

Wireless post-processing and interfacing ECG, blood pressure and blood oxygen measurement systems

By

Kundan Thind

Electrical and Biomedical Engineering Design Project (4BI6)

Department of Electrical and Computer Engineering

McMaster University

Hamilton, Ontario, Canada

Wireless post-processing and interfacing ECG, blood pressure and blood oxygen measurement systems

By

Kundan Thind

Electrical and Biomedical Engineering

Faculty Advisor: Dr. Jim Reilly

Electrical and Biomedical Engineering Design Project Report

Submitted in partial fulfillment of degree of

Bachelor of Engineering

McMaster University

Hamilton, Ontario, Canada

April 27, 2009

Copyright © April, 2009 by Kundan Thind

ABSTRACT

Canadian hospitals are known to have high wait times. This is especially prominent in walk-in clinics. One of the basic diagnoses that every patient in walk-in clinic has to go through includes checking the blood pressure and health of the heart. The physician usually does this himself. Quick Doc is a portable system that measures ECG, blood oxygen content and blood pressure. Quick Doc could be used to perform the basic vital measurements and would help save physician's valuable time. The way it works is that the transducers communicate the measured signals to the base station via a wireless link, where the signals are processed to yield meaningful result. The physiological signals from the human body in question range in mV. These signals are amplified to a range of Volts before transporting via the wireless link. Due to their inherent nature, a lot of associated noise is present. The noise is contributed from a variety of sources that range anywhere from unwanted muscle activity to 60Hz power interference. Most of the processing is done at the base station. This post-processing at the base station involves heavy filtering that provides an output, which is valuable information as the interpretation of the information could lead to simple predictions in quantitative measure of blood pressure, heart rate and blood oxygen content. These measures in turn, serve as the basis of many diagnostics in medicine. The report will center on assimilation of different ideas that led to Quick Doc. The specific focus will be on interfacing the signals from body to the base station with a wireless link as well as post-processing of these signals. The end result is electrocardiogram signal, blood pressure and blood oxygen content that is mostly consistent with commercial devices available in the market. This report goes step by step from acquiring signals to complex processing to yield a valid output. These outputs are presented as numbers, similar to what commercial devices produce. In addition, experiments with different techniques, failure of certain methodologies, and findings from those mistakes are also presented.

Key words: ECG, blood pressure, Blood oxygen, wireless, post-processing, filtering, noise

Acknowledgements

I would like to express my gratitude to Dr. Jim Reilly, who acted as advisor for project Quick Doc. I would also like to thank Dr. Thomas E. Doyle, the supervisor for the project. Their excellent guidance guided me throughout the project. I would like to also thank Jason Thong, Teaching assistant for his assistance in the lab.

I would like to thank my colleagues Hamzah Qureshi, Ashwin Ayyaswamy and Doralice Ferreira for the assistance and constant support.

Kundan Thind

Table of Contents

Abstract		iii
Acknowledgements		iv
Contents		v
List of Tables		7
List of Figures		8
Nomenclature		10
1	Introduction	11
	1.1 Background	11
	1.2 Objective and scope	13
	1.3 Methodology	15
	1.4 Conclusion	18
2	Literature	19
	2.1 Processing guidelines	19
3	Statement of problem and methodology of solution	21
	3.1 Statement of Problem	21
	3.2 Theoretical background	25
	3.2.1 Electrocardiography Signal	25
	3.2.2 Blood pressure	26
	3.2.3 Blood oxygen content	26
	3.2.4 Wireless module	27
	3.2.5 Data acquisition system	27
	3.2.6 Post-processing	28
4	Hardware Design	29
	4.1 Wireless link	29
	4.1.1 Transmitter	30
	4.1.2 Receiver	31

4.1.3	Antenna	32
4.2	Data Acquisition system	33
5	Software design implementation	35
5.1	Introduction	35
5.2	Electrocardiography algorithm	35
5.3	Blood pressure algorithm	42
5.4	Blood oxygen saturation algorithm	48
6	ECG case study: arrhythmia	53
6.1	Introduction	53
6.2	Case study algorithm	53
7	Results and discussion	60
7.1	Wireless module	60
7.2	Data acquisition system	60
7.3	Electrocardiography signal	61
7.4	Blood pressure measurement	61
7.5	Blood oxygen content measurement	62
8	Conclusions and recommendations	64
A		65
A.1	MATLAB code for importing input data	65
A.2	MATLAB code for ECG processing	67
A.3	MATLAB code for blood pressure measurement	72
A.4	MATLAB code for blood oxygen content measurement	74
Bibliography		77
Vitae		79

List of Tables

Table 5.1 Table describing filter characteristic used in ECG algorithm	41
Table 5.2 Table describing filter characteristic used in blood pressure algorithm	46
Table 5.3 Table describing filter characteristic used in blood oxygen saturation algorithm	53

List of Figures

Figure 1.1 Schematic of the Quick Doc system [1]	15
Figure 3.1 Basic block layout for Quick Doc, the area outlined represents the personal contribution towards the project [2]	22
Figure 3.2 Schematic representation of a normal ECG [3]	25
Figure 4.1 Picture of receiver and transmitter module [4]	29
Figure 4.2 Simple schematic of the wireless transmitter used in Quick Doc [4]	30
Figure 4.3 Simple schematic of wireless receiver used in Quick Doc [4]	31
Figure 4.4 Antenna dimensions [4]	32
Figure 4.5 Picture of the data acquisition system used in Quick Doc [5]	33
Figure 5.1 A flow chart representing the algorithm for ECG processing	35
Figure 5.2 Raw ECG data	36
Figure 5.3 Raw ECG data after FFT manipulation	37
Figure 5.4 ECG data after removal of power noise	38
Figure 5.5 ECG after passing through a low pass filter	39
Figure 5.6 ECG data after a smoothing window	40
Figure 5.7 The green stars represent the local peaks in ECG signal	41
Figure 5.8 A flow chart representing the algorithm for Blood pressure measurement processing	42
Figure 5.9 RAW blood pressure input data	43
Figure 5.10 Graph illustrating the original blood pressure signal and amplified peaks signal	46
Figure 5.11 Picture of commercial device with blood pressure reading	47
Figure 5.12 A flow chart representing the algorithm blood oxygen saturation algorithm	48
Figure 5.13 Infra red and red data from the transducer for blood oxygen content	49

Figure 5.14 Infra red diode signal after filtering, Fourier manipulation and peak detection	51
Figure 5.15 Red diode signal after filtering, Fourier manipulation and peak detection	52
Figure 5.16 Hamzah Qureshi using his blood oxygen saturation module	53
Figure 6.1 A flow chart representing the algorithm for ECG processing	54
Figure 6.2 Raw ECG case study data	55
Figure 6.3 ECG case study data after Fourier manipulation	56
Figure 6.4 ECG case study data after power noise removal	57
Figure 6.5 ECG case study data after passing through a low pass filter	57
Figure 6.6 the output after implementation of a smoothing window	58
Figure 6.7 ECG case study data zoomed in	59
Figure 6.8 ECG case study after peak detection, discrepancy between different peaks	60

Nomenclature

- *ECG* – Electrocardiography, electrical activity of the heart
- *P wave* – component of ECG representing atria depolarization
- *QRS complex* – component of ECG representing ventricular depolarization
- *T wave* – component of ECG representing ventricular repolarization
- *MAP*- Mean Arterial Pressure; represents the average pressure in systemic circulation
- *Systolic Pressure* – Highest pressure in systemic circulation
- *Diastolic Pressure* – Lowest pressure in systemic circulation
- *LPF* - Low pass Filter, used to remove high frequency components
- *HPF* – High Pass Filter, used to remove low frequency components
- *Notch Filter* – filter used to attenuate specific frequency in frequency domain such as noise from power lines
- *Fourier Transform* – provides a frequency domain representation of a signal in time domain
- *DC offset* – Direct Current offset representative of a constant Voltage value
- *Peak detection* – finding local peaks in the given signal

Chapter 1

Introduction

1.1 Background

Health care is one of the primary concerns in one's life. The value of life would not be the same without it. In a country such as Canada, free healthcare through medicare provides a worry free lifestyle concerning health care. This universal health care system is not implemented in many other countries. Even though the system is well established, there are some minor flaws that present some concerns.

One of the biggest concerns is the wait time in hospitals and clinics. The following facts were presented in 2004 by CBC (Canadian Broadcasting Corporation) in collaboration with a number of organizations including: The College of Family Physicians, the Canadian Medical Association, and the Royal College of Physicians and Surgeons:

- Average number of patients seen by a family physician per week is 117 [6]
- Total hours worked by a family physician are 70-80 per week. This includes the on-call time and about 70% of physicians are on-call [6]
- New patients are not accepted by approximately 60% of the existing doctors [6]
- Only 20.2 % of physicians reported acceptance of new patients [6]

The Canadian Family Physician organization performed a poll in 2006 which indicates around 17% of Canadians do not have family physicians. This amounts to around 5 million Canadian residents. It was also found that around 2 million of these 5 million cannot find a doctor [7]. These findings along with the aforementioned stats prove that we need more health care professionals. Increasing the number of health care professionals requires a big investment in medical schools and hospitals that are

vital to the training of doctors. Although the expansion of facilities is happening slowly, an alternative approach could be to improve the existing facilities to cut the waiting times.

Improving the existing instrumentation could help in cutting wait times. Constant improvements in the instrumentation have helped in bringing down wait times and increasing patient comfort. Further improvements could be made in the management system for patients. One of the most intriguing ways in which patient wait time is used to reduce the overall wait time is seen in some optometry clinics. As the patient checks in, a scan is performed on the auto-refractor machine. This machine prints out an approximation of the eye lens weakness, myopic or hypermetropic condition as well as the astigmatism angle. This gives a general idea to the optometrist where to begin with the eye exam. This saves the actual time spent in the eye exam room and utilizes the patient wait time while saving Optometrist's time as well.

The idea of Quick Doc sparked from this management system implemented in optometry clinics. The most common diagnosis performed by a physician as a patient walks into the clinic is blood pressure, health of heart and temperature. Quick Doc is meant to perform the blood pressure measurement, electrocardiograph reading and blood oxygen content measurement. The device is wireless in nature and consists of transducers placed on patient's wrist. Information is communicated to a base station which performs the post-processing and provides the output.

Blood pressure measurement provides an indication of the health of the heart. It is known that cardiovascular problems are very prominent in Canada and can be linked to obesity and stress. Although blood pressure is a good indication of heart function, electrocardiography signal which is the measurement of electrical activity of the heart, provides an even better insight in functioning of the heart. When these are combined together, the diagnosis could lead to narrowing down the problem and elimination of certain problems completely.

Blood oxygen content refers to the saturation level of haemoglobin in the blood. The number is usually out of one hundred percent and in a healthy individual; it is usually over ninety percent. Again, this data could be combined with ECG and blood pressure to look at heart and lung function at the same time. For example, if the ECG data is normal and the blood pressure is within the normal limit, abnormal blood oxygen saturation could indicate a potential problem with lungs or amount of haemoglobin present in blood. This could further lead the diagnosis to either an inappropriate amount of perfusion in the lungs or conditions pertaining to anaemia.

This report presents the insight from acquiring the signal to post-processing and outputting the final result. Various manipulations are done in the processing of initial signals. These include removal of noise, Fourier transformations, peak detection and removal of floating DC values. The other members in this project consist of Hamzah Qureshi who is responsible for hardware of the blood oxygen measurement system. Ashwin Ayyaswamy is responsible for hardware of ECG and finally, Doralice Ferreira is responsible for the hardware of blood pressure measurement. My contribution is to transport these signals wirelessly and process them to yield some sensible results.

1.2 Objective and Scope

As described earlier, there is a severe shortage of health care professionals, especially doctors and the clinical wait times have been increasing to a level that is out of the comfort zone of a patient. This system, Quick Doc is aimed at reducing both physician work load and clinical wait times. The system incorporates measurement of several physiological signals that aid in basic diagnosis. These signals are described by the inputs from blood pressure measurement, electrocardiography recording and blood oxygen content measurement. These signals are transported wirelessly to a base station where all the post-processing takes place. The post-processing reveals final values that

could be used in the diagnosis of the patient. These three signals could be used in diagnosis individually or in conjunction with one another. This idea would help a physician to narrow down a problem quickly, and without utilizing any time in the actual acquiring of the signal. The acquiring and processing of signals is meant to take place while a patient is waiting in a clinic. Therefore, this device would help save both physician and patient's valuable time. The physical form of the final device is meant to look like a glove that could be easily worn over the hand. Another advantage is that the glove does not have any wires that restrict patient movement in any way.

Implementation of this device in walk-in clinics or busy hospitals would be an ideal situation where all the information from the wearable glove could be sent to the base station. The base station could be the nurse's station, where the data is processed and meaningful results are displayed. This could be utilized by the nurses and doctors to analyze patients that are higher priority than others based on their health condition. It also serves as an initial indication of basic physiological signals for the doctor to begin the diagnosis.

Most of the patients dislike the wires hooked up to their peripherals for continuous diagnosis. This is usually the case in intensive care units. The wires have a psychological stigma associated with them. Of course wires also limit the mobility of the patient to a great extent due to the nature of attachments. This limits the patients to stay in their bed or room and not go anywhere even if they are capable to do so. Implementing the Quick Doc system would help to get rid of the mobility problem as the patients could walk around freely and enjoy themselves. This also limits the frustration associated with wired measurements devices that could be partly responsible for ill health of the patient. Since Quick Doc transports the signals wirelessly, patients would perceive this system differently than a regular wired measurement device. Also, the wireless range of Quick Doc is up to 1000 feet which provides a lot of freedom in terms of patient mobility.

Another area that Quick Doc could have a big impact is physiotherapy. The rehabilitation programs aim to reinforce the physical strength in a patient. Here the

wireless system is extremely important as the whole exercise program is based on the mobility of the patient. Several dangling wires in this case are cumbersome for the patient. Quick Doc on the other hand is a wearable glove that performs similar task. The physiotherapy program would be accelerated to a greater extent with the aid of Quick Doc. Patient comfort is also bound to increase with the replacement of wired diagnostic devices.

Thus Quick Doc aims to reduce clinical wait time and increase patient comfort. This is mainly done by the integration of wireless component to the measurement device and by the small form factor. The small form factor is achieved by assimilating all three transducers near the wrist area. This idea is unique in terms of the way the three physiological signals are integrated and their wireless transmission.

1.3 Methodology

The entire project is sub-divided into four parts. These include the hardware implementation of an electrocardiography device, blood pressure measurement device, blood oxygen content device and the wireless transmission and post-processing of signals from all three hardware devices. The system can be represented by a simplistic diagram as following:

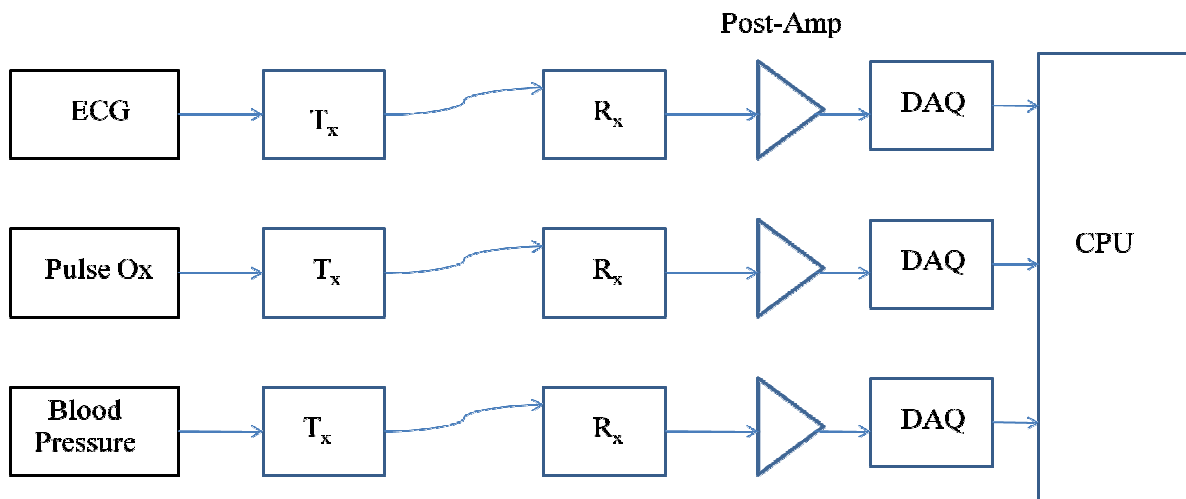


Figure 1.1 Schematic of the Quick Doc system [1]

The physiological signals are recorded by the various transducers and then transmitted wirelessly to a base station. Here after the transportation of signal via a wireless link, a data acquisition system is used to sample the signals in a digital format. The signals are then read into the computer via the Universal Serial Bus port. Since the signals have to be converted from analog to digital domain, sampling frequency and input Voltage range is important criteria while importing the data into the computer. The signal Voltage is limited from -10 V to +10 V.

A precise representation in digital domain would be required for all the physiological signals hence it was a better choice to go with an analog to digital converter that has a high resolution. The analog to digital converter is built into the data acquisition system. The highest sampling frequency from the data acquisition system is around 240 Hz. This frequency, when linked with Nyquist sampling allows us to correctly reconstruct signals with a frequency up to 120 Hz.

It is known that electrocardiography is the signal with the highest frequency components in the three types of signals acquired by Quick Doc. The QRS complex in electrocardiography has the frequency around 40 Hz. This is the highest frequency that needs to be read in from the data acquisition system. So a sampling rate of 80 samples per second is required. Therefore the rate of 240 samples per second is sufficient to cover the frequency range of all physiological signals in question. The frequency spectra of blood oxygen content ranges from 0.1 Hz to around 5 Hz. The blood pressure has a very similar frequency range. Consequently, we have complete range of frequencies covered.

The wireless transmission is done via a digital Radio Frequency based transmitter and receiver. They work at a center frequency of 869 MHz and have a bandwidth of 28000 Hz. The range of this wireless link is up to 1000 feet. An added benefit in using these wireless modules is that they are low power, which aids in the whole scope of the project since the transmitter is based on the glove that is worn on the hand. Special consideration was made to the antennas for these devices. The transmission frequency for

the antennas had to be centered on the same frequency as the wireless link itself. The form factor of the antennas was also an important consideration as the transmitter is located on the glove.

