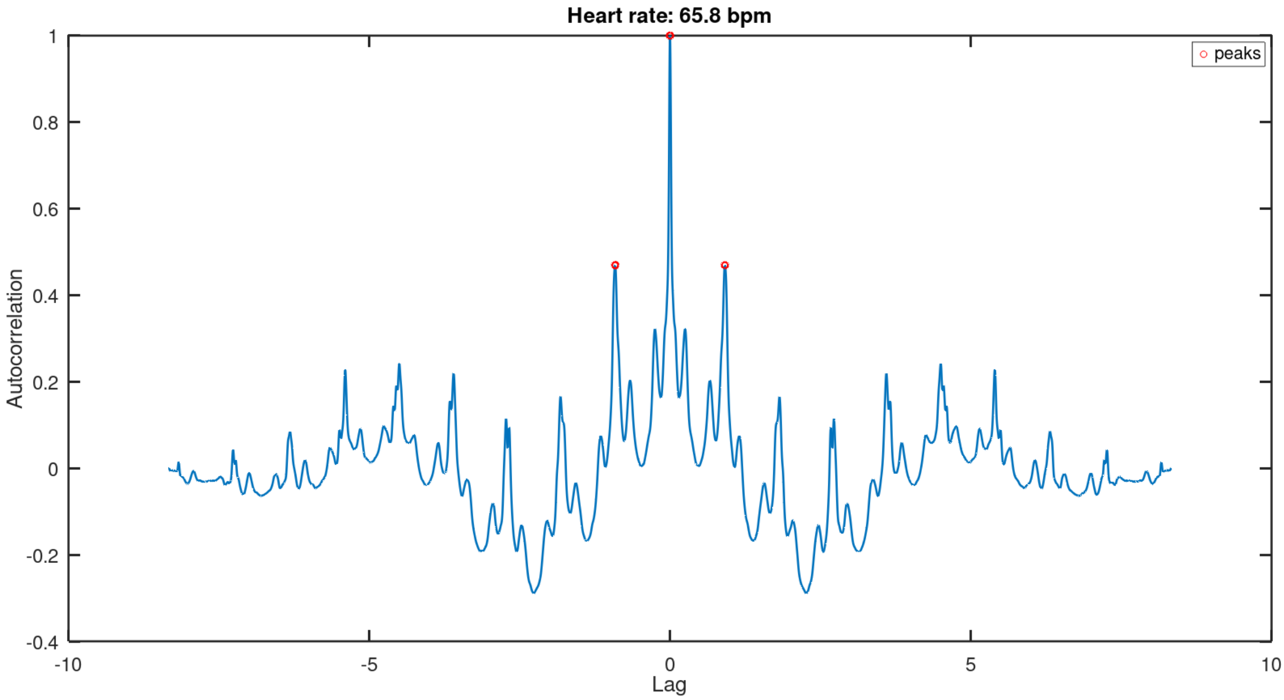


Figure 9: autocorrelation can help verify the presence of cycles and determine their durations.

