Mobile On-Call System: Applying Multivariate Normal Analysis in a Personal Vital Stats Monitor

by

Sandra Escandor

Electrical and Biomedical Engineering Design Project (4BI6)
Department of Electrical and Computer Engineering
McMaster University
Hamilton, Ontario, Canada

Electrical and Biomedical 4BI6
Project Report

Mobile On-Call System: Applying Multivariate Normal Analysis in a Personal Vital Stats Monitor

by

Sandra Escandor (0562174)

Department of Electrical and Computer Engineering
Faculty Advisor: Prof. Deen

Electrical and Biomedical Engineering Project Report
Submitted in partial fulfillment of the requirements
for the degree of Bachelor of Engineering

McMaster University
Hamilton, Ontario, Canada
March 15, 2009

ABSTRACT

Patients who are recovering from illness and surgeries require a large amount of resources dedicated to them in order to monitor their well-being. At the same time, hospitals have a limited amount of resources. A proposed system, called Mobile On-Call, is a low-cost and low-power device that can be used for monitoring patients' well-being, which can, for example, reduce the number of hospital beds being used by patients during their recovery. The goal of this project is to create a cell phone based program to monitor a user's health, based on electrocardiogram data, blood pressure, and body temperature. The three vital sign signals are sent over a Bluetooth transmission interface device onto a cell phone. Data analysis is done in the cell phone on the three vital signs, such that a normal pattern for the patient is determined. If samples of the three vital signs show outlier data, a message to a physician or healthcare practitioner can be sent if deemed necessary.

Keywords: wireless, Bluetooth, multivariate normal analysis, biomedical monitoring

ACKNOWLEDGEMENTS

I would like to thank Dr. Jim Reilly, Tyler Ackland, Mike Kinsner, and Steven Li for their help, input, guidance, and encouragement. Without them, this project would not be the quality that it is right now. I would also like to thank them for their time and patience during the project research phase. I would like to thank Dhvani Parekh and Kirsten Zernask for being supportive team members. Last, I would also like to thank Dr. Jamal Deen for his guidance and for allowing me to work on this project.

CONTENTS

LIST OF FIGURES

LIST OF EQUATIONS

1 Introduction

The idea of monitoring a patient's medical vital signs remotely from the patient's home is becoming more crucial as hospitals try to conserve resources. The technology required to perform vital signs monitoring must meet specific requirements such as that it be cost effective, relatively simple to implement, efficient in power usage, non-invasive, and non-intrusive.

This project is a subset of an overall system that measures a patient's blood pressure, temperature, and heart rate. The system then analyzes the measured data, and compresses it. The compressed data is then sent to a monitoring physician/health worker upon request. If the vital signs are found to be irregular, the system alerts the monitoring physician/health worker immediately, so proper action can be taken.

The primary objective of this project is to create a relatively simple method of analyzing the trends of the patient's vital signs using multivariate normal analysis. Using this trend analysis, the vital signs monitor can send a message to the monitoring clinician if there is any unusual data measured from the patient. The monitoring clinician can then make decisions on the proper course of action. This trend analysis algorithm was implemented in J2ME to take advantage of an existing J2ME math application that includes matrix manipulation functionality.

The secondary objective of this project is to create a method that wirelessly transmits measured vital signs from the patient to a Sony Ericsson cell phone. In order to reduce the complexity of implementation, the PIC18F2550 microcontroller was chosen for the transmitter interface system, and a serial to Bluetooth transmitter was used to connect the transmitter interface system to the cell phone.

The scope of the primary objective is limited to the assumption that the user of the technology is an elderly patient over the age of 65 and that he or she is confined to bed rest. The focus for the secondary objective is limited to the Bluetooth protocol, since it is the transmitter technology implemented in the available Sony Ericsson cell phone.

2 Literature Review

Two currently existing products were examined. The first system, from Welch Allyn, is called the Acuity Central Monitoring System. It is composed of a wearable patient monitor, called a Micropaq, and central station. The central station is composed of a dedicated computer, which shows the vital sign measurements. It is also connected to a switch that connects the dedicated computer to several wireless access points. These wireless access points are the connection points that the Micropaq uses to send measured data to the dedicated computer (See Fig. A2.1 - Appendix). The following is a table summarizing the advantages and disadvantages of the Acuity Central Monitoring System.

Advantages	**Disadvantages**
Has an oxygen monitor	Needs an access point, which means more hardware
Overall system (Micropaq wearable monitor, access point, and Acuity Central Monitoring Station) can monitor more than one patient, provided that one wearable monitor is available for one patient.	Needs to communicate with a central station (called Acuity Central Monitoring Station) laptop to store data.
Patient data (ie, ECG waveform) can be displayed in a large window (due to Acuity Central Monitoring Station)	Only shows data for at most 24 hours. "Acuity LT stores patient review information for up to 24 hours after Acuity monitoring has stopped"
Can allow modification of minimum and maximum vital sign values that will trigger an alarm	Minimum and maximum vital sign values can be incorrectly set by an inexperienced user of the Acuity Central Monitoring Station.
Warns when Micropaq wearable monitor is out of range of wireless network	Since Micropaq wearable monitor requires communication with a central station, it can go out of range if user is located too far from central station.

Table 2.1: Acuity Central Monitoring System and Micropaq Overview

A basic Acuity Central Monitoring System costs $40 000, and one Micropaq costs $3 300 [1].

The second existing product examined, is the Mark of Fitness MF-77 Wrist Blood Pressure Monitor (See Fig. A2.3 - Appendix). The cost for only the blood pressure monitor is $74.95, and the interface kit costs $64.95 [2]. Originally, this system was not created to automatically interface with a PC/Smartphone in order to store and analyze the measured data, but IBM Research has been able to modify this unit to operate with Bluetooth and use a Sony Ericsson P900 cell phone as a communications hub [3]. The following is a table summarizing the advantages and disadvantages of the Mark of Fitness MF-77.

Advantages	**Disadvantages**
Display on actual measuring device	Does not automatically interface with a PC/Smartphone to store and analyze data
Has built in graphing on device	Uses inflation cuff which can be intrusive to user
Can show histograms and correlation charts on data measured	Requires an interface cable to store data into PC in order to do data analysis
Low cost	Only measures blood pressure and pulse/min

Table 2.2: Advantages and disadvantages of the Mark of Fitness MF-77

Other vital sign monitors researched [4],[5],[6] require the monitoring clinician to observe past results and determine the trend. One method for remote vital sign monitoring found, uses a two-phase reverse neural network learning algorithm [7]. This method requires a knowledge-base in order to compare vital signs that are considered normal for individuals of a specific age group, but there are some conflicting research results on this topic. With respect to temperature, there is conflicting research on the accepted value of the temperature at which an elderly person is considered having a fever. One study found that lowering the threshold for recognizing fevers in elderly nursing home residents would help in recognizing the onset of a fever [8]. Another study done on the normal

body temperature found that the idea of "older is colder" does not necessarily apply to all older adults and that "clinicians cannot adequately assess fever without first establishing what is 'normal' for different age groups" [9].

Another method for remote vital sign monitoring found, uses a support vector machine, in order to aid in identifying trauma patients who were in a hypovolemic state and being transported by helicopter [10]. The outcome of this research shows that the classification of the patient was successful 80% of the time, even though the environment caused high noise and missing data. This suggests that support vector machines are useful when the patient's environment can cause the vital sign monitor to miss data.

With respect to learning algorithms, there were three types examined for this project: support vector machines, artificial neural networks, and multivariate normal analysis. Support vector machines and artificial neural networks are well-suited to data that have high dimensionality. Both support vector machines and artificial neural networks require effort to train in order to be used [11]. In comparison to a support vector machine, an artificial neural network has local minima on its error surface [11]. These local minima cause more complexity in training the artificial neural network [11]. In comparison, a support vector machine has no local minima, so training is less complex [11]. Although a support vector machine does not have a complex training procedure in comparison to an artificial neural network, a support vector machine requires a lengthy training time, and is related to the number of training data required [11].

In comparison to the support vector machine and artificial neural network, multivariate normal analysis has a much simpler training method, since only the covariance matrix and mean need to be modified [12]. The downside to multivariate normal analysis is that it requires faster searching and updating for data sets that have high dimensionality [13]. Multivariate normal analysis is well suited for data having smaller dimensionality [12].

Support Vector Machines	**Artificial Neural Networks**	**Memory-based Learning (Multivariate Normal Analysis)**
Doesn't rely on data having Gaussian probability distribution [12].	Doesn't rely on data having Gaussian probability distribution [12].	Require faster searching and updating for data sets that have high dimensionality [13].
No local minima, so training is less complex than training a neural network [11].	Has local minima on its error surface, therefore training can be complex, since the goal is to find the global minimum on the error surface [11].	"Experimental results demonstrate the equal or superior power of memory-based methods on many problems" [13].
Requires long training time. The number of training data determines the amount of difficulty in training [11].	Vulnerable to outliers. Outliers need to be removed from training data [11].	Efficiency issues in terms of lookup table performance when dealing with data sets having high dimensionality [13].
	Predictions are fast after training, but training uses a lot of computational resources [14].	Only need to train covariance matrix and mean – less complex than classifier [12].

Table 2.3: Comparison chart of statistical learning algorithms examined

3 Statement of Problem and Methodology of Solution

3.1 Learning

People differ in the regular or normal values for their vital signs. For example, a patient, Person A may have a resting heart rate that can be expected to be 70 bpm, with a standard deviation of 2 bpm. Another patient, Person B may have a resting heart rate that can be expected to be 60 bpm, with a standard deviation of 4 bpm. This means that irregular values for resting heart rate for Person A will be different from the irregular values for resting heart rate for Person B. Monitoring a patient's normal values requires an algorithm that can detect trends and is personalized to suite the user.

The methodology of the solution proposed by this project uses multivariate normal analysis. Multivariate normal analysis was chosen due to the simplicity in its implementation. Because the scope of the project is limited to patients who are confined to bed rest, the need to take into account missing data is less.

The only dimensions required for the data space are resting heart rate, systolic blood pressure, and temperature values. Resting heart rate was specifically chosen as the heart rate type due to the fact that "high resting heart rate is a predictor for total and cardiovascular mortality" [22]. Also, systolic blood pressure was chosen since it is "more important than diastolic when determining risk" [23]. Samples of resting heart rate, systolic blood pressure, and temperature can be approximated to have Gaussian distributions [12].

The idea behind the methodology of this solution can be best illustrated by the following example. A scatter plot of the three sample distributions for a user can be made in the three-dimensional Cartesian space. The samples will come from data that has been obtained from the user over a 24-hour period. The plot will cluster at the region that

signifies the user's normal resting heart rate, blood pressure, and temperature. Ellipsoids with their center at the mean of the data cluster enclose ever larger numbers of data points as the ellipsoids' size increase, and "the link between the size of these ellipsoids and the enclosed probability is the chi-square distribution" [15]. The value obtained from the chi-square distribution for a specific quantile is a confidence region [12]. We can use the value obtained from a specific quantile of the chi-square distribution as a distance comparison for each point obtained [12]. More specifically, looking at a table for chi-square distribution values for a specific quantile, with three degrees of freedom (since the degrees of freedom is equal to the number of dimensions of the data), we can use the value obtained and compare it with the Mahalanobis distance of a specific data point, where the Mahalanobis distance is the distance calculated from the origin of the data cluster to the examined data point [15].

