Table of Contents

Biosignal Analysis Using Matlab - Project 3 - Group 10

Project : ECG Signal Component Analysis

Juan Pablo Florez Suarez, Student ID :12536870

Aida Daniela Becerra Lopez, Student ID : 12432114

MATLAB Version : R2025b File Version : V00 .00

1. (1P) Clear the workspace.

```
close all
clear
clc
```

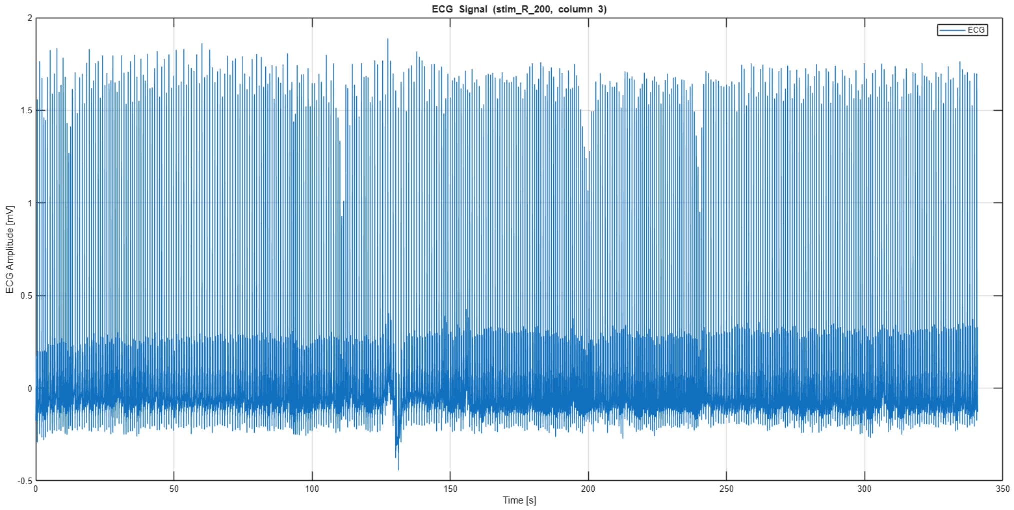
2. (1P) Load the file stim_R_200.csv into the workspace and remove header if applicable.

```
filename = 'stim_R_200.csv';
Fs = 10e3; % sampling frequency [Hz]
```

```
% Robust CSV import (handles potential header lines)
opts = detectImportOptions(filename);
tbl  = readtable(filename, opts);

% Use only the 3rd column (task definition)
if width(tbl) < 3
    error('Input file has %d columns, but column 3 is required.',
width(tbl));
end

ecg = tbl{:,3};
ecg = ecg(:);

N = numel(ecg);
time = (0:N-1)'/Fs;
```

3. (1P) Plot the ECG signal in a figure. Label axes and name the signal.

```
h(1).hf = figure('Name','ECG Signal');
h(1).ha = axes('Parent', h(1).hf);

h(1).hp(1) = plot(h(1).ha, time, ecg, 'DisplayName','ECG');
grid(h(1).ha, 'on');

h(1).ht = title(h(1).ha, 'ECG Signal (stim\_R\_200, column 3)');
h(1).hx = xlabel(h(1).ha, 'Time [s]');
h(1).hy = ylabel(h(1).ha, 'ECG Amplitude [mV]');

h(1).hl = legend(h(1).ha, 'show', 'Location', 'best');
```



4. (27P) Analysis of the ECG signal

4.b (2P) Find the QRS-complexes in the ECG signal.

R peak detection (Pan–Tompkins style front-end)

```
[R_locs, ~, bp_ecg] = detectRpeak(ecg, Fs, false); %Checked function

% QRS delineation: Q onset, Q peak, R peak, S peak, S offset
[Q_on, Q_peak_fil, R_peak, S_peak_fil, S_off] = delineateQRS(bp_ecg, R_locs,
Fs, false); %Checked function

% Re-locate Q and S peaks on the ORIGINAL ECG
padQ_ms = 5; % safety margin before R (ms)
padS_ms = 5; % safety margin after R (ms)

% convert ms to samples
padQ = round(padQ_ms * 1e-3 * Fs);
padS = round(padS_ms * 1e-3 * Fs);

Q_peak = refineMinBetween(ecg, Q_on, R_peak - padQ);
S_peak = refineMinBetween(ecg, R_peak + padS, S_off);
```

4.a (2P) Find the P-peaks in the ECG signal.

