Tele-cardiology sensor networks for remote ECG monitoring

Meng Jiang

Follow this and additional works at: https://scholarworks.rit.edu/theses

Recommended Citation

This Thesis is brought to you for free and open access by RIT Scholar Works. It has been accepted for inclusion in Theses by an authorized administrator of RIT Scholar Works. For more information, please contact ritscholarworks@rit.edu.
Tele-cardiology Sensor Networks for Remote ECG Monitoring

by

Meng Jiang

A Thesis Submitted in Partial Fulfillment of the Requirements for the Degree of
Master of Science in Computer Engineering

Supervised by

Dr. Fei Hu
Department of Computer Engineering
Kate Gleason College of Engineering
Rochester Institute of Technology
Rochester, NY
October, 2006

Approved By:

_____________________________________________    ___________    ___
Dr. Fei Hu
Primary Advisor – R.I.T. Dept. of Computer Engineering

_____________________________________________         ______________
Dr. Marcin Lukowiak
Secondary Advisor – R.I.T. Dept. of Computer Engineering

_____________________________________________         ______________
Dr. Xiaojun Cao
Secondary Advisor – R.I.T. Dept. of Network Security and System Administration
Thesis Release Permission Form

Rochester Institute of Technology
Kate Gleason College of Engineering

Title: Tele-cardiology Sensor Networks for Remote ECG Monitoring

I, MENG JIANG, HEREBY GRANT PERMISSION TO THE WALLACE MEMORIAL LIBRARY TO REPRODUCE MY THESIS IN WHOLE OR PART.

_________________________________
Meng Jiang

_________________________________
Date
Dedication

To my grandpa, who although have left us, but have never left my heart.

To my family, for the values they have taught me and the man they raised me to be.

To Lu, for her years of silent support and always having faith in me.
Acknowledgements

I would like to thank Dr. Fei Hu for advising this work and opening a new door for me in the fields of wireless sensor networks and signal processing. None of this would have been possible without his guidance, both academically and personally. Also thanks to Dr. Lukowiak and Dr. Cao for serving on my committee.

Special thanks to Cindy Lui and Daniel Fava for proof reading parts of this thesis.
Abstract

One of today’s most pressing matters in medical care is the response time to patients in need. The scope of this thesis is to suggest a solution that would help reduce response time in emergency situations utilizing wireless sensor networks technology. Wireless sensor network researches have recently gained unprecedented momentum in both industries and academia, especially its potential applications in Emergency Medical Services and Intensive Care Units. The enhanced power efficiency, minimized production cost, condensed physical layout, as well as reduced wired connections, presents a much more proficient and simplified approach to the continuous monitoring of patients’ physiological status.

This thesis focuses on the areas of remote ECG feature extraction utilizing wavelet transformation concepts and sensor networks technology. The proposed sensor network system provides the following contributions. The low-cost, low-power wearable platforms are to be distributed to patients of concern and will provide continuous ECG monitoring by measuring electrical potentials between various points of the body using a galvanometer. The system is enabled with integrated RF communication capability that will relay the signals wirelessly to a workstation monitor. The workstation is equipped with ECG signal processing software that performs ECG characteristic extractions via wavelet transformation. Lastly, a low-complex, end-to-end security scheme is also incorporated into this system to ensure patient privacy. Other notable features include location tracking algorithms for patient tracking, and MATLAB Server environment for internal communication.
# Table of Contents

**THESIS RELEASE PERMISSION FORM** ................................................................................II  
**DEDICATION** .................................................................................................................. III  
**ACKNOWLEDGEMENTS** ................................................................................................. IV  
**ABSTRACT** ...................................................................................................................... V  
**LIST OF FIGURES** ......................................................................................................... VIII  
**LIST OF TABLES** ........................................................................................................... X  
**LIST OF EQUATIONS** .................................................................................................... XI  
**GLOSSARY** ....................................................................................................................... XII  

## CHAPTER 1  INTRODUCTION .................................................................................. 1  
## CHAPTER 2  BACKGROUND ................................................................................... 4  

2.2. WAVELET ANALYSIS ............................................................................................... 6  
2.3. WIRELESS SENSOR NETWORKS .............................................................................. 10  
2.4. WORK ENVIRONMENT ............................................................................................ 11  
  2.4.1 TinyOS ................................................................................................................. 11  
  2.4.2 NesC ..................................................................................................................... 12  
  2.4.3 MATLAB ............................................................................................................. 12  

## CHAPTER 3  MOBILE PLATFORM .......................................................................... 13  

3.1. TELOSB MOTE ........................................................................................................... 14  
3.2. SENSOR BOARD ....................................................................................................... 15  
  3.2.1 Circuit Design......................................................................................................... 15  
  3.2.2 Layout Design......................................................................................................... 16  
  3.2.3 Electronic Components ......................................................................................... 17
3.3. ECG SIGNAL SIMULATOR .................................................................................................. 19

CHAPTER 4 DATA COMMUNICATION .................................................................................. 21

4.1. SENSOR NETWORK COMMUNICATION ........................................................................... 21

4.1.1 Network Configuration .................................................................................................. 21

4.1.2 VitalDust Plus – Flavor RIT ........................................................................................ 23

4.1.3 Network Security ........................................................................................................ 24

4.2. MATLAB SERVER ........................................................................................................ 29

CHAPTER 5 FEATURE EXTRACTION .................................................................................. 32

5.1. ANALYSIS OVERVIEW .................................................................................................... 32

5.2. ALGORITHMS .................................................................................................................. 36

5.3. IMPLEMENTATION .......................................................................................................... 40

5.3.1 R Peak Detection ........................................................................................................ 43

5.3.2 QS Detection ............................................................................................................... 43

5.3.3 Zero Level Detection .................................................................................................. 44

5.3.4 PT Wave Detection .................................................................................................... 44

5.3.5 Results ......................................................................................................................... 45

CHAPTER 6 LOCATION TRACKING ................................................................................... 47

6.1. INTRODUCTION ............................................................................................................. 47

6.2. MOTETRACK .................................................................................................................. 48

6.3. DEPLOYMENT ................................................................................................................ 50

6.4. PROBLEMS ................................................................................................................... 54

CHAPTER 7 DEPLOYMENT PROCEDURE ........................................................................... 56

CHAPTER 8 CONCLUSION .................................................................................................. 60

BIBLIOGRAPHY ................................................................................................................... 62
List of Figures

Figure 2.1 Example of a Normal ECG Trace .......................................................... 5
Figure 2.2 Electrocardiogram of a Healthy Heart ............................................... 6
Figure 2.3 Sine Wave and Wavelet Comparisons ............................................... 7
Figure 2.4 Continuous Wavelet Transformation (CWT) ..................................... 8
Figure 2.5 Filter Banks Signal Decomposition .................................................. 9
Figure 2.6 Wavelet Decomposition Tree .......................................................... 10
Figure 2.7 Sensor Motes - Mica2, Mica2Dot, Telosb ...................................... 11
Figure 3.1 Mobile Platform ............................................................................ 13
Figure 3.2 TelosB Block Diagram ................................................................... 14
Figure 3.3 Sensor Board Schematics .............................................................. 15
Figure 3.4 Sensor Board Layout Diagram ....................................................... 16
Figure 3.5 Complete Sensor Board .................................................................. 18
Figure 3.6 3-Lead ECG Snap Set Leadwires .................................................... 19
Figure 3.7 Model 430B Patient Simulator .......................................................... 19
Figure 3.8 Mobile Platform Patient Simulation ................................................. 20
Figure 4.1 Wireless Sensor Network .................................................................. 22
Figure 4.2 VitalDust Plus - Flavor RIT .............................................................. 23
Figure 4.3 TinySec and Original Data Packets ................................................... 26
Figure 4.4 Data Displayed Every 600 packets ..................................................... 30
Figure 5.1 Filter Banks Analysis ....................................................................... 34
Figure 5.2 Haar Wavelet ................................................................................... 36
Figure 5.3 Daubechies Wavelets ...................................................................... 37
Figure 5.4 Symlets Wavelets................................................................. 38
Figure 5.5 Equivalent frequency responses of the DWT at scales $2^k$.............. 39
Figure 5.6 WT at the First Five Scales of ECG-like Simulated Waves ............ 40
Figure 5.7 5-Level Decomposition Using DB6 .......................................... 41
Figure 5.8 Normal ECG Signal and Frequency Distribution........................ 42
Figure 5.9 Sample Feature Extraction......................................................... 45
Figure 5.10 Real-time Feature Extraction.................................................... 46
Figure 6.1 Example of stored Reference Signatures ..................................... 49
Figure 6.2 Experiment Field and Beacon Placements.................................... 51
Figure 6.3 Data Collection Stage ............................................................... 53
Figure 6.4 Location Tracking................................................................. 54
Figure 7.1 System Overview ................................................................. 56
Figure 7.2 Port Forwarding ................................................................. 57
Figure 7.3 Data Collection ................................................................. 58
Figure 7.4 Feature Extraction a)Tab Mote30 b)Tab Mote40 .................... 59
Figure 7.5 Feature Extraction Result....................................................... 59
List of Tables

