Design of Software for an Electrocardiogram Analyzer

by

Robert Tisma

Electrical Biomedical Engineering Project Report

Submitted in partial fulfillment of the
Degree of Bachelor of Engineering

McMaster University
Hamilton, ON
April 30, 2010

Abstract

With cardiovascular disease (CVD) being one of the top causes of death in Canada, a solution must be developed to help treat patients in the early stages of CVD. In addition, with increasing patient waiting times, the demand for a system that can diagnose heart disease is also increasing. The ECG Analyzer provides a method of monitoring the heart and diagnosing heart disease in real-time using pattern recognition. The analyzer incorporates a feature extraction component and a classification component. Feature extraction uses QRS detection to acquire disease characterizing information for subsequent use in the classification component. The classification component is achieved through the use of support vector machines (SVM). Results demonstrated that the QRS detection algorithm and SVM classification performed reasonably well with classification error rates as low as 19.09%. Different kernel functions were used and the polynomial function was found to be the best option. At the conclusion of testing, it was noted that classification accuracy could be increased by using a higher dimensional feature vector in conjunction with feature selective classification.

Acknowledgements

I would like to thank our course supervisor Dr. Doyle, for his guidance and his support throughout the duration of the course. As well, I would like to thank Dr. Patriciu and Tyler Ackland for their providing last minute help. I would also like to thank TIS Technologies for providing last minute components. Also thanks to my family for supporting me throughout the stress and development of the project. A special thanks must be extended to Mile Stanojčić, for his support and guidance throughout the project. Last but not least, I would like to thank my good friend and partner, Mike Chrapala, for collaborating with me in creating this wonderful product.

Thank you.

Table of contents

Abstract ..... **ii**

Acknowledgments ..... **iii**

Contents ..... **iv**

List of Tables ..... **vi**

List of Figures ..... **vii**

Nomenclature ..... **viii**

1. Introduction **1**

1.1 Background 1

1.2 Objectives 5

1.3 Methodology 6

1.4 Scope 6

2. Literature Review **8**

2.1 Standard Clinical ECG Features 8

2.2 Feature Extraction 9

2.3 QRS Detection 10

2.4 Support Vector Machines 12

3. Statement of Problem **14**

3.1 Overview 14

3.2 Feature Extraction 15

3.3 Microcontroller Interface 16

3.4 Classification 16

4. Experimental and Design Procedures **18**

4.1 Introduction 18

4.2 Design of QRS Detector 19

4.3 Design of Support Vector Machine 28

4.5 Experimental Procedures 35

5. Results and Discussion **36**

5.1 Results of QRS Detector 36
i) Filtering 36
ii) Differentiation 40
iii) Nonlinear Transformation: Squaring 42
iv) Moving Window Integrator (MWI) 43
5.2 Results of Support Vector Machine 47
5.3 Overall Performance 53
5. Conclusion 54
6.1 Conclusion of the Design Procedure 54
6.2 Future Recommendations 55
Appendix A – Computer Programs Used 56
References 57
Vitae 59

List of Tables

2.1	Typical ECG features and their normal values at a heart rate of 60 bpm	10
5.1	Statistical Results Using Kernel Functions	50
5.2	Statistical Results From The Use Of Various Kernel Functions Including Outliers.	50

List of Figures

1.1	The Anatomy of the Heart [3]	2
1.2	The Conduction System of the Heart [4]	4
1.3	Key Features of an ECG Signal [6]	4
2.1	The QRS detection algorithm	10
2.2	2D example using a 1D classifying hyperplane	12
4.1	Amplitude Response of the digital BPF.	22
4.2	Relationship between the QRS complex and the MWI	23
4.3	Relationship between the QRS complex, the MWI (with noise) and the adaptive thresholds	25
4.4	Linear Separating Hyperplanes	30
4.5	SVM network map	33
4.6	Mapping data to a new feature space	34
5.1	Digital Band Pass Filter.	37
5.2	Spectral plots of BPF	38
5.3	PSD plots demonstrating the result of the digital band pass filter.	39
5.4	Plot of data analysis demonstrating the result of the derivative function	41
5.5	Plot of demonstrating the result of the squaring function	43
5.6	Plot of data analysis demonstrating the result of the MWI.	44
5.7	3D Mesh plot of varying MWI window size (τ samples)	45
5.9	Final plots of ECG data analyzed with the QRS detector	46
5.10	QRS detection on Testing ECG data	48
5.11	3D scatter plots representing the extracted features and testing data (d).	49
5.12	A plot of the trained and classified data for various kernel functions	52

Nomenclature

ADC - Analog-to-Digital Converter

AV – Atrioventricular

BPF – Band Pass Filter

ECG - Electrocardiogram

HPF – High Pass Filter

LPF – Low Pass Filter

MWI - Moving Window Integrator

RBF – (Gaussian) Radial Basis Function

SA - Sinoartrial

SVC – Support Vector Classification

SVM – Support Vector Machine

SVR – Support Vector Regression

Chapter 1

Introduction

1.1 Background

Cardiovascular disease (CVD) such as heart disease and stroke is a serious health concern that takes the life of one Canadian every seven minutes [1]. More specifically, in 2005 it accounted for 31% of all deaths in Canada and has more deaths than any other disease [1]. The high prevalence of these devastating conditions supports the significance of early diagnosis in order to identify and aid at-risk individuals. Currently, there are various medical instruments used to record and monitor cardiovascular health in patients. However, most of these require a qualified professional to analyze and interpret the results in order to diagnose the problem and therefore prescribe the appropriate treatment. Possible obstacles such as increasing demand and lack of properly trained medical personnel, patients at risk of heart disease or stroke may not be aware of their condition nor have the resources to purchase expensive medical equipment in an attempt to monitor their own health. Thus, problems such as these must be addressed in order for emerging medical technologies to improve patient care.

One of the easiest and inexpensive methods of monitoring the heart for CVD is the use of an Electrocardiogram (ECG). ECG machines provide a means of quantifying electrical activity of the cardiovascular system, which is represented graphically for specialists to analyze. When an individual complains of chest pain, a series of steps are executed to determine the extent and severity of the heart problem. The first step is a general classification of the patient's symptoms as either a heart or non-heart related issue [2]. Upon classification as a heart-related complication, the second step is for the patient to undergo laboratory testing for specific blood factors associated with CVD [2]. Furthermore, along with laboratory tests, another option is to undergo diagnostic imaging such as echocardiogram and perfusion scintigraphy [2]. However, as previously mentioned these forms of medical assessment may result in prolonged waiting periods

and again ultimately require professional analysis in order to determine a conclusive diagnosis [2]. Therefore, in order to proactively diagnose cardiovascular complications an ECG test should be performed prior to secondary extensive and invasive procedures. In summary, ECG's can be considered the principle method of monitoring the heart in real-time and are able to provide medical specialists with quick and reliable information regarding a patient's condition.

The waveform of an ECG recording is able to provide vital information regarding an individual's heart condition by observing specific features in combination with the known relationship between cardiac contraction and relaxation and electrical activity [10]. The heart is a four chambered pump that sends deoxygenated blood to the lungs, while simultaneously distributing oxygenated blood to the rest of the body. As illustrated in Figure 1.1, the heart is divided so that two chambers are implicated in the deoxygenation process and the remainder are involved in the oxygenation along with all the corresponding major arteries and veins.

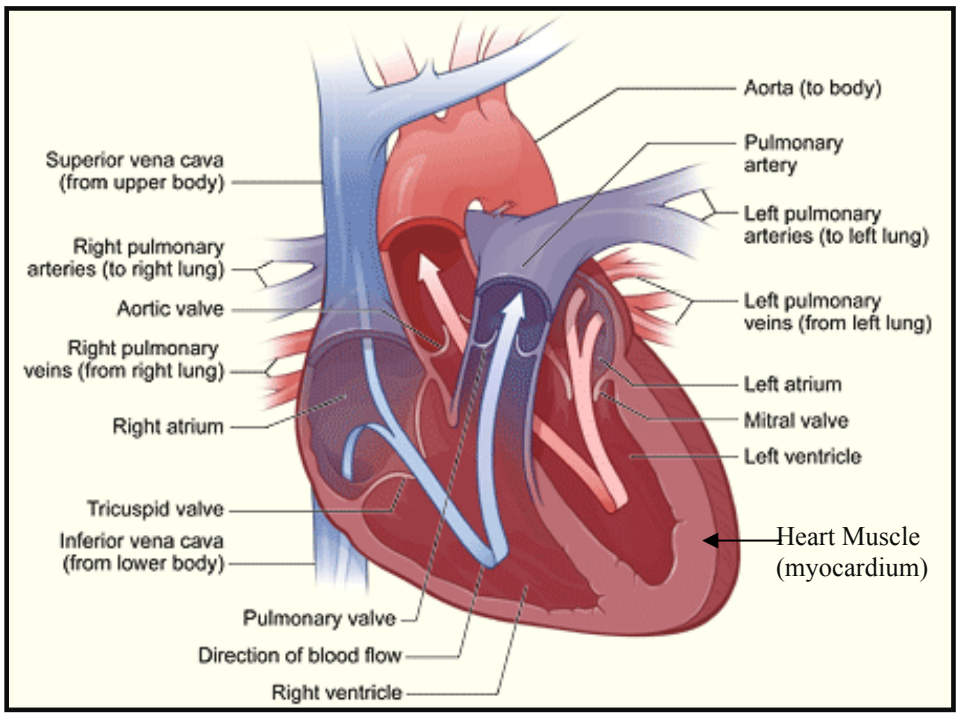


Figure 1.1: The Anatomy of the Heart [3]

The cardiac cycle functions to distribute blood throughout the body, can be best described by two main events: atrial systole and ventricular systole. In atrial systole, the

left and right artia contract forcing blood into the left and right ventricles, respectively [4]. After blood is pumped into the ventricles, the atria undergo a diastolic phase where they would relax. In ventricular systole, the left and right ventricles contract causing the respective closing of the mitral and tricuspid valves in order for pressure to accumulate in the ventricles [4]. Upon accumulation of atrial pressure to its threshold, the aortic valve opens resulting in blood flow to the aorta. Similar events occur within the right ventricle with the pulmonary valve opening and resulting in blood flow to the pulmonary artery. This cyclical pattern continues by coupling ventricular relaxation with atrial filling of blood, in preparation for the following cycle.

The cardiac cycle is a process that is governed by the electrical activity of the myocardium, also referred to as the conduction system. The conduction system begins with the sinoartrial (SA) node (refer to Figure 1.2). The SA node's cellular architecture is composed of cells that do not have a stable resting potential and as a result, they repeatedly depolarize to a specific threshold value in order to trigger an action potential [4]. Therefore, the SA node is considered the heart's pacemaker and generates an action potential that triggers a series of contractions throughout the heart. The action potential then propagates throughout both the left and right aria, producing a subsequent contraction in the atrial muscle fibers [4]. Next, the action potential reaches the atrioventricular (AV) node and propagates to the AV bundle or Bundle of His [4]. The Bundle of His is the only site where action potentials initiated by the SA node are able to conduct from the artia to the ventricles [4]. Lastly, the action potential eventually spreads to the ventricular myocardium, resulting in ventricular systole [4]. In conclusion, the conduction system regulates the periodicity of cardiac contractions and relaxations, thus serving as a vital element to diagnosing numerous CVD.

In order to assess the functioning of the cardiac cycle and conduction, an ECG test is conducted to generate quantified readings translating to the electrical activity of the heart. As previously mentioned, the ECG contains information regarding events that have occurred in the heart by analyzing the electrical potentials in respect to time [4]. As represented in Figure 1.3, this typical ECG waveform consists of 5 main features: the P, Q, R, S and T waves [4]. The ECG signal begins with the depolarization of atrial contractile fibers (atrial systole), which creates a characteristic

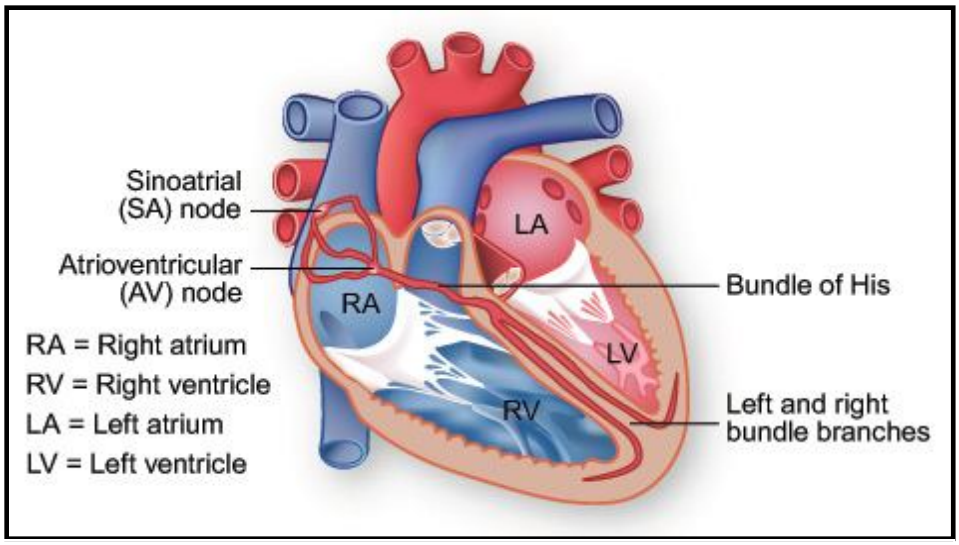


Figure 1.2: The Conduction System of the Heart [4]

wave known as the P wave [4]. Once the P wave has been generated, a slight delay takes place as the action potential from the SA node travels to the Bundle of His and consequently ventricular myocardium [4]. The ventricular contractile fibers then begin to depolarize (ventricular systole), producing the Q, R and S waves that form the QRS

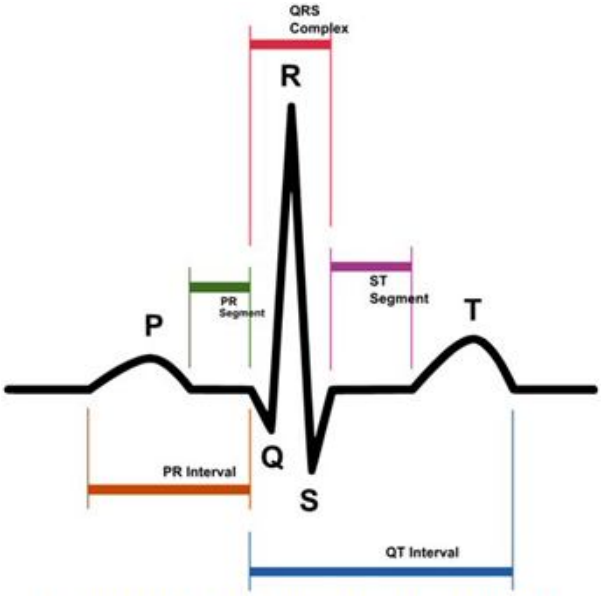


Figure 1.3: Key features of an ECG signal

complex [4]. During the time of ventricular systole, the atria repolarize but are difficult to detect on an ECG signal due to the QRS complex masking it [4]. The repolarization of ventricular contractile fibers then takes place to produce the characteristic T wave shape,

ultimately followed by relaxation [4]. Taking into consideration that the ECG signal describes electrical activity and the extent of myocardial activity of the heart, thus this non-invasive test is able to serve as a representation of the current state of a given heart.

In order for an ECG signal to be useful, an interpretative mechanism is required in order to describe the events occurring in the heart. To date, a diagnosis of a heart condition is made primarily by trained professionals based on the interpretation of ECG patterns [7]. One of the potential problems with ECG interpretation is the chance of human error [7]. Some factors contributing to human error include: fatigue (paramedics), habituation, and psychological factors (emotion, reaction to stressful situations, etc) [7]. A promising solution to many of these factors is the implementation of automated ECG analyzing machines. ECG analyzers interpret data by first extracting features, which are then used to classify patterns synonymous with cardiac defects [10]. Using pattern recognition, ECG analyzing machines are in theory immune to the effect of human error, however are limited to the technologies and algorithms provided [10]. Thus, ECG analyzers can essentially replace human interpretation and therefore ease, if not eliminate, the responsibility of interpretation by medical professionals. This will further increase the probability of detecting heart conditions much earlier.

1.2 Objectives

The objectives of this project are to provide diagnosis of heart conditions using data acquired from an ECG signal in real-time. By doing so, an individual will be able to not only monitor their heart during cardiac cycles, but also have a portable device that is efficient and affordable. One of the advantages to having such a device is that people can monitor and diagnose any heart related problem in the comfort of their own home. Most ECG analyzers employed in the healthcare industry are very costly and contain additional elements that for the purpose of this project are not useful. This project contains two main components: hardware and software. The hardware component consists of an ECG machine that measures a single-lead ECG signal from an individual, which was the responsibility of Michael Chrapala. In terms of the software components, the PIC24HJ32GP202 microcontroller will be used to convert analog ECG signals into digital form. These signals will then undergo feature extraction in order to be classified

using a Support Vector Machine (SVM), which is equipped with a library of ECG signals from heart condition patients. Robert Tisma is responsible for this component. At the conclusion of the present project, an affordable, compact, and effective ECG analyzer will have been successfully created to measure and classify the ECG signal of an individual having some form of heart disease.

