EXPERIMENT 3

AIM: DETECTION OF QRS COMPLEX IN ECG

SOFTWARE: MATLAB

THOERY:

The QRS complex in an ECG represents ventricular depolarization and is crucial for determining heart rate and detecting abnormalities. This experiment employs the Pan-Tompkins algorithm for QRS detection using signal processing techniques.

The ECG signal is first preprocessed using a bandpass filter (0.5–50 Hz) to remove noise. The signal is then differentiated to highlight rapid changes, squared to enhance peak prominence, and passed through a moving window integrator (MWI) to smoothen the waveform. Peak detection is performed using an adaptive threshold to locate the R-peaks. The heart rate is calculated by measuring the intervals between consecutive R-peaks.

This method ensures accurate QRS detection, enabling real-time heart rate monitoring and cardiac analysis.

CODES:

clc; clear all; close all;

%Load ECG Signal ( select only one channel )

[ ecg\_signal, fs ] = rdsamp('s0016lre');

%check if ecg\_signal has multiple columns

if size(ecg\_signal,2) > 1

ecg\_signal = ecg\_signal(:,1); %Select the first channel

end

%Display basic info

disp(['Sampling Frequency: ', num2str(fs), ' Hz']);

disp(['Signal Length: ', num2str(length(ecg\_signal)), ' samples ']);

%Plot ECG Signal

t = (0:length(ecg\_signal)-1) / fs; %Time axis

figure;

plot(t,ecg\_signal);

xlabel('Time(s)');

ylabel('Amplitude');

title('Raw ECG Signal from PhysioNet');

grid on;

%Step 1: Bandpass Filtering (0.5 - 50 Hz)

low\_freq = 0.5;

high\_freq = 50;

[b, a] = butter(3, [low\_freq high\_freq] / (fs/2), 'bandpass'); %Bandpass Butterworth Filter

filtered\_ecg = filtfilt(b, a, ecg\_signal);

%Step 2: Differentiation

diff\_ecg = diff(filtered\_ecg);

%Step 3: Squaring

squared\_ecg = diff\_ecg.^2;

%Step 4: Moving Window Integration

window\_size = round(0.150 \* fs); % 150ms window

window = ones(1, window\_size) / window\_size;

mid\_ecg = filter(window, 1, squared\_ecg);

%Ensure mid\_ecg is a vector

mwi\_ecg = mid\_ecg(:); %Convert to column vector if needed

%Step 5: Peak Detection with Adaptive Threshold

threshold = 0.6\*max(mwi\_ecg); %Initial Threshold

[~, r\_peaks] = findpeaks(mwi\_ecg, 'MinPeakHeight', threshold, 'MinPeakDistance', round(0.6\*fs));

%Plot Results

figure;

subplot(3,1,1); plot(ecg\_signal); title('Original ECG Signal');

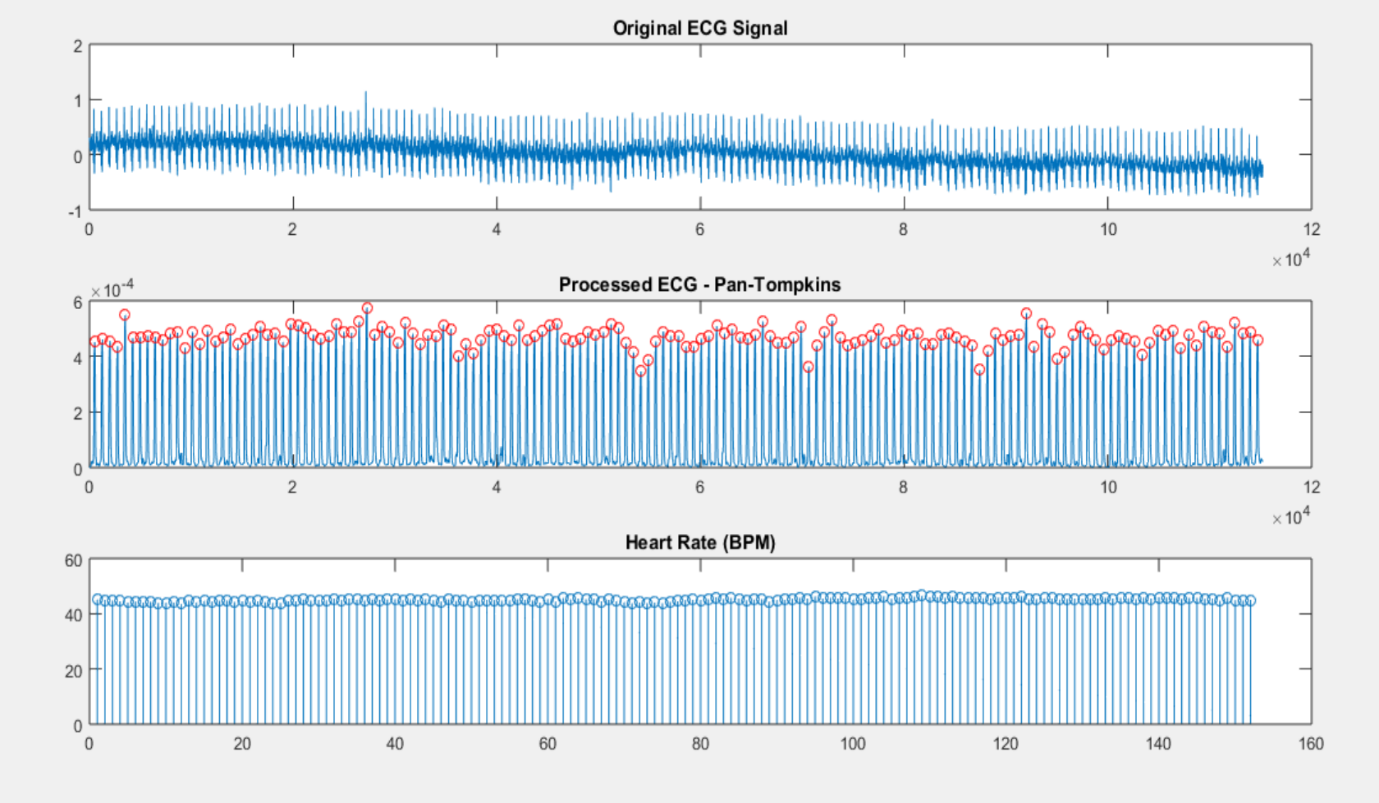
subplot(3,1,2); plot(mwi\_ecg); title('Processed ECG - Pan-Tompkins');