Table 3.1 Sensor Board Part List ................................................................. 18
Table 4.1 TinySecMode Interface ............................................................... 28
List of Equations

\[ CWT(a, \tau) = \frac{1}{\sqrt{a}} \int s(t)\psi\left(\frac{t-\tau}{a}\right)dt \]  \hspace{1cm} (2.1) ................................................................. 8

\[ DWT(m, n) = 2 \sum_{k} s(k)\psi(2^{-m}k - n) \]  \hspace{1cm} (2.2) ................................................................. 9

\[ y[n] = (x * g)[n] = \sum_{k=-\infty}^{\infty} x[k] \cdot g[n-k] \]  \hspace{1cm} (5.1) ............................................................... 34

\[ y[n] = (x * h)[n] = \sum_{k=-\infty}^{\infty} x[k] \cdot h[n-k] \]  \hspace{1cm} (5.2) ............................................................... 35

\[ y_{\text{low}}[n] = \sum_{k=-\infty}^{\infty} x[k] \cdot g[2 \cdot n - k] \]  \hspace{1cm} (5.3) ............................................................... 35

\[ y_{\text{high}}[n] = \sum_{k=-\infty}^{\infty} x[k] \cdot h[2 \cdot n - k] \]  \hspace{1cm} (5.4) ............................................................... 35

\[ (y \downarrow k)[n] = y[k \cdot n] \]  \hspace{1cm} (5.5) ............................................................... 35

\[ y_{\text{low}} = (x * g) \downarrow 2 \]  \hspace{1cm} (5.6) ............................................................... 35

\[ y_{\text{high}} = (x * h) \downarrow 2 \]  \hspace{1cm} (5.7) ............................................................... 35
Glossary

#

3-Lead System  AN ECG signal measuring system utilizing 3-Lead placements. There are also other systems requiring up to a total of 12-Leads.

B

BPM  Beats per Minute.

C

Cardiac arrhythmia  A group of conditions in which the muscle contraction of the heart is irregular or is faster or slower than normal.

CodeBlue  Led by Harvard University, the CodeBlue project is a combined hardware and software platform tailored for medical monitoring.

E

ECG  Electrocardiogram – an electrical recording of the heart and is used in the investigation of heart disease.

EMS  Emergency Medical Services – responsible for providing pre-hospital (or out-of-hospital) care by paramedics, emergency medical technicians, and medical first responders.
Endocardium  The innermost layer of cells, embryologically and biologically that lines blood vessels. Recently, it has become evident that the endocardium controls myocardial function.

G

Galvanometer  An instrument for detecting and measuring electric currents.

GUI  Graphical User Interface.

Grossman, Alex  A Croatian physicist at the University of Aix-Marseille II in Luminy campus who did pioneering work on wavelet analysis.

H

Haar, Alfred  A Hungarian mathematician who contributed the Haar measure, Haar wavelet, and Haar transform. (October 11, 1885 – March 16, 1933)

I

ICU  Intensive Care Unit – a specialized facility in a hospital that provides intensive care medicine.

Ischemia  A restriction in blood supply, generally due to factors in the blood vessels, with resultant damage or dysfunction of tissue.
Mitral stenosis  A narrowing of the orifice of the mitral valve of the heart.

Morlet, Jean  A French geophysicist who did pioneering work in the field of wavelet analysis.

Myocardium  The muscular tissue of the heart.

P

Papillary muscle  Serves to limit the movements of the mitral and tricuspid valves and prevent them from being inverted.

PCB  Printed Circuit Board.

Purkinje fibers  Located in the inner ventricular walls of the heart, just beneath the endocardium. These fibers are specialized myocardial fibers that conduct an electrical stimulus or impulse that enables the heart to contract in a coordinated fashion.

R

RAM  Random Access Memory.

S

STFT  Short-time Fourier transform or alternatively short-term Fourier transform, where a Fourier-related transform is used to determine the sinusoidal frequency and phase content of local sections of a signal as it changes over time.
T

Tmote Sky  Originally known as Telos RevB or TelosB. A low power wireless sensor module designed at the University of California, Berkeley, by TinyOS developers and manufactured by Moteiv Corp.

V

Ventricle  A heart chamber which collects blood from an atrium (another heart chamber that is smaller than a ventricle) and pumps it out of the heart.
Chapter 1 Introduction

“Signal processing has become an essential part of contemporary scientific and technological activity.”[14] This statement marks an attempt by the author Yves Meyer, in his book, to explain to his readers the status of signal processing in the modern world. Signal processing has always been a fundamental ability in the day to day functions since the beginning of time. Perhaps unnoticed, people have always practiced such skills as reading, listening, and seeing. However, the importance of signal processing has only risen to a new height due to the recent developments in machine automation, which brings out the emphasis focused on by the scope of this thesis.

ECG monitoring and interpretation have always been tasks conventionally assigned to trained medical care personals. Although being more comprehensive in the related knowledge, the constraints to manpower are also very obvious. Fatigue factors and overwhelming workloads are both possible causes to delayed emergency response that may have reduced the chances for patients’ survival. By automating this process, the system frees the medical professionals of the tedious tasks to center their attentions on something much more demanding.

There is still a significant gap between the existing sensor network solutions and the needs in medical care. This thesis is an endeavor to help close the gap by suggesting a solution for a ECG vital sign monitoring system designed to reduce the medical response time for the patients in need. This monitoring system provides many useful applications, giving support to the current medical care structure, especially for ICU patients and under emergency relief situations. Under these scenarios, the physiological statuses of multiple
patients are continuously observed for immediate medical decisions that may well increase their chances of survival.

Based on these motivations, there have been numeral attempts to develop medical systems similar to the proposed work in this thesis. Such efforts are primarily led by the academia but extending deeply into the industries. However, most research efforts have been focusing on either the vital sign monitoring aspect or the ECG feature extraction using standard databases both falling short of expectation. Having analyzed the existing solutions, this thesis vows to bridge the two major research efforts and bring out a more realizable product to directly benefit the consumers in the medical field.

This thesis offers the following contributions to the proposed system, foremost is the wearable ECG monitoring platform (Chapter 3) based on a 3-Lead System and a design under the CodeBlue project [1]. The ECG data collected using these mobile platforms are then transmitted wirelessly using Tmote Sky via radio frequencies to a receiving mote connected to the workstation monitor. These communications are then encrypted to better protect patient privacies [3] (Chapter 4). The received patient data on the workstation is processed using wavelet transforms [10, 14] to provide feature extraction capabilities [7] in order to locate the characteristic points of the ECG waves (Chapter 5). In addition to the above deliverables, another important feature, device location tracking [2], is proposed and studied as a future expansion to the product to ensure the goal of shortened medical response time (Chapter 6).

The rest of this document is organized as the following. Chapter 2 provides some background information on the topics of ECG signals, wavelet transformation theories, as well as others works supporting this thesis. As mentioned above, Chapters 3,4,5,6 offers
more insights regarding the works behind the wearable platform, wireless communication, signal processing, and location tracking, respectively. Chapter 7 abridges the previous four chapters and presents an overview of the entire system, together with feature extraction outcomes. Chapter 8 provides the final remarks of this work, future outlook, and recommendations for any subsequent work.
Chapter 2  Background

This chapter provides a brief introduction to the subjects of ECG interpretation, wavelet analysis, wireless sensor networks, and the supporting work environment. It is by no means a thorough tutorial for any of the above topics, but merely an attempt to provide some basic knowledge necessary to understand the underlying research work that went into preparing for this thesis. If it were found desirable to appreciate more of the in-depth theoretical explanations, please refer to the bibliography section for further reading materials.

2.1. ECG interpretation [11]

ECG is abbreviated from the word electrocardiogram, or alternatively called EKG, which is the abbreviation of the German word elektrokardiogramm. Produced by an electrocardiograph, the signal is constructed by measuring electrical potentials between various points of the body using a galvanometer. ECG signals have a wide array of applications throughout the medical field in determining whether the heart is functioning properly or suffering from any abnormalities. It helps to screen and diagnosis cardiovascular diseases such as ischemia, cardiac arrhythmia, and mitral stenosis, etc.

Figure 2.1 shows an example of a normal ECG trace, which consists of a P wave, a QRS complex and a T wave. A small U wave may also be sometimes visible, but is neglected in this work for its inconsistency. The P wave is the electrical signature of the current that causes atrial contraction; the QRS complex corresponds to the current that
causes contraction of the left and right ventricles; the T wave represents the repolarization of the ventricles; and the U wave, although not always visible, is considered to be a representation of the papillary muscles or Purkinje fibers. [12] The presence or lack of presence of these waves as well as the QT interval and PR interval are meaningful parameters in the screening and diagnosis of cardiovascular diseases. Figure 2.2 below is an example of a healthy ECG which shows clearly all of the components mentioned above.

