2011

On Application of Wireless Sensor Networks for Healthcare Monitoring

Elham Dolatabodi

On Application of Wireless Sensor Networks for Healthcare Monitoring

(Spine title: WSNs for Healthcare Monitoring)

(Thesis format: Monograph)

by

Elham Dolatabadi

Graduate Program
in
Engineering Science
Electrical and Computer Engineering

A thesis submitted in partial fulfillment
of the requirements for the degree of
Master of Engineering Science

School of Graduate and Postdoctoral Studies
The University of Western Ontario
London, Ontario, Canada

Certificate of Examination

THE UNIVERSITY OF WESTERN ONTARIO
SCHOOL OF GRADUATE AND POSTDOCTORAL STUDIES
CERTIFICATE OF EXAMINATION

Chief Advisor:

Dr. Serguei L. Primak

Advisory Committee:

Examining Board:

Dr. Abdallah Shami

Dr. Robert Sobot

Dr. Thomas R. Jenkyn

The thesis by
Elham Dolatabadi
entitled:
On Application of Wireless Sensor Networks for Healthcare Monitoring
is accepted in partial fulfillment of the
requirements for the degree of
Master of Engineering Science

Date:

Chair of Examining Board
Firstname Lastname

Abstract

With the recent advances in embedded systems and very low power wireless technologies, there has been a great interest in the development and application of a new class of distributed Wireless body area network for health monitoring.

The first part of the thesis presents a remote patient monitoring system within the scope of Body Area Network standardization. In this regime, wireless sensor networks are used to continuously acquire the patient's Electrocardiogram signs and transmit data to the base station via IEEE.802.15. The personal Server (PS) which is responsible to provide real-time displaying, storing, and analyzing the patient's vital signs is developed in MATLAB. It also transfers ECG streams in real-time to a remote client such as a physician or medical center through internet. The PS has the potential to be integrated with home or hospital computer systems. A prototype of this system has been developed and implemented. The developed system takes advantage of two important features for healthcare monitoring: (i) ECG data acquisition using wearable sensors and (ii) real-time data remote through internet. The fact that our system is interacting with sensor network nodes using MATLAB makes it distinct from other previous works. The second part is devoted to the study of indoor body-area channel model for 2.4 GHz narrowband communications. To understand the narrowband radio propagation near the body, several measurements are carried out in two separate environments for different on body locations. On the basis of these measurements, we have characterized the fading statistics on body links and we have provided a physical interpretation of our results.

Keywords: WBAN, ECG, statistics, PDF, CDF, GMM, MATLAB, TinyOS.

Acknowledgements

First of all, I would like to express my sincere gratitude to my supervisor Prof. Serguei Primak for his invaluable advice and prolific suggestions. The completion of this research would not have been possible without his abundant guidance.

I also would like to thank all my colleagues in the Bell Centre for Information Engineering (BCIE) for their sincere help and friendship during this time. Moreover, I am grateful of the support of faculty of Engineering for Western Engineering students.

I would like to show gratitude of TELUS research and NSERC discovery grants for financial support to this study. We also gratefully acknowledge support from the CMC in providing the Crossbow wireless sensor network kit for our research.

Finally, I am really indebted to my husband and my parents who unremittingly supported me to make this work possible.

DEDICATION

To my beloved husband, Ali.

Table of Contents

List of Tables

List of Figures

Acronyms

WSN	*Wireless Sensor Network*
WBAN	*Wireless Body Area Network*
BAN	*Body Area Network*
ECG	*Electrocardiogram*
TinyOS	*Tiny Microthreading Operating System(TinyOS)*
nesC	*Network Embedded System C*
PS	*Personal Server*
DWT	*Discrete Wavelet Transform*
IIR Filters	*Infinite Impulse Response Filters*
GUI	*Graphical User Interface*
TCP	*Transmission Control Protocol*
IP	*Internet Protocol*
RSSI	*Received Signal Strength*
LQI	*Link Quality Indicator*
PRR	*Packet Reception Rate*
AIC	*Akaike Information Criterion*
PDF	*Probability Density Function*
CDF	*Cumulative Density Function*
GMM	*Gaussian Mixture Model*
EM	*Expectation-maximization algorithm*
SNR	*Signal to Noise Ratio*
BER	*Bit Error Rate*
PER	*Packet Error Rate*
OQPSK	*Orthogonal Quadrature Phase Shift Keying*

Chapter 1
Introduction

During the past decades, advances in miniaturization, low power wireless communication, sensor design, and energy storage technologies enabled development of Wireless Sensor Network (WSN) with pervasive computing capabilities [1]. A WSN is a network of many small sensing and communicating devices called sensor nodes (or motes). Each node has a CPU, a power supply and a radio transceiver for communication [2].

Sensor Networks have shown potential application in changing the way of living ranging from medical to military and from home to industry [3]. There are various applications for WSNs which can be categorized into Health, environment, military and other commercial applications. The two important applications are discussed below:

1. Healthcare monitoring applications, which include (i) activities of daily living monitoring, (ii) fall and movement detection, (iii) location tracking, (iv) medication intake monitoring, and (v) medical status monitoring [4].
2. Environment applications in which the sensors are employed to detect environmental temperature, humidity, and pressure or disasters such as earthquakes, volcanoes, and tornadoes [5].

1.1 Healthcare Monitoring Applications

Pervasive health or patient monitoring systems integrated into a telemedicine system are new information technology that will be able to support early finding of abnormal conditions and prevention of serious consequences. The concept of health monitoring is advanced by which health parameters are automatically monitored at home without disturbing daily activities. The most vital signs monitored are ECG, EEG, pulse oximetry, body temperature, heart rate, and blood pressure.

In this work, the main focus is to develop a prototype system which provides a long-term continuous monitoring of Electrocardiogram (ECG) data; however, we believe that many different body sensors can be potentially integrated to the wireless platform developed by our group. The reason we chose ECG sensor is that Cardiovascular disease is the leading cause of death in the world; Claimed 34.3 percent of all deaths in the United States [6], more than 30 in Asia [7], and an estimated 4.3 million people in Europe [8]. These deaths can often be prevented with proper healthcare and continuous monitoring with the help of WSN. Most of the cardiac abnormalities occur outside of the hospital and new strategies are needed to reduce the time before treatment and to detect the onset of cardiac events. The ECG is the only valuable, noninvasive, easily repeatable, and inexpensive diagnostic tool that is immediately available to assess the probability of cardiac events such as myocardial infarction, ischemia, and ventricular hypertrophy, and it is unequaled in the analysis of cardiac arrhythmias [9]. Also, in the past few decades the clinical information that can be derived from the electrocardiogram has grown continually. Wireless technologies have a significant role in the cardiac patient remote monitoring in different environments such as homes and hospitals.

By definition, ECG records the variations in voltage that are produced by depolarization and repolarization of heart muscles. The variations in voltage are measured at the surface of the body using surface electrodes [10].

Today, three different types of ECG recording are available in general use [11]:

1. In the first one which has been the standard approach for almost half a century, a short sampling of patient cardiac data is needed. For this type of standard ECG, the signs are recorded from 12 leads for a short sampling (lower than thirty seconds).

2. Some cardiac conditions which are irregular or intermittent would not be diagnosed by standard ECG as it provides a snapshot of the patient's cardiovascular activity in time. Continues ECG monitoring should be used for these kinds of diagnosis. Continuous ECG monitoring can be done offline or remotely. In this regard, ECG signals can be stored to be diagnosed offline after completion of data collection. Holter monitors and event recorders became available portable tools for offline ECG recording at hospitals.

3. Although Holter monitors are valuable tools for continuous monitoring, these devices only provide recording and monitoring capabilities and no real-time classification of ECGs as the diagnosis is performed offline. Remote real time diagnosis might be needed to improve the level of diagnosis. Many cardiac abnormalities cannot be diagnosed by offline processing without real-time feedback provided by Holter monitors. In this system, cell phone or personal digital assistants are utilized for collecting the ECG data and sending them to a remote computer [12].

Therefore, significant research efforts on cardiac monitoring in real-time have been started in the past few years. Several research groups and commercial vendors are already developing prototype system for transmitting the vital physiological data such as ECG from a patient to a remote monitoring station. A large number of applications have been developed in US, EU and Asia, deployed at different stages (pilots, clinical trials or regular operation [13]) and operating in different settings (ambulatory, home, mobile, and clinical [14]). The deployment of telehealth system pursues a variety of goals, ranging from organizational to medical improvement [15].

Using telecommunications for remote diagnosis was the basis for several earlier products and projects. However, the first projects such as ECG Holter recorder for the cardiopulmonary diagnostics [16] and the European project "EPI-MEDICS" [17] for the early detection of cardiac events, used wired connection to the patient. EPI-MEDICS [18] is the European project stands for Enhanced Personal, Intelligent and Mobile system for early detection and interpretation of Cardiological Syndromes, uses embedded Personal ECG Monitor (PEM) for the early detection and prevention of cardiac events. Using EPI-MEDICS, a patient can record an ECG data in patient's electronic health record (EHR) on a smart media card in the PEM device. All the data will be processed and in case that a medium alarm situation has been detected the PEM connects to the patient's mobile phone via BlueTooth and sends an alarm message to a dedicated PEM server. The PEM server notifies this event to the patient's attending physician by a SMS (Short Message Service) or an email. The GP receives the SMS message and calls the patient back to tell him that he handles the alarm. He then connects to Internet and reaches quickly and reviews the alarm [17]. However, the system will normally require wired connection to a wearable device.

New wireless technology for telecardiology gives new possibilities for monitoring vital parameters using body worn wireless sensors. Implementation of wearable biomedical sensors, makes patients feel freedom of movement. They will experience a greater physical mobility as it removes cables around patient's body and frees them from the confinement of wires [3]. One example of ECG measuring systems is Physical Activities Healthcare System (PATHS) [19] which has a wearable wireless sensor unit (W2SU) including ECG sensor, dual-axis accelerometer sensor, a hand-held Bluetooth device to receive data from the wearable unit. The wearable unit is equipped with a Secure Digital Memory Card (SD Card) interface and a high speed USB port; Hence, the recorded vital data can be transmitted to the hand-held device for further use or can be displayed on a monitor. Furthermore, the hand-held device can relay data and abnormality alarms to the Internet for the use of healthcare professionals.

With the advent of body worn wireless sensors, the radio channel between sensor nodes became more difficult to characterize due to the stochastic influence of human body. Thus, a number of researches relating to characterization of on-body propagation have appeared in the literature. These literatures investigated how the human body affects the wireless connection link qualities by characterizing the wireless radio propagation between the nodes placed on human body. Propagation model can be characterized by study the path loss exponent under various condition [20]-[27]. Another related studies focus on the variability of the signal power in close spatial proximity to a particular location and fading statistics [28]-[31]. Furthermore, some literatures explore the impact of realistic channels on network level performance and link estimation metrics [32]-[34].

1.2 Thesis Organizations

Current advances in wireless communications integrated with development in pervasive and wearable biomedical sensor technologies would have a radical impact on healthcare monitoring systems.

In this study, we develop a wireless body area network for health monitoring which supports ECG wearable sensor and contains an online personal server for real-time data acquisition, visualization, memorizing, analyzing, and transmission to local network and internet. We will also characterize the narrowband body area propagation environment through measurements in order to predict the link level performance. In this work we try to bridge the gap between wireless sensor networks and the effect of human body on wireless links.

This thesis is organized in two main parts. The first part - Chapter 2 provides a background on Wireless Sensor Network and Chapter 3 introduces the WBAN architecture that can meet the requirements for real-time ECG streaming and monitoring. Our system supports following requirements: (i) periodic transmission of ECG vital signs, (ii) addressing practical issues such as user comfort and mobility by deploying wearable and portable sensors and wireless platforms, (iii) displaying the ECG streams on a friendly user GUI and (iv) providing high speed access to the local networks such as internet. In terms of real-time remote access to the ECG data, our developed system is totally adopted for immediate data transmission. In this study, the personal server as an interface between sensor nodes and local network is implemented on MATLAB environment for a first time. Finally, the proposed system implementation and some practical results will be shown at the end of this chapter.

The second part - Chapter 4 is devoted to the study of indoor body-area channel model for 2.4 GHz narrowband communications. In order for a Body Area Network (BAN) to work in the preferred manner, it is essential to assure that the communication between the mounted sensor nodes and base node takes place in the right way. We also considered the nature of local environment on the performance of radio communication systems. The measurement set up, environment, and procedure used for extracting channel parameters for two separate environments are all described in this chapter. The main contributions, we obtained are summarized in the followings;

1. Identification of link estimation metrics (RSSI, LQI, and PRR) for several body locations in the real channel condition.

2. Characterization of the fading statistics on body links based on the variation of received signal level (RSSI) over different locations and times.

3. Evaluation of the bit error rate (BER) performance of the narrowband-based WBAN system.

In addition to the main parts, the remainder of the thesis focuses on conclusion and two Appendixes. The step by step software design manual for developing a Health Monitoring System based on WSN using the IRIS platform in TinyOS is presented in Appendix A. The Appendix B is a collection of MATLAB functions built on the MATLAB technical computing environment. It is a framework for communicating with motes supporting the serial port interface.

Chapter 2

Background

In the following chapter, we will discuss the notion of a wireless sensor network, its applications, and related challenges. The wireless body area network, the hardware and software platform, commonly designed for health application of wireless sensor network will also be considered.

2.1 Wireless Body Area Network

Integration of low power wireless sensor network devices into medical settings gives raise to Wireless Body Area Network (WBAN) for measuring and monitoring patient's vital signs. WBAN is a radio frequency based wireless networking technology that interconnects intelligent nodes capable of sensing, sampling, processing, and communicating of biological signals, attached on or around human body [35]. This allows people in the hospital, at home, or on the move to monitor changes in their physiological parameters. Therefore, the need for face-to-face contact with the care professional will be reduced and patients feel freedom of movement. They will experience a greater physical mobility as it removes cables around patient's body and frees them from the confinement of wires [4]. WBAN has also made the costs for pervasive healthcare systems inexpensive. Three group of people are interacting with WBAN system [3].

Chronically Ill: Chronic conditions are health conditions that either have symptoms on a constant basis or flare up episodically. Such as: diabetes, heart disease, pulmonary conditions, hypertension, mental disorders, stroke and cancer. Based on last statistics, 162 million people are suffering from chronic conditions in the United States [36]. Chronic diseases are among the most prevalent, costly and preventable of all health problems. According to the Centers of Disease Control and Prevention, seven in ten Americans who die each year die of a chronic disease [37]. Hence, Real-time monitoring of patients' physiological status and analysis systems seem crucial for this group of people as it helps chronically ill people with different degrees of cognitive and physical disabilities will be capable of having more dependent life. In this regime, they will be able to closely monitor changes in their vital physiological signs and bring feedback to maintain an optimal health status. For real-time systems, the physician will have access to vital signs of chronic patients at their home without need of hospitalization. Moreover, emergency situation will be identified with a little latency and in a few seconds.

Elderly: this group includes peoples being past middle age and approaching old age. 80% of older adults have at least one chronic condition, and 50% have at least two [38]. Besides, they are more susceptible to sudden falls. Health monitoring and analysis systems are crucial for many older people live independently and avoid expensive trips to the emergency room or nursing homes.

Children: this group is known as a person younger than the age of majority who are not capable of taking care of themselves like babies, infants, toddlers and needed to be continuously monitored.

Healthcare monitoring application of Wireless Body Area Network involves a number of issues and challenges including needs to interoperability, privacy and security, low-power communication, biosensor design, power consumption, communication link between the implanted device and external monitoring control equipment which needs to be resolved. Some of leading challenges are listed below,

- Security, Privacy, and Reliability: Security must be guaranteed all the way through the healthcare application state. In terms of security requirements, the patients' sensitive health information must be viewed only by authorized parties, resistant against security tracks. Moreover, the application must ensure a well defined degree of privacy with precisely planed rules. The application should also be reliable. Reliable data communication, analysis, and measurement are the three main categories involved in the issues of reliability.

- Node size: It is essential for the wireless medical sensors to be light weight and small since they should be worn by patients. The size and weight of batteries determine the node sizes. However, the batteries' capacity depends on their size. With advances in technology and integrated circuits, it can be expected to have efficient sensors with small sizes.

- Energy efficiency: as the sensor nodes are designed to be used for a long period of time, the power sources should be efficient enough and long lasting [5].

2.2 WSNs Software Platform

TinyOS [39] is an open-source operating system designed for wireless embedded sensor networks. However, Programming TinyOS can be challenging because it requires using a new language, nesC.

2.2.1 Tiny Microthreading Operating System(TinyOS)

TinyOS features a component-based architecture, which enables rapid innovation and implementation while minimizing code size as required by the severe memory constraints inherent in sensor networks. TinyOSs component library includes network protocols, distributed services, sensor drivers, and data acquisition tools all of which can be used for a custom application. TinyOSs event-driven execution model enables fine-grained power management yet allows the scheduling flexibility made necessary by the unpredictable nature of wireless communication and physical world interfaces.

TinyOS is not an operating system (OS) in the traditional sense; it is a programming framework for embedded systems and set of components that enable building an application-specific OS into each application. The reason for this is to ensure that the application code has an extremely small memory foot print. In addition TinyOS is designed to have no file system, supports only static memory allocation, implement a simple task model, and provide minimal device and networking abstractions.

TinyOS has a component-based programming model (codified by the nesC language). Like other operating systems, TinyOS organizes its software components into layers. The lower the layer the closer it is to the hardware; the higher the component, the closer it is to the application. A complete TinyOS application is a graph of components, each of which is an independent computational entity [40].

2.2.2 Network Embedded System C (NesC)

The TinyOS operating system, libraries, and applications are all written in nesC, a new structured component-based language. The nesC language is primarily intended for embedded systems such as sensor networks. The nesC has a C-like syntax, but supports the TinyOS concurrency model, as well as mechanisms for structuring, naming, and linking together software components into robust network embedded systems. The principal goal is to allow application designers to build components that can be easily composed into complete, concurrent systems, and yet perform extensive checking at compile time.

2.3 WSNs Hardware Platform

The sensor network hardware platforms usually consist of three components:

2.3.1 IRIS mote

In this experiment, the IRIS developed by UC Berkeley and manufactured by Crossbow Technology [41] operates as a primary embedded platform for the ECG sensor. The IRIS is the company's latest generation of Motes. The name "motes" refers to a general class of technologies comprising an embedded microcontroller and low-power radio. In comparison with previous company's motes, the IRIS Mote platform provides more improved RF range, lower sleep current, and double the program memory. Each IRIS board makes use of ATmega1281 microprocessor [42] as data processing unit and AT86RF230 [43], a low-power 2.4 GHz radio transceiver especially designed for ZigBee/IEEE 802.15.4 as data transmission unit. The Atmel ATmega128/128L devices are members of the Atmel's 8-bit AVR microcontroller family. Peripherals include a 10-bit analog to digital converter, real time clock, timers, and asynchronous/synchronous serial interfaces (with SPI and I2C modes). The ATmega128L device is specified for supply voltages of 2.7 through 5.5 volts. Internal memory includes 128 KBytes program flash, 4 KBytes SRAM, and 4 KBytes EEPROM. IRIS has programmable output power from -17 dBm up to 3 dBm and receiver sensitivity

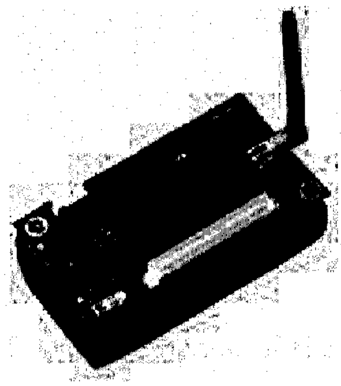


Figure 2.1: Photo of the XM2110IRIS with standard antenna.

of -101 dBm. As a power supply, two AA batteries are typically used, but a mote is powered through USB bus if connected to an interface board.

2.3.2 RF230 transceiver

The radio used by the IRIS is an IEEE 802.15.4 compliant RF transceiver designed for low-power and low-voltage wireless applications. It uses Atmel's AT86RF230 radio that employs O-QPSK ("offset quadrature phase shift keying") with half sine pulse shaping. The 802.15.4 radio includes a DSSS (digital direct sequence spread spectrum) baseband modem providing a spreading gain of 9 dB and an effective data rate of 250 kbps. The radio is a highly integrated solution for wireless communication in the 2.4 GHz unlicensed ISM band [41].

2.3.3 MIB520 USB interface

The MIB520 provides USB connectivity to the IRIS and MICA family of Motes for communication and in-system programming. It supplies power to the devices through USB bus. MIB520CB has a male connector while MIB520CA has female connector.

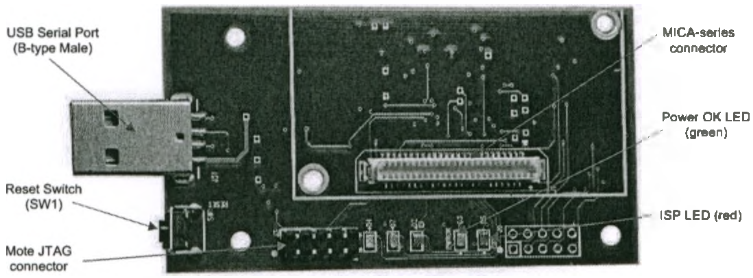


Figure 2.2: Photo of top view of the MIB520CB.

2.4 Summary

The IRIS Mote operates as a primary embedded platform for the ECG sensor in our system. In February 2008, Crossbow announced the availability of the TinyOS 2.x Operating System for Crossbow's advanced IRIS Motes. IRIS Motes are also supported by Crossbow's MoteWorks software development environment based on open-source TinyOS 1.x. We preferred to run our application in the TinyOS 2.x Operating System because it provides a better hardware abstraction model, improved timers, sensor interfaces, power management, arbitration, and much more. Our developed wireless body area network for health monitoring is composed of two IRIS motes, one MIB520 USB interface board, and one analogue ECG sensor. The personal Server (PS) which is responsible to provide real-time displaying, storing, and analyzing the patient's vital signs is developed in MATLAB. It also transfers ECG streams in real-time to a remote client such as a physician or medical center through internet. The complete hardware and software architectures for developing a healthcare monitoring instrument using a Wireless Sensor Network are given in Chapter 3.

Chapter 3

WBAN System Architecture

3.1 Introduction

With the growing needs in communications and recent advances in very-low power wireless technologies, there has been considerable interest in the development and application of wireless networks around humans. The wearable sensors technology as an important development in the healthcare monitoring system has been focused by several international researches. Body worn devices make patients experience a greater physical mobility as they remove cables around their body.

Fensli *et al.* [44] measures the ECG-signal by a wearable sensor which continuously transmits the signal using RF radio transmitter to the Personal Digital Assistant (PDA). In case of observing abnormalities, PDA will start transmitting the recorded ECG-signal to the Clinical Diagnostic Station. The system is supposed to be able to detect the rare occurrences of cardiac arrhythmias and to follow up critical patients from their home [45]. However, the system prototype seems to be too large for routine use by out patients. Similarly, MobiHealth, the European Commission's wide-ranging project started in May 2002 [46] has developed health body area network system by which the patients can be monitored and receive medical care in emergency situations. They were successful in designing and implementing a Health Body Area Network service platform by developing a concept of a 3G enabled Body

Area Network [47]. In a recent project, CodeBlue [48], performed at Harvard University, the combined hardware and software sensor Networks platform for Medical Care has been developed. CodeBlue is one of the most comprehensive projects including combination of mote, ad-hoc network [49], multi-hop communication design, and location tracking system called MoteTrack [50]. The hardware design part has composed of three mote-based sensors; pulse oximeter, ECG [11], and motion sensors. These sensor boards are connected to commercially available Mica2, Micaz, and Telos Motes. They also developed a new miniaturized sensor mote designed for medical use. CodeBlue is a protocol and middleware framework based on publish/subscribe routing framework in which sensors publish data to a specific channel and end-user devices subscribe to channels of interest [51]. A multi-tier WBAN system prototype has been also proposed by Otto *et al.* [52] and Jovanov *et al.* [53]. Their system handles data transmission between the WBAN and a medical server. The communication between the sensor nodes and network nodes is single-hop, slotted and uses ZigBee or Bluetooth. The Zheng *et al.* [13] provided long-term continuous monitoring of cardiovascular patients using wearable sensor with ability to automatically launch an emergency call once it detects abnormal situations. Their Mobihealth care system composed of three parts; the wearable sensor called WS, the PPU providing real-time collection and analysis and MSC which is composed of a MDS (medical data server) and MT (monitoring terminals). In new studies, the work carried out by Chipara *et al.* [54] is similar to CodeBlue project. They have developed a system for detecting clinical deterioration using WSN based on real-time vital signs. Moreover, Yan *et al.* [55] developed a WSN based e-health system which estimates the location of targets without making any interference to their normal life.

Despite the increased interest in the WSN areas, there are few studies on the system development with 'real-time' remote access to the vital signs via internet which is the other essential factor for Health monitoring. In terms of real-time data transmission through local network, majority of previous works focused on sending an alarm to the medical center in case of observing abnormalities while our system is completely adopted for real time data transmission. To the best of our knowledge, this is a first work which develops MATLAB based system for interaction with sensor network nodes. MATLAB contains advanced numerical computing ability, powerful libraries and elaborate toolboxes for plotting and analyzing data in real-time that eases data processing operations.

In this chapter, we will introduce the WBAN system developed in the Bell Centre for Information Engineering (BCIE) for remote patient monitoring in real-time and illustrated in Figure 3.1. The prototype system developed by our group can be functionally divided into two subsystems: (i) WBAN nodes, and (ii) Personal Server, being presented in this chapter. The primary function of WBAN nodes is to sample ECG signs and transfer the data to a personal server through IEEE 802.15.4. The Personal server implemented on computer controls, display the data and also transfer vital information to the local network through internet.

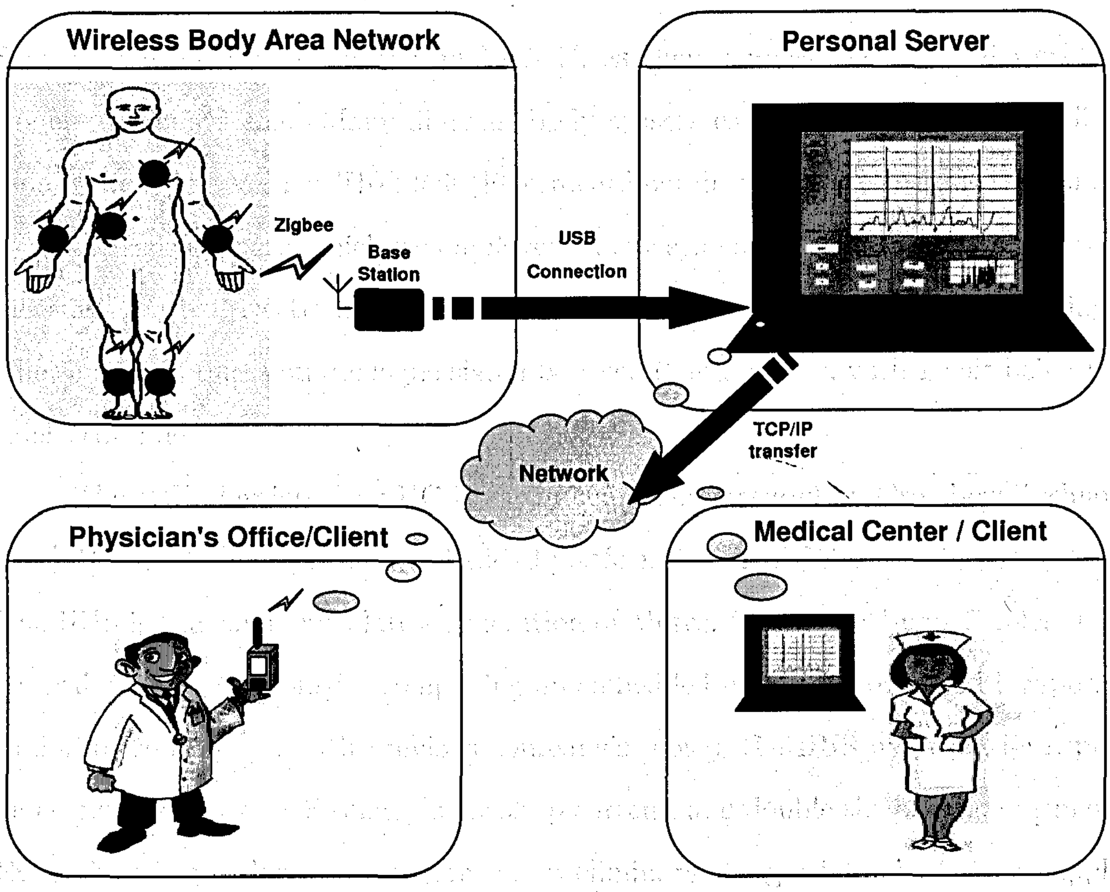


Figure 3.1: WBAN System for real-time patient monitoring.

3.2 Wireless Body Area Network nodes

This section describes development of a combined hardware and software platform for the Wireless Body Area Network nodes.

3.2.1 Hardware Design

Sensors play an important role in WBAN, as they connect the physical world to the electronic systems. Many different body sensors can be used in remote medical monitoring. The Vernier EKG [56] (Electrocardiogram or ECG) Sensor is the analog wearable sensor board which we employed in our system to acquire ECG signals. It measures cardiac electrical potential waveforms for standard 3-lead ECG tracings. The sensor produces an analogue signal between 0 and 5 volts, with 1 volt being the isoelectric line.

The IRIS developed by UC Berkeley and manufactured by Crossbow Technology [41] operates as a primary embedded platform for the ECG sensor in our system. The IRIS is the company's latest generation of Motes. The name "motes" refers to a general class of technologies comprising an embedded microcontroller and low-power radio. In comparison with previous company's motes, the IRIS mote platform provides more improved RF range, lower sleep current, and double the program memory. Even though crossbow motes were not preliminary designed for body-worn applications, they have been successfully used in several body health monitoring projects such as Harvard's CodeBlue project [48], remote monitoring cardiac activity [57], profiling the body channel for patients with chronic illnesses [58], and finally mounting the motes onto fast-moving athletes [59].

The block diagrams of hardware platforms for WBAN nodes are indicated in Figure 3.2. Each IRIS board contains ATmega1281 microprocessor [42] as a data processing unit and AT86RF230 [43], a low-power 2.4 GHz radio transceiver especially designed for ZigBee/IEEE 802.15.4, as a data transmission unit. The RF230 single-chip radio transceiver provides a complete radio transceiver interface between the antenna and the microcontroller. It comprises the analog radio transceiver and the digital demodulation including time and frequency synchronization, and data buffering. The AT86RF230 is designed for low-power and low-voltage wireless applications. All RF-critical components except the antenna, crystal and de-coupling capacitors are integrated on-chip. Therefore, the AT86RF230 is particularly suitable for 2.4 GHz IEEE 802.15.4 and ZigBee systems applications. The transmit modulation scheme is OQPSK with half-sine pulse shaping and 32-length block coding (spreading).

The IRIS platform is powered by 2 AA batteries and has a detachable, quarter wave, monopole antenna and a 51-pin expansion connector for external sensor boards. We choose to connect our ECG sensor to Crossbow's IRIS platform via the MDA100 [60] sensor board prototyping area which supports connection to all eight channels of the mote's analog to digital converter node (the integration is shown in Figure 3.3). On the patient body, the ECG sensor node is mounted on the waist and lets the subject feel freedom of movement (Figure 3.4). The ECG signs are acquired by the sensor node and transmitted to the base station in realtime. The base station (Figure 3.5), is composed of MIB520 [41] and is responsible for data collection, communication and in-system programming to PC.

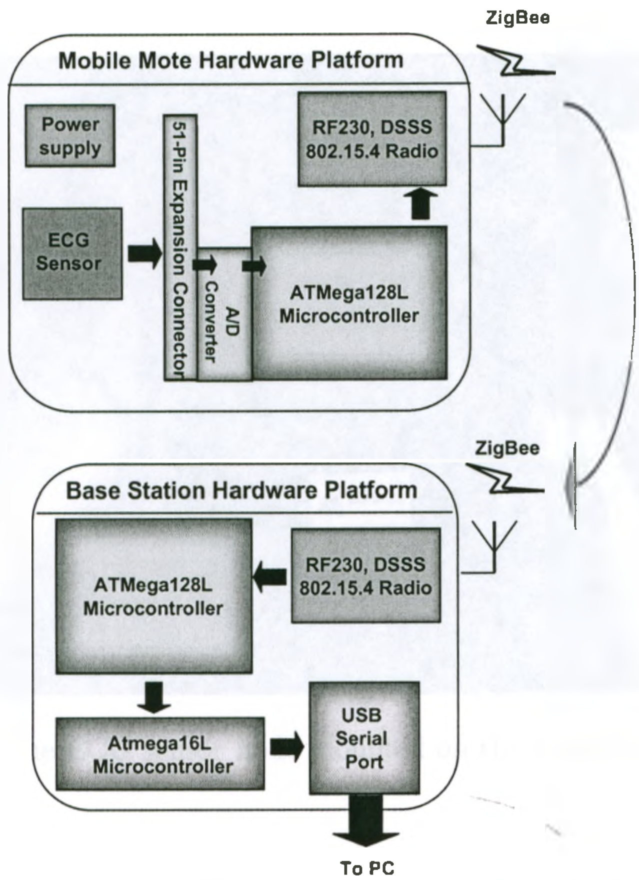


Figure 3.2: The block diagrams of hardware platforms for WBAN nodes.

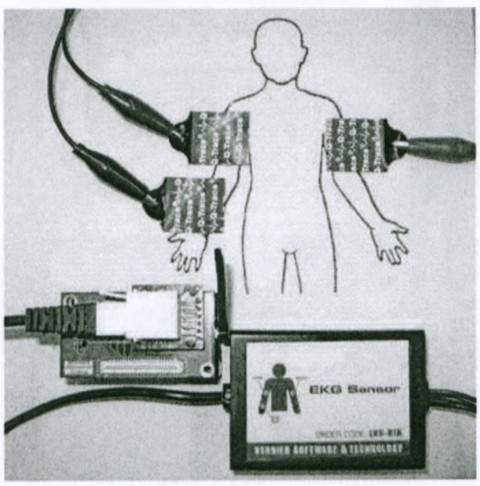


Figure 3.3: The Vernier EKG sensor integrated to IRIS mote.

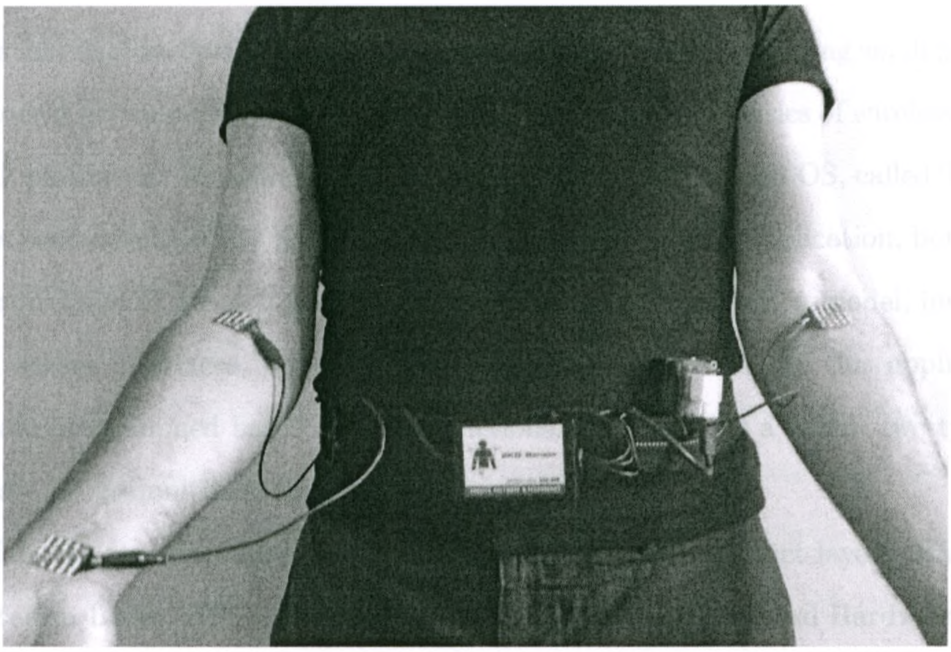


Figure 3.4: The ECG sensor node mounted on the waist of the subject.

