Trinity College Dublin
Coláiste na Tríonóide, Baile Átha Cliath
The University of Dublin

MoBiLE: Movement Tracking using Bluetooth Low Energy

by

Neil Savio Carvalho, B.E. Electronics and Telecomm.

Dissertation

Presented to the

University of Dublin, Trinity College

in partial fulfillment of the requirements

for the Degree of

Master of Science in Computer Science

University of Dublin, Trinity College

August 2015

Declaration

I, the undersigned, declare that this work has not previously been submitted as an exercise for a degree at this, or any other University, and that unless otherwise stated, is my own work.

Neil Savio Carvalho

August 27, 2015

Permission to Lend and/or Copy

I, the undersigned, agree that Trinity College Library may lend or copy this thesis upon request.

Neil Savio Carvalho

August 27, 2015

Acknowledgments

I would like to convey my utmost gratitude to my supervisor Dr. Jonathan Dukes for his invaluable support, innovative thinking and guidance over the course of my thesis. He helped me find a solution to the research question in a very scientific and methodical way.

I would like to also thank Dr. John Dingliana for his support with the animation aspects of this project. I would also like to thank John Keogh and Tom Kearney for their awesome assistance with the hardware aspects of this project.

I would also like to thank my friends Smita Shivaram and Łukasz Jajeśńica for their continued motivation and support throughout this course. Lastly, I would like to thank my parents Oliveiro and Ingrid Carvalho and my sister Simantha Carvalho for their unconditional love and support throughout my life. It has been an extraordinary year for me and I am grateful to Dr. Ciarán Goldrick for giving me this unique opportunity as well as guidance to join and complete this course.

Neil Savio Carvalho

University of Dublin, Trinity College

August 2015

MoBiLE:

Movement Tracking using Bluetooth Low Energy

Neil Savio Carvalho, M.Sc.
University of Dublin, Trinity College, 2015

Supervisor: Dr. Jonathan Dukes

With the ever rising costs, non-uniformity in procedures and localized medical data in healthcare, eHealth and mHealth technologies strive to develop pervasive, low cost, uniform and ubiquitous health monitoring from the comfort of one's home. While the costs of rehabilitation can be about €60 per session and about 10 sessions are required on an average with a physiotherapist, a motion capture system with therapeutic feedback could provide a highly economical alternative with a one-time payment of €50-60 only. The Motion sensing over Bluetooth Low Energy (MoBiLE) system aims to design, implement and evaluate a low cost, lightweight and energy efficient remote exercise monitoring system for physiotherapy. This monitoring system processes a real time data stream from motion sensors and utilizes this data to animate human motion using augmented reality. A prototype MoBiLE system built with Mbed, nRF51DK, Python and Processing is designed, implemented and evaluated. This system uses Bluetooth Low Energy (BLE), a low power communication protocol and implements a novel Power Efficient Data Adaptive

(PEDA) algorithm to conserve battery life. This proposed solution allows the user to use the motion capture sensor with a coin cell battery which can last for hours . Initial experiments comparing this algorithm with other raw data techniques show that this technique is able to conserve considerable power and produce comparable animation results.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

According to World Health Organization, approximately 15% of the people in the world suffer from disabilities. Amongst them, about 110-190 million people have difficulties performing day to day activities. Stroke is the leading cause of permanent disabilities affecting 5 million people every year and causing lasting impairment. A range of barriers are faced by them and 51-53% of them are unable to afford health services and transportation. These services are also constrained by limited availability of medical therapists. Another issue is that the health care workers have inadequate knowledge and skills to properly treat them. The World Health Organization (WHO) has recommended that normative tools be developed along with guidelines to strengthen health care by shaping innovative and low cost rehabilitation technologies.

eHealth, mHealth and Ambient Intelligence in Healthcare(AmI) are new paradigms aimed at taking medical practices away from a localized medical information system to a pervasive, anticipatory and unobtrusive system as the older medical methods introduce many opportunities for error and inefficiencies. Examples include treating patients, conducting research, educating the health workforce, tracking diseases and monitoring public health. eHealth for patient care included improved access to health advice, remote consultations, tele-medicine and quicker access to emergency services. An example of a wireless body sensor network is show in Fig. 1.1. This eHealth system would proactively provide bio-feedback, be adaptive and sensitive to human gestures and activities. They could include devices that are embedded into regular clothing (e-Textiles) or everyday objects that can communicate with other devices to actively monitor lifestyle. An AmI is characterized by context awareness, personalization, anticipatory sensing, adaptability, ubiquity and transparency.

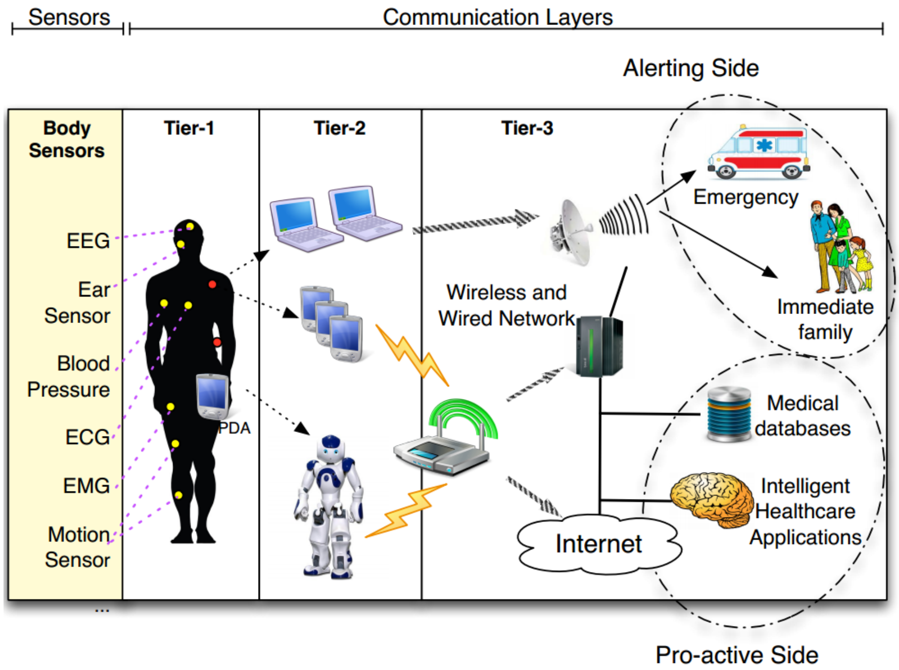

Figure 1.1: Example of a Wireless Body Sensor Network [Acampora et al. 2013]

This dissertation describes the design and implementation of a MoBiLE system which seeks to reduce bandwidth by reducing the number of packets sent over the air, reduce power consumption, thereby increasing battery life and evaluate the performance of Bluetooth Low Energy in real-time streaming of eHealth data.

1.1 Motivation

Currently, there are a limited systems built for movement tracking. However, they are face a variety of problems. Many motion capture systems based on video capture like Microsoft

Kinect suffer from problems such as being limited by line of sight and are not suitable for outdoor use. These systems need to be installed before use. Some of the other systems like marker-based motion capture are obtrusive and hence are not convenient for everyday use or rehabilitation. Further limitations are that these systems are not scalable or cannot be integrated into a centralized system or they currently have no mechanism to transmit data to a cloud where the transmitted data can be used for remotely monitored. The biggest drawback is the lack of a common and widely available communication technology for interoperability between different systems. All these factors drive the need for a system that can not only capture real time motion, but also has commonly used technologies, is ubiquitous, low cost, lightweight and battery efficient and can provide visual feedback.

1.2 Research Area

Our main research goal is to develop and evaluate a MoBiLE system that unobtrusively senses hand movements and that can be modularly embedded in everyday garments. The feasibility of providing quantitative assessment and evaluation based on identification of physio-therapeutic movements by leveraging augmented reality is explored during the course of this research. The quality of performance of these tasks are assessed by applying performance criteria. We will also investigate intelligent bio-feedback technologies which would simulate a clinical setting.

1.3 Objectives

The main research objective of this project is as follows:

- To research the State of the Art in Ambient Intelligence in rehabilitation for recovery from strokes, cardiac disorders, sports related injuries etc.
- To investigate feasibility of using a communication technology like Bluetooth low Energy can be used as a low cost, reliable and power efficient technology for real-

time data streaming of motion sensor data and discuss the advantages of such a protocol over other communication protocols like Zigbee and Bluetooth.

- To propose and evaluate a novel Power Efficient Data Adaptive (PEDA) algorithm to transmit motion sensor data.
- To design a Motion Processing Unit (MPU) Service as a GATT specification to transfer motion sensor data.
- To design an Augmented Reality System to animate real-time human motion.
- To identify future areas, improvements and issues.

1.4 Road-map

Chapter 2 discusses different state of the art medical applications of Wireless Body Sensor Networks for eHealth. It also discusses and compares various wireless communication protocols like Zigbee, ANT and Bluetooth Low Energy which can be used for real-time streaming of sensor data. Current techniques and products for motion tracking and estimation for remote rehabilitation are also discussed. Chapter 3 describes the proposed design of the system i.e. the system architecture and how various components of the system interact with each other to implement a common goal. Chapter 4 discusses the prototype implementation of this design with flowcharts explaining how the individual functionalities were realized in hardware and software components. It also contains a discussion chapter detailing various alternate techniques implemented as a part of this thesis. Chapter 5 provides a detailed evaluation testing and comparing the power consumption and packet size of various algorithms. Chapter 6 discusses about the main contributions of this thesis and also the future work regarding the improvements in the proposed system and bio-feedback that can be researched for remote physiotherapy.

Chapter 2

Background

2.1 Introduction

In this chapter, we look at the emerging field of Ambient Intelligence in healthcare. We survey and examine various state of the art techniques within the area of remote physiotherapy monitoring for Wireless Body Sensor Networks (WBSN). Various different techniques for healthcare monitoring and the technologies behind them will be analyzed and compared.

We discuss functional and technical challenges of Wireless Body Sensor Networks (WBSN) such as power consumption, cost, real time streaming and size and survey various ways to implement it. We also investigate different ways in which human motion can be detected for remote monitoring for physiotherapy applications. We review techniques related to detecting both human activities like walking, running, standing as well as human limb motion. We propose to preliminarily survey various techniques and implement this model with a 3-axis accelerometer, then integrate it with gyroscopes and then magnetometers and understand the performance for detecting human motion. We will then discuss more advanced techniques which make use of Inertial Motion Sensor (IMU), a network of IMUs and then IMU with digital motion processors.

Secondly, we discuss wireless communication techniques like Bluetooth low Energy and ZigBee. We also evaluate the advantages of Bluetooth Low Energy in terms of power consumption and size in comparison to ZigBee. We assess whether Bluetooth Low Energy can be used for data streaming for real-time medical data like heart beats or motion data.

Thirdly, we examine different Augmented Reality techniques to create interactive 3D human models that can be reconstructed using real time signal from IMUs. We discuss various methodologies like quaternion and Kalman filters to create this 3D model. We also discuss various algorithms like Hidden Markov Models or State Vector Machines or neural networks to automatically verify whether the exercise done is accurate or not. We also discuss regarding the various products currently available in the market and the drawbacks from them.

2.1.1 Requirements of Wireless Body Sensor Networks

Advanced information management and body sensor networks are required which can enable remote monitoring of patients at home. Due to the recent advances in wireless communication, reducing sensor sizes and improving energy storage techniques, the vision of a pervasive Wireless Sensor Network is a step closer to becoming a reality. This monitoring could include movement tracking, physiological data, vital signs like ECG (Electro Cardio Gram), EEG (Electro EncephaloGraphy), pulse, temperature, etc. Continuous sensing of this medical data and transmission of this sensitive data from various devices to a central system for medical monitoring is required. Due to the difference in the type of data being transmitted, these networks have different requirements from traditional Wireless Sensor Networks. [Zatout, 2012] outlines the various requirement of Wireless Body Sensor Networks as follows:

1. **Quality of Service:** Quality of Service (QoS) in terms of bandwidth allocation, data transfer delay, packet error rate and jitter are important parameters to be considered while transmitting real time motion sensor data.

2. **Energy Consumption :** A crucial factor in WBSN is energy consumption as it

affects the sensor lifetime

3. **Scalability:** Any WBSN should accommodate the possibility of adding more sensors to the network. It should also be adaptable to any future changes in the network

4. **Location:** The WBSN should not be restricted to home constraints. It should be available for ubiquitous and unobtrusive use in indoor and outdoor environments.

5. **Comfort Level:** Another important characteristic to consider is the comfort level of the individual as he wears such a device. The device should be lightweight and easy to wear.

6. **Security:** Security is an important aspect while building WBSN's as data flows in an unencrypted format and can be easily sniffed. Currently, a lot of wireless communication techniques have encrypted modes for sensitive data transfer. This is particularly useful like in case of a heart pace monitor.

2.2 Motion data acquisition methods using wireless technologies

This section discuss different methods to perform motion data acquisition. Accelerometers and Gyroscopes measure acceleration and rotation, but individually they are unable to provide adequate information to be able to deduce orientation, velocity and position. Inertial Measurements Units (IMU) are boards contains any combination of sensors like accelerometers, gyroscopes and magnetometer to measure these parameters. Each of the sensors have three axis also called Degrees of Freedom (DOF), so a combination of these sensors and their axis produce 3 DOF, 6DOF and 9 DOF IMUs. These IMUs are increasingly being used in motion sensing in a variety of products and the DOFs are decided based on their applications.

2.2.1 Inertial Measurement Units

[Giggins et al. 2013] attempt to find out whether a single Inertial Measurement Unit (IMU) on the shin would be sufficient to provide effective biofeedback during lower limb exercises and under different exercise conditions. The authors mention various different techniques previously used to achieve this goal by capturing data using accelerometers, gyroscopes, force capture systems, optical motion capture systems, multiple IMUs attached on different parts of the body and then using various techniques to identify correct exercise procedures like using Classifiers (AdaBoost), Kinematic models and extended Kalman filters.

[Buonocunto and Marinoni 2014] proposes a low cost, small, energy efficient, high performance and flexible tracking system based on Inertial Measurement Units (IMU) as a body area network. The key objectives for this implementation are flexibility in terms of adaptability to other applications, accuracy in terms of precision to 1 degree, real-time feedback in terms of a maximum bound on the delay in milliseconds between movement and visualization, and low power consumption. Mean angle error, position error and execution and delay times for this system seemed to be comparable to other systems like Polhemus system with minor errors for fast movements.[Peppoloni et al. 2013] propose a kinematic for articulation of the human upper limb with more accuracy by using an accurate joint angle, velocity and acceleration estimation obtained from fusion measurements from 7 degrees of freedom (DoFs) i.e. tri-axial gyroscopes, triaxial accelerometers and magnetometers with the use of Unscented Kalman Filter (UKF).The algorithm consists of three models together used for joint angle estimation i.e. upper limb kinematics model, system dynamics model and sensor measurement model. The kinematic model consists of 2 DoFs motion of clavicle, 3 DoFs motion for the shoulder and 2 DoFs motion for the forearm. The results show that the system is able to track joint angles with good accuracy and slightly outperform the previous 5 DoF model.

[Benocci et al. 2010] investigates whether ZigBee-based data streaming could be utilized to implement an accelerometer data-triggered fall detection system while meeting the key requirements of Wireless Body Sensor Networks (WBSN) like higher communication efficiency and low power consumption in comparison with a Bluetooth serial port profile solution. The author compares and contrasts the two technologies in terms of their coverage area i.e. 10m for Bluetooth and 50m for ZigBee, throughput i.e. 720Kbps for Bluetooth and 250kbps for ZigBee, security, power consumption i.e. 60mA for Bluetooth and 40mA for ZigBee and network topology i.e. piconet. The prototype was built using a tri-axial accelerometer and worn on the lower back at the waist to monitor the body center of mass acceleration. The data is then sent via the two technologies to a C++ application on the desktop or a mobile application which extracts the relevant features and generates an alarm to a concerned medical official when a fall is detected. The fall detection algorithm uses a single-threshold algorithm as it consumes lower computational power. The transmission performance of both Bluetooth and ZigBee is evaluated by comparing their power, lifetime and throughput. The study show that optimized ZigBee shows an improvement of 88% more lifetime then Bluetooth and 40% more than a non-optimized case and improved performance.

[Jihyoung et al. 2011] propose a cost efficient, unobtrusive and low complexity wearable upper limb tracking technique for home rehabilitation therapy using Inertial Measurement Units, Zigbee, kinematic modelling and dynamic time warping. The first phase is a data collection phase using inertial measurement units consisting of tri-axial accelerometers and tri-axial gyroscopes placed on the upper limb. The second phase is an inertial tracking phase where the position of the arm is estimated in a world coordinate system using fusion values obtained by integrating gyroscope measurements and by double integrating accelerometer measurements. The accuracy of motion estimation is 85% as compared to the previous methods and the accuracy of motion matching is around 88.5% as compared to the reference motion library. [Lingfei et al. 2012] propose a wearable wireless multisensory system design which enables realtime measurement of human activities like energy expenditure and breathing while addressing key challenges of such sensor networks like limited battery life and data synchronization. To overcome these challenges, two algorithms i.e. multi-data packaging and slot data synchronization have been developed for energy efficiency and synchronization. The processor of the SNs are woken from sleep every 33.3 ms, the sensors sample signals and stores the data in a 43 buffer. An activity prediction algorithm based on support vector machines was also implemented. It is observed from these experiments that the time deviation is reduced to 23.85ms compared to a deviation of 293.92ms without the algorithm. The battery life also shows a significant improvement of 68h with this algorithm compared to 16h without it.

2.2.2 Remote Therapy Products

This section discusses the various commercial products available in the market for remote monitoring of patients. [Jarochowski et al. 2011] propose a Zigbee based WBSN that monitors patients by collecting the data from sensors. sending this data via Zigbee motes to a server application that enables specialists to provide correct prescriptions to patients for rehabilitation. Intelligent Sweet Home is another system monitors human lifestyle by continuously checking their health status [Park et al. 2007; Jung et al. 2005] have developed an Intelligent Sweet Home (ISH) which can continuously check a person's health status or intentions actively supporting people with disabilities.[Helmer et al. 2010] proposed a rehabilitation system aimed at improving the life of patients suffering from Chronic Obstructive Pulmonary Disease (COPD) through remote rehabilitation training and monitoring.

2.3 Wireless data communication technologies

A variety of communication technologies like Zigbee, ANT, Bluetooth and Bluetooth Smart have been implemented in WBSN. In this section, we discuss the implementations of these technologies, their applications, evaluations and drawbacks and based on this data, we make a choice regarding the communication technology to be used for this dissertation.