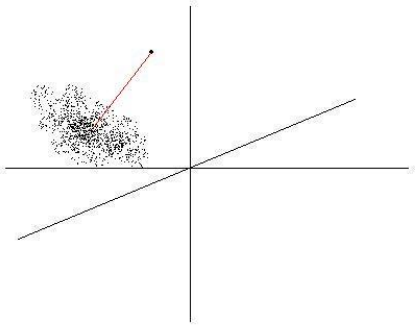

Figure 3.1: Example of a Data Cluster

In Figure 3.1, it can be seen that the data clusters about a center and has an ellipsoidal shape. An outlier can be seen to be quite far from the center of the data cluster. The line from the center of the data cluster to the outlier signifies the outlier's Mahalanobis distance.

Using a confidence region of 90%, any samples taken after the 24-hour period can be compared with the statistical distance required to be within the 90% confidence range. If

a future sample is calculated to be outside of the 90% confidence range, but within the 95% confidence range, a warning can be flagged. If a sample is calculated to be outside of the 95% confidence region, it is determined to be an outlier. This outlier signifies an irregular measurement taken from the user. If the threshold is reached for the number of outliers counted, then a signal is sent to the monitoring clinician.

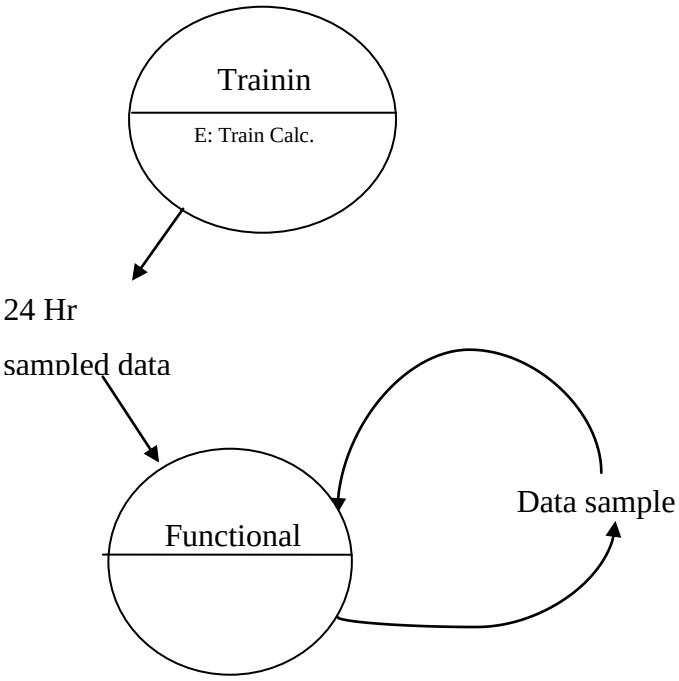

Figure 3.2: State Diagram

To be more specific, before classification can be done, training the learning algorithm is required. Training requires that the sample data over 24 hours, with each sample data point having an associated value of blood pressure, temperature, and heart rate, be placed into a matrix. The mean vector and covariance matrix can be obtained from the sample data matrix, which will be required for classification [12]. The mean vector is calculated by Equation 3.1. The covariance matrix is calculated by Equation 3.2.

$$[x_m \quad y_m \quad z_m \ ] = [\frac{\sum_{i=0}^{N-1} x_i}{N} \quad \frac{\sum_{i=0}^{N-1} y_i}{N} \quad \frac{\sum_{i=0}^{N-1} z_i}{N}]$$

Equation 3.1: Mean Vector Calculation

$$\begin{matrix} cov(x,x) & cov(x,y) & cov(x,z) \\ cov(y,x) & cov(y,y) & cov(y,z) \\ cov(z,x) & cov(z,y) & cov(z,z) \end{matrix}$$

Equation 3.2: Covariance Matrix Calculation

$$\frac{\sum_{i=0}^{N}(x_i - x_m)(y_i - y_m)}{N-1}$$

Equation 3.3: Covariance calculation of one element, where N is the number of rows in the training data.

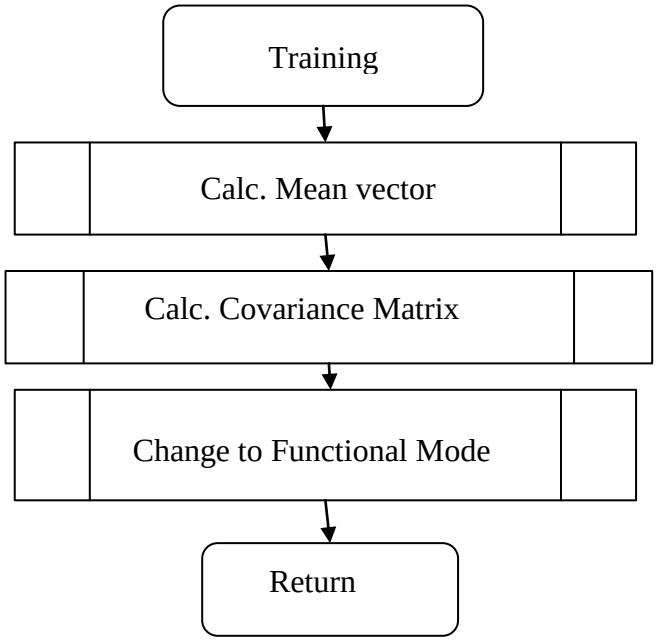

Figure 3.3: Training mode flow chart

Once the mean vector and the covariance matrix are calculated, the algorithm will go into functional mode. During functional mode, new data points are obtained. A Mahalanobis distance is calculated for one data point. This Mahalanobis distance is compared with a chi-square value that is associated with the required confidence region to be considered within a normal range, which was chosen to be 90%. If the Mahalanobis distance of one

data point is outside of the 90% confidence region, then the Mahalanobis distance is checked to see if it is within the confidence region between 90% and 95%. If the Mahalanobis distance is determined to be within that region, then a warning flag is sent to the caller of the algorithm. If the Mahalanobis distance is determined to be outside of the 95% confidence region, then it is considered an outlier, and a danger flag is sent to the caller of the algorithm.

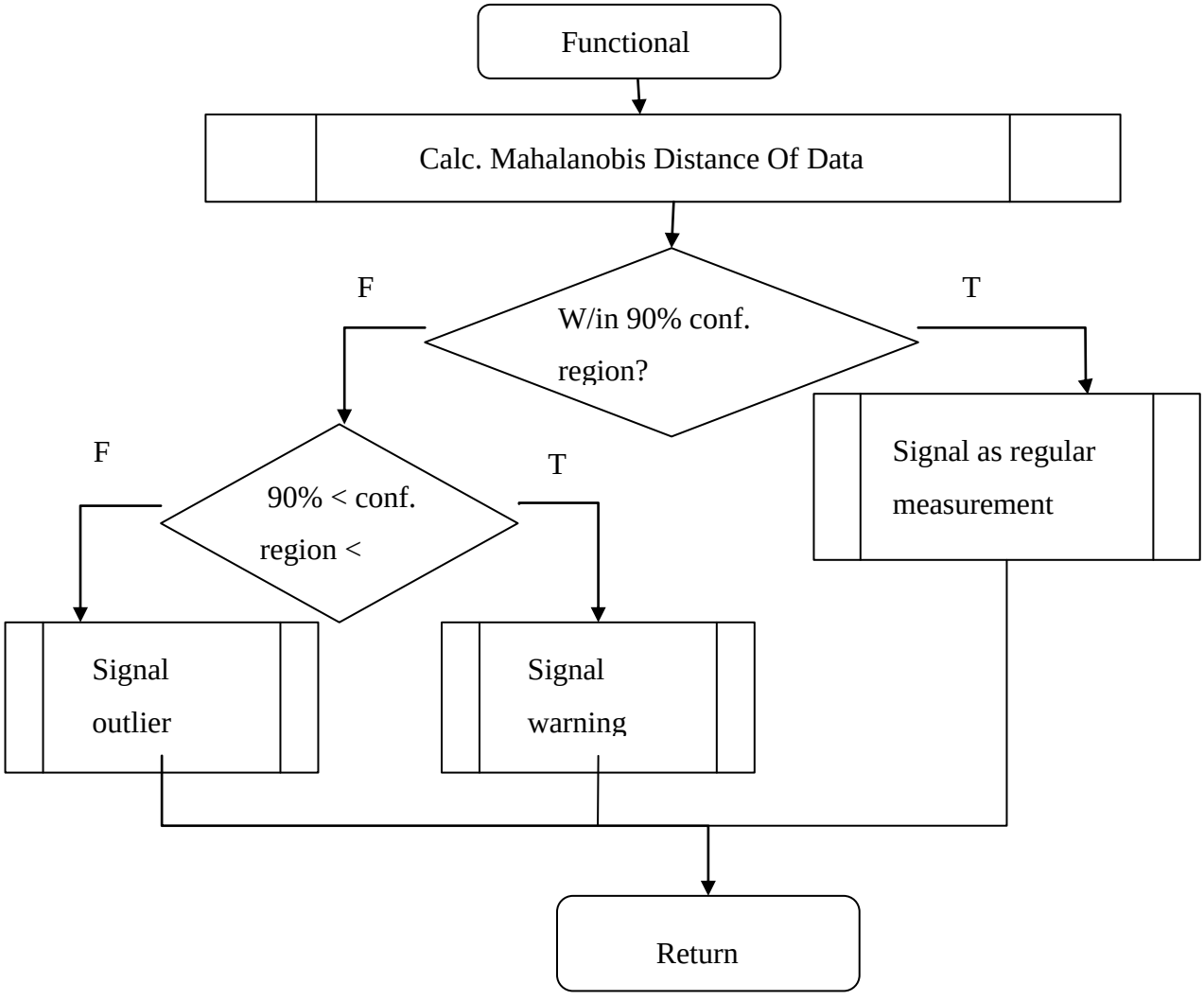

Figure 3.4: Functional Mode flow chart

$$\boldsymbol{D^2 = [\,X - \,\mu]^T[S]^{-1}[\,X - \,\mu]}$$

Equation 3.4: The Mahalanobis distance

In Equation 3.4, X is the data point vector, μ is the mean vector, and [S] is the covariance matrix [15].

The multivariate normal analysis algorithm is implemented using J2ME, Java for mobile phones. Netbeans is used as the programming environment. Jasymca, an open-source symbol calculator, is used for matrix manipulation functionality.

Implementing the algorithm in J2ME and using an open-source code for the matrix manipulation functionality was chosen so that having a laptop with Matlab as an intermediary between the sensors and the cell phone would be circumvented, in order to reduce the cost of the system.

3.2 Matrix Functionality

In order to calculate the Mahalanobis distance, the Jasymca open source mobile math application's Matrix, Vector, and Unexakt classes are used.