We also compute P onset/offset to enable PR-interval and PR-segment.

```
[P_on, P_peak, P_amp, P_off] = detectPwave(ecg, R_peak, Q_on, Fs, false, ...
                                            'BaseAmpFrac', 0.5, ...
                                            'BaseSlopeFrac', 0.40, ...
                                            'DerivSmooth_ms', 10, ...
                                            'OnsetWin_ms', 50, ...
                                            'OffsetWin_ms', 40, ...
                                            'BaseAmpFrac_off', 0.65, ...
                                            'BaseSlopeFrac_off', 0.80);
%Checked function
```

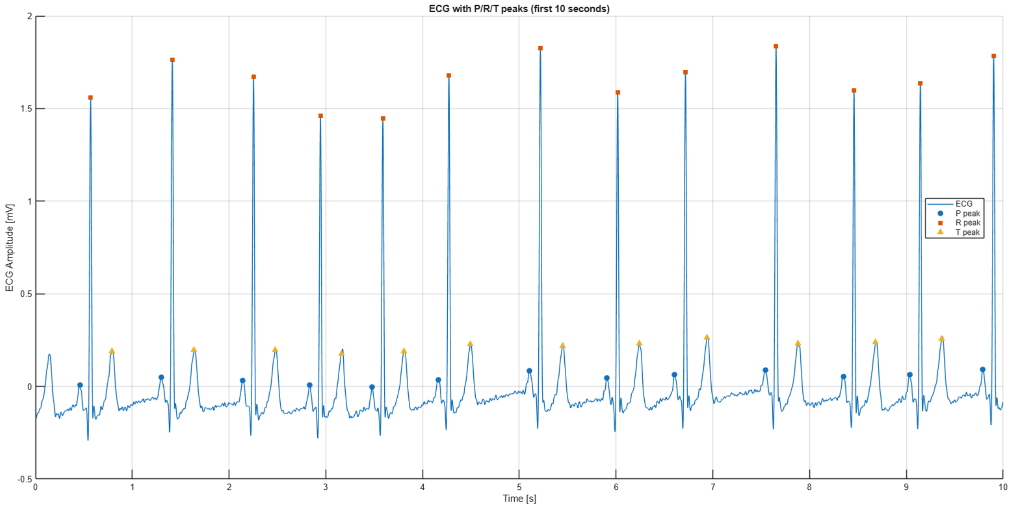
4.c (2P) Find the T-peaks in the ECG signal.

We also compute T onset/offset to enable QT-interval, ST-interval, ST-segment.

```
[T_on, T_peak, ~, T_off] = detectTwave(ecg, R_peak, S_off, Fs, false,...
                                        'OnsetWin_ms', 80, ...
                                        'OffsetWin_ms', 120, ...
                                        'BaseAmpFrac', 0.75, ...
                                        'BaseSlopeFrac', 0.55);
```

4.d (1P) Plot ECG and mark found peaks.

```
h(2).hf = figure('Name','ECG with P/R/T peaks');
h(2).ha = axes('Parent', h(2).hf);
h(2).ha = plotECGWithMarkers(ecg, Fs, 'ECG with P/R/T peaks (first 10
seconds)','Parent', ...
    h(2).ha,'tmin', 0, 'tmax', 10, P_peak, 'P peak', R_peak, 'R
peak',T_peak, 'T peak');

% Title / labels
h(2).ht = get(h(2).ha, 'Title');
h(2).hx = get(h(2).ha, 'XLabel');
h(2).hy = get(h(2).ha, 'YLabel');

% Legend
h(2).hl = legend(h(2).ha);

% All line objects in this axes
h(2).hp = flipud(findobj(h(2).ha, 'Type', 'line'));
```



