ECE-744 SPRING 2014

# EMBEDDED DIGITAL SYSTEM DESIGN USING TIME - FREQUENCY ANALYSIS

## PROJECT IMPLEMENTATION REPORT

## ECG SIGNAL PROCESSING

BY
VENKATA SAI SARATH MAREPALLI
(A20319735)

# EXECUTIVE SUMMARY

In this project I am doing a simple implementation of a Time-Frequency analysis on a raw ECG signal, An ECG signal is obtained from an on-line database and it is analyzed using the basic toolbox in Matlab program and then its features are determined finally its rate is determined using its features. The ECG signal is treated with FFT and Wigner Distribution techniques to remove all the unwanted data in it and the peaks for each beat are determined, all the results and explanations are done in the appendix section and also the code used for this implementation is attached along with the submission.

# INTRODUCTION

Electrocardiogram (ECG or EKG) is a diagnostic tool which measures the electric activity of the heart in detail. By analyzing the ECG a variety of heart conditions can be studied and various problems can be prevented including death.

An ECG is developed by the nerve impulse in sync with the heart when the heart pumps blood for once that means the contraction and relaxation of the heart generates a nerve impulse and this activity generates an ECG pulse. Due to this activity current us diffused around the surface of the body, which intern develops a voltage drop varing between μV to a few mV. As it is a very small impulse it requires a large amount of amplification.

# THEORY

**Electrocardiogram:**

A typical ECG wave consists of a QRS complex, P wave and a T wave, also a small amount of U wave but it is viisble only in 50% to 75% of the ECG results. The isoelectric line is the base voltage of an ECG signal and it is generally the portion of the tracing following the T wave and preceding the next P wave. This electrical activity can be recorded from the surface of the human body, Thus ECG is nothing but a voltmeter using up to 12 electrodes that is 12 different leads used to detect the electrical activity of the heart. The electrical activity of the heart is generally sensed by monitoring electrodes placed on the skin surface. The electrical signal is very small (normally 0.0001 to 0.003 volt). These signals are within the frequency range of 0.05 to 100 Hertz (Hz.).

**Matlab:**

MATLAB( Matrix Laboratory) is a  multi-paradigm numerical computing environment and fourth-generation programming language. Matlab has various toolboxes which are used in this project. The toolbox used is a signal processing toolbox which has the filters and FFT transforms in it.

# IMPLEMENTATION

In this project we are trying to detect the heart beat rate(bpm) by eliminating the noise. So firstly we get the raw ECG signal from the sources specified below in the appendix and load it into the matlab and plot the signal and thus we can detect there are certain amount of samples on the signal. The main task of Electrocardiogram processing is R-peaks detection and we detect that there are some irregular distances between the peaks and presence of a low frequency components due to patients breathing.

So the above mentioned characters are first eliminated and then the analysis is continued. The ECG signal obtained from the source is quite uneven so in the first step we straighten it we apply Fast Fourier Transform(FFT) to it, The FFT helps in removing the low frequency component and this is obtained by first generating the FFT for the whole duration of the signal and later applying Inverse Fast Fourier Transform to the result to get the original form of the signal minus the Low frequency component.

As the second step the local maxima's are located and this is done by using a window technique because the windowed filter percives only maxima in the window and ignores the minima.

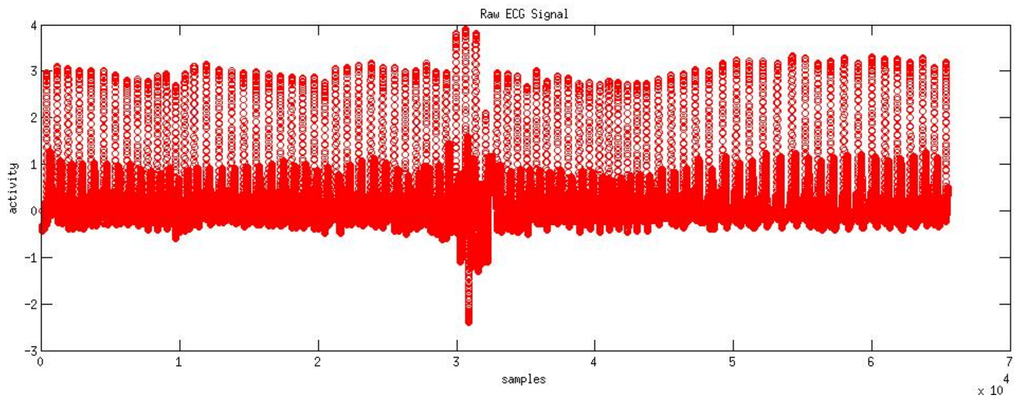
Even though we ignore the minima samples in a particular time we are still left out with some samples of lower frequency this are elimnated by passing them through a threshold filter and then the final output is plotted.

