Link: <https://biomedicalsignalandimage.blogspot.in/2016/02/matlab-code-to-plot-ecg-signal.html>

<https://physionet.org/cgi-bin/atm/ATM>

<http://www.hrsonline.org/Patient-Resources/Heart-Diseases-Disorders?gclid=Cj0KEQjw2-bHBRDEh6qk5b6yqKIBEiQAFUz29tinnjI8vyptP97IXCW8Yf7zunI08S35wlBZ_YbaGowaAjne8P8HAQ>

<http://www.ambulancetechnicianstudy.co.uk/ecgbasics.html>

<https://www.memorialcare.org/services/glossary/a/abnormal-heart-rhythms-arrhythmias>

Matlab Code to plot ECG signal

close all;

clear all;

clc;

%

%S=load('I:\BIOM\_Signal\_processing\Hw5\ECGsignal\_1.mat')

y1=xlsread('J:\BIOM\_Signal\_processing\Hw5\ECGsignal\_1.xls');

fs = 250 % find the sampling rate or frequency

T = 1/fs;% sampling rate or frequency

% find the length of the data per second

N = length(y1);

ls = size(y1);

t = (0 : N-1)/fs;% sampling period

figure; %subplot(1,2,1);

%subplot(1,2,1);

plot(t,y1);

title ('plot of the orignal ECG signal');

xlabel ('time (sec)');

ylabel ('ECG Amplitute (mv)');

grid on;

figure;

%%% Create a period

y1new = y1(10:350);

ax = axis; axis([ax(1:2) -3.2 3.2])

t2 = (0 : length (y1new)-1)/fs;% sampling period

%subplot(1,2,2);

plot (y1new);

title ('plot one typical period of the signal amplitude spectrume of the  ECG signal');

xlabel ('time (msec)');

ylabel ('ECG Amplitute  (mv)');

figure;

%%% Create a period

y1new1 = y1(40:200);

ax = axis; axis([ax(1:2) -3.2 3.2])

t2 = (0 : length (y1new)-1)/fs;% sampling period

%subplot(1,2,2);

plot (y1new1);

title ('plot PQRST cycle of the ECG signal');

xlabel ('time (msec)');

ylabel ('ECG Amplitute  (mv)');

%% heart rate analysis

% count the dominat peak

beat\_count =0;

for k = 2 : length(y1)-1

    %the peak has to be greater than 1 and greater than the value before it and greater then the value after it.

    if(y1(k)> y1(k-1) && y1(k) > y1(k+1) && y1(k)> 1)

         beat\_count = beat\_count +1;

    end

end

display (k);

disp('dominant peaks');

%% divide the peak count by the duration in minute

duration\_in\_sec = N/fs;

duration\_in\_minute = duration\_in\_sec/60;

BPM = beat\_count/duration\_in\_minute;

Matlab Code to study signal

%%Calculate HR

close all; clear all; clc;

fs = 250 % find the sampling rate or frequency

T = 1/fs;% sampling rate or frequency

window = 120; % 2 min 0r 120 second

%%Select a filename in .mat format and load the file.

fignum = 0;

load('J:\BIOM\_Signal\_processing\exam1\ECGsignalTest\_1.mat') % contains hr\_sig and fs

% Make time axis for ECG signal

tx = [0:length(hr\_sig)-1]/fs;

fignum = fignum + 1;

figure(fignum)

plot(tx,hr\_sig)

xlabel('Time (s)')

ylabel('Amplitude (mV)')

title('ECG signal')

xlim([30.3,31]) % Used to zoom in on single ECG waveform

disp('Contents of workspace after loading file:')

%whos

%% copy the data and put into excel

y1=xlsread('J:\BIOM\_Signal\_processing\exam1\ecg\_1.xls');

% find the length of the data per second

N = length(y1);

ls = size(y1);

t = (0 : N-1) / fs;% sampling period

fignum = fignum + 1; %% keep track of figures

figure(fignum)

plot(t,y1);

title ('plot of the original of ECG signal')

xlabel ('time (sec)')

ylabel ('Amplitute (mv)')

grid on;

%% find PP interval

 i = 0;  %% to make the code start from 0.

 rr = 0; %% each time the code run, rr distance two peaks

 hold off % for the next graph

 rrinterval = zeros(3600,1); % create an array to strore 2 peaks

beat\_count =0;

for k = 2 : length(y1)-1

    %the peak has to be greater than 1 and greater than the value before it and greater then the value after it.

    if(y1(k)> y1(k-1) && y1(k) > y1(k+1) && y1(k)> 1);

     beat\_count = beat\_count +1;

     if beat\_count ==1;

        rr =0;

     else

         rr = k-i;

         rrinterval(k)=rr;

         i=k;

     end

    else

        rrinterval(k)= rr;

    end

end

%% heart rate analysis

% count the dominat peak

beat\_count =0;

for k = 2 : length(y1)-1

    %the peak has to be greater than 1 and greater than the value before it and greater then the value after it.

    if(y1(k)> y1(k-1) && y1(k) > y1(k+1) && y1(k)> 1)

         beat\_count = beat\_count +1;

    end

end

display (k);

disp('dominant peaks');

%% this section is calculate heart rate of the ECG

%% divide the peak count by the duration in minute

duration\_in\_sec = N/fs;

duration\_in\_minute = duration\_in\_sec/60;

BPM = beat\_count/duration\_in\_minute; %% this is calculation heart rate

msgbox(strcat('Heart-rate is = ',mat2str(BPM),' BPM'));

%Compute the spectrum of the ECG

%b) Compute the spectrum of the ECG and provide remarks on the spectral features of the ECG ( see reference “ECG Statistics, Noise, Artifacts, and Missing Data”).

%%%  DFT to describe the signal in the frequency

NFFT = 2 ^ nextpow2(N);

Y = fft(y1, NFFT) / N;

f = (fs / 2 \* linspace(0, 1, NFFT / 2+1))'; % Vector containing frequencies in Hz

amp = ( 2 \* abs(Y(1: NFFT / 2+1))); % Vector containing corresponding amplitudes

figure;

plot (f, amp);

title ('plot single-sided amplitude spectrume of the ECG signal');

xlabel ('frequency (Hz)');

ylabel ('|y(f)|');

grid on;

figure;

psdest = psd(spectrum.periodogram,y1,'fs',fs);

plot(psdest)

title ('plot single-sided PSD of the ECG signal');

xlabel ('frequency (Hz)');

ylabel ('|y(f)|');

grid on;

figure;

psdest1 = psd(spectrum.periodogram,y1,'NFFT',length(y1),'Fs',fs);

plot(psdest1)

avgpower(psdest1,[58,62]);

title ('plot single-sided PSD of the ECG signal')

xlabel ('frequency (Hz)');

ylabel ('|y(f)|');

grid on;

max\_value=max(y1);

mean\_value=mean(y1);

threshold=(max\_value-mean\_value)/2;

%% create a subset to zoom into the signal make easy to verify mark position

y1\_1500 = y1(1:1850);

t2 = 1:length(y1\_1500);

figure;

plot(t2,y1\_1500);

title ('plot of subset of the ECG signal')

xlabel ('time (msec)')

ylabel ('Amplitute (mv)')

grid on

%c) Write code to automatically detect the various features of the ECG (PQRST) and use that to mark the ECG waveform features

%% used the snip code from this website.

%%%%http://www.mathworks.com/help/signal/examples/peak-analysis.html

%Detrending Data

%The above signal shows a baseline shift and therefore does not represent the true amplitude. In order to remove the trend, fit a low order polynomial to the signal and use the polynomial to detrend it.

[p,s,mu] = polyfit((1:numel(y1\_1500))',y1\_1500,6);

f\_y = polyval(p,(1:numel(y1\_1500))',[],mu);

ECG\_data = y1\_1500 - f\_y;        % Detrend data

N1= length (y1\_1500);

t1 = (0 : N1-1) / fs;% sampling period

figure

%plot(t1,ECG\_data); grid on

plot(t2,ECG\_data); grid on

ax = axis; axis([ax(1:2) -2.2 2.2])

%ax = axis; axis([ax(1:2) -3.2 3.2])

title('Detrended ECG Signal')

xlabel('time msec'); ylabel('Voltage(mV)')

legend('Detrended ECG Signal')

%Thresholding to Find Peaks of Interest

%The QRS-complex consists of three major components: Q-wave, R-wave, S-wave. The R-waves can be detected by thresholding peaks above 0.5mV. Notice that the R-waves are separated by more than 200 samples. Use this information to remove unwanted peaks by specifying a 'MinPeakDistance'.

