

# **INTELLIGENT DETERMINATION OF ECG HEART BEAT RATE**

**GRADUATION PROJECT SUBMITTED TO  
THE FACULTY OF ENGINEERING  
OF  
NEAR EAST UNIVERSITY**

**By**

**Buse Uğur  
Simon Gideon Idris  
Tareq Tarazi**

**In Fulfillment of the Requirements for  
The Degree of Bachelor of Science  
In  
Biomedical Engineering**

**NICOSIA -2014**

# **INTELLIGENT DETERMINATION OF ECG HEART BEAT RATE**

**GRADUATION PROJECT SUBMITTED TO  
THE FACULTY OF ENGINEERING  
OF  
NEAR EAST UNIVERSITY**

**By**

**Buse Uğur  
Simon Gideon Idris  
Tareq Tarazi**

***Supervisor: Ali Işın***

**In Fulfillment of the Requirements for  
The Degree of Bachelor of Science**

**In  
Biomedical Engineering  
NICOSIA -2014**

## **Contents:**

### **CHAPTER 1**

1.1 Introduction .....	1
1.1.1 Characteristics of normal Electrocardiogram.....	1
1.1.2 Why is the ECG important.....	2
1.2 Principal Methods for ECG Analysis.....	3
1.2.1 Introduction.....	3
1.2.2 QRS Detection-Brief Review.....	3
1.3 Purpose and Motivation.....	4
1.3.1 Introduction.....	4

### **CHAPTER 2**

#### **Origin and Spread of Cardiac Excitation**

2.1 Cell to Cell spread.....	5
2.2 Atrial Excitation.....	5
2.3 The AV node.....	5
2.4 Ventricular Excitation.....	6
2.4.1 Bundle of His and Purkinje Fibres.....	6

### **CHAPTER 3**

#### **Block Diagram in the System**

3.1 Introduction.....	7
3.2 ECG Data Acquisition.....	7

### **CHAPTER 4**

#### **Signal Preprocessing**

4.1 Introduction.....	9
4.2 Types of Filters.....	11
4.2.1 Low Pass Filter.....	11
4.2.2 High Pass Filter.....	12
4.2.3 Comb Filter (Band-stop Filter).....	13
4.3 The Preprocessing Steps.....	14
4.3.1 Removing DC Components of the ECG Signal.....	14
4.3.2 Baseline Wandering.....	15
4.3.3 Removing Low Frequency and High Frequency Noise.....	15
4.3.4 Removing 60Hz Powerline Interference.....	16

## **5CHAPTER**

### **QRS Detection**

5.1 Introduction.....	18
5.2 Pan-Tompkins QRS Detection Algorithm.....	18
5.2.1 Differentiation.....	19
5.2.2 Squaring.....	20
5.2.3 Integration.....	21
5.2.4 Thresholding.....	22

## **CHAPTER 6**

### **Search Procedure for QRS**

6.1 Location of R Peaks.....	23
6.2 Calculation of Hear Beat Rate.....	23
6.3 Result.....	24
6.4 Discussion.....	24

## **CHAPTER 7**

Conclusion & Future Work.....	25
References .....	26
Appendix .....	27

# CHAPTER 1

## INTRODUCTION

---

### *1.1 Introduction*

The heart is one of the most critical organs in the human body, thus the development of methods for monitoring its functionality is crucial. Electrocardiography is considered to be one of the most powerful diagnostic tools in medicine that is routinely used for the assessment of the functionality of the heart. The Electrocardiogram (ECG) is the conventional method for non-invasive interpretation of the electrical activity of the heart in real-time. The electrical cardiac signals are recorded by an external device, by attaching electrodes to the outer surface of the skin of the patient's thorax. These currents stimulate the cardiac muscle and cause the contractions and relaxations of the heart<sup>1</sup>. The electrical signals travel through the electrodes to the ECG device, which records them as characteristic waves. Different waves reflect the activity of different areas of the heart which generate the respective flowing electrical currents. Figure 1 shows a schematic representation of a normal ECG and its various waves.

#### *1.1.1 Characteristics of normal Electrocardiogram*

A normal ECG consists of a P wave, a QRS complex, and a T wave. The P wave is caused by electric currents produced by the depolarization of the atria before their contraction, while the QRS complex is caused by electric currents produced by the depolarization of the ventricles prior to their contraction, during the extending of the depolarization in the ventricular myocardium. The QRS complex usually consists of three different waves, the Q, R, and S waves. Note that both the P-wave, and the waves that form the QRS complex, are depolarization waves. The T wave is caused by the electric currents produced during recovery of the ventricles from the state of depolarization. This process takes place in the ventricular myocardium 0.25s to 0.35s after the depolarization. The T wave is characterized as the wave of repolarization. The Figure 1 shows a representation of an ECG with the waves and complexes annotated

---

<sup>1</sup> <http://en.wikipedia.org/wiki/Electrocardiography>

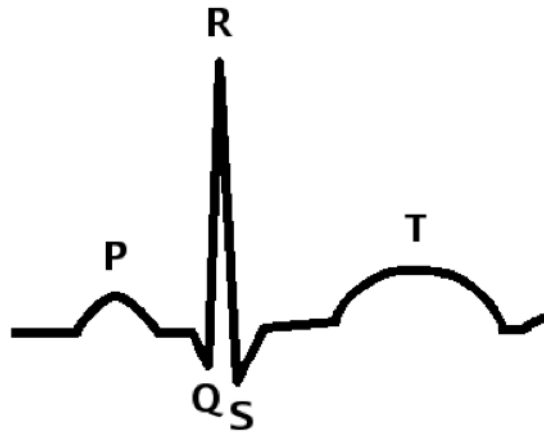


Figure 1: Normal ECG with the waves that is consisted noted.

### *1.1.2 Why is the ECG important?*

The ECG has been established as the most common, easiest, way for and rapid diagnosis and management of numerous cardiovascular incidents. A significant number of patients treated in the emergency room (ER) and in the intensive care unit (ICU), present with cardiovascular complaints. In those cases, they need early, accurate diagnosis as well as rapid, appropriate therapy, reinforce the importance of electrocardiography. Some examples of incidents that are ideally managed with an ECG are chest pain (presenting ST-segment elevation), acute myocardial infarction, acute coronary syndrome, arrhythmias, and even suspected pulmonary embolism.<sup>2</sup>

## **1.2 Principal Methods for ECG Analysis**

### *1.2.1 Introduction*

---

<sup>2</sup> Amal Mattu, William Brady, "ECGs for the Emergency Physician", BMJ Books, BMA House, Tavistock Square, London WC1H 9JR, pp. ix, 2003.

Since digital electrocardiography has been established as the fundamental way for ECG data acquisition, algorithms or automatic ECG analysis, and more specifically automatic QRS complex detection has been the focus of intense research activity<sup>3</sup>. The QRS complex is perhaps the most significant waveform within the ECG and thus its detection is the crucial first step in every automated algorithm for ECG analysis. Due to their characteristic shape, the QRS complexes serve as the reference point for the automated heart rate determination<sup>4</sup>. After detection, analysis and feature extraction, provide useful information about the current state of the heart. Software QRS detection has been a research topic for more than thirty years. Within the last decade, numerous new approaches have been proposed and compared, in order to find the optimum automatic QRS detection method<sup>5</sup>.

### *1.2.2 QRS Detection-Brief Review*

Algorithms, which have been developed for the purpose of QRS detection and analysis, have been derived from Artificial Neural Networks, genetic algorithms, wavelet transforms and filter banks. Neural Networks have been used for QRS detection, by training adaptive, non-linear ECG signal predictors. In another method, an estimate of the ECG samples was derived by a number of adaptive filters. That estimate was given as a weighted summation of previous samples, with the weights adapting according to the statistics of the signal. Other approaches included signal derivatives, for detection of the steep slope of the QRS complex, cross-correlation methods, where an initial template was aligned to the current ECG signal, and syntactic approaches, where the ECG signal was represented as a piecewise linear approximation and was analyzed using syntactic rules. Almost all of the proposed algorithms so far, share a common algorithmic structure, that is, a preprocessing stage, including filtering, a feature extraction stage, and a decision stage in which peak detection and decision logic are included.

---

<sup>3</sup> Carsten Meyer, Jose Fernandez Gavela, Matthew Harris, “Combining Algorithms in Automatic Detection of QRS Complexes in ECG Signals”, IEEE transactions on Information Technology in Medicine, Vol. 10, No.3, July 2003.

<sup>4</sup> Franco Chiarugi, “New Developments in the Automatic Analysis of the Surface ECG: The Case of Atrial Fibrillation”, Hellenic Journal of Cardiology 49, pp.207--- 221, 2008.

<sup>5</sup> Bert-Uwe Kohler, Carsten Hennig, Reinhold Orglmeister, “The Principles of Software QRS Detection: Reviewing and Comparing Algorithms for Detecting this Important ECG Waveform”, IEEE Engineering in Medicine and Biology, February, 2002.