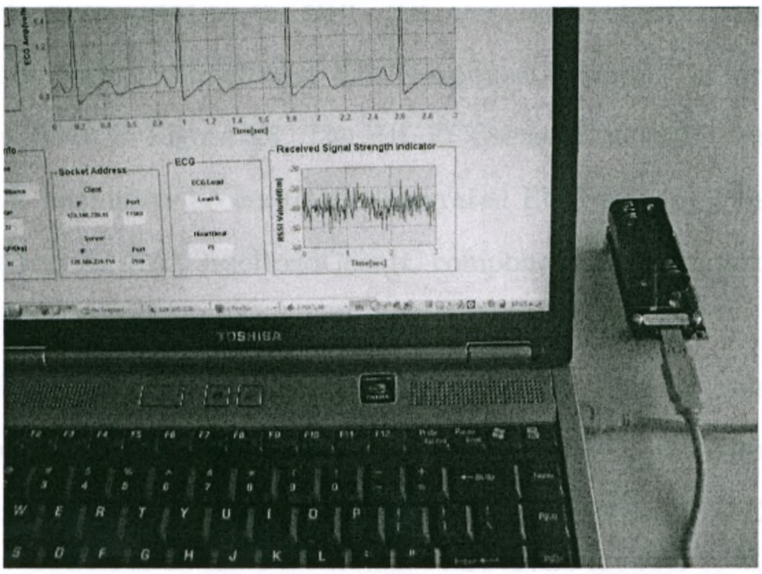


Figure 3.5: The base station which is composed of MIB520.

3.2.2 Software Design

It is the fact that an operating system framework is needed for retaining small physical size, modest active power load, and tiny inactive load characteristics of wireless sensor network platforms. To address this problem, a tiny microthreaded OS, called TinyOS [39] has been developed by researchers at UC Berkeley. In this application, both IRIS motes run TinyOS 2.x as it provides a better hardware abstraction model, improved timers, sensor interfaces, power management, and arbitration. In this application, motes are programmed based on their functions and tasks. As a result, two types of motes are programmed: Mobile and Base.

TinyOS organizes its software components into three distinct layers: Hardware Presentation Layer (HPL), Hardware Adaptation Layer (HAL), and Hardware Interface Layer (HIL) [61]. Each layer has clearly defined responsibilities and is dependent on interfaces provided by lower layers. All components in HPL layer have access the hardware in the usual way (by I/O) and are Hardware dependent. Components in HAL use interfaces provided by the HPL. HIL contains hardware-independent interfaces. The block diagram of software HIL components running on both mobile and base motes are shown in Figure 3.6. The HIL ECGsensorC.nc component is a generic component that virtualizes access to the integrated ECG sensor. This generic component is linked to the HAL AdcReadClientC component providing arbitrated access to the Atmega128 ADC.

The Mobile IRIS mote that is connected to an ECG sensor runs an appropriate software which uses the ADC to sample the analog ECG data from the sensor, construct a message out of them, and send it over IEEE 802.15.4 (ZigBee) [9] to the IRIS mote connected to the base station. The ADC channel continuously reads the ECG data and when enough ECG samples are collected in the message buffer, the application passes the message to the networking stack. ECG usually digitized at sampling frequency of 500-1000 Hz. Even though 1 kHz is less than Nyquist frequency for ECG signal, fine temporal precision is needed with respect to expected variations in cardiac period. The sampling rate and packet transmission rate over the network are dependent on each other such that a higher sampling rate requires higher packet transmission rates.

One of the challenges is to determine the suitable sampling rate of the ECG signal. On one hand, this is due to limitation of IRIS mote packet transmissions rate. IRIS mote can handle only 200 packet transmissions in a second which leads to a period interval of 5 ms for each packet. In order to satisfy desired sampling rate, each packet includes ten ECG data with a period interval of 1.4 ms. On the other hand, one has to find appropriate TinyOS Source and Sink Independent Driver (SID) interface compatible with ECG sampling rate. As shown in Figure 3.6, the application uses TimerMilliC which gives an independent millisecond granularity timer. On the IRIS motes, the 32 KHz external clock is divided by 32, so the best resolution we can get from that timer is 1 ms. To avoid timing issues with IRIS, a lower limit of 5 ms should be used. As a result, we have two options to get 1.4 ms sampling interval from Timer hardware in Atmega128: (i) using the Alarm interface which is asynchronous, and (ii) running the ADC module in streaming mode. As ECG provides a continuous stream of data, we programmed an IRIS to read data in block instead of individually by using ReadStream SID interface.

Moreover, the motes are programmed to obtain link estimation metrics such as the received signal strength indicator (RSSI), packet error rate (PER), and sequence numbers. The packet number, showing the sequence number, is increased by one each time it is sent to the destination node. The sequence number is utilized to identify a packet within a burst. The received signal strength (RSSI) and link quality indicator (LQI) information are also requested for each packet by the receiver. The RSSI is an estimation of the received signal power within the bandwidth of an IEEE 802.15.4 channel. The IEEE 802.15.4 standard determines the LQI measurement as a characterization of the strength and quality of a received packet. The LQI values are associated with an expected packet error rate (PER). All data (20 bytes for the ECG data, 1 byte for RSSI, 1 byte for LQI, and 1 byte for Counter (showing sequence numbers)) are collected into one TinyOS message and then sent to the base station (see Figure 3.7). A TinyOS message is a generic message structure with a reserved payload for application data. More details on our software design is performed in our application note [62] and has been presented in Appendix A.

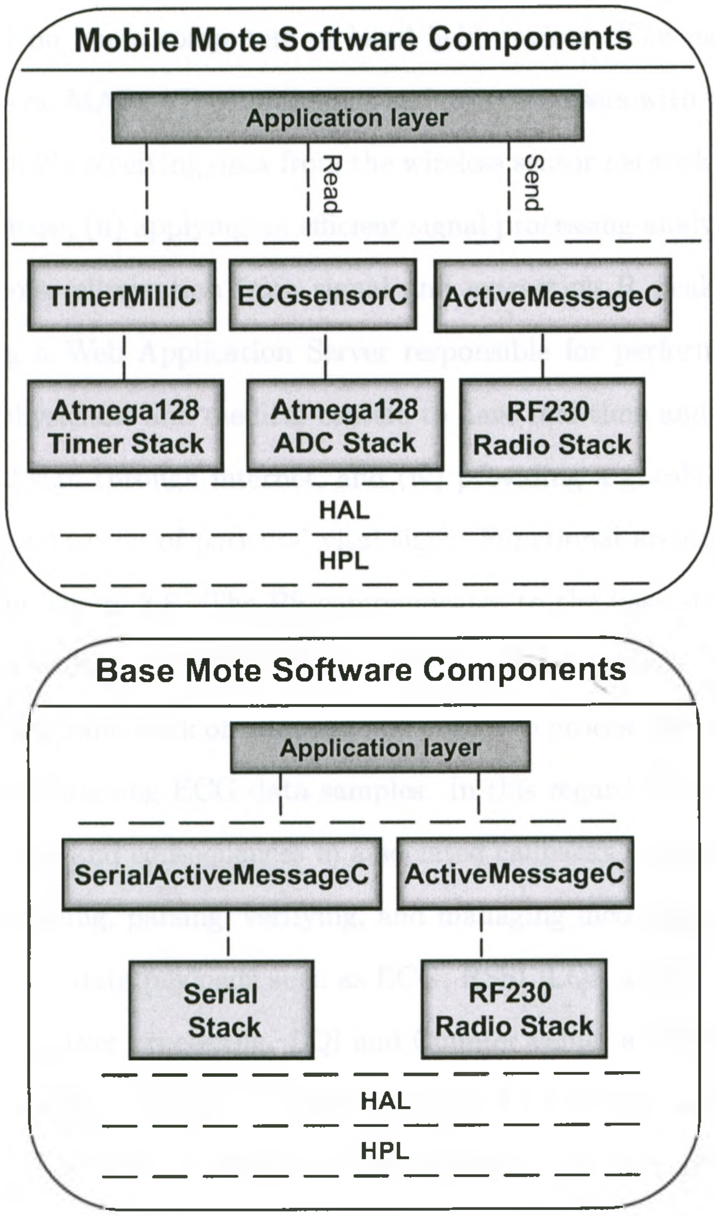


Figure 3.6: The block diagram of software HIL components running on both mobile and base motes.

3.3 Development of Personal Server

The personal server (PS) which is the core processing element has the potential to be implemented on home computers or hand held devices. The multithreading PS application runs on MATLAB environment and provides users with the following services: (i) continually receiving data from the wireless sensor network and storing it in a specified database, (ii) applying an efficient signal processing analysis on ECG data for powerline noise elimination from signal and extracting R peaks of ECG wave, (iii) establishing a Web Application Server responsible for performing an interface which enables physicians and medical centers to have real-time and continues access to patients vital sign through internet, and (iv) providing a graphical user interface for real-time visualization of patients' vital signs. Functional architecture of PS services is shown in Figure 3.8. The PS communicates to the base station using serial port interface provided by MATLAB Instrument Control Toolbox. PS deploys dedicated threads with framework of callbacks and events to process the incoming TinyOS 802.15.4 frames containing ECG data samples. In this regard, an event occurs after a packet is received and consequences in associated callbacks. Callback functions are responsible for reading, parsing, verifying, and managing incoming packets. In addition to reading, the data payloads such as ECG, RSSI, LQI, and Counter values will be extracted for further processing. LQI and Counter values are utilized for tracking lost packets in a packet stream. A buffer is assigned for storing packet's contents (a FIFO-first in first out) which isolates the processes of reception and storing. Signal processing (Digital notch filtering and ECG peak extraction) and TCP/IP algorithms are executed in their own callbacks and applied to all packets saved in FIFO buffer in real-time.

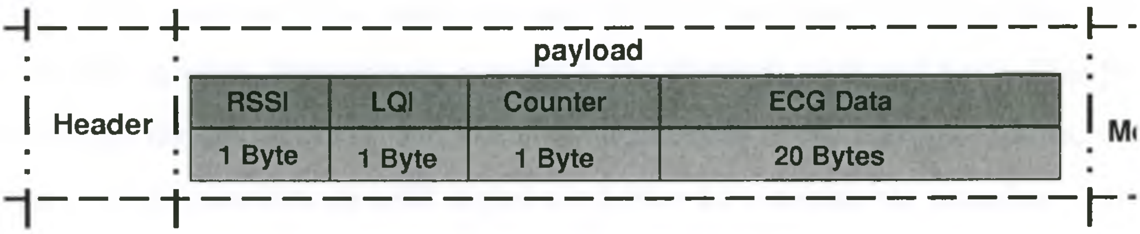


Figure 3.7: Tinyos message fields and payload contents.

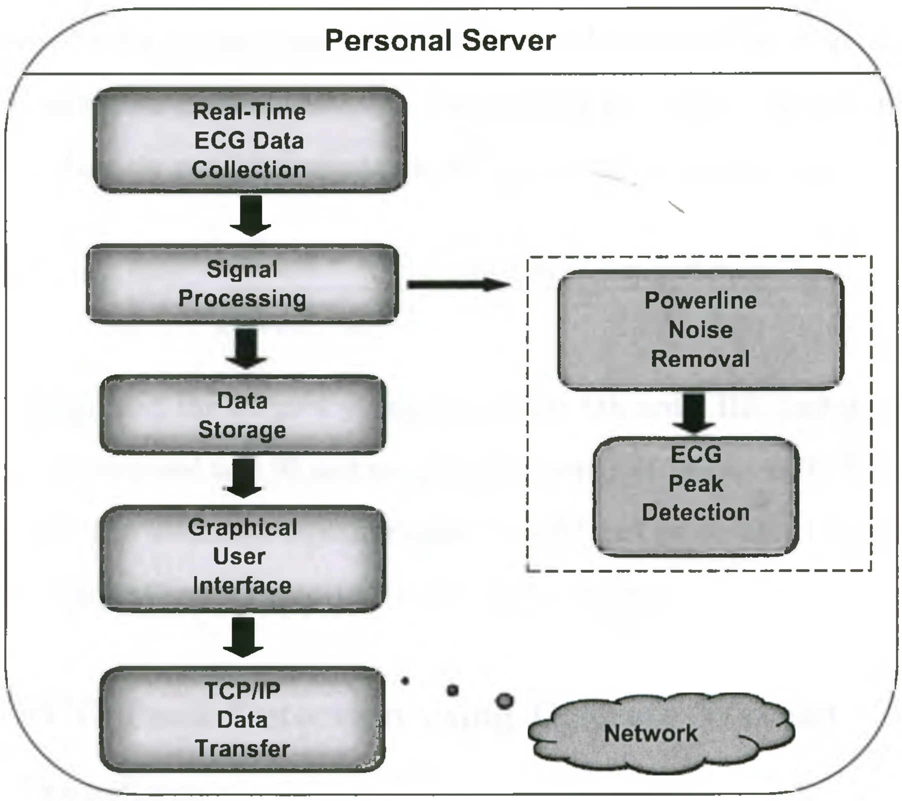


Figure 3.8: Functional architecture of Personal Server services.

3.3.1 Design of IIR digital notch filter

The 60 Hz powerline noise passes through the ECG waveform because of magnetic induction in wires, displacement currents in the electrode leads and body. This interference distorts the data, and obscures the behavior of the body [63]. Thus, the signal should pass through a IIR digital notch filter for removing the powerline interference in an ECG. Here, we investigate the use of pole-zero placements at the desired frequency locations on the unit circle in the design of IIR notch filter. Based on this method, the zeros are located on the unit circle while the poles are located inside the circle at the same angular values and within a radial distance from the zeros in order to flatten the frequency response away from the notch frequencies. In theory, the number of poles (order) that a notch filter has determines the drop-off rate; the higher the order the greater the rate. The transfer function of an IIR notch filter designed by the pole zero placements on the unit circle is given by [64]:

$$H(Z) = \frac{\prod_{i=0}^{N}(z - z_{(i)})}{\prod_{i=0}^{N}(p - p_{(i)})} = \frac{a_0 + a_1 z^{-1} + ... + a_n z^{-n}}{1 + b_1 p^{-1} + ... + b_n p^{-n}}. \tag{3.1}$$

Pole-zero design and the frequency response of the 6th order IIR digital notch filter with a pole radius equal to 0.95 and sampling frequency of 740 for ECG data is shown in Figure 3.9. The de-noised signal (Figure 3.10(b)) can be obtained by passing the noisy ECG signal (Figure 3.10(a)) through the notch filter.

3.3.2 ECG Peak Detection using Discrete Wavelet Transform

The peak of the QRS complex, or R-wave, is a readily identifiable fiduciary mark on the ECG which can be used to indicate a heartbeat. A wavelet analysis based

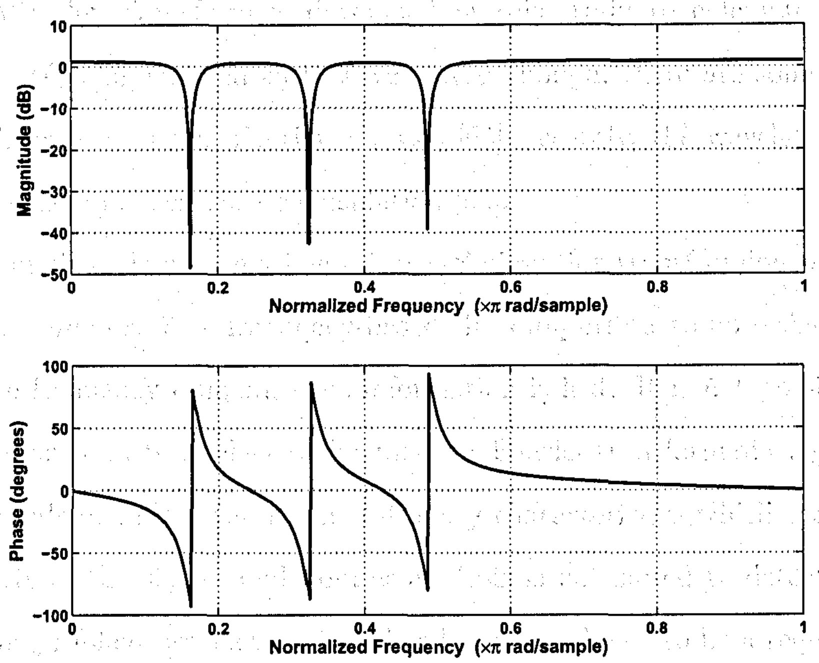


Figure 3.9: Frequency response of the notch filter.

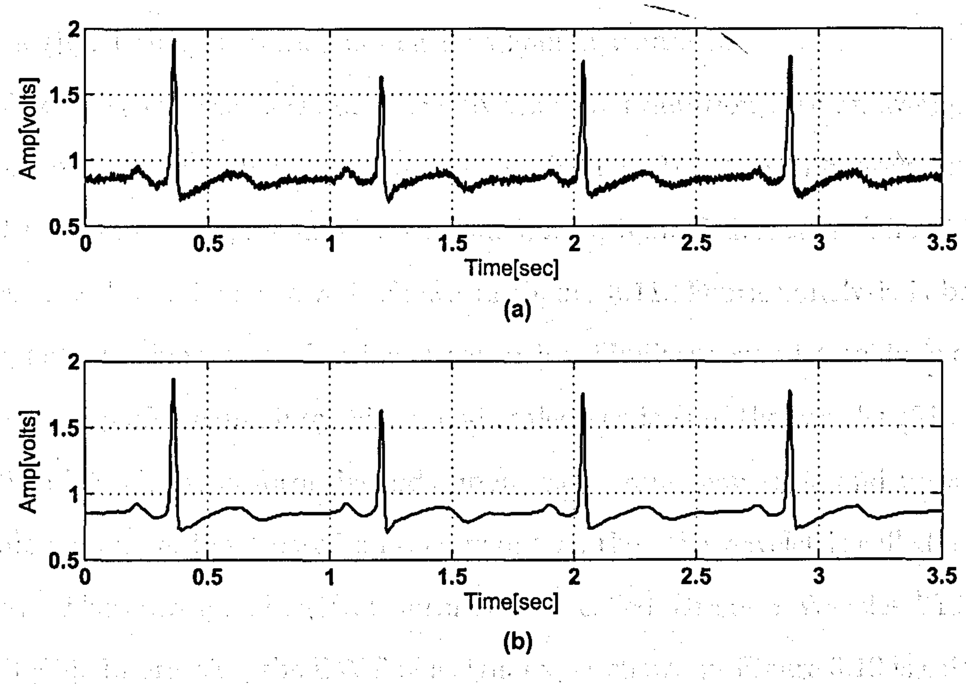


Figure 3.10: (a) Noisy ECG Signal. (b) Filtered ECG signal.

on Discrete Wavelet Transform is developed in this study to calculate particular parameters of ECG signal such as R wave. Even though, there are some works on detection of heartbeats using Fourier analysis [65], recently, the wavelet analysis is found more suitable for heartbeats calculation [66].

Fourier analysis is as a mathematical technique for transforming our view of the signal from time-based to frequency-based. It is important to note that in transforming to the frequency domain, time information is lost. It is not possible to tell when a particular event took place by looking at a Fourier transform of a signal. Most biomedical signals include several non stationary characteristics which are the most important parts of the signal, and Fourier analysis is not suited to detecting them. To address this problem, wavelet analysis has been introduced to be a requisite addition to the analyst's collection of tools and continue to enjoy a burgeoning popularity today. Wavelet analysis is capable of revealing aspects of data which might be missed by other signal analysis techniques such as Fourier transform.

A wavelet is a waveform of efficiently restricted duration with an average value of zero. To compare with wavelet, sine waves which are the basis of Fourier transform extend from minus to plus infinity without having limited duration. The difference between wavelet and sine wave is shown in Figure 3.11. Fourier analysis is breaking up a signal into sine waves of various frequencies. Similarly, wavelet analysis consists of breaking up of a signal into shifted and scaled versions of the wavelet [67].

The Wavelet transform depends upon two parameters; scale and position. If the scale parameter is set based on powers of two, then the wavelet is called a dyadic wavelet. The corresponding transform is also called Discrete Wavelet Transform (DWT) [68]. In practice, the DWT of a signal x, as shown in Figure 3.12 is calculated by passing it through a series of complementary filters; low pass and high pass filter. For many signals, the low-frequency component of the signal called approximations

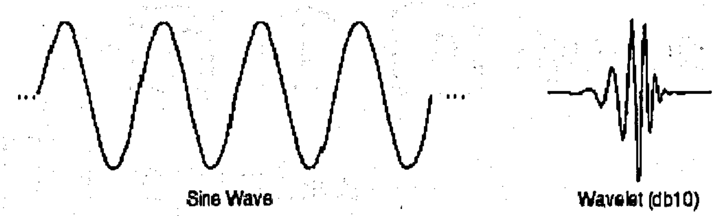


Figure 3.11: A wavelet is a waveform of effectively limited duration that has an average value of zero.

is high scale and the most important part. The high-frequency content, called detail is low scale and passes on flavor or nuance. As the output signals of both filters wind up with twice as much data as the signal passed through filters, the filter outputs are then down sampled by 2. The downsampling process indicated in the Figure 3.12 produces DWT coefficients. A plot of wavelet coefficients clearly illustrates the exact location in time of the particular events. This decomposition has divided the time resolution in two as only half of each filter output characterizes the signal. However, each output has half the frequency band of the input so the frequency resolution has been doubled. The decomposition process can be repeated further in order to break down the signal into many lower resolution components. This is called the wavelet decomposition tree (The ECG signal's wavelet decomposition tree is displayed in Figure 3.13).

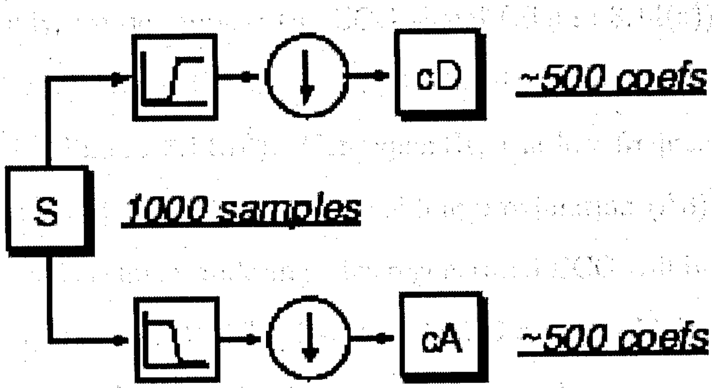


Figure 3.12: Block diagram of filter analysis for DWT.

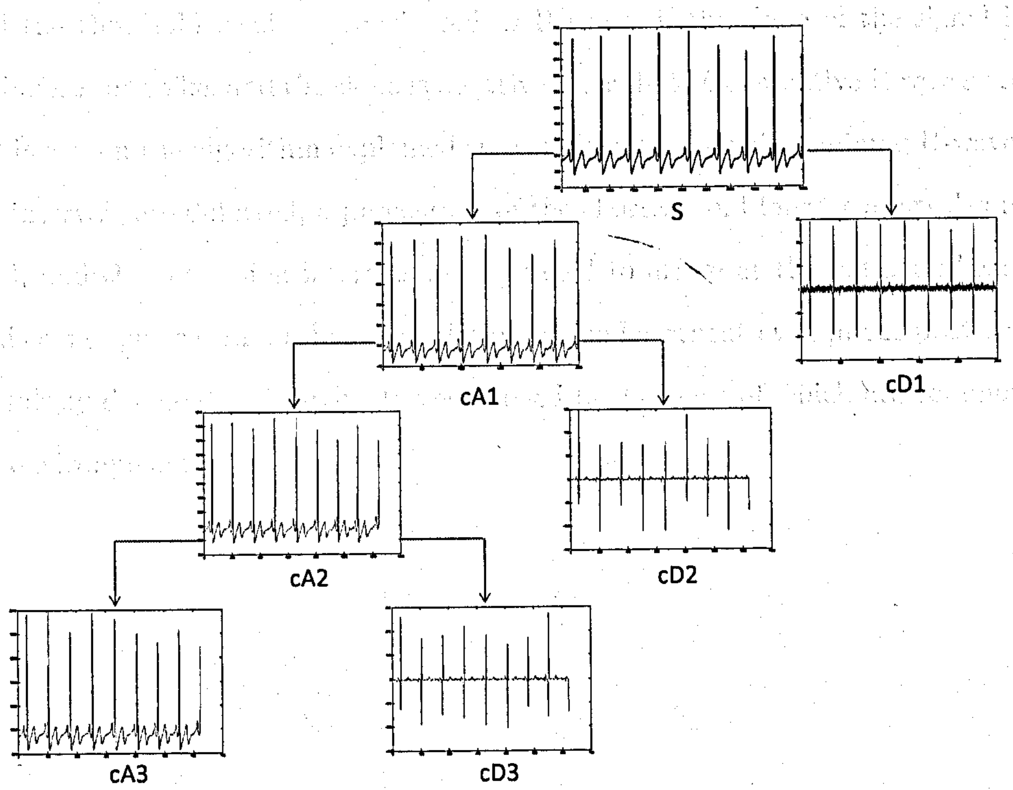


Figure 3.13: The ECG signal's wavelet decomposition tree.

In this study, we decompose the ECG signal (Figure 3.14(a)) into 3 levels by applying DWT and rebuild the Level 3 approximation (A3) and detail (D3) from DWT coefficients (Figure 3.14(b)). Consequently, the low frequency ECG (Figure 3.14(c)) will be reconstructed from the Level 3 approximation (A3) and detail (D3). In order to remove baseline wandering, this regenerated ECG will be subtracted from original ECG (Figure 3.14(d)). The de-trended ECG contains high amplitudes spikes which are R-waves. A threshold level is set up based on the average amplitudes of the R-waves. As all the noisy spikes from supply lines were removed by the notch filter described in previous section, all the spikes in the signal are denoting R peaks. Peaks exceeded the threshold level are considered as R-waves if the slope of the signal is positive before the spike, and the slope is negative after that. Consecutive R waves are detected based on the algorithm explained steps by steps. After determining R-wave, the R-R intervals are detected, a percentage of the shortest and longest intervals are discarded, and the remaining intervals are averaged to arrive at the patient's heart rate. This consequences in a robust calculation of the heartbeat even in the presence of noise falsely detected as heartbeats and missed beats, both of which are common in noisy environments.

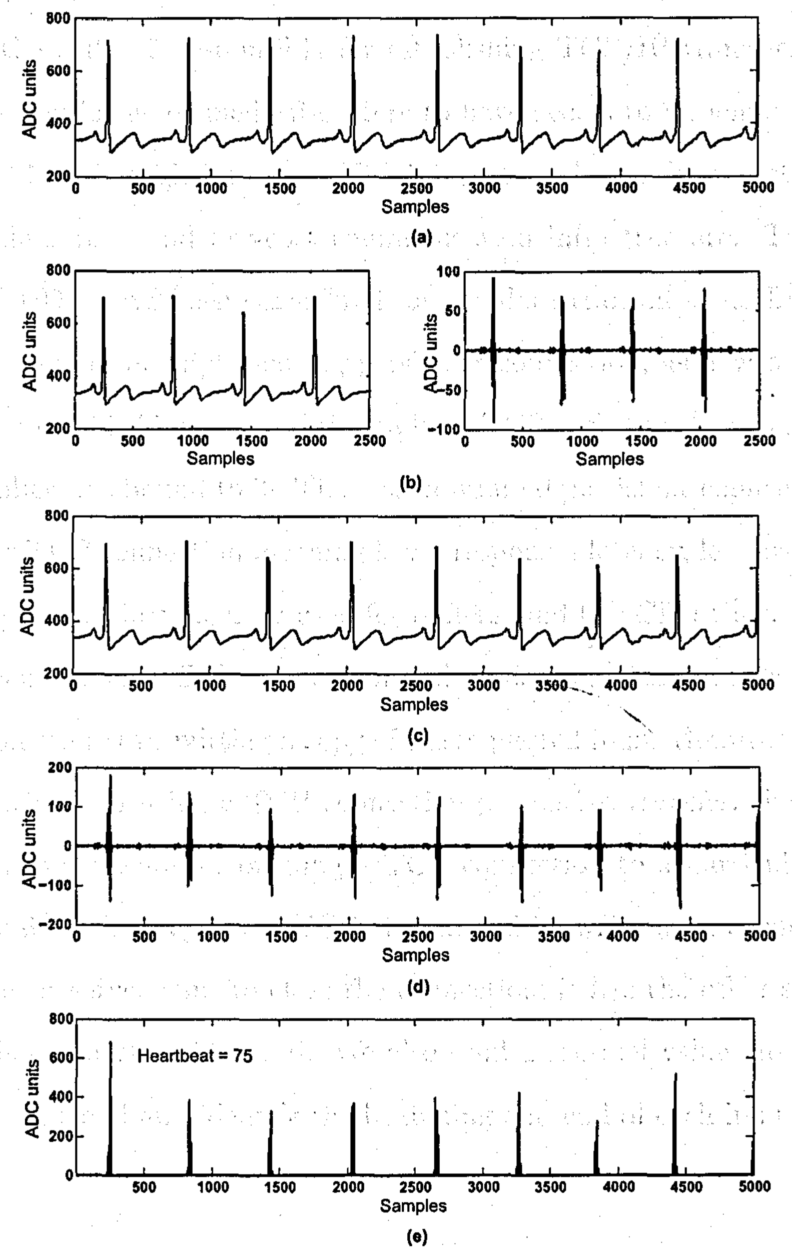


Figure 3.14: (a) Original Signal. (b) The Level 3 approximation (A3) and detail (D3). (c) Low level ECG. (d) De-trended ECG. (e) R-detected ECG.

3.3.3 TCP/IP Data Transfer

Once three cycles of ECG are stored in the FIFO buffer, the previous callbacks pass the data to the callback responsible for establishing TCP/IP transfer. This application allows physicians or medical centers to have access to patient's vital sign via cell phone or PC through internet. The internet is chosen for remote monitoring as it is a wide spread and low-cost communication infrastructure. Two protocols, TCP/IP and UDP have been considered for the data transmission. Between them, TCP/IP was chosen for implementing persistent connection, as it was more reliable than UDP. For establishing a connection, a local TCP socket was created and a specified port number was bound to it. The reason we used persistent connection was that it lies in fewer TCP connection meaning lower responses latency, less overhead on the underlying networks, less memory used for buffers, and less CPU time. In usual connection, a client opens a TCP connection and sends a request, and the server closes the connection after transmitting a copy of the requested item. However, in persistent connection, instead of using a TCP connection per packet transfer, the client leaves the connection in place after opening a TCP connection to a particular server. In each transmission, three cycles of ECG stream will be written on connection buffer. When a client or server aims to close the connection, it lets the other side know the intent and the connection is closed. We also send a sentinel value after the item in order to mark the end and identify the beginning and end of each item sent over the connection.

3.3.4 Graphical User Interface

A graphical user interface (GUI) is embedded into PS which enables users to perform multiple interactive tasks. The GUI, shown in Figure 3.15, allows users to switch between three different options: Monitor, Record, and Send. The upper-left panel contains the three tasks which should be enabled by the users. In this Scenario, the users can visualize the ECG data in real-time by choosing the Monitor option. The Record option stores all the ECG data received from the base station and lets the users to save all the important information including name, age, and weight into the database specified for them. The data transmission to clients over the internet occurs when the users select Send option. In addition to transferring data, the socket addresses of both server and client will be shown in Socket Address panel. The Start and Stop push buttons are responsible for opening and closing the serial port connection. The ECG signal is plotted on an axis located on the upper right of GUI. In addition to plotting the real-time ECG signal, the RSSI value of each received packets will be plotted on the axis in the lower right. This option is extremely useful as it makes the users aware of their positions and distances from the base station.

3.4 Implementation and Verification

To evaluate the performance of our developed system in a real practice, the system has been tested on a real subject, and all important steps comprising real-time ECG acquisition, peak detection, data storage and transmission via TCP/IP through internet have been assessed. The ECG wireless sensor node (Vernier EKG sensor integrated to IRIS mote) was placed on the subject's chest and three electrode tabs were attached to his arms. The ECG data were successfully collected, sampled at 740 Hz frequency by IRIS mote and transmitted over sensor network to the base station. The first test was to check the performance of notch filter on the powerline noise removal. The 60 Hz hum noise on the subject's ECG data has been removed via two different methods: (i) wiring the ECG sensor to the earth ground which confined

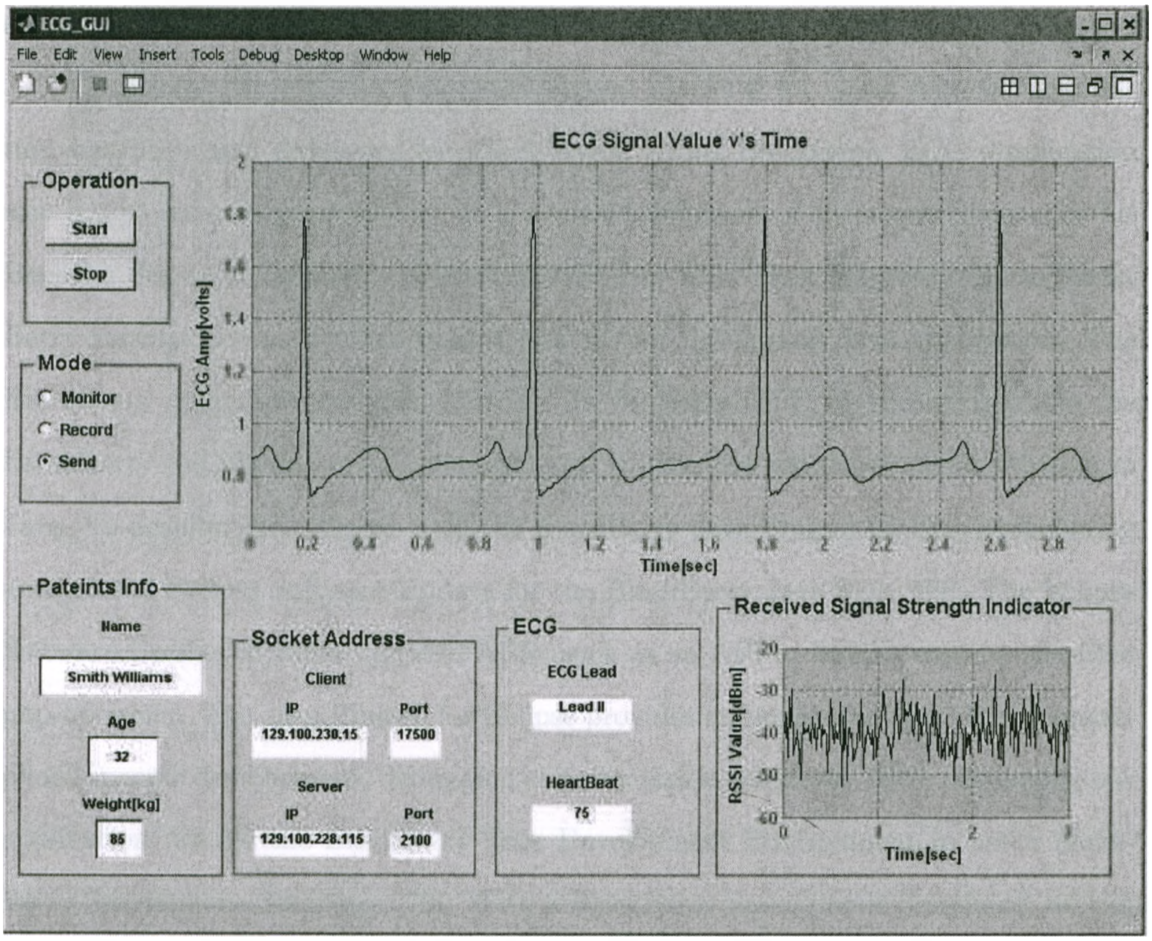


Figure 3.15: The graphical user interface developed in MATLAB for providing real-time monitoring, recording and transmission of the ECG signs.

the subject's movement, (ii) using designed IIR digital notch filter which gives the subject free movement. The results obtained by two methods had excellent consistency which verifies the performance of notch filter. In the next step, subject's heart beat has been calculated by ECG peak detection using DWT and compared to the simultaneous results obtained by high accurate digital wrist heart beat meter; the results were well matched. Finally data transfer was also evaluated by communication between two computers and one Computer and a Black Berry cell phone (server and client), via TCP/IP; the monitoring of received ECG data on the client system was successfully and precisely performed with 5 seconds delay.

