Experiment No. 04

4.1 Experiment Name

Development of a MATLAB based GUI app for processing ECG signal to determine Cardiac state

4.2 Objectives

- To develop a MATLAB-based GUI application that effectively processes ECG signals
- To contribute to the efficient determination of cardiac states, enabling timely diagnosis and intervention

4.3 Theory

The electrocardiogram (ECG) stands as a pivotal diagnostic tool, capturing the heart's electrical activity over time. In the realm of ECG signal processing, essential steps include filtering, noise reduction, and feature extraction. Leveraging MATLAB's capabilities in data analysis, signal processing, and visualization proves ideal for developing tools tailored to process and analyze ECG signals. The Graphical User Interface (GUI) feature within MATLAB enhances user interaction, paving the way for user-friendly applications. The integration of automated analysis within the GUI application streamlines this complex process, offering a swift and accurate means of assessing cardiac health.

4.4 Apparatus

- ❖ MATLAB

4.5 MATALAB Code

```
function varargout = ecg(varargin)
gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...
'gui_Singleton',  gui_Singleton, ...
'gui_OpeningFcn', @ecg_OpeningFcn, ...
'gui_OutputFcn',  @ecg_OutputFcn, ...
'gui_LayoutFcn',  [] , ...
'gui_Callback',   []);
if nargin && ischar(varargin{1})
gui_State.gui_Callback = str2func(varargin{1});
end

if nargout
[varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
gui_mainfcn(gui_State, varargin{:});
end
function ecg_OpeningFcn(hObject, eventdata, handles, varargin)
handles.output = hObject;
guidata(hObject, handles);
function varargout = ecg_OutputFcn(hObject, eventdata, handles)
varargout{1} = handles.output;
function ecgsig_Callback(hObject, eventdata, handles)
fullfile = dlmread('ECG_Signal.tsv');  %Load ECG signal during walking
Fs = 250; %The data are sampled at 250Hz
ecgsig = fullfile(:,2);  %reading ecg signal
samples = 1:length(ecgsig);  %No. of samples
tx = samples./Fs;  %Getting time vector
f = 1./tx; % frequency of ecg
set(handles.listbox1,'String',ecgsig);
function listbox1_Callback(hObject, eventdata, handles)
function listbox1_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
set(hObject,'BackgroundColor','white');
```

```
end
function plot_ecg_Callback(hObject, eventdata, handles)
ecgsig = str2num(get(handles.listbox1,'String'));
Fs = 250; %The data are sampled at 250Hz
samples = 1:length(ecgsig);  %No. of samples
tx = samples./Fs;  %Getting time vector
f = 1./tx; % frequency of ecg
axes(handles.axes1);
plot(tx,ecgsig);
title('Original Lead II ECG Signal')
legend('Original ECG');
xlabel('Time (sec)')
ylabel('Voltage (volt)')
grid on
function baseline_drift_Callback(hObject, eventdata, handles)
ecgsig = str2num(get(handles.listbox1,'String'));
Fs = 250; %The data are sampled at 250Hz
samples = 1:length(ecgsig);  %No. of samples
tx = samples./Fs;  %Getting time vector
f = 1./tx; % frequency of ecg
for i = 1:1:length(ecgsig)
if i == 1
m(i) = ecgsig(i+1)-ecgsig(i);
end
if i>1
m(i) = ecgsig(i-1)-ecgsig(i);
end
end
denoised = m;
wp=20; ws=60; rp=0.5; rs=25; %Design a Butterworth filter of order 9 for smoothing noise
[N, Wn] = buttord(wp/(Fs/2), ws/(Fs/2), rp, rs);
[b, a]=butter(N, Wn);
yy=filter(b,a,denoised);
axes(handles.axes1);
plot(tx, yy);
title('Baseline noise removed ECG')
legend('Baseline noise removed ECG');
xlabel('Time (sec)')
ylabel('Voltage (volt)')
grid on
function r_peak_Callback(hObject, eventdata, handles)
ecgsig = str2num(get(handles.listbox1,'String'));
Fs = 250; %The data are sampled at 250Hz
samples = 1:length(ecgsig);  %No. of samples
tx = samples./Fs;  %Getting time vector
f = 1./tx; % frequency of ecg

for i = 1:1:length(ecgsig)
if i == 1
m(i) = ecgsig(i+1)-ecgsig(i);
end
if i>1
m(i) = ecgsig(i-1)-ecgsig(i);
end
end
base_remove_signal = m;
for i = 1:1:length(base_remove_signal)
if base_remove_signal(i)>=0.0005
n(i)=base_remove_signal(i);
end
end
r_peak=n;
plot(r_peak);
title('R Peak Detected Signal')
legend('R Peak Detected Signal');
xlabel('Time (sec)')
ylabel('Voltage (volt)')
grid on
```

```
end
end
```