3 Finding the QRS complex:

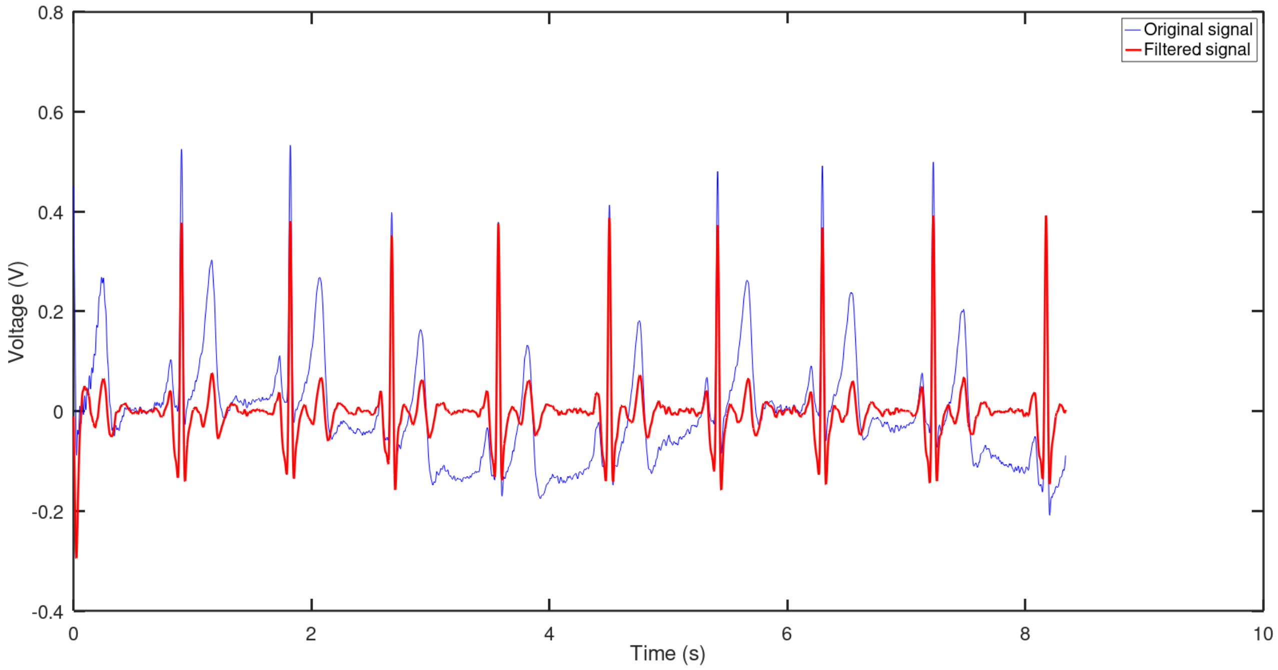
Detects QRS complex in an ECG signal, make the procedure automatic and use it on the processed signal `ecg3` or `ecg2` and on the unprocessed ECG signal `EKG1`.
apply banspass filter to highlight the frequency content of the rapid heart depolarization.

```
% section 5 — Find QRS complex
function [qrs_peaks, ecg_filtered] = detect_qrs_complex(ecg_signal, fs, f1, f2,
    order, threshold_factor)
% Detect QRS complex in an ECG signal using the Pan–Tompkins algorithm
% Inputs:
```

```
%   ecg_signal: the ECG signal to process
%   fs: the sampling frequency of the ECG signal
%   f1: the lower cutoff frequency for the bandpass filter
%   f2: the higher cutoff frequency for the bandpass filter
%   order: the order of the Butterworth filter
%   threshold_factor: the factor to multiply the max amplitude of the filtered