In a joint project, the smartphone-based platform for client remote visualization has been also developed by Eliud Kyale. This BlackBerry ECG Application was completely designed by him as a partial fulfillment of the requirements for his Bachelor degree at University of Western Ontario. This client program runs on BlackBerry smartphone as it offers unique features for developing Java application. The BlackBerry communicates with Personal Server using TCP/IP connection. On the BlackBerry cell phone, we had the option of writing the cell phone client software in Java. To develop the code, we used the BlackBerry Java Plug-in for Eclipse including an updated Eclipse Software Update for the BlackBerry Java SDK [69]. The Eclipse Software Update includes updated APIs, such as an API to build a context-sensitive pop-up menu. The Java Plug-in for Eclipse provides an integrated BlackBerry smartphone specific development, debugging and simulation workflow. After developing our application, we set the BlackBerry Java Development Environment to use a BlackBerry Smartphone Simulator [70]. With a BlackBerry Smartphone Simulator, we can run and debug applications as if they were on an actual BlackBerry smartphone. In this application, we used Simulator Model Bold 9700. We designed a communication protocol called EKGCommProtocol which establishes the communication between the BlackBerry and the server. This protocol includes commands such as Handshake Command, Send Data Command and Closed Connection Command. As three cycles of ECG data (almost 5 KB) arrives at the phone in real time, the TCP/IP thread sends the data to the plotting thread for displaying the ECG graph on the screen. To improve the aesthetics of the ECG signals display on the application, we have included an animated background to mimic the heartbeat rhythm. A screenshot of the Black Berry Java development environment and simulator for the Bold 9700 is shown in Figure 3.16.

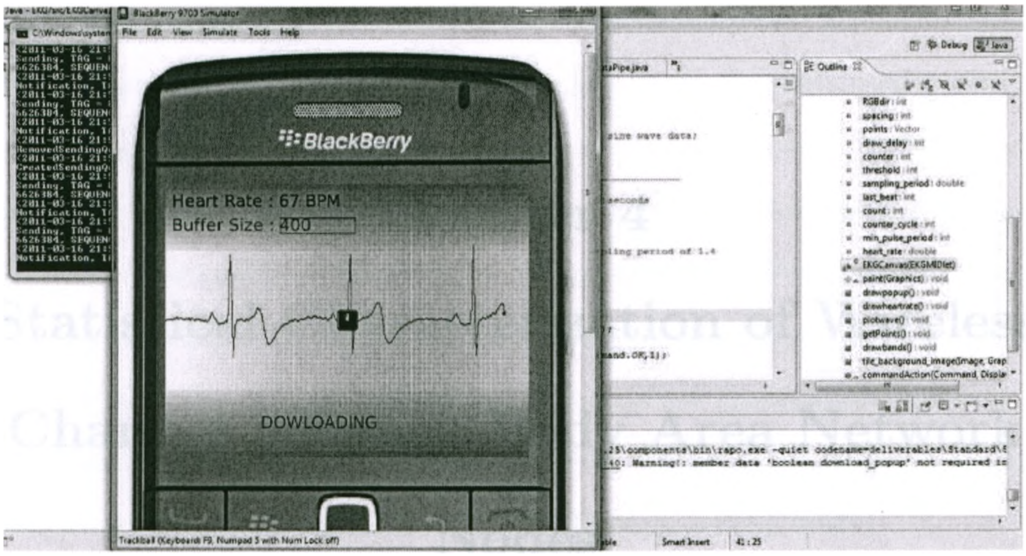


Figure 3.16: A screenshot of the Black Berry Java development environment and Smartphone Simulator for the Bold 9700.

3.5 Conclusion

This Chapter presented a system prototype of wireless sensors for medical care within the framework of Body Area Network standardization. Our approach to develop a combined hardware and software platform for the WBAN nodes were described and discussed. In addition to developing WBAN nodes, the personal server interfacing between the WBAN sensor nodes and Wide Area Network was implemented on PC. The PS was responsible for providing multiple functionalities such as real-time visualization, memorizing, analyzing and communication with clients requesting connection over internet. The developed system takes advantage of interaction with sensor network nodes using MATLAB. Several test has been performed to assure the feasibility and accuracy of the system. The system features a clean design based on the unique characteristics of patient monitoring in general hospital units which include of (i) maintaining periodic and real-time transmission of ECG signs , (ii) providing a user feel of comfort, and (iii) giving high speed access to wireless networks.

Chapter 4

Statistical Characterization of Wireless Channel between Body Area Network Nodes

4.1 Introduction

In this chapter we characterize the wireless channel structure for the communication link between the sensor node placed on human body surface and the external receiver node placed near the computer server. In order for a Body Area Network (BAN) to operate in the desired manner, it is essential to guarantee that the communication between the mounted sensor nodes and base node takes place in the right way. In BAN, the dynamic nature of the human body, complexity of the human tissues structure and body shape give variation to the traditional radio channel [71]. Hence, it becomes difficult to extract a simple channel model.

In wireless communication, two types of mathematical channel modeling are employed to describe the radio channel and evaluate the performance of different physical layer proposals; (i) a stochastic modeling and (ii) theoretical modeling.

The statistical modeling has proven to be a powerful approach to consideration of most problems related to information transmission. It is the fact that any real signal, propagation channel, and interference all have stochastic nature [72]. In this approach, the essential properties of the signal propagation, such as narrowband fading, are captured by probability distributions. It is involved in collecting a set of measurements for a very large population of transmit-receive (T-R) paths in a specific environment [73]. The alternative approach -theoretical modeling- contains fundamental principles of electromagnetic propagation and precise modeling of a specific situation at radio link level. This approach benefits from computer programs that (1) emulate the physical environment, (2) use wave propagation physics to predict the radio signal produced at any receive point from any transmit point, and (3) account for transmission through walls and diffraction around walls.

Thus far, a limited number of studies have been carried out to characterize the statistical on-body wireless channel. These studies exist on modeling the pathloss and amplitude distribution of both narrowband and UWB wireless BAN's at different frequencies [24, 28]. Fading characteristics within the European 868 MHz band have previously been investigated for body worn communication in WBAN applications [74]. The authors used Mica2Dot wireless sensor motes (based on CC1000) in their study. Alomainy *et al.* [27] studied on body pathloss distribution for a stationary user in an anechoic chamber and laboratory environment using microstrip patch antennas at 2.45 GHz. The authors suggested that the pathloss distribution was well described by the Lognormal distribution and that the human body is the leading shadowing factor in WBAN. Fort *et al.* [24] have carried out one of the most comprehensive studies on describing both the UWB and narrowband radio channel around the human body in a typical indoor environment. They have published several papers in this regard [20]-[24]. They also developed practical models which have been used to evaluate their performance in WBANs. They have been able to report some first order statistics for small scale fading in an indoor environment while the user was stationary [22] and in the case of partial movement of arms [23]. Furthermore, to understand the narrowband radio propagation near the body, they have measured electromagnetic waves of the 915 MHz and 2.45 GHz around torso [24]. Their measurement showed that the pathloss of energy diffracting around the body follows the expected exponential trend, but flattens out due to the influence of multipath components reflecting from the surrounding environment. The small-scale fading has been represented by a Ricean distribution with a K-factor decreasing as the amplitude of the initial diffracting wave decreases. It approaches the Rayleigh distribution when the transmitter and receiver located on opposite sides of the body; in this situation the multipath reflections from the surrounding environment are dominant. Reusens *et al.* [31] have characterized the physical layer in terms of pathloss, delay spread, and mean excess delay for narrowband communication at 2.45 GHz between two half wavelength dipoles near a realistic human body. In their study, the CDF of the deviation of measured pathloss and models were described by a Lognormal distribution. In addition, the pathloss exponent was about 3.1 when the subject located in free space. Kim *et al.* [26] has introduced the dynamic on-body channel measurements obtained using a real-time channel sounding systems. The measurement were performed using a male subject in a radio anechoic chamber and the investigation of the statistical characteristics in specific action scenarios were presented. In their studies, they found that the Weibull distribution has the best match under dynamic channel behavior.

Beside the amplitude distribution and the pathloss model, the link estimation is also an important factor in protocol design and has been focused by many researchers interested in extracting asymmetry correlation between RSSI, LQI and PRR. One of the first attempts on systematic measurements of packet delivery in wireless sensor networks has been performed by Zhao *et al.* [75]. They positioned Mica motes in a simple linear topology in three separate environments. From their measurements (without any encoding), they indicated that links with PRR of at least 95% has high

RSSI while the converse is not true. It was concluded that such PRR was most likely due to multipath effects as there was little correlation between high RSSI and PRR. Later, Polastre *et al.* [76] presented preliminary evaluation results for Telos motes (based on CC2420) and recommended that the average LQI is a better indicator of PRR than RSSI. Soon after, Srinivasan *et al.* [77] at Stanford University presented preliminary evaluation for Telos motes. Their results contradicted the previous works as they showed that RSSI has a good correlation with PRR rather than LQI. Moreover, they explored a temporal and spatial correlation of packet delivery and link asymmetry in their recent publication [78]. Their study clarified a fundamental challenge in packet-based network studies: one can only measure successfully received packets. Simply disabling CRC checking does not result in meaningful data, as random RF noise can often appear to be a start symbol. Therefore, all measurements are biased. A number of studies have discussed the interference caused by the human body and diverging environments on radio communications. In this regard, Jea *et al.* [34] presented some results on connectivity in a body area network using the mica2dot motes. They have measured link qualities between nodes at different on body positions under various RF power levels for two scenarios of standing and walking. Their results suggested good connectivity among all nodes on a body beyond a certain transmit power. The works carried out by Papagiannaki *et al.* [33] took another steps toward understanding of the on body 2.4 GHz link characteristics. They used Intel Mote 2 devices and placed them on three areas: the chest, the right side of the waist and the right ankle, while setting the transmit power of the radio at 0 dBm. They showed that nodes location, as well as body location, significantly affects connectivity.

In all the above mentioned works on sensor networking, the authors presented preliminary results which are the initial step toward a better exploring the characteristics of wireless communication between on body sensor nodes especially for prior crossbow motes such as micaZ, Mica2Dot, and Telos. In this thesis, we will evaluate the newer set of crossbow motes called IRIS (based on RF230) and we believe that measuring these network characteristics can facilitate exploration of protocols functionalities in that network. In this network, it is necessary for the wireless communication channel between nodes to be studied in the environment that wearable systems are implemented. In fact, the nature of local environment is responsible for the performance of radio communication systems. Environment plays an important role in variation of communication channel and link estimation metrics due to complex propagation mechanisms such as diffraction, reflection and scattering. Furthermore, the propagation channels vary as the mobile device moves behind the walls, desks, and other objects.

4.2 On-Body measurements

This section describes the measurement set up, environment, and procedure used for extracting channel parameters.

4.2.1 System

The body worn measurement system consists of two Crossbow IRIS wireless sensor motes and one MIB520 base station. Each IRIS board contains ATmega1281 microprocessor [42] as a data processing unit and AT86RF230 [43], a low-power radio transceiver. The radio used by the IRIS is an IEEE 802.15.4 compliant RF transceiver designed for low-power and low-voltage wireless applications. It uses Atmel's AT86RF230 radio that employs O-QPSK ("offset quadrature phase shift keying") with half sine pulse shaping. The 802.15.4 radio includes a DSSS (digital direct sequence spread spectrum) baseband modem providing a spreading gain of 9 dB and an effective data rate of 250 kbps. The radio is a highly integrated solution for wireless communication in the 2.4 GHz unlicensed ISM band [41]. In this experiment, one of the IRIS mote acts as the base station receiving the packets and the other IRIS mote is mounted on the subject's waist as a RF source. For IRIS, RF transmission power is programmable from 3 dBm to -17.2 dBm. Lower transmission power can be useful by minimizing interference, maximizing energy efficiency, and decreasing human exposure to electromagnetic radiation. On the other hand, the higher transmission power reduces the fading. We therefore programmed the transmission power to the medium value of 0.5 dBm for having both sides advantages. Moreover, the radio was tuned within the channel 11 of IEEE 802.15.4 with a bandwidth of 5 MHz. The receiver is kept stationary on the desk (with the height of 90 cm), connected to computer server via USB. The Computer Server is programmed to record the link estimation metrics (RSSI, LQI and the number of received packets), in each cycle of broadcast.

The Sending IRIS broadcasts packets with a rate of 70 packets/second for a period of 10 seconds. The motes were programmed with TinyOS and running the code developed and explained in Chapter 3. A man of age 28, mass of 80 kg and height of 178 cm, served as a test subject. The IRIS mote was fixed on his waist and all the measurements were taken in two distinct indoor environments; office room and corridor.

4.2.2 Environment

The experiments were performed in two different environments: (1) The indoor office labeled with 344 and (2) corridor labeled with TA 37 in Figure 4.1. The floor and ceiling of the indoor office and corridor consist of 3 1/2 inch concrete over a corrugated steel deck. The interior partitions are 1/2 inch drywall over $2'' \times 4''$ steel studs. The office area also contains several PCs, chairs, desks, and shelves.

4.2.3 Procedure

The floor plan of environments was uniformly meshed having total number of 31 and 36 nodes in the office and corridor (shown in Figure 4.2), respectively. For each action scenario, several measurements were conducted on the labeled nodes. However, there were some nodes in corridor marked with green color in Figure 4.2 which were out of network coverage and couldn't communicate with base station. The measurements therefore were made at 24 locations marked with red circles in corridor. The measurements in each environment include three action scenarios of standing, sitting, and walking. The test subject was instructed to stand/sit motionless with arms hanging at his side on each node in both test environment for 10s. The test subject then asked to walk in a natural manner along the marked locations. For robustness, the measurements were repeated for two individual trials of each action scenario. It should be noted that during experiment, the receiver was kept stationary near the computer server inside an office area marked with black rectangle. Roughly 342 snapshots (interval of 10s) at each marked location and each action scenario were taken via IRIS motes.

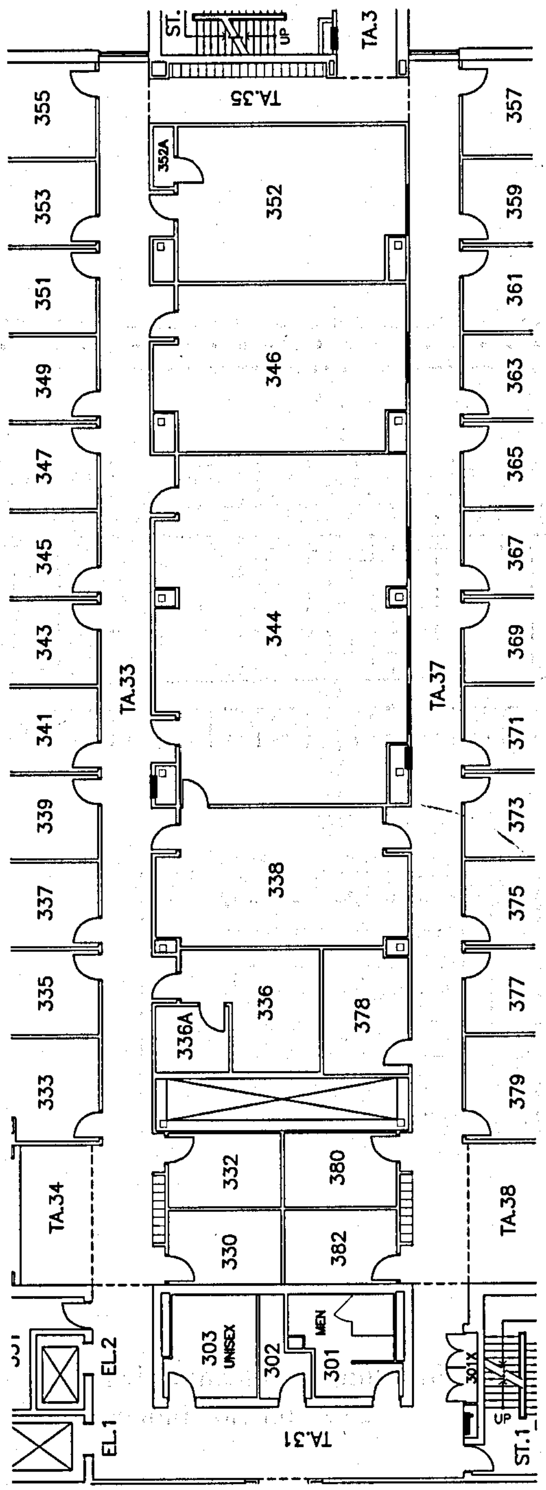


Figure 4.1: The floor plan of the indoor office and Corridor area.

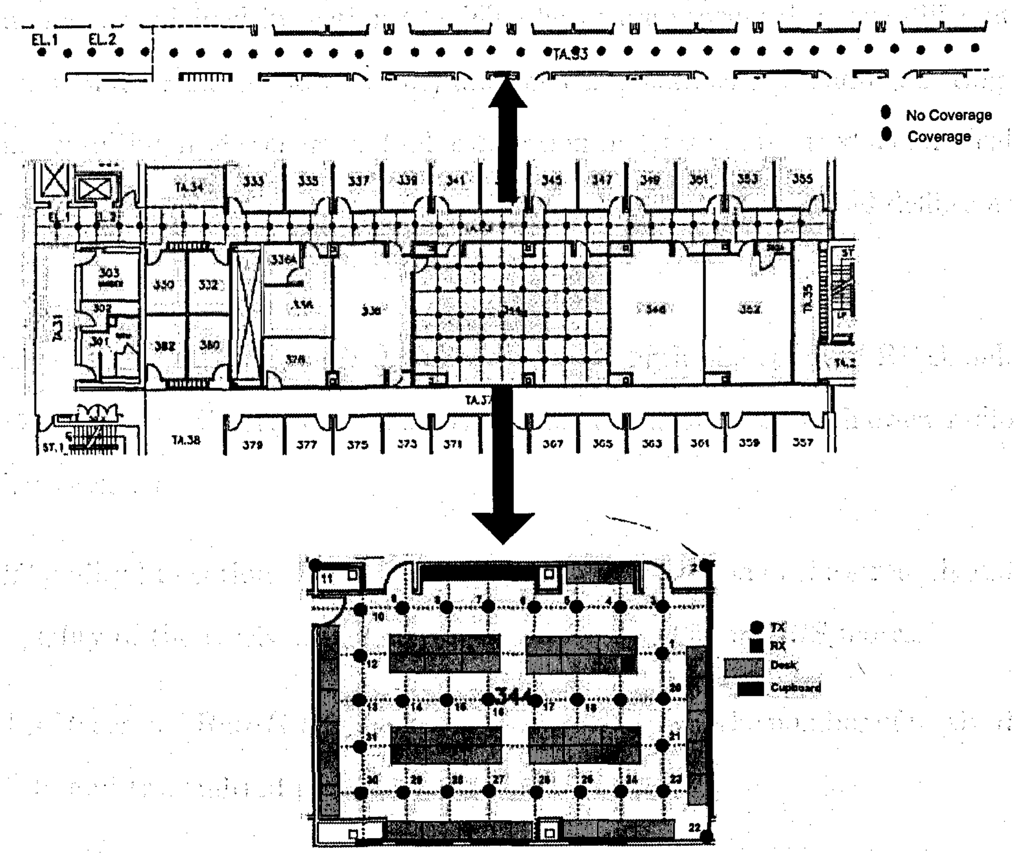


Figure 4.2: The floor plan of environments was uniformly meshed having total number of 31 and 36 nodes in the office and corridor area.

4.3 Spatial correlation of link estimation metrics

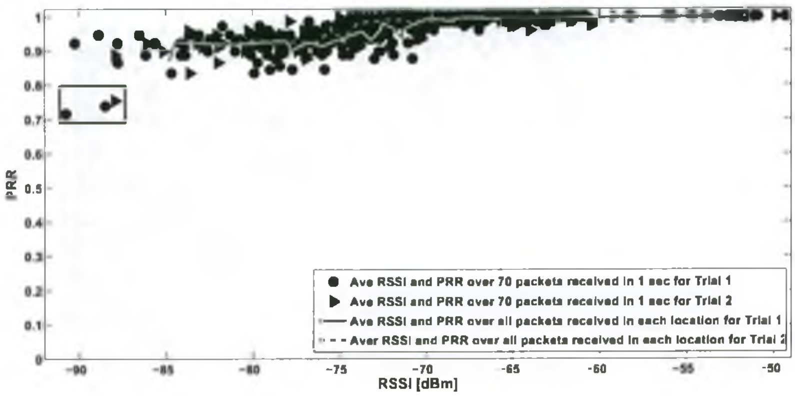
Since link estimation is an important factor in protocol design, in this section, we shall examine the correlation between link estimation metrics, i.e: RSSI, LQI, and PRR, for the real channel condition. The data used in this section are based on the measurement explained in section 4.2. The data were processed in two different environments (Indoor office and corridor) and for three actions of standing, sitting, and walking resulting in 6 scenarios. Link estimation metrics are referred as physical layer parameters that the node can measure. They can be categorized and defines as follow:

- Received signal Strength (RSSI), which is the strength of a received RF signal. The RSSI is read directly from the AT86RF230 Radio and sent with every radio packet received.
- Link Quality Indication (LQI), which is the characterization of the strength and the quality of the received packets varying from 0 to 255 in IRIS motes.
- Packet Reception Rate (PRR), which is the ratio between the number of received packets and transmitted packets.

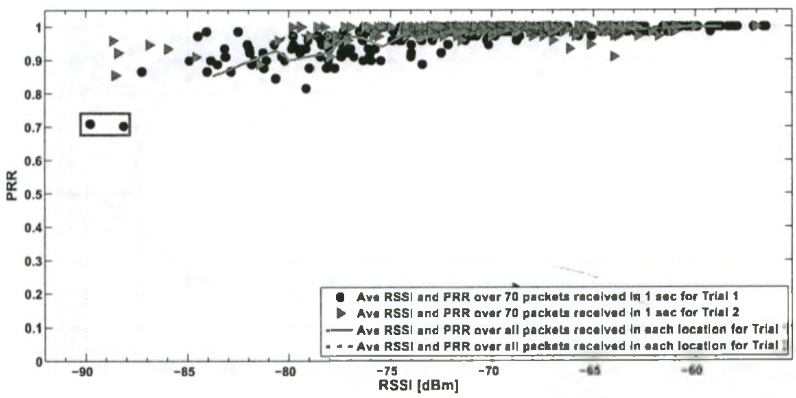
4.3.1 Packet Reception Rate (PRR)

In order to extract the correlation between physical layer parameters, we have plotted the PRR versus variations of RSSI, LQI, and distance. As explained in section 4.2.3, for each environment, the measurements were conducted at different grid nodes. In each node, the link estimation metrics were recorded during 10 seconds and repeated for two trials. Figure 4.3 and 4.4 depict the plots of PRR versus variations of RSSI for office area and corridor, respectively. For the PRR versus RSSI plots, we first plotted average RSSI and PRR (computed over 70 packets received) at each grid node shown with circles and triangles for first and second trials respectively. On the same axis, we then plotted the variation of PRR versus the mean RSSI which are computed over all packets received in each gride node shown by dashed and solid lines (only for standing and sitting scenarios). In each location grid, receiving 70 packets takes approximately 1 second, so circles and triangles display the channel variation over time while the dashed and solid lines indicate the spatial behavior of the channel.

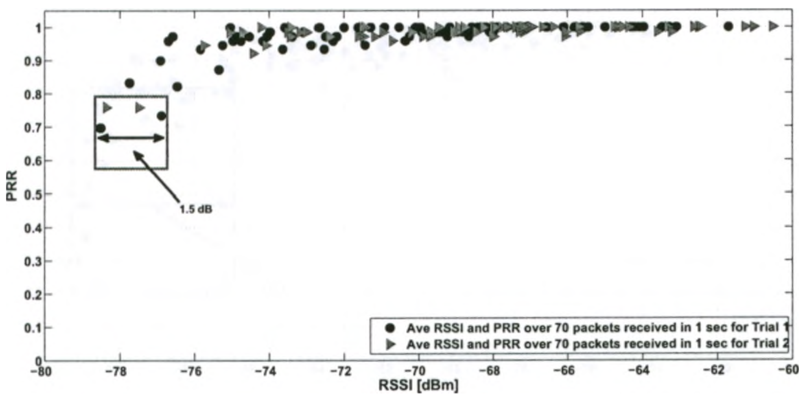
In general, the diagrams show a strong correlation between RSSI and PRR values which proves the stability of RSSI over different locations and times. In detail, it can be concluded from Figures 4.3(a) and 4.3(b) that for all RSSI values measured in standing and sitting scenarios inside office area, the PRR is at least 80% indicating a very reliable link. However, small variations are observed in RSSI values which are made by attenuation and noise floor. We will investigate modeling their variations in next section. Moreover, the few outliers surrounded by red triangles in both plots (Figures 4.3(a) and 4.3(b)) have PRR values less than 80%. However, for the walking scenario shown in Figure 4.3(c), for the RSSI values less than -75 dBm, the PRR values are less than 80% which point out the weakness of link for the subject walks along marked locations inside office.



(a)

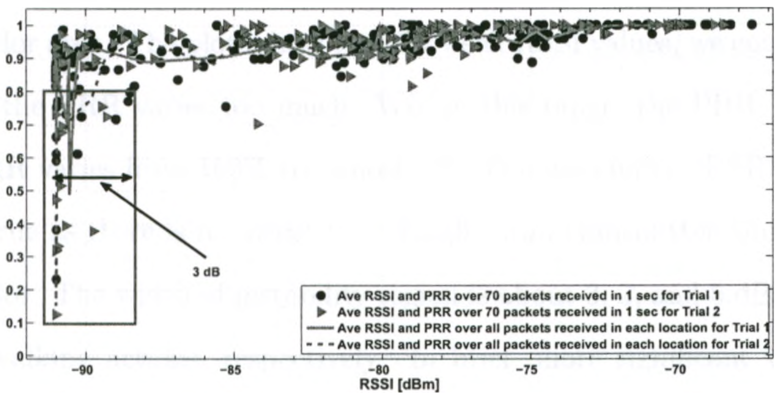


(b)

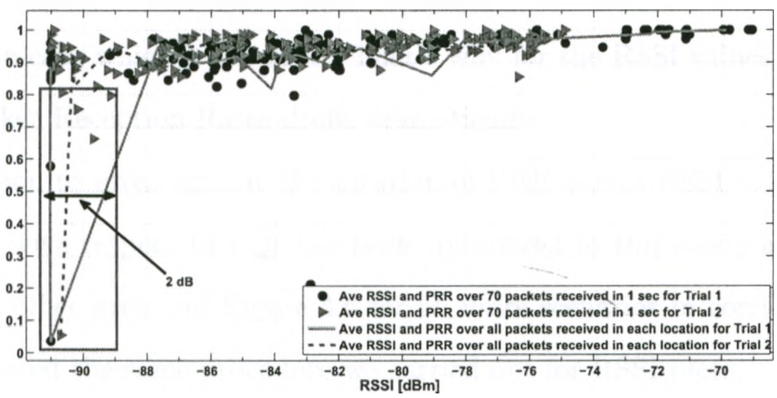


(c)

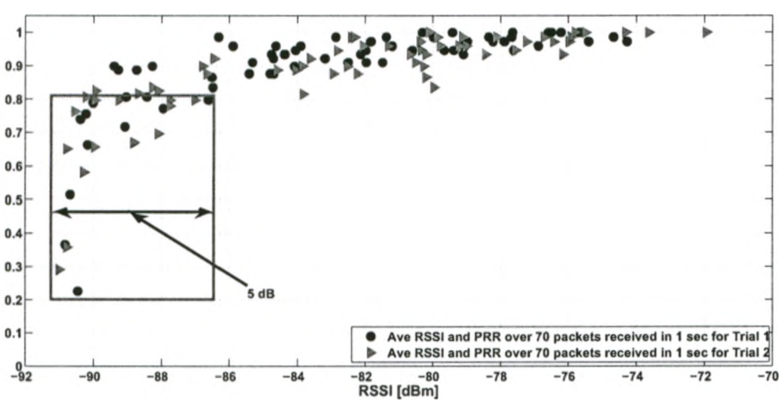
Figure 4.3: Variation of PRR with respect to RSSI for three actions of (a) Standing, (b) Sitting, and (c) Walking in indoor office area.



(a)



(b)

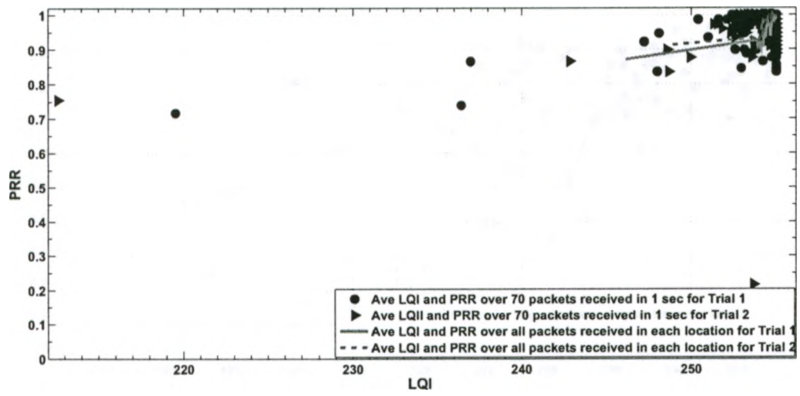


(c)

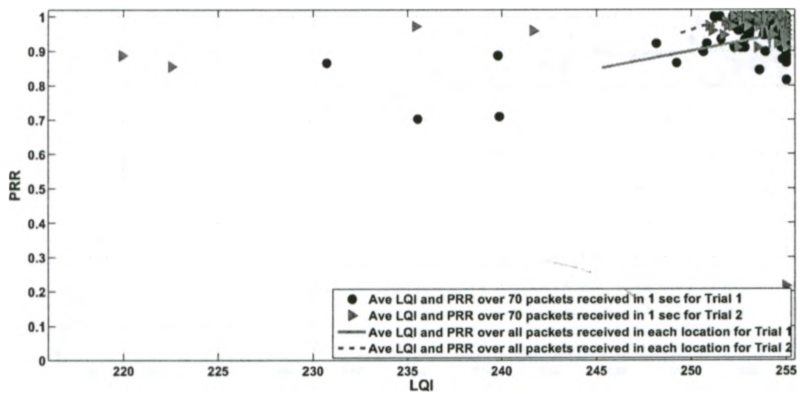
Figure 4.4: Variation of PRR with respect to RSSI for three actions of (a) Standing, (b) Sitting, and (c) Walking in corridor area.

Figure 4.4 displays the variation of PRR versus RSSI values for measurements taken in corridor area. The plots show that for weak RSSI values, we enter a rectangle region where the PRR varies too much. Within this range, the PRR-RSSI curve is sharp and PRR varies from 100% to almost 0%. The instability of PRR vs. RSSI is particularly true as there is no direct line of sight from transmitter node to the base node in corridor. The width of instability region is about 3, 2, and 5 dB for standing, sitting, and walking actions, respectively. In brief, more significant variations are found when the transmitter is located and the link is weak for RSSI values below -86 dBm. It is important to note that the -86 dBm is close to the sensitivity threshold of RF230 radio chip which is -91 dBm. That's why for the RSSI values less than -86 dBm, the Packet Reception Rates drops dramatically.

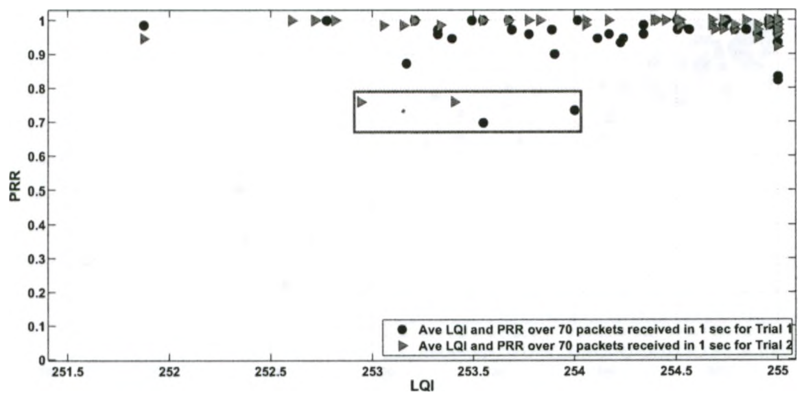
In addition to investigation of varication of PRR versus RSSI values, the variation of PRR with respect to LQI has been considered in this study as depicted in Figure 4.5 for office area and Figure 4.6 for all users scenarios in corridor. For LQI plots, we repeated the same procedure we carried out for RSSI plots.



