

## **Acknowledgement**

First and foremost, I would like to express my sincere gratitude to my supervisor, Professor Lee Boon Giin for guiding and encouraging me patiently throughout the completion of this thesis. He had allowed me freedom to explore the research topics I am interested in without any restrictions, yet he always ensured that I was progressing steadily and in the right way. Also, he had given to me the deep insights into how to write well-structured journal articles. His teaching, supervision and attitude will remain forever grateful to me.

Thanks attended to my fellow ubiquitous sensor network laboratory members for their helping in providing an excellent research environment which made this research possible. Their sharing and comments have meant a lot to me in succeeding my research study.

Special thanks to my family, my girlfriend and my close friends for supporting me throughout this research work. My research life and study as graduate student went well and smoothly because of their kind understanding and encouragement.

# Contents

1	Introduction.....	1
1.1	Motivation and Research Objectives .....	2
1.2	Contributions.....	4
1.3	Thesis Organization .....	4
2	Literature Review.....	6
2.1	Vision-based Method .....	7
2.2	Non-vision-based Method.....	8
3	System Design .....	12
3.1	Design of Hardware .....	14
3.1.1	Processing Module.....	14
3.1.2	Sensor Module .....	17
3.1.3	3D-Printed Casing.....	18
3.2	Design of Software .....	19
3.2.1	What is Qt? .....	19
3.2.2	Qt Creator – IDE .....	21
3.2.3	Libraries .....	22
3.2.4	Smart Home Application .....	24
4	Experiment and Evaluation.....	28

4.1	Dynamic Time Warping Algorithm.....	28
4.1.1	One-Dimensional DTW .....	29
4.1.2	N-Dimension – ND DTW .....	31
4.2	Implementation .....	31
4.2.1	Pre-processing.....	31
4.2.2	Training Phase.....	34
4.2.3	Classification Threshold .....	35
4.3	Experiment Setup.....	36
4.4	Result and Discussion .....	38
5	Conclusions and Future Work.....	46
	References.....	47

## List of Figures

- Figure 2-1 Vision based hand gesture recognition system.
- Figure 2-2 Hand gesture recognition system using sEMG sensor.
- Figure 2-3 A Wearable device uses IMUs for recognizing hand gestures.
- Figure 2-4 The flex sensor glove for detecting hand gestures.
- Figure 3-1 System design overview.
- Figure 3-2 Proposed smart wearable controller system.
- Figure 3-3 Proposed smart wearable controller system that can be applied to various applications with Qt cross-platform software.
- Figure 3-4 (a) Schematic and (b) PCB layer design of processing module.
- Figure 3-5 (a) Schematic and (b) PCB layer design of sensor module.
- Figure 3-6 The flowchart of proposed system fabrication using 3D printer.
- Figure 3-7 Appearance design (a) part design and (b) overview design.
- Figure 3-8 Qt cross-platform application development framework.
- Figure 3-9 Qt creator IDE.
- Figure 3-10 Screenshot of the smart home application.
- Figure 3-11 System overview of smart home application.
- Figure 4-1 Two identical time-series signals, with phase differences measured with (a) Euclidean distance and (b) Dynamic Time Warping.
- Figure 4-2 Cost matrix and the minimum warp path (indicated by the red line).
- Figure 4-3 The comparison of angular data with and without complimentary filter.
- Figure 4-4 The predefined 18 gestures which are divided into (a) simple gestures, (b) common gestures, (c) and complex gestures.

- Figure 4-5 Four finger gestures which are (a) zoom out, (b) zoom in, (c) rotate right and (f) rotate left.
- Figure 4-6 Self-development application for the purposes of training phase and testing phase.
- Figure 4-7 The accuracy rate of each groups in 18 hand gestures and 4 fingers gestures.
- Figure 4-8 Orientation samples obtained from 3 IMUs to the gestures including (a) rotate right, (b) rotate left, (c) zoom in and (d) zoom out.
- Figure 4-9 Sample of orientation obtained from 3 IMUs legend.

## List of Tables

Table 3-1	Processing module hardware specification.
Table 3-2	Qt Essentials
Table 3-3	Qt Add-Ons
Table 4-1	Gesture recognition accuracy rate with 1 IMU on back of the palm.
Table 4-2	Table 4 1 Gesture recognition accuracy rate with 2 IMUs on the thumb and the index finger.
Table 4-3	Table 4 1 Gesture recognition accuracy rate with 3 IMUs on back of the palm and the thumb and the index finger.
Table 4-4	Comparison of related systems

# 1 Introduction

Gesture recognition has ~~become~~ become one of the ~~hottest-popular research~~ fields of ~~research~~ that since it served as an intelligent and a natural interface between the human and the computer (HCI) in recent years. In ~~the past~~ last decades, ~~there are~~ limited ways for human to interact with computers, for instance, -computer mouse and keyboard which played a significant roles in ~~human-computer interaction~~ HCI. ~~However~~ Fortunately, ~~thanks-given to~~ the rapid and advanced development of technology ~~about both hardware and software~~, scientists and engineers are able to explore alternative ways to ~~started not only developing assistive technologies which would~~ recognize and imitate human gestures to perform complex and difficult tasks in various areas. Moreover, wearable computing trends allowed devices to be portable and smaller in size yet covered more functionalities compared to the traditional bulky, inconvenient and limited usage of the devices ~~to perform tasks that we consider difficult or strenuous to make many HCI types, but also developing portable and wearable devices to make more convenient for users.~~

Gestures are physical movements of ~~different parts of the~~ the human body to express their feelings and emotions before languages are even invented. ~~that are expressive and meaningful to human beings.~~ In fact, ~~We~~ human perform gestures to convey information or to interact with the environment. Gesture recognition can be found widely adopted in various applications such as aids for the hearing impaired people, sign language recognition, virtual environments navigation, automation control, factory manufacture tasks ~~has a wide variety of applications that include, but is not restricted, developing aids for the hearing impaired, recognizing sign language,~~

~~navigating in virtual environments, and automation of manufacturing tasks [1] and etc. This study is aimed to design and implement a smart wearable dissertation is concentrating on smart wearable~~ controlling system by recognizing the hand and fingers gestures ~~recognition~~ in real-time.

The following sections in ~~This~~ chapter discussed the motivation and research objective, contributions and thesis organization ~~of this dissertation~~.

## 1.1 Motivation and Research Objectives

Gesture recognitions technology are widely explored and implemented in academic and industrial areas in the past decade. Among the developed gesture recognition technology from industries, ~~V~~ision-based approach [2] is the most common method to detect the user gestures ~~via via video camera video camera~~. ~~Thise~~ computer vision-based techniques interpreted the motions into information or commands by the means of acquiring, processing, analyzing, and understanding images the of gestures of human body and such as hand movements, facial expression and head motionsorientation. For ~~exampleinstance~~, the Vicon system [3] captured the motions by using consisting of multiple infrared high-speed infrared cameras and further processed by an-its associated embedded software-can be used to capture the motion data [3]. HoweverUnfortunatelly, the vision-based approach is prone to be affected by several aspects, for instance, background environment, video frequency, detected boundaries and etc.

On the other hand, other popular gesture recognitions techniques employed Kinect sensors and physiological sensors ~~for to detect the human motion-detection~~. ~~Thesese~~ typical sensors involved are ~~typically the~~ depth sensor, motion sensor, flex sensor,



~~and multi-array microphone sound sensor and others. Those methods too suffered from several. However there are a lot of limitations. For instance, for example using the electromyography (EMG) sensor which measured the muscle activity to detect certain motions are limited by the requirement of ,the system needs to use electrodes (wet or dry type). [4] to be attached on the specific parts of human body which [4] that are uncomfortable and unfriendly-disturbing to users. Given such limitations, those gesture methods are not well-suited to be adopted as controllers for home appliances, automations and etc.~~

~~Therefore, practically, it needs a portable and smart device that is not only wearable, but also high accuracy in recognizing gesture. In order~~

~~Thus, to solve problems, this dissertation study proposed a smart wearable controlling quaternion-based gesture recognition system that could resolve aforementioned issues by analyzing the motions of hand and fingers-gesture recognition. This system is a quaternion-based gesture recognition system. The goal of this system is to develop a smart wearable gesture recognition system by analyzing the motion of hand and fingers. In order to reduce the discomfort of the wearable device, In order to reduce the redundancy of the sensors, this study only aimed to utilize only 4 three inertial motion unit (IMU) sensors to detect the motions. Those sensors are which are placed on the thumb, index finger and back side of the palm. To demonstrate the feasibility of the proposed system, this study evaluates the gesture recognitions performance by introducing three levels of gesture difficulties, from simple (one direction) to complex gestures (involved multiple directions) with total of 18 gestures. In addition, 4 gestures involved only fingers are included as well~~

~~This study evaluates the performance of the gesture recognition by introducing a dictionary of 18 gestures from simple gestures to complex gestures and 4 gestures for~~

~~finger gesture, in particular. Meanwhile, a mobile-based home appliances controller application prototype is developed to simulate the sensitivity and accuracy of the proposed gestures. The IMU sensors data are~~The IMU sensors data are transmitted to ~~at the application stimulated application built on a mobile device~~ through Bluetooth communication ~~and further analyzed by the embedded. The gestures are determined by an implemented~~ dynamic time warping (DTW) classification ~~method in the stimulated application to correctness of the detected gestures.~~

## 1.2 Contributions

Throughout the research progress, it had been contributed during the system implementation. Collaboration works are required in both hardware and software side to perfect the system. The fundamental ideas are as follows:

- ①. Full design of a smart wearable controlling system is described.
- ②. Studying and developing DTW algorithm for gesture recognition and further improve the algorithm for higher accuracy.
- ③. Developing cross-platform application to simulate the detected gestures and extend applications for the system on the mobile device.
- ④. Evaluating the system to optimize for standalone system.

## 1.3 ~~Dissertation~~ Thesis Organization

~~This thesis e above dissertation~~ described the motivation, research objectives and ~~contributions~~ distribution in developing the proposed smart wearable controlling system ~~by using hand and finger gesture recognition. The contents of the rest chapters are as followed.~~ Chapter 2 briefly introduced the related works and researches that

had been ~~done-conducted by other researchers in the past decades~~. Chapter 3 introduced the architecture and design of the system. ~~Next e~~Chapter 4 described the ~~processing of system flow with experiments setup, the experimental design, and the environment setup and~~ finalized by the results ~~data analysis. d~~Discussion on the ~~principle results are also presented in chapter 4. Finally~~Lastly, chapter 5 summarized ~~the-concluded the thesis with work done and described~~ the potential future work ~~which-that~~ could further improved the performance of the proposed system.

## 2 Literature Review

Generally, ~~h~~Hand gesture recognition is ~~generally divided~~categorized into static ~~gesture recognition~~ and dynamic gestures ~~recognition~~. Static gesture recognition analyzed the hand sign without the requirement of any hand movements, typically with static image processing is the recognition of hand shape, read out the meaning of hand expression. In contrast to static hand gesture, the generation of ~~d~~Dynamic hand gestures ~~recognition is the recognition operation based~~ required movement of hand, fingers or both based on the ~~on obtained~~ trajectory parameters throughout a consecutive time frames period, ~~which is described by the space characteristics changing with time.~~ Various algorithms are developed for the static gestures recognition with computer vision processing methods, such as neural network (NN), Bayes network and random forest (RF). However, these static-based methods required a trained classifier and not suitable for real-time cursor control. As such, dynamic-based methods gained its popularity to replace the static-based methods as cursor control in recent research trends.

~~Algorithm on static gesture recognition have developed rapidly in recent years, such as gesture recognition based on artificial neural network and computer vision. However, simple static gesture cannot meet the requirements of the industry application and methods on dynamic gesture recognition become a focus in the area of research instead.~~

This chapter discussed the current works on the gesture recognitions technology worldwide covered from academic researches to industrial fields which incorporates

various types of sensors in augmented reality (AR) to virtual reality (VR). Two main methods are discussed here: vision-based and non-vision-based methods.