[~,locs\_Rwave] = findpeaks(ECG\_data,'MinPeakHeight',0.5,...

                                    'MinPeakDistance',120);

%Finding Local Minima in Signal

%Local minima can be detected by finding peaks on an inverted version of the original signal.

ECG\_inverted = -ECG\_data;

[~,locs\_Swave] = findpeaks(ECG\_inverted,'MinPeakHeight',0.4,...

                                       'MinPeakDistance',120);

%The following plot shows the R-waves and S-waves detected in the signal.

figure

hold on

plot(t2,ECG\_data);

plot(locs\_Rwave,ECG\_data(locs\_Rwave),'rv','MarkerFaceColor','r');

plot(locs\_Swave,ECG\_data(locs\_Swave),'rs','MarkerFaceColor','b');

%axis([0 1850 -1.1 1.1]); grid on;

axis([0 1850 -2.2 2.2]); grid on;

legend('ECG Signal','R-waves','S-waves');

xlabel('time msec'); ylabel('Voltage(mV)')

title('R-wave and S-wave in ECG Signal')

[~,locs\_Twave] = findpeaks(ECG\_data,'MinPeakHeight',-0.02,...

                                      'MinPeakDistance',40);

%% The following code detect and mark T

figure;

hold on

plot(t2,ECG\_data);

plot(locs\_Twave,ECG\_data(locs\_Twave),'X','MarkerFaceColor','y');

plot(locs\_Rwave,ECG\_data(locs\_Rwave),'rv','MarkerFaceColor','r');

plot(locs\_Swave,ECG\_data(locs\_Swave),'rs','MarkerFaceColor','b');

grid on

title('Thresholding Peaks in Signal')

xlabel('time msec'); ylabel('Voltage(mV)')

ax = axis; axis([0 1850 -2.2 2.2])

legend('ECG signal','T-wave','R-wave','S-wave');

Matlab code to detect and mark PQRST on your ECG

close all;

clear all;

clc;

%%Select a filename in .mat format and load the file.

%[fname path]=uigetfile('\*.mat');

%fname=strcat(path,fname);

%y1 = load(fname );

%file =load('I:\BIOM\_Signal\_processing\Hw5\ECGsignal\_1.mat')

load('I:\BIOM\_Signal\_processing\Hw5\ECGsignal\_1.mat')

disp('Contents of workspace after loading file:')

whos

fs = 250; % find the sampling rate or frequency

y1=xlsread('I:\BIOM\_Signal\_processing\Hw5\ECGsignal\_1.xls');

T = 1/fs;% sampling rate or frequency

% find the length of the data per second

N = length(y1);

ls = size(y1);

t = (0 : N-1) / fs;% sampling period

%t = (0 : N-1) \*T;

%t = (0:1:length(y1)-1)/fs;

%subplot (2,2,2)

%plot (t,data);

figure; %subplot(1,2,1);

      plot(t,y1);

    %plot(x,y2, 'g');

title ('plot of the original of ECG signal')

xlabel ('time (sec)')

ylabel ('Amplitute (mv)')

grid on;

y1\_n=(y1-min(y1))/(max(y1)-min(y1));         % normalize between 0-1

fnyquist = fs/2;

%% find P

m1=max(y1)\*.60;

P=find(y1>=m1);

y1\_1500 = y1(1:1850);

t2 = 1:length(y1\_1500);

figure;

plot(t2,y1\_1500);

title ('plot of subset of the ECG signal')

xlabel ('time (msec)')

ylabel ('Amplitute (mv)')

grid on

%% used the snip code from this website.

%%%%http://www.mathworks.com/help/signal/examples/peak-analysis.html

%Detrending Data

%The above signal shows a baseline shift and therefore does not represent the true amplitude. In order to remove the trend, fit a low order polynomial to the signal and use the polynomial to detrend it.

[p,s,mu] = polyfit((1:numel(y1\_1500))',y1\_1500,6);

f\_y = polyval(p,(1:numel(y1\_1500))',[],mu);

ECG\_data = y1\_1500 - f\_y;        % Detrend data

N1= length (y1\_1500);

t1 = (0 : N1-1) / fs;% sampling period

figure

%plot(t1,ECG\_data); grid on

plot(t2,ECG\_data); grid on

ax = axis; axis([ax(1:2) -2.2 2.2])

%ax = axis; axis([ax(1:2) -3.2 3.2])

title('Detrended ECG Signal')

xlabel('time msec'); ylabel('Voltage(mV)')

legend('Detrended ECG Signal')

%Thresholding to Find Peaks of Interest

%The QRS-complex consists of three major components: Q-wave, R-wave, S-wave. The R-waves can be detected by thresholding peaks above 0.5mV. Notice that the R-waves are separated by more than 200 samples. Use this information to remove unwanted peaks by specifying a 'MinPeakDistance'.

[~,locs\_Rwave] = findpeaks(ECG\_data,'MinPeakHeight',0.5,...

                                    'MinPeakDistance',120);

%Finding Local Minima in Signal

%Local minima can be detected by finding peaks on an inverted version of the original signal.

ECG\_inverted = -ECG\_data;

[~,locs\_Swave] = findpeaks(ECG\_inverted,'MinPeakHeight',0.4,...

                                       'MinPeakDistance',120);

%The following plot shows the R-waves and S-waves detected in the signal.

figure

hold on

plot(t2,ECG\_data);

plot(locs\_Rwave,ECG\_data(locs\_Rwave),'rv','MarkerFaceColor','r');

plot(locs\_Swave,ECG\_data(locs\_Swave),'rs','MarkerFaceColor','b');

%axis([0 1850 -1.1 1.1]); grid on;

axis([0 1850 -2.2 2.2]); grid on;

legend('ECG Signal','R-waves','S-waves');

xlabel('time msec'); ylabel('Voltage(mV)')

title('R-wave and S-wave in ECG Signal')

[~,locs\_Twave] = findpeaks(ECG\_data,'MinPeakHeight',-0.02,...

                                      'MinPeakDistance',50);

figure;

hold on

plot(t2,ECG\_data);

plot(locs\_Twave,ECG\_data(locs\_Twave),'X','MarkerFaceColor','y');

plot(locs\_Rwave,ECG\_data(locs\_Rwave),'rv','MarkerFaceColor','r');

plot(locs\_Swave,ECG\_data(locs\_Swave),'rs','MarkerFaceColor','b');

grid on

title('Thresholding Peaks in Signal')

xlabel('time msec'); ylabel('Voltage(mV)')

ax = axis; axis([0 1850 -2.2 2.2])

legend('ECG signal','T-wave','R-wave','S-wave');

[~,locs\_Pwave] = findpeaks(ECG\_data,'MinPeakHeight',-0.09,...