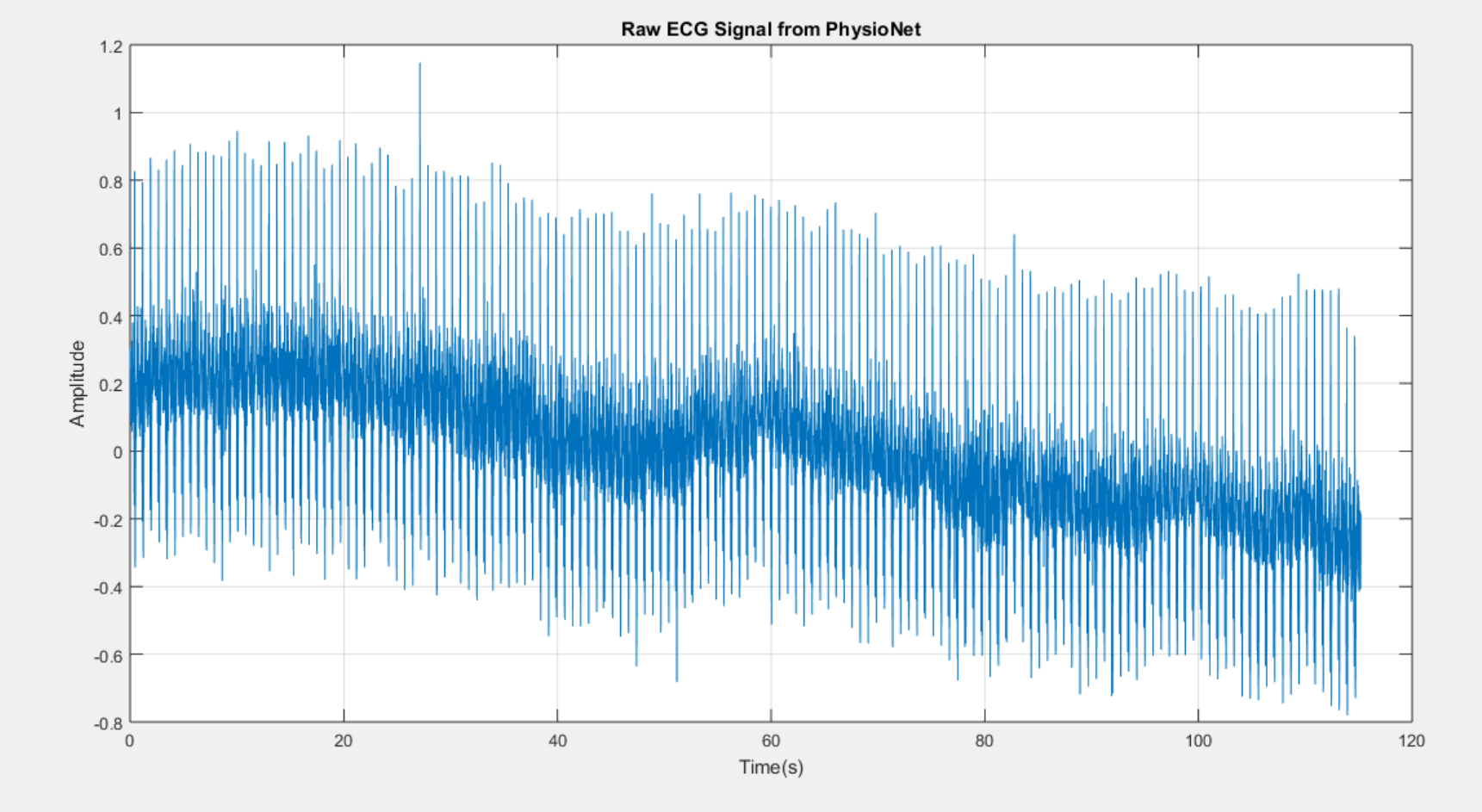
hold on; plot(r\_peaks, mwi\_ecg(r\_peaks), 'ro'); hold off;

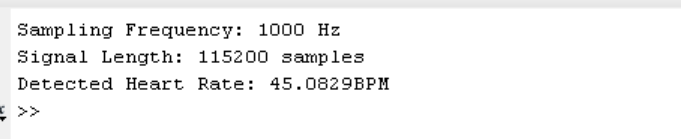
subplot(3,1,3); stem(diff(r\_peaks)/ fs\*60); title('Heart Rate (BPM)');

disp(['Detected Heart Rate: ', num2str(mean(diff(r\_peaks) / fs\*60)), 'BPM']);

OUTPUT:







CONCLUTION:

The detection of QRS complexes in an ECG signal is crucial for cardiac monitoring and diagnostics. By using bandpass filtering, differentiation, squaring, moving window integration, and adaptive peak detection, the Pan-Tompkins algorithm successfully identifies QRS complexes and estimates heart rate. This method is efficient, robust, and widely used in real-time ECG analysis and wearable health monitoring devices.