    signal to set the threshold for QRS detection
% Output:
%   qrs_peaks: the indices of the R-peaks in the ECG signal

% Apply bandpass filter to the ECG signal
[b, a] = butter(order, [f1 f2]/(fs/2), 'bandpass');
ecg_filtered = filtfilt(b, a, ecg_signal);

% Set the threshold for QRS detection
threshold = threshold_factor * max(ecg_filtered);

% Find the R-peaks using a simple thresholding method
qrs_peaks = find(ecg_filtered > threshold);

end
```

apply the function to `ecg2` and `EKG1`

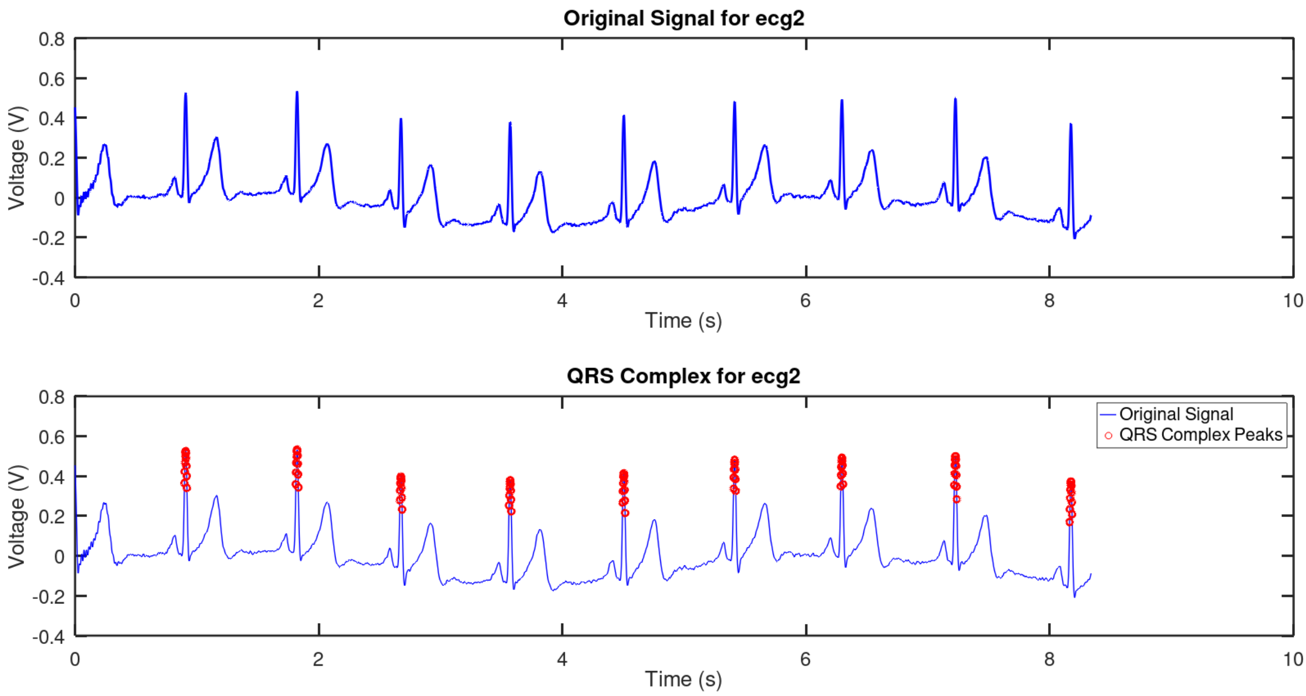
```
f1 = 5;
f2 = 35;
order = 2;
threshold_factor = 0.5;

% Apply QRS detection to ecg2
[qrs_peaks_ecg2, ecg2_filtered]= detect_qrs_complex(ecg2, fs, f1, f2, order,
    threshold_factor);

figure
plot(time, ecg2, 'b', time, ecg2_filtered, 'r', 'linewidth', 2);
set(gca, "linewidth", 2, "fontsize", 24);
xlabel("Time (s)");
ylabel("Voltage (V)");
legend("Original signal", "Filtered signal");
```

