ANALYSIS OF INERTIAL MEASUREMENT ACCURACY USING COMPLEMENTARY FILTER FOR MPU6050 SENSOR

NURUL HUDA BINTI ZANI

UNIVERSITI TEKNIKAL MALAYSIA MELAKA

ANALYSIS OF INERTIAL MEASUREMENT ACCURACY USING COMPLEMENTARY FILTER FOR MPU6050 SENSOR

NURUL HUDA BINTI ZANI

This report is submitted in partial fulfilment of the requirements for the degree of Bachelor of Electronic Engineering with Honours

Faculty of Electronics and Computer Engineering
Universiti Teknikal Malaysia Melaka

2021

DECLARATION

I declare that this report entitled "Analysis of Inertial Measurement Accuracy using MPU6050 for Emergency Department." is the result of my work except for quotes cited in the references.

Signature :

Author : Nurul Huda Binti Zani

Date : 15 June 2021

APPROVAL

I hereby declare that I have read this thesis and in my opinion, this thesis is sufficient in terms of scope and quality for the award of Bachelor of Electronic Engineering with Honours.

Signature :

Supervisor Name :

DR. JUWITA BT MOHD SULTAN
Pensyarah Kanan
Fakulti Kejuruteraan Elektronik Dan Kejuruteraan Komputer
Universiti Teknikal Malaysia Melaka (UTeM)
Hang Tuah Jaya
76100 Durian Tunggal

Date : 25 June 2021

DEDICATION

I wish to recognise my supervisor Dr Juwita Bt Mohd Sultan, for her counsel and direction in this thesis with genuine appreciation. Her determined feedback brought trust and certainty in me, indeed at the only discouraging minutes. She was a source of my motivation. Also, not forgetting my friend’s support has made studying at UTeM an enhancing and developmental experience. Even though the sees and conclusions communicated in this proposal stay my claim, my capacity to lock in with the talks about and talks remain my supervisor’s obligation and the other speakers in this UTeM instruction.

ABSTRACT

Inertial can be defined as disinclination to motion, action or change. The inertia of an object is the propensity to remain at rest or, if in motion, stays in motion at a steady speed. This project is to analyse the accuracy of the inertial measurement for the MPU 6050 sensor. MPU6050 is one of the low-cost ultrasonic range sensors that contains a 3-axis gyroscope for total 3-axis orientation measurement. It is used to analyse movement detection. As for this project, the application of the MPU6050 sensor is to the firefighter department by utilising accurate internal measurement. The accurate inertial measurement will help detect movement and the location of the firefighter in a rescuing operation. The sensor will then be connected wirelessly by using ESP8266 NodeMCU for the data transmission. Finally, the results can be viewed on an IoT platform. In order to improve the obtained results, a complementary filter is being used in this project. It is expected that the location's accuracy could be improved by 98%. Therefore, a precise location of the firefighter can be provided.

ABSTRAK

Inersia dapat didefinisikan sebagai disinklinasi terhadap gerakan, tindakan atau perubahan. Inersia objek adalah kecenderungan untuk tetap dalam keadaan rehat atau, jika bergerak, tetap bergerak pada kelajuan yang stabil. Projek ini adalah untuk menganalisis ketepatan pengukuran inersia untuk sensor MPU6050. MPU6050 adalah salah satu sensor jarak ultrasonik kos rendah yang mengandungi giroskop 3 paksi untuk pengukuran orientasi 3 paksi total. Ia digunakan untuk menganalisis pengesanan pergerakan. Untuk projek ini, penerapan sensor MPU6050 adalah untuk pemadam kebakaran dengan menggunakan pengukuran dalaman yang tepat. Pengukuran inersia yang tepat akan membantu mengesan pergerakan dan lokasi pemadam kebakaran dalam operasi menyelamatkan. Sensor kemudian akan disambungkan tanpa wayar dengan menggunakan ESP8266 NodeMCU untuk penghantaran data. Akhirnya, hasilnya dapat dilihat di platform IoT. Untuk meningkatkan hasil yang diperoleh, saringan pelengkap digunakan dalam projek ini. Diharapkan ketepatan lokasi dapat ditingkatkan sebanyak 98%. Oleh itu, lokasi tepat anggota bomba dapat disediakan.

ACKNOWLEDGEMENTS

This thesis has been attempted in halfway satisfaction of the Bachelor Degree of Engineering Electronic. I want to thank specific educators and individuals for their contributions to the creation of this research study. I need to thank my family with a genuine appreciation for their unlimited bolster. My earnest much obliged moreover go to my companions. I am too thankful to the graduates and researchers who take an interest in this think about. Uncommon appreciated going to the IEEE site to contribute to endorsing certain writings that made a different type in this report for all the courses.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF SYMBOLS AND ABBREVIATIONS

For examples:

IMU	:	Inertial Measurement Unit
INS	:	Inertial Navigation System
MEMS	:	Microelectromechanical System
VR	:	Virtual Reality

CHAPTER 1

INTRODUCTION

1.1 Introduction

The word inertial originally comes from inertia. The inertia of an object is the propensity to remain at rest or if in motion, stays in motion at a steady speed [1]. This project is to analyse the accuracy of the inertial measurement for the MPU6050 sensor. This thesis focus on analysing the accuracy of the MPU6050 sensor by using the complementary filter. The inertial measurement unit is applied to measure the accuracy in terms of 3-axis accelerometers and 3-axis gyroscopes. In this chapter, the background of the MPU6050 sensor is explained together with the problem statement. The next part is the objectives which define the aim of this project. Then, the general flow of this project is described in the scope section. Finally, the organisation of the thesis is mention in the outline chapter.

1.2 Background of Project

An Inertial Measurement Unit (IMU) is the centre of inertial positioning and navigation systems. Every IMU comprises at any rate three accelerometers and three gyroscopes (angular-rate sensors) [2]. Before IMUs are collected, gyroscopes and accelerometers are dependent upon independent tests. It is essential to decide their boundaries when they are parts of IMUs because the yield boundaries of gyroscopes and accelerometers are fixed to the reference axes of the IMU [3]. Along these lines, alignment is a mandatory stage in getting ready IMUs for an activity or adjusting inertial frameworks. By alignment, it implies that an IMU's bounds or errors have been identified and ready to be used in any application [4]. The MPU-6050 is the world's first and only 6-axis motion tracking devices designed for the low power, low cost, and high-performance requirements of smartphones, tablets and wearable sensors [5]. An assortment of utilisations is financially used by utilising the MPU6050 sensor as an elective instrument.

1.3 Problem Statement

Localisation plays a vital role for a particular organisation. The location's accuracy brings a huge impact, especially to the emergency responders team. While rescuing operations such as in a building, the movement and the position of the team members could not be determined, especially if there is an unexpected accident that occurred [6]. This project focus on applying the MPU6050 sensor used by the firefighters by utilising accurate internal measurement. MPU6050 is one of the low-cost motion tracking sensor that contains a 3-axis gyroscope and 3-axis accelerometer. IoT platform display the person's motion and orientation in terms of three axes.

Besides that, a complimentary filter is applied to improve the accuracy percentage. Therefore, the MPU6050 sensor’s inertial measurement could also be improved.

1.4 Objectives

i. To analyses the accuracy of the MPU6050 sensor’s inertial measurement.

ii. To improve the achievable accuracy rate to 95% using complimentary filter.

iii. To prompt the accuracy by visualising the result using IoT Platform.

1.5 Scope of Project

The project starts by collecting data using the MPU6050 sensor. The sensor is used to extract an Inertial Measurement Unit of MPU6050 that comprises a 3-axis accelerometer and 3-axis gyroscope. When the information needed is completed, it sends the client’s information by utilising ESP8266 NodeMCU. This data is at that point handled using the Arduino IDE program and communicated over a Wi-Fi interface to the IoT platform, such as a smartphone or a tablet, to view the motion and orientation of the emergency responder. The accuracy of the sensor is measured and optimised by using the complementary filter. The obtained results are displayed as in x,y and z-axis using the IoT platform.

1.6 Thesis Outline

This thesis outline is organised into five chapters to cover the research related to MPU6050. The following are the outline for the thesis:

1. Chapter 1

This chapter will cover the background of the Inertial Measurement Unit of the MPU6050 sensor. Also, the problems statement, objectives, project scope, and the project structure of the project is explained in this chapter.

2. Chapter 2

This chapter presents a literature review of the MPU6050 sensor's Inertial Measurement Unit. The three accelerometers and gyroscopes are described and explained. This page also includes previous research on this project.

3. Chapter 3

This chapter describes the methodology used in this project. First, a flowchart is used to describe the process in detail, beginning with a literature review, followed by the results collected, a strategy for enhancing accuracy, and lastly, a summary of the results.

4. Chapter 4

This chapter focuses on the analysis and discussion of the accuracy of the MPU6050 sensor's Inertial Measurement Unit, which will be gained at the end of the experiment. In addition, the results of using the complementary filter are also displayed on an IoT platform.

5. Chapter 5

This chapter concludes the project and discusses the project's future activities. This chapter also includes a summary of the project's results, and it ends with a suggestion for future work that could be done with this project.

CHAPTER 2

BACKGROUND STUDY

2.1 Introduction

An inertial measurement unit (IMU) is an electronic device that uses a combination of accelerometers, gyroscopes, and sometimes magnetometers to detect and report a body's specific motion, angular rate, and orientation [7][8]. Inertial Navigation Systems (INS) frequently use IMUs to calculate attitude, angular rates, linear velocity, and location relative to a global reference frame [9]. This section will focus on the theories, methods, and gaps in the existing research of the IMU sensor. The error characteristic is also explained along with the background of the inertial measurement unit (IMU). The last part contains developing an optimal inertial measurement unit (IMU) made by previous research to minimise the error characteristic and improve the accuracy.

2.2 Inertial Measurement Unit

A three-axis accelerometer and a three-axis gyroscope combined is known as an inertial sensor [10]. Inertial measurement units (IMU) are the names given to devices that contain these sensors. As demonstrated in Figure 1, inertial sensors are now included in almost every modern devices, such as Wii controllers and virtual reality (VR) headsets. Nowadays, many gyroscopes and accelerometers are based on microelectromechanical system (MEMS) technology. MEMS components are compact, light, and low-cost, with low power consumption and quick startup times [11].

Figure 1: Example of devices containing inertial sensors; A Samsung gear VR (left image) [12], a Wii controller having an accelerometer and a MotionPlus expansion device containing a gyroscope (right image) [13].

The number of applications for inertial sensors is constantly growing.In general, inertial sensors can offer information about the attitude of any object to which they are rigidly attached [14]. Furthermore, multiple inertial sensors can also be used to acquire information on the pose of separate connected objects. As a result, inertial sensors can be employed to track human motion as shown in Figure 2.

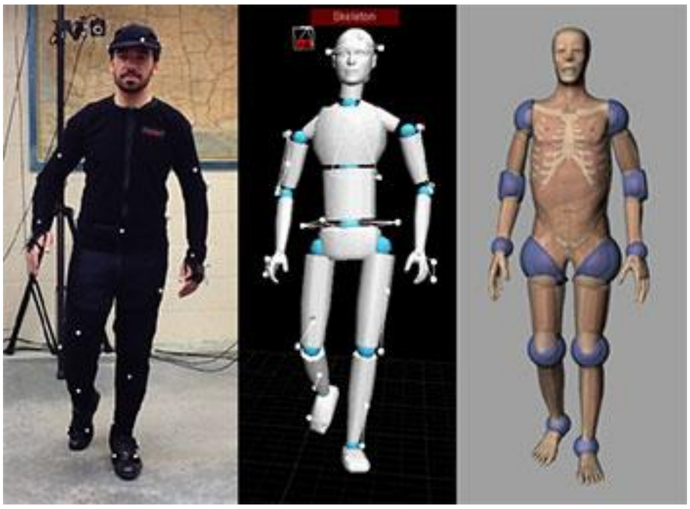


Figure 2: Actor wearing 20 IMUs to capture his real-life motion data (left) is acquired on a motion capture platform (centre) and used to determine the posture of the CHAD phantom (right) [15].

2.3 Error Characteristic of IMU

Several effects caused the error for IMU measurement. The gyroscope measures the angular velocity, whereas the accelerometer determines the particular force. There are, however, many reasons why this is not the case. A steadily changing sensor bias and measurement noise are two explanations that contribute to error inertial measurements [16].

2.3.1 Sensor Bias

The amount of departure or drift the sensor has from its mean value of the output rate is defined as bias stability (also known as bias instability). Essentially, the bias stability measurement determines how stable a gyroscope and accelerometer bias are over time [17]. Lower bias instability is advantageous since it leads to fewer errors being measured. The sensor errors in the inertial measurements are illustrated in Figure 3.