There are several flavors of the system used to monitor patient ECG information differing primarily on lead placements, ranging from 3-leads to 12-leads. The 3-lead
system is non-diagnostic and is meant for rhythm interpretation, while the 12-lead system, on the other hand, is diagnostic. Although the 12-lead system provides a more thorough coverage of ECG functionalities, it is also more costly both financially and in terms of transport time. Therefore, a 3-lead system is chosen for this application.

2.2. Wavelet Analysis

“The objective of signal processing are to analyze accurately, code efficiently, transmit rapidly, and then to reconstruct carefully at the receiver the delicate oscillations or fluctuations of this function of time. This is important because all of the information contained in the signal is effectively present and hidden in the complicated arabesques appearing in its graphical representation.”[14] This is especially true in this application where the feature extraction of the ECG signals is to locate the interested characteristic points that can be used to detect possible cardiovascular abnormalities. The topic is further complicated, since most of the time the desired ECG signals are either corrupted or embedded in noises. The answer to all of these is wavelet analysis.

Having recently received unprecedented attention in academia, the application of wavelet analysis is still a novel topic dating back only a few years. The name wavelet
itself is only invented in the early 1980s by Jean Morlet and Alex Grossman when they translated the original name ondlette into English. The French word meant “small wave”, and by translating “onde” into “wave”, the name wavelet was first introduced. The development of wavelets can be traced back to as early as the early 20th century when Alfred Haar invented the first wavelet, Haar wavelet, although the importance of this science, at that time, was yet fully understood.

“A wavelet is a waveform of effectively limited duration that has an average value of zero.”[15] Figure 2.3 shows a comparison between a sine wave and a wavelet. Sinusoids do not have limited duration and extends from minus to plus infinity. The smooth and predictable sine wave marks the foundations for Fourier transforms, where signals are transformed into multiple sine waves at various frequencies. On the other hand, the irregular wavelets represents well of signals with sharp changes. Very similar to the STFT, wavelet analysis is actually converting signals into shifted and scaled versions of the mother wavelet.

![Sine Wave and Wavelet Comparisons](image)

Figure 2.3 Sine Wave and Wavelet Comparisons

Same as the Fourier transforms, the wavelet analysis may be divided into two categories, the continuous wavelet transform and the discrete wavelet transform. “The continuous wavelet transform (CWT) is defined as the sum over all time of the signal
multiplied by scaled, shifted versions of the wavelet function $ψ$” [15] or alternatively as shown in Equation 2.1.

$$CWT(a, τ) = \frac{1}{\sqrt{a}} \int s(t)ψ\left(\frac{t-τ}{a}\right)dt$$

(2.1)

In this equation, the parameter $a$ is the scaling factor that stretches or compresses the function. The parameter $τ$ is the translation factor that shifts the mother wavelet along the axis. The parameter $s(t)$ is an integrable signal whose sum is to be multiplied by the translated mother wavelet. And finally, the mother wavelet is denoted by $ψ(t)$, which is a function of the scaling and translation factors just as the result of the continuous wavelet transformation CWT. “The greater the scale factor $a$ is, the wider is the basis function and consequently, the corresponding coefficient gives information about lower frequency components of the signal …” [9] Figure 2.4 shows graphically of this transformation.

![Figure 2.4 Continuous Wavelet Transformation (CWT)](image)

However, living in a digitalize world, it is often desirable to work with discretized signals. By switching into the discrete domain, it is possible to not only save a fair amount of work, but also by choosing carefully of the scales and positions based on powers of two, receive results that are just as accurate. This is called the discrete wavelet transform (DWT) as defined in Equation 2.2.
\[ DWT(m, n) = 2^n \sum_{k} s(k) \psi(2^{-m} k - n) \] (2.2)

In this equation, the parameters \( a \) and \( \tau \) are replaced by \( m \) and \( n \), which are discretized scales and translations where \( a = 2^m \) and \( \tau = n \cdot 2^m \) where \( m \) and \( n \) are integers. The discrete wavelet \( \psi(k) \) may or may not be a sampled version of the continuous wavelet. It is possible to have a discretized wavelet not having a continuous time version.

Often, Discrete Wavelet Transform is also referred to as decomposition by wavelet filter banks. This is because DWT uses two filters, a low pass filter (LPF) and a high pass filter (HPF) to decompose the signal into different scales. The output coefficients of the LPF are called approximations while the output coefficients of the HPF are called details. The approximations of the signal are what define its identity while the details only imparts nuance. Figure 2.5 shows this process graphically, where \( D \) and \( A \) denoting details and approximations, while \( c \) representing coefficients.

The downward arrow in Figure 2.5 symbolizes a process called one-stage DWT of a signal. This is important because without it, the decomposition of signals at each
level produces twice as much data as originally started with. This produces too much burden for the procedure where the original intend was to simplify the problem. Therefore by keeping only one point out of two in each of the two samples, the notion of down-sampling is introduced and providing the additional parameter of coefficients.

Furthermore, the decomposition process is iterative. The approximation signal may be passed down to be decomposed again by breaking the signal into many levels of lower resolution components. This is called multiple-level decomposition and may be represented in a wavelet decomposition tree shown in Figure 2.6. Only the last level of approximation is save among all levels of details, which provides sufficient data to fully reconstruct the original signal using complementary filters.

![Figure 2.6 Wavelet Decomposition Tree](image)

### 2.3. Wireless Sensor Networks

Wireless sensor networks (WSN) research is originally motivated by military applications such as battlefield surveillance. As the field slowly matured and technology rapidly advanced, it has found itself merging into many of the civilian applications as well, such as environment and habitat monitoring, home automations, traffic control, and more recently healthcare applications. Often equipped with wireless communication
devices and microcontrollers, “a wireless sensor network is a computer network consisting of spatially distributed autonomous devices using sensors to cooperatively monitor physical or environmental conditions, such as temperature, sound, vibration, pressure, motion or pollutions, at different locations.” [12]

Figure 2.6 shows three different types of sensor motes that were used in this research work. They are most often referred as motes and come with some sort of a power source. Although their prices and sizes are not yet desirable for the applications they are intended for, the eventual goal is to be able to produce cheap smart dust sensors that can be deployed in quantity over areas of interests.

![Figure 2.7 Sensor Motes - Mica2, Mica2Dot, TelosB](image)

2.4. Work Environment

2.4.1 TinyOS

Developed primarily by the University of California, Berkeley in cooperation with Intel Research, TinyOS is an open-source embedded operating system designed for wireless sensor networks. Written in NesC programming language, TinyOS offers a component-based architecture and is able to operate within the severe memory constraints posted by sensor networks. The copy of TinyOS used in this thesis is version
1.1.15, released in December of 2005. For more details, please refer to the network configuration section of Chapter 4.

### 2.4.2 NesC

NesC is a programming language designed for applications targeting the TinyOS platform. Again by University of California, Berkeley and Intel Research, it is an extension to the C programming language that is component based as the TinyOS operating system. The most important feature of this programming language is that it produces fairly small sized code to be able to load on to sensor network nodes. For more details, please refer to the network configuration section of Chapter 4.

### 2.4.3 MATLAB

Created by The MathWorks and short for Matrix Laboratory, MATLAB provides a high level numerical computation environment and programming language. A powerful and also popular tool in both the industry and academia, it allows for easy toolbox extension using its native M-code programming language. It is intended to “perform computationally intensive tasks faster than with traditional programming languages such as C, C++, and Fortran.”[16] MATLAB also provides support for a Java Virtual Machine (JVM) so it is possible to use the Java interpreter via MATLAB commands, as well as creating and running programs that create and access Java objects. The copy of MATLAB used in this thesis is MATLAB 6.5 Release13 with the wavelet toolbox library. For more details, please refer to the MATLAB server section of Chapter 4.
Chapter 3  Mobile Platform

This chapter focuses on the functionalities and the construction of the hardware mobile platforms. These mobile platforms are essentially the wearable devices that would be distributed among patients in order to offer continuous monitoring of the patients’ vital signs. As shown in Figure 3.1, each platform is composed of a customized sensor board providing connections to a 3-Lead ECG monitoring system, which is housed on a commercially available TelosB sensor mote. While the sensor board gathers useful patient ECG data, the sensor mote provides limited processing capabilities and more importantly wireless communication for transmitting the signals back to the workstation for feature extraction.

Figure 3.1 Mobile Platform
3.1. **TelosB Mote**

The TelosB mote is also sometimes referred to as the Tmote Sky. Designed at University of California, Berkeley, by TinyOS developers, it is an ultra low power wireless module intended for sensor networks applications. Regarded as the next-generation mote platform, it offers the most on-chip RAM of 10kB and also the first to provide IEEE 802.15.4 Chipcon radio with an integrated on-board antenna providing up to 125 meters of range. Constructed around a TI MSP430 microcontroller, the TelosB was the ideal choice for this project for its on-board ADC peripherals with expansion bays, from which the customized sensor board is connected to. The overall block diagram of the TelosB is shown in Figure 3.2 below.