1.3 Methodology

The software component is dissected into the main components that need to be resolved. The first problem is the use of the PIC24HJ32GP202. This microcontroller has the ability to process information at 40 MIPS as well contains an extensive amount of peripheral devices [8]. One of the main features of the PIC24 is the onboard analog-to-digital converter (ADC). The ADC can be configured in a variety of operating modes which makes it very customizable for almost any application. It also has digital input and output ports, which can be useful when trying to display a message to the user. For the project, the ADC will be used to sample the analog output of the ECG machine and then digital processing will take place on the microcontroller.

The processing involves the extraction of features from the digital ECG signal. Since the processing is a real-time implementation, the time needed to extract features and classify them, must take less time than the sample period. This is a major concern as the code must be efficient for the processing to keep up with the ADC. When the microcontroller has classified the input data, an external LED would be lit which corresponds to one of the heart diseases that trained the SVM. The library that will be used to train the SVM will be available ECG recordings of individuals with heart conditions.

1.4 Scope

When processing biological signal, there must be great awareness of the variability in data. Since biological signals, such as the ECG, vary in shape and size from beat to beat, they require special consideration in terms of processing. The ECG signal obtained from the hardware component cannot be completely filtered of all its noise, thus can present

some unpredictable results in the software component. The algorithms that will be used are geared towards variable biological data, but still a degree of variability exists which could result in false positives or false negatives. The software component will work under the assumption that all human ECG signal have a characteristic wave form, such as Figure 1.3. Although the software will be able to differentiate between healthy and diseased ECG signals in general, there will always exist data that does not follow certain assumption. The main goal is to maximize the effectiveness of classification following the assumptions that ECG signals are variable but consistent in shape and size.

As well, the effectiveness of classification is primarily based on the accuracy of the extracted ECG features. Since no ECG signal is ideal, there is a probability of false positives and false negatives. To overcome problems such as this, extended data sets and averages will help to alleviate errors as well as random events. Therefore a threshold or tolerance level must be established in order to properly process and classify ECG data.

Chapter 2

Literature Review

The ECG analyzers that are currently available are essentially designed for applications in the veterinary [9] and healthcare system [7]. However, due to the strong dependence on pattern recognition, ECG analyzers are not prominent in the field of medicine due to their poor reliability [7]. In order for an ECG analyzer to classify data, it must be equipped with data from individuals who currently have a known heart condition. In other words, the classification of a system is reflected upon the information it is programmed with and since only a limited amount of data can be used to train the system, it is easily rendered unreliable by users [7]. Additionally, due to the complex and variable nature of the ECG signal, research has been conducted to help create efficient methods of extracting data from test readouts, along with better methods of training learning machines [10].

2.1 Standard Clinical ECG Features

When designing a system that analyzes ECG signals, it is important to have an understanding of the relationship between signal features and both the anatomical and physiological components of the heart. As previously mentioned, an ECG signal is composed of five main features: the P, Q, R, S and T wave (refer to Figure 1.3). In order to identify particular segments, a reference point for each beat must be established. This is called the fiducial point and is normally represented as the peak position of the R wave [10]. A key factor of any ECG signal is the RR interval, which is the time between two adjacent R waves corresponding to the instantaneous heart rate [10]. The instantaneous heart rate can be calculated using Equation 2.1, where HR_n represents the n^{th} instantaneous heart rate [10].

$$HR_n = \frac{60}{RR_n} \times \frac{beats}{min} \qquad \textit{(Equation 4.1)}$$

Due to the stochastic nature of the ECG signal, the RR interval varies on a beat-to-beat basis. When the heart rate is of importance to processing, the average heart rate is normally calculated to help mask oscillation in the RR interval [10].

Furthermore, ECG morphology is also correlated with the current RR interval. A common phenomenon that occurs during respiration is the Bainbridge reflex [10]. The Bainbridge reflex is characterized by the periodical oscillation of RR_n, as a result of the expansion and contraction of the lungs [10]. The periodic motion of the lungs results in changes to the intrathoracic pressure, which in turn affects the cardiac filling volume [10]. During inspiration, the thoracic pressure decreases causing an increase in venous return [10]. This results in right atrium expansion and increases the local heart rate [10]. On the other hand, respiration results in the reverse process and facilitates a decrease in local heart rate [10]. Thus, the ECG signal obtained during thoracic expansion and contraction results in changes to the amplitude and timing of features.

Some of the other aspects that must be taken into account are the different noise sources in an ECG signal. Common noise artifacts are: power line interference (60 Hz mains noise), loss of contact between the electrode and the skin (contact noise), movement of electrodes from the skin contact area (patient-electrode motion artifacts), electomyographic noise (muscle motion artifact), artifacts contributed by signal processing devices (device noise), quantization noise and aliasing as a result of analog to digital conversion, and baseline drift [10]. These types of artifacts affect the overall ECG signal and are capable of creating inconsistencies during data analysis.

2.2 Feature Extraction

The analysis of ECG peak values requires referencing to a specific level called the isoelectric level. The isoelectric level is one of the most important features of the ECG signal [10]. ECG signals are prone to some baseline drift where the vertical translation in the waveform takes place. When the waveform translates vertically, the isoelectric level

follows, and when measurements of the peak values are taken they are adjusted to present the true value [10]. If peaks were simply measured from the 0V line, incorrect patterns and correlations would be found leading to false results.

Along with the other main features of an ECG signal previously described, the QRS complex width, local heart rate, R wave peak, P wave peak and T wave peak are all fundamental. To find the value of these features, a QRS detection algorithm must first be used to detect the fiducial point. Once this point is established, the isoelectric level can be determined. Upon doing so, all other features can then be measured using specific timing thresholds. Some examples of these typical values are shown in Table 2.2.

Table 2.1: Typical ECG features and their normal values at a heart rate of 60 bpm[1]

Feature	Normal Value	Normal Limit
QRS Width	100 ms	±20 ms
P amplitude	0.15 mV	±0.05 mV
QRS height	1.5 mV	±0.5 mV
T amplitude	0.3 mV	±0.2 mV

[1] Adapted from [10].

2.3 QRS Detection

The QRS detection is the initial component to any feature extraction step. The main

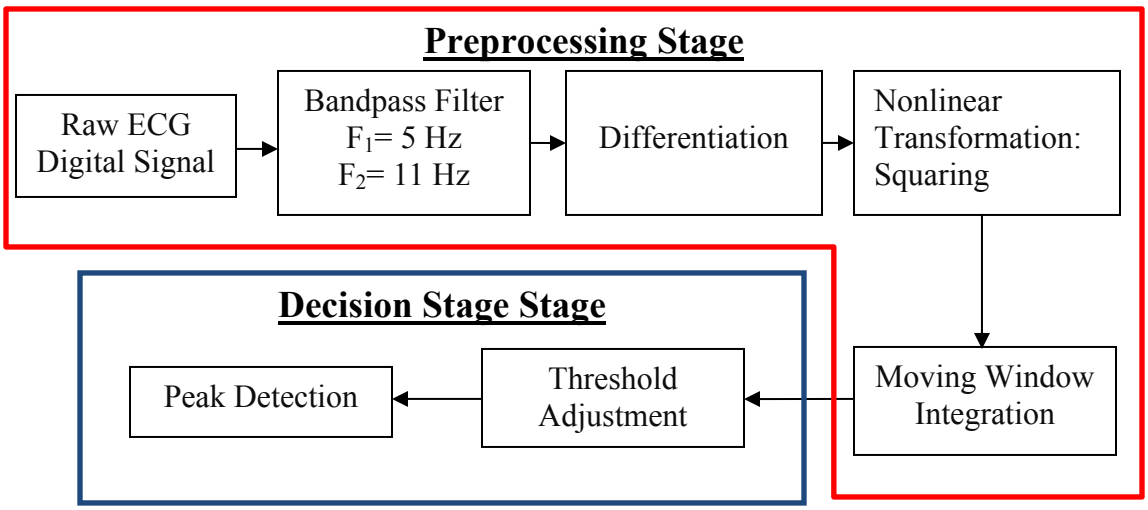


Figure 2.1: The QRS detection algorithm proposed by [11], consisting of a Preprocessing stage, and Decision stage.

purpose is to determine the location of the QRS complex in order for features to be uniquely described on a beat-to-beat basis. Once the QRS complex has been obtained, other features may be extracted using the QRS complex as a fiducial mark for every heartbeat. The algorithm presented by Pan and Tompkins [11], uses a series of preprocessing stages followed by a decision stage. In the preprocessing stage, the signal is band pass filtered (BPF) by cascading a low pass filter (LPF) and high pass filter (HPF) together. The filter contains frequency cutoffs of about 5 Hz and 11 Hz, which maximizes the QRS energy needed for the detection stage. The inherent noise due to power line interference, muscle noise, baseline wander and T wave interference are all reduced with the band pass filter [11]. Next is the differentiation stage where the filtered ECG signal undergoes differentiation. The purpose of this component is to obtain information for the next stage. The nonlinear transformation component is a function that squares the output of the differentiation block to make them positive and emphasize high slope values. The next component is the Moving Window Integrator (MWI), which calculates an average value for a specific number of previous samples from the nonlinear transformation component. At the conclusion of this stage, the raw ECG data will be represented in a time-averaged form that will aid in the decision process of the next stage (refer to Figure 2.1, [11]). Lastly, the main components of the decision stage are the threshold adjustment and the peak detection. The threshold adjustment component consists of adaptive thresholds that are automatically adjusted to float over the noise. There are two sets of thresholds: one for the MWI and the other for the filtered ECG signal that are used to distinguish true and false QRS detection in the subsequent component. The peak detection component obtains information from the MWI and compares it with the current thresholds. Essentially, if the input signal is greater than the upper threshold of both sets of thresholds within a certain time period, then a QRS candidate has been detected. As well, if a QRS peak has not been identified in a certain time frame, then a search-back routine is performed to find the missed peak [11]. The decision component also incorporates an average and instantaneous heart rate calculation that helps reduce false detection [11]. Furthermore, the T wave may also be detected along with the QRS complex by identifying peaks between two successive QRS complexes but below a certain threshold and time period. This algorithm allows for

unique identification of QRS complexes, thus providing a fiducial mark for further analysis.

2.4 Support Vector Machines

The learning regime that will be used in the project is called Support Vector Machines (SVM). This technique is a linear classifier that uses a *p*-dimensional feature vector and separates the classes using a *p-1* dimensional hyperplane [12]. An illustration is provided in Figure 2.2.

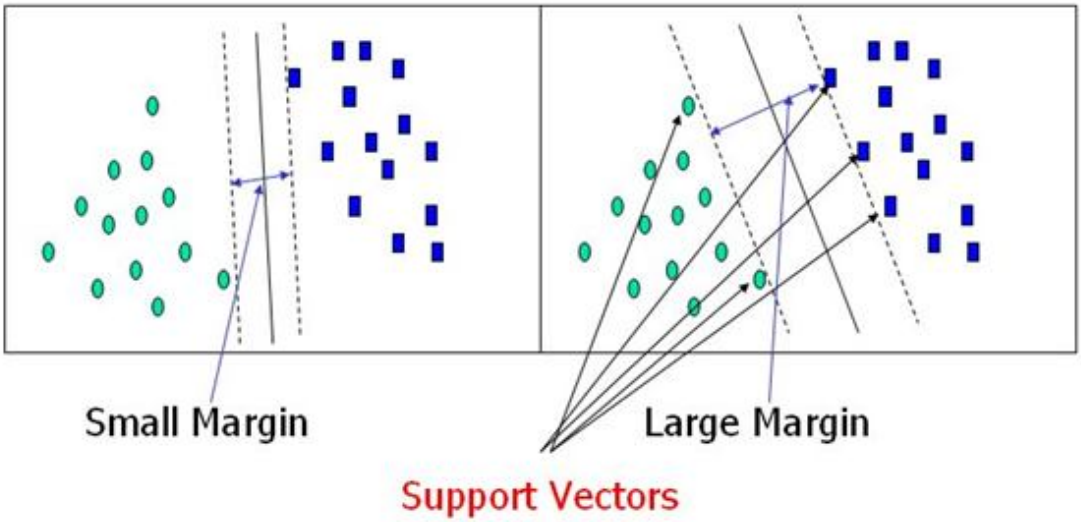


Figure 2.2: 2-dimentional example of data being classified with a 1-dimensional hyperplane [12].

The whole purpose is to create a classification by separating data points with a maximal margin using a hyperplane. In the training phase, the margin separating the two classes continually adjusts with increasing data points until a definitive pattern is established [12]. In the execution phase, data points are classified based on their position relative to the defined hyperplane [12].

The SVM that will be incorporated in this project is a library of SVM functions called LIBSVM [13]. LIBSVM is an easy to use tool that is very well documented and is available for more than one programming language [13]. The library consists of the following SVM implementations: C-Support Vector Classification (C-SVC), υ-Support

Vector Classification (υ-SVC), ε-Support Vector Regression (ε -SVR) and υ-Support Vector Regression (υ-SVR). C-SVC and υ-SVC use a set of training data called a training vector with corresponding class labels in order to create an optimal separating hyperplane [13]. The data points that define the maximal width of the margin and ultimately the optimal hyperplane, are labeled as support vectors [13]. The training data is used to create Lagrangian coefficients that define the hyperplane, which are then used in the execution phase [13]. Once an SVM model has been created, the system is then ready to classify input data into appropriate classes [13]. C-SVC has a restriction on the range of values available for the Lagrangian coefficients, while υ-SVC uses a parameter, υ, which has an upper bound on the fraction of training errors and a lower bound on the fraction of support vectors [13]. ε-SVR is similar to SVC except that a function or model is created with a maximum of ε deviation from obtained targets of training data and at the same time be as flat as possible [13]. The last SVM implementation, υ-SVR, uses the parameter υ to control the number of support vectors in the regression [13].

Chapter 3

Statement of Problem

In this chapter, a greater understanding of the issues concerning the design and development of an ECG analyzer as an improved alternative to current diagnostic procedures will be presented. More specifically, it will describe the problems regarding feature extraction, microcontroller interface and classification stages. When considering the feature extraction step, the expectations and minimum requirements will be addressed followed by the respective reasoning. The problems regarding the microcontroller interface will discuss what is expected of the microcontroller component and the challenges that must be overcome. Finally, the classification stage will focus on determining the appropriate categorization and translation of data in order to make accurate diagnosis. Upon reading these sections, there will be a strong conception of the underlying theory that is required to grasp the methodology and assumptions associated with it, along with the reasoning behind the implementation of an ECG analyzer.

3.1 Overview

The ECG Analyzer is portable device that uses a microcontroller to analyze patient ECG data, and provide a probability based diagnosis based on CVD ECG data stored in a library. The analyzer extracts features from an ECG signal which are sent to the SVM, where a diagnosis is made based on probability. The feature extraction stage uses a QRS detection algorithm to extract data from ECG signals. These features are then input to an SVM, which separates data by finding the optimal separating hyperplane. Overall, the expectations at the conclusion of the project are that QRS detection operates with minimal error, and that a diagnosis is made with the smallest error using SVM classification.

3.2 Feature Extraction

The feature extraction step is the method of acquiring specific useful information regarding an ECG signal. The objective of this step is not only to obtain data that define CVD, but to implement the procedure in an efficient and real-time manner. The goal of any ECG analyzer is online analysis, and therefore efficient implementation is required to ensure patient data is quick and accessible. Therefore the idea of this component is to extract features from patient ECG data that define patterns associated with heart disease using an efficient real-time implementation.

With the use of QRS detection, the aforementioned problems can be addressed given specific conditions and limits. The detection essentially consists of processing raw ECG data into a form that represents features associated with the QRS complex more easily. In addition, the assumptions are that human ECG data carry the same characterizing elements that are only distinctive in magnitude. This implies that CVD ECG data can be characterized by the same elements that define normal healthy ECG data, and have unique values. Features such as QRS width, R wave height, P wave height, P wave width, T wave height, T wave width and local heart rate are all examples of what QRS detection can output.

For the purpose of this project, the features of interest are the QRS width, the R wave peak value, and the local heart rate. These 3 features were chosen as the most relevant since they can be easily detected and be used to classify many diseases [10]. The assumption was that the data extracted by the QRS detector would be sufficient to provide insight to patterns in ECG signals. One of the challenges that must be overcome, are extracting information in the presence of variations in ECG morphology. With diseases such as atrial flutter which differ considerably in morphology in comparison to normal ECGs, special consideration must be made in the implementation of the detector [10]. As side from attribute related differences, QRS detection also faces the challenge of noise. Due to the stochastic nature of the ECG signal, procedures must be developed to avoid the effects of noise contaminated signals. The feature extraction step is an important aspect to the classification of heart-disease related ECG signals since it depends on the quality and speed of the obtained information.

