ECE 701: Project Report

Automated ECG Interpretation using the Extended Kalman Filter and Support Vector Machines

Tuesday August 16, 2011

Dr. Tom Doyle

Robert Tisma
0658942

Contents

1 Abstract **3**

2 Introduction **4**
2.1 The Electrocardiogram and the Heart . . . 4
2.2 Objective . . . 5

3 Methodology **7**
3.1 Overview . . . 7
3.2 ECG Dynamical Model . . . 7
3.3 Preprocessing . . . 8
3.3.1 Diagnostic Stage . . . 10
3.3.2 Phase Wrapping Stage . . . 10
3.4 EKF for ECG Segmentation . . . 11
3.4.1 Modified State Equations . . . 12
3.4.2 Initialization Procedure . . . 12
3.4.3 Linearization of EKF Equations . . . 13
3.5 Support Vector Machines . . . 17
3.5.1 Theory and Application . . . 17
3.5.2 Decision Method . . . 19

4 Results **21**
4.1 Normal Sinus Rhythm . . . 21
4.2 Atrial Flutter . . . 23
4.3 Supraventricular Tachyarrhythmia . . . 25
4.4 Atrial Fibrillation . . . 27

5 Discussion and Conclusion **29**

6 Acknowledgements **33**

List of Figures

2.1 The Anatomy of the Heart [4]. . . . 5
2.2 The Conduction System of the Heart [3]. . . . 6
2.3 The Key Features of an ECG Signal [5]. . . . 6

3.1 (a) 5 Individual Gaussian Functions. (b) Sum of 5 Gaussian Kernels. Adapted from [9]. . . . 8
3.2 The 3D trajectory generated by the EDM. Adapted from [9]. . . . 9
3.3 The MWI and the calculated threshold level. . . . 10
3.4 The phase wrapped ECG signal. . . . 11
3.5 Overview of the EKF Feature Extraction Process (Adapted from [9]) . . . 17
3.6 Example of an optimal hyperplane in an SVM (adapted from [17]) . . . 17

4.1 (a) The MWI, peak detection and phase calculation for a Normal Sinus Rhythm. (b) The phase wrapped ECG signal for a Normal Sinus Rhythm . . . 21
4.2 Raw ECG versus EKF filtered ECG for a Normal Sinus Rhythm . . . 21
4.3 Performance of the a_T state estimate for a Normal Sinus Rhythm . . . 21
4.4 Performance of the b_T state estimate for a Normal Sinus Rhythm . . . 22
4.5 Performance of the θ_T state estimate for a Normal Sinus Rhythm . . . 22
4.6 (a) The MWI, peak detection and phase calculation for Atrial Flutter. (b) The phase wrapped ECG signal for Atrial Flutter . . . 23
4.7 Raw ECG versus EKF filtered ECG for Atrial Flutter . . . 23
4.8 Performance of the a_T state estimate for Atrial Flutter . . . 23
4.9 Performance of the b_T state estimate for Atrial Flutter . . . 24
4.10 Performance of the θ_T state estimate for Atrial Flutter . . . 24
4.11 (a) The MWI, peak detection and phase calculation for Supraventricular Tachyarrhythmia. (b) The phase wrapped ECG signal for Supraventricular Tachyarrhythmia . . . 25
4.12 Raw ECG versus EKF filtered ECG for Supraventricular Tachyarrhythmia . . . 25
4.13 Performance of the a_T state estimate for Supraventricular Tachyarrhythmia . . . 25
4.14 Performance of the b_T state estimate for Supraventricular Tachyarrhythmia . . . 26
4.15 Performance of the θ_T state estimate for Supraventricular Tachyarrhythmia . . . 26
4.16 (a) The MWI, peak detection and phase calculation for Atrial Fibrillation. (b) The phase wrapped ECG signal for Atrial Fibrillation . . . 27
4.17 Raw ECG versus EKF filtered ECG for Atrial Fibrillation . . . 27
4.18 Performance of the a_T state estimate for Atrial Fibrillation . . . 27
4.19 Performance of the b_T state estimate for Atrial Fibrillation . . . 28
4.20 Performance of the θ_T state estimate for Atrial Fibrillation . . . 28

5.1 The raw EKF and a model fit using nonlinear least squares regression (adapted from [9]) . . . 30
5.2 The estimated state variables from a typical ECG signal. (a) Input signal, (b) θ_P, (c) θ_Q, (d) a_R, (e) b_S and (f) b_T (adapted from [9]). . . . 31
5.3 Segmentation results for typical ECG data (adapted from [9]). . . . 32

Chapter 1

Abstract

The purpose of this project was to develop an algorithm that uses the Extended Kalman Filter (EKF) and Support Vector Machines (SVMs), to autonomously interpret ECG data. With the valuable data that the ECG signal captures, information regarding the health of an individuals heart can be monitored and diagnosed using a series of signal processing steps. To properly analyze the ECG data, an ECG Dynamical Model (EDM) was incorporated into the feature extraction component. The algorithm begins with a preprocessing step which detects R-wave peaks followed by a linear phase mapping of heart beats. In addition to this, non-stationary statistics regarding the mean ECG amplitude, mean phase and ECG standard deviation, were calculated. In the segmentation step, the recorded statistics were used to initialize the matrices and vectors for EKF implementation. Once initialization was complete, the nonlinear EDM was linearized and incorporated into the EKF calculations. The EKF essentially estimates the optimal parameters that best fit the model to the observed data, in the least mean square error sense. The SVM then classifies the state estimates using a library of previously trained classifiers, to output a diagnosis for the presented ECG signal. Although results for the EKF step were not accurate enough for SVM implementation, other authors demonstrated exceptional results.

Chapter 2

Introduction

The Electrocardiogram (ECG) is one of the most widely used medical devices in the world today. Its non-invasive feature makes it a very attractive device for monitoring heart activity in humans, along with animals. In addition, it provides cardiologists and paramedics with quality data that can accurately represent the status of a patients heart and provide useful insight into the onset of cardiovascular disease. In addition to monitoring and diagnosis, the invention of the ECG has provided great insight into the structure and function of both healthy and diseased hearts [1]. Furthermore, the ECG has been implemented into miniaturized systems that allow for patients to monitor their heart activity for extended periods of time (called a Holter monitor). Aside from these positive aspects, the ECG machines used in clinical settings are very expensive. 12-lead ECG systems as well as Holter monitors are beyond affordable for people seeking basic monitoring or occasional use. In this case, ECG devices that monitor and diagnose patients are rarely available at affordable costs which motivates engineers and scientists around the world to develop and manufacture house-hold ECG devices that can offer similar functionality at reduced costs.

2.1 The Electrocardiogram and the Heart

The ECG is defined as the measurement of bioelectric potentials as they propagate throughout the heart [2]. The heart is a four chambered pump that delivers de-oxygenated blood to the lungs while at the same time, distributing oxygenated blood to he rest of the body [3]. Figure 2.1 illustrates the various components of the heart and the directions of blood flow. The contractions that occur in the ventricles and atria can be described by the conduction system, shown in Figure 2.2. This system starts with the sinoatrial (SA) node which is composed of cells that do not have a stable resting membrane potential. This results in multiple depolarization episodes that eventually trigger an action potential [3]. Since the SA node generates action potentials that trigger multiple contractions throughout the heart, it is considered to be the body's pacemaker. Once a successful action potential has been generated, it then propagates to both the left and right atria, where a resulting contraction occurs [3]. The action potential then reaches the atrioventricular (AV) node and then propagates to the Bundle of His, which is the only place where action potentials are able to conduct from the atria to the ventricles [3]. Finally, the action potential then spreads to the ventricular myocardium causing contraction of the ventricles. Since the heart activity is dictated by the conduction cycle, it therefore has the ability to reveal important information regarding the condition of a patient's cardiac health.

Since the condition of the heart can be adequately described from the condition of the conduction path, the ECG can be used to record any phenomena that occurs in the heart. The ECG measures the electric potential from the surface of the body and records the movement of dipoles during the conduction cycle. As a result, this gives the recorded signal a characteristic morphology as depicted in Figure 2.3. The ECG can be described by 5 main features: the P, Q, R, S, and T waves. The ECG signal begins with the depolarization of the atria, which creates the characteristic wave known as the P wave [3]. After the onset of the P wave a small delay occurs as the action potential from the SA node travels to the Bundle of His and therefore the ventricular muscle fibres [3]. The ventricular wall then begins to depolarize (contract), producing the Q, R, and S waves that form the characteristic QRS complex [3]. During this time, the atria repolarize (relax) but are usually difficult to detect in an ECG signal due to the masking nature of the QRS complex [3]. Next, the repolarization of the ventricular walls takes place, followed by relaxation to produce the characteristic T wave [3]. Given that the ECG describes the electrical activity found in the heart, it can therefore be used to detect indications of heart disease.

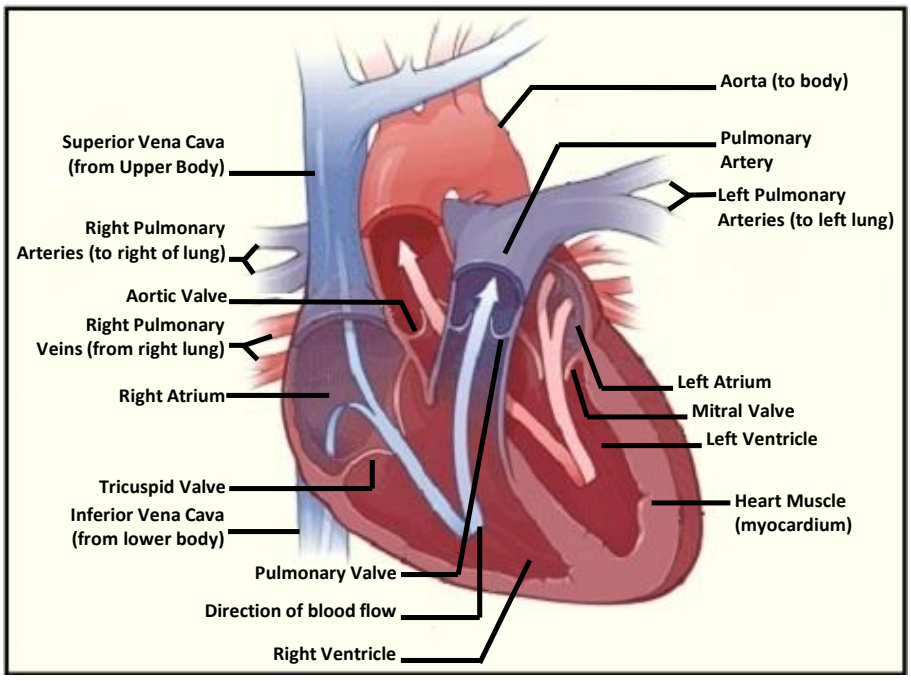

Figure 2.1: The Anatomy of the Heart [4].

To date, diagnosis of heart conditions are primarily executed by trained professionals. They use recorded ECG data and interpret it manually by analyzing the various features that are present, such as wave amplitude, wave width etc [7]. One issue associated with manual ECG interpretation is the risk of human error [7]. Unlike machines, humans are susceptible to fatigue (paramedics), habituation and psychological factors (emotion, reaction to stressful situations etc.), which can affect the interpretation of patient data [7]. On the other hand, automated ECG interpretation devices are immune to these issues. They interpret data by first extracting features which are then used to classify patterns related to heart disease [6]. Using advanced techniques in machine learning, these devices are able to classify data based on training data from a database. Although this method of automated interpretation is immune to human error, it is still limited to the approximations and inaccuracies embedded in the available technologies and algorithms. Thus, automated ECG interpreters can benefit not only patients by increasing the probability of detecting heart conditions much earlier, but medical facilities as it would ease the responsibilities of interpretation by medical professionals.

2.2 Objective

The objective of this project is create a real-time algorithm for diagnosing heart conditions using data acquired from an ECG signal and a database of reference diseased ECGs. By creating a real-time algorithm, it can be incorporated into a small battery powered device (using a microcontroller) and made affordable for everyone. The algorithm will use the Extended Kalman Filter and Support Vector Machine to extract and diagnose data from the ECG signal in a real-time fashion. The real-time feature is an essential component since many medical emergency scenarios require accurate and immediate results.