                                      'MinPeakDistance',25);

figure;

hold on

plot(t2,ECG\_data);

plot(locs\_Pwave,ECG\_data(locs\_Pwave),'x','MarkerFaceColor','y');

plot(locs\_Twave,ECG\_data(locs\_Twave),'X','MarkerFaceColor','g');

plot(locs\_Rwave,ECG\_data(locs\_Rwave),'rv','MarkerFaceColor','r');

plot(locs\_Swave,ECG\_data(locs\_Swave),'rs','MarkerFaceColor','b');

grid on

title('Thresholding Peaks in Signal')

xlabel('time msec'); ylabel('Voltage(mV)')

ax = axis; axis([0 1850 -2.2 2.2])

legend('ECG signal','P-wave','T-wave','R-wave','S-wave');

[~,locs\_qwave] = findpeaks(ECG\_data,'MinPeakHeight',-0.2);

figure;

hold on

plot(t2,ECG\_data);

plot(locs\_qwave,ECG\_data(locs\_qwave),'x','MarkerFaceColor','y');

% link and zoom in to show the changes

%linkaxes(ax(1:2),'xy');

%axis(ax,[60 230 0.006 -0.04])

%Next, we try and determine the locations of the Q-waves. Thresholding the peaks to locate the Q-waves results in detection of unwanted peaks as the Q-waves are buried in noise. We filter the signal first and then find the peaks. Savitzky-Golay filtering is used to remove noise in the signal.

smoothECG = sgolayfilt(ECG\_data,1,3);

figure

plot(t2,ECG\_data,'b',t2,smoothECG,'r'); grid on

axis tight;

xlabel('time msec'); ylabel('Voltage(mV)');

legend('ECG Signal','Filtered Signal')

title('Filtering Noisy ECG Signal')

%We perform peak detection on the smooth signal and use logical indexing to find the locations of the Q-waves.

%[~,min\_locs] = findpeaks(-smoothECG,'MinPeakDistance',29);

%[~,min\_locs] = findpeaks(smoothECG,'MinPeakDistance',2);%Twave

[~,min\_locs] = findpeaks(smoothECG,'MinPeakDistance',50);

% Peaks between -0.2mV and -0.5mV

%locs\_Qwave = min\_locs(smoothECG(min\_locs)>-0.3 &

%-smoothECG(min\_locs)<-0.1); %Twave

locs\_Qwave = min\_locs(smoothECG(min\_locs)>-0.3 & -smoothECG(min\_locs)<-0.11);

figure

hold on

plot(t2,smoothECG);

plot(locs\_Qwave,smoothECG(locs\_Qwave),'rs','MarkerFaceColor','g');

plot(locs\_Rwave,smoothECG(locs\_Rwave),'rv','MarkerFaceColor','r');

plot(locs\_Swave,smoothECG(locs\_Swave),'rs','MarkerFaceColor','b');

grid on

title('Thresholding Peaks in Signal')

xlabel('time msec'); ylabel('Voltage(mV)')

ax = axis; axis([0 1850 -2.2 2.2])

legend('Smooth ECG signal','T-interval','R-wave','S-wave');

%The above figure shows that the QRS-complex successfully detected in the noisy ECG signal.

%Error Between Noisy and Smooth Signal

%Notice the average difference between the QRS-complex in the raw and the detrended filtered signal.

% Values of the Extrema

[val\_Qwave, val\_Rwave, val\_Swave] = deal(smoothECG(locs\_Qwave), smoothECG(locs\_Rwave), smoothECG(locs\_Swave));

meanError\_Qwave = mean((y1\_1500(locs\_Qwave) - val\_Qwave))

meanError\_Rwave = mean((y1\_1500(locs\_Rwave) - val\_Rwave))

meanError\_Swave = mean((y1\_1500(locs\_Swave) - val\_Swave))

%% find PP interval

 i = 0;  %% to make the code start from 0.

 rr = 0; %% each time the code run, rr distance two peaks

 hold off % for the next graph

 rrinterval = zeros(3600,1); % create an array to strore 2 peaks

beat\_count =0;

for k = 2 : length(y1)-1

    %the peak has to be greater than 1 and greater than the value before it and greater then the value after it.

    if(y1(k)> y1(k-1) && y1(k) > y1(k+1) && y1(k)> 1);

     beat\_count = beat\_count +1;

     if beat\_count ==1;

        rr =0;

     else

         rr = k-i;

         rrinterval(k)=rr;

         i=k;

     end

    else

        rrinterval(k)= rr;

    end

end

figure;

plot (rrinterval);

xlabel('Time in sec\*10^-2'), ylabel('Distance betweeen 2 Heatbeats (R-R) in sec\*10^-2'), title('R-R intervals');

%% find PP interval

%% heart rate analysis

% count the dominat peak

beat\_count =0;

for k = 2 : length(y1)-1

    %the peak has to be greater than 1 and greater than the value before it and greater then the value after it.

    if(y1(k)> y1(k-1) && y1(k) > y1(k+1) && y1(k)> 1)

         beat\_count = beat\_count +1;

    end

end

display (k);

disp('dominant peaks');

%% divide the peak count by the duration in minute

duration\_in\_sec = N/fs;

duration\_in\_minute = duration\_in\_sec/60;

BPM = beat\_count/duration\_in\_minute;

%%%  DFT to describe the signal in the frequency

NFFT = 2 ^ nextpow2(N);

Y = fft(y1, NFFT) / N;

f = (fs / 2 \* linspace(0, 1, NFFT / 2+1))'; % Vector containing frequencies in Hz

amp = ( 2 \* abs(Y(1: NFFT / 2+1))); % Vector containing corresponding amplitudes

figure;

plot (f, amp);

title ('plot single-sided amplitude spectrume of the ECG signal')

xlabel ('frequency (Hz)')

ylabel ('|y(f)|')

grid on;

max\_value=max(y1);

mean\_value=mean(y1);

threshold=(max\_value-mean\_value)/2;

%% downsampling ½ sample frequency

close all;

clear all;

clc;

%%Select a filename in .mat format and load the file.

%[fname path]=uigetfile('\*.mat');

%fname=strcat(path,fname);

%y1 = load(fname );

%file =load('I:\BIOM\_Signal\_processing\Hw5\ECGsignal\_1.mat')

load('I:\BIOM\_Signal\_processing\Hw5\ECGsignal\_1.mat')

disp('Contents of workspace after loading file:')

whos

fs = 250; % find the sampling rate or frequency

fs2 = 250\*1/2;

y1=xlsread('I:\BIOM\_Signal\_processing\Hw5\ECGsignal\_1.xls');

T = 1/fs;% sampling rate or frequency

% find the length of the data per second

N = length(y1);

ls = size(y1);

t = (0 : N-1) / fs;% sampling period

%t = (0 : N-1) \*T;

%t = (0:1:length(y1)-1)/fs;

%subplot (2,2,2)

%plot (t,data);

figure; %subplot(1,2,1);

      plot(t,y1);

    %plot(x,y2, 'g');

title ('plot of the original of ECG signal')

xlabel ('time (sec)')

ylabel ('Amplitute (mv)')

grid on;

%%%%%%%%%%%%%

% down sampling 1/2 of frequency sample

y2 = resample(y1,fs2,fs);

N2 = length(y2);

ls2 = size(y2);

t22 = (0 : N2-1) / fs2;% sampling period

figure; %subplot(1,2,1);

      plot(t22,y2);

title ('plot of the down sampling 1/2 frequency sample of ECG signal')

xlabel ('time (sec)')

ylabel ('Amplitute (mv)')

grid on;

%y1\_n=(y1-min(y1))/(max(y1)-min(y1));         % normalize between 0-1

fnyquist = fs2/2;

%% find P

m1=max(y2)\*.60;

P=find(y2>=m1);

y1\_1500 = y2(1:1850);

t2 = 1:length(y1\_1500);

figure;

plot(t2,y1\_1500);

title ('plot of subset of down sampling 1/2 frequency sample the ECG signal')

xlabel ('time (msec)')

ylabel ('Amplitute (mv)')

grid on

%% used the snip code from this website.

%%%%http://www.mathworks.com/help/signal/examples/peak-analysis.html

%Detrending Data

%The above signal shows a baseline shift and therefore does not represent the true amplitude. In order to remove the trend, fit a low order polynomial to the signal and use the polynomial to detrend it.

[p,s,mu] = polyfit((1:numel(y1\_1500))',y1\_1500,6);

f\_y = polyval(p,(1:numel(y1\_1500))',[],mu);

ECG\_data = y1\_1500 - f\_y;        % Detrend data

N1= length (y1\_1500);

t1 = (0 : N1-1) / fs2;% sampling period

figure

%plot(t1,ECG\_data); grid on

plot(t2,ECG\_data); grid on

ax = axis; axis([ax(1:2) -2.2 2.2])

%ax = axis; axis([ax(1:2) -3.2 3.2])

title('Detrended down sampling 1/2 frequency sample ECG Signal')

xlabel('time msec'); ylabel('Voltage(mV)')

legend('Detrended ECG Signal')

%Thresholding to Find Peaks of Interest

%The QRS-complex consists of three major components: Q-wave, R-wave, S-wave. The R-waves can be detected by thresholding peaks above 0.5mV. Notice that the R-waves are separated by more than 200 samples. Use this information to remove unwanted peaks by specifying a 'MinPeakDistance'.