(a)

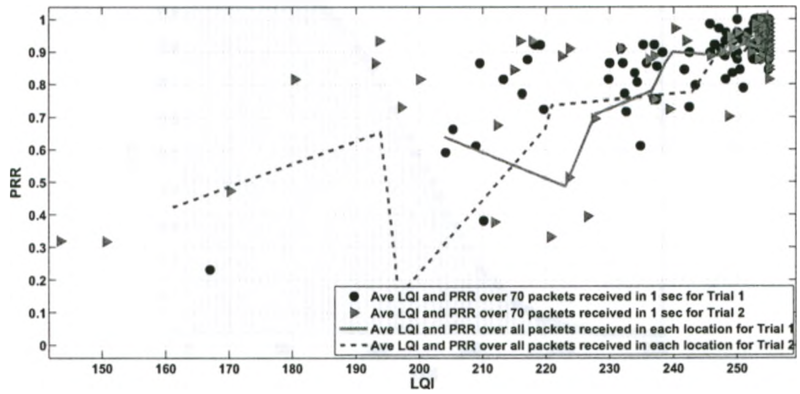


(b)

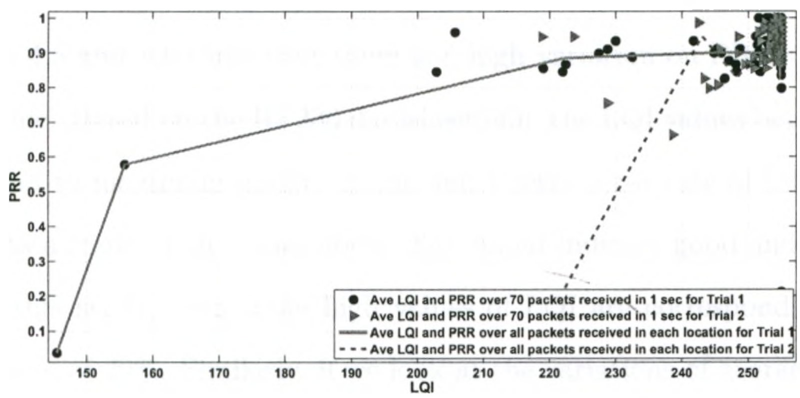


(c)

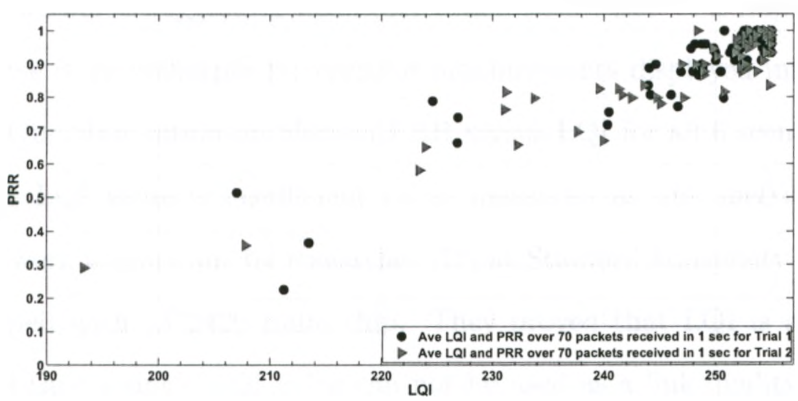
Figure 4.5: PRR versus LQI for three actions of (a) Standing, (b) Sitting, and (c) Walking in indoor office area.



(a)



(b)



(c)

Figure 4.6: PRR versus LQI for three actions of(a) Standing, (b) Sitting, and (c) Walking and in corridor area.

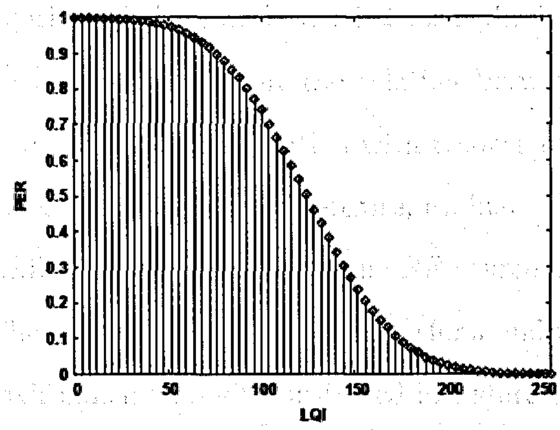
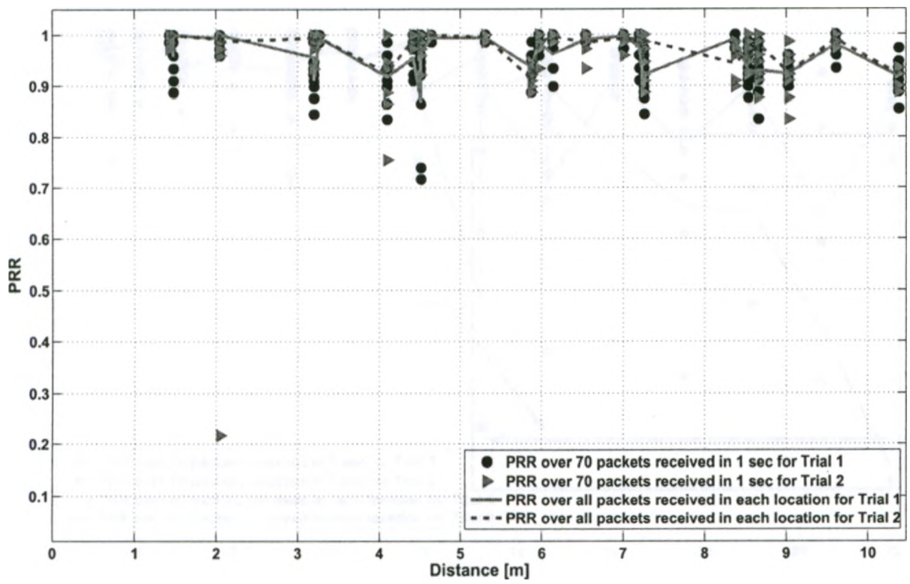


Figure 4.7: Conditional Packet Error Rate versus LQI

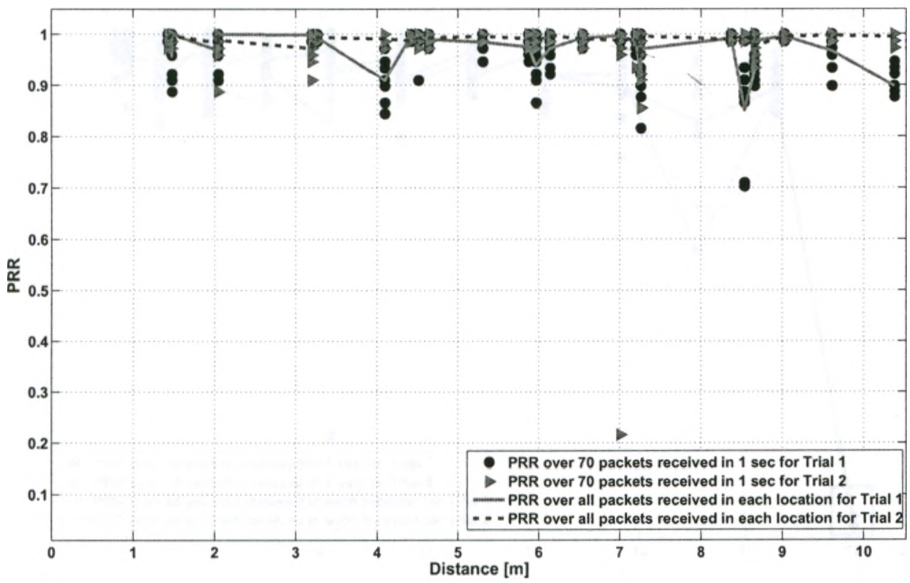
Figures 4.5 and 4.6 show that there is a high variation on PRR values for any given LQI value. Based on the RF230 datasheet [43], the LQI values between 200 and 255 correspond to maximum quality frame and Packet error rate of 0% as shown in Figure 4.7. As a result, LQI values above 250 should indicate good and reliable link. However, in this study, even some high values of LQI are corresponding to Packet Reception Rates of 70%. Similarly, if we look at the variations of average LQI values computed for all received packets in each location, we can figure out that they don't follow a smooth curve suggesting weak correlation with PRR. It is important to note that this curve is even sharper for corridor measurements displayed in Figure 4.6(a) and 4.6(b). Our observation on plots of PRR versus LQI for all 6 scenarios suggests that a single LQI value is insufficient to be measured as link metrics. This fact support the work carried out by researcher [78] at Stanford University on the micaZ motes equipped with CC2420 radio chip. They proved that LQI is a probabilistic quantity and thus a single LQI value can not be used as a link quality indicator for intermediate links.

Besides investigation of the correlation between physical layer parameters of the network, we also tried to figure out the relation between PRR and distance. Our focus lies on verifying the PRR variation with respect to the distance between transmitter and receiver nodes. We can, therefore, evaluate the performance of link estimation metrics while the subject wearing the RF source is mobile and moves to different locations. The plots of PRR versus distance for standing and sitting scenarios performed in both environments are demonstrated in Figure 4.8 and 4.9. Remember that for walking scenario, the subject was asked to walk in a natural manner along all the marked locations inside office and corridor without any stops. Hence, all the link estimation metrics were recorded continually without specifying subject's location. So, we couldn't extract corresponding PRR for each distance between nodes and our observations are limited to standing and sitting scenarios.

The plots of PRR versus distance (Figures 4.8 and 4.9) show that PRR values and distances between nodes are not correlated. For measurements taken in office area (Figure 4.8(a) and 4.8(b)), at every distances between nodes, the PRR values are above 80% and no specific relation between PRR and distance is found. The same behavior can be observed from the measurements made in corridor for sitting scenario (Figure 4.9(a)). However, for user standing state in corridor, there is a distance range in which PRR values drop below 80%. The width of this range is almost 9 meters as shown in Figure 4.9(b).

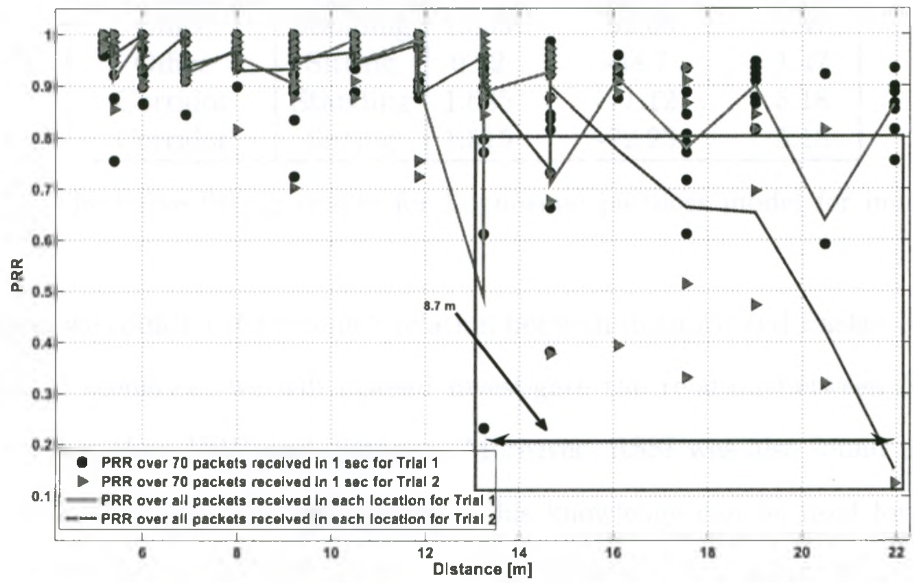


(a)

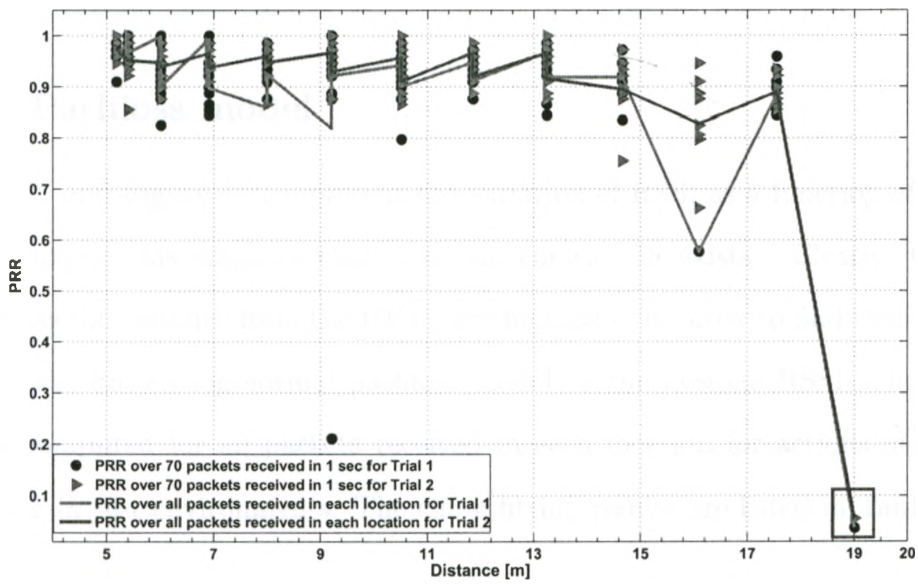


(b)

Figure 4.8: PRR versus distance for two actions of (a) Standing and (b) Sitting in indoor office area.



(a)



(b)

Figure 4.9: PRR versus distance for two actions of (a) Standing and (b) Sitting in corridor area.

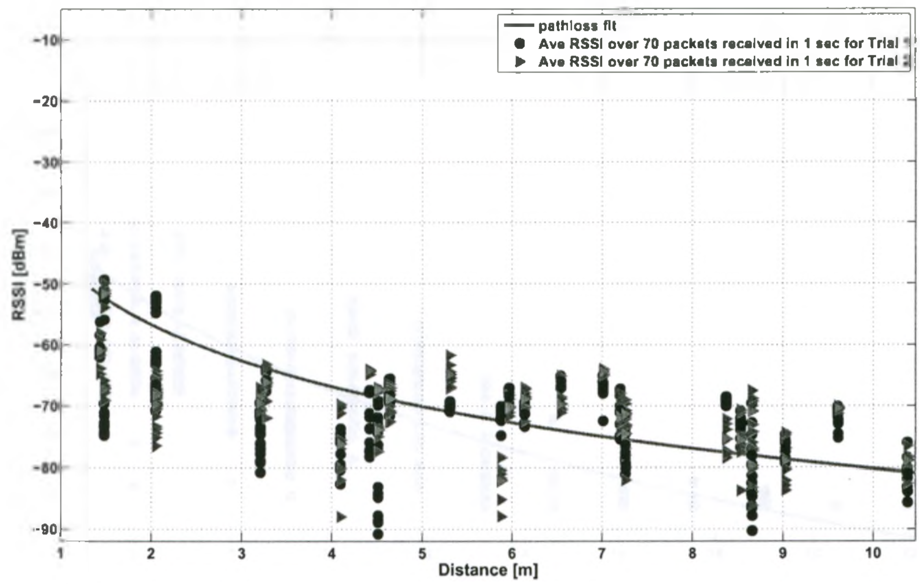
Environment	State	n	$RSSI_0$ [dBm]	d_0 [m]
Office	Standing	1.503	-52.06	1.47
Office	Sitting	0.92	-58.7	1.47
Corridor	Standing	1.685	-71.12	5.18
Corridor	Sitting	1.515	-72.23	5.18

Table 4.1: The curve fitting results for Lognormal pathloss model for indoor office environment.

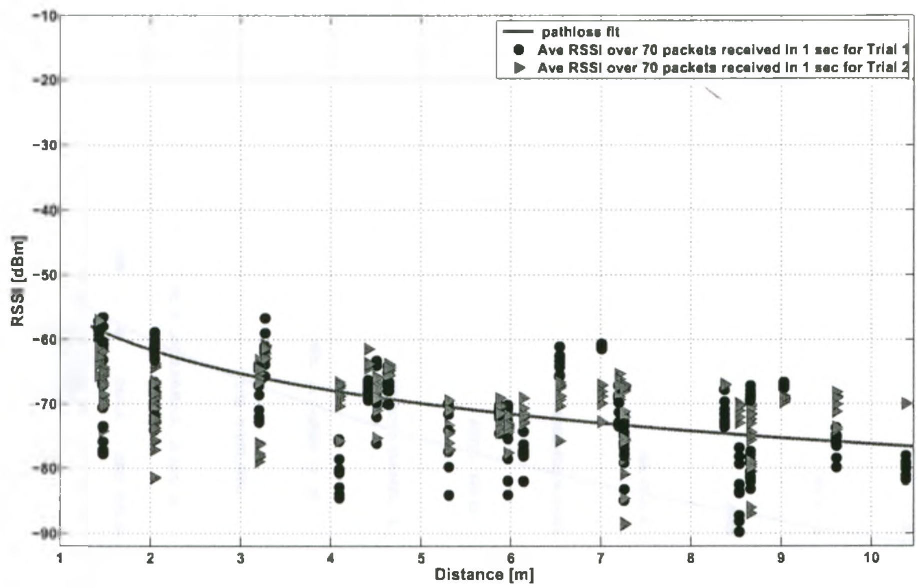
Since, we couldn't determine a relation between distance and Packet Reception Rates for all scenarios, we will instead investigate the relation between RSSI and distance rather than PRR and distance. Moreover, RSSI was also found as a good candidate for indicating the link quality. This knowledge can be used for tracking the user while wearing the RF source. This relation can also adapt the route if the links reliability suddenly drops. Note that RSSI values are usually used for distance estimation.

4.3.2 Pathloss model

Figure 4.10 and Figure 4.11 represent the variation of RSSI as a function of distance between nodes; this suggests that a strong correlation exists. Clearly, the RSSI decreases as the distance from the RF source increases. In order to find their relation, we fitted the known log-normal pathloss model to the average RSSI values versus distance computed for all packets received in each location for actions of standing and sitting in test environments. The curve fitting results are listed in Table 4.1.

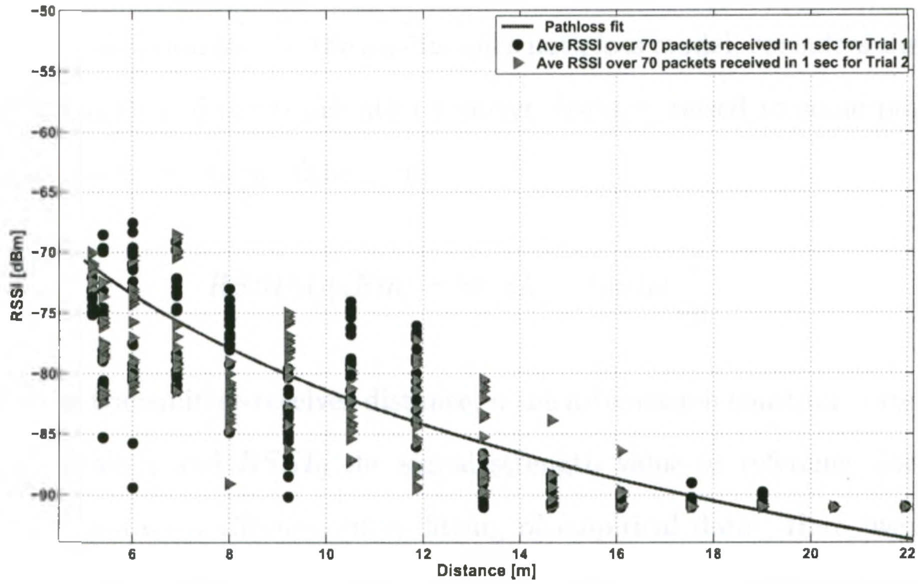


(a)

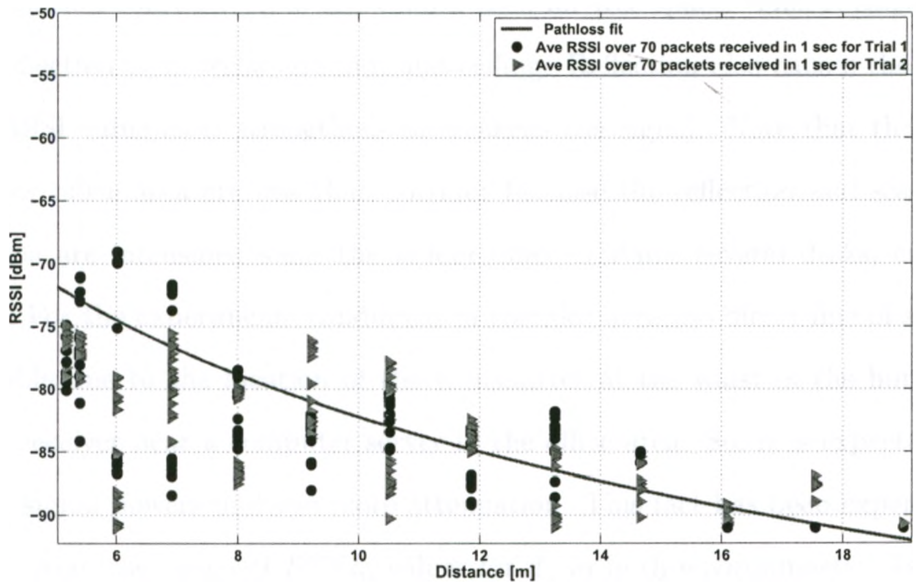


(b)

Figure 4.10: RSSI versus distance for two actions of (a) Standing and (b) Sitting in indoor office area.



(a)



(b)

Figure 4.11: RSSI versus distance for two actions of (a) Standing and (b) Sitting in corridor area.

Before comparing results, we briefly review the log-normal pathloss. The widely used radio propagation model, the log-distance path loss model, considers the received power as a function of the transmitter-receiver distance raised to some power. The log-distance path loss model defined as:

$$RSSI(d)[dBm] = RSSI_0 - 10n\lg(\frac{d}{d_0}), \tag{4.1}$$

where d is the transmitter-receiver distance, n the attenuation constant (rate at which the signal decays), and $RSSI_0$ the signal strength value at reference distance d_0. Usually, n is obtained through curve fitting of empirical data. Here, we used the Curve Fitting Tool (*CFTool*) available in MATLAB toolbox for fitting the Lognormal pathloss model to the empirical RSSI values. As listed in Table 4.1 for all scenarios, an attenuation constant, n, is less than 2 used for free space. This is caused by the signal reflection from walls, ground, and ceiling. Reflection also causes the variation in the RSSI value as it strengthens or weakens the signal. Note that the n values for indoor office area are less than corridor because the reflection and scattering in office area are intensified since the indoor office contains several desks, cupboards, and PC. For the experiments conducted in corridor area, no direct line of sight path is available due to the location of the transmitter at the waist of the human body and the receiver near a computer server in the office area. So, it is expected for the received signal powers undergo more attenuation. This fact has been experimentally observed from the range of $RSSI_0$ values for d_0 in both environments. As listed in Table 4.1, the $RSSI_0$ values for corridor environment are -71.12 dBm, -72.23 dBm which are more than the values estimated in indoor office, -52.06 dBm and 58.7 dBm.

As the RSSI values have strong correlation with the packet reception rate and distance, they can be counted as good link indicators. We also focus the rest of our study on providing a statistical model to determine the variation of the received signal level over time and location. The proposed model would characterize the effect of multipath and pathloss pattern on received signal attenuation and distortion.

4.4 Fitting of Amplitude Distribution

A reliable statistical model is needed to determine how much the received signal level (RSSI) can vary. Using this approach, we can characterize the fading statistics on body links that occur with body motion and change of body position in the test environment. Based on the measurements explained in section 4.2, a total of 242820 samples of received power are logged at a rate of 70 Hz for two different environments. For each environment, the data are analyzed for three actions of standing, sitting, and walking resulting in 6 scenarios. We analyze the combination of multipath and pathloss effects by processing the distribution of the received signal power obtained over different locations inside the environment.

We characterize both large scale and small scale statistics by fitting the received powers at different locations of the measurement grid to the well known distributions including Normal, Lognormal, Gamma, Rician, and Weibull [26]. The probability density function (PDF) of the well-known distributions for the power of received signal, x, is given by:

- Normal

$$f(x|\mu,\sigma) = \frac{1}{\sigma\sqrt{2\pi}} exp\{\frac{-(x-\mu)^2}{2\sigma^2}\} \tag{4.2}$$

- Lognormal

$$f(x|\mu,\sigma) = \frac{1}{x\sigma\sqrt{2\pi}} exp\{\frac{-(ln(x)-\mu)^2}{2\sigma^2}\} \tag{4.3}$$

- Gamma

$$f(x|a,b) = \frac{1}{b^a\Gamma(a)}x^{a-1}exp\{\frac{x}{b}\} \tag{4.4}$$

- Weibull

$$f(x|a,b) = \begin{cases} ba^{-b}x^{b-1}exp\{\frac{-x}{a}\}^b & \text{if } x \geq 0 \\ 0, & \text{else} \end{cases}$$

- Rician

$$f(x|s,\sigma) = \frac{1}{\sigma^2}exp\{\frac{-(x^2+s^2)}{2\sigma^2}\}I_0(\frac{xs}{\sigma^2}) \tag{4.5}$$

All distribution parameter estimations are obtained on a 95 confidence interval using the maximum likelihood estimation via *mle(.)* function available in the statistics toolbox of MATLAB. We used the Akaike information criterion (AIC) rather than a hypothesis tests such as KStets for choosing the best model among the set of candidates [23]. The Akaike criterion will be briefly described in the next section.

4.4.1 Akaike Information Criterion

The AIC is an entropy-based model selection criterion that rewards goodness of fit, while at the same time penalizing for increased model orders with the aim of avoiding over fitting. The first-order AIC is defined as follows [79];

$$AIC = -2\log_e(\ell(\widehat{\theta})|data) + 2k, \tag{4.6}$$

where the expression $\log_e(\ell(\widehat{\theta})|data)$ is the numerical value of the log-likelihood at its maximum point. The maximum point on the log-likelihood function keeps up a correspondence to the values of the maximum likelihood estimates. The number of estimable parameters in the model is denoted by k. The first term $(\log_e(\ell(\widehat{\theta})|data))$

represents the log-likelihood estimates which indicates that better models have a lower AIC and tends to decrease as more parameters are added to the approximation model. On the other hand, the second term, $2k$, increases as more parameters are added to the approximation model. This is a trade of between under fitting and over fitting which helps us to select the model that has the best fit with the least number of parameters.

The AIC is on a relative scale which is strongly dependent on sample size and an individual AIC value, by itself, is not meaningful due to the unknown constant scale. In practice, the relative values of AIC among the model sets are used to select the best models. Thus, AIC differences Equation (4.7) and weights Equation (4.9) are usually interpreted as follows:

$$\Delta_i = AIC_i - min(AIC), \tag{4.7}$$

where AIC_i is the AIC value for the model index i. Δ_i values are used for a quick comparison, ranking of candidate models, and AIC weights. Clearly, the best model among the set of models has a delta AIC of 0. The larger Δ_i is, the less plausible it is in the way that values between 37 indicate that the model has considerably less support, while values greater than ten indicate that the model is very unlikely. The Akaike weight (Equation (4.9)) gives an efficient way to scale and interpret the Δ_i values. It is also called as the probability of being the best model in the set [23].

$$\omega_i = \frac{exp(\frac{-\Delta_i}{2})}{\sum_{r=1}^{R} exp(\frac{-\Delta_r}{2})}, \tag{4.8}$$

where R is the number of models. The ω_i depends on the whole set and it should be recomputed if a model is added or dropped. Moreover, the ratio of two AIC weights depicts how much more likely one model is better compared to the other. To summarize, in fitting application, the AIC value, difference, and weight should be computed for each of the candidate models.

4.4.2 Analysis of the results for Indoor office environment

Five probability distributions (Normal, Lognormal, Gamma, Rician, and Weibull) which all have been used to model WBAN channel [26], are considered in this study. Table 4.2 shows the summary of distribution parameter estimates, subsequent AIC values, deltas (Δ_i) and weights (ω_i). The results are reported separately for measurements taken in different actions of standing, sitting, and walking scenarios.

Model selection is based on the AIC theory. According to Akaike's theory, (explained in section 4.4.1) the most accurate model has the smallest AIC and its Akaike weight is one. Clearly, the AIC values prove that the Normal distribution provides a stronger fit to the received amplitude for action of standing. In addition to Normal distribution, this action is also supported by Rician distribution as its AIC weight is equal to 0.4. AIC indices calculated for the sitting scenario favored the Lognormal distribution to the received power signals. Although the AIC can be deployed to choose the best fitting distributions in the set of candidates, it is not surprising if none of the models provides the best fit to the measured received signal. In other words, if all the models are very poor, AIC will still choose the one estimated to be best even though that relatively best model might become poor in an absolute sense. Thus, every effort must be made to ensure that the set of models is well founded. We, therefore, need to confirm the best model graphically using Probability Density Function (PDF), and Cumulative Density Function (CDF). The theoretical Normal, Lognormal, Gamma, Rician, and Weibull models fitted to empirical PDF and CDF of the received amplitude are illustrated in Figure 4.12, Figure 4.13, and Figure 4.14.

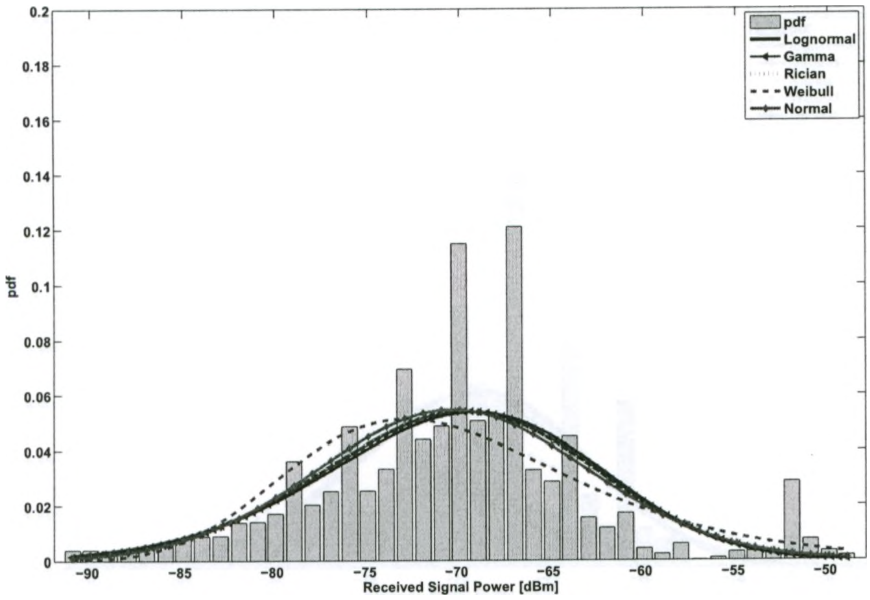
Table 4.2: MLE parameter estimated and AIC values computed for five common distributions for indoor office environment.

State	Distribution	*AIC* 10^5	Δ_i 10^3	ω_i	Parameter Estimate	Mean (dBm)	Std. Dev. (dB)
Standing	Normal	2.98048	0	0.8	μ = -70.42 $\sigma = 7.3$	-70.43	53.29
	Lognormal	3.00026	1.978	0	$\mu = 4.24$ $\sigma = 0.11$	-70.43	56.72
	Gamma	2.99146	1.098	0	a = 90.08 b = 0.78	-70.43	55.05
	Rician	2.98052	0.004	0.11	s = 70.04 $\sigma = 7.32$	-70.42	53.29
	Weibull	3.01208	3.160	0	a = 73.67 b = 6.38	-70.16	68.40
Sitting	Normal	2.90162	1.860	0	μ = -69.72 $\sigma = 6.38$	-69.72	40.76
	Lognormal	2.88302	0	1	$\mu = 4.24$ $\sigma = 0.09$	-69.72	39.56
	Gamma	2.88798	0.496	0	a = 122.32 b = 0.57	-69.72	39.74
	Rician	2.90154	1.852	0	s = 69.42	-69.72	40.75

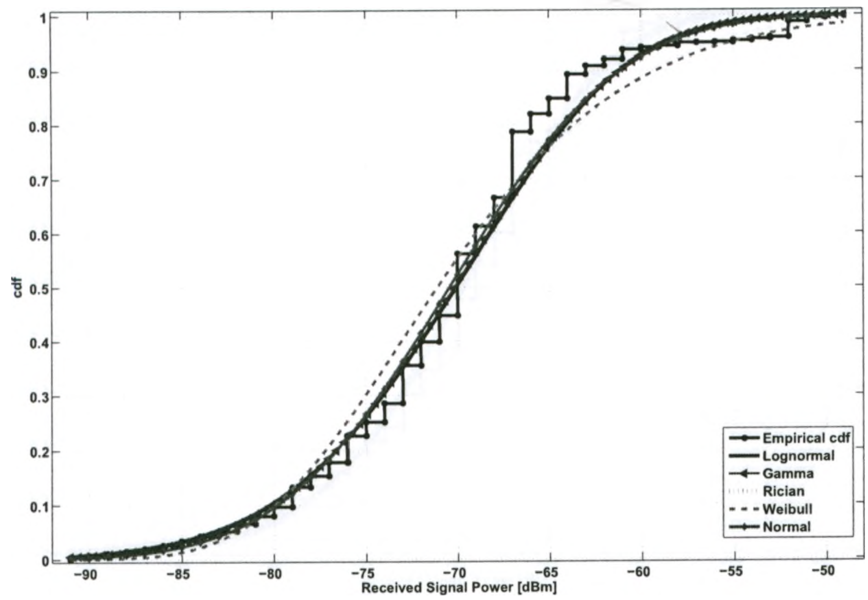
Continued on next page

Table 4.2 – continued from previous page

State	Distribution	AIC 10^5	Δ_i 10^3	ω_i	Parameter Estimate	Mean (dBm)	Std. Dev. (dB)
	Weibull	2.99662	11.360	0	$\sigma = 6.4$ a = 72.37 b = 10.42	-69.31	64.28
Walking	Normal	0.68208	2.912	0	$\mu = 69.35$ $\sigma = 6.37$	-69.35	40.65
	Lognormal	0.67917	0	1	$\mu = 4.23$ $\sigma = 0.091$	-69.35	40.03
	Gamma	0.67984	0.668	0	a = 120.19 b = 0.57	-69.35	40.01
	Rician	0.68207	2.9	0	s = 69.05 $\sigma = 6.38$	-69.35	60.12
	Weibull	0.70076	21.592	0	a = 72.33 b = 10.68	-69.35	74.35

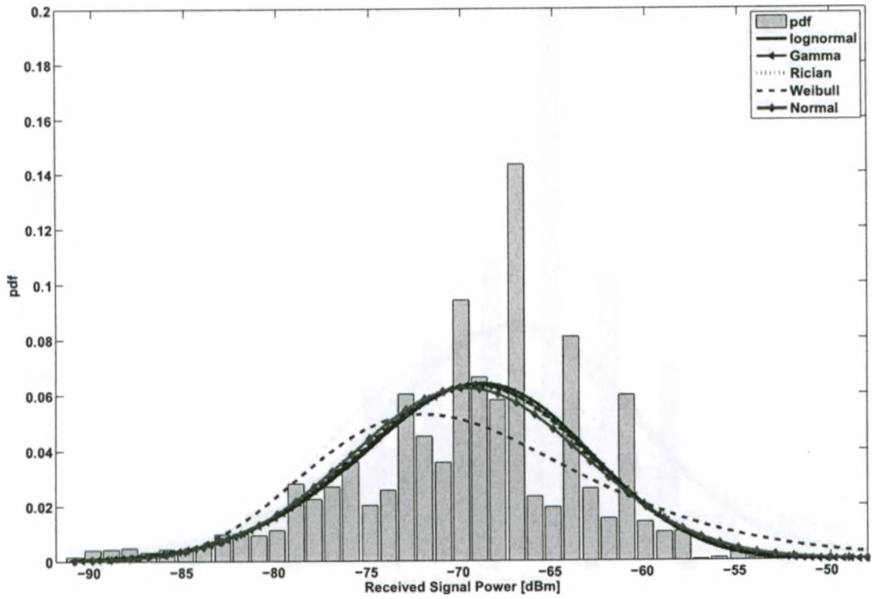


(a)

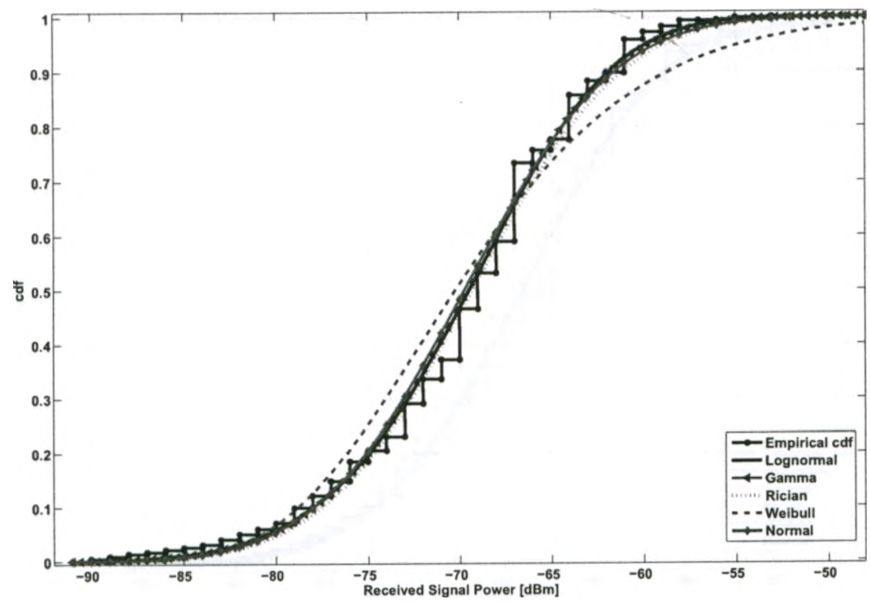


(b)

Figure 4.12: Empirical and MLE fitted (a) PDF and (b) CDF of the received signal power for the action of standing in indoor office area.