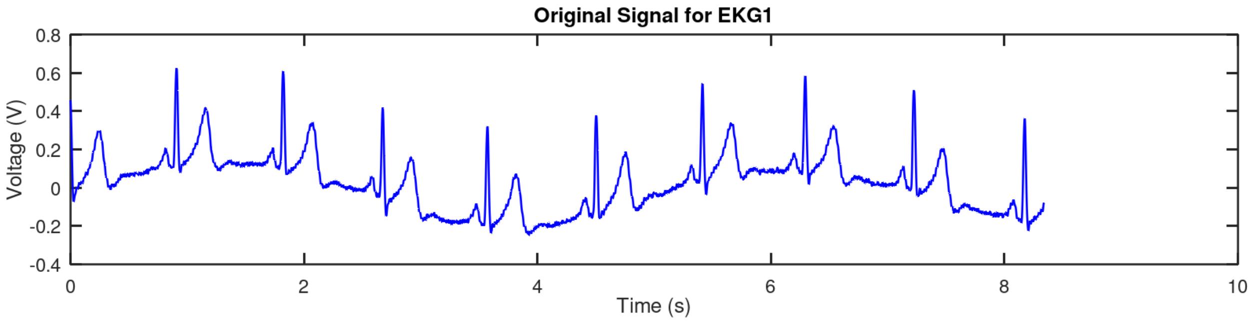


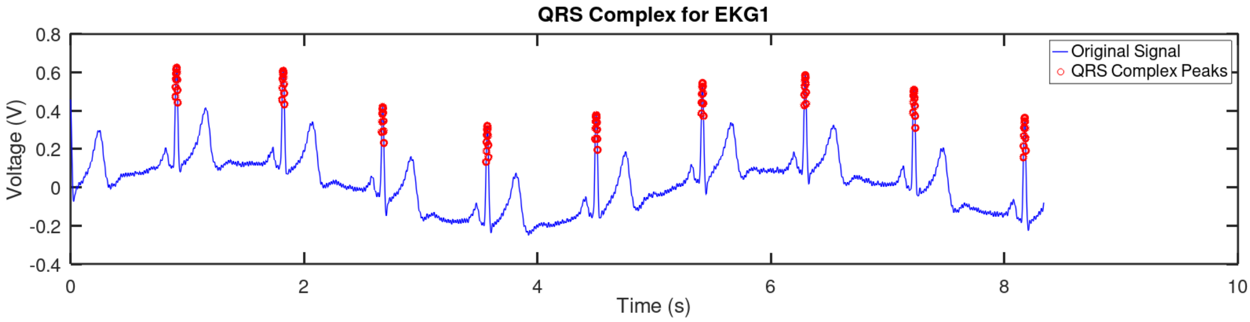
Plot the original signal with the detected QRS complex for both `ecg2` and `EKG1`

```
figure;
subplot(2, 1, 1);
plot(time, ecg2, 'b', 'linewidth', 2);
set(gca, 'linewidth', 2, 'fontsize', 24);
xlabel('Time (s)');
ylabel('Voltage (V)');
title('Original Signal for ecg2');

subplot(2, 1, 2);
plot(time, ecg2, 'b', 'linewidth', 1);
hold on;
plot(time(qrs_peaks_ecg2), ecg2(qrs_peaks_ecg2), 'ro', 'linewidth', 2);
hold off;
set(gca, 'linewidth', 2, 'fontsize', 24);
xlabel('Time (s)');
ylabel('Voltage (V)');
title('QRS Complex for ecg2');
legend('Original Signal', 'QRS Complex Peaks');

% Plot the original signal with the detected QRS complex for EKG1
figure;
subplot(2, 1, 1);
plot(time, EKG1, 'b', 'linewidth', 2);
set(gca, 'linewidth', 2, 'fontsize', 24);
xlabel('Time (s)');
ylabel('Voltage (V)');
```

```
title('Original Signal for EKG1');

subplot(2, 1, 2);
plot(time, EKG1, 'b', 'linewidth', 1);
hold on;
plot(time(qrs_peaks_EKG1), EKG1(qrs_peaks_EKG1), 'ro', 'linewidth', 2);
hold off;
set(gca, 'linewidth', 2, 'fontsize', 24);
xlabel('Time (s)');
ylabel('Voltage (V)');
title('QRS Complex for EKG1');
legend('Original Signal', 'QRS Complex Peaks');
```