[~,locs\_Rwave] = findpeaks(ECG\_data,'MinPeakHeight',0.5,...

                                    'MinPeakDistance',60);

%Finding Local Minima in Signal

%Local minima can be detected by finding peaks on an inverted version of the original signal.

ECG\_inverted = -ECG\_data;

[~,locs\_Swave] = findpeaks(ECG\_inverted,'MinPeakHeight',0.4,...

                                       'MinPeakDistance',60);

%The following plot shows the R-waves and S-waves detected in the signal.

figure

hold on

plot(t2,ECG\_data);

plot(locs\_Rwave,ECG\_data(locs\_Rwave),'rv','MarkerFaceColor','r');

plot(locs\_Swave,ECG\_data(locs\_Swave),'rs','MarkerFaceColor','b');

%axis([0 1850 -1.1 1.1]); grid on;

axis([0 1850 -2.2 2.2]); grid on;

legend('ECG Signal','R-waves','S-waves');

xlabel('time msec'); ylabel('Voltage(mV)')

title('R-wave and S-wave in down sampling 1/2 frequency sample of ECG Signal')

[~,locs\_Twave] = findpeaks(ECG\_data,'MinPeakHeight',-0.02,...

                                      'MinPeakDistance',25);

figure;

hold on

plot(t2,ECG\_data);

plot(locs\_Twave,ECG\_data(locs\_Twave),'X','MarkerFaceColor','y');

plot(locs\_Rwave,ECG\_data(locs\_Rwave),'rv','MarkerFaceColor','r');

plot(locs\_Swave,ECG\_data(locs\_Swave),'rs','MarkerFaceColor','b');

grid on

title('Thresholding Peaks in down sampling 1/2 frequency sample Signal')

xlabel('time msec'); ylabel('Voltage(mV)')

ax = axis; axis([0 1850 -2.2 2.2])

legend('ECG signal','T-wave','R-wave','S-wave');

[~,locs\_Pwave] = findpeaks(ECG\_data,'MinPeakHeight',-0.09,...

                                      'MinPeakDistance',12);

figure;

hold on

plot(t2,ECG\_data);

plot(locs\_Pwave,ECG\_data(locs\_Pwave),'x','MarkerFaceColor','y');

plot(locs\_Twave,ECG\_data(locs\_Twave),'X','MarkerFaceColor','g');

plot(locs\_Rwave,ECG\_data(locs\_Rwave),'rv','MarkerFaceColor','r');

plot(locs\_Swave,ECG\_data(locs\_Swave),'rs','MarkerFaceColor','b');

grid on

title('Thresholding Peaks in down sampling 1/2 frequency sample Signal')

xlabel('time msec'); ylabel('Voltage(mV)')

ax = axis; axis([0 1850 -2.2 2.2])

legend('ECG signal','P-wave','T-wave','R-wave','S-wave');

[~,locs\_qwave] = findpeaks(ECG\_data,'MinPeakHeight',-0.2);

figure;

hold on

plot(t2,ECG\_data);

plot(locs\_qwave,ECG\_data(locs\_qwave),'x','MarkerFaceColor','y');

% link and zoom in to show the changes

%linkaxes(ax(1:2),'xy');

%axis(ax,[60 230 0.006 -0.04])

%Next, we try and determine the locations of the Q-waves. Thresholding the peaks to locate the Q-waves results in detection of unwanted peaks as the Q-waves are buried in noise. We filter the signal first and then find the peaks. Savitzky-Golay filtering is used to remove noise in the signal.

smoothECG = sgolayfilt(ECG\_data,1,3);

figure

plot(t2,ECG\_data,'b',t2,smoothECG,'r'); grid on

axis tight;

xlabel('time msec'); ylabel('Voltage(mV)');

legend('ECG Signal','Filtered Signal')

title('Filtering Noisy of down sampling 1/2 frequency sample ECG Signal')

%We perform peak detection on the smooth signal and use logical indexing to find the locations of the Q-waves.

%[~,min\_locs] = findpeaks(-smoothECG,'MinPeakDistance',29);

%[~,min\_locs] = findpeaks(smoothECG,'MinPeakDistance',2);%Twave

[~,min\_locs] = findpeaks(smoothECG,'MinPeakDistance',25);

% Peaks between -0.2mV and -0.5mV

%locs\_Qwave = min\_locs(smoothECG(min\_locs)>-0.3 &

%-smoothECG(min\_locs)<-0.1); %Twave

locs\_Qwave = min\_locs(smoothECG(min\_locs)>-0.3 & -smoothECG(min\_locs)<-0.11);

figure

hold on

plot(t2,smoothECG);

plot(locs\_Qwave,smoothECG(locs\_Qwave),'rs','MarkerFaceColor','g');

plot(locs\_Rwave,smoothECG(locs\_Rwave),'rv','MarkerFaceColor','r');

plot(locs\_Swave,smoothECG(locs\_Swave),'rs','MarkerFaceColor','b');

grid on

title('Thresholding Peaks down sampling 1/2 frequency sample in Signal')

xlabel('time msec'); ylabel('Voltage(mV)')

ax = axis; axis([0 1850 -2.2 2.2])

legend('Smooth ECG signal','T-interval','R-wave','S-wave');

%The above figure shows that the QRS-complex successfully detected in the noisy ECG signal.

%Error Between Noisy and Smooth Signal

%Notice the average difference between the QRS-complex in the raw and the detrended filtered signal.

% Values of the Extrema

[val\_Qwave, val\_Rwave, val\_Swave] = deal(smoothECG(locs\_Qwave), smoothECG(locs\_Rwave), smoothECG(locs\_Swave));

meanError\_Qwave = mean((y1\_1500(locs\_Qwave) - val\_Qwave))

meanError\_Rwave = mean((y1\_1500(locs\_Rwave) - val\_Rwave))

meanError\_Swave = mean((y1\_1500(locs\_Swave) - val\_Swave))

%% find PP interval

 i = 0;  %% to make the code start from 0.

 rr = 0; %% each time the code run, rr distance two peaks

 hold off % for the next graph

 rrinterval = zeros(3600,1); % create an array to strore 2 peaks

beat\_count =0;

for k = 2 : length(y1)-1

    %the peak has to be greater than 1 and greater than the value before it and greater then the value after it.

    if(y1(k)> y1(k-1) && y1(k) > y1(k+1) && y1(k)> 1);

     beat\_count = beat\_count +1;

     if beat\_count ==1;

        rr =0;

     else

         rr = k-i;

         rrinterval(k)=rr;

         i=k;

     end

    else

        rrinterval(k)= rr;

    end

end

figure;

plot (rrinterval);

xlabel('Time in sec\*10^-2'), ylabel('Distance betweeen 2 Heatbeats (R-R) in sec\*10^-2'), title('R-R down sampling 1/2 frequency sample intervals');

%% find PP interval

%% heart rate analysis

% count the dominat peak

beat\_count =0;

for k = 2 : length(y2)-1

    %the peak has to be greater than 1 and greater than the value before it and greater then the value after it.

    if(y2(k)> y2(k-1) && y2(k) > y2(k+1) && y2(k)> 1)

         beat\_count = beat\_count +1;

    end

end

display (k);

disp('dominant peaks');

%% divide the peak count by the duration in minute

duration\_in\_sec = N/fs2;

duration\_in\_minute = duration\_in\_sec/60;

BPM = beat\_count/duration\_in\_minute;

%%%  DFT to describe the signal in the frequency

NFFT = 2 ^ nextpow2(N2);