## 1.3 Purpose and Motivation

### *1.3.1 Introduction*

The purpose of this project was to analyze the ECG and more specifically to find the heart beat rate contained in the ECG through reliable detection of the QRS complexes. To avoid erroneous results, QRS identification should be preceded by artifact detection and removal from the ECG signal, so that the QRS detection would be more reliable. The ECG data used for this work was found in the PhysioNetlibrary ([www.physionet.org](http://www.physionet.org)) in the MIMICII database. MIMIC II database's recordings are taken from ICU patients and, since it is a real-life database, it contains lots of artifacts, which interfere with the normal and abnormal ECG signals.



# CHAPTER 2

## ORIGIN & SPREAD OF CARDIAC EXCITATION

---

### *2.1 Cell to Cell spread*

Electrical excitation generated by the pacemaker cells in the SA node is spread between cell-cell via gap junctions. These allow the free movement of ions between cells.

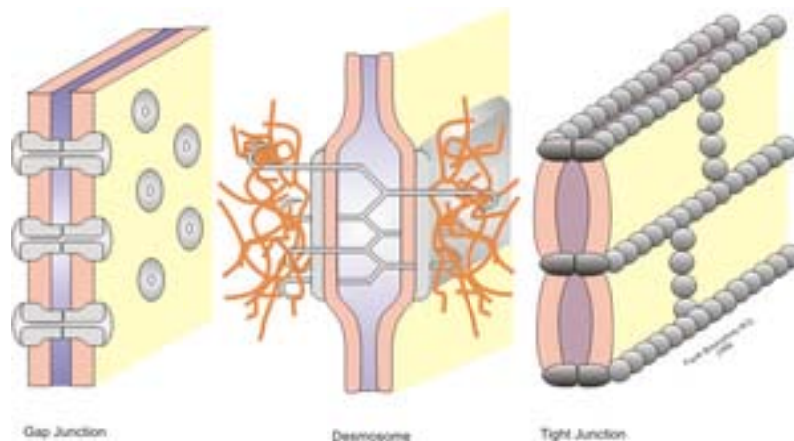


Illustration of tight junction, gap junction, and Desmosomal junction (desmosome)

### *2.2 Atrial Excitation*

Inter-atrial pathway and the Internodal pathway: The former allows for almost simultaneous contraction of both atria due to the rapid electrical spread via gap junctions. The latter directs this excitation towards the atrio-ventricular node, which (because there is a layer of fibrous, non-conducting tissue between ventricles and atria) is the only point of electrical communication between the atria and ventricles.

### *2.3 The AV node*

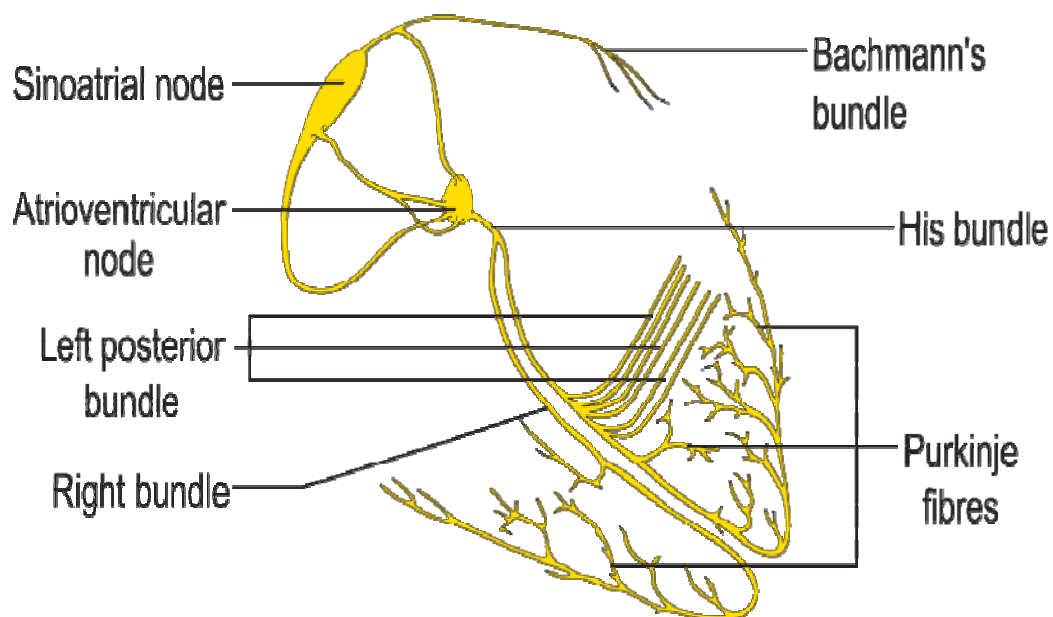
The atrio-ventricular node is located at the base of the right atrium, just above the junction of the atria and ventricles. The AV node is the ONLY point of contact between atria and ventricles i.e. in order from SA activity to reach the ventricles, it must go through the AV node. The AV node cells are unique in that they are smaller in diameter and thus have a smaller conduction velocity. This has two major implications:

1. The pacemaker activity of the AV node produces a slower firing rate (around 40-60/min compared with the SA node's 70-80/min). However, the signal generated by the SA node, if it reaches the AV node, overrides this.
2. The signal from the SA node is delayed at the AV (slow conduction) to allow atrial systole to precede ventricular systole.

#### *2.4 Ventricular Excitation:*

##### *2.4.1 Bundle of His and Purkinje Fibres*

After excitation reaches the AV node and atrial systole has ended, it spreads rapidly down the ventricles via specialized cells that form the Bundle of His (of which there is a left and a right bundle) and to the rest of the myocardium via similar Purkinje fibres. NB Purkinje fibres do not excite every cardiac cell- full excitation is completed by gap junction spread.



A graphical representation of the Electrical conduction system of the heart showing the Sinoatrial node, Atrioventricular node, Bundle of His, Purkinje fibers, and Bachmann's bundle (Photo credit: Wikipedia)

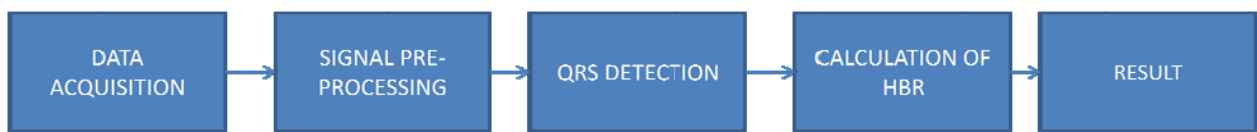
# CHAPTER 3

## BLOCK DIAGRAM OF THE SYSTEM

---

### *3.1 Introduction*

The first step is data acquisition. The data could be collected from real subjects but in this project it is collected from online database. Next step is signal pre-processing. In this step obtained ECG signals are filtered to remove noises. The third step is QRS detection which corresponds to the period of ventricular contraction or depolarization of heart.



Fourth step is to calculate the number of QRS complexes which corresponds to the heart beat rate of the signal.

### *3.2 ECG Data Acquisition.*

The ECG data used for the development and testing of the algorithm was taken from the Physionet library and specifically from the MIT-BIH Arrhythmia database. PhysioNet is an online forum for distribution and exchange of recorded biomedical signals and open-source analysis software, also providing facilities for cooperative analysis of data and evaluation of proposed new algorithms. PhysioBank is a large and growing archive of well-characterized digital recordings of biomedical signals for use by the biomedical researchers. It is freely accessible online through the World Wide Web (<http://www.physionet.org/>).

The database contains 23 records (numbered from 100 to 124, some numbers missing) chosen at random from this set, and 25 records (numbered from 200 to 234, again some numbers missing) selected from the same set to include a variety of rare but clinically important phenomena. Each of the 48 records is slightly over 30 minutes long. ECG records 100, 118 and 217 (ECG1, ECG2 and ECG3 ) are used in the training and records 101, 231 and 107 (ECG1, ECG2, ECG3) are used in testing of the neural network in this project. Each of these records is slightly over 30 minutes long, has a sample frequency of 360Hz and contains 2 channels (2 signals recorded from different angles on chest). By using Physionet's built-in

web tool only 1 minute long sections of each record is extracted as .m files that can be readily used in Matlab. As a result 6 recordings (3 for training and 3 for testing) each containing 21600 samples and approximately 60-80 waveforms depending on heart rate and class (normal, rbbb or paced) is obtained and loaded into Matlab environment. After loading the data into Matlab one of the channels is removed and only one channel for each recording is used for the rest of the program(channel MLII).

In the end of the Data acquisition, a total of 214 waveforms (as 3 separate recordings, each representing a ECG 1, ECG 2 and ECG 3 waveforms) for heart beat calculation are prepared ready for next step which is signal pre-processing.