As mentioned previously, many hospitals are over crowded and under employed with doctors. Thus, by creating a system that can be implemented into a portable device, patients can monitor and diagnose them selves at home, making hospitals and medical clinics less crowded. Since most of the professional ECG devices are extremely expensive, a system like the one proposed would be a cost effective and efficient solution once implemented into a basic microcontroller. People could use this technology to detect the onset of cardiac problems and then schedule appointments for professional examinations. This would therefore decrease the waiting times in many medical

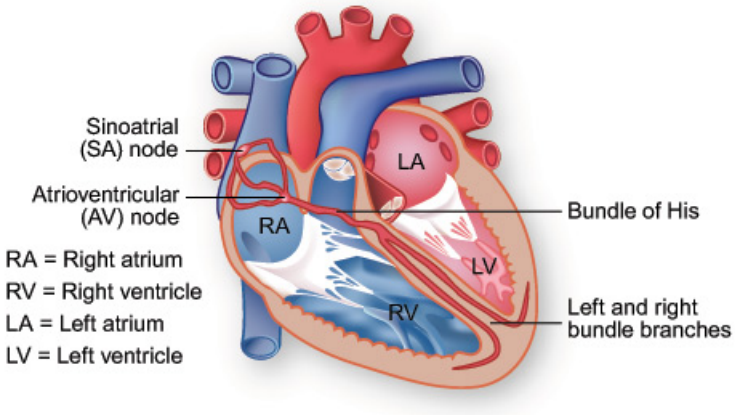

Figure 2.2: The Conduction System of the Heart [3].

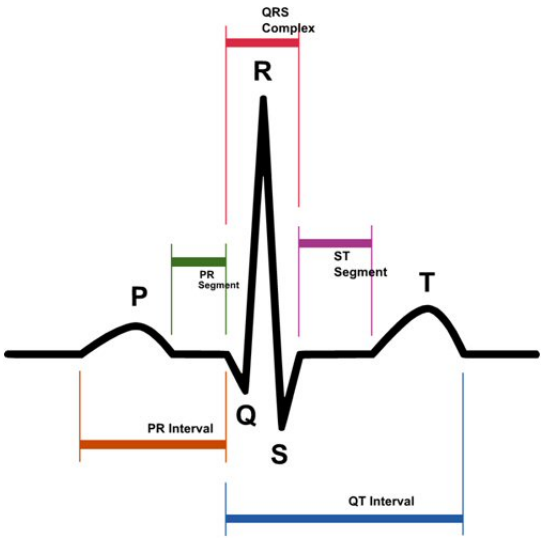

Figure 2.3: The Key Features of an ECG Signal [5].

settings since most of the tedious and interpretative work can be done by the device.

Chapter 3

Methodology

3.1 Overview

The process of taking patient ECG data and providing a diagnosis, requires a series of processing steps. There are 3 main steps to this process: preprocessing, feature extraction and classification. The preprocessing step is one of the most essential components in the analysis of ECG data. All subsequent steps are heavily dependant on the output of this step, as accuracy and speed are key to the real-time function of the algorithm. It essentially filters and transforms raw ECG data to accentuate easy-to-detect features and to reduce noise and artifacts as much as possible. The next step is the feature extraction step. The whole purpose is to use the preprocessed data and to extract information essential for classification of ECG signals. The data that is extracted is primarily referred to as the fiducial points, which represents the peak positions and amplitudes of all 5 waveforms shown in Figure 2.3. In this project, the Extended Kalman Filter will be used for the feature extraction step, in addition to its native filtering abilities. In the last step, the extracted features are used for the classification of heart disease. This component uses SVMs to classify feature data based on a database of trained models. These 3 steps are all iterated in a real-time fashion to produce an efficient and accurate classification of ECG data.

3.2 ECG Dynamical Model

In order to successfully extract features from ECG data, a model of the typical ECG morphology must be known. The model approximates certain morphological features using Gaussian Kernels (GKs) for classifying heart disease. Thus, if the model parameters of an ECG signal are known, then classification can be conducted using the extracted fiducial points. The model approximates ECG data by building the characteristic waves using a series of GKs, described in (3.1).

$$\gamma_i = a_i \exp\left(-\frac{(\phi-\theta_i)^2}{b_i^2}\right), \qquad i \in \{P, Q, R, S, T\} \tag{3.1}$$

The general GK model approximates a realistic ECG signal morphology, by summing various GK functions using (3.1). In this project, the ECG signal was approximated using 5 GKs (one for every characteristic wave), where a_i represents the amplitude, θ_i represents the mean of the GK (phase position), and b_i represents the standard deviation (spread) of the GK. These 5 GKs combine to form an approximation of the ECG signal, as shown in Figure 3.1. The basic assumption was that the GKs were symmetric and that any skewness present in real ECG data was negligible. As well, the onset of any GK was assumed to be 3 times the spread ($3b_i$) from the mean, which approximated a 99% confidence bound [9]. In other literature such as [10], the problem regarding asymmetric Gaussian functions was overcome by introducing additional GKs that would sum with nearby GKs to produce a skewed waveform. For the purpose of simplicity, the 5 GK method was sufficient.

In order to use the ECG Dynamical Model (EDM), the developments upto now need to be applied in a realistic time-varying fashion. The EDM is dependant on not only the morphological components through the GK functions but the time characteristics as well. Since the heart beats in a quasi-periodic manner, it is only natural that the EDM is cyclical and uses the heart rate, ω (in radians), as a parameter. To simulate the noisy and

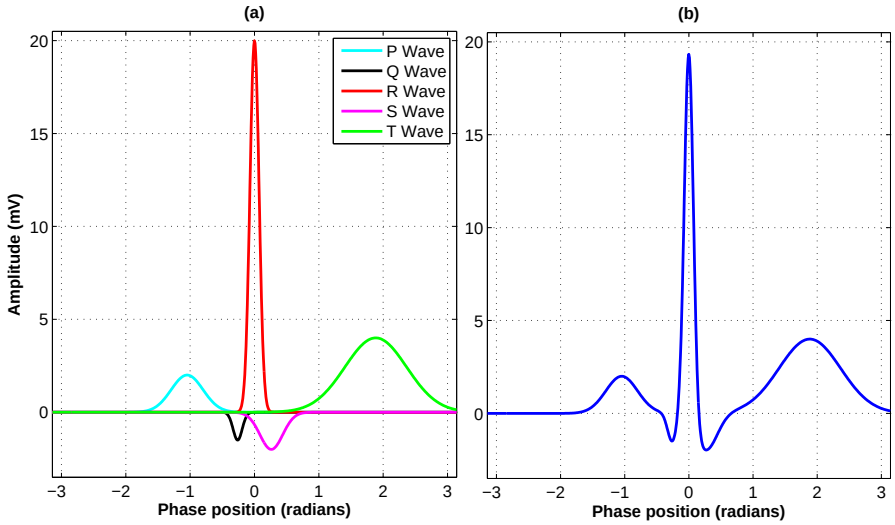

Figure 3.1: (a) 5 Individual Gaussian Functions. (b) Sum of 5 Gaussian Kernels. Adapted from [9].

quasi-periodic behaviour found in realistic ECG data, variables used in the EDM are considered random Gaussian variables. The idea for the proposed synthetic model unifies the morphology and pulse timing in a discrete-time non-linear dynamical model, shown in (3.2) and (3.3) [9].

State Equations:

$$\left.\begin{aligned} \phi_{k+1} &= \phi_k + \omega\delta \\ s_{k+1} &= s_k - \sum_{i\in\{P,Q,R,S,T\}} \delta \frac{a_{i_k}\omega}{b_{i_k}^2} \Delta\theta_{i_k} \exp\left(-\frac{\Delta\theta_{i_k}^2}{2b_{i_k}^2}\right) + \eta \end{aligned}\right\} \tag{3.2}$$

Observation Equations:

$$\left.\begin{aligned} \Phi_k &= \phi_k + v_{\phi_k} \\ z_k &= s_k + v_{s_k} \end{aligned}\right\} \tag{3.3}$$

In (3.2) and (3.3), δ represents the sample time, s_k represents the value of the ECG at time k, η is the random additive white noise that represents baseline wander effects and other sources of process noise, a_{i_k}, b_{i_k} and θ_{i_k} represent the amplitude, spread and location of the Gaussian functions at time k, respectively. Furthermore, the beat-to-beat heart rate, f, can be transformed using $\omega = 2\pi f$. Since the EDM varies with time, the relative cyclic position of the Gaussian functions must be calculated using (3.4).

$$\Delta\theta_{i_k} = (\phi_k - \theta_{i_k}) mod(2\pi) \tag{3.4}$$

As previously mentioned, the ECG signal is quasi-periodic, which means that on each cycle or heart beat, the relative amplitude, spread and position of each GK function is a random Gaussian variable. Thus, the EDM can be thought of as a model that generates a 3D trajectory which contains the circular limit cycle in the polar plane and is pushed up and down as it approaches the centers of each characteristic wave, shown in Figure 3.2. The EDM can be looked at as a phase wrapped representation of the ECG data. Each point in the ECG of a single heart beat is assigned a phase between $-\pi$ and π. This allows for heart beats of different duration to be compared and properly analysed in subsequent steps.

3.3 Preprocessing

The preprocessing step is one of the most important steps for both the feature extraction step and the classification step. The purpose of it is to process the ECG data into a form that allows subsequent processes to function efficiently and with as much accuracy as possible. Simple features in ECG data such as the R-wave peaks

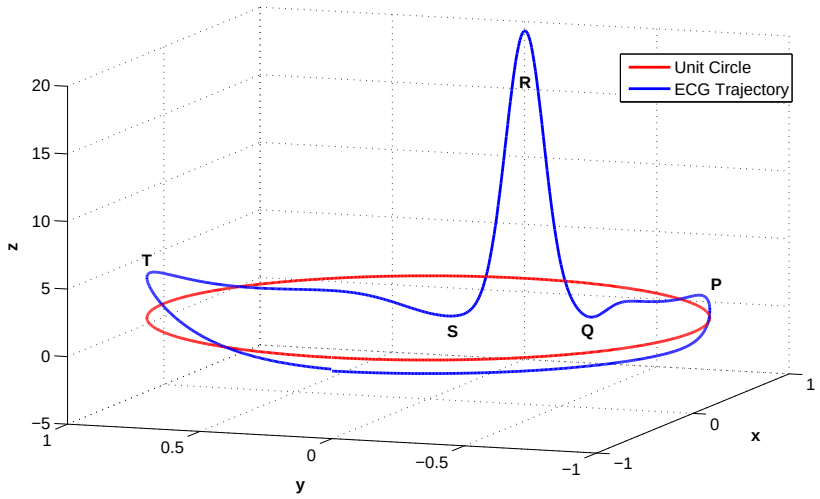

Figure 3.2: The 3D trajectory generated by the EDM. Adapted from [9].

which stand out the most (refer to Figure 2.3), are extracted through a series of transformations which provide a framework for the overall project. There are 2 stages involved in the processing step: the diagnostic stage, and the phase wrapping stage.

The data that is presented to the system is in a circular buffer $\mathbf{z}_k$ of size $T_w F_s$, where T_w is the time duration of the longest possible beat (in this case 3 seconds which corresponds to a heart rate of 20 bpm) and F_s is the sample rate. In order for the EKF step to work, at least one complete heart beat must be present in the buffer at any time, and since 3 seconds is assumed as the longest possible heart beat, the size of the circular buffer is $3F_s$.

Before any evaluation can occur, the ECG signal must be transformed into a useful signal for the remainder of the processes. First, the buffer is differentiated with respect to time, using (3.6). Note that the buffer, shown in (3.5), is an N_w-stage delay line with $N_w + 1$ taps, where $N_w = T_w F_s$ [11].

$$\mathbf{z}_k = [z_k, z_{k-1}, \ldots, z_{k-N_w}]^T \tag{3.5}$$

$$\mathbf{d}_k = \frac{\Delta \mathbf{z}_k}{T_s} = \frac{\mathbf{z}_{k+1} - \mathbf{z}_k}{T_s} \tag{3.6}$$

Then the derivative $\mathbf{d}_k$, of the buffer undergoes a non-linear transformation, that emphasizes high frequency components of the ECG signal [11]. The simplest implementation of this technique, is by simply squaring the derivative, shown in (3.7).

$$\mathbf{q}_k = \mathbf{d}_k^2 = \left(\frac{\Delta \mathbf{z}_k}{T_s} \right)^2 \tag{3.7}$$

The final transformation, called the Moving Window Integrator (MWI), is simply the sum of N_{mwi} most recent samples from the buffer in the non-linear transformation stage, $\mathbf{q}_k$, shown in (3.8) [11]. The size of N_{mwi} must not be too large as it would cause MWI data from adjacent beats to overlap. As well, the size must not be chosen too small since it could lead to noisy behaviour. In this project, $N_{mwi} = 75$ at a sample rate of $T_s = 350Hz$.

$$\mathbf{y}_k = \sum_{i=k-N_{mwi}}^{k} \mathbf{q}_k[i] \tag{3.8}$$

These transformations are the necessary tools needed for any operation and are therefore integrated into each step of this project.