## 2.1 Vision-based Method

Vision-based method [5] recognized the gesture patterns by analyzing the hand postures in images. Figure 2-1 showed the architecture and design of the typical vision-based method gesture recognition system.

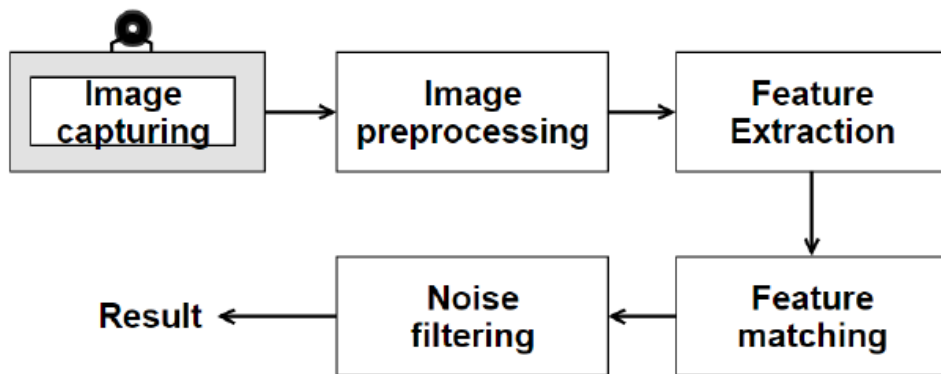


Figure 2-1 Vision based hand gesture recognition system.

One simplest way to identify the fingers from the images is to assign and segment the color markers for each fingers on the glove. Lamberti et al. [6] proposed a hand gesture recognition system by distinguish three different colors on a customized wool glove with a learning vector quantization classifier. Those colors are represented as palm, fingers, and other parts of hand respectively. Likewise, Iwai et al. [7] proposed an equivalent approach that divided the color glove into twelve regions with twelve different colors which consisted of a wrist region, ten finger regions and a region corresponded to the rest part of a hand.

## 2.2 Non-vision-based Method

Electromyography (EMG) sensor [8] is oOne of the most common gesture recognition using physiological sensors ~~are utilizing the EMG sensor [5]. The~~ EMG measures the electrical potentials generated from the muscle contractions in responses to different gestures. Indeed, mMost studies ~~used~~ ~~regarded to~~ surface-EMG (sEMG) for gesture recognition. The sEMG are-is usually focusing-employed by attaching the sensor electrodes on the specific locations of arm, shoulder and hand ~~movement which is~~. Figure 2-4 illustrates in Figure 2-2 ~~the sEMG sensor system.~~

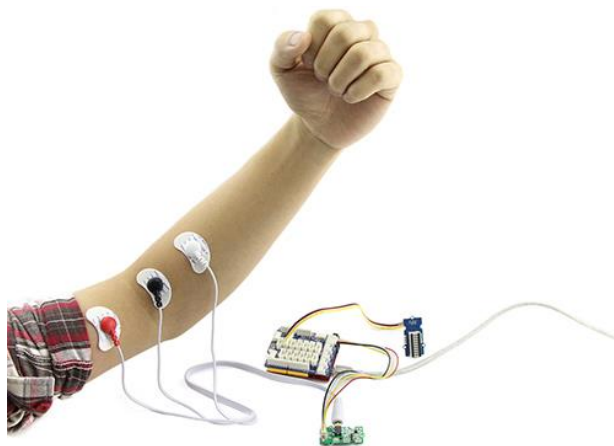


Figure 2-2 Hand gesture recognition system using sEMG-sensor system sensor.

Englehart et al. [9] conducted a study to recognize and discriminate the limb movement patterns by analyzing used pattern recognition to process four channels of EMG sensors attached on different parts of the limb, with the task of discriminating multiple classes of limb movement processed by using the feature vector of zero crossing rate (ZCR) and absolute mean value of sensors data. Momen et al. [10] used the recorded sEMGs from forearm extensor and flexor muscles to differentiate numerous type of hand movements by deploying fuzzy C-means clustering. On the

other hand, Uchida et al. [11] classified four types of gestures ~~type~~ with a neural network classifier based on the ~~extracted~~ features extracted using the using Fourier Transform (FFT) algorithmanalysis.

Meanwhile, motion sensors such as ~~Another common sensor is~~ IMU sensor is widely adopted for gesture recognition approach. The IMU sensor is a self-contained system that measures linear and angular motion, with built-in ~~usually with a triad of~~ three-axis gyroscopes and ~~triad three-axis of~~ accelerometers that converted these sensor data into 3-dimensional space orientations, in order to turn the sensor data into 3D space orientation in real time. Figure 2-3 ~~depicted shows~~ a wearable device developed by Gest Company [12] using IMUs for hand gestures recognizing recognition using motion sensors~~hand gestures~~.



Figure 2-3 A Wearable device uses IMUs for recognizing hand gestures.

Meanwhile, The most recent gesture recognition system that is accelerometer-based is the uWave. uWave was developed by Liu [13] is a user-dependent system that supports personalized and customized gestures recognition on mobile phones. uWave only utilizing one trained sample, stored as template for each gesture pattern. The

~~core processing method of uWave is DTW algorithm. uWave functions by utilizing only one training sample, stored in a template for each gesture pattern. The core of the uWave is DTW and the system's database undergoes two types of adaptations: positive and negative adaptation.~~ Meanwhile, Kim et al. [14] designed a glove that consisted of three tri-axis accelerometer sensors which are placed on the thumb, middle finger, and back of the palm respectively. The sensors created a 3-D hand motion tracking patterns that detected simple rule-based method gestures such as scissor, rock and paper based on sensors' horizontal (z-axis) and vertical (x-axis) angular positions. Similarly, Lu et al. [15] implemented a gesture recognition glove known as YoBu that utilized total of 18 inertial motion unit (IMU) sensors. Each finger consisted of three IMUs that are placed on the finger joints. The remaining three IMU sensors are placed on the arm, forearm and upper arm respectively. An extreme kernel-based learning machine algorithm is implemented to identify the specific gestures based on the 54-dimension extracted features.

Meanwhile, the utilization of flex sensor is another common approach that can identified the hand gestures by measuring the deflection or bending degrees of fingers as illustrated in Figure 2-4 which is developed by Ben [16].



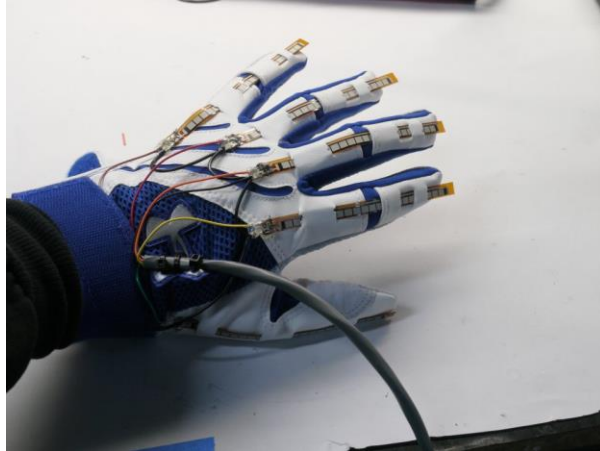


Figure 2-4 The flex sensor glove for detecting hand gestures.

On the other hand, Pathk et al. [17] presented a framework for hand gesture recognition in the study by utilizing the fusion of flex, accelerometer and contact sensors. The proposed framework detect the gestures by searching for the matches between the input sensor values with the stored template values for each respective signs. Dawane et al. [18] proposed a similar way to identify the gestures by template matching method based on the values of flex sensors that are attached on a glove corresponded to each finger. This method is improved by Patil et al. [19] which further divided the bending of each finger into three flexions. These flexions are categorized as straight (finger open), partial bend, and complete bend (finger close).

### 3 System Design

The design of the proposed system is shown in Figure 3-1. Essentially, the system is built up with three modules: sensor module, processing module and mobile application. The sensor module consisted of three IMU sensors attached on thumb and index finger, and back of the palm are communicated with the processing module through I2C protocol. The processing module remove the internal noises and motion artifact (i.e., hand shacking) of the received sensors data and further transmitted the filtered data to the mobile application via Bluetooth communication. Gestures are classified by the trained DTW model (see Chapter 4) and the respective pre-defined command in the prototype home controller application is triggered. The proposed smart wearable controller system is depicted in Figure 3-2.

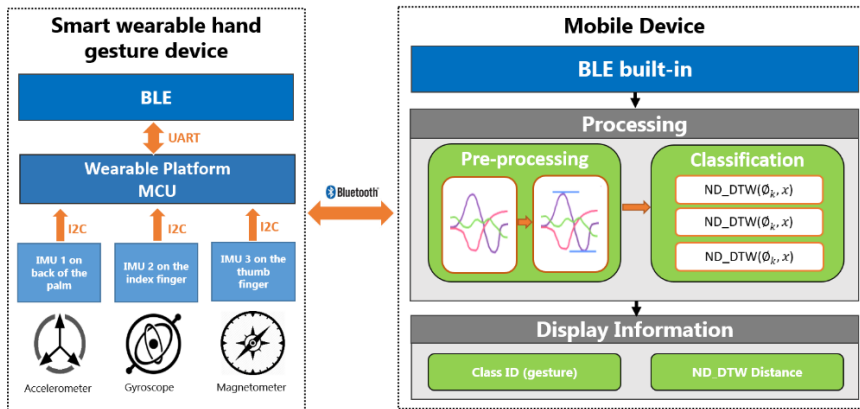


Figure 3-1 System design overview.

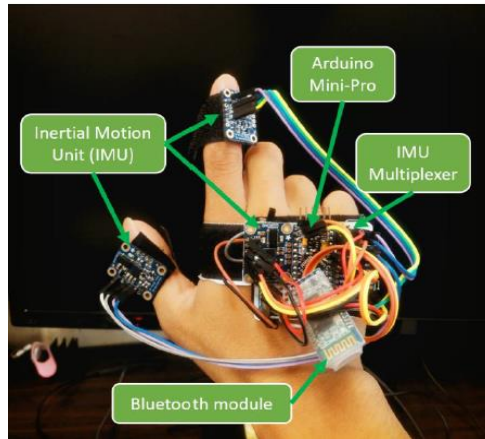


Figure 3-2 Proposed smart wearable controller system.

Our proposed system can be applied to various applications as described in Figure 3-3, i.e, home appliances and games controller. Qt cross platform software (see section 3.2) is used for developing the prototype mobile application in this study as it is capable to run on various software and hardware platforms with little or no change in the underlying codebase such as Android, iOS and embedded system (Raspberry Pi).

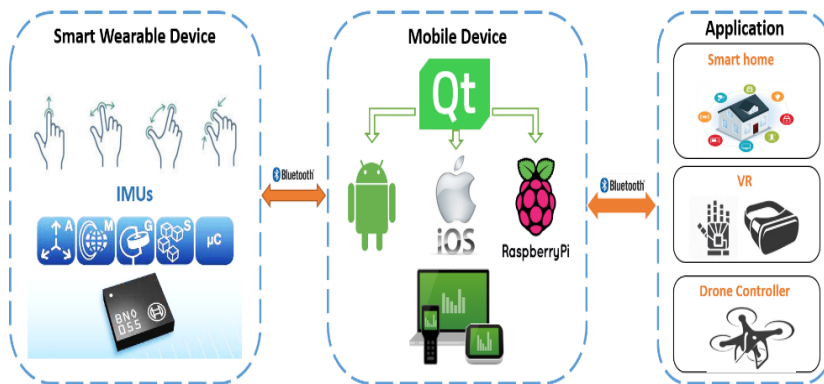


Figure 3-3 Proposed smart wearable controller system that can be applied to various applications with Qt cross-platform software.