Y = fft(y2, NFFT) / N2;

f = (fs2 / 2 \* linspace(0, 1, NFFT / 2+1))'; % Vector containing frequencies in Hz

amp = ( 2 \* abs(Y(1: NFFT / 2+1))); % Vector containing corresponding amplitudes

figure;

plot (f, amp);

title ('plot single-sided amplitude spectrume of 1/2 frequency sample ECG signal')

xlabel ('frequency (Hz)')

ylabel ('|y(f)|')

grid on;

max\_value=max(y1);

mean\_value=mean(y1);

threshold=(max\_value-mean\_value)/2;

%%Downsampling ¼ sample frequency

close all;

clear all;

clc;

load('I:\BIOM\_Signal\_processing\Hw5\ECGsignal\_1.mat')

disp('Contents of workspace after loading file:')

whos

fs = 250; % find the sampling rate or frequency

fs1 = 250\*1/2;

fs2 = 250\*1/4;

y1=xlsread('I:\BIOM\_Signal\_processing\Hw5\ECGsignal\_1.xls');

T = 1/fs;% sampling rate or frequency

% find the length of the data per second

N = length(y1);

ls = size(y1);

t = (0 : N-1) / fs;% sampling period

figure; %subplot(1,2,1);

      plot(t,y1);

    %plot(x,y2, 'g');

title ('plot of the original of ECG signal')

xlabel ('time (sec)')

ylabel ('Amplitute (mv)')

grid on;

% down sampling 1/2 of frequency sample

y2a = resample(y1,fs1,fs);

N1 = length(y2a);

ls1 = size(y2a);

t21 = (0 : N1-1) / fs1;% sampling period

figure; %subplot(1,2,1);

      plot(t21,y2a);

title ('plot of the down sampling 1/2 frequency sample of ECG signal')

xlabel ('time (sec)')

ylabel ('Amplitute (mv)')

grid on;

%%%%%%%%%%%%%

% down sampling 1/4 of frequency sample

y2 = resample(y1,63,250);

N2 = length(y2);

ls2 = size(y2);

t22 = (0 : N2-1) / fs2;% sampling period

figure; %subplot(1,2,1);

      plot(t22,y2);

title ('plot of the down sampling 1/4 frequency sample of ECG signal')

xlabel ('time (sec)')

ylabel ('Amplitute (mv)')

grid on;

%% find P

m1=max(y2)\*.60;

P=find(y2>=m1);

y1\_1500 = y2(1:1850);

t2 = 1:length(y1\_1500);

figure;

plot(t2,y1\_1500);

title ('plot of subset of down sampling 1/4 frequency sample the ECG signal')

xlabel ('time (msec)')

ylabel ('Amplitute (mv)')

grid on

%% used the snip code from this website.

%%%%http://www.mathworks.com/help/signal/examples/peak-analysis.html

%Detrending Data

%The above signal shows a baseline shift and therefore does not represent the true amplitude. In order to remove the trend, fit a low order polynomial to the signal and use the polynomial to detrend it.

[p,s,mu] = polyfit((1:numel(y1\_1500))',y1\_1500,6);

f\_y = polyval(p,(1:numel(y1\_1500))',[],mu);

ECG\_data = y1\_1500 - f\_y;        % Detrend data

N1= length (y1\_1500);

t1 = (0 : N1-1) / fs2;% sampling period

figure

%plot(t1,ECG\_data); grid on

plot(t2,ECG\_data); grid on

ax = axis; axis([ax(1:2) -2.2 2.2])

%ax = axis; axis([ax(1:2) -3.2 3.2])

title('Detrended down sampling 1/4 frequency sample ECG Signal')

xlabel('time msec'); ylabel('Voltage(mV)')

legend('Detrended ECG Signal')

%Thresholding to Find Peaks of Interest

%The QRS-complex consists of three major components: Q-wave, R-wave, S-wave. The R-waves can be detected by thresholding peaks above 0.5mV. Notice that the R-waves are separated by more than 200 samples. Use this information to remove unwanted peaks by specifying a 'MinPeakDistance'.

[~,locs\_Rwave] = findpeaks(ECG\_data,'MinPeakHeight',0.5,...

                                    'MinPeakDistance',30);

%Finding Local Minima in Signal

%Local minima can be detected by finding peaks on an inverted version of the original signal.

ECG\_inverted = -ECG\_data;

[~,locs\_Swave] = findpeaks(ECG\_inverted,'MinPeakHeight',0.4,...

                                       'MinPeakDistance',30);

%The following plot shows the R-waves and S-waves detected in the signal.

figure

hold on

plot(t2,ECG\_data);

plot(locs\_Rwave,ECG\_data(locs\_Rwave),'rv','MarkerFaceColor','r');

plot(locs\_Swave,ECG\_data(locs\_Swave),'rs','MarkerFaceColor','b');

%axis([0 1850 -1.1 1.1]); grid on;

axis([0 1850 -2.2 2.2]); grid on;

legend('ECG Signal','R-waves','S-waves');

xlabel('time msec'); ylabel('Voltage(mV)')

title('R-wave and S-wave in down sampling 1/4 frequency sample of ECG Signal')

[~,locs\_Twave] = findpeaks(ECG\_data,'MinPeakHeight',-0.02,...

                                      'MinPeakDistance',13);

figure;

hold on

plot(t2,ECG\_data);

plot(locs\_Twave,ECG\_data(locs\_Twave),'X','MarkerFaceColor','y');

plot(locs\_Rwave,ECG\_data(locs\_Rwave),'rv','MarkerFaceColor','r');

plot(locs\_Swave,ECG\_data(locs\_Swave),'rs','MarkerFaceColor','b');

grid on

title('Thresholding Peaks in down sampling 1/4 frequency sample Signal')

xlabel('time msec'); ylabel('Voltage(mV)')

ax = axis; axis([0 1850 -2.2 2.2])

legend('ECG signal','T-wave','R-wave','S-wave');

[~,locs\_Pwave] = findpeaks(ECG\_data,'MinPeakHeight',-0.09,...

                                      'MinPeakDistance',6);

figure;

hold on

plot(t2,ECG\_data);

plot(locs\_Pwave,ECG\_data(locs\_Pwave),'x','MarkerFaceColor','y');

plot(locs\_Twave,ECG\_data(locs\_Twave),'X','MarkerFaceColor','g');

plot(locs\_Rwave,ECG\_data(locs\_Rwave),'rv','MarkerFaceColor','r');

plot(locs\_Swave,ECG\_data(locs\_Swave),'rs','MarkerFaceColor','b');

grid on

title('Thresholding Peaks in down sampling 1/4 frequency sample Signal')

xlabel('time msec'); ylabel('Voltage(mV)')

ax = axis; axis([0 1850 -2.2 2.2])

legend('ECG signal','P-wave','T-wave','R-wave','S-wave');

[~,locs\_qwave] = findpeaks(ECG\_data,'MinPeakHeight',-0.2);

figure;

hold on

plot(t2,ECG\_data);

plot(locs\_qwave,ECG\_data(locs\_qwave),'x','MarkerFaceColor','y');

% link and zoom in to show the changes

%linkaxes(ax(1:2),'xy');

%axis(ax,[60 230 0.006 -0.04])

%Next, we try and determine the locations of the Q-waves. Thresholding the peaks to locate the Q-waves results in detection of unwanted peaks as the Q-waves are buried in noise. We filter the signal first and then find the peaks. Savitzky-Golay filtering is used to remove noise in the signal.

smoothECG = sgolayfilt(ECG\_data,1,3);

figure

plot(t2,ECG\_data,'b',t2,smoothECG,'r'); grid on

axis tight;

xlabel('time msec'); ylabel('Voltage(mV)');

legend('ECG Signal','Filtered Signal')

title('Filtering Noisy of down sampling 1/4 frequency sample ECG Signal')

%We perform peak detection on the smooth signal and use logical indexing to find the locations of the Q-waves.