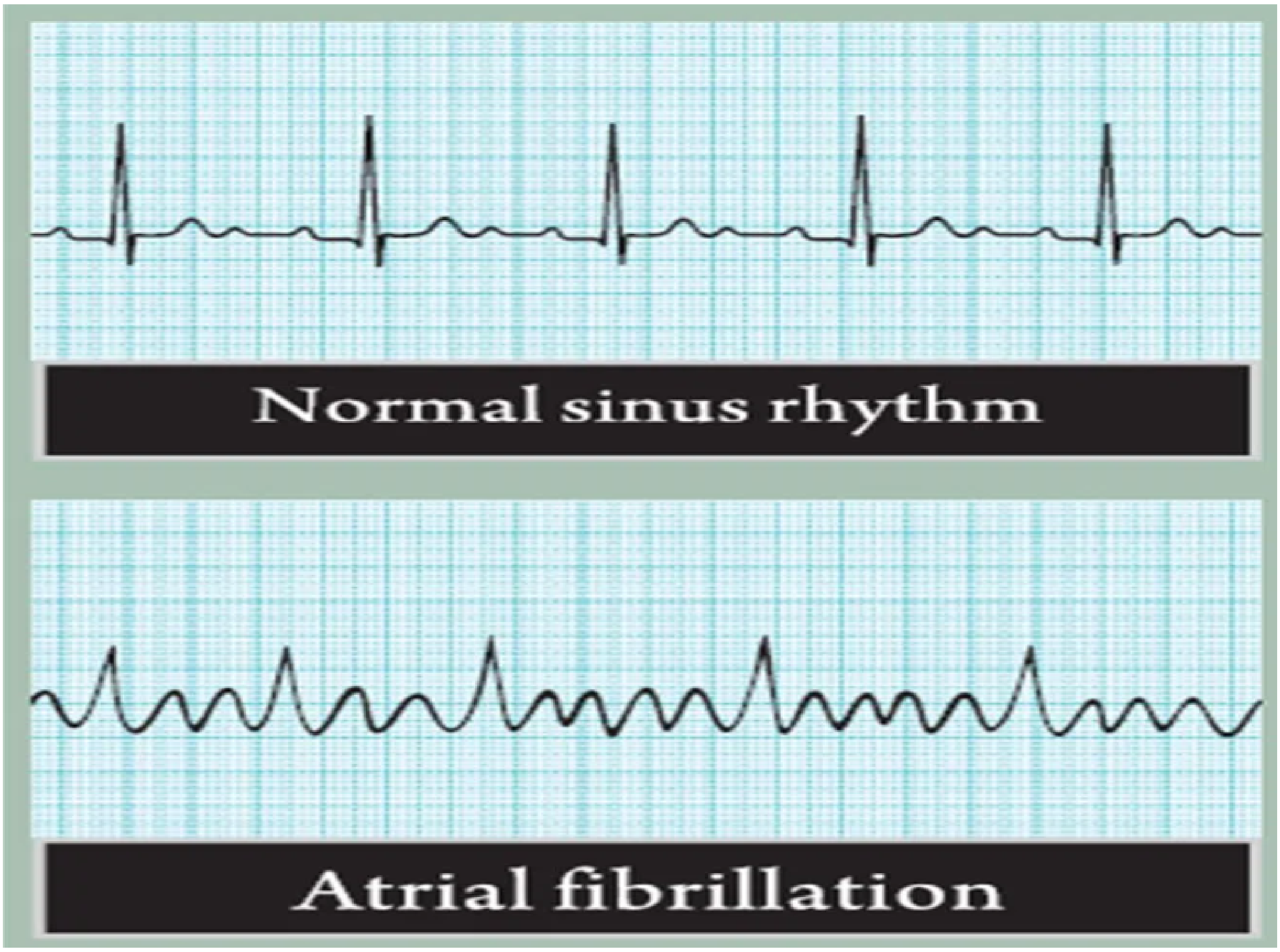


4 Make a survey about various heart diseases diagnosed from the waves of ECG signals.

Electrocardiogram (ECG) is a non-invasive diagnostic tool that uses electrodes placed on the skin to detect and record the electrical activity of the heart. ECG signals are used to diagnose various heart diseases and monitor heart health.

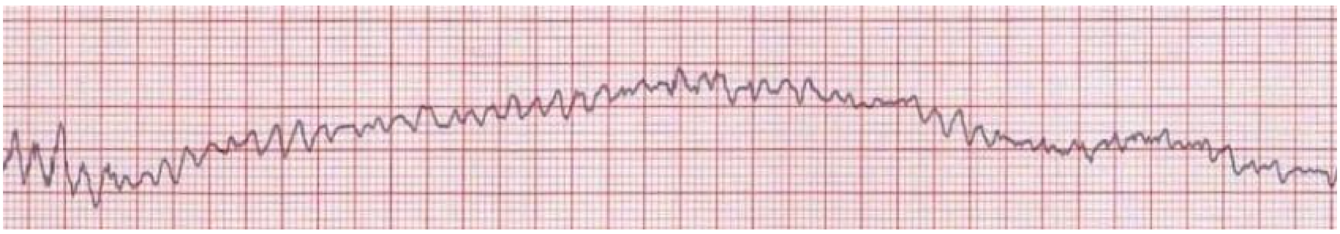
4.1 Atrial fibrillation

This is a heart condition characterized by an irregular, rapid heart rate that occurs when the upper chambers of the heart (the atria) beat irregularly. The ECG waveform in atrial fibrillation often shows an absence of P waves and an irregularly irregular rhythm. The most common type of test for helping to diagnose AF is the electrocardiogram (or ECG) which measures the electrical activity of your heart to show whether or not there is any irregularity. An ECG records the heart's rhythm and activity on a moving strip of paper or a line on a screen.