## 3.1 Design of Hardware

### 3.1.1 Processing Module

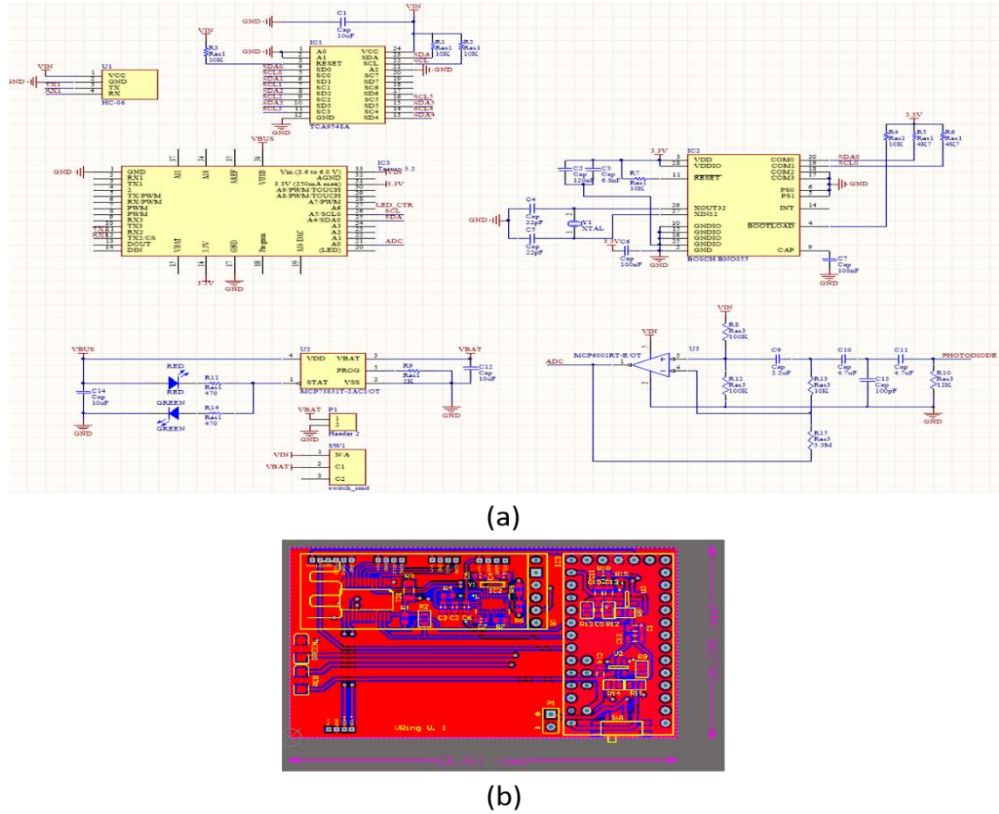


Figure 3-4 (a) Schematic and (b) PCB layer design of processing module.

The schematic design of the processing module is shown in Figure 3-4 (a). This processing module consisted of a microcontroller unit (MCU) (Teensy 3.2) [20], an IMU (BNO055) [21], an I2C multiplexer (TCA9548A) [22], Bluetooth module (HC-06) [23] and battery charger (MCP73831T) [24] which is powered by a 3.7 V 240

mAh lithium-ion battery. Table 3-1 listed the details of hardware specification for the proposed wearable controller system (Figure 3-2).

The BNO055 IMU is a 9-axis absolute orientation sensor consisted of an accelerometer, a gyroscope, and a magnetometer sensor (10-pins) operated at 3.3 V power which equipped with digital bi-directional I2C for data communication with processing module. The configuration of I2C protocol restricted multiple IMUs to communicate with the processing module using the same address using the same SDA/SCL pins. To resolve such issue, I2C multiplexer TCA9548A is adopted as communication medium between IMUs and the processing module. The TCA9548A multiplexer has eight bi-directional switches that are controlled by I2C bus. Each pair of SCL/SDA pins to connect each IMU could be configured by the setting of programmable control register. Thus, multiple IMUs data is able to be obtained successfully with a single MCU instead of three separated. The HC-06 Bluetooth module is responsible to transmit the IMUs value to the Bluetooth-enabled mobile device. Meanwhile, the proposed wearable system has the charging capability with the MCP73831 battery charger component which had advanced linear charging management controller, suitable to use in space-limited and low-cost system, powered up by USB cable.

Table 3-1 Processing module hardware specification.

Part	Components	Specification
IMU sensor [21]	Sensor	Tri-axial 16 bits gyroscope Tri-axial 14 bits accelerometer Geomagnetic sensor
	Voltage supplies Small package	2.4V to 3.6V Footprint 3.8 x 5.2 mm <sup>2</sup> Height 1.13 mm <sup>2</sup>
Teensy 3.2	Voltage supplies Processor	3.3V to 5V Cortex-M4, 72MHz (96MHz)

[20]	Flash Memory Ram EEPROM I2C	256 KB 64 KB 2 KB 2 ports
Bluetooth HC-06 [23]	Operating voltage Working frequency Transmission range	3.3V – 5V 2.4 GHz ISM band 10 m
I2C multiplexer TCA9548A [22]	Operating voltage Clock Frequency I2C	1.65V to 5.5 V 0 to 400 kHz 3 address pins, 8 I2C bus
Battery Changer MCP73831 [24]	Operating voltage Voltage Regulation Charge Current Charge Status	3.75V to 6V 4.2V, 4.35V, 4.4V, 4.5V 15mA to 500 mA Tri-State, Open-Drain Output

Figure 3-4 (b) depicted the layout of the proposed processing module's printed circuit board (PCB) layer. The most crucial issue in designing wearable device is the size and dimension of the device. In order to minimize the size of the PCB, all hardware components utilized the surface-mount technology (SMT). The SMT technology is a method for producing electronic circuits in which the components are mounted or placed directly onto the surface of PCBs. Subsequently, the device with SMT technology enabled hardware is known as surface-mount device (SMD).

### 3.1.2 Sensor Module

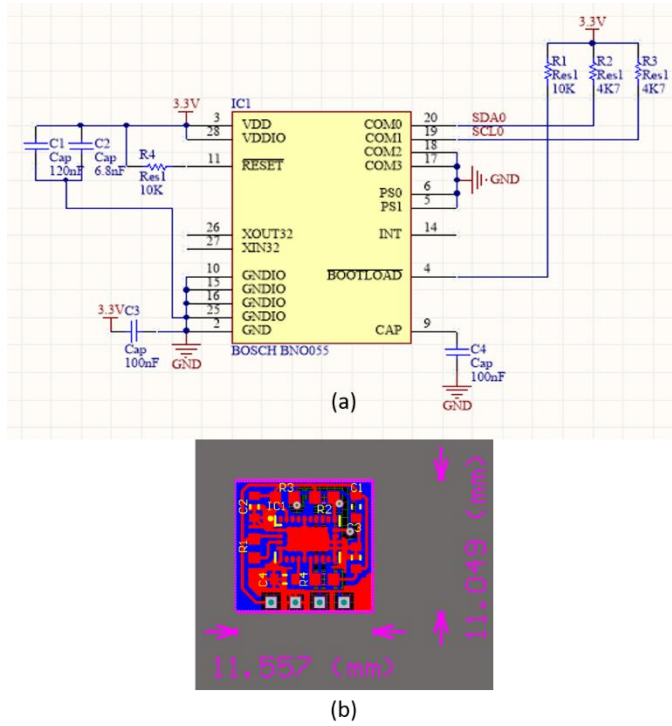


Figure 3-5 (a) Schematic and (b) PCB layer design of sensor module.

Figure 3-5 showed the schematic and PCB layer design of the sensor module. Two sensor modules are attached on the index finger and the thumb finger respectively which are connected to the processing module. Each sensor module consisted of an IMU (BNO055) sensor. IMU sensor on the back side of the palm utilized external crystal of 32.768 kHz to obtain high accuracy of sensor reading and in the meantime performed better with low power consumption. In contrast, IMU sensors in sensor module used their default internal clock with deviation of  $\pm 3\%$ , but overall the system accuracy is not affected.

### 3.1.3 3D-Printed Casing

3D printer has emerged as powerful tool for customized model design. Therefore, in this study, the processing module and sensor module are put into specifically designed processing module case and sensor module case which are fabricated by 3D printer (3DISON Printer [25]), in order to realize a real-life application. The fabricating material is Polylactic Acid (PLA) filament.

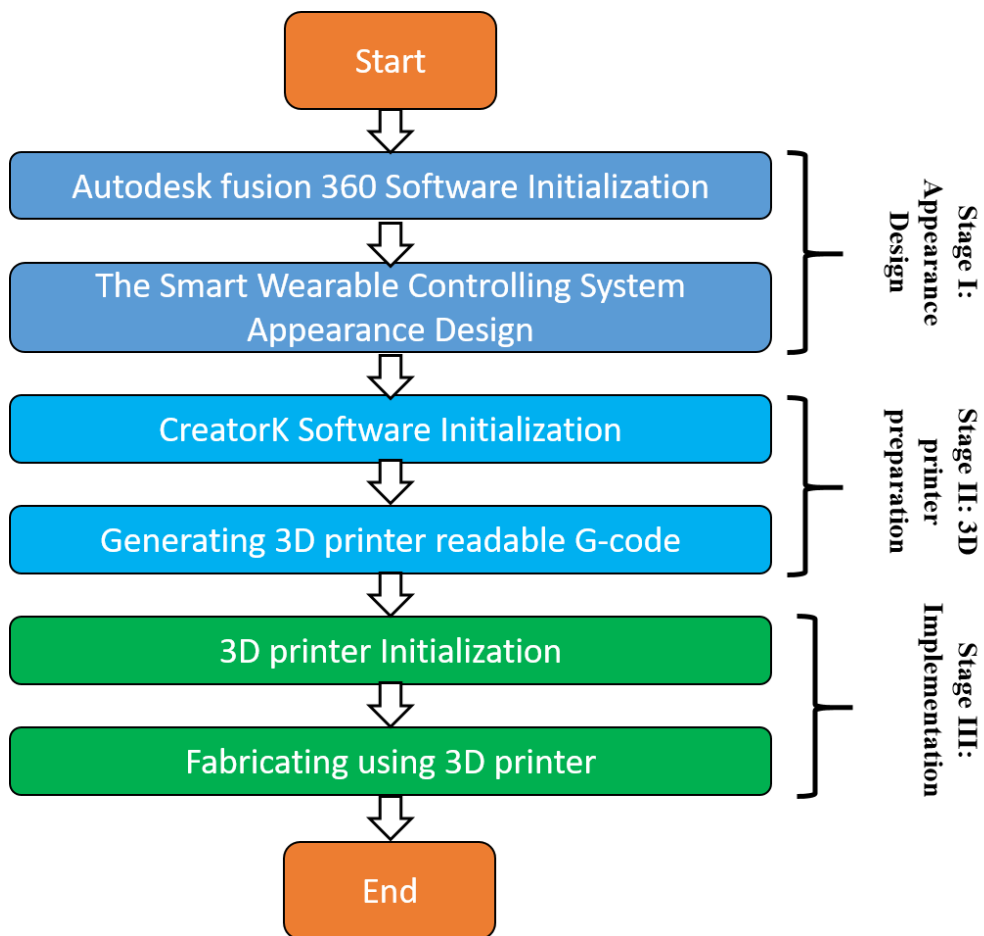


Figure 3-6 The flowchart of proposed system fabrication using 3D printer.



As shown in Figure 3-6, the design of 3D printer are in three stages – Stage I: appearance modeling of proposed system (see Figure 3-7) using Autodesk 360 Fusion software, Stage II: 3D printer preparation which involved the generation of 3D printer readable G-code from “.stl” format of Autodesk 360 Fusion; Stage III: Printing of 3D case.