%[~,min\_locs] = findpeaks(-smoothECG,'MinPeakDistance',29);

%[~,min\_locs] = findpeaks(smoothECG,'MinPeakDistance',2);%Twave

[~,min\_locs] = findpeaks(smoothECG,'MinPeakDistance',25);

% Peaks between -0.2mV and -0.5mV

%locs\_Qwave = min\_locs(smoothECG(min\_locs)>-0.3 &

%-smoothECG(min\_locs)<-0.1); %Twave

locs\_Qwave = min\_locs(smoothECG(min\_locs)>-0.3 & -smoothECG(min\_locs)<-0.11);

figure

hold on

plot(t2,smoothECG);

plot(locs\_Qwave,smoothECG(locs\_Qwave),'rs','MarkerFaceColor','g');

plot(locs\_Rwave,smoothECG(locs\_Rwave),'rv','MarkerFaceColor','r');

plot(locs\_Swave,smoothECG(locs\_Swave),'rs','MarkerFaceColor','b');

grid on

title('Thresholding Peaks down sampling 1/4 frequency sample in Signal')

xlabel('time msec'); ylabel('Voltage(mV)')

ax = axis; axis([0 1850 -2.2 2.2])

legend('Smooth ECG signal','T-interval','R-wave','S-wave');

%The above figure shows that the QRS-complex successfully detected in the noisy ECG signal.

%Error Between Noisy and Smooth Signal

%Notice the average difference between the QRS-complex in the raw and the detrended filtered signal.

% Values of the Extrema

[val\_Qwave, val\_Rwave, val\_Swave] = deal(smoothECG(locs\_Qwave), smoothECG(locs\_Rwave), smoothECG(locs\_Swave));

meanError\_Qwave = mean((y1\_1500(locs\_Qwave) - val\_Qwave))

meanError\_Rwave = mean((y1\_1500(locs\_Rwave) - val\_Rwave))

meanError\_Swave = mean((y1\_1500(locs\_Swave) - val\_Swave))

%% find PP interval

 i = 0;  %% to make the code start from 0.

 rr = 0; %% each time the code run, rr distance two peaks

 hold off % for the next graph

 rrinterval = zeros(3600,1); % create an array to strore 2 peaks

beat\_count =0;

for k = 2 : length(y1)-1

    %the peak has to be greater than 1 and greater than the value before it and greater then the value after it.

    if(y1(k)> y1(k-1) && y1(k) > y1(k+1) && y1(k)> 1);

     beat\_count = beat\_count +1;

     if beat\_count ==1;

        rr =0;

     else

         rr = k-i;

         rrinterval(k)=rr;

         i=k;

     end

    else

        rrinterval(k)= rr;

    end

end

figure;

plot (rrinterval);

xlabel('Time in sec\*10^-2'), ylabel('Distance betweeen 2 Heatbeats (R-R) in sec\*10^-2'), title('R-R down sampling 1/4 frequency sample intervals');

%% find PP interval

%% heart rate analysis

% count the dominat peak

beat\_count =0;

for k = 2 : length(y2)-1

    %the peak has to be greater than 1 and greater than the value before it and greater then the value after it.

    if(y2(k)> y2(k-1) && y2(k) > y2(k+1) && y2(k)> 1)

         beat\_count = beat\_count +1;

    end

end

display (k);

disp('dominant peaks');

%% divide the peak count by the duration in minute

duration\_in\_sec = N/fs2;

duration\_in\_minute = duration\_in\_sec/60;

BPM = beat\_count/duration\_in\_minute;

%%%  DFT to describe the signal in the frequency

NFFT = 2 ^ nextpow2(N2);

Y = fft(y2, NFFT) / N2;

f = (fs2 / 2 \* linspace(0, 1, NFFT / 2+1))'; % Vector containing frequencies in Hz

amp = ( 2 \* abs(Y(1: NFFT / 2+1))); % Vector containing corresponding amplitudes

figure;

plot (f, amp);

title ('plot single-sided amplitude spectrume of 1/4 frequency sample ECG signal')

xlabel ('frequency (Hz)')

ylabel ('|y(f)|')

grid on;

max\_value=max(y1);

mean\_value=mean(y1);

threshold=(max\_value-mean\_value)/2;

%% upsampling 2 sample frequency

close all;

clear all;

clc;

load('I:\BIOM\_Signal\_processing\Hw5\ECGsignal\_1.mat')

disp('Contents of workspace after loading file:')

whos

fs = 250; % find the sampling rate or frequency

fs2 = 250\*2;

y1=xlsread('I:\BIOM\_Signal\_processing\Hw5\ECGsignal\_1.xls');

T = 1/fs;% sampling rate or frequency

% find the length of the data per second

N = length(y1);

ls = size(y1);

t = (0 : N-1) / fs;% sampling period

figure; %subplot(1,2,1);

      plot(t,y1);

    %plot(x,y2, 'g');

title ('plot of the original of ECG signal')

xlabel ('time (sec)')

ylabel ('Amplitute (mv)')

grid on;

% up sampling 2 of frequency sample

y2 = resample(y1,500,250);

N2 = length(y2);

ls2 = size(y2);

t22 = (0 : N2-1) / fs2;% sampling period

figure; %subplot(1,2,1);

      plot(t22,y2);

title ('plot of the up sampling 2 frequency sample of ECG signal')

xlabel ('time (sec)')

ylabel ('Amplitute (mv)')

grid on;

%% find P

m1=max(y2)\*.60;

P=find(y2>=m1);

y1\_1500 = y2(1:1850);

t2 = 1:length(y1\_1500);

figure;

plot(t2,y1\_1500);

title ('plot of subset of upsampling 2 frequency sample the ECG signal')

xlabel ('time (msec)')

ylabel ('Amplitute (mv)')

grid on

%% used the snip code from this website.

%%%%http://www.mathworks.com/help/signal/examples/peak-analysis.html

%Detrending Data

%The above signal shows a baseline shift and therefore does not represent the true amplitude. In order to remove the trend, fit a low order polynomial to the signal and use the polynomial to detrend it.

[p,s,mu] = polyfit((1:numel(y1\_1500))',y1\_1500,6);

f\_y = polyval(p,(1:numel(y1\_1500))',[],mu);

ECG\_data = y1\_1500 - f\_y;        % Detrend data

N1= length (y1\_1500);

t1 = (0 : N1-1) / fs2;% sampling period

figure

%plot(t1,ECG\_data); grid on

plot(t2,ECG\_data); grid on

ax = axis; axis([ax(1:2) -2.2 2.2])

%ax = axis; axis([ax(1:2) -3.2 3.2])

title('Detrended upsampling 2 frequency sample ECG Signal')

xlabel('time msec'); ylabel('Voltage(mV)')

legend('Detrended ECG Signal')

%Thresholding to Find Peaks of Interest

%The QRS-complex consists of three major components: Q-wave, R-wave, S-wave. The R-waves can be detected by thresholding peaks above 0.5mV. Notice that the R-waves are separated by more than 200 samples. Use this information to remove unwanted peaks by specifying a 'MinPeakDistance'.

[~,locs\_Rwave] = findpeaks(ECG\_data,'MinPeakHeight',0.5,...

                                    'MinPeakDistance',240);

%Finding Local Minima in Signal

%Local minima can be detected by finding peaks on an inverted version of the original signal.

ECG\_inverted = -ECG\_data;

[~,locs\_Swave] = findpeaks(ECG\_inverted,'MinPeakHeight',0.4,...

                                       'MinPeakDistance',240);

%The following plot shows the R-waves and S-waves detected in the signal.

figure

hold on

plot(t2,ECG\_data);

plot(locs\_Rwave,ECG\_data(locs\_Rwave),'rv','MarkerFaceColor','r');

plot(locs\_Swave,ECG\_data(locs\_Swave),'rs','MarkerFaceColor','b');

%axis([0 1850 -1.1 1.1]); grid on;

axis([0 1850 -2.2 2.2]); grid on;

legend('ECG Signal','R-waves','S-waves');

xlabel('time msec'); ylabel('Voltage(mV)')

title('R-wave and S-wave in upsampling 2 frequency sample of ECG Signal')

[~,locs\_Twave] = findpeaks(ECG\_data,'MinPeakHeight',-0.02,...