The data acquisition system used has the input range of -10 V to +10 V and has a precision of 10 bits. In order to maximize the use of 10 bit precision, the input to the data acquisition system needs to be as close to the maximum range as possible. During the implementation, the signal range of around 5 V was achieved which is an acceptable level. The analog signal from wireless receiver are sampled at 240 samples per second, converting it into a digital format. This is sent to the computer via a Universal Serial Bus port. The proprietary software that is provided with the data acquisition system reads the signal and displays it on the computer monitor. This signal is the raw signal; no valuable information could be extracted from this graph. The input signal is then saved in the memory.

Computer software called MATLAB is then used to open this input signal and perform processing on it. MATLAB is used to implement various filters such as high pass, low pass, band reject. It is also used to perform the Fourier transform of the signal. Fourier transform takes the signal in frequency domain. Here, various manipulations are performed, which are discussed in detail further in the report. Other techniques such as local peak detection, graphing the output and playing back the electrocardiography after processing so it appears to be in real time only delayed by a certain constant time is also done in MATLAB. Finally combination of above techniques, a few other algorithms and usage of some formulas yield the final results. These results are displayed on the computer screen.

1.4 Conclusion

Quick Doc is meant to assimilate three types of physiological signals into one device. This device is wireless and portable in nature that aids in patient comfort. The integration of three types of physiological signals: blood pressure measurement, blood oxygen content measurement and electrocardiography reading give the physician a scope for basic diagnosis. This results in saving clinical wait time as well as physician's valuable time.

Chapter 2

Literature

2.1 Processing guidelines

The analog signal acquired from the patient is usually converted to digital domain. This is done by sampling the signal at a given frequency. The sampling frequency is a critical decision and is discussed later in this chapter. Once the data is sampled, it could take one of the pre-defined values in the digital domain. The digital data is also easier to work with since it can be imported to computer. However, it is not as intuitive to work with as the analog signal.

Although most of the signals converted from analog to digital domain follow the Nyquist theorem for sampling, the theorem is applicable to highly ideal cases. Nyquist defines a sampling rate of more than twice is needed than the frequency of the input signal. Since ECG is a combination of various peaks, the frequency spectrum is spread over a large frequency range. Applying Nyquist, the sampling frequency should be more than twice of the highest frequency in the ECG signal. In most of the commercial ECG devices, oversampling is used. Oversampling is performed to get a better representation of the signal in digital domain. Oversampling provides some lenience in terms of filter design. If ECG is sampled at Nyquist rate, a highly ideal filter needs to be implemented, which is very expensive in analog domain and computationally intense in digital domain. Thus oversampling solves this problem as the transition bands between band pass and band stop filters have larger bandwidth. A classic case of band pass and band stop in case of ECG would be while trying to isolate the ECG signal from the power noise [8].

ECG processing primarily deals with the filtering of the original signal. As with any signal, ECG signal is contaminated with the power noise. Power noise results from the power supply source, the wall socket is one example of such noise. Since most of the

devices are powered from the wall socket, some noise exists in the instruments. Furthermore, some power noise could be experienced in the form of electromagnetic waves. An easy way to remove power noise is to implement a band stop or a notch filter in the frequency domain [8]. Power noise is based around 60Hz and is removed using a sharp notch filter.

Low pass filtering is very essential for an ECG signal. This is because the activities from muscles have a similar frequency domain and led to contamination of the ECG signal. This can be removed using a low pass filter. The cut-off for a low pass filter can be 40Hz; this preserves all the components of an ECG signal including the sharpest peak represented by the QRS complex and eliminates high frequency noise from muscle activity. In very high precision ECG systems, a cut-off of 100 Hz is used [8]. This is not necessary for a basic ECG application.

High pass filtering is required to get rid of the very slowly varying components in the ECG signal. Low frequency contamination comes in the form of baseline sway in the ECG. Usually the cut-off used is from 0.05 Hz to 0.67 Hz. This solves the problem of varying baseline sway. Another interesting implementation of this technique can be done in the frequency domain. Once the signal is converted to frequency domain, the lower components in the frequency domain, around 0Hz could be removed [8]. This would remove the DC off set and the swaying baseline and provide a perfectly flat ECG signal.

Techniques, such as local peak detection could be used to find the local peaks. This would enable tracking a peak per cycle of ECG. Usually the easiest peak to track is QRS since it is the highest peak in the standard ECG signal. If this peak is detected every cycle of the ECG, the number of peaks could be counted over a given time interval to provide the heart rate of the patient.

Chapter 3

Statement of Problem and Methodology of Solution

3.1 Statement of Problem

As stated in chapter 1, the Canadian Family Physician organization performed a poll in 2006 that shows around 17% of Canadians do not have family physicians. This amounts to around 5 million Canadian residents. It was also found that around 2 million of these 5 million cannot find a doctor [7]. These findings imply that more health care professionals are required. This requires a huge initial investment in the facilities to accommodate for the training and experience of these health care professionals. The investment in facilities such as hospital is happening slowly, other approaches to cut waiting could be explored in the meantime.

Improvements in existing machinery and technology could help cut the wait times. It has been seen in the past that developments in this field have led to a drastic reduction in wait time as well as increase in patient comfort. It could be inferred that patient comfort is somewhat related to reduced wait time. Therefore improving technology leads to both. In certain health care professions such as optometry such improvements have already taken place. A scan on the auto-refractor machine is performed as the patient is waiting to see the optometrist. This machine prints out an approximation of the eye lens weakness, myopic or hypermetropic condition as well as the astigmatism angle. This gives a general idea to the optometrist where to begin with the eye exam. This saves the actual time spend in the eye exam room and utilizes the patient wait time while saving optometrist's time as well.

The idea of Quick Doc sparked from this management system implemented in optometry clinics. Most common diagnosis performed by a physician as a patient walks into the clinic as blood pressure, health of heart and temperature. Quick Doc is meant to do perform the blood pressure measurement, electrocardiograph reading and blood oxygen content measurement. The device is wireless in nature and consists of transducers placed on patient's wrist. The information is communicated to a base station which performs the post-processing and provides the output. This information could be used by the physicians for diagnosis.

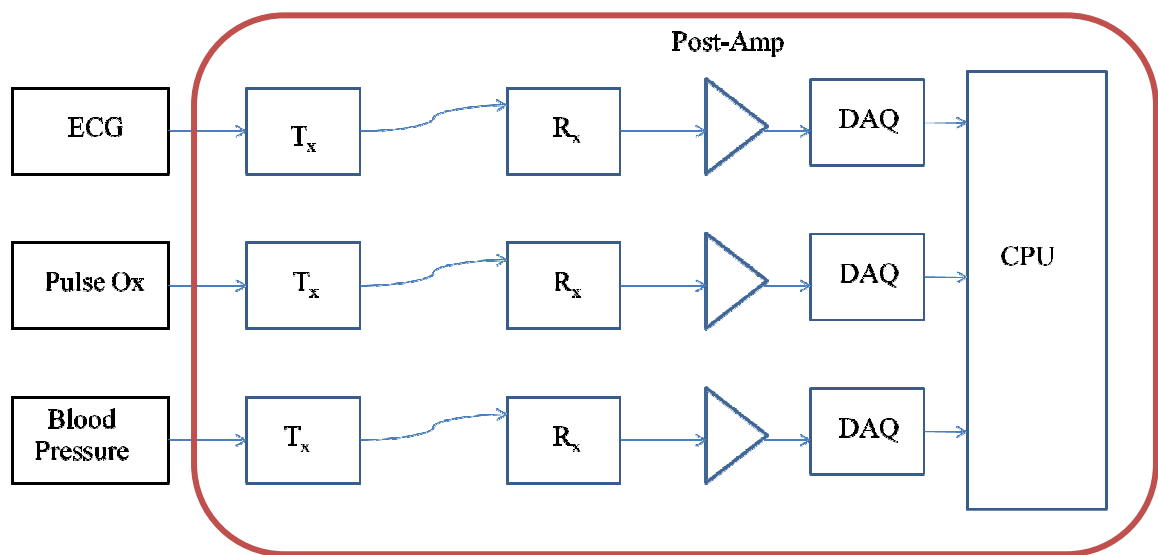


Figure 3.1 Basic block layout for Quick Doc, the area outlined represents the personal contribution towards the project [1]

A wireless transmitter and receiver are required to transmit the physiological signals wirelessly. Although this is an integral part of the project, the bigger problem is to read the signals from the transducer and process them using the software MATLAB. The criteria and algorithms for other modules will be discussed prior to the wireless components which are added in the end.

The physiological signals need to read into the computer. This requires converting from analog to digital domain. The two biggest issues in this procedure are the Voltage

levels of input signals coming from the transducers and the sampling rate to convert from analog to digital domain.

The output Voltage levels from ECG transducer are around 2 to 5 mV. This is quantitatively a very small signal. It needs to be passed through relatively few stages of amplifiers and filters before it could be sampled into the computer. The level of signal when sampling to convert to digital domain should be in range of Volts. The output from blood oxygen content measurement apparatus is around 2 to 5 mV. Again this needs to amplify and filters before it could be sampled into the computer. The blood pressure measurement device has an output of around 2-5 V. Therefore, no extra amplification is needed for blood pressure signal; it could be sampled into the computer as such. This limit of a few Volts arises from the hardware limitations of the data acquisition system being used. The input range is from -10 V to +10 V. In order to maximize the use of 10 bit resolution, the input signal needs to be as close to the input range as possible.

The other issue is regarding the sampling frequency. It is always easier to use the maximum sampling frequency as possible or limited by the hardware however that would be wasteful in terms of the bandwidth being used. Among the three physiological signals being acquired it is known that ECG has the highest frequency components. The highest frequency in ECG is around 35 to 40 Hz and represents the QRS complex. As long as this is sampled properly, the remaining frequency spectrum should be sampled properly as well. To sample 40Hz in a proper way, following Nyquist rate, around 80 samples per second are required. This would theoretically cover the entire frequency domain for all three physiological signals. However, there is always an issue of added noise. The signals might need to be sampled faster than this, as the processing to remove noise and such could also lead to deterioration of the signal quality. Consequently, the signals should be sampled at some quantity that is higher than 80 samples per second. The actual data acquisition system used is discussed in detail in the hardware implementation chapter of this report.

After the signals are imported into the computer, the post-processing is done. A lot of noise associated with the signals could be anticipated. Thus the post-processing would be heavily based on filtering techniques. Some of the other possible ideas such as detecting local peaks, detecting highest peak would also be required. A notch filter to remove power noise based at 60 Hz is required. Peak detection is required to count the number of peaks in the ECG signal to find the heart rate of the patient. It is also required to find peaks in the blood pressure transducer output signal. Due to the nature of this signal, the local and highest peaks need to be detected to find the blood pressure of the patient. Local peak detection is also required in the blood oxygen content measurement signal. The amplitude of peaks determines the blood oxygen saturation levels. All these and many more techniques have been discussed in much more detail in further chapters of this report, in the Software Implementation chapter.

The biggest constraint on the wireless module is that it has to be of a small form factor and low power requirement. The module also has to have an analog input and analog output. This greatly reduces the problem definition as an analog output signal is obtained from the receiver, which is fed to the data acquisition system, where it is sampled and converted into frequency domain. The simplest form of wireless communication is using some sort of radio frequency module that works on the principle of modulation. The modulation could be analog or frequency, although frequency modulation is preferred due to its superior nature. The final module chosen is discussed in detail under the chapter Hardware Implementation of this report.

3.2 Theoretical Background

3.2.1 Electrocardiography signal

The electrocardiography is the measurement of electrical activity of the heart. The waveform shown below represents a perfect ECG signal. The heart is divided into atria and ventricles. Atria store blood temporarily and then transfer it into the ventricles via the valves. The ventricles then pump the blood to the entire body.

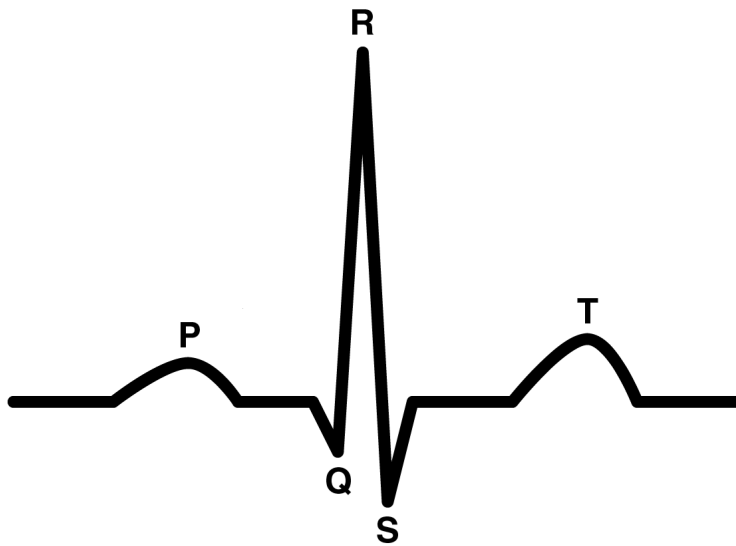


Figure 3.2 Schematic representation of a normal ECG [3]

The P wave represents atria depolarization. This leads to pumping of blood from atria to the ventricles. After that the QRS complex represents the ventricular depolarization. As seen above this is the biggest spike in the ECG signal. The ventricular depolarization leads to the pumping of blood to the aorta and pulmonary vein which leads to supplying blood to the entire body. The T wave represents the ventricular repolarisation. This is when the ventricle is restored to its normal relaxed form. It should be noted that atria repolarization is masked by ventricular depolarization. The magnitude of ventricular depolarization is much bigger than atria repolarization.

3.2.2 Blood Pressure

The blood pressure associated with this project is the arterial blood pressure measured at the brachial artery. The brachial artery is the artery on the medial side of the arms. Blood pressure normally consists of two parts, systolic and diastolic blood pressure. The systolic blood pressure represents the pressure in the artery after the ventricular contraction. This is usually the higher number and it is rated for a normal value of 120 mm Hg. Diastolic blood pressure represents the pressure when ventricles are filling with blood from the atria. The pressure in systemic arteries is at a minimum level at this time. Consequently, diastolic represents the lower limit of the blood pressure and is usually rated at 80 mm Hg for a normal value.

Hypertension is a condition when the blood pressure is over its normal limit. It is defined as a systolic blood pressure of over 140 mm Hg and diastolic blood pressure of over 90 mm Hg. The leading cause of hypertension is obesity, which leads to arteriosclerosis. The hardened arteries have far less compliance than normal arteries, which leads to higher blood pressure. On the other hand hypotension refers to a blood pressure of less than 100 mm Hg and 60 mm Hg for systolic and diastolic respectively. Exact cause for hypotension is unknown.

3.2.3 Blood Oxygen content

Blood oxygen content represents the saturation of haemoglobin in blood with oxygen. The normal level is over 90 percent. This level reflects on the proper functioning of the lungs and heart.

3.2.4 Wireless module

The radio frequency wireless module could be based on amplitude modulation of frequency modulation although frequency modulation is preferred. In digital frequency modulation, frequency-shift keying is often used. This enables the carrier frequency to change between a few pre-defined discrete carrier frequencies. This digital approach is much more reliable and immune to noise and provides a better signal to noise ratio. Usually the carrier frequency is centered to a much higher value than the input frequency range of the device. Transmission, via high frequency is more reliable than using low frequency, although the processing of the signal is better done at low frequency. Thus in certain devices, an intermediate frequency is used in the receiver module which is much lower than the carrier frequency. This frequency is used for a better result from the filters and post-processing apparatus, which are known to work more efficiently at a lower frequency than the carrier frequency.

In certain receiver modules, surface acoustic wave filters are also implemented. These convert the electromagnetic wave to a mechanical wave. As the mechanical wave travels through the medium it is delayed by some amount. The delayed waves are recombined to form an implementation of a finite impulse response filter. These types of filters are limited to a frequency range of a few GHz.

3.2.5 Data Acquisition System

The data acquisition system is required to have analog input as the wireless module has an analog output. As the highest frequency in the ECG spectrum is around 35-40 Hz, the sampling frequency should be at least around 80 Hz. The bit resolution is another issue that needs to be addressed. Higher bit resolution ensures better reconstruction of the sampled data. Usually a bit resolution of 8 bits is preferred. The data acquisition system should be able to supply the data to the computer via a USB or some other peripheral port. This data would then be imported by MATLAB to perform necessary calculations and post-processing.

3.2.6 Post-processing

The post-processing will rely heavily on filtering. It is known the physiological signals have very low amplitude and could be contaminated by various sources of noise. For example the ECG signal is expected to have amplitude of 2-5 mV. This is extremely low and could be contaminated by electrical signals from the muscle activity near the ECG transducers. Also the blood oxygen content measurement is also expected to have amplitude of about 5 mV. This is further contaminated by muscle activity. Other sources of noise include the instrument itself. Also the power noise plays a big role in signal contamination. Power noise is centered around 60 Hz so it does not affect the physiological signals as adversely because the highest frequency component of physiological signal is around 40 Hz. Apart from this, a few algorithms could be implemented in frequency domain. This would require the Fourier transform of the input data. Detection of peaks could be another important algorithm that is required for all three physiological signals. Graphing the output, especially ECG to show it running in real time only delayed by a fixed time would require a variety of small algorithms implemented in MATLAB.

However for the most part, it would depend on the nature of input data what type of algorithm to apply. Consequently, it will require a lot of trial and error approaches. The final design criterion that works for all three types of physiological signals is discussed in software implementation further in this report.

Chapter 4

Hardware Design

4.1 Wireless link

The wireless link used is manufactured by Linx technologies. It is known as the ES series and is based on Frequency modulation and Frequency-Shift Keying technique. In Frequency-Shift Keying the data is transferred via frequency modulation in a way that carrier frequency changes between certain set discrete values.



Figure 4.1 Picture of receiver and transmitter module [4]

4.1.1 Transmitter

The transmitter is a 10 pin device that is capable of both analog and digital input for the data to be transmitted. The analog input is utilized for the purpose of this project. The transmission frequency is centered around 869 Mega Hz. The bandwidth is about 28000 Hz.