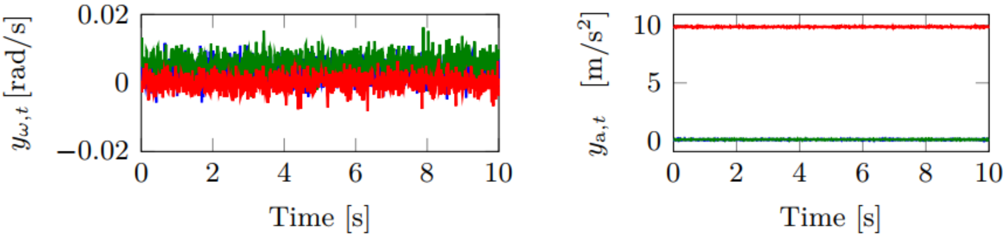


Figure 3: Inertial measurements for 10 seconds of static data conducted by [18]. The measures, as can be seen, are noise-corrupted and biassed.

2.3.2 Noise Measurement

These are the flaws that occur when measuring the IMU's output. White noise, random angle/velocity noise, flicker noise, angular rate/acceleration random walk, and ramp noise are among them [19]. AV technique is used to identify which noise terms exist in the IMU data, and the result is shown as below conducted by;

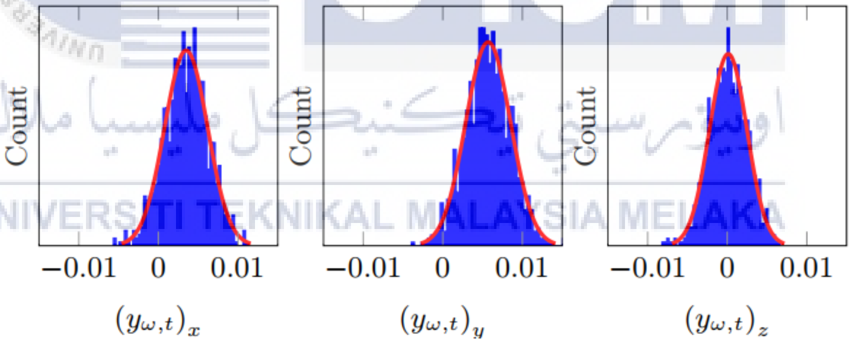


Figure 4: Histogram (blue) of the gyroscope measurements from a stationary sensor and a Gaussian fit (red) to the data. As can be seen, the measurement noise looks quite Gaussian [20].

2.3.3 Summary of error characteristics

Figures 3 illustrate gyroscope and accelerometer measurements for roughly 10 seconds of stationary data. The gyroscope is only supposed to measure the earth's angular velocity because the IMU is stationary. The gyroscope data, however, are distorted by noise, as shown in Figure 4. This noise appears to be Gaussian in nature. In addition, the measurements seem to be biased.

2.4 Previous Study

Sensor bias and measurement errors are the most common sources of estimate errors when using IMU for inertial estimation. In theory, all of these faults should be considered; therefore digital filter for optimising the IMU is introduce to reduce IMU errors.

2.4.1 Moving Average Filters

A consider that conducts by [21] is employing Moving Average Filters. It was executed to channel the IMU signal in real-time within the Arduino controller. Moving Average Filters takes a settled subset of the information focuses from a given arrangement of data. Indeed, even though this channel gave a sensibly great smoothing reaction, the channel’s reaction was ceaselessly slacking behind the input (Figure 5, Top) with variable offsets. It is found that the greatest counterbalanced between the sifted and unfiltered information to be around 9◦ (Figure 5, Bottom).

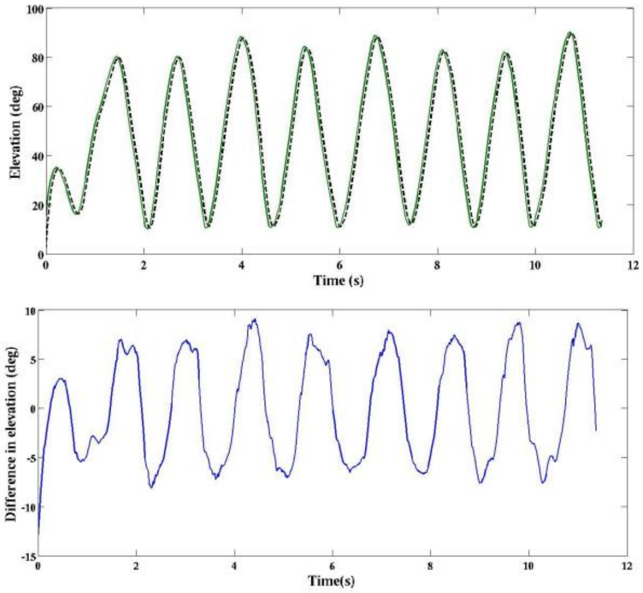


Figure 5. (Top) Execution of the Moving Average Filter. The continuous green line is the unfiltered information, and the dark sprint-line is the sifted reaction. (Bottom) The counterbalanced between the filtered and unfiltered data [22].

In outline, this consider analysed IMU estimation moving normal channels were executed within the controller to decrease the clamour, but it is found that these channels not proficient sufficient.

2.4.2 Complementary Filter

The complementary filter gives the best way to combine the accelerometer and the gyroscope information to get precise and responsive pitch/roll/yaw value. This channel is implied to determine one single value by combining two distinctive estimations. Centring on one case conducted by [23], the accelerometer flag produces high recurrence noise, whereas the gyroscope comes about to contain more recurrence noise. These information combination methods apply both low and high pass channels as communicated in Eq. (1):

$$Hs = HLP(s) + HHP(s) = 1 \tag{1}$$

Using this equation, this study overcomes the delay problem. The primary portion of Eq. (1) keeps up a high-frequency reaction, whereas low recurrence commotion is taken care of by the last-mentioned portion of Eq. (1). Sometimes recently getting real-time information, utilising MATLAB for Complementary filters to watch computational time, fetched and complexity contrasts. Overall, the execution exactness was moreover famous, as appeared in Fig. 6.

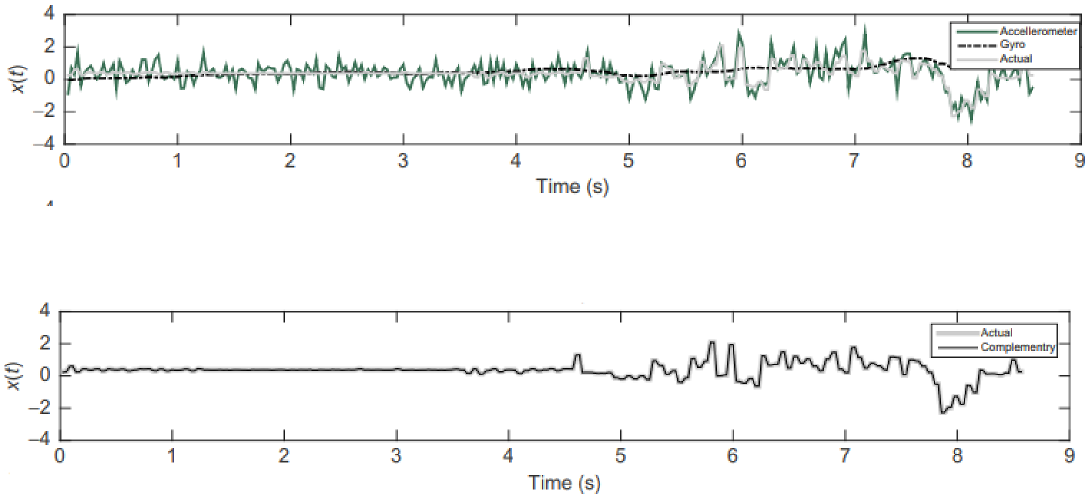


Figure 6. Actual vs Complimentary Filter graph [24].

Considering the complementary filter work does not depend on any suspicions for handle flow, subsequently, it does not endure the issues that the Kalman Filter should confront. With zero expectation computations, the complexity is low, and the handling time is short; the Complementary filter has demonstrated its worth, as seen within the exploratory comes about Figure 6.

2.4.3 Kalman Filter

The Kalman filter, named after Rudolf Kálmán, co-inventor of electrical engineering, offers a particular pick up from the combination of pulverisation and FIR filter. The term ‘filter’ that characterises the Kalman filter may be a bit of a misnomer. It is more practically equivalent to a “recursive estimator. Even though much more complicated than a single condition, we can streamline the utilise case here by dropping out the state networks and able to get the math appeared below:

$$X_k = K_k \times Z_k \ (1 - K_k) \times X_k{-}1 \tag{2}$$

Where:

X_k= Current estimation

K_k = Kalman gain

Z_k = Measured value

X_k–1 = Previous estimation

Each k is distinguishing discrete-time interims or tests of each sensor hub yield. The introductory Kalman picks up or covariance coefficients are utilised inside the IMU enrol settings to set up the anticipated relationship between the sensor yield vectors. The result of contemplating [25] is shown in Figure 6, which shows the height estimated from accelerometer data plotted within the best board. The footboard is designed to rise from the conjunction of the two sensors.

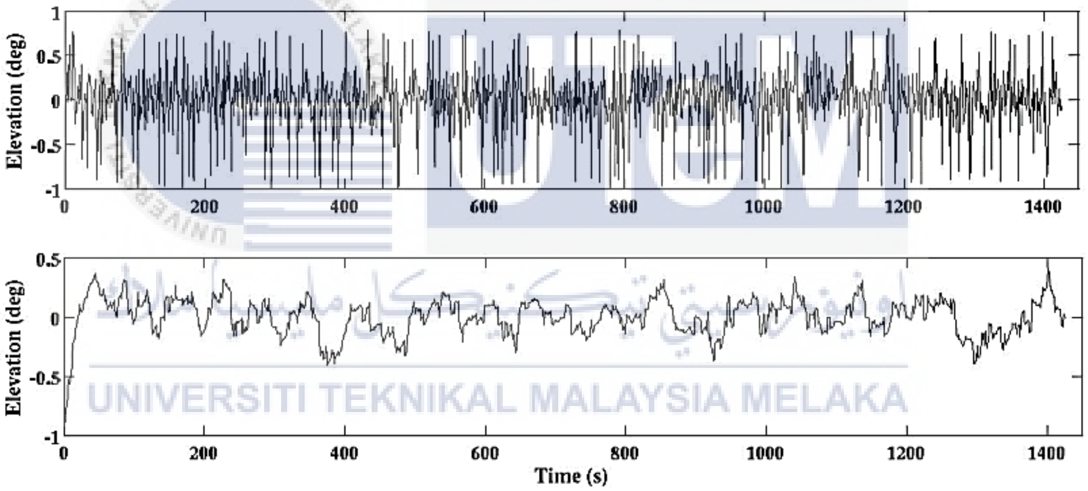


Figure 6: (Top) Rise calculated from the accelerometer versus time. (Bottom) value from accelerometer and gyroscope after the Kalman filter versus time [26].

2.4.4 Accuracy Comparison Method

A study conducted by [27] compared the relocation measured by HC-SR04 with the relocation calculated from the accelerometer information of MPU6050. The objective is to reply whether the information from an accelerometer alone is adequate to decide the robot’s uprooting with sensible precision. After Calibration, an HC-SR04 is set 150mm absent from the divider, as appeared in Fig. 9 for each run. The HC-SR04 unit can degree separations between 20cm to 150cm precisely with 1-1.5% of a mistake than physical estimations employing a tape ruler. Deducting the separate measured by the ultrasonic sensor from the beginning and last positions gives the ground truth information to compare the uprooting calculated utilising the accelerometer's information.

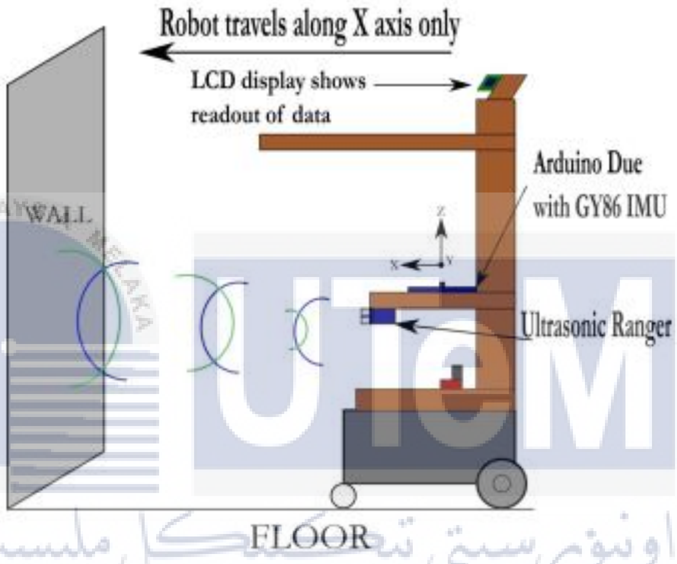

Robot travels along X axis only
LCD display shows readout of data
Arduino Due with GY86 IMU
z
x
y
Ultrasonic Ranger
WALL
FLOOR

Figure 7: Schematic diagram of the experiment conducted [28].

2.5 Project Significant

The IMU sensor application did not carries any harm to the environment and the mankind. The project is designed to transmit and receive data through WIFI transmission which is not emitting hazardous substances to humans and environment. This project considered to be safe to use on humans and environmentally friendly. The sustainability of this project also conserving; the life span of the component used in this project can last up to 7 to 9 years in storage.