(12P) Compute intervals/segments and store their durations.

```
intervals = computeECGIntervals(R_peak, P_on, P_off, Q_on, S_off, T_on,
T_off, Fs);
```

4.e RR Interval

```
RR_s = intervals.RR_s;        %checked
```

4.f PR interval

```
PR_s = intervals.PR_s;       %checked
```

4.g QT interval

```
QT_s = intervals.QT_s;       %checked
```

4.h ST interval

```
STint_s = intervals.STint_s; %checked
```

4.i QRS interval

```
QRS_s = intervals.QRS_s;     %checked
```

4.j PR segment

```
PRseg_s = intervals.PRseg_s; %checked
```

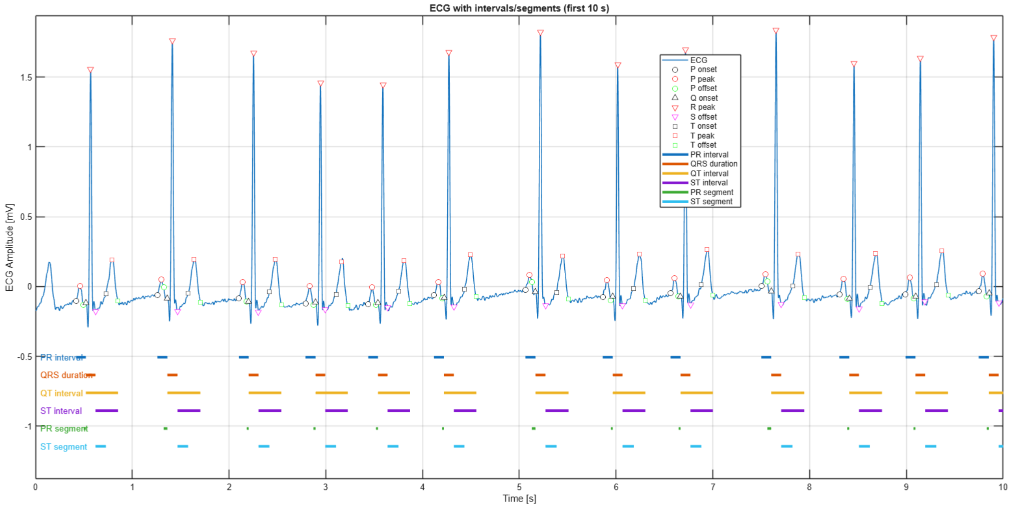
4.k ST segment

```
STseg_s = intervals.STseg_s; %checked

% HR bpm
mean(intervals.HR_bpm);
```

4.l (1P) Plot ECG and mark the found intervals and segments.

```
markers = struct('P_on',P_on, 'P_peak',P_peak, 'P_off',P_off, ...
    'Q_on',Q_on, 'R',R_peak, 'S_off',S_off, ...
    'T_on',T_on, 'T_peak',T_peak, 'T_off',T_off);

h(3).hf = figure('Name','ECG with intervals/segments');
h(3).ha = axes('Parent', h(3).hf);

h(3).ha = plotECGWithIntervals(ecg, Fs, ...
    'ECG with intervals/segments (first 10 s)', intervals, markers, ...
    'Parent', h(3).ha,'tmin', 0, 'tmax', 10);

% Title / labels (text objects stored in the axes)
h(3).ht = get(h(3).ha, 'Title');
h(3).hx = get(h(3).ha, 'XLabel');
h(3).hy = get(h(3).ha, 'YLabel');

% Legend handle (returns existing legend if already created)
h(3).hl = legend(h(3).ha);
```

```
% All line objects in this axes:
h(3).hp = flipud(findobj(h(3).ha, 'Type', 'line'));
```



4.m (2P) Determine respiratory component and calculate the envelope.