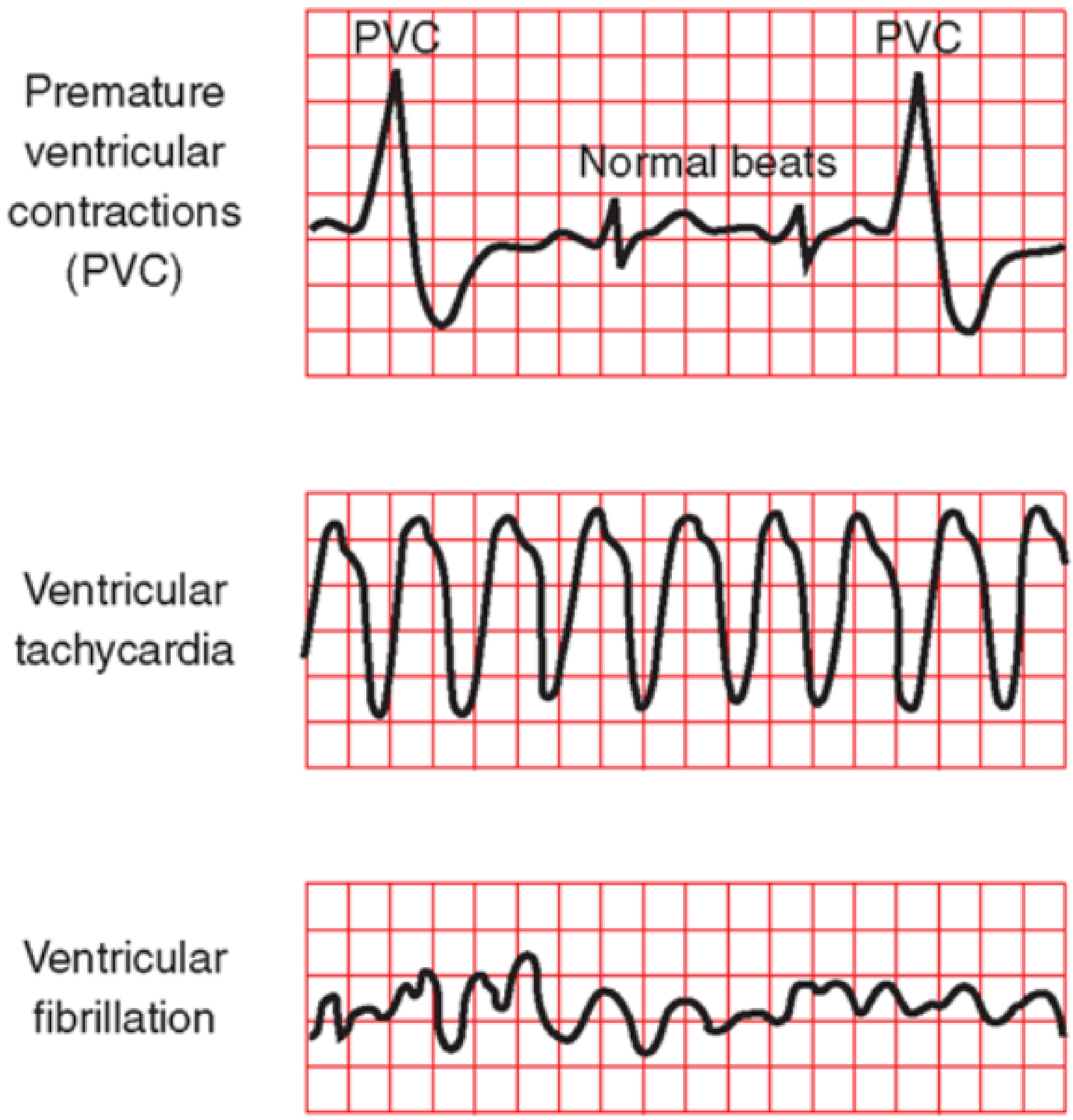
5 Ventricular fibrillation

Ventricular fibrillation (VF) is the most important shockable cardiac arrest rhythm. It is invariably fatal unless advanced life support is