2.6 Summary

A single IMU must be calibrated and filtered and an adequate method for estimating three displacements and orientation while experimenting. A filtered IMU sensor combination must be used to avoid the introduction estimation blunders offered by the error’s disturbance. For the accuracy measurement, the value of IMU reading must be compared with any reliable ground truth.

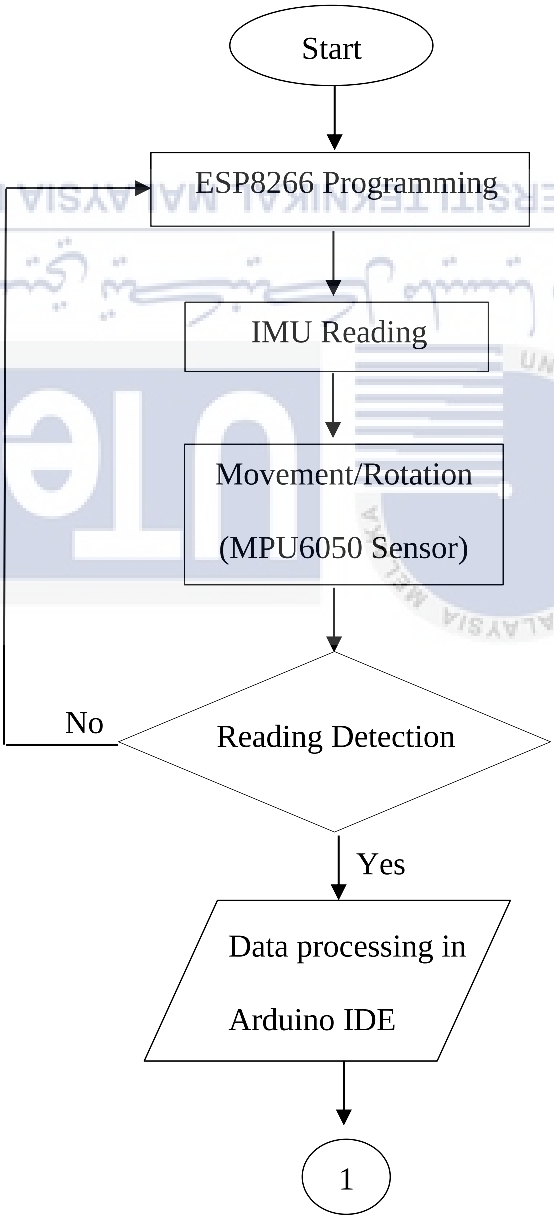
CHAPTER 3

METHODOLOGY

3.1 Introduction

This project consists of two parts, hardware and software part. The first part is the hardware part which explains details on the circuit construction and the components used. The schematic diagram for all the parts is also presented here. The second part is the software. Details on programming applied for this project is described. The use of a complementary filter is also shown in this section. The two primary devices for this project is the MPU6050 and ESP8266 NodeMCU. MPU6050 is a Micro Electro-mechanical system (MEMS), which consists of a three-axis accelerometer and three-axis gyroscope. It is used to measure velocity, orientation, acceleration, displacement and other motion like features.

Meanwhile, ESP8266 NodeMCU acts as an Access Point (AP) using Arduino IDE. This allows the project to connect directly via Wi-Fi without a wireless router. The project starts by collecting data using the MPU6050 sensor. The sensor is used to read the Inertial Measurement Unit of an emergency responder. When the information needed is completed, it sends the client's information by utilising ESP8266 NodeMCU. The graphical results are displayed using an IoT Platform. The block diagram of this project is shown in the figure below.


Start
ESP8266 Programming
IMU Reading
Movement/Rotation (MPU6050 Sensor)
Reading Detection
No
Yes
Data processing in Arduino IDE
1

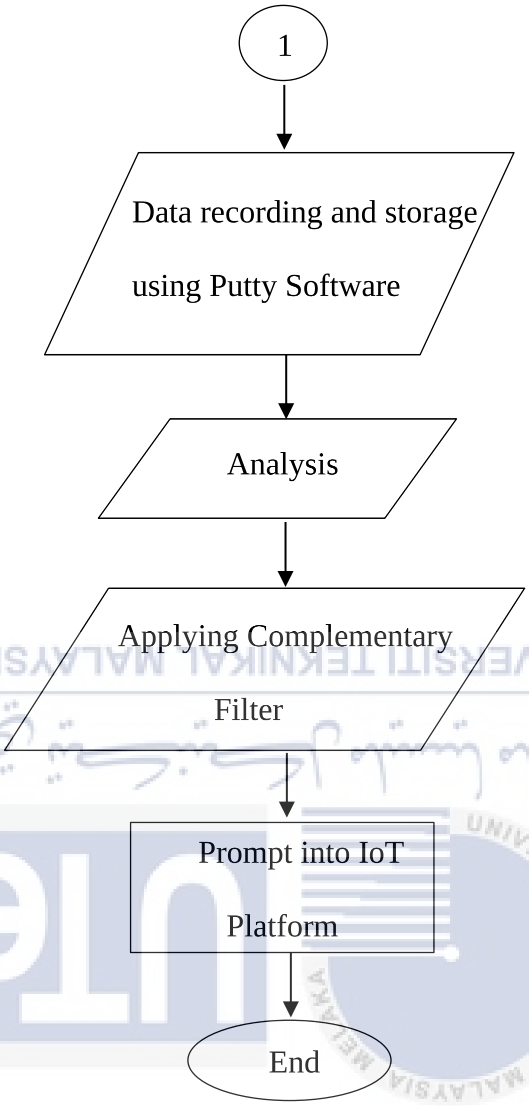

1
Data recording and storage using Putty Software
Analysis
Applying Complementary Filter
Prompt into IoT Platform
End

Figure 7. Block Diagram of the project.

3.2 Hardware Design and Development

This project consists of two primary devices, which is the MPU6050 and ESP8266 NodeMCU. MPU6050 is a Micro Electro-mechanical system (MEMS), which consists of a three-axis accelerometer and three-axis gyroscope.

3.2.1 MPU6050

A low-power, low-cost 6-axis Motion Tracking chip, which includes a 3-axis gyroscope, 3-axis accelerometer, and a Digital Motion Processor (DMP) all in a small

4mm x 4mm container, is at the heart of the module. The module comes with an on-board LD3985 3.3V regulator, with a 5V logic power supply microcontroller like Arduino. The MPU6050 consumes less than 3.6mA during measurements and only 5µA during idle. This low power consumption allows the implementation of battery-driven devices. In addition, the module has a power LED that lights up when the module is powered [29].

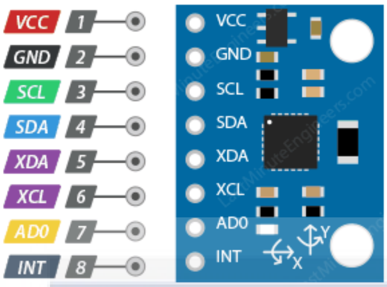


Figure 8: MPU6050 Module Hardware Pinout Overview [30].

3.2.2 ESP8266 NodeMCU

ESP8266 microcontroller may be a small breakout board. Inside, the ESP chips wear a 32-bit RISC CPU that ordinarily works at 80 MHz or 160 MHz and has 64 kB of Slam for enlightening and 96 kB for information. There are 16 general-purpose IO (GPIO) pins designed as input or yield stick and single analogue input with a 10-bit ADC. The controller bolsters I2C, SPI communication as well as RS-232. On the ESP-01, there are, as it were, eight pins uncovered and available [31]. Figure 9 appears the breakout board for ESP8266 NodeMCU.

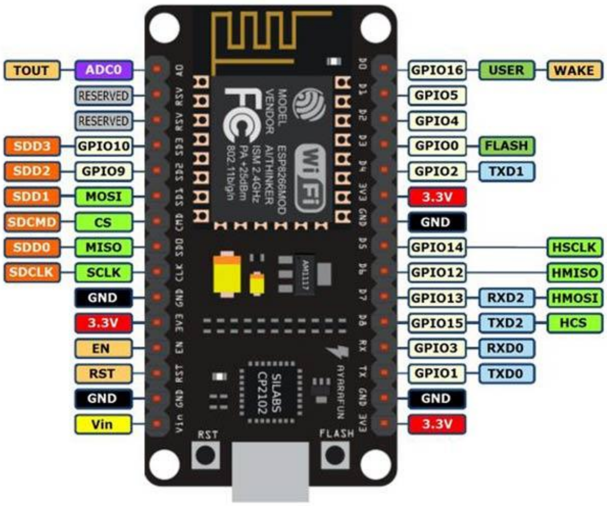


Figure 9: ESP8266 NodeMCU pinout diagram [32].

3.2.3 Full Circuit Design

The circuit design is shown in Fig. 13 MPU-6050 sensor is visible on the right next to the blue ESP8266 board. A power bank 2500mAh fueled the system ostensibly conveys 3.7 V and utilises a micro USB to associate with the ESP8266 NodeMCU. The VCC pin is connected to the 3.3V of the ESP8266 module to power up. Both the grounds of the two devices are connected in common. The SCL pin of MPU6050 is associated with the default SCL pin of ESP8266 NodeMCU.

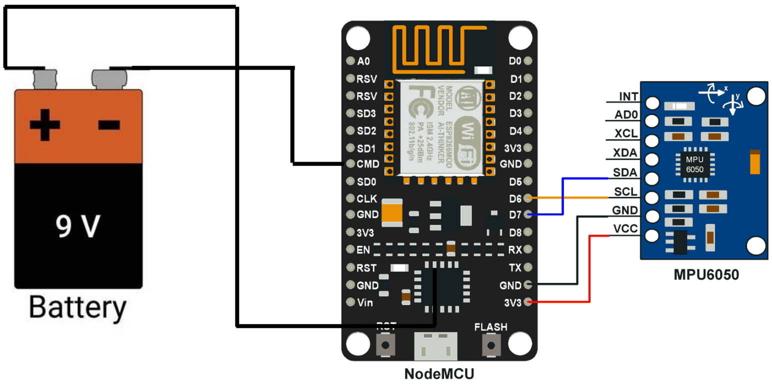


Figure 10. Schematic Design for the project.

3.3 Software Design and Development

This section involves defining the specification software design of this project that is used to achieve the objective. Putty Software linked to Serial Monitor Arduino is used to get the raw data from the MPU6050's gyroscope and accelerometer. The data from Putty Software is then imported into Excel, where it is converted into a graph. The accuracy of the MPU6050's Inertial Measurement Unit will then be conducted concurrently bt using static and dynamic analyses.

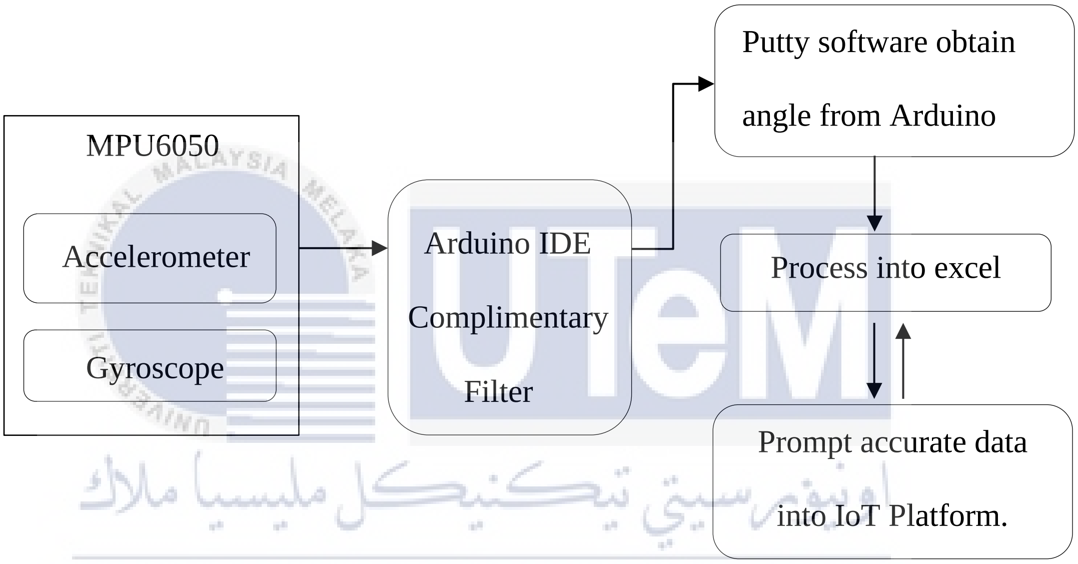


Figure 11: Block diagram of software development.

3.3.1 Arduino IDE

The Arduino Integrated Development Environment - or Arduino Software (IDE) - contains an editor for composing code, a message zone, a content comfort, a toolbar with buttons for typical capacities, and menus arrangement [33]. In addition, this program interfaces with ESP8266 NodeMCU equipment to transfer programs and communicate with them. Thus, Arduino IDE will compute the algorithm for this project from the MPU6050 sensor and display the Inertial Measurement Unit from it.