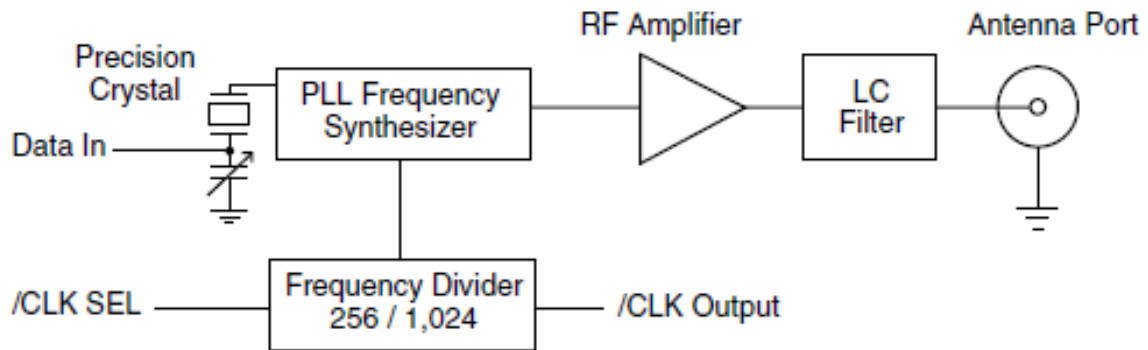


Figure 4.2 Simple schematic of the wireless transmitter used in Quick Doc [4]

The Data In refers to the input from the external source. This input is limited from 0 to 5 Volts. The clock select is responsible for the frequency divide that could be chosen between 256 and 1024. This is fed to the PLL frequency synthesizer, where the input data is modulated with the frequency from precision crystal and controlled by frequency divider. The frequency divider ensures the variation of transmission frequency between various discrete values. This is the principle of frequency-shift keying. The output from PLL frequency synthesizer is fed to a RF amplifier. This amplifier amplifies the signal, which is passed through a filter and then to an antennae. The filter is necessary to remove any noise present in the signal. The antenna impedance is around 50 ohms and this is critically matched to avoid any bouncing back of the signal from antenna to the filter. The Voltage requirement of this device is 4 Volts and current requirement is 5.5 mA to about 8.5 mA.

4.1.2 Receiver

The receiver operates at a frequency of 869 MegaHz with a bandwidth of 28000 Hz. The receiver is a 16 pin device and is capable of both analog and digital output. In this case the analog output is utilized.

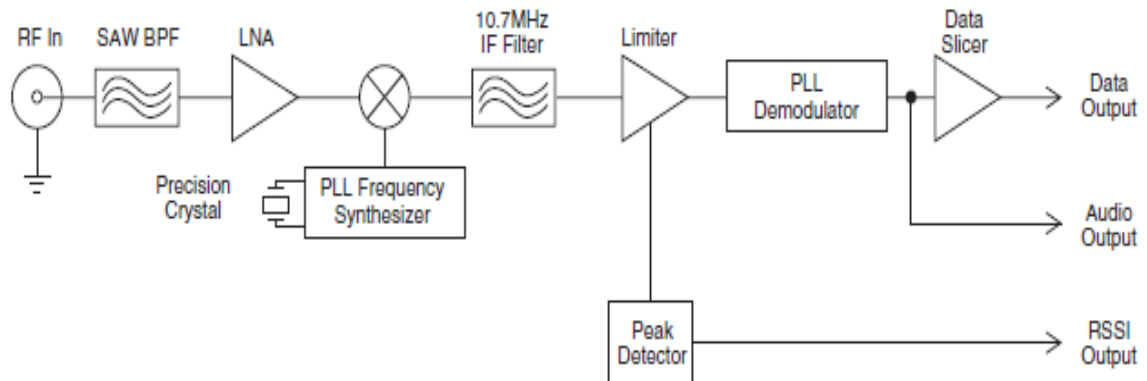


Figure 4.3 Simple schematic of wireless receiver used in Quick Doc [4]

The Radio Frequency input is from the antenna that also operates at 869 MegaHz. This is fed into a SAW band pass filter. The Surface Acoustic Wave filter works by converting the electromechanical wave into a mechanical wave and is then converted back to the electrical equivalent. The delayed mechanical waves are combined to form a finite impulse response filter. This is then combined with the output from a precision crystal. This is what leads to the demodulation of the input signal. An intermediate frequency of 10.7 MHz is used to perform further filtering. The intermediate frequency is chosen such that it is easier to work with than the high input frequency. A peak detection block follows the intermediate frequency filter. A demodulator further demodulates the signal and it is sent to the audio output. This audio output represents the analog output. The analog output is the final signal of interest for the scope of this project. The output signal is around 55 millV. This is further amplified before input into the data acquisition system. The Voltage requirement for this device is 5 Volts and current is limited from 5.5 to 6.5 milliamps.

4.1.3 Antenna

The antenna used is the JJB series from linx technologies. It is a monopole antenna with a very compact package. The maximum width of the antenna is 7 mm. the input impedance is 50 ohms which is perfectly matched with output from transmitter and input to receiver module of the wireless link. The center frequency is 868Mhz which is very close to 869 of the wireless modules. The bandwidth is 25000Hz. This is convenient as the bandwidth of the wireless modules have similar bound.

This antenna is especially convenient because of its small size. The size is crucial for the overall small form factor of the transducer part of Quick Doc.

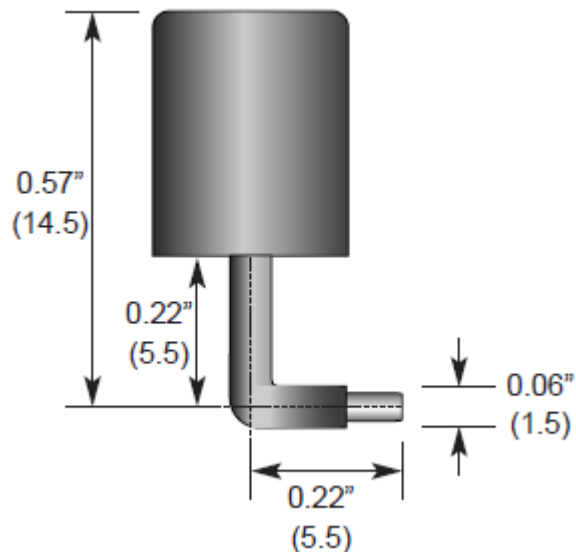


Figure 4.4 Antenna dimensions [4]

4.2 Data Acquisition System

The data acquisition system is produced by Dataq instruments. The model number is DI-148U. It enables USB acquisition of analog data. The device houses eight analog inputs that could range from -10 V to +10 V. It also has certain digital outputs however, those were not used in this project. Although the device is capable of sampling up to 14,400 samples per second, a sampling rate of 240 samples per second is used for this project. This covers all the necessary frequencies from the three physiological input signals.



Figure 4.5 Picture of the data acquisition system used in Quick Doc [5]

The inputs to the data acquisition system are sampled with a 10 bit resolution. This also requires the analog input to be as close to maximum input of -10 V to +10 V as possible to utilize the maximum bit precision. As discussed prior, the output from receiver of the wireless module is around 55 millV. This is amplified by 56 times and the final input to the data acquisition system is around 3.1 V. This is convenient and the input is read properly by the device.

Once the data is read, it appears on the computer screen as a continuously updating graph. The proprietary graphing software is provided by Dataq communications and comes with the purchase of the device. The data displayed is the raw data and is not processed in any manner. As such it includes a lot of noise. This data is then saved on the memory of the computer. The format data is saved in is also of the format set by Dataq instruments. This data is later imported into MATLAB for necessary post-processing.

Chapter 5

Software Design Implementation

5.1 Introduction

The software used to implement all of the post-processing is MATLAB. Once the data for each physiological signal is imported, a set algorithm is followed to get valid result from the raw data. These algorithms will be described in the following sections along with the explanation of each block in the algorithm.

5.2 Electrocardiography Algorithm

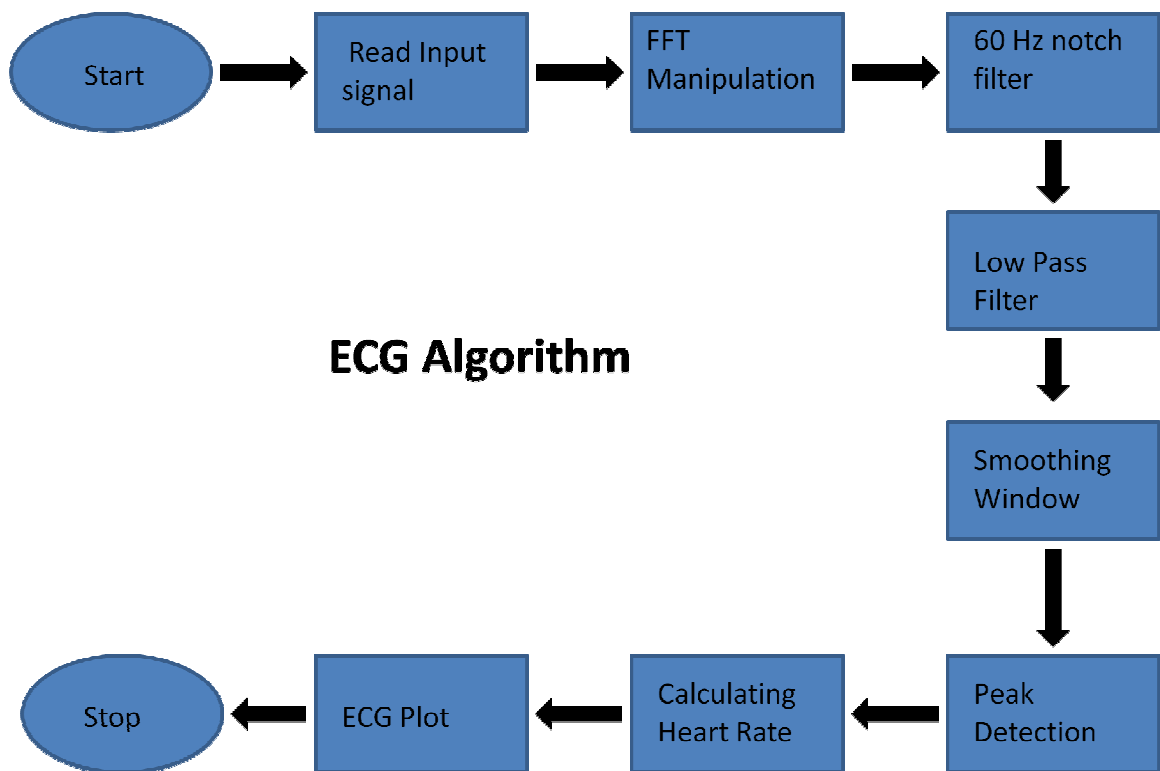


Figure 5.1 A flow chart representing the algorithm for ECG processing

In the following section, the algorithm is followed step by step along with the output at each step. The output at each step serves as a test point and provides an indication whether the whole algorithm is heading in the right direction or not. As previously described, the signal is acquired by the data acquisition system and then imported into the computer via a USB. The data is stored and then opened in the software MATLAB. Once the data is read into MATLAB, the data looks like the following.

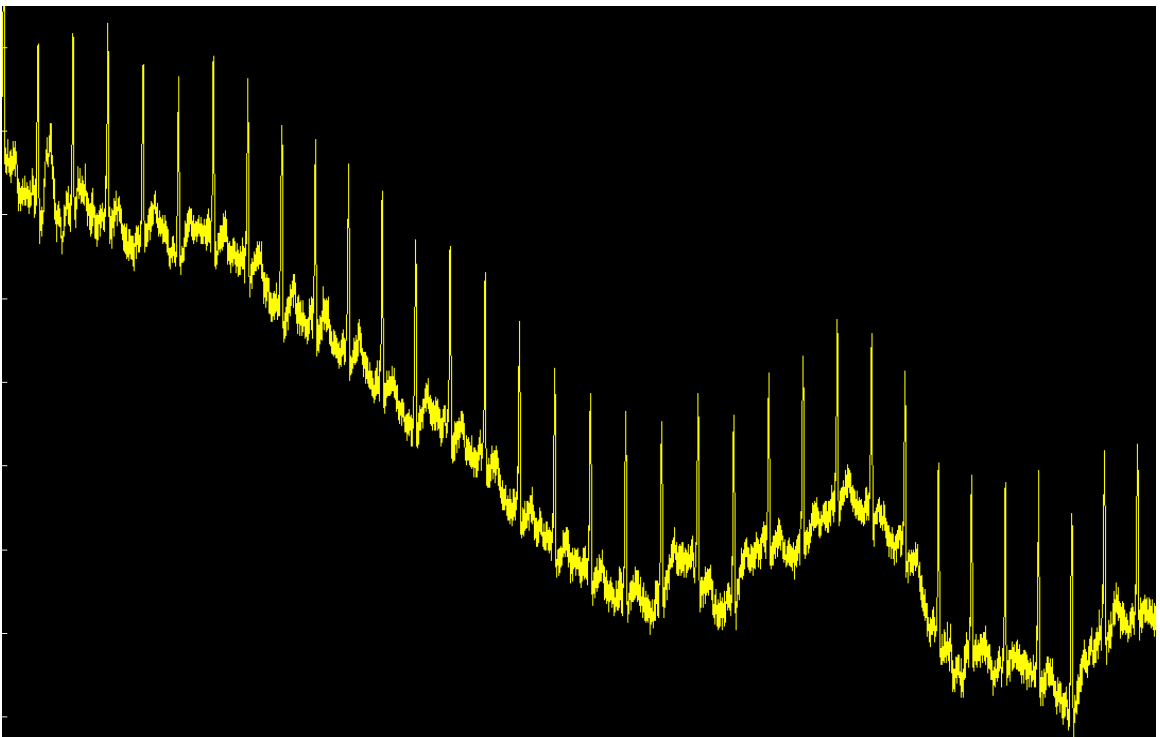


Figure 5.2 Raw ECG data

This data is directly read from the transducer placed on the wrist. The data goes from transducer to several filters, amplifiers, wireless link, post-wireless amplifier, the data acquisition system and finally read into MATLAB. As seen from the figure, the data does not completely look like the ideal ECG signal as described before. This data needs certain post-processing.

As seen from the picture of raw data, a floating DC value is observed. This is a consequence of the floating ground for the transducer. This poses a problem as the commercial ECG machines seen on market produce a perfectly straight line ECG output. This problem is solved by performing some fast Fourier transformation manipulation. The data is converted into frequency domain by taking the Fourier transform of the input data. Once this is done, the extremely low frequency components based around 0Hz are removed. This gets rid of the DC as well as the floating DC offset. As observed about the floating DC is very low frequency and is therefore removed once the frequency around 0 Hz is removed. Once the data is converted back to time domain following output is obtained. As it is clear from the figure, there is no DC offset or floating DC. Now this data could be utilized for further post-processing.

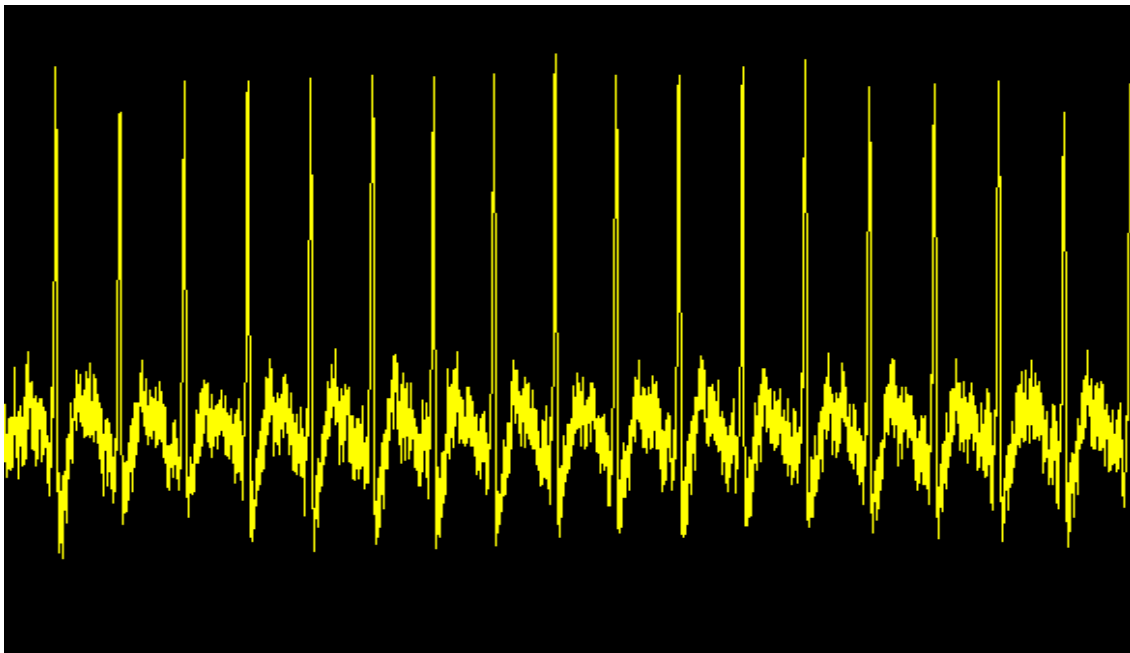


Figure 5.3 Raw ECG data after FFT manipulation

The above figure clearly has some issues with the noise associated with the input. Power noise based around 60 Hz is the biggest culprit in contamination by noise. This could be removed by implementing a band stop filter or a notch filter at 60 Hz. For this purpose a notch filter of the fifth order is used. This leads to removal of 60 Hz noise and

gives us the following result. This figure compared to the previous one reveals a much lesser noise level, particularly noted between the peaks that represent the QRS complex in the ECG waveform.