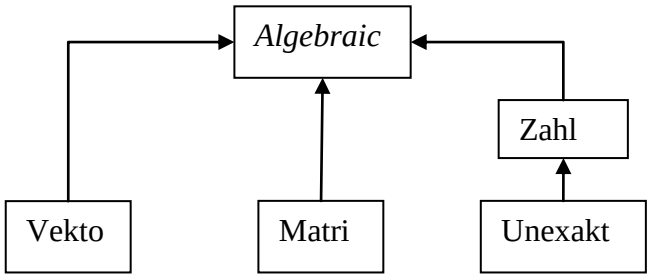

Figure 3.5 Class Diagram of Jasymca Matrix/Math classes used for project

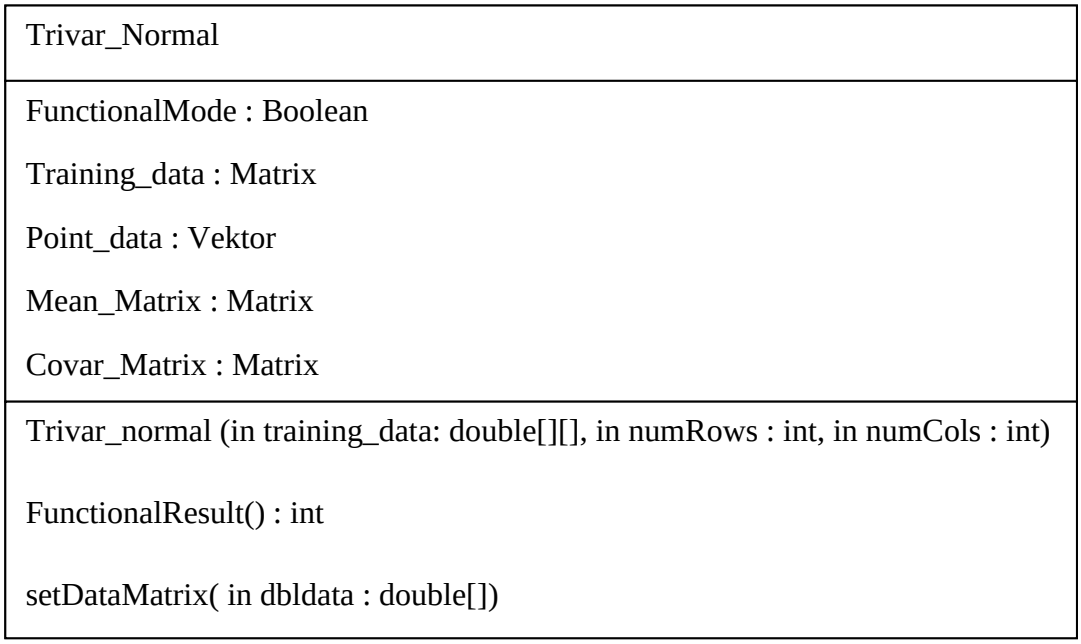

Figure 3.6: Trivar_Normal Class Diagram

Figure 3.6 shows the most important attributes and methods that exist in the Trivar_Normal class, which is the class created in order to realize the multivariate normal analysis algorithm.

Using the Trivar_Normal class is straightforward. First, it is instantiated by calling the Trivar_normal constructor. The constructor requires that the training data given in its parameter list be a double array. The Trivar_normal constructor also needs the number of rows and the number of columns of the training data double array. Once a Trivar_Normal object has been instantiated, its setDataMatrix method can be called and a data point can be given to it. Note that the data point must be a double array with three elements. The FunctionalResult method can be called after setting a data point. This method will return an integer that signifies the type of point (i.e normal, warning, or outlier).

3.3 R-R Peak Detection

The second issue is to be able to detect the RR peaks on the ECG signal from the ECG sensor, so that the proper heart rate can be calculated accurately and reliably. An algorithm developed for a wearable cardio respiratory system for in-home monitoring was chosen as a starting point [16]. It is important to note that RR peak detection algorithms that have fixed amplitude thresholds result in missing many real peaks [16]. The algorithm developed for a wearable cardio respiratory system uses variable amplitude thresholds in order to detect as many real peaks as possible.

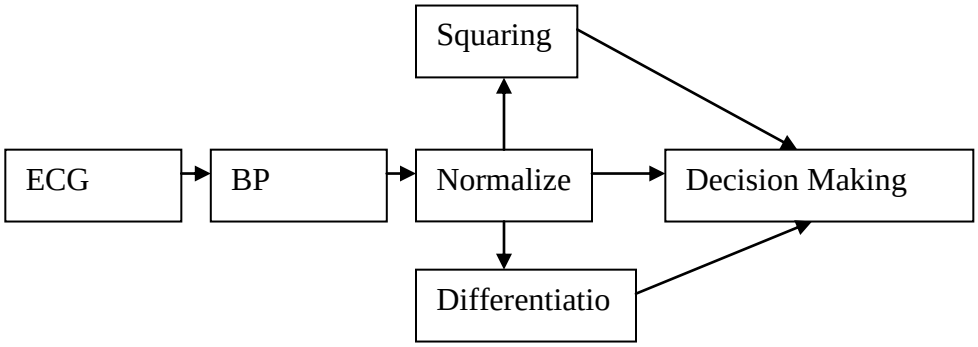

Figure 3.7: Flow chart for QRS Detection [16]

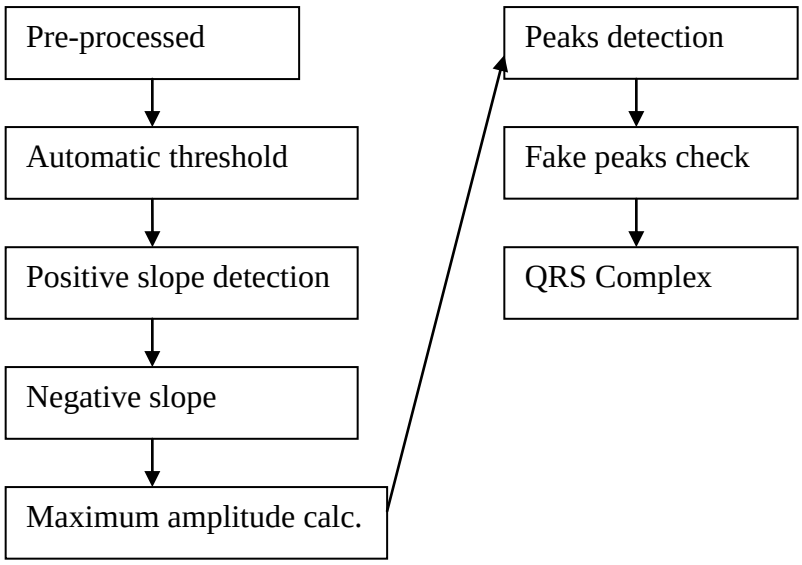

Figure 3.8: Flow chart for Decision Making Stage [16]

The algorithm used for RR peak detection in this project was developed from the algorithm mentioned above. A stripped down version of the algorithm described above was used, in order to create simplifications for the project. It consists of the following components for RR peak detection: Variable positive and negative slope threshold calculation, positive slope detection, negative slope detection, maximum amplitude calculation, and hear rate calculation.

The positive slope threshold calculations are done by taking the data and passing through the whole set once to determine the average positive slope. Specifically, any slope that is greater than zero is counted, and is added to a running total of slope. After the whole data set is passed, the running total of slope is divided by the number of counts of slope greater than zero. Once the average positive slope is calculated, the positive slope threshold is then determined as:

$$Positive\ Slope\ Threshold = Average\ Positive\ Slope + Average\ Positive\ Slope\ \times 2.5$$

Equation 3.5: Calculation of Positive Slope Threshold

With respect to negative slope threshold calculations, a similar process is done, as previously described with positive threshold calculations, but instead, any slope that is less than zero is counted.

Positive slope detection is done by passing through the data set and calculating the slope at each data element point. The slope at a data element position is calculated by taking the value of the data element ahead, and the value of the previous data element and dividing by the sample time period. If the slope is greater than the positive slope threshold, then it is counted. If there are two consecutive positive slopes that surpass the positive slope threshold, then the data sample number is considered the data sample number of the start of the first R peak location. Once the start of the first R peak location has been determined, the end of the R peak must be found, within fifteen data sample positions after finding the start of the first R peak location. A new first R peak location is found if

the first R peak end location has not been found within fifteen data sample positions. Once the first R peak is found, the second R peak is found in the same manner.

After the first R peak starting position, first R peak ending position, second R peak starting position, and second R peak ending position are found, the maximum value between each starting and ending positions are found. These two maximum values correspond to the location of the points of the two R peaks. Once the locations of the points of the two R peaks are found, the heart rate (in beats per minute) is calculated as:

$$Heart\ Rate\ (bpm) = \frac{60}{Sample\ \ Period \times (Second\ \ R\ Peak\ \ Location - First\ \ R\ Peak\ \ Location\ )}$$

Equation 3.6: Heart Rate Calculation

3.4 Bluetooth Transmission

The third issue that needs to be solved in order to create the overall system is to determine a way of sending data from the sensors to the cell phone. The method must be efficient in terms of power usage, inexpensive, and simple to implement. The method must also be compatible with the cell phone data transmission protocol.

The chosen method was to use the PIC18F2550 to perform analog to digital conversion of the three sensor outputs. After digital conversion of each sensor value, the results are converted to their ASCII representation, and then sent to the USART of the microcontroller. The results are converted to their ASCII representation so that it is easier for the data compression subsystem to process this data. Because the microcontroller operates at 0 to 5 V output levels, but the RS-232 protocol for Bluetooth requires a voltage range of -12 to 12 V, a MAX-232 TTL/RS-232 chip needs to be connected at the send/receive transmission pins of the microcontroller [17]. A DB-9 connector is connected to the send/receive transmission pins of the MAX-232 chip. A serial to Bluetooth adapter is then connected to the DB-9 connector. The board design for the PICDEM2 Plus was modified to fit the purpose of this project [18].

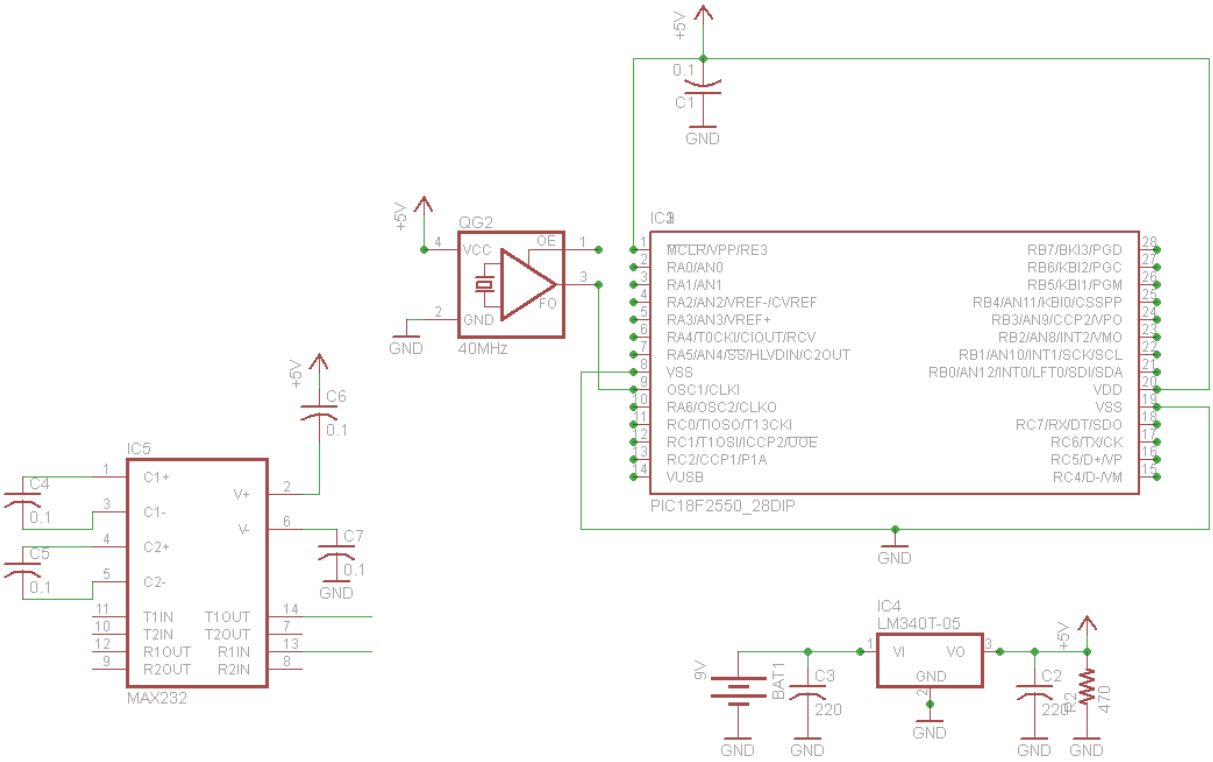

Figure 3.9: Schematic of Bluetooth Interface

The inputs for the three sensors, ECG, temperature, and blood pressure need to be connected to pin 2, pin 3, and pin 4, respectively. It is important to note that the voltage on any pin with respect to the V_{ss} pin of the microcontroller can range only between -0.3 V to (V_{DD} + 0.3 V) [19].