3.3.2 Putty

PuTTY is a telnet and SSH client developed by Simon Tatham for the Windows platform. PuTTY is free, open-source software with source code that is developed and maintained by a group of volunteers [34]. In this project, putty is used to transfer data from serial monitor of Arduino IDE into Excel.

3.3.3 IoT Platform

The value of the 3-axis of gyroscope and accelerometer of MPU6050 is shown graphically and connected to IoT. This will help to analyse the variation in the acquisition of data and to act accordingly. The data obtained can be seen globally. The IoT platform that is chosen for this project is Blynk. It is designed for the Internet of Things. It can control hardware remotely, display sensor data, store data and visualise it [35]. Blynk is a Platform with IOS and Android apps to control the microcontroller used in this project.

3.4 Data Collection

The quantitative research method in this project is the output value from the sensor. To collect quantitative data for this study, Putty software is used to store the reading of the MPU6050 sensor that has been calibrated and then convert it into a graph to an adequate understanding of the result. Step on how to extract data from the MPU6050 sensor is shown in this section.

3.4.1 Calibration

There are numerous ways to extricate valuable information from the MPU6050. One way is to examine the raw sensor information values amid the calibration handle Figure 12 and utilise that information to compute the new orientation.

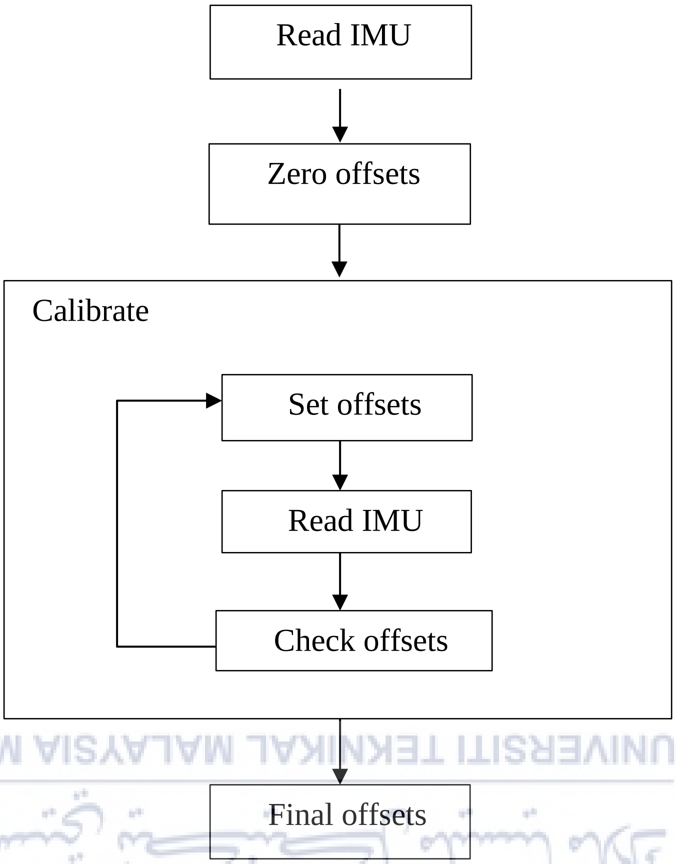


Figure 12. The block diagram for MPU6050 Calibration [36].

The calibration handle begins by focusing on the IMU offsets. Next, an beginning 1000 estimations are studied from the IMU, and the mean values are computed. At that point, drop into the most calibration schedule, which circles until the offsets are inside resilience.

3.4.2 Complimentary Filter

After the Calibration for the raw data from MPU6050 is complete. At this point, a filter is utilised to relieve the vibration impacts that the accelerometer is subjected to and, more vitally, the long-term float impacts of the gyroscope. The requirement for an elective to Kalman filter emerges from the truth that the Kalman filter is exceptionally lumbering, troublesome to get it and challenging to actualise on a littler 8-bit microcontroller [37]. Hence, a complementary filter serves this reason for streamlining the challenges confronted while executing straightforwardly to begin arranging high pass and low pass filter. Hence, the filter's errand assesses a steady point from different sources containing incorrect information and shows clamour with various recurrence substances.

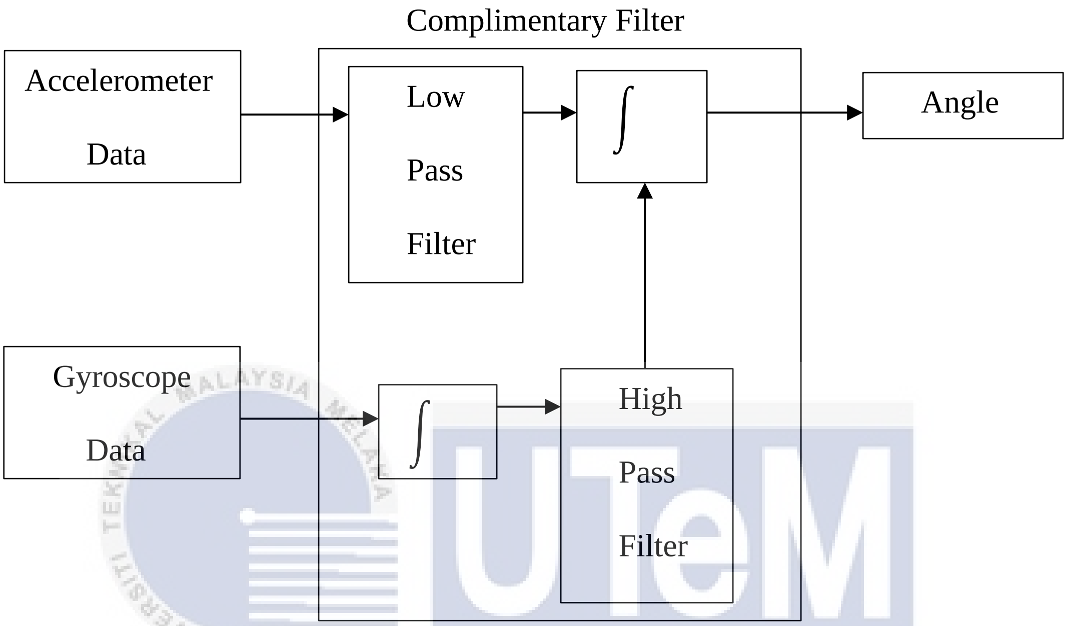


Figure 13. Block diagram of complementary digital filter system for MEMS gyroscope and accelerometer [38].

Within the datasheet of MPU6050, the values of the registers 0x43, 0x45 & 0x47 hold the raw gyro information where the gyroscope yields are at precise speeds. The rates are separated by 131 values when changing from raw data to degree/second. When the gyroscope and accelerometer information is scaled, it is balanced by the equilibrium, then enhanced into the complementary filter. The filter's constant devotion comprises one so that the delivered output is precise and direct measurement. Numerical connection of complementary filter as appeared in (3).

$$Angle = a\times(angle+gyro\times dt) + (a1) \times (accelerometer) \tag{3}$$

3.5 Accuracy Analysis

The utilise of the complementary filter requires the input of the gyroscope’s special rate to be coordinated. This to develop a state of mind point sometimes bolsters into a high pass filter to decrease the impact of inclination float. The genuine issue is based on movement, which can be interpreted explicitly into degrees. In this manner, the IMU-sensor’s information is changed to roll, pitch, and yaw. The experiment aims to attain high accuracy estimation of the situating point by the actualising advanced complementary filter. This to be able to analyse actual uprooting. The practical experiments were categorised into two different sections:

1. Static analysis.
2. Dynamic analysis.

3.5.1 Static analysis

The static investigation was made to confirm the exactness and solidness of degrees (roll, pitch, yaw) from each sensor’s tilt when the IMU-sensor was consistent with no revolutions or increasing physical speed was presented. The estimations can be watched in segment 4.1 (result area). These tests were carried out to ensure that the IMU was stable and precise enough to continue various tests without errors.

3.5.2 Dynamic analysis

The dynamic examination comprised tests where the IMU-sensor was settled on a Tripod that can pivot 360°. The setup of the tests can be seen in Fig. 14.The

Tripod is a point sort of device, and the sensor was settled in a position that it measures yaw, pitch, and roll when revolutions are presented around the fixing hub.

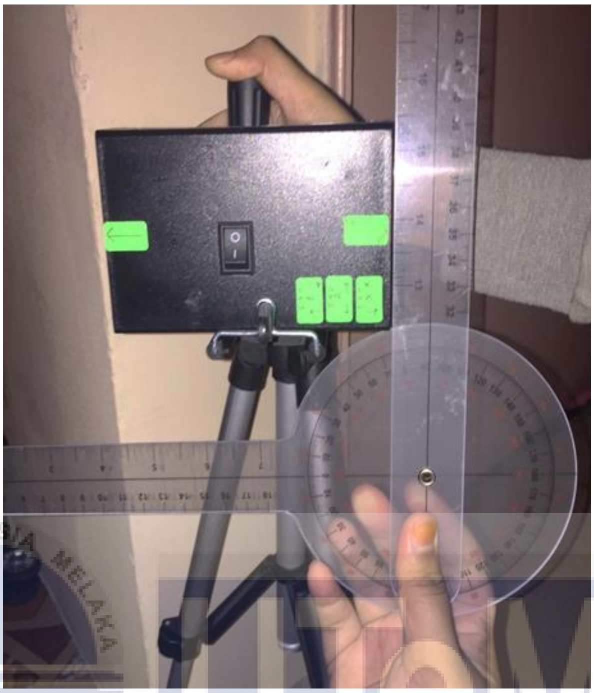


Figure 14. The setup of the experiment using a Goniometer tool.

Tests where the IMU-sensor was settled on a tripod amid a fixing to confirm the accelerometer's precision and gyroscope in their distinctive setups. A test of fixing with a 360° ruler where pitch, roll and yaw was the precise uprooting to get. The result is discussed in section 4.

3.6 Project Implementation

The MPU6050 sensor had a test rate of 100 Hz. The gadget measurements are 58 mm, 31 mm, and had a mass of approximately 22 g. In this consider, the device is mounted on a person to keep an eye on them throughout the operation. As seen in the image, the MPU6050 unit was attached to the chest (Fig. 15).

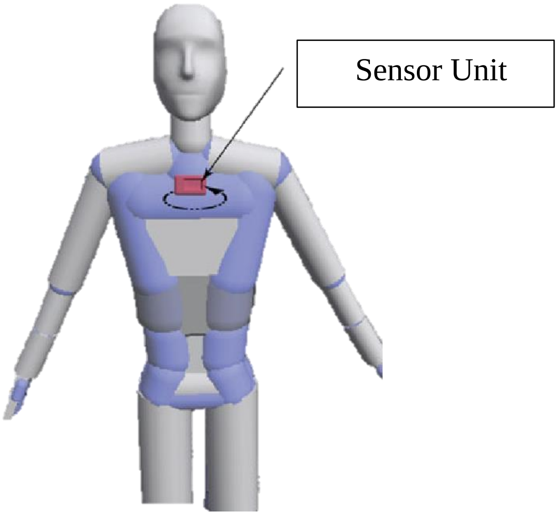


Figure 15. The sensor units are positioned on the chest, with the axis aligned with the vertical line.

It is perfect for selecting markers that are not likely to move from their unique position due to skin development or extension. In this way, the sensor unit is mounted on the chest of the user.

3.7 Summary

To summarise, raw data from the MPU6050's gyroscope and accelerometer must first be retrieved. The gyroscope and accelerometer can measure rotation angles in the experiment described in this report. For the accelerometer point estimation and the real gyroscope rate, a complementary filter was used. The input of the actual gyroscope rate is synchronised to raise a particular point recently, and it is reinforced into a high pass filter to cancel out the effect of float. The correct angle is then be calculated, and the value will be used to analyse the accuracy of the MPU6050's Inertial Measurement Unit utilising static and dynamic analysis.

CHAPTER 4

RESULTS AND DISCUSSION

4.1 Introduction

This section demonstrated the hardware and software findings utilising an ESP8266 NodeMCU connected to a computer and an MPU6050 Inertial Measurement Unit (IMU) sensor. While the hardware is linked to the computer, the Arduino IDE software application is run to perform the Complimentary Filter into the project. The circuit’s connection is also examined. The data from Arduino then transfer into excel sheet by using Putty, hence shows the accurate reading of Yaw, Pitch and Roll from the IMU sensor. The MPU6050 sensor is also connected wirelessly via internet therefore the results obtained can be view using Iot Platform.