![TelosB Block Diagram](image)
3.2. Sensor Board

3.2.1 Circuit Design

The design of the sensor board is contributed by Harvard University as part of their ongoing research in project CodeBlue.\[13\] As mentioned before, vital sign monitoring has been a frequently visited topic with CodeBlue being one of the most successful in academia. It focuses on the exploration of wireless sensor network technology for a range of medical applications and offers “a combined hardware and software platform that provides protocols for device recovery and publish/subscribe multihop routing, as well as simple query interface that is tailored for medical monitoring” \[1\]. Although not yet complete at the time of this documentation, its design, Figure 3.3, for the wearable ECG platform has seen much utility as the foundation for the work conducted in this thesis.

Figure 3.3 Sensor Board Schematics
3.2.2 Layout Design

The design of this sensor board may be translated into layout diagram as shown in Figure 3.4, which is used for PCB board fabrication, sponsored by PCBExpress. This is a multi-layered layout design. The layers as shown by order are respectively: top silkscreen, top solder mask, top copper, bottom copper, bottom solder mask, and the drill layer. For convenience, the board outline is part of each layer.

![Sensor Board Layout Diagram](image)

**Figure 3.4 Sensor Board Layout Diagram**

The two layers of coppers are laid onto the otherwise none conductive board to create the schematic design onto the PCB. Multi-layered PCBs are formed by bonding together separately etched thin boards. Defined by the solder mask layers, the bare coppers are typically plated with solder to avoid copper oxidization, which would make
the PCB not solderable. On the other hand, the areas that are not meant to be soldered are covered with solder resist. Lastly, silk-screening is also called screen printing, where line art and text may be printed onto the outer surfaces of a PCB. It is often used to indicate individual components. The Gerber file of each layer was sent to the PCBExpress for fabrication together with a drill file that included all the automatic drilling information such as hole sizes and machine coordinates.

### 3.2.3 Electronic Components

A list of components was soldered onto each sensor board PCB according to the schematic shown in Figure 3.3. The complete part list may be found below in Table 3.1, where the only category requiring explanation is “Package”. The different styles of packaging information are the surface mounting types that correspond to the PCB layout choices. The complete sensor board is shown in Figure 3.5 without the TelosB mote.

<table>
<thead>
<tr>
<th>Quantity</th>
<th>Component</th>
<th>Package</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>INA321</td>
<td>8-MSOP</td>
<td>Instrumentation Amplifier</td>
</tr>
<tr>
<td>1</td>
<td>OPA4336</td>
<td>16-SSOP</td>
<td>Operational Amplifier</td>
</tr>
<tr>
<td>2</td>
<td>100k, 5%</td>
<td>0805</td>
<td>Resistors</td>
</tr>
<tr>
<td>3</td>
<td>1M, 5%</td>
<td>0805</td>
<td>Resistors</td>
</tr>
<tr>
<td>3</td>
<td>47k, 5%</td>
<td>0805</td>
<td>Resistors</td>
</tr>
<tr>
<td>2</td>
<td>2.2M, 5%</td>
<td>0805</td>
<td>Resistors</td>
</tr>
<tr>
<td>1</td>
<td>3.3k, 5%</td>
<td>0805</td>
<td>Resistors</td>
</tr>
<tr>
<td>1</td>
<td>4.7k, 5%</td>
<td>0805</td>
<td>Resistors</td>
</tr>
<tr>
<td>1</td>
<td>806, 5%</td>
<td>0805</td>
<td>Resistors</td>
</tr>
</tbody>
</table>
Table 3.1 Sensor Board Part List

<table>
<thead>
<tr>
<th>Quantity</th>
<th>Value</th>
<th>Ref</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>0.1uF</td>
<td>0805</td>
<td>Capacitors</td>
</tr>
<tr>
<td>2</td>
<td>0.1uF</td>
<td>1913</td>
<td>Capacitors</td>
</tr>
<tr>
<td>1</td>
<td>4.7nF</td>
<td>1206</td>
<td>Capacitors</td>
</tr>
<tr>
<td>1</td>
<td>1uF</td>
<td>1210</td>
<td>Capacitors</td>
</tr>
<tr>
<td>3</td>
<td>J539-ND</td>
<td>N/A</td>
<td>MCX Jack</td>
</tr>
<tr>
<td>3</td>
<td>J680-ND</td>
<td>N/A</td>
<td>MCX Plug</td>
</tr>
<tr>
<td>3</td>
<td>Wires</td>
<td>N/A</td>
<td>20 AWG Wires</td>
</tr>
<tr>
<td>3</td>
<td>A14299-AD</td>
<td>N/A</td>
<td>Wire Crimps</td>
</tr>
<tr>
<td>1</td>
<td>A26455-ND</td>
<td>N/A</td>
<td>Connection Receptor</td>
</tr>
</tbody>
</table>

Figure 3.5 Complete Sensor Board

The ECG lead extensions from the sensor board are pin-compatible and color coded to standard 3-Lead ECG monitoring systems. While there are different flavors of physiological chest leads, this system was designed to match any 3-Lead ECG Snap Set.
Leadwires as shown in Figure 3.6. The Snap Set may be used to collect data by attaching to it the appropriate jellied ECG conductive adhesive electrodes if real people were to be used for testing purposes. An alternative would be ECG signal simulators.

Figure 3.6 3-Lead ECG Snap Set Leadwires

3.3. ECG Signal Simulator

Figure 3.7 Model 430B Patient Simulator
The testing simulator of choice in this project is the Model 430B, 12-lead ECG simulator as shown in Figure 3.7. This simulator provides a complete PQRST waveform at six preset rates (60, 75, 100, 120, 150, and 200 BPM) as well as six preset amplitudes (0.1, 0.2, 0.5, 1.0, 2.0, and 5.0 mV). It is also capable of generating square waves using its 5 ECG snaps plus 10 banana jacks. This provides a good testing interface even if this project will be adapted into a 12-lead monitoring system in the future.

Figure 3.8 Mobile Platform Patient Simulation
Chapter 4  Data Communication

This chapter focuses on the communication methods necessary to transmit the patient data collected via mobile platforms to the feature extraction unit on the workstation. This is a two-step process. The first step involves the sensor network communication that takes place between the mobile platforms and the receiving sensor mote connected to the workstation. After this step, all of the useful patient data have been collected and now reside onboard the workstation. The next level of communication occurs within the workstation environment, where a MATLAB server is created to transfer data from a Java runtime environment into the MATLAB workspace via localhost connection. This is the final procedure before sending the patient data for signal processing, which leads to feature extraction.

4.1.  Sensor Network Communication

4.1.1  Network Configuration

Wireless communication is an important makeup of this project for it greatly increases the functional range and mobility of the system. It is also one of the advantages that come along the selection sensor networks as the backbone communication architecture. The sensor mote of choice in this application is the TelosB devices with IEEE 802.15.4-compliant radio capabilities and a range of 125 meters. Other alternatives include the Mica2, Mica2Dot motes, although due to their non-standard radio chips, their
designs are now deprecated in favor of the MicaZ device with standard radio as TelosB does.

These sensor motes operate under the TinyOS environment and may be programmed using the NesC language. Indeed, the TinyOS system and libraries themselves are also written in NesC. This style of program is component-base structured and supports concurrency modeling. NesC applications are built out of components with bidirectional interfaces and are then linked together to form an executable. The concurrency model provides two threads of execution: tasks and hardware event handlers. Tasks are scheduled functions that run to completion without the ability to preempt one another. Hardware event handlers are responses to desired hardware interrupts that preempt executions of tasks and other hardware event handlers.

![Figure 4.1 Wireless Sensor Network](image)

Figure 4.1 Wireless Sensor Network
The wireless sensor network is composed of two groups of devices classified based on their operations. As seen in Figure 4.1, the two groups are mobile platforms and the receiving station. The mobile platforms, as previously explained, are meant for patient data collection and are distributed to the patients as wearable devices. The quantity of mobile platforms in each system may vary depending on the number of patients needs to be observed. There is, however, only one receiving station in each system setup. Connected to the workstation via an USB port, the receiving station is meant for data gathering and actively communicates with each of the mobile platforms in use.

### 4.1.2 VitalDust Plus – Flavor RIT

![VitalDust Plus - Flavor RIT](image)

**Figure 4.2 VitalDust Plus - Flavor RIT**
The software used to govern the sensor network communication and displaying the received patient data on the workstation is based on a program called VitalDust Plus. VitalDust Plus is developed by a group of PhD students at Harvard University led by Dr. Matt Welsh. This software is essentially a stripped down version of the CodeBlue software that provides a simple demonstration of its wireless pulse oximeter and wireless ECK devices. The software has two parts, the TinyOS software for the mobile platforms to sample and transmit vital sign data over the radio, and a Java GUI application to display the vital signs in a graphical form.