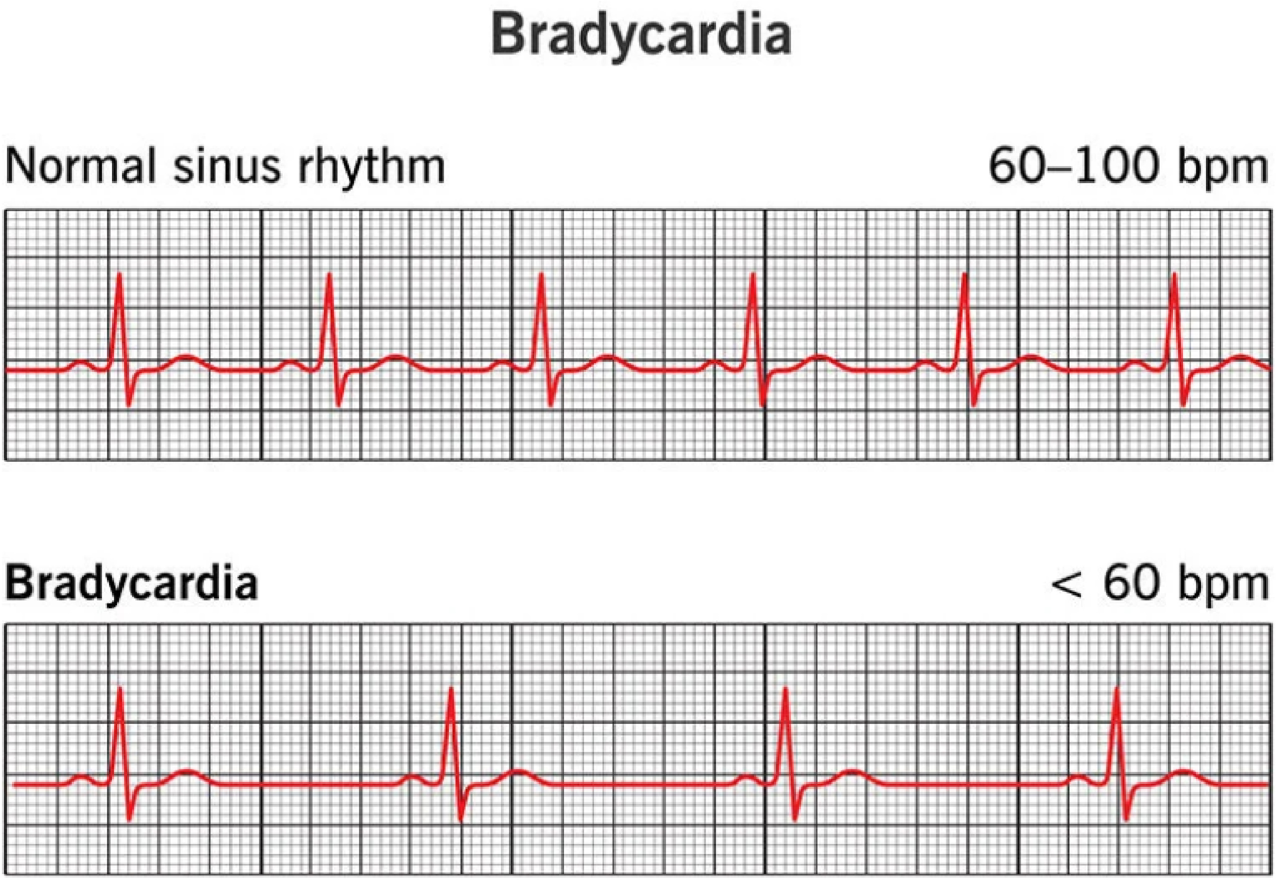
6 Tachycardia

Tachycardia is a fast heart rate, which can be caused by stress, exercise, or an underlying medical condition. ECG signals can detect the fast heart rate and help diagnose Tachycardia



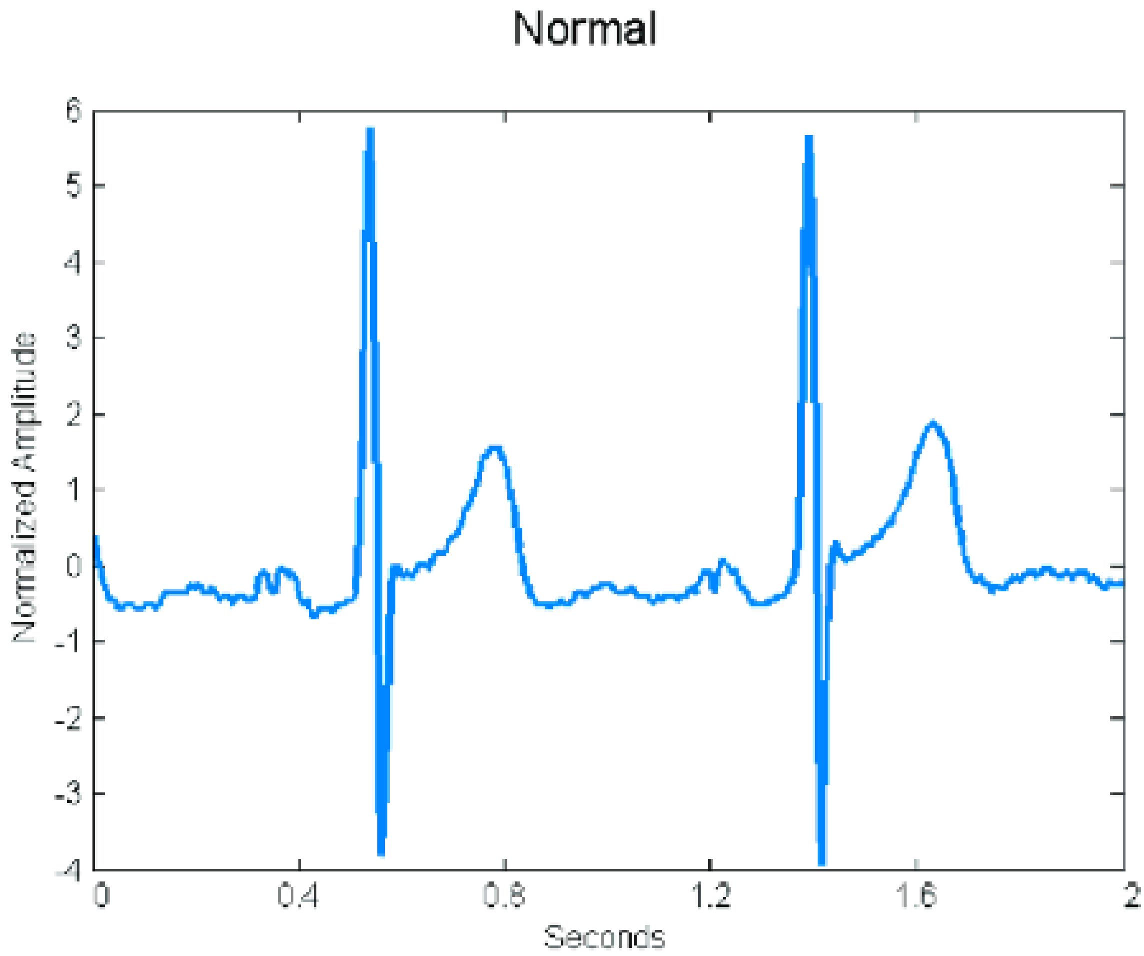
7 Bradycardia

Bradycardia is a slow heart rate. Bradycardia can be caused by medications, heart disease, or an underlying medical condition. ECG signals can detect the slow heart rate and help diagnose Bradycardia