2.3.1 Zigbee

Zigbee is a low power short distance communication protocol that is built on an 802.15.4 protocol for used in wireless sensor networks. Compared to Bluetooth, this technology provides a low latency as the physical layer Direct Sequence Spread Spectrum allows nodes to switch to sleep mode without lose of synchronization. It provides simple forms of guaranteed QoS. Zigbee is a mesh network used in low-power wireless space that could span entire buildings.Its energy utilization is 2.5 times more as compared to BLE. Due to the use of mesh networks, it also has high latency. It is also susceptible to interference as it does not employ channel hopping. It works on the 2.4HGHz band worldwide and has data rates varying from 20-250Kbps. Zigbee's power requirements and complexity make it unsuitable for devices which need to run for extensive periods on a limited power source like a coin cell battery. Application areas include industrial control, building automation, medical data collection and home entertainment control.

2.3.2 ANT

ANT is a low power proprietary multi-cast wireless sensor network technology established by Dynastream. It is a communication protocol stack that operates in the 2.4 GHz ISM band and establishes communication standards for data representation, signaling, authentication and error detection. Data Frames carry only 7 bytes instead of 15-35 bytes in IEEE 802.15.4. It supports two peer to peer and star topologies. ANT is less popular than IEEE 802.15.4. It is a ultra-low power, low bandwidth wireless protocol and immunity to interference. It also requires lesser transactions per connection event and lesser overhead to send the same amount of data as compared of Bluetooth Low Energy.

2.3.3 Bluetooth

Bluetooth is a widely used wireless technology for exchanging data like audio, video, files over short distances managed by the Bluetooth Special Interest Group operating in the 2.4GHz band and used in mobile devices as well as Personal Area Networks. It has 79 designated channels and can transmit data via adaptive frequency hopping on any of the channels. Several profiles are defined like Health Device Profile, Headset Profile or headset profile which are needed to be implemented by the device to use the desired services.This technology can be used in a piconet with seven active slaves per piconet (255 in park mode). However, Bluetooth is not the best choice for Wireless Body Sensor Networks due to its high energy consumption, high synchronization costs and the complex network topology.

2.3.4 Bluetooth Low Energy

Bluetooth Low Energy (BLE) also known as Bluetooth Smart is a ultra low power, multi-vendor inter-operable wireless technology powered by coin-cell batteries for use in devices such as sports & fitness, beacons, wearables, healthcare and entertainment devices. Bluetooth 4.2 allows two types of implementation:

1. **Dual-mode chips:** Bluetooth Low Energy and Classic Bluetooth are integrated together on a single chip. The results in an architecture where the most part of Classic Bluetooth technology's existing radio and functionality is shared with BLE resulting in a minimum cost increase compared to Classic Bluetooth technology.

2. **Single-mode chips:** These devices are standalone BLE devices optimized for use with small coin size batteries, having low cost and low power consumption. A typical single-mode device is a heart rate sensor.

Following are the characteristics of Bluetooth Low Energy:

1. Data is transferred in short packets i.e. 8 octets up to max 27 octets at 1Mbps.

2. BLE employs adaptive frequency hopping to minimize interference from other technologies in the 2.4GHz ISM Band.

3. The controller allows the host to sleep for long periods of time and is woken by the controller only when the host needs to carry out a task.

4. Connection setup and data transfer can take place in a very short interval of time i.e. 7.5ms.

5. It has a 48-bit number that uniquely identifies a BLE device similar to an Ethernet Media Access Control address.

6. It can provide a range of over 100 meters.

7. It uses 24-bit Cyclic Redundancy Check (CRC) on all packets, thereby ensuring that the data sent is robust.

8. AES-128 encryption can be used to provide security and authentication

9. BLE uses 32 bit access address which theoretically allows billions of devices to be connected using a star topology.

BLE Protocol Stack

Fig. 2.1 depicts the three basic building blocks i.e. application, host and controller of the BLE Protocol Stack. Each of these blocks are then split into more layers

1. **Application:** It is the highest layer and responsible for logic, data handling and User interface.

2. **Host:** It includes the following layers:

   (a) **Generic Access Profile (GAP):** It defines how two BLE devices perform device discovery, connection establishment and standards to ensure interoperability.

(b) **Generic Attribute Profile (GATT):** The GATT sits on the Attribute Protocol (ATT) and defines how data is organized and exchanges between BLE Devices. It has a fundamental difference from ATT as the data is encapsulated in GATT Services which contain characteristics.

(c) **Logical Link and Adaptation Protocol(L2CAP) :** It provides two main functions:

i. Fragmentation and Recombination: Processing data packets received from the upper layers, breaking them into smaller packets so that they fit into the 27-byte max payload size.

ii. Encapsulation: Taking multiple protocols from upper layers and encapsulating them into a standard BLE packet format.

(d) **Attribute Protocol (ATT) :** A simple client/server stateless protocol which uses attributes as the for organizing data. Each attribute contains a 16-bit attribute handle, a UUID, set of permissions and a data value.

(e) **Security Manager (SM) :** It is a protocol and a set of security algorithms that allow two BLE devices to communicate with each other securely over an encrypted link. It provides support for pairing, bonding and encryption re-establishment.

(f) **Host Controller Interface (HCI), Host :** It is a standard protocol that facilitates communication between a host and a controller via a serial interface. An example of a host-controller setup would be a smartphone or a tablet, where the host would run on the main CPU while the controller is a hardware chip connected via USB or UART. It has a set of commands and events for the host and controller to communicate with each other.

3. **Controller:** The controller has the following layers:

(a) **Link Layer(LL):** This layer is a time constrained layer isolated from the upper layers by the HCI. The functionalities provides by this layer include Preamble, access address and air protocol framing, CRC generation and verification, random number generation and AES encryption. BLE has an asymmetric architecture in its lower layers as master configuration requires more resources then slave.

(b) **Physical Layer (PHY)**: This layer performs analog communications and is capable of modulating, demodulating and digitizing signals. The radio uses the 2.4GHz ISM band to communicate and divides the band into 40 channels from which 37 are used for connection data using adaptive frequency hopping spread spectrum while last three are used for advertising, connection setup and broadcasting data. The modulation used is Gaussian Frequency Shift Keying (GFSK) that is similar to Bluetooth.

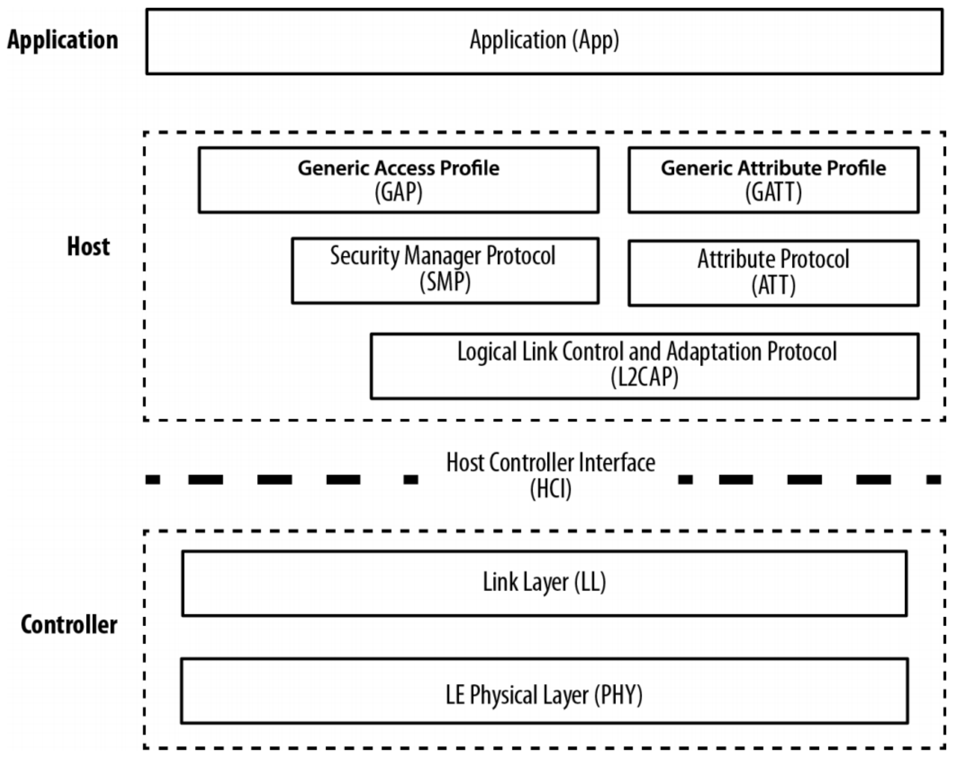

Figure 2.1: BLE Protocol Stack [Townsend et al. 2014]

Generic Access Profile (GAP)

GAP is one of the crucial building blocks that allow BLE devices to interoperate with each other. It is a framework that defines how devices perform device discovery, data broadcasting and connection establishment in a standardized format. This is the lowest API stack available for application programmers to develop their BLE applications. There are different roles defines which impose rules and behaviors allowing devices to communicate with each other. A device can operate in more than one role simultaneously. GAP defines four roles that any device can adopt :

1. **Broadcaster:** This role is used for periodically transmitting advertisement packets with data. For example, this role could be used by a beacon that constantly transmits its location and some other data like facilities available around it. The data sent by a broadcaster is accessible to everybody. This corresponds to the Link Layer advertiser role.

2. **Observer:** This role is optimized for applications that would act as a data sink to aggregate data from broadcasting devices. For example, a smartphone that collects data from many beacons and tries to perform position estimation. This corresponds to the Link Layer scanner role.

3. **Central:** This role is used to enable to device to establish multiple connections to peers. Only the device in a central role can initiate BLE connections and allows devices to join the network This corresponds to the Link Layer master role. The computing requirements of this device is higher and hence the central role is usually played by a smartphone or a tablet.

4. **Peripheral:** This role uses advertising packets to allow central devices to discover it and connect to it. The peripheral has lower computing requirements as compared to a central in terms of processing power and memory and are inexpensive. This corresponds to the Link Layer slave role.

Generic Attribute Profile (GATT)

The GATT defines how two BLE devices exchange Service data between them.
GATT defines two main roles:

- **Client:** It sends requests to the servers like read, write, notifications and receives responses from it. The GATT Client can discover services hosted by the GATT service by performing service discovery and does not need *apriori* information about it. It can then read or write attributes from or to the server as well as receive notifications (server initiated updates)

- **Server:** It receives requests from the client and sends responses back. It can also send notifications or indications to the client. Its role can be compares to a Web Service.

All GATT roles are independent of GAP roles and are compatible with each other.

GATT Service

Fig. 2.2 depicts a Generic GATT Service. The topmost building logical block is a profile, which is composed of many services, which contain characteristics which have attributes. A group of services for a particular requirement form a profile. A GATT Service is basically a set of characteristics bundled together for a particular purpose. For example, Device Information Service would have characteristics like Manufacturer Name String, Model Number String and Serial Number String. A Characteristic act as a user data container which is similar to an attribute. They contain two attributes: characteristic declaration which basically contains a UUID, value handle and properties of the characteristic and a characteristic value which is the attribute that contains the actual user data. The characteristic may also contain descriptors like the Client Characteristic Configuration Descriptor which is used for Server-Initiated Updates.

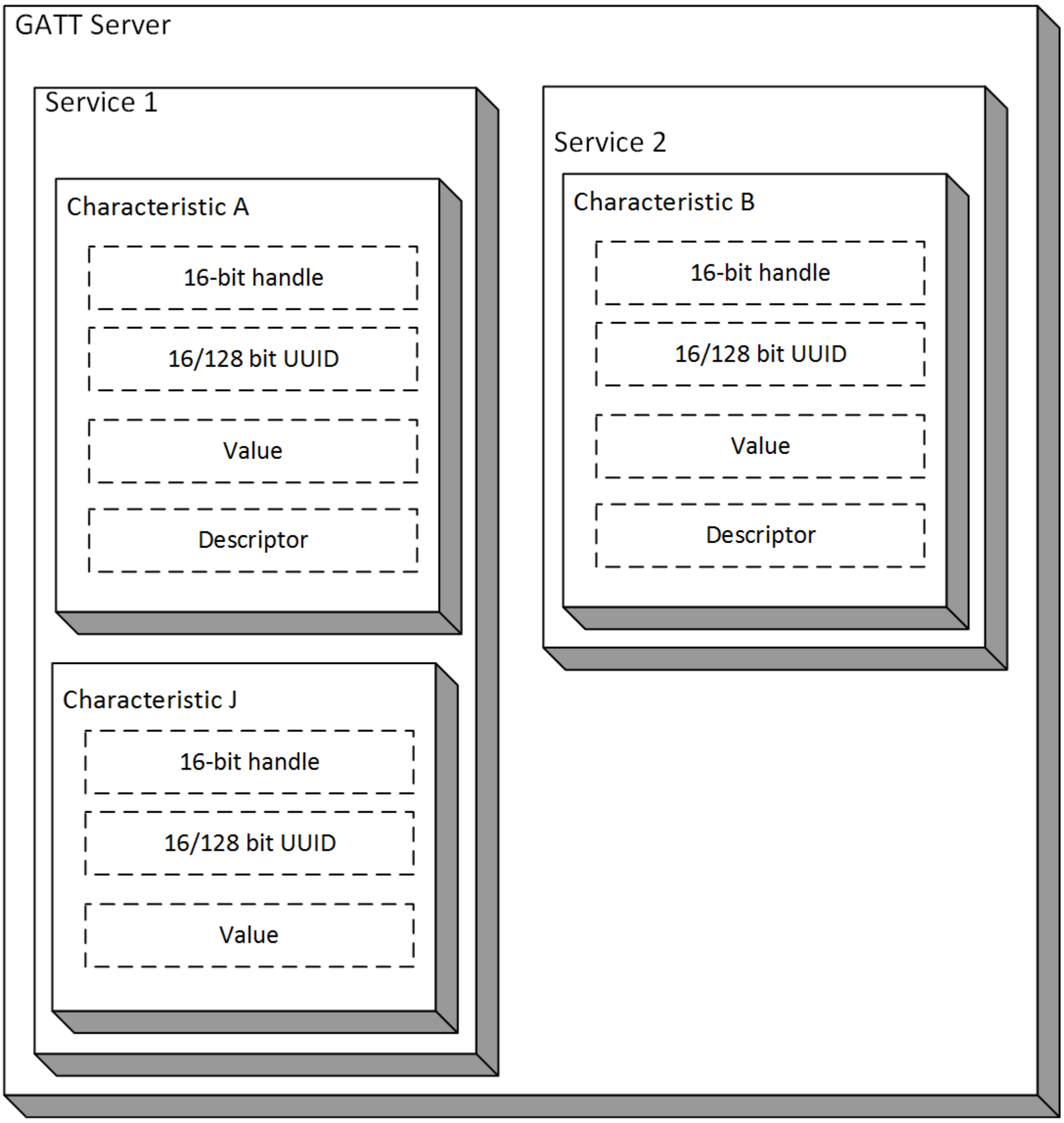

Figure 2.2: Generic GATT Service

Some of the commonly used attribute requests are:

- **Requests**: The client issues writes to or reads from a characteristic and the server needs to send acknowledgments back.

- **Commands**: The client issues writes to or reads from a characteristic and the server doesn't to send acknowledgments back.

- **Notifications**: The server sends data to a Central and the latter does need to send acknowledgments back.

- **Indications**: The client sends data to a Central and the latter needs to send acknowledgments back.

Connections and Parameters

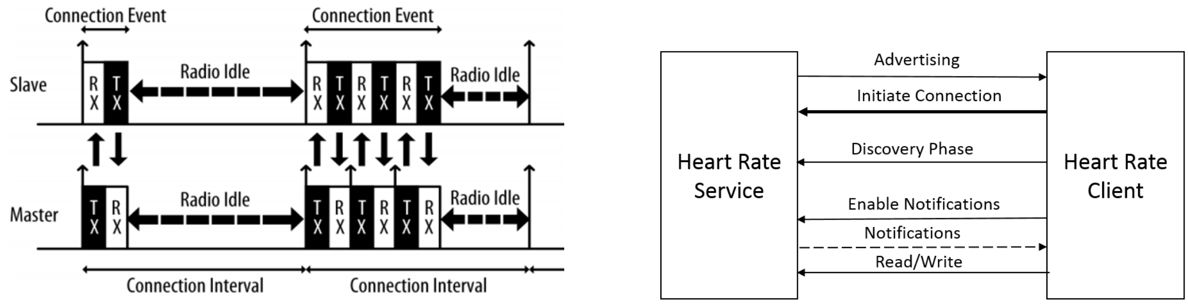

Figure 2.3: Basic BLE Connection (i) Master and Slave Connection Interval [Townsend et al. 2014] (ii) Master Slave Connection Sequence

Bluetooth Low Energy is built around a Client-Server architecture. A server is basically a device with resources which can be connected to by clients. First, the master starts scanning to look for advertising peripherals that are accepting connection requests. Based on the Bluetooth Address, a suitable slave is detected and the master initiates a connection request to the slave using one of the three channels. If the slave responds, the master establishes a connection with the slave. The frequency hop increment is also sent with the connection request so that both the master and slave follow the same hopping sequence. The remaining 37 channels are then used for data transmission. Fig. 2.3 shows how a basic BLE connection is established.

Connection Parameters These are connection parameters that defines how often the data would be transferred within a connection. The parameters can be setup on both the Client and the peripheral but it entirely depends on the Client which configuration parameters it wants to use. These configurations are constraints by the type of the BLE chip on the client so it varies between devices.

- **Connection Interval:** This is the time between the start of two consecutive connection events. This value can be set between 4s (low data rates -lower power consumption) or 7.5ms (high data rates - higher power consumption).
- **Slave Latency:** This defines how many connection events a slave can skip without being disconnected
- **Connection supervision timeout:** This defines the maximum time interval between two received valid data packets before the connection is to be considered as lost. According to the BLE specifications, It should satify the below equations:

$$Supervision\_Timeout * 4 > (1 + slave\_latency) * connection\_interval_{max} \quad (2.1)$$

$$Supervision\_Timeout > (1 + slave\_latency) * connection\_interval_{max} * 2 \quad (2.2)$$

... where *connection_interval_max* is given in milliseconds

The number of packets that can be transmitted per event varies with the type of the Bluetooth Device. For example, an iPhone 6, Nexus 4 and nRF8001 can transmit upto 6, 4 and 1 packet respectively. given the data above, the throughput for a BLE connection can be calculated using the connection interval I, number of packets transmitted per interval N and number of bytes transmitted per packet.The maximum throughput is calculated as:

$$DataRate = N * 20B * \frac{1}{I} \quad (2.3)$$

So for a BLE device with N=6 (maximum no. of packets) and I=0.0075ms (if we take the min connection interval), the maximum achievable data rate would be 128kbps. These parameters are tuned depending on the applications and factors like latency and power consumption. For example, if it is a system requirement that a connection interval cannot be skipped.

[Laine et al. 2014] proposes a technical design and challenges for a mobile Bluetooth-based gateway that mediates a connection between a local sensor network and a remote server system using ZigBee, Bluetooth and WLAN/3G for remote health care monitoring. It outlines the various requirements for such ubiquitous health care systems like high user or object mobility, unobtrusive monitoring, contextual and personalized user access to information, reliability, speed, low latency, low cost and power and flexibility in adding new nodes. The design mainly consists of a ZigBee-based body area network which sends data over Bluetooth to the mobile gateway which then sends this data to a centralized database as well as alerts regarding this data are sent to the medical professionals via SMS or calls over data mobile network 3G/4G.Thus by integrating Bluetooth and ZigBee, the problems regarding single point data aggregation and low data speeds are resolved.