4.2 Hardware

Following the successful testing of the circuit on photoboard, a PCB is created. PCBs are widely used in the electronics industry and are also cost-effective. It can collect a number of the most complicated circuits in tiny spaces while avoiding the risk of loose connections. PCB layout were made using software which is ISIS Protues. This is done by converting circuit's schematic diagram shown in Figure 10 into a PCB layout as shown in the Figure below;

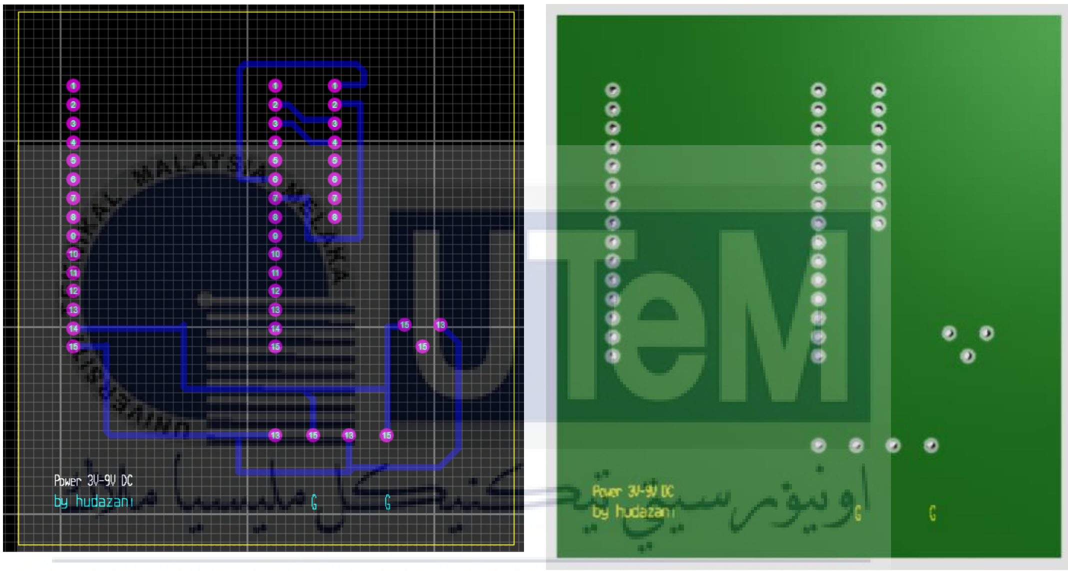


Figure 16: (Left) PCB Layout design by using ISIS Protues Software, (Right) 3D View for the PCB Layout Design.

Component are soldered onto the PCB according to the schematic design connection. The complete circuit after all the component were soldered is shown in the Figure 17. For the finishing, the PCB was then put into a black general purpose box as shown in Figure 18. The enclosure allows the PCB to easily enclose the main board. Good enclosure keeps the PCB safe and to ensure that it does not suffer any mechanical or temperature damage.

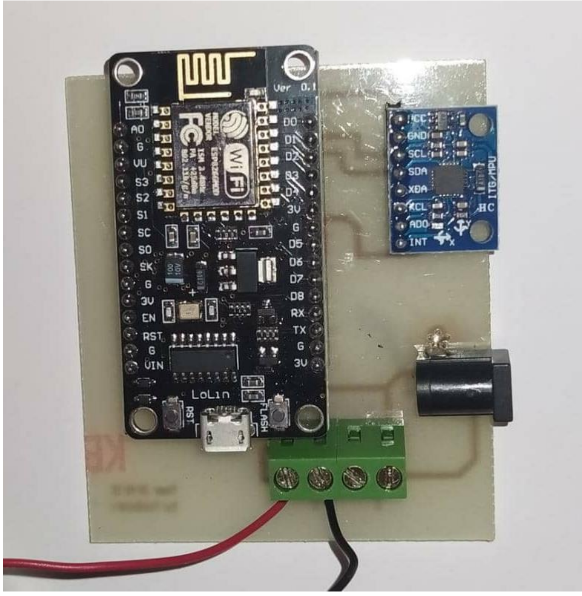


Figure 17: Complete Printed Circuit Board (PCB) of the project.

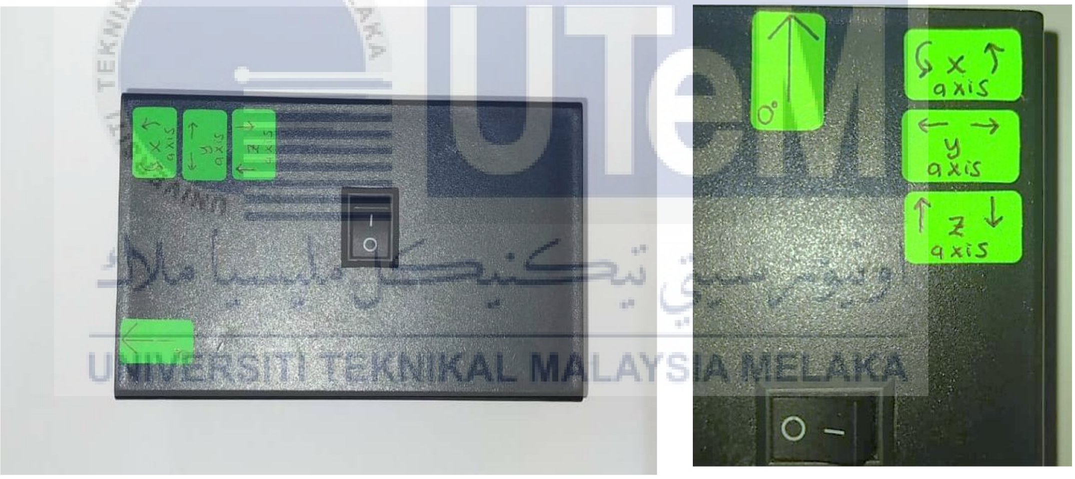
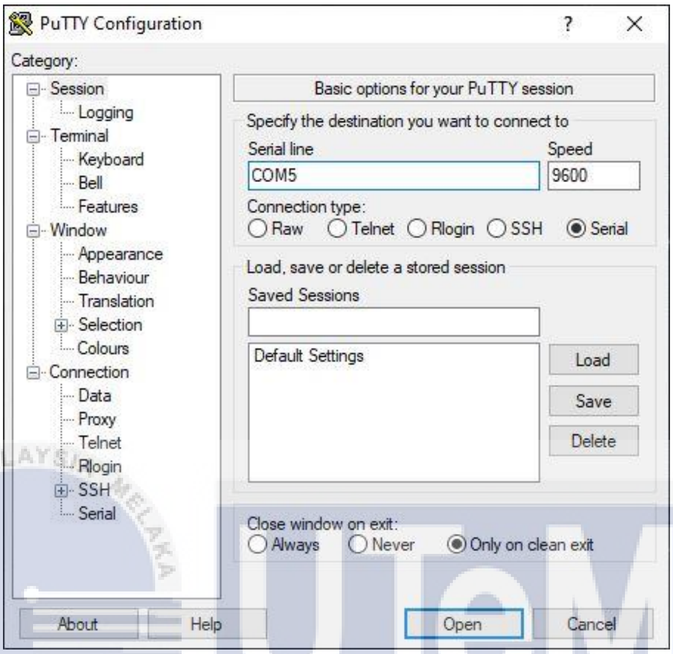


Figure 18: Printed Circuit Board in an enclosure with axis labelling.

4.3 Software

Arduino IDE is used to program the ESP8266 to read the IMU value from MPU6050 sensor. When the IMU starts to move and rotates, it will give a signal and reading to the Arduino. The serial plotter of the Arduino displayed the reading of yaw, pitch and roll. However, Arduino serial plotter cannot record the display results. So, Putty software is used to record and store the data. The imported data which has been save from the computer will be converted into the excel datasheet. Figure 19 shows the configuration of Putty software recording value from serial monitor and Figure 20 shows the value from MPU6050 sensor.



Serial line: COM5

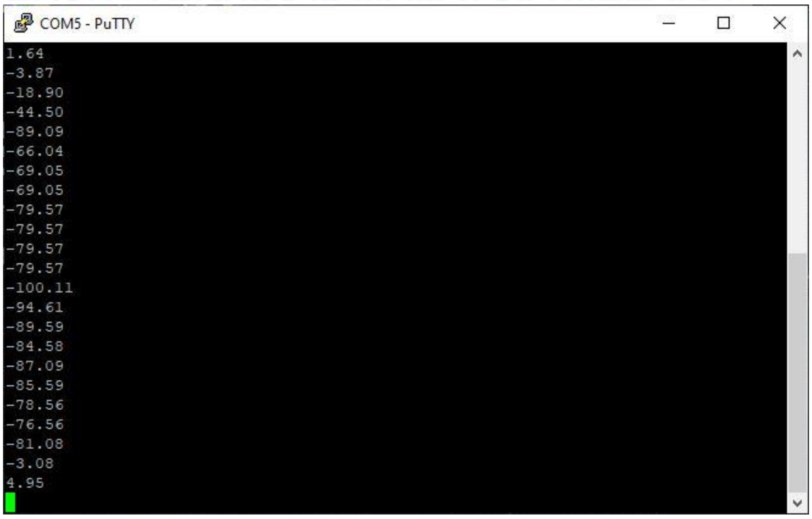
Speed: 9600

Connection type: ○ Raw ○ Telnet ○ Rlogin ○ SSH ● Serial

Default Settings

Close window on exit: ○ Always ○ Never ● Only on clean exit

Figure 19: Configuration Putty software to connect with Serial monitor Arduino to read the output value from MPU6050 sensor.



```
1.64
-3.87
-18.90
-44.50
-89.09
-66.04
-69.05
-69.05
-79.57
-79.57
-79.57
-79.57
-100.11
-94.61
-89.59
-84.58
-87.09
-85.59
-78.56
-76.56
-81.08
-3.08
4.95
```

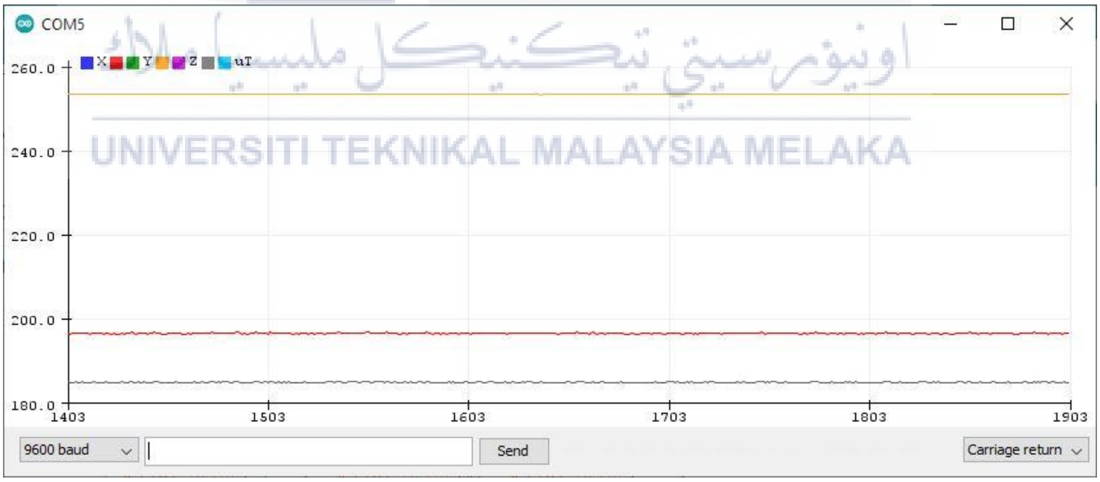
Figure 20: The output value of Y-axis from MPU6050 sensor on Putty serial monitor.

4.3.1 Analysis of Data Accuracy

This part demonstrates the differences between results after and before applying to the Complimentary filter. As mentioned in Chapter 2, applying the raw data onto the Complimentary filter is to improve the accuracy. The result is shown in a table; the data represent the X, Y and Z-axis. It is tabulated accordingly. The value of each axis is acquired by performing the Static and Dynamic Analysis as mentioned in Chapter 3. The data obtained by rotating the IMU in different direction and angle, from 0° until 180° consistently with no revolutions or increasing physical speed, was presented. Goniometer tool is used as the threshold angle. As mentioned before the unit is in degree (°) per second.

a) Result on Serial Plotter of Arduino IDE

This section shows the result obtain straight from the Serial Plotter of Arduino IDE for Static Analysis as explain in Chapter 3.



9600 baud

Carriage return

Figure 21: MPU6050 sensor reading before the complimentary filter is applied on stationary position for 10 seconds.

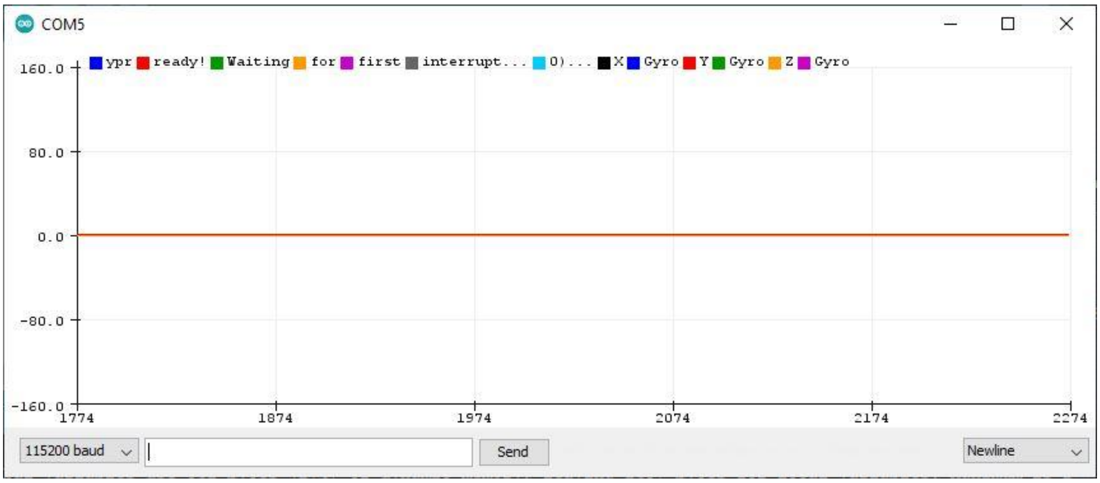


Figure 22: MPU6050 sensor reading after the complimentary filter is applied on stationary position for 10 seconds.