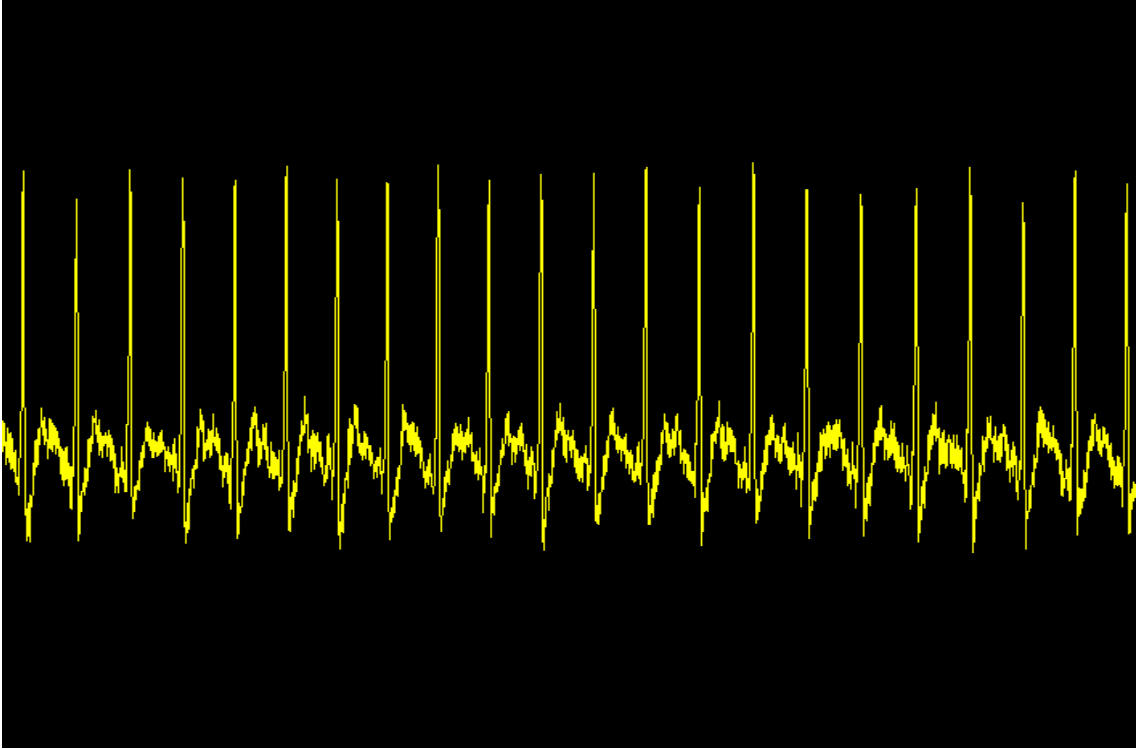


Figure 5.4 ECG data after removal of power noise

Even though the noise has been removed, the P and T wavelets are not clearly visible. The QRS wavelet is seen clearly as the big peak that repeats every so often. As discussed earlier, the highest peak in ECG is around 34 Hz. The above data has frequency component of up to 50 Hz after the implementation of a notch filter. This data is further passed through a low pass filter based at 35 Hz of third order is used.

This yields a significantly improved result and the P and T wavelets could be seen much clearer than the prior figure. This is displayed in the following figure.

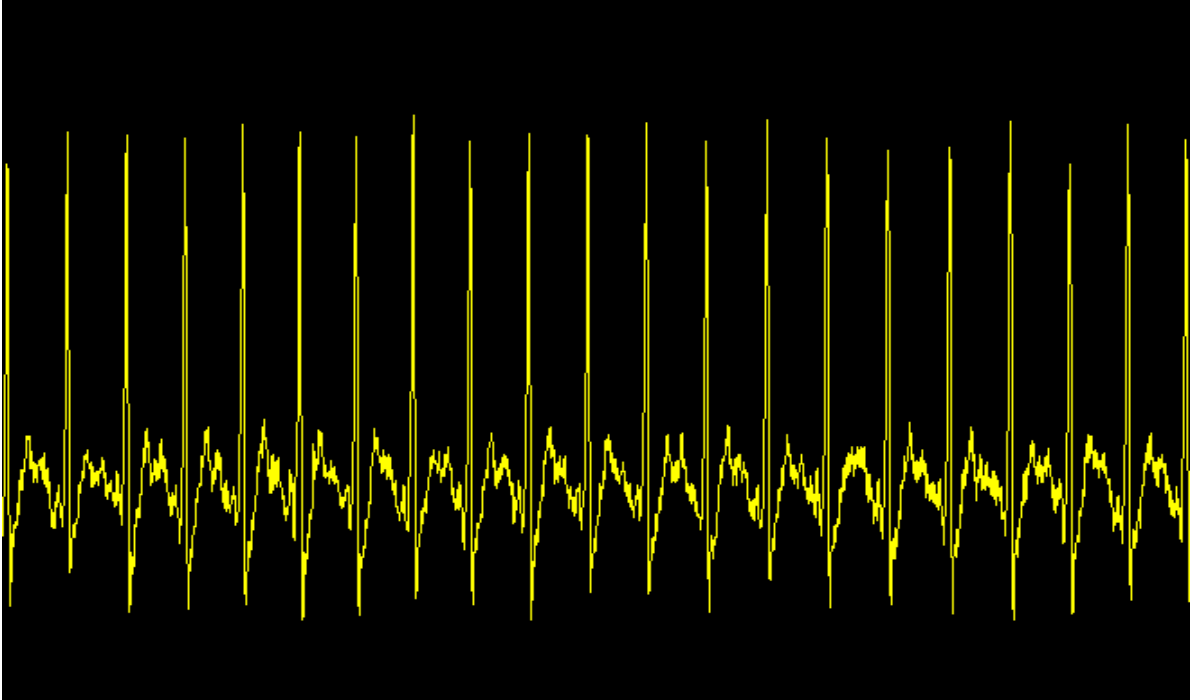


Figure 5.5 ECG after passing through a low pass filter

Although above data does look somewhat like an ECG signal, it is still not comparable to the output produced by the commercial ECG machines. As seen in the above figure, the QRS complex is fairly well represented but the P and T wavelets are not. Although the plot follows the general outline, it does not represent them accurately. One solution to the above is the implementation of a smoothing window. The way this works is that each sample after the smoothing window is an average of a few samples around it. This further serves as a low pass filter. A crucial step is choosing the number of samples to be included in the smoothing window. As it is known already the highest frequency in ECG is around 34 Hz and the sampling rate for the device is 240 samples per second. Therefore, 6 samples would represent the highest frequency. As long as the smoothing window encompasses less than or equal to that number of samples, the output should not have the higher frequency components suppressed. Output after a smoothing window is shown in the following figure.

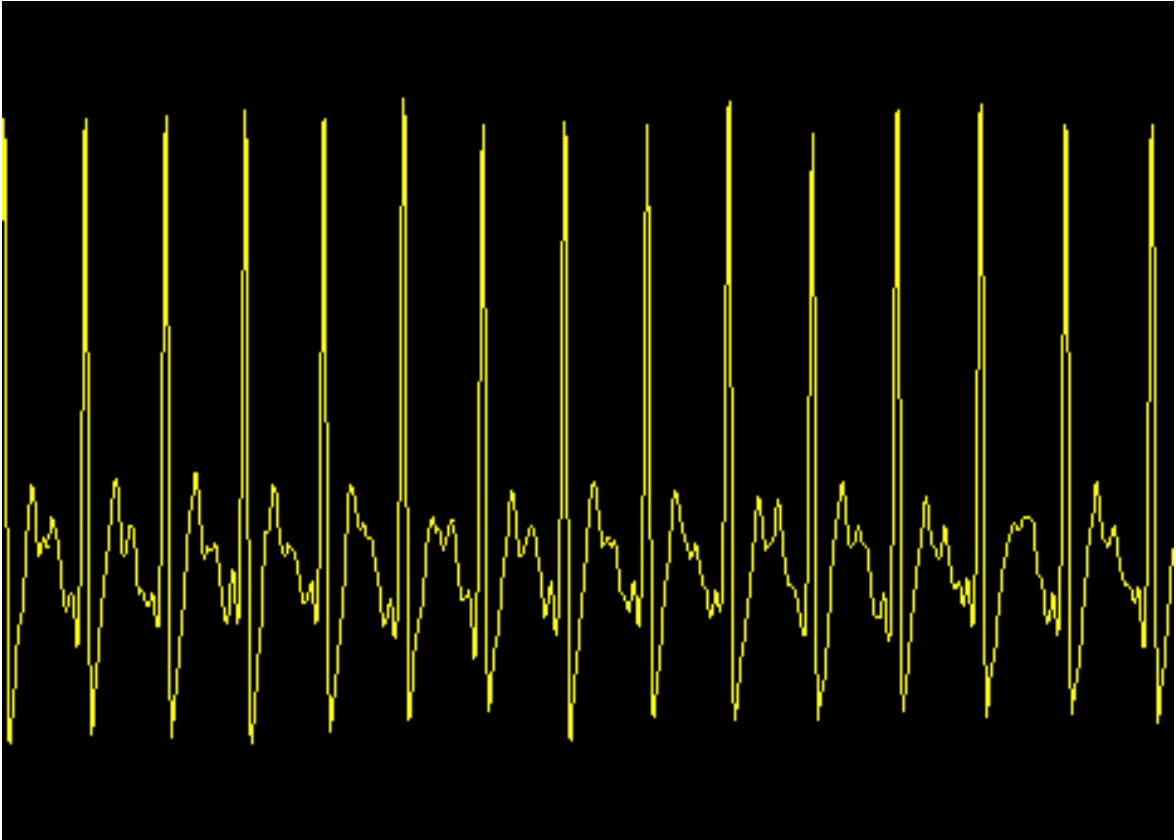


Figure 5.6 ECG data after a smoothing window.

As seen above, distinct P and T wavelets could be identified. The QRS peak is seen as it was in the previous figure. Consequently, implementation of smoothing window did not disrupt the higher frequency component, the QRS peak. This identification of P, QRS and T is important as this represents the complete ideal ECG waveform. The shape of the waveform is represented accurately. Peak detection is still required to count the number of peaks for heart rate detection. This is done by identifying variations among local maxima and minima. A threshold is specified and if the difference between a local minima and maxima exceeds that threshold, a local peak is detected. This algorithm works very well as long the data being analyzed is not contaminated by noise. As seen from the above figure, the data does not look noisy and is ideal for local peak detection. The following figure demonstrates the peak detection.

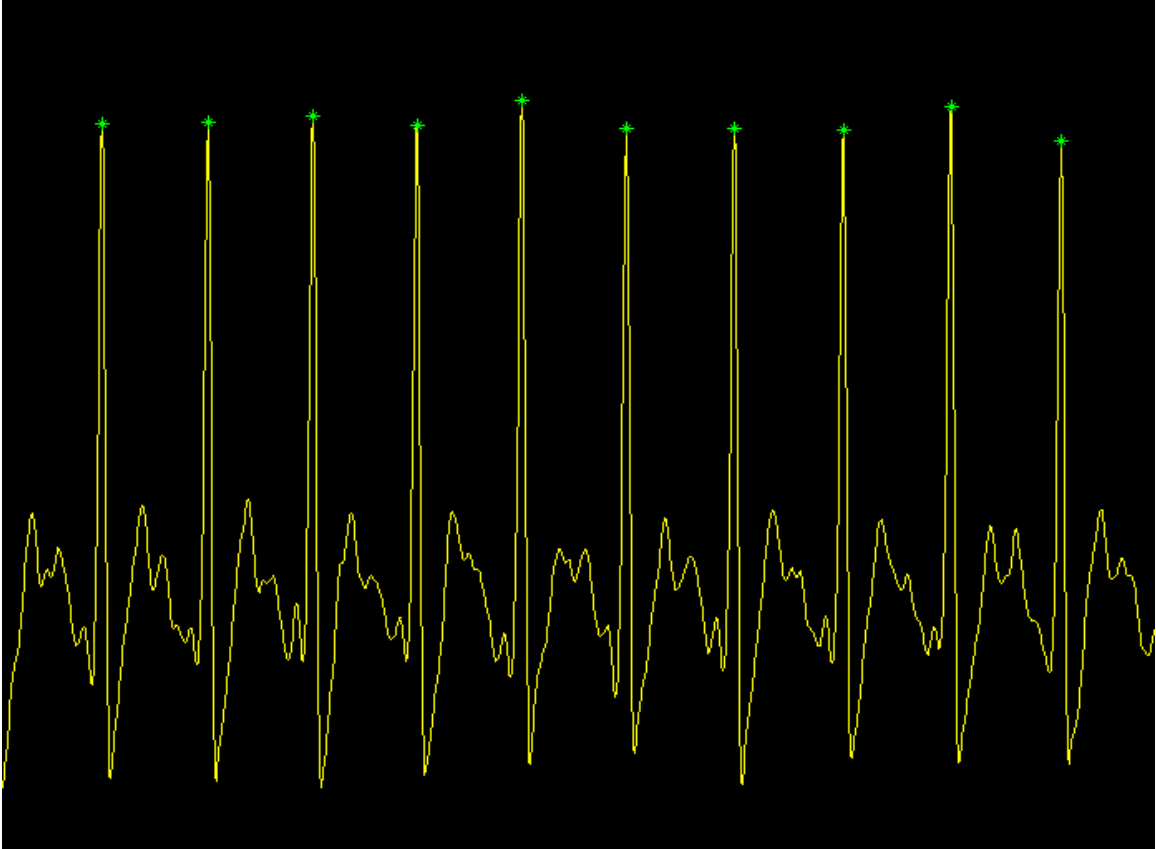


Figure 5.7 The green stars represent the local peaks in ECG signal

The detection of peaks provides the heart rate of the patient. Heart rate is a good indication of health of the patient. This number could be used to detect bradycardia or tachycardia.

Type of Filter	Cut-off frequency	Order
Notch filter	Around 60Hz	5
Low pass filter	35Hz	3
High pass filter(using Fourier manipulation)	Very low, around 0 Hz to 0.05Hz	1

Table 5.1 Table describing filter characteristic used in ECG algorithm

5.3 Blood Pressure Algorithm

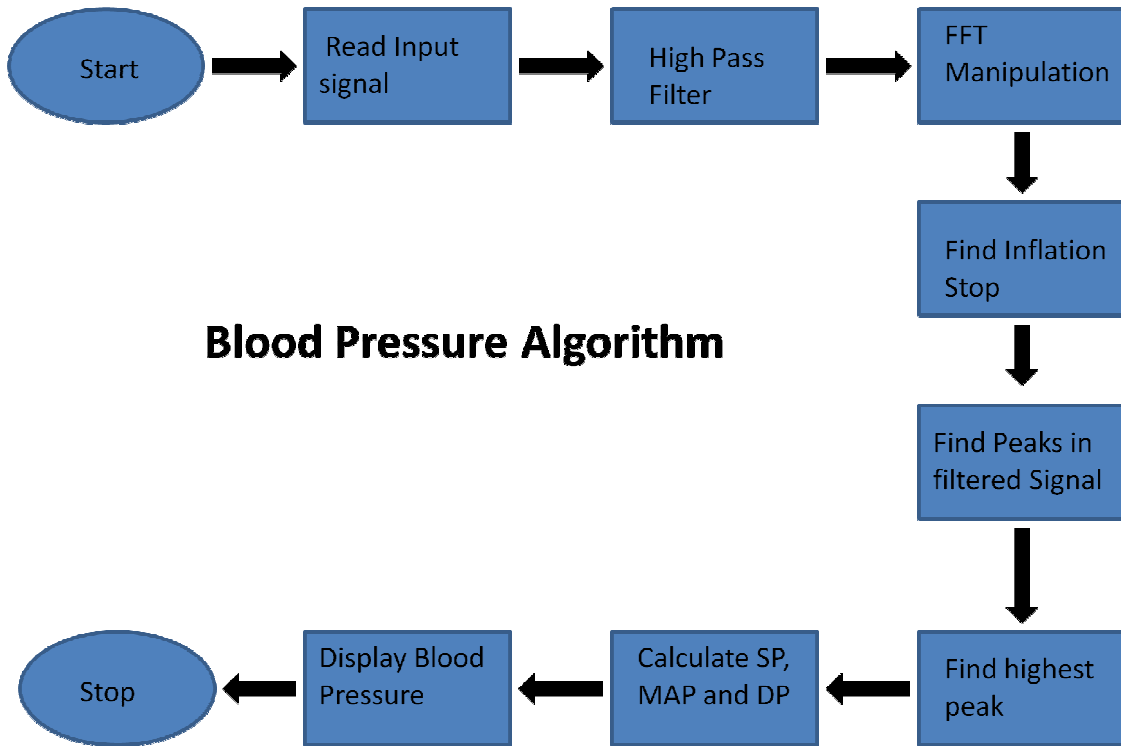


Figure 5.8 A flow chart representing the algorithm for Blood pressure measurement processing

In the following section, the algorithm is followed step by step along with the output at each step. The output at each step serves as a test point and provides an indication whether the whole algorithm is heading in the right direction or not. As previously described the signal is acquired by the data acquisition system and then imported into the computer via a USB. The data is stored and then opened in the software MATLAB. Once the data is read into MATLAB, the data looks like the following.

The figure below shows the input signal from the pressure transducer used for blood pressure measurement. As seen by the shape of graph, it goes up to a maximum value first and then decays slowly to an almost flat line. This is the Voltage representation of the pressure in the cuff worn by the patient. The relationship is linear in nature. Initial part of the graph is seen to rise up representing increasing pressure in the cuff. This represents the pumping of air in to the cuff. The idea is to pump enough pressure in the cuff so that blood stops flowing in the arterial artery. As the pressure is released in the cuff, the blood starts flowing again the pressure when it flows again provides an indication of systolic blood pressure. The diastolic pressure is a function of mean arterial blood pressure.

The peaks seen in the rising part of graph are the fast pressure changes in the cuff whenever the air is pumped into the cuff. After the inflation stop, the pressure is slowly released in the cuff. The following part of graph shows this decay in pressure which is accompanied by little oscillations about the almost linear pressure decay. These little oscillations are a result of blood flow in the artery that causes a minute pressure change in the cuff and are therefore are of extreme importance for the purpose of measuring blood pressure.

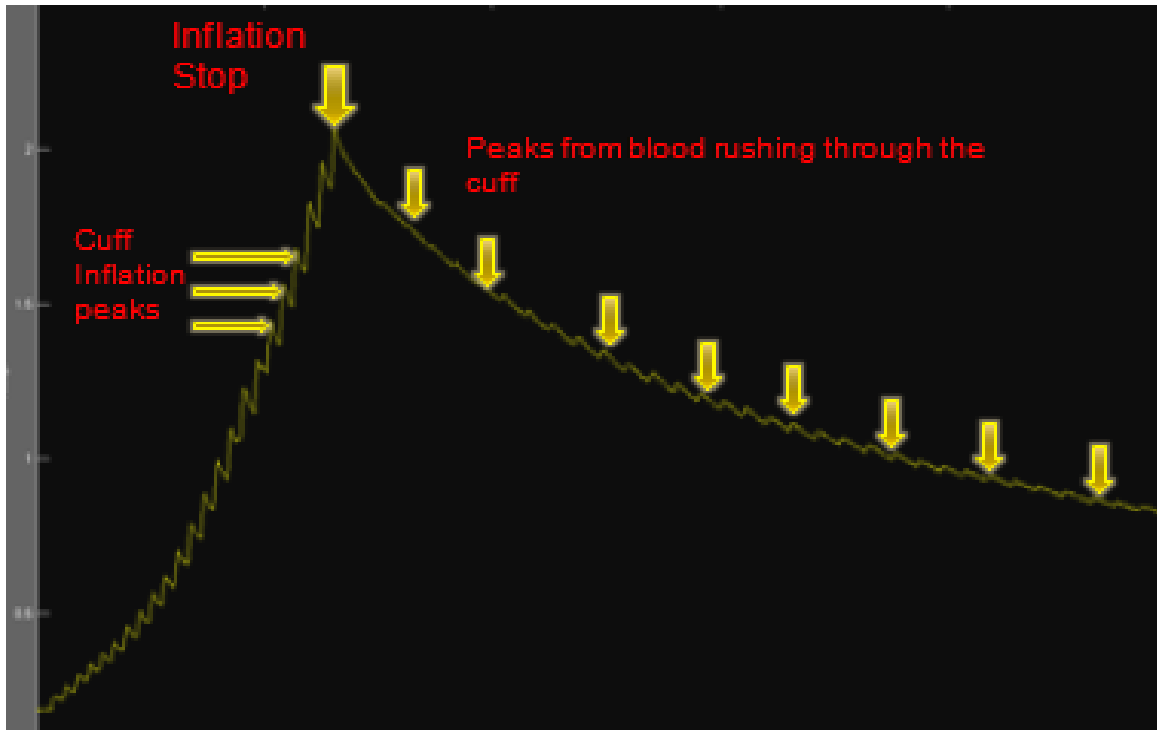


Figure 5.9 RAW blood pressure input data

The objective is to isolate the oscillations after the inflation stop that represents pressure change in the cuff due to blood flow. Once isolated, the amplitude of each peak needs to be measured. This almost seems impossible when looking at the input data. An alternative method used here is to isolate the peaks, amplify them and then map the data obtained after filtering and amplification on to the original input signal. This is essential as the original input represents the true mapping of pressure to Voltage and any filtering or amplification would result in disruption of that relationship.