[Linde and Tucker 2013] describe a novel system of transferring audio using BLE between electronic devices which can be used in assistive learning technologies (e.g. hearing aid). When a electronic device receives a Protocol Data Unit (PDU) in its BLE Link Layer, it reads the header of this data to check if the PDU contains audio data. If the PDU contain audio data, the device can then send this audio data to its audio layer in the BLE stack for further processing. This system claims to result in much lesser power consumption then traditional Bluetooth Classic. [Touati et al. 2013] propose and implement a remote real time ECG monitoring system interfaced with a local computer. Due to the ease of integration, and low power advantage of BLE devices, its more easily adaptable for use in ECG monitoring as compared to Zigbee. The evaluations show that the captured ECG data over BLE is almost identical to that captured over traditional devices and thus, proving that BLE shows great potential for healthcare applications.

2.3.5 Time synchronization using Bluetooth Low Energy

In this section, we discuss the different methods used to achieve time synchronization using Bluetooth Low Energy. [Casamassima et al. 2013] analyses two algorithms to synchronize various devices during data streaming or logging connected in a Bluetooth piconet to achieve an accuracy of 1ms with the use of the piconet internal clock. They also aim to achieve higher performance and good noise immunity with no communication overhead while sustaining sufficient data throughput. The first algorithm analyzed in this paper are piconet clock synchronization and correction method which consists of reading the piconet clock at regular intervals of time using a command (iWarp) by switching from data mode to command mode and then storing the clock value in the flash memory or sending the packet to the master node. The other algorithm is using a low power performance Sniff mode in which the device stays synchronized by listening to the piconet at regular intervals of time for a short instant. Experiments show that these algorithms show an error of 1.039ms and 0.313ms respectively which is low. The first algorithm has an overhead of only 4 bytes for the master node. The sniff mode also reduces power consumption, improves accuracy and delivers a better performance.

[Ghoshdastider et al. 2014] propose an algorithm to achieve time synchronization using a synchronization add-on and synchronization center for a collaborative brain computer interface using BLE. The proposed system consists of multiple BCI systems where Visually and Auditory evoked potential (WEP/AEP) simulator plays stimuli data in real-time and is connected to a PC. On the other side, users are mounted with EEG devices like cap, vest, armband, Inertial measurement units (IMU) that acquire and transmit data to a remote PC. The first step of this implementation consisted of the SC broadcasting a Tick signal periodically and N number of times. The EEG system with the lowest timestamp receives the tick first and is selected as the SyncParent and provides a global time in the network. The latency of each module after synchronization is assessed using wires connected to the general purpose input/output pins of each APP processor. It is found that this method gives an average latency of 37.781 ms which is well below the 100ms latency target required for such applications.

2.3.6 Comparison of Bluetooth Low Energy with other protocols

A detailed comparison of the difference in features between various communications technologies is given in Table 2.1.

Features	Wireless Technology			
	Zigbee	**Bluetooth**	**ANT**	**BLE**
Industry	Zigbee Alliance	Bluetooth SIG	Dynastream	Bluetooth SIG
RF Frequency	2.4 Ghz, 868 MHz	2.4 Ghz (79 channels)	2.4 GHz-2.4835 GHz	2.4 GHz-2.4835 GHz (40 channels)
Data Rate	20,40 & 250 Kbps	1-3 Mbps	1-3 Mbps	1 Mbps
Application throughput	$\sim$ 120 kbps	0.7-2.1 Mbps	20 kbps	$\sim$ 270Kbps
Maximum range	10-100 m	10-100 m	30 metres at 0 dBm	$>$ 100m
Energy	Very Low	Low	Very Low	Very Low
Nodes	65000 (20 Kbps), 255 (250 Kbps)	8 (7 slave+1 master),	8	8 (7 slave+1 master),
Battery	3.1 days	2.2 hours	Day/Months	Day/Months

Table 2.1: Comparison between different wireless technologies

The detailed evaluation shows that amongst the four protocols, BLE seems to have high throughput, low energy consumption and is not proprietary like ANT so seems to be a good choice to implement a motion sensing system. More detailed evaluation from various research papers are discussed to understand the different metrics related to BLE. [Young-Jin and Hui-Sup, 2013] propose an experiment to evaluate the power optimization and transmission performance of medical data i.e. ECG using Bluetooth Low energy (BLE) due to its low extremely low peak, idle and average current consumption. The authors discuss the different available Bluetooth 4.0 modes i.e. single-mode and dual-modes and their features like ultra-low-power behavior. A new ECG data packet format is designed and an optimum connection interval studied. The results show that the data rates obtained are about 6.66kbps and a battery life of 9-13 hours on an average.Another paper by [Dementyev et al. 2013] proposes an analysis to evaluate the power consumption and connection intervals of different wireless protocols like Bluetooth Low Energy, ZigBee and ANT for an optimum cyclic sleep scenario. The authors aim to reduce sleep and awake time, low latency, low connection time while maintaining high data rates for the above protocols. The results show that for all sleep intervals, BLE outperforms ZigBee and ANT for sleep current consumption. However, as BLE employs a frequency hopping scheme, it takes longer connection time then ZigBee and ANT and consumes more active current then ANT protocol. In terms of duty cycle, BLE still performs better than the other two protocols. The findings suggest that the optimal sleep interval is 10.0s for BLE, 14.3s for ZigBee and 15.3s for ANT.

[Siekkinen et al. 2012] proposes a similar comparative study of energy consumption of Bluetooth Low Energy (BLE), IPv6 with BLE, Wi-Fi and ZigBee/802.15.4 by measuring the consumption using a power monitor and deriving characteristic models based on the ratio of energy per bit. The study also considers the impacts of interference by WiFi during data transmission, effects of different overhead for each of the protocols and limitations of the current implemented BLE stack transmitted by each protocol. The energy consumption is then studied for two states: non-connected and connected. For

BLE, connection interval and no of packets sent per connection maximize energy utility and that its energy consumption is 2.5 times more than ZigBee. Interference immunity for BLE is 60% at 1.5 metres while for ZigBee it is 35% at the same distance. The author concludes the research by saying that ratio of energy per bit transmitted is very less for BLE compared to ZigBee and Wi-Fi but better immunity to interference can be implemented using Adaptive frequency hopping.

[Laine et al. 2014] proposes a technical design and challenges for a mobile Bluetooth-based gateway that mediates a connection between a local sensor network and a remote server system using ZigBee, Bluetooth and WLAN/3G for remote health care monitoring. The design mainly consists of a ZigBee-based body area network which sends data over Bluetooth to the mobile gateway which then sends this data to a centralized database as well as alerts regarding this data are sent to the medical professionals via SMS or calls over data mobile network 3G/4G.The main conclusions of this paper are by integrating Bluetooth and ZigBee, the problems regarding single point data aggregation and low data speeds are resolved. It provides mobile sensor control by designing control Bluetooth packets to modify its features like sensing period, start and stop time and also the ability to start/stop all sensors at once.

2.4 Movement Tracking

This section discusses the various methods implemented to deduce the activity or motion of any individual utilizing different techniques like RSSI, machine learning methods and rule based techniques given the sensor data. All these procedures try to exploit some common trait between activities to form a training set and then use this set to make decisions regarding what and how an exercise is performed.

2.4.1 Radio Propagation Measurements

[Munoz et al. 2014] evaluate radio propagation measurements like received signal strength (RSS) of four on-body channels from waist to the wrist, ankle, chest and back during different sport activities like jogging, rowing and cycling using wireless sensor nodes. During activities like walking or jogging, harmonics are introduced due to the repetitive nature of activities which give an indication of speed of motion. The variations for RSS were 3dB for a motionless user, 15 dB for all activities. The mean power loss varies widely for jogging and rowing and not for cycling while the standard deviation varies most for cycling. Using a combination of data from various channels, the relevant activity can be determined using a statistical model Rayleigh distribution which pioneers an alternate method of noninvasive physiological monitoring using radio propagation channels.

2.4.2 Bayes Classifier

[Tianxiang et al. 2013] propose a technique to identify whether a particular gait cycle belongs to the motion of a human being by using Inertial Measurement Units (IMU) and discriminant analysis using Bayes classifier. The technique also addresses issues like speed of motion, variations in the placement of sensors and differences in the direction of motion i.e. curved vs straight path. This proposed technique involves an auto-segmentation phase, feature extraction phase and a training and classification phase. It is observed from the experiments that individuals gait movement are identified with an accuracy of 99.3% using this method for different speeds, motion directions and asymmetric foot movements. The authors conclude that this asymmetric feature-based auto-segmentation method is a robust, efficient and light method to detect human motion.

2.4.3 Zero Vector Crossing

[Lin and Kulic, 2012] propose a novel approach for automatic segmentation of human motion using a two stage process consisting of identification of motion segment candidates by Zero Velocity Crossings (ZVC) and Zero velocity peaks (ZVP) and recognition of segment locations by a Hidden Markov Model. The requirements of such an algorithm like ability to take into account spatial and temporal variability, give real-time feedback and being computationally light are discussed. During the training phase, a feature template is constructed for each Degree of Freedom (DoF) by analyzing the ZVCs and peaks between two consecutive ZVCs to find out whether a given peak is a characteristic feature or not. The HMM template is constructed using Baum-Welch algorithm by creating an 8-state left-right model and determining a threshold using exemplars. During the segmentation phase, a potential segment point is said to be located if there is a sequence of peaks and ZVCs matching the known template. For comparison with other techniques, ZVC only, fixed window HMM and manual segmentation were also implemented and parameters like correct, false positive and false negative was computed for each of the methods. The results show that although the template construction takes maximum time for a Feature based HMM using ZVC, the segmentation time and false positive is very low over fixed HMM while the accuracy is higher at 89methods.

2.4.4 Dynamic time warping

[Fernandez et al. 2013] The authors propose an effective, affordable and simplistic method for human segmentation and labelling using principal component analysis to reduce input data dimensions, labelling using machine learning techniques and clustering to refine results. The main goal is to identify the key exercise poses within captured motion data and label them with good accuracy and performance. The proposed approach consist of two phases i.e. training phase during which the system learns the features of training data using frame-wise feature extraction and testing phase in which body poses, also called landmark candidates, are discovered within individual motion cycles which correspond to learnt body poses also known as ground-truth landmarks. It is concluded from the results that this method has better precision of 90% as compared to 48% by DTW and 14% by

HACA. DTA has higher accuracy of 98% as compared to the proposed method which has an accuracy of 81% mainly due to the higher sensitivity of DTW to symmetry of motion data.

2.4.5 Rule Based Techniques

[Wenbing et al. 2014] propose a reusable, customizable, lightweight and robust rule-based real-time motion assessment and feedback system for rehabilitation exercises using motion sensors. Rule-based matching would be defined by a rule set, and hence is less computationally intensive, scalable and hence can provide real time feedback. Exercise movements are defined in terms of sequence of monotonic segments which is a sequence of configurations where the joint angles are either non-decreasing or non-increasing. Three types of rules which defines different types of movements, i.e. rules for dynamic movement in which each rule is created in terms of two or more reference configurations of joint segments like joint angles, rules for static poses which defines minimum and maximum allowed angle for a joint or distance between different joints and rules for movement invariance which defines the relative angle between the body plane and the moving body segment. These set of rules are then encoded in the form of XML with exercise name, their rules and the individual configurations like delimiting joint names, joint angles, and maximum and minimum exercise duration for each rule. The assessment for the static and invariance rules are straight forward as they contain parameters which are available with every frame through calculation of joint angle, distance and body segment orientation. For dynamic rules, each rule is assessed using a finite state machine.

2.4.6 Motion rehabilitation techniques with visual feedback

[Khan et al. 2013] proposes and implements a single tri-axial accelerometer sensor based approach for physical activity recognition using a lower level state recognition scheme based on artificial-neural nets (ANN) and statistical signal features and an upper level activity recognition scheme based on derived augmented feature vectors and linear discriminant analysis.Using the state and acceleration signal, the augmented feature vector is constructed using Autoregressive coefficients, Signal Magnitude Area and Tilt Angle, Linear discriminant analysis is performed and using Artificial-neural nets ANN, the activity can be classified.The evaluation consisted of tests for State recognition, feature augmentation and hierarchical AR (HAR) system. For HAR, a comparative study between Single level and HAL shows average accuracy at 71.6 and 97.9 respectively. The main conclusions of this paper are that it proves that using a single accelerometer and a two level hierarchical state and activity recognition scheme, the activity of the individual can be recognized accurately and robustly. The author claims that the performance of such a system was comparable to previously built systems with more accelerometers and single level recognition systems.

2.4.7 MultiLabel Classification

[Taylor et al. 2012] propose a method to assess the quality of human motion during rehabilitation by detecting errors with the help of various machine learning techniques like Binary Relevance (BR) Algorithm using AdaBoost classifier with Classifier Chain model and RakEL using Label Powerset method and decision trees. From the evaluation, it is found that RakEL has a specificity of 99% and sensitivity of 84% when compared to BR which has a specificity of 98% and sensitivity of 78%. This study proves that a strong multi-label classification method can accurately classify not only correct exercises techniques but also multiple error classes and give feedback to the person reducing the costs for personal home care.

2.5 Movement Reconstruction using Adaptive human motion reconstruction (AMR)

This section discusses a technique to perform 3D human motion reconstruction using Inverse Kinematics.[Shih-Yeh et al. 2011] proposes a 3D human motion reconstruction system utilizing multiple tri-axial accelerometers and gyroscopes to measure limb motion and locations as well as reconstruction using a 3D Adaptive human Motion Reconstruction (AMR) algorithm with a Body Correction algorithm (BCA). The joint movements and limb locations are calculated and modelled using kinematic theory i.e. Forward Kinematics and Inverse Kinematics to calculate limb motion. The 3d skeleton model is reconstructed using 3DS Max Software. The next step is to apply BCA to compute the actual limb proportions.The results show that there is a 6% difference in accuracy with and without BCA and that this method shows an average accuracy of 95.72%.

2.6 Quaternions

A quaternion is a vector having four components i.e. an element of 4D space used to represent orientation of a body or coordinate frames in a 3-D space. The orientation of frame B relative to frame A can be achieved by rotation of angle θ around an axis defined in frame A [Hanson, 2007]. They are denoted by:

$$\begin{aligned} q &= a + bi + cj + dk \\ where \quad i^2 &= j^2 = k^2 = -1 \\ ij &= jk = -ji \\ a, b, c, d &\sum R \end{aligned} \tag{2.4}$$

Unit quaternions or versors, provide a practical notation for translations and rotations of 3D objects. They follow the below constraints:

$$a^2 + b^2 + c^2 + d^2 = 1 \tag{2.5}$$

The quaternions (a,b,c,d) represents a rotation around the axis which is directed by vector (a,b,c) by an angle of:

$$\alpha = 2\cos^{-1} a = 2\sin^{-1} \sqrt{b^2 + c^2 + d^2}. \tag{2.6}$$

The following equation shows how a rotation can be achieved by multiplying the quaternion with a vector:

$$\begin{bmatrix} a^2 + b^2 - c^2 - d^2 & 2bc - 2ad & 2ac + 2bd \\ 2bc + 2ad & a^2 - b^2 + c^2 - d^2 & 2cd - 2ab \\ 2bd - 2ac & 2ab + 2cd & a^2 - b^2 - c^2 + d^2 \end{bmatrix} \tag{2.7}$$

Orthogonal rotation matrices consume 9 numbers while a quaternions are more compact in the information they contain and consume only 4 numbers.The conversion between rotation i.e. axis and angle to quaternion and vice versa is very easy to achieve.Quaternions also do not have a phenomenon called Gimbal lock which normally occurs with Euler angle. If one angle pitch is rotated 90 up or down, resulting int yaw or roll corresponding to the same motion, a degree of freedom of rotation is lost and then rotation can take place in 2D only.

2.7 Summary

This Chapter gives a detailed background information about eHealth and various implementations on Wireless Body Sensor Networks used to implement eHealth and AmI. The review discusses a wide spectrum of techniques and frameworks and their evaluations using different methods. This in-depth research also provides a good understanding of how the current implementations work and possible improvements. In Section 2.2, we have discussed various techniques for movement tracking using IMU and also their drawbacks like the lack of an easily adaptable standard in terms of protocols and communication technologies. These systems also also have the drawbacks of having high power consumption and being obtrusive. Section 2.3 deals with different communication technologies delving deeper into Bluetooth Low energy, constraints related with other protocols and time synchronization methods for BLE. It also discusses real time data streaming using BLE. This section provides an understanding of the underlying difference between various technologies and the reason as to why BLE is selected as the choice for real-time motion sensing. Finally, Section 2.4 deals with the various methods that remote and automatic monitoring of rehabilitation for patients can be performed. [Buonocunto and Marinoni 2014; Laine et al. 2014; Young-Jin and Hui-Sup 2013] provide a good understanding on how to use IMU's to generate motion sensor data while [Shih-Yeh et al. 2011; Wenbing et al. 2014] and Quaternions present methods to use this data for remote rehabilitation. Based on the drawbacks identified within these techniques, a MoBiLE system has been proposed to realize our research objectives.

Chapter 3

Design

Wireless Body Sensor Networks for medical applications typically work on coin cell batteries that should last for atleast weeks or months. They should not hinder the normal activities of an individual and they should be economical.

3.1 Requirements

Keeping all these points in mind and previous research as discussed in the previous chapter, the requirements of the MoBiLE system for real-time motion data streaming and reconstruction are as follows:

1. The GATT Service should contain a characteristic with motion sensor data packets.
2. These packets can be available in different formats and controlled by a control flag.
3. The GATT Service should transmit data packets only when necessary in order to conserve bandwidth, energy and enhance battery life.
4. The functions of the service itself like starting or stopping it should be controllable by the Client.
5. The Client should be able to receive data at different frame rates.

6. The Client should be able to connect to more than one device at a time and have minimal data and packet loss.

7. The system should be built with modules with clear separation of functionalities among them.

8. The GUI can be built so that the Client can interface with the Service with ease.

9. The implementation should be economical and unobtrusive to the end user. It should also be implemented with readily available technologies so that it can be easily integrable in day to day life.