As the final step the correct Heart rate of the sample data is calculated.

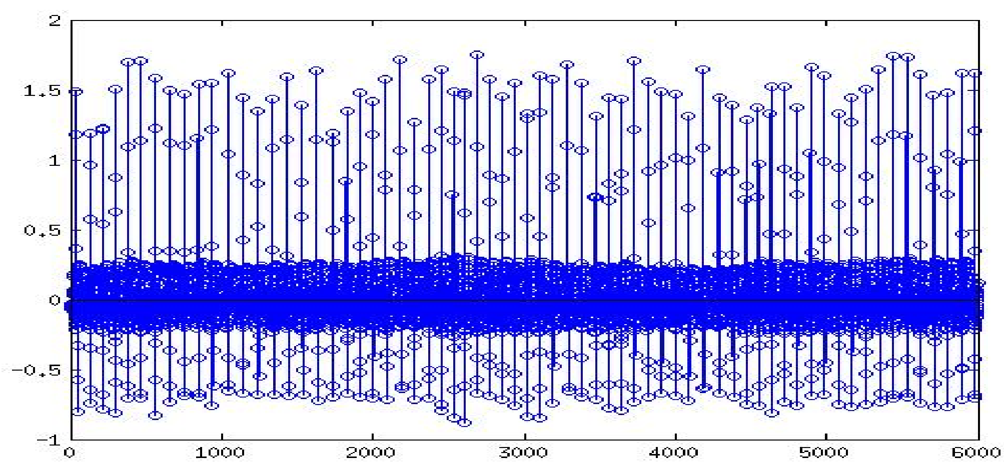
# RESULTS

The source ECG signal is plotted and is shown below at every stage the outputs are plotted and are shown below. As the main aim of the project is to determine the Heart beat rate we calculate the beat rate of the final noise reduced wave and this is obtained as 67 bpm(beats per minute).