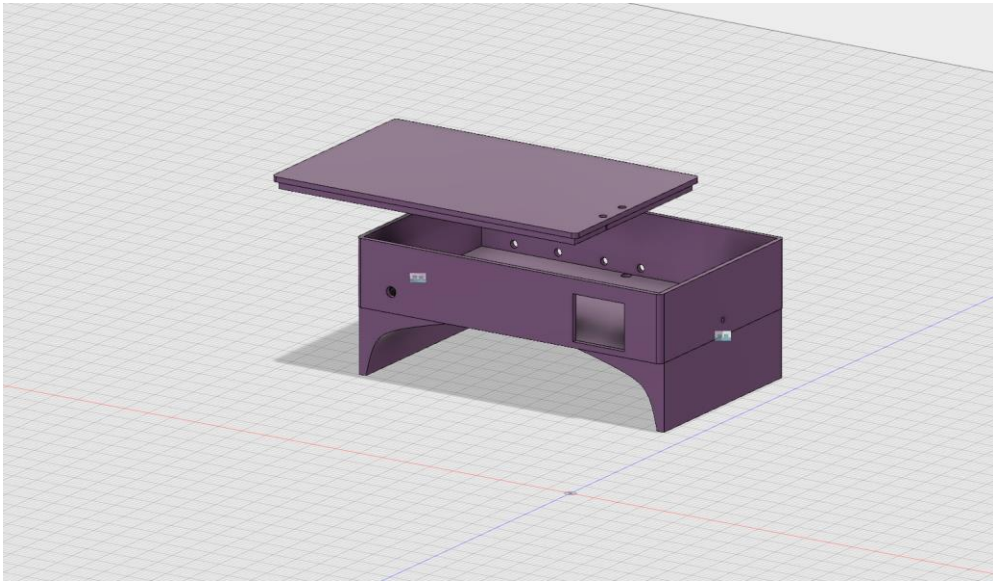


Figure 3-7 Appearance design (a) part design and (b) overview design.

## **3.2 Design of Software**

### **3.2.1 What is Qt?**

Qt [26] is a cross-platform application framework that is used for developing software that can be run on various software and hardware platforms with little or no change in the underlying codebase, while still being a native application with native

capabilities and speed. Supported platforms include Linux, OS X, Windows, VxWorks, QNX, Android, iOS and others as shown in Figure 3-8.



Figure 3-8 Qt cross-platform application development framework.

Qt is not a programming language on its own. It is a framework written in C++. A preprocessor, the MOC (Meta-Object Compiler) is used to extend the C++ language with features like signals and slots. Before the compilation step, the MOC parses the source files written in Qt-extended C++ and generates standard compliant C++ sources from them. Thus the framework itself and applications/libraries using it can be compiled by any standard compliant C++ compiler like Clang, GCC, ICC, MinGW and MSVC.

With Qt, GUIs can be written directly in C++ using its Widgets module. Qt also comes with an interactive graphical tool called Qt Designer [27] which functions as a code generator for Widgets-based GUIs. Alternative way to write GUIs with Qt is to use the QtQuick module [28]. GUIs using QtQuick are written in QML (Qt Meta-Object Language). QML is a declarative object description language that integrates Javascript for procedural programming. QtQuick provides the necessary modules for GUI development with QML. Usually, the whole application could be written in

QML, but in common cases, only GUI is written in QML instead of the whole application whereas the application's backend is written in C++.

### 3.2.2 Qt Creator – IDE

Qt comes with its own Integrated Development Environment (IDE), namely Qt Creator [29] as illustrated in Figure 3-9. It could run on Linux, OS X and Windows operating system. Moreover, it offers intelligent code completion, syntax highlighting, an integrated help system, debugger and profiler integration and also all major version control system integration compatibility. Qt Creator can be developed in Windows operating system and also could be used as add-on in Microsoft Visual Studio.

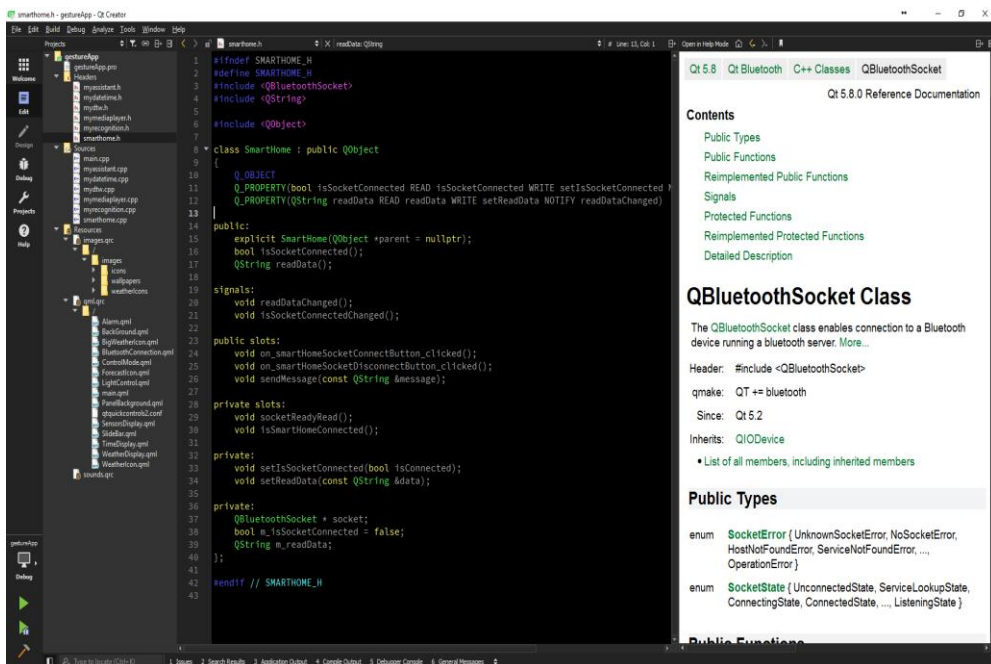


Figure 3-9 Qt creator IDE.

### 3.2.3 Libraries

Technically, Qt is divided into essential and add-on modules [30]. Qt essentials defined the foundation of Qt on all the software platforms. They are available on all supported development platforms and on the tested target platforms which are source and binary compatible for Qt5. Essential modules are general and useful for major Qt applications. The modules supported by the Qt essential are listed in the Table 3-2 Qt Essentials below.

Table 3-2 Qt Essentials.

Module	Description
Qt Core	Core non-graphical classes used by other modules.
Qt Gui	Base classes for graphical user interface (GUI) components. Includes OpenGL.
Qt Multimedia	Classes for audio, video, radio and camera functionality.
Qt Multimedia widgets	Widget-based classes for implementing multimedia functionality.
Qt Network	Classes to make network programming easier and more portable.
Qt QML	Classes for QML and JavaScript languages.
Qt Quick	A declarative framework for building highly dynamic applications with custom user interfaces.
Qt Quick Controls	Reusable Qt Quick based UI controls to create classic desktop-style user interfaces.
Qt Quick Dialogs	Types for creating and interacting with system dialogs from a Qt Quick application.
Qt Quick Layouts	Layouts are items that are used to arrange Qt Quick 2 based items in the user interface.
Qt SQL	Classes for database integration using SQL.
Qt Test	Classes for unit testing Qt applications and libraries.
Qt Widgets	Classes to extend Qt GUI with C++ widgets.

A special purpose module is treated as an add-on module which is available specifically on some software development platforms. Many add-on modules are either with full feature lists and backwards compatibility, or are only applicable to

certain platforms. Each add-on module had their own compatibility level. The list of Qt add-ons modules available is shown in the Table 3-3 below.

Table 3-3 Qt Add-Ons.

Module	Development Platforms	Target Platforms	Description
Active Qt	Windows		Classes for applications which use ActiveX and COM
Qt 3D	All		Functionality for near-realtime simulation systems with support for 2D and 3D rendering.
Enginio	All	All	A Backend-as-a-Service solution to ease the backend development for connected and data-driven applications.
Qt Android Extras	All	Android	Provides platform-specific APIs for Android.
Qt Bluetooth	All	Android, iOS, Linux and macOS	Provides access to Bluetooth hardware.
Qt Canvas 3D	All		Enables OpenGL-like 3D drawing calls from Qt Quick applications using JavaScript.
Qt Concurrent			Classes for writing multi-threaded programs without using low-level threading primitives.
Qt D-Bus	All		Classes for inter-process communication over the D-Bus protocol.
Qt Gamepad	All	Android, iOS, macOS, tvOS (including the tvOS remote), Linux, Windows	Enables Qt applications to support the use of gamepad hardware.
Qt Graphical Effects	All		Graphical effects for use with Qt Quick 2.
Qt Image Formats	All		Plugins for additional image formats: TIFF, MNG, TGA, WBMP.
Qt Location	All	All	Displays map, navigation, and place content in a QML application.
Qt Mac Extras	All	macOS	Provides platform-specific APIs for macOS.
Qt NFC	All	Android and Linux	Provides access to Near-Field communication (NFC) hardware.
Qt OpenGL (Deprecated)			OpenGL support classes. Deprecated in favor of the QOpenGL* classes in the Qt GUI module.
Qt Platform Headers			Provides classes that encapsulate platform-specific information, tied to a given runtime configuration of a platform plugin.
Qt Positioning	All	Android, iOS, macOS, Linux, WinRT.	Provides access to position, satellite and area monitoring classes.
Qt Print Support	All		Classes to make printing easier and more portable.
Qt Quick Controls 2	All		Provides lightweight QML types for creating performant user interfaces for embedded and mobile devices. These controls achieve improved efficiency by employing a simplified styling architecture when compared to Qt Quick Controls. These types work in conjunction with Qt Quick and Qt Quick Layouts.
Qt Quick Extras	All		Provides a specialized set of controls that can be used to build interfaces in Qt Quick.

Qt Quick Widgets	All		Provides a C++ widget class for displaying a Qt Quick user interface.
Qt Script (Deprecated)	All		Classes for making Qt applications scriptable. Deprecated in favor of the QJS* classes in the Qt QML module.
Qt SCXML	All	All	Provides classes and tools for creating state machines from SCXML files and embedding them in applications.
Qt Script Tools (Deprecated)	All		Additional components for applications that use Qt Script.
Qt Sensors	All	Android, Qt for iOS, WinRT and Mer.	Provides access to sensor hardware and motion gesture recognition.
Qt Serial Bus	Linux	Linux and Boot to Qt targets.	Provides access to serial industrial bus interface. Currently the module supports the CAN bus and Modbus protocols.
Qt Serial Port	All	Windows, Linux, and macOS.	Provides access to hardware and virtual serial ports.
Qt SVG	All		Classes for displaying the contents of SVG files. Supports a subset of the SVG 1.2 Tiny standard.
Qt WebChannel	All	All	Provides access to QObject or QML objects from HTML clients for seamless integration of Qt applications with HTML/JavaScript clients.
Qt WebSockets	All	All	Provides WebSocket communication compliant with RFC 6455.
Qt WebView	All	Platforms with a native web engine	Displays web content in a QML application by using APIs native to the platform, without the need to include a full web browser stack.
Qt Windows Extras	All	Windows	Provides platform-specific APIs for Windows.
Qt X11 Extras	All	Linux/X11	Provides platform-specific APIs for X11.
Qt XML			C++ implementations of SAX and DOM. Note: This module is not required to use QXmlStreamReader and QXmlStreamWriter. They are already part of Qt Core.
Qt XML Patterns			Support for XPath, XQuery, XSLT and XML schema validation.
Qt Wayland Compositor	Linux	Linux and Boot to Qt targets.	Provides a framework to develop a Wayland compositor.