The positive Vref of the analog to digital converter is connected to 5 V. The negative Vref of the analog to digital converter is connected to 0 V. The least significant bit calculation is shown in Equation 3.7:

$$\boldsymbol{LSB = \frac{V_{ref}}{2^N}}$$

Equation 3.7: Least Significant Bit Calculation

Therefore, the least significant bit is 4.88 mV. This means that every 4.88 mV in the analog to digital converter input is counted as one bit.

The procedure to send data from the microcontroller starts when a timer interrupt occurs, which calls the interrupt handler. In the interrupt handler, three channels are opened one by one. Specifically, after a channel opens, the analog to digital conversion starts, and the value is stored, and then the next channel is opened, and so on. After the three channels have all been through the analog to digital conversion, each of the values from the channels are converted from their floating point representation to an ASCII representation of the floating point number, and this representation is sent to the USART. The code to perform the data transmission has been modified to fit this project from a general code originally by Steven Li.

The idea behind sampling is as follows. Initially, a counter is set which starts at the starting value count, in this case 15651. The counter will count up to 65535, since we have a 16-bit clock counter. Once the counter reaches 65535, the counter is reset to 0, and the interrupt service routine is started. The interrupt service routine then resets the counter to 15651 to start counting again, and then it samples the channels and sends out all of the sample values during the same time that the counter increments back to 65535.

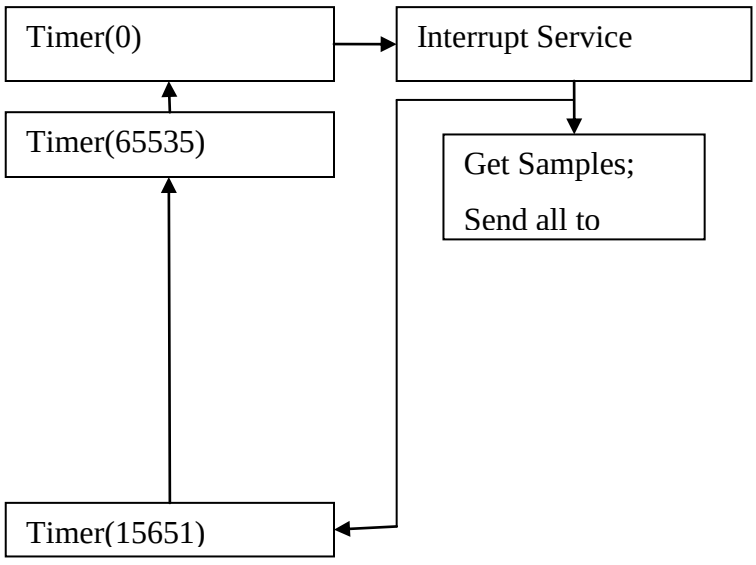

Figure 3.10: Diagram of timing

For example, since the ECG requirement of the overall system specify a sampling rate of 200 Hz, we calculate the period that it takes for a 200 Hz sampling rate, which is over a period of $5x10^{-3}$ ms.

$$\boldsymbol{Internal\ Instruction\ Cycle} = \left(\frac{\boldsymbol{Oscillator\ Frequency}}{\mathbf{4}}\right)^{\mathbf{-1}}$$

Equation 3.8: Determining Internal Instruction Clock Cycle Period [20]

Therefore, we know that using a 40 MHz clock, one instruction takes: $(40\ MHz / 4)^{-1} = 1x10^{-7}$ s to complete.

$$\boldsymbol{C} = \frac{\boldsymbol{One\ Period\ of\ Sampling}}{\boldsymbol{Internal\ Instruction\ Cycle}}$$

Equation 3.9: Number of Counts Required for Specified Sampling Rate

$$\boldsymbol{Counter\ Start\ Value} = (\mathbf{2}^{\boldsymbol{x}} - \mathbf{1}) - \boldsymbol{C}$$

Equation 3.10: Determining the Start Value of the Timer Counter

From Equation 3.9, we can see that it requires 50000 timer count increments in order to have a 200 Hz sampling rate. Therefore, from Equation 3.10, we can see that the timer counter needs to start at 15535.

It is important to note that it has been assumed that the number of instructions required for calling the interrupt service routine, and then also to reset the timer counter to the specific start value takes only one instruction cycle, and therefore, the value that is obtained by Equations 3.8, 3.9, and 3.10 is an estimate of the value required. In order to achieve the 200 Hz sampling rate, a frequency counter needs to be connected to pin 6 of the microcontroller to measure the current timer frequency, and then small adjustments

can be made to the timer counter to get 200 Hz. In this case, the start value for the timer counter was found to be 15651, which is not too far from the calculated value.

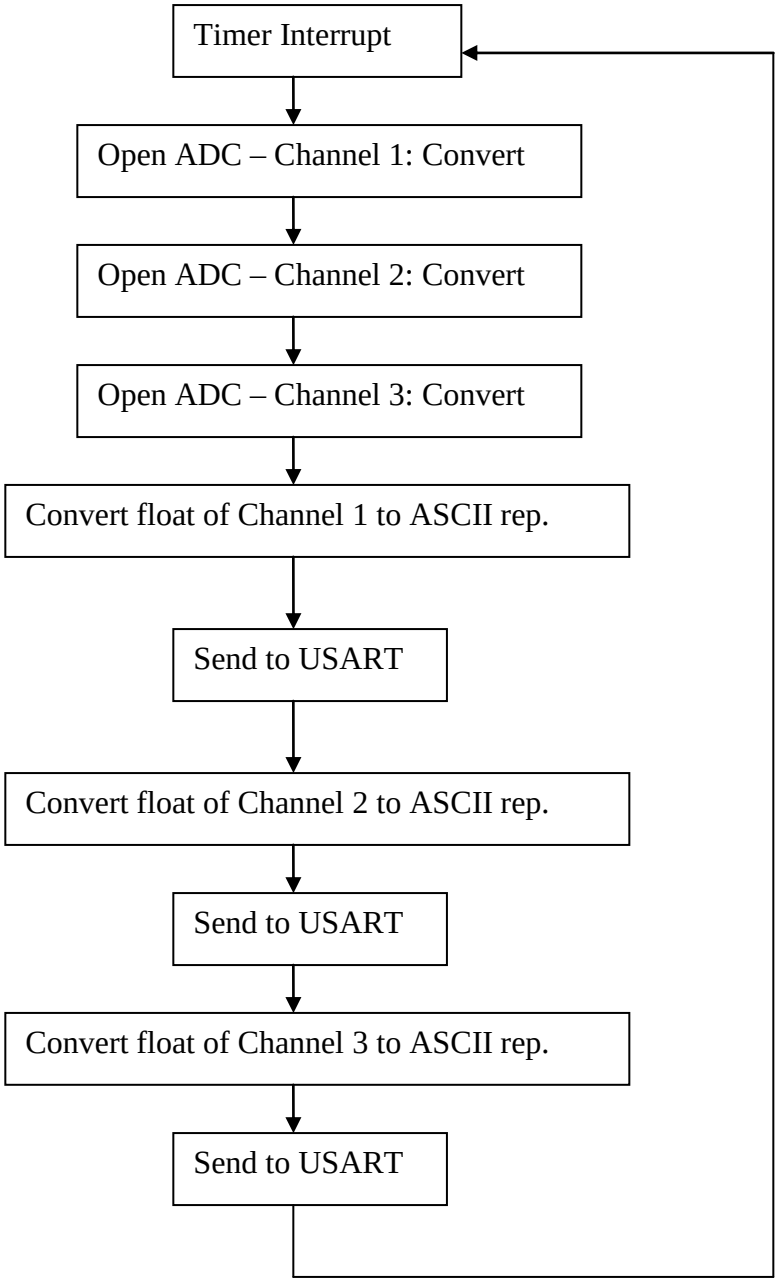

Figure 3.11: Data Transmission Flow chart

4 Experimental and Design Procedures

4.1 J2ME Environment Set-Up

Netbeans IDE and Sony Ericsson mobile development kit v.2.5.06 is downloaded on a laptop running Windows Vista. The Sony Ericsson development kit is installed first. Afterwards, the Netbeans IDE is installed and set up to use the Sony Ericsson mobile development kit. Next, Jasymca is downloaded and then built using the KToolbar utility packaged with the Sony Ericsson mobile development kit.

It is important to note that Jasymca needs to be placed into a JAR package file before it can be used in the project in Netbeans, so that the various math classes that are required can be used. After placing Jasymca into a JAR package file, it is added into the project for the multivariate normal analysis application.

A template for a simple mobile phone application is used as a starting point, so that the multivariate normal analysis code could be started using the Sony Ericsson emulator.

4.2 J2ME Debugging

The procedure for debugging a J2ME application is not the same as debugging regular Java applications. The idea is that the emulator starts the application in debug mode, acting like a server, and the debugger must connect to the emulator remotely [17].

Before debugging the code in Netbeans, the KToolbar utility is started and the project to debug is opened. Then, in Netbeans, the project can be started in debug mode, and a port number for the debugging session will be displayed. This port number is then entered in as the “Remote debugging port” in KToolbar. After Netbeans is able to connect to the emulator, the debugging session can proceed.

4.3 Matlab Data Generation and Algorithm Testing

The multivariate normal analysis sample observation data and test point data were created using Matlab. Heart rate, temperature, and systolic blood pressure data were randomly created using the Gaussian normal distribution random number generator in Matlab. Using Matlab, sample data is generated and used for multivariate normal analysis testing. The testing is done on both the Matlab code generated for the multivariate normal analysis, in order to verify the algorithm, and also on the Java code. It is expected that the Java results behave exactly like the Matlab results.

The following is a description of the Matlab data generation. First, three Gaussian random variables are generated, one each for heart rate, temperature, and systolic blood pressure. Each observation contains these three random variables. One hundred observations (for example) is generated and placed into an observation matrix. The next step, after the creation of the observation matrix, is to find its covariance matrix, which will be called the observation covariance matrix. After the covariance matrix of the observation matrix is found, the observation covariance matrix eigenvectors and eigenvalues are then obtained. The eigenvectors will give the principal directions of the ellipsoids. The eigenvalues give the magnitude of the corresponding eigenvectors. Since the classification of the three data point types (outlier, warning, and normal) need to be tested, the three data points need to be generated. In order to create an outlier data point, the following steps must occur. First, an eigenvector of the covariance matrix must be multiplied by the square root of its eigenvalue, which then must be multiplied by the 0.95 quantile of the chi square distribution. This creates a vector that extends out from the center of the data cluster, towards the surface of the ellipsoid for the 0.95 quantile of the chi square distribution. Next, the mean vector of the observation matrix must be added to the vector created from the previous step mentioned. This creates a new vector that extends from the origin (0,0,0) to the surface of the ellipsoid for the 0.95 quantile of the chi square distribution, and this now corresponds to an outlier data point.