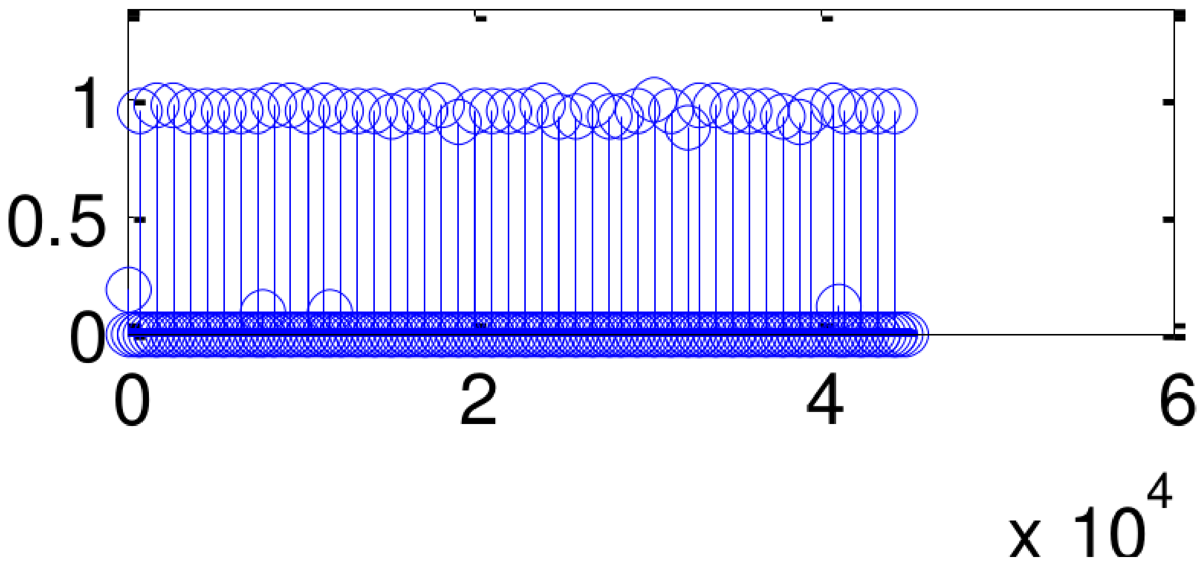
**A) Initial Wave:**



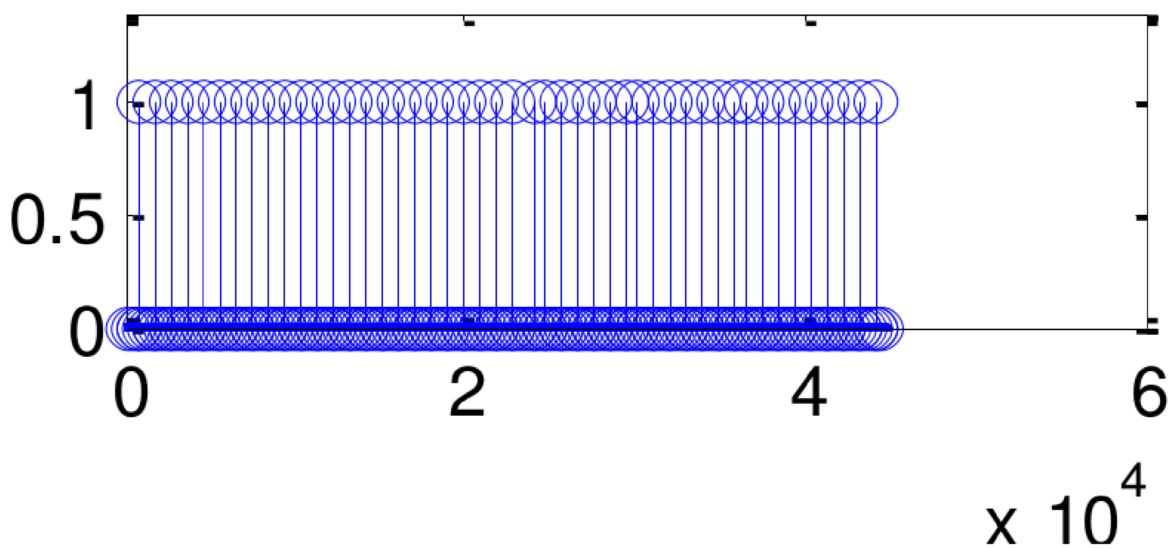

**B) Signal After FFT:**



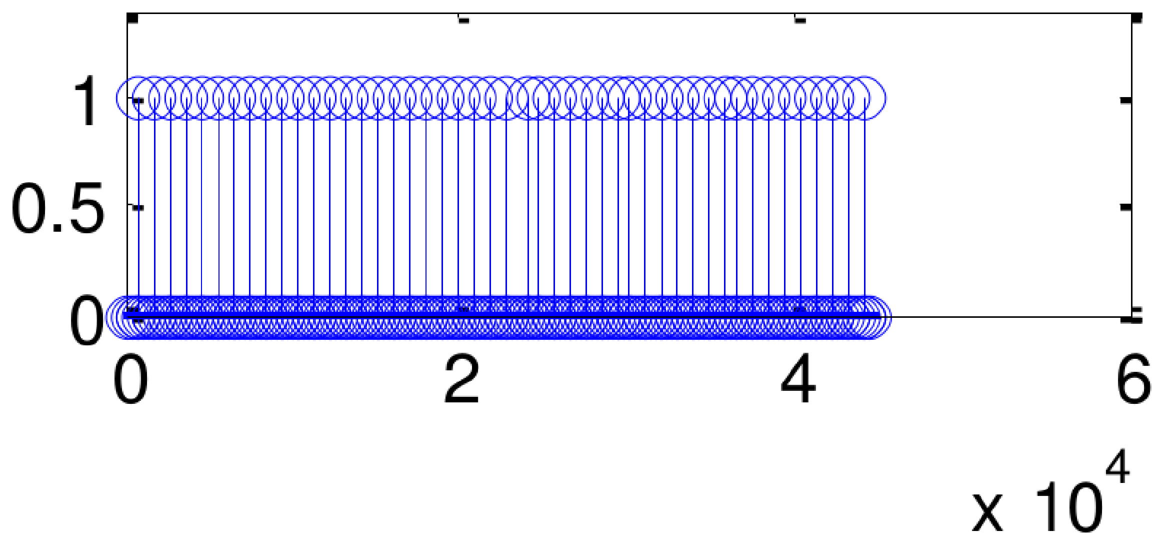

**C) Signal After passing through Windowed High Pass Filter:**



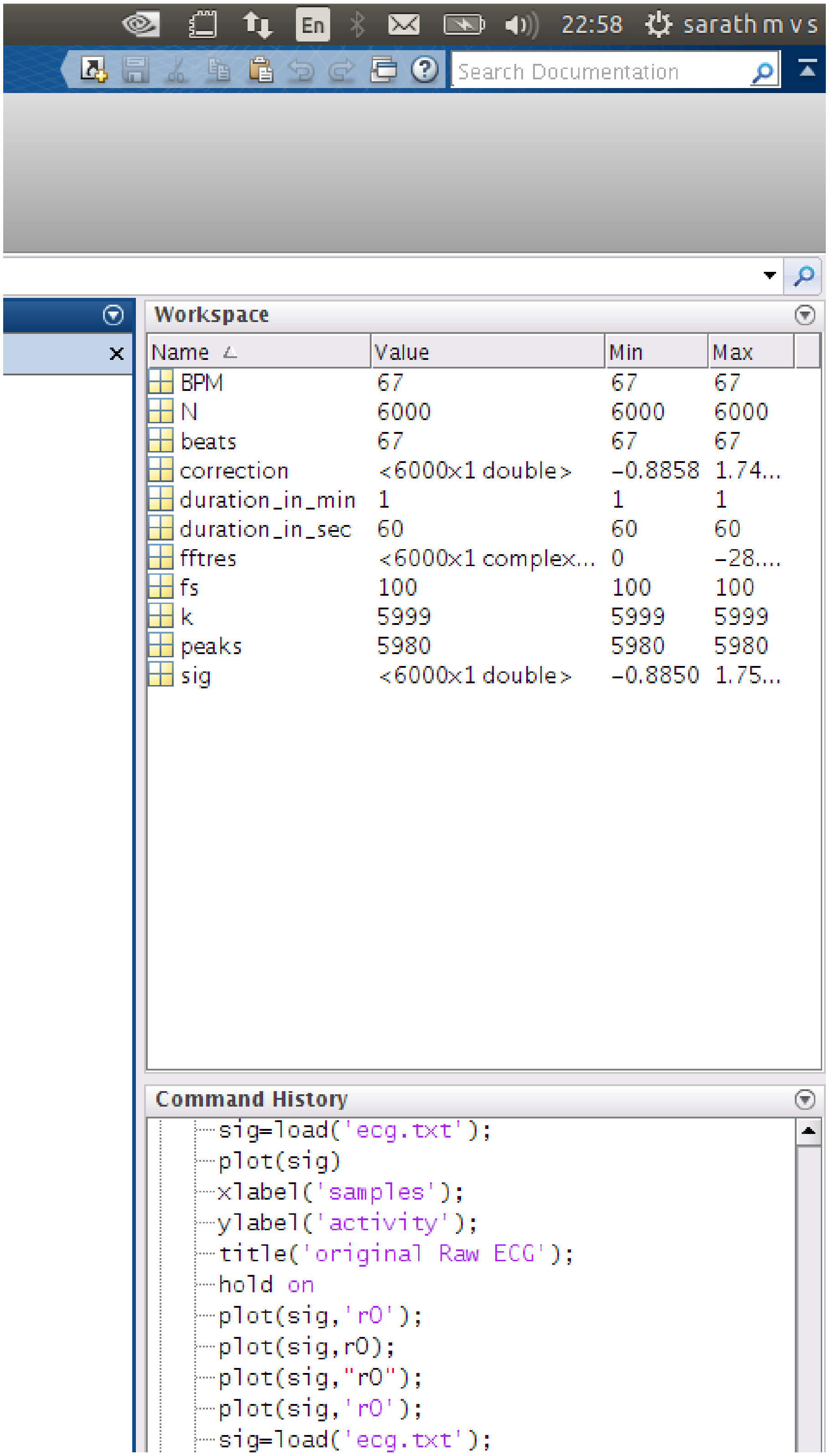

**D) Signal After Passing Through Threshold Filter:**



**E) Signal After passing through Threshold Filter:**



**F) ScreenShot showing the BPM:**



# FUTURE POSSIBILITIES

The same ECG signal processing can be done by Wigner Function to obtain a better results and also this code can be exported to C language by using a functionality in Matlab which is got by using code

%#codegen Matlab

The above code has a capability of checking the entire code and convert it into C/C++ code a sign on the corner of the matlab window indicates the state of the code.

# CONCLUSION

Thus in this project I have analyzed the ECG wave and also removed the noise and unwanted signals from the signal and thus able to get the BPM of the heart. The future possibilities of this project implementation have been identified and described above.