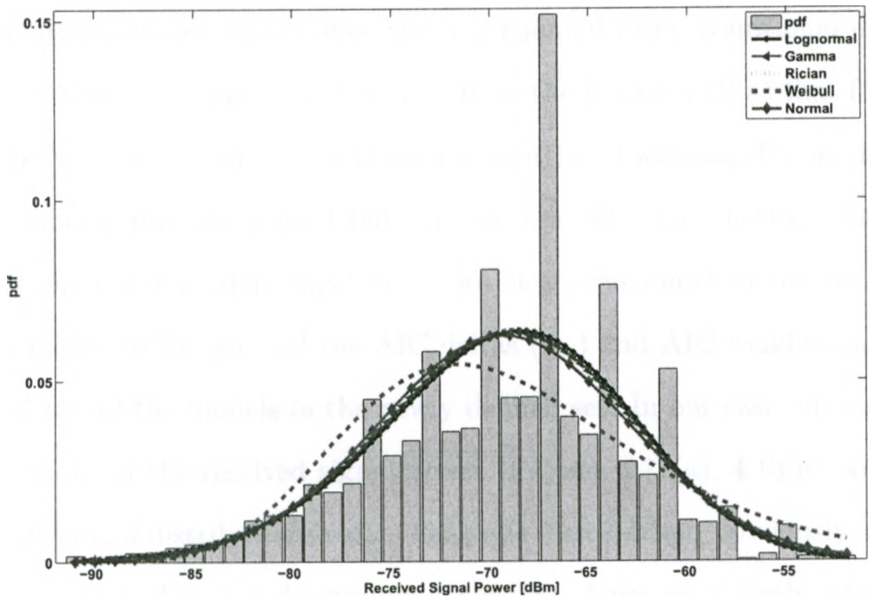


(a)

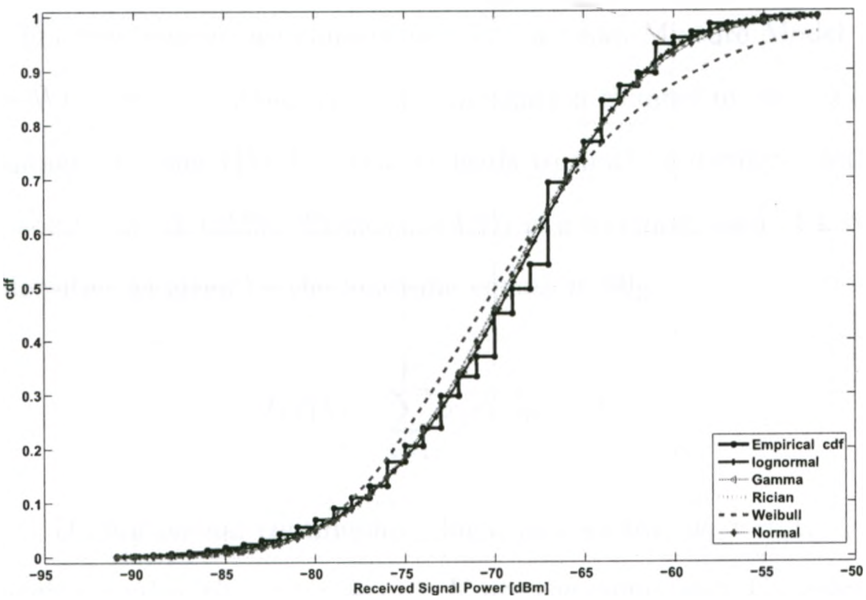


(b)

Figure 4.13: Empirical and MLE fitted (a) PDF and (b) CDF of the received signal power for the action of sitting in indoor office area.



(a)



(b)

Figure 4.14: Empirical and MLE fitted (a) PDF and (b) CDF of the received signal power for the action of walking in indoor office area.

The PDF and CDF plots (Figure 4.12, Figure 4.13, and Figure 4.14) suggest that the analysis of the indoor office wireless channel using traditional distribution may not be entirely appropriate. It is clear from the Figure 4.12(a) that for standing scenario the Normal and for sitting (Figure 4.13(a)) and walking (Figure 4.14(a)) the lognormal models provide a good but not the best fit. This notices that we need to consider the other models which have not been concluded in the set. We shall add a new model to the set and the AIC deltas (Δ_i) and AIC weights (ω_i) must be recomputed for all the models in the newly defined set. In our case, an inspection of the distributions of the received signal power (Figures 4.12(a), 4.13(a), and 4.14(a)) reflects a mixture of distributions rather than one distribution. Obviously, their PDFs consist of several random and deterministic clusters. A practical method for modeling the probability density function including multiple clusters is to fit the resulting amplitude distributions with a mixture of a number of Gaussian probability density functions. For this reason, we concentrate on Gaussian Mixture Model (GMM) to explain the Wireless body Area Network propagation channel in an indoor office. A main advantage of using GMM is that it leads to mathematically tractable signal processing solutions. A GMM (Equation (4.9)) is a weighted sum of k components Gaussian densities as given by the following equation [80],

$$f(x|\lambda) = \sum_{i=1}^{K} w_i g(x|\mu_i, \sigma_i), \tag{4.9}$$

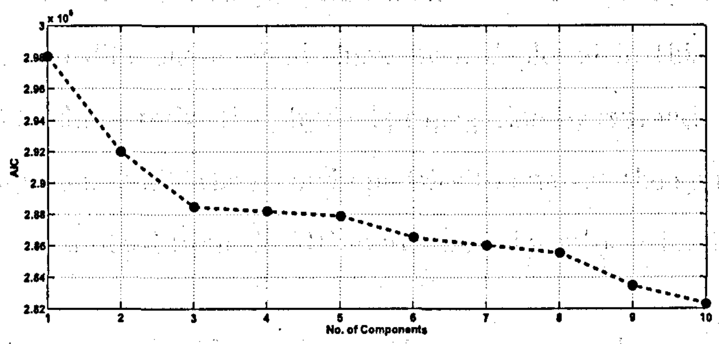
where x is a D-dimensional continuous-valued data vector, $w_i, i = 1, ..., K$, are the mixture weights, and $g(x|\mu_i, \sigma_i), i = 1, ..., K$ are the component Gaussian densities, with the mean vector μ_k and covariance matrix σ_i. The mixture weights satisfy the constraint that $\Sigma_{i=1}^{K} w_i = 1$. The mixture weight can be interpreted as the expected fraction of the number of vectors from the process x_i associated with the

i^{th} mixture. The complete Gaussian mixture model is parameterized by the mean vectors, covariance matrices and mixture weights from all component densities. These parameters are collectively represented by the notation,

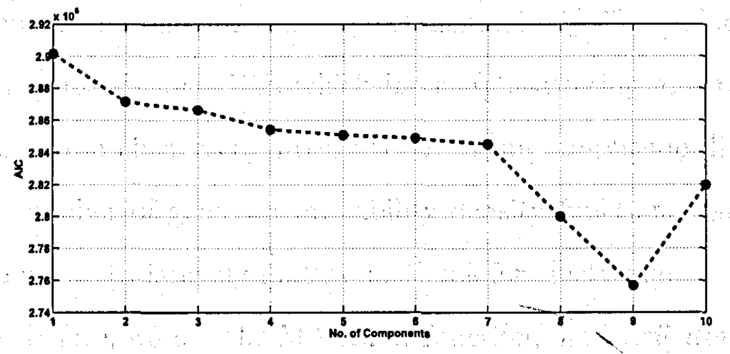
$$\lambda = \{w_i, \mu_i, \sigma_i\}, i = 1, 2, ..., K. \tag{4.10}$$

The EM algorithm [81] which is an iterative maximum likelihood (ML) estimation method has been employed to calculate the parameters of a k-mixture Gaussian PDF model for measured signals. In the statistics toolbox of MATLAB, the function *fit (gmdistribution)* fits GMM to the data utilizing the expectation maximization (EM) algorithm. This algorithm assigns posterior probabilities to each component density with respect to each observation. Clusters are assigned by selecting the component that maximizes the posterior probability. Similar to k-means clustering, GMM uses an iterative algorithm that converges to a local optimum.

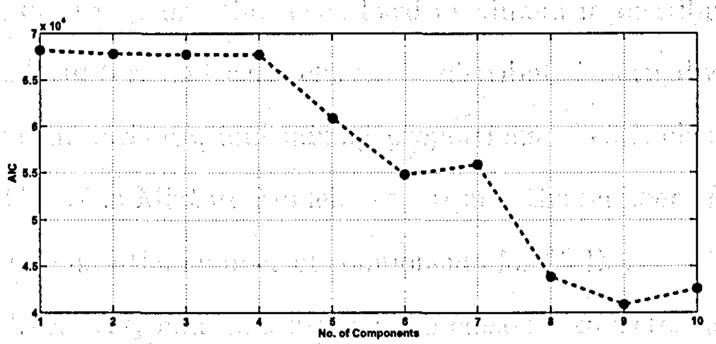
Besides graphical visualization using PDF and CDF, the Akaike Information Criterion (AIC) is also used to determine an appropriate number of components for a model when the number of components is unspecified in GMM. Here, we computed AIC for GMM with finite number of components ($k = 1, 2, ..., 10$) and the results are shown in Figure 4.15. As indicated in the figure, the AIC value decreases as the number of component increases.



(a)



(b)



(c)

Figure 4.15: The Akaike Information Criterion computed for GMM with finite number of components ($k = 1, 2, ..., 10$) for the actions of (a)standing, (b)sitting, and (c)walking.

In order to determine the appropriate number of GMM components with respect to AIC values, two different methods have been deployed in this study. The first approach seeks to characterize the physical propagation by extracting several clusters from the distribution, While the second one tries to analyze the system performance for the signals transmitted from the RF source worn by subject in different scenarios.

4.4.2.1 First approach: Cluster and Ray components

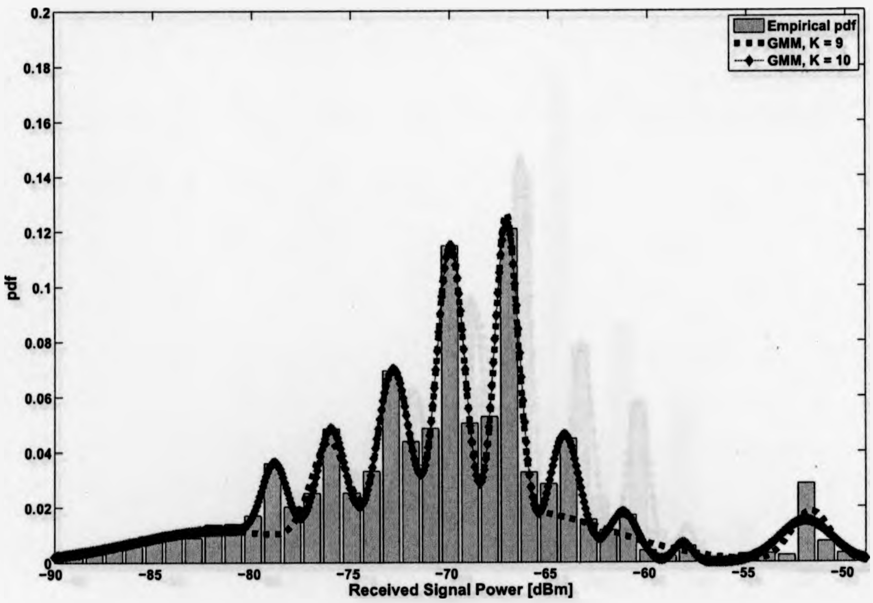
The distributions of received signal consist of several clusters due to especial condition of the measurement environment and shape of human body. The indoor office contains several desks, cupboards, and PCs which intensify scattering and reflection effects. We characterize these clusters by fitting the resulting amplitude distributions with a mixture of a number of gaussian probability density functions; Gaussian Mixture Models are the most statistically mature methods for clustering.

As stated before, we use the MATLAB function, *gmdistribution.fit*, in order to fit the multivariate Gaussian mixture to the data set. Each multivariate Gaussian component is defined by its mean and variance, and the mixing proportions. For the sake of simplicity, we assume that each local maximum in distribution corresponds to an individual cluster, and each cluster is distributed normally. We, therefore, estimate the mean, variance, and mixing proportions of each cluster as a separate component of Gaussian Mixture Model. As a result, the number of discrete picks or local maximum defines the number of components for GMM.

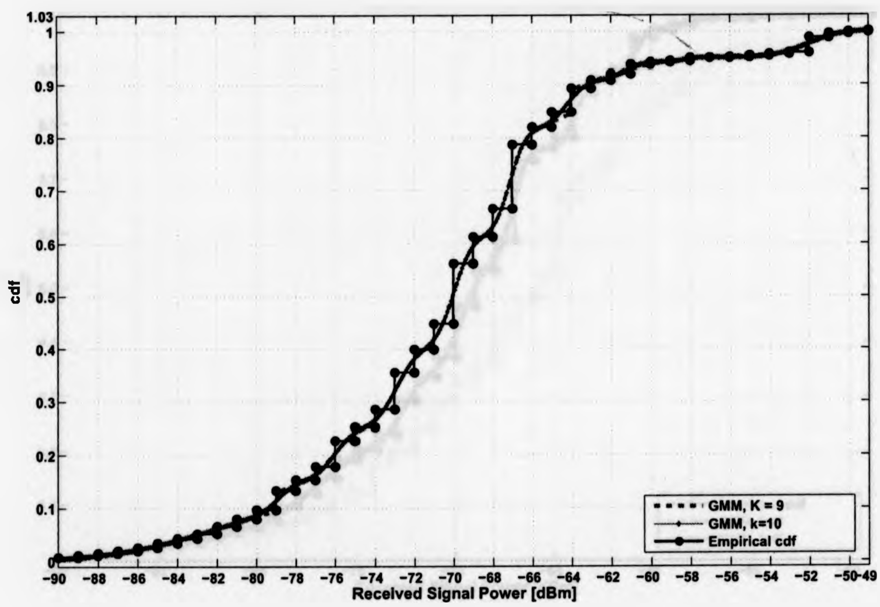
Figures 4.16, 1.17, and 4.18 display the clusters extracted from all measurements in an indoor office. For standing scenario, we found that the GMM with 10 components gives the best fit with the empirical distribution (Figure 4.16(a)). This is particularly true as it can totally characterize the mixture of clusters observed in distribution and its subsequent AIC value is the minimum among the AIC values of the GMM with finite components from 1 to 10 (Fig. 4.15(a)). However, we tried to fit the GMM with less components to the data set but an ill-conditioned covariance matrix occurred while estimating the parameters in a GMM with number of components between 5 and 10. For instance, we applied the GMM with 9 components to the empirical distribution and the ill-conditioned covariance matrices were removed by adding a very small positive number to the diagonal of covariance matrix. The empirical PDF and CDF diagrams of GMM with 9 components are plotted to ensure that GMM with 10 components provides a better fit. The results are shown in the same plot (Figure 1.16). It is obvious from the PDF plot (Figure 4.16(a)) that 9 components can't characterize all the clusters as well as 10 components. However, the difference between CDF plots, Figure 4.16(b), is negligible.

The GMM with 9 components was also found to account for user sitting scenario in this environment (Figure 4.17(a) and 4.17(b)). Similar to standing scenario, there were an ill-conditioned covariance matrix for estimating the parameters in a Gaussian Mixture Model with number of components less than 9, for instance 8 components.

Finally, the GMM with 9 components provided a superior fit to the measured received power for walking scenario (Figure 4.18(a) and 4.18(b)).

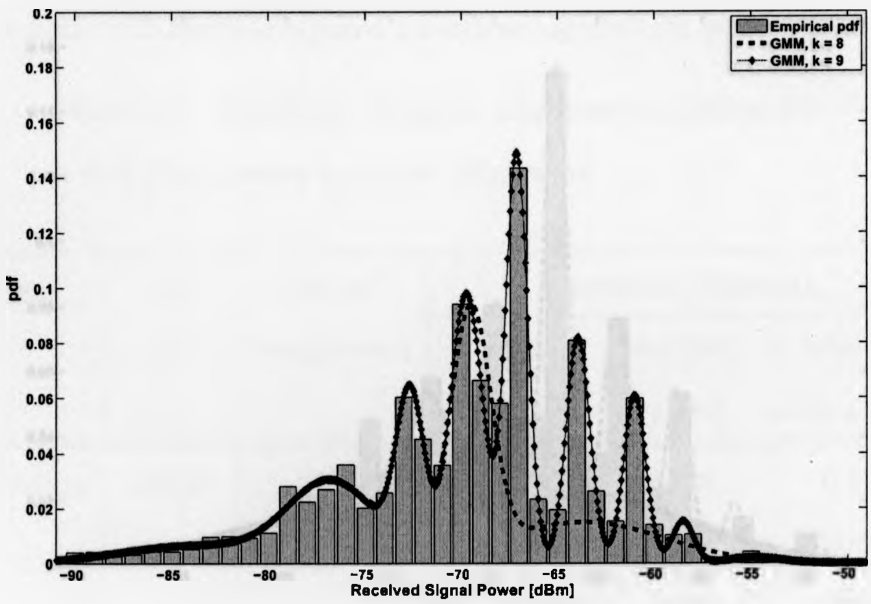


(a)

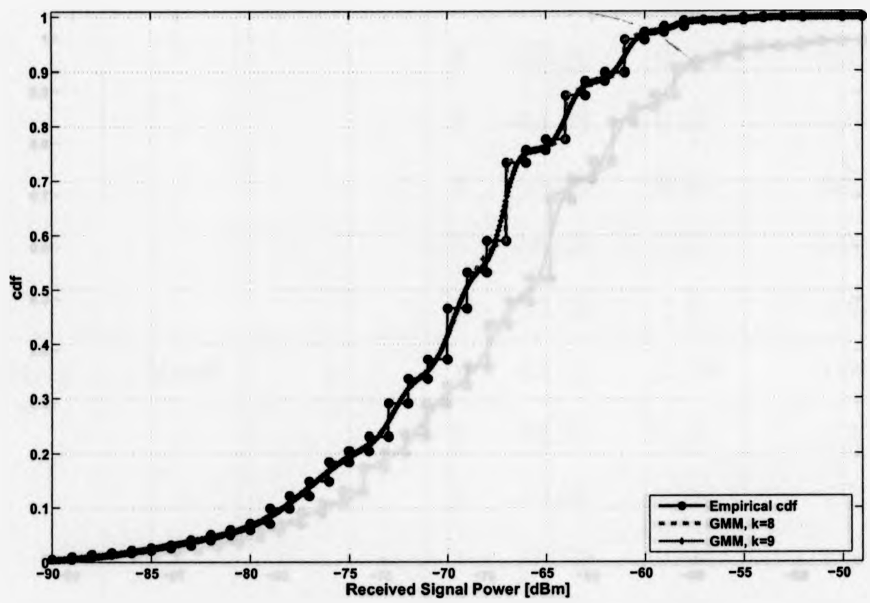


(b)

Figure 4.16: Estimated GMM (a) PDF and (b) CDF with 9 and 10 components for the action of standing in indoor office area.

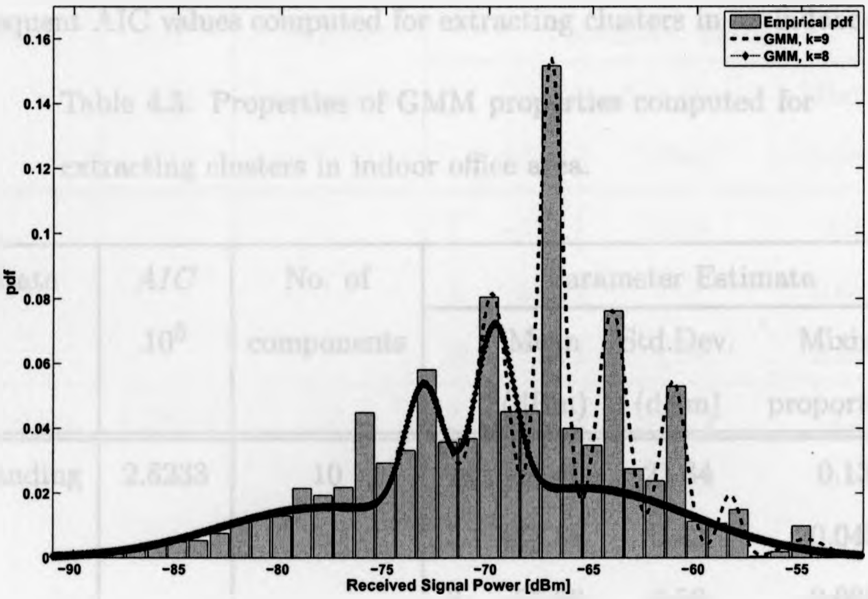


(a)

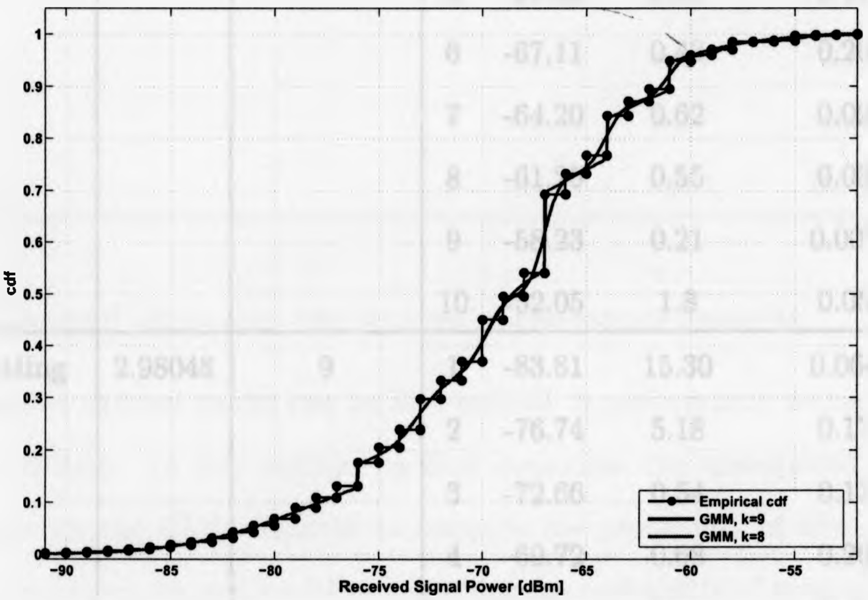


(b)

Figure 4.17: Estimated GMM (a) PDF and (b) CDF with 8 and 9 components for the action of sitting in indoor office area.



(a)



(b)

Figure 4.18: Estimated GMM (a) PDF and (b) CDF with 8 and 9 components for the action of walking in indoor office area.

Table 4.3 presents the detail of GMM parameter estimated using EM algorithm and subsequent AIC values computed for extracting clusters in an indoor offices.

Table 4.3: Properties of GMM properties computed for extracting clusters in indoor office area.

State	*AIC* 10^5	No. of components	Parameter Estimate			
				Mean (dBm)	Std.Dev. (dBm)	Mixing proportions
Standing	2.8233	10	1	-81.46	21.34	0.13
			2	-78.89	0.42	0.043
			3	-75.99	0.56	0.083
			4	-72.88	0.73	0.14
			5	-69.96	0.53	0.21
			6	-67.11	0.43	0.20
			7	-64.20	0.62	0.09
			8	-61.25	0.55	0.03
			9	-58.23	0.21	0.007
			10	-52.05	1.8	0.05
Sitting	2.98048	9	1	-83.81	15.30	0.064
			2	-76.74	5.18	0.17
			3	-72.66	0.54	0.11
			4	-69.72	0.68	0.20
			5	-67.11	0.33	0.21
			6	-63.94	0.38	0.13

Continued on next page

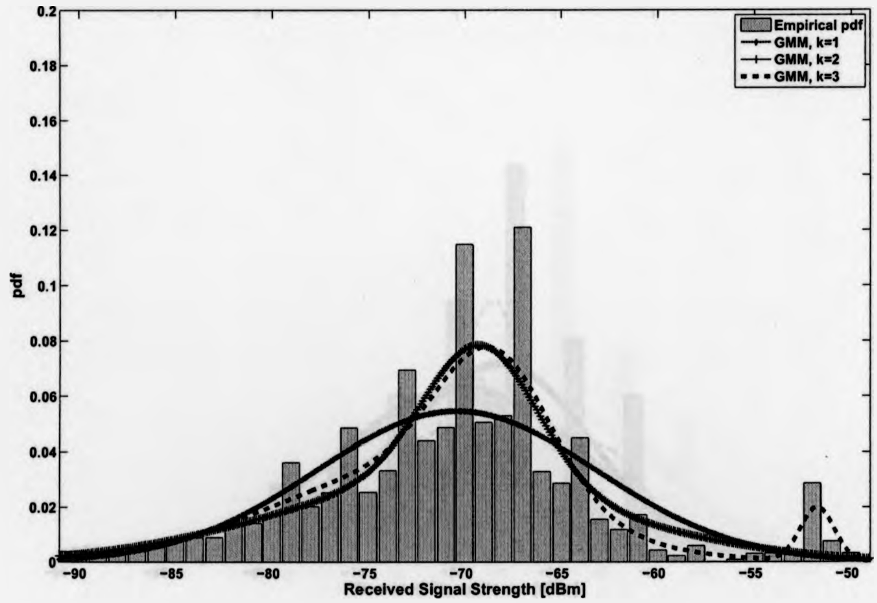
Table 4.3 – continued from previous page

State	AIC 10^5	No. of components	Parameter Estimate			
				Mean (dBm)	Std.Dev. (dBm)	Mixing proportions
			7	-61.01	0.33	0.08
			8	-58.51	0.3	0.021
			9	-54.78	1.84	0.006
Walking	2.98048	13	1	-77.82	25.12	0.2
			2	-76	2e-20	0.04
			3	-73.00	1.088	0.11
			4	-69.83	0.57	0.14
			5	-67.01	0.35	0.22
			6	-64.06	0.54	0.14
			7	-61.07	0.40	0.08
			7	-58.41	0.27	0.025
			9	-54.77	1.057	0.014

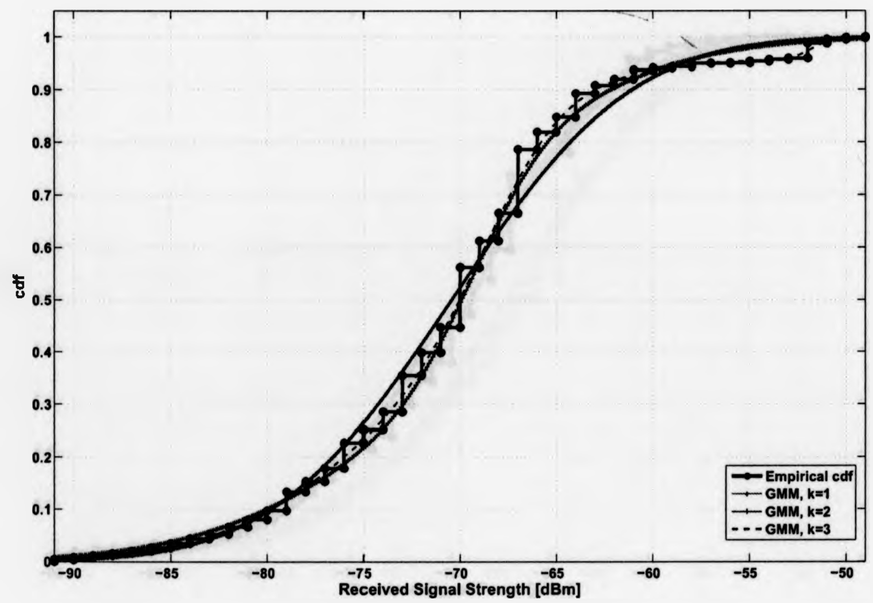
4.4.2.2 Second approach: the system performance analysis

The presented channel model can be also utilized in performance evaluation of the prototype system. In this section, we first determine the appropriate number of components for the GMM required to compute the probability of error. Then we apply the proposed channel model in evaluating the probability of error (section 4.5) for the modulation signals.

To obtain the error probabilities, one should average the conditional probability of error for the OQPSK modulation over the probability density function of signal which depends on the ratio and not on any other detailed characteristics of the signals distribution. As a result, in contrast with the previous approach, here we don't need to characterize the channel with many components. Theretofore, we shall minimize the number of components. For this sake, AIC will be employed to determine an appropriate number of components for a model. In this case, the GMM with 1,2, and 3 were applied to the data set in this environment. The results of fitted model and empirical PDF and CDF of the received signal powers are displayed in Figures 4.19, 4.20 and 4.21 for different scenarios of standing, sitting, and walking, respectively. Table 4.4 presents the estimated GMM with 1,2, and 3 number of component and subsequent AIC values, deltas and weights. Based on the AIC values, we found that GMM with three components obtains the superior fit compare to all three user status models in this environment. It is particularly true as their subsequent AIC values and deltas are minimum. The goodness of this fitting can be also observed from the PDF plots of GMM fitted to the data set (Figures 4.19(a), 4.20(a), and 4.21(a)). The parameters of the GMM with 3 components are listed in Table 4.5.

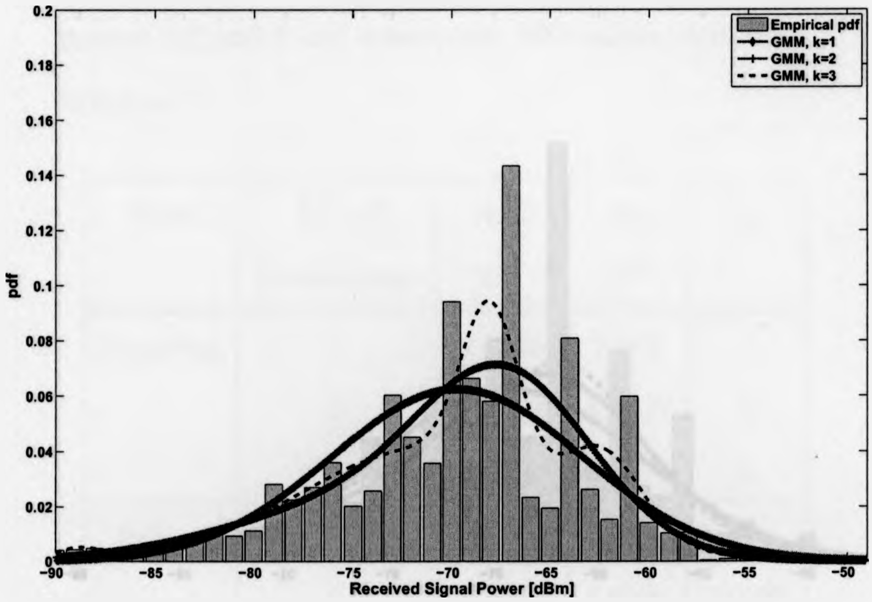


(a)

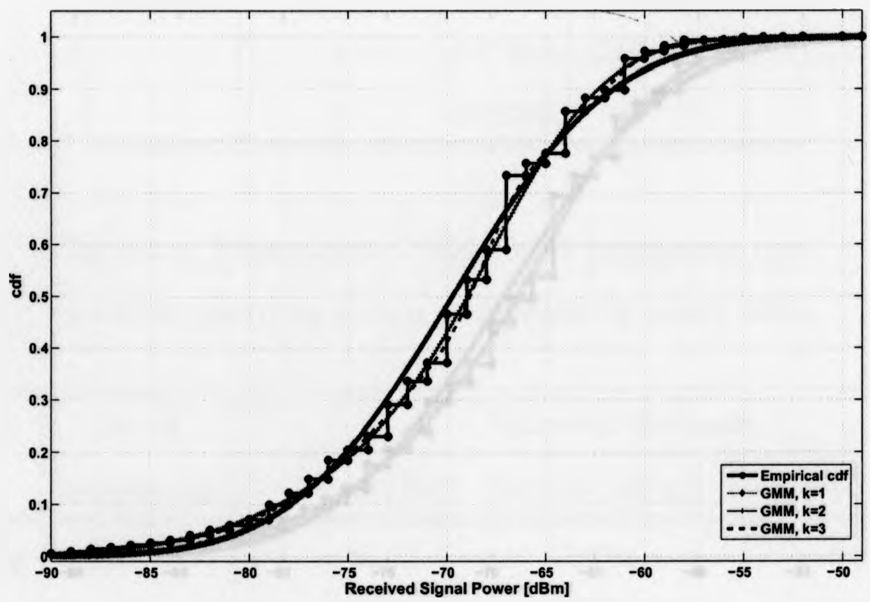


(b)

Figure 4.19: Estimated GMM (a) PDF and (b) CDF with 1,2 and 3 components for the action of standing in indoor office area.