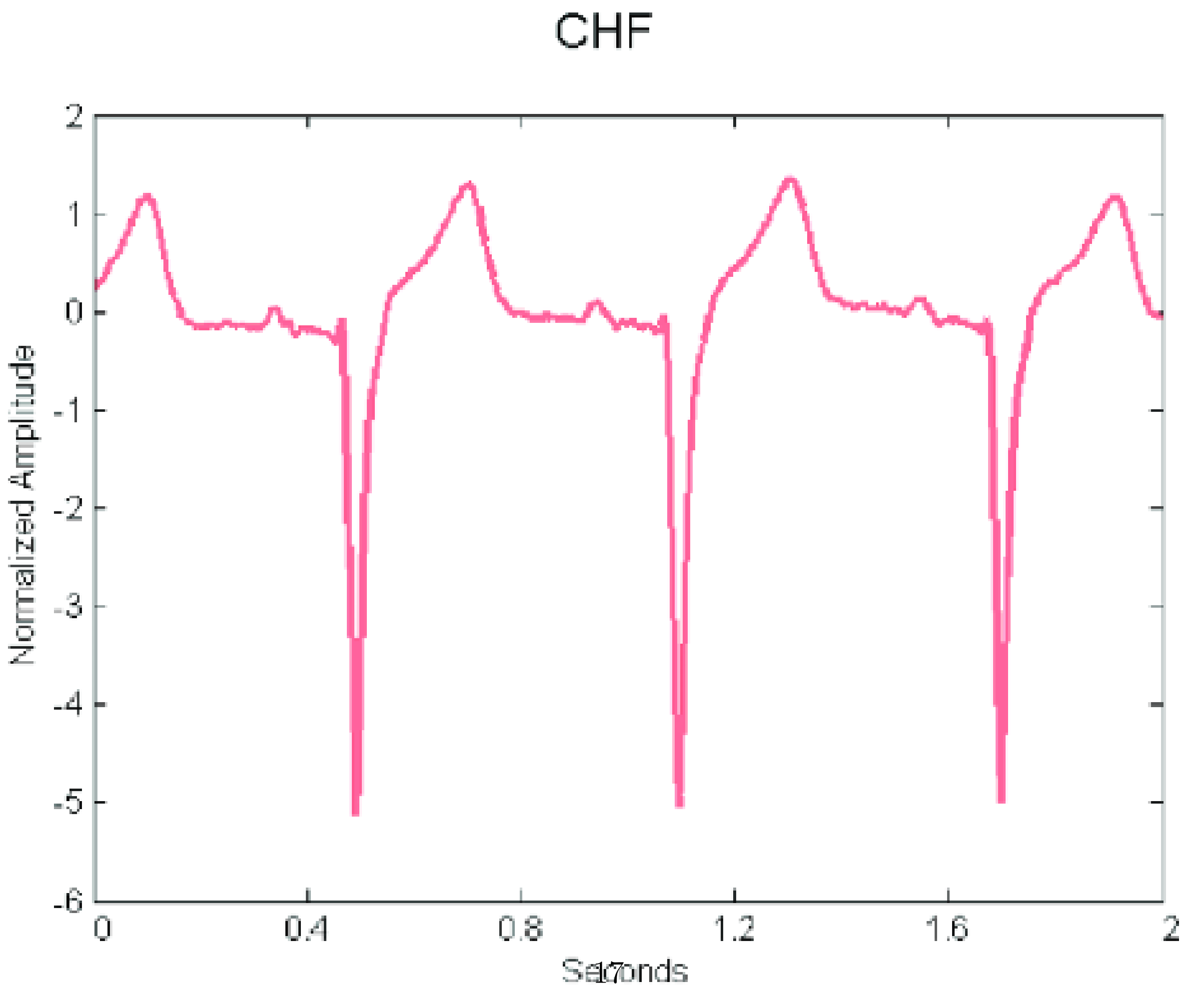


8 Congestive heart failure

Congestive heart failure (CHF) is a chronic condition in which the heart is unable to pump blood effectively, leading to fluid buildup in the lungs and other parts of the body.


Normal
Normalized Amplitude
6
5
4
3
2
1
0
-1
-2
-3
-4
0
0.4
0.8
1.2
1.6
2
Seconds

(a)


CHF
Normalized Amplitude
2
1
0
-1
-2
-3
-4
-5
-6
0
0.4
0.8
1.2
1.6
2
Seconds