The peaks are isolated by high pass filtering the input signal. Once high pass filtered, it is amplified to an appropriate amount. The high pass filter has a cut-off of 4 Hz and is of the third order. Once passed through the high pass filter, the peaks are amplified by a factor of 10.

Fourier transform manipulation is performed to remove the very low frequency components. As seen from the picture of raw data, a slowly varying graph that goes up and down is observed. This slowly varying change is not required in the high peak

detection. This problem is solved by performing some fast Fourier transformation manipulation. The data is converted into frequency domain by taking the Fourier transform of the input data. Once this is done, the extremely low frequency components based around 0Hz are removed. DC, as well as the slowly changing graph represented by very low frequency is therefore discarded once the frequency around 0 Hz is removed. Once the data is converted back to time domain the output shown in yellow in the following graph is obtained. As it is clear from the figure, there is no DC offset or slowly varying low frequency component. Now this data could be utilized for further post-processing.

Once the amplified peaks are obtained, the inflation stop point detection is needed. This is crucial to isolate the peaks after the inflation is stopped. The inflation stop point is detected by scanning the original data and looking for the highest point in the data array. Once this is done, a local peak detection algorithm is run on the amplified peak data array. This is done by identifying variations among local maxima and minima. A threshold is specified and if the difference between a local minima and maxima exceeds that threshold, a local peak is detected. This algorithm works very well as long the data being analyzed is not contaminated by noise. The detected peaks are represented by red stars on the yellow graph.

After the detection of inflation stop and local peaks, the highest peak needs to be detected after the inflation stop to find the blood pressure associated with the patient. This is done by looking for the highest peak in the amplified peaks data starting from inflation stop till the end of the array.

Once the highest peak is found from the amplified peak array, its index is mapped back to the original data. This provides the original Voltage reading, which could directly converted to the pressure in the cuff. In this particular algorithm, instead of using just the highest peak in the original data, two peaks around the highest peak are also used. An average of the highest peak and the two around it is computed and utilized to find the

average Voltage level and hence the average pressure in the cuff. The mean arterial pressure is thus give by the formula:

$$\text{Mean arterial pressure} = ((10*\text{average_peak})+0.2) / 0.133 \text{ [9]}$$

average_peak is the average of the highest peak and the two around it.

In a similar fashion, systolic pressure is the function of third and fourth peak after the inflation stop point. The index of third and fourth peak are found in the amplified peaks array and mapped back to the original data. Again average of the two peaks is taken. Thus systolic pressure is described by the following formula:

$$\text{Systolic Pressure} = ((10*\text{syspeaksavg})+0.2)/ 0.133 \text{ [9]}$$

syspeaksavg is the average of third and fourth peak after inflation stop point.

Finally, the diastolic pressure is given by the following formula, which is a function of mean arterial pressure and systolic pressure:

$$\text{Diastolic pressure} = (3/2)*(\text{Mean arterial pressure} - (1/3)*\text{Systolic pressure}) \text{ [9]}$$

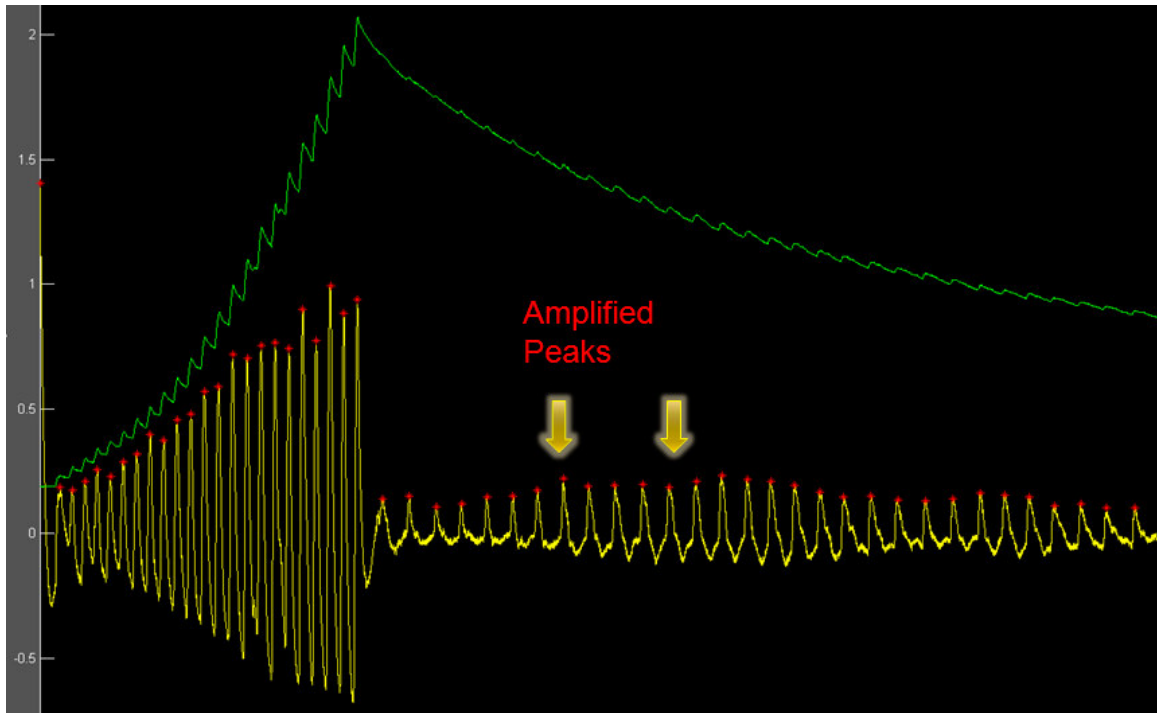


Figure 5.10 Graph illustrating the original blood pressure signal and amplified peaks signal

Type of Filter	Cut-off frequency	Order
Notch filter	Around 60Hz	5
High pass filter	4 Hz	3
High pass filter(using Fourier manipulation)	Very low, around 0 Hz to 0.05Hz	1

Table 5.2 Table describing filter characteristic used in blood pressure algorithm

Following are the results from above algorithm.

Systolic_Pressure = 131.1291

Mean_Arterial_Pressure = 95.0893

Diastolic_Pressure = 77.0694

This is confirmed by the following picture that was taken just a few minutes after the measurement of blood pressure using the transducer and above mentioned algorithm. The picture shows a commercial blood pressure measurement device made by Omron that is a reputable company. The discrepancy is very little as the systolic and diastolic blood pressures are both very similar to the numbers shown above.



Figure 5.11 Picture of commercial device with blood pressure reading

5.4 Blood Oxygen Saturation Algorithm

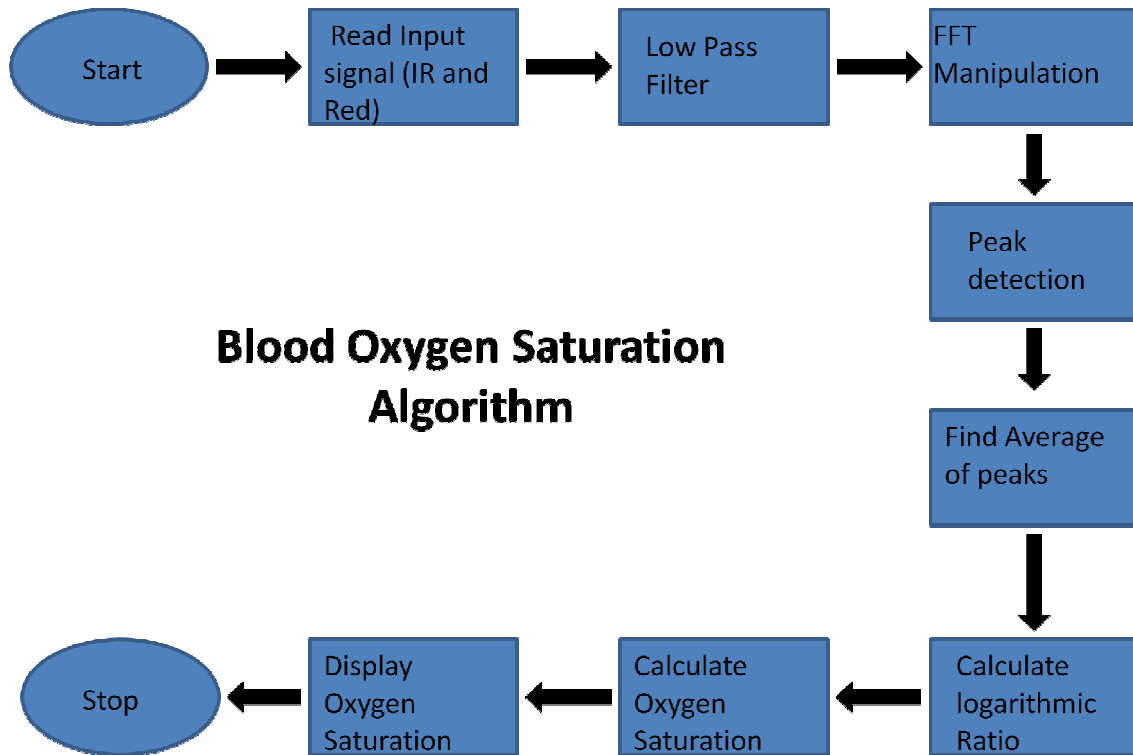


Figure 5.12 A flow chart representing the algorithm blood oxygen saturation algorithm

In the following section, the algorithm is followed step by step along with the output at each step. The output at each step serves as a test point and provides an indication to whether the whole algorithm is heading in the right direction or not. As previously mentioned the signal is acquired by the data acquisition system and then imported into the computer via a USB. The data is stored and then opened in the software MATLAB. Once the data is read into MATLAB, the data looks like the following. The data shown on top is from the infra red diode and one shown below is from red diode. The red amplitude appears to be greater than infra red amplitude.

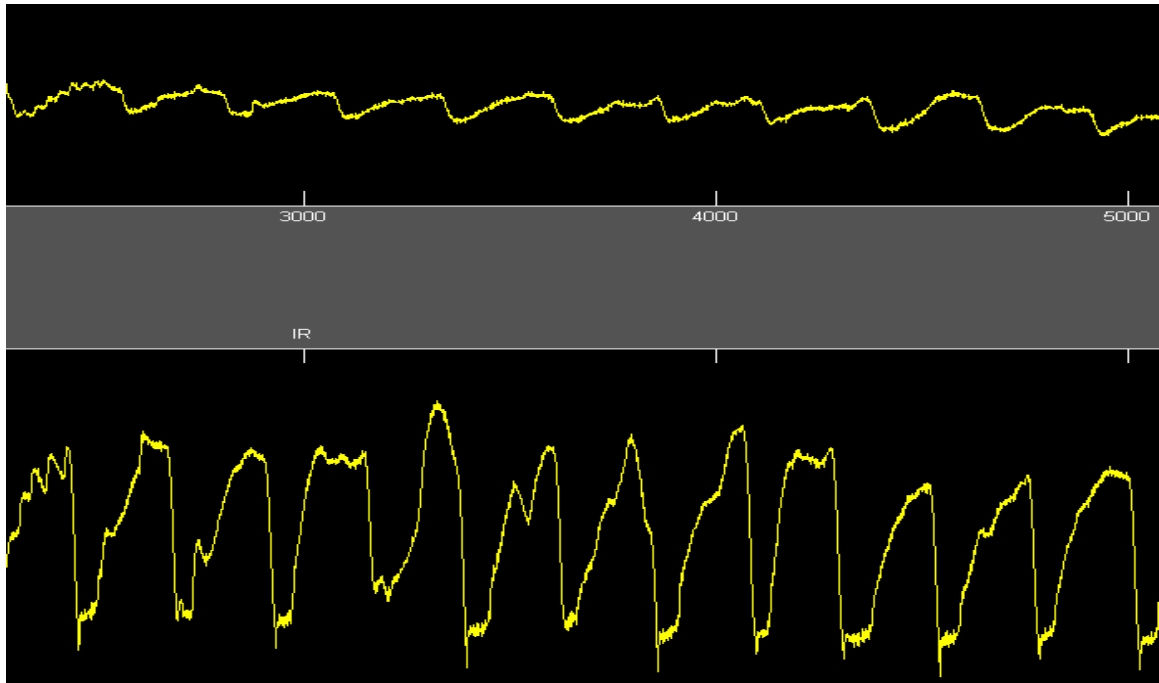


Figure 5.13 Infra red and red data from the transducer for blood oxygen content

Once the data is in the computer, it is passed through a low pass filter. The filter is a third order Butterworth filter with cut-off frequency of 5 Hz. This also gets rid of the 60Hz power noise. As it is known, the input signal is a function of heart beat of the patient, a cut-off of 5 Hz guarantees a wider spectrum of patients with varying heart rate to be okay when using this blood oxygen measurement device.

After the low pass filter, Fourier manipulation is performed to get rid of the DC offset and floating DC value. The data is converted into frequency domain by taking the Fourier transform of the input data. Once this is done, the extremely low frequency components based around 0Hz are removed. This gets rid of the DC as well as the floating DC offset. As observed about the floating DC is very low frequency and is therefore removed once the frequency around 0 Hz is removed. Once the data is converted back to time domain, the following output is obtained. It is evident from the following figure, there is no DC offset or floating DC. Now this data could be utilized for further post-processing.

Local peak detection is performed after this step to find the amplitude of the peaks. This is done by identifying variations among local maxima and minima. A threshold is specified and if the difference between a local minima and maxima exceeds that threshold, a local peak is detected. This algorithm works very well as long the data being analyzed is not contaminated by noise. As seen from the above figure, the data does not look noisy and is ideal for local peak detection. Following figure demonstrates the peak detection. The peaks are shown by green stars.

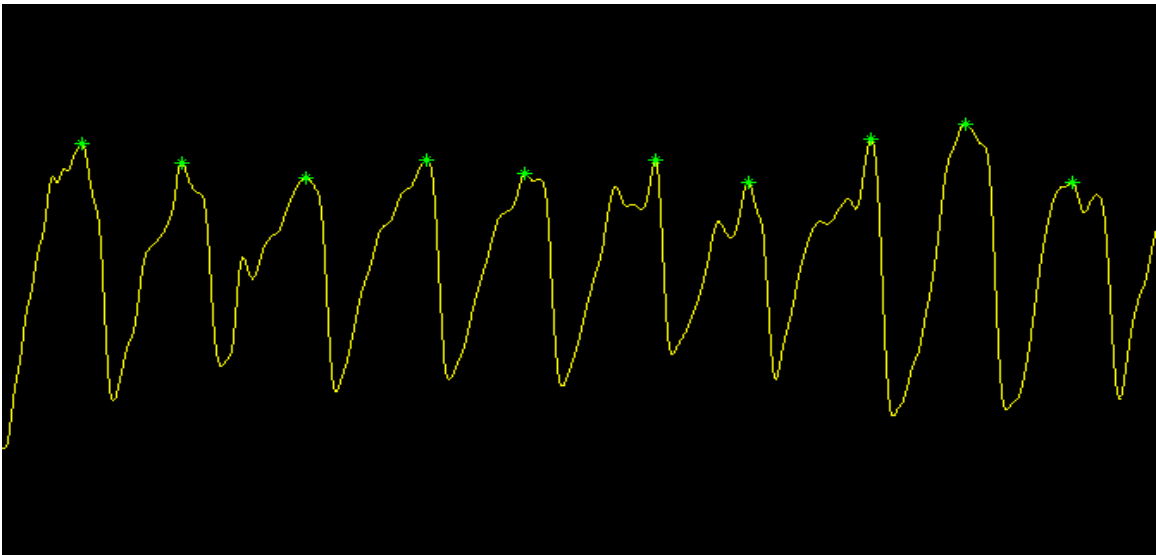


Figure 5.14 Infra red diode signal after filtering, Fourier manipulation and peak detection

Similar procedure is performed on the infra red data. After the low pass filtering, Fourier manipulation to get rid of DC value and local peak detection, the data is given in the following figure.