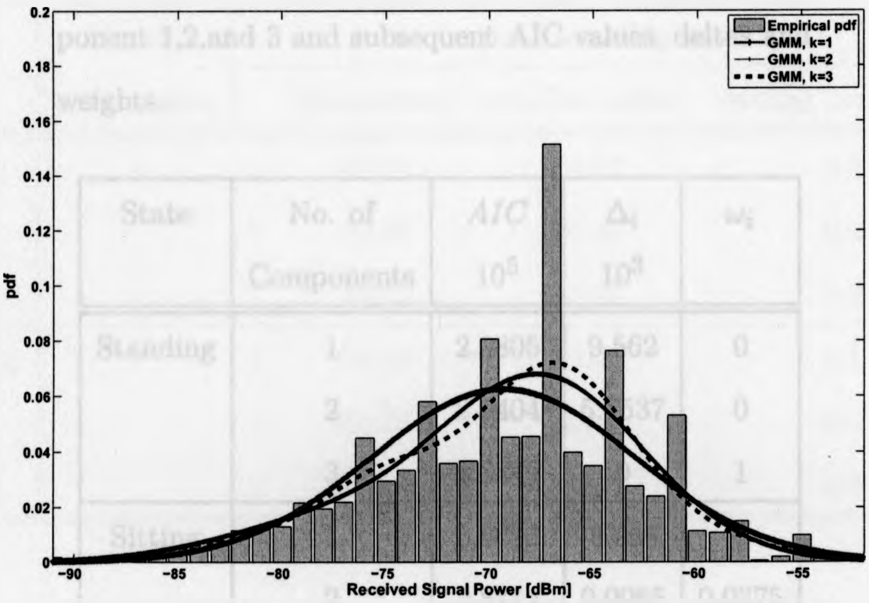


(a)

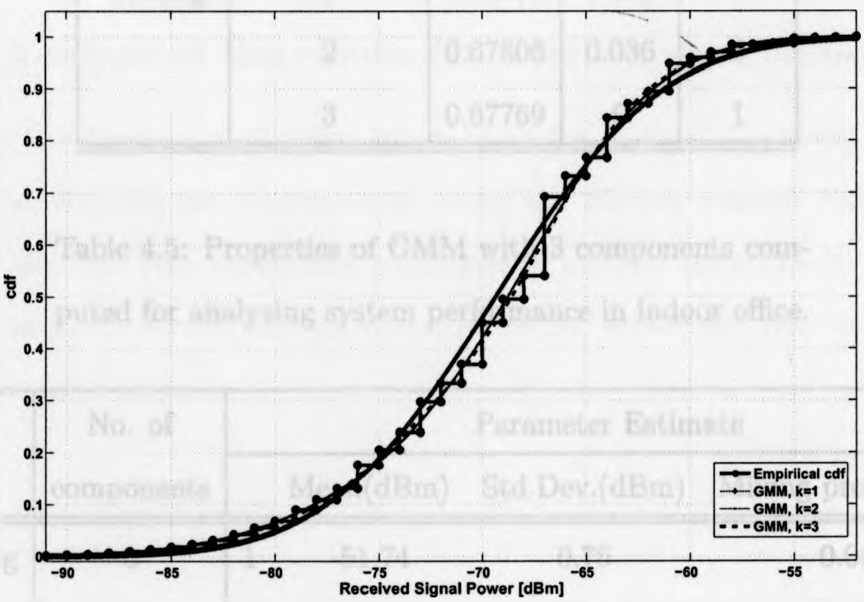


(b)

Figure 4.20: Estimated GMM (a) PDF and (b) CDF with 1,2 and 3 components for the action of sitting in indoor office area.



(a)



(b)

Figure 4.21: Estimated GMM (a) PDF and (b) CDF with 1,2 and 3 components for the action of walking in indoor office area.

Table 4.4: Estimated GMM with finite number of component 1,2,and 3 and subsequent AIC values, deltas and weights.

State	No. of Components	AIC 10^5	Δ_i 10^3	ω_i
Standing	1	2.9805	9.562	0
	2	2.9404	5.5537	0
	3	2.8849	0	1
Sitting	1	2.9016	3.026	0
	2	2.8714	0.0065	0.0375
	3	2.8715	0	0.9625
Walking	1	0.68208	0.438	0
	2	0.67806	0.036	0
	3	0.67769	0	1

Table 4.5: Properties of GMM with 3 components computed for analyzing system performance in indoor office.

State	No. of components	Parameter Estimate			
			Mean(dBm)	Std.Dev.(dBm)	Mixing proportions
Standing	3	1	-51.74	0.76	0.04
		2	-73.06	50.77	0.58
		3	-68.48	8.67	0.37

Continued on next page

Table 4.5 – continued from previous page

State	No. of components	Parameter Estimate			
			Mean(dBm)	Std.Dev.(dBm)	Mixing proportions
Sitting	3	1	-67.63	16.97	0.36
		2	-74.32	46.46	0.36
		3	-66.52	20.91	0.28
Walking	3	1	-76.99	32.95	0.15
		2	-68.53	43.9	35.03
		3	-66.28	5.6	9.61

We will use this channel model in order to calculate the probability of error for the O-QPSK modulated signals in the next section 4.5.

4.4.3 Analysis of the results for Corridor environment

Again, the five common probability distributions (i.e. Normal, Lognormal, Gamma, Rician, and Weibull) are considered to model the wireless channel between BAN nodes for the case that experiment performed in corridor environment. Table 4.6 briefly presents the distribution parameter estimates, subsequent AIC values, deltas (Δ_i) and weights (ω_i).

Table 4.6: MLE parameter estimated and AIC values computed for five common distributions in corridor environment.

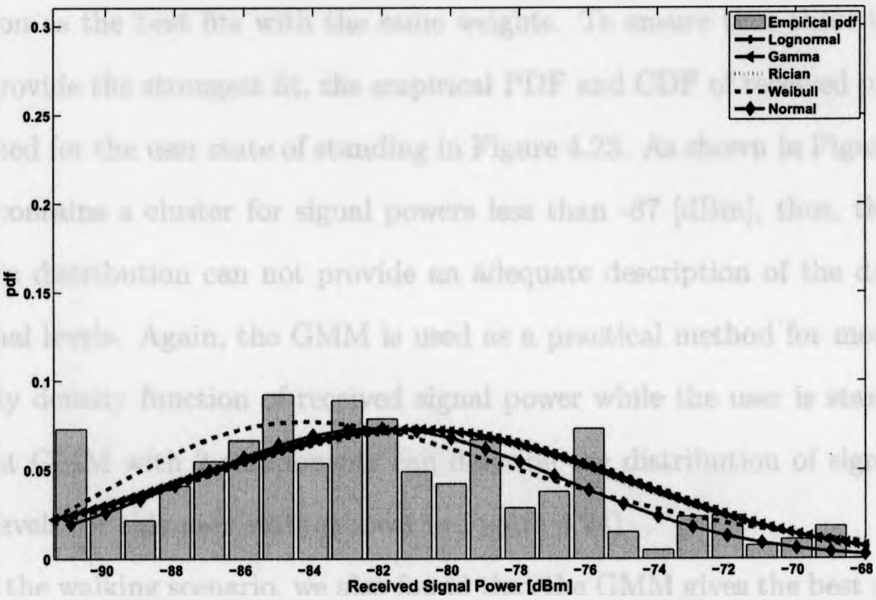
State	Distribution	AIC 10^5	Δ_i 10^3	ω_i	Parameter Estimate	Mean (dBm)	Std. Dev. (dB)
Standing	Normal	2.00224	0	0.5	μ = -81.64 $\sigma = 46.51$	-81.64	6.82
	Lognormal	2.00496	272	0	$\mu = 4.4$ $\sigma = 0.08$	-81.64	47.44
	Gamma	2.00360	136	0	a = 141.98 b = 0.57	-81.64	46.94
	Rician	2.00224	0	0.5	s = 81.35 $\sigma = 6.83$	-81.64	46.51
	Weibull	3.01208	4360	0	a = 84.77 b = 13.7	-81.62	52.98
Sitting	Normal	1.771	1860	0	$\mu = 82.17$ $\sigma = 5.44$	-82.17	29.65
	Lognormal	1.6761	1558	0	$\mu = 4.4$ $\sigma = 0.06$	-82.17	30.74
	Gamma	1.6732	1262	0	a = 222.91 b = 0.36	-82.17	30.28
	Rician	1.6683	772	0	s = 81.98 $\sigma = 5.45$	-82.17	29.65

Continued on next page

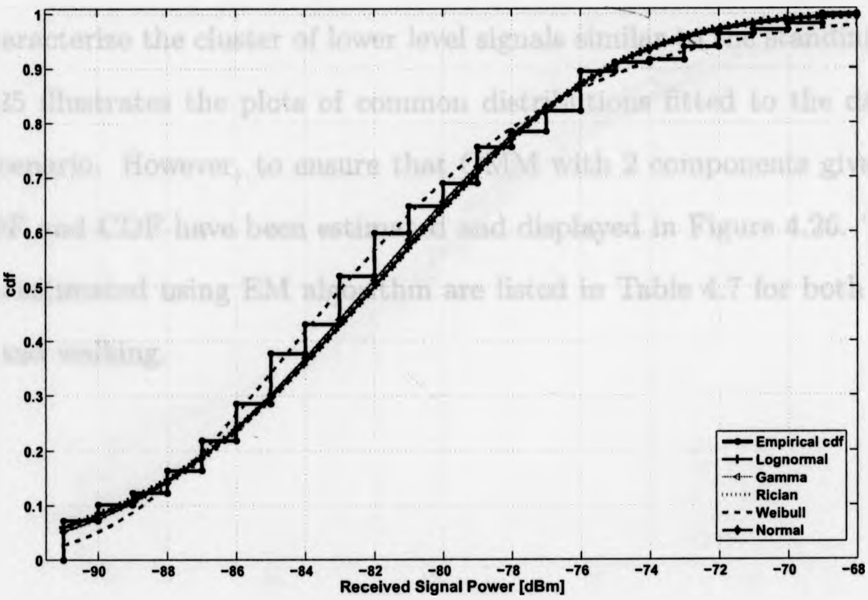
Table 4.6 – continued from previous page

State	Distribution	AIC 10^5	Δ_i 10^3	ω_i	Parameter Estimate	Mean (dBm)	Std. Dev. (dB)
	Weibull	1.6606	0	1	a = 84.64 b = 17.68	-82.13	32.88
Walking	Normal	5.9109	421	0	μ = 83.00 σ = 6.28	-83.00	39.42
	Lognormal	5.9350	662	0	μ = 4.41 σ =0.076	-83.01	40.48
	Gamma	5.9258	570	0	a = 171.27 b = 0.48	-83.00	40.23
	Rician	5.9108	420	0	s = 82.77 σ = 6.29	-83.00	39.42
	Weibull	5.8688	0	1	a = 85.86 b = 15.91	-83.06	41.18

Weibull distribution suggests a superior fit to the distribution of received power while the user is sitting in different locations along corridor. The AIC indices also verify the accuracy of this fitting. It is apparat in the the Figures 4.22(a) and 4.22(b) that the weibull model provides the best fit among all candidates.



(a)

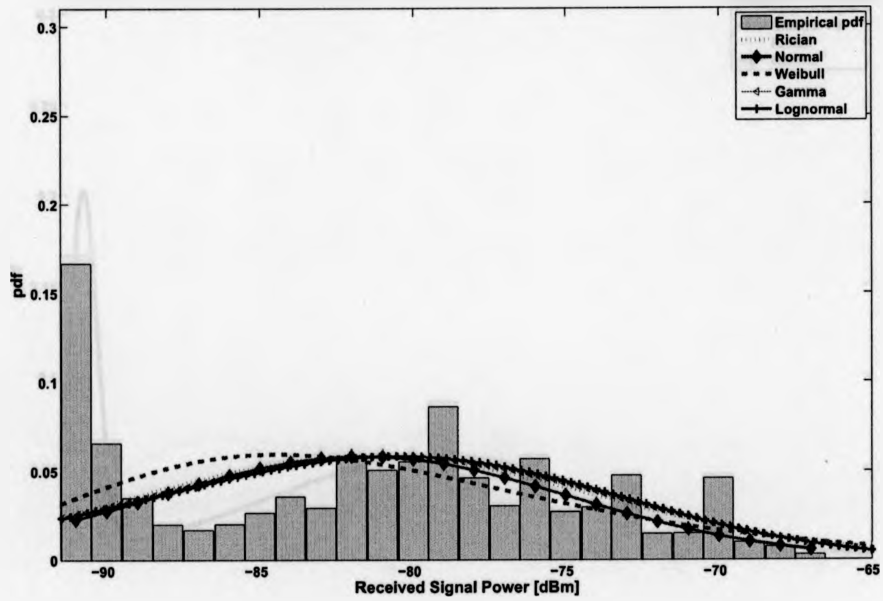


(b)

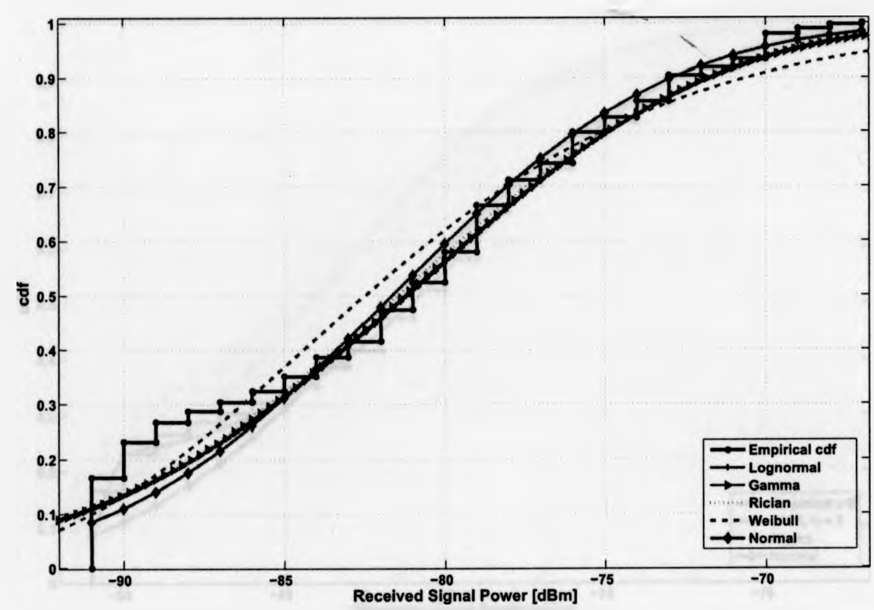
Figure 4.22: Empirical and MLE fitted (a) PDF and (b) CDF of the received signal power for the sitting scenario in corridor area.

For the action of standing, the AIC theory suggests the Normal and Rician distribution as the best fits with the same weights. To ensure that these two distributions provide the strongest fit, the empirical PDF and CDF of received power have been plotted for the user state of standing in Figure 4.23. As shown in Figure 4.23(a), its PDF contains a cluster for signal powers less than -87 [dBm], thus, the Normal and Rician distribution can not provide an adequate description of the data at the lower signal levels. Again, the GMM is used as a practical method for modeling the probability density function of received signal power while the user is standing. We found that GMM with 2 components can describe the distribution of signal powers from all levels for this user state (shown in Figure 4.24).

For the walking scenario, we also found that the GMM gives the best agreement with the empirical distribution. Although the AIC indices listed in Table 4.6 favored the Weibull distribution as the best model for user action of walking, this model cannot characterize the cluster of lower level signals similar to the standing scenario. Figure 4.25 illustrates the plots of common distributions fitted to the data set for walking scenario. However, to ensure that GMM with 2 components gives a better fit, its PDF and CDF have been estimated and displayed in Figure 4.26. The GMM properties estimated using EM algorithm are listed in Table 4.7 for both actions of standing and walking.

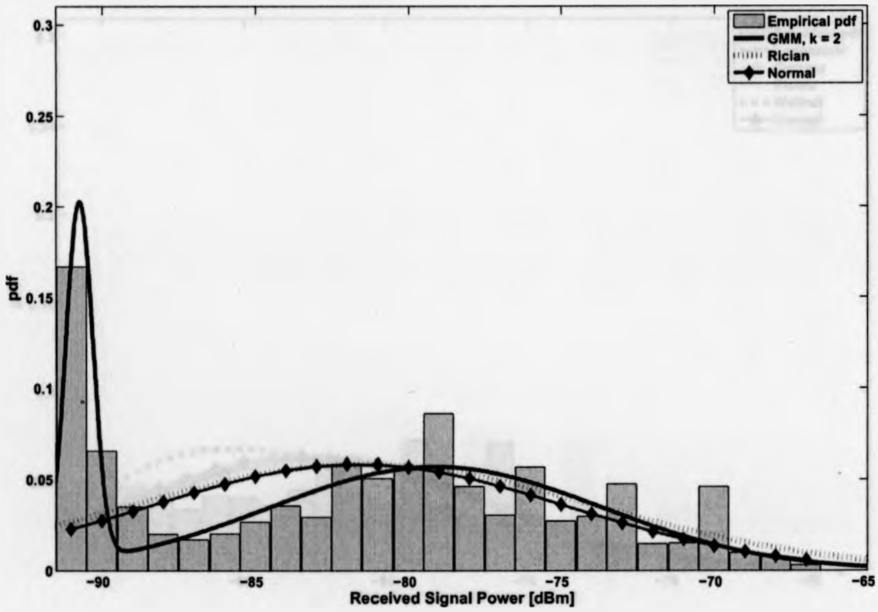


(a)

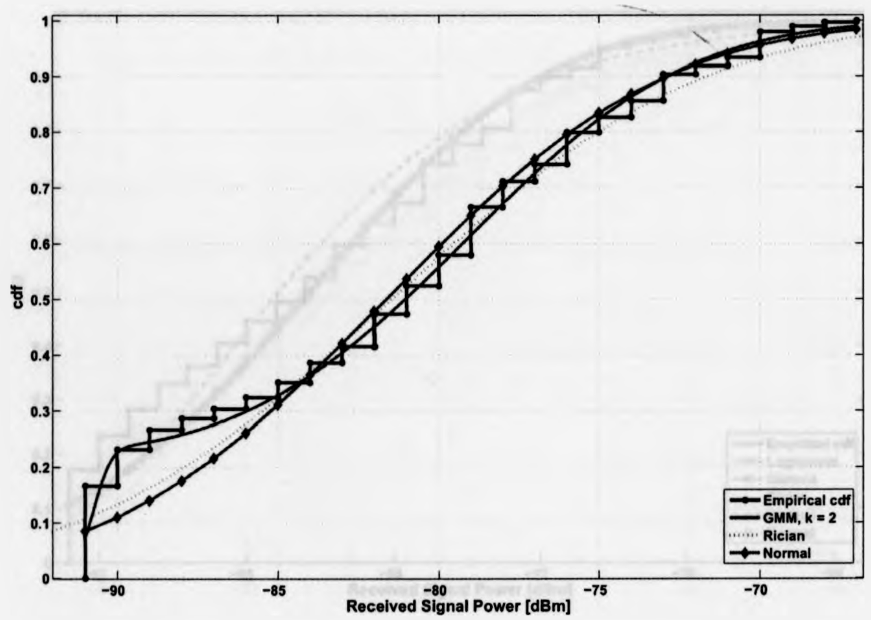


(b)

Figure 4.23: Empirical and MLE fitted (a) PDF and (b) CDF of the received signal power for the standing scenario in corridor area.

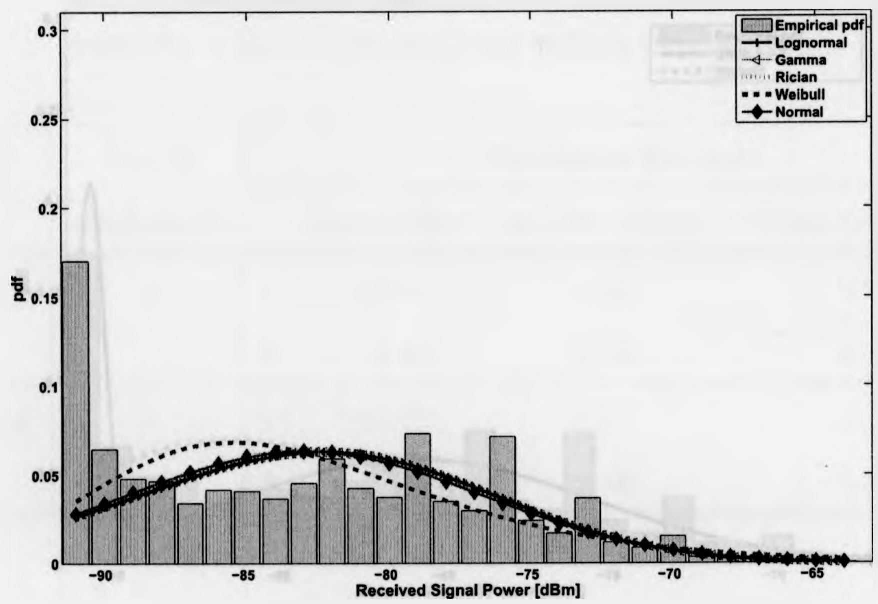


(a)

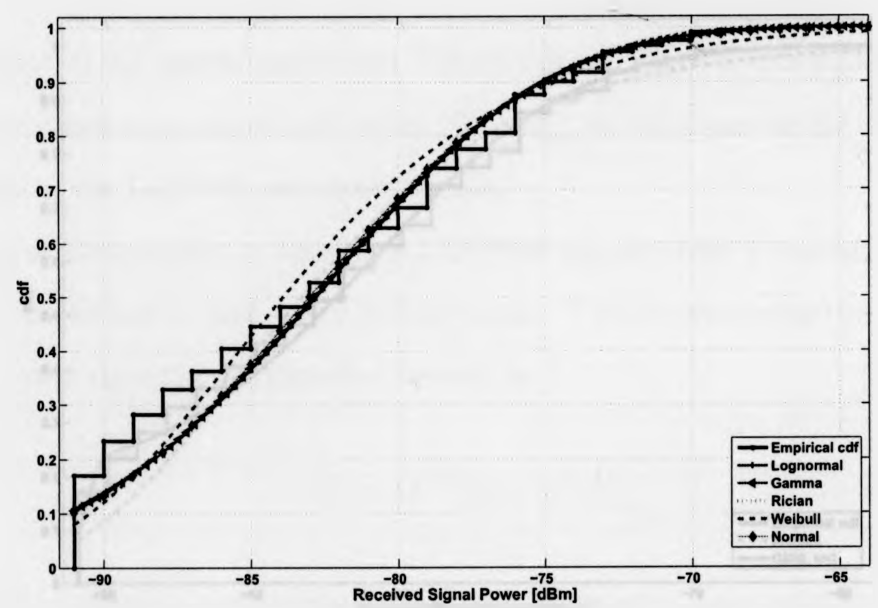


(b)

Figure 4.24: Estimated GMM and MLE fitted (a) PDF and (b) CDF of the received signal power for the standing scenario in corridor area.

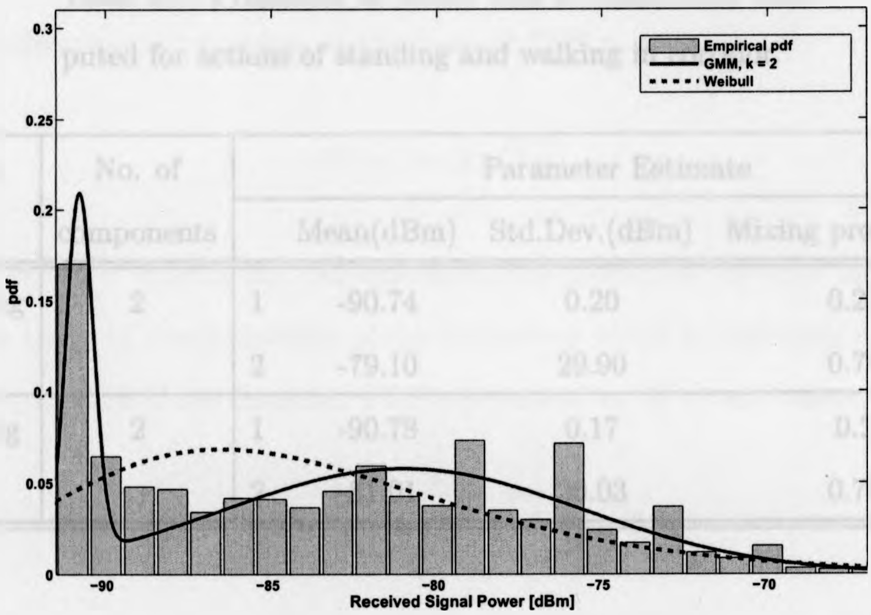


(a)

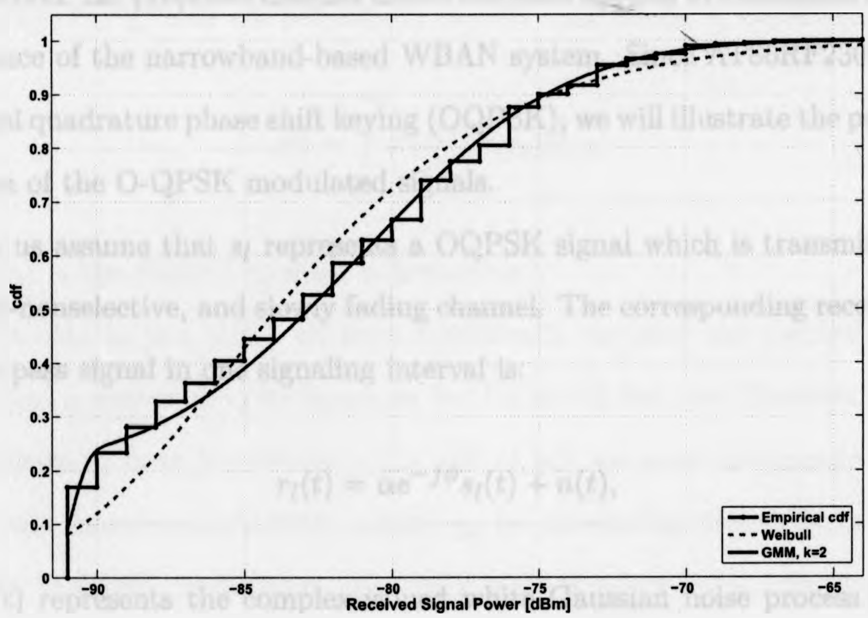


(b)

Figure 4.25: Empirical and MLE fitted (a) PDF and (b) CDF of the received signal power for the walking scenario in corridor area.



(a)



(b)

Figure 4.26: Estimated GMM and MLE fitted (a) PDF and (b) CDF of the received signal power for the walking scenario in corridor area.

Table 4.7: Properties of GMM with 2 components computed for actions of standing and walking in corridor.

State	No. of components	Parameter Estimate			
			Mean(dBm)	Std.Dev.(dBm)	Mixing proportions
Standing	2	1	-90.74	0.20	0.22
		2	-79.10	29.90	0.78
Walking	2	1	-90.78	0.17	0.2
		2	-81.01	30.03	0.79

4.5 Probability of Error for OQPSK modulation

In this section, the proposed channel model has been applied in evaluation of the BER performance of the narrowband-based WBAN system. Since AT86RF230 radio uses orthogonal quadrature phase shift keying (OQPSK), we will illustrate the performance evaluation of the O-QPSK modulated signals.

Let us assume that s_l represents a OQPSK signal which is transmitted over a frequency-nonselective, and slowly fading channel. The corresponding received equivalent low pass signal in one signaling interval is:

$$r_l(t) = \alpha e^{-j\phi} s_l(t) + n(t), \tag{4.11}$$

where $n(t)$ represents the complex-valued white Gaussian noise process corrupting the signal.

For error probability analysis, we assumed our modeled WBAN channel fading is sufficiently slow that the phase shift ϕ can be estimated from the received signal without error. For a fixed attenuation α, the conditional bit error probability for QPSK modulation as a function of the received SNR γ_b is:

$$P_2(\gamma_b) = Q(\sqrt{2\gamma_b}), \tag{4.12}$$

where $\gamma_b = \frac{P_\alpha P_s}{P_n} = \alpha^2 \varepsilon_b / N_0$. As explained, the probability of error is always expressed in terms of the probability of the Q-function which is frequently used for the area under the tail of the Gaussian PDF represented by $Q(x)$ and defined as:

$$Q(x) = \frac{1}{\sqrt{2\pi}} \int_x^\infty e^{-\frac{x^2}{2}} dx, x \geq 0, \tag{4.13}$$

Equation (4.12) is viewed as conditional error probabilities, where the condition is that α is fixed. In order to obtain the error probabilities when α is random, we must average $P_2(\gamma_b)$ over the probability density function of γ_b explained by [82]:

$$P_2 = \int_0^\infty P_2(\gamma_b) p(\gamma_b) d(\gamma_b), \tag{4.14}$$

where $p(\gamma_b)$ is the PDF of γ_b when α is random.

Note that in this study we have statistically modeled the distributions of received signal (referred as r_l in Equation (4.11)) which has been discussed in Section 4.4. As shown in both Equations (4.12) and (4.14), we need to compute the probability density function of the SNR values (γ_b) for calculating the conditional bit error probability of QPSK modulation. The PDF of SNR values ($p(\gamma_b)$) can be estimated using the distribution model of the received signals (RSSI values) with parameters explained and listed in Sections 4.4.2.2 and 4.4.3.

From Equation (4.11), the expression for the SNR values γ_b as a function of received signal and noise power is:

$$\gamma_b = \frac{P_\alpha P_s}{P_n} = \frac{P_r}{P_n} - 1, \tag{4.15}$$

where P_s is the transmitted signal power, P_r is the received signal power, and P_n is the noise power. The noise power can be calculated by:

$$P_n = N_0 B, \tag{4.16}$$

where noise spectral density N_0 is the noise power per unit of bandwidth and is given by $N_0 = kT_e$, k is Boltzmann's constant in joules per kelvin, and T_e is the receiver system noise temperature in kelvins. B is the total bandwidth (Hz) over which that noise power is measured. In the 2.4 GHz band there are 16 ZigBee channels, with each channel requiring $B = 5$ MHz of bandwidth.

The noise temperature T_e can be computed from the noise factor of a device F by:

$$F = 1 + \frac{T_e}{T_o}, \tag{4.17}$$

Where T_o is the operating temperature. For the RF230 radio chip, the F is given to be 6 for the $T_o = 22c^\circ$.

Based on Equations (4.16) and (4.17), we can obtain the noise power P_n and substitute Equation (4.16) into Equation (4.15) and calculate the range of SNR values in terms of received power signals. Now, the parameters of the PDF of γ_b can be computed by linear transformation of parameters of the PDF of r_l shown in Equation (4.15). The statistical details corresponding to the distribution model of the received signals are indicated in Tables 4.5 and 4.7 for Indoor office area and corridor, respectively. Table 4.8 shows the computed parameters of probability density function of SNR values γ_b for three different actions of standing, sitting, and walking in two distinct environment.

Table 4.8: The parameters of the probability density function of SNR values γ_b.

Environment	State	distribution	Parameter Estimate		
			Mean (dBm)	Std.Dev. (dBm)	Mixing proportions
Indoor office	Standing	GMM, k=3	35.37	0.76	0.04
			14.05	50.77	0.58
			18.63	8.67	0.37
	Sitting	GMM, k=3	19.48	16.97	0.36
			19.48	46.46	0.36
			20.59	20.91	0.28
	Walking	GMM, k=3	10.12	32.95	0.15
			18.58	43.9	35.03
			20.83	5.6	9.61
Corridor	Standing	GMM, k=2	-3.62	0.2	0.22
			8.01	29.9	0.78
	Sitting	Weibul	4.98	32.88	-
	Walking	GMM, k=2	-3.66	0.17	0.2
			6.1	30.03	0.79

The distribution models fitted to empirical PDF of SNR values for the actions of standing, sitting, and walking in both environments are illustrated in Figure 4.27(a), 4.28(a), and 4.29(a), 4.30(a), 4.31(a), and 4.32(a). The conditional probability of error which are computed for all 6 scenarios are also demonstrated in Figure 4.27(b), 4.28(b), and 4.29(b), 4.30(b), 4.31(b), and 4.32(b).

After extraction of the probability density function of SNR values for all scenarios, we can carry out the integration for $P_2(\gamma_b)$ as given by Equation (4.14). The result of this integration for binary OQPSK modulated signals in all six scenarios are listed in Table 4.9. An acceptable average probability of error against $\overline{\gamma_b}$ is also indicated in Figure 4.33, where $\overline{\gamma_b}$ is the average SNR (signal to noise ratio), defined as:

$$\overline{\gamma_b} = \frac{\varepsilon_b}{N_0} E(\alpha^2), \tag{4.18}$$

Table 4.9: An average Probability of error for binary OQPSK modulated signals in all six scenarios.

Environment	State	Average SNR $\overline{\gamma_b}$	BER
Indoor office	Standing	16.69	0.005
	Sitting	17.39	0.0039
	Walking	17.76	0.0011
Corridor	Standing	5.47	0.082
	Sitting	4.94	0.083
	Walking	4.1	0.09

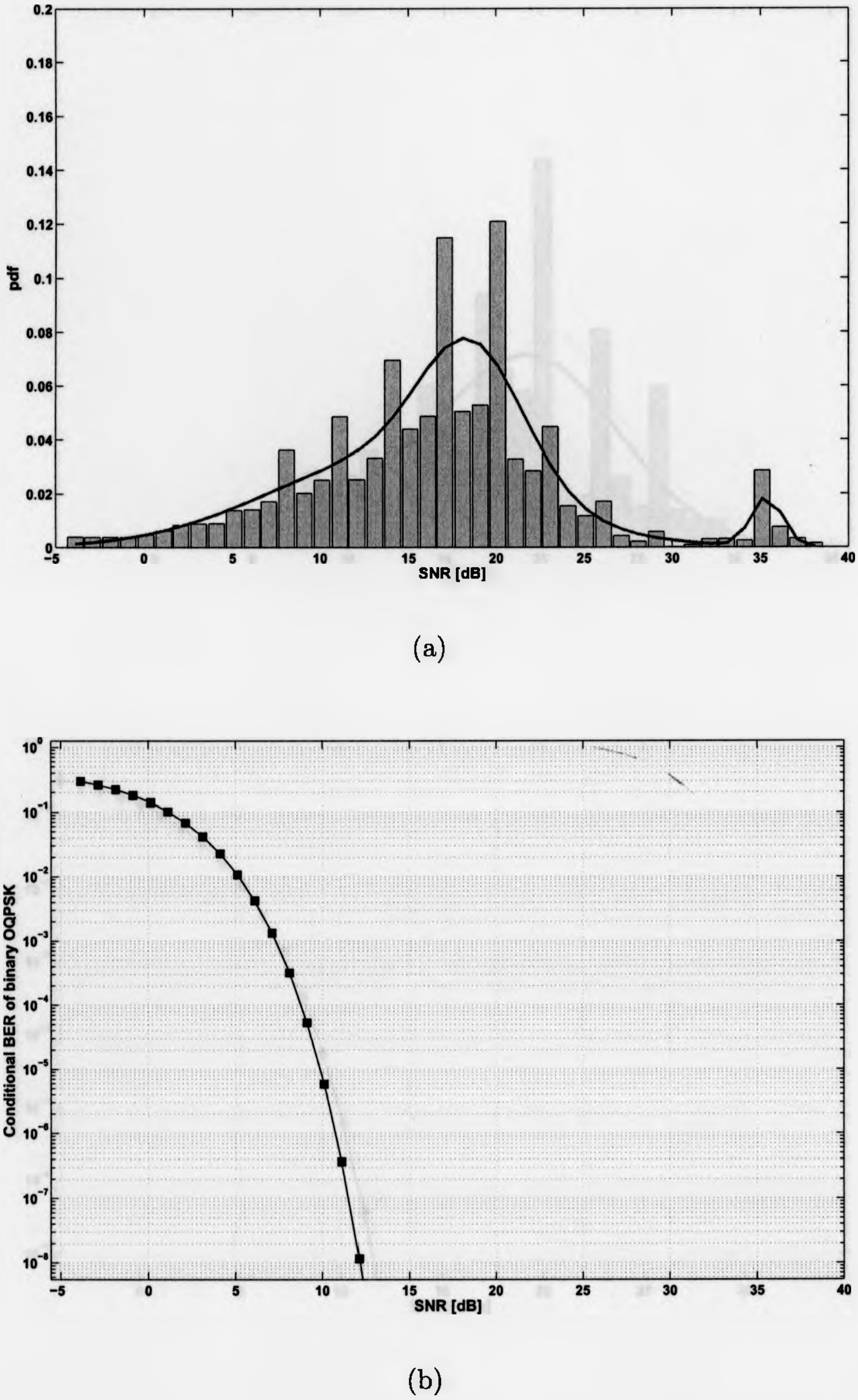


Figure 4.27: (a) The distribution models fitted to empirical pdf of SNR values, (b) the conditional probability of error computed for the action of standing in indoor office area.