VitalDust Plus – Flavor RIT, from now on referred to as Flavor RIT, made several functional additions to the Java applications. The most notable modifications are the inclusions of MATLAB support and the ability to select data, at run time, from only the desired patient for feature extraction. Some of the unused features are also removed from the original graphical user interface. Figure 4.2 shows a screen shot of Flavor RIT while it’s receiving patient data from two separate mobile platforms: mote30 and mote40. The patient data field is displaying the ECG waveform associated with the selected mobile platform. Only data from the currently selected mobile platform are sent to MATLAB for signal processing. The link quality field shows the quality of the wireless signal also associated with the selected mobile platform.

4.1.3 Network Security

While sensor networks have shown promising future, it is not without problems. One of major areas of concern is security. A wireless network provides an open medium allowing any malicious users the opportunities of intrusions, making security even more
crucial compared to the traditional wired networks. The situation is further complicated in the case of sensor networks. Sensor network devices offer limited computation and communication capabilities and have restraint battery life, thus the implementation of conventional security algorithms would also be impractical. However, networking security is especially a concern in this application because lives are at stake if anything were to go wrong.

For these reasons, TinySec, the first fully implemented link layer security architecture is selected. There are two different security options offered by TinySec: authenticated encryption and authentication only. The authenticated encryption option encrypts the data payload and authenticates the packet with a MAC, which is computed over the encrypted data and packet header. In authentication only mode, TinySec authenticates the entire packet with a MAC, but the data payload is not encrypted. [3]

The reason for using the link-layer to provide security measurements is due to the difference between sensor networks and conventional networks. Conventional networks is rarely limited by energy and experience mostly end-to-end communication where the intermediate routers only need to view message headers. However in sensor networks, one of the major concerns is energy preservation. To save power, redundant messages from nearby sensor nodes with similar or correlated environmental events are pruned by intermediate nodes accessing, modifying, or suppressing the contents of messages. Therefore the reading of only the message header would not be sufficient. Another problem of the end-to-end security mechanisms is that message integrity is only checked at the final destination. This is not desirable in sensor networks because the transmitting of adversary messages waste precious energy and bandwidth.
TinySec was designed to provide access control, message integrity, and message confidentiality. Access control means the prevention of unauthorized network access by malicious users. It is important for the sensor nodes to be able to distinguish such packets from the authentic messages to not waste valuable resources processing them. This is also true on the related topic of message integrity where a message from an authorized sender may have been modified in-flight by an adversary. The solution is provided by including a message authentication code (MAC) with each packet. Confidentiality is to provide semantic security, which implies that adversaries should not have more than a 50% chance in correctly answering any yes or no question. This is done via message encryption and helps to keep private information secret from unauthorized users.

The addition of the TinySec security algorithms only introduces minor overhead for the transmission and processing of the messages. The difference between the original data packets and the TinySec protected data packets may be seen in Figure 4.3.

![TinySec and Original Data Packets](image)

(a) TinySec-AE packet format

(b) TinySec-Auth packet format

(c) TinyOS packet format

**Figure 4.3 TinySec and Original Data Packets**
As explained earlier, the fields that have been hatched in Figure 4.3 are protected by the MAC, and the shaded data field in TinySec-AE indicates that the data field is also encrypted. The TinySec security creates only 1 byte of overhead in using the authentication only option and 5 bytes of overhead using the authentication encryption option. During the security options, the CRC bit is replaced by the MAC to provide access control and message integrity. There are some common fields such as the destination address (Dest), active message (AM) type, and length. The active message type specifies the handler function used to decode the messages. The group (Grp) field of the original packet was used to prevent different sensor networks to interfere each other, but it is eliminated in TinySec because of the MAC that already enforces access control. The counter (Ctr) and source (Src) of TinySec-AE are used to provide confidentiality protection.

The implementation of TinySec into existing TinyOS applications proves to be another highlight of this security algorithm. First of all, TinySec is readily available to all TinyOS users for it is already incorporated into the official TinyOS release. In order to apply the algorithm, there are also no required code changes for most usages. If the Makerules from the tinyos-1.x/apps direction is included, the only adjustment is to add the flag TINYSEC=true to the makefile. This can also be done by simply attaching the line to the end of the make command: e.g., make telosb TINYSEC=true. This provides a very basic but robust level of security for typical applications. At a data rate of 19.2 Kbps, it would take over 20 months of MAC forgery attempts to successfully send a forged message, which is certainly over the promised amount of battery life on a sensor network device.
The TinySec default setting provides authentication to all messages, but with encryptions turned off. To use encryption, the TinySecMode interface must be included, and the commands with different modes are shown in Table 4.1 below. The default is TINYSEC_AUTH_ONLY for sending and TINYSEC_RECEIVE_AUTHENTICATED for receiving. The choice of implementation for this thesis is to provide authentication capabilities only to ensure the message integrity and access control on the medical network. However, the encryption option was disregarded out of the consideration that ECG signals contain enough confidentiality to worth the insertion of 4 bytes of additional overhead.

<table>
<thead>
<tr>
<th>Table 4.1 TinySecMode Interface</th>
</tr>
</thead>
</table>

There are two commands in the TinySecMode:

```c
command result_t setTransmitMode(uint8_t mode);
command result_t setReceiveMode(uint8_t mode);
```

**setTransmitMode takes one of three values as its argument:**

- TINYSEC_AUTH_ONLY
- TINYSEC_ENCRYPT_AND_AUTH
- TINYSEC_DISABLED

**setReceiveMode takes one of three values as its argument:**

- TINYSEC_RECEIVE_AUTHENTICATED
- TINYSEC_RECEIVE_CRC
- TINYSEC_RECEIVE_ANY
4.2. MATLAB Server

Although MATLAB has proven itself to be a very powerful instrument in both academia and industry, it does not provide command line support for its functions and libraries outside the MATLAB working environment. This is particularly cumbersome for its intended applications in this thesis. MATLAB and its wavelet toolbox provide a good option for the desired signal wavelet analysis and feature extraction, but the patient data are passed in automatically via a Java application in a complete separate working environment. While MATLAB does provide Java Virtual Machine support, it is not possible the other way around to access MATLAB functions from a Java program outside. To maintain the real-time behavior of this application, the patient data must be passed into the MATLAB workspace promptly for signal processing.

The solution to the above problems is to setup a MATLAB server establishing a connection to the localhost that enables communication within the workstation. A number of additional files are required to make this work classified into the server side and the client side. The MATLAB server is based on a small application named MatlabControl.java developed by Kamin Whitehouse during his studies at University of California, Berkeley. This is a Java program intended to access MATLAB commands while running inside the MATLAB working environment. This is made possible by MATLAB’s support for the Java Virtual Environment and the abilities to execute normal Java programs.

The MATLAB server is based on the MatlabControl file. It establishes a localhost connection and awaits communication from the outside programs. Upon receiving messages, it either redirects them to the appropriate MATLAB functions via
MatlabControl.java, or responds with a predefined solution back to the awaiting clients. One of the problems that exist with running a Java program inside the MATLAB environment is the fact that MATLAB provides only one single thread, therefore the termination of any Java application initiated from inside MATLAB would also exit MATLAB as well.

The client side of program is incorporated into the Flavor RIT application by reading patient data from Flavor RIT and communicates it to the MATLAB server via the established localhost connection. However due to the continuous input of patient data
from the mobile platforms, it is impossible to send all of them at the same time, especially during times when there are more than one connected mobile platform. The design choice was to only send in data associated with the currently selected in Flavor RIT for wavelet analysis after every 600 packets have been collected. This provides a meaningful mediation for data processing and data displaying. A sample extraction result is shown in Figure 4.4.
Chapter 5  Feature Extraction

Feature extraction is a commonly used term in image processing and pattern recognition. It is a form of dimensionality reduction that locates points of interest from a multidimensional space. In the scope of this thesis, feature extraction is conducted by applying wavelet analysis techniques to patient data, thus providing ECG characteristic point detection capabilities. Since most recently published detectors are based on standard database libraries, this real-time application is an attempt to expand the horizons of current research efforts. It also offers a significant function extension to existing vital sign monitoring systems and brings them one step closer to medical care realization.

5.1. Analysis Overview

“The analysis of ECG is widely used for diagnosing many cardiac diseases, which are the main cause of mortality in developed countries.”[9] The automatic detection of ECG wave is an important topic, especially for extended recordings, because it provides many clinical insights can be derived from the information found in the intervals and amplitudes defined by the significant points. The performance of such automatic systems relies heavily on the accuracy and reliability in the detection of the QRS complex, which is necessary to determine the heart rate, and as reference for beat alignments. The QRS complex is the most characteristic waveform of the signal with higher amplitudes. It may be used as references for the detection of other waves, such as the P and T complexes, which are also useful at times. The feature extraction methods applied in this thesis
focuses on the detection of the QRS complex and characteristic points in addition to attempting to locate the associated P and T waves if there are any.