3.3 Microcontroller Interface

The second component to the project is the microcontroller interface, which will execute the program for the ECG analyzer. The objective is to use the microcontroller to process the input data from the ECG machine. The microcontroller will convert the analog signal to a digital signal, where information can be easily extracted as well as output a diagnosis based on a percentage match between the input data and the stored library of CVD ECG data. The main challenge is implementing the processing elements in an efficient manner, as to minimize the time required for diagnosis and maximize the amount of available on-board memory. In addition, the ECG analyzer will be a portable device which provides the advantage of not only being used in medical centers, but in households also.

In the medical field, the speed by which a patient is diagnosed and emitted from the medical center is an important factor. With a many patients requiring medical attention, the hospital must be able to provide effective diagnosis and subsequent treatment in order decrease patient waiting times. With that being said, an ECG analyzer that can diagnosis a patient's heart condition much faster than a human, would be beneficial in reducing patient line-ups and concentrating medical attention to those of more severe symptoms. For this reason, the ECG analyzer must be programmed to run efficiently and reliably, in order to diagnose patient heart symptoms in a short time.

3.4 Classification

The classification stage represents the process of using the extracted features from QRS detection to match patterns found in a patient's ECG signal, to patterns associated with CVD. The system will be realized using a SVM to recognize patterns linked to heart disease. The SVM model will be trained using a library of heart disease related ECGs. Once the model is developed, a patient's ECG signal will be classified according to the percentage match with each disease in the library. By using LIBSVM, the pattern recognition task can be easily and efficiently incorporated onto the microcontroller, which supports the notion of a portable ECG analyzer.

In order to employ this component, the challenges involved must be taken into consideration. In this project, the SVM will use the extracted features from the QRS detector to classify and therefore diagnose input ECG data. In terms of errors in classification, the quality of QRS detection and limitations to SVM implementation are of great concern. One cause of misclassified data, can be attributed to errors in QRS detection. Since it provides the SVM with extracted features for classification, the errors from incorrectly detected features translates to incorrectly classified data. The other challenge is that errors in classification can result from inseparable data. Since the SVM attempts to find an optimal hyperplane that separates the CVD ECG data from patient data (which is stochastic in nature), errors are inevitably produced as a result of data being inseparable. Therefore, errors in classification are a product of incorrectly extracted features and inseparable data. Therefore the objective of the classification stage, is to choose the SVM model that provides the least error in diagnosing patient ECG data.

Chapter 4

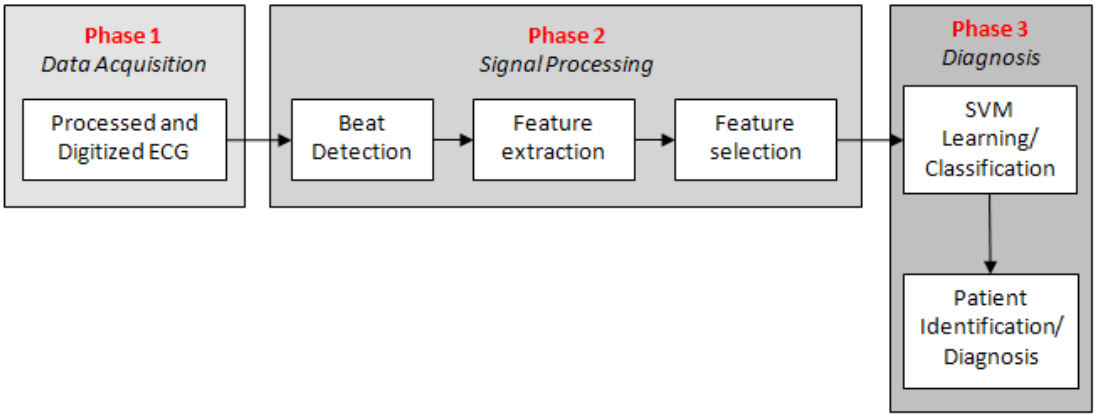
Experimental and Design Procedures

In this chapter, the design and experimental procedures will be discussed and ultimately address how the problems presented in Chapter 3 will be resolved. The experimental procedures presented will discuss the design of the QRS detector and the SVM. These will be used to classify ECG signals from at-risk individuals for CVD related patterns. As well, the steps involved in testing the systems ability to analyze and classify ECG data will be presented. Furthermore, the SVM component will be mathematically derived to provide a concrete understanding of the elements involved with classification. At the conclusion of this chapter, one should have a clear understanding of how the QRS detector and SVM were presently implemented and tested.

4.1 Introduction

In order to develop an understanding of the processes involved, an overview of the design and experimental procedure is required. The design of the project can be best described by understanding Schematic 4.1. The project is composed of three phases. The first phase describes the acquisition of ECG data from the hardware and analog-to-digital conversion. The ECG signal (hardware component) was obtained by the ADC in the laboratory and due to technical difficulties with the PIC24 microcontroller, eliminated the option of real-time processing. The information was then converted to a format compatible with MATLAB and used for all subsequent phases. Although the program used was constructed for real-time application, it was still able to operate offline or by using stored ECG data. Phase two is composed of the detection of the QRS complex, which is then outputted to the third phase. The last phase is better known as diagnosis and is where the information is classified. A problem that occurred during the design procedure was that the test data obtained from MIT/BIH arrhythmia database could not

be properly calibrated to fit the systems requirements. Also, taking into account that ECG data could not be obtained from individuals with heart conditions, therefore the design of the project was slightly modified. Instead of diagnosing patient ECG with a particular disease, the system was changed to output the identity of the patient (Phase 3), using their inputted ECG signal. The assumption was that if the system could distinguish the identity based on two sets of data, then the system would be able to diagnosis ECG data since the method of classification is based on the same underlying principles. In this chapter, Phase 2 and Phase 3 are discussed in detail followed by a discussion of the experimental procedures.



Schematic 4.1: Block diagram of project components

4.2 Design of QRS Detector

As previously mentioned, the QRS detector extracts information regarding the QRS complex of an ECG signal on a beat-to-beat basis. Referring to Figure 2.1, the preprocessing phase modifies the input signal into a form that can provide useful information about the current health of a patient's heart. The decision stage then creates an adaptive threshold that is used for peak detection and feature extraction. In this section, a peak is a local maximum observed when the signal changes direction whereas a feature is a common or characteristic component of a signal that can be used to identify underlying patterns [11].

i) Data acquisition and analog-to-digital conversion

The first step digitized the acquired ECG data and stored it into memory for further processing. Since all the data obtained from the ADC was stored on disk, the program was modified to obtain information from memory. Samples were taken one at a time, much like an ADC would. As time progresses, the system adds each sample to a buffer array that is capable of holding a short length of data. This short length is necessary due to memory on a microcontroller having to be conserved as much as possible. Based on the preprocessing components outlined in Figure 2.1, the length of the buffer was determined to have a minimum length of 32 samples. In addition, the buffer array was implemented using a circular array. The circular array avoided the need to shift data for incoming data. By doing so, this method simply overwrites the oldest sample in the buffer with the incoming sample. All other preprocessing components that required memory, implemented a circular array to avoid array shifting, thus increasing the speed of QRS detection, and reserving memory for other important components.

ii) Band Pass Filter

The second step was the design of filters. A band pass filter (BPF) was implemented to reduce the interference of muscular noise (60 Hz), baseline wander and T wave interference [11]. The BPF is the result of cascading a high pass filter (HPF) with a low pass filter (LPF). The transfer function of the second-order LPF is described in Equation 4.1.

$$H_{LPF}(z) = \frac{1}{36}\left(\frac{1 - z^{-6}}{1 - z^{-1}}\right)^2 \qquad \textbf{(Equation 4.1)}$$

In order to implement the LPF, the inverse Z-transform of Equation 4.1 must be done to obtain the difference equation (refer to Equation 4.2). By doing so, x[n] represents the input sample at time $t_n = nT_s$ with T_s being the sample period (T_s = 5ms) and y[n] represents the output of the filter.

$$y[n] = \frac{1}{36}(x[n] - 2x[n-6] + x[n-12]) + 2y[n-1] - y[n-2]$$

The LPF has a 3-dB cutoff frequency of about 11 Hz and produces a delay of 5 samples at the output [11]. In addition, the filters were cascaded together by providing the input

of the HPF with the output of the LPF [11]. The transfer function of the HPF is shown in Equation 4.3 followed by its difference equation in Equation 4.4 [11].

$$H_{HPF}(z) = \frac{-1 + 32z^{-16} + z^{-32}}{32(1 - z^{-1})} \qquad \textit{(Equation 4.3)}$$

$$y[n] = \frac{1}{32}(32x[n-16] - x[n] + x[n-32]) - y[n-1] \qquad \textit{(Equation 4.4)}$$

The HPF has a 3-dB cutoff frequency of about 5 Hz with a total delay of 16 samples [11]. In order to correct the delay caused by the HPF and LPF, every sample of the output is shifted by 21 samples to match the input and filtered signals. This was a critical step in the design procedure since the slightest misalignment would result in false timings of extracted features. Therefore, bearing in mind the objective of the present project was to operate in real-time, it is vital that delays in data must be accounted for and corrected.

The entire process requires the most recent 32 samples, and calculates the output with a maximum of 5 terms [11]. Since the sample rate of the ADC was 200 Hz and the calculation size was small, the filter would perform very well in real-time without requiring excessive computing power [11].

The amplitude response of the digital BPF is displayed in Figure 4.1. The filter is seen to attenuate lower frequencies (below 5 Hz) steadily to -60 dB, which is a result of the HPF and the high frequencies (above 11 Hz) with a sharp transition band. Therefore, the filter created by Pan and Tompkins [11] was used due to the filters amplitude response showing good performance.

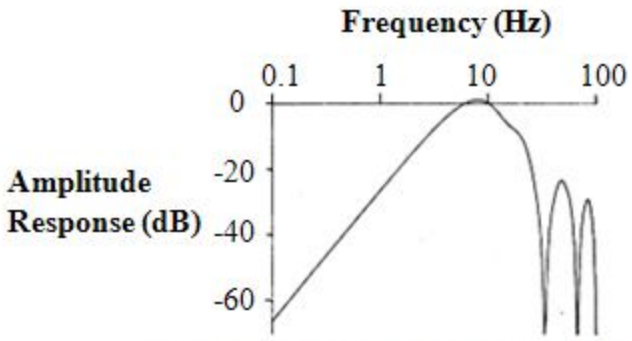


Figure 4.1: Amplitude Response of the digital BPF. Passband (3-dB) is 5-11 Hz [11].

iii) Derivative Function

After the raw input data was filtered, it then underwent differentiation. Equation 4.5a shows the difference equation that was used to differentiate the filtered data. Since the delay was one sample, output samples were shifted to lineup corresponding to the input waveform. Since derivatives provide information regarding local minima, local maxima and the rate of change (the slope), they were used to obtain critical information for recognizing patterns associated with the QRS complex.

$$y[n] = \frac{x[n] - x[n-1]}{T_s} \qquad \textit{(Equation 4.5a)}$$

iv) Non Linear Transformation: Squaring

In the fourth step, the output from the differentiator underwent a non-linear transformation; squaring. The purpose of the transformation was to emphasize large slopes from smaller ones [11]. Since frequencies can be described using slope values, this step emphasizes high frequency components from the signal [11]. This allows future steps to locate areas of increased activity, such as the QRS complex. The squaring function is shown in Equation 4.5b.

$$y[n] = (x[n])^2 \qquad \textit{(Equation 4.5b)}$$

v) MWI

In the final processing step, the MWI was calculated using the output from the squaring function. The purpose of the MWI was to extract feature information from the input ECG data along with the slope of the R wave, so that the time average of a specific window size can be determined [11]. This was calculated using Equation 4.6, where τ represents the window of integration size (MWI window size).

$$y[n] = \frac{1}{\tau}\sum_{k=0}^{\tau} x[n-k] \qquad \textit{(Equation 4.6)}$$

One specific relationship that the MWI has with an ECG signal is the width of the QRS complex. For every heartbeat, the QRS width corresponds to the rising edge of the MWI and is therefore calculated by finding the time duration of the rising edge [11]. Figure 4.2 illustrates the relationship between the QRS width and the MWI. Using this property, the width of the QRS complex was determined by finding the time duration from the local minimum prior to the R wave peak to the local maximum after the R wave peak. The use of the MWI allowed other features of the ECG signal to be extracted as well.

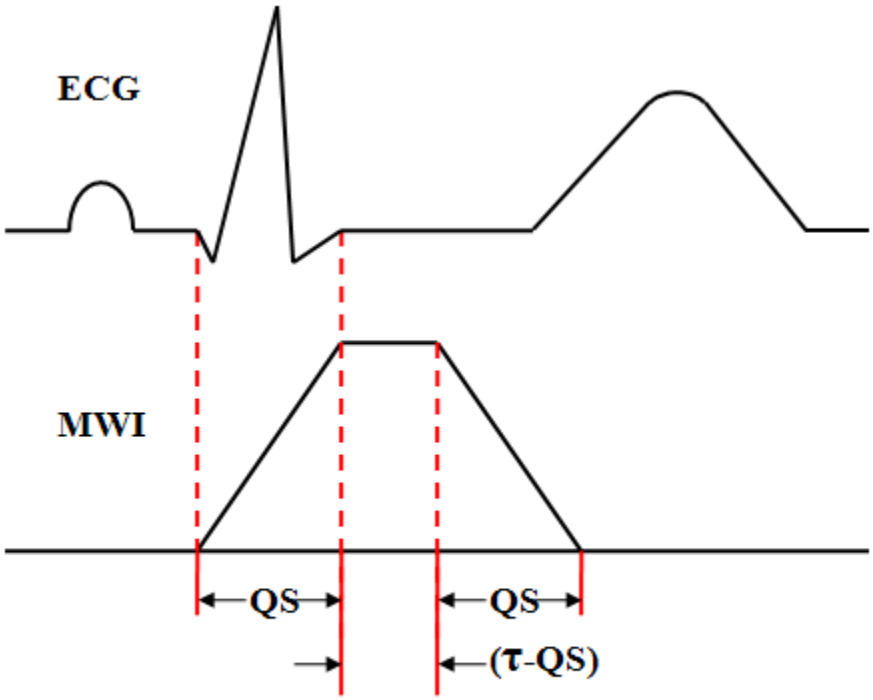


Figure 4.2: The ideal relationship between the QRS complex and the MWI [11]. QS: QRS width. τ:MWI window size.

vi) Threshold Adjustment

Once the MWI was calculated, the data continued into the decision stage. The first step was the threshold adjustment. The purpose of the thresholds was to provide specific limits for finding the R wave peak and QRS width. Using the MWI, two thresholds were set and were automatically adjusted with every input sample. The first threshold, called THR_NOISE was automatically adjusted to float above the noise [11]. Since the signal-to-noise ratio (SNR) was improved by the BPF, a low threshold was possible. Any MWI value that was found below THR_NOISE was considered to be a result of noise and therefore anything above the THR_NOISE was considered useful information. The second threshold, THR_SIGNAL was used for detecting QRS candidates. This threshold was also automatically updated to represent the decision boundary for QRS complexes and non-QRS complexes.

The thresholds were adjusted based on local minima and maxima found in the MWI waveform. Once the QRS detection was initiated, a brief diagnostic period of 1000 samples (5 seconds) was allowed for finding the largest peaks and lowest minimums. Once the diagnostic period ended, the program used the results from the diagnostic period as running estimates of signal and noise peaks, and continued for 30 to 50 seconds to establish THR_NOISE and THR_SIGNAL. Algorithm 4.1 describes the process involved in automatically adjusting the two thresholds [11].

```
if (CURRENTPK > THR_SIGNAL)

      SPKI = SPKI = 0.125*CURRENTPK + 0.875*SPKI;

elseif(CURRENTPK > THR_NOISE && CURRENTPK < THR_SIGNAL)

      NPKI = 0.125*CURRENTPK + 0.875*NPKI;

end

THR_SIGNAL = NPKI + 0.25*(SPKI - NPKI);
THR_NOISE = 0.5*THR_SIG;
```

(Algorithm 4.1)

In Algorithm 4.1, CURRENTPK represents the current peak detected in the MWI waveform, SPKI is the running estimate of the signal peak, and NPKI is the running estimate of the noise peak [11]. Using this algorithm, the threshold constantly adapted to

the MWI for every heartbeat, but by the time the diagnostic period passed, the thresholds already stabilized. The use of thresholds was an essential component as they provided the QRS detection step with appropriate information to extract features.

In the final step, the system used the calculated thresholds to determine whether or not a QRS complex occurred. The main goal from this step was to extract the start and end of the QRS complex as well as the position and height of the R wave peak, as depicted in Figure 4.3.