Figure 21 shows that the graph before the complimentary filter is applied contains a value for each axis, even though it should be 0 in degree because it is stationary. In contrast to Figure 22, which displays the accurate value of 0 degrees because no force or rotation is applied. This demonstrates the effectiveness of a complementary filter, which must be used on the MPU6050 sensor prior to any application.

b) Result on datasheet Excel before Complimentary filter is applied.

This section contains the result before the complimentary filter is applied using Dynamic Analysis.

Table 1: Percent of error for the raw output value.

No.	Threshold (◦)	Raw Output Reading (◦)			Percent of Error (%)		
	X, Y, Z	X	Y	Z	X	Y	Z
1	0	1.5	0.16	0	0.15	0.016	0
2	10	2.51	8.85	7.05	0.75	0.115	0.295
3	20	1	14.88	-10.02	0.95	0.256	0.5
4	30	1	16.38	27.55	9.67	0.454	0.08167
5	40	4.51	19.89	31.05	8.87	0.5028	0.22375
6	50	5.01	13.38	46.11	9	0.7324	0.0778

7	60	5.01	19.9	38.6	0.9165	0.6683	0.35667
8	70	5.01	32.43	36.59	0.9284	0.5367	0.47729
9	80	7.02	32.93	34.59	0.9123	0.5884	0.56763
10	90	9.02	59.48	35.09	0.8998	0.3391	0.61011
11	100	9.02	54.96	33.09	0.9098	0.4504	0.6691
12	110	11.52	56.97	31.08	0.8953	0.4821	0.71745
13	120	12.53	51.46	31.08	0.8956	0.5712	0.741
14	130	12.53	63.48	29.08	0.9036	0.5117	0.77631
15	140	12.53	68	27.07	0.9105	0.5143	0.80664
16	150	10.03	62.49	31.07	0.8002	0.5834	0.79287
17	160	10.53	46.96	44.63	0.6908	0.7065	0.72106
18	170	4.52	10.88	46.63	0.9734	0.936	0.72571
19	180	14.06	10.88	26.55	0.9219	0.9396	0.8525
20	-170	14.06	-1.64	122.11	-1.083	-0.99	-1.7183
21	-160	15.56	-3.87	124.62	-1.097	-0.976	-1.7789
22	-150	17.07	-18.9	168.21	-1.114	-0.874	-2.1214
23	-140	20.07	-44.5	-55.98	-1.143	-0.682	-0.6001
24	-130	23.58	-89.09	-51.98	-1.181	-0.315	-0.6002
25	-120	28.09	-66.04	-78.08	-1.234	-0.45	-0.3493
26	-110	25.58	-69.05	-95.62	-1.233	-0.372	-0.1307
27	-100	20.58	-69.05	-97.12	-1.206	-0.31	-0.0288
28	-90	26.1	-79.57	-101.12	-1.29	-0.116	-0.1236
29	-80	26.6	-79.57	-105.14	-1.333	-0.005	-0.3143
30	-70	27.61	-79.57	-107.64	-1.394	-0.137	-0.5377
31	-60	27.61	-79.57	-117.74	-1.46	-0.326	-0.9623
32	-50	29.61	-100.1	-144.74	-1.592	-1.002	-1.8948
33	-40	30.62	-94.61	-150.26	-1.766	-1.365	-2.7565
34	-30	21.08	-89.59	-160.28	-1.703	-1.986	-4.3427
35	-20	23.08	-84.58	-174.84	-2.154	-3.229	-7.742
36	-10	28.1	-87.09	-169.21	-3.81	-7.709	-15.921
37	0	31.1	-18.9	-55.98	-3.11	-18.9	55.98
				Total	12.046	-29.84	24.05

Table 2: Percent of accuracy for each axis.

Axis	Percent of Accuracy
X	11.04%
Y	28%
Z	22%
Total Average	20.35%

Table 1 shows the data obtained solely from the MPU6050 sensor's raw value without the filter. The X, Y, and Z-axis readings straight from the sensor, as well as the percent of inaccuracy and the threshold, are in three columns. When the threshold value is 0°, there isn't much difference; nonetheless, the sensor's output value is ∓1.5. When the value of the threshold grows, the raw and threshold values diverge significantly. The error margin is about 0.99 percent. The threshold value on the highlighted yellow box, for example, is 90 degrees, but the raw value readings are 9.02 degrees for the X-axis, 59.48 degrees for the Y-axis, and 35.09 degrees for the Z-axis, leading to a high percent of error. As shown in Table 2, the total average percent of accuracy for MPU6050 sensor before filtering is just 20.35 percent.

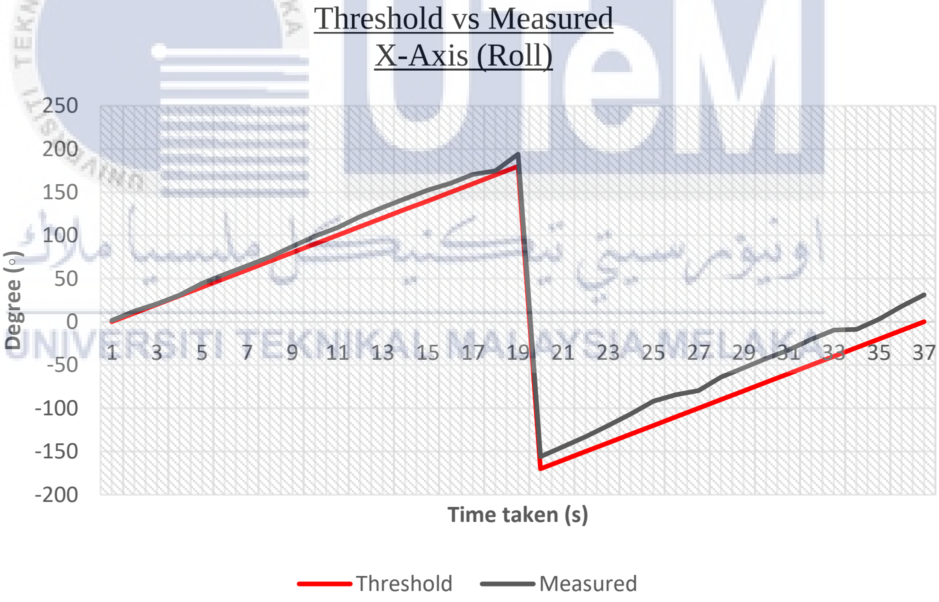


Figure 23. Graph of Threshold vs Measured value for X-axis.

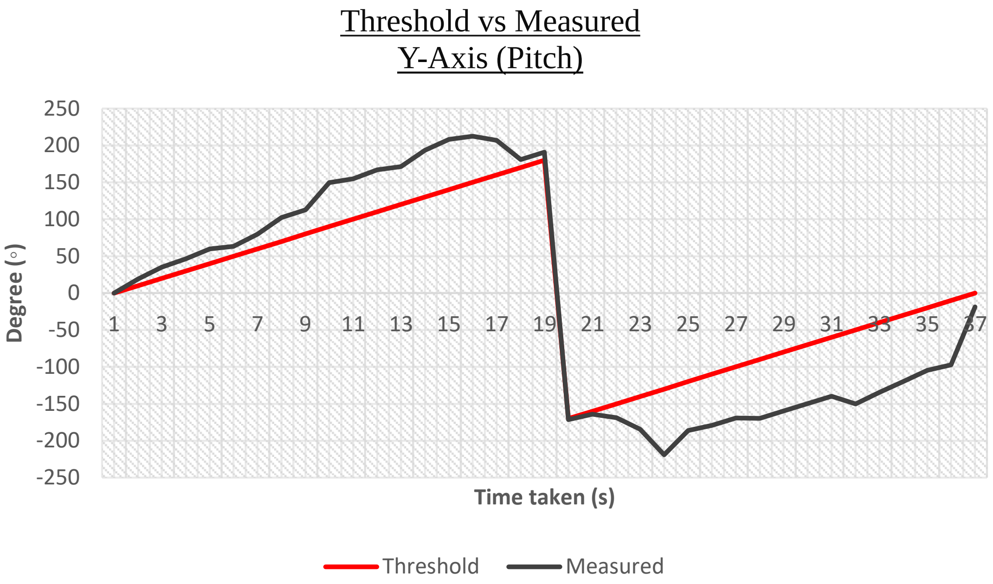


Figure 24. Graph of Threshold vs Measured value for Y-axis.

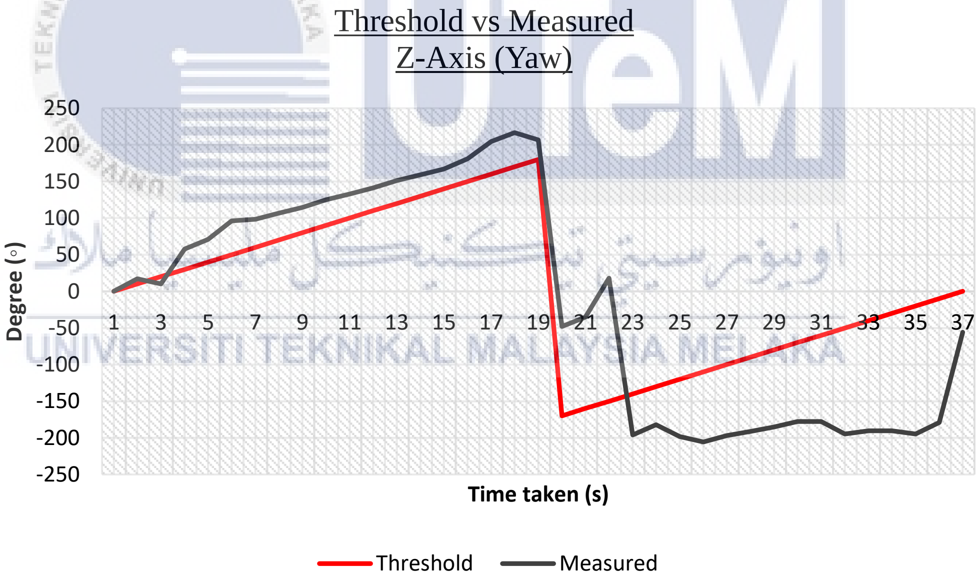


Figure 25. Graph of Threshold vs Measured value for Z-axis.

The plotted graph of the results on the X-axis, Y-axis, and Z-axis is shown in Figures 23, 24, and 25. Once the measurement is obtained, all of the axes' degrees of rotation are plotted on the graph, along with the threshold. With these, it is evident that the threshold and the raw value arriving from the MPU6050 sensor before it is filtered have a noticeable difference graph pattern between the threshold (red line) and the measured value of the MPU6050 sensor (black line).

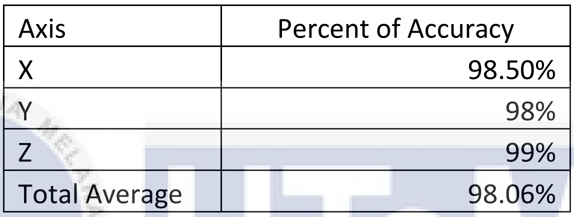
c) Result on datasheet Excel after Complimentary Filter is applied.

Table 3: Value of IMU sensor after Complimentary Filter is applied.