Wavelet transform is a perceived as a very promising technique for this type of applications because it is localized in both the frequency and time domain. It may be used “to distinguish ECG waves from serious noise, artifacts, and baseline drift.”[4] Wavelet transformation represents the temporal features of a signal at different resolution providing better analysis of ECG signals, which is characterized by cyclic occurring patterns at difference frequencies. The wavelet transformation is not difficult to apply as a mathematical tool for decomposing signals. The real difficulty comes at choosing a mother wavelet that optimally fits the signal depending on the application and the signal itself.

It is necessary to consider a few important characteristics when selecting the optimal mother wavelet. “The first two are the ability to reconstruct the signal from the wavelet decompositions and to preserve the energy under the transformation. Another characteristic is symmetry, which is important in avoiding a drift of the information.”[8] Although there are many different proposed wavelet analysis algorithms in academia, most of them have one commonality, which is the selection of discrete wavelet transform over continuous wavelet transforms.

Discrete wavelet transform has its natural advantages when applied towards ECG analysis. Conventionally, ECG feature extraction is preceded by a bandpass or a matched filter to suppress the P and T waves and noises before sending the signal for characteristic detection. By using discrete wavelet transform, “frequency domain filtering is implicitly performed, making the system robust and allowing the direct application over raw ECG
Again, this is made possible due to the nature of the discrete wavelet transform. Discrete wavelet transform is also referred to as decomposition by wavelet filter banks as shown in Figure 5.1.

![Figure 5.1 Filter Banks Analysis](image)

It uses two filtering banks, a low pass filter and a high pass filter to decompose the signals into its different scales. This process may be iterative into many levels as shown in Figure 5.1. It can be seen that the discrete wavelet transform of a signal $x[n]$ is calculated by passing it through a series of filter banks, of which the result may be interpreted as the convolution of the signal itself with the corresponding impulse responses as shown in Equations 5.1 and 5.2. From the equations, $g$ is the impulse response of a low pass filter while $h$ is the high-pass filter. The output from the high-pass filter is regarded to as the detail coefficients and the output from the low-pass filter is regarded as the approximation coefficients. The filters are known as a quadrature mirror filter where “a filter bank which splits an input signal into two bands which are usually then sub-sampled by a factor of 2.” [12]

$$y[n] = (x \ast g)[n] = \sum_{k=-\infty}^{\infty} x[k] \cdot g[n - k]$$  \hspace{1cm} (5.1)
To compensate for the fact that the amount of data would double after each filtering stage due to the dual bank architecture, the filter outputs are down-sampled by two. However, according to Nyquist’s rule, this measure does not compromise the original level of accuracy. In this case, Equations 5.1 and 5.2 may be rewritten as Equations 5.3 and 5.4.

\[ y[n] = (x \ast h)[n] = \sum_{k=-\infty}^{\infty} x[k] \cdot h[n - k] \]  
\[ \text{(5.2)} \]

Equation 5.5 shows the overall discrete wavelet transform using the down-sampling operator ↓.

\[ (y \downarrow k)[n] = y[k \cdot n] \]  
\[ \text{(5.5)} \]

To be more precise, it may be separated into Equations 5.6 and 5.7 below.

\[ y_{\text{low}} = (x \ast g) \downarrow 2 \]  
\[ \text{(5.6)} \]

\[ y_{\text{high}} = (x \ast h) \downarrow 2 \]  
\[ \text{(5.7)} \]
5.2. **Algorithms**

Due to the nature of wavelet transformations the original signal is decomposed into wavelets that are dilations of a mother wavelet by a scale factor. The key to a successful wavelet analysis is by selecting the optimal mother wavelet function that fits the signals taking into consideration the application and the signal itself. This is no simple task. Although wavelet analysis is a rather new method from an historical point of view, it has spurred tremendous interest and, as a result, dozens families of wavelets were born. This section will introduce a few of the basic and prominent ones.

The earliest mother wavelet recording refers back all the way to the early 1900s when Alfred Haar first introduced the Haar wavelet.

![Haar Wavelet](image)

As shown Figure 5.2, the Haar wavelet is the first and simplest wavelet. It is discontinuous and therefore not differentiable. Resembles a step function, it is indeed a special case of the Daubechies wavelet and is also known as db1, where the number represents the order.

The Daubechies wavelets are invented by Ingrid Daubechies, a top wavelet researcher who made practical the fields of discrete wavelet analysis. The Daubechies
family wavelets are represented as dbN, where N is the order. Figure 5.3 shows the next
eight members of the family skipping db1, which is the Haar wavelet.

![Figure 5.3 Daubechies Wavelets](image)

The order number of the Daubechies wavelet may extend to a much higher range. This is a family of orthogonal wavelets defining a discrete wavelet transform characterized by a maximal number of vanishing moments from some given support. It provides solutions for applications such as self-similarity properties of a signal or fractal programs, signal discontinuities, etc. Daubechies went on to propose another wavelet family marked by its symmetrical attribute and are derived from the db family. This new family of wavelets is referred as the symlets and is shown in Figure 5.4. The two families of wavelets exhibit very similar properties.

There are also a wide range of other wavelet families such as the Biorthogonal family that uses two wavelets, one for decomposition and other for reconstruction; the Morlet family that has no scaling functions; the Meyer family whose wavelet and scaling functions are defined in the frequency domain; as well as many other real and complex wavelets. Again the key is to find one that would best represent the signals at hand.
One of the most notable feature extraction algorithms for ECG characteristic-point detection is presented by Cuiwei Li and her colleagues in 1995 at the Biomedical Engineering Institute of Xi’an Jiaotong University. [4] This often referenced publication proposed a multi-scale QRS detector including a method for detecting monophastic P and T waves. The algorithm utilizes the discrete wavelet transform advantage and may be applied over raw ECG signals without any pre-filtering. A quadratic spline wavelet with compact support and one vanishing moment proposed by Stephane Mallat and Sifen Zhong in [18] was used as the mother wavelet.

Its study concluded that “most of the energy of ECG signal lies within the scales of $2^1$ to $2^5$. For scales higher than $2^4$, the energy of the QRS is very low. The P and T waves have significant components at scale $2^5$ although the influence of baseline wandering is important at this scale.” [9] This result is based on the equivalent responses from Figure 5.5 and according to the spectrum of the ECG signal waves by NV Thakor and his colleagues in [19].
This algorithm uses the information of local maxima, minima and zero crossing at different scales. It associates each change in the signal with a line of maxima or minima crossing the scales as shown in Figure 5.6. The algorithm follows four steps: detection of QRS complexes, detection and identification of the QRS individual waves and boundaries, T wave detection and delineation, and lastly P wave detection and delineation. However, only the QRS detector was validated.

Even though this algorithm was not actually implemented, it contributed many fundamental theories in the field of ECG characteristic point feature extractions. The algorithm’s complexity posed challenges in assuring the performance of the proposed system in real-time settings. Its implementation of a non industry standard mother wavelet also made its implementation more worrying for conventional situations. An
alternative method was selected to ensure the performance and easy implementation for this proposed system. The selected algorithm is presented in the next section.

![Figure 5.6 WT at the First Five Scales of ECG-like Simulated Waves](image)

5.3. **Implementation**

The implemented feature extraction algorithm is based on the work presented by S. Z. Mahmoodabadi and his colleagues in [7]. While there are no absolute guidelines in selecting a wavelet family, it is of utmost importance that the wavelet function closely matches the signal to be processed. [20] As described in the last section, there are many flavors of wavelet families available. Although the Haar wavelet provides the benefits of simplicity, it does not take into consideration of the finer details of a signal. The Daubechies Wavelets are conceptually more complex than the Haar wavelet and are
similar in shapes to QRS complex of ECG waves. Their energy spectrum is also concentrated around the low frequencies, making them the mother wavelet of choice in this application.

The specific Daubechies wavelet use in this implementation is db6 as suggested by S. Z. Mahmoodabadi. Figure 5.7 shows a 5-level signal decomposition of a sample ECG waveform using this wavelet. It also includes a comparison between the original ECG signal and the reconstructed ECG signal. This is important because one of the key criteria of a good mother wavelet is its ability to fully reconstruct the signal from the wavelet decompositions.

![Figure 5.7 5-Level Decomposition Using DB6](image)

The high frequency components of the ECG signal decreases as lower details are removed from the original signal. As the lower details are removed, signal becomes
smoother and the noises on the T and P waves disappears since noises are marked of high
frequency components picked up along the ways of transmission. This is the contribution
of the discrete wavelet transform where noise filtration is performed implicitly. This is
explained by the ECG signal frequency distribution, which is shown in Figure 5.8.