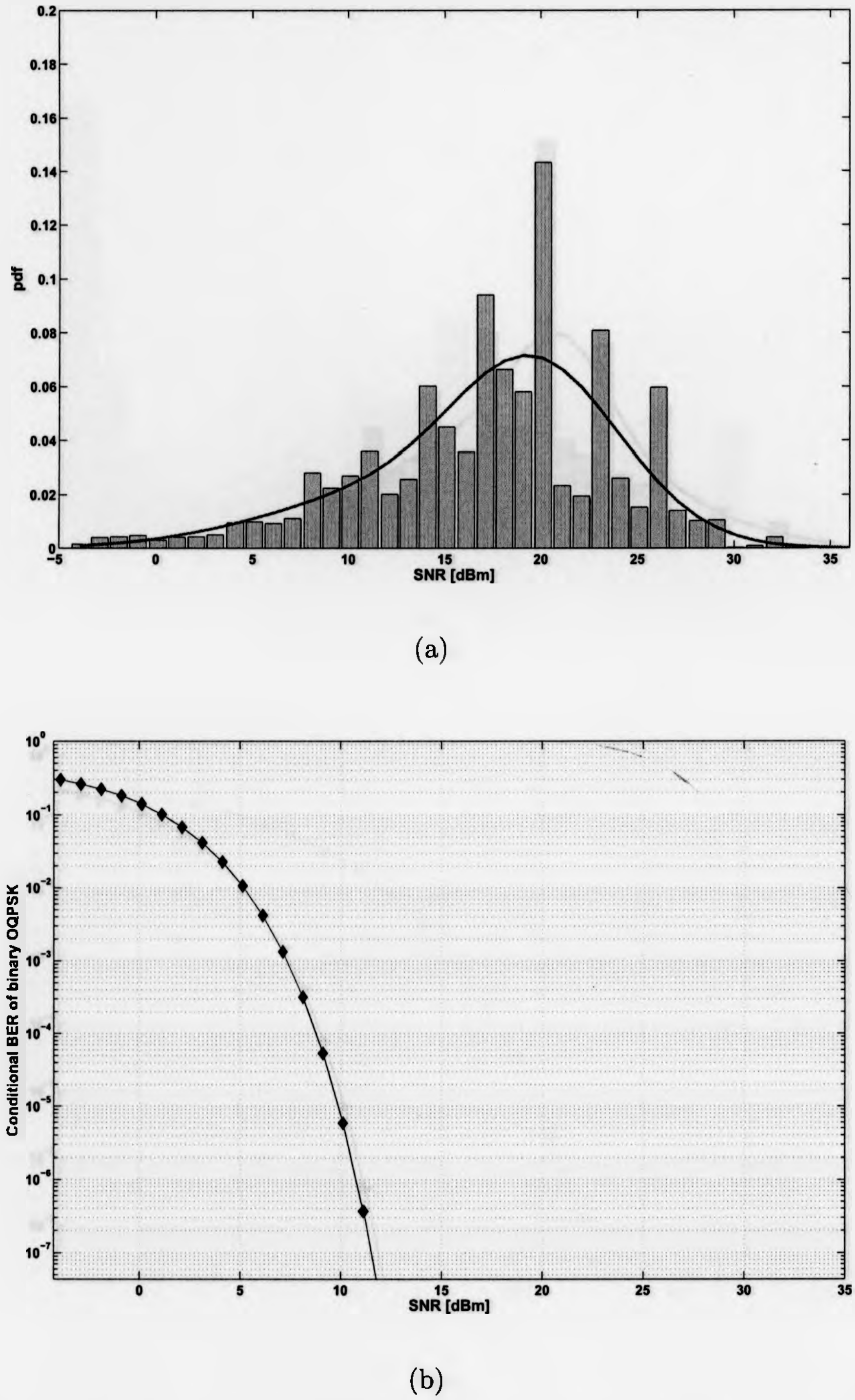
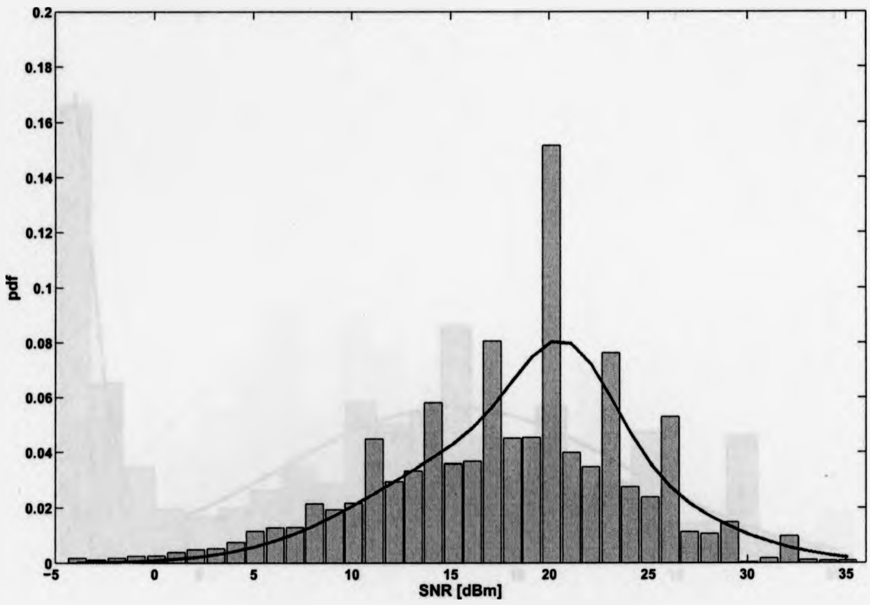
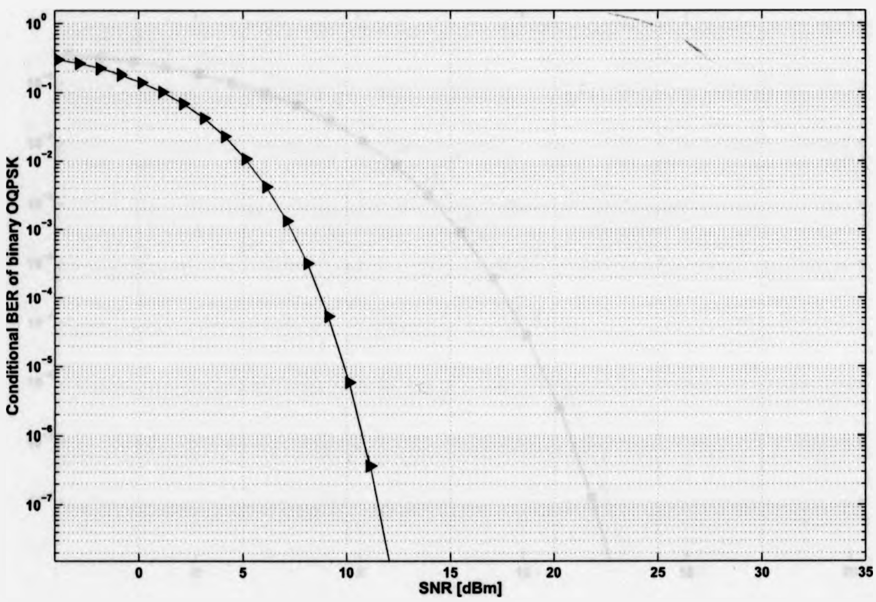


Figure 4.28: (a) The distribution models fitted to empirical pdf of SNR values and (b) the conditional probability of error computed for the action of sitting in indoor office area.

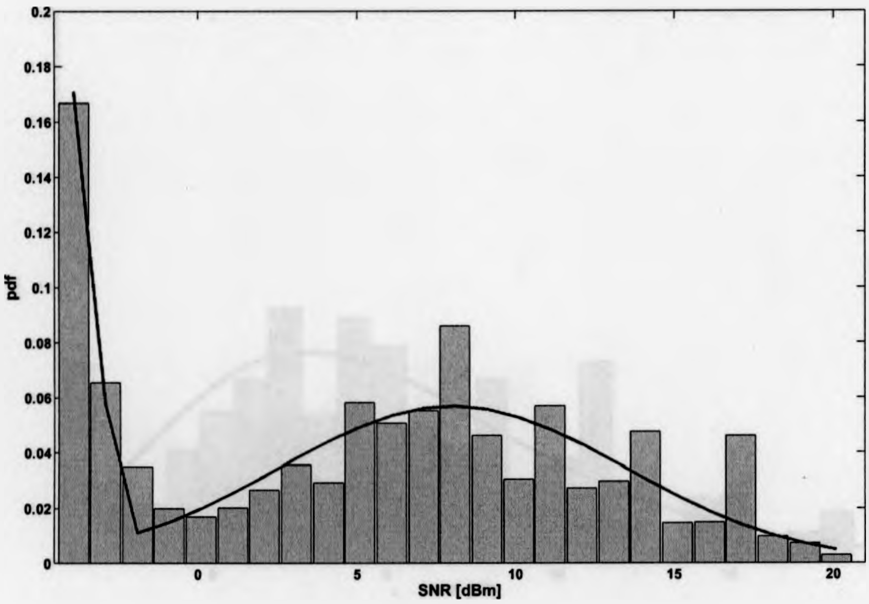


(a)

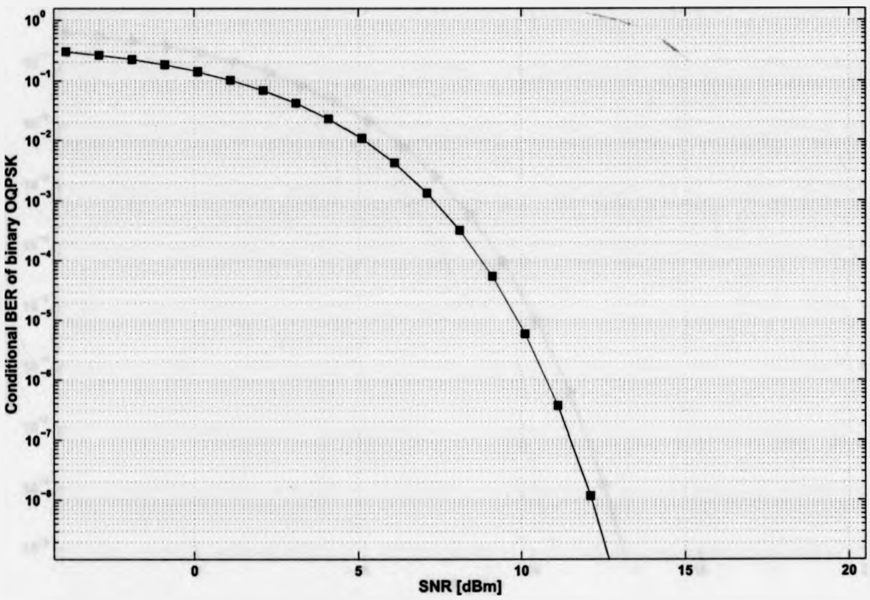


(b)

Figure 4.29: (a) The distribution models fitted to empirical pdf of SNR values and (b) the conditional probability of error computed for the action of walking in corridor area.

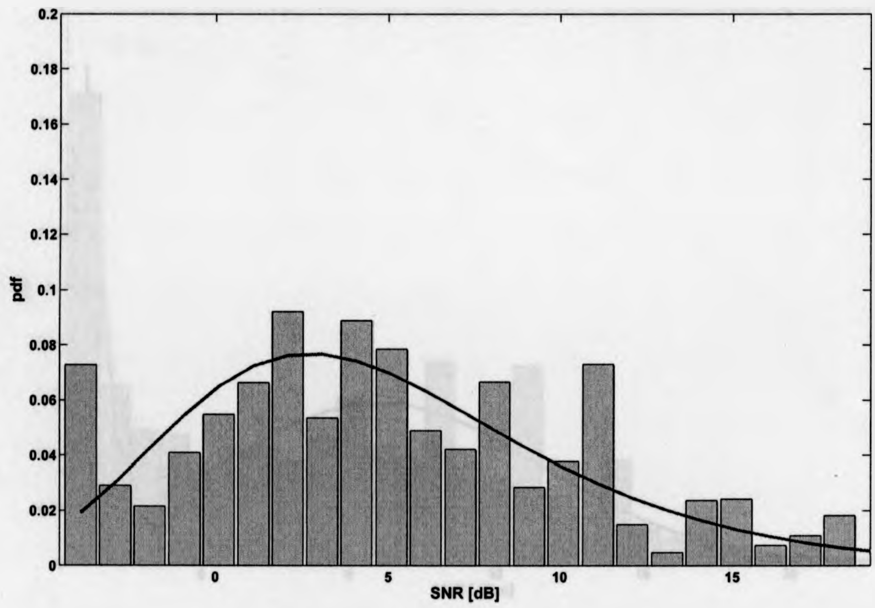


(a)

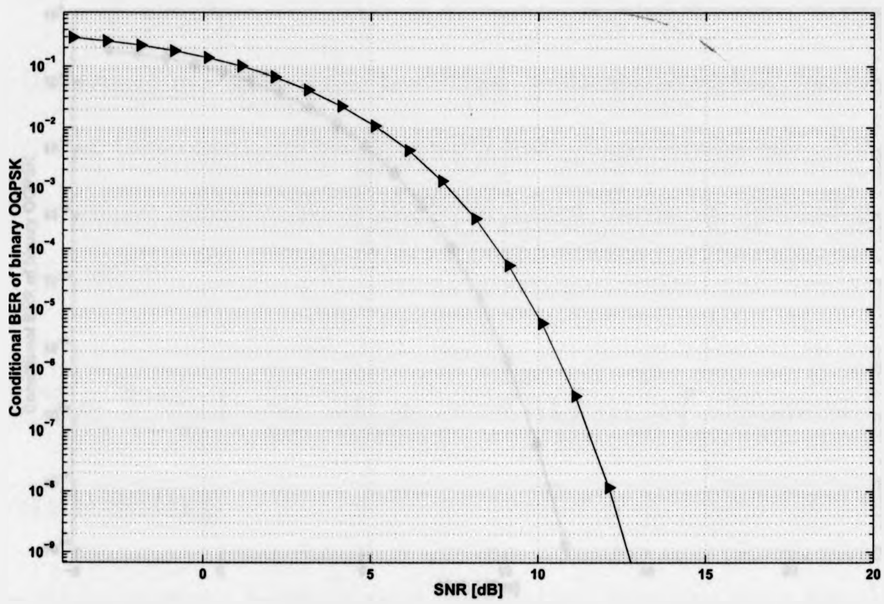


(b)

Figure 4.30: (a) The distribution models fitted to empirical pdf of SNR values and (b) the conditional probability of error computed for the action of standing in corridor area.

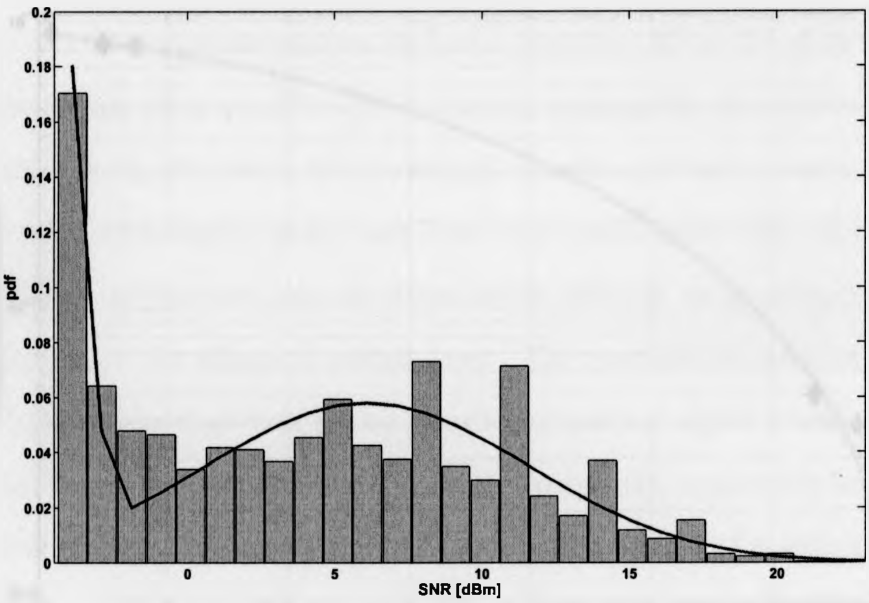


(a)

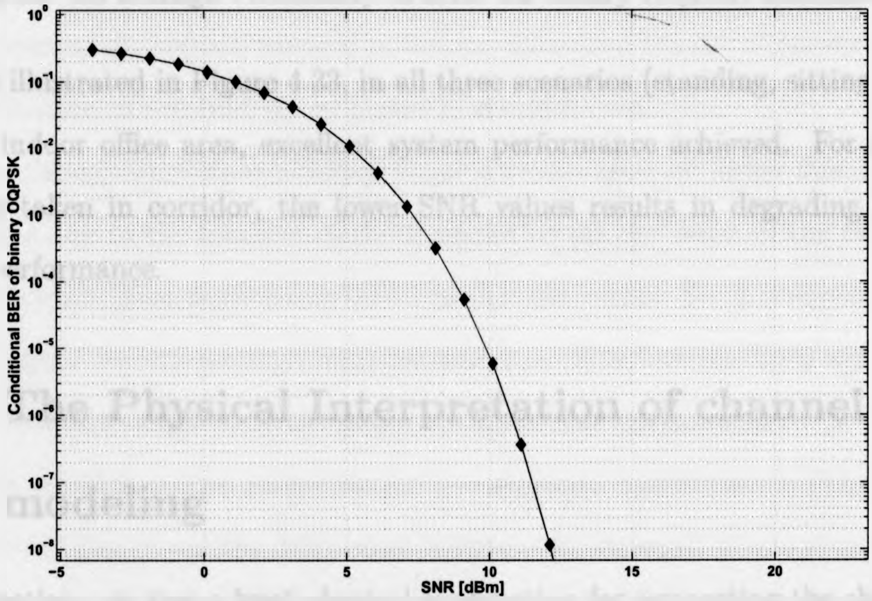


(b)

Figure 4.31: (a) The distribution models fitted to empirical pdf of SNR values and (b) the conditional probability of error computed for the action of sitting in corridor area.



(a)



(b)

Figure 4.32: (a) The distribution models fitted to empirical pdf of SNR values and (b) the conditional probability of error computed for the action of walking in corridor area.

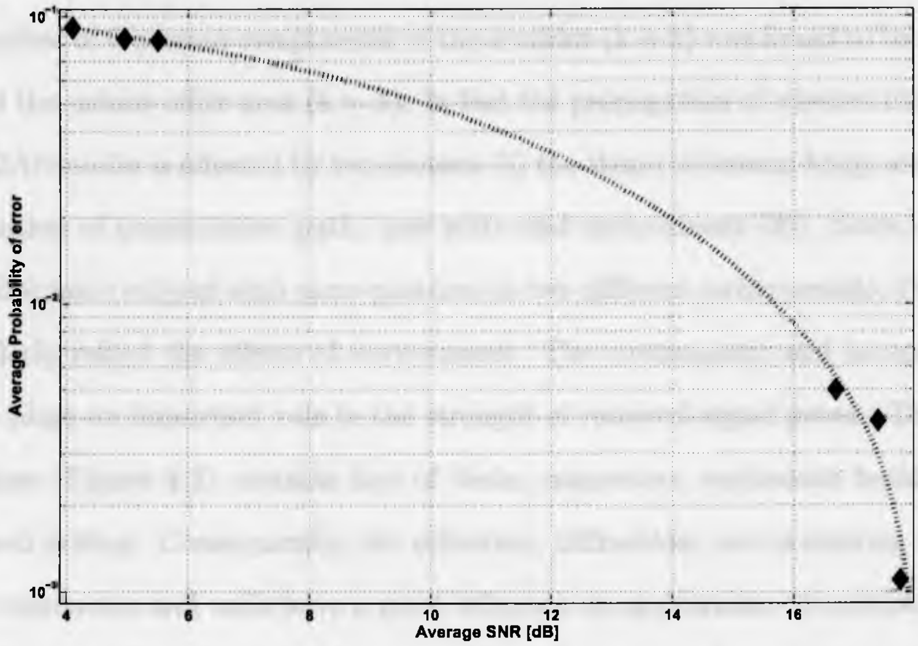


Figure 4.33: An average Probability of error for binary OQPSK modulated signals.

As illustrated in Figure 4.33, in all three scenarios (standing, sitting, and walking) for indoor office area, excellent system performance achieved. For other three scenarios taken in corridor, the lower SNR values results in degrading the overall system performance.

4.6 The Physical Interpretation of channel modeling

In this section, we give a brief physical explanation for supporting the characterized fading statistics and measurement results on link estimation metrics for different body postures presented in previous sections.

The Gaussian Mixture Model was found to account for the majority of user states models in each of the test environments except the sitting scenario in corridor. The number of Gaussian components in the corridor (k = 2) was found to be less than those of the indoor office area (k = 3). In fact the propagation of wireless channel between BAN nodes is affected by two factors; (i) the shape of human body, and (ii) the combination of transmission path, user state and environment [30]. Since, we mentioned the same subject with same postures in two different environments, the results particularly reflect the effects of environment. The environment and its subsequent objects plays an important role in the strength of received signal power. The indoor office area (Figure 4.2) contains lots of desks, computers, cupboards besides walls, floor, and ceiling. Consequently, the reflection, diffraction, and scattering from the office components and walls have a great influence on appearance of multiple clusters in distribution of received signal power. All of these obstacles near the mobile unit cause signal shadowing manifested as an attenuation of the direct wave. Hence, the distributions of the signals received in indoor office area consist of more clusters than the distribution of received signal power in corridor area.

For the experiments conducted in the corridor area, no direct line of sight path is available due to the location of the transmitter (at the waist of the human body) and the receiver (near a computer server in the office area). So, it is expected that the received signal powers undergo more attenuation. This fact has been experimentally proven from the range of RSSI values for both test environments. If we compare the range of RSSI values measured for two separate environments (Figure 4.8 and 4.16), we will figure out the differences between the strength of the received powers. Moreover, based on the pathloss model given in section 4.3.2, the reference RSSI values ($RSSI_0$) in corridor (standing = -71.12 dBm and sitting = -72.23 dBm) are less than those of the indoor office area (standing = -52.06 dBm and sitting = -58.7 dBm) listed in table 4.1. The estimated attenuation coefficient n is about

1.5 for line of sight communication while the user is standing in indoor office area. However, the pathloss exponent n for the sitting state decreased to 0.92. The similar trend is observed when comparing the non-line of sight pathloss exponent of standing ($n = 1.7$) and sitting ($n = 1.5$) scenarios in corridor area. It can be concluded that when the user sits, less significant signal attenuation is observed. This is due to the fact that the reflection is less pronounced in sitting scenario than standing scenario.

Furthermore, as it was illustrated before in Figure 4.3, and 4.4, significant variations are found for the computed packet reception rates (PRR) with respect to RSSI values in corridor. The PRR drop is particularly true as the transmission path is weak and there is no direct line of sight from transmitter (IRIS node) to the base node. More significant variation on the PRR plot versus RSSI were observed when the user starts moving. It is true because when the user walks, he blocks the line of sight between the two antennas. Moreover, the scatter and reflected contributions, and diffraction around the body is expected to be intensified due to changes in geometrical body shape and posture. It is concluded that significant attenuation can occur which also affects the packet recaption rate indicated in Figures 4.3(c) and 4.4(c).

4.7 Conclusion

In this chapter, we focused on the spatial correlations of link estimation metrics and characterization of fading statistics on body area propagation in an indoor environment. The body worn nodes were operating at 2.4 GHz for three user states (standing, sitting, and walking) in two separate environments (indoor office area and corridor area) resulting in 6 scenarios.

In this study, we showed that the RSSI can be a promising indicator rather than LQI when its value is above the sensitivity threshold of RF230 (-91 dBm). Above this threshold, RSSI shows stable behavior and good correlation with PRR. However, no specific correlation has been found between LQI and PRR for all user states. We also derived channel models on pathloss through plotting variation of RSSI with respect to distances between nodes. The pathloss exponent is about 1.5 for line of sight communication while the user is standing in indoor office area. It decreased to 0.92 for the sitting state. For the user states in corridor, the pathloss exponents are about 1.6.

A reliable statistical model was extracted to determine the variation of received signal level (RSSI). Based on this model, it has been concluded that the user states and local environment were responsible for the variation of received signal strength. For all the user states in indoor offi e area and actions of standing and walking in corridor area, the Gaussian Mixture Model provided the best fit. Using GMM, we have shown that the distribution of received signal strength consists of cluster of components diffracting around the body and reflecting from the surrounding objects in the environment. The number of Gaussian components in the corridor (k = 2) was found to be less than those of the indoor office area (k = 3). Weibull distribution suggests a superior fit to the distribution of received power while the user sits in different locations in corridor. Furthermore, performance evaluation for narrowband WBAN employing OQPSK modulation was provided through BER analysis by deploying proposed channel model.

Chapter 5

Conclusion

This thesis deals with the development of prototype system for real-time patient monitoring within the standardization of WBAN. It is going to be a very useful technology with the capability of offering a wide range of benefits to patients, physicians, and society through continuous monitoring and early detection of serious conditions.

A lot of research has been focused on pervasive health or patient monitoring systems. However, majority of previous works focused solely on developing prototype system for WBAN or characterizing the statistical on-body wireless channel. The results of those studies were helpful in pointing out the dark spots and uncertainties in our knowledge of subject and consequently giving important considerations when designing our own experience. However, our study is different from the prior studies presented here as; (a) we have carried out a project which is the combination of developing unique prototype WBAN system and investigating a wide range of issues affecting the wireless channel structure for the communication link between WBAN nodes, (b) we have evaluated the newer set of crossbow motes called IRIS (based on RF230) and we believe that measuring these network characteristics can facilitate us explore how protocols may work in that network, and (c) the characterized wireless communication channel model and measurements between nodes are specific to our particular environment that wearable systems are implemented.

The main contributions of this study are summarized in the followings:

1. This study discusses our effort on developing combined hardware and software platform for the proposed WBAN patient monitoring system. We have developed a system of wearable ECG sensor, wireless sensor network, and a personal server implemented on desktop computer. The personal server interfacing between the WBAN sensor nodes and Wide Area Network is implemented on PC. The PS is responsible for providing multiple functionalities such as real-time visualization, memorizing, analyzing and communication with clients requesting connection over internet. The prototype system takes advantage of (i) maintaining periodic and real-time transmission of ECG signs, (ii) providing a user feel of comfort, (iii) giving high speed access to wireless networks, and (iv) increasing the ECG sampling rates from 200 Hz to 800 Hz.

2. It focuses on signal processing analysis of heart beat calculation and noise removal, implementation, and validation of a patient monitoring system. The key contribution of this thesis is developing MATLAB based system for interaction with sensor network nodes. MATLAB contains advanced numerical computing ability, powerful libraries and elaborate toolboxes for plotting and analyzing data in real-time that eases data processing operations. Moreover, it supports patient mobility using low-cost reliable system.

3. It investigates how the human body affects wireless link communication by attaching IRIS sensor nodes onto the waist of the human body. Despite wire connection, wireless connections are unstable and vulnerable to environments. We believe that the measurements carried out in this part of the thesis support predicting the impact of realistic channel on network level performance. In this regard, the wireless channel structure for the communication link between the sensor node placed on human body surface and the external receiver node

placed near the computer server has been characterized.

- Firstly, we identified spatial correlation of link estimation metrics (RSSI, LQI, and PRR) as three characteristics of a network. We believe that observing network characteristics is essential in understanding how protocols may work in that network. Moreover, they can be employed to predict link level performance and development of more effective antennas. For instance, in this study, we figured out that the RSSI can be a promising indicator rather than LQI when its value is above the sensitivity threshold of RF230 (-91 dBm). Above this threshold, RSSI is correlated with PRR and the distance between nodes. As a result, protocol designers should better choose RSSI over LQI as inexpensive link estimators.

- Secondly, we characterized the fading statistics on body links by modeling the distribution of the received signal power. The Akaike information criterion (AIC) has been employed to choose the best model among the set of candidates. Our modeling is clearly different from the previous work which tried to use well known probability density functions for modeling. In this study, the Gaussian Mixture Model is found to account for the majority of user states models in each of the two test environments and the number of Gaussian components is describing the cluster of components diffracting around the body and reflecting off of the surrounding objects in the environment.

- Finally, the proposed channel model has been applied in evaluating the BER performance of the narrowband-based WBAN system. We expect these experimental results can be deployed to assist developing reliable and efficient wireless connections between on-body sensor nodes for future biomedical researches.

We believe that the work carried out as part of this thesis can be the basis of considerable future research. This study presented a telecardiology system in which an ECG sensor is integrated into the wireless platform. However, we believe that many other different body sensors such as EEG, blood pressure, and etc can be potentially integrated to our developed wireless platform or other sensor network platforms such as Telos, MicaZ, and Mica2. The system can become more sophisticated and comprehensive by utilizing several on body sensor networks. As this application scenario would be different from our particular system prototype, challenges such as network configuration and power management need to be revisited. Based on the new set up, the sensor nodes should be able to join or leave the network at any time. Dynamic management of the resources such as sensor functionalities and communication bandwidth should be also considered.

Security and privacy are two open resources issues which should be addressed before WBAN technologies become widely applied. Different levels of security should be recognized, and appropriate encryption mechanism should be improved to identify security attacks. Following security requirements such as data confidentiality, data authenticity, and access control should be assured before acceptance of WBAN. Although providing adequate security is a crucial factor in the widely approval of WBAN, little research has been carried out to follow security requirements. Privacy requires effective authentication techniques such as human faces, hand features, or EEG signals. Using privacy, the users can have control over their information in such a way that they can decide which information to be transferred. Note that security and privacy protection mechanisms make use of noticeable energy. Other WBAN devices also require energy for data collection, analyzing, and transferring. So, another possibility for future work is to develop algorithms in order to decrease radio transmissions and enhance energy saving capabilities. It is a fact that the sensor nodes are designed to be used for a long period of time and the power sources should be efficient enough and long lasting.

References

[1] S. Borkar, "Design challenges of technology scaling," *IEEE Micro*, vol. 19, no. 4, 1999. 1

[2] M. van de Goor, "Indoor localization in wireless sensor networks," Master's thesis, 2009. 1

[3] H. Alemdar and C. Ersoy, "Wireless sensor networks for healthcare: A survey," *Computer Networks*, vol. 54, no. 15, Oct 2010. 1, 5, 8

[4] S. Kumar, K. Kambhatla, F. Hu, M. Lifson, and Y. Xiao, "Ubiquitous Computing for Remote Cardiac Patient Monitoring: A Survey," *International Journal of Telemedicine and Applications*, 2008. 1, 8

[5] I. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci. 1, 10

[6] Cardiovascular disease statistics. [Online]. Available: http://www.americanheart.org/presenter.jhtml?identifier=4478 2

[7] (2001, February) Cardiovascular epidemiology in the asia-pacific region. 2

[8] Eurika. [Online]. Available: http://finance.yahoo.com/news/Study-Shows-Cardiovascular-prnews-2927369358.html 2

[9] F. Vergari, V. Auteri, C. Corsi, and C. Lamberti, "A ZigBee−based ECG transmission for a low cost solution in home care services delivery," Dipartimento di Elettronica, Universita' di Bologna,Viale Risorgimento, Tech. Rep. 2, 23

[10] J. Anderson and S. E. DiCarlo, "Virtual experiment for understanding the electrocardiogram and the mean electrical axis," *Advan Physiol Educ*, vol. 23, pp. 1–17, 2000. 3

[11] T. Fulford-Jones, G.-Y. Wei, and M. Welsh, "A portable, low-power, wireless two-lead ekg system," in *26th IEEE EMBS Annual International Conference*, September 2004. 3, 16

[12] J. Proulx, R. Clifford, S. Sorensen, D. J. Lee, and J. Archibald, "Development and evaluation of a bluetooth EKG monitoring sensor," *Proceedings of the 19th IEEE Symposium on Computer-Based Medical Systems (CBMS'06)*, pp. 2141–2144, June 2006. 3

[13] J. W. Zheng, Z. B. Zhang, T. H. Wu, and Y. Zhang, "A wearable mobihealth care system supporting real-time diagnosis and alarm," *Medical and Biological Engineering and Computing*, vol. 45, pp. 877–885, July 2007. 4, 16

[14] Mobile remote monitoring project. MobiHealth. [Online]. Available: http://www.mobihealth.com/home/en/home.php 4

[15] C. Orwat, A. Graefe, and T. Faulwasser, "Towards pervasive computing in health care a literature review," *BMC Medical Informatics and Decision Making*, vol. 8, p. 119, June 2008. 4

[16] Schiller, the art of diagnostics. [Online]. Available: http://www.schiller.ch/index.php?id=1 4

[17] F. Gouaux, L. Simon-Chautemps, J. Fayn, and et al., "Ambient intelligence and pervasive systems for the monitoring of citizens at cardiac risk: New solutions from the epi-medics project," in *Proceedings of the Annual Conference on Computers in Cardiology*, vol. 29, Sep 2002, p. 289292. 4

[18] P. Rubel, F. Gouaux, J. Fayn, and et al., "Towards intelligent and mobile systems for early detection and interpretation of cardiological syndromes," in *Proceedings of the Annual Conference on Computers in Cardiology*, vol. 28, Jan 2001, pp. 193 – 196. 4

[19] Z. Li and G. Zhang, "A physical activities healthcare system based on wireless sensing technology," in *13th IEEE International Conference on Embedded and Real-Time Computing Systems and Applications*, 2007. 5

[20] A. Fort, C. Desset, J. Ryckaert, P. D. Doncker, L. V. Biesen, and P. Wambacq, "Characterization of the ultra wideband body area propagation channel," in *IEEE Conference on Ultra-Wideband*, September 2005. 5, 42

[21] ——, "Ultra wide-band body area channel model," in *IEEE Conference on Communications*, May 2005.

[22] A. Fort, J. Ryckaert, C. Desset, P. D. Doncker, P. Wambacq, and L. V. Biesen, "Ultra-wideband channel model for communication around the human body," *IEEE Journal on Selected Areas in Communications*, vol. 24, pp. 927–933, April 2006. 42

[23] A. Fort, C. Desset, J. Ryckaert, P. D. Doncker, L. V. Biesen, and S. Donnay, "An ultra-wideband body area propagation channel model-from statistics to implementation," *IEEE Journal on Microwave theory, techniques*, vol. 54, pp. 1820–1826, June 2006. 42, 66, 67

[24] S. Drude, "Indoor body-area channel model for narrowband communications," in *IET Microwaves, Antennas and Propagation*, vol. 1, December 2007, pp. 1197–1203. 42, 43

[25] K. Takizawa, T. Aoyagi, , and R. Kohno, "Channel modeling and performance evaluation of UWB-based wireless body area networks," June 2009.