3.2 System Architecture

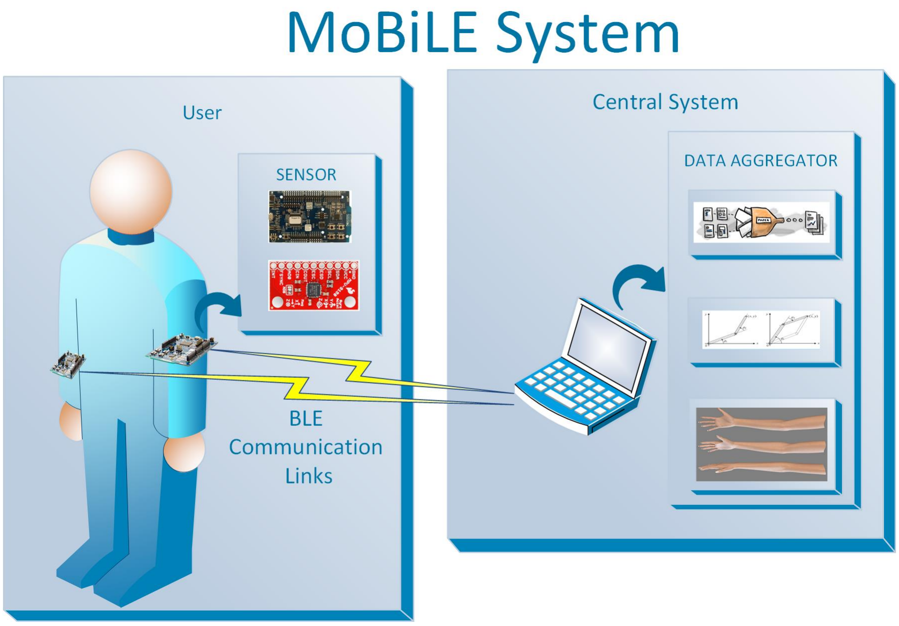

Figure 3.1: System Architecture

Fig. 3.1 depicts the functional requirements of the MoBiLE system. This system consists of two main modules:

1. **User Module:** This module consists of two parts: the motion sensor along with a transmitter which can be embedded onto one device and can be unobtrusively worn by the User while performing daily exercises.

2. **Central System Module:** This module consists of three main parts:-

   (a) **Data Receiver :** A client front end application that can receive data via a wireless communication protocol

   (b) **Data Aggregator:** A Data aggregator that can collect data from multiple motion sensors

   (c) **Augmented View Renderer:** This module can then use the aggregated data and create 3D human motion animation to facilitate remote monitoring.

3.3 MPU GATT Service

Fig. 3.2 illustrates the Motion Capture Profile which exposes the MPU GATT Service and Device Information Service running on a server. This is a handle by handle description of the different characteristics designed for use for the MPU GATT Service. There are four main characteristics for the MPU GATT Service:

1. **Sensor Value Characteristic:** The main characteristic which holds motion data values. A Client can read data from this characteristic or receive notifications i.e. data pushed from the server to the client. To enable notifications, a value of 0x01 would need to be written to the Client Characteristic Configuration Descriptor (CCCD) with handle 0x0F on the peripheral device.

2. **Body Location Characteristic:** This indicates the location of the sensor on the body. The sensor is attachable on different locations of the body like chest, wrist, fingers, arm, hand and foot. This characteristic enables the medical professionals to understand which part of the body is performing which type of exercise and give feedback accordingly. It is a read only characteristic

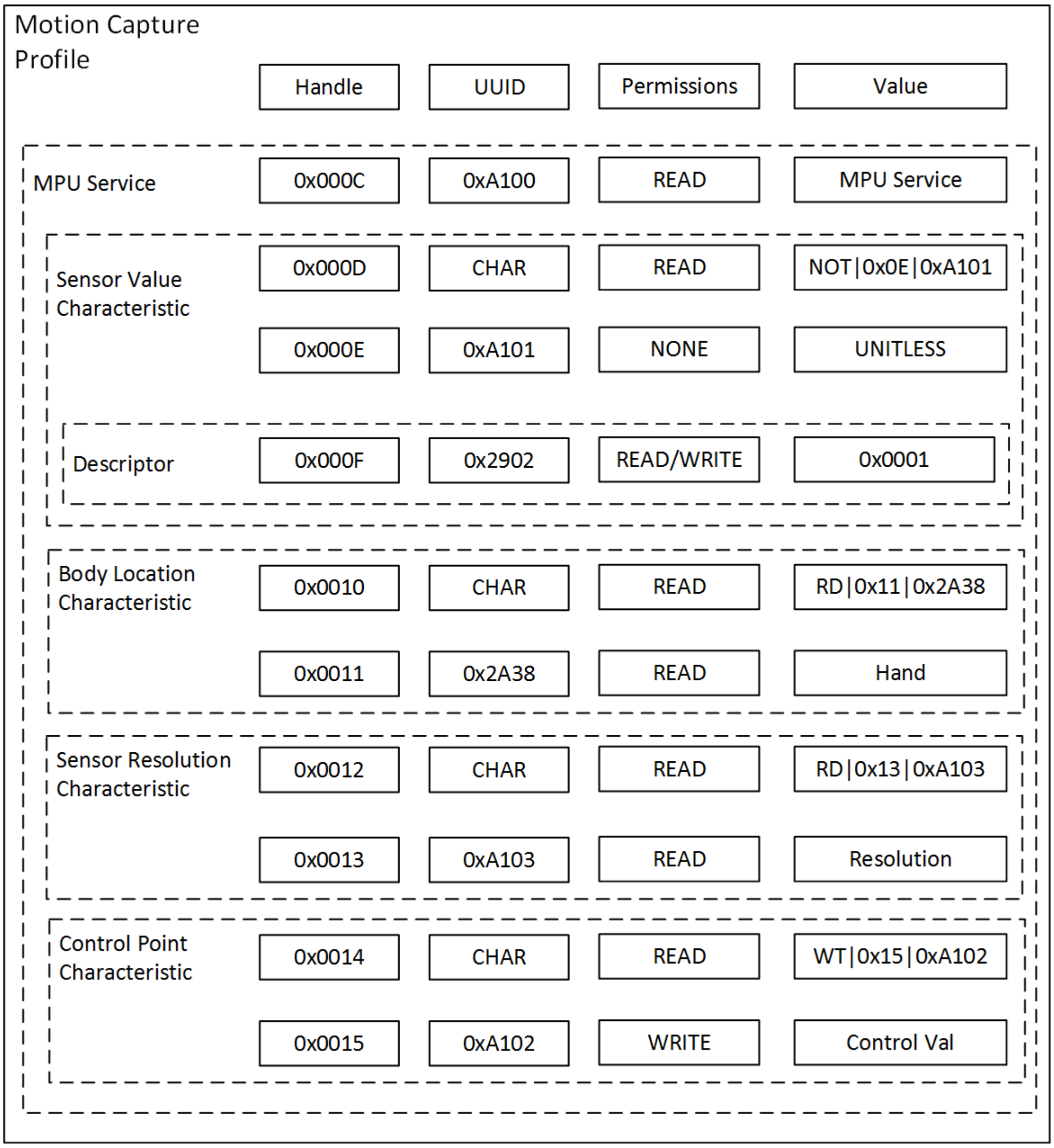

Figure 3.2: Designed MPU GATT Service

3. **Sensor Resolution Characteristic:** This is required for calculations on the Client side. It enables the client to understand what resolution is used for the accelerometer, gyroscope and magnetometer as different types of sensors have different ranges of resolution and hence this can be determined while the MPU Server is active. It is a read only characteristic

4. **Control Point Characteristic:** This is another important characteristic which is a write only characteristic. It is used to control the device and configure modes of operation.

3.3.1 Modes of operation and Control Point

Control Point Characteristic

Names	Field Requirement	Format	Min Value	Max Value	Additional Information		
					Key	**Value**	
MPU Sensor Control	Mandatory	8bit	N/A	N/A	0	1	Stop MPU Service
					1	0	Set resolution to 16 bits
						1	Set resolution to 8 bits
					2	0	Set frame rate to 30 FPS
						1	Set frame rate to 15 FPS
					3:4	00	Raw Data
						01	Differential Data
						10	Processed Data
						11	DMP Data
					5:7	000	Normal Mode
						001	Acc and Gyro
						010	Acc only
						xxx	Reserved

Table 3.1: MPU Control Point Characteristic

The control point characteristic is the Single Point of Contact for controlling the MPU Service using any GATT client like the Master Emulator Android app or using the various GATT clients discussed later. It is the only write characteristic and allows the central to have a high level of control on the rate of flow of the data stream, type of data and the ability to start/stop the service. It allows Central devices to remote control or fine tune the services provided by the MPU Service.

Table 3.1 describes the packet structure of the MPU Control point in a BLE GATT Specification Standard. Bit 0 is used to stop the MPU Service. Bit 1 is to set 16 bit or 8 bit format. Bit 2 is used to toggle the frame rate between 30 and 15 frames per second. Bits 3,4 are used to control the type of data. bits 5:7 are used to control the mode of operation. The format of the Flag field transmitted as the first byte within the BLE packet follows the same format with one bit shift to the right as the stop MPU Service is not required.

Packet Structures

This section discusses the 6 modes of operation for the MPU Service as illustrated with the below tables and also the control point values associated with them.

Case 1 : 16 bit raw sensor Data (9DOF)

This mode is used to send 6 degrees - accelerometer and gyroscope data in 16 bit format and 3 degrees - magnetometer data in 8bit format. The Control point characteristic required to be written to the GATT Service is 0x00.

(x,y)	0	1	2	3	4	5	6	7	8	9
0	Flags	Seq No.(H)	Seq No.(L)	Acc_x (H)	Acc_x (L)	Acc_y (H)	Acc_y (L)	Acc_z (H)	Acc_z (L)	Gyr_x (H)
1	Gyr_x (L)	Gyr_y (H)	Gyr_y (L)	Gyr_z (H)	Gyr_z (L)	Mag_x (H)	Mag_y (H)	Mag_z (H)		

Table 3.2: 16 bit raw sensor Data

Case 2 : 8 bit raw sensor Data

This mode is used to send all 9 DOF raw data in 8 bit format. The Control point characteristic required to be written to the GATT Service is 0x02.

(x,y)	0	1	2	3	4	5	6	7	8	9
0	Flags	Seq No.(H)	Seq No.(L)	Acc_x	Acc_y	Acc_z	Gyr_x	Gyr_y	Gyr_z	Mag_x
1	Mag_y	Mag_z								

Table 3.3: 8 bit raw sensor Data

Case 3 : 16 bit Processed DMP Data

This mode is used to send all DMP quaternion data in 16 bit format. The Control point characteristic required to be written to the GATT Service is 0x18.

(x,y)	0	1	2	3	4	5	6	7	8	9
0	Flags	Seq No.(H)	Seq No.(L)	$Quat_x$ (H)	$Quat_x$ (L)	$Quat_y$ (H)	$Quat_y$ (L)	$Quat_z$ (H)	$Quat_z$ (L)	$Quat_w$ (H)
1	$Quat_w$ (L)									

Table 3.4: 16 bit Processed DMP Data

Case 4 : 8 bit Differential sensor Data

This mode is used to send 8 bit differential quaternion data. The Control point characteristic required to be written to the GATT Service is 0x12. This mode is used only for very slow changes in sensor values as it is not able to react to large changes in sensor values.

(x,y)	0	1	2	3	4	5	6	7	8	9
0	Flags	Seq No.(H)	Seq No.(L)	$Quat_x$	$Quat_y$	$Quat_z$	$Quat_w$			

Table 3.5: 8 bit Differential sensor Data

Case 5 : Power Saving Mode for Quaternion Data PEDA Algorithm

This mode is used to send all different quaternion data in a variable bit format using the PEDA algorithm. The size of the packets are variable from 5-12 bytes. The bitmap flag is a flag in which the first four bits (LSB) indicate the presence of the differential Quaternion value while the higher four bits (MSB) indicate whether the differential data is 8 or 16 bit. The Control point characteristic required to be written to the GATT Service is 0x4A.

(x,y)	0	1	2	3	4-12
0	Flags	Seq No.(H)	Seq No.(L)	Bitmap	Variable

Table 3.6: Power Saving Mode for Quaternion Data PEDA Algorithm

Case 6 : Accelerometer and Gyroscope data(6 DOF)

This mode is used to send 6DOF data in a 16 bit format to the MPU Client. The Control point characteristic required to be written to the GATT Service is 0x80.

(x,y)	0	1	2	3	4	5	6	7	8	9
0	Flags	Seq No.(H)	Seq No.(L)	Acc_x (H)	Acc_x (L)	Acc_y (H)	Acc_y (L)	Acc_z (H)	Acc_z (L)	Gyr_x (H)
1	Gyr_x (L)	Gyr_y (H)	Gyr_y (L)	Gyr_z (H)	Gyr_z (L)					

Table 3.7: 16 bit raw sensor Data 6DOF

Detailed specifications of the implementation for the MPU Gatt Service is given in Table B.4 and Table B.5. These contain detailed definitions of characteristics of the GATT Service and the different configurable modes and their properties.

3.4 Onboard Algorithms

3.4.1 Power Efficient Data Adaptive (PEDA) Algorithm

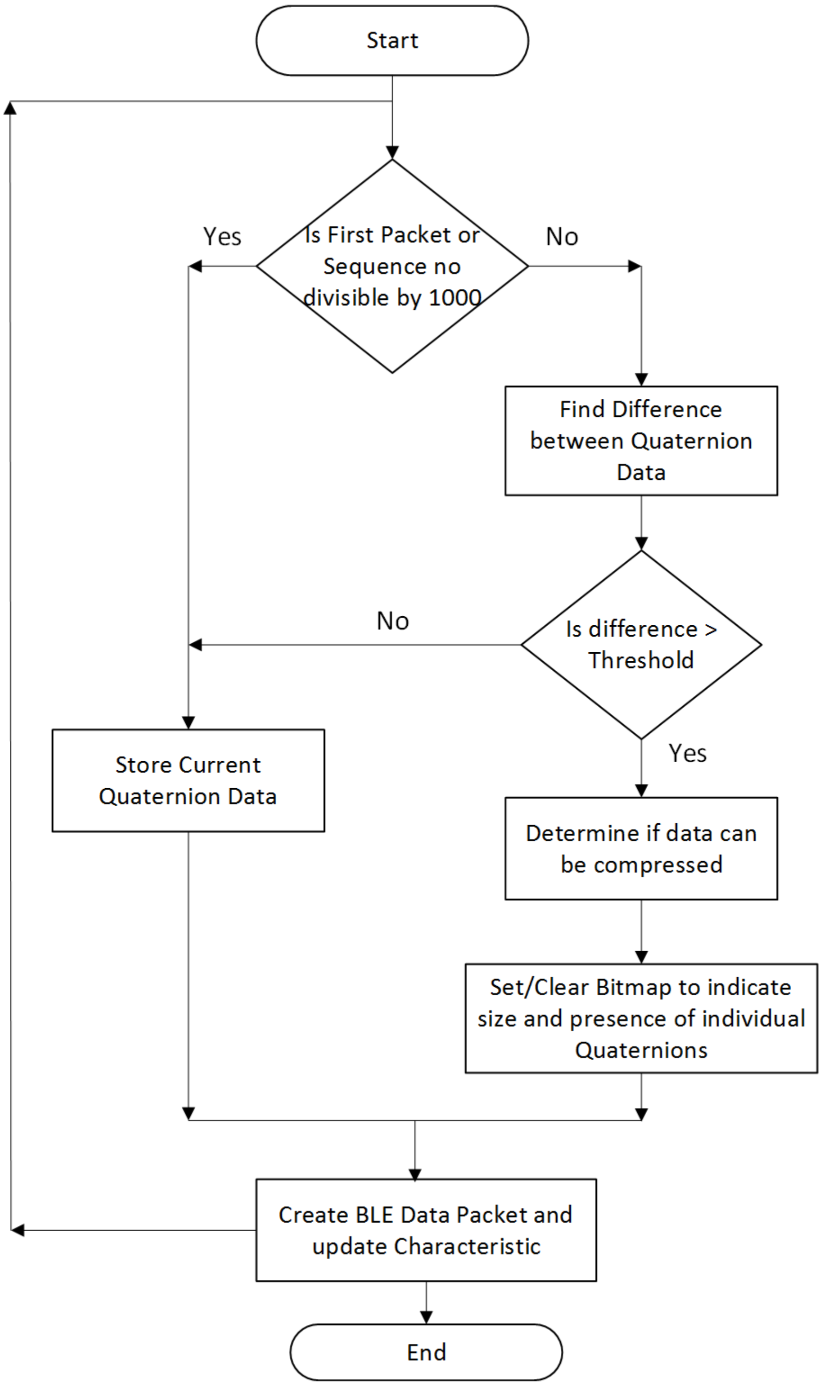

Figure 3.3: PEDA Algorithm

Fig. 3.3 illustrates a novel differential algorithm which aims to reduce the number of packets sent per connection interval by comparing the difference in the consecutive quaternion values against a predefined threshold. The algorithm has the following steps:

- The first data packet or one with a sequence number divisible by 1000 is sent without any changes during the connection.
- The subsequent quaternion data values are then compared with the previous values.
- If the difference is above some predefined threshold, a function is applied to try to convert the data into an 8 bit value if possible. This procedure is followed for every quaternion value. If the difference is more then what a 8bit value can contain, the difference is sent as a 16 bit value.
- The Bitmap Flag is then used to indicate the presence of a Quaternion value and its resolution i.e. 8 or 16 bits.
- The resultant quaternion bytes are then packed into a data frame and the MPU Sensor Value characteristic is updated with this frame.

3.4.2 Motion Fusion Algorithm

Motion Fusion combines output from three sensors to produce data like quaternions, rotation matrices, gravity and euler angles which can be used for animation. The algorithm has the following steps:

1. First the 9 DOF values i.e. accelerometer, gyroscope and magnetometer values are normalized.
2. The orientation of the earth frame with respect to the sensor frame at a particular time t can be computed by numerically integrating the quaternion derivative.

3. The accelerometer measure's the magnitude and direction of the gravity in the sensor frame along with linear accelerations. Likewise, the magnetometer will measure the magnitude and direction of the earths magnetic field in the sensor frame.
4. Gradient decent algorithm is then used to compute an orientation estimation using a Jacobian function for a field in any direction.
5. Compute rate of change of quaternion .
6. Perform filter fusion by integrating all the measured values to yield quaternion.
7. Perform gyroscope and magnetometer bias drift compensation that could occur due to temperature and motion.