	Threshold (◦)	Filtered Output Reading (◦)			Percent of Error (%)		
No.	X, Y, Z	X	Y	Z	X	Y	Z
1	0	0.796	0.419	0.271	0.0204	0.016	0
2	10	9.231	10.325	11.314	0.0769	0.0325	0.013
3	20	19.25	20.515	20.831	0.0375	0.02575	0.0415
4	30	28.89	30.274	30.304	0.037	0.00913	0.01013
5	40	38.2	39.907	41.726	0.045	0.00233	0.04315
6	50	47.61	50.133	49.847	0.0478	0.00266	0.00306
7	60	57.35	60.872	60.741	0.04417	0.01453	0.01235
8	70	70.13	70.655	70.413	0.00186	0.00936	0.0059
9	80	78.85	75.905	81.308	0.01438	0.05119	0.01635
10	90	92.91	94.503	91.381	0.03233	0.05003	0.01534
11	100	102.36	106.809	102.254	0.0236	0.06809	0.02254
12	110	111.35	109.608	110.502	0.01227	0.00356	0.00456
13	120	123.385	122.304	122.243	0.02821	0.0192	0.01869
14	130	129.581	130.711	132.054	0.00322	0.00547	0.0158
15	140	138.381	145.738	141.097	0.01156	0.04099	0.00784
16	150	148.857	153.158	151.07	0.00762	0.02105	0.00713
17	160	158.164	162.246	160.63	0.01148	0.01404	0.00394
18	170	171.955	171.779	168.63	0.0115	0.01046	0.00806
19	180	179.363	179.838	180.961	0.00354	0.0009	0.00534
20	-170	-165.88	-170.32	-169.21	-0.0243	-0.0019	-0.0046
21	-160	-162.27	-160.29	-162.62	-0.0142	-0.0018	-0.0164
22	-150	-154.85	-150.28	-148.21	-0.0323	-0.0018	-0.0119
23	-140	-142.46	-140.52	-140.98	-0.0176	-0.0037	-0.007
24	-130	-134.92	-128.7	-131.14	-0.0378	-0.01	-0.0088
25	-120	-117.02	-120.26	-120.66	-0.0248	-0.0022	-0.0055
26	-110	-109.77	-112.58	-112.62	-0.0021	-0.0235	-0.0238
27	-100	-99.156	-105.77	-99.12	-0.0084	-0.0577	-0.0088
28	-90	-91.239	-93.91	-91.966	-0.0138	-0.0434	-0.0218

29	-80	-82.364	-79.339	-78.533	-0.0296	-0.0083	-0.0183
30	-70	-70.857	-69.775	-69.879	-0.0122	-0.0032	-0.0017
31	-60	-61.689	-58.506	-61.826	-0.0282	-0.0249	-0.0304
32	-50	-50.712	-48.351	-50.433	-0.0142	-0.033	-0.0087
33	-40	-42.617	-40.965	-41.101	-0.0654	-0.0241	-0.0275
34	-30	-32.76	-31.615	-30.123	-0.092	-0.0538	-0.0041
35	-20	-20.45	-20.565	-19.879	-0.0225	-0.0283	-0.006
36	-10	-10.133	-10.535	-10.141	-0.0133	-0.0535	-0.0141
37	0	-5.248	-0.395	-0.129	-0.0526	-0.039	0.0012
				Total	-0.035	-0.0168	0.03627

Table 4: Percent of Accuracy for each axis**.**



Axis	Percent of Accuracy
X	98.50%
Y	98%
Z	99%
Total Average	98.06%

Table 4 shows the data obtained after the complimentary filter is applied on the MPU6050 sensor's. The X, Y, and Z-axis readings from the sensor and the percent of inaccuracy and the threshold are in those three columns. When the threshold value is 0°, there isn't much difference; nonetheless, the sensor's output value is ∓0.7. When the value of the threshold grows, the raw and threshold values did not diverge much as before. In fact, the error margin is less than 0.03 percent. The threshold value on the highlighted yellow box, for example, is 90 degrees, and the raw value readings are 92.91 degrees for the X-axis, 94.503 degrees for the Y-axis, and 91.38 degrees for the Z-axis, leading to a low percent of error. As shown in Table 5, the total average percent accuracy for the MPU6050 sensor after filtering is 98.06 percent.

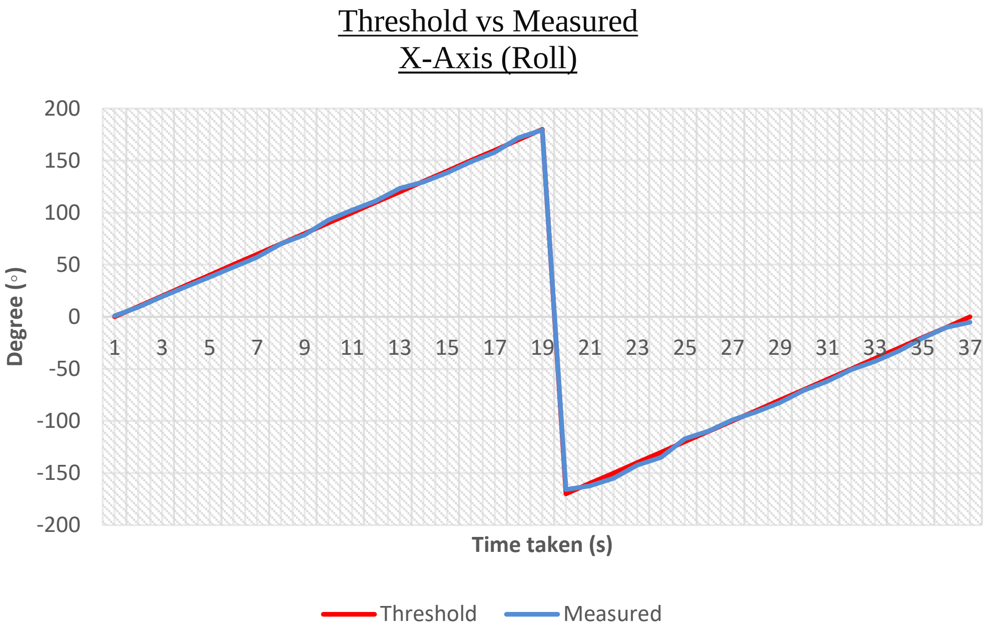


Figure 26. Graph of Threshold vs Measured value for X-axis.

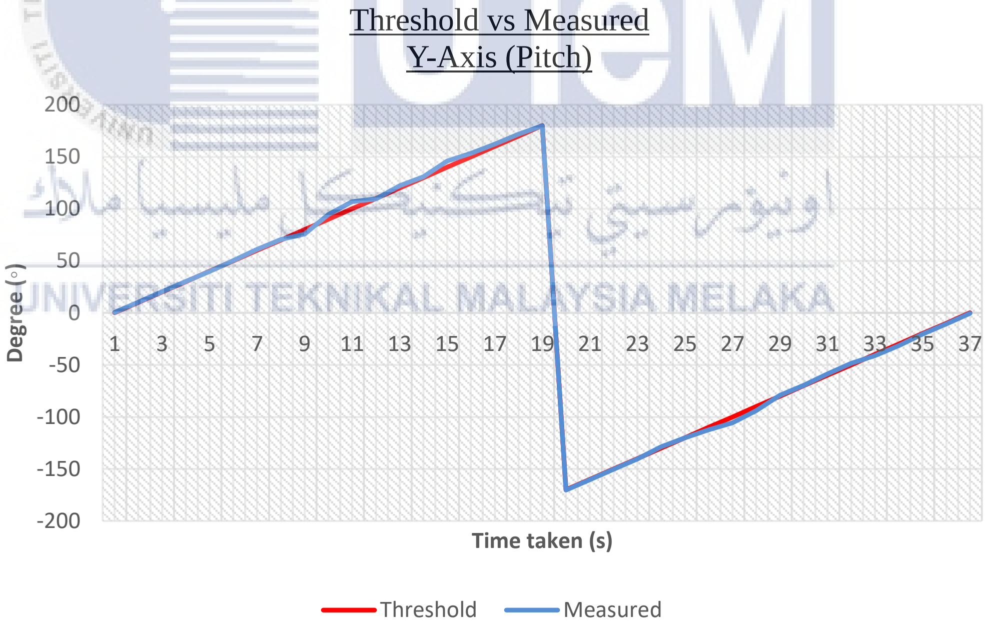


Figure 27. Graph of Threshold vs Measured value for Y-axis.

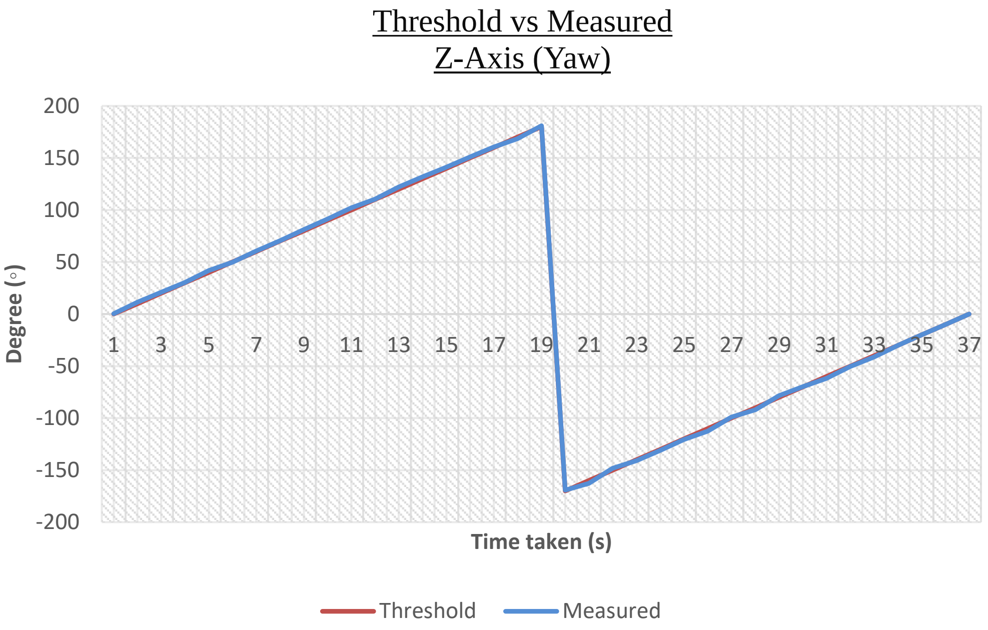


Figure 28. Graph of Threshold vs Measured value for Z-axis.

Figure 26, 27 and 28 show the plotted graph of the results on excel on the X-axis, Y-axis, and Z-axis. The red line is for the threshold; meanwhile, the blue line reads the measurement from MPU6050 sensor. The graph shows the value of MPU6050 after adding a complementary filter. Those three graphs indicate that the pattern of the X, Y and Z-axis results is remarkably similar. Also, it was almost identical to the threshold. Thus, the graph's pattern was virtually the same, but the values attained were not much different.

d) Comparison before and after Complimentary Filter

Table 5: Reading of MPU6050 sensor Raw vs Filtered output.

No.	Threshold (◦)	Raw Output Reading (◦)			Filtered Output Reading (◦)		
	X, Y, Z	X	Y	Z	X	Y	Z
1	0	1.5	0.16	0	0.796	0.419	0.271
2	10	2.51	8.85	7.05	9.231	10.325	11.314
3	20	1	14.88	-10.02	19.25	20.515	20.831

4	30	1	16.38	27.55	28.89	30.274	30.304
5	40	4.51	19.89	31.05	38.2	39.907	41.726
6	50	5.01	13.38	46.11	47.61	50.133	49.847
7	60	5.01	19.9	38.6	57.35	60.872	60.741
8	70	5.01	32.43	36.59	70.13	70.655	70.413
9	80	7.02	32.93	34.59	78.85	78.905	81.308
10	90	9.02	59.48	35.09	92.91	90.503	91.381
11	100	9.02	54.96	33.09	102.36	100.809	102.254
12	110	11.52	56.97	31.08	111.35	109.608	110.502
13	120	12.53	51.46	31.08	123.39	122.304	122.243
14	130	12.53	63.48	29.08	129.58	130.711	132.054
15	140	12.53	68	27.07	138.38	142.738	141.097
16	150	10.03	62.49	31.07	148.86	150.158	151.07
17	160	10.53	46.96	44.63	158.16	162.246	160.63
18	170	4.52	10.88	46.63	171.96	171.779	168.63
19	180	14.06	10.88	26.55	179.36	179.838	180.961
∑	1710	138.9	644.36	546.9	1706.6	1722.7	1727.58

The contrast between before and after the complementing filter is applied shown in Table 4. The total value for the threshold is 1710, as indicated by the green coloured box. This means that the lower the gap between the measurement value and the threshold, the higher the accuracy. The total value obtained by the raw value for each axis in the orange colour box, for example, is 138.9 for the X-axis, 644.36 for the Y-axis, and 546.9 for the Z-axis. There is a substantial gap between the raw data and the threshold value of 1710. This demonstrates that the raw values have a deficient level of precision.

Meanwhile, the filtered value, represented by the blue coloured box, differs just slightly from the threshold value, ranging from 1705 to 1720. This implies that the filtered value has a high degree of accuracy. Take a look at the graph pattern in Figure 27 before and after the complimentary filter, using the X-axis as an example.

As a result, the graph's layout between filtered and threshold was nearly identical, but the graph's raw value pattern has a big difference.