3.3.1 Diagnostic Stage

The whole purpose of the diagnostic stage is to calculate the limits of the transformed data and the threshold. In order to detect the R-wave peaks, a threshold must be set. The maximum and minimum MWI values are found while $t < T_1$ where a value of $T_1 = 5$ seconds was used, and is then incorporated into the calculation of the threshold. In addition, the threshold can be made adaptive, as described in [11]. This calculation is a simple calibration that is made for the subsequent stage. Figure 3.3 demonstrates how the MWI should look, along with the detected R-peak and threshold level, for synthetic ECG data.

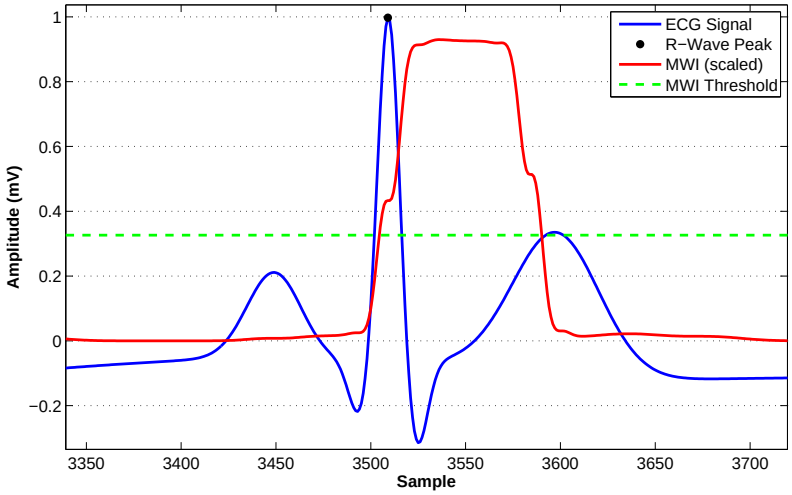

Figure 3.3: The MWI and the calculated threshold level.

3.3.2 Phase Wrapping Stage

In the phase wrapping stage, the threshold value from the previous stage is used to detect R-wave peaks and perform other calculations important to the EKF step. The algorithm works by searching for R-wave peaks in the $\mathbf{z}_k$ vector, only when the MWI signal is greater than the threshold value. This basically narrows the search amongst several noisy and smaller peaks, by concentrating on ECG samples that change quickly in time using values of the MWI that are above the threshold. The basic assumption is that when the previous value of the MWI is below the threshold, and the current value is above, the R-wave peak must be in the vicinity. For data that experiences severe baseline wander, the implementation of an adaptive threshold could result in better performance [11].

Once the R-wave peaks are detected, the system phase wraps the ECG signal. This is done by linearly assigning a value of $-\pi$ to π, to samples in between 2 peaks, with the peaks representing a phase of 0. The algorithm starts by counting the number of samples from one beat to the next. The number of samples, N_s is calculated using (3.9), where $RR_{(b-1),b}$ is the peak-to-peak time interval, between the peak for the current beat, b, and the peak for the previous beat $(b-1)$. Once N_s is calculated, the data samples between the current R-peak, R_b, and previous R-peak, R_{b-1}, are assigned a phase using (3.10)

$$N_s = RR_{(b-1),b} \times F_s \tag{3.9}$$

$$\phi_{k_{(b-1)}-N_s+i} = \left(\frac{2\pi i}{N_s} + \pi\right) mod(2\pi) - \pi, \qquad i = \{1, 2, \ldots, N_s\} \tag{3.10}$$

In this equation, $\phi_{k_{(b-1)}-N_s+i}$ represents the phase at time $k_{(b-1)} - N_s + i$ and $k_{(b-1)}$is the time of the R-peak corresponding to the $(b-1)^{th}$ heart beat. In addition, the size of the data buffer described previously can be explained by the nature of the calculation used in (3.9) and (3.10). Since the algorithm requires 2 R-peaks for the phase calculation, the buffer in (3.5) must be able to hold all the data of at least one long heart beat. This implementation is a semi-real time procedure, since it can only phase wraps the $(b-1)^{th}$ heart beat, once the b^{th}

R-peak has occurred. Figure 3.4 shows an examples of what the phase calculation looks like.

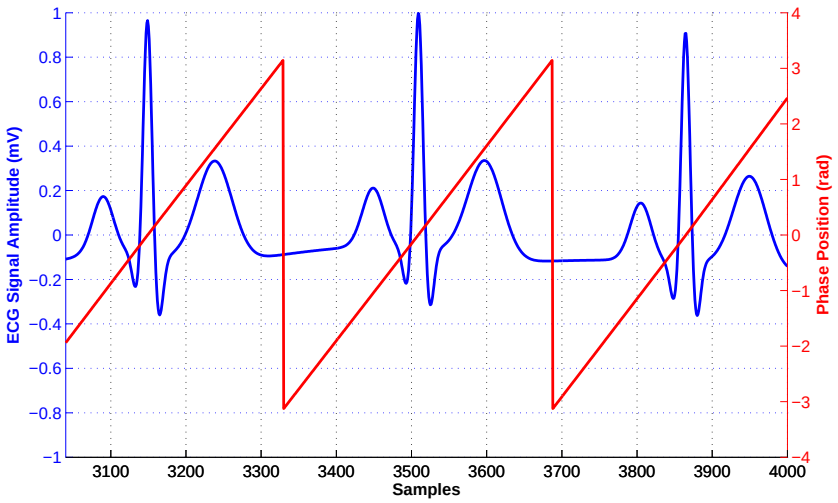

Figure 3.4: The phase wrapped ECG signal.

With the R-peak of a heart beat detected and the phase calculation completed, the algorithm can calculate statistics regarding the ECG signal. There are 3 essential statistics: the mean ECG signal, the mean phase, and the standard deviation (SD). This process is vital for the initialization procedure, described in Section 3.4.2. In order to calculate any of the mentioned statistics, the data for each heart beat must be grouped into *bins*. These bins serve the purpose of collecting ECG data that fits a certain condition. They are assigned the responsibility of storing the mean ECG, mean phase and ECG SD for a certain phase range. For example, if there are β bins and N_s samples in-between $R_{(b-1)}$ and R_b, then each bin, j, would have the responsibility of storing the 3 different statistics for phases between $\frac{2\pi(j-1)}{\beta} - \pi$ and $\frac{2\pi j}{\beta} - \pi$, where $j = \{1, 2, \ldots, \beta\}$.

To calculate the running mean of the ECG signal, (3.11) is used [12]. In this equation, $M_{h-1}(j)$ represents the $(h-1)^{th}$ or old estimate of the mean, $M_h(j)$ is the h^{th} or new estimate and $r_h(j) = x_h(j)$, for each bin j. The mean phase is found in similar manner by using $r_h(j) = \phi_h(j)$. Both variations are initialized using $M_0(j) = 0$.

$$M_h(j) = M_{h-1}(j) + \frac{1}{h}\left(r_h(j) - M_{h-1}(j)\right) \tag{3.11}$$

The calculation for the variance of the ECG signal uses the formulation in (3.12) [13]. As with (3.11), $V_h(j)$ is the current variance, $V_{h-1}(j)$ is the previous variance and $r_h(j) = z_h(j)$, where j is the bin number.

$$V_h(j) = \left(\frac{h-2}{h-1}\right) V_{h-1} + \frac{1}{h}(r_h(j) - M_{h-1}(j))^2 \tag{3.12}$$

To conclude the calculation, SD can be found by simply square rooting the variance. In addition there are 2 special cases: $0 > \frac{N_s}{\beta} \geq 1$ and $\frac{N_s}{\beta} > 1$. The first case guarantees there will be at most 1 sample per bin. This would result in a noisy approximation of the statistics describing the wrapped ECG signal, since some bins could be empty. The second case is the preferred case in which 1 or more samples are guaranteed to be eligible for any bin at any one time. For the purpose of the project, a value of $\beta = 256$ was used, since this would satisfy the preferred relationship for an estimated average heart rate of 60 bpm which corresponds to $N_s = 350$ samples at $F_s = 350Hz$. The calculations in the phase wrapping stage continue while $T_1 < t < T_2$ and are then used in the EKF step.

3.4 EKF for ECG Segmentation

In this section, the calculations and vectors from Section 3.3, are used to filter and extract features from the ECG signal. Following the definition of (3.2) and (3.3), the EDM is revised to suite the feature extraction step by

introducing new states. The 3 statistics from Section 3.3.2 are then used in the non-linear parameter estimation of the new proposed EDM. This is a vital step since the convergence and accuracy of the EKF is heavily dependant on the estimated initial state values. Once the initialization procedure is completed, the EKF begins estimating the hidden states of the system and therefore extracting features.

3.4.1 Modified State Equations

The equations described in the EDM in Section 3.2 only use 2 states to describe the system at any point in time: the phase ϕ_k and the ECG amplitude s_k. In order to have information regarding the peak positions and widths of the characteristic waves, the GK parameters for each wave must be estimated from the ECG signal using the EKF. To do this, the EDM is modified to include first order autoregressive (AR) dynamics describing the GK parameters for each characteristic wave [9]. The modified EDM state equations using AR dynamics are shown in (3.13).

Modified State Equations:

$$\left.\begin{aligned}
\phi_{k+1} &= \phi_k + \omega_k \cdot T_s \\
s_{k+1} &= s_k - \sum_{i=1}^{N} T_s \frac{a_{i_k}\omega_k}{b_{i_k}^2}\Delta\theta_{i_k} \exp\left(-\frac{\Delta\theta_{i_k}^2}{2b_{i_k}^2}\right) + \eta_k \\
a_{1_{k+1}} &= a_{1_k} + u_{1_k} \\
&\vdots \\
b_{1_{k+1}} &= b_{1_k} + u_{(N+1)_k} \\
&\vdots \\
\theta_{1_{k+1}} &= \theta_{1_k} + u_{(2N+1)_k} \\
&\vdots \\
\theta_{N_{k+1}} &= \theta_{N_k} + u_{(3N)_k}
\end{aligned}\right\} \tag{3.13}$$

In (3.13), the new variables correspond to the variables in (3.1), where N is the number of characteristic waves (in this project $N = 5$). The AR noise components, which are $\{u_{1_k}, u_{2_k}, \ldots, u_{(3N)_k}\}$, are considered zero mean Gaussian random variables and represent the variation in parameter values due to the quasi-periodic nature of the heart. Although the state equations were modified, the observation equations in (3.3) remain the same, since the phase and amplitude are the only states that are observable. With the introduction of these new hidden states, the EKF can properly estimate the parameters that approximate the true ECG signal being observed.

3.4.2 Initialization Procedure

In this step, the 3 statistics from Section 3.3.2 are used in conjunction with the equations in (3.2) and (3.3), to initialize the EKF. As mentioned previously, the initialization of the EKF is one of the most important procedures. If the state estimates are inaccurate, the filtering and feature extraction performance of the EKF, could be compromised drastically. Once $t = T_2$, the mean ECG signal is used for the model fitting procedure using a non-linear least squares approach [8]. This procedure initializes the state vector, shown in (3.14), which is a $(3N + 2) \times 1$ vector.

$$\mathbf{x}_k = [\phi_k, s_k, a_{1_k}, \ldots, a_{N_k}, b_{1_k}, \ldots, b_{N_k}, \theta_{1_k}, \ldots, \theta_{N_k}]^T \tag{3.14}$$

The process and observation noise vectors, shown in (3.15) and (3.16) respectively, are critical to the initialization procedure. These zero mean Gaussian random vectors provide the necessary formulations for the covariance matrices that follow. As well, $\mathbf{w}_k$ is a $(3N + 2) \times 1$ vector and $\mathbf{v}_k$ is a 2×1 vector.

$$\mathbf{w}_k = [\omega_k, \eta_k, u_{1_k}, \ldots, u_{3N_k}]^T \tag{3.15}$$

$$\mathbf{v}_k = [v_{\phi_k}, v_{s_k}]^T \tag{3.16}$$

The process and observation noise covariance matrices, shown in (3.17) and (3.18) respectively, represent the variance

between the elements of their corresponding noise vectors [14]. In addition, (3.17) is a $(3N+2)\times(3N+2)$ matrix, and (3.18) is a 2×2 matrix.

$$\mathbf{Q}_k = E[\mathbf{w}_k\mathbf{w}_k^T] \tag{3.17}$$
$$\mathbf{R}_k = E[\mathbf{v}_k\mathbf{v}_k^T] \tag{3.18}$$

In the EKF framework, the noise covariance matrices are considered constant matrices that are initialized at $t = T_2$. Estimates can be defined as being either *prior* or *posterior* estimates. The prior state estimate, $\widehat{\mathbf{x}}_k^-$ shown in (3.19), represents the state estimate at time k using the observations $\mathbf{y}_{k-1}$ to $\mathbf{y}_1$ [8]. The observation vector, $\mathbf{y}_k$ is defined in (3.21) which is basically the vectorized form of (3.3). The equation in (3.20) is the posterior state estimate, $\widehat{\mathbf{x}}_k^+$ which represents the state estimate at time k using the observations $\mathbf{y}_k$ to $\mathbf{y}_1$ [8].