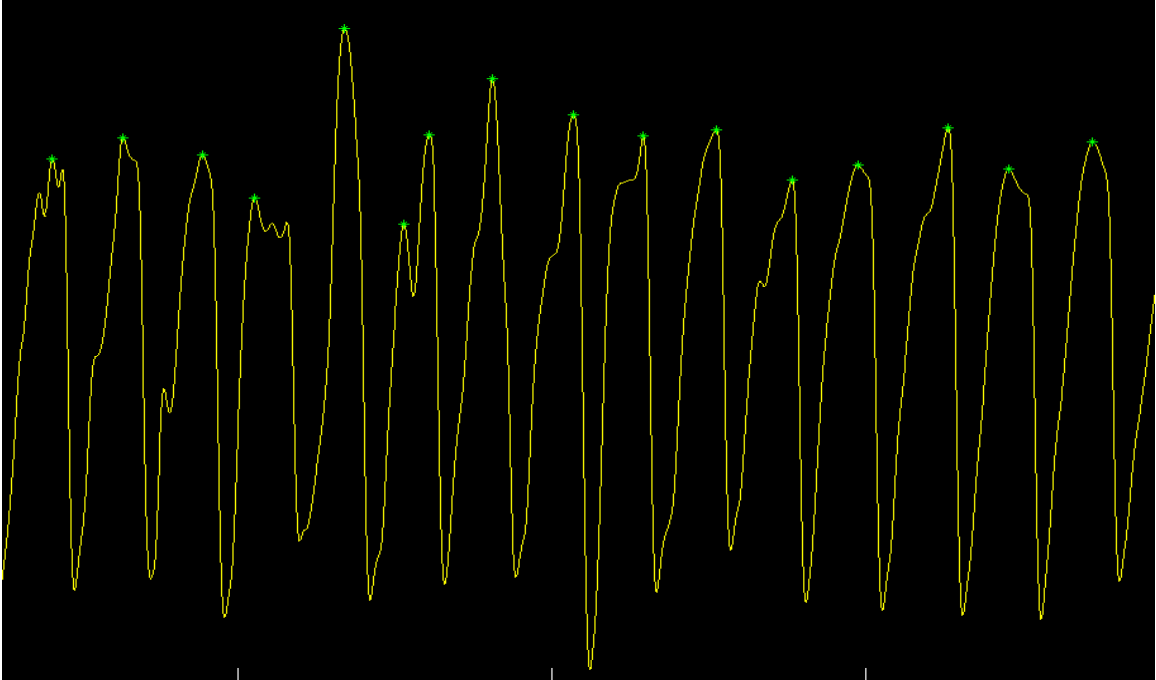


Figure 5.15 Red diode signal after filtering, Fourier manipulation and peak detection

Finally the average of red and infra red diode signals is computed and is given by the following numbers.

$$ir_avg = 0.1170$$

$$red_avg = 0.4657$$

A logarithmic ratio is computed after this, given by

$$raio = \log(red_avg)/\log(ir_avg) [10]$$

And this is given by the following number

$$ratio = 0.3561$$

Finally, the blood oxygen saturation is given by the formula

$$saturation = (ratio*(3.713935628) - 17.28085507)/(ratio*(-2.788022512) - 15.56913667)) [10]$$

This provides a saturation given by following number

$$saturation = 0.9635$$

The saturation is around 96.35% which indicates a healthy heart and lungs.

Type of Filter	Cut-off frequency	Order
Notch filter	Around 60Hz	5
Low pass filter	5Hz	3
High pass filter(using Fourier manipulation)	Very low, around 0 Hz to 0.05Hz	1

Table 5.3 Table describing filter characteristic used in blood oxygen saturation algorithm

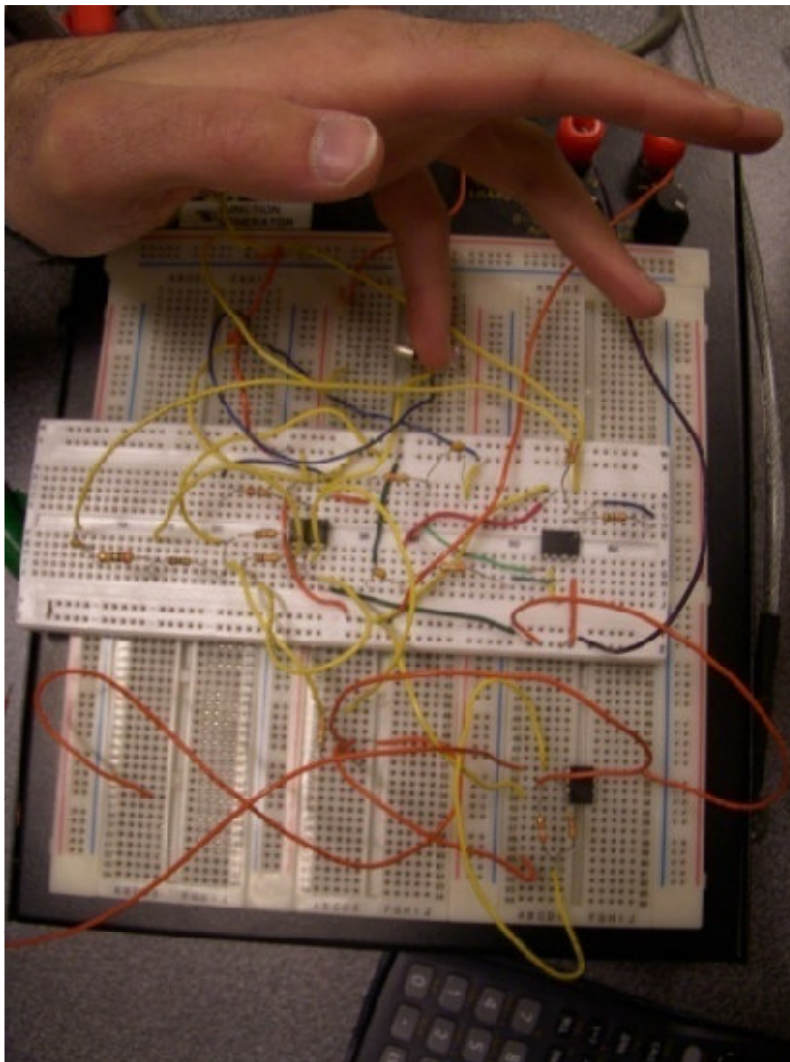


Figure 5.16 Hamzah Qureshi using his blood oxygen saturation module

Chapter 6

ECG Case Study: Arrhythmia

6.1 Introduction

The software used to implement all of the post-processing is MATLAB. Once the data for each physiological signal is imported, a set algorithm is followed to get valid result from the raw data. These algorithms will be described in the following sections along with the explanation of each block in the algorithm. The algorithm looks very similar to the regular ECG described in the previous chapter. The basic principles are the same except for a few minute changes. The block diagram is exactly the same as regular ECG.

6.2 Case study Algorithm

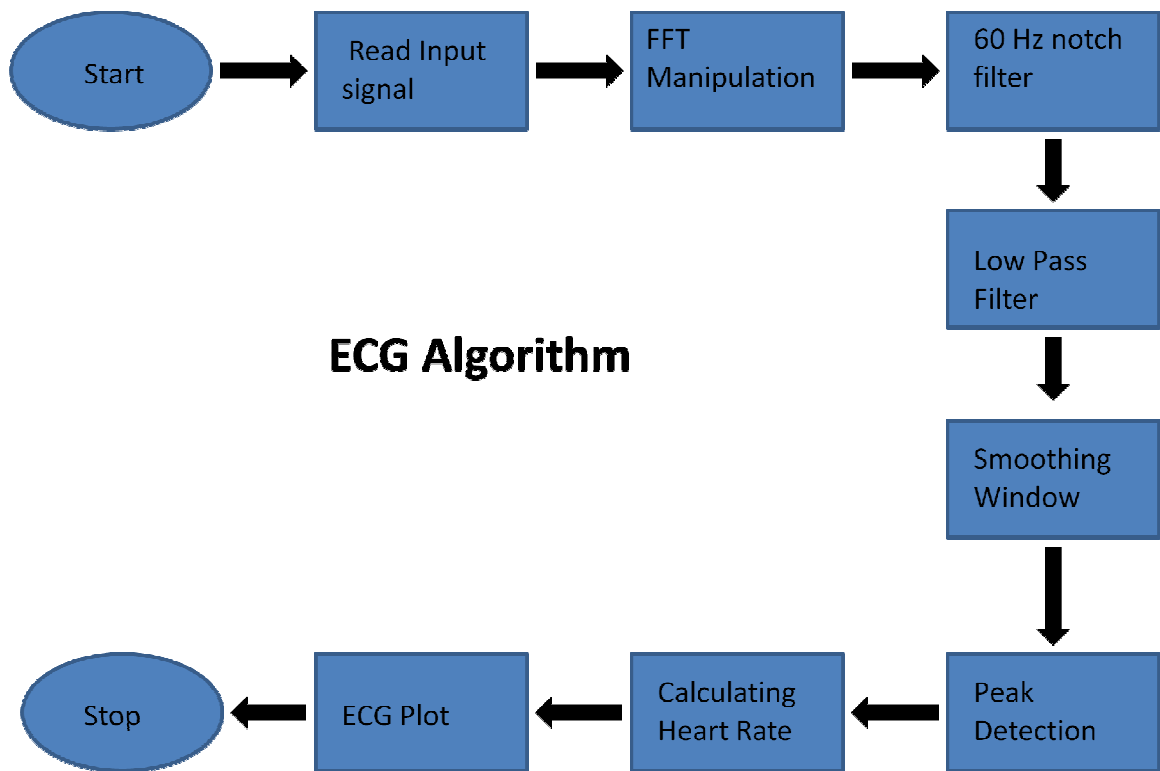


Figure 6.1 A flow chart representing the algorithm for ECG processing

As previously described the signal is acquired by the data acquisition system and then imported into the computer via a USB. The data is stored and then opened in the software MATLAB. Once the data is read into MATLAB, the ECG case study data looks like the following.

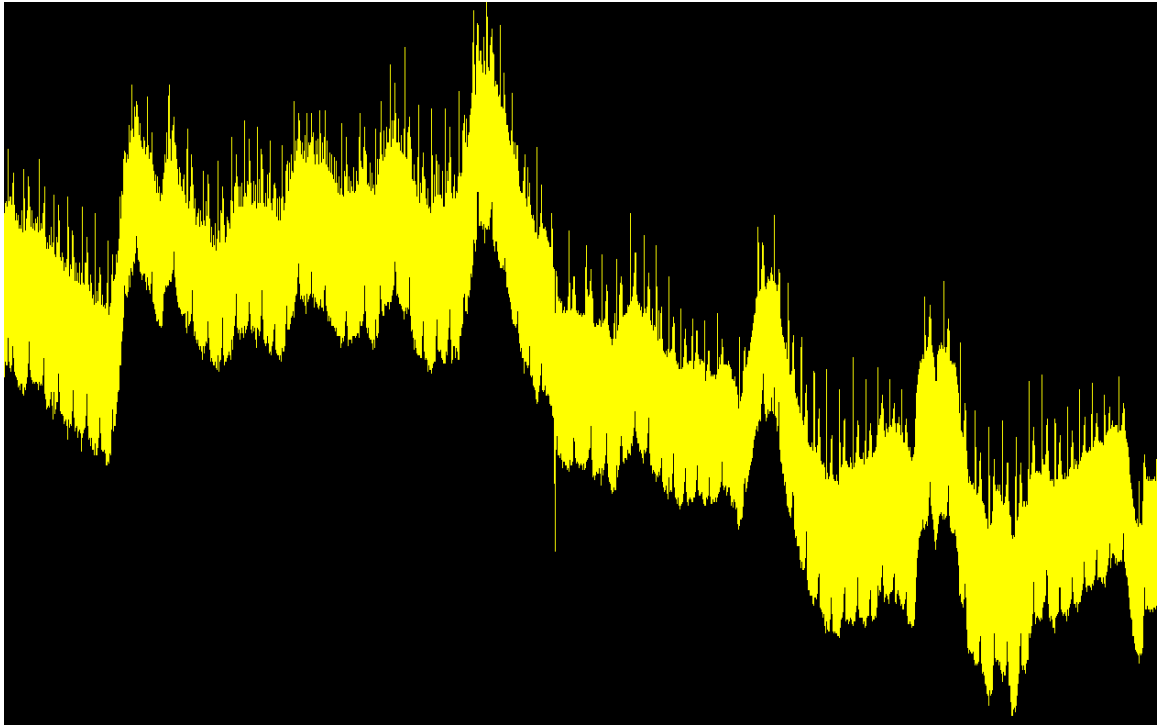


Figure 6.2 Raw ECG case study data

As seen from the picture of raw data, a floating DC value is observed. This is a consequence of the floating ground for the transducer. This poses a problem as the commercial ECG machines seen on market produce a perfectly straight line ECG output. This problem is solved by performing some fast Fourier transformation manipulation. The data is converted into frequency domain by taking the Fourier transform of the input data. Once this is done, the extremely low frequency components based around 0Hz are removed. This removes the DC as well as the floating DC offset. As observed, the floating DC is very low frequency and is therefore removed once the frequency around 0 Hz is removed. Once the data is converted back to time domain, the following output is

obtained. As it is apparent from the figure, there is no DC offset or floating DC. Now this data is able to be utilized for further post-processing.

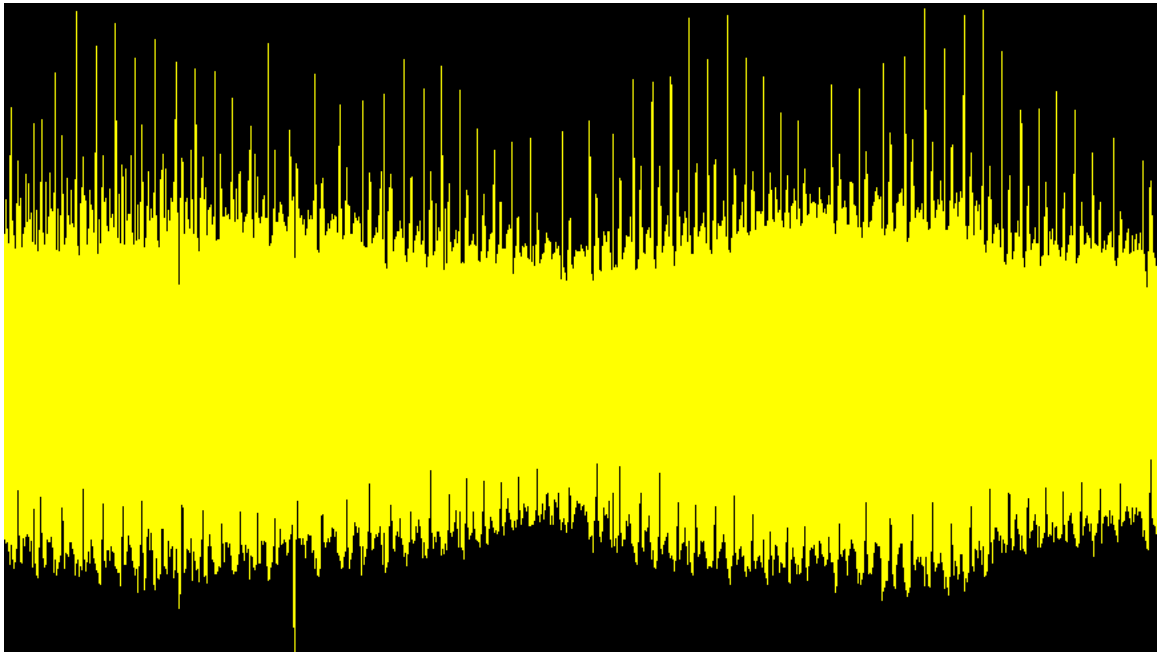


Figure 6.3 ECG case study data after Fourier manipulation

The above figure undoubtedly has some issues with the noise associated with the input. Power noise based around 60 Hz is the biggest culprit in contamination by noise. This could be removed by implementing a band stop filter or a notch filter at 60 Hz. For this purpose a notch filter of the fifth order is used. This leads to removal of 60 Hz noise

and it provide us the following result.

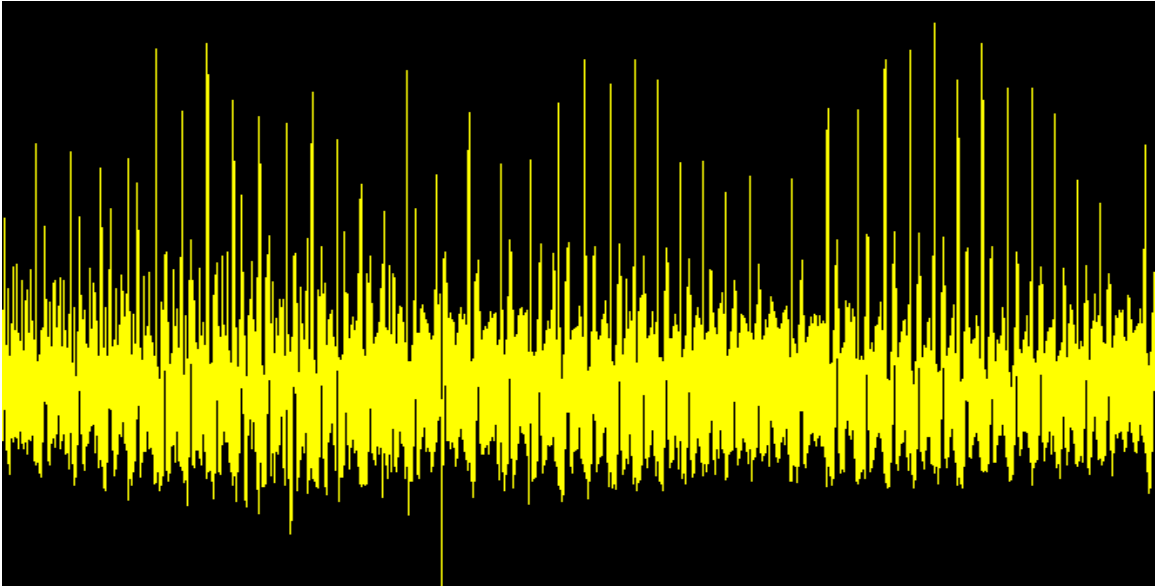


Figure 6.4 ECG case study data after power noise removal

As it is obvious the data still has a lot of noise associated with it. To get rid of some of the noise, a low pass filter is implemented. This filter is of the type Butterworth, third order and has a cut-off frequency of 35Hz. The output after the low pass filter is shown below.