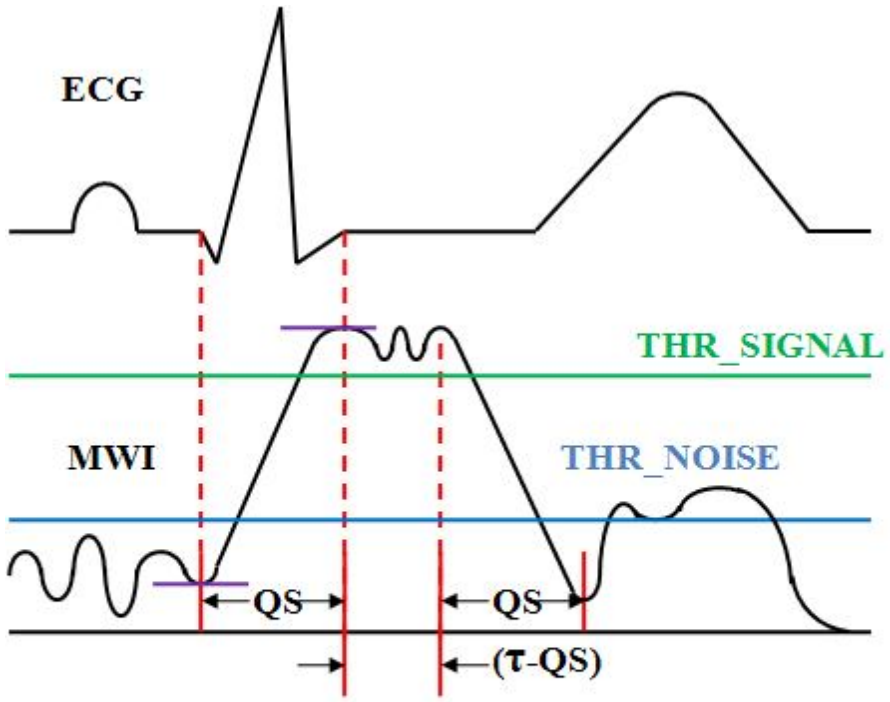


Figure 4.3: The actual relationship between the QRS complex, the MWI (with noise) and the adaptive thresholds THR_NOISE and THR_SIGNAL. QS: QRS width. τ:MWI window size.

To detect the start position of the QRS complex, local minima must be analyzed, which is shown by the first purple horizontal line in Figure 4.3. The local minimum or CURRENTMIN, was found by analyzing the change in MWI or MWI_D[n] slope from a negative value to a positive one. Since the start of the QRS complex corresponds to the beginning of the rising edge in the MWI waveform (refer to Figure 4.2), the CURRENTMIN closest to *n* where MWI[n] = THR_NOISE is considered the beginning of the QRS complex. In addition, the QRS start was assumed to represent the isoelectric level. Algorithm 4.2 describes in detail the code designed for detecting the start of the

QRS complex. In this algorithm, *window* represents the size of the circular buffer array, *count* represents the time (in samples) elapsed, *n* represents the current new position in the circular buffer, *beat_num* represents the current beat number, *MWI_D* represents the derivative of MWI[n], *CURRENT_R* represents the peak of the most recent R wave, *x[n]* represents the raw ECG data, *D[n]* represents the derivative of *x[n]* and *CURRENT_R_pos* represents the position of the most recent R wave peak.

```
n = mod(count,window);

if (MWI[n] <= THR_NOISE && MWI_D <=0)
      QRS_width = 1000*(QRS_end_pos-QRS_start_pos)/Fs;

      if (found_QRS_end == 1)
          QRS_width_memory[beat_num] =  QRS_width;
      end

      CURRENTMIN = MWI[n];
      CURRENTMIN_pos = n;
      QRS_start_pos = count;

      CURRENT_R = 0;
      R_found = 0;
      found_QRS_end = 0;

elseif (MWI[n] > THR_NOISE)
      if (x[n] >= CURRENT_R && D[n] >= 0)
            CURRENT_R = x[n];
            CURRENT_R_pos = count;
            CURRENT_R_index = n;
      else
            if (R_found == 1)
            %This assignment is so that R_found > 0,
            so that a peak does not need to be
            searched for any longer, until, MWI falls
            below THR_NOISE, in which case R_found
            resets to 0.

                  R_found = 2;

            elseif (R_found == 0)
                  R_found = 1;
            end
      end
```

(Algorithm 4.2)

The position of the R wave peak was found by determining the maximum value of x[n], when MWI[n] > THR_NOISE and when D[n] >=0. Since the R wave cannot occur when MWI[n] is below THR_NOISE, it was found by observing the derivative, D[n] of x[n], when MWI[n] >THR_NOISE. As well, since the R wave peak is characterized by a zero crossing in D[n], the maximum value of x[n] before D[n] = 0 (under the condition MWI[n] > THR_NOISE), was determined to be the R wave peak and position. Once the R wave peak was found, it was subtracted by the QRS start value to provide a comparable value relative to the isoelectric level.

With the R wave peak position and QRS start established, the last piece of information that was found was the QRS end using Algorithm 4.3. This was found by determining the peak in the MWI waveform, for MWI[n] > THR_SIGNAL and after a certain time interval. In Figure 4.3, this peak is shown as the purple horizontal line above

```
if (MWI_D >= 0 && MWI[n] >= CURRENTPK)

      CURRENTPK = MWI[n];

elseif (MWI_D < 0 && CURRENTPK > 0 )

      if (found_QRS_end == 0 && CURRENTPK > THR_SIGNAL
      && new_D >mod(CURRENT_R_index +
      time_thresh,window))

            SPKI = 0.125* CURRENTPK + 0.875*SPKI;
            QRS_end_pos = count;
            found_QRS_end = 1;

      elseif (CURRENTPK > THR_NOISE && CURRENTPK <
            THR_SIGNAL)

            NPKI = 0.125* CURRENTPK + 0.875*NPKI;

      end

      THR_SIGNAL = NPKI + 0.25*(SPKI - NPKI);
      THR_NOISE = 0.5*THR_SIG;

      CURRENTPK = 0; %reset currentpeak so that can
      find a new peak, even if lower than previous
end
```

(Algorithm 4.3)

THR_SIGNAL. In the Algorithm 4.3, *CURRENT_R_index* represents the R wave peak position in the circular buffer and *time_thresh* represents the extra time after

CURRENT_R_index in which peaks may be considered. This variable prevented peaks from being detected before the occurrence of the QRS end. With the QRS end position determined, the QRS width may be calculated by subtracting the QRS end by the QRS start.

With the R wave peak position and height detected, heart rate calculations were performed, using Equation 2.1. At the end of the analysis, the QRS detector outputted a feature vector described by Equation 4.7 under the conditions presented in Equations 4.8-4.10. In Equation 4.7, R_{Rpeak} is the R wave peak value in volts (V), R_{IHR} is the instantaneous heartrate in beats/min and R_{QW} is the QRS width in milliseconds (ms).

$$\boldsymbol{R} = \left[R_{Rpeak}, R_{IHR}, R_{QW}\right] \qquad \textbf{(Equation 4.7)}$$

$$R_{Rpeak} > 0.2\ V \qquad \textbf{(Equation 4.8)}$$

$$40\ beats/min < R_{IHR} < 200\ beats/min \qquad \textbf{(Equation 4.9)}$$

$$80\ ms < R_{QW} < 200\ ms \qquad \textbf{(Equation 4.10)}$$

As mentioned above, patient identity and therefore heart disease, can be characterized by these features extracted from a particular ECG signal and will therefore be used in Phase 3 of Schematic 4.1 to classify the data.

4.3 Design of Support Vector Machine

In the final phase of the project, the extracted features from Phase 2 were used to build a SVM model. The goal of SVM was to use available examples to produce a model or function that can classify unseen examples with little error [16]. As described in Chapter 2.4, a *(p-1)*-dimensional hyperplane is used to separate 2 *p*-dimensional classes, by finding its optimal position and orientation that maximizes the distance to the nearest data point of each class (refer to Figure 2.2) [16]. In order to understand the classification system that was used, a detailed derivation and explanation of SVM is necessary.

In order to generate a model, an SVM must first be trained using data with known labels. The idea is to separate a feature vector; D, with data; $\boldsymbol{x}$, and class or label; y, as shown in Equation 4.11a [16].

$$D = \{(\boldsymbol{x}_1, y_1), (\boldsymbol{x}_2, y_2), (\boldsymbol{x}_3, y_3), \dots, (\boldsymbol{x}_m, y_m)\},\ x \in \mathbb{R}^p, y \in \{-1, +1\} \quad \textbf{(Equation 4.11a)}$$

In the above equation, where m is the number of training samples in the data set, p is the number of dimensions, y is either positive or negative and corresponds to the positive or negative class [16]. The two classes are then separated with the hyperplane defined in Equation 4.11b, where $\boldsymbol{w}$ is normal to the hyperplane and $|b|/\|\boldsymbol{w}\|$ is the perpendicular distance to the origin [16][17].

$$\langle \boldsymbol{w}, \boldsymbol{x} \rangle + b = 0 \quad \textbf{(Equation 4.11b)}$$

As well $\langle \boldsymbol{w}, \boldsymbol{x} \rangle$ represents the dot product of $\boldsymbol{w}$ with $\boldsymbol{x}$ [16]. In order to find the optimal hyperplane with the largest margin, training data from the positive class (y = +1) must satisfy the inequality in Equation 4.12 and the negative class (y = -1) must satisfy Equation 4.13, where $\boldsymbol{x}_i$ represents the i^{th} input training data sample [17].

$$\langle \boldsymbol{w}, \boldsymbol{x}_i \rangle + b \geq +1, \qquad y_i = +1 \quad \textbf{(Equation 4.12)}$$

$$\langle \boldsymbol{w}, \boldsymbol{x}_i \rangle + b \leq -1, \qquad y_i = -1 \quad \textbf{(Equation 4.13)}$$

Equations 4.12 and 4.13 can be combined to form Equation 4.14 [17].

$$y_i(\langle \boldsymbol{w}, \boldsymbol{x}_i \rangle + b) - 1 \geq 0 \quad \textbf{(Equation 4.14)}$$

Thus, data points that satisfy the equality of the Equation 4.12 (positive class), can be described by the hyperplane H_1 in Equation 4.15, with the perpendicular distance from the origin,$|1 - b|/\|\boldsymbol{w}\|$ [17]. Similarly, the hyperplane H_2 in Equation 4.16, can be described by the equality of Equation 4.13 (negative class), with perpendicular distance from the origin,$|-1 - b|/\|\boldsymbol{w}\|$ [17]. Therefore, the distance from H_0 to H_1, d_1, is equivalent to the distance from H_0 to H_2, d_2, as shown in Equation 4.17 [17]. The purpose is to maximize the margin, defined by Equation 4.18, by determining a hyperplane H_0 that minimizes $\Psi(\boldsymbol{w})$ in Equation 4.19 [16]. Figure 4.3 illustrates the optimization problem where solid circles refer to the negative class, and hollow circles refer to the positive class [17]. It can be seen in Figure 4.4, that the support vectors (circled) define the optimal separating hyperplane, and that if the non-circled data point were removed, there would be no change.

$$H_1: \langle \boldsymbol{w}, \boldsymbol{x}_i \rangle + b = 1 \qquad \textit{(Equation 4.15)}$$

$$H_2: \langle \boldsymbol{w}, \boldsymbol{x}_i \rangle + b = -1 \qquad \textit{(Equation 4.16)}$$

$$d_2 = d_1 = \frac{1}{\|\boldsymbol{w}\|} \qquad \textit{(Equation 4.17)}$$

$$d_{12} = d_1 + d_2 = \frac{2}{\|\boldsymbol{w}\|} \qquad \textit{(Equation 4.18)}$$

$$\Psi(\mathbf{w}) = \frac{1}{2}\|\boldsymbol{w}\|^2 \qquad \textit{(Equation 4.19)}$$

In order to minimize Equation 4.19 under the constraints of Equation 4.14, the Lagrangian, shown in Equation 4.20, must be minimized with respect to $\boldsymbol{w}$, b and maximized with respect to $\alpha_i \geq 0$, where α_i are the positive Lagrange multipliers [16][17].

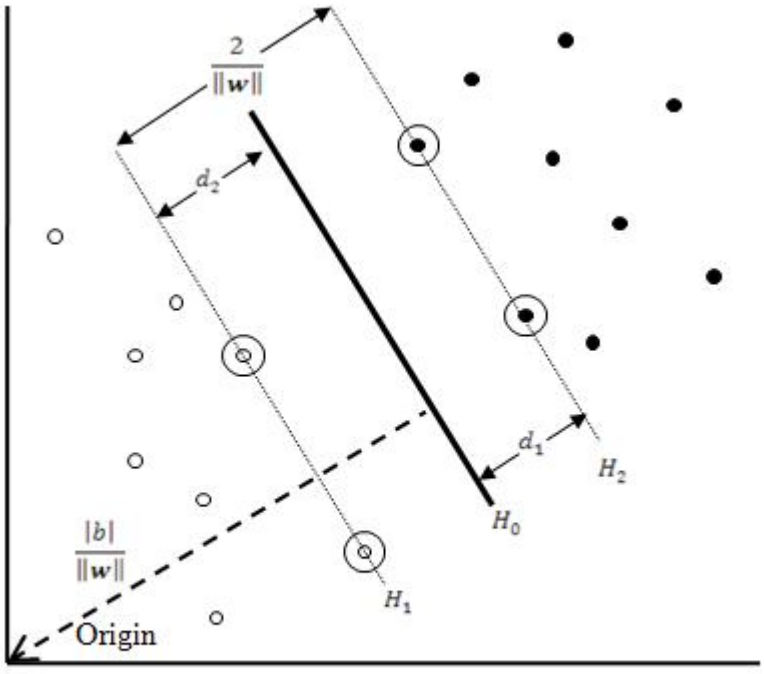


Figure 4.4: Linear separating hyperplanes. Support Vectors are circled [17].

$$\Psi(\boldsymbol{w}, b, \alpha) \equiv \frac{1}{2}\|\boldsymbol{w}\|^2 - \sum_{i=1}^{m} \alpha_i y_i(\langle \boldsymbol{w}, \boldsymbol{x}_i \rangle + b) + \sum_{i=1}^{m} \alpha_i \quad \textbf{(Equation 4.20)}$$

To make the optimization easier, the primal problem in Equation 4.20 can be transformed to the dual problem by maximizing $\Psi(\boldsymbol{w}, b, \alpha)$ subject to the constraints of Equation 4.21 and 4.22a [17]. This particular formulation is called the Wolfe dual, and has the property that the maximum of $\Psi(\boldsymbol{w}, b, \alpha)$ subject to the constraints in Equation 4.21 and 4.22a, occurs at the same values of $\boldsymbol{w}, b$and α, as the minimum of $\Psi(\boldsymbol{w}, b, \alpha)$ subject to the constraint of Equation 4.22b [16][17].

$$\frac{\partial \Psi}{\partial \boldsymbol{w}} = 0 \quad \Rightarrow \quad \boldsymbol{w} = \sum_i \alpha_i y_i \boldsymbol{x}_i \qquad \textbf{(Equation 4.21)}$$

$$\frac{\partial \Psi}{\partial b} = 0 \quad \Rightarrow \quad \sum_i \alpha_i y_i = 0 \qquad \textbf{(Equation 4.22a)}$$

$$\frac{\partial \Psi}{\partial \alpha} = 0 \quad \Rightarrow \quad \sum_{i=1}^{m} y_i(\langle \boldsymbol{w}, \boldsymbol{x}_i \rangle + b) = \sum_{i=1}^{m} 1 = m \qquad \textbf{(Equation 4.22b)}$$

The dual of Equation 4.20 can be formulated by substituting the constraints from Equation 4.21 and 4.22a into Equation 4.20, to give Equation 4.23 [16][17].

$$W(\alpha) = \sum_i \alpha_i - \frac{1}{2}\sum_i \sum_j \alpha_i \alpha_j y_i y_j \langle \boldsymbol{x}_i, \boldsymbol{x}_j \rangle \qquad \textbf{(Equation 4.23)}$$

The problem is then to minimize Equation 4.23, subject to the constraints in Equation 4.22a and $0 \leq \alpha_i \leq$ C, where C is a constant [18]. This step is a quadratic programming problem, and can be solved using developed techniques, such as the Sequential Minimal Optimization method, which will output α_i corresponding to training samples and as a result, provide the optimal hyperplane as described in Equation 4.24 [18].

$$\boldsymbol{w}^* = \sum_{i=1}^{m} \alpha_i y_i \boldsymbol{x}_i \qquad \textbf{(Equation 4.24)}$$

Using Equation 4.24, the bias,b, can be found, by using Equation 4.25, where $\boldsymbol{x}_{SV}^{+}$ and $\boldsymbol{x}_{SV}^{-}$ are the positive and negative support vectors, respectively [18].

$$b = -\frac{1}{2}(\langle \boldsymbol{w}^*, \boldsymbol{x}_{SV}^{+} \rangle + \langle \boldsymbol{w}^*, \boldsymbol{x}_{SV}^{-} \rangle) \qquad \textbf{(Equation 4.25)}$$

In addition, the output α_i values will only be non-zero for support vectors as described in Equation 4.26 where solutions are either $\alpha_i = 0$ or the equality of Equation 4.14 [18].

$$\alpha_i(y_i[\langle \boldsymbol{w}^*, \boldsymbol{x}_i \rangle + b] - 1) = 0 \qquad \textbf{(Equation 4.26)}$$

The classifier may then be described by Equation 4.27 where sgn is the function described in Equation 4.28, and **R** is the input data to be classified (test data) [18].

$$f(\boldsymbol{R}) = sgn(\langle \boldsymbol{w}^*, \boldsymbol{R} \rangle + b) \qquad \textbf{(Equation 4.27)}$$

$$sgn(x) = \begin{cases} -1, & x < 0 \\ 0, & x = 0 \\ 1, & x > 0 \end{cases} \qquad \textbf{(Equation 4.28)}$$

Equation 4.27 takes an test input data, **R**, and classifies it as the positive class if $\langle \boldsymbol{w}^*, \boldsymbol{R} \rangle + b > 0$, and the negative class if $\langle \boldsymbol{w}^*, \boldsymbol{R} \rangle + b < 0$. Therefore a binary classifier has been produced, using the α_i values from the training phase, the training vectors $\boldsymbol{x}_i$ with corresponding labels y_i, and **R**, as shown in Equation 4.29.

$$f(\boldsymbol{R}) = \sum_{i=1}^{m} \alpha_i y_i \langle \boldsymbol{x}_i, \boldsymbol{R} \rangle + b \qquad \textbf{(Equation 4.29)}$$

For the purpose of the project, the input feature vector to the classifier is described in Equation 4.7. A network map of the SVM that was implemented is illustrated in Figure 4.5. The map indicates that every components of the input data, was multiplied with the corresponding component for each training data sample, then multiplied by the Lagrange coefficient, α_i and the label, y_i corresponding to the ith training data sample, and finally summed (along with b) to output -1 for a negative

classification or 1 for a positive classification. In addition, Figure 4.5 displays the SVM model assuming the input data was linearly separable.