![Figure 5.8 Normal ECG Signal and Frequency Distribution](image)

The feature extraction algorithm follows the following steps: R wave detection, Q
and S wave detection, zero level detection, and lastly P and T wave detection. There are
actually four algorithms, each focusing on one certain feature of the ECG signal. The
result of the previous detections may be used as references in the later detections.
Although sequential in nature, all of the algorithms are applied directly at one run over
the entire digitized ECG signals collected using the mobile platforms. The detailed
algorithms of the feature detection methods are listed below.
5.3.1 R Peak Detection

The detection of the R peaks is the first step of feature extraction. The patient data are broken into segments of 600 points and only one segment is analyzed at a time. The R peaks have the largest amplitudes among all the waves, making them the easiest to detect and good reference points for future detections. The signal was processed using the db6 wavelet up to 8 levels. However for the detection of R peaks, only details up to level 2 were kept and all the rest removed. This procedure removed lower frequencies considering QRS waves have comparatively higher frequency than other waves. The attained data is then squared to stress the signal. A threshold equals to 30% of the maximum value is sub-sequentially applied to set a practical lower limit to help to remove the unrelated noisy peaks. At this point, the data set is ready for peak detection through a very simple search algorithm that produces very accurate results.

5.3.2 QS Detection

The detection of Q and S peaks is associated directly with the detection of R peaks. Q and S peaks occurs about the R peaks within 0.1 seconds. Therefore this detection algorithm requires the results from the previous part for setting up windows of interest. Only details up to level 2 are used for searching of the extermum points about each R peak formally detected. The point preceding the R peak denotes the Q peak and the point following the R peak denotes the S peak. “A normal QRS complex indicates that the electrical impulse has progressed normally from the bundle of His to the Purkinje network through the right and left bundle branches and that normal depolarization of the right and ventricles have occurred.” [7]
5.3.3 Zero Level Detection

The zero level of a recording electrocardiogram is the point where there is no current flowing around the heart. This point is difficult to attain because there are many stray currents existing in the body resulting from skin potentials and differences in ionic concentrations in different parts of the body. Conventionally researchers are conditioned to consider the TP segment as the zero point reference level. However, the point right after the end of the QRS complex marks the real zero level potential. This is known as the J point, where even the current of injury disappears. [11] There are two zero level points, one before the Q peak and one after the S peak. Decomposition details of up to level $2^4$ and the approximation level $2^8$ are used for J point detections. Although not part of the actual feature extraction procedure, these points are used as reference points in the detection of the PT waves.

5.3.4 PT Wave Detection

Although the work in this thesis focuses on the detection of the QRS complexes, an effort has been made to provide some information regarding the P and T waves that may present in the ECG signals. PT waves provide meaning supplement information to the QRS complexes in detection of cardiac diseases. The detection of the PT waves in this algorithm require knowledge of the J points while using decomposed signals of only the level $2^8$ approximation signals. P wave is detected before the first J point preceding
the R peaks, while T wave is detected after the second J point following the R peaks. The J points may be considered to be the onset and offset points of the waves, respectively.

5.3.5 Results

All four of the above algorithms are applied at once in real-time to the collected patient data via the mobile platforms. The patient data are segmented by Flavor RIT and the MATLAB server into 600-point packages. Each package is sent to the MATLAB workspace at one time for signal extraction. This process is repeated packet after packet producing the results in a MATLAB figure. Some of the sample results are shown before. Figure 5.9 shows the feature extraction applied to a software generated sample ECG data set while Figure 5.10 shows the feature extraction result applied over the real-time data collected via the mobile platforms.

Figure 5.9 Sample Feature Extraction
Figure 5.10 Real-time Feature Extraction

The red cross sign denotes the R peaks, and as seen, it is located 100% of the time. The Q and S peaks are denoted respectively by black plus sign and black multiplication sign. Although with occasion miss placements (1 in Figure 5.10), they still provide very accurate detections. The black diamonds and black squares denote the P and T waves, they show the approximation of the PT waves if they were present in the ECG signals.
Chapter 6  Location Tracking

This chapter focuses on the potential future expansion of location tracking capabilities for this monitoring system. Location tracking is a desirable ability due to the motivation for creating this system, which is to produce a device that may shorten the medical response time required for patients in need. By knowing the exact positions of the patients, the system can quickly route the medical professionals to the desired locations, thus saving traveling time that could have been used to help the patients. This chapter conducted feasibility studies and environment tests to recommend MoteTrack as the location tracking algorithm of choice. With sufficient hardware devices, it may be easily incorporated into the current systems.

6.1. Introduction

Determining the location of a particular sensor in a wireless sensor network is an extremely difficult problem facing the wireless sensor network research community. GPS is far too expensive a solution for wireless sensor networks. The goal of producing wireless sensor nodes for less than one dollar would be severely compromised. Additionally GPS consumes far too much power to be a realistic localization solution for sensor networks that run on limited battery power.

Much research has been spent in order to solve the localization problem. Some localization algorithms require no infrastructure. These can be launched in hostile territory with no immediate setup procedure. MoteTrack is not such an algorithm.
MoteTrack relies upon beacons with known locations densely spread throughout a known area. Once beacons have been established, further setup is required in that reference signatures must be taken to properly train the algorithm. As such, MoteTrack is not suitable for hostile environments, but is perfect for tracking locations in a predetermined area. Hospitals and office buildings are examples of environments in which the MoteTrack algorithm would be effective and appropriate.

MoteTrack, developed at Harvard University, is a robust, decentralized approach to RF-based location tracking. Its purpose is the accurate location tracking of motes, which are small, lower-power, battery operated devices that can be readily embedded into equipment or the environment. Using radio signal information alone, it is possible to determine the location of a roaming node at close to meter-level accuracy. MoteTrack can tolerate the failure of up to 60% of the beacon nodes without severely degrading accuracy, making the system suitable for deployment in highly volatile conditions.

6.2. MoteTrack

MoteTrack does not rely upon any back-end server or network infrastructure. The location of each mobile node is computed using a received signal strength indicator (RSSI) signature from numerous beacon nodes to a database of signatures that is replicated across the beacon nodes themselves. Each beacon node locally stores a slice of the entire reference signature database.
In MoteTrack, a building or other area is populated with a number of motes acting as beacon nodes. Beacon nodes broadcast periodic signatures, which consist of the format \{sourceID, powerLevel, meanRSSI\}. The sourceID is the unique identifier of the beacon node, powerLevel is the transmission power level used to broadcast the message, and meanRSSI is the mean received signal strength indication (RSSI) of a set of beacon messages received over some time interval. Each mobile node that wishes to determine its location listens for some period of time to acquire a signature, consisting of the set of beacon messages received over some time interval. Finally, a reference signature is defined as a signature combined with a known three-dimensional location \( (x, y, z) \).

The location estimation problem consists of a two-phase process: an offline collection of reference signatures followed by online location estimation. As in other signature-based systems, the reference signature database is acquired manually by a user with a laptop and a radio receiver. Each reference signature, shown as gray dots in Figure 6.1 Example of stored Reference Signatures.
6.1, consists of a set of signature tuples of the form \{sourceID, powerLevel, meanRSSI\}. sourceID is the beacon node ID, powerLevel is the transmit power level of the beacon message, and meanRSSI is the mean received signal strength indication (RSSI) of a set of beacon messages received over some time interval. Each signature is mapped to a known location by the user acquiring the signature database. In MoteTrack, beacon nodes broadcast beacon messages at a range of transmission power levels. Using multiple transmission power levels will cause a signal to propagate at various levels in its medium and therefore exhibit different characteristics at the receiver. In the most extreme case, a slight increase in the transmission power may make the difference between whether or not a signal is heard by a receiver. Varying transmission power therefore diversifies the set of measurements obtained by receiving nodes and in fact increases the accuracy of tracking by several meters.

The MoteTrack algorithm assumes that the most relevant (closest in signature space) reference signatures are stored on the beacon node with the strongest signal. The mobile node sends a request to the beacon node from which it received the strongest RSSI, and only that beacon node estimates the mobile node’s location. As long as this beacon node stores an appropriate slice of the reference signature database, this should produce very accurate results. The communication cost is very low because only one reply is sent to the mobile node containing its location coordinates.

6.3. Deployment

The first step towards implementing the MoteTrack algorithm is to determine the field in which tracking of a mobile mote would be desired. Beacon mote code must be
altered to indicate how often each beacon must transmit their signal, which contains the beacon identification that is used by mobile motes to determine their locations. This length of time is defined by the FREQ_LISTEN_PERIOD constant and must be updated whenever the algorithm is run.

This code must be compiled and loaded to the beacon motes. In the process of loading beacons, one must provide a unique identification number for each one. These beacons must then be placed throughout the environment at predetermined locations which are recorded as coordinates in a map. To make MoteTrack use this map properly the METERS_PER_PIXEL parameter must be defined. This parameter is the conversion between meters and pixels on the map. This is used by MoteTrack to place dots on the map to indicate mobile mote locations. The environment used for this experiment was the Department of Computer Engineering at Rochester Institute of Technology. See Figure 6.2 for the chosen beacons and their locations.
Once the beacons have been placed, the next step is to produce a reference signature database. This is accomplished by programming a mobile mote to collect reference data and provide that data to an attached laptop computer. To program a mobile mote to collect reference signatures, the length of time that the mobile mote will record the transmissions from the beacons it can see for each reference signature point must be defined. This is the DATA_COLLECTION_PERIOD parameter. Once this is set, the mobile mote mode DATA_COLLECTION must be defined preventing the mote from operating in normal mode, and instead will forward all collected data directly to the attached computer.