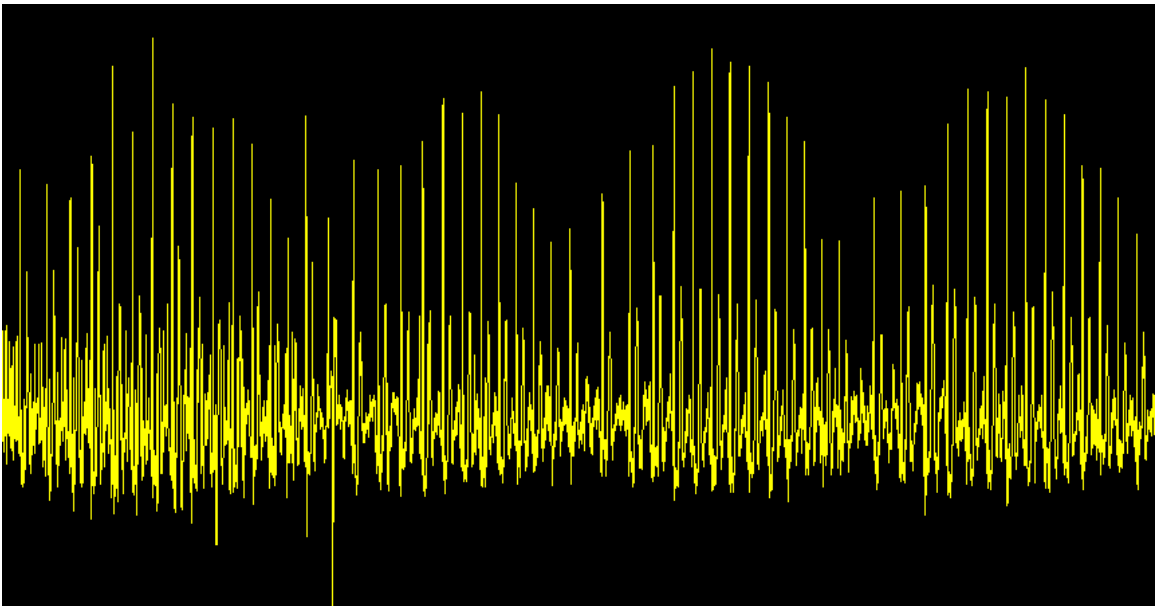


Figure 6.5 ECG case study data after passing through a low pass filter

If the data above is analyzed carefully, it is seen that the QRS complex seems to increase and decrease in amplitude in a periodic fashion. This is definitely intriguing as further filtering should be performed to reveal the true nature of ECG waveform. To address the issue of additional noise, a smoothing window is implemented. The smoothing window just represents each sample as an average of that sample and a few around it. The number of samples to be averaged is a crucial decision in the implementation of a smoothing window. It further depends on the sampling frequency of the input signal. In this particular case, the number of samples to be averaged is eight. Output after a smoothing window is shown in the following figure.

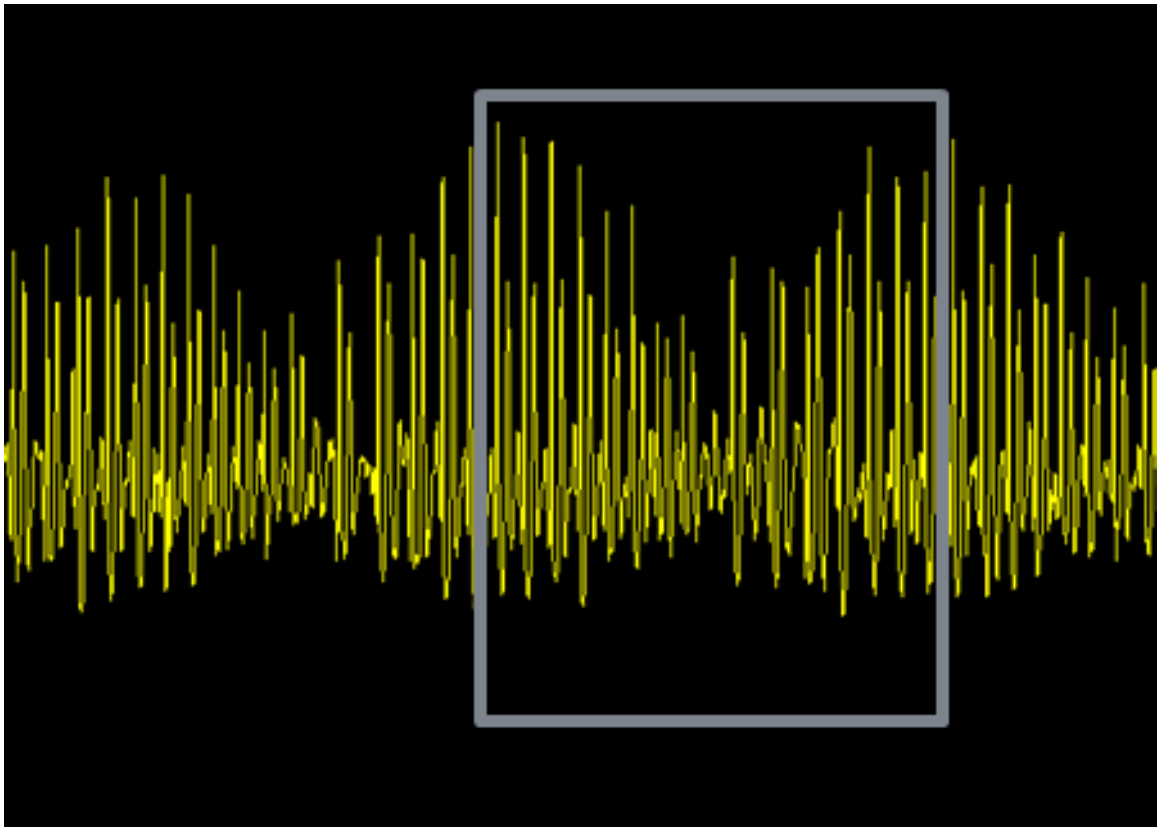


Figure 6.6 the output after implementation of a smoothing window

Demonstrated in the figure above, the periodicity in the amplitude of overall ECG is apparent. Although a perfect ECG waveform cannot be distinguished, a region of

ECG is picked and magnified in the figure shown below. This would provide some more insight in to the issue of varying ECG amplitude.

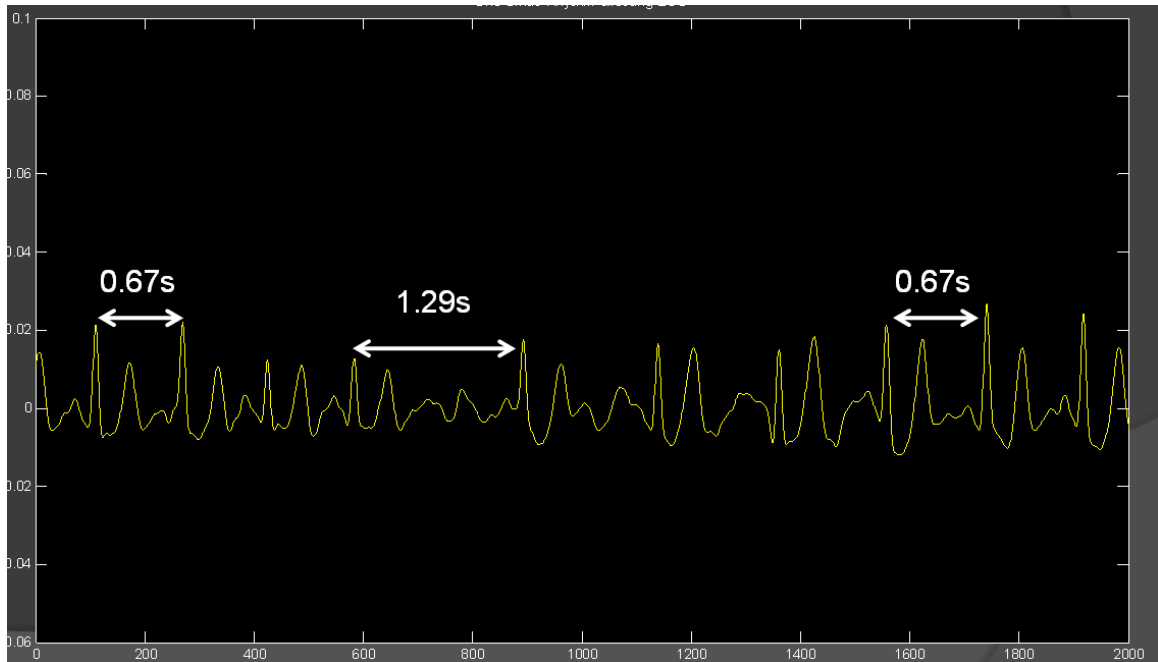


Figure 6.7 ECG case study data zoomed in

If the picture above is analyzed carefully, it is evident that the interval between adjacent QRS complexes is increasing and decreasing. The time interval shown above is the interval between QRS complexes. This represents one heart beat. The time varies between 0.67 second to 1.29 second. This is equivalent to a heart rate of 89 and 46 respectively. These findings reveal some sort of problem with the heart. Furthermore, the following figure is shown with peak detection and another critical discrepancy is revealed.

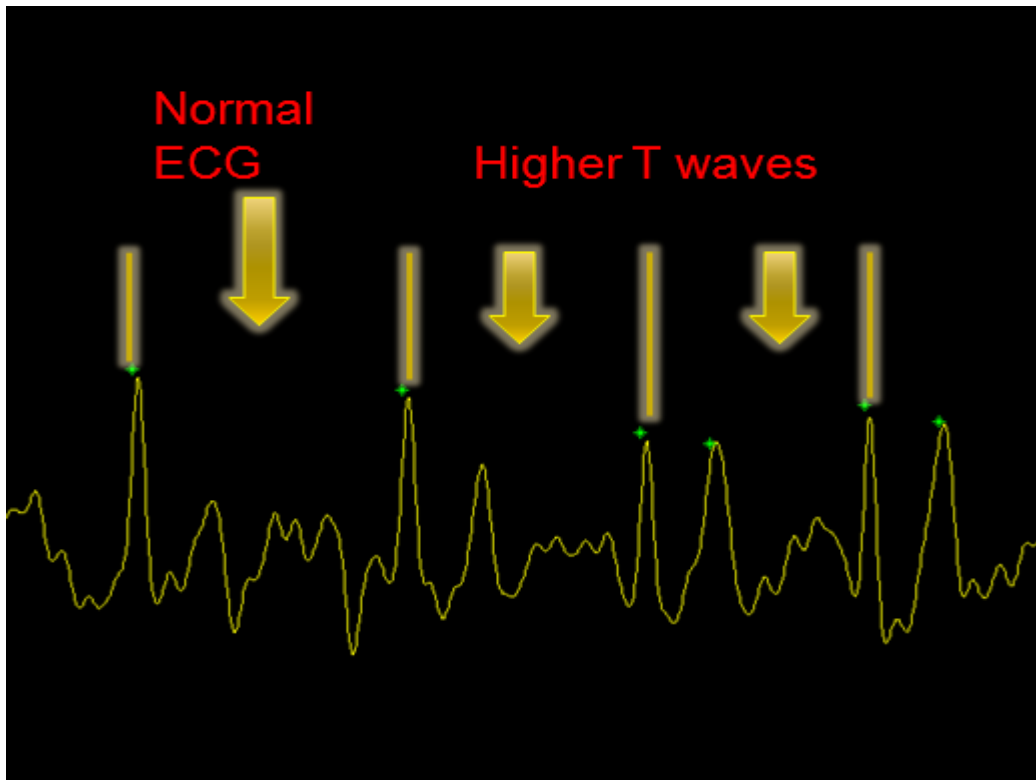


Figure 6.8 ECG case study after peak detection, discrepancy between different peaks

The graph above shows the deviation from normal ECG and an increase in the amplitude of T waves. The T waves are seen to have almost the same amplitude as the QRS complex. This is definitely not a characteristic of the regular ECG waveform. After analyzing this figure and the figure before, it is apparent that the patient has certain problem associated with the heart.

Researching varying heart rate and higher T waveform revealed the arrhythmia that the patient suffered from. The result was sinus arrhythmia in which the breathing action leads to the stimulation of the vagus nerve. Vagus nerve is responsible for sympathetic stimulation of the heart. When the breathing action leads to stimulation of vagus nerve, it further leads to stimulation of the heart [11]. This is much evident in figure 6.6 where the ECG can be seen varying with the breathing in and breathing out action performed by the patient.

Chapter 7

Results and Discussion

7.1 Wireless Module

The wireless module worked without any major problems. The input to the transmitter had to be maximized to a level of 5 Volts. The higher input ensured better quality of the reconstructed data on the receiver end utilizing the inherent bit resolution. Specialized antennas were hard to find in the market but worked flawlessly nonetheless. The form factor of the antennas makes them extremely portable and in tune with the whole small form factor for the complete design. The receiver has an output of around 55 millV. This was amplified around 56 times using a simple operational amplifier and a few resistors. The circuit is the most basic amplifier circuit with negative feedback of the operational amplifier connected to ground. This provided an output of 3.1 V at the data acquisition module.

7.2 Data Acquisition module

The data acquisition module takes the analog input from the wireless module and samples it to convert it into digital domain. The sampling frequency is 240 Hz. Going by the Nyquist rate, this frequency is able to represent a frequency of up to 120 Hz. The highest frequency component in the associated physiological signals is around 35Hz. Therefore, this adequately covers the entire frequency domain of the physiological signals. The analog to digital converter has a bit resolution of 10 bits. This is spread over a span of -10 V to +10 V.

After the conversion into digital domain, the data is sent to the computer via a USB. This data is handled by the proprietary software provided by dataq instruments, the manufacturer of data acquisition system. The data is saved in this software and later on imported into MATLAB for post-processing. The import is done via the MATLAB code files provided by Dataq instruments as well. The code is shown in the appendix. The readme file has full instructions on how to import the data into MATLAB.

7.3 Electrocardiography signal

As it is demonstrated in figure 5.7 the final ECG data obtained looks very similar to the standard ECG waveform. All of the components, P wave, QRS complex and T wave are represented accurately. The processing of ECG signal includes Fourier manipulation, removal of power noise, low pass filtering, smoothing window, peak detection, plotting the ECG and calculation of the heart rate. Fourier manipulation gets rid of the DC offset and the floating DC problem. The notch filter helps remove the power noise that is centered around 60Hz. Low pass filtering further gets rid of some of the other noise such as electrical activity from muscle stimulation. The smoothing window enables a smoother looking plot of the ECG. Finally, the local peaks are calculated using local maxima and minima. The QRS complex is used as the local peak in ECG waveform since it is the highest peak. Peak detection also helps identify the heart rate of the patient. Finally, the ECG is plotted in real time. It looks fairly similar to the plot shown on the commercial ECG machines, only delayed by a fixed time offset.

The case study on ECG provides a proof of the robustness of the post-processing algorithm. The simple two electrode ECG hardware along with some detailed post-processing was able to indicate sinus arrhythmia. This is a good indication of the scope of this project.

7.4 Blood pressure measurement

Blood pressure measurement also relies heavily on the post-processing. The input signal is just the pressure variation in the cuff on patients arm. From that input signal, the oscillations in pressure in the cuff due to blood flow are isolated. This is done by using a high pass filter. Once this is done, Fourier manipulation is applied to get rid of the DC. The filtered and amplified peak array is of extreme importance to find the blood pressure of the patient. From the peaks, depending on their location in the time domain and the amplitude, mean arterial pressure and systolic pressure is calculated. The diastolic pressure is a function of mean arterial pressure and systolic pressure. From a collection of data samples, blood pressure was calculated and recorded. Blood pressure was also

recorded using a commercial Omron machine on the same patient. The error rate in systolic pressure is around 3.7 percent and in diastolic pressure is around 2.6 percent (based on readings shown in chapter 5). This proves that the numbers are similar to the commercial device.

7.5 Blood oxygen content measurement

Blood saturation level is calculated by using an infra red and a red diode. The output from the transducer is a periodic waveform that is a function of the heart rate. Both infra red and red signals are recorded. These are then passed through a notch filter to reject the power noise. Fourier manipulation is also performed that rejects most of the DC offset and floating DC. The signals are then passed via a low pass filter. Peak detection is performed after this and amplitudes are calculated. The amplitudes are then put into a formula that returns blood oxygen saturation level. From the analysis in chapter 5, the blood saturation level obtained was around 96 percent. This number represents a healthy patient with no heart or lung problems.

Chapter 8

Conclusions and Recommendations

The objective was to create a device that records three different physiological signals, transmits them wirelessly and performs certain processing to yield meaningful results that could be interpreted by health care professionals for basic diagnosis.

The team as a whole was able to create this device. The signals were processed in a similar fashion to the commercially available devices. The accuracy of Quick Doc was not too far off the commercial devices. It should also be noted that no funding was available for the project. The most basic, cheaper electrical components were used to construct the project. This provides a proof of concept as in how simple components with a deeper understanding of the functioning could be used to construct fairly useful devices. This is fully in sync with the whole idea of what engineering is based around.

In the end the device is merely a proof of concept and cannot be used commercially. Due to time and funding constraints, the physical form of the device is not exactly as mentioned in the project proposal. The objectives changed to a great extent over the past eight months. For future implications, more emphasis should be put into the hardware of the device. The transducers play a big role in the quality of physiological signal obtained. The software model can only work around a predictable set of input signals. Although it is possible to isolate the signal and noise, in certain cases the spike from electrical activity cannot be separated from ECG signal as both have same frequency range as well as amplitude range.

The software model for future practice could be modelled on a microcontroller instead of using software such as MATLAB. Most of the commercial devices have the embedded microcontroller based structure. Due to the unpredictability of input signals and because of the lack of research in this field led the team Quick Doc to using MATLAB. Ultimately, the device is an evidence of a proof of concept.

Appendices

(Note: all the files in following appendices are to be placed in the same directory. Detailed instructions on how to import data and run the program are provided on a separate CD. It also includes all the data files, code files and some recorded video files)

A.1 MATLAB code for importing input data

A.1.1 ReadWinDaqFile.m

%This M-file reads data from a WinDaq file and plots the data.

%ReadWinDaqFile takes 4 parameters:

%WinDaqFileName = full path to the WinDaq (.wdq) data file.

%StartScan = scan point to start collecting data from, 0 = beginning of %file.

%NumScanPts = number of channel scans to be returned. One scan is defined %as one pass through all enabled channels (i.e., one sample per channel).

%This value must be between 1 and 32,767. NOTE: the scan starts at 0, so %if you want 200 scans, enter NumScanPts = 199

%DataFormat = format of the data returned. Valid settings are: 0 for 2's %complement binary data (FormatBinary); and 1 for data scaled in %engineering units (FormatScaled).