[26] M. Kim and J. ichi Takada, "Statistical model for 4.5-GHz narrowband on-body propagation channel with specific actions," *IEEE Antennas Wireless Propag. Lett.*, vol. 8, pp. 1250–1254, 2009. 43, 65, 68

[27] A. Alomainy, Y. Hao, A. Owadally, C. Parini, Y. Nechayev, C. Constantinou, and P. Hall, "Statistical analysis and performance evaluation for on-body radio propagation with microstrip patch antennas," *IEEE Trans. Antennas Propag.*, vol. 55, pp. 245–248, January 2007. 5, 42

[28] D. Smith, L. Hanlen, D. Miniutti, J. Zhang, D. Rodda, and B. Gilbert, "Statistical characterization of the dynamic narrowband body area channel," in *Proc. ISABEL 2008*, Aalborg, 25-28, October 2008. 5, 42

[29] W. G. Scanlon and S. L. Cotton, "Understanding on-body fading channels at 2.45 GHz using measurements based on user state and environment," in *Antennas and Propagation Conference*, Loughborough, 17-18 March 2008.

[30] S. L. Cotton and W. G. Scanlon, "An experimental investigation into the influence of user state and environment on fading characteristics in wireless body area networks at 2.45 ghz," *IEEE Trans. Wireless Commun.*, vol. 8, pp. 6–12, Jan 2009. 110

[31] E. Reusens, W. Joseph, B. Latre, B. Braem, G. Vermeeren, E. Tanghe, L. Martens, I. Moerman, and C. Blondia, "Characterization of on-body communication channel and energy efficient topology design for wireless body area networks," *IEEE Trans. Inf. Technol. Biomed.*, vol. 13, pp. 933–945, Nov 2009. 5, 43

[32] R. C. Shah and M. Yarvis, "Characteristics of on-body 802.15.4 networks," in *2nd IEEE Workshop on Wireless Mesh Networks*, Sep 2006. 5

[33] K. Papagiannaki, M. Yarvis, and W. S. Conner, "Experimental characterization of home wireless networks and design implications," in *Proc. IEEE Infocom*, April 2006. 44

[34] D. Jea and M. B. Srivastava, "Channels characteristics for on-body mica2dot wireless sensor networks," in *Proc. Mobiquitous*, 2005. 5, 44

[35] H. Cao, V. Leung, C. Chow, and H. Chan, "Enabling technologies for wireless body area networks: A survey and outlook," *IEEE Commun. Mag.*, vol. 47, pp. 84–93, Dec 2009. 8

[36] Bedroussian, Armen, DeVol, and Ross. (2007, October) An Unhealthy America: The economic burden of chronic disease charting a new course to save lives and increase productivity and economic growth. Milken Institute. 9

[37] (2001, June) America's most ignored health problem: Caring for the chronically ill. Washington: Alliance for Health Reform. 9

[38] (2009, February) Chronic disease prevention and promotion. Center for Disease Control. [Online]. Available: http://www.cdc.gov/nccdphp/ 9

[39] Tinyos. [Online]. Available: http://docs.tinyos.net 10, 23

[40] P. Levis and D. Gay, *TinyOS Programming*. Camebridge University press, 2009. 11

[41] *IRIS*, Crossbow Technology. [Online]. Available: http://www.xbow.com 12, 13, 19, 20, 46

[42] *8-bit AVR Microcontroller*, ATMEL. [Online]. Available: http://www.atmel.com/dyn/resources 12, 19, 45

[43] *AVR Low Power 2.4 GHz Transceiver for ZigBee, IEEE 802.15.4, 6LoWPAN, RF4CE and ISM Applications*, ATMEL. [Online]. Available: http://www.atmel.com/dyn/resources 12, 20, 45, 57

[44] R. Fensli, E. Gunnarson, and O. Hejlesen, "A wireless ECG system for continuous event recording and communication to a clinical alarm station," in *Proceedings of the 26th Annual International Conference of the IEEE EMBS*, Sep 2004. 15

[45] R. Fensli, E. Gunnarson, and T. Gundersen, "A wearable ECG-recording system for continuous arrhythmia monitoring in a wireless tele-home-care situation," in *18th IEEE Symposium on Computer-Based Medical Systems*, June 2005. 15

[46] D. Konstantas, V. Jones, and R. Bults, "Mobihealth innovative 2.5/3G mobile services and applications for healthcare," *IST Mobile and Wireless Telecommunications Summit*, 2002. 15

[47] A. V. Halteren, R. Bults, K. Wac, N. Dokovsky, G. Koprinkov, I. Widya, D. Konstantas, and V. Jones, "Wireless body area networks for healthcare : the mobihealth project," *Wearable eHealth Systems for Personalised Health Management: State of the Art and Future Challenges*, vol. 108, pp. 181–193, 2004. 16

[48] V. Shnayder, B. Chen, K. Lorincz, T. R. F. FulfordJones, and M. Welsh, "Sensor networks for medical care," *Division of Engineering and Applied Science, Harvard University, Tech. Rep.*, Jan 2005. 16, 19

[49] D. Malan, T. Fulford-Jones, M. Welsh, and S. Moulton, "CodeBlue: An Ad Hoc sensor network infrastructure for emergency medical care," in *MobiSys 2004 Workshop on Applications of Mobile Embedded Systems*, June 2004. 16

[50] K. Lorincz and M. Welsh, "Motetrack: A robust, decentralized approach to rf-based location tracking," in *International Workshop on Location- and Context-Awareness (LoCA 2005)*, May 2005. 16

[51] K. Lorincz, D. Malan, T. Fulford-Jones, A. Nawoj, A. Clavel, V. Shnayder, G. Mainland, S. Moulton, and M. Welsh, "Sensor networks for emergency response: Challenges and opportunities," in *IEEE Pervasive Computing*, Oct/Dec 2004. 16

[52] E. Jovanov, A. Milenkovic, C. Sanders, C. Otto, and P. C. de Groen, "A wireless body area network of intelligent motion sensors for computer assisted physical rehabilitation," *Journal of NeuroEngineering and Rehabilitation*, vol. 2, pp. 16–23, March 2005. 16

[53] C. Otto, A. Milenkovic, C. Sanders, and E. Jovanov, "System architecture of a wireless body area sensor network for ubiquitous health monitoring," *Journal of Mobile Multimedia*, vol. 1, pp. 307–326, Oct 2006. 16

[54] O. Chipara, C. Lu, T. C. Bailey, and G.-C. Roman, "Reliable clinical monitoring using wireless sensor networks: experiences in a step-down hospital unit," in *SenSys '10 Proceedings of the 8th ACM Conference on Embedded Networked Sensor Systems*, Nov 2010, pp. 155–168. 16

[55] H. Yan, H. Huo, Y. Xu, and M. Gidlund, "Wireless sensor network based e-health system - implementation and experimental results," *IEEE Trans. Consum. Electron.*, vol. 56, no. 4, pp. 2288– 2295, Nov 2010. 16

[56] *Vernier Software and Technology.* [Online]. Available: http://www2.vernier.com/booklets/ekg-bta.pdf 19

[57] J. P. Shah, P. Aroul, A. Hande, and D. Bhatia, "Remote Cardiac Activity Monitoring Using Multi-Hop Wireless Sensor Networks," in *Proceedings of the 19th IEEE Symposium on Computer-Based Medical Systems (CBMS 06)*, Nov 2007, pp. 142–145. 19

[58] S. Xiao, A. Dhamdhere, V. Sivaraman, and A. Burdett, "Transmission Power Control in Body Area Sensor Networks for Healthcare Monitoring," *IEEE J. Sel. Areas Commun.*, vol. 27, pp. 37–48, Jan 2009. 19

[59] V. Sivaramany, S. Grovery, A. Kurusingaly, A. Dhamdherey, D. Ostryz, and A. Burdett, "Mobility in the soccer field : from empirical data collection to modelling correlated connectivity," *IEEE Trans. Inf. Technol. Biomed.*, vol. 14, pp. 726–733, May 2010. 19

[60] *MTS/MDA, sensor, data acquisition boards*, Crossbow Technology. [Online]. Available: http://www.xbow.com/Products 20

[61] V. Handziski, J. Polastre, J. H. Hauer, C. Sharp, A. Wolisz, D. Culler, and D. Gay, "Hardware abstraction architecture," TinyOS Documentation Wiki, TinyOS Enhancement Proposals, Tech. Rep. [Online]. Available: http://www.tinyos.net/tinyos-2.1.0/doc/html/tep2.html 23

[62] E. Dolatabadi and S. Primak, "Application note: Wireless sensor network for health monitoring," CMC Microsystems, Tech. Rep. [Online]. Available: http://www.cmc.ca/ 25

[63] J. C. Huhta and J. G. Webster, "60Hz interference in electrocardiography," *IEEE Trans. Biomed. Circuits Syst.*, vol. 20, 2007. 29

[64] M. Ferdjallah and R. E. Barr, "Adaptive digital notch filter design on the unit circle for the removal of powerline noise from biomedical signals," *IEEE Trans. Biomed. Circuits Syst.*, vol. 41, June 1994. 29

[65] J. M. Kortelainen and J. Virkkala, "FFT averaging of multichannel BCG signals from bed mattress sensor to improve estimation of heart beat interval," in *29th Annual International Conference of the IEEE EMBS*, Lyon, Aug 2007. 31

[66] S. S. Joshi and C. V. Ghule, "DWT based beat rate detection in ECG analysis," in *International Conference and Workshop on Emerging Trends in Technology (ICWET 2010)*, TCET, Mumbai, India, Feb. 2010, pp. 765–769. 31

[67] *Wavelet Toolbox*, Mathworks. [Online]. Available: http://www.mathworks.com/help/toolbox/wavelet/gs/bsjspmn.html 31

[68] M. A. Khayer and M. A. Haque, "ECG peak detection using wavelet transform," in *3rd International Conference on Electrical and Computer Engineering (ICECE 2004)*, Dhaka, Bangladesh, Dec 2004. 31

[69] *BlackBerry Java Plug-in for Eclipse.* [Online]. Available: http://us.blackberry.com/developers/javaappdev/javaplugin.jsp 39

[70] *BlackBerry Smartphone Simulators.* [Online]. Available: http://us.blackberry.com/developers/resources/simulators.jsp 39

[71] K. Y. Yazdandoost, H. Sawada, S. T. Choi, J. ichi Takada, and R. Kohno, "Channel characterization for ban communications," IEEE

P802.15 Working Group for Wireless Personal Area Networks(WPANs), Tech. Rep. [Online]. Available: http://www.ap.ide.titech.ac.jp/publications/Archive/IEEE802-15-07-0641-00-0ban280703Kamya29.pdf 41

[72] S. Primak, V. Kontorovich, and V. Lyandres, *Stochastic Methods and Their Applications to Communications: Stochastic Differential Equations Approach.* John Wiley and Sons, 2005. 41

[73] K. A. zge, "Channel modeling approaches to wireless system design and analysis," Ph.D. dissertation, New Brunswick, Rutgers, The State University of New Jersey, New Brunswick, New Jersey, October 2010. 42

[74] W. G. Scanlon and S. L. Cotton, "A statistical analysis of indoor multipath fading for a narrowband wireless body area network," in *IEEE 17th International Symposium on Personal, Indoor and Mobile Radio Communications*, Helsinki, 11-14 Sept 2006. 42

[75] J. Zhao and R. Govindan, "Understanding packet delivery performance in dense wireless sensor networks," in *Proceedings of the First ACM Conference on Embedded Network Sensor Systems*, 2003. 43

[76] J. Polastre, R. Szewczyk, and D. E. Culler, "Telos: enabling ultra-lowpowerwireless research," in *IPSN*, 2005. 44

[77] K. Srinivasan and P. Levis, "RSSI is under appreciated," in *Proceedings of the Third Workshop on Embedded Networked Sensors*, 2006. 44

[78] K. Srinivasan, P. Dutta, A. Tavakoli, and P. Levis, "An empirical study of low-power wireless," *ACM Transactions on Sensor Networks*, vol. 6, Feb 2010. 44, 57

[79] K. P. Burnham and D. R. Anderson, *Model Selection and Multimodel Inference: A Practical Information-Theoretic Approach.* New York: Springer-Verlag, 2002. 66

[80] D. Reynolds, "Gaussian mixture models," MIT Lincoln Laboratory, USA, Tech. Rep. [Online]. Available: dar@ll.mit.edu 74

[81] *Expectation-maximization algorithm.* [Online]. Available: http://en.wikipedia.org/wiki/Expectation-maximization_algorithm 75

[82] J. Proakis, *Digital Communications.* McGraw-Hill Higher Education, 2001. 99

[83] *Serial Port I/O*, Mathworks. [Online]. Available: http://www.mathworks.com/help/techdoc/matlab_external/f38496.html 137

[84] *serial*, Mathworks. [Online]. Available: http://www.mathworks.com/help/techdoc/ref/serial.html 138

[85] *fread*, Mathworks. [Online]. Available: http://www.mathworks.com/help/techdoc/ref/serial.fread.html 139

[86] *fopen*, Mathworks. [Online]. Available: http://www.mathworks.com/help/techdoc/ref/serial.fopen.html 139

[87] *fclose*, Mathworks. [Online]. Available: http://www.mathworks.com/help/techdoc/ref/serial.fclose.html 139

[88] *Events and Callbacks*, Mathworks. [Online]. Available: http://www.mathworks.com/help/techdoc/matlab-external/f73779.html 140

[89] *function-handle*, Mathworks. [Online]. Available: http://www.mathworks.com/help/techdoc/ref/function-handle.html 141

Appendix A
TinyOS programming for developing a Health Monitoring System

In February 2008, Crossbow announced the availability of the TinyOS 2.x Operating System for Crossbow's advanced IRIS Motes. IRIS Motes are also supported by Crossbow's MoteWorks software development environment based on open-source TinyOS 1.x. We preferred to run our application in the TinyOS 2.x Operating System because it provides a better hardware abstraction model, improved timers, sensor interfaces, power management, arbitration, and much more.

A.1 Creating the Sensor Board Driver

The capabilities of a physical sensor are made available to a TinyOS application through a sensor driver. TinyOS device drivers provide hardware-independent interfaces (HIL) and hardwaredependent interfaces (HAL). Custom sensor drivers must also follow HIL/HAL architecture for consistency. In TinyOS, each sensor should be characterized as an individual component. For a sensor board that contains different sensors, the collection of components should be represented. That is, a sensor board is a set of sensor components each with a predetermined name, intended for connection to several TinyOS platforms. These are the steps needed to create our ECG Sensor driver components:

- Each sensor board must have its own directory named < *sensorboard* >. Default TinyOS 2.x sensor boards are placed in "*tos/sensorboards/*< *sensorboard* >", but sensor board directories can be placed anywhere as long as the nesC compiler receives a *-I* directive pointing to the sensor board's directory. We placed our Vernier EKG sensor board in the default *TinyOS 2.x* sensor board directory (see Figure A.1).

- Each sensor board directory must have *a .sensor* file. This file is a script which is executed as part of the *ncc nesC* compiler frontend. It can add or modify any compile-time options necessary for a particular sensor board.

- Because our ECG sensor is analog, we need to create only two components: one HIL component to present the sensor itself (*ECGC.nc*) and one HAL component to select the suitable hardware resources (*ECGdeviceC.nc*).

  1. The sensor HIL component should contain one or more Source and Sink Independent Drivers (SID) interfaces for reading data (Figure A.2). SID presents two interfaces for reading data provided by the sensor board: *Read* and *ReadStream*. When a client requests data through the *Read* or *ReadStream* interface, the HIL will request access to the HAL using the Resource interface.

  2. The *ECGsensorC.nc* component should be a generic component that virtualizes access to the sensor. As illustrated in Figure A.3 , this generic component is linked to two components: *AdcReadClientC* and *ECGdeviceC*. The *AdcReadStreamClientC* provides arbitrated access via *ReadStream* interface, respectively, to the Atmega128 ADC. The *ECGdeviceC*, shown in Figure A.4, is a sensor's HAL component and provides implementation of *Atm128AdcConfig* and the ADC parameters such as channel. As explained before, the HIL will ask for access to the HAL and after being accepted,

it will pull the client's ADC configuration using the *Adcconfigure* interface and convert the client's *ReadStream* command to a chip-dependent HAL command. Once the HAL signals the conversion result to the HIL, HIL will release the ADC through the Resource interface and signal the conversion result to the client through the *ReadStream* interface [11].

3. The actual implementation of the ECG sensor's HAL component (*ECGdeviceC*) is in the *ECGdevicep* module to provide power control and the ADC configuration. The *MicaBusC* component returns the ADC channel number for the ADC pins (see Figure A.5). For this example, we chose the 5th channel of the ADC. The *ECGdeviceP* component selects the appropriate hardware resources, such as ADC port 5, reference voltage 2.56, and ADC prescaler settings (see Figure A.6).

   Note: The header file *Atm128adc.h*, in *tos/chips/atm128/adc/*, contains all configuration settings of the 8 channel 10-bit *Atm128ADC*. *Atm128AdcC*, *Atm128AdcMultiple*, and *Atm128AdcSingle* are example HAL components that implement ADC conversion.

A.2 How to Program the Mobile Mote and the Base Station

The way motes are programmed depends on their functions. Two types of motes are programmed in this application: Mobile and Base.

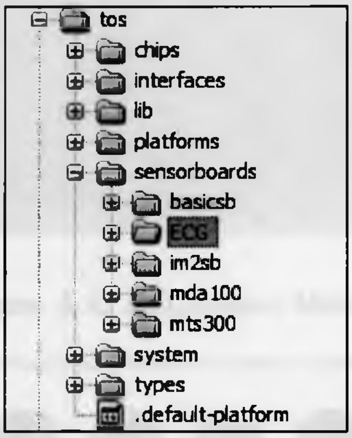


Figure A.1: TinyOS 2.x Sensor Board Directory.

```
interface ReadStream<val_t> {
  command error_t postBuffer(val_t* buf, uint16_t count);
  command error_t read(uint32_t usPeriod);

  event void bufferDone(error_t result,
          val_t* buf, uint16_t count);
  event void readDone(error_t result, uint32_t usActualPeriod);
}
```

Figure A.2: *ReadStream* Interface.

```
#include "ECG.h"

generic configuration ECGsensorC() {
  provides interface Read<uint16_t>;
}
implementation {
  components new AdcReadClientC(), ECGdeviceC;

  Read = AdcReadClientC;
  AdcReadClientC.Atm128AdcConfig -> ECGdeviceC;
}
```

Figure A.3: *ECGsensorC.nc* component.

A.2.1 Mobile Mote, *Oscilloscope* Application

The Mobile IRIS mote is connected to the ECG sensor and runs an application which uses the ADC to sample the analog ECG data from the sensor, constructs a message out of them, and sends it over the radio to the IRIS mote connected to the base station. The ADC channel continuously reads the ECG data and when

```
#include "ECG.h"

configuration ECGdevicec{
  provides {
    interface Atm128AdcConfig;
  }
}
implementation {
  components ECGdevicep, MicaBusC;

  Atm128AdcConfig = ECGdevicep;
  ECGdevicep.ECGAdc -> MicaBusC.Adc5;
}
```

Figure A.4: *ECGdevicec* Module.

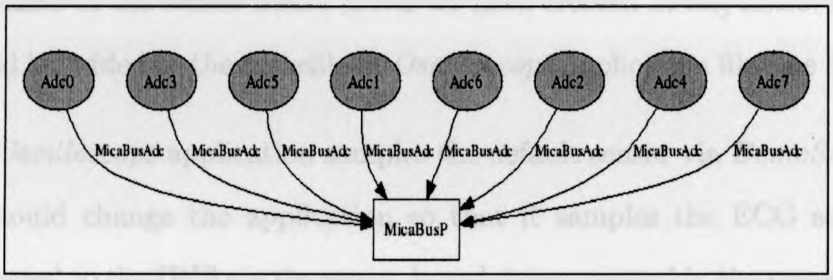


Figure A.5: MicaBusC Component and its Adc Interfaces.

```
module ECGdevicep
{
  provides {
    interface Atm128AdcConfig;
  }
  uses {
    interface MicaBusAdc as ECGAdc;
  }
}
implementation
{
  async command uint8_t Atm128AdcConfig.getChannel() {
    return call ECGAdc.getChannel();
  }

  async command uint8_t Atm128AdcConfig.getRefVoltage() {
    return ATM128_ADC_VREF_OFF;
  }

  async command uint8_t Atm128AdcConfig.getPrescaler() {
    return ATM128_ADC_PRESCALE;
  }
}
```

Figure A.6: Component *ECGdeviceP*.

enough ECG samples are collected in the message buffer, the application passes the message to the networking stack. All details about radio communication and packet protocols in TinyOS are explained in [12]. *Oscilloscope* is an application available as part of TinyOS that lets us monitor sensor readings on the PC. By making minor modifications on the *Oscilloscope* application, we can visualize our Medical Sensor

```
COMPONENT=OscilloscopeAppC
SENSORBOARD = ECG
include $(MAKERULES)
```

Figure A.7: *Oscilloscope* Application Makefile.

reading. The *Oscilloscope* application is located in *tinyos-2.x/apps/Oscilloscope*.

To modify the *Oscilloscope* application:

1. The name of the sensor board driver we have created in *tos/sensorboards/ECG* should be added to the makefile in *Oscilloscope* application file (see Figure A.7).

2. The *Oscilloscope* application samples the default sensor via *DemoSensorC*, and we should change the application so that it samples the ECG sensor board connected to the IRIS via the sensor board driver created in the previous section. Remember that *ECGsensorC*, located in the *ECGG* sensor board driver, is a means to achieve sensor data acquisition from a platform specific *ECGdeviceC* component. As illustrated in Figure A.8, the DemoSensorC component has been changed to the *ECGsensorC* component.

3. The actual implementation of the application is in Oscilloscope.nc, (*tinyos-2.x/apps/Oscilloscope/OscilloscopeC.nc*). This module (see Figure A.9) is a grouping of different TinyOS applications explained completely on the TinyOS tutorial homepage [3]. It uses a timer to periodically sample the connected external sensor. Whenever 10 samples of ECG are collected in the buffer, a sending packet is created and will be sent to the destination mote via the *AM-Send* interface.

```
configuration OscilloscopeAppC { }
implementation
{
  components OscilloscopeC, MainC, ActiveMessageC, LedsC,
    new TimerMilliC(), new ECGsensorC() as Sensor,
    new AMSenderC(AM_OSCILLOSCOPE), new AMReceiverC(AM_OSCILLOSCOPE);

  OscilloscopeC.Boot -> MainC;
  OscilloscopeC.RadioControl -> ActiveMessageC;
  OscilloscopeC.AMSend -> AMSenderC;
  OscilloscopeC.Receive -> AMReceiverC;
  OscilloscopeC.Timer -> TimerMilliC;
  OscilloscopeC.Read -> Sensor;
  OscilloscopeC.Leds -> LedsC;


}
```

Figure A.8: *OscilloscopeAppC* Configuration.

```
module OscilloscopeC @safe()
{
  uses {
    interface Boot;
    interface SplitControl as RadioControl;
    interface AMSend;
    interface Receive;
    interface Timer<TMilli>;
    interface Read<uint16_t>;
    interface Leds;
  }
}
```

Figure A.9: *OscilloscopeC* Module Signature.

A.2.2 Base station

The IRIS Mote connected to the MIB520 USB interface should run the *BaseStation* application (the *BaseStaion* application can be found in *tinyos-2.x/apps/BaseStation*). *BaseStation* (see Figure A.10) is a basic TinyOS utility application. It forwards packets between the UART and the radio. It replaces the *GenericBase* of TinyOS 1.0 and the *TOSBase* of TinyOS 1.1. It provides a link between the serial port and radio network. It receives packets and sends them to the serial port via *UARTSend.*

```
configuration BaseStationC {
}
implementation {
  components MainC, BaseStationP, LedsC;
  components ActiveMessageC as Radio, SerialActiveMessageC as Serial;

  MainC.Boot <- BaseStationP;

  BaseStationP.RadioControl -> Radio;
  BaseStationP.SerialControl -> Serial;

  BaseStationP.UartSend -> Serial;
  BaseStationP.UartReceive -> Serial.Receive;
  BaseStationP.UartPacket -> Serial;
  BaseStationP.UartAMPacket -> Serial;

  BaseStationP.RadioSend -> Radio;
  BaseStationP.RadioReceive -> Radio.Receive;
  BaseStationP.RadioSnoop -> Radio.Snoop;
  BaseStationP.RadioPacket -> Radio;
  BaseStationP.RadioAMPacket -> Radio;

  BaseStationP.Leds -> LedsC;
}
```

Figure A.10: *BaseStationC* Component Configuration.

A.3 How to Determine the Suitable Sampling Rate for the Wearable Medical Sensor Connected to an IRIS platform

The features of a physiological signal such as ECG are very different from environmental signals such as temperature. Therefore, physiological signals are usually digitized at sampling rates from 500 Hz to 1000 Hz which lead to sampling periods from 1 to 2 ms.

The *Oscilloscope* application uses *TimerMilliC* which gives an independent millisecond granularity timer. On the IRIS motes (*Atmega128 microcontroller*), the 32 KHz external clock is divided by 32, so the best resolution we can get from that timer is 1 ms. To avoid timing issues with IRIS, a lower limit of 5 ms was used. As a result, we have two options to arrive at the 1 to 2 ms sampling interval from Timer hardware in Atmega128:

- using the Alarm interface which is asynchronous,
- running the ADC module in streaming mode.

Because most biomedical sensors such as ECG provide a continuous stream of readings, we programmed an IRIS to read data in blocks rather than in individual units. The *ReadStream* interface is used in TinyOS for a continuous stream of readings. It is intended for buffered high data rate reading, especially from medical sensor devices. Step by step instructions on how to use the *ReadStream* interface and to run the ADC module in streaming mode are explained as follows:

- First of all, some modification should be done in the sensor driver we have created in the previous section in order to run the ADC module in streaming mode. The *Read* interface provided by *AdcReadClientC* should be replaced by the *ReadStream* interface provided by *AdcReadStreamClientC* in the HIL component (*ECGSensorC*) of the ECG sensor board directory (*tos/sensorboard/ECG*). Or, we can create a new HIL component with the name of *ECGSensorStreamC* which uses the *ReadStream* interface (see Figure A.11).

- To use this interface, we should allocate a buffer, and declare the period in microseconds to acquire the samples to be buffered (see Figure A.12). Whenever enough samples of ECG are collected in the buffer, a sending packet is created and will be sent to the destination mote. As a result, the sampling rate and packet transmission rate over the network are dependent on each other such that higher sampling rates requires higher packet transmission rates. The IRIS mote can handle only 200 packet transmissions in a second which leads to a period interval of 5 ms for each packet. Therefore, we should define the size of the packet based on our desired sampling interval. In this application, the buffer contains 5 ECG samples.

- Whenever the timer is fired, by calling *ReadStream.postBuffer()*, the contents of the buffer are passed into the device. Then, with a call to *ReadStream.read()*

```
#include "ECG.h"

generic configuration ECGsensorStream() {
  provides interface ReadStream<uint16_t>;
}
implementation {
  components ECGdeviceC, new AdcReadStreamClientC();

  ReadStream = AdcReadStreamClientC;
  AdcReadStreamClientC.Atm128AdcConfig -> ECGdevicec;
}
```

Figure A.11: *ECGSensorStreamC* Component Configuration.

```
 enum {
 ECG_NSAMPLES = 5.
 us_FREQUENCY = 1000,
};
 uint16_t ECGSamples[ECG_NSAMPLES];
```

Figure A.12: Buffer Allocation and Declaring the Sample Period in Microseconds.

the sampling process will be started and the device begins filling the buffer based on the sampling interval in microseconds shown in Figure A.13 with the declaration of $\mu_{FREQUENCY}$ period. Figure 18 indicates two commands for calling *ReadStream.postBuffer*() and *ReadStream.read*().

- A *bufferDone*() event will be signaled once the buffer has been filled with data. If the lower layer finishes signaling *readDone*() and then finds that no more buffers have been posted, it will consider the read to be finished, and signal *readDone*() (see Figure A.14).

Note: T is the millisecond period the timer fires periodically, n the number of values the buffer should hold, and u the microsecond period between the samples in the buffer. T should be larger than $u.n$. If the difference between T and $u.n$ is small and about $1 - 2$ ms, the sampling rate is close to $1/u$.

```
/* At each sample period:
   - if local sample buffer is full, send accumulated samples
   - read next sample
*/
event void Timer.fired() {
   .
   .
   .

  call ReadStream.postBuffer(ADCSamples, ECG_NSAMPLES);
      call ReadStream.read(us_FREQUENCY);
      }
```

Figure A.13: Implementing the *postBuffer* Command.

```
event void AMSend.sendDone(message_t* msg, error_t error) {
  if (error == SUCCESS)
    report_sent();
  else
    report_problem();

  sendBusy = FALSE;
}

event void ReadStream.readDone(error_t result, uint32_t usActualPeriod) {
  if (result != SUCCESS)
    {
      report_problem();
    }
    for (i=0; i<10; i++)
  {
  local.readings[i] = ADCSamples[i];
      }
}
event void ReadStream.bufferDone(error_t result, uint16_t *Samples, uint16_t
ECG_NSAMPLES){}
}
```

Figure A.14: Implementation of *readDone* and *bufferDone* Events.

Appendix B

MATLAB programming for communicating with motes

This appendix presents the MATLAB-based infrastructure for communicating with IRIS motes via serial port. The IRIS mote can directly control a serial port (programming boards basically connect the mote's serial port pins to the actual serial port on the board).

The MATLAB serial port interface [83] provides direct access to the IRIS mote connected to our computer serial port via MIB520 USB. The serial port object provides us following functionalities:

- Configuration the serial port communications
- Using serial port control pins
- Writing and reading data
- Using events and callbacks
- Recording information to disk

In order to establish the serial port interface, we need to create a serial port object.

B.1 Creating a Serial Port Object

Using *serial* [84] function, we can create a serial port object. First, we should specify the port as serial requires the name of the serial port connected to the mote as an input argument. The MIB520 USB driver creates two sequentially numbered virtual COM ports. One of the COM port is for PC to MIB520 data communication which can be also used for MATLAB serial port communication. In order to create a serial port object associated with a serial port, we should enter

s = serial('port');

Before starting writing and reading data, both the serial port object and the IRIS mote must have matching communication settings. Configuring serial port communications involves specifying values for Serial Port Communication Properties such as *BaudRate*. *BaudRate* specifies the rate at which bits are transmitted. The default baud rate for IRIS platform is 57600. To specify the BaudRate, we should enter

s.BaudRate = 57600;

B.2 Reading Data

For serial port objects, we can determine whether read operations are synchronous or asynchronous using the *ReadAsyncMode* property. We can configure *ReadAsyncMode* to continuous or manual. If *ReadAsyncMode* is continuous, the serial port object continuously checks the mote to find out if data is available to be read. In case that data is available, it is asynchronously stored in the input buffer. The input buffer is a computer memory allocated by the serial port object to store data that is to be read from the mote. When the serial port is reading data from the mote, the data flow follows these two steps:

```
% Creating serial port object
s = serial('COM5');
IBS= 1000000;
% Configure property values for serial
s.InputBufferSize = IBS;
s.BaudRate = 57600;
s.ReadAsyncMode= 'continuous';
```

Figure B.1: creating a serial port object and configuring its communication settings.

1. The data read from the device is stored in the input buffer.

2. The data in the input buffer is returned to the MATLAB variable specified by the read function.

We can specify the maximum number of bytes that we can store in the input buffer with *InputBufferSize*. The *BytesAvailable* property shows the number of bytes currently available to be read from the input buffer. Figure B.1 illustrates how to create a serial port object and configure its communication settings.

To transfer the data from the input buffer to the MATLAB workspace, we can use *fread* [85] function. *fread* function is used to read binary data from the device. Reading binary data means that we can return numerical values to MATLAB. By default, *fread* returns numerical values in double precision arrays. However, we can specify many other precisions as described in the *fread* reference pages. Before we can use the serial port object to read data, we have to connect it to our mote with *fopen* [86] function. Figure B.2 illustrates the commands used for reading binary data from a mote. When we don't need to communicate with the device, we can disconnect it from the serial port object with the *fclose* [87] function (Figure B.3).

```
%Connect to the device
fopen(s)

%read data
out = fread(s,s.BytesAvailable,'uint8');
```

Figure B.2: The commands used for reading binary data from a mote.

```
%Disconnect and clean up
fclose(s)
delete(s)
clear s
```

Figure B.3: Disconnecting the mote from the serial port object.

B.3 Events and Callbacks

In this application, the PS implemented on MATLAB environment is responsible for reading and displaying ECG signals in real-time. Hence, the serial port application should be able to display the ECG signals after a specified number of bytes is available in the input buffer. We can enhance the power and flexibility of our serial port application by using events [88]. An event occurs after a condition is met and might result in one or more callbacks. While the serial port object is connected to the mote, we can use events to display ECG signals, records, analyze, and transfer data. All event types have an associated callback property. Callback functions are MATLAB functions that we make to suit our specific application needs. We should execute a callback when a particular event occurs by specifying the name of the callback function as the value for the associated callback property.

B.3.1 Bytes Available Event

For this application, *Bytes available* has been selected as the serial port event type and it involves in following callback properties;

```
B_per_packet = 37;    % The Number of Bytes in each Packet
%Creating Callback Event for serial
set(s, 'BytesAvailableFcnCount', B_per_packet);
set(s, 'BytesAvailableFcnMode', 'byte');
```

Figure B.4: Creating Callback Event for serial and configure its associated callback properties.

- *BytesAvailableFcn*, Specify the callback function to execute when a specified number of bytes is available in the input buffer.
- *BytesAvailableFcnCount*, Specify the number of bytes that must be available in the input buffer to generate a bytes-available event.
- *BytesAvailableFcnMode*, Specify if the bytes-available event is generated after a specified number of bytes is available in the input buffer.

A bytes available event is generated immediately after a predetermined number of bytes are available in the input buffer as determined by the *BytesAvailableFcnMode* property. If *BytesAvailableFcnMode* is byte, the bytes-available event executes the callback function specified for the *BytesAvailableFcn* property every time the number of bytes specified by *BytesAvailableFcnCount* is stored in the input buffer. This event can be generated only during an asynchronous read operation. Figure B.4 displays how to create a serial port event type and configure its associated callback properties.

B.3.2 Creating and Executing Callback Functions

After creating the event and setting up its configurations, we need to specify the callback function which should be executed when a specific event type occurs. We can specify the callback function as a function handle [89] or as a string cell array element by including the name of the file as the value for the associated callback property. Figure B.5 shows how we can specify the callback function as a cell array.

```
%Creating Callback function
set(s, 'BytesAvailableFcn', {'ReadECGcallbackFcn'});

% Connecting a serial port object to the device with the fopen
function.
fopen(s);
```

Figure B.5: Creating Callback function.

```
% ReadECGcallbackFcn callback function
function ReadECGcallbackFcn(s,event)
fread(s,B_per_packet);
.
.
.
end
```

Figure B.6: The *ReadECGcallbackFcn* callback function.

The *ReadECGcallbackFcn* callback function will be executed every time 37 bytes are available in the input buffer.

Callback functions (Figure B.6) must have at least two input arguments. The first argument is the serial port object. The second argument is a variable that captures the event information such as Break interrupt, Bytes available, Error, and Timer. This event information relates only to the event that caused the callback function to execute. We can also pass additional parameters to the callback function by including both the callback function and the parameters as elements of a cell array.

In this application, the *ReadECGcallbackFcn* callback function does multiple taxes such as recovering the payload data from the IEEE.802.15.4 message frame, and extracting the ECG readings, RSSI, LQI, and packet bursts. Moreover, it displays the ECG signals for three cycles, plots the RSSI value, and calculate the heartbeat. At the end it will transfer the ECG data and Heartbeat to the internet using TCP/IP.