$$\widehat{\mathbf{x}}_k^- = \widehat{\mathbf{x}}_{k|k-1} = E[\mathbf{x}_k|\mathbf{y}_{k-1},\ldots,\mathbf{y}_1] \tag{3.19}$$
$$\widehat{\mathbf{x}}_k^+ = \widehat{\mathbf{x}}_{k|k} = E[\mathbf{x}_k|\mathbf{y}_k,\ldots,\mathbf{y}_1] \tag{3.20}$$
$$\mathbf{y}_k = [\Phi_k, z_k]^T \tag{3.21}$$

Due to noise and model inaccuracies, the observation and state estimate share a difference. This difference, called the *estimate error*, can be defined as

$$\mathbf{e}_k^- = \mathbf{x}_k - \hat{\mathbf{x}}_{k|k-1} \tag{3.22}$$
$$\mathbf{e}_k^+ = \mathbf{x}_k - \hat{\mathbf{x}}_{k|k} \tag{3.23}$$

The prior and posterior state estimate errors correspond to (3.22) and (3.23), respectively. These equations are then used to define the prior and posteriror error covariance matrices in (3.24) and (3.25), respectively, which are both $(3N+2)\times(3N+2)$ matrices. They are initialized as diagonal matrices using the SD ECG and mean ECG statistics from the mean extraction step at $t = T_2$.

$$\mathbf{P}_k^- = \mathbf{P}_{k|k-1} = E[\mathbf{e}_k^-\mathbf{e}_k^{-T}] \tag{3.24}$$
$$\mathbf{P}_k^+ = \mathbf{P}_{k|k} = E[\mathbf{e}_k^+\mathbf{e}_k^{+T}] \tag{3.25}$$

Once these matrices are initialized, the EKF can begin the ECG filtering and segmentation process.

3.4.3 Linearization of EKF Equations

In order for the EDM described in (3.13) and (3.3) to be incorporated into the EKF, a linearization process must occur. The EKF is essentially a nonlinear extension of the Kalman Filter which uses a recursive data processing algorithm to estimate the state of a noisy linear dynamic system [15]. Since the EDM is a nonlinear dynamical model, a linear approximation would have to be formulated in order to work with the KF. The general dynamic model shown in (3.26), can be approximated using (3.27), where f is the state evolution function and g represents the relationship between the state and the observations [8].

$$\left.\begin{aligned}\mathbf{x}_{k+1} &= f(\mathbf{x}_k, \mathbf{w}_k, k)\\ \mathbf{y}_k &= g(\mathbf{x}_k, \mathbf{v}_k, k)\end{aligned}\right\} \tag{3.26}$$

$$\left.\begin{aligned}\mathbf{x}_{k+1} &\approx f(\widehat{\mathbf{x}}_k, \widehat{\mathbf{w}}_k, k) + \mathbf{A}_k(\mathbf{x}_k - \widehat{\mathbf{x}}_k) + \mathbf{F}_k(\mathbf{w}_k - \widehat{\mathbf{w}}_k)\\ \mathbf{y}_k &\approx g(\widehat{\mathbf{x}}_k, \widehat{\mathbf{v}}_k, k) + \mathbf{C}_k(\mathbf{x}_k - \widehat{\mathbf{x}}_k) + \mathbf{G}_k(\mathbf{v}_k - \widehat{\mathbf{v}}_k)\end{aligned}\right\} \tag{3.27}$$

The linearizing matrices used in (3.27) are defined in (3.28) to (3.35) [8]. The $\mathbf{0}$ and $\mathbf{I}$ matrices are the zero and identity matrices, respectively, with their size indicated by the subscript.

$$\mathbf{A}_k = \left.\frac{\partial f(\mathbf{x}_k, \widehat{\mathbf{w}}_k, k)}{\partial \mathbf{x}_k}\right|_{\mathbf{x}_k=\widehat{\mathbf{x}}_k} = \begin{bmatrix} A_{11} & 0 & & \mathbf{0}_{1\times 3N} & \\ A_{21} & A_{22} & \boldsymbol{\Pi}^a_{1\times N} & \boldsymbol{\Pi}^b_{1\times N} & \boldsymbol{\Pi}^\theta_{1\times N} \\ \mathbf{0}_{3N\times 2} & & & \mathbf{I}_{3N\times 3N} & \end{bmatrix} \tag{3.28}$$

$$
\left.
\begin{aligned}
A_{11} &= \frac{\partial \phi_{k+1}}{\partial \phi_k} = 1 \\
A_{21} &= \frac{\partial s_{k+1}}{\partial \phi_k} = -\sum_{i=1}^{N} T_s \frac{a_{i_k}\omega_k}{b_{i_k}^2}\left[1 - \frac{\Delta\theta_{i_k}^2}{b_{i_k}^2}\right]\exp\left(-\frac{\Delta\theta_{i_k}^2}{2b_{i_k}^2}\right) \\
A_{22} &= \frac{\partial s_{k+1}}{\partial s_k} = 1 \\
\mathbf{\Pi}_{1\times N}^{a} &= [\Pi^{a_{1_k}}, \ldots, \Pi^{a_{N_k}}] \\
\mathbf{\Pi}_{1\times N}^{b} &= [\Pi^{b_{1_k}}, \ldots, \Pi^{b_{N_k}}] \\
\mathbf{\Pi}_{1\times N}^{\theta} &= [\Pi^{\theta_{1_k}}, \ldots, \Pi^{\theta_{N_k}}] \\
\Pi^{a_{i_k}} &= \frac{\partial s_{k+1}}{\partial a_{i_k}} = -T_s \frac{\omega_k \Delta\theta_{i_k}}{b_{i_k}^2}\exp\left(-\frac{\Delta\theta_{i_k}^2}{2b_{i_k}^2}\right) \\
\Pi^{b_{i_k}} &= \frac{\partial s_{k+1}}{\partial b_{i_k}} = 2T_s \frac{a_{i_k}\omega_k\Delta\theta_{i_k}}{b_{i_k}^3}\left[1 - \frac{\Delta\theta_{i_k}^2}{2b_{i_k}^2}\right]\exp\left(-\frac{\Delta\theta_{i_k}^2}{2b_{i_k}^2}\right) \\
\Pi^{\theta_{i_k}} &= \frac{\partial s_{k+1}}{\partial \theta_{i_k}} = T_s \frac{a_{i_k}\omega_k}{b_{i_k}^2}\left[1 - \frac{\Delta\theta_{i_k}^2}{b_{i_k}^2}\right]\exp\left(-\frac{\Delta\theta_{i_k}^2}{2b_{i_k}^2}\right)
\end{aligned}
\right\} \tag{3.29}
$$

$$
\mathbf{F}_k = \left.\frac{\partial f(\widehat{\mathbf{x}}_k, \mathbf{w}_k, k)}{\partial \mathbf{w}_k}\right|_{\mathbf{w}_k = \widehat{\mathbf{w}}_k} = \begin{bmatrix} F_{11} & 0 & \mathbf{0}_{2\times 3N} \\ F_{21} & F_{22} & \\ \mathbf{0}_{3N\times 2} & & \mathbf{I}_{3N\times 3N} \end{bmatrix} \tag{3.30}
$$

$$
\left.
\begin{aligned}
F_{11} &= \frac{\partial \phi_{k+1}}{\partial \omega_k} = T_s \\
F_{21} &= \frac{\partial s_{k+1}}{\partial \omega_k} = -\sum_{i=1}^{N} T_s \frac{a_{i_k}\Delta\theta_{i_k}}{b_{i_k}^2}\exp\left(-\frac{\Delta\theta_{i_k}^2}{2b_{i_k}^2}\right) \\
F_{22} &= \frac{\partial s_{k+1}}{\partial \eta_k} = 1
\end{aligned}
\right\} \tag{3.31}
$$

$$
\mathbf{C}_k = \left.\frac{\partial g(\mathbf{x}_k, \widehat{\mathbf{w}}_k, k)}{\partial \mathbf{x}_k}\right|_{\mathbf{x}_k = \widehat{\mathbf{x}}_k} = \begin{bmatrix} \frac{\partial \Phi_k}{\partial \phi_k} & \frac{\partial \Phi_k}{\partial s_k} & \mathbf{\Lambda}_{1\times N}^{a} & \mathbf{\Lambda}_{1\times N}^{b} & \mathbf{\Lambda}_{1\times N}^{\theta} \\ \frac{\partial z_k}{\partial \phi_k} & \frac{\partial z_k}{\partial s_k} & \mathbf{\Gamma}_{1\times N}^{a} & \mathbf{\Gamma}_{1\times N}^{b} & \mathbf{\Gamma}_{1\times N}^{\theta} \end{bmatrix} = \begin{bmatrix} \mathbf{I}_{2\times 2} & \mathbf{0}_{2\times 3N} \end{bmatrix} \tag{3.32}
$$

$$\left.\begin{aligned}
\mathbf{\Lambda}^a_{1\times N} &= [\Lambda^{a_{1_k}},\ldots,\Lambda^{a_{N_k}}] = \mathbf{0}_{1\times N} \quad where, \Lambda^{a_{i_k}} = \frac{\partial \Phi_k}{\partial a_{i_k}} = 0 \\
\mathbf{\Lambda}^b_{1\times N} &= \left[\Lambda^{b_{1_k}},\ldots,\Lambda^{b_{N_k}}\right] = \mathbf{0}_{1\times N} \quad where, \Lambda^{b_{i_k}} = \frac{\partial \Phi_k}{\partial b_{i_k}} = 0 \\
\mathbf{\Lambda}^\theta_{1\times N} &= \left[\Lambda^{\theta_{1_k}},\ldots,\Lambda^{\theta_{N_k}}\right] = \mathbf{0}_{1\times N} \quad where, \Lambda^{\theta_{i_k}} = \frac{\partial \Phi_k}{\partial \theta_{i_k}} = 0 \\
\mathbf{\Gamma}^a_{1\times N} &= [\Gamma^{a_{1_k}},\ldots,\Gamma^{a_{N_k}}] = \mathbf{0}_{1\times N} \quad where, \Gamma^{a_{i_k}} = \frac{\partial z_k}{\partial a_{i_k}} = 0 \\
\mathbf{\Gamma}^b_{1\times N} &= \left[\Gamma^{b_{1_k}},\ldots,\Gamma^{b_{N_k}}\right] = \mathbf{0}_{1\times N} \quad where, \Gamma^{b_{i_k}} = \frac{\partial z_k}{\partial b_{i_k}} = 0 \\
\mathbf{\Gamma}^\theta_{1\times N} &= \left[\Gamma^{\theta_{1_k}},\ldots,\Gamma^{\theta_{N_k}}\right] = \mathbf{0}_{1\times N} \quad where, \Gamma^{\theta_{i_k}} = \frac{\partial z_k}{\partial \theta_{i_k}} = 0 \\
\frac{\partial \Phi_k}{\partial \phi_k} &= 1, \quad \frac{\partial \Phi_k}{\partial s_k} = 0, \quad \frac{\partial z_k}{\partial \phi_k} = 0, \quad \frac{\partial z_k}{\partial s_k} = 1
\end{aligned}\right\} \tag{3.33}$$

$$\mathbf{G}_k = \left.\frac{\partial g(\widehat{\mathbf{x}}_k, \mathbf{v}_k, k)}{\partial \mathbf{v}_k}\right|_{\mathbf{v}_k=\widehat{\mathbf{v}}_k} = \begin{bmatrix} \frac{\partial \Phi_k}{\partial v_{\phi_k}} & \frac{\partial \Phi_k}{\partial v_{s_k}} \\ \frac{\partial z_k}{\partial v_{\phi_k}} & \frac{\partial z_k}{\partial v_{s_k}} \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} = \mathbf{I}_{2\times 2} \tag{3.34}$$

$$\frac{\partial \Phi_k}{\partial v_{\phi_k}} = 1, \quad \frac{\partial \Phi_k}{\partial v_{s_k}} = 0, \quad \frac{\partial z_k}{\partial v_{\phi_k}} = 0, \quad \frac{\partial z_k}{\partial v_{s_k}} = 1 \tag{3.35}$$

Using these equations, the EKF can be fully implemented. The EKF formulation consists of 2 parts: predict and update. The formulation is shown below:

Predict :

$$\widehat{\mathbf{x}}^-_{k+1} = f\left(\widehat{\mathbf{x}}^+_k, \mathbf{w}_k, k\right)\Big|_{\mathbf{w}=\mathbf{0}} \tag{3.36}$$

$$\mathbf{P}^-_{k+1} = \mathbf{A}_k\mathbf{P}^+_k\mathbf{A}^T_k + \mathbf{F}_k\mathbf{Q}_k\mathbf{F}^T_k \tag{3.37}$$

Update :

$$\widehat{\mathbf{x}}^+_k = \widehat{\mathbf{x}}^-_k + \mathbf{K}_k\left[\mathbf{y}_k - g\left(\widehat{\mathbf{x}}^-_k, \mathbf{v}_k, k\right)|_{\mathbf{v}=\mathbf{0}}\right] \tag{3.38}$$

$$\mathbf{K}_k = \mathbf{P}^-_k\mathbf{C}^T_k\left[\mathbf{C}_k\mathbf{P}^-_k\mathbf{C}^T_k + \mathbf{G}_k\mathbf{R}_k\mathbf{G}^T_k\right]^{-1} \tag{3.39}$$

$$\mathbf{P}^+_k = \mathbf{P}^-_k - \mathbf{K}_k\mathbf{C}_k\mathbf{P}^-_k \tag{3.40}$$

When $t > T_2$, the initialized vectors and matrices from Section 3.4.2 are used to update the equations in (3.38) to (3.40). On the next iteration, the EKF updates the linearization matrices $\mathbf{A}_k$,$\mathbf{F}_k$, $\mathbf{C}_k$ and $\mathbf{G}_k$ and then *predicts* the state and the error covariance matrix using the posterior estimates from the previous time. The prior state prediction $\widehat{\mathbf{x}}^-_{k+1}$ is obtained by evaluating (3.13) using the posterior state estimate and process noise vector from the time before. The equations are shown in (3.36) and (3.37).