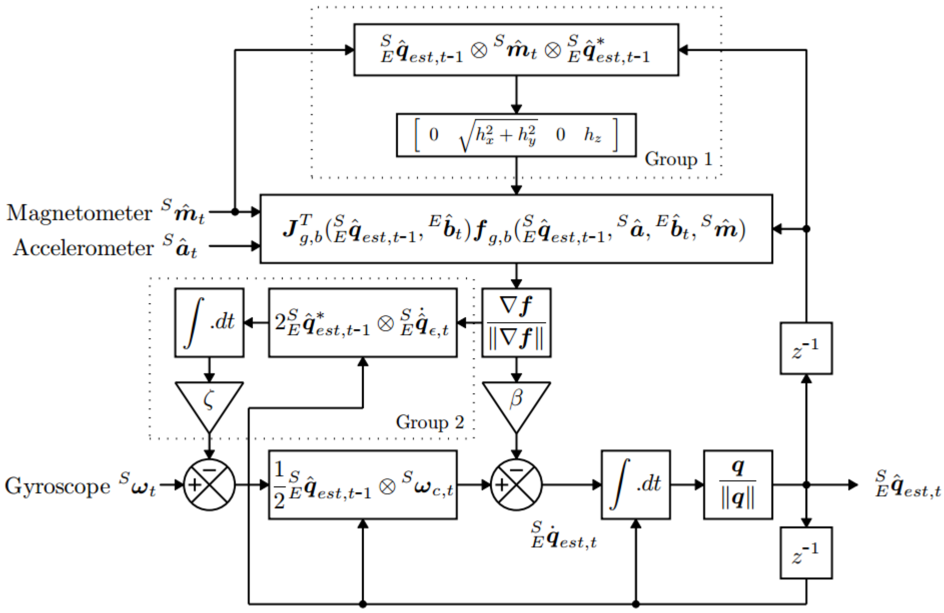

Figure 3.4: Block diagram representation of the complete orientation filter [Madgwick]

3.5 Summary

This Chapter provides high level design specifications for the implementation of the MoBiLE system. The main building blocks of this system are discussed along with the functions and requirements of each block. The blueprint for the GATT Service for transmitting real-time motion sensing data is also proposed which contains details about various characteristics like Sensor Data, Body Sensor Location and their modes of access. The different modes of operation of this Service are also discussed. The design of a generic GATT Client that can interact with this service is also discussed .Lastly, motion fusion and differential data algorithms are discussed which can be used to conserve bandwidth. The next Chapter describes a prototype implementation of the proposed design which will then be used in later chapters to evaluate the operation and performance of the proposed approach.

Chapter 4

Prototype Implementation

4.1 System Architecture

Fig. 4.1 shows a basic flow of the system built to implement an end to end flow of a motion capture system using Bluetooth Low energy while taking into consideration different constraints like low power consumption, reliable communication and real time streaming of data between a Motion Processing Unit and a BLE enabled device.

1. The first step is to configure the MPU 9150 device to acquire sensor data like accelerometer, gyroscope and magnetometer (9 DOF) or process it into Quaternions using a Digital Motion Processor.

2. Initialize the MPU GATT service on the nRF51DK peripheral and transmit data via Bluetooth low energy to the Central BLE Client via BLE at 30/15 frames per second by using a ticker or callback function.

3. On the Client side, aggregate the data received from the two sensors using the MPU GATT Client implemented in Python with a NRF51 Dongle, process this data and send it via a Socket connection to the 3D Human motion renderer.

4. Animate two human hands in real-time using the data received via the socket connection in Processing. Processing is an language written in Java and uses OpenGL.

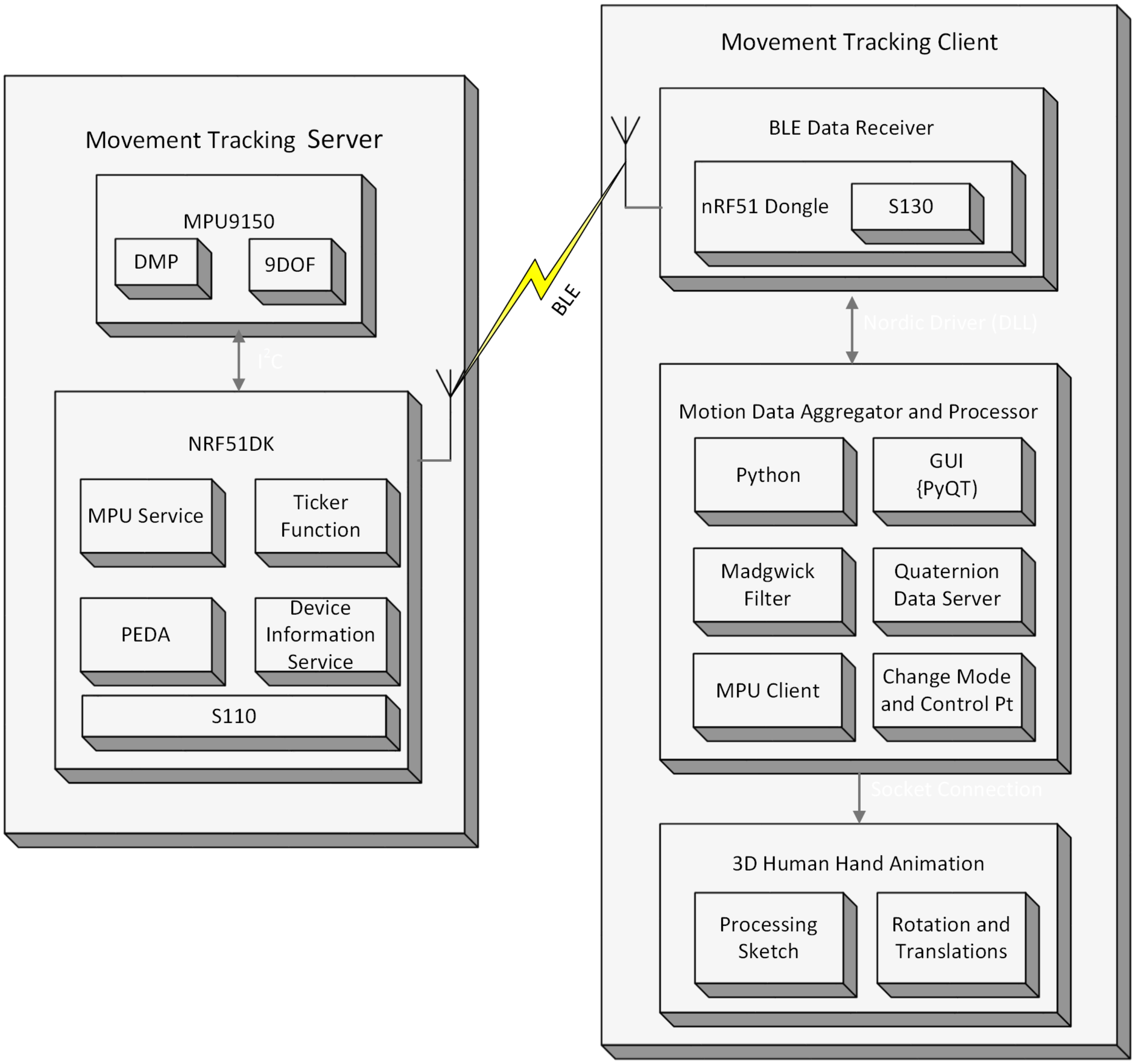

Figure 4.1: System Architecture

4.2 Hardware Devices

4.2.1 Nordic Device

The Nordic devices [Nordic] consist of SoftDevices which are precompiled and binary hex files implementing the BLE protocol stack for the nRF51 devices.The devices contain a Application Programming Interface (API) which is a coding bundle written in C that provides a high level of abstraction to the application from the SoftDevice Implementation. The SoftDevice also allows the developer to program their code as a ARM Cortex project without worrying about the vendor specific BLE implementation. Fig. 4.2 shows the System on Chip application with the SoftDevice and Fig. 4.3 shows the BLE Energy stack. As the SoftDevice is a precompiled hex file, application developers provided with run time isolation and the system would have a deterministic behavior. A Softdevice consists of 3 main modules:

- **SoC Library:** This library provides API for managing hardware resources.
- **SoftDevice Manager:** API for managing the state of the SoftDevice. Can be used even when the SoftDevice is disabled.
- **Protocol Stack:** Wireless protocol stack which provides abstract control of the RF transceiver suited for wireless applications. In addition, it also has an nRF API to support both the SoftDevice manager and the SoC library.

Standard BLE profiles like Heart Rate Monitor as well as other proprietary use cases can be implemented using an abstract and flexible interface with the SoftDevice. The API sits on the Generic Attribute Protocol (GATT), Generic Access Profile (GAP), Logical Link Control and Adaptation Protocol (L2CAP). The nRF51DK implements all Service and Profiles using the SoftDevice. The SoftDevice is implemented using thread-safe Supervisor Calls (SVC) and all interactions with this stack are event driven and asynchronous. Some of the supported profiles are: HID over GATT, Health Thermometer, Proximity and Heart Rate.

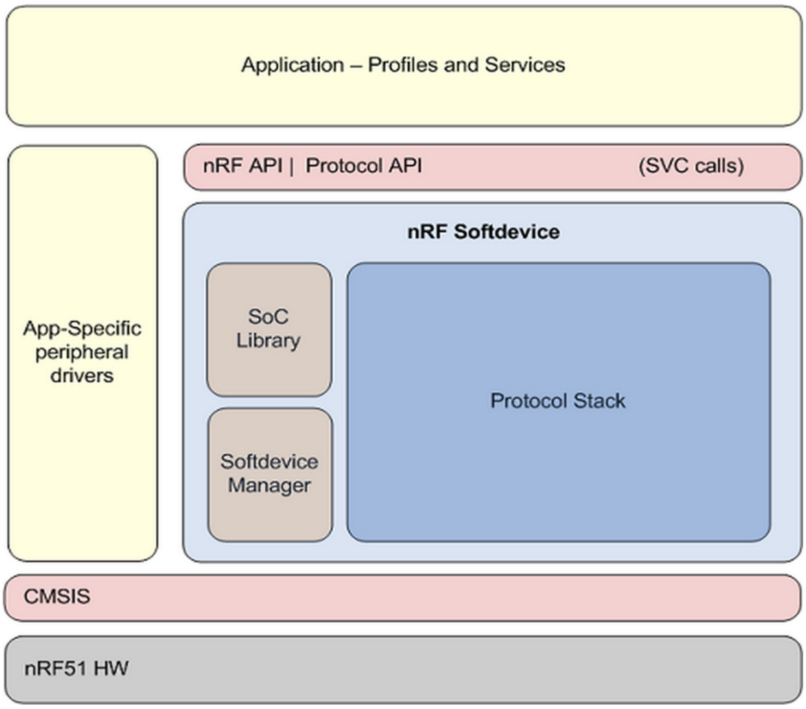

Figure 4.2: SoC Application with SoftDevice [Nordic]

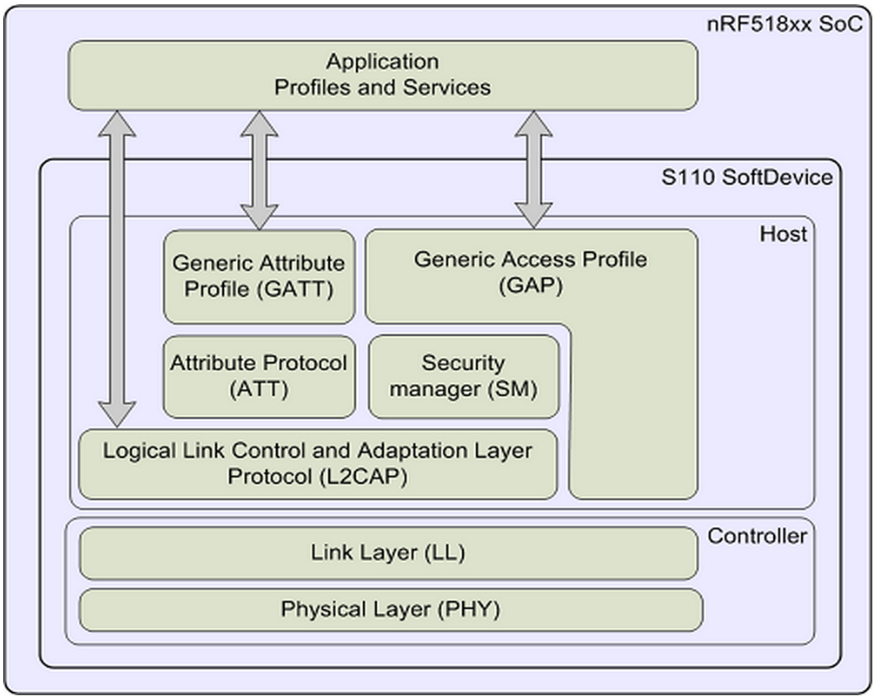

Figure 4.3: BLE Energy stack [Nordic]

The program and data memory and peripherals are sandboxed to prevent SoftDevice program corruption by the application ensuring robust and predictable performance. The RAM and Program Memory each are divided into two regions using registers for the SoftDevice and application.

Soft Device	Description
S110	Can support Peripheral and Broadcaster Roles concurrently.
S120	Bluetooth low energy Central protocol stack solution supporting up to eight simultaneous Central role connections.
S130	Can support up to 3 central, 1 Peripheral, Observer and Broadcaster roles concurrently on one device.

Table 4.1: Types of SoftDevices

Fig. 4.4 compares the dimensions of the nRF51DK (101 x 63 mm) , nRF51 Dongle (16 x 28 mm) and a €2 coin (25 x 25 mm).

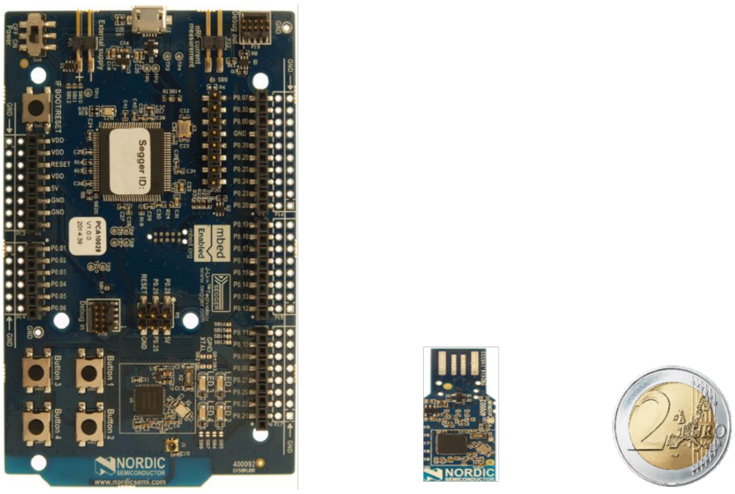

Figure 4.4: Size comparison between nRF51DK, nRF51 Dongle and a €2 coin

4.2.2 Two Wire Interface - nRF51DK

The two-wire interface driver for the nRF51DK consists of two layers:

1. Hardware Abstraction Layer : Basic APIs to access registers of TWI
2. Driver Layer : Higher level APIs to use TWI.

The TWI connection has a 7-bit addressing mode.The nRF51DK (microcontroller) is connected to MPU9150 IMU Sensor via Vcc,GND, SDA and SCL Line.

4.2.3 MPU9150

MPU-9150 [Sparkfun] by InvenSense is the world's first 9DOF Motion Tracking device (MEMS) and a 3^{rd} generation motion processor which is designed for low cost, power and higher performance. It is used in a plethora of consumer electronics like wearable sensors, tablets, game controllers, and smartphones. It combines two chips MPU6050, which contains a 3-axis accelerometer and 3-axis gyroscope and an onboard Digital Motion Processor (DMP) which can process complex Motion Fusion and gesture recognition algorithms and a 3-axis magnetometer, AK8975. The 16-bit accelerometer output has a programmable full scale range of $\pm 2g$,$\pm 4g$,$\pm 8g$ and $\pm 16g$. The 16-bit gyroscope output has full scale range of $\pm 200dps$,$\pm 500dps$,$\pm 1000dps$ and $\pm 2000dps$.The 13-bit magnetometer output has a full scale range of $\pm 1200\mu$T. The device is capable of tracking both slow and fast motions. It has a 1024 byte onboard FIFO buffer which enables the system processor to read the sensor output in bursts and then enter a low power mode thereby reducing power consumption.

I^2C connection

The breakout board designed by SparkFun using the MPU9150 provides a ready to use I^2C pullup resistors and addressing. This I^2C port is able to connect in a primary mode i.e. act as a slave or a auxillary mode i.e. act as a master to other devices like pressue sensors. The device communicates with the registers using I^2C at 400KHz. I^2C is a Two Wire Interface (TWI) consisting of Signals Serial Data(SDA) and Serial Clock (SCL). The MPU 9150 is works as a slave device while communications with the nRF51DK. The slave address of the MPU6050 can be 0x68/69 depending on the logic level of pin AD0 allowing two MPU9150 to be connected on the same line. The address for the AK8975 is 0x0C.

The nRF51DK initiates communication via I^2C by sending the START condition. It then sends the address of MPU9150 followed by the 8^th bit to indicate read bit. The master then waits for an acknowledgment signal (ACK) from the slave. the MPU9150 can now start sending data to the master. If the master needs to write to the internal MPU9150 registers, it again transmits a start condition, followed by the I^2C address and a write bit. Following an ACK from the MPU, it will then put the register address on the bus and only then it is able to write to that register. All data transfer is terminated by the master device with the STOP condition.

4.3 Motion Processing Unit

The code for the MPU Service is written using the Mbed platform [ARM]. Mbed is a platform for IoT and BLE devices based on 32-bit ARM Cortex-M microcontrollers. Fig. 4.5 shows the algorithm implemented for the MPU GATT Service.

1. A ticker is initialized and setup as a callback function to periodically perform data acquisition from the MPU9150. This controls the frequency of notifications i.e. how often the nRF51DK would wake up from its sleep mode and push data to the Client. The device on which the MPU Service is running is assigned a name "Mordic" which is short for Motion sensing Using Nordic Devices.

2. The BLE parameters like min and max connection interval and slave latency are initialized to their optimal values .For this service, min connection interval is set to 7.5ms to get maximum throughput. Services like Device Information Service and MPU Service are configured to their operational values and attached to the BLE Device.