In order to create a warning data point, the following steps must occur. First, an eigenvector of the covariance matrix must be multiplied by the square root of its eigenvalue, which then must be multiplied by the 0.9 quantile of the chi square distribution. This creates a vector that extends out from the center of the data cluster, towards the surface of the ellipsoid for the 0.9 quantile of the chi square distribution. Next, the mean vector of the observation matrix must be added to the vector created from the previous step mentioned. A small offset must be added to get the point within the warning region. This creates a new vector that extends from the origin (0,0,0) to slightly outside of the surface of the ellipsoid for the 0.9 quantile of the chi square distribution, and this now corresponds to a warning region.

In order to create a regular data point, an observation within less than the 0.9 quantile of the chi square distribution may be used, and similar steps outlined for generating a warning data point can be used.

After the three data points have been created, the Mahalanobis distance formula in Equation 3.4, can be applied to each of the three data points. A check must be done in order to determine if the distance formula matches the expectations for each data point types.

In order to test the results in Java, the Matlab code must generate the appropriate Java array expressions for the observation data points, as well as the three data point types mentioned above.

$$General\ Test\ Data\ Point = M + \Lambda\sqrt{\lambda\alpha}$$

Equation 4.1: Creating a test data point using the eigenvector and eigenvalue

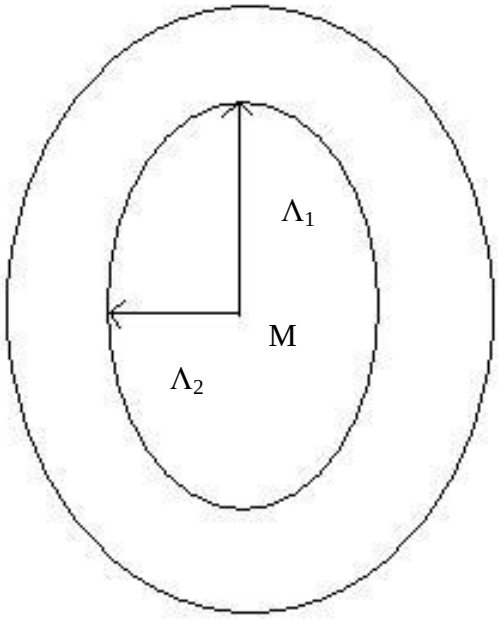

Figure 4.1 Example of eigenvectors and their eigenvalues for a quantile of chi square distribution for a two dimensional data set

For R-R peak detection, Matlab is also used. First, a ten second ECG data sample, for example, record cdb/08730_01, is obtained from the MIT-BIH ECG compression test database [24]. The ECG data sample is read into Matlab. The R-R peak detection algorithm is applied on the ECG data sample and the heart rate calculation is also applied in order to get the heart rate output. A manual calculation check is done in order to determine whether the heart rate output is correct, using Equation 4.2. The Matlab code also generates the appropriate Java array expression for the ECG data sample so that the data sample can be entered into the Java code for testing the Java ported algorithm.

$$Heart\ Rate\ (bpm) = 60/(Second\ R\ Peak\ Time\ Stamp - First\ R\ Peak\ Time\ Stamp)$$

Equation 4.2: Calculating Heart rate from R-R peak times

4.4 Bluetooth Transmission Testing

The transmission module testing requires the following setup, illustrated in Figure 3.1.

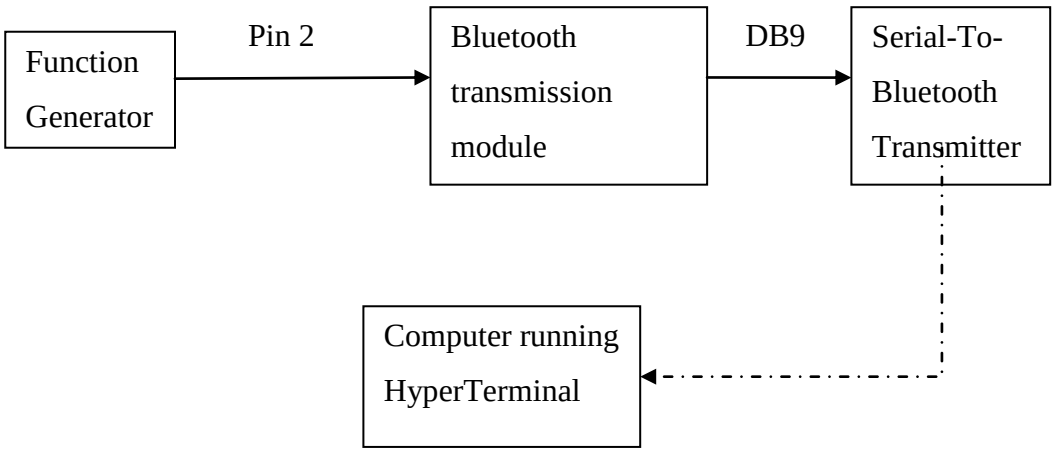

Figure 4.2: Bluetooth transmission testing set-up

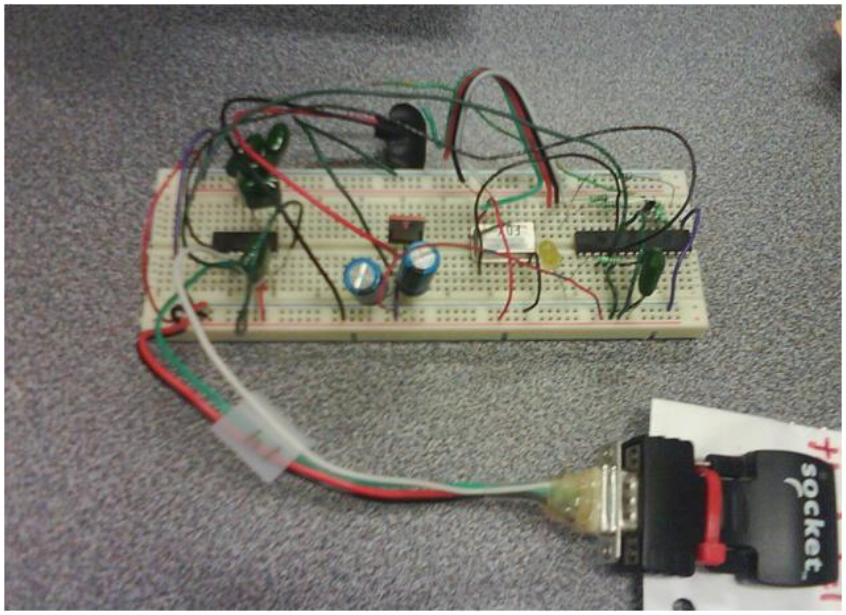

Figure 4.3: Bluetooth interface with Bluetooth transmitter (“socket”) connected

The positive V_{ref} (pin 5 of the microcontroller) is connected to 5 V. The negative V_{ref} (pin 4 of the microcontroller) is connected to 0 V. The voltage input which will act as the test input voltage is connected (pin 2 of the microcontroller). The Bluetooth transmission module is then connected to the Serial-To-Bluetooth Transmitter. The Bluetooth connection on the computer is enabled and the Serial-To-Bluetooth transmitter ID is selected in order for the computer to connect to the Serial-To-Bluetooth transmitter. Once the computer discovers the Serial-To-Bluetooth transmitter, HyperTerminal can be started. In HyperTerminal, the proper COM port must be selected for the Bluetooth transmitter located on the computer. When establishing a HyperTerminal connection, the baud rate of 115200 is used. The comma separated values received by HyperTerminal are the values from the three different input channels. Each value in the comma separated value received by HyperTerminal is the decimal representation of the converted voltage.

For example, if the input voltage is 0 V, the analog to digital converter will convert this to 0 (binary). The ftoa function used in the Bluetooth transmission interface module converts this value to a decimal value of 0 (decimal). If the input voltage is 5 V, the analog to digital converter will convert this to 10000000000 (binary). The ftoa function used in the Bluetooth transmission interface module converts this value to a decimal value of 1024 (decimal). This means that in order to interpret the decimal value in terms of voltages, we would need to multiply the decimal value result with the least significant bit voltage, which in this case is 4.88 mV (which was calculated using Equation 3.7). Using Equation 4.3, if the received decimal value is 512, for example, then the corresponding voltage is 2.5 V.

$$\boldsymbol{Voltage = LSB_{Voltage} \times Decimal\ Data\ Value\ Transmitted}$$

Equation 4.3: Corresponding Voltage from Transmitted Data

There were two tests done on the transmitter in order to determine proper data transmission. The first test was done using an input voltage, first at 0.0034 V and then switching to 4.981 V. The second test was done using a 6 Hz sine wave with amplitude of 4.970 V.

5 RESULTS AND DISCUSSION

5.1 Bluetooth Testing

Figure 5.1 and Figure 5.2 are the graphs of the data received by HyperTerminal. In Figure 5.1, a voltage of 0.0034 V is first applied on pin 2 (channel 1), and then a voltage of 4.981 is applied. At the 0.0034 V input, the data received by HyperTerminal corresponds to a bit conversion of approximately 0 (decimal), which is expected. At the 4.981 V input, the data received by HyperTerminal corresponds to a bit conversion of 1023 (decimal), which is also expected. Figure 5.2, shows an input that is a 6 Hz sinusoidal voltage with amplitude varying from 0 to 4.970 V. The decimal representation data received by HyperTerminal shows a sinusoidal trend, which is expected. Also note that the sinusoidal trend has a minimum of approximately 0 and a maximum of approximately 1023, as expected. Note that the x-axis corresponds to the number of the data sample in the data stream.

