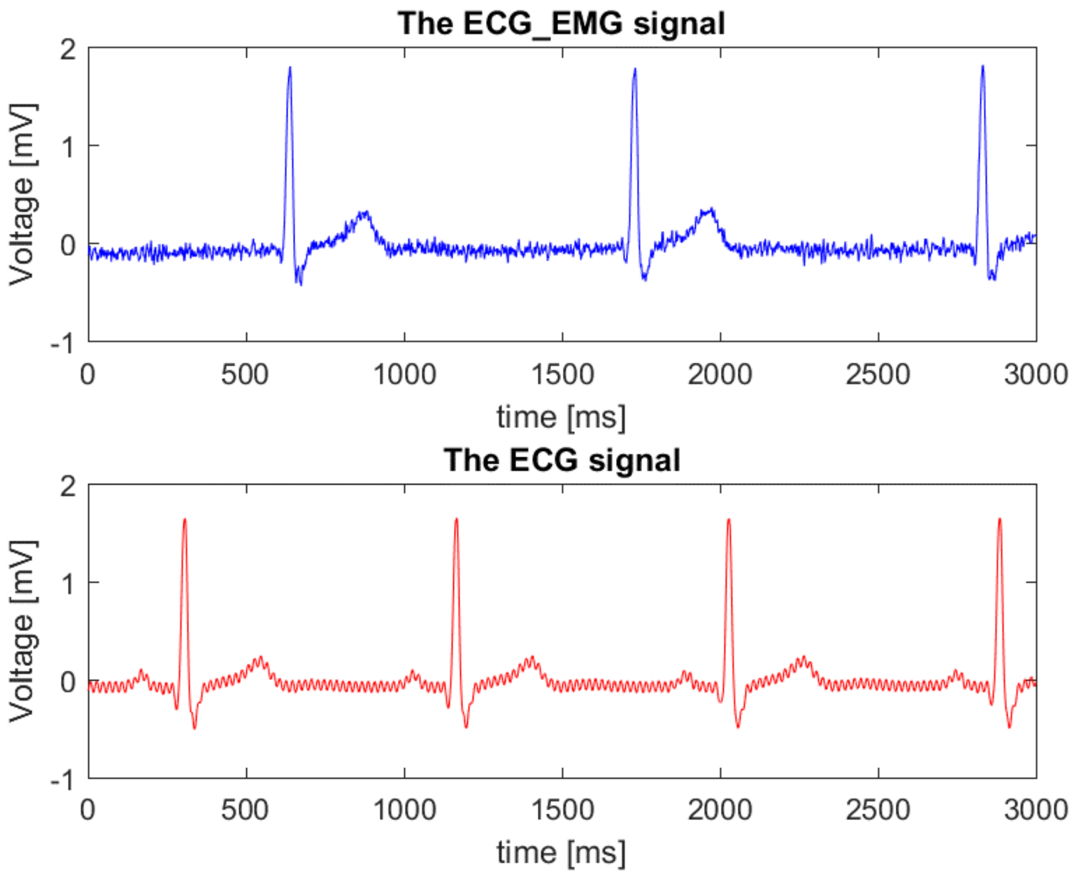
Assignment for BIE103 module: Introduction to MATLAB

Point a

The plot with the ECG and ECG_EMG signals for the first 3 seconds is shown in the following figure