### 3.2.4 Smart Home Application

In order to stimulate the effectively of the proposed smart wearable controller system, a smart home application as depicted in Figure 3-10 is developed in Android platform with encoded template of 7 hand gestures and 2 finger gestures. Once the Bluetooth connection is established between the wearable controller system and the smart home application, user can control the application with gestures. The functions recognized

by the gestures included displaying of date and time, temperature, humidity, dust level and UV level at home. Other controller functions included turning the lamps on/off, controlling the brightness of the lamps, displaying the weather forecast, alarm setting or playing songs. The summary of the gestures are listed as follow:-

- ①. Swipe up - turn on the lamp.
- ②. Swipe down - turn off the lamp.
- ③. Swipe left - display content of previous tab in application.
- ④. Swipe right - display content of next tab in application.
- ⑤. W shape - display weather forecast.
- ⑥. M shape - play music.
- ⑦. S shape - stop music.
- ⑧. Rotate left (finger gesture) – increase the brightness of lamp.
- ⑨. Rotate right (finger gesture) – decrease the brightness of lamp.

Figure 3-10 displayed the screenshot of the design of the smart home application. For the simulation purposes, the temperature sensor, humidity sensor, dust measurement sensor, UV sensor and relay for the lamps are connected to an Arduino UNO [31] which are connected to the smart home application via Bluetooth communication.

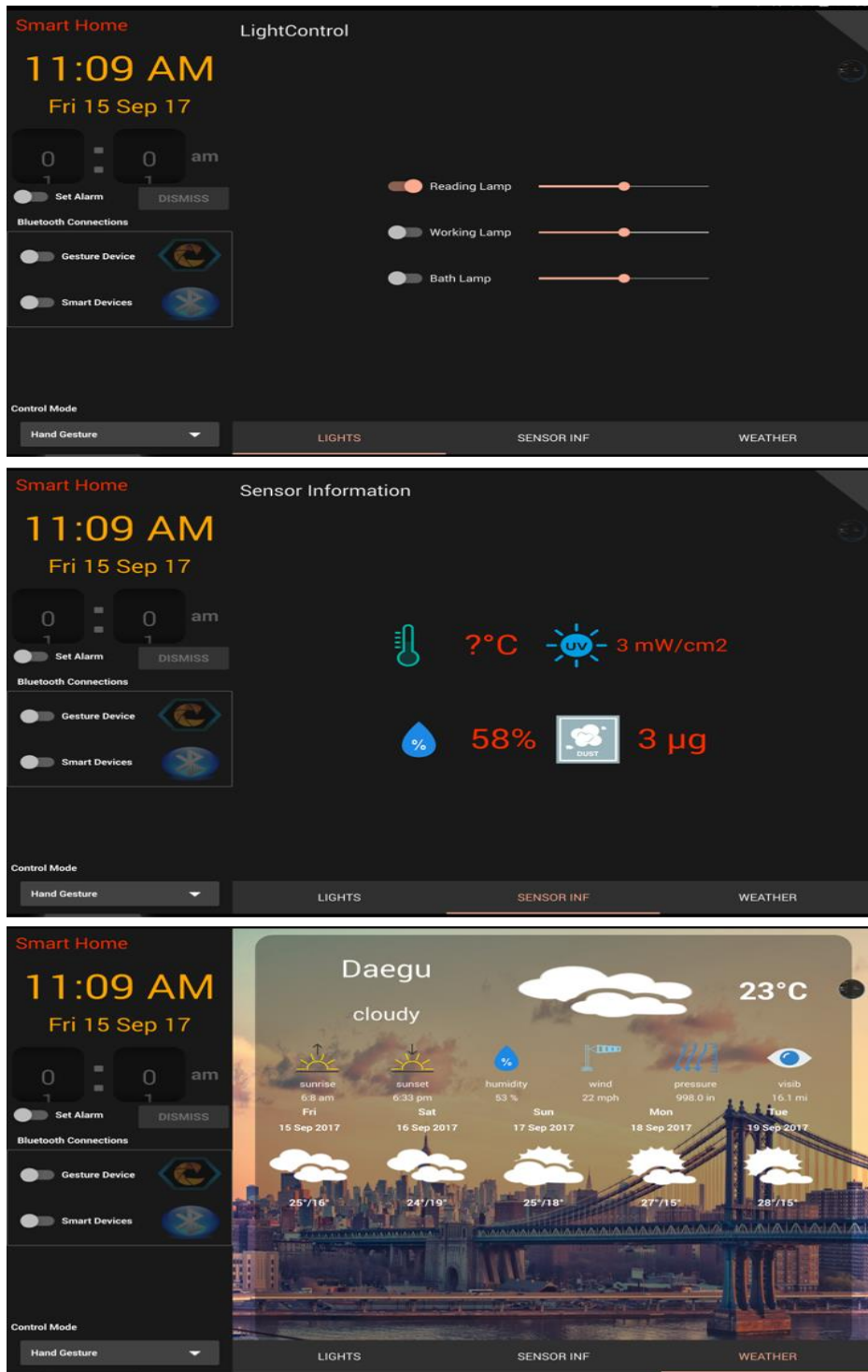


Figure 3-10 Screenshot of the smart home application.



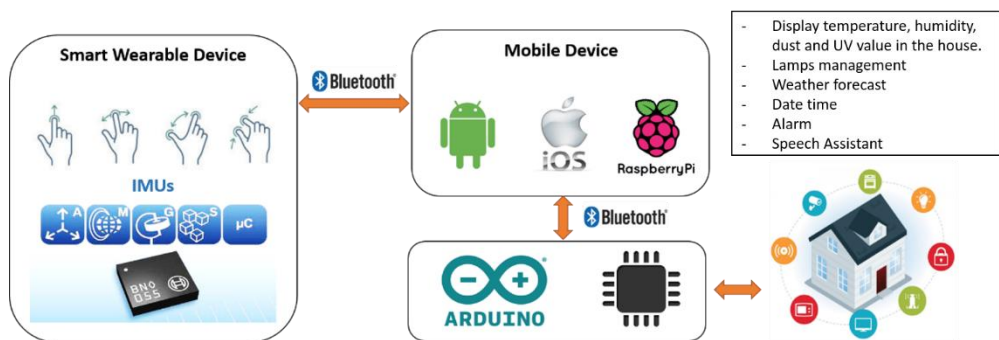


Figure 3-11 System overview of smart home application.

## 4 Experiment and Evaluation

### 4.1 Dynamic Time Warping Algorithm

In the proposed system, Dynamic Time Warping (DTW) is used to recognize the gesture pattern in real-time. DTW is an algorithm that can compute the similarity between two time-series signals, even if the lengths of the time-series signals do not match. One of the main issues to use distance measurement method (such as Euclidean distance) to measure the similarity between two time-series signals is that the result is often unintuitive. For instance, even two time-series signals are identical, but difference in phase, the distance measurement method will result in poor measurement. [32] as illustrated in Figure 4-1.

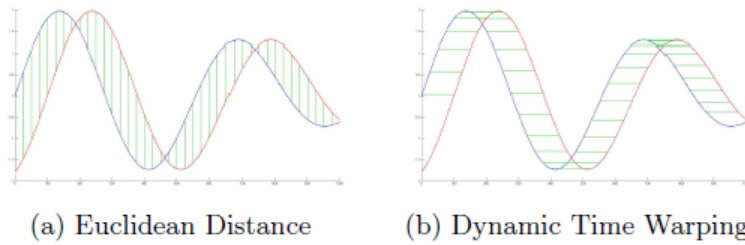


Figure 4-1 Two identical time-series signals, with phase differences measured with (a) Euclidean distance and (b) Dynamic Time Warping.

In fact, DTW overcame the limitation by ignoring both the local and global shifts in the time dimension [33].

### 4.1.1 One-Dimensional DTW

The foundation algorithm for DTW is as follows. Given two, one-dimensional, time-series,  $x = \{x_1, x_2, \dots, x_{|x|}\}^T$  and  $y = \{y_1, y_2, \dots, y_{|y|}\}^T$ , with respective lengths  $|x|$  and  $|y|$ , construct a warping path  $w = \{w_1, w_2, \dots, w_{|w|}\}^T$  so that  $|x|$ , the length of  $w$  is:

$$\max\{|x|, |y|\} \leq |w| < |x| + |y| \quad (1)$$

Where the  $k_{th}$  value of  $w$  is give by:

$$w_k = (x_i, y_i) \quad (2)$$

A number of constraints are placed on the warping path which are as follows:

- The warping path must start at:  $w_1 = (1, 1)$
- The warping path must end at:  $w_{|w|} = (|x|, |y|)$
- The warping path must be continuous, i.e if  $w_k = (i, j)$  then  $w_{k+1}$  must equal either  $(i, j)$ ,  $(i+1, j)$ ,  $(i, j+1)$  or  $(i+1, j+1)$
- The warping path must exhibit monotonic behavior, i.e the warping path cannot move backwards.

There are exponentially many warping paths that satisfy the above conditions.

However, we are only interested in finding the warping cost given by:

$$\min \frac{1}{|w|} \sum_{k=1}^{|w|} DIST(w_{k_i}, w_{k_j}) \quad (3)$$

Where  $DIST(w_{k_i}, w_{k_j})$  is the distance function (typically Euclidean) between point  $i$  in time-series  $x$  and point  $j$  in time-series  $y$ , given by  $w_k$ . The minimum total warping path can be found by using dynamic programming to fill a two-dimensional ( $|x|$  by  $|y|$ ) cost matrix  $\mathbf{C}$ . Each cell in the cost matrix represents the accumulated minimum

warping cost so far in the warping between the time-series  $x$  and  $y$  up to the position of that cell. The value in the cell at  $C_{(i,j)}$  is therefore given by:

$$C_{(i,j)} = DIST(i, j) + \min\{C_{(i-1,j)}, C_{(i,j-1)}, C_{(i-1,j-1)}\} \quad (4)$$

which is the distance between point  $i$  in the time-series  $x$  and point  $j$  in the time-series  $y$ , plus the minimum accumulated distance from the three previous neighbor cells of cell  $i,j$  (the cell above it, the cell to its left and the cell at its diagonal). When the cost matrix has been filled, the minimum possible warping path can easily be calculated by navigating through the cost matrix in reverse order, starting at  $C_{(|x|,|y|)}$ , until cell  $C_{(1,1)}$  has been reached, as illustrated in Figure 4-2. The process is repeated at each step until  $C_{(1,1)}$  has been reached where the minimum value is found. The warping distance between  $x$  and  $y$  is defined as

$$DTW(x, y) = \frac{1}{|w|} \sum_{k=1}^{|w|} DIST(w_{k_i}, w_{k_j}) \quad (5)$$

Here,  $\frac{1}{|w|}$  is used as a normalization factor to allow the comparison of warping paths of varying lengths.

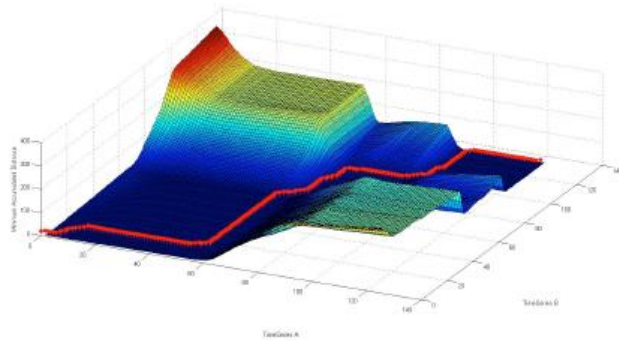


Figure 4-2 Cost matrix and the minimum warp path (indicated by the red line).