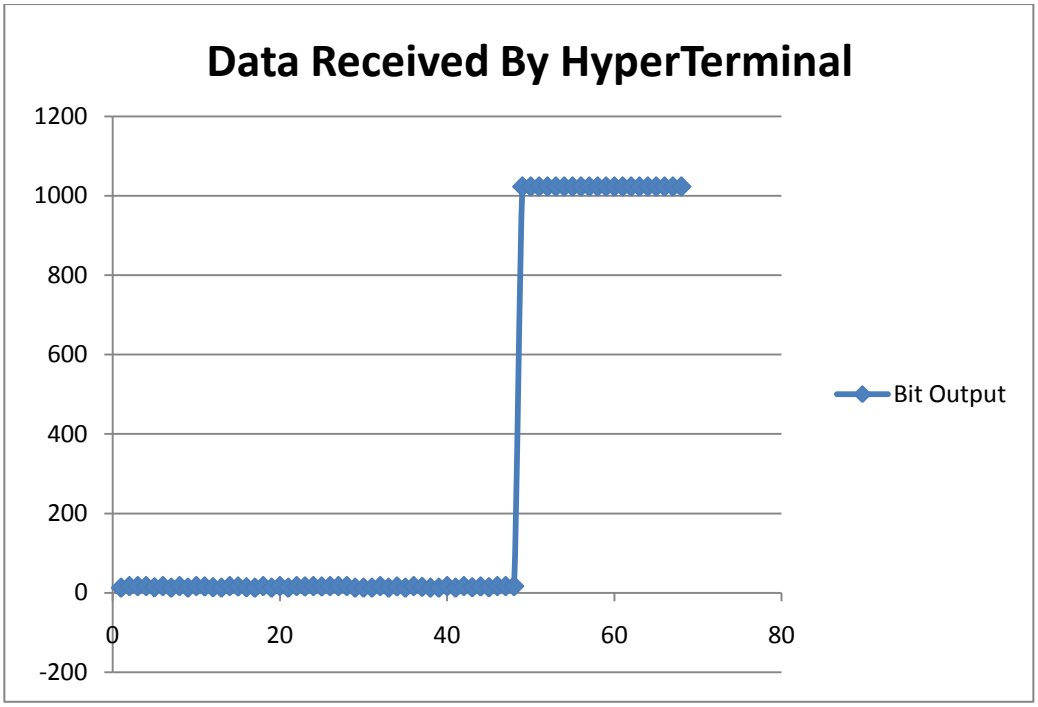

Figure 5.1: DC Voltage at 0.0034 V then at 4.981 V

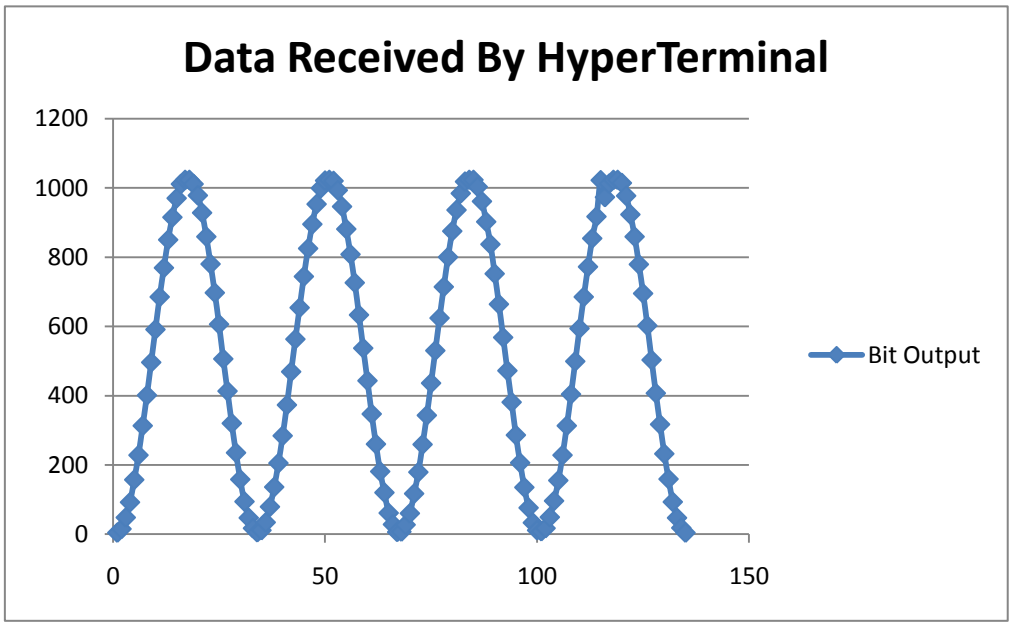

Figure 5.2: 6 Hz sinusoidal input amplitude 4.970 V

5.2 Multivariate Normal Analysis

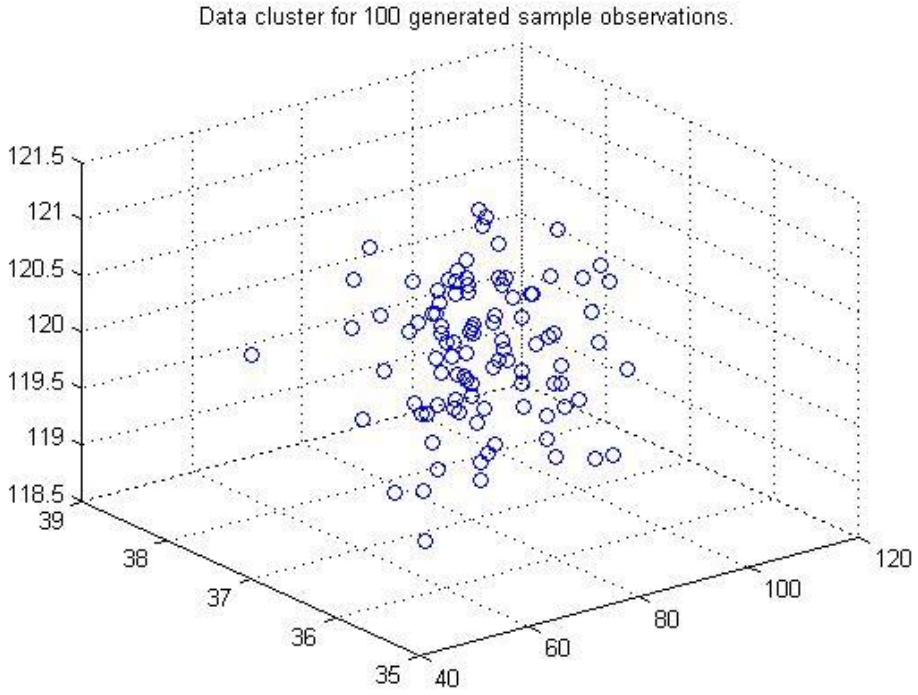

Figure 5.3 Data cluster for 100 generated sample observations plotted in Matlab.

Data points generated by Matlab and used as input:

point1 = { 82.10, 37.84, 118.88 } //Outlier Data Test Point

point2 = { 89.75, 37.70, 120.27 } //Regular Data Test Point

point3 = { 56.36, 36.74, 119.80 } //Warning Data Test Point

Both Matlab and Java algorithms correctly identify the outlier, regular, and warning data test points. Although, there is an additional condition in the Java algorithm that states that if an outlier is within the preset range that the monitoring physician has created (for the case of sending an alarm if any of the vital signs are outside of the ranges), then the outlier data point is categorized as a warning data point. Of course, without this additional condition, the outlier data test point would be categorized as a danger point.

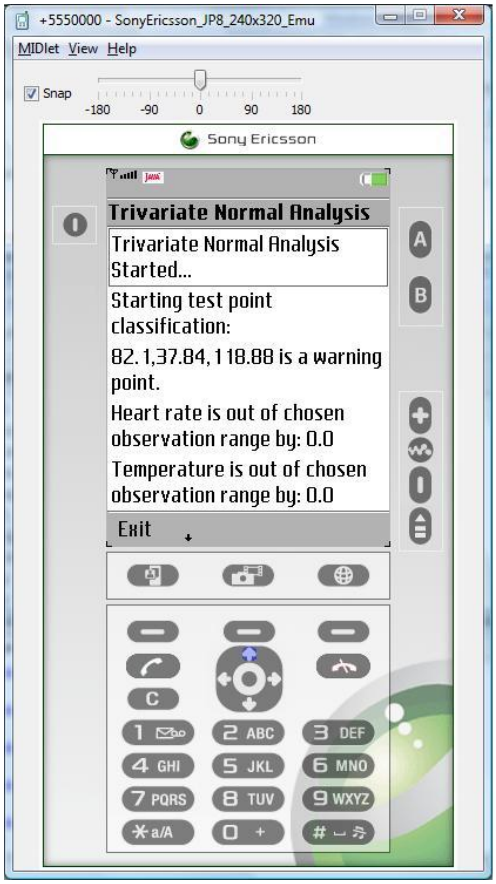

Figure 5.4: Multivariate Normal Analysis running on Sony Ericsson Emulator

5.3 R-R Peak Detection and Heart Rate Calculation

Calculation of heart rate using the ECG signal in Figure 5.5 was done, first, by taking the difference between the second R-peak time and the first R-peak time. The heart rate was then calculated by dividing sixty seconds by the difference between the second R-peak time and the first R-peak time. The manual heart rate calculation results in 39 bpm.