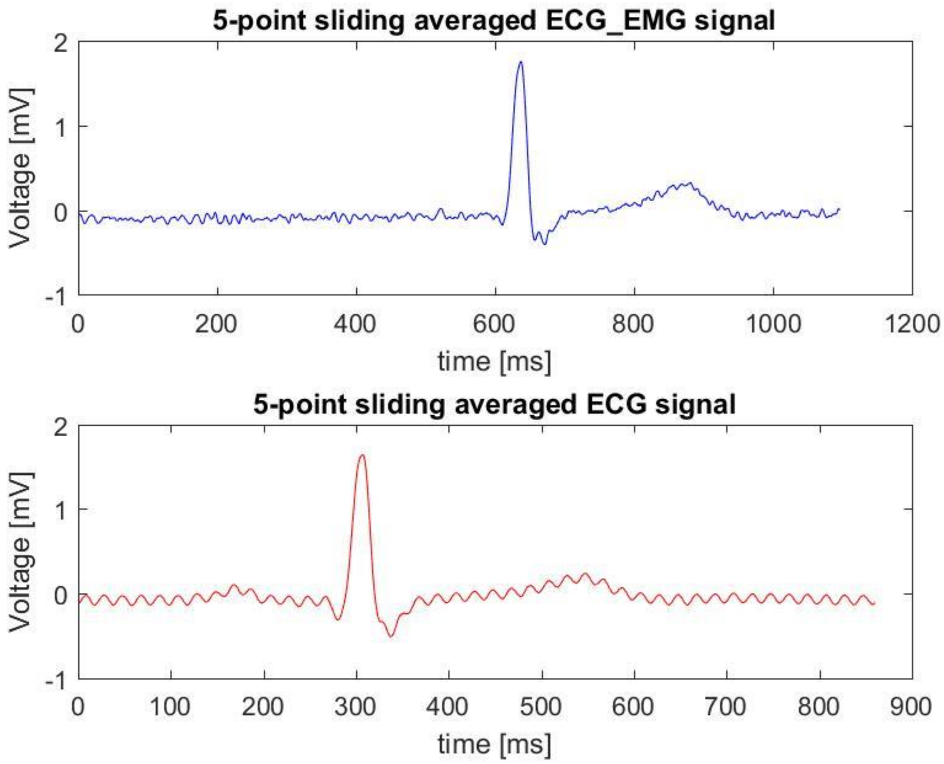


Point b

```
load ecg_wavs;

max_ecg_emg=length(ecg_emg); % the length of the ecg_emg signal
c=zeros(max_ecg_emg-4,1);
% set the length of the averaged vector for ecg_emg signal
for j=1:(max_ecg_emg-4)
     c(j)=(ecg_emg(j)+ecg_emg(j+1)+ecg_emg(j+2)+ecg_emg(j+3)+ecg_emg(j+4))/5;
end

max_ecg=length(ecg50hz); % the length of the ecg50hz signal
b=zeros(max_ecg-4,1); % set the length of the averaged vector for ecg50hz
signal
for i=1:(max_ecg-4)

b(i)=(ecg50hz(i)+ecg50hz(i+1)+ecg50hz(i+2)+ecg50hz(i+3)+ecg50hz(i+4))/5;
end
```

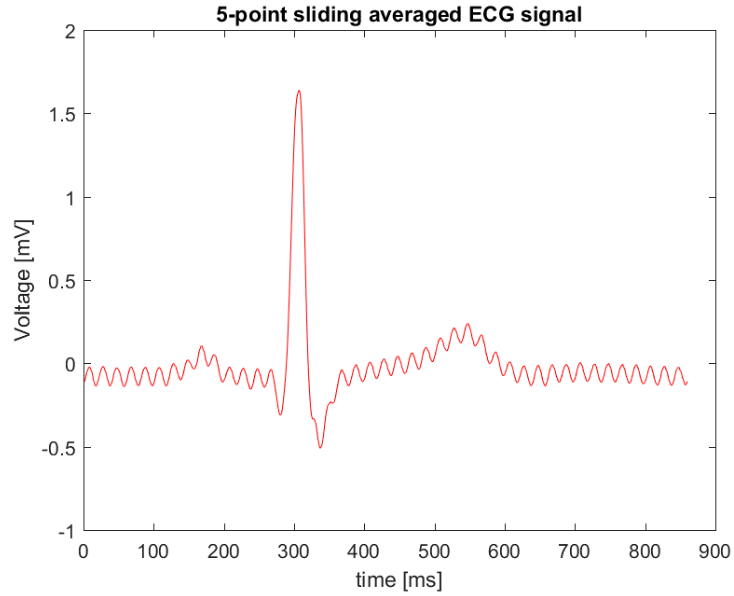
Point c

```
l_x1=860; % approximate length of the first cycle of ecg50hz
x1_ecg50hz=ecg50hz(1:l_x1); % first cycle of ecg50hz
l_y1=1096; % approximate length of the first cycle of ecg_emg
y1_ecg_emg=ecg_emg(1:l_y1); % first cycle of ecg_emg
```