On every time step, the EKF recursively calculates the error covariance matrix using information from the past and present. This matrix essentially depicts the uncertainties of the estimates using information from the past. The Kalman gain, $\mathbf{K}_k$, incorporates the prior error covariance matrix along with the measurement covariance matrix, to calculate the effective gain on the correction term, shown in square brackets in (3.38). When $\mathbf{P}^-_k$ is very large,

the Kalman gain is also large, which causes the residual, $\mathbf{y}_k - g\left(\widehat{\mathbf{x}}_k^-, \mathbf{v}_k, k\right)|_{\mathbf{v}=\mathbf{0}}$, to be come a stronger contributor to the posterior state estimate. When $\mathbf{P}_k^-$ is small in value, the Kalman gain tends towards a small value as well, resulting in nearly complete confidence in the model. Thus the value of the Kalman gain and therefore the dependency of the residual, grow with increasing uncertainty of the estimates. This recursive process, estimates the posterior state values by varying the effect the measurements and prior state estimates.

On every heart beat, the EKF uses the previous state estimates to calculate the lineaization matrices, the prediction equations and the update equations. Since the EKF estimates the GK parameters, the fiducial points from the ECG signal are extracted on a beat-to-beat basis. However, the estimated GK parameters, are not a stationary value, but instead vary through out the heart beat. In the EKF framework, there are 2 types of signal samples: *in-kernel* samples and *out-kernel* samples [9]. When the observation signal forms any part of the N waves, it is said to be an in-kernel sample for the i^{th} Gaussian kernel. For example, if the signal being observed is that of the P wave, then that observation is an in-kernel sample for $i = P$. When the observation signal does not correspond to the Gaussian kernel in question, it is considered an out-kernel sample. In the same example, the observation is an in-kernel sample for $i = P$, but is an out-kernel sample for $i = \{Q, R, S, T\}$. Since the behaviour of the ECG signal is quasi-periodic, the state estimates fluctuate rapidly at the onset of an in-kernel sample, and can fluctuate very slowly during out-kernel samples (quasi-constant) [9]. Thus, the onset and offset of the in-kernel samples are found through *fluctuation extraction*, where the zero-crossings corresponding to the slope signals (the derivative of the estimate) are detected. Using (3.41), the correct onset and offset of each GK can be found, where f_i^{on} is the time of the onset of the i^{th} GK and f_i^{off} is the time of the offset [9]

$$\begin{aligned} P_{on} &= min(\theta_P - 3b_P, f_P^{on}), & P_{off} &= max(\theta_P + 3b_P, f_P^{off}) \\ QRS_{on} &= min(\theta_Q - 3b_Q, f_Q^{on}), & QRS_{off} &= max(\theta_S + 3b_S, f_S^{off}) \\ T_{on} &= min(\theta_T - 3b_T, f_T^{on}) & T_{off} &= max(\theta_T + 3b_T, f_T^{off}) \end{aligned} \tag{3.41}$$

Finally, once the fiducial points are obtained they are used to calculate the features associated with the ECG signal for the current heart beat. Table 3.1 summarizes the relevance of each feature and their corresponding extraction formulas.

Table 3.1: Chart of ECG Extracted Features using the Gaussian kernel parameters (Adapted from [9])

Feature	**Relevance to Fiducial Points**	**Extraction Formula**
P_{amp}	Amplitude of the ECG signal at P_{peak}	a_P
QRS_{amp}^-	Minimum amplitude of the ECG signal at Q_{peak}, R_{peak}, S_{peak}	$min(a_Q, a_R, a_S)$
QRS_{amp}^+	Maximum amplitude of the ECG signal at Q_{peak}, R_{peak}, S_{peak}	$max(a_Q, a_R, a_S)$
T_{amp}	Amplitude of the ECG signal at P_{peak}	a_T
P_{dur}	Difference between the onset and offset of the P wave	$P_{off} - P_{on}$
QRS_{dur}	Difference between the onset of the Q wave and offset of the S wave	$S_{off} - Q_{on}$
T_{dur}	Difference between the onset and offset of the T wave	$T_{off} - T_{on}$
RR_{int}	Difference between the peak location of two successive R wave	$R'_{peak} - R_{peak}$
QT_{int}^P	Difference between the peak locations of the Q wave and T wave	$T_{peak} - Q_{peak}$
QT_{int}	Difference between the peak location of the Q wave and offset of T wave	$T_{off} - Q_{peak}$
TP_{int}	Difference between the peak locations of the T wave and next P wave	$P'_{peak} - T_{peak}$

To conclude the implementation of the EKF in ECG segmentation, Figure 3.5 summarizes the process described in this section and how they relate to each other.

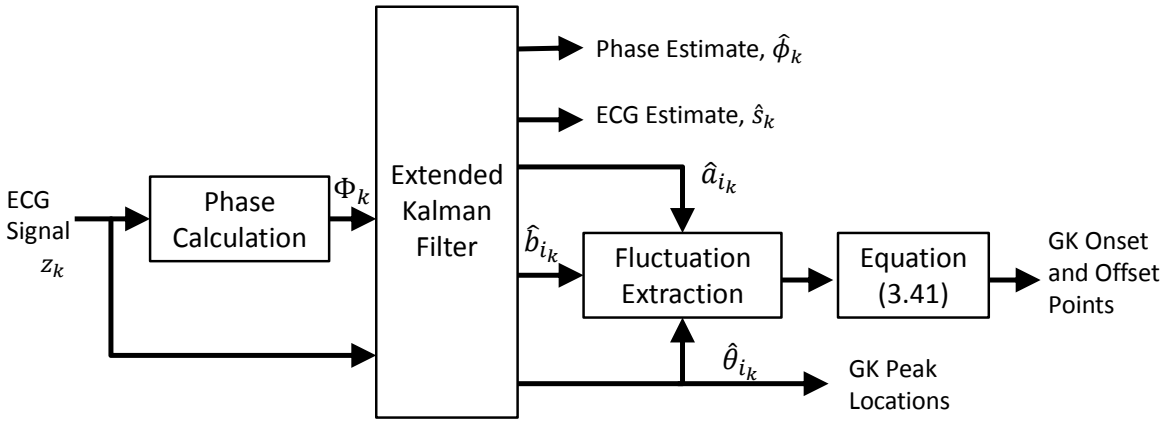

Figure 3.5: Overview of the EKF Feature Extraction Process (Adapted from [9])

3.5 Support Vector Machines

3.5.1 Theory and Application

With the features extracted from the ECG signal in Section 3.4, the SVM can classify the data for heart disease. The SVM is essentially a classifier that uses an $(n-1)$-dimensional hyperplane to separate a n-dimensional feature vector into different classes [14]. The SVM determines the optimal position and orientation for the $(n-1)$-hyperplane, that maximizes the distance to the nearest data point of each class (also known as *support vectors*) [16]. The SVM is a binary classifier in which one class is assigned a value of $y_i = +1$ and the other a value of $y_i = -1$. Figure 3.6 illustrates the classification of linearly separable data.

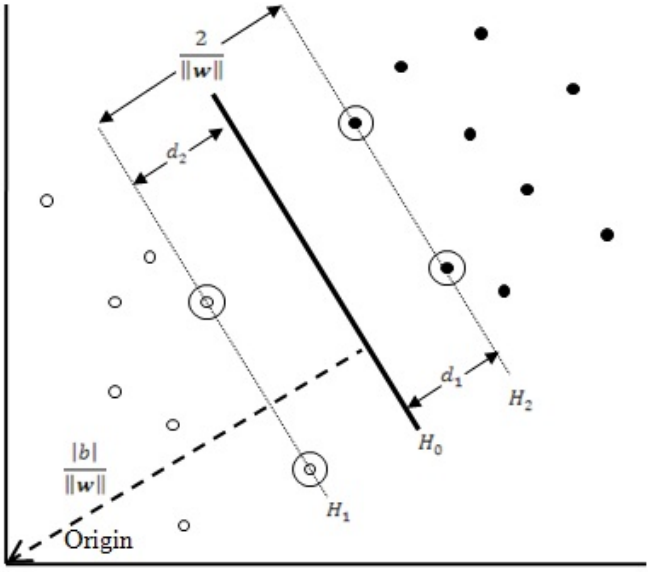

Figure 3.6: Example of an optimal hyperplane in an SVM (adapted from [17])

The procedure for implementing a SVM consists of a training phase and execution phase. The training phase is completed off-line, and consists of using labelled training data to train the SVM to classify data with as little error as possible. If the training data set of size m is defined as

$$D = \{(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), (\mathbf{x}_3, y_3), \ldots, (\mathbf{x}_m, y_m)\}, \qquad x \epsilon \Re^n, y \epsilon \{-1, 1\} \tag{3.42}$$

where $\mathbf{w}$ is the $(n \times 1)$ weight vector, $\mathbf{x}$ is the $(n \times 1)$ feature vector and y is the corresponding class label, then the optimal separating hyperplane can be described by (3.43) [17].

$$\mathbf{w}^T\mathbf{x} + b = 0 \tag{3.43}$$

The classification for the positive class, $y_i = +1$, and the negative class, $y_i = -1$, can be described using the equations in (3.44) and (3.45), respectively, where x_i and y_i correspond to the i^{th} input data pair from (3.42).

$$\mathbf{w}^T\mathbf{x}_i + b \geq +1 \qquad y_i = +1,\ i\epsilon\{1, 2, \ldots m\} \tag{3.44}$$

$$\mathbf{w}^T\mathbf{x}_i + b \leq -1 \qquad y_i = -1,\ i\epsilon\{1, 2, \ldots m\} \tag{3.45}$$

Alternatively, the previous two equations can be combined using (3.46).

$$y_i(\mathbf{w}^T\mathbf{x}_i + b) - 1 \geq 0 \qquad i\epsilon\{1, 2, \ldots m\} \tag{3.46}$$

The SVM essentially calculates the optimal hyperplane by finding the parameters $\mathbf{w}$, b and α that maximize the margin of size $\frac{2}{\|w\|}$. The optimization is done by minimizing the Lagrangian function, shown in (3.47), with respect to $\mathbf{w}$, b and maximizing with respect to $\alpha_i \geq 0$, where α_i is a positive Lagrangian multiplier [17].

$$L(\mathbf{w}, b, \alpha) = \frac{1}{2}\| \mathbf{w} \|^2 - \sum_{i=1}^{m} \alpha_i \left[y_i(\mathbf{w}^T\mathbf{x}_i + b) - 1 \right] \tag{3.47}$$

The minimum of $L(\mathbf{w}, b, \alpha)$ with respect to $\mathbf{w}$ and b, are shown in (3.48) and (3.49) [17].

$$\frac{\partial L}{\partial \mathbf{w}} = 0 \quad \Longrightarrow \quad \mathbf{w} = \sum_{i=1}^{m} \alpha_i \mathbf{x}_i y_i \tag{3.48}$$

$$\frac{\partial L}{\partial b} = 0 \quad \Longrightarrow \quad \sum_{i=1}^{m} \alpha_i y_i = 0 \tag{3.49}$$

The optimization is made easier by transforming the primal problem in (3.47) to its corresponding dual problem, which is defined using (3.50) [16].