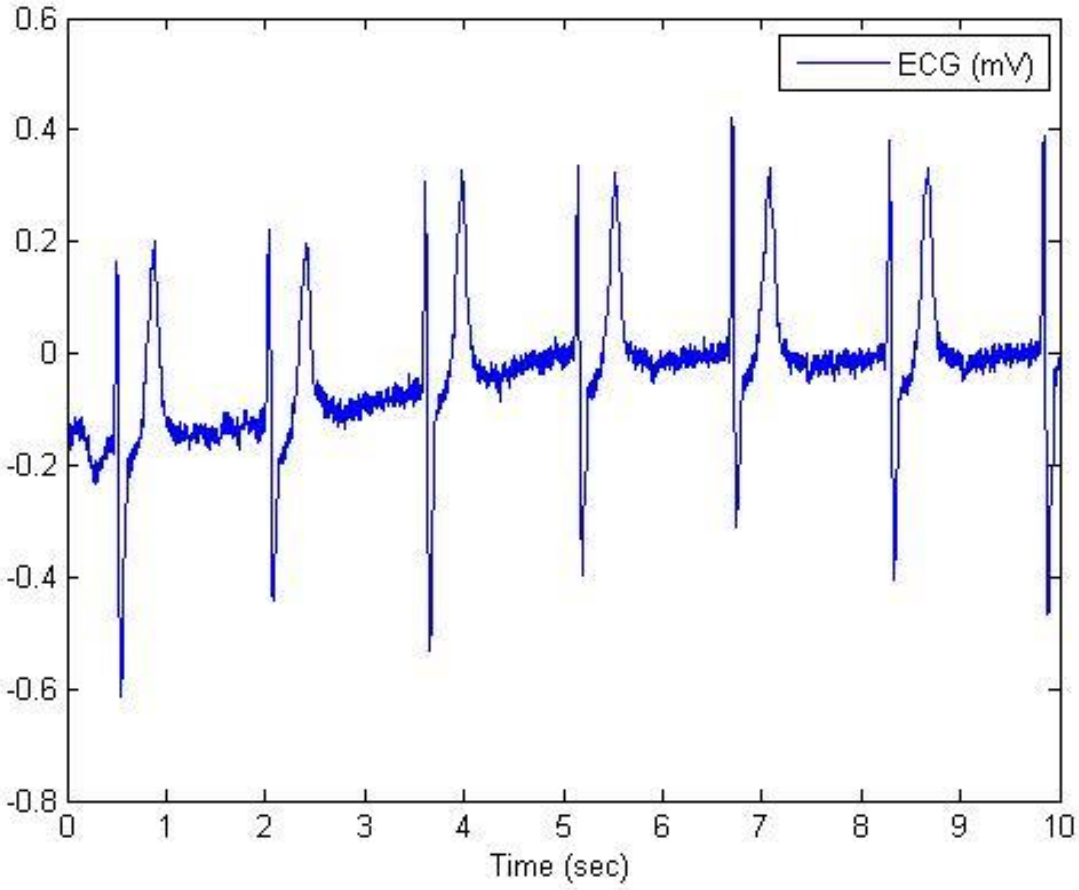

Figure 5.5: ECG Sample 1 in time domain

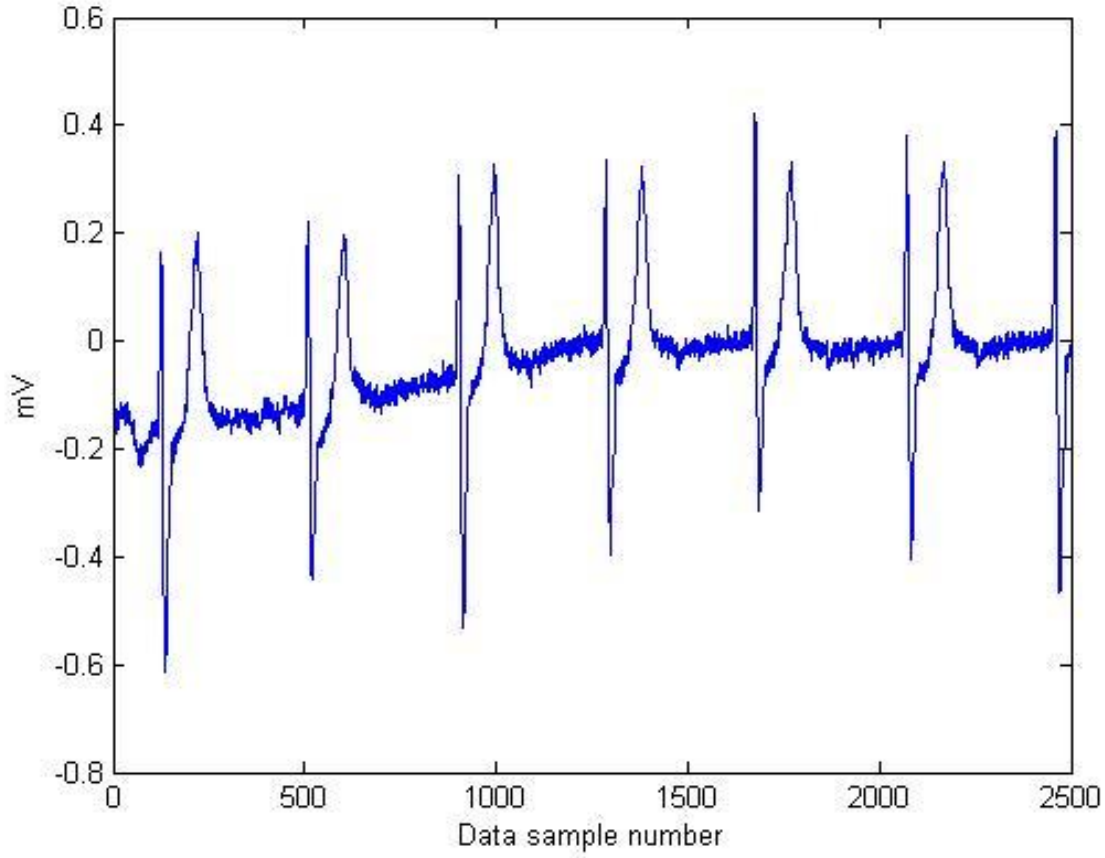

Figure 5.6: ECG Sample 1 in sample number domain

The Java R-R peak detection algorithm and heart rate calculation matches the manual heart rate calculation for ECG data sample 1. The corresponding ECG data in the data sample number domain shows the location of the R-R peak in terms of the index of the array of each of the data points. This can be used to determine if the proper data sample is obtained.

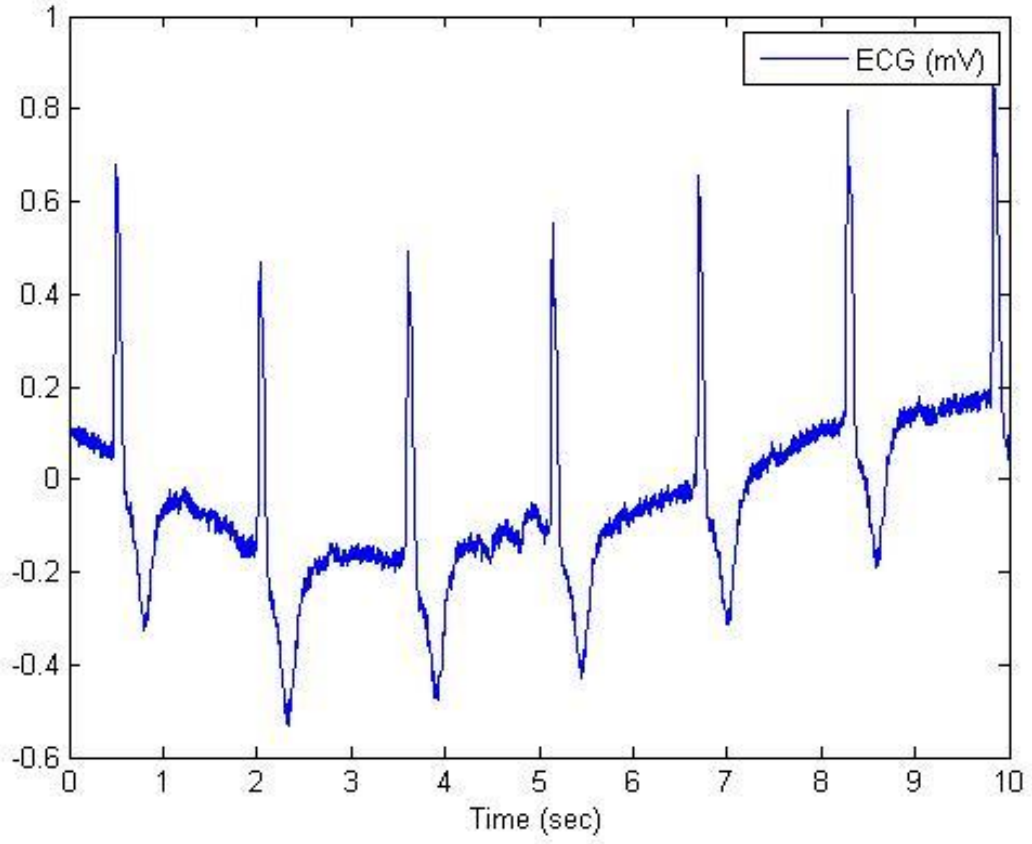

Figure 5.7: ECG data sample 2 in time domain

The resulting manual heart rate calculation for the data set shown in Figure 5.7 is 39 bpm.

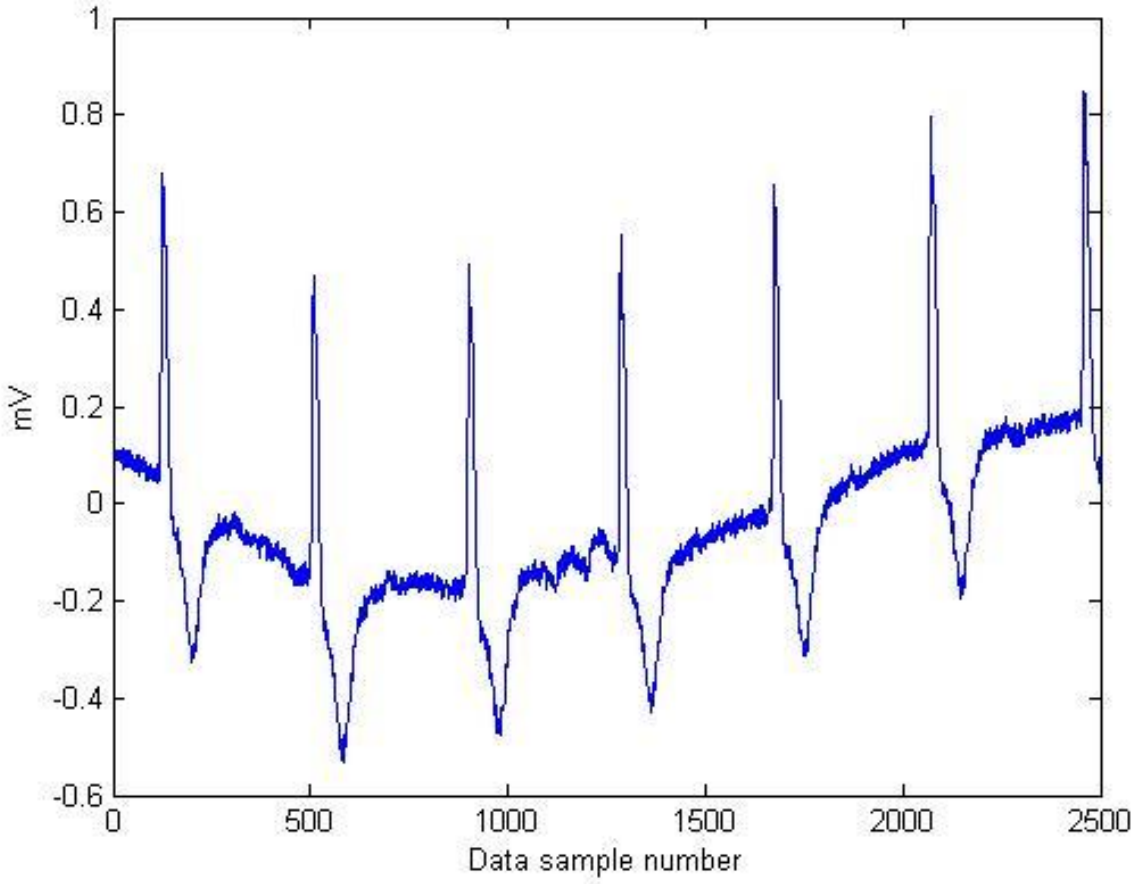

Figure 5.8: ECG data sample 2 in sample number domain

The corresponding ECG data in the data sample number domain shows the location of the R-R peak in terms of the index of the array of each of the data points. This can be used to determine if the proper data sample is obtained.

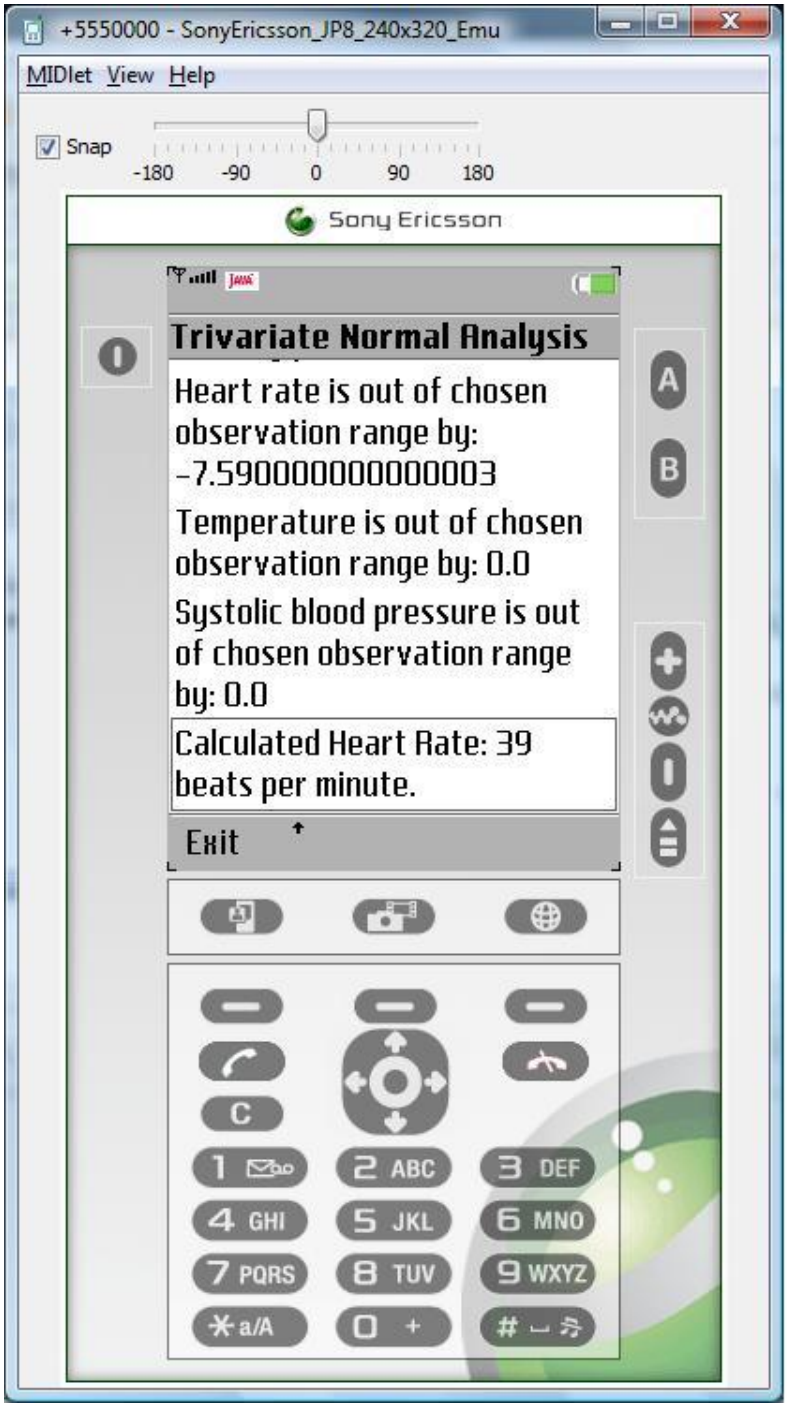

Figure 5.9: Heart Rate calculation running on Sony Ericsson Emulator

6 Conclusions and Recommendations

The application of the multivariate normal analysis learning algorithm for data analysis of a patient's heart rate, temperature, and blood pressure is presented. A QRS complex detection algorithm is also presented. Finally, a method for transmitting data from the three sensors to a cell phone using the Bluetooth protocol is presented.