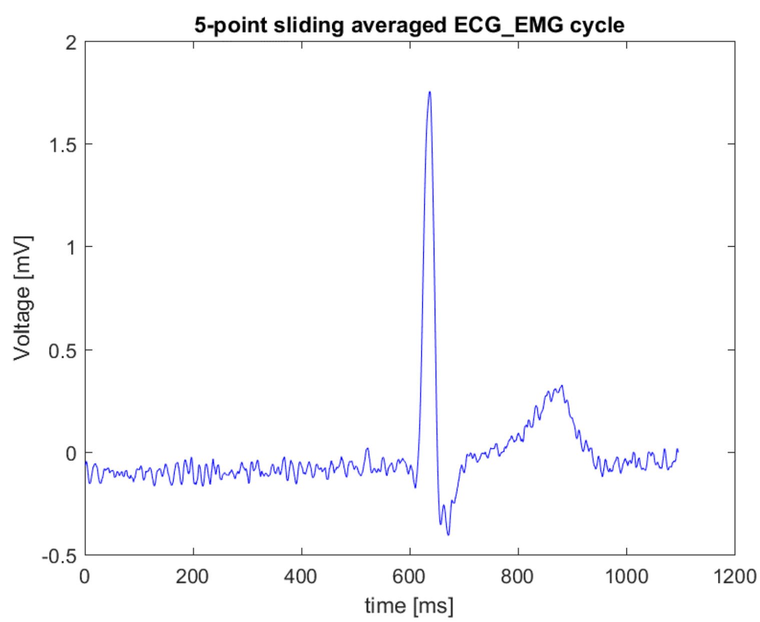

```
figure % only the first cycle is plotted
subplot(2,1,1), plot(c(1:l_y1),'-b')
title('5-point sliding averaged ECG\_EMG signal')
xlabel('time [ms]');
ylabel('Voltage [mV]');
figure % only the first cycle is plotted
subplot(2,1,2), plot(b(1:l_x1),'-r')
title('5-point sliding averaged ECG signal')
xlabel('time [ms]');
ylabel('Voltage [mV]');
```

The 5-point sliding average filter applied on the ecg50hz signal is plotted in the following figure




```
figure, plot(b(1:l_x1),'-r')
title('5-point sliding averaged ECG signal')
xlabel('time [ms]');
ylabel('Voltage [mV]');
```

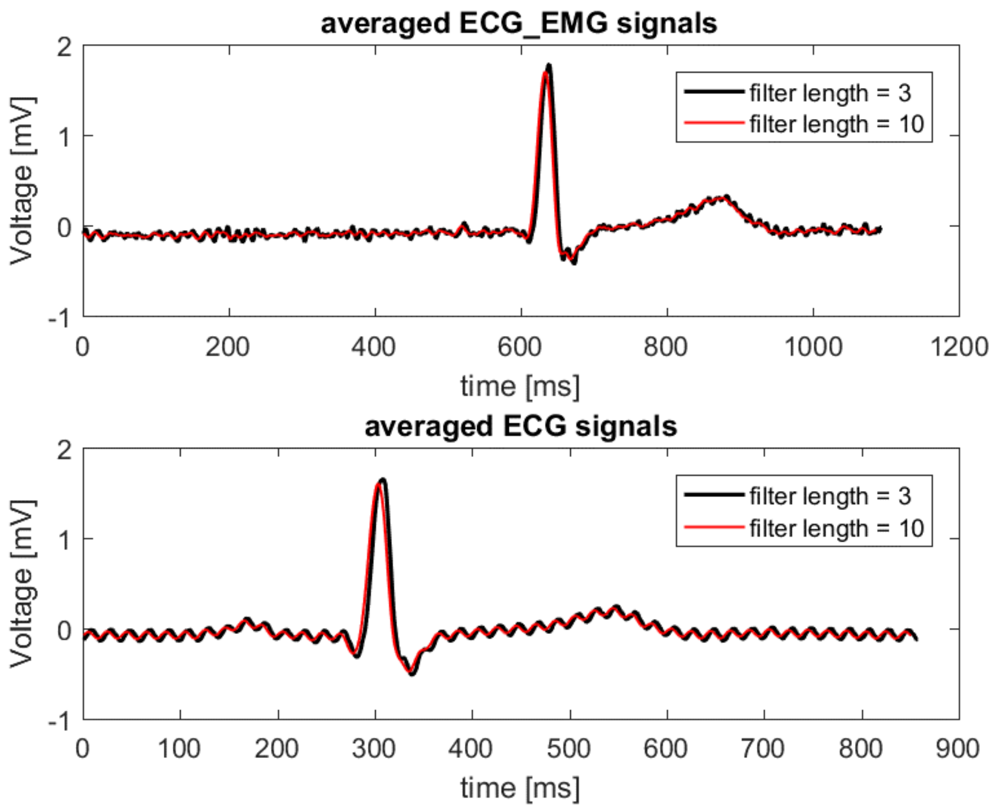
The 5-point sliding average filter applied on the ecg_emg signal is plotted in the following figure




```
figure % only the first cycle is plotted
plot(c(1:l_y1),'-b')
title('5-point sliding averaged ECG\_EMG cycle')
xlabel('time [ms]');
ylabel('Voltage [mV]');
```