# CHAPTER 4

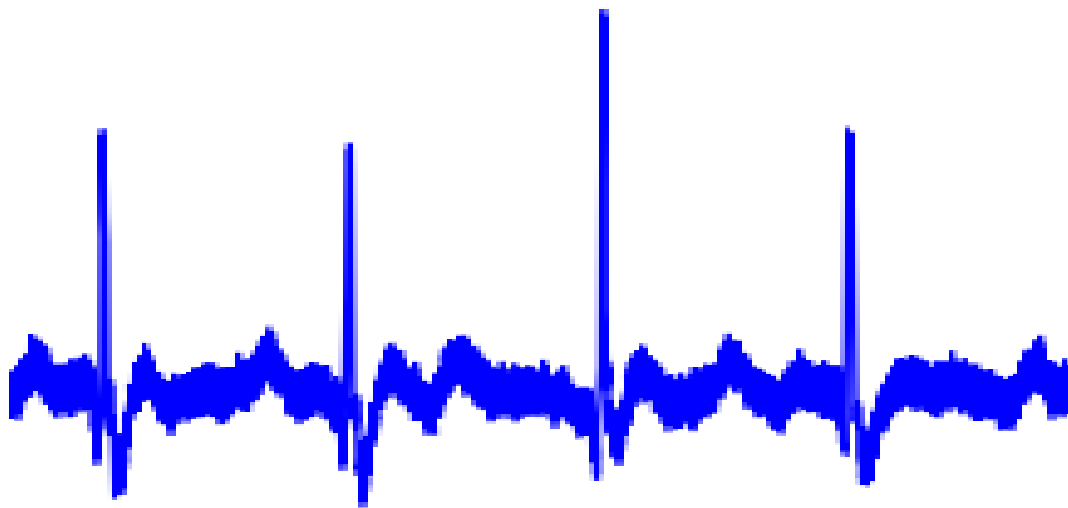
## SIGNAL PREPROCESSING

---

### *4.1 Introduction*

In general, the aim of preprocessing steps is to improve the general quality of the ECG for more accurate analysis and measurement. Noises may disturb the ECG to such an extent that measurements from the original signals are unreliable. The main categories of noise are: low frequency base line wander (BW) caused by respiration and body movements, high frequency random noises caused by mains interference (50 or 60Hz) and muscular activity and random shifts of the ECG signal amplitude caused by poor electrode contact and body movements, DC noise.

This project, at signal pre-processing step is focused on noise removal and after this step processing of the signal will continue with QRS detection. A number linear and non-linear technique has been developed to eliminate these artifacts. The preprocessing comprises of four steps: removal of DC noise, removal of base line wander (elimination of very low frequencies) removal of high frequency noise and QRS detection <sup>6</sup>.

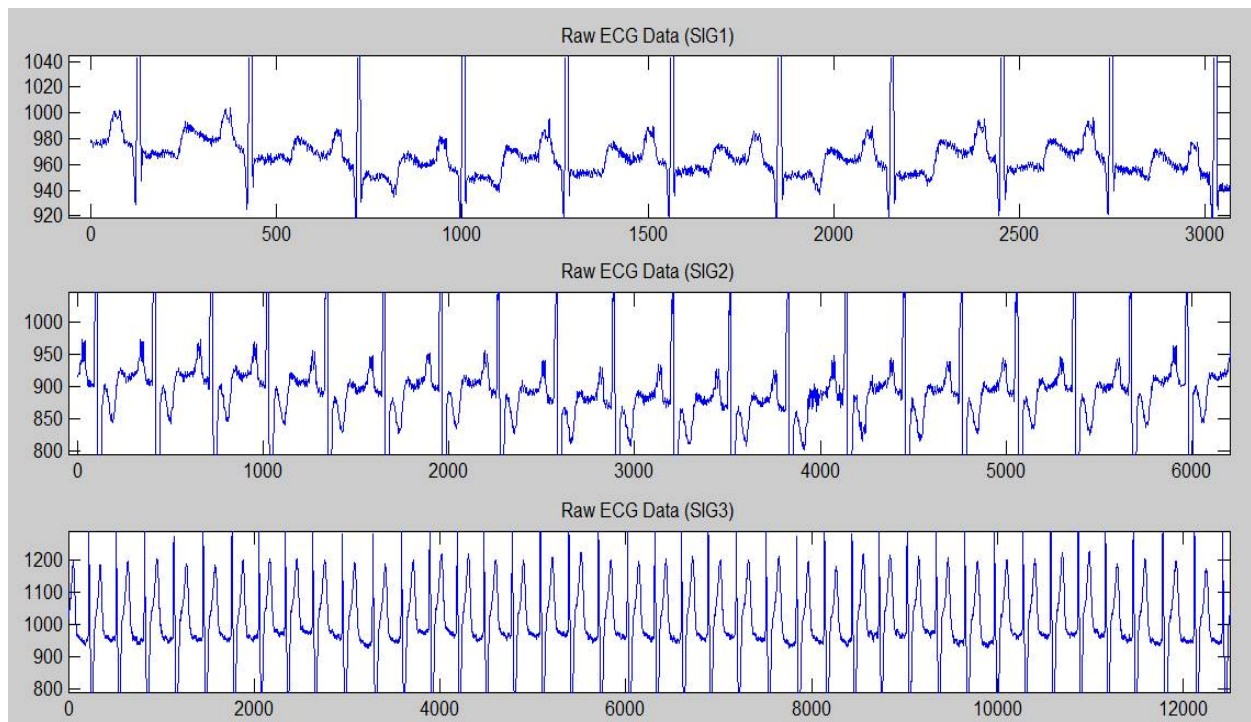


Typical ECG signal with noise

---

<sup>6</sup> <http://en.wikipedia.org/wiki/Electrocardiography>

## Sample Raw Noisy ECG Record Before Pre-processing



### *4.2 Types of Filters*

In this project, we have used the following types of filters to remove the unwanted noise from the ECG signal;

1. Low Pass Filter
2. High Pass Filter
3. Comb Filter (Band Stop Filter)

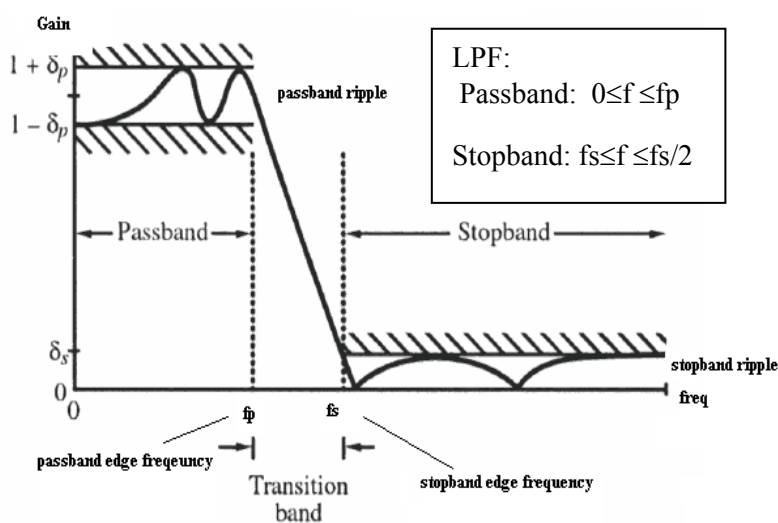
#### 4.2.1 Low Pass Filter:

A low-pass filter is a [filter](#) that passes low-[frequency signals](#) and [attenuates](#) (reduces the [amplitude](#) of) signals with frequencies higher than the [cutoff frequency](#). The actual amount of attenuation for each frequency varies depending on specific filter design. It is sometimes called a high-cut filter, or treble cut filter in audio applications. A low-pass filter is the opposite of a [high-pass filter](#).

The low pass filter has the property that low-frequency excitation signal components down to and including direct current, are transmitted, while high-frequency components, up to and including infinite ones are blocked. The range of low frequencies, which are passed, is called the pass band or the bandwidth of the filter. It extends from  $\omega=0$  to  $\omega=\omega_c$  rad/sec ( $f_c$  in Hz).

The highest frequency to be transmitted is  $\omega_c$ , which is also called the cutoff frequency.

Frequencies above cutoff are prevented from passing through the filter and they constitute the filter stopband.



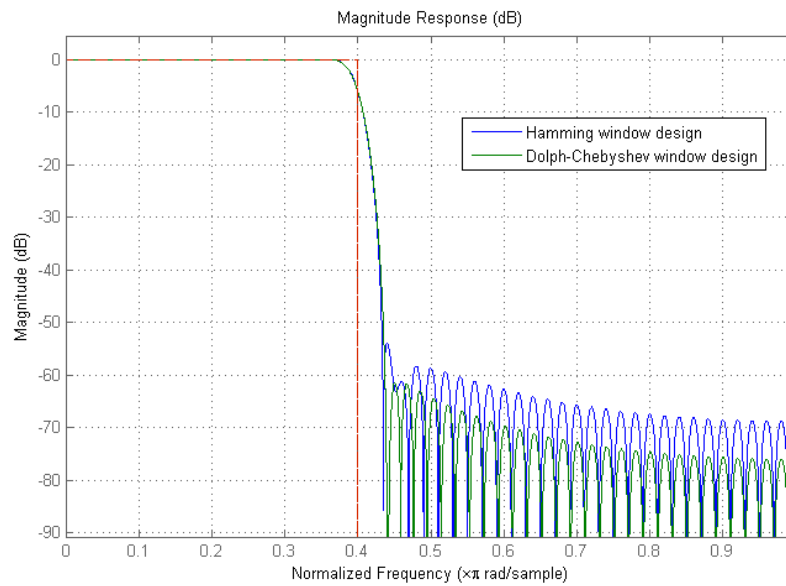
#### LPF

- Passes all low-frequency components below  $f_p$  and blocks all higher frequency components above  $f_s$

- In reality, you can't design 'square' type of filters so, there needs to be transition between the bands.