Since the extracted ECG features were not linearly separable, an alternative solution was found that transformed the data into a higher dimensional space.

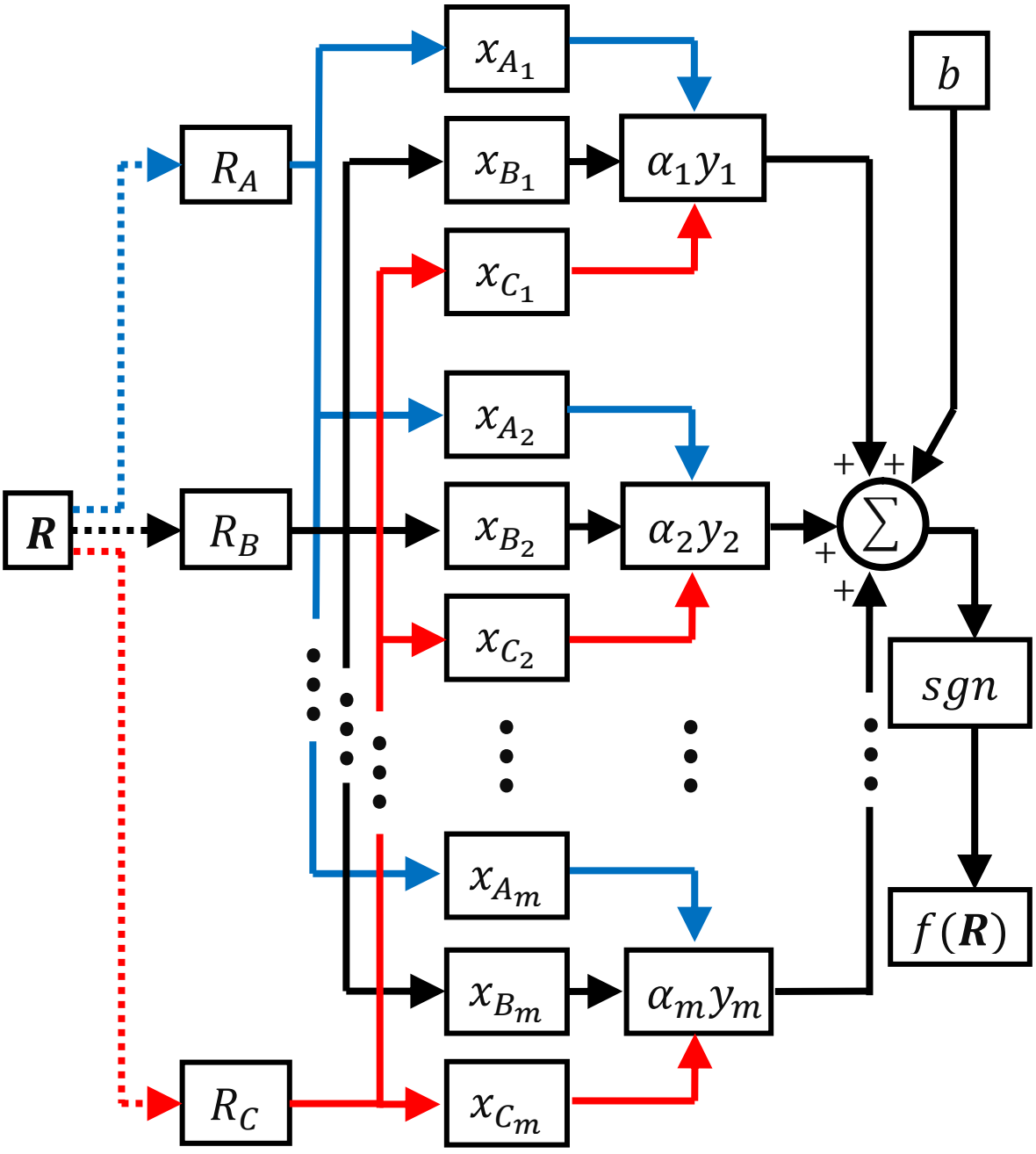


Figure 4.5: The SVM network map used for classification of ECG data. The feature vectors are: R wave peak (R_A), instantaneous heart rate (R_B) and the QRS width (R_C).

The solution consisted of mapping input data to a new features space, where data was successfully separated and classified. Figure 4.6 illustrates the how the linearly inseparable data can be mapped to a higher dimension to produce separable data.

Figure 4.6: Linearly inseparable data can be mapped to higher dimensions to create separable data [18].

The input data from Figure 4.5 was mapped using the transformation in Equation 4.30 [18]. If a kernel function is defined as $K(\boldsymbol{x}_a, \boldsymbol{x}_b) = \langle \Phi(\boldsymbol{x}_a), \Phi(\boldsymbol{x}_b) \rangle$, then Equation 4.29 can be expressed as Equation 4.31 [18].

$$\boldsymbol{R} \longrightarrow \Phi(\boldsymbol{R}) \qquad \textit{(Equation 4.30)}$$

$$f(\boldsymbol{R}) = \sum_{i=1}^{m} \alpha_i y_i K(\boldsymbol{x}_i, \boldsymbol{R}) + b \qquad \textit{(Equation 4.31)}$$

Examples of the kernel functions are the linear, quadratic, Gaussian Radial Basis Function (RBF) and polynomial kernels, as shown in Equations 4.32-4.35 [19].

$$Linear: \quad K(\boldsymbol{x}_a, \boldsymbol{x}_b) = \langle \boldsymbol{x}_a, \boldsymbol{x}_b \rangle + c \qquad \textit{(Equation 4.32)}$$

$$Quadratic: \quad K(\boldsymbol{x}_a, \boldsymbol{x}_b) = 1 - \frac{\|\boldsymbol{x}_a - \boldsymbol{x}_b\|}{\|\boldsymbol{x}_a - \boldsymbol{x}_b\| + \theta} \qquad \textit{(Equation 4.33)}$$

$$RBF: \quad K(\boldsymbol{x}_a, \boldsymbol{x}_b) = \exp\left(-\frac{\|\boldsymbol{x}_a - \boldsymbol{x}_b\|^2}{2\sigma^2}\right) \qquad \textit{(Equation 4.34)}$$

$$Polynomial: \quad K(\boldsymbol{x}_a, \boldsymbol{x}_b) = (\gamma \langle \boldsymbol{x}_a, \boldsymbol{x}_b \rangle + c)^d \qquad \textit{(Equation 4.35)}$$

By using these kernels, the classification system transformed the problem into a new feature space in efforts to improve the effectiveness of the SVM.

4.3 Experimental Procedures

The experimental procedures consisted of testing the QRS detector and the SVM using several kernel functions. As previously mentioned, the QRS detector from Chapter 4.2 used raw ECG data to output a vector defined by Equation 4.7. To test the effectiveness of QRS detection, 4 sets of data were presented to the system which was Rob and Mike's training and testing data. The outputs were then analyzed qualitatively by observing heart beats for incorrect QRS detection. The training data was then used to train the SVM to identify Mike's ECG as the positive class and any other ECG (i.e Rob's data) as Not-Mike's ECG. To test the SVM, a data set containing labeled test ECG data from Rob and Mike, was input. The output of the classification stage was then compared to the input for true positives, true negatives, false positives and false negatives, which were then used to calculate the relative error in classification for each kernel function. As well, the efficiency if the QRS detector was assessed by analyzing the classification error before and after the elimination of outliers.

Chapter 5

Results and Discussion

As previously mentioned (refer to Schematic 4.1), the first component of the overall system is the QRS detector or feature extraction stage. This element analyzed the data and would output features necessary for the classification stage. The SVM is the classification stage that uses the input feature vector to train the SVM and then execute classification on non-training data. In this implementation of the ECG analyzer, the signal from human test subjects was used to train the SVM, and then non-training data from the subjects was used to distinguish between them.

5.1 Results of QRS Detector

This component of the ECG analyzer represents the feature extraction stage. To test the feature extraction stage, experimental ECG data obtained from the hardware component of the project was used and analyzed to output information regarding desired features that were then classified.

i) Filtering

In this component of QRS detection, test ECG data obtained from the hardware component was filtered with a BPF. The output of the filter is shown in Figure 5.1.

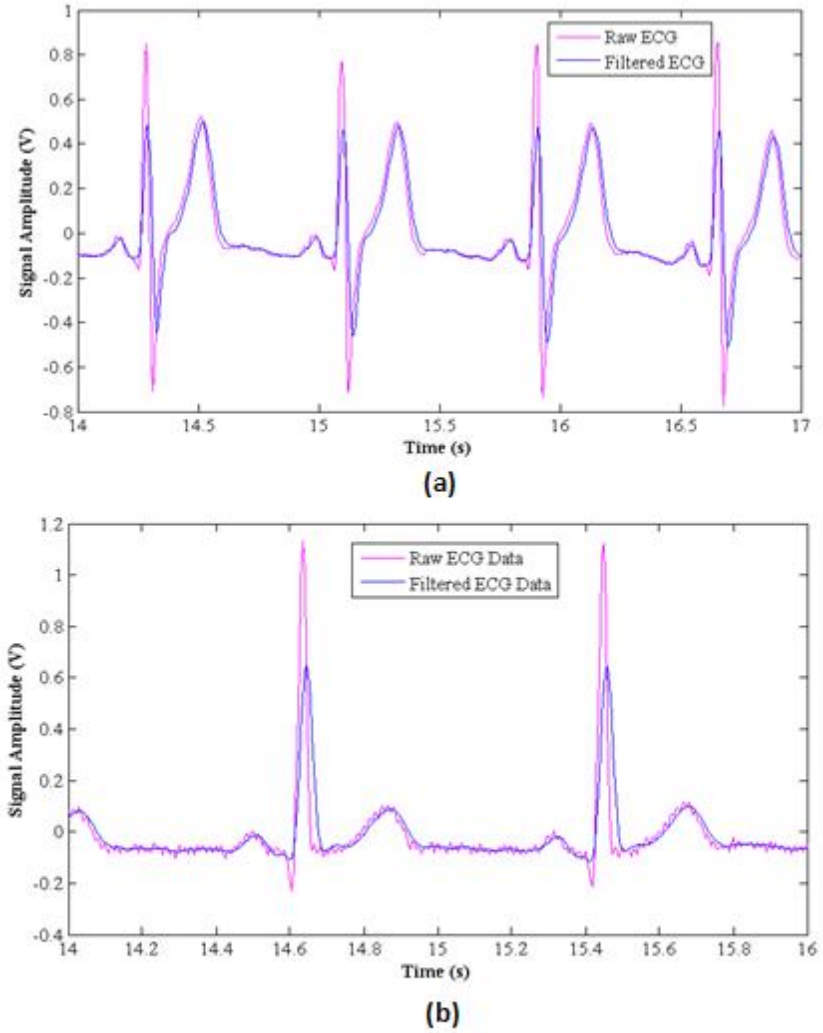
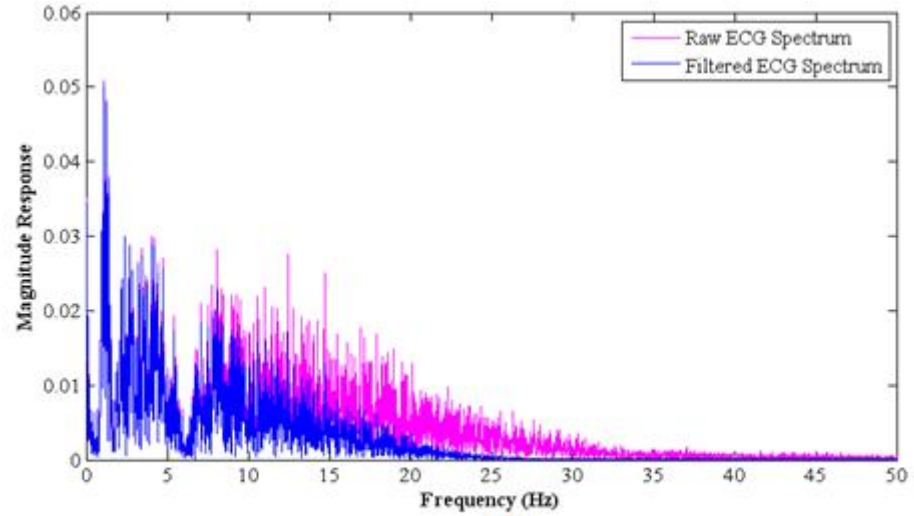
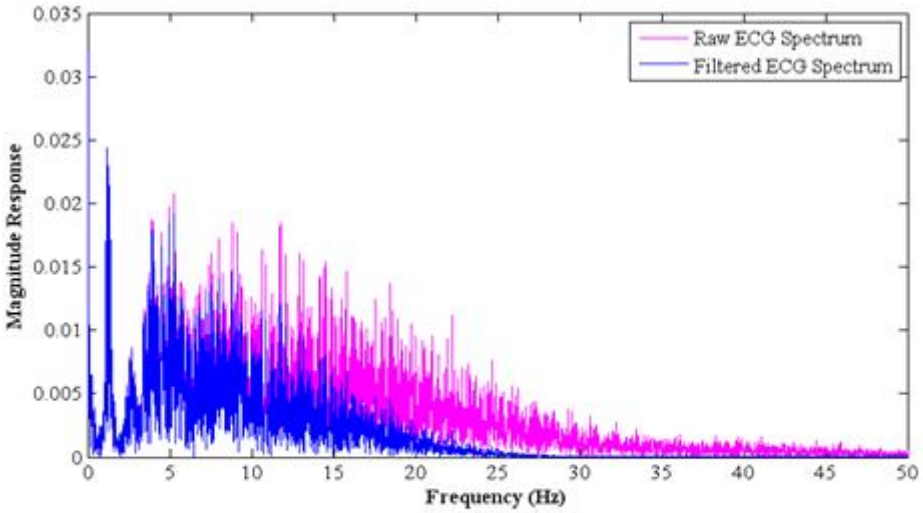


Figure 5.1: Example plots of Robs data (a), and Mikes data (b), demonstrating the result of the digital band pass filter.

As represented in Figure 5.1 above, the filtered waveform is a smoother version of the raw ECG data. Furthermore, there exists a slight manipulation in the shape of the filtered output taking into consideration the high frequency components that were attenuated. The slight difference in the shape of the filtered ECG signal indicates that higher frequency components contribute to the characteristic shape of the ECG signal.



(a)



(b)

Figure 5.2: Spectral plots of Robs data (a), and Mikes data (b), demonstrating the result of the digital band pass filter.

In addition to the time domain, a signal may also be represented in the frequency domain to provide information regarding a filters performance (refer to Figure 5.2, a representative spectrum of Figure 5.1). As the Figure 5.2 depicts, the frequencies above 11 Hz are attenuated, which is consistent with the notion that the LPF is operational. However, due to the lack of low frequency attenuation, the filter created a perfect overlap

for frequencies below 5 Hz. Thus, this suggests that the HPF component was not working properly.