Point d

The averaged cycle for both the *ecg_emg* and *ecg50hz* signals for filter lengths of 3 and 10 is shown in the following figure. It can be noticed that the smoothing effect is higher for a higher filter length. The smoothing effect is more evident for the ECG_EMG signal than for the ECG signal. Also, there is a shift of the smoothed signals that depends on the filter length.




```
[~,locs_wave] =
findpeaks(ecg50hz(1:3000),'MinPeakHeight',1.5,'MinPeakDistance',800)
first_cycle_length_ecg50hz = mean (diff(locs_wave))
% locs_wave gives the location of the peaks whose MinPeakHeight is higher
% than 1.5 and MinPeakDistance between the waves is higher than 800
[~,locs_wave1] =
findpeaks(ecg_emg(1:3000),'MinPeakHeight',1.5,'MinPeakDistance',800)
first_cycle_length_ecg_emg = mean (diff(locs_wave1))
% the approximate cycle of the ecg_emg signal has 1096 samples (the distance
between the peaks)
```

Use the function from point e

```
average3_x1_ecg50hz=Maria_Albu_function(x1_ecg50hz,3); % averaged ecg50hz
signal, length of filter is 3
average10_x1_ecg50hz=Maria_Albu_function(x1_ecg50hz,10); % averaged ecg50hz
signal, length of filter is 10

average3_y1_ecg_emg=Maria_Albu_function(y1_ecg_emg,3); % averaged ecg_emg
signal, length of filter is 3
average10_y1_ecg_emg=Maria_Albu_function(y1_ecg_emg,10); % averaged ecg_emg
signal, length of filter is 10

figure % only the first cycle is plotted
subplot(2,1,1), plot(average3_y1_ecg_emg,'k','Linewidth',1.5)
title('averaged ECG\_EMG signals')
hold on;
plot(average10_y1_ecg_emg,'r','Linewidth',1)
xlabel('time [ms]');
ylabel('Voltage [mV]');
legend('filter length = 3','filter length = 10')

subplot(2,1,2), plot(average3_x1_ecg50hz,'k','Linewidth',1.5)
title('averaged ECG signals')
hold on;
plot(average10_x1_ecg50hz,'r','Linewidth',1)
xlabel('time [ms]');
ylabel('Voltage [mV]');
legend('filter length = 3','filter length = 10')
```

Point e

The function is Maria_Albu_function.m

It takes an input signal and the filter length and generates the smoothed signal.

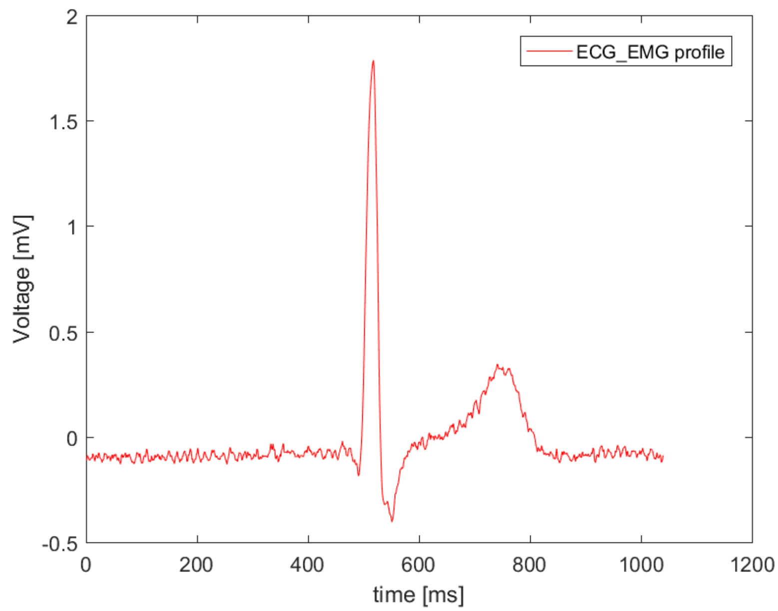
```
function [smoothed] = Maria_Albu_function(x, Nl)
% Inputs:
% x -> The arbitrary input signal
% Nl-> filter length
% Output:
% smoothed-> smoothed signal

max_x=length(x); % find the length of the input signal
smoothed=zeros(max_x-Nl+1,1); % % set the length of the smoothed signal
for j=1:max_x-Nl+1
      smoothed(j)=mean(x(j:j+Nl-1));
end
```