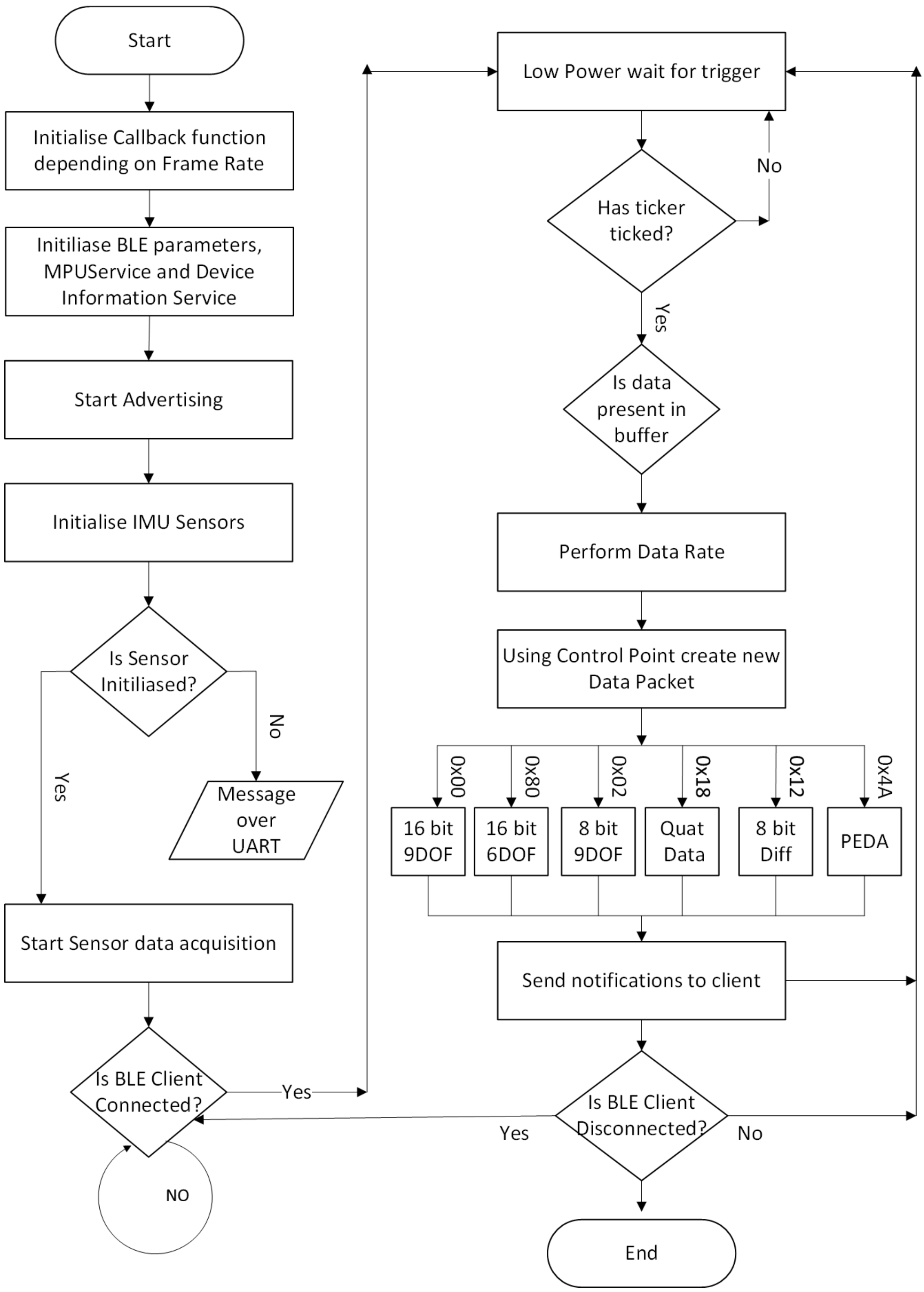

Figure 4.5: MPU Service

3. The MPU 9150 sensor is then initialized along with its Digital Motion Processor to

transmit data via I^2C to the microcontroller.

4. If the MPU9150 initialization is not successful, then it will display an error message on the console via UART. MPU9150 begins acquiring data, sending it via I^2C to the Nordic device which stores it in a buffer.

5. The BLE peripheral i.e. nRF51DK then starts advertising data.

6. The nRF51DK goes into a low power wait mode at this stage waiting for a connection from a central device.

7. Once the MPU Client connects to the MPU Service and enables notifications, the MPU Server sends data present in buffer to the Client.

8. The data packet format and content is dependent of the Flag and Control point characteristic which can be changed during a connection by the Central device. The control point characteristic format, different packet types and their formats are explained in section are based on Section 3.2.

4.4 Client-side processing

4.4.1 Motion Data Aggregator and Processor

Fig. 4.6 depicts how the MPU Client is designed to act as a data aggregator and process data and send it to Augmented Reality renderer. The MPU Client consists of two main parts: Hardware which is the nRF51 Dongle and Software which is acts as a master controller for the dongle. The software is written with Python, QT Framework and a nRF51 Driver (C++).The process follows the below steps:

1. The nRF51 Dongle is reset, connection parameters setup to receive data at 30fps and the socket connection is also initialized so that this MPU data aggregator module can be used as a socket server.

2. A Graphical User Interface is built with PyQT4 to control the MPU Service like starting/stopping notifications and modifying the control point characteristic.

3. On configuring the maximum number of peer devices that the central can connect to, the central continuously starts scanning for Mordic Devices. Once it detects Mordic devices, it can automatically connect to them.

4. It then performs service discovery, acquires a list of all characteristics belonging to the service along with their respective handles and also descriptors if any.

5. Notifications are then enabled for the Mordic devices so that real time streaming of sensor data can commence.

6. The received data is checked for packet loss and data rates.

7. The sensor is then decoded using the MPU Helper class which helps to convert data belongs to different modes(1-6) into a standard Quaternion format with a device number specific tag so that two independent data streams can be sent over a socket connection for animation.

8. The received data is also displayed in the GUI.

9. Enable/ Disable notifications is implemented by writing a 0/1 to the descriptor handle of the MPU Sensor data characteristic.

10. Control Point characteristic is changed by issuing a write command with a specific value to the MPU Service. The change is reflected instantly and can be verified by checking the flag field on the GUI.

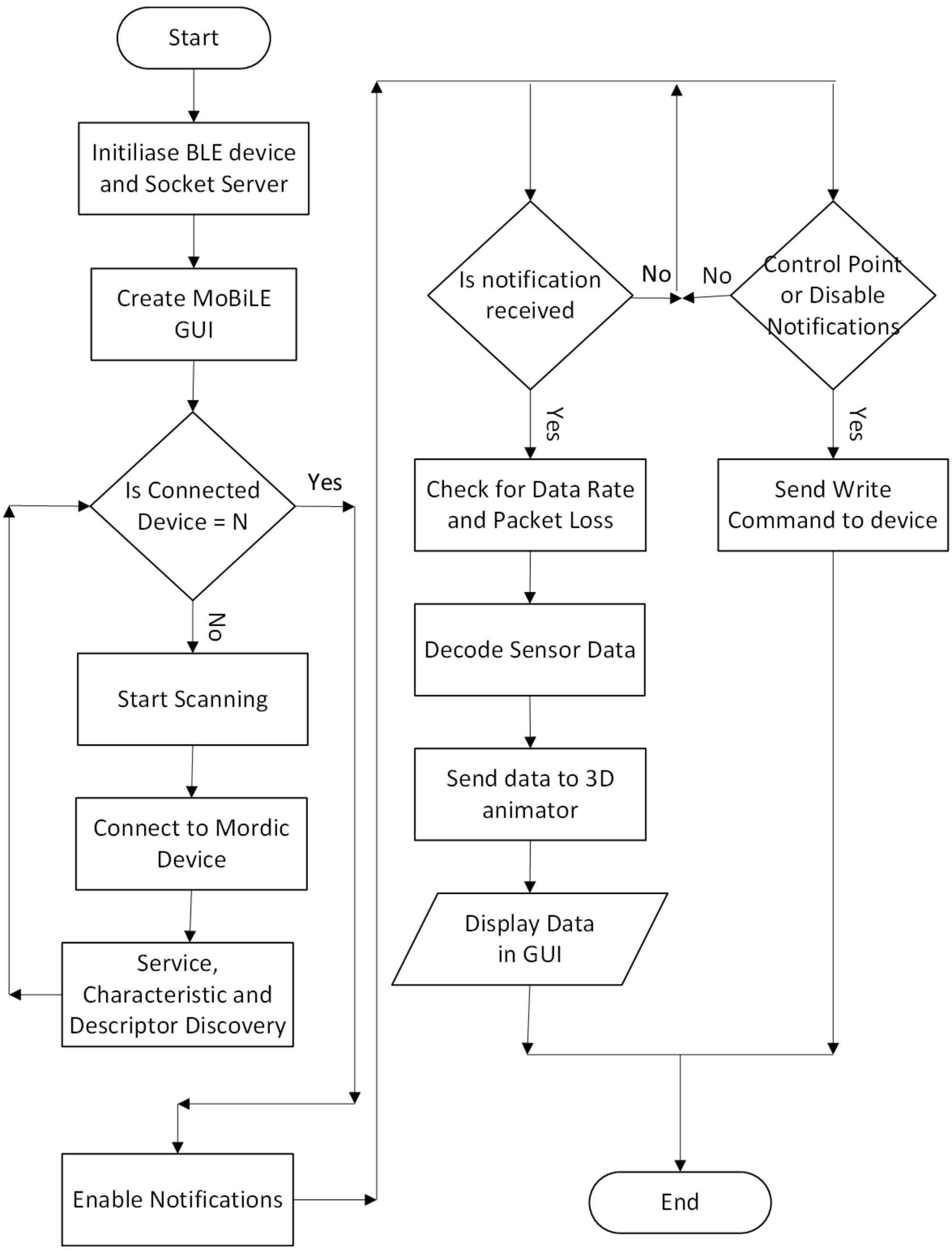

Figure 4.6: MPU Client

4.4.2 3D Human Hand Animation

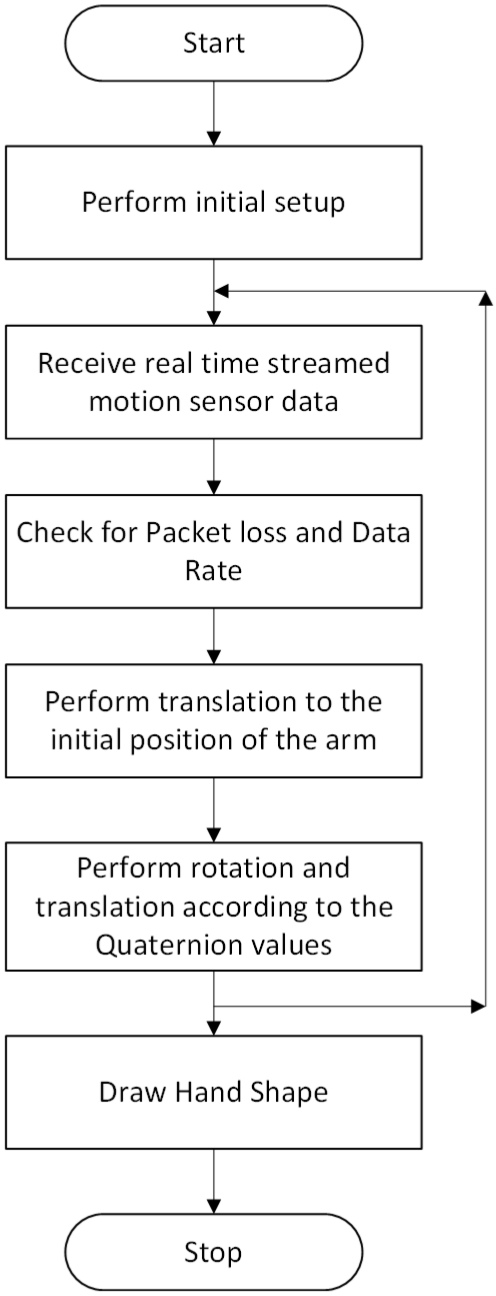

Figure 4.7: 3D Human Hand Animation

Fig. 4.7 describes how animation using real-time motion sensor data is implemented for this system. Processing [Fry and Reas] is a open source animation software in Java which is used to implement 3D human motion, socket client and data capture from UART for tests.

It makes use of sketchbooks as an IDE which consists sketches contains an initial setup() method and a run() method which is executed depending on the frame rate. Processing animation works like a stack wherein you push all elements onto a matrix(stack), rotate and translate only objects you need and then push them back from the matrix again. However, the object is not rotated in processing as it is the frame of reference for that object that is actually rotated.

- The sketch performs preliminary setups like configuring the socket client connection, initializing the shapes for both the arms.
- Once the connection has been established, data sent over the socket is then checked for packet loss and data rate to determine if there are any losses.
- All the current data is pushed back on a matrix, the frame for the hand is then rotated and translated according to the generated quaternions using a library called ToxicLibs. The hand is then drawn there. The matrix is popped back again with the values before the push. This is done to preserve the previous state of the animation.
- The above steps are then repeated for the second hand.
- The background is redrawn as the trace of the hand motion would otherwise be visible.

Fig. 4.7 illustrates how real-time sensor data is used by Processing to animate 3D human motion. The figures show a set of different hand movements i.e. vertical, circular and horizontal motions in real-time and their 3D rendering by the MoBiLE system. The system is able to process and render all the data with almost no latency.

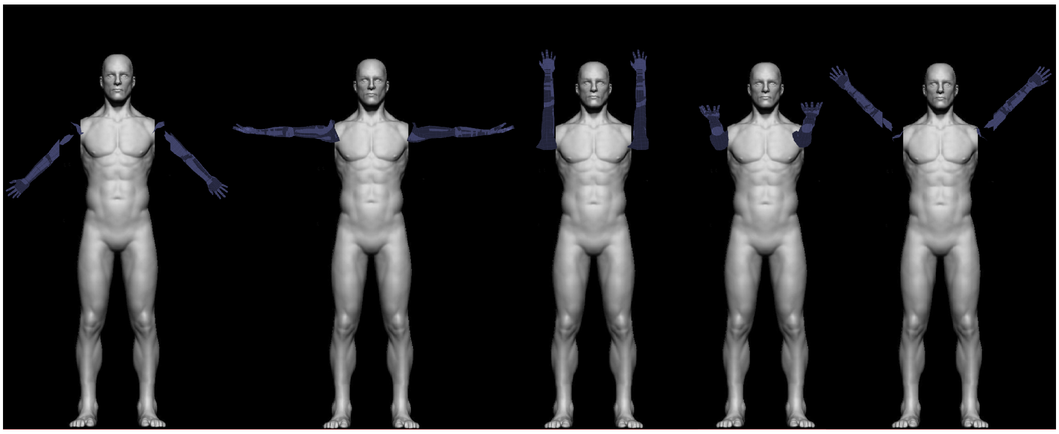

Figure 4.8: 3D Human Motion real time animation

4.5 Discussion

Bluetooth Low Energy(BLE) is an emerging technology which is ubiquitous as it is widely available in most mobile phones and laptops these days. However, it faces considerable issues as it is still in its nascent stages of development in terms of how the API for the BLE GATT Client has been developed. For the purpose of this project, we have investigated a variety of methods to implement the functionality of a MPU GATT Client. The requirements of this client is focused on two main points:

1. The Client should be able to receive data at 30 frames per second.
2. The Client should be able to connect to more then one device at one time and have minimal data and packet loss.

While evaluating these criteria, three hardware devices were considered:

- **Built-in BLE sensor in laptop with BlueZ:** This was the first choice for a BLE sensor as it is readily available in most laptops. Using the native BLE libraries provided by Windows, code was implemented with C# in order to connect with the Heart Rate Service. Although, the code could interact with the Heart Rate Service, Windows BLE APIs are not mature enough or customizable to interact with devices other then the standard BLE profile implemented by Windows. Linux was then chosen as the operating environment for this project as its BLE APIs are more mature and its native BLE API, BlueZ was selected to act as a Master Controller to operate the BLE sensor. Implemented in python, Bluez Gatt Client was able to control the sensor to interact with the MPU Service. However, high packet loss and low data rates as well as unstable connections made this an unsuitable option.

- *Bluetooth V4.0 Adapter Mini Dongle with BlueZ:* The second choice was to assess whether an external BLE Device i.e. a BLE Dongle coupled with BlueZ as the controller could be have a better performance than the inbuilt BLE sensor. The code written in python was used to evaluate this setup. It was found that although the connections were stable, the packet loss and data rates did not satisfy the requirements of this system.

- **NRF51 Dongle :** The final option was to evaluate whether the NRF51 Dongle, can be used to implement the BLE Client along with different software drivers. The following choices were made:

  - **NRF51 Dongle and Master Emulator API:** Code was written in C# to operate the dongle utilizing the Master Emulator API. Experiments showed that although there was zero packet loss and high data rates achievable, the API could support only a single connection at one time.

  - **NRF51 Dongle and Mbed API:** In July 2015, Mbed invited BLE developers to test out their experimental GATT Client API built on S130 stack. The Mbed GATT Client API was implemented in C++ and the dongle was able to connect to a maximum of 3 devices using the API code. However, the ability to control the device using a GUI and other such options were limited with this option.

– **NRF51 Dongle and Nordic Driver API:** Relatively newer (v0.5), with strong Python bindings, this is the preferred choice as it satisfies all the necessary requirements as stated above. Zero packet loss, high data rates, ability to have concurrent connections and the ability to modify characteristics dynamically on the GATT Service make this an ideal solution.

4.6 Summary

This Chapter specifies the implementation details of an end to end movement tracking system with augmented reality using MPU9150, NRF51DK, I^2C, BLE, Python, C++ and Processing 3D. The most challenging part of this implementation was to select and implement a MPU GATT Client that could connect simultaneously to more than device at once while maintaining constant data rates and zero packet loss as the GATT Client is still in its nascent stages of development. Another challenging aspect was to perform data acquisition using the IMU. The integration of individual components to create the MoBiLE system which could in real-time stream motion sensor data to produce 3D human motion provided a deeper understanding of the objectives of this thesis.

Chapter 5

Evaluation

Certain experiments have been conducted to evaluate the following metrics which give an indication about performance of the implemented prototype.

5.1 Experiment 1: Packet Size for different modes of operation

Aim: To determine the sizes of LE packet sizes for different modes of operation of the GATT Service.

Setup: The system needs to be configured using the following steps for this experiment:

- The nRF Bluetooth Sniffer is a debugging tool for Bluetooth low energy (BLE) applications which sniffs all packets between two communicating BLE Devices or from an advertising BLE peripheral over-the-air to solve any potential issues. It is able to detect payload data, Bluetooth address and RSSI.
- The nRF Sniffer firmware needs to be loaded on the BLE device which would act as the Sniffer.
- There is a command line utility ”ble-sniffer.exe” to setup the Sniffer, configure it to listen to all advertised packets or listen to a particular peripheral within a

connection. The packets are then captured by the utility along with a network protocol analyzer called Wireshark.

- The configured nRF sniffer needs to be placed in between the BLE central and peripheral to capture the data packets.

Procedure: The following steps are executed to evaluate this aim:

- Start the Sniffer utility and then start Wireshark by pressing W.
- In the Sniffer utility, Press 'l' to list all nearby devices.
- Select the BLE Peripheral whose packets are needed to be analyzed.
- Connect both the BLE central and peripheral.
- Analyze the data packets as they appear in Wireshark. Change the different modes of operation of the MPU Service using the GUI and determine how many packets are sent during different modes in best and worst case scenarios.

Observations : Graph 5.1 shows the comparison between the different modes of operation. It shows that Mode 1 i.e. raw 9 DOF data has the highest packet size of 21 bytes. Mode 3 i.e. Quaternion data consumes about 14 bytes of data as it constantly sends data even if the sensors are at rest. Although mode 4 has a low payload of only 10 bytes, the resolution is very poor for real-time motion data streaming. The PEDA algorithm i.e. Mode 5 has a variable packet size due to its differential characteristics which varies between 9-15 bytes which in normal scenarios outperforms all other modes in terms of packet size and reproducible human motion.