Another method of analyzing a filters performance is to analyze the power density of the signals. In Figure 5.3, the power spectral density (PSD) was calculated from data used in Figure 5.1 and plotted in an overlapping manner to provide qualitative assessment of the filters performance. Based on the findings in Figure 5.3, it is evident that power

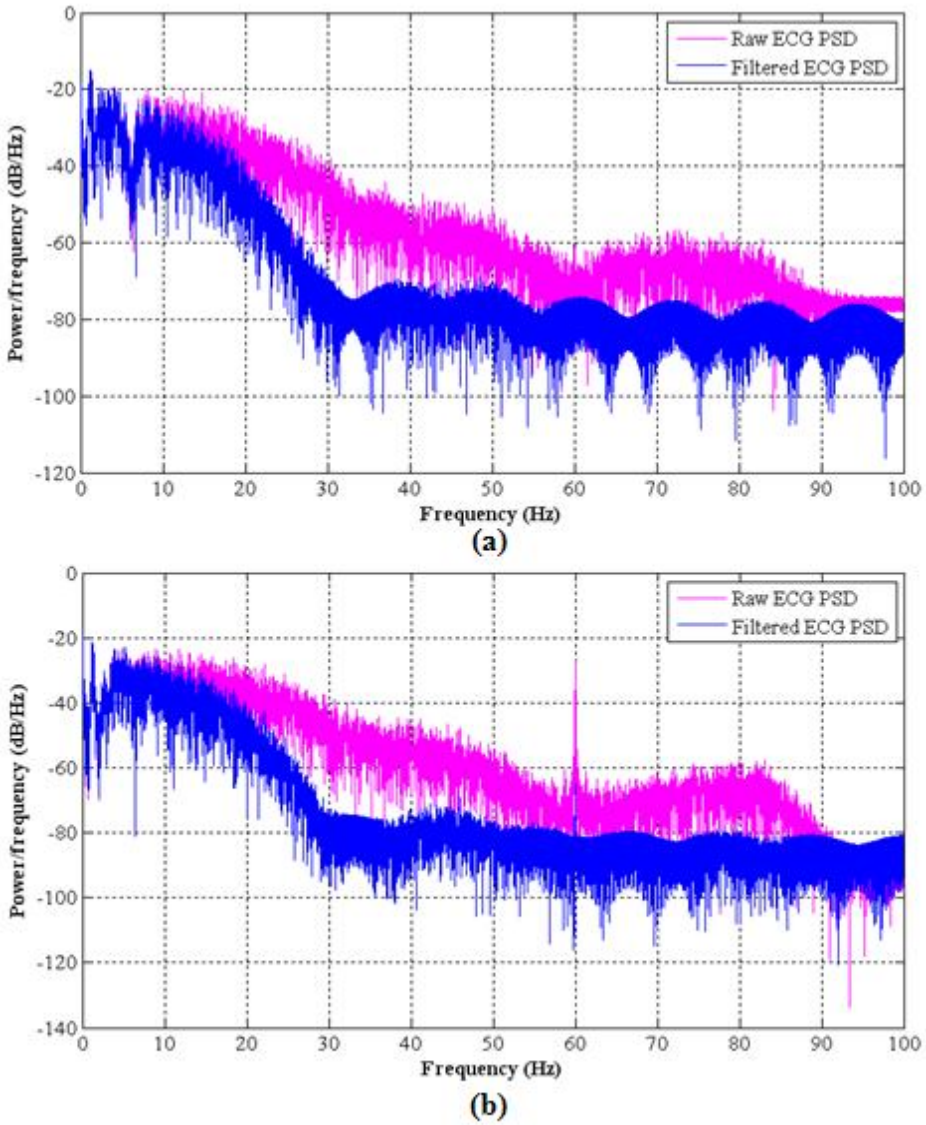


Figure 5.3: PSD plots of Robs data (a), and Mikes data (b), demonstrating the result of the digital band pass filter.

from the QRS complex is conserved from raw to filtered data. This is an important feature due to all relevant information being derived from the position of the QRS

complex. In addition, the absence of the HPF component can also be noted since there is no decrease in power density for frequencies below 5 Hz. When considering the time domain, frequency domain and PSD signals, the BPF was regarded as operational only for frequencies above 11 Hz. However, the lack of attenuation for frequencies below 5 Hz is a result of the non-operational HPF. Although the filter component did not operate as originally designed, there were very few errors in the classification component and thus demonstrating the system can successfully run even in the absence of a HPF.

ii) Differentiation

Upon completion of the filtering stage, the differentiation stage was used to distinguish filtered data with respect to time and outputted to the subsequent stages.

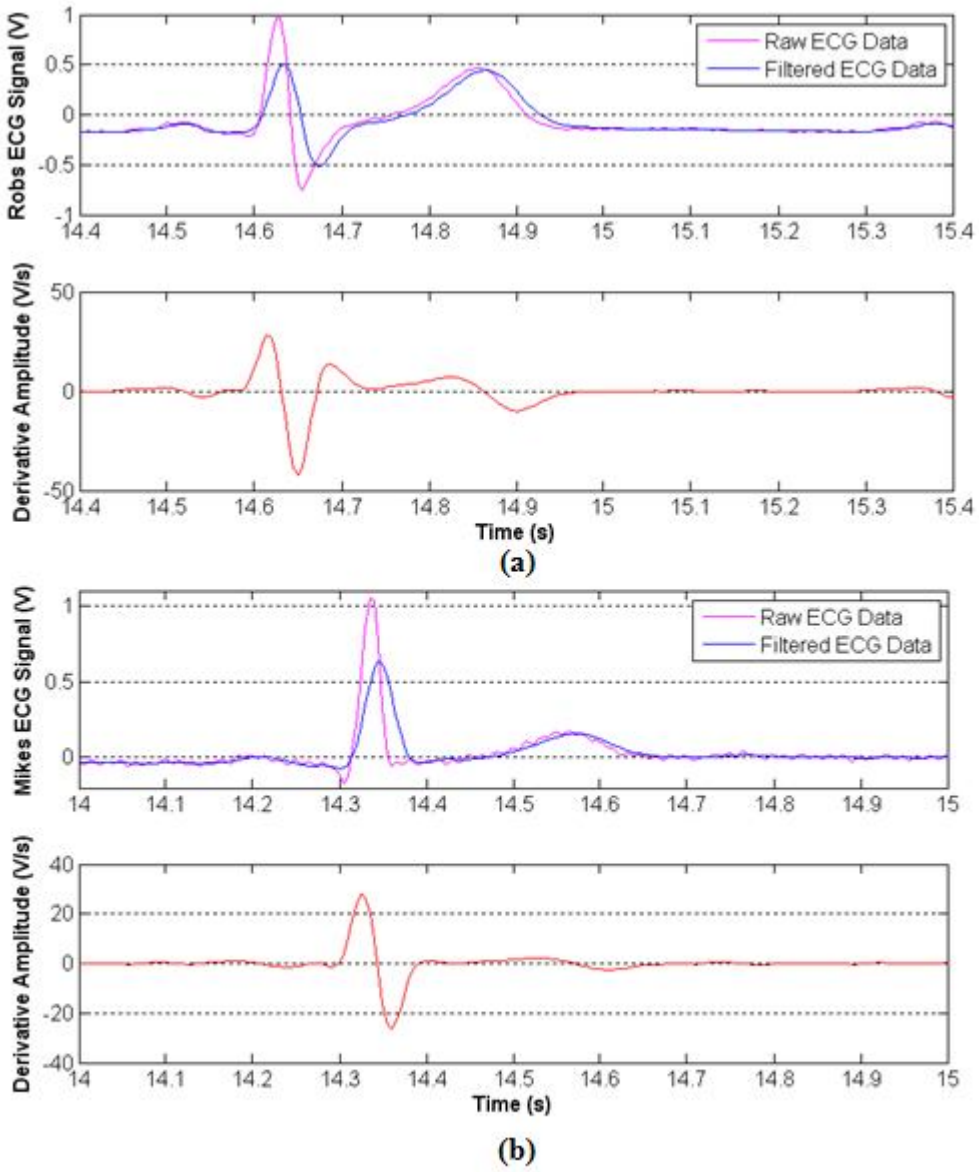


Figure 5.4: Plot of Robs data analysis (a), and Mikes data analysis (b), demonstrating the result of the derivative function.

The results of differentiating Figure 5.1 data are shown in Figure 5.4. The data demonstrates that for values that change rapidly, the output of the derivative component is a high value. This was confirmed by matching the location of positive peak in the differentiated waveform with the filtered waveform, and observing the slope. On the other hand, values that change gradually over time are represented by less prominent maxima and minima in the differentiated waveform. This can be seen at $t \approx 14.9$ s (Figure 5.4a), where the differentiated waveform has a small value and the slope of the

filtered data is smaller in magnitude than the QRS data. The maxima and minima of the ECG signals were seen to occur when the output of the differentiating function was zero and therefore used as a time marker for any peaks or minimums. An example of this is at $t \approx 14.35$ s (Figure 5.4b), where the peak in the filtered ECG waveform resulted in a zero crossing in the differentiated waveform.

The relationship between the filtered signal and its derivative can describe locations of features, while providing insight on these occurring events. An example of this is the P wave which can be described by smaller minima and maxima due to the relatively slow depolarizing nature of the atria. On the other hand, by observing the large values in minima and maxima from the differentiated waveform, the QRS complex can be identified, which indicate that ventricular depolarization occurred incredibly fast. The T wave was seen to have values greater than the P wave, however insignificant in comparison to the QRS complex. These results show that the relationship between a signal and its derivative provide useful information needed to describe occurring events.

iii) Nonlinear Transformation: Squaring

This component squares the output of the differentiation block, as shown in Figure 5.5. One can observe that the output of the squaring function creates positive values of the derivative function and emphasizes steep slopes of the filtered output much more than flatter slopes. Since peak values in the squared output represent the steepest slope, the fast changing profile of the QRS complex can be related. An example is shown by the peaks of the squared output, in the time interval [14.6, 14.7] (refer to Figure 5.5a). The P and T wave cannot be properly recognized since the values of their peaks (squared function) are far less in comparison with QRS peaks. Since rapidly changing events are emphasized much more than slower ones, the squaring function emphasizes the QRS complex the most and thus makes the output of the nonlinear transformation stage an essential step in the analysis of QRS complexes.

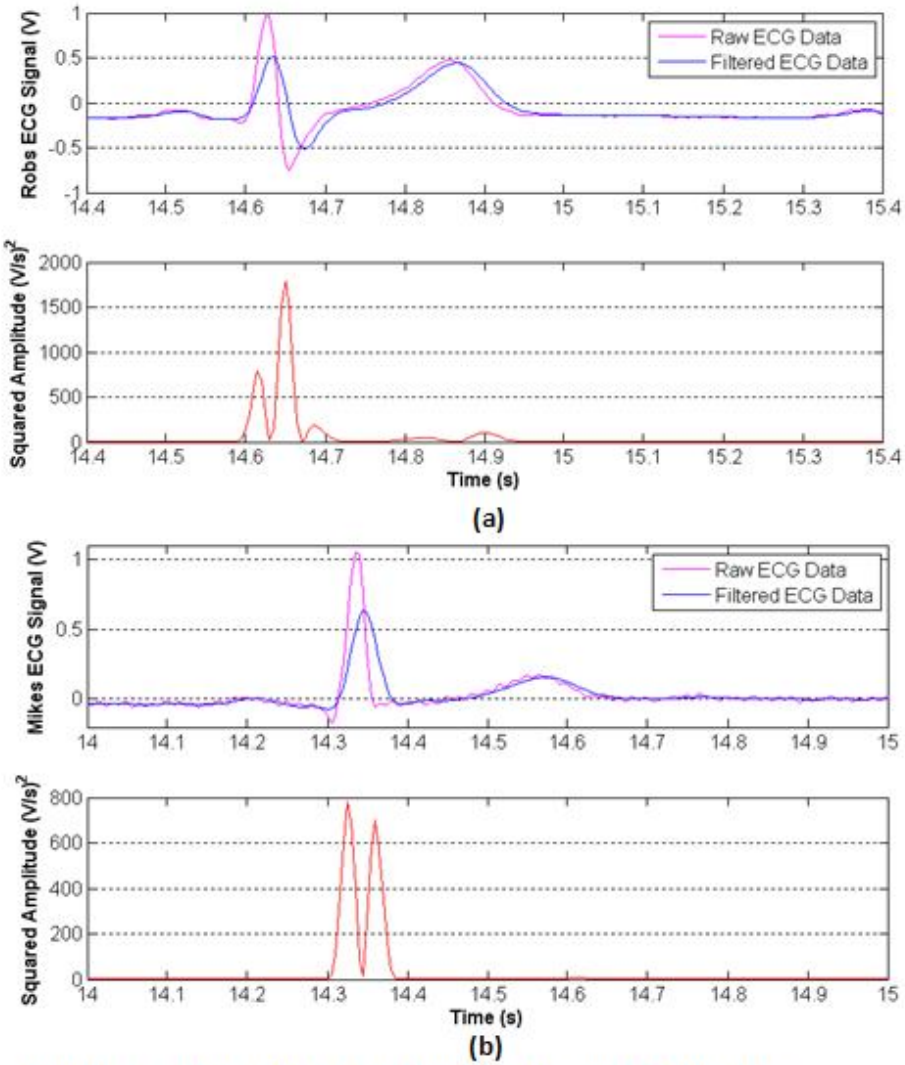
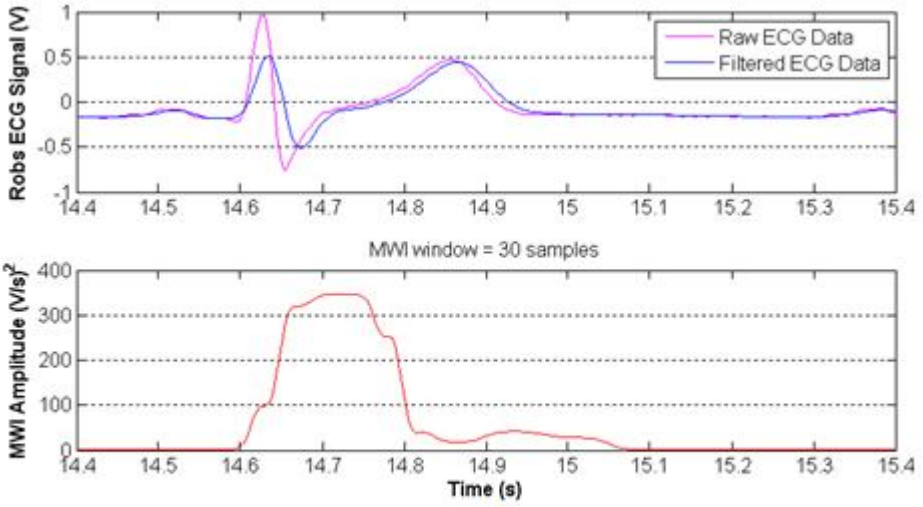


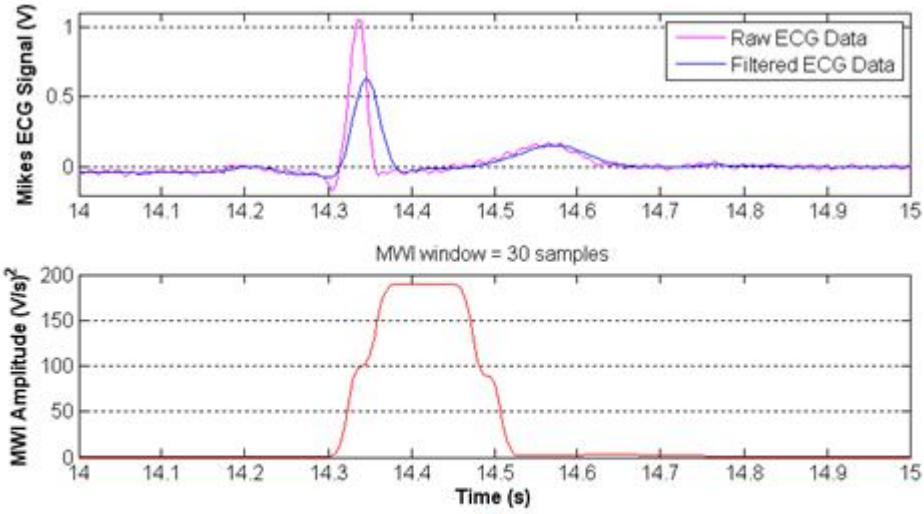
Figure 5.5: Plot of Robs data analysis (a), and Mikes data analysis (b), demonstrating the result of the squaring function.

iv) Moving Window Integrator (MWI)

The results from the MWI showed considerable impact on the outcome of the preprocessing stage, described in Figure 2.1. The MWI function obtained a time average of the squaring function, which consisted of calculating the mean for τ samples for every point in time, and resulted in a shape described in Figure 5.6.



(a)



(b)

Figure 5.6: Plot of Robs data analysis (a), and Mikes data analysis (b), demonstrating the result of the MWI.