Points f and g

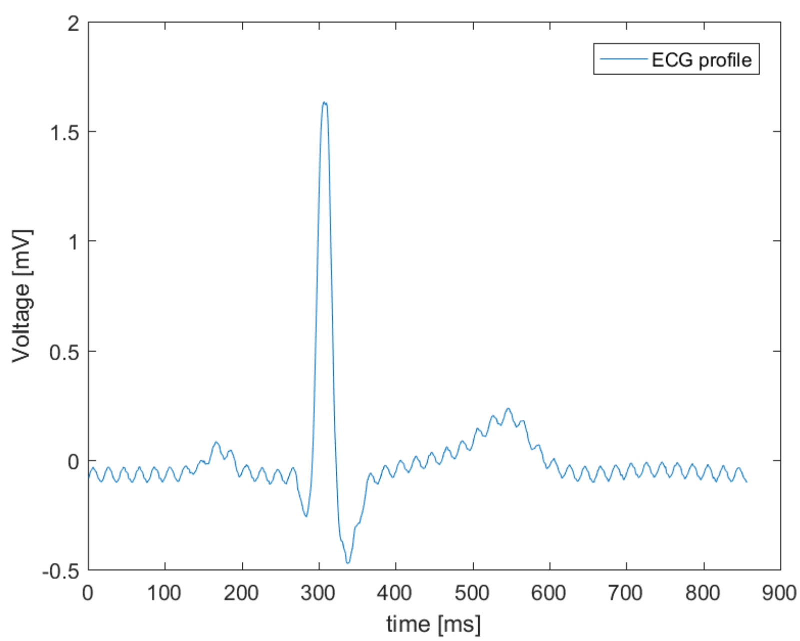
The function used is Maria_Albu_function2.m

It takes as an input the ECG or ECG_EMG signals and generates the averages of all the cycles.

The ECG_EMG profile is plotted in the following figure.



The ECG profile is plotted in the following figure.



It can be noticed that the smoothing method that obtain the ECG_EMG profile gives better results that the averaging method, while for the ECG signal the results are pretty similar.

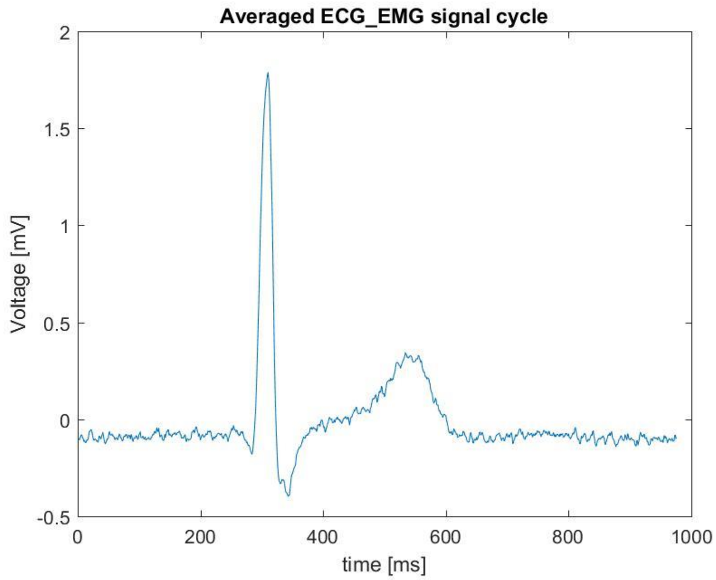
The following function takes an input ECG or ECG_EMG signal and a parameter that represents the minimum expected distance between peaks, `MinPeakDistance` (800 is a good value for both signals).