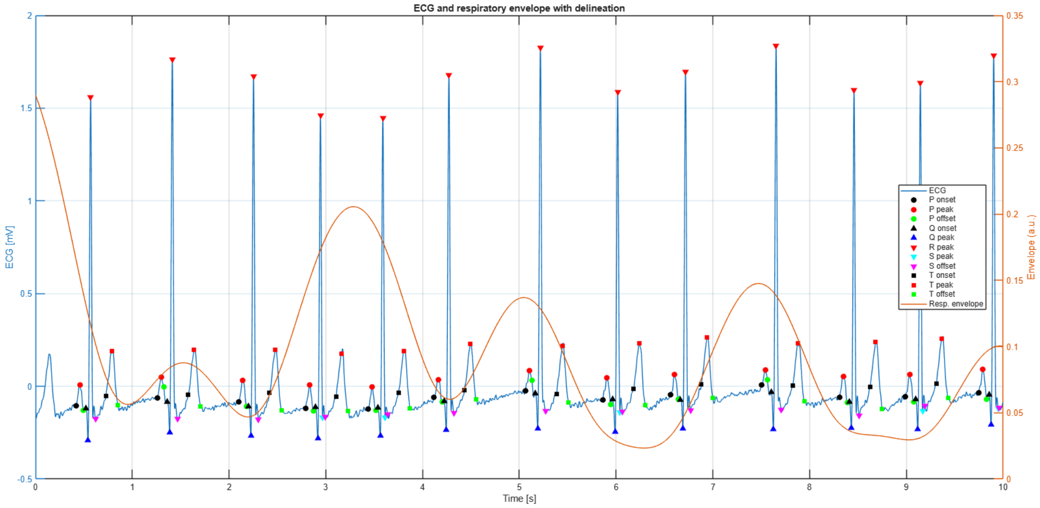
We use ECG-derived respiration (EDR) from beat-to-beat R-peak amplitude modulation.

```
[resp, env, t_env] = ecgRespEnvelope(ecg, R_peak, Fs, false); %checked
```

4.n (1P) Plot ECG and envelope in one figure and add all components.

```
h(4).hf = figure('Name','ECG + Respiratory Envelope');
h(4).ha = axes('Parent', h(4).hf);

yyaxis(h(4).ha, 'left');
h(4).hp = gobjects(0);

h(4).hp(end+1) = plot(h(4).ha, time, ecg, 'DisplayName','ECG');
grid(h(4).ha,'on');
hold(h(4).ha,'on');

% Helper masks
iPon  = P_on(isfinite(P_on));
iPpk  = P_peak(isfinite(P_peak));
iPoff = P_off(isfinite(P_off));

iQon  = Q_on(isfinite(Q_on));
iQpk  = Q_peak(isfinite(Q_peak));
```

```
iRpk  = R_peak(isfinite(R_peak));

iSpk  = S_peak(isfinite(S_peak));
iSoff = S_off(isfinite(S_off));

iTon  = T_on(isfinite(T_on));
iTpk  = T_peak(isfinite(T_peak));
iToff = T_off(isfinite(T_off));

% Markers
h(4).hp(end+1) = plot(h(4).ha, time(iPon),  ecg(iPon),  'ko',
'MarkerFaceColor','k', 'LineStyle','none', 'DisplayName','P onset');
h(4).hp(end+1) = plot(h(4).ha, time(iPpk),  ecg(iPpk),  'ro',
'MarkerFaceColor','r', 'LineStyle','none', 'DisplayName','P peak');
h(4).hp(end+1) = plot(h(4).ha, time(iPoff), ecg(iPoff), 'go',
'MarkerFaceColor','g', 'LineStyle','none', 'DisplayName','P offset');

h(4).hp(end+1) = plot(h(4).ha, time(iQon),  ecg(iQon),  'k^',
'MarkerFaceColor','k', 'LineStyle','none', 'DisplayName','Q onset');
h(4).hp(end+1) = plot(h(4).ha, time(iQpk),  ecg(iQpk),  'b^',
'MarkerFaceColor','b', 'LineStyle','none', 'DisplayName','Q peak');

h(4).hp(end+1) = plot(h(4).ha, time(iRpk),  ecg(iRpk),  'rv',
'MarkerFaceColor','r', 'LineStyle','none', 'DisplayName','R peak');

h(4).hp(end+1) = plot(h(4).ha, time(iSpk),  ecg(iSpk),  'cv',
'MarkerFaceColor','c', 'LineStyle','none', 'DisplayName','S peak');
h(4).hp(end+1) = plot(h(4).ha, time(iSoff), ecg(iSoff), 'mv',
'MarkerFaceColor','m', 'LineStyle','none', 'DisplayName','S offset');

h(4).hp(end+1) = plot(h(4).ha, time(iTon),  ecg(iTon),  'ks',
'MarkerFaceColor','k', 'LineStyle','none', 'DisplayName','T onset');
h(4).hp(end+1) = plot(h(4).ha, time(iTpk),  ecg(iTpk),  'rs',
'MarkerFaceColor','r', 'LineStyle','none', 'DisplayName','T peak');
h(4).hp(end+1) = plot(h(4).ha, time(iToff), ecg(iToff), 'gs',
'MarkerFaceColor','g', 'LineStyle','none', 'DisplayName','T offset');

% Right axis line
yyaxis(h(4).ha,'right');
hold(h(4).ha,'on');
grid(h(4).ha,'on');
h(4).hp(end+1) = plot(h(4).ha, t_env, env, 'DisplayName','Resp. envelope');

% Labels/titles: these return handles
h(4).ht = title(h(4).ha, 'ECG and respiratory envelope with delineation');
h(4).hx = xlabel(h(4).ha, 'Time [s]');
yyaxis(h(4).ha,'left');
h(4).hy(1) = ylabel(h(4).ha, 'ECG [mV]');
yyaxis(h(4).ha,'right');
h(4).hy(2) = ylabel(h(4).ha, 'Envelope (a.u.)');

h(4).hl = legend(h(4).ha, 'show', 'Location','best');

xlim(h(4).ha, [0 10]);
```