*function dqAnalysis =
ReadWinDaqFile(WinDaqFileName,StartScan,NumScanPts,DataFormat)*

*%Create the readdataqfile object in a figure window
readdataqfile1 = actxcontrol('DATAQFILE.ReadDataqFileCtrl.1',[10,10,5,5]);*

*%Register the ReadDataqFile Events
readdataqfile1.registerevent({'FileError','dataqfileerror','EndOfFile','endoffile','ControlError','controlerror'});*

*%Select the WinDaq file that you want to read from
set(readdataqfile1, 'FileName', WinDaqFileName);*

```

%Open the WinDaq file
readdataqfile1.Open;

%Move the pointer to the StartScan position from the beginning of the
%file(0)
readdataqfile1.MoveTo(StartScan, 0);

%Display data on the screen, and put into matrix dqAnalysis
dqAnalysis = readdataqfile1.GetData(NumScanPts, DataFormat);

%View the number of Channels and data rows in the file and assign them to
%variables dqRows, dqChannels
[dqChannels,dqDataPts] = size(dqAnalysis);

%Create a time vector, dqTime, of integers from 1 to dqDataPts.
EndScan = dqDataPts + StartScan - 1;
dqTime = StartScan:EndScan;

%Plot the channels versus data points and annotate the plot
set(0,'defaultaxeslinestyleorder','-')
set(0,'defaultaxescolororder',[1 0 0;0 1 0;0 0 1;1 1 0;0 0 0;1 0 1;0 1 1])
plot(dqTime,dqAnalysis)
xlabel('WinDaq Data Scans'), ylabel('Units Measured'), grid on

%This function closes ReadDataqFile and deletes it. Delete the next two
%lines if you want to use readdataqfile1 after you run the function.
%NOTE: the output variable (value = dqAnalysis data matrix) will still be
%useable for computations.
readdataqfile1.Close
readdataqfile1.delete

%For more information about using the DATAQ ActiveX controls in MATLAB, go
%to http://www.dataq.com/develop/
%For help with this file, please post a message to the DATAQ Forum at
%http://support.dataq.com

```

A.1.2endoffile.m

```

%This function is triggered by the event EndOfFile for the ReadDataqFile
%control. You must register the event EndOfFile as endoffile so that it will
%call endoffile.m when the event fires.
function endoffile(varargin) %varargin holds a variable number of arguments
disp('End of file has been reached!') %display end of file message

```

A.1.3 dataqfileerror.m

```
%This function is triggered by the event FileError for the ReadDataqFile
%control. You must register the event FileError as newdata so that it will
%call newdata.m when the event fires.
function dataqfileerror(varargin) %varargin holds a variable number of arguments
disp(varargin) %display the error
```

A.1.4 controlerror.m

```
%This function is triggered by the event ControlError for the ReadDataqFile
%and DataqSDK controls. You must register the event ControlError as
%controlerror so that it will call controlerror.m when the event fires.
function controlerror(varargin) %varargin holds a variable number of arguments
disp(varargin) %display the error
```

A.2 MATLAB code for ECG processing

A.2.1 ECG_filter.m

```
clear;
clc;

%reading data from the saved file
%input = ReadWinDaqFile('C:\Documents and Settings\brown bear\Desktop\BI6
recordings\casestudyecg.wdq',0,7200,1);
%input = ReadWinDaqFile('C:\Documents and Settings\brown bear\Desktop\BI6
recordings\casestudyecg.wdq',0,7200,1);
input=importdata('ashEcg.mat');
fs=240;
% sampling the data down, picking one in every 10 samples
% and typecasting as double
%A=double(input(1:10:length(input)));
%A(:)=-A(:);
%subplot 311, plot(input); title('Original Signal');
figure,plot(input); title('Original Signal');
```

```

fresult=fft(input);
fresult(1:round(length(input)*2/fs))=0;
fresult(end-round(length(fresult)*2/fs):end)=0;
ecg=real(iff(fresult));

%subplot 312,plot(ecg);title('ECG after fft manipulation');

figure,plot(ecg);title('ECG after fft manipulation');

%getting rid of 60Hz noise
%stop band from 50Hz to 120 Hz
StopBand=[2*50/240 2*119/240];
[b,a]=butter(5,StopBand,'stop');
ecg1=filter(b,a,ecg);
%subplot 313, plot(ecg1);title('ECG after removal of 60 Hz noise');
figure,plot(ecg1);title('ECG after removal of 60 Hz noise');

%implementing a low pass filter to smooth out some of
%the high frequency components
lowCutoff=2*30/240;
[b,a]=butter(3,lowCutoff);
ecg2=filter(b,a,ecg1);

figure,
%subplot 211, plot(ecg2);title('ECG after low pass filter of 30 Hz');
plot(ecg2);title('ECG after low pass filter of 30 Hz');

%implementing a smoothing window to remove
%any additional sharp peaks

ecg3=fastsmooth(ecg2,6,3);
%subplot 212, plot(ecg3);title('ECG with a smoothing window');
figure,plot(ecg3);title('ECG with a smoothing window');
%detecting the peaks
[highpeaks, lowpeaks]=peakdet(ecg3,.05);

figure,plot(ecg3);
hold on;
plot(highpeaks(:,1),highpeaks(:,2),'*g');title('Peak detection');

heart_rate_per_min=round((length(highpeaks)/(length(input)/fs))*60)

%plotting the ecg
plot_ECG(ecg3,fs);

```

A.2.2 plot_ecg.m

```
function [] = plot_ECG(data, fs)
%data=importdata('out2.mat');
%fs=4410;
%sample data down by a factor of 100
%sd=100;
%data=data(1:sd:length(data));
%delay depends on the sampling frequency
%fn=fs/100;
update_time=0.1;
%tw=1/fn;
ini_wait=0.5;
plot_start=1;
plot_end=round(fs*update_time);
% fs=2;
% time = length(data)/fs;
whitebg('black');
%scrnz=get(0,'ScreenSize');
%figure('Position',[1 scrsz(4)/3 scrsz(3)/3 scrsz(4)/3])

%one window updates the ecg for 5 seconds
%userinput=input('Enter the length of ECG window in seconds:');
prompt=('Enter the length of ECG plot window in seconds');
name=('Length of ECG window');
numlines=1;
%defaultanswer='5';
userinput=inputdlg(prompt,name,numlines);
windowSize=(str2num(userinput{1}))*fs;
figure,axis([0 windowSize 2*min(data) 2*max(data)]);
myWait(ini_wait);
windowStart=1;
windowEnd=windowSize;

while (plot_end<length(data))
    while(plot_end<windowEnd)
        for i=plot_start:plot_end
            %axis([0 length(data) 2*min(data) 2*max(data)])
            %plot(i,data(i));
            tempx(i)=data(i);
            tempy(i)=i;
            %hold on;
            %myWait(tw);
        end;
        axis([windowStart windowEnd 2*min(data) 2*max(data)]);
        plot(plot_start:plot_end,data(plot_start:plot_end),'g');
```

```

hold on;
myWait(update_time);
plot_start=plot_end;
plot_end=plot_end+round(fs*update_time);
end;
windowStart=windowEnd;
windowEnd=windowEnd+windowSize;
end;

```

A.2.3 myWait.m

```

% Waits for the specified number of seconds
function myWait(DeltaT)
if(DeltaT>0) %end condition
    t=timer('timerfcn','myWait(0)','StartDelay',DeltaT);
    start(t);
    wait(t);
end

```

A.2.4 peakdet.m

```

function [maxtab, mintab]=peakdet(v, delta, x)
%PEAKDET Detect peaks in a vector
% [MAXTAB, MINTAB] = PEAKDET(V, DELTA) finds the local
% maxima and minima ("peaks") in the vector V.
% MAXTAB and MINTAB consists of two columns. Column 1
% contains indices in V, and column 2 the found values.
%
% With [MAXTAB, MINTAB] = PEAKDET(V, DELTA, X) the indices
% in MAXTAB and MINTAB are replaced with the corresponding
% X-values.
%
% A point is considered a maximum peak if it has the maximal
% value, and was preceded (to the left) by a value lower by
% DELTA.

% Eli Billauer, 3.4.05 (Explicitly not copyrighted).
% This function is released to the public domain; Any use is allowed.

maxtab = [];
mintab = [];

v = v(:); % Just in case this wasn't a proper vector

```

```

if nargin < 3
    x = (1:length(v))';
else
    x = x(:);
    if length(v)~= length(x)
        error('Input vectors v and x must have same length');
    end
end

if (length(delta(:)))>1
    error('Input argument DELTA must be a scalar');
end

if delta <= 0
    error('Input argument DELTA must be positive');
end

mn = Inf; mx = -Inf;
mnpos = NaN; mxpos = NaN;

lookformax = 1;

for i=1:length(v)
    this = v(i);
    if this > mx, mx = this; mxpos = x(i); end
    if this < mn, mn = this; mnpos = x(i); end

    if lookformax
        if this < mx-delta
            maxtab = [maxtab ; mxpos mx];
            mn = this; mnpos = x(i);
            lookformax = 0;
        end
    else
        if this > mn+delta
            mintab = [mintab ; mnpos mn];
            mx = this; mxpos = x(i);
            lookformax = 1;
        end
    end
end
end

```

A.2.4 fastsmooth.m (from Haver 2008)

```
function SmoothY=fastsmooth(Y,w,type,ends)
% fastsmooth(Y,w,type,ends) smooths vector Y with smooth
% of width w. Version 2.0, May 2008.
% The argument "type" determines the smooth type:
% If type=1, rectangular (sliding-average or boxcar)
% If type=2, triangular (2 passes of sliding-average)
% If type=3, pseudo-Gaussian (3 passes of sliding-average)
% The argument "ends" controls how the "ends" of the signal
% (the first w/2 points and the last w/2 points) are handled.
% If ends=0, the ends are zero. (In this mode the elapsed
% time is independent of the smooth width). The fastest.
% If ends=1, the ends are smoothed with progressively
% smaller smooths the closer to the end. (In this mode the
% elapsed time increases with increasing smooth widths).
% fastsmooth(Y,w,type) smooths with ends=0.
% fastsmooth(Y,w) smooths with type=1 and ends=0.
% Example:
% fastsmooth([1 1 1 10 10 10 1 1 1 1],3)=[0 1 4 7 10 7 4 1 1 0]
% fastsmooth([1 1 1 10 10 10 1 1 1 1],3,1,1)=[1 1 4 7 10 7 4 1 1 1]
% T. C. O'Haver, 2008.
if nargin==2, ends=0; type=1; end
if nargin==3, ends=0; end
switch type
case 1
    SmoothY=sa(Y,w,ends);
case 2
    SmoothY=sa(sa(Y,w,ends),w,ends);
case 3
    SmoothY=sa(sa(sa(Y,w,ends),w,ends),w,ends);
end

function SmoothY=sa(Y,smoothwidth,ends)
w=round(smoothwidth);
SumPoints=sum(Y(1:w));
s=zeros(size(Y));
halfw=round(w/2);
L=length(Y);
for k=1:L-w,
    s(k+halfw-1)=SumPoints;
    SumPoints=SumPoints-Y(k);
    SumPoints=SumPoints+Y(k+w);
end
s(k+halfw)=sum(Y(L-w+1:L));
SmoothY=s./w;
```



```

% Taper the ends of the signal if ends=1.
if ends==1,
    startpoint=(smoothwidth + 1)/2;
    SmoothY(1)=(Y(1)+Y(2))./2;
    for k=2:startpoint,
        SmoothY(k)=mean(Y(1:(2*k-1)));
        SmoothY(L-k+1)=mean(Y(L-2*k+2:L));
    end
    SmoothY(L)=(Y(L)+Y(L-1))./2;
end

```

A.3 MATLAB code for blood pressure measurement

A.3.1 bp_daq_one_ch.m

```

clear;
clc;

gain=10;

%input=importdata('bp_data.mat');
input=importdata('bp_daq_ori_data_kundan.mat');
bp_ori=input;

fs=240;
fc=.4;

%getting rid of 60Hz noise
%stop band from 50Hz to 120 Hz
StopBand=[2*50/240 2*119/240];
[b,a]=butter(5,StopBand,'stop');
input=filter(b,a,input);

[b,a]=butter(3,fc*2/fs,'high');
bp_peaks=filter(b,a,input);

fresult=fft(bp_peaks);
fresult(1:round(length(input)*1/fs))=0;
fresult(end-round(length(fresult)*1/fs):end)=0;
bp_peaks=real(ifft(fresult));

bp_peaks=gain*bp_peaks;

%bp_peaks=input(2,:);
% sampling the data down, picking one in every 10 samples

```

```

% and typecasting as double
%A=double(input(1:10:length(input)));
%A=double(input);
fs=240;
time=length(bp_ori)/fs;

% end;

[Voltage_inf_stop,inf_stop_index]=max(bp_ori);

%finding peaks in the bp signal
[maxtab, mintab] = peakdet(bp_peaks, gain*.01);
%looking for the highest peak
highest_peak=0;
index=0;

%looking for highest peak, look only
%beyond the inflation point
%index represents highest peak location

for i=1:length(maxtab)
    if(maxtab(i,1)>inf_stop_index)
        if(maxtab(i,2)>highest_peak)
            highest_peak=maxtab(i,2);
            index=i;
        end;
    end;
end;

%averaging the value around highest peak
number_peaks=3;
average_peak=0;
for i=1:number_peaks
    average_peak=average_peak+bp_ori(maxtab(index-2+i,1))/number_peaks;
end;

%looking for the nearest peak in maxtab
%when stopped inflating based on index inf_stop_index

for i=1:length(maxtab)
    diff(i)=inf_stop_index-maxtab(i,1);
end;

[a,stop_index]=min(abs(diff));

figure, plot(bp_peaks)

```

```

hold on; %plot(mintab(:,1), mintab(:,2), 'g*');
plot(maxtab(:,1), maxtab(:,2), 'r*');
hold on; plot(input,'k');

```

```

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

```

```

%calculations to calculate BP
%for systolic, using first two peaks
syspeakavg=(input(maxtab(stop_index+3,1))+input(maxtab(stop_index+4,1)))/(2);
Systolic_Pressure=(10*syspeakavg+0.2)/.133

```

```

Mean_Arterial_Pressure=((average_peak)*10+0.2)/.133

```

```

Diastolic_Pressure=(3/2)*(Mean_Arterial_Pressure - (1/3)*Systolic_Pressure)

```

A.4 MATLAB code for blood oxygen content measurement

A.3.1 blood_ox.m

```

clc;
clear;

ir=importdata('wireless_ir.mat');
red=importdata('wireless_red.mat');

subplot 211, plot(ir);title('IR');
subplot 212, plot(red);title('Red');

fs=240;

%getting rid of 60Hz noise
%stop band from 50Hz to 120 Hz
StopBand=[2*50/240 2*119/240];
[b,a]=butter(5,StopBand,'stop');
ir=filter(b,a,ir);

hcutoff=2*5/fs;
[b,a]=butter(3,hcutoff);
ir=filter(b,a,ir);

```

```

fresult=fft(ir);
fresult(1:round(length(ir)*.5/fs))=0;
fresult(end-round(length(fresult)*.5/fs):end)=0;
ir=real(ifft(fresult));

sum=0;
for i=1:length(ir)
    sum=sum+red(i);
end;
offset=sum/length(ir);

figure, plot(ir);

[highpeaks lowpeaks] = peakdet(ir,.1);
figure,plot(ir);
hold on;
plot(highpeaks(:,1),highpeaks(:,2),'*g');title('Peak detection IR');

ir_avg=0;
count=0;
for i=1:length(highpeaks(:,2))
    if(highpeaks(i,2)<.4)
        ir_avg=ir_avg+highpeaks(i,2);
        count=count+1;
    end;
end;
ir_avg=ir_avg/count

%red

%getting rid of 60Hz noise
%stop band from 50Hz to 120 Hz
StopBand=[2*50/240 2*119/240];
[b,a]=butter(5,StopBand,'stop');
red=filter(b,a,red);

hcutoff=2*5/fs;
[b,a]=butter(3,hcutoff);
red=filter(b,a,red);

fresult=fft(red);
fresult(1:round(length(red)*.5/fs))=0;

```

```

fresult(end-round(length(fresult)*.5/fs):end)=0;
red=real(iff(fresult));

%getting rid of DC
sum=0;
for i=1:length(red)
    sum=sum+red(i);
end;
offset=sum/length(red);

red=red-offset;
figure, plot(red);

[highpeaks lowpeaks] = peakdet(red,.1);
figure,plot(red);
hold on;
plot(highpeaks(:,1),highpeaks(:,2),'*g');title('Peak detection Red');

red_avg=0;
count=0;
for i=1:length(highpeaks(:,2))
    if(highpeaks(i,2)<.7)
        red_avg=red_avg+highpeaks(i,2);
        count=count+1;
    end;
end;
red_avg=red_avg/count

ratio=log(red_avg)/log(ir_avg)

saturation =(ratio*(3.713935628) - 17.28085507)/(ratio*(-2.788022512) - 15.56913667)

```

References

- [1] Hamzah Qureshi, blood oxymeter “Schematic of Quick Doc system”, December 20 2008.
- [2] Kundan Thind, Integration of various physiological signal, “Layout of Quick Doc: Personal contribution”, in association with Hamzah Qureshi. January 2009
- [3] “VIBES Electrocardiogram mosaic”, 2006. [Online] Available:
<http://www.vanth.org/vibes/electro.html> [Accessed: November 5 2008]
- [4] “A low cost RF module available at 900Mhz”, 2005. [Online]. Available:
<http://www.linxtechnologies.com/Products/RF-Modules/ES-Series-Digital-and-Analog-Wireless-Module/> [Accessed: February 5 2009]
- [5] “Low cost compact data acquisition kit”, 2005. [Online]. Available:
<http://www.dataq.com/support/documentation/pdf/datasheets/148ds.pdf> [Accessed: November 10 2008]
- [6] “CBC News indepth: HEALTH CARE,” May 12 2005. [Online]. Available:
<http://www.cbc.ca/news/background/healthcare/familydoctors.html> [Accessed: March 20, 2009].
- [7] Tom Bailey, “Waiting for a family doctor”, College of family physicians of Canada. Vol. 53, no. 3, pp 579–580. [Online]. Available:
<http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=1949098>, [Accessed: March 20, 2009].

[8] Richard E. Gregg, Sophia H. Zhou, James M. Lindauer, Eric D. Helfenbein, Karen K. Giuliano, "What is inside the electrocardiograph?" ,Journal of Electrocardiology. Vol 41, pp. 8-14. May 2007

[9] Doralice Ferriera, blood pressure measurement, "peaks for blood pressure estimation", January 2009.

[10] Dr. Neil Townsend. Medical Electronics. Michaelmas Term, 2001.

[11] "Definition of Sinus Arrhythmia", June 1999. [Online]. Available: <http://www.medterms.com/script/main/art.asp?articlekey=9745> [Accessed: march 25 2009]

[12] "Frequency-shift keying – Wikipedia, the free encyclopedia", April 2009. [Online]. Available: http://en.wikipedia.org/wiki/Frequency-shift_keying [Accessed: April 20 2009]

[13] "Electronic Filter –Wikipedia, the free encyclopaedia", April 2009. [Online]. Available: http://en.wikipedia.org/wiki/Electronic_filter [Accessed: April 20 2009]

Vita

Name: Kundan Thind

Date of Birth: February 1, 1987

Secondary Education: Orchard Park Secondary School, Stoney Creek(2005)

Post Secondary Education: Electrical and Biomedical engineering, McMaster
University