### 4.1.2 N-Dimension – ND DTW

The previous section described the standard implementation of DTW for two uni-dimensional time-series signals. However, in gesture recognition computation of time-series signals required multiple dimensions features. In this case, computation of 3-axis absolute orientation of IMUs. Thus, alternative DTW that could compute the distances between two N-dimensional time-series is a necessity as proposed by the [34], [35]. The measurement summed all the distances errors between the input N-dimensional time-series signals and the template signals as described below:

$$DIST(i, j) = \sqrt{\sum_{n=1}^N (i_n - j_n)^2} \quad (6)$$

## 4.2 Implementation

### 4.2.1 Pre-processing

Before the signals are being processed by the DTW, pre-processing is required to remove the signals noises. BNO055 sensor is the 9-degree-of-freedom (DOF) IMU sensor integrated with a MEMs magnetometer, gyroscope and accelerometer under a single die which are processed by a high-speed ARM Cortex-M0 processor. The speed and direction of the movements could be measured by accelerometer sensor and gyroscope sensor respectively. However, the gravity by earth was also included in the measurement where any forces applied to the object will affect measurement entirely. In fact, the accelerometer data is reliable only in the long run. As the result,

the measurements had the tendency to drift, not returning to zero when the system returned to its original position due to the overtime integration. In contrast, the gyroscope data is reliable only in the short term, and it will start to drift in the long run.

Therefore, to resolve such this issue, measurement by the fusion of accelerometer and gyroscope data with complementary filter are considered. The computation of complementary filter is described as (7):

$$\text{angle} = 0.98 * (\text{angle} + \text{gyrData} * \text{dt}) + 0.02 * (\text{accData}) \quad (7)$$

where gyrData and accData are gyroscope values and accelerometer from X-, Y- and Z-axis respectively whereas dt is sample rate to obtain data.

The angular data is updated repeatedly with the gyroscope readings and accelerometer readings that are processed to obtain the angular position by an atan2 function [36]. Basically,  $0.98 * (\text{angle} + \text{gyrData} * \text{dt})$  part resembles a high-pass filter on the integrated gyro angle estimate and  $0.02 * (\text{accData})$  part resembles a low pass portion acting on the accelerometer. Figure 4-3 illustrated the differences of angular measurement with and without the utilization of complementary filter.

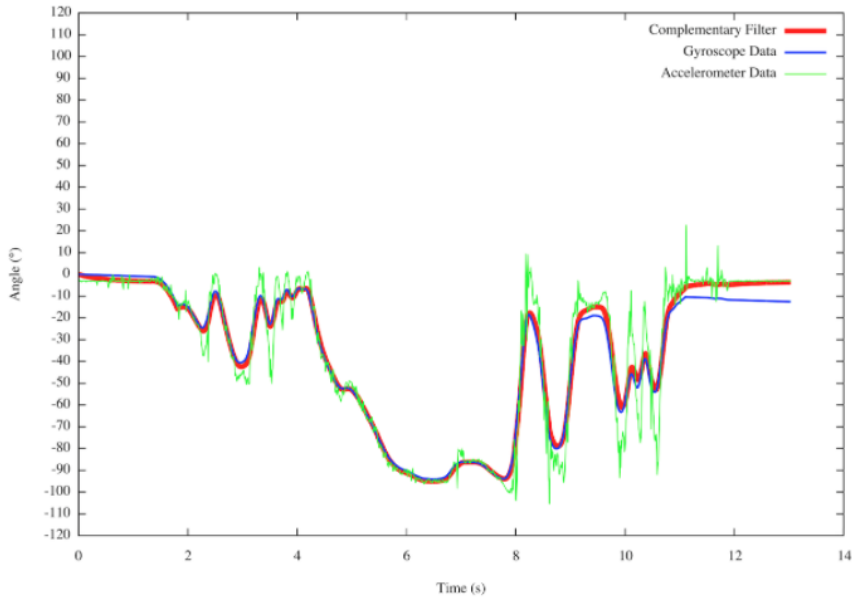


Figure 4-3 The comparison of angular data with and without complimentary filter.

DTW is a useful tool for computing the distance between two time-series. It is, however, a computational costly algorithm to use for real-time recognition, as every value in the cost matrix must be filled. Clearly this is unusable for real time recognition purposes, particularly if the unknown time-series is being matched against a large database of gestures. Hence, to improve the efficiency of DTW, the warping path is constrained in the sense that the warping path is not allowed to drift too far from the diagonal with small warping window. The size of the warping window can be controlled by varying the parameter  $r$ . The warping window is then set as the distance  $r$ , from the diagonal to directly above and to the right of the diagonal. This type of global constraint is referred to as the Sakoe-Chiba band [37] and Itakura [38]. On the other hand, to increase the efficiency of the signals detection and analysis, any intended gesture is processed only if the button attached on the device is pressed and hold.

### 4.2.2 Training Phase

Firstly, template is required to store all the intended gestures in database for template matching in gesture recognition by recording  $M_g$  training samples for each of the  $G$  gestures that are required to be recognized. After the training data has been recorded, each of the  $G$  templates can be found by computing the distance between each of the  $M_g$  training samples for the  $g_{th}$  gesture and searching for the training samples that provided the minimum normalized warping distances when compared with the other  $M_g-1$  training samples in that class. The  $g_{th}$  template ( $c$ ) is therefore given by:

$$\phi_g = \arg \min_i \frac{1}{M_g - 1} \sum_{j=1}^{M_g} 1\{ND\_DTW(X_i, X_j)\} \quad 1 \leq i \leq M_g \quad (8)$$

where the  $1\{.\}$  that surrounds the ND-DTW function is the indicator bracket, giving 1 when  $i \neq j$  or 0 otherwise,  $X_i$  and  $X_j$  are  $i_{th}$  and  $j_{th}$   $N$ -dimensional training examples for  $g_{th}$  gesture in the form of  $X = \{x_1, x_2, \dots, x_N\}$  and  $x = \{x_1, x_2, \dots, x_{|x|}\}^T$ . The ND-DTW function in (8) is the conversion of standard 1-D DTW to  $N$ -dimensions DTW:

$$ND\_DTW(X, Y) = \min \frac{1}{M_g - 1} \sum_{k=1}^{|w|} DIST(w_{k_i}, w_{k_j}) \quad (9)$$

$$DIST(i, j) = \sqrt{\sum_{n=1}^N (i_n - j_n)^2}$$



### 4.2.3 Classification Threshold

After the ND-DTW algorithm has been trained, an unknown gesture is able to recognize by taking its N-dimensional time-series  $\mathbf{X}$  and computed its total warping distance between  $\mathbf{X}$  and each of the  $G$  templates in the model (10) that resulted in the minimum normalized total warping distance as below:

$$c = \arg \min_g ND - DTW(\phi_g, \mathbf{X}) \text{ with } 1 \leq g \leq G \quad (10)$$

However, this method often give false positive results if the N-dimensional input time-series  $\mathbf{X}$  is not matched with any of the gestures in the template. This false classification issue can be resolved by defining a classification threshold for each gesture in the template during the training phase. If the computed distance is above the classification threshold, the algorithm will indicated that the particular gesture as no match.

$$c = \begin{cases} c & \text{if } (d \leq \tau_g) \\ 0 & \text{otherwise} \end{cases} \quad (11)$$

where  $d$  is the total normalized warping distance between  $\phi_g$  and  $\mathbf{X}$  and  $\tau_g$  is the classification threshold for the  $g_{th}$  template.

The classification threshold for each template can be set as the average total normalized warping distance between  $\phi_g$  and the other  $\mathbf{M}_g - 1$  training samples for that gesture with  $\gamma$  standard deviations:

$$\tau_g = \mu_g + (\sigma_g \gamma) \quad (12)$$

$$\mu_g = \frac{1}{M_g - 1} \sum_{i=1}^{M_g} 1\{ND - DWT(\phi_g, X_i)\} \quad (13)$$

$$\sigma_g = \sqrt{\frac{1}{M_g - 1} \sum_{i=1}^{M_g} 1 \left\{ \left( ND - DWT(\phi_g, X_i) - \mu_g \right)^2 \right\}} \quad (14)$$

where the  $1\{\cdot\}$  surrounded the ND-DTW function is the indicator bracket that output value of 1 if  $i$  is not equal to the index of the training samples with the minimum normalized total warping distance or value of 0 otherwise and  $X_i$  is the  $i_{th}$  training samples for the  $g_{th}$  class.  $\gamma$  could be initially set to any number during the training phase and later be adjusted by the user in the real-time gestures predication phase until a suitable classification/rejection level has been achieved.

### 4.3 Experiment Setup

Six right handed subjects are voluntarily participated in the experiments to validate the accuracy of the proposed smart wearable system. Subjects are requested to perform 25 repetitions of 22 gestures. The gestures consisted of 18 hand gesture and 4 fingers gestures. The defined 18 hand gestures are spanned the two planes: XZ and YZ planes. The gestures could be divided into simple gestures (only one direction), common gestures and complex gestures as illustrated in Figure 4-4. Moreover, 4 fingers gestures involved only fingers, namely zoom-out (increasing distance between thumb and index finger), zoom-in (decreasing distance between thumb and index finger), rotate right and rotate left as described in Figure 4-5. The data collected from all 6 participants will be referred to as the numbers-shapes dataset.

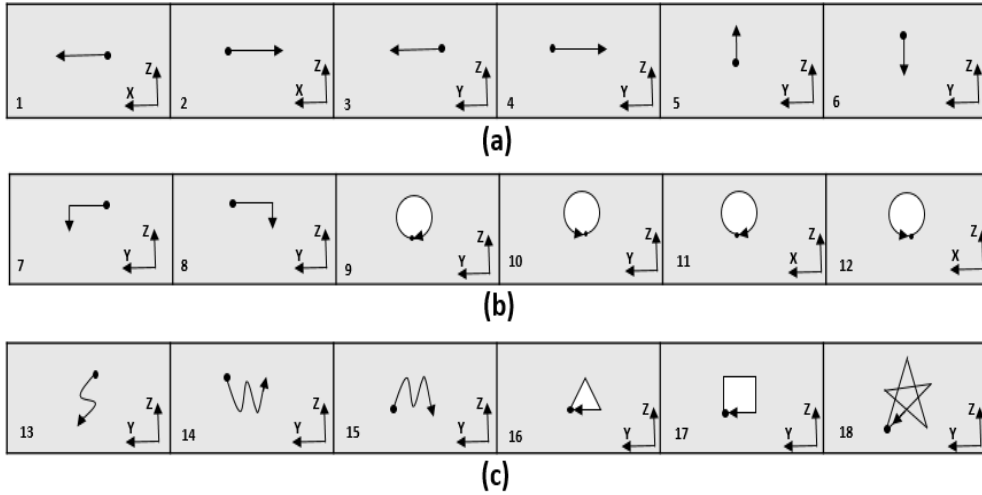


Figure 4-4 The predefined 18 gestures which are divided into (a) simple gestures, (b) common gestures, (c) and complex gestures.

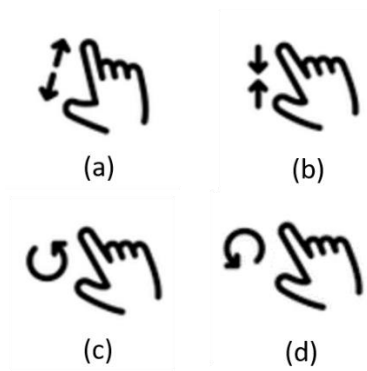


Figure 4-5 Four finger gestures which are (a) zoom out, (b) zoom in, (c) rotate right and (f) rotate left.

In order to distinguish between the total 22 gestures (hand and fingers gestures), all the unknown or non-recognized gestures will be treated as “unknown gesture”. An application is developed to serve the purposes for training to estimate the accuracy of gestures and testing in real-time as shown in Figure 4-6.