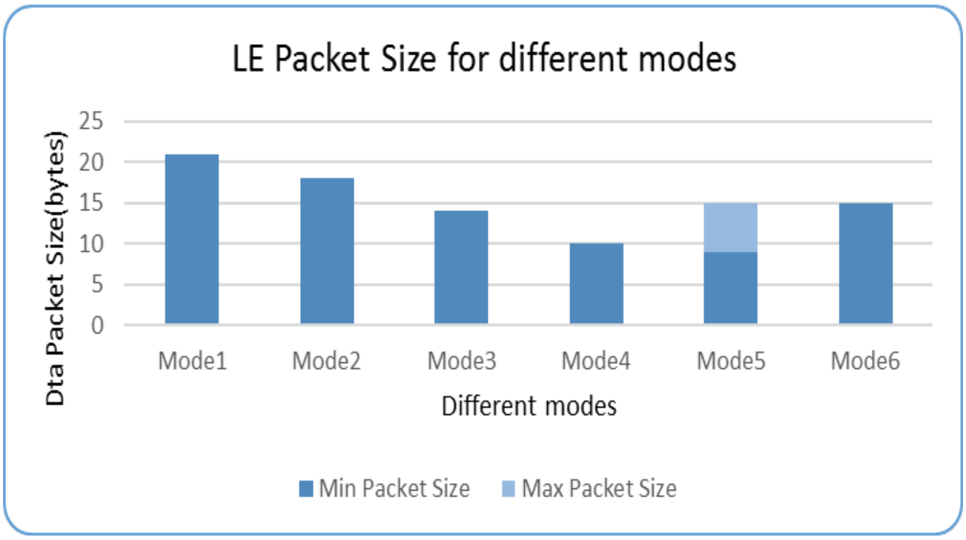

Figure 5.1: Packet size for different modes

5.2 Trace Drive Test Execution (TDTE)

Trace Driven Test Execution is an evaluation method that provides homogeneous test data for scenarios where it is not possible to provide exacting data for different test scenarios. The execution steps for this testing technique is outlined below:

1. The first phase is generate the trace using a standard set of exercises. These consist of circular, horizontal and vertical motions along with different speeds.

2. This real-time motion data set is then transmitted over BLE to the MPU Client. The data aggregator in Processing collects all the data, formats it into a Quaternion or Sensor data array ad stores it in a C++ header file on the file system.

3. The second phase is the test execution phase. This trace data is then loaded onto a MPU Test Service. Data is then sent from this Service to the Client by looping over the data for 10 minutes.

4. The final step is generic step where the relevant tests associated with the experiment are performed.

Fig. 5.2 shows the flowchart for a trace drive test execution.

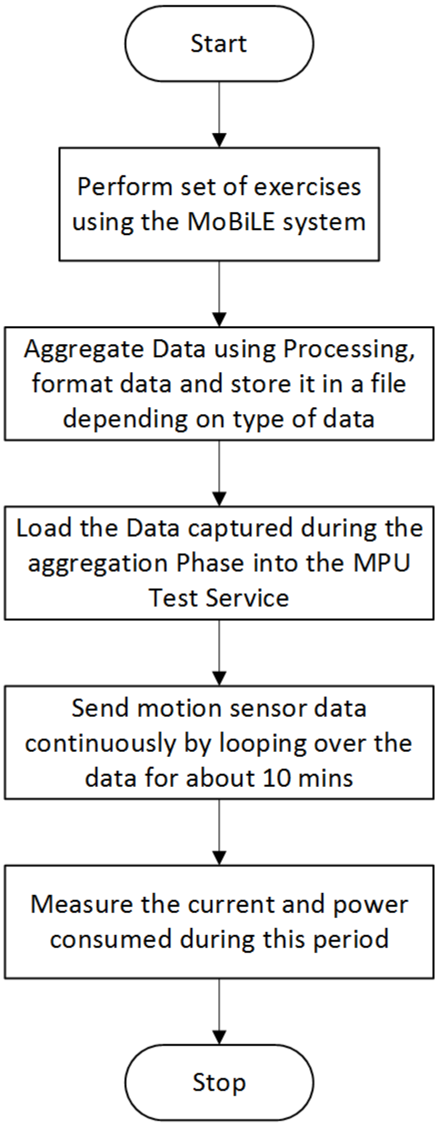

Figure 5.2: Trace Driven Test Execution

5.3 Experiment 2: Current Consumption for different modes and scenarios

Aim: To determine the voltage consumption for the different modes and scenarios **Procedure:**

The following steps are executed to evaluate this aim:

- To setup the device for voltage measurement, shorted solder bridge SB9 needs to be cut and a resistance of 10 ohms is needed to be mounted at R6 on the nRF51 DK board.
- TDTE is utilized for this experiment. The trace data is generated and then loaded on the MPU Test Service.
- The voltage consumed during each connection is then measured using a multimeter across P22 on the Nordic Board.
- Current is then deduced using Ohm's Law as given in 5.1.

$$Current = \frac{Voltage}{Resistance} \tag{5.1}$$

- The experiment is performed for three modes Quaternion data, Differential data and Raw data (6DOF) and at varying loads of sensor motion.

Observations: Graph 5.3 shows the variation in the current consumption for the three

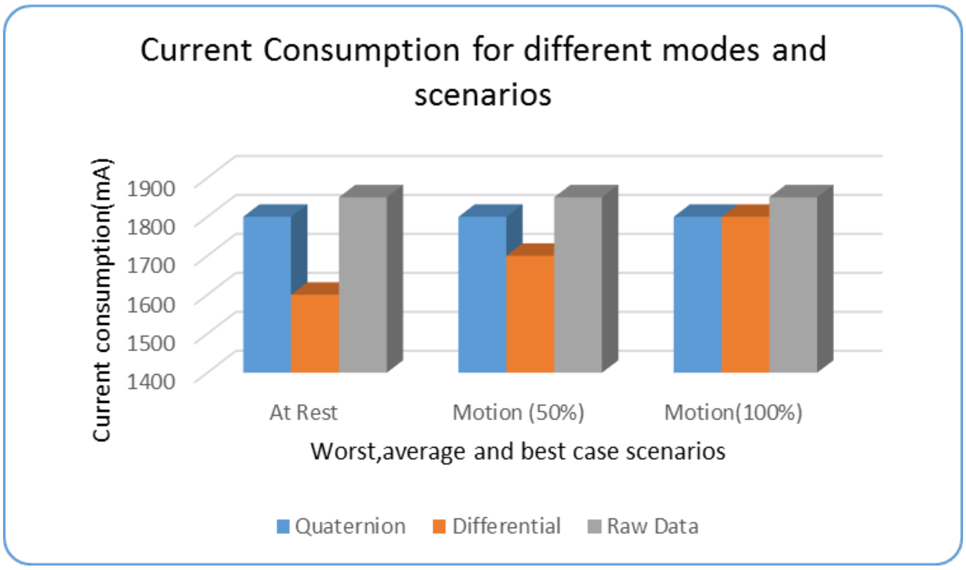

Figure 5.3: Current consumption across different modes

main modes by averaging results over 1000 packets. The baseline current consumption when two devices are connected and have MPU GATT service and Client running is 1.5mA. The current consumed when the sensor is at rest is 1.85mA per packet for the raw data stream mode and 1.8mA per packet for a quaternion data stream while the differential data only consumes 1.6mA per packet. For an average case scenarios, the current consumption remains the same for quaternion and raw data modes, however, for the PEDA data stream, the current consumption increases to 1.7mA. In worst case scenario, the PEDA algorithm saturates to the quaternion stream current consumption. If we take the baseline consumption, the PEDA algorithm consumes only 0.1mA at rest, 0.2mA in average case scenarios and 3mA in worst case scenarios as compared to 0.3mA current consumption in other modes of operation and hence can be effectively utilized for real-time sensor data streaming.

5.4 Experiment 3: Battery usage graphs for different modes of operation

Aim: To develop a battery usage profile for different modes and scenarios.

Procedure: The following steps are executed to evaluate this aim:

- To setup the device for voltage measurement, shorted solder bridge SB9 needs to be cut and a resistance of 10 ohms is needed to be mounted at R6 on the nRF51 DK board.
- TDTE is utilized for this experiment. The trace data is generated and then loaded on the MPU Test Service.
- A lithium coin cell battery CR2032 is used as a power source for these experiments. The voltage consumed during each connection is then measured using a multimeter across battery using Vcc and GND on the nRF51DK.
- The voltage is measured every 5 mins in a 45 minute interval to create battery usage

profile. All the starting voltage values are normalized to 2.717V.

- The experiment is performed for three modes Quaternion data, Differential data and Raw data (6DOF).

Observations: Graph 5.3 shows the battery usage for the different modes of operation

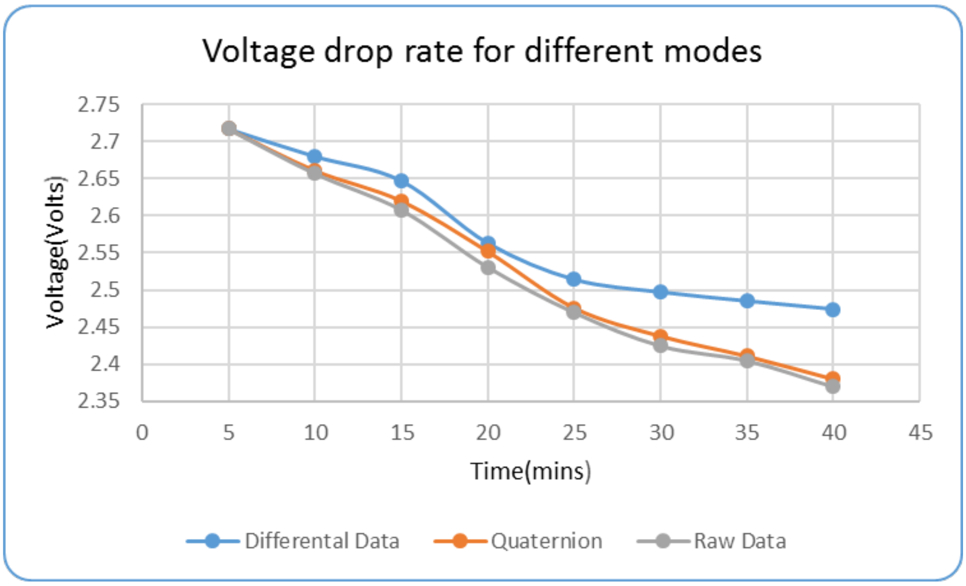

Figure 5.4: Battery usage graphs for different modes of operation

for the MPU Service. Battery life is an important requirement of remote monitoring systems and even a small improvement in current consumption would in a longer battery life and up to an improvement of 3-4 hours. The graph shows periods of rapid drop in battery and periods of almost negligible battery consumption. This is due to the fact that the data has periods of activity and inactivity. The PEDA algorithm consumes low battery during the inactive period and hence has a step decrease while the other two modes continuously keep transmitting data over BLE and hence have an almost constant decreasing slope. These results show that the battery usage over time is lower for the PEDA algorithm as compared to other modes of operation.

5.5 Summary

The proposed system has been evaluated against various metrics and the following insights have been gained:

- The size of the data packets sent within a connection are variable. The PEDA algorithm takes advantage of this feature varying the packets depending on the data resulting in the device sending data only when necessary.
- The PEDA algorithm shows a step power usage profile as compared to the other modes which show a steep response due to its ability to intelligently attempt to conserve bandwidth and power as and when possible.
- The GATT Service allows a remote control feature allowing a soft control of the service giving the end user the possibility to choose his own modes.
- The 3D Human motion is accurate and has a very quick response time. These results are very promising for real-time rehabilitation.

Chapter 6

Conclusion

AmI in HealthCare aims to bring low cost, light weight, pervasive and unobtrusive medical care for an individual at the comfort of his own home. The current technologies focus on developing embeddable, low-power communication devices that allow a them to wirelessly transmit data to other centralized medical system. Different approaches investigating and implementing various solutions have been discussed and some of the concepts like quaternions adopted for the design. The prototype developed has demonstrated that most of the requirements for Wireless Body Sensor Networks are satisfied by Bluetooth Low Energy owing to its low cost, power consumption, lightweight, good bit rates and proliferation in mobile devices, laptops and sensors. nRF5x series provide a compact BLE transceiver which when combined with an IMU can provide an efficient movement tracking system. Due to the rapid and unusually high growth in the adoption of BLE devices and advancement in machine learning, remote medical monitoring could be possible in the near future.

6.1 Contribution

The most important contribution of this thesis is the design and implementation of a MPU GATT service as well as an evaluation of the feasibility of Bluetooth Low Energy to support real-time transfer of motion sensor data. A novel Power Efficient Data Adaptive Algorithm (PEDA) was implementation and evaluated showing good preliminary results. This research also investigated and implemented four different methods to implement a GATT client to interface with the MPU GATT Service evaluating their functionalities and highlighting their drawbacks.

6.2 Future Work

Although the evaluations for the MoBiLE system are very promising, further research is required take this prototype to the next level. Our ultimate goal is to create a system of wearable devices that can monitor not only motion but also vital signs like heart beat and transmit all this information to a centralized medical system which would enable remote human health monitoring without human intervention based on a set of rules. Enhancement to the PEDA algorithm is required so that it can estimate motion data patterns and go into predictive sleep modes. More detailed analysis of power consumption using the PEDA Algorithm is needed with more complex exercise patterns. Lastly, considerable amount of work is required to facilitate remote monitoring of exercises through the use of inverse kinematics, machine learning for automatic monitoring of exercises and rule based rehabilitation.

Appendix A

Abbreviations

Short Term	Expanded Term
AmI	Ambient Intelligence in Healthcare
BLE	Bluetooth Low Energy
DMO	Digital Motion Processor
DOF	Degree's of Freedom
GAP	Generic Access Profile
GATT	Generic Attribute Profile
IMU	Inertial Measurement Unit
MPU	Motion Processing Unit
PEDA	Power Efficient Data Adaptive
TDTE	Trace Drive Test Execution

Appendix B

MPU GATT Service Design

Name: MPU Sensor Service

Type: org.bluetooth.service.motion_sensor_data

Assigned Number:0xA100

Abstract:

This service exposes DMP motion sensor data consisting of raw, differential and processed data intended for physiotherapy

Summary:

The MPU Sensor service exposes DMP motion sensor data consisting of raw, differential and processed data intended for physiotherapy

Service Dependencies

This service is not dependent upon other services

GATT Requirements

Sub-Procedure	Server Requirement
Write Characteristic Value	Mandatory if the Heart Rate Control Point characteristic is supported, otherwise excluded for this service.
Notifications	Mandatory
Read Characteristic Descriptors	Mandatory
Write Characteristic Descriptors	Mandatory

Table B.1: GATT Requirements

Transport Dependencies

Transport	Supported
Classic	false
Low Energy	true

Table B.2: Transport Dependencies

Error Codes

Name	Code	Description
Motion Sensor Measurement not Supported	0x90	

Table B.3: List of Error Code

Service Characteristics

<table>
<tr><th>Overview</th><th>Properties</th><th>Security</th><th>Descriptors</th></tr>
<tr>
<td>Name: MPU Sensor Measurement
Description: This characteri stic is used to send motion sensor measurement.
Type: org.bluetooth. characteristic. mpu_sensor_ measurement
Requirement: Mandatory</td>
<td><table>
<tr><th>Property</th><th>Require-ment</th></tr>
<tr><td>Read</td><td>Mandatory</td></tr>
<tr><td>Write</td><td>Excluded</td></tr>
<tr><td>Write Without Response</td><td>Excluded</td></tr>
<tr><td>Signed Write</td><td>Excluded</td></tr>
<tr><td>Notify</td><td>Mandatory</td></tr>
<tr><td>Indicate</td><td>Excluded</td></tr>
<tr><td>Write Auxili-aries</td><td>Excluded</td></tr>
<tr><td>Broadcast</td><td>Excluded</td></tr>
<tr><td>Extended Properties</td><td></td></tr>
</table></td>
<td>None</td>
<td><table>
<tr><th>Overview</th><th>Permissions</th></tr>
<tr><td>Name: Client Characte-ristic Configur-ation
Type: org. bluetooth. descriptor. gatt.client_ characteri-stic_config-uration
Require-ment: Mandatory</td>
<td><table>
<tr><th>Permi-ssion</th><th>Requir-ement</th></tr>
<tr><td>Read</td><td>Manda-tory</td></tr>
<tr><td>Write</td><td>Manda-tory</td></tr>
</table></td></tr>
</table></td>
</tr>
<tr>
<td>Name: Body Sensor Location
Description: This characteri stic is used to describe the intended locati-on of sensor measurement for the device.
Type: org.bluetooth. characteristic. body_sensor_ location
Requirement: Optional</td>
<td><table>
<tr><th>Property</th><th>Require-ment</th></tr>
<tr><td>Read</td><td>Mandatory</td></tr>
<tr><td>Write</td><td>Excluded</td></tr>
<tr><td>Write Without Response</td><td>Excluded</td></tr>
<tr><td>Signed Write</td><td>Excluded</td></tr>
<tr><td>Notify</td><td>Excluded</td></tr>
<tr><td>Indicate</td><td>Excluded</td></tr>
<tr><td>Write Auxili-aries</td><td>Excluded</td></tr>
<tr><td>Broadcast</td><td>Excluded</td></tr>
<tr><td>Extended Properties</td><td></td></tr>
</table></td>
<td>None</td>
<td>None</td>
</tr>
</table>

Continued on next page

Table B.4 – *Continued from previous page*

Overview	**Properties**	**Security**	**Descriptors**
Name: MPU Sensor Resolution **Description:** This characteri stic is used to describe the resolution factor for the sensors. **Type:** org.bluetooth. characteristic. MPU_sensor_ resolution **Requirement:** Mandatory	<table><tr><th>Property</th><th>Requirement</th></tr><tr><td>Read</td><td>Mandatory</td></tr><tr><td>Write</td><td>Excluded</td></tr><tr><td>Write Without Response</td><td>Excluded</td></tr><tr><td>Signed Write</td><td>Excluded</td></tr><tr><td>Notify</td><td>Excluded</td></tr><tr><td>Indicate</td><td>Excluded</td></tr><tr><td>Write Auxiliaries</td><td>Excluded</td></tr><tr><td>Broadcast</td><td>Excluded</td></tr><tr><td>Extended Properties</td><td></td></tr></table>	None	None
Name: MPU Control Point **Description:** This characteri stic is used to enable the client to write control points to a Server to control behaviour. **Type:** org.bluetooth. characteristic. MPU_control_ point **Requirement:** Mandatory	<table><tr><th>Property</th><th>Requirement</th></tr><tr><td>Read</td><td>Excluded</td></tr><tr><td>Write</td><td>Mandatory</td></tr><tr><td>Write Without Response</td><td>Excluded</td></tr><tr><td>Signed Write</td><td>Excluded</td></tr><tr><td>Notify</td><td>Excluded</td></tr><tr><td>Indicate</td><td>Excluded</td></tr><tr><td>Write Auxiliaries</td><td>Excluded</td></tr><tr><td>Broadcast</td><td>Excluded</td></tr><tr><td>Extended Properties</td><td></td></tr></table>	None	None