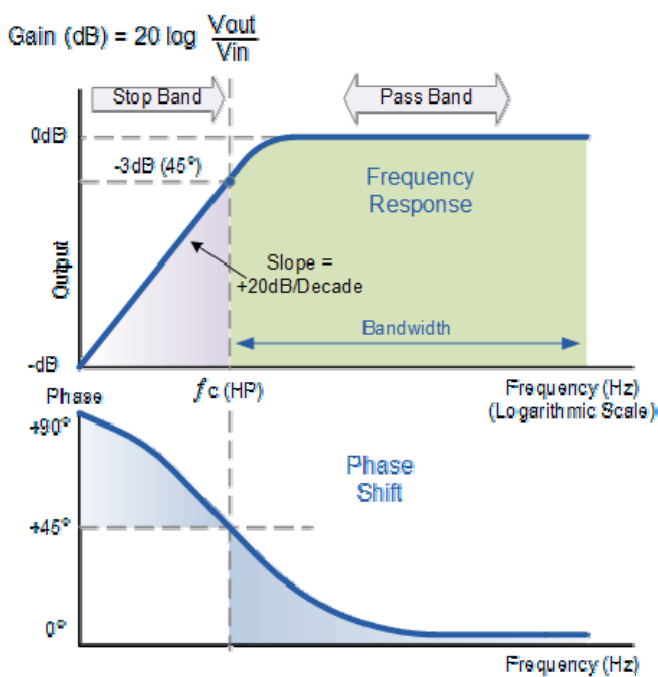
An ideal low pass filter

requires an infinite impulse response. Truncating (or windowing) the impulse response results in the so-called window method of FIR filter design. Consider a simple design of a low pass filter with a cutoff frequency of  $0.4\pi$  radians per sample



#### 4.2.2 High Pass Filter:

A high-pass filter (HPF) is an [electronic filter](#) that passes high-[frequency signals](#) but [attenuates](#) (reduces the [amplitude](#) of) signals with frequencies lower than the [cutoff frequency](#). The actual amount of attenuation for each frequency varies from filter to filter. A high-pass [filter](#) is usually modeled as a [linear time-invariant system](#). It is sometimes called a low-cut filter or bass-cut filter. High-pass filters have many uses, such as blocking DC from circuitry sensitive to non-zero average voltages or [RF](#) devices. They can also be used in conjunction with a [low-pass filter](#) to make a [bandpass filter](#).

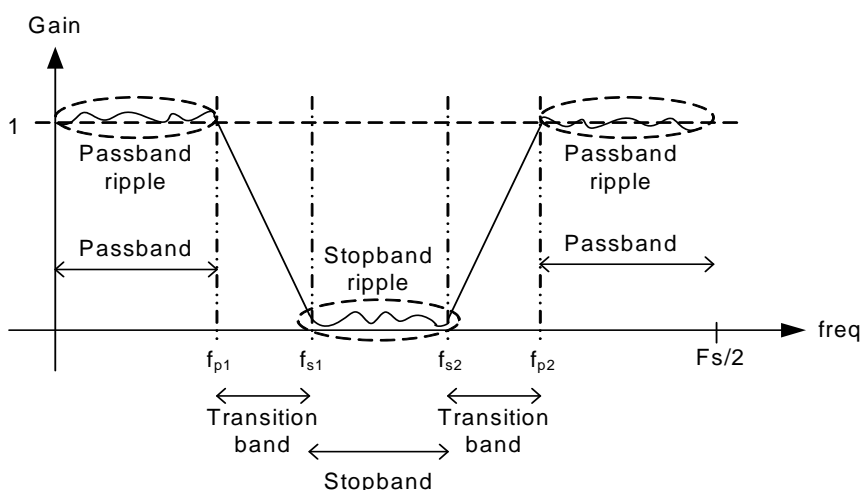


Passes all high-frequency components above the cut-off frequency,  $f_c$  and blocks all lower frequency components below  $f_c$ .



#### 4.2.3 Comb Filter (Band-stop Filter):

A comb filter is a [filter](#) that has a series of very deep notches in its [frequency response](#) with the spacing of all of the notches at multiples of the frequency of the lowest notch (they are all harmonically related). It got its name from looking like a comb when plotted on a frequency response graph. Comb filters are produced when a signal is time delayed and added back to itself. Some frequencies will cancel and others will be reinforced, which can dramatically change the tonal color of the sound. In practice this is common problem that occurs when a stereo mix is collapsed to mono because many stereo effects, such as chorus and flanging, achieve their stereo imaging by using some form of the [Haas effect](#). A static comb filter will make its audio sound kind of hollow or “phasey” depending upon how severe it is. Add some [modulation](#) and you have a flanger. Comb filtering is one of the main ingredients in the distinctive sound of a jet aircraft passing overhead. The difference in the time arrival to your ears of the direct sound versus the sound reflected off of the ground causes various frequencies to be cancelled or reinforced. As the plane moves these distances are all changing, thus causing the coloration of the sound to change. Again, it’s the same principle used in a flanger<sup>7</sup>.



Passes all frequency components lower and higher than edge passband frequencies,  $\text{freq}_{(\text{allow})} < f_{p1}$ ;  $\text{freq}_{(\text{allow})} > f_{p2}$  and blocks all frequencies between  $f_{s1} < \text{freq}_{(\text{block})} < f_{s2}$

<sup>7</sup> <http://www.mathworks.com/help/signal/ref/fdesign.lowpass.html>

### 4.3 The Preprocessing Steps

1. Removing DC Components of the ECG Signal
2. Removing Low Frequency and High Frequency Noise
3. Removing 60 Hz Power Line Interference

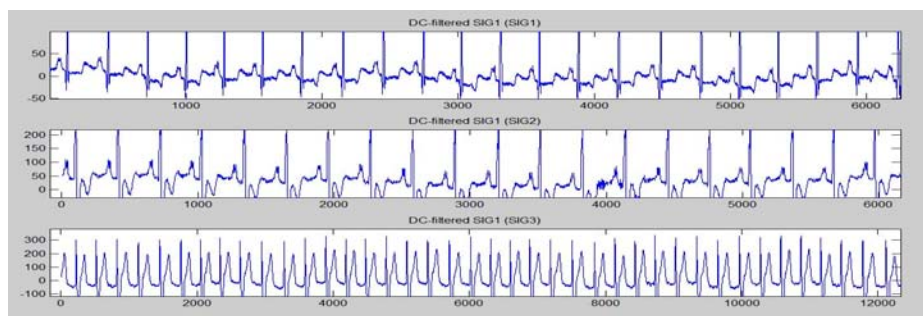
#### 4.3.1 Removing DC Components of the ECG Signal

As it can be clearly seen from Figure 3.4, ECG signals taken from MIT-BIH database contain baseline (sections of ECG where there is no electrical activity of heart) amplitudes higher than zero (around 950 in Figure 3.4). In this step by subtracting the mean of the signal from itself, the unwanted dc component is removed and the signal baseline amplitude is pulled back to level zero.

$$\text{ECGSignal} = \text{ECGSignal} - \text{mean}(\text{ECGSignal})$$

#### 4.2.2

##### Baseline



##### Wandering

Baseline wander (BW) is an extraneous low-frequency activity in the ECG which may interfere with the signal analysis, rendering clinical interpretation inaccurate and misleading. ST-T changes in the ECG are measured with reference to the isoelectric line. When BW is present the isoelectric line is no longer well defined and hence ST analysis becomes inaccurate. BW may result from various noise sources including perspiration, respiration, body movements, and poor electrode contact. Its spectral content is usually well below 1Hz, but may contain higher frequencies during strenuous exercise. Two major techniques are usually employed for BW removal, namely linear filtering and polynomial fitting. Linear filtering involves the design of an LTI high pass filter with cut off so that the clinical information in the ECG is preserved and as much as possible and the BW is removed. This cut off is based on the lowest heart rate which is around 40beats/min during bradycardia, which implies that,

the lowest 8Hz frequency component in the ECG 0.67Hz . Therefore the cut off chosen is around 0.5 Hz.<sup>8</sup>

#### *4.2.3 Removing Low Frequency and High Frequency Noise*

De-noising is the primary processing to remove all the high frequency as well as power supply interference from the ECG signal. ECG data used for the system contains low and high frequency noise components that may be caused by the sources explained in the previous chapter. Before the design of the software both frequency domain and time domain filters were tested for noise removal. It is observed that time domain filters provide better noise removal on the signals obtained from MIT-BIH database than frequency domain filters (butterworth filters in our case). Because of this and since the most of the noise present in the database are random noise time domain filters were chosen to filter unwanted high and low frequency noise.