4.6 GUI Layout

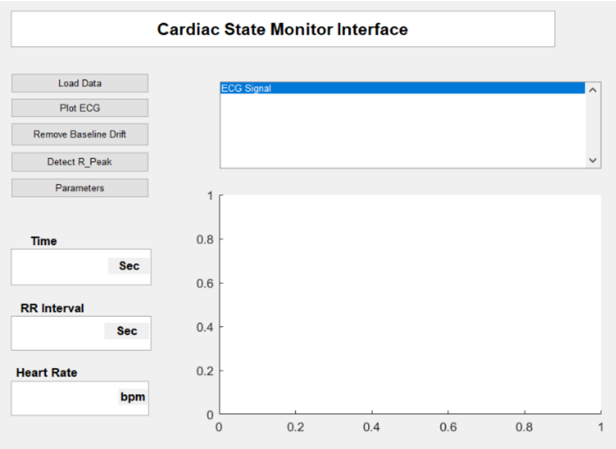

Fig.4.1: *GUI of Cardiac States Monitor*

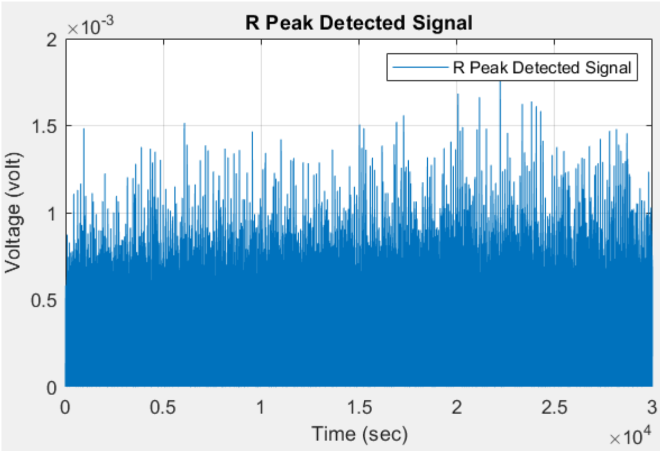

Fig.4.2: R-peak Detected signal

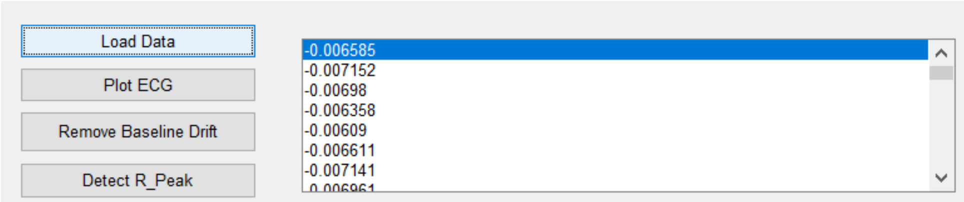

Fig.4.3: Load Data

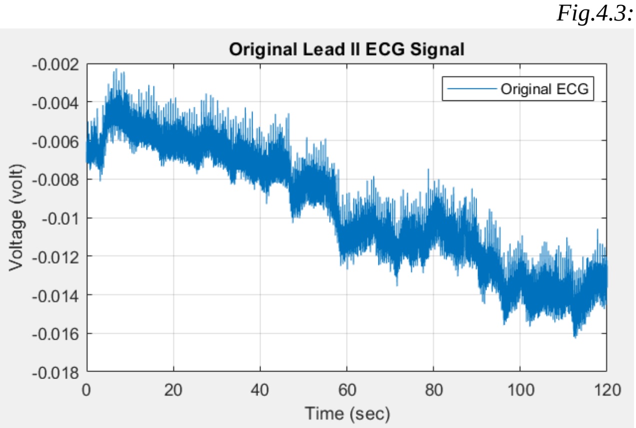

Fig.4.4: Original Lead II ECG Signal

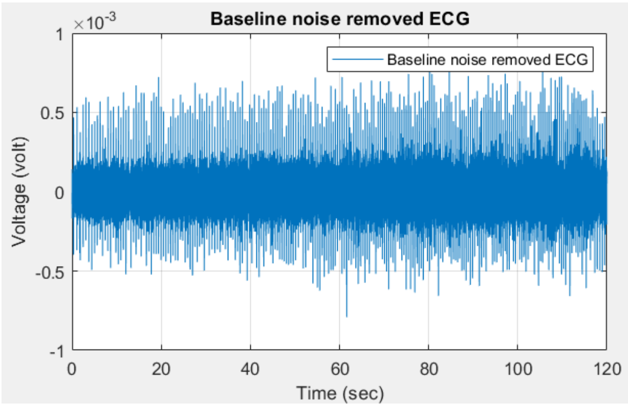

Fig.4.5: Baseline noise removed ECG Signal

4.7 Discussion & Conclusion

The experiment successfully developed a MATLAB-based GUI application for ECG signal processing, facilitating efficient cardiac state determination. MATLAB's robust capabilities in data analysis and signal processing, combined with a user-friendly GUI, presented a valuable tool for healthcare professionals. Processed ECG signals allowed for the assessment of normal rhythms, arrhythmias, and patterns indicative of cardiac disorders. The automated analysis within the GUI demonstrated efficiency, promising accelerated decision-making in clinical settings. Further validation and testing in diverse scenarios are essential for ensuring the reliability and applicability of this innovative tool.