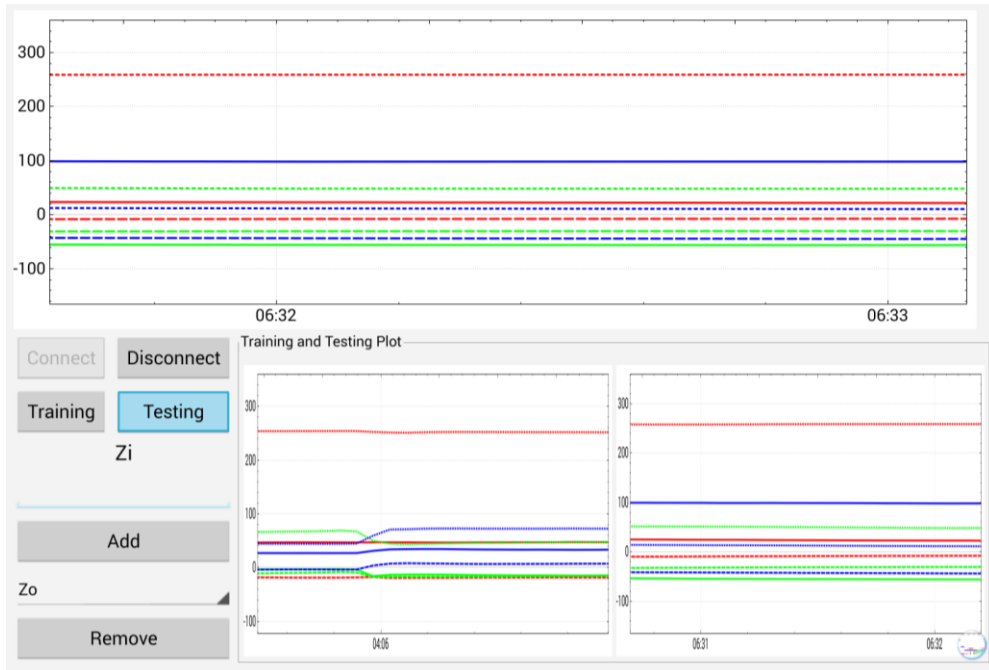


Figure 4-6 Self-development application for the purposes of training phase and testing phase.

## 4.4 Result and Discussion

Experiments are conducted to validate the gesture recognition accuracy rate utilizing the ND-DTW algorithm. In addition, the impact to accuracy rate based on the quantity of IMUs used are also taken into consideration. The ND-DTW model was trained using 6-fold cross validation method with each samples are treated as testing samples once.

Table 4-1 Gesture recognition accuracy rate with 1 IMU on back of the palm.



















Gesture/ Subjects	A	B	C	D	E	F	Mean Accuracy
	100%	100%	99%	100%	100%	100%	99.83%
	99%	100%	100%	100%	100%	99%	99.67%
	100%	100%	100%	99%	100%	100%	99.83%
	100%	99%	98%	100%	100%	99.00%	99.33%
	99%	100%	100%	100%	98%	100%	99.50%
	100%	98%	99%	100%	98%	100%	99.17%
	97%	100%	98%	96%	100%	96%	97.83%
	98%	100%	95%	100%	98%	100%	98.50%
	100%	98%	96%	98%	97%	95%	97.33%
	97%	98%	100%	96%	96%	98%	97.50%
	100%	97%	95%	96%	100%	97%	97.50%
	97%	96%	100%	98%	99%	98%	98.00%
	100%	98%	96%	95%	100%	97%	97.67%
	96%	96%	94%	98%	97%	95%	96.00%
	94%	98%	93%	95%	96%	100%	96.00%
	95%	95%	93%	98%	98%	95%	95.67%
	98%	95%	93%	90%	95%	95%	94.33%
	93%	90%	92%	93%	90%	95%	92.17%
Mean Accuracy	97.94%	97.67%	96.72%	97.33%	97.89%	97.72%	97.55%

Table 4-2 Table 4 1 Gesture recognition accuracy rate with 2 IMUs on the thumb and the index finger.





































Gesture/ Subjects	A	B	C	D	E	F	Mean Accuracy
	98%	99%	99%	100%	99%	100%	99.17%
	99%	100%	99%	98%	100%	99%	99.17%
	100%	99%	100%	99%	100%	99%	99.50%
	99%	99%	98%	100%	100%	99.00%	99.17%
	99%	100%	100%	99%	98%	100%	99.33%
	100%	98%	99%	100%	98%	100%	99.17%
	97%	100%	98%	96%	100%	96%	97.83%
	98%	100%	95%	100%	98%	100%	98.50%
	100%	98%	96%	98%	97%	95%	97.33%
	97%	98%	100%	96%	96%	98%	97.50%
	100%	97%	95%	96%	100%	97%	97.50%
	97%	96%	100%	98%	99%	98%	98.00%
	100%	98%	96%	95%	100%	97%	97.67%
	96%	96%	94%	98%	97%	95%	96.00%
	94%	98%	93%	95%	96%	100%	96.00%
	95%	95%	93%	98%	98%	95%	95.67%
	98%	95%	93%	90%	95%	95%	94.33%
	93%	90%	92%	93%	90%	95%	92.17%
Zoom in	93%	92%	93%	94%	95%	92%	93.17%
Zoom out	96%	94%	95%	94%	92%	93%	94.00%
Rotate right	96%	93%	97%	96%	98%	98%	96.33%
Rotate left	97%	91%	93%	97%	95%	93%	94.33%
Mean Accuracy	97.36%	96.64%	96.27%	96.82%	97.32%	97.00%	96.90%

Table 4-3 Table 4 1 Gesture recognition accuracy rate with 3 IMUs on back of the palm and the thumb and the index finger.

Gesture/ Subjects	A	B	C	D	E	F	Mean Accuracy
	98%	97%	98%	99%	100%	97%	98.17%
	98%	100%	98%	98%	100%	97%	98.50%
	98%	97%	100%	98%	100%	98%	98.50%
	97%	99%	98%	99%	96%	99.00%	98.00%
	99%	100%	98%	97%	98%	98%	98.33%
	99%	98%	99%	100%	98%	100%	99.00%
	98%	97%	98%	96%	98%	97%	97.33%
	96%	97%	96%	98%	98%	97%	97.00%
	100%	98%	96%	98%	97%	95%	97.33%
	98%	96%	96%	97%	96%	97%	96.67%
	97%	95%	96%	95%	97%	98%	96.33%
	94%	95%	94%	97%	96%	97%	95.50%
	97%	95%	94%	96%	97%	97%	96.00%
	94%	95%	93%	95%	95%	96%	94.67%
	94%	96%	94%	97%	96%	94%	95.17%
	97%	94%	97%	95%	94%	93%	95.00%
	95%	97%	93%	92%	94%	93%	94.00%
	92%	89%	89%	91%	94%	95%	91.67%
Zoom in	92%	89%	92%	93%	92%	94%	92.00%
Zoom out	93%	90%	94%	93%	92%	95%	92.83%
Rotate right	96%	93%	97%	96%	93%	93%	94.67%
Rotate left	95%	92%	92%	94%	92%	93%	93.00%
Mean Accuracy	96.23%	95.41%	95.55%	96.09%	96.05%	96.05%	95.89%

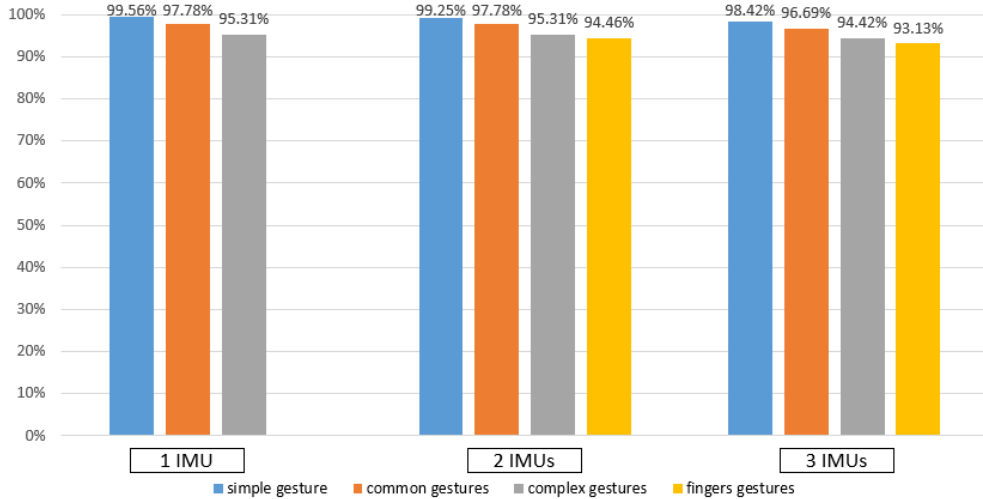


Figure 4-7 The accuracy rate of each groups in 18 hand gestures and 4 fingers gestures.

Table 4-1 depicted the accuracy of the gestures performed by all subjects using just one IMU data (1-IMU) on back of the palm. However, gestures performed with combination or solely with fingers are not recognizable by using only 1-IMU. On the other hand, Table 4-2 and Table 4-3 showed the accuracy of 18 hand gestures and 4 fingers gestures utilizing the 3 IMUs and 2 IMUs respectively. The results indicated that simple gestures had the highest accuracy in all 3 scenarios, followed by common gestures and lastly the complex gestures as shown in Figure 4-7.

Utilizing only one IMU data provided higher mean of accuracy rate and lower processing time followed by utilizing two IMUs and lastly three IMUs. This is due to the complexity of N-dimensional DTW which involved the similarity computation between two time-series where multiple IMUs might produce the higher mean errors as compared to just only one IMU.

In addition, according to the results in Table 4-2 and Table 4-3, among the 4 fingers gestures, rotate right gesture had the highest recognition accuracy rate of 96.33% and



94.67%. Meanwhile zoom in gesture had the lowest mean recognition accuracy among all subjects. The reason is that the subject B and C had a smaller hand size, thus, the distance between both thumb and index finger had shorter range compared to other subjects. Figure 4-8 illustrates a set of orientation samples obtained from the 3 IMUs with respect to the four gestures as stated below.

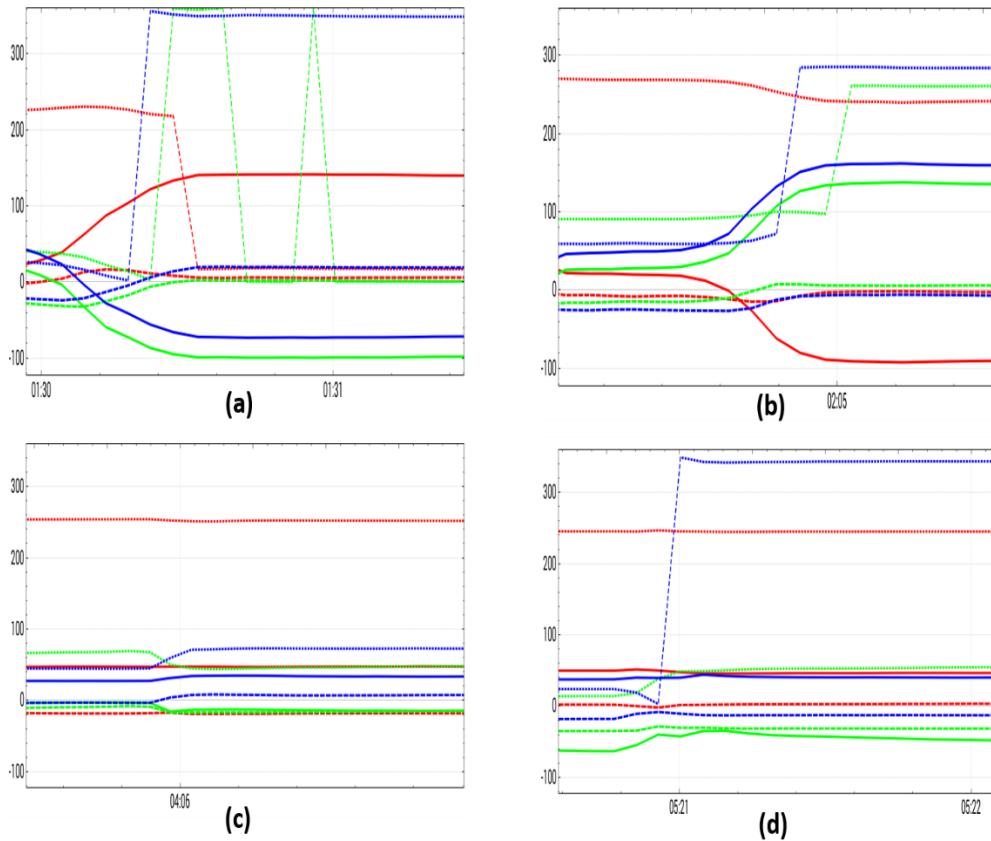


Figure 4-8 Orientation samples obtained from 3 IMUs to the gestures including (a) rotate right, (b) rotate left, (c) zoom in and (d) zoom out.