To remove high frequency random noise, mostly caused by patients muscle contractions during recording, from the ECG signals a 10 point moving average (low pass) filter which passes low frequencies but attenuates high frequencies chosen and the signals are filtered by using Matlab's filter function

$$B=(1/10)*ones(1,10);$$

$$A=1;$$

$$ECGSignal=filter(B,A,ECGSignal)$$

After the removal of high frequency noise from the signal next step is to remove low frequency noise components. This low frequency noise shows itself as baseline wandering that is caused mostly by the respiration of the patient. To remove this low frequency noise, a derivative based (high pass) filter (Figure 3.6) that passes high frequencies but attenuates low frequencies used.<sup>9</sup>

$$B=(1/1.0025)*[1 \ -1];$$

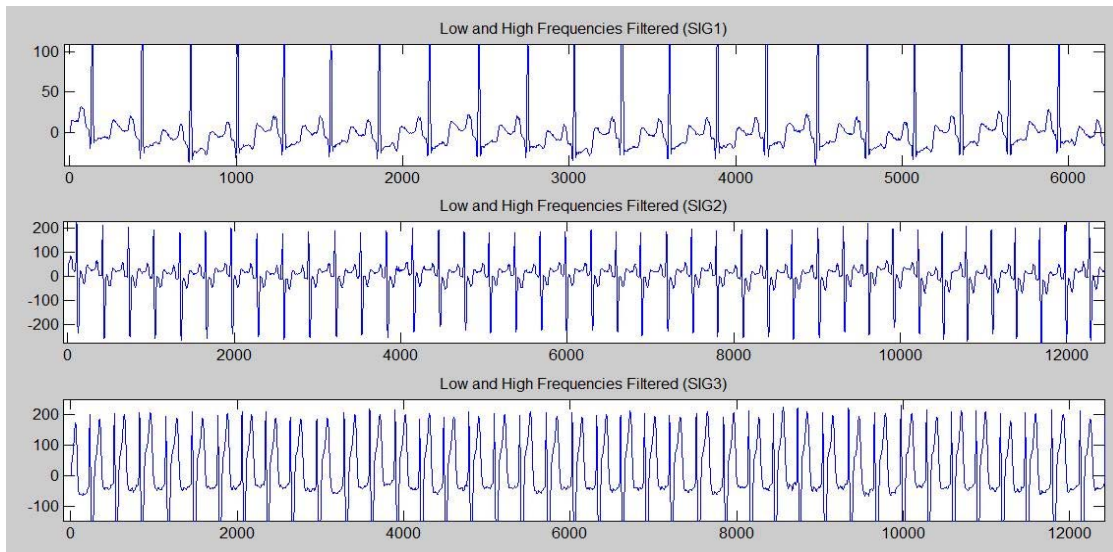
$$A=[1 \ -0.995];$$

---

<sup>8</sup> <http://www.ece.uic.edu/~jmorisak/blpf.html>

<sup>9</sup> [http://www.electronics-tutorials.ws/filter/filter\\_3.html](http://www.electronics-tutorials.ws/filter/filter_3.html)

ECGSignal=filter(B,A,ECGSignal)

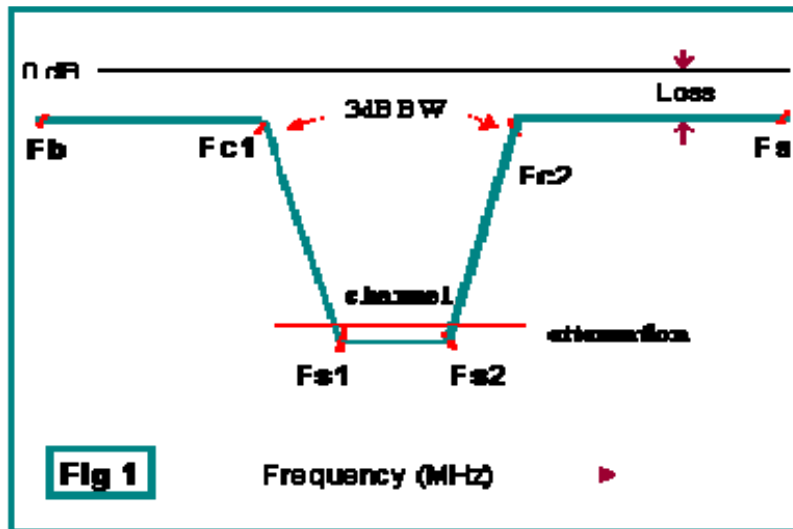


#### 4.2.4 Removing 60Hz Powerline Interference

Powerline interference is a noise caused by the electricity current flowing in wires and power lines. Powerline interference that is present in our ECG signals consists of 60Hz pickup and harmonics. Since frequency of 60Hz overlaps with our ECG signal frequency range we have to suppress only 60Hz frequency components and its harmonics without disturbing the frequencies around. To achieve this, comb filter is used and 60Hz powerline interference with its harmonics is removed from the ECG signals. Comb filter is a band-stop filter which attenuates a certain band of frequencies and their harmonics.<sup>10</sup>

---

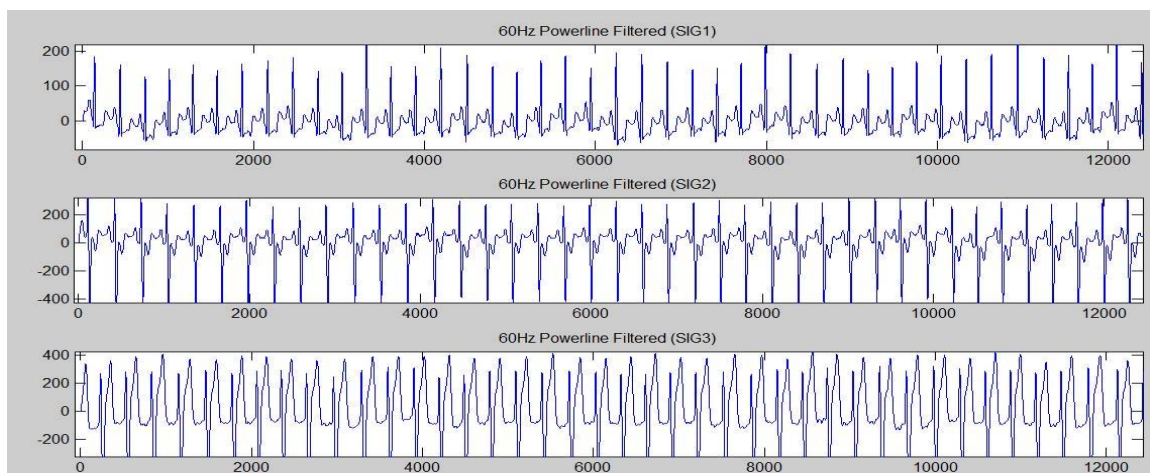
<sup>10</sup> <http://www.sweetwater.com/insync/comb-filter/>



```

B=conv([1 1],[0.6310 -0.2149 0.1512 -0.1288 0.1227 -0.1288 0.1512 -0.2149 0.6310]);
A=1;
ECGSignal=filter(B,A,ECGSignal)

```



# CHAPTER 5

## QRS DETECTION

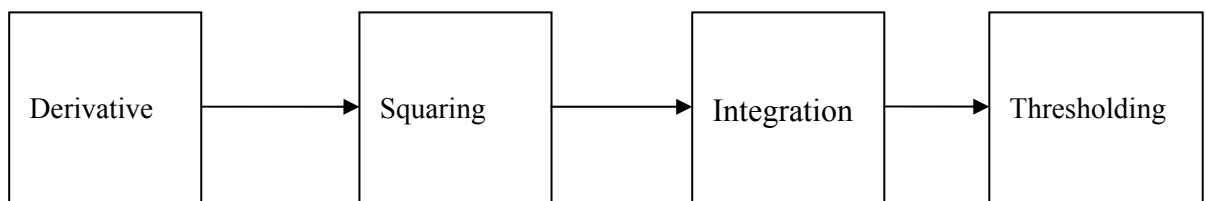
---

### *5.1 Introduction*

The main tasks of a QRS complex detector include detecting QRS complexes of heartbeats and generating a stable fiducial point for each individual heartbeat. The fiducial point and its placement should be robust and insensitive to subtle morphological variability in the QRS complex. In the literature, there are some excellent QRS complex detectors present [6,7] In this project however the real time QRS detector proposed by Pan and Tomkins has been used for its intrinsic simplicity and low computational complexity.