$$\max_{\alpha} W(\alpha) = \max_{\alpha} \{ \min_{\mathbf{w}, b} L(\mathbf{w}, b, \alpha) \} \tag{3.50}$$

By using (3.47) to (3.50), the optimal Lagrangian multipliers that solve the dual problem are defined in (3.51), with the constraints in (3.52) and (3.53) [16].

$$\boldsymbol{\alpha}^* = \arg\min_{\alpha} \frac{1}{2} \sum_{i=1}^{m} \sum_{j=1}^{m} \alpha_i \alpha_j y_i y_j \mathbf{x}_i^T \mathbf{x}_i - \sum_{i=1}^{m} \alpha_i \tag{3.51}$$

$$\alpha_i \geq 0, \quad i = \{1, 2, \ldots, m\} \tag{3.52}$$

$$\sum_{i=1}^{m} \alpha_i y_i = 0 \tag{3.53}$$

Since the patterns associated with the ECG morphology are generally nonlinear, the patterns are not linearly separable. To solve this problem, the feature vectors are mapped to a higher dimensional feature space, thereby exploiting the separability in higher dimensions. This is done by applying the following transformation to the feature vector:

$$\mathbf{x} \longrightarrow \Phi(\mathbf{x}) \tag{3.54}$$

where,

$$\begin{aligned} K(\mathbf{x}_a, \mathbf{x}_b) &= \Phi(\mathbf{x}_a)^T \Phi(\mathbf{x}_b) \\ &= \exp\left(-\frac{\| \mathbf{x}_a - \mathbf{x}_b \|^2}{2\sigma^2} \right) \end{aligned} \tag{3.55}$$

The function $K(\mathbf{x}_a, \mathbf{x}_b)$ is the Gaussian Radian Basis Function (RBF) kernel, and is among many other kernels, such as the linear, quadratic and polynomial kernels [17]. Using the nonlinear mapping in (3.54) on (3.51), the optimal Lagrangian multiplier for the nonlinear case can be described by (3.56), in conjunction with (3.52) and (3.53) [16].

$$\begin{aligned}\boldsymbol{\alpha}^* &= \arg\min_{\alpha} \frac{1}{2}\sum_{i=1}^{m}\sum_{j=1}^{m}\alpha_i\alpha_j y_i y_j \Phi(\mathbf{x}_i)^T\Phi(\mathbf{x}_j) - \sum_{i=1}^{m}\alpha_i \\ &= \arg\min_{\alpha} \frac{1}{2}\sum_{i=1}^{m}\sum_{j=1}^{m}\alpha_i\alpha_j y_i y_j K(\mathbf{x}_i, \mathbf{x}_j) - \sum_{i=1}^{m}\alpha_i \end{aligned} \tag{3.56}$$

The optimization problem presented in (3.56) is a quadratic programming optimization problem. Several algorithms exist for finding the optimal Lagrange multipliers, such as the Sequential Minimal Optimization (SMO) algorithm. It is important to note that data points for which $\alpha_i > 0$ are called *Support Sectors* (SVs) and lie on one of the two boundaries, while non-support vectors have a value of $\alpha_i = 0$ [17]. Thus, the SVM creates a classification model using training points and only retaining the important ones (support vectors), since non-support vectors are assigned a value of $\alpha = 0$. Once the optimal Lagrange multipliers are calculated using the SMO optimization method, the execution phase of the SVM can begin. Using α_i and the support vectors from the training set, the optimal separating hyperplane can be defined as

$$\mathbf{w}^* = \sum_{i \epsilon SVs} \alpha_i^* y_i \Phi(\mathbf{x}_i) \tag{3.57}$$

and the bias can be calculated using support vectors, as shown in (3.58), where $\mathbf{x}^+$ is any SV from the positive class ($y_i = +1$) and $\mathbf{x}^-$ is any SV from the negative class ($y_i = -1$) [17].

$$b^* = -\frac{1}{2}\sum_{i \epsilon SVs} \alpha_i^* y_i \left[K(\mathbf{x}_+, \mathbf{x}_i) + K(\mathbf{x}_-, \mathbf{x}_i)\right] \tag{3.58}$$

The overall classification of a feature vector $\mathbf{x}$, is defined in (3.60), where the explicit calculation of $\mathbf{w}^*$ is avoided through the use of the kernel function (also known the as *kernel trick*) [17]. Thus, once α_i^* and b^* are calculated, the classification of the state estimate can begin.

$$f(\mathbf{x}) = sign\left(\sum_{i \epsilon SVs} \mathbf{w}^{*T}\Phi(\mathbf{x}) + b^*\right) \tag{3.59}$$

$$= sign\left(\sum_{i \epsilon SVs} \alpha_i^* y_i K(\mathbf{x}_i, \mathbf{x}) + b^*\right). \tag{3.60}$$

where

$$sign(a) = \begin{cases} 1 & \text{if } a \geq 0 \\ -1 & \text{if } a < 0 \end{cases} \tag{3.61}$$

3.5.2 Decision Method

In terms of heart disease classification, a library of Lagrange multipliers, biases and support vectors are used to diagnose patient ECG data. The binary classifier described in Section 3.5.1 is trained using various labelled ECG examples that are both diseased and normal. Additional information regarding local heart rates and higher order statistics could be used as features for the SVM classifier. By using (3.56), the SVM calculates the optimal Lagrange multiplier for each diseased ECG that is presented and compares it to all other ECG signals. Once the library or models of diseased and normal ECG signals is constructed, the SVM can finally be used for the execution or classification stage.

When the state estimates of an ECG signal are input into the trained SVM, a series of classifications must occur in order to properly diagnose the patient amongst the many diseases. This is considered a multi-classification problem, where more than one class or disease is used in the decision criteria. One method of multi-classification is the One-against-all (OvA) method, where an N-class problem is divided into N two-class SVM classifiers. The

i^{th} SVM classifier is trained with labelled samples from the i^{th} class (which are positive examples) and samples from all the other classes (which are negative examples) [18]. When a state estimate is presented to this system, it is classified by each of the N-SVM classifiers, and a tally of each classifier is recorded. As more state estimates are input, the system counts the number of correct classifications for each classifier. After a certain number of test heart beats, the system diagnoses the patient by providing the percentage of correct classification for each disease in the library. This would therefore provide the patient with their probability of having a certain heart disease.

Chapter 4

Results

4.1 Normal Sinus Rhythm

The result for the preprocessing step for normal ECG data, is demonstrated in Figure 4.1. The results from the EKF step, are shown in Figures 4.2 to 4.5. Since estimation functions equally among the different kernels, only state estimates for the T wave were shown.

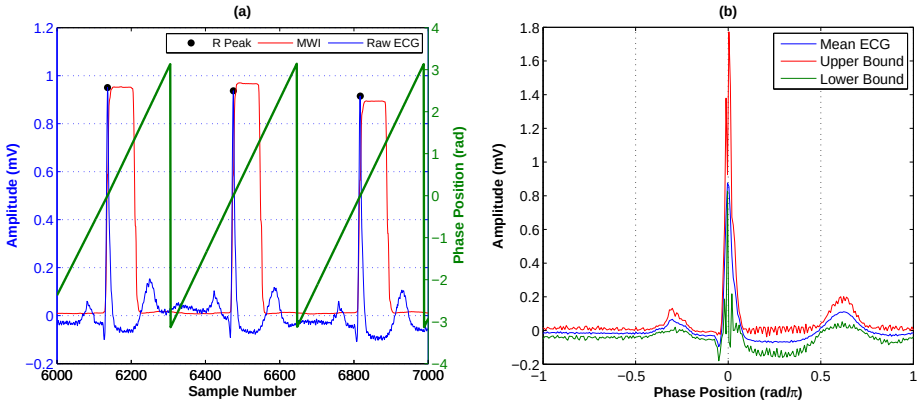

Figure 4.1: (a) The MWI, peak detection and phase calculation for a Normal Sinus Rhythm. (b) The phase wrapped ECG signal for a Normal Sinus Rhythm

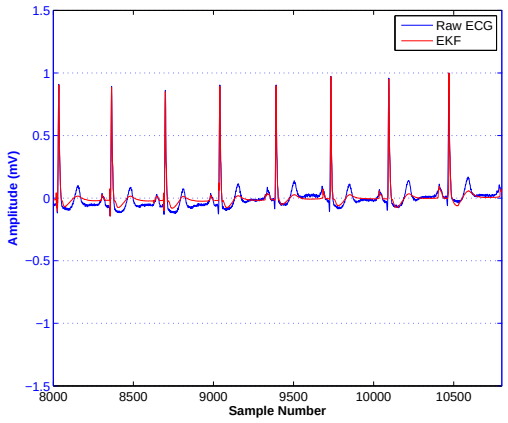

Figure 4.2: Raw ECG versus EKF filtered ECG for a Normal Sinus Rhythm

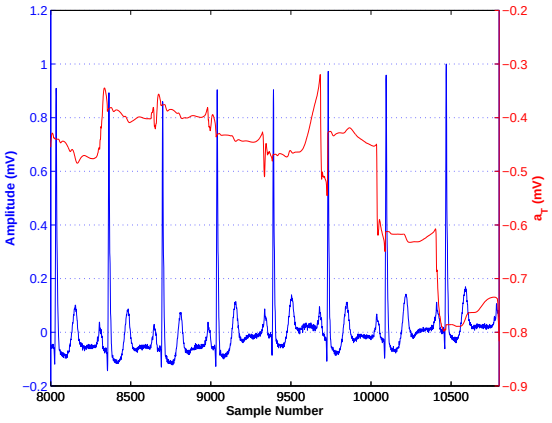

Figure 4.3: Performance of the a_T state estimate for a Normal Sinus Rhythm

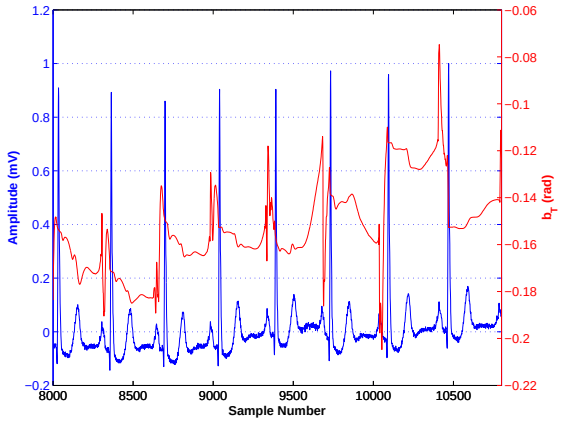

Figure 4.4: Performance of the b_T state estimate for a Normal Sinus Rhythm

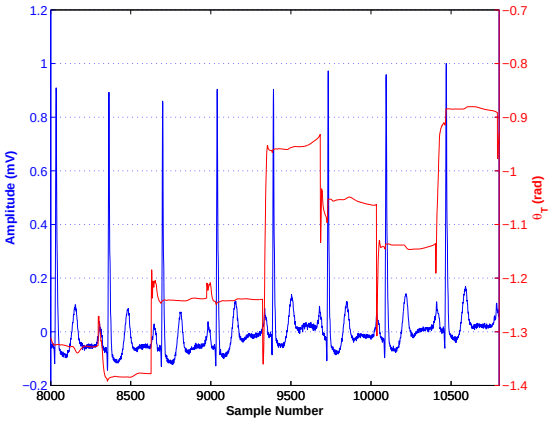

Figure 4.5: Performance of the θ_T state estimate for a Normal Sinus Rhythm

4.2 Atrial Flutter

The result for the preprocessing step for ECG data from a patient with Atrial Flutter, is demonstrated in Figure 4.6. The results from the EKF step, are shown in Figures 4.7 to 4.10. Since estimation functions equally among the different kernels, only state estimates for the T wave were shown.