The multivariate learning algorithm and QRS complex detection algorithm still require refinement and more development in order to be useful in a home vital sign monitoring setting. The Bluetooth transmitter module requires further testing with the cell phone.

It can be seen that the multivariate learning algorithm can be applied to the cell phone and that the expected classifications are done by the algorithm. Once the multivariate normal analysis learning algorithm has been optimized, the algorithm can be combined with an activity classifier. An example of a recently developed activity classifier is one developed at Imperial College London by Lo, et al [21]. Once combined with an activity classifier, the multivariate normal analysis learning algorithm can be used to determine trends for each different type of activity.

With respect to the QRS complex detection algorithm, a simplified version of the one applied for a wearable cardio respiratory system for in-home monitoring was chosen as a starting point [16]. The more complex version should be applied if the system is to be developed for a patient, since the more complex algorithm is more sensitive to varying positive and negative slope thresholds.

The Bluetooth transmission module can also be refined. One such refinement would be a handshake protocol between the transmission module and cell phone, so that the cell phone can detect dropped data packets, and the cell phone can then ask the transmission module for a retransmission of the dropped data packets.

Overall, it has been shown that applying this system is possible and that creating a more refined and complex version is achievable.

Appendix A

A1. List of Computer Programs Used

Microsoft Office 2007 (Word and Powerpoint), MATLAB, NetBeans IDE, Sony Ericsson Java ME SDK For CLDC, Jasymca, MPLAB IDE v8.50

A2. Summary of Existing Technologies Examined

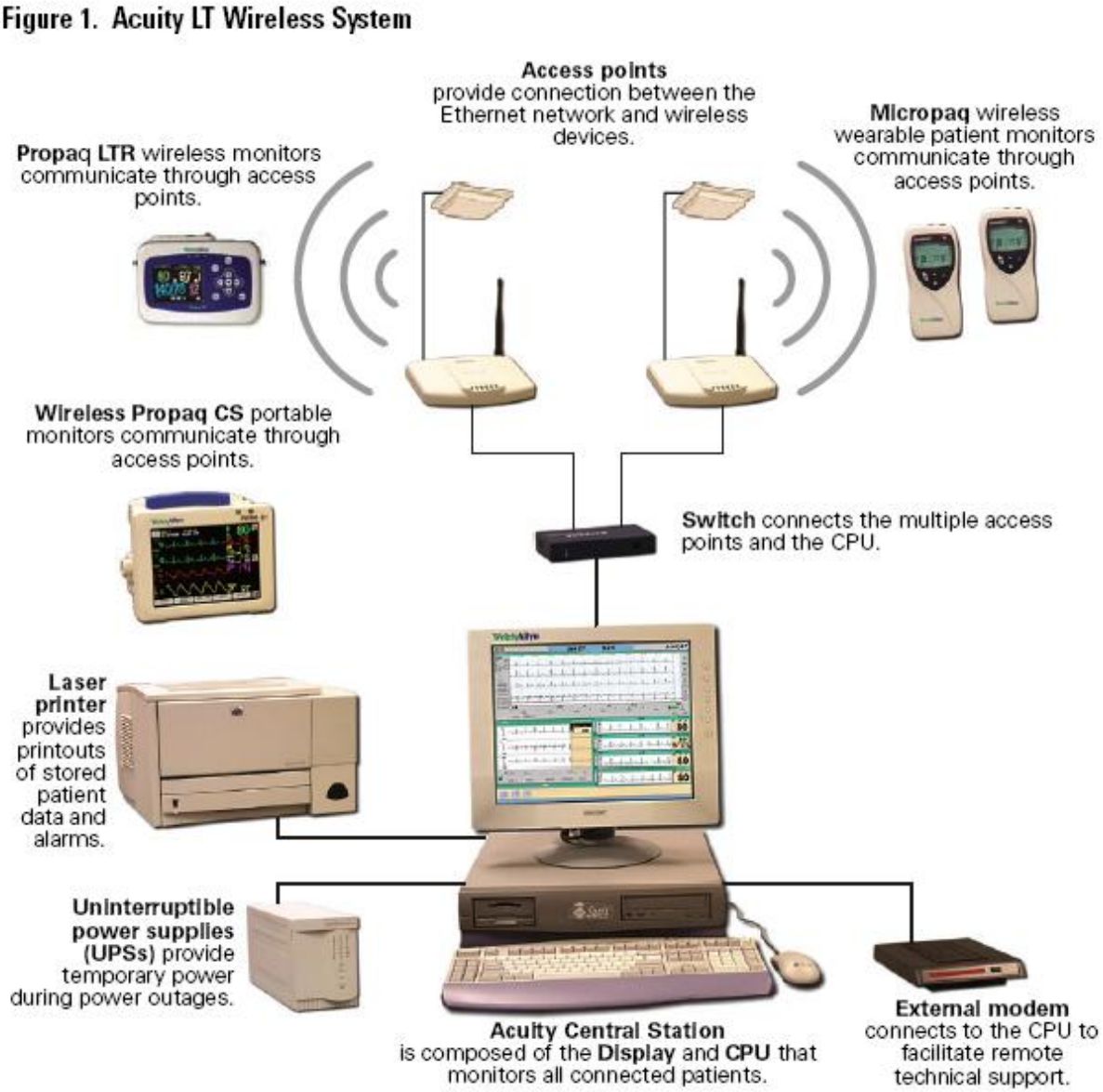

Figure A2.1: Acuity Central Monitoring Station System

Figure A2.2: Micropaq wireless wearable patient monitor

Micropaq Specifications
• Wireless = 2.4 GHz WLAN; IEEE 802.11 compliant from Symbol Technologies • Weighs 16 oz with battery • 7-3/16" x 3-1/2" x 1-9/16" (size of human hand) • Drop resistant (50 g shock) • Water resistant (IPX 1) • Backlit LCD with automatic power save • Shows 1 or 2 ECG leads • Heart rate, SpO2 numerics, pulse bar

Table A2.1: Micropaq Specifications

Figure A2.3: Mark of Fitness MF-77 Blood Pressure Monitor

Figure A2.4: Mark of Fitness Computer Interface Kit

References

[1] T. Andison. (2010, *Email communication with Welch Allyn sales rep.*

[2] Anonymous Mark of fitness MF-77 blood pressure monitor. *2009* Available: http://www.bloodpressuremonitor.bz.libaccess.lib.mcmaster.ca/mark_of_fitness_mf77_blood_pressure_monitors.asp

[3] Anonymous Remote monitoring of health conditions - IBM researchers, working with medical device manufacturers and mobile phone handset manufacturers, have created a unique solution to track vital health signs. *2009* Available: http://domino.research.ibm.com/comm/pr.nsf/pages/news.20031112_mobilehealth.html

[4] T. xin, G. 'Xing-ming, C. min and y. Yan. 6-8 July 2007). Wireless telemedicine physiological monitoring center based on virtual instruments. *Bioinformatics and Biomedical Engineering, 2007. ICBBE 2007. the 1st International Conference on* pp. 1157-1160.

[5] R. Elgharably, E. Marzban, S. Belal, B. Ahmad, I. AbdElLatif, R. Atef and I. ElBabli. 18-20 Dec. 2008). Wireless-enabled telemedicine system for remote monitoring. *Biomedical Engineering Conference, 2008. CIBEC 2008. Cairo International* pp. 1-4.

[6] J. M. Kang, T. Yoo and H. C. Kim. Oct. 2006). A wrist-worn integrated health monitoring instrument with a tele-reporting device for telemedicine and telecare. *Instrumentation and Measurement, IEEE Transactions on 55(Issue: 5),* pp. 1655-1661.

[7] I. Jung, S. C. Park and G. Wang, "Two Phase Reverse Neural Network Based Human Vital Sign Prediction System," *Computational Intelligence for Modelling, Control and Automation, 2006 and International Conference on Intelligent Agents, Web Technologies and Internet Commerce, International Conference on,* pp. 135-135, Nov. 28 2006-Dec. 1 2006.

[8] S. Castle, D. Norman, M. Yeh, D. Miller and T. Yoshikawa, "Fever response in elderly nursing home residents: are the older truly colder?" *J Am Geriatr Soc,* 1991 Sep.

[9] S. Lu and Y. Dai, "Normal body temperature and the effects of age, sex, ambient temperature and body mass index on normal oral temperature: A prospective, comparative study " *International Journal of Nursing Studies,* vol. 46, 2009.

[10] L. Chen, T. M. McKenna, A. T. Reisner, A. Gribok and J. Reifman, "Decision tool for the early diagnosis of trauma patient hypovolemia," *Journal of Biomedical Informatics,* vol. Volume 41, pp. 469, 2008.

[11] S. Abe. (2005, *Support Vector Machines for Pattern Classification.*

[12] J. Reilly, "Meetings," Nov 20, 2009.

[13] M. Kárný, K. Warwick and V. (. Kůrková, *Dealing with Complexity : A Neural Network Approach.* 1998,

[14] J. Schneider and A. Moore. A locally weighted learning tutorial using vizier 1.0

[15] D. S. Wilks. (2005, "Multivariate statistics," in *Statistical Methods in the Atmospheric Sciences* (2nd ed.) Anonymous

[16] M. Adnane, Z. Jiang and S. Choi. (2009, Development of QRS detection algorithm designed for wearable cardiorespiratory system. *Computer Methods and Programs in Biomedicine 93(1),*

[17] D. Ibrahim. (2008, "Chapter 4: Functions and libraries in mikroC," in *Advanced PIC Microcontroller Projects in C : From USB to RTOS with the PIC 18F Series*

[18] Anonymous PICDEM 2 Plus Demonstration board user's guide. Available: ww1.microchip.com/downloads/en/DeviceDoc/51275d.pdf

[19] Anonymous "PIC18F2455/2550/4455/4550 Data Sheet,"

[20] Anonymous Timers: Timer0 tutorial (part 2). Available: ww1.microchip.com/downloads/en/DeviceDoc/51702A.pdf

[21] B. Lo, L. Atallah, O. Aziz, M. El ElHew, A. Darzi and G. Yang. (2007, "Real-time pervasive monitoring for postoperative care," in *4th International Workshop on Wearable and Implantable Body Sensor Networks (BSN 2007)*

[22] A. Diaz, M. G. Bourassa, M. Guertin and J. Tardif, "Long-term prognostic value of resting heart rate in patients with suspected or proven coronary artery disease," European Heart Journal, vol. 26, 2005.

[23] T. G. Pickering, "1.2," in Ambulatory Monitoring and Blood Pressure Variability. Science Press 1991

[24] "MIT-BIH ECG compression test database", http://www.physionet.org/cgi-bin/ATM

Vitae

NAME: Sandra Escandor
EDUCATION: Mohawk College of Applied Arts and Technology (2001-2004)

Sandra Escandor is currently completing level IV of Electrical and Biomedical Engineering at McMaster University in Hamilton, Ontario.