### *5.2 Pan-Tompkins QRS Detection Algorithm*

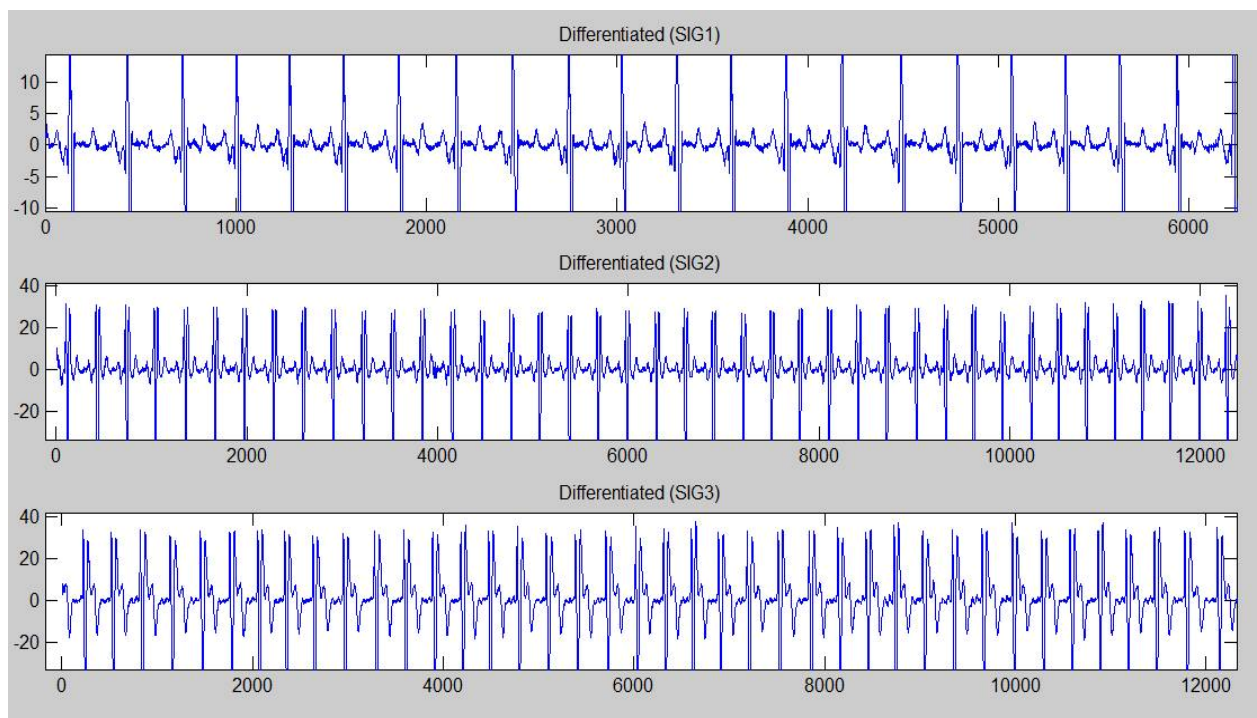
Pan and Tompkins proposed a real-time QRS detection algorithm based on slope, amplitude, and width of QRS complexes. The algorithm includes differentiation followed by squaring and then moving window integration. The information about the slope is obtained in the derivative stage. The squaring process intensifies the slope of the frequency response curve of the derivative and helps restrict the false positive waves caused by the T waves with higher than usual spectral energies. The moving window integrator produces a signal that includes information about both the width and slope of the QRS complex. Then a simple threshold is applied to detect the peaks.



### 5.2.1 Differentiation

The derivative procedure suppresses the low-frequency components of the P and T waves, and provides a large gain to the high-frequency components arising from the high slopes of the QRS complex. Derivative operation is implemented in Matlab by using diff function which finds the differences between the adjacent values in the signal.

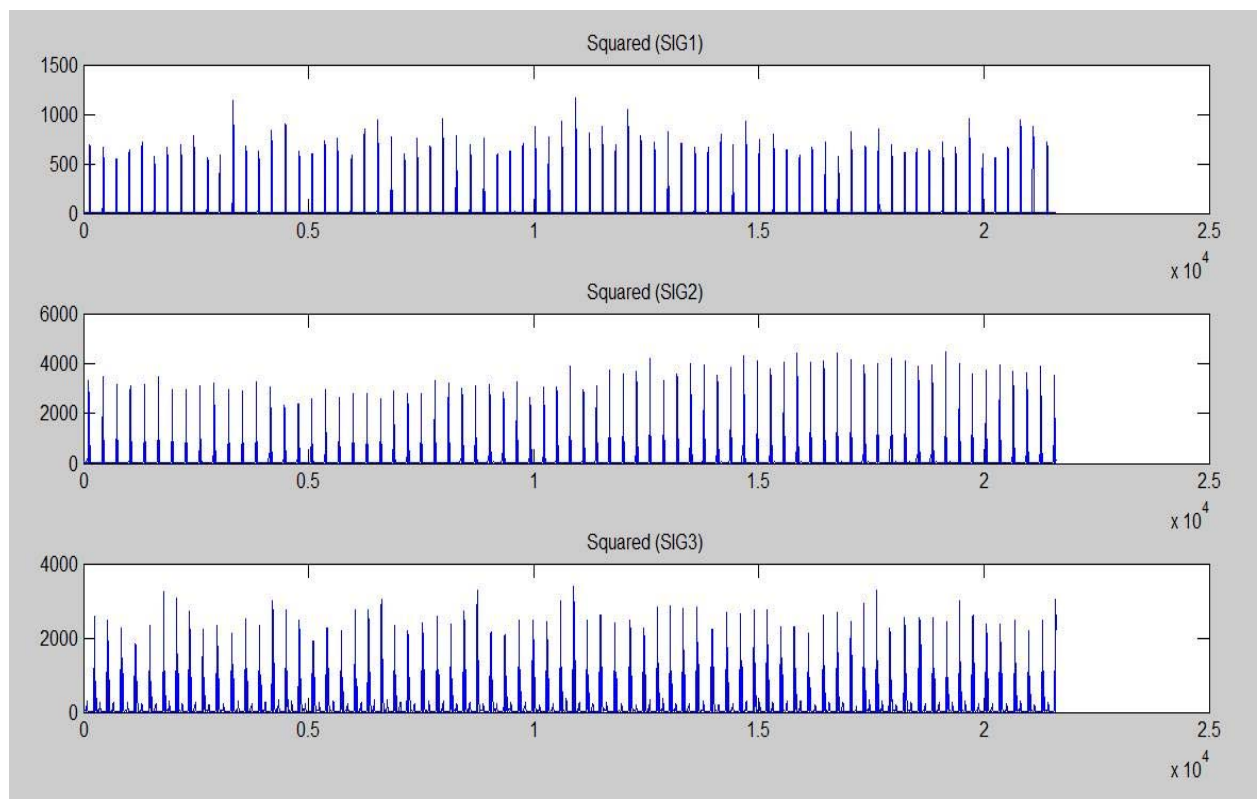
Derivative=diff(ECGSignal)



### 5.2.2 Squaring:

The squaring operation makes the result positive and emphasizes large differences resulting from QRS complexes; the small differences arising from P and T waves are suppressed. QRS complex is further enhanced. Squaring operation is implemented simply by multiplying the signal by itself in Matlab.

Squaring=derivative.\*derivative



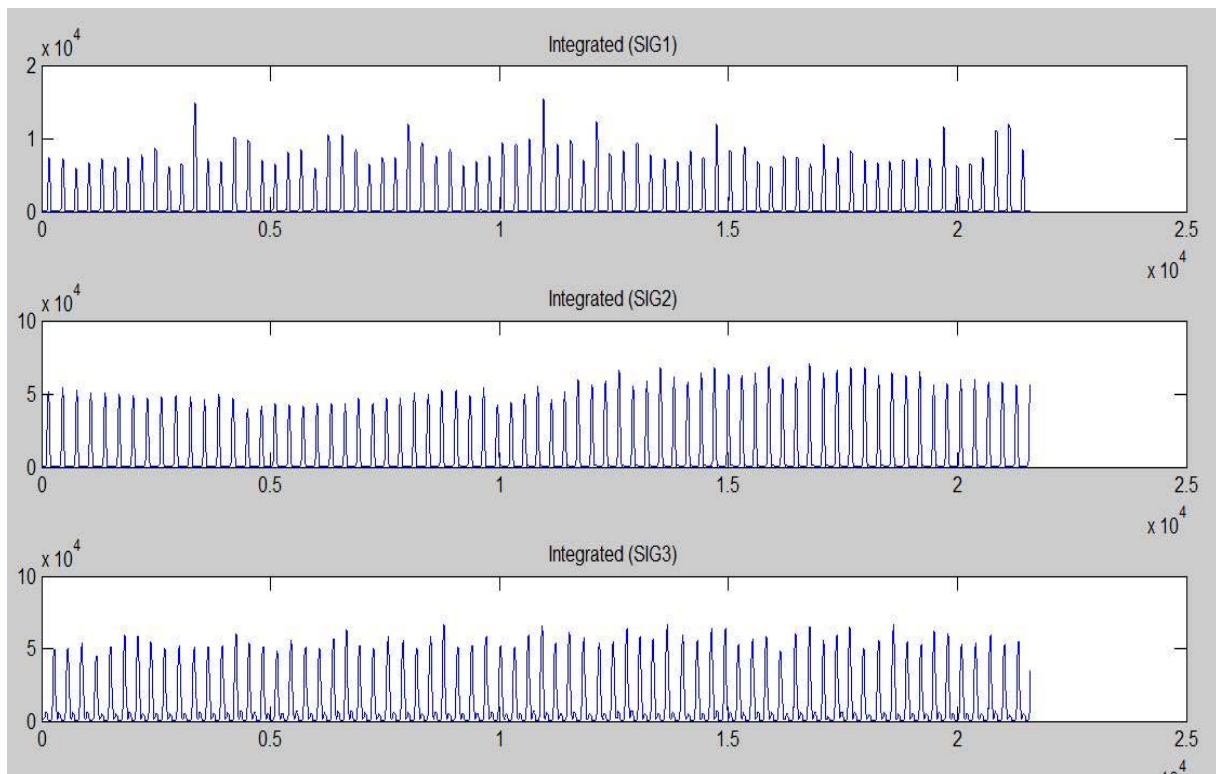


### 5.2.3 Integration

The output of a derivative based operation may contain multiple peaks within the duration of a single QRS complex. A moving window integrator is applied to perform smoothing of the output of the preceding operations so that multiple peaks are avoided. This step is performed in Matlab by using medfilt1 function and a window width of 54 is found to be suitable for sampling frequency 360Hz.

```
window=ones[1,54];
```

```
Integration=medfilt1(filter(window,1,squaring),10);
```