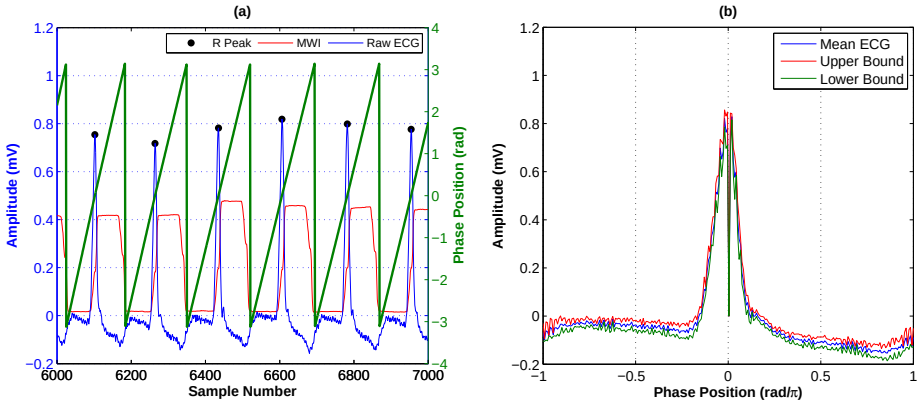

Figure 4.6: (a) The MWI, peak detection and phase calculation for Atrial Flutter. (b) The phase wrapped ECG signal for Atrial Flutter

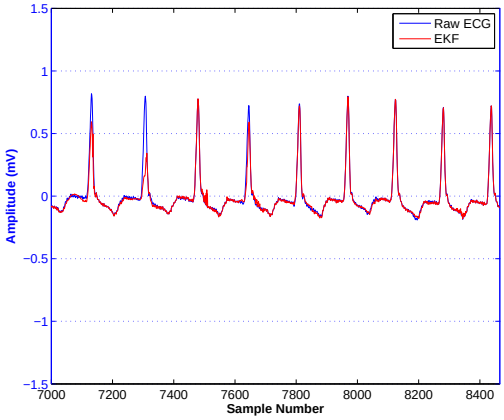

Figure 4.7: Raw ECG versus EKF filtered ECG for Atrial Flutter

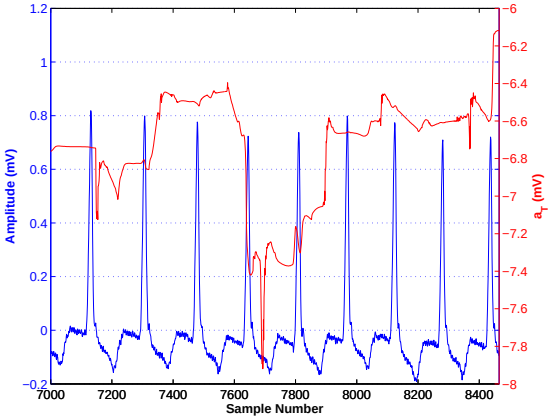

Figure 4.8: Performance of the a_T state estimate for Atrial Flutter

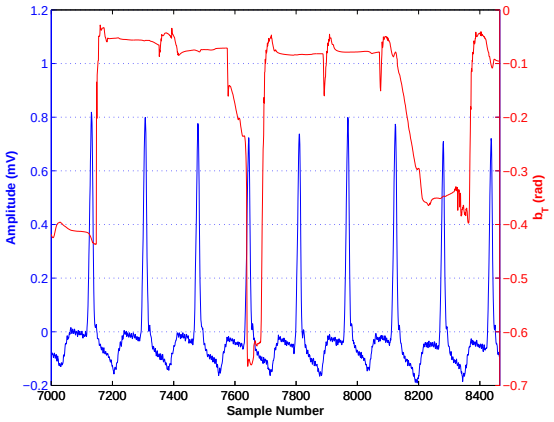

Figure 4.9: Performance of the b_T state estimate for Atrial Flutter

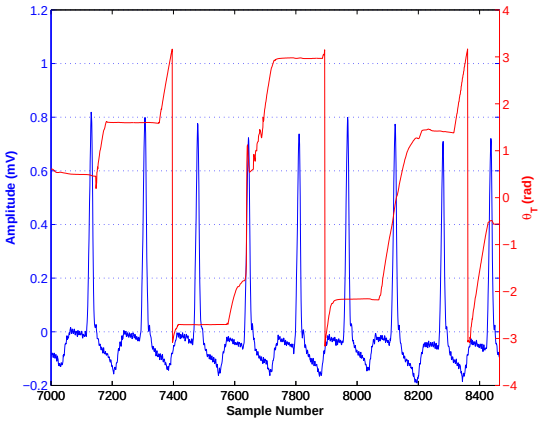

Figure 4.10: Performance of the θ_T state estimate for Atrial Flutter

4.3 Supraventricular Tachyarrhythmia

The result for the preprocessing step for ECG data from a patient with Supraventricular Tachyarrhythmia, is demonstrated in Figure 4.11. The results from the EKF step, are shown in Figures 4.12 to 4.15. Since estimation functions equally among the different kernels, only state estimates for the T wave were shown.

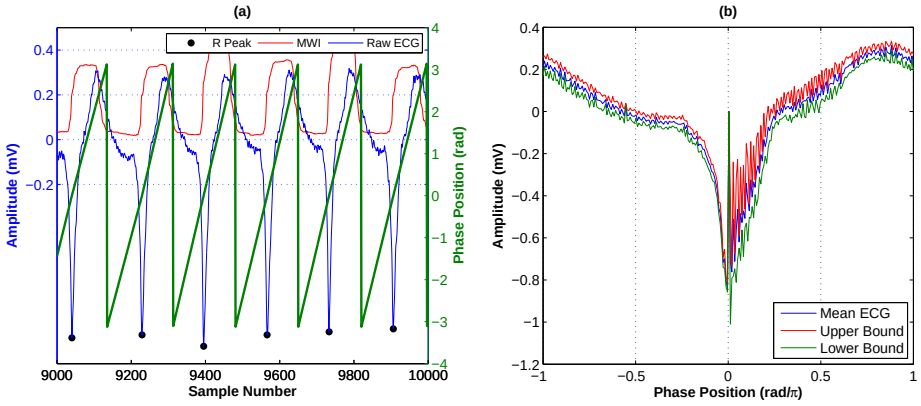

Figure 4.11: (a) The MWI, peak detection and phase calculation for Supraventricular Tachyarrhythmia. (b) The phase wrapped ECG signal for Supraventricular Tachyarrhythmia

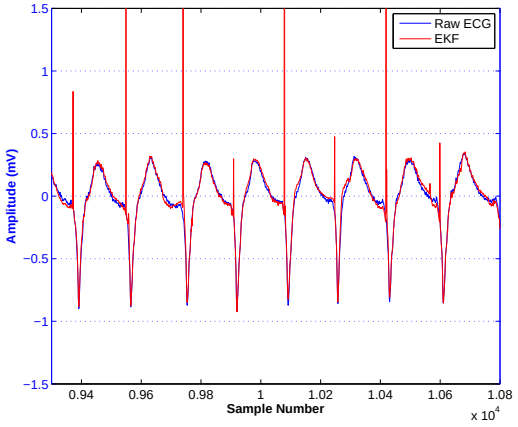

Figure 4.12: Raw ECG versus EKF filtered ECG for Supraventricular Tachyarrhythmia

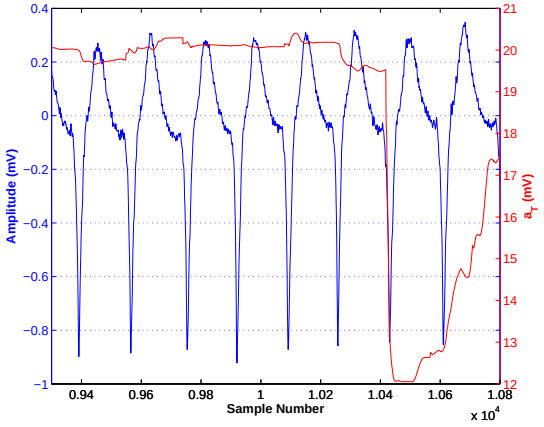

Figure 4.13: Performance of the a_T state estimate for Supraventricular Tachyarrhythmia

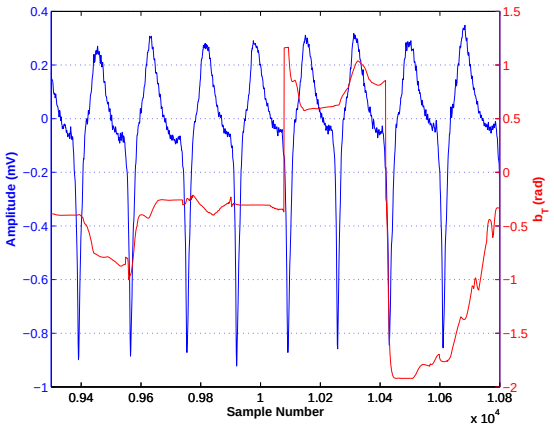

Figure 4.14: Performance of the b_T state estimate for Supraventricular Tachyarrhythmia

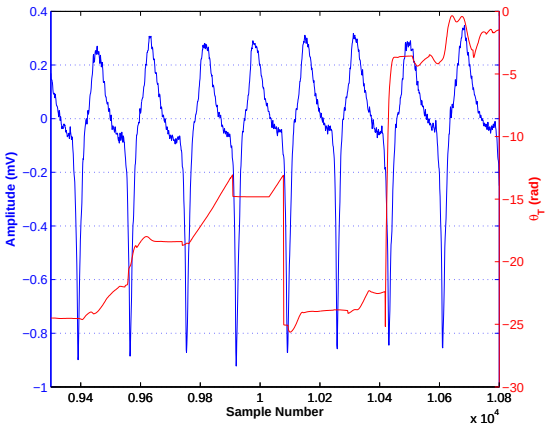

Figure 4.15: Performance of the θ_T state estimate for Supraventricular Tachyarrhythmia

4.4 Atrial Fibrillation

The result for the preprocessing step for ECG data from a patient with Atrial Fibrillation, is demonstrated in Figure 4.16. The results from the EKF step, are shown in Figures 4.17 to 4.20. Since estimation functions equally among the different kernels, only state estimates for the T wave were shown.

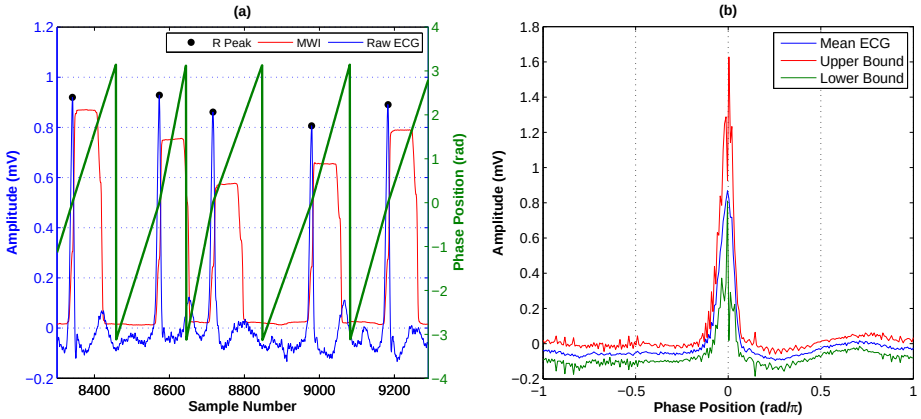

Figure 4.16: (a) The MWI, peak detection and phase calculation for Atrial Fibrillation. (b) The phase wrapped ECG signal for Atrial Fibrillation