Table B.4: MPU Service Characteristics

Name: MPU Sensor Measurement

Type: org.bluetooth.service.mpu_sensor_measurement
Assigned Number:0xA101

Names	Field Requirement	Format	Min Value	Max Value	Additional Information
Flags	Mandatory	8bit	N/A	N/A	Bit-Field Please refer to Table
Accelerometer-X Axis Measurement Value (uint 8) **Information:** Note: The presence and format of the Heart Rate Measurement Value field is dependent upon bits 0 and bits [2:3] of the Flags field respectively. **Unit:** org.bluetooth.unit.acceleration. metres_per_second	C1 ,C4,C5	uint8	N/A	N/A	None
Accelerometer-Y Axis Measurement Value (uint 8) **Information:** Note: The presence and format of the Heart Rate Measurement Value field is dependent upon bits 0 and bits [2:3] of the Flags field respectively. **Unit:** org.bluetooth.unit.acceleration. metres_per_second	C1 ,C4,C5	uint8	N/A	N/A	None
Accelerometer-Z Axis Measurement Value (uint 8) **Information:** Note: The presence and format of the Heart Rate Measurement Value field is dependent upon bits 0 and bits [2:3] of the Flags field respectively. **Unit:** org.bluetooth.unit.acceleration. metres_per_second	C1 ,C4,C5	uint8	N/A	N/A	None
Accelerometer-X Axis Measurement Value (uint 16) **Information:** Note: The presence and format of the Heart Rate Measurement Value field is dependent upon bits 0 and bits [2:3] of the Flags field respectively. **Unit:** org.bluetooth.unit.acceleration. metres_per_second	C2 ,C4,C5	uint8	N/A	N/A	None

Continued on next page

Table B.5 – *Continued from previous page*

Names	Field Requirement	Format	Min Value	Max Value	Additional Information
Accelerometer-Y Axis Measurement Value (uint 16) **Information:** Note: The presence and format of the Heart Rate Measurement Value field is dependent upon bits 0 and bits [2:3] of the Flags field respectively. **Unit:** org.bluetooth.unit.acceleration. metres_per_second	C2 ,C4,C5	uint8	N/A	N/A	None
Accelerometer-Z Axis Measurement Value (uint 16) **Information:** Note: The presence and format of the Heart Rate Measurement Value field is dependent upon bits 0 and bits [2:3] of the Flags field respectively. **Unit:** org.bluetooth.unit.acceleration. metres_per_second	C2 ,C4,C5	uint8	N/A	N/A	None
Gyroscope-X Axis Measurement Value (uint 8) **Information:** Note: The presence and format of the Gyroscope-X Axis Value field is dependent upon bit 0 and bits [2:6]of the Flags field. **Unit:** org.bluetooth.unit.acceleration. angular_velocity.radian_per_second	C1,C4	uint8	N/A	N/A	None
Gyroscope-Y Axis Measurement Value (uint 8) **Information:** Note: The presence and format of the Gyroscope-X Axis Value field is dependent upon bit 0 and bits [2:6]of the Flags field. **Unit:** org.bluetooth.unit.acceleration. angular_velocity.radian_per_second	C1,C4	uint8	N/A	N/A	None

Continued on next page

Table B.5 – *Continued from previous page*

Names	Field Requirement	Format	Min Value	Max Value	Additional Information
Gyroscope-Z Axis Measurement Value (uint 8) **Information:** Note: The presence and format of the Gyroscope-X Axis Value field is dependent upon bit 0 and bits [2:6]of the Flags field. **Unit:** org.bluetooth.unit.acceleration. angular_velocity.radian_per_second	C1,C4	uint8	N/A	N/A	None
Gyroscope-X Axis Measurement Value (uint 16) **Information:** Note: The presence and format of the Gyroscope-X Axis Value field is dependent upon bit 0 and bits [2:6]of the Flags field. **Unit:** org.bluetooth.unit.acceleration. angular_velocity.radian_per_second	C2,C4	uint8	N/A	N/A	None
Gyroscope-Y Axis Measurement Value (uint 16) **Information:** Note: The presence and format of the Gyroscope-X Axis Value field is dependent upon bit 0 and bits [2:6]of the Flags field. **Unit:** org.bluetooth.unit.acceleration. angular_velocity.radian_per_second	C2,C4	uint8	N/A	N/A	None
Gyroscope-Z Axis Measurement Value (uint 16) **Information:** Note: The presence and format of the Gyroscope-X Axis Value field is dependent upon bit 0 and bits [2:6]of the Flags field. **Unit:** org.bluetooth.unit.acceleration. angular_velocity.radian_per_second	C2,C4	uint8	N/A	N/A	None

Continued on next page

Table B.5 – *Continued from previous page*

Names	Field Requirement	Format	Min Value	Max Value	Additional Information
Magnetometer-X Axis Measurement Value (uint 8) **Information:** Note: The presence and format of the Magnetometer-X Axis Measurement Value field is dependent upon bit 0 and bits [2:6]of the Flags field. **Unit:** org.bluetooth.unit.acceleration. magnetic_flux_density.tesla	C1	uint8	N/A	N/A	None
Magnetometer-Y Axis Measurement Value (uint 8) **Information:** Note: The presence and format of the Magnetometer-X Axis Measurement Value field is dependent upon bit 0 and bits [2:6]of the Flags field. **Unit:** org.bluetooth.unit.acceleration. magnetic_flux_density.tesla	C1	uint8	N/A	N/A	None
Magnetometer-Z Axis Measurement Value (uint8) **Information:** Note: The presence and format of the Magnetometer-X Axis Measurement Value field is dependent upon bit 0 and bits [2:6]of the Flags field. **Unit:** org.bluetooth.unit.acceleration. magnetic_flux_density.tesla	C1	uint8	N/A	N/A	None
Magnetometer-X Axis Measurement Value (uint16) **Information:** Note: The presence and format of the Magnetometer-X Axis Measurement Value field is dependent upon bit 0 and bits [2:6]of the Flags field. **Unit:** org.bluetooth.unit.acceleration. magnetic_flux_density.tesla	C2	uint8	N/A	N/A	None

Continued on next page

Table B.5 – *Continued from previous page*

Names	Field Requirement	Format	Min Value	Max Value	Additional Information
Magnetometer-Y Axis Measurement Value (uint16) **Information:** Note: The presence and format of the Magnetometer-X Axis Measurement Value field is dependent upon bit 0 and bits [2:6]of the Flags field. **Unit:** org.bluetooth.unit.acceleration. magnetic_flux_density.tesla	C2	uint8	N/A	N/A	None
Magnetometer-Z Axis Measurement Value (uint16) **Information:** Note: The presence and format of the Magnetometer-X Axis Measurement Value field is dependent upon bit 0 and bits [2:6]of the Flags field. **Unit:** org.bluetooth.unit.acceleration. magnetic_flux_density.tesla	C2	uint8	N/A	N/A	None
Quaternion Measurement Value q1 (uint16) **Information:** Note:The presence of the Quaternion Measurement Value field is dependent upon bits [4:6] of the Flags field. **Unit:** org.bluetooth.unit.acceleration. unitless	C3	uint8	N/A	N/A	None
Quaternion Measurement Value q2 (uint16) **Information:** Note:The presence of the Quaternion Measurement Value field is dependent upon bits [4:6] of the Flags field. **Unit:** org.bluetooth.unit.acceleration. unitless	C3	uint8	N/A	N/A	None
Quaternion Measurement Value q3 (uint16) **Information:** Note:The presence of the Quaternion Measurement Value field is dependent upon bits [4:6] of the Flags field. **Unit:** org.bluetooth.unit.acceleration. unitless	C3	uint8	N/A	N/A	None

Continued on next page

Table B.5 – *Continued from previous page*

Names	Field Requirement	Format	Min Value	Max Value	Additional Information
Quaternion Measurement Value q4 (uint16) **Information:** Note:The presence of the Quaternion Measurement Value field is dependent upon bits [4:6] of the Flags field. **Unit:** org.bluetooth.unit.acceleration.unitless	C3	uint8	N/A	N/A	None

Table B.5: MPU Sensor Measurement Characteristic

Name: MPU Sensor Flag

Bit	Size	Name	Definition: Key	Value	Requires
0	1	Sensor Data Type	0	MPU Sensor Data Format is set to UINT8. Units: metres per second squared	C1
			1	MPU Sensor Data Format is set to UINT16. Units: metres per second squared	C2
1	1	MPU Service Frame Rate	0	MPU Service frame rate is set to 30 frame per second.	-
			1	MPU Service frame rate is set to 15 frame per second.	-
2	2	Sensor Data Type	00	MPU service sends Raw Data	C1 or C2
			01	MPU service sends Differential Data	-
			10	MPU service sends Processed Data	C3
			11	MPU service sends DMP Data	C3
4	3	Mode of Operation	000	MobiLE systems running in Normal Mode	C1
			001	Sensing sending Accelerometer and Gyroscope Data	C4
			010	Select Acc only	C5
			XXX	Reserved	-

Table B.6: Flag Field from MPU Sensor Characteristic

Bibliography

[Acampora et al. 2013] Acampora, G. , Cook, D.J. , Rashidi, P., Vasilakos, A.V., A Survey on Ambient Intelligence in Healthcare, *Proceedings of the IEEE* 101, Nr. 12, pp. 2470–2494, Dec, 2013.

[ARM] ARM, Mbed, *Mbed Developer*, URL: `https://developer.mbed.org/`

[Benocci et al. 2010] Benocci, M. , Tacconi, C. , Farella, E. , Benini, L. , Chiari, L., Vanzago, L., "Accelerometer-based fall detection using optimized ZigBee data streaming, *Microelectronics Journal* ,vol. no. 41, pp. 703–710, November, 2010.

[Buonocunto and Marinoni 2014] Buonocunto, P., Marinoni, M., "Tracking limbs motion using a wireless network of inertial measurement units". in Proceedings of the *9th IEEE International Symposium on Industrial Embedded Systems (SIES).* Pisa, Italy, June, 2014, pp. 66–76.

[Casamassima et al. 2013] Casamassima, F. , Farella, E., Benini, L., "Synchronization methods for Bluetooth based WBANs". in Proceedings of the *IEEE International Conference on Body Sensor Networks.* Cambridge, MA, USA, May, 2013, pp. 1–6.

[Dementyev et al. 2013] Dementyev, A. , Hodges, S. , Taylor, S., Smith, J., "Power consumption analysis of Bluetooth Low Energy, ZigBee and ANT sensor nodes in a cyclic sleep scenario". in Proceedings of the *IEEE International Wireless Symposium (IWS).* Beijing, China, April, 2013, pp. 1–4.

[Fernandez et al. 2013] Fernandez, P. , Qinggang, M., Chung, P.W.H., "A Machine Learning Method for Identification of Key Body Poses in Cyclic Physical Exercises". in Proceedings of the *IEEE International Conference on Systems, Man, and Cybernetics (SMC)*. Manchester, United Kingdom, October, 2013, pp. 1605–1610.

[Fry and Reas] Fry, B., Reas, C., *3D Processing*, URL: `https://processing.org/`

[Ghoshdastider et al. 2014] Ghoshdastider, U. , Viga, R., Kraft, M., "Wireless time synchronization of a collaborative brain-computer-interface using bluetooth low energy". in Proceedings of the *IEEE SENSORS*. Valencia, Spain, November, 2014, pp. 2250–2254.

[Giggins et al. 2013] Giggins, O. , Kelly, D., Caulfield, B., "Evaluating rehabilitation exercise performance using a single inertial measurement unit". in Proceedings of the *7th International Conference on Pervasive Computing Technologies for Healthcare (PervasiveHealth)*. Venice, Italy, May, 2013, pp. 49–56.

[Hanson 2007] Hanson, Andrew J., Visualizing quaternions: course notes for siggraph 2007. in Proceedings of the *In ACM SIGGRAPH 2007 courses, SIGGRAPH 07*, 2007

[Helmer et al. 2010] Helmer, A. , Song, B. , Ludwig, W. , Schulze, M. , Eichelberg, M. , Hein, A. , U. Tegtbur, R. K. , Haux, R., Marschollek, M., "A sensor-enhanced health information system to support automatically controlled exercise training of COPD patients". in Proceedings of the *4th International Conference on Pervasive Computing Technologies for Healthcare (PervasiveHealth)*. Munich , Germany, March, 2010, pp. 1–6.

[Jarochowski et al. 2011] Jarochowski, B. P. , Shin, S. , Ryu, D., Kim, H., "Ubiquitous rehabilitation center: An implementation of a wireless sensor network based rehabilitation management system". in Proceedings of the *International Conference on Convergence Information Technology*. Washington, DC, USA, 2011, pp. 23492358.

[Jihyoung et al. 2011] Jihyoung, K. , Sungwon, Y., Mario, G., "StrokeTrack:wireless inertial motion tracking of human arms for stroke telerehabilitation". in Proceedings of the *First ACM Workshop on Mobile Systems, Applications and Services for Healthcare (mHealthSys).* Seattle, Washington, USA, November, 2011, pp. .

[Jung et al. 2005] Jung, J.-W. , Do, J.-H. , Kim, Y.-M. , Suh, K.-S. , Kim, D.-J., Bien, Z., "Advanced robotic residence for the elderly/the handicapped:Realization and user evaluation". in Proceedings of the *9th International Conference on Rehabilitation Robotics,ICORR.* Chicago, Illinois, July, 2005, pp. 492495.

[Khan et al. 2013] Khan, A. M. , Young-Koo, L. , Lee, S. Y., Tae-Seong, K., "A Triaxial Accelerometer-Based Physical-Activity Recognition via Augmented-Signal Features and a Hierarchical Recognizer, *IEEE Transactions on Information Technology in Biomedicine* ,vol. no. 14, pp. 1166–1172, June, 2013.

[Laine et al. 2014] Laine, T.H. , Chaewoo, L., Haejung, S., "Wireless time synchronization of a collaborative brain-computer-interface using bluetooth low energy". in Proceedings of the *Eighth International Conference on Innovative Mobile and Internet Services in Ubiquitous Computing (IMIS).* Birmingham, UK, July, 2014, pp. 139–145.

[Lin and Kulic 2012] Lin, J. F. S., Kulic, D., "Segmenting human motion for automated rehabilitation exercise analysis". in Proceedings of the *Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC).* San Diego, CA. ,USA , September, 2012, pp. 2881–2884.

[Linde and Tucker 2013] Linde, J., Tucker, B.J., *Audio transfer using the bluetooth low energy standard.* February 21, 2013. – URL: `http://www.google.com/patents/US20130045684`. – US Patent App. 13/403,605

[Lingfei et al. 2012] Lingfei, M. , Shaopeng, L. , Gao, R.X. , John, D. , Staudenmayer, J.W., Freedson, P.S., "Wireless Design of a Multisensor System for Physical Activity Monitoring, *Microelectronics Journal* ,vol. no. 59, pp. 3230–3237, November, 2012.

[Madgwick] Madgwick, S.O.H., *An efficient orientation filter for inertial and inertial/magnetic sensor arrays*, URL: `http://www.x-io.co.uklres/doc/madgwick-interaIJeport.pdf`

[Munoz et al. 2014] Munoz, M. O. , Foster, R., Yang, H., "Exploring Physiological Parameters in Dynamic WBAN Channels", *IEEE Transactions on Antennas and Propagation* ,vol. no. 62, pp. 5268–5281, October, 2014.

[Nordic] Nordic, Semiconductors, *nrf51DK*, URL: `https://processing.org/`

[Park et al. 2007] Park, K.-H. , Bien, Z. , Lee, J.-J. , Kim, B. , Lim, J.-T. , Kim, J.-O. , Lee, H. , Stefanov, D. , Kim, D.-J. , Jung, J.-W. , Do, J.-H. , Seo, K.-H. , Kim, C. , Song, W.-G., Lee, W.-J., "Robotic smart house to assist people with movement disabilities, *Autonomous Robots* ,vol. no. 22, pp. 183–198, February, 2007.

[Peppoloni et al. 2013] Peppoloni, L. , Filippeschi, A. , Ruffaldi, E., Avizzano, C.A., "A novel 7 degrees of freedom model for upper limb kinematic reconstruction based on wearable sensors". in Proceedings of the *11th IEEE International Symposium on Intelligent Systems and Informatics (SISY).* Subotica, Serbia, September, 2013, pp. 105–110.

[Shih-Yeh et al. 2011] Shih-Yeh, C. , Wei-Tsong, L. , Han-Chieh, C. , Yueh-Min, H., Lai, C.-F., "Adaptive reconstruction of human motion on wireless body sensor networks". in Proceedings of the *Proceedings of the International Conference on Wireless Communications and Signal Processing (WCSP).* Nanjing, China, November, 2011, pp. 1–5.

[Siekkinen et al. 2012] Siekkinen, M. , Hiienkari, M. , Nurminen, J. K., Nieminen, J., "How low energy is bluetooth low energy? Comparative measurements with ZigBee/802.15.4". in Proceedings of the *IEEE Wireless Communications and Networking Conference Workshops (WCNCW).* Paris, France, April, 2012, pp. 232–237.

[Sparkfun] Sparkfun, *MPU 9150*, URL: `https://www.sparkfun.com/products/11486`

[Taylor et al. 2012] Taylor, P.E. , Almeida, G.J.M. , Hodgins, J.K., Kanade, T., "Multi-label classification for the analysis of human motion quality". in Proceedings of the *Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC).* San Diego, CA , USA, September, 2012, pp. 2214–2218.

[Tianxiang et al. 2013] Tianxiang, Z. , Karg, M. , Lin, J.F-S. , Kulic, D., Venture, G., "IMU based single stride identification of humans". in Proceedings of the *IEEE ROMAN.* Gyeongju, South Korea, August, 2013, pp. 220–225.

[Touati et al. 2013] Touati, F. , Tabish, R., Mnaouer, A. B., A Real-time {BLE} Enabled {ECG} System for Remote Monitoring, {*APCBEE*} *Procedia* 7, pp. 124 – 131, 2013.. – The 3rd International Conference on Biomedical Engineering and Technology - {ICBET} 2013

[Townsend et al. 2014] Townsend, K. , Cuf, C. , Akiba, Davidson, R., *"Getting Started with Bluetooth Low Energy".* Sebastopol, CA, USA O'Reilly Media, April, 2014 .

[Wenbing et al. 2014] Wenbing, Z. , Lun, R. , Espy, D.D., Reinthal, M.A., "Rule based realtime motion assessment for rehabilitation exercises". in Proceedings of the *IEEE Symposium on Computational Intelligence in Healthcare and e-health (CICARE).* Orlando, Florida, USA, December, 2014, pp. 133–140.

[Young-Jin and Hui-Sup 2013] Young-Jin, P., Hui-Sup, C., "Transmission of ECG data with the patch-type ECG sensor system using Bluetooth Low Energy". in Proceedings of the *International Conference on ICT Convergence (ICTC).* Jeju Island,South Korea, October, 2013, pp. 289–294.

[Zatout 2012] Zatout, Y., Using wireless technologies for healthcare monitoring at home: A survey. in Proceedings of the *IEEE 14th International Conference on e-Health Networking, Applications and Services (Healthcom)*, Oct, 2012, pp. 383–386.