The output of the function is the ECG or ECG_EMG profile.

```
function [averaged_signal] = Maria_Albu_function2(x, MinPeakDistance)
% Inputs:
% x-> The arbitrary input: ECG or ECG_EMG signals
% MinPeakDistance-> Minimum Peak Distance
% Output:
% averaged_signal-> smoothed signal

len_x=length(x); % find the length of the input signal
val_max=max(x); % Maximum value of the input signal
MinPeakHeight=0.9*val_max; % set the threshold to 90% of the maximum value of
the input signal

loc_wave=[]; % find the points above the threshold and put them in loc_wave
vector
for i=1:len_x
    if x(i)>MinPeakHeight;
    loc_wave=[loc_wave i];
    end
end

len_locs_wave=length(loc_wave); % the number of points above the threshold
% find the points above the threshold that are at least at MinPeakDistance
from their neighbour

locs_wave=[]; % put them in the locs_wave vector
for j=1:len_locs_wave-1
    if loc_wave(j+1)-loc_wave(j)>MinPeakDistance
        locs_wave=[locs_wave loc_wave(j)];
    end
end

vec2=min(diff(locs_wave)); % the period of peaks is the minimum value of the
difference between peaks
A2=min(min(locs_wave),fix(vec2/2));  % set the A value from the left of the
peak
l_locs_wave=length(locs_wave); % number of found peaks
matrice=zeros(l_locs_wave,vec2); % create the matrix where each row is a
cycle related to peaks
for i=1:l_locs_wave
    matrice(i,:)=x(locs_wave(i)-A2+1:locs_wave(i)+vec2-A2);
end
averaged_signal=mean(matrice); % the average signal cycle
```

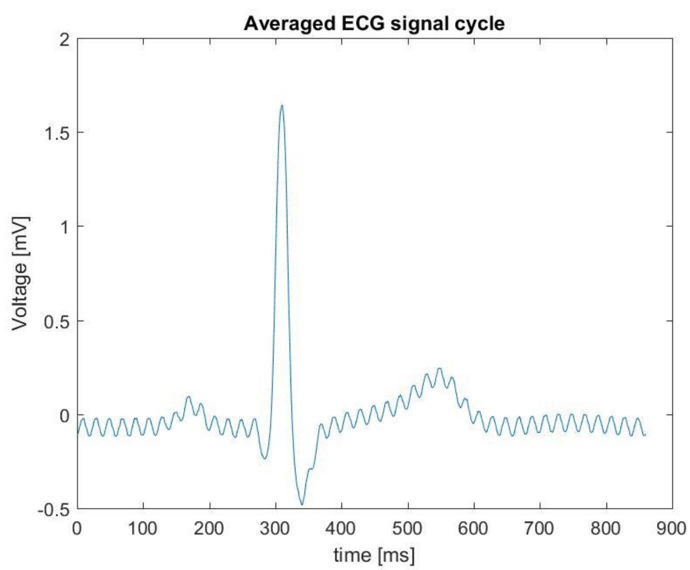
```
MinPeakDistance=800;
[averaged_ecg50hz] = Maria_Albu_function2(ecg50hz,MinPeakDistance);
[averaged_ecg_emg] = Maria_Albu_function2(ecg_emg,MinPeakDistance);

figure,
plot(averaged_ecg50hz);
legend('ECG profile')
xlabel('time [ms]');
ylabel('Voltage [mV]');

figure
plot(averaged_ecg_emg,'r');
legend( 'ECG\_EMG profile')
xlabel('time [ms]');
ylabel('Voltage [mV]');


% An alternative to find the peaks
[~,locs_wave1] = findpeaks(ecg_emg,'MinPeakHeight',1.5,'MinPeakDistance',800)
l_ecg_emg = min (diff(locs_wave1));
A1=min(min(locs_wave),fix(l_ecg_emg/2));
l_locs_wave1=length(locs_wave1); % number of peaks
matrice=zeros(l_locs_wave1,l_ecg_emg);
for i=1:l_locs_wave1
    matrice(i,:)=ecg_emg(locs_wave1(i)-A1+1:locs_wave1(i)+l_ecg_emg-A1);
end

[~,locs_wave] = findpeaks(ecg50hz,'MinPeakHeight',1.5,'MinPeakDistance',800);
vec2=min (diff(locs_wave));
l_ecg50hz = min (diff(locs_wave));
A2=min(min(locs_wave),fix(l_ecg50hz/2));
l_locs_wave=length(locs_wave); % number of peaks
matrice1=zeros(l_locs_wave,l_ecg50hz);
for i=1:l_locs_wave
    matrice1(i,:)=ecg50hz(locs_wave(i)-A2+1:locs_wave(i)+l_ecg50hz-A2);
end
```

Plotting the averaged ECG_EMG signal cycle




```
figure
averaged_signal=mean(matrice);
plot(averaged_signal)
xlabel('time [ms]');
ylabel('Voltage [mV]');
title('Averaged ECG_EMG signal cycle')
```

Plotting the averaged ECG signal cycle




```
figure
averaged_signal1=mean(matrice1);
plot(averaged_signal1)
xlabel('time [ms]');
ylabel('Voltage [mV]');
title('Averaged ECG signal cycle')
```