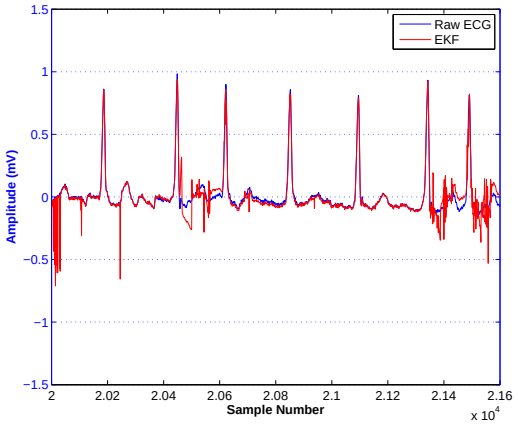

Figure 4.17: Raw ECG versus EKF filtered ECG for Atrial Fibrillation

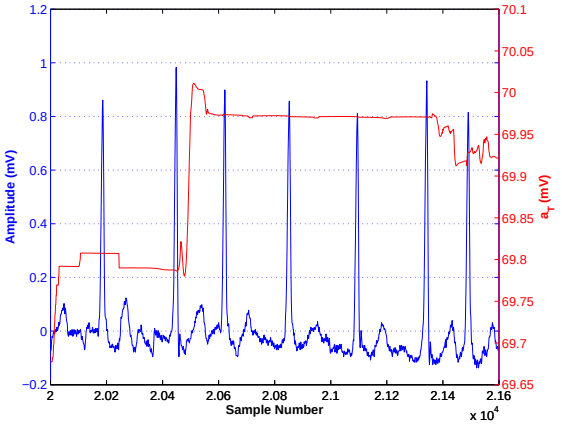

Figure 4.18: Performance of the a_T state estimate for Atrial Fibrillation

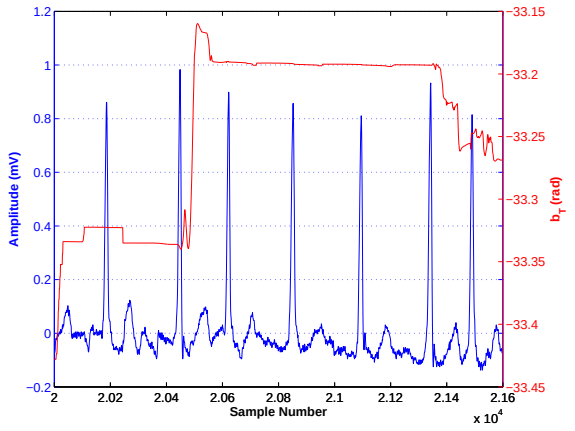

Figure 4.19: Performance of the b_T state estimate for Atrial Fibrillation

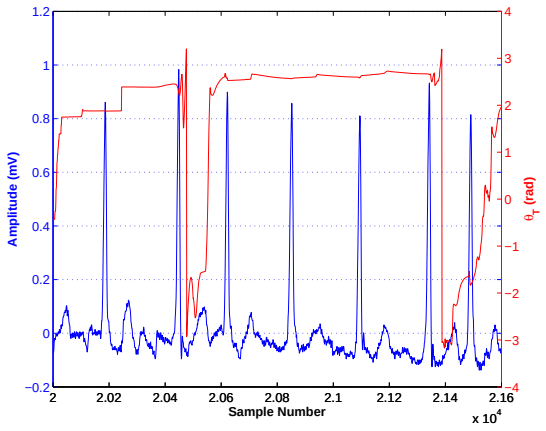

Figure 4.20: Performance of the θ_T state estimate for Atrial Fibrillation

Chapter 5

Discussion and Conclusion

One of the most successful portions of this project, was the preprocessing stage. As mentioned in Section 3.3, the whole purpose of this stage was to detect R wave peaks and to generate linear phase assignments to intra-beat samples. By referring to Figures 4.1(a), 4.6(a), 4.11(a) and 4.16(a), it is clear that the phase calculations and peak detections were completed successfully. With the assumption of negligible baseline wander, the algorithm still performed relatively well on ECG samples with strong baseline wander. In low SNR conditions, the algorithm failed to properly extract R-peak waves and therefore calculate the phase. To extend this algorithm for more distorted signals, the implementation of a simple bandpass filter to attenuate baseline wander and high frequency noise, as well as an adaptive MWI threshold would improve the peak detection performance in the preprocessing step.

In the phase wrapping component, the algorithm successfully mapped linear phase assignments to sample between adjacent R peaks. Using this, the algorithm was able to collect essential ECG statistics necessary for the EKF step, as shown in Figures 4.1(b), 4.6(b), 4.11(b) and 4.16(b). The issue with the general preprocessing step, is that a delay between the time ECG data is presented to the system and the time ECG segmentation begins, introduces a delayed response of the entire system. The diagnostic stage requires $T_1 = 5$ seconds to properly initialize the system, and the phase wrapping component requires $T_2 = 10$ seconds to collect enough statistics. With an overall delay of 15 seconds, medical practitioners may be discouraged from using this algorithm during emergency situations, but can generally be considered negligible. In addition to delays, the data buffers must be very large to account for extremely low heart rate scenarios. Since the phase calculations occur at the conclusion of a heart beat, any samples of data presented to the EKF that have not been phase wrapped, would create errors in the estimation process. To combat this issue, the size of the buffer was made to hold at least all the data for the longest possible heart rate, which was assumed 20 bpm (or 3 seconds of data). Aside from the signal delays, the overall functioning of the phase wrapping component was successfully implemented in a real-time fashion.

In the EKF implementation, the incorrect initialization procedure was observed to be detrimental to the segmentation process. The whole purpose of the EKF initialization procedure, was to estimate the parameters that model and best fit the mean ECGs in Figures 4.1(b), 4.6(b), 4.11(b) and 4.16(b). In this project, the implementation of a nonlinear least squares regression on the mean ECG was not successful. The parameters modelling the mean ECGs of each disease were estimated manually and generalized using the data from Figure 4.1(b). This simple yet catastrophic approximation, resulted in poor filtering performance for the input ECG data, as shown for Figures 4.2, 4.7, 4.12 and 4.17. The poor performance can be assessed qualitatively, by observing the EKF output as it follows the unfiltered data and introduces noisy peaks. The state estimates of the GK parameters were as well corrupted, as shown in the remaining figures. As an example, the θ_T estimates for a normal sinus rhythm, shown in Figure 4.5, contains noisy peaks as well as negative values, that do not describe the T wave kernel parameters appropriately. The sudden jumps in the waveform (in red) should occur around the T waves and not the other characteristic waves. In addition, by observing Figure 4.1(b) it can be seen that the T wave occurs near the 0.625 rad/π mark which is positive compared to the estimates. Similar errors occur in diseased ECG signals, which solidifies the important concept of correct initialization. Furthermore, the application of the SVM component was not achieved since the errors in the estimates were extremely large.

The idea that the EKF can extract features from an ECG signal, was successfully accomplished by Sayadi and Shamsollahi in [9]. They successfully implemented a non-linear least squares regression on mean ECG data, as shown in Figure 5.1. Using correctly fitted mean data, the group was able to properly initialize $\mathbf{P}_k$, $\mathbf{Q}_k$ and $\mathbf{R}_k$

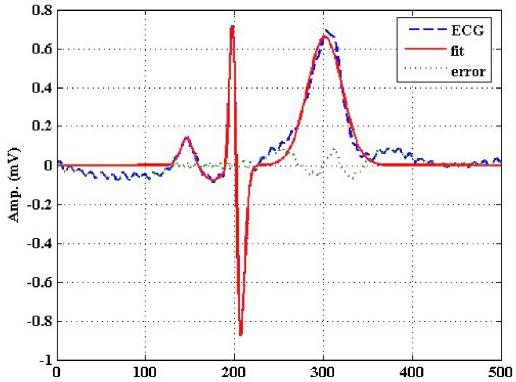

Figure 5.1: The raw EKF and a model fit using nonlinear least squares regression (adapted from [9])

as well the linearizing matrices $\mathbf{A}_k$ and $\mathbf{F}_k$. After using the equations (3.36) to (3.25) along with the fluctuation extraction step in (3.41) and Table 3.1, they obtained the results shown in Figure 5.2. At the conclusion of their research, they obtained the fiducial points of all the characteristic waves, shown in Figure. These figures indicate that feature extraction from ECG data using the EKF is possible by ensuring adequate initialization and following the correct implementation details. These results could then be fed into SVM classifiers to diagnose patient ECG data.

The current implementation requires several improvements to decrease errors in calculations. Aside from initialization, the introduction of more than 5 GKs could increase the performance of the system. As well, the use of better estimators such as the Extended Kalman Smoother and Cubature Kalman Filter, would increase the performance of the system considerably. For the classification task, the use of different and more advanced decision strategies for multi-classification, would yield faster results when used in a real-time scenario [18]. Overall, this project was an interesting and innovative approach to utilizing state estimation in conjunction with machine learning, to analyze ECG data and automatically diagnosis patients for heart disease.

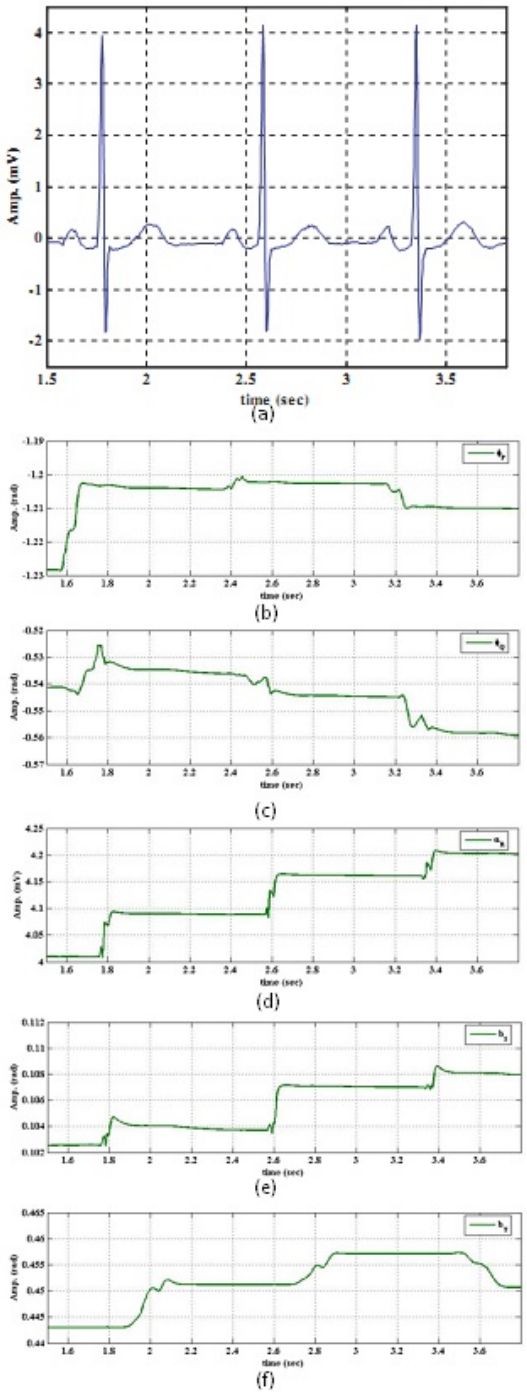

Figure 5.2: The estimated state variables from a typical ECG signal. (a) Input signal, (b) θ_P, (c) θ_Q, (d) a_R, (e) b_S and (f) b_T (adapted from [9]).

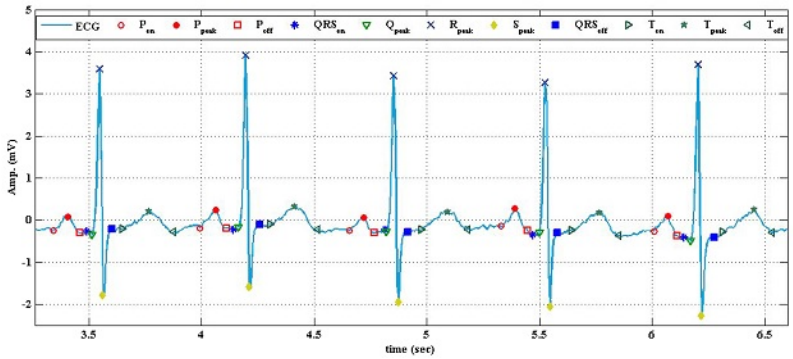

Figure 5.3: Segmentation results for typical ECG data (adapted from [9]).

Chapter 6

Acknowledgements

I would like to thank Dr. T. E. Doyle for his outstanding support and expertise knowledge, throughout the duration of this project.

Bibliography

[1] A. Dupre, S. Vieau and p. A. Iaizzo, "Basic ECG Theory, 12-Lead Recordings and Their Interpretation" in Handbook of Cardiac Anatomy, Physiology, and Devices, New York: Humana Press, 2006.

[2] Kundu M., Nasipuri M., Basu D.K. Knowledge-based ECG interpretation: A critical review, (2000) Pattern Recognition, 33 (3), pp. 351-373.

[3] 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.

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

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

[6] 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.

[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] Sameni R., Shamsollahi M. B., Jutten C., Clifford G. D. "A Nonlinear Bayesian Filtering Framework for ECG Denoising", IEEE Transactions On Biomedical Engineering, Volume 54, Issue 12, Dec. 2007 Page(s):2172 - 2185 .

[9] O. Sayadi and M.B. Shamsollahi, "A model-based Bayesian framework for ECG beat segmentation", Physiological Measurement, Vol. 30, pp. 335-352, 2009.

[10] Sayadi, O., Shamsollahi M. B., Clifford G. D."Synthetic ECG generation and bayesian filtering using a Gaussian wave-based dynamical model", IOP Physiol. Meas. Physiol. Meas. 31 (Aug 2010) pp 1309-1329.

[11] J. Pan and W. J. Tompkins, A real-time QRS detection algorithm. IEEE Trans. Biomed. Eng., BME-32 (3):230-236, 1985.

[12] R. Sutton and A. Barto, "Reinforcement learning". MIT Press, 1998, vol. 9. [Online]. Available: http://www.mitpressjournals.org/doi/abs/10.1162/089892999563184

[13] Weisstein, Eric W. "Sample Variance Computation." From MathWorld–A Wolfram Web Resource. http://mathworld.wolfram.com/SampleVarianceComputation.html

[14] Simon Haykin. "Neural Networks and Learning Machines", 3rd Ed., New Jersey: Pearson Education, Inc., 2009.

[15] P. S. Maybeck, Stochastic models, estimation and control, Academic Press, Inc., New York, USA (1979).

[16] S.R. Gunn, "Support Vector Machines for Classifcation 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] Madzarov G., Gjorgjevikj D., Chorbev I., "A Multi-class SVM Classifier Utilizing Binary Decision Tree", Informatica, Vol. 33, No. 1, pp. 233-242, May 2009.