Figure 4-9 illustrates the description of the signals plotted in the figures above which are obtained from 3 IMUs with IMU 1, IMU 2, IMU 3 attached on the back of hand, the index finger and the thumb finger, respectively.

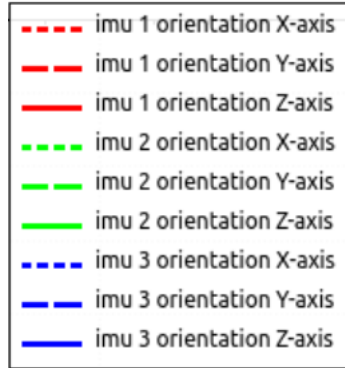


Figure 4-9 Sample of orientation obtained from 3 IMUs legend.

On the other hand, Table 4-4 indicates the comparison of the related existing system of gestures recognition with the proposed smart wearable controller system. Chang et al. [39] proposed a hierarchical hand motions recognition method based on one IMU sensor and two sEMG sensors, which is able to distinguish 6 gestures with an average accuracy rate of 95.6%. Meanwhile, Georgi et al. [40] evaluated the performances of a wearable gesture recognition system that captured the motions of arm, hand, and finger. He measured the movements and muscle activities at the forearm with an IMU and 16 sEMG sensors respectively to distinguish 12 different gestures with average accuracy rate of 97.8%. On the other hand, Khezri et al. [41] used 4 sEMG sensors to identify 6 gestures with an average classification accuracy rate of 92%. Likewise, Englehart et al. [42] showed high accuracy rate for 6 gestures at 98% by using 4 sEMG sensors. Liu et al. [13] presented the uWave for 8 gestures recognition with similar DTW algorithm using a single three-axis accelerometer with highest accuracy rate of 98.6%.

Table 4-4 Comparison of related systems.

Author	Type of sensors	Quantity of sensors	Number of gesture	Accuracy rate	Paper
Chang	IMU EMG	3	6	95.6%	[39]
Georgi	IMU EMG	17	12	97.8%	[40]
Khezri	EMG	4	6	92%	[41]
Englehart	EMG	4	6	98%	[42]
Liu	IMU	1	8	98.6%	[13]
Proposed study	IMU	1	18	97.55%	
Proposed study	IMU	2	22	96.90%	
Proposed study	IMU	3	22	95.89%	

## 5 Conclusions and Future Work

This study is focused on the design and implementation of smart wearable controller system using hand and fingers for gesture recognition. The system utilized 3 IMUs sensors, to recognize total of 22 pre-defined gestures with DTW algorithm with low cost implementation. Moreover, the study evaluated the IMUs quality impacted to the accuracy rate of system. By utilizing a single IMU, the system is capable of recognizing 18 gestures with accuracy rate of 97.55%. Meanwhile, gesture recognition by using two and three IMUs to detect 22 gestures are able to reach the accuracy rate of 96.90% and 95.89% accuracy rate, respectively. The proposed system is performed with low energy consumption at 10 hours operating time, powered with 3.7V – 240mAh battery. In fact, the proposed system can be applied to various application fields such as industry automation, gaming controller, robotics, sign language and etc.

Future works include:

- ①. Optimization of PCB design in term of size and dimensionality to reduce power consumption and its wearable comfortability.
- ②. Implementation of recognition algorithm in embedded hardware as standalone system which is compatible to different software platforms.
- ③. Recognition of more complex gestures that involved three dimensional gestures by utilizing hand and fingers motion.
- ④. Inclusion of more functionalities for various applications and allow gestures self-customization by users.

## References

- [1] S. Mitra and T. Acharya, " Gesture Recognition: A Survey," *IEEE Trans. Syst. Man Cybern. Part C Appl. Rev.*, p. 311–324, 2007.
- [2] Wu, Y. and Huang, T. S., "Vision-based gesture recognition: a review," *Proceedings of the International Gesture Workshop on Gesture-Based Communication in Human-Computer Interaction*, pp. 103-115, March 17 – 19, 1999.
- [3] L. Sigal, A. O. Balan and M.J. Black. Humaneva, "Synchronized video and motion capture dataset and baseline algorithm for evaluation of articulated human motion," *Int. J. Comput. Vis.*, p. 87, 2010.
- [4] C. Castellini and P. d. Smagt, "Surface EMG in advanced hand prosthetics," *Biological cybernetics*, pp. 35-47, 2009.
- [5] Y. Wu and T. S. Huang, "Vision-Based Gesture Recognition: A Review," *Gesture-Based Communication in Human-Computer Interaction Lecture Notes in Computer Science*, pp. 103-115, 1999.
- [6] L. Lamberti and F. Camastra, "Real-time hand gesture recognition using a color glove," *Int. Conf. Image Analy. Process.*, 2011.

- [7] Y. Iwai, K. Watanabe, Y. Yagi and M. Yachida, "Gesture recognition using colored gloves," *Proceedings of 13th International Conference Pattern Recognition*, 1996.
- [8] Q. Wang, X. Chen, R. C. Y. Chen and X. Zhang, "Electromyograph-based locomotion pattern recognition and personal positioning toward improved context-awareness applications," *IEEE Trans. Syst. Man. Cybern.*, pp. 1216-1227, 2013.
- [9] Englehart, K. and Hudgins, B., "A robust real-time control scheme for multifunction myoelectric control," *IEEE Trans. Biomed. Eng.*, pp. 848-854, Jul. 2003.
- [10] Momen, K., Krishnan, S. and Chau, T., "Real-time classification of forearm electromyographic signals corresponding to user-selected intentional movements for multifunction prosthesis control," *IEEE Trans. Neural Syst. And Rehabil. Eng.*, pp. 535-542, Dec. 2007.
- [11] N. Uchida, A. Hiraiwa, N. Sonehara and K. Shimohara, "EMG pattern recognition by neural networks for multi fingers control," in *14th Annual Int. Conf. Eng. Med. Biol. Soc.*, Paris, 1992.
- [12] Gest, [Online]. Available: <https://gest.co/>. [Accessed 20 3 2017].
- [13] J. Liu, Z. Wang, L. Zhong, J. Wickramasuriya and V. Venu, "uWave: Accelerometer-based Personalized Gesture recognition and Its Application," *Pervasive and Mobile Computing*, 2009.

- [14] J. Kim, N. D. Thang and T. Kim, "3-D hand motion tracking and gesture recognition using a data glove," *IEEE Int. Symp. Industrial Elec*, pp. 5-8, 2009.
- [15] D. Lu, Y. Yu and H. Liu, "Gesture recognition using data glove: an extreme learning machine method," *Proceedings of 2016 IEEE International Conference on Robotics and Biometrics*, pp. 3-7, 2016.
- [16] B. Krasnow. [Online]. Available: <http://benkrasnow.blogspot.kr/2010/12/diy-10-finger-flex-sensor-gloves-for.html>. [Accessed 17 10 2017].
- [17] V. Pathak, S. Mongia and G. Chitranshi, "A framework for hand gesture recognition based on fusion of Flex, Contact and accelerometer sensor," *2015 Third International Conference on Image Information Processing (ICIIP)*, pp. 312-319, 2015.
- [18] S. P. Dawane and S. H. G. A., "Hand gesture recognition for deaf and dumb people using gsm module," *Int. J. Sci. R.*, pp. 2226-2230, 2017.
- [19] K. Patil, G. Pendharkar and G. N. Gaikwad, "American sign language detection," *Int. J. Sci. R. Pub.*, pp. 1-6, 2014.
- [20] T. 3.2. [Online]. Available: <https://www.pjrc.com/teensy/teensy31.html>.
- [21] BNO055. [Online]. Available: <https://learn.adafruit.com/adafruit-bno055-absolute-orientation-sensor/processing-test?view=all>.
- [22] I. multiplexer. [Online]. Available: <https://learn.adafruit.com/adafruit-tca9548a-1-to-8-i2c-multiplexer-breakout/wiring-and-test?view=all>.

- [23] HC-06. [Online]. Available: <http://www.devicemart.co.kr/1272836>.
- [24] MCP73831. [Online]. Available: <https://www.sparkfun.com/datasheets/Prototyping/Batteries/MCP73831T.pdf>.
- [25] "3DISON," [Online]. Available: <http://www.3disonprinter.com/>. [Accessed 17 10 2017].
- [26] Q. Company, "About Qt," [Online]. Available: [https://wiki.qt.io/About\\_Qt](https://wiki.qt.io/About_Qt).
- [27] Q. Company, "Qt Designer Manual," [Online]. Available: <http://doc.qt.io/qt-5/qtdesigner-manual.html>.
- [28] Q. Company, "QtQuick," [Online]. Available: <https://wiki.qt.io/QtQuick>.
- [29] Q. Company, "Qt Creator," [Online]. Available: [https://wiki.qt.io/Qt\\_Creator](https://wiki.qt.io/Qt_Creator).
- [30] Q. Company, "Qt Modules," [Online]. Available: <http://doc.qt.io/qt-5/qtmodules.html>.
- [31] A. UNO. [Online]. Available: <https://store.arduino.cc/usa/arduino-uno-rev3>.
- [32] N. Gillian, R. B. Knapp and S. O'Modhain, "Recognition Of Multivariate Temporal Musical Gestures Using N-Dimensional Dynamic Time Warping," in *NIME*, 2011.
- [33] S. Salvador and P. Chan, "Toward accurate dynamic Toward accurate dynamic," *Intelligent Data Analysis*, pp. 561-580, 2007.



- [34] M. Ko, G. West, S. Venkatesh and M. Kumar., "Using dynamic time warping for online temporal fusion in multisensor systems," *Information Fusion*, pp. 370-388, 2008.
- [35] G. t. Holt, M. Reinders and E. Hendriks., "Multi-dimensional dynamic time warping for gesture recognition," *In Thirteenth annual conference of the Advanced School for Computing and Imaging*, 2007.
- [36] wiki, "atan2," [Online]. Available: <https://en.wikipedia.org/wiki/Atan2>.
- [37] H. Sakoe and S. Chiba, "Dynamic progeamming algorithm optimization for spoken wordrecognition," *Reading in speech recognition*, p. 159, 1990.
- [38] F. Itakura, "Minimum prediction residual principle applied to speech recognition," *Readings in speech recognition*, p. 154, 1990.
- [39] W. Change, L. Dai, S. Sheng, J. T. C. TAN, C. Zhu and F. Duan, "A hierarchical hand motions recognition method based on IMU and sEMG sensors," in *IEEE Conference on robotics and biomimetics*, Zhuhai, China, 2015.
- [40] M. Georgi, C. Amma and T. Schultz, "Recognizing Hand and Finger Gestures with IMU based Motion and EMG based Muscle Activity Sensing," *Proc. of the Int. Conf. Bio-inspired Syst. and Signal Processing*, pp. 99-108, 2015.
- [41] M. Khezri and M. Jahed, "A Neuro–Fuzzy Inference System for sEMG-Based Identification of Hand Motion Commands," *IEEE TRANSACTIONS ON INDUSTRIAL ELECTRONICS*, 2011.

- [42] K. Englehart, B. Hudgin and P. A. Parker, "A Wavelet-Based Continuous Classification Scheme for Multifunction Myoelectric Control," *IEEE TRANSACTIONS ON BIOMEDICAL ENGINEERING*, 2001.