                                      'MinPeakDistance',100);

figure;

hold on

plot(t2,ECG\_data);

plot(locs\_Twave,ECG\_data(locs\_Twave),'X','MarkerFaceColor','y');

plot(locs\_Rwave,ECG\_data(locs\_Rwave),'rv','MarkerFaceColor','r');

plot(locs\_Swave,ECG\_data(locs\_Swave),'rs','MarkerFaceColor','b');

grid on

title('Thresholding Peaks in upsampling 2 frequency sample Signal')

xlabel('time msec'); ylabel('Voltage(mV)')

ax = axis; axis([0 1850 -2.2 2.2])

legend('ECG signal','T-wave','R-wave','S-wave');

[~,locs\_Pwave] = findpeaks(ECG\_data,'MinPeakHeight',-0.09,...

                                      'MinPeakDistance',52);

figure;

hold on

plot(t2,ECG\_data);

plot(locs\_Pwave,ECG\_data(locs\_Pwave),'x','MarkerFaceColor','y');

plot(locs\_Twave,ECG\_data(locs\_Twave),'X','MarkerFaceColor','g');

plot(locs\_Rwave,ECG\_data(locs\_Rwave),'rv','MarkerFaceColor','r');

plot(locs\_Swave,ECG\_data(locs\_Swave),'rs','MarkerFaceColor','b');

grid on

title('Thresholding Peaks in upsampling 2 frequency sample Signal')

xlabel('time msec'); ylabel('Voltage(mV)')

ax = axis; axis([0 1850 -2.2 2.2])

legend('ECG signal','P-wave','T-wave','R-wave','S-wave');

[~,locs\_qwave] = findpeaks(ECG\_data,'MinPeakHeight',-0.2);

figure;

hold on

plot(t2,ECG\_data);

plot(locs\_qwave,ECG\_data(locs\_qwave),'x','MarkerFaceColor','y');

% link and zoom in to show the changes

%linkaxes(ax(1:2),'xy');

%axis(ax,[60 230 0.006 -0.04])

%Next, we try and determine the locations of the Q-waves. Thresholding the peaks to locate the Q-waves results in detection of unwanted peaks as the Q-waves are buried in noise. We filter the signal first and then find the peaks. Savitzky-Golay filtering is used to remove noise in the signal.

smoothECG = sgolayfilt(ECG\_data,1,3);

figure

plot(t2,ECG\_data,'b',t2,smoothECG,'r'); grid on

axis tight;

xlabel('time msec'); ylabel('Voltage(mV)');

legend('ECG Signal','Filtered Signal')

title('Filtering Noisy of upsampling 2 frequency sample ECG Signal')

%We perform peak detection on the smooth signal and use logical indexing to find the locations of the Q-waves.

%[~,min\_locs] = findpeaks(-smoothECG,'MinPeakDistance',29);

%[~,min\_locs] = findpeaks(smoothECG,'MinPeakDistance',2);%Twave

[~,min\_locs] = findpeaks(smoothECG,'MinPeakDistance',25);

% Peaks between -0.2mV and -0.5mV

%locs\_Qwave = min\_locs(smoothECG(min\_locs)>-0.3 &

%-smoothECG(min\_locs)<-0.1); %Twave

locs\_Qwave = min\_locs(smoothECG(min\_locs)>-0.3 & -smoothECG(min\_locs)<-0.11);

figure

hold on

plot(t2,smoothECG);

plot(locs\_Qwave,smoothECG(locs\_Qwave),'rs','MarkerFaceColor','g');

plot(locs\_Rwave,smoothECG(locs\_Rwave),'rv','MarkerFaceColor','r');

plot(locs\_Swave,smoothECG(locs\_Swave),'rs','MarkerFaceColor','b');

grid on

title('Thresholding Peaks down sampling 2 frequency sample in Signal')

xlabel('time msec'); ylabel('Voltage(mV)')

ax = axis; axis([0 1850 -2.2 2.2])

legend('Smooth ECG signal','T-wave','R-wave','S-wave');

%The above figure shows that the QRS-complex successfully detected in the noisy ECG signal.

%Error Between Noisy and Smooth Signal

%Notice the average difference between the QRS-complex in the raw and the detrended filtered signal.

% Values of the Extrema

[val\_Qwave, val\_Rwave, val\_Swave] = deal(smoothECG(locs\_Qwave), smoothECG(locs\_Rwave), smoothECG(locs\_Swave));

meanError\_Qwave = mean((y1\_1500(locs\_Qwave) - val\_Qwave))

meanError\_Rwave = mean((y1\_1500(locs\_Rwave) - val\_Rwave))

meanError\_Swave = mean((y1\_1500(locs\_Swave) - val\_Swave))

%% find PP interval

 i = 0;  %% to make the code start from 0.

 rr = 0; %% each time the code run, rr distance two peaks

 hold off % for the next graph

 rrinterval = zeros(3600,1); % create an array to strore 2 peaks

beat\_count =0;

for k = 2 : length(y1)-1

    %the peak has to be greater than 1 and greater than the value before it and greater then the value after it.

    if(y1(k)> y1(k-1) && y1(k) > y1(k+1) && y1(k)> 1);

     beat\_count = beat\_count +1;

     if beat\_count ==1;

        rr =0;

     else

         rr = k-i;

         rrinterval(k)=rr;

         i=k;

     end

    else

        rrinterval(k)= rr;

    end

end

figure;

plot (rrinterval);

xlabel('Time in sec\*10^-2'), ylabel('Distance betweeen 2 Heatbeats (R-R) in sec\*10^-2'), title('R-R down sampling 2 frequency sample intervals');

%% find PP interval

%% heart rate analysis

% count the dominat peak

beat\_count =0;

for k = 2 : length(y2)-1

    %the peak has to be greater than 1 and greater than the value before it and greater then the value after it.

    if(y2(k)> y2(k-1) && y2(k) > y2(k+1) && y2(k)> 1)

         beat\_count = beat\_count +1;

    end

end

display (k);

disp('dominant peaks');

%% divide the peak count by the duration in minute

duration\_in\_sec = N/fs2;

duration\_in\_minute = duration\_in\_sec/60;

BPM = beat\_count/duration\_in\_minute;

%%%  DFT to describe the signal in the frequency

NFFT = 2 ^ nextpow2(N2);

Y = fft(y2, NFFT) / N2;

f = (fs2 / 2 \* linspace(0, 1, NFFT / 2+1))'; % Vector containing frequencies in Hz

amp = ( 2 \* abs(Y(1: NFFT / 2+1))); % Vector containing corresponding amplitudes

figure;

plot (f, amp);

title ('plot single-sided amplitude spectrume of upsampling 2 frequency sample ECG signal')

xlabel ('frequency (Hz)')

ylabel ('|y(f)|')

grid on;

max\_value=max(y1);

mean\_value=mean(y1);

threshold=(max\_value-mean\_value)/2;

### Matlab code to plot the FFT of the windowed segments of ECG signal

clear all;  
clc;  
fs = 250 % find the sampling rate or frequency  
T = 1/fs;% sampling rate or frequency  
window = 120; % 2 min 0r 120 second   
fignum = 0;  
load('J:\BIOM\_Signal\_processing\exam1\ECGsignalTest\_1.mat') % contains hr\_sig and fs  
% Make time axis for ECG signal  
tx = [0:length(hr\_sig)-1]/fs;  
fignum = fignum + 1;  
figure(fignum)  
%subplot (211)  
plot(tx,hr\_sig)  
xlabel('Time (s)')  
ylabel('Amplitude (mV)')  
title('Zoom into ECG signal')  
xlim([30.3,31]) % Used to zoom in on single ECG waveform  
%load('\\tsclient\E\BIOM\_Signal\_processing\exam1\ECGsignalTest\_1.mat')  
%disp('Contents of workspace after loading file:')  
%whos  
%% copy the data and put into excel   
y1=xlsread('J:\BIOM\_Signal\_processing\exam1\ecg\_1.xls');  
% find the length of the data per second  
N = length(y1); %length of signal  
ls = size(y1);  
t = (0 : N-1) / fs;% sampling period or time vector  
fignum = fignum + 1; %% keep track of figures  
figure(fignum)  
subplot (211), plot(t,y1);  
title ('plot of the original of ECG signal')  
xlabel ('time (sec)')  
ylabel ('Amplitute (mv)')  
grid on;  
%Compute the spectrum of the ECG and provide remarks on the spectral features of the ECG ( see reference “ECG Statistics, Noise, Artifacts, and Missing Data”).  
%%%  DFT to describe the signal in the frequency  
NFFT = 2 ^ nextpow2(N);  
Y = fft(y1, NFFT) / N;  
f = (fs / 2 \* linspace(0, 1, NFFT / 2+1))'; % Vector containing frequencies in Hz  
amp = ( 2 \* abs(Y(1: NFFT / 2+1))); % Vector containing corresponding amplitudes  
%figure;  
subplot(212), plot (f, amp);  
title ('plot single-sided amplitude spectrume of the ECG signal');  
xlabel ('frequency (Hz)');  
ylabel ('|y(f)|');  
grid on;  
  