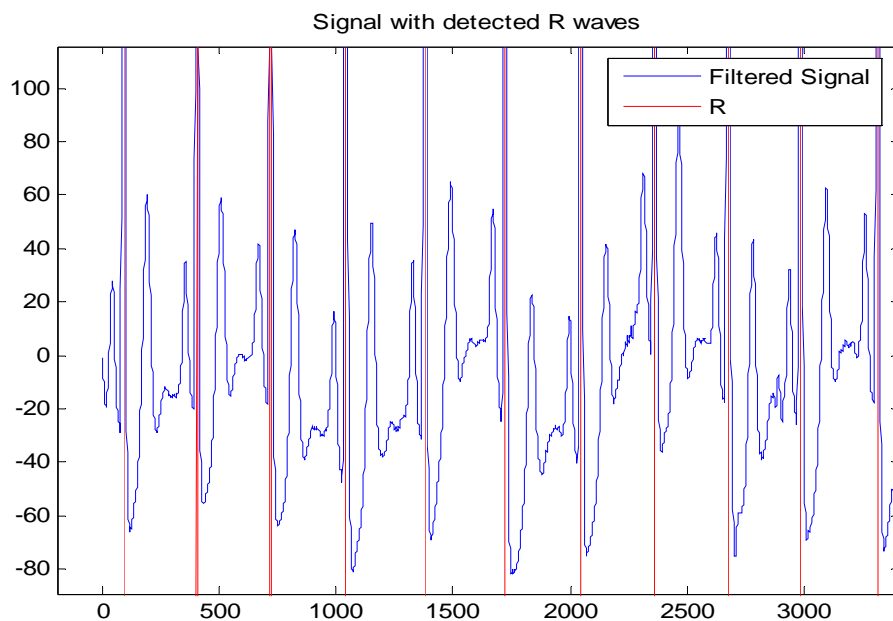


#### 5.2.4 Thresholding:

Maximum value of the signal that had passed from above steps is taken and multiplied by a threshold percentage value. This is done because the output of preceding operations may contain noise peaks. These noise peaks do not have as large amplitude as R peaks but if we take all the peaks present in the output of above steps as R peaks then noise peaks will also be classified as R peaks (QRS complexes). So by taking a certain percentage of the highest peak amplitude as a threshold we avoid this. Different values for threshold percentage were tested and value 0.2 found to be suitable for removing noise peaks in our signals. This threshold value is used for searching R peak in search procedures.

$\text{maxvalue} = \max(\text{integration})$

$\text{threshold} = \text{maxvalue} * 0.2$



# CHAPTER 6

## SEARCH PROCEDURES FOR QRS

---

### *6.1 Location of R Peaks*

In the last step of QRS detection, regions of the output signal, of the preceding steps, that is above the threshold value is found. Starting and ending locations of each region is recorded. Then each specific region is again searched on the original ECG signal for a maximum value which represents the exact R peak of that wave. Locations of all R peaks are then recorded and the QRS searching algorithm is finalized .

```
position_region=integration>threshold
```

```
left=find(diff([0 position_region])==1)
```

```
right=find(diff([position_region 0])==1)
```

```
for i=1:length(left)
```

```
    [maxvalue(i) maxlocation(i)]=max(ECGSignal(left(i):right(i)))
```

```
end
```

### *6.2 Calculation of Hear Beat Rate*

At this stage in the project, we have been able to process the ECG signals and are left with the determination of the heart beat rate. The number of R peaks have also been calculated which corresponds to the number of “Left Leg” of each QRS complex as detected by the Pan-Tompkins Algorithm since no two QRS complex had two “left legs”.

The matlab code below counts the number of left legs of the ECG signals

```
for i=1:length(left)
    [maxvalue(i) maxlocation(i)]=max(ECGSignal(left(i):right(i)))
end
```

### *6.3 Result*

- The Heart Beat Rate of SIG1 is 74
- The Heart Beat Rate of SIG2 is 72
- The Heart Beat Rate of SIG3 is 71

### *6.4 Discussion*

In this project, three ECG signals were taken from the MIT data base, they were processed and the heart beat rates are determined in a very effective manner. These heart beats rates were evaluated using Matlab codes which has proven to be simple intelligent and cost effective

# CHAPTER 7

## CONCLUSION & FUTURE WORK

---

In conclusion, a method based on QRS detection algorithm was used in order to process the signal and identify the QRS complex. This algorithm is called Pan-Tompkins algorithm and is made up of the following procedures:

- a) Differentiation
- b) Squaring
- c) Integration
- d) Thresholding.

This algorithm processes the signal and is then followed by QRS searching procedure which produces the number of R-peaks contained in the signal and this number corresponds to the heart beat rate of the individual ECG signals.

In the future this work could be extended as an ECG-based, automated diagnostic tool for various diseases of the cardiovascular system. More work is necessary in order to detect the P and T waves as well, which provide additional information about specific functionalities of the heart. Of course, additional research leading to successful detection and classification of the various types of abnormal beats will be the hallmark of the success of this algorithm as a diagnostic tool.

Real-time patient data acquisition can be added and more advanced filters can be designed to remove real-time dynamic noises. More complex feature extraction methods can be applied. System can be adapted to classify most of the cardiac conditions that can be faced in a clinical environment.

# REFERENCES

---

- [1] <http://en.wikipedia.org/wiki/Electrocardiography>
- [2] Amal Mattu, William Brady, “*ECGs for the Emergency Physician*”, BMJ Books, BMA House, Tavistock Square, London WC1H 9JR, pp. ix, 2003.
- [3] Carsten Meyer, Jose Fernandez Gavela, Matthew Harris, “*Combining Algorithms in Automatic Detection of QRS Complexes in ECG Signals*”, IEEE transactions on Information Technology in Medicine, Vol. 10, No.3, July 2003.
- [4] Franco Chiarugi, “*New Developments in the Automatic Analysis of the Surface ECG: The Case of Atrial Fibrillation*”, Hellenic Journal of Cardiology 49, pp.207--- 221, 2008.
- [5] Bert-Uwe Kohler, Carsten Hennig, Reinhold Orglmeister, “*The Principles of Software QRS Detection: Reviewing and Comparing Algorithms for Detecting this Important ECG Waveform*”, IEEE Engineering in Medicine and Biology, February, 2002.
- [6] <http://www.ece.uic.edu/~jmorisak/blpf.html>
- [7] [http://en.wikipedia.org/wiki/Low-pass\\_filter](http://en.wikipedia.org/wiki/Low-pass_filter)
- [8] [http://www.electronics-tutorials.ws/filter/filter\\_3.html](http://www.electronics-tutorials.ws/filter/filter_3.html)
- [9] <http://www.sweetwater.com/insync/comb-filter/>

# APPENDIX

## Matlab Codes For The Project

---

```
%Loading ECG DATA
%www.physionet.org/physiobank/database/html/mitdbdir/mitdbdir.htm
%Each 1 min record contains approximately 60-80 waveforms
%sample rate of the signals=360Hz
%loading ECG SIG1,SIG2 and SIG3 waveform recordings

disp('Loading ECG Records That Will Be Used In Heart Beat Rate Calculation')
SIG1=load('C:\Users\USER\Desktop\100m.mat'); %1min record containing SIG1 beat
waveforms
SIG2=load('C:\Users\USER\Desktop\118m.mat'); %1min record containing SIG2 beat
waveforms
SIG3=load('C:\Users\USER\Desktop\217m.mat'); %1min record containing SIG3 beat
waveforms

%above ecg .mat files contains 2 channel ECG records.
%we only need one channel so extracting only one of the channels
%(channel MLII)

SIG1=SIG1.val(1,:);
SIG2=SIG2.val(1,:);
SIG3=SIG3.val(1,:);

%%%%%%optional plot%%%%%%%%
subplot(3,1,1);plot(SIG1)
title('Raw ECG Data (SIG1)')
subplot(3,1,2);plot(SIG2)
title('Raw ECG Data (SIG2)')
subplot(3,1,3);plot(SIG3)
title('Raw ECG Data (SIG3)')
figure

%mean filter code to remove DC Components noise
SIG1=SIG1-mean(SIG1);
plot(SIG1);
title('Mean-filtered SIG1 ecg')
figure

SIG2=SIG2-mean(SIG2);
plot(SIG2);
title('Mean-filtered SIG2 ecg')
figure
```

```
SIG3=SIG3-mean(SIG3);
plot(SIG3);
title('Mean-filtered SIG3 ecg')
figure
```

```
subplot(3,1,1);plot(SIG1)
title('DC-filtered SIG1 (SIG1)')
subplot(3,1,2);plot(SIG2)
title('DC-filtered SIG1 (SIG2)')
subplot(3,1,3);plot(SIG3)
title('DC-filtered SIG1 (SIG3)')
figure
```