In Figure 5.6, the MWI resulting from the data in Figure 5.5, demonstrates the differences between the two data sets. By comparing (a) and (b) of Figure 5.6, the MWI waveform can be seen to differ in both shape and magnitude, demonstrating that ECG signals are unique to every individual. In addition, the shape of the MWI output is also characterized by the window size, τ. Figure 5.7 demonstrates the effect of varying τ for the given sample rate of 200 Hz. It can be seen that if τ is too large, the MWI function merges QRS complexes and T waves together. As well, if τ is too small, the MWI

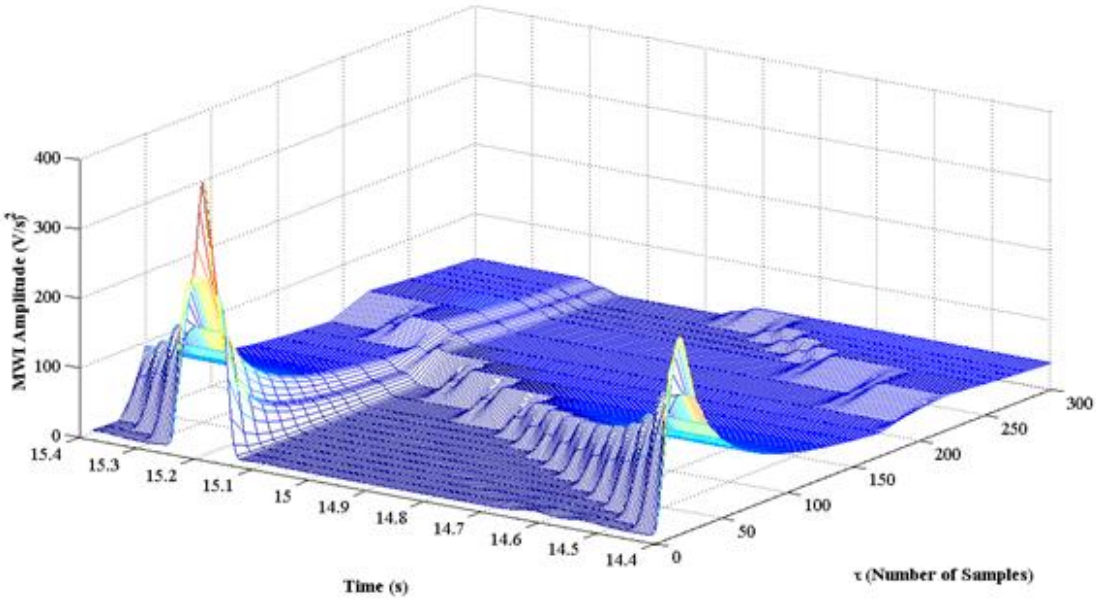


Figure 5.7: 3-Dimensional Mesh plot illustrating the effect of varying MWI window size (τ samples) on the same time interval. The following values of τ were used: 10, 20, 30, 40, 50, 60, 70, 80, 90, 100, 110, 130, 150, 170, 190, 220, 250, 260, 270, 280, 300.

function has several peaks and its rising edge does not represent the QRS width appropriately. As a result, the features extracted from the MWI are a combination of unique ECG data and window size. Taking into account the best window size was determined empirically to be 150 ms [11], the value for τ used was 30 samples. The use of this value resulted in non-merged MWI data as evidenced by Figure 5.7 at $\tau = 30$ samples. With this window size, the MWI data can be used for determining information relevant to patterns in ECG signal.

Due to the MWI output, some features may be extracted for use in the classification stage. As described in Chapter 4.2, one of the most important purposes of

the MWI waveform is its relation to the QRS complex (refer to Figure 4.2). In this figure, the width of the QRS complex corresponds to the width of the rising edge of the MWI waveform. This can be observed using the test data in Figure 5.6, by comparing the width of the rising edge of the MWI waveform to the width of the QRS complex. The features that were extracted using the components of the QRS detector are illustrated in Figure 5.9. This figure shows the final output of the QRS detector using the training data from Figure 5.1.

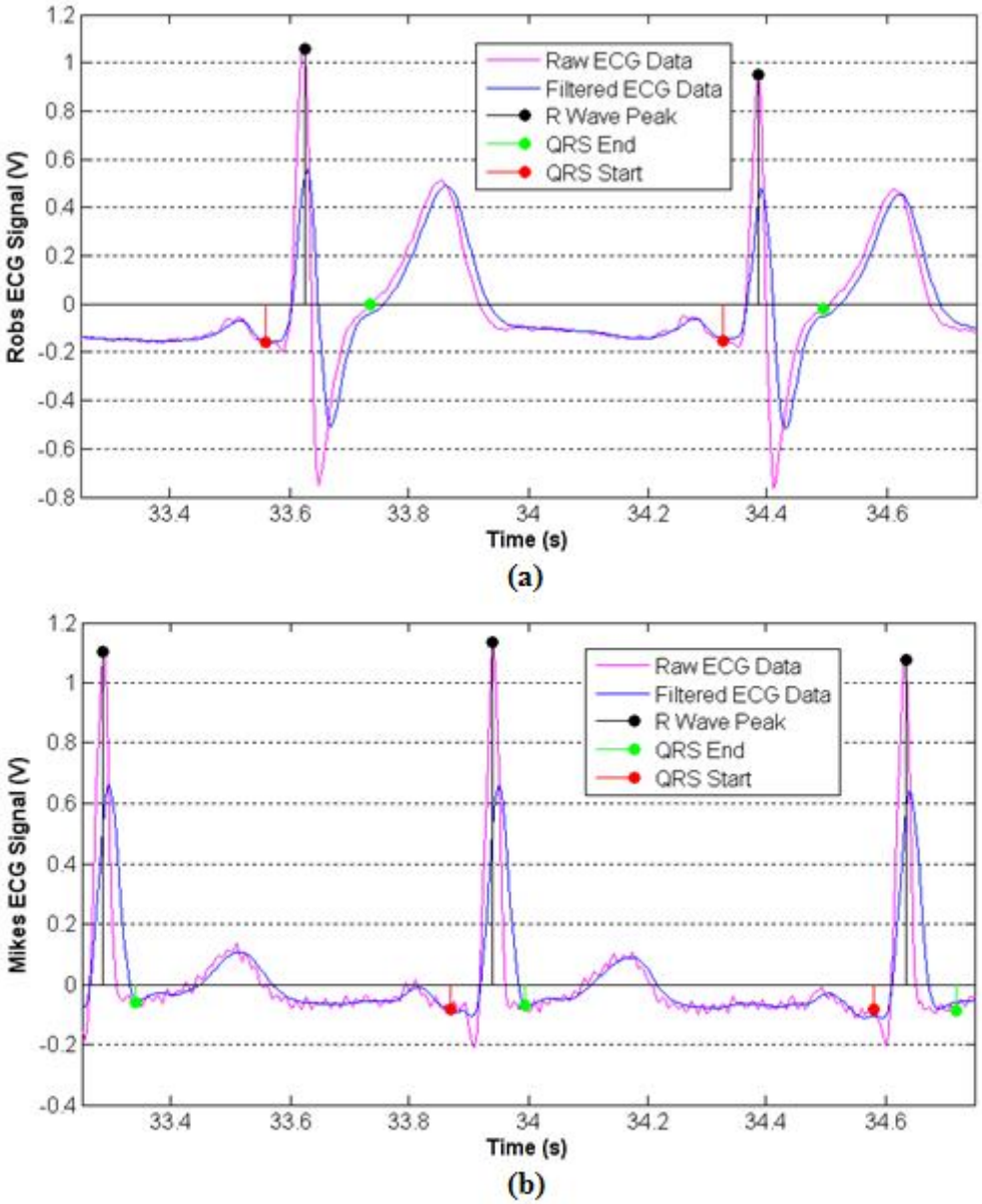
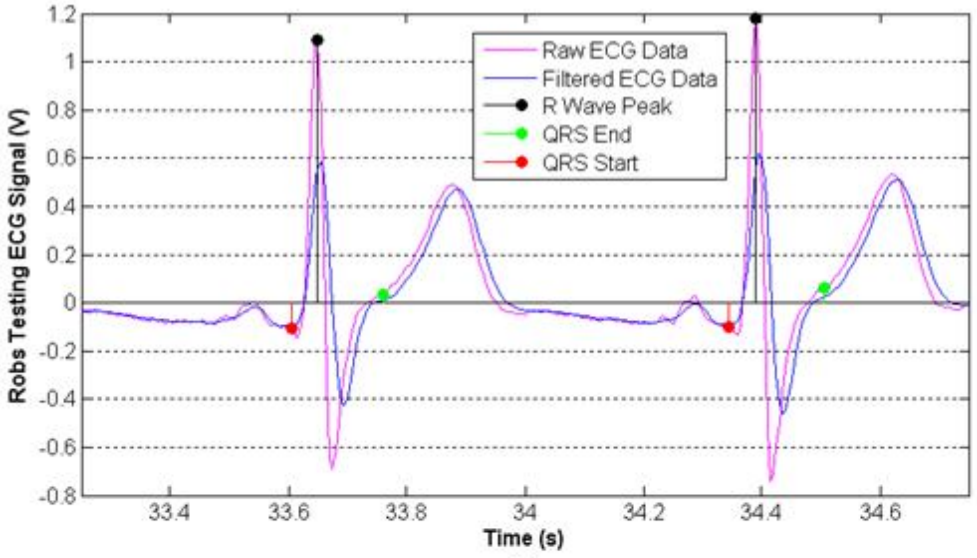


Figure 5.9: Final plots of Rob (a), and Mikes (b) ECG data analyzed with the QRS detector.

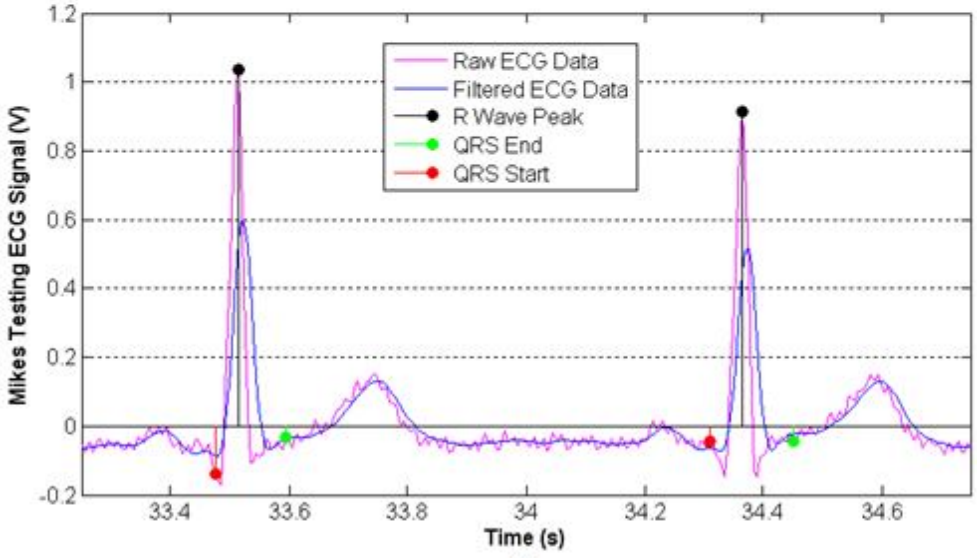
As observed in the illustration, the QRS detector detected the start and end of the QRS complex along with the position of the R wave peak, for Rob and Mike's ECG signals. The results from this figure demonstrate that the QRS detector can function properly, regardless of the difference in ECG shape and amplitude.

5.2 Results of Support Vector Machine

As previously described, the pattern recognition component is comprised of the training and classification phase. That being said, analyzing error associated with the classification system is strongly dependant on the error associated with the inputs. As shown in Figure 5.10, the QRS readings of Mike and Rob's test data supports the notion that the algorithm is in fact working, which is expressed by the detected features. Also,



(a)



(b)

Figure 5.10: Example of QRS detection on Rob (a) and Mikes (b) Testing ECG data. The results from QRS detection of training data are the same as Figure 5.9.

although the ECG morphology is different, the QRS detector is still able to extract the relevant features. When analyzing these diagrams, the presence of more than one cluster of data is evident, as shown in Figure 5.11. Under ideal circumstances, there would be only one present. However, due to errors in QRS detection there was more than one

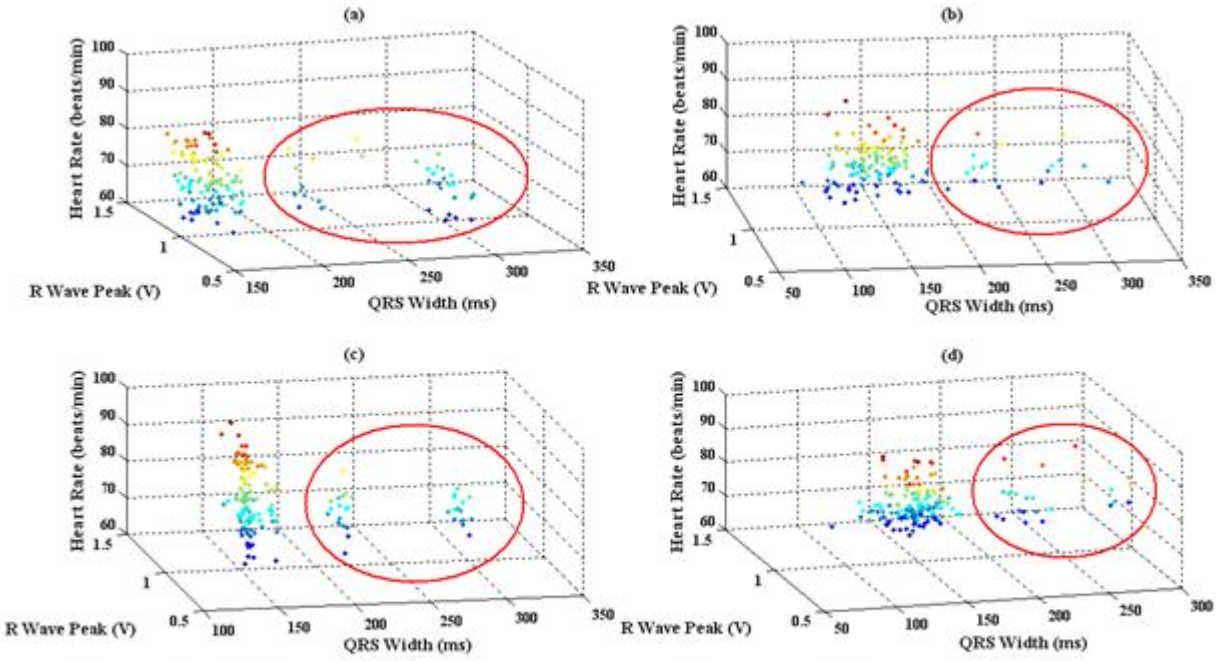


Figure 5.11: 3D scatter plots representing the extracted features from Robs training data (a), Mikes training data (b), Robs testing data (c), and Mikes testing data (d).

observed. These clusters of data (red ellipses) represent data points that are numerically distant from the remaining values and are better known as outliers [15]. Although a restriction was placed on the extracted features, not all outliers were eliminated. These clusters of data are a result of errors associated with the QRS detection algorithm. Since the outliers of Figure 5.11 were common for very wide QRS complexes (greater than 200ms), they were eliminated from the training data. By outliers being reduced from the feature vector, the SVM model was able to properly train and execute to find the optimal separating hyperplane required for classification. As shown in Figure 5.12, using the testing ECG data from Figure 5.9, SVM classification results were obtained for a variety of kernel functions. In order to display the effectiveness of the different SVM's, 2-Dimensional feature vectors were used due to the graphic ease in expressing such plots. Looking at Figure 5.12a, it represents the classification of an SVM model trained with a

linear kernel. Since the data was not completely separable using a linear hyperplane, the overall error for the linear kernel was 36.1%. When considering Figure 5.12b, using a quadratic kernel for training the SVM model had an overall error of 17.43%. The SVM shown in Figure 5.12c was trained using the Gaussian Radial Basis Function (RBF) kernel, which had an overall error of 19.92%. The last variation found on Figure 5.12d, represents the output of classification when the SVM was trained using the polynomial kernel. The overall error of the polynomial kernel was 19.09%, which was the lowest of all kernels tested. A summary of the classification test is displayed in Table 5.1 and Table 5.2 (note: values are measured in number of heart beats).

Table 5.1: Table showing the statistical results from the use of various kernel functions. Positives correspond Mike's data and Negatives correspond to Not-Mike's data. The feature vectors are 2D, comprising of QRS width and R wave peak value.