4.o (0.5P) Effect of breathing on ECG signal (comment).

Breathing mainly affects the ECG through 1) baseline wander (low-frequency drift) 2) amplitude modulation of the QRS complexes (ECG-derived respiration / EDR), due to changes in thoracic impedance and heart orientation during inhalation/exhalation. Additionally, breathing can cause slight heart-rate variability (respiratory sinus arrhythmia).

4.p (0.5P) Effect of apnoea on ECG signal (comment).

During apnoea (temporary cessation of breathing), the respiration-related modulation decreases: baseline wander and QRS-amplitude modulation are reduced and the EDR envelope becomes flatter. Respiratory sinus arrhythmia also tends to diminish (less HR modulation).

4.q (2P) Compare respiratory frequency with cardiac component; report ranges.

```
freqOut = compareRespCardiacFreq(resp, Fs, R_peak, false); %checked

% Report key results in the command window

fprintf('\n==== Frequency comparison ====\n');
fprintf('Respiration dominant frequency (Welch): %.3f Hz (%.1f breaths/min)
\n', ...
    freqOut.resp_Hz_dominant, 60*freqOut.resp_Hz_dominant);
fprintf('Respiration frequency range (time-domain): [%.3f, %.3f] Hz\n', ...
    freqOut.resp_Hz_range(1), freqOut.resp_Hz_range(2));
```

```
fprintf('Cardiac mean frequency (RR): %.3f Hz (%.1f bpm)\n', ...
    freqOut.card_Hz_mean, 60*freqOut.card_Hz_mean);
fprintf('Cardiac frequency range (RR): [%.3f, %.3f] Hz\n', ...
    freqOut.card_Hz_range(1), freqOut.card_Hz_range(2));

% Typical ranges (for comparison):
%   Respiration: ~0.1-0.5 Hz (6-30 breaths/min)
%   Cardiac:     ~0.8-2.0 Hz (48-120 bpm)
```

```
==== Frequency comparison ====
Respiration dominant frequency (Welch): 0.293 Hz (17.6 breaths/min)
Respiration frequency range (time-domain): [0.176, 0.806] Hz
Cardiac mean frequency (RR): 1.232 Hz (73.9 bpm)
Cardiac frequency range (RR): [0.735, 1.610] Hz
```

Published with MATLAB® R2025b