**%Removing Low and High Frequency Noise**

**%Low Frequency**

```
B=(1/10)*ones(1,10);
A=1;
SIG1=filter(B,A,SIG1);
```

```
B=(1/10)*ones(1,10);
A=1;
SIG2=filter(B,A,SIG2);
```

```
B=(1/10)*ones(1,10);
A=1;
SIG3=filter(B,A,SIG3);
```

**%High Frequency**

```
B=(1/1.0025)*[1 -1];
A=[1 -0.995];
SIG1=filter(B,A,SIG1);
```

```
B=(1/1.0025)*[1 -1];
A=[1 -0.995];
SIG2=filter(B,A,SIG2);
```

```
B=(1/1.0025)*[1 -1];
A=[1 -0.995];
SIG3=filter(B,A,SIG3);
```

```
subplot(3,1,1);plot(SIG1)
title('Low and High Frequencies Filtered (SIG1)')
subplot(3,1,2);plot(SIG2)
title('Low and High Frequencies Filtered (SIG2)')
subplot(3,1,3);plot(SIG3)
title('Low and High Frequencies Filtered (SIG3)')
figure
```

**%60Hz Powerline interference Removal**

```
B=conv([1 1],[0.6310 -0.2149 0.1512 -0.1288 0.1227 -0.1288 0.1512 -0.2149 0.6310]);
```



```

A=1;
SIG1=filter(B,A,SIG1);

B=conv([1 1],[0.6310 -0.2149 0.1512 -0.1288 0.1227 -0.1288 0.1512 -0.2149 0.6310]);
A=1;
SIG2=filter(B,A,SIG2);

B=conv([1 1],[0.6310 -0.2149 0.1512 -0.1288 0.1227 -0.1288 0.1512 -0.2149 0.6310]);
A=1;
SIG3=filter(B,A,SIG3);

subplot(3,1,1);plot(SIG1)
title('60Hz Powerline Filtered (SIG1)')
subplot(3,1,2);plot(SIG2)
title('60Hz Powerline Filtered (SIG2)')
subplot(3,1,3);plot(SIG3)
title('60Hz Powerline Filtered (SIG3)')
figure

```

#### %QRS Detection Using Pan-Tompkins Algorithm

##### %Differentiation

```
Derivative_SIG1=diff(SIG1);
```

```
Derivative_SIG2=diff(SIG2);
```

```
Derivative_SIG3=diff(SIG3);
```

```

subplot(3,1,1);plot(Derivative_SIG1)
title('Differentiated (SIG1)')
subplot(3,1,2);plot(Derivative_SIG2)
title('Differentiated (SIG2)')
subplot(3,1,3);plot(Derivative_SIG3)
title('Differentiated (SIG3)')
figure

```

##### %Squaring

```
Squaring_SIG1=(Derivative_SIG1.*Derivative_SIG1);
```

```
Squaring_SIG2=(Derivative_SIG2.*Derivative_SIG2);
```

```
Squaring_SIG3=(Derivative_SIG3.*Derivative_SIG3);
```

```

subplot(3,1,1);plot(Squaring_SIG1)
title('Squared (SIG1)')
subplot(3,1,2);plot(Squaring_SIG2)
title('Squared (SIG2)')
subplot(3,1,3);plot(Squaring_SIG3)
title('Squared (SIG3)')
figure

```

```
%Integration
```

```
window=ones(1,54);
```

```
Integration_SIG1=medfilt1(filter(window,1,Squaring_SIG1),10);
```

```
Integration_SIG2=medfilt1(filter(window,1,Squaring_SIG2),10);
```

```
Integration_SIG3=medfilt1(filter(window,1,Squaring_SIG3),10);
```

```
subplot(3,1,1);plot(Integration_SIG1)
```

```
title('Integrated (SIG1)')
```

```
subplot(3,1,2);plot(Integration_SIG2)
```

```
title('Integrated (SIG2)')
```

```
subplot(3,1,3);plot(Integration_SIG3)
```

```
title('Integrated (SIG3)')
```

```
figure
```

```
%Thresholding
```

```
maxvalue=max(Integration_SIG1);
```

```
Threshold_SIG1=maxvalue*0.2;
```

```
maxvalue=max(Integration_SIG2);
```

```
Threshold_SIG2=maxvalue*0.2;
```

```
maxvalue=max(Integration_SIG3);
```

```
Threshold_SIG3=maxvalue*0.2;
```

```
subplot(3,1,1);plot(Threshold_SIG1)
```

```
title('Threshold (SIG1)')
```

```
subplot(3,1,2);plot(Threshold_SIG2)
```

```
title('Threshold (SIG2)')
```

```
subplot(3,1,3);plot(Threshold_SIG3)
```

```
title('Threshold (SIG3)')
```

```
figure
```

```
%QRS Searching
```

```
position_region_SIG1=Integration_SIG1>Threshold_SIG1;
```

```
left=find(diff([0 position_region_SIG1])==1);
```

```
right=find(diff([position_region_SIG1 0])==1);
```

```
for i=1:length(left)
```

```
[maxvalue_SIG1(i) maxlocation_SIG1(i)]=max(Integration_SIG1(left(i):right(i)));
```

```
end
```

```
%plot(maxvalue_SIG1);
```

```
%title('maxvalue_SIG1 ecg')
```

```
%figure
```

```

plot(maxlocation_SIG1);
title('maxlocation_SIG1')
figure

```

```

position_region_SIG2=Integration_SIG2>Threshold_SIG2;
left=find(diff([0 position_region_SIG2])==1);
right=find(diff([position_region_SIG2 0])==1);

```

```

for i=1:length(left)
[maxvalue_SIG2(i) maxlocation_SIG2(i)]=max(Integration_SIG2(left(i):right(i)));
end

```

```

%plot(maxvalue_SIG2);
%title('maxvalue_SIG2 ecg')
%figure

```

```

plot(maxlocation_SIG2);
title('maxlocation_SIG2')
figure

```

```

position_region_SIG3=Integration_SIG3>Threshold_SIG3;
left=find(diff([0 position_region_SIG3])==1);
right=find(diff([position_region_SIG3 0])==1);

```

```

for i=1:length(left)
[maxvalue_SIG3(i) maxlocation_SIG3(i)]=max(Integration_SIG3(left(i):right(i)));
end

```

```

%plot(maxvalue_SIG3);
%title('maxvalue_SIG3 ecg')
%figure

```

```

plot(maxlocation_SIG3);
title('maxlocation_SIG3')
figure

```

```

%subplot(3,1,1);plot(position_region_SIG1)
%title('position_region (SIG1)')
%subplot(3,1,2);plot(position_region_SIG2)
%title('position_region (SIG2)')
%subplot(3,1,3);plot(position_region_SIG3)
%title('position_region (SIG3)')
%figure

```

```

disp('The Heart Beat Rate of SIG1 is ')
disp(length(maxlocation_SIG1));

```

```
disp('The Heart Beat Rate of SIG2 is ')
disp(length(maxlocation_SIG2));
```

```
disp('The Heart Beat Rate of SIG3 is ')
disp(length(maxlocation_SIG3));
```

```
%%%%optional plot%%%%
```

```
maxposition_SIG1=ones(1,21600)*(-max(SIG1)); %putting
```

```
%%%%max value to R locations inorder to plot R locations
```

```
maxposition_SIG1(1,maxlocation_SIG1)=max(SIG1);
```

```
maxposition_SIG2=ones(1,21600)*(-max(SIG2));
```

```
maxposition_SIG2(1,maxlocation_SIG2)=max(SIG2);
```

```
maxposition_SIG3=ones(1,21600)*(-max(SIG3));
```

```
maxposition_SIG3(1,maxlocation_SIG3)=max(SIG3);
figure
```

```
subplot(3,1,1);
plot(SIG1)
title('Signal with detected R waves');
```

```
hold on
plot(maxposition_SIG1,'r')
legend('Filtered Signal','r');
```

```
subplot(3,1,2);plot(SIG2)
hold on
plot(maxposition_SIG2,'r')
```

```
subplot(3,1,3);plot(SIG3)
hold on
plot(maxposition_SIG3,'r')
```