# APPENDIX

**Matlab Code:**

```
% first we load the Given Raw ECG signal and plot it on a graph and denote
% the number of samples present in the signal

sig1 =load ( 'ecg.txt ' ) ;
plot ( sig1 ) ;
xlabel ( 'samples' ) ;
ylabel (  'activity' ) ;
title ( 'Raw ECG Signal' ) ;

% we measure the length of the signal for further calculations the length
% of the signal in this case is the duration of the signal
fs = 100 ;
L = length ( sig1 ) ;
% now we denote the possible samples on the plot using a matlab command
hold on ;
plot ( sig1 , 'R0' ) ;

% After detecting the length and sampling rate of the signal we then
% proceed to the first step in the algorithm that is passing it through a
%  FFT transformation and lower frequencies are removed.

fftres = fft ( sig1 ) ;
fftres (1: round (length ( fftres ) / fs ) ) = 0 ;
fftres ( end - round ( length ( fftres ) / fs ) : end ) = 0 ;
% after the FFT transformation accordinf to the algorithm we get back the
% original signal by running a inverse FFt on the FFT
correction = real ( ifft ( fftres ) ) ;
```

```
% passing the FFT treated signal into a windowed signal whosse size is
% determined by the input automatically

windowsize = floor ( fs * L / 1000 ) ;
if  rem ( windowsize , 2 ) == 0
        windowsize = windowsize + 1 ;
end

filter1 = winopt1 ( correction , windowsize ) ;
% scaling the ecg
peak1 = filter1 / ( max ( filter1 ) / 7 ) ;
% filtering thus obtained ecg by a threshold filter

for data = 1 : 1 : length ( peak1 )
            if  peak1 ( data ) < 4
                        peak1 ( data ) = 0 ;
             else
                         peak1 ( data ) = 1 ;
            end
end
position = find ( peak1 ) ;
dist = position ( 2 ) - position ( 1 ) ;
for data = 1 : 1 : length ( positions ) - 1
             if positions ( data + 1 ) - positions ( data ) < dist
                         dist = position ( data + 1 ) - position ( data ) ;
            end
end

% setting the filter window size
```

```
dist2 = floor ( 0.04 * fs ) ;
if rem ( dist2 , 2 ) == 0
    dist2 = dist2 + 1 ;
end
windowsize = 2 * dist - dist2 ;

% passing the output onto a second filter
filter2 = winopt 2 ( correction , winsize ) ;
peak2 = filter2 ;

for data 1 : 1 : length ( peak2 )
          if peak2 ( data ) < 4
                    peak2 ( data ) = 0 ;
          else
                    peak2 ( data ) = 1 ;
           end
end

% determining the Heart Rate Beats per minute
beats = 0 ;
for k = 2 :  length ( sig ) - 1
          if ( sig ( k ) > sig ( k-1 ) & sig ( k ) > sig ( k + 1 ) & sig ( k ) > 1 )
                    peaks=k;
                    beats=beats+1;
          end
end

fs = 100 ;
N = length ( sig1 ) ;
```

```
duration_in_sec = N / fs ;
duration_in_min = duration_in_sec / 60 ;
BPM = beats / duration_in_min ;

% plotting the entire operation
figure ( ECG ) ; set ( ecg , 'name' , strcat ( plotname , '-stage' ) ) ;

% plotting the original data
subplot ( 3 , 2 , 1 ) ; plot ( ( sig1 , min ( sig1 ) ) / ( max ( sig1 ) - min ( sig1 ) ) ) ;
title ( 'Original Raw ECG');

% After passing into the FFT the output is as follows
subplot ( 3 , 2 , 2 ) ; plot ( ( correction , min ( correction ) ) / ( max ( correction ) - min (correction ) ) ) ;
title ( 'After FFT' ) ;

%  after passing into threshold filter
subplot ( 3 , 2 , 3 ) ; stem ( ( filter1 , min ( filter1 ) ) / ( max ( filter1 ) - min ( filter1 ) ) ) ;
title ( 'After Thresholdfilter' )  ;

% after passing into the final second filter
subplot ( 3 , 2 , 4 ) ; stem ( ( filter2 , min ( filter2 ) ) / ( max ( filter2 ) - min ( filter2 ) ) ) ;
title ( 'detection of peaks' ) ;
```

# REFERENCES


1. http://www.physionet.org/physiobank/database/#ecg
2. http://en.wikipedia.org/wiki/MATLAB
3. Study and Analysis of ECG Signal Using MATLAB & LABVIEW as Effective Tools
   M. K. Islam, A. N. M. M. Haque, G. Tangim, T. Ahammad, and M. R. H. Khondokar, Member IACSIT
4. ECG Signal Processing Using Wigner Function
   Marius Branzila, Valeriu David, Electrical measurements and materials, Gheorghe Asachi Technical University of Iasi, TUIASI