%%% Create a period  
figure;  
y1new = y1(180:300);  
N1 = length(y1new);  
%hold off  
subplot (2,1,1), plot (y1new);  
title ('plot one  typical PQRST segments of the signal amplitude spectrume of the orginal ECG signal')  
xlabel ('time (msec)')  
ylabel ('Amplitute  (mv)')  
grid on;  
  
%%%  DFT to describe the signal in the frequency of 1 window  
NFFT1 = 2 ^ nextpow2(N1); % FFT transform length  
Y2 = fft(y1new, NFFT1) / N1; % FFT, Discard unnecessary part of spectrum and scale amplitude  
f1 = (fs / 2 \* linspace(0, 1, NFFT1 / 2+1))'; % Vector containing frequencies in Hz  
amp1 = ( 2 \* abs(Y2(1: NFFT1 / 2+1))); % Vector containing corresponding amplitudes  
%figure;  
subplot (2,1,2), plot (f1, amp1);  
title ('single-sided amplitude spectrume of one PQRST segments of ECG signal');  
xlabel ('frequency (Hz)');  
ylabel ('|y(f)|');  
grid on;  
  
xx=zeros(1,length(y1));   
Xtwz = zeros(1,N); % pre-allocate STFT output array  
M = length(y1new);           % M = window length, N = FFT length  
zp = zeros(N-M,1);       % zero padding (to be inserted)  
xoff = 0;                % current offset in input signal x  
Mo2 = (M-1)/2;           % Assume M odd for simplicity here  
for m=1:N  
    xt = y1(xoff+1:xoff+M); % extract frame of input data  
    xtw = y1new.\* xt;         % apply window to current frame  
    xtwz = [xtw(Mo2+1:M); zp; xtw(1:Mo2)]; % windowed, zero padded  
    Xtwz(m,:) = fft(xtwz); % STFT for frame m  
    %xoff = xoff + R;       % advance in-pointer by hop-size R  
end  
  
figure;  
plot (xtw);  
  
figure;  
plot (Xtwz);  
  
  
  
  
%% window  
% Stride through the input with 50% overlap   
% calculate windowed length “fftlen” FFT of each block,  
% then inverse transform and overlap-add.  
for i=1:1024/2:(length(y1)-1024)  
%ff=fft(y1new.\*y1(i:(i+128-1))',128);  
%  
% User processing of FFT data would go here...  
%  
%xx(i:(i+NFFT-1))= xx(i:(i+NFFT-1))+ ifft(ff,NFFT);  
end   
  
%% create a sub-sample of original ECG  
y1\_sub = y1(5000:7600);  
tsub = 1:length(y1\_sub);  
figure;  
subplot (2,2,1), plot(tsub,y1\_sub);   
title ('locate ST-interval (Angina pectoris) of the original ECG signal')  
xlabel ('time (msec)')  
ylabel ('Amplitute (mv)')  
grid on  
  
%% create a sub-sample of original ECG for 1 ST  
y1\_1 = y1(5042:5250);  
tsub\_1 = 1:length(y1\_1);  
N1\_1=length(y1\_1);  
%figure;  
subplot (2,2,2), plot(tsub\_1,y1\_1);   
title ('Zoom-in ST-interval (Angina pectoris) of the original ECG signal')  
xlabel ('time (msec)')  
ylabel ('Amplitute (mv)')  
grid on  
  
  
%Compute the spectrum of Angina pectoris the ECG and provide remarks on the spectral features of the ECG .  
%%%  DFT to describe the signal in the frequency  
NFFT1\_1 = 2 ^ nextpow2(N1\_1);  
Y2\_1 = fft(y1\_1, NFFT1\_1) / N1\_1;  
f1\_1 = (fs / 2 \* linspace(0, 1, NFFT1\_1 / 2+1))'; % Vector containing frequencies in Hz  
amp1\_1 = ( 2 \* abs(Y2\_1(1: NFFT1\_1 / 2+1))); % Vector containing corresponding amplitudes  
%figure;  
subplot (2,2,3), plot (f1\_1, amp1\_1);  
title ('single-sided amplitude spectrume of Angina pectoris segments of ECG signal');  
xlabel ('frequency (Hz)');  
ylabel ('|y(f)|');  
grid on;

%Create a 4 Hz sinusoidal waveform (ie, a waveform of the form A\*sin(w0\*t) where w0 = 2\*pi\*f, A is amplitude, and f is the sinusoidal frequency)

clc;  
clear all;  
close all;  
  
fm =1;  
f= 4; %4 Hz sinusoidal  
w0 = 2\*pi\*f;  
Ts = 1/f;                     % sampling period  
Tmax = 2.0;                    % signal duration  
B= 1;  
ka=1; %modulation coefficient  
  
%t = [0:Ts:Tmax];               % time vector  
t=0:0.001:1;  
set(0,'defaultlinelinewidth',2);  
A = B\*cos(2\*pi\*f\*t);   
S=A.\*sin(w0\*t);%waveform Signal  
  
AM=A.\*(1+ka\*S); %Amplitude Modulated wave  
  
subplot(3,1,1);%Plotting frame divided in to 3 rows and this fig appear at 1st  
plot(t,S);  
xlabel('Time');  
ylabel('Amplitude');  
title('A\*sin(2\*pi\*f\*t)waveform');  
grid on;  
  
subplot(3,1,2); %plotting the  wave  
plot(t,A);  
xlabel('Time');  
ylabel('Amplitude');  
title('A = B\*cos(w\*t) waveform');  
subplot(3,1,3); %plotting the amplitude modulated wave  
plot(t,AM);  
xlabel('Time');  
ylabel('Amplitude');  
title('AM modulation signal');  
  
  
%Compute the spectrum of Angina pectoris the ECG and provide remarks on the spectral features of the ECG .  
%%%  DFT to describe the signal in the frequency  
N1 = length (S);  
NFFT1 = 2 ^ nextpow2(N1);  
Y2 = fft(S, NFFT1) / N1;  
f1 = (f / 2 \* linspace(0, 1, NFFT1 / 2+1))'; % Vector containing frequencies in Hz  
amp1 = ( 2 \* abs(Y2(1: NFFT1 / 2+1))); % Vector containing corresponding amplitudes  
figure;  
 subplot(211), plot(f1, amp1);  
title ('single-sided amplitude spectrume of  A\*sin(2\*pi\*f\*t)waveform ');  
xlabel ('frequency (Hz)');  
ylabel ('|y(f)|');  
grid on;  
  
%Compute the spectrum of Angina pectoris the ECG and provide remarks on the spectral features of the ECG .  
%%%  DFT to describe the signal in the frequency  
N1\_1 = length (AM);  
NFFT1\_1 = 2 ^ nextpow2(N1\_1);  
Y2\_1 = fft(AM, NFFT1\_1) / N1\_1;  
f1\_1 = (f / 2 \* linspace(0, 1, NFFT1\_1 / 2+1))'; % Vector containing frequencies in Hz  
amp1\_1 = ( 2 \* abs(Y2\_1(1: NFFT1\_1 / 2+1))); % Vector containing corresponding amplitudes  
%figure;  
 subplot(212); plot(f1\_1, amp1\_1);  
title ('single-sided amplitude spectrume of AM modulation signal ');  
xlabel ('frequency (Hz)');  
ylabel ('|y(f)|');  
grid on;