This code must then be compiled and loaded onto the mobile mote. Once that has been completed, a serial forwarding program must be started to ensure proper communication between the mote and the laptop computer. Finally, the data collection program may be started. This program requires the mobile mote to be moved to various locations within the environment. At certain locations the user may indicate a position on the map, and start the data collection process for that location. This will cause the mobile mote to record power strength and identification numbers from each beacon it is within range of. These signatures are recorded as .dat files to be used later in generating the database. See Figure 6.3 for an example of this program.

A .dat file is produced for each and every location for which a signature is recorded. These dat files must be combined into one and provided to a program provided by MoteTrack that generates two database files to be used in the main MoteTrack program’s algorithms. Using these reference signatures, the algorithm can estimate location based on the power strength and beacon identification. This data collection
process prevents the use of MoteTrack in hostile environments. It is simply not feasible to deploy beacon motes at predetermined locations and create a reference signature database.

The system has now been set up and trained. It is now ready for mobile mote tracking. To accomplish this, the mobile mote must be reprogrammed yet again. The DATA_COLLECTION definition must be replaced by a NORMAL_MODE definition. In addition, the ESTLOC_PERIOD must be defined. This parameter tells the mobile mote how long it must record data to be used in the estimation algorithms. A longer ESTLOC_PERIOD would produce a more accurate result, but would refresh less frequently. Once all of these parameters have been defined, the mobile mote code must be compiled and loaded to the mobile mote.

As with the data collection procedure, the serial forwarder program must be started to enable transmissions. Once live, the main MoteTrack program may be started.
The mobile mote collects data and sends the collected data in large packets to the connected computer. The MoteTrack program running on the computer uses this data and compares it to the reference signature data base to estimate location. See Figure 6.4.

![Figure 6.4 Location Tracking](image)

### 6.4. Problems

Due to hardware limitation, MoteTrack was deployed and tested using only Mica2 and Mica2Dot motes as beacons and mobile mote. However, it was soon realized of the shortcomings of using these devices. The MoteTrack code, designed for TelosB and MicaZ platforms, would need to be updated for the Mica2 family platform. In particular, the RSSI scaling operations were incorrect and required modification. More troubling is the fact that the Mica2 and Mica2Dot motes only support one frequency channel. Typically beacon motes report the frequency channel upon which it is transmitting. When installed on Mica2 and Mica2Dot motes, however, the code was designed to send no channel information at all. On the mobile mote side, whenever a mobile mote receives a
signal it first checks to see if the signal was transmitted on a channel it knows about. If no channel is present it will never record the signal. This means that nothing will ever be recorded by the mobile mote and the algorithm can not run. For this reason, frequency channel reporting must be added to the Mica2 platform beacon mote code manually.

Once these changes were made, the MoteTrack program could be run and data could be recorded. However the performance of MoteTrack was fairly poor. The estimated location tended to gravitate towards the center of the tested area. While this changed as the mobile mote was moved throughout the environment, actual location was rarely accurate. On the positive side, the MoteTrack program also attempts to determine which beacon is the closest to the mobile mote. This was extremely accurate. MoteTrack was almost always able to correctly identify the nearest beacon. This indicates that the received signal strength and beacon reporting operations worked properly.

Communication is also not possible between the Mica2, Mica2Dot platforms used for MoteTrack feasibility testing and the TelosB platform used for monitoring system implementation due to their different radio standards. This has put a final restriction on the inclusion MoteTrack location tracking capability into this release of the product as proposed by this thesis. However this section has shown prominent outlook of the algorithm as a possible future extension, and with sufficient TelosB hardware platforms, it is a fairly simple task to implement the algorithm and also provide acceptable location tracking accuracies.
Figure 7.1 shows an overview of the entire system. It includes two mobile platforms, one patient simulator, one receiving station, the workstation, and several Mica2, Mica2Dot family sensor motes. The patient simulator is used to generate ECG signals for testing purposes, which eliminates the inconvenience of having a live testing subject. One of the two mobile platforms is connected to the patient simulator to collect meaningful data while the other is meant for testing the multi-platform support capability. The receiving station receives the patient and communicates with the workstation via the USB port. The Flavor RIT software picks up the signal, displays it onto the screen, and then sends them via the MATLAB server for feature extraction. The MATLAB program
segments the data into 600-point packets and applies feature extraction algorithms to one segment at a time. The Mica2 and Mica2Dot sensor motes are used for MoteTrack application, although they are not deployed as shown in the picture, they are there to provide an overview of the system.

This chapter will provide instruction on installing the system software. The first step is to start the MATLAB server. After placing the feature extraction folder into system path, start the server by typing `Start_Extract` in the command. This would create a Java object and starts the server connection to localhost on port 4444. This localhost port number may be altered in the MATLAB server and client files if it is already taken by other applications.

![Figure 7.2 Port Forwarding](image)

Figure 7.2 shows the start up procedure of the Flavor RIT software after the MATLAB server is up and running. Due to the using of a sensor mote as the receiving station, it is important to configure the data port for successful data transmission. The
receiving mote is running the generic program TOSBase, which may be found with any TinyOS distribution. The first step is to find out the actual port number assigned to the sensor mote by using the command `motelist`. In Figure 7.2, it is assigned to the COM port 3. The next step is to configure the MOTECOM system variable by issuing the command `export MOTECOM=serial@COM3:telos`. This associates the MOTECOM variable with the serial communication port COM3 at the data rate defined by telos, which is 57600.

After having configured the environment, it is now possible to start the Flavor RIT program by typing in `vitaldust.gui.java.VitalDust` in the command line, and the system would be up and running as shown in Figure 7.3. Once the first 600 data points have been collected, the data would be sent out for feature extraction and a MATLAB window would appear with the associated characteristic points as shown in Figure 7.4 a).
Because there are actually two mobile platforms sending data simultaneously in the current set up, Flavor RIT would have two data tabs available. Figure 7.4 shows the data associated with tab Mote30, the mote connected to the patient simulator. While by switching to tab Mote40, there should no longer be any meaningful data in both windows as shown in Figure 7.4 b). Figure 7.5 shows an enlarged copy of the feature extraction result. The top row displays the original signal while the bottom row shows the feature extractions.
Chapter 8  Conclusion

The objective of this thesis was to take advantage of the modern day technology and create a tele-cardiology sensor network for remote ECG monitoring purposes. While technological advancements have seen changes to many aspects of the daily lives, there is still a significant gap between the existing solutions and the needs in the medical field. One of the most pressing issues in medical care today is the response time to patients in need. Although fully capable of providing cures, many patients pass away due to delays of treatments. Thanks to the recent developments in wireless sensor networks and wavelet signal analysis, such misfortunes may be avoided.

This thesis is an endeavor to suggest a solution utilizing these technologies and provide a remote ECG monitoring system designed for the medical environment. This system provides continuous vital sign monitoring capabilities without the exhaustion of any manpower. In fact, it is intended to give support to the current health care environments and free up medical professionals for more urgent functions. By automating the vital sign monitoring process, the most updated information for all patients are made available at all times.

This system is composed of two major components. Based on wireless sensor network technology, there are the wearable mobile platforms distributed to the patients of concern. These mobile platforms are responsible for gathering patient vital sign using a 3-Lead ECG monitoring system. The gathered data are transmitted wirelessly over radio to the receiving station connected to a workstation where the data are processed.
The second part of the system is based on wavelet analysis theories. This takes place onboard the workstation where patient data are gathered. Feature extraction techniques are applied to the patient data and the characteristic points of interests extracted. These data provide meaningful information for the diagnosis of possible cardiovascular diseases. This is especially useful for extended recordings of ECG signals where human processing is not only time consuming, but also error prone.

In addition to these functionalities, the system is designed to also provide security measurements against malicious attacks and stealing of patient information. More focus is paid on malicious attack than to patient information security. The decision is made based on the fact that any disturbance of the system may cause potentially harm to human lives, while ECG vital sign data do not expose an extended amount of patient privacy. The security measurement implemented is TinySec, which is designed for sensor network applications.

Finally a future expansion possibility is studied for location tracking. This is an important expansion because the original intention of this system is to decrease the amount of time required for medical response to patients in need. By having the exactly patient locations in hand, it is possible to further reduce the response time. The algorithm of choice is the MoteTrack application. The study shows much potential for future inclusion of the application provided with sufficient telosB hardware platforms.

Although there have been many research efforts in both of the fields of vital sign monitoring and ECG signal feature extraction. Most of them stay theoretical at the best. This thesis marks an attempt to bridge the two research fields by providing a product that is more realizable and would directly benefit the consumers in the medical field.
Bibliography