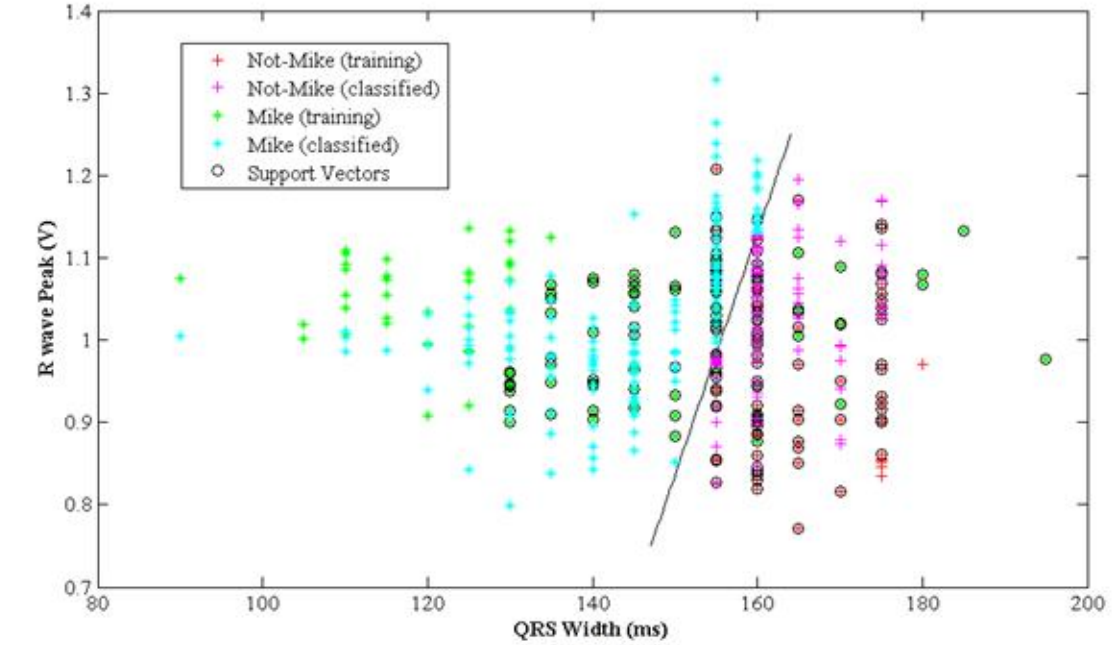
Kernel Function	Total Input +	Total Input -	Total Output +	Total Output -	True +	True -	False +	False -	Overall Error
Linear	134	107	147	94	97	57	50	37	**36.10%**
Quadratic	134	107	116	125	104	95	12	30	**17.43%**
RBF	134	107	120	121	103	90	17	31	**19.92%**
Polynomial	134	107	116	125	102	93	14	32	**19.09%**

Table 5.2: Table showing the statistical results from the use of various kernel functions including outliers. Positives correspond Mike's data and Negatives correspond to Not-Mike's data. The feature vectors are 2D, comprising of QRS width and R wave peak value.

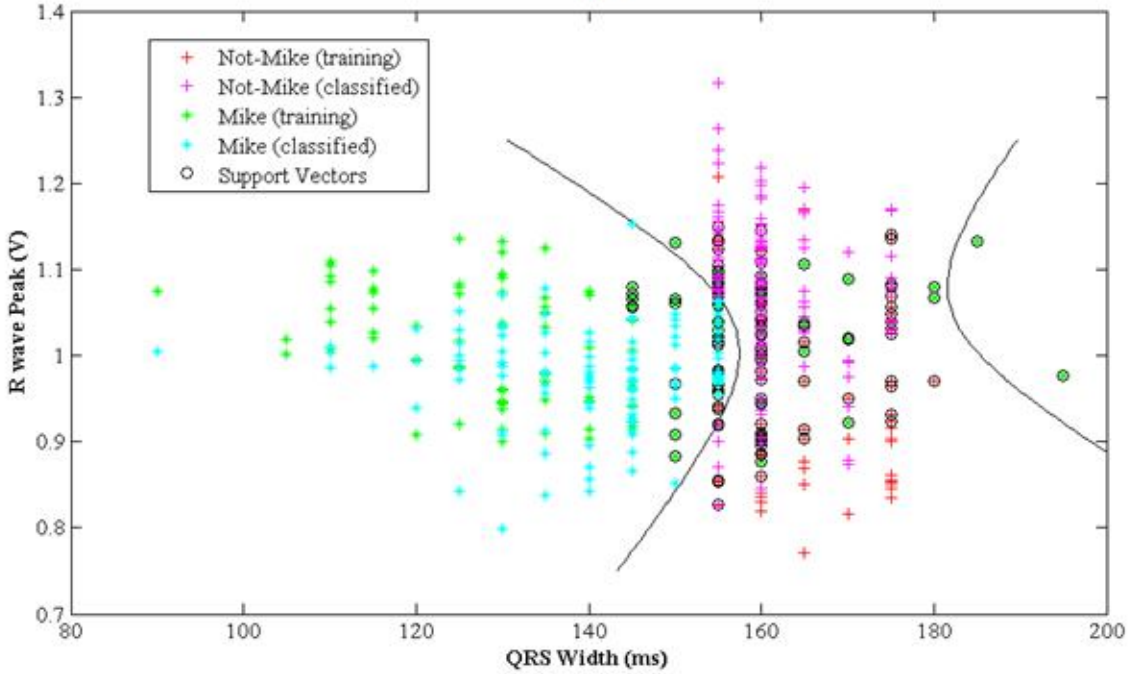
Kernel Function	Total Input +	Total Input -	Total Output +	Total Output -	True +	True -	False +	False -	Overall Error
Linear	157	138	211	84	106	33	105	51	**52.88%**
Quadratic	157	138	168	127	112	82	56	45	**34.24%**
RBF	157	138	132	163	105	111	27	52	**26.78%**
Polynomial	157	138	136	159	110	112	26	47	**24.75%**

The results seen in Table 5.1 and Figure 5.12 indicate that the classification system performed reasonably well if outliers (features with QRS widths greater than 200ms) were eliminated. Table 5.2 demonstrates the results when the outliers were not removed. When comparing the error terms for each kernel in Table 5.1 and Table 5.2, it is clear that the outliers present a significant error to the classification system. Due to the

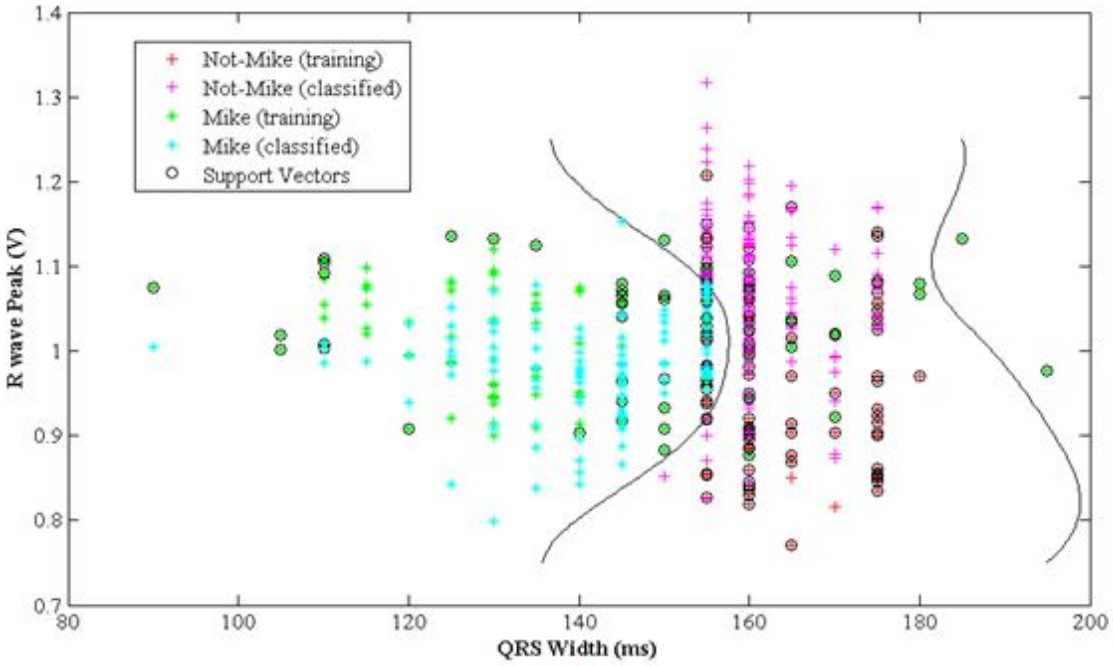
outliers arising as a result of incorrect QRS detection, the overall operation of the QRS detector can be described by the error terms presented in Tables 5.1 and 5.2. In addition, since patterns in both test ECG and labeled disease ECG data, are characterized with the same feature vectors, the results from successful patient identity translate to successful diagnosis. Furthermore, both show that the detector performs moderately well for the quadratic, RBF and polynomial kernels.



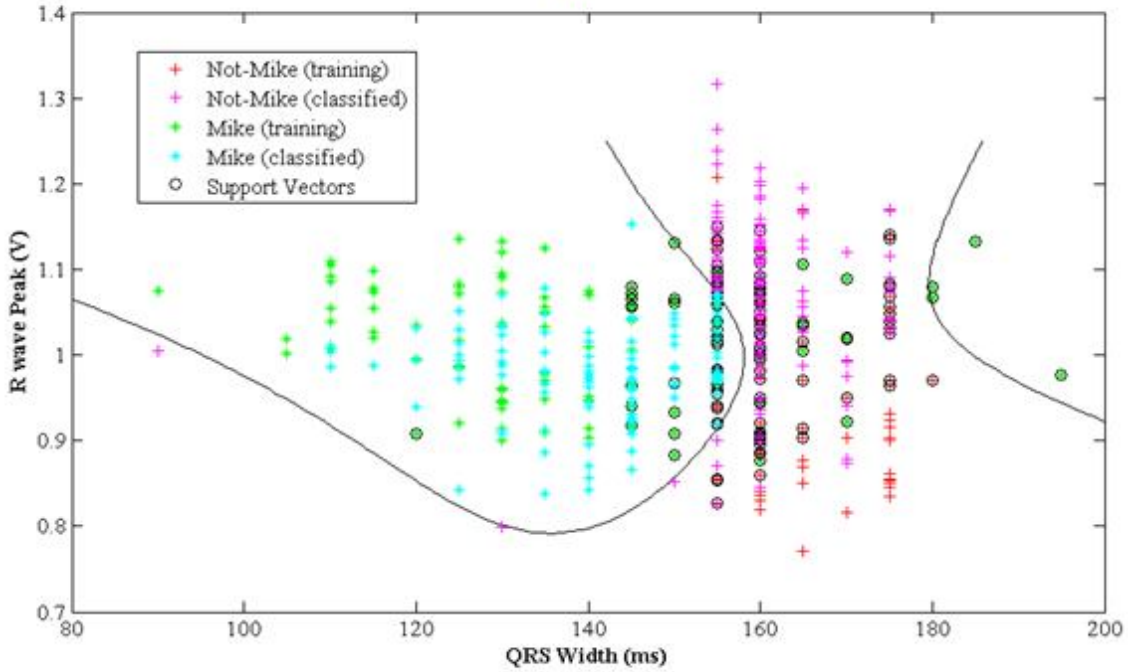
(a)



(b)



(c)



(d)

Figure 5.12: A plot of the trained and classified data for various kernel functions using Mike and Rob's ECG data. (a) was trained using the linear kernel, (b) was trained using the quadratic kernel, (c) was trained using the Gaussian Radial Basis Function (RBF) kernel with a scaling factor of 1, and (d) was trained using the polynomial kernel with an order of 3.

5.3 Overall Performance

The overall performance of the system was considered acceptable given the fact components of the QRS detector were not operating properly. As mentioned earlier, the HPF in the filter stage was not operation, and results from the classification stage provide a higher bound for error associated with classification. Therefore, correction of the HPF could result decrease the error in classification. Another restricting aspect was the dimension of the feature vector in Equation 4.7. Since the feature vector used to generate the statistics in Table 5.1 and Table 5.2 were 3-dimensional, the ability for the SVM to classify testing data was restricted to 3 features. A better ECG analyzer would include many more features in defining patterns associated with heart-disease. Given the scope of the project, the use of 3 features was enough to prove the functionality of an ECG analyzer using fundamental concepts.

Chapter 6

Conclusion

6.1 Conclusion of the Design Procedure

The purpose of the project was to design an ECG Analyzer that takes human ECG data, and provides a diagnosis for the state of the heart. The project was composed of two main components: QRS detection and SVM classification.

The QRS detection component was successful as it was able to analyze various types of ECG data and identify the QRS start, QRS end and R wave peak and position. Since the detector was able to extract features from different ECG data (i.e Rob and Mike's data), the detector could therefore be applied to any normal ECG waveform. Although the majority of heart beats that were detected in Rob and Mike's training and testing ECG data, were correctly identified, there existed some error. The error could be the result of the malfunctioning HPF or the detection ability, as evidenced by the outliers in Figure 5.11. The effects of error are shown in Table 5.1 and Table 5.2, where the elimination of outliers resulted in more accurate detections.

On the other hand, the classification system performed considerably well. Since the feature vector was 3-dimensional, the classification was restricted to patterns found in only 3 features. The results in Table 5.1 and Table 5.2 indicated that the SVM was capable of distinguishing between Mike's data and Not-Mike's data with about 19% to 35% error. The error is the result of incorrectly detected peaks from the feature extraction stage. This was evidenced by the increase in classification error when outliers were included in the test data. Therefore the performance of the feature extraction stage is reflected by the classification errors, which indicates that the QRS detector had some errors. In addition, the SVM classifier operated with the least error when the polynomial kernel was used, and the most error when the linear error was used (refer to Table 5.1 and Table 5.2).

Overall, the ECG analyzer was able to distinguish the identity for any input ECG data, as either Mike or Not-Mike with reasonable accuracy. Since the ECG analyzer was capable of determining the identity of input data using only 3 features, the system could therefore be capable of diagnosing patients using probability. With that being said, the project fulfilled its requirements of detecting QRS complexes and identifying ECG origin, inferring patient diagnosis.

6.2 Future Recommendations

Although the ECG analyzer operated with moderate results, there was still room for improvement. One source of improvement would be the size of the feature vector. By using more features, patterns could be defined better for specific CVDs. This would result in decreased classification errors, but the use of too many features could result in decreased generalization. To combat this problem, a feature selection component would be incorporated where only certain features are used to classify particular heart diseases. Additionally, the selection of a kernel function that provides the best generalization would have to be determined to further increase classification accuracy. One other improvement would be to implement the software onto to a microcontroller to provide portability.

Appendix A – Computer Programs Used

MATLAB R2008b

Microsoft Word 2007

Microsoft PowerPoint 2007

Paint

References

[1] Canada. Statistics Canada. *Mortality Summary List of Causes 2005,* Ottawa: Health Statictics Division, 2009.

[2] "Heart Attack and Acute Coronary Syndrome," May 18, 2009. [Online]. Available: http://adam.about.com/reports/000012_4.htm. [Accessed: April 2, 2010].

[3] "The Heart Basics," [Online]. Available: http://home.comcast.net/~pegglestoncbsd/cardiovascular.htm#Anatomy_. [Accessed: April 2, 2010].

[4] G. J. Tortora and B. Derrickson, "The Cardiovascular System: The Heart," in Principles of Anatomy and Physiology, 12th ed., New Jersey: John Wiley & Sons, Inc., 2009, pp. 717-749.

[6] F. Melgani, "QRS Complex," Oct., 2008. [Online]. Available: http://www.ing.unitn.it/~melganif/images/QRS.jpg. [Accessed April 2, 2010].

[7] C. D. Nugent, "Neural Networks in ECG Classification: What is Next for Adaptive Systems?" in *Neural Networks in Healthcare,* R. Begg, J. Kamruzzaman and R. Sarker, Pennsylvania: Idea Group Publishing, 2006, pp. 81-104.

[8] Microchip Technology Inc., "PIC24HJ32GP202/204 and PIC24HJ16GP304 Data Sheet," 2009, [Online]. Available: http://ww1.microchip.com/downloads/en/DeviceDoc/70289F.pdf. [Accessed April 2, 2010].

[9] PC-Vet Wireless ECG, "Vmed Technology: Advanced instrumentation for veterinarians," 2009. [Online]. Available: http://www.vmedtech.com/wireless_ecg.htm.

[10] G. D. Clifford, F. Azuaje and P. E. McSharry, "ECG Statistics, Noise, Artifacts, and Missing Data" in *Advanced Methods and Tools for ECG Data Analysis*, Boston: Artech House, 2006, pp. 55-93.

[11] J. Pan and W. J. Tompkins, "A Real-Time QRS Detection Algorithm," *IEEE T Bio-Med Eng*, vol. BME-32, no. 3, pp. 230-236, March 1985.

[12] DTREG, “SVM – Support Vector Machines,” 2007. [Online]. Available: http://www.dtreg.com/svm.htm. [Accessed April 2, 2010].

[13] C. C Chang and C. J. Lin, “ LIBSVM – A Library for Support Vector Machines,” Feb. 27, 2009. [Online]. Available: http://www.csie.ntu.edu.tw/~cjlin/libsvm. [Accessed April 6, 2010].

[14] A. J. Smola and B. Scholkopf, “A Tutorial on Support Vector Regression,” *NeuroCOLT2 Technical Report Series.* p. 1-71, Oct., 1998.

[15] J. Renze , “Math World,” Apri 19, 2010. [Online]. Available: http://mathworld.wolfram.com/Outlier.html. [Accessed April 6, 2010].

[16] S.R. Gunn, “Support Vector Machines for Classification and Regression”. *Faculty of Engineering, Science and Mathematics School of Electronics and Computer Science*; 1998.

[17] C.J.C. Burges, “A Tutorial on Support Vector Machines for Pattern Recognition,”*Data mining and Knowledge Discovery*, vol.2, issue.2, pp. 121-167, June 1998.

[18] D. Boswell, “Introduction to Support Vector Machines”. 2002.

[19] T. Hofmann, B. Schölkopf, and A. J. Smola. “Kernel Methods in Machine Learning,”*The Annals of Statistics*, vol.36, no. 3, pp. 1171-1220. 2008.

Vitae

NAME: Robert Tisma

PLACE OF BIRTH: Mississauga, Ontario

YEAR OF BIRTH: 1988

SECONDARY EDUCATION: Rick Hansen Secondary School