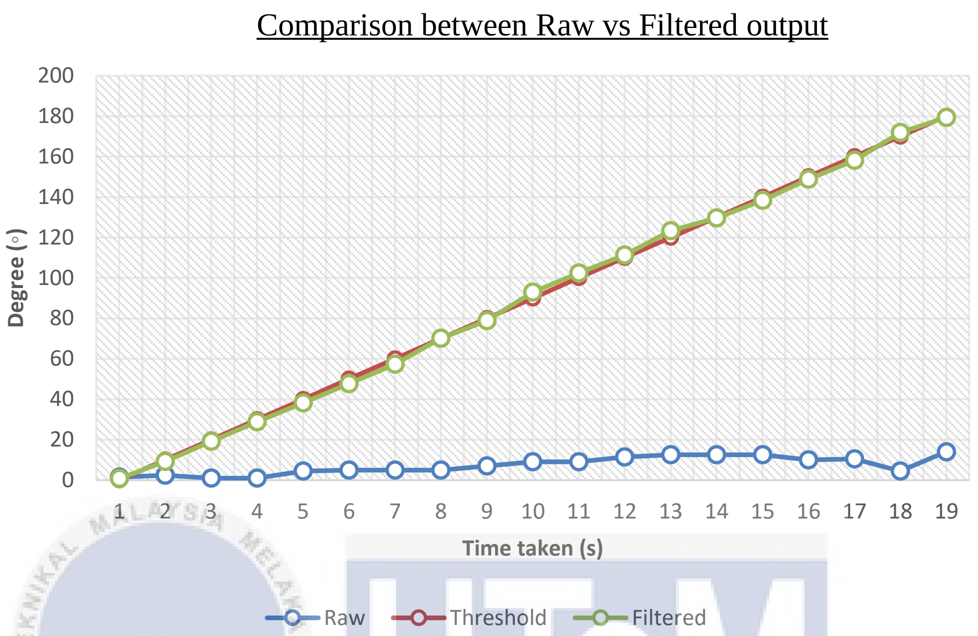
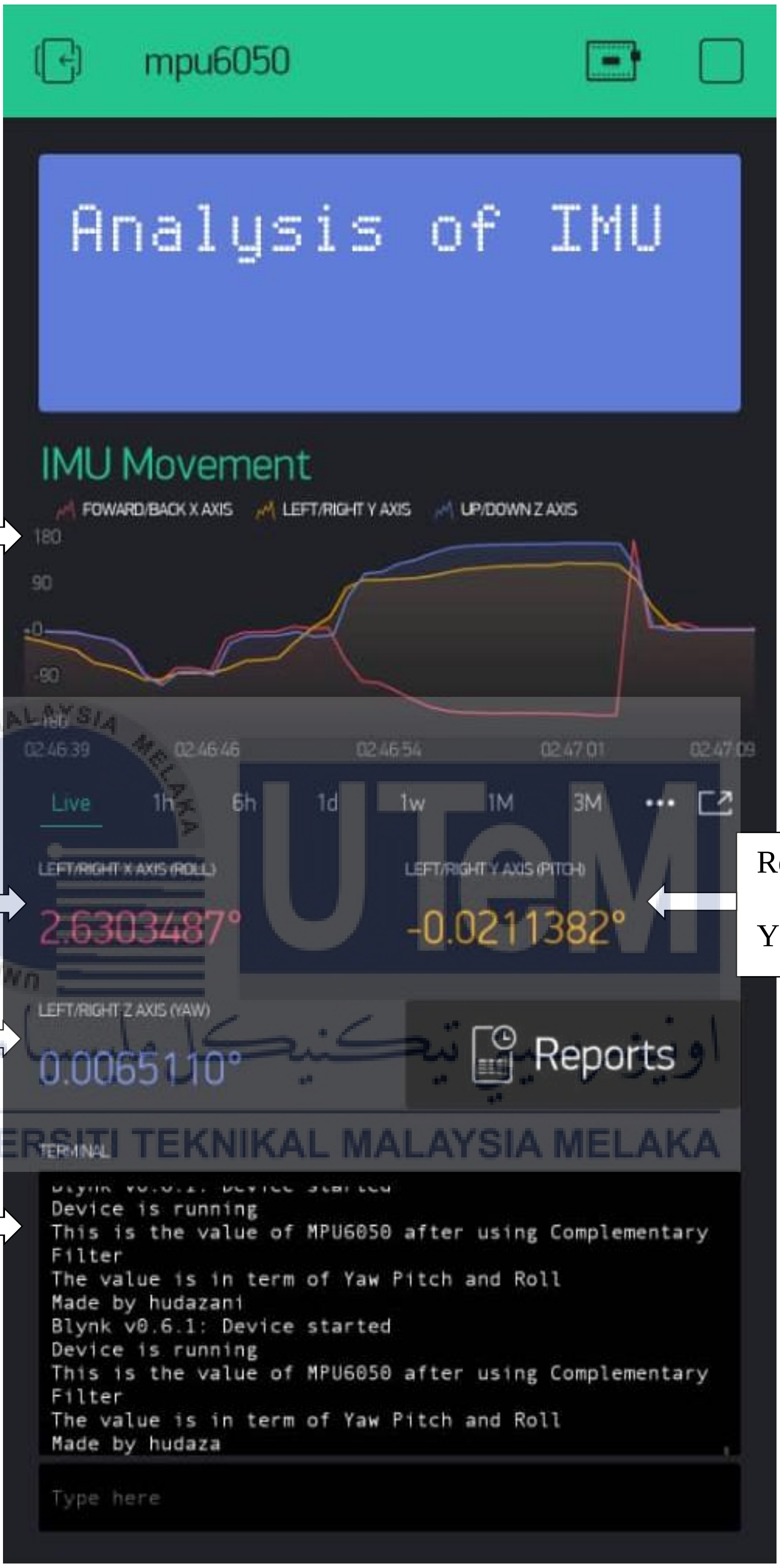


Figure 29: Graph of comparison between before and after the complimentary filter is applied.

e) IoT Platform

To complete the project's final objective, which is to display the results of the MPU6050 sensor's accuracy using the IoT Platform. Blynk application is used to visualise the accurate data of the MPU6050 sensor in degree. The Blynk application interface after connecting to the MPU6050 sensor via the internet is shown in Figure 30. The Blynk app displays the value for the X-axis, which is red coloured text, the Y-axis, which is yellow coloured text, and the Z-axis, which is blue coloured text. This allows any user to read the axis value using only their smartphone.



LEFT/RIGHT X AXIS (ROLL) 2.6303487°

LEFT/RIGHT Y AXIS (PITCH) -0.0211382°

LEFT/RIGHT Z AXIS (YAW) 0.0065110°

```
Blynk v0.6.1: Device started
Device is running
This is the value of MPU6050 after using Complementary
Filter
The value is in term of Yaw Pitch and Roll
Made by hudazani
Blynk v0.6.1: Device started
Device is running
This is the value of MPU6050 after using Complementary
Filter
The value is in term of Yaw Pitch and Roll
Made by hudaza
```

Type here

Graph output of X,Y and Z-axis

Reading for X-axis (Roll)

Reading for Y-axis (Pitch)

Reading for Z-axis (Yaw)

Terminal

Figure 30. Graph reading after Complimentary Filtered is applied on the IoT platform (Blynk).

4.4 Limitation

MPU6050 is a simple sensor in dealing with it and extracting accurate data with the right complementary filter. However, if the sensor is utilised in an uncertain environment and the shortcomings are not taken into consideration, the sensor may be unreliable. It must be determined whether heat affect the angular rate sensitivity or the zero-rate level on the gyroscope. The vibration created by the surface itself with various tightening methods also needs to be carefully examined. For example, the sensor may be placed on a rubber cushion to reduce vibration as much as necessary. If the sensor is positioned in an inconvenient area, as indicated above, the project's long-term sustainability may be affected.

4.5 Summary

It can be concluded and verified that the complementary filter can obtain a smoother line of a graph from the MPU6050 sensor. The angle produced by using the complementary filter is more accurate and exact, with no reduction in mistakes. Furthermore, when the MPU6050 moves on the axis applied, there are no vibrations on the axes. In summary, after the filter coefficient is adequately calibrated, this complementary filter is productive, constant, and reasonable in IMU application.

CHAPTER 5

CONCLUSION AND FUTURE WORKS

5.1 Introduction

This chapter explains the conclusion based on the analysis of the results. The conclusion is presented as the summarisation obtained throughout this project. Besides that, several ideas are also proposed as the future development for this project.

5.2 Conclusion

Finally, the project's objective of analysing the accuracy of the MPU6050 sensor's inertial measurement and improving the achievable accuracy rate to 98 percent utilising complementary filters was met. The concept was examined with the help of NodeMCU-based IMU. Static and dynamic analyses have been used to test it. The negative numbers indicate that the angle's direction was different (anticlockwise). The acceleration of a moving or vibrating object was recorded by an accelerometer, while a gyroscope monitored the angular rate or orientation. Because the acceleration values are not constant, the graph for raw value differed significantly from the graph for threshold. A complementary filter was created by combining a high pass filter and a low pass filter. As a result, the gyroscope data should be gathered and merged to build a complementary filter. Now the signal data is smooth and accurate. In brief, all of the objectives were fulfilled, including analysing the MPU6050 sensor's inertial measurement accuracy and increasing the achievable accuracy rate to 98 percent by applying complementary filters. Last but not least, this project's final aim of prompting accuracy by visualising the results utilising IoT Platform has been completed.

5.3 Future Work

If the IMU is to be installed within the tool in the future, it must be determined whether heat affect the angular rate sensitivity or the zero-rate level on the gyroscope. The vibration created by the tool itself with various tightening methods also needs to be carefully examined. For example, the sensor may be placed on a rubber cushion to reduce vibration as much as necessary. Further examination of the qualities and settings modified in software for the gyroscope can increase the accuracy and readings for the specific actual angle application.

REFERENCES

[1] J. Svacha, J. Paulos, G. Loianno and V. Kumar, "IMU-Based Inertia Estimation for a Quadrotor Using Newton-Euler Dynamics," in IEEE Robotics and Automation Letters, vol. 5, no. 3, pp. 3861-3867, July 2020.

[2] H. Benzerrouk and A. V. Nebylov, "Robust IMU/UWB integration for indoor pedestrian navigation," 2018 25th Saint Petersburg International Conference on Integrated Navigation Systems (ICINS), 2018, pp. 1-5.

[3] K. Ogata, H. Tanaka and Y. Matsumoto, "A Robust Position and Posture Measurement System Using Visual Markers and an Inertia Measurement Unit," 2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), 2019, pp. 7497-7502.

[4] M. Meghji et al., "An Algorithm for the Automatic Detection and Quantification of Athletes’ Change of Direction Incidents Using IMU Sensor Data," in IEEE Sensors Journal, vol. 19, no. 12, pp. 4518-4527, 15 June15, 2019.

[5] Y. Wang and A. M. Shkel, "A Review on ZUPT-Aided Pedestrian Inertial Navigation," 2020 27th Saint Petersburg International Conference on Integrated Navigation Systems (ICINS), 2020, pp. 1-4.

[6] R. S. Kulikov, "Integrated UWB/IMU system for high rate indoor navigation with cm-level accuracy," 2018 Moscow Workshop on Electronic and Networking Technologies (MWENT), 2018, pp. 1-4, doi: 10.1109/MWENT.2018.8337235.

[7] H. Xing, Z. Chen, C. Wang, M. Guo and R. Zhang, "Quaternion-based Complementary Filter for Aiding in the Self-Alignment of the MEMS IMU," 2019 IEEE International Symposium on Inertial Sensors and Systems (INERTIAL), 2019, pp. 1-4.

[8] Z. Zhe, W. Jian-bin, S. Bo and T. Guo-feng, "Adaptive Complementary Filtering Algorithm for IMU Based on MEMS," 2020 Chinese Control And Decision Conference (CCDC), 2020, pp. 5409-5416.

[9] S. Sheikhpour, M. M. Atia and S. Waslander, "A Flexible Simulation and Design Environment for IMU/GNSS Sensors Integration," 2018 IEEE 61st International Midwest Symposium on Circuits and Systems (MWSCAS), 2018, pp. 472-475.

[10] Marco Iosa, Pietro Picerno, Stefano Paolucci & Giovanni Morone (2016) Wearable inertial sensors for human movement analysis, Expert Review of Medical Devices, 13:7, 641-659.

[11] N. Barbour and G. Schmidt. Inertial sensor technology trends. IEEE Sensors Journal, 1(4):332{339, 2001}.

[12] M. Qadri, M. S. Hussain, S. Jawed and S. A. Iftikhar, "Virtual Tourism Using Samsung Gear VR Headset," 2019 International Conference on Information

Science and Communication Technology (ICISCT), 2019, pp. 1-10.

[13] Steven E. Jones; George K. Thiruvathukal, "Core Controller: The Wii Remote," in Codename Revolution: The Nintendo Wii Platform , MIT Press, 2012, pp.53-77.

[14] M. Perlmutter and L. Robin. High-performance, low cost inertial MEMS: A market in motion! In Proceedings of the IEEE/ION Position Location and Navigation Symposium (PLANS), pages 225{229, Myrtle Beach, South Carolina, USA, 2012.

[15] C. Kwak and I. V. Bajić, "Online MoCap Data Coding With Bit Allocation, Rate Control, and Motion-Adaptive Post-Processing," in IEEE Transactions on Multimedia, vol. 19, no. 6, pp. 1127-1141, June 2017.

[16] P. Promrit, S. Chokchaitam and M. Ikura, "In-Vehicle MEMS IMU Calibration Using Accelerometer," 2018 IEEE 5th International Conference on Smart Instrumentation, Measurement and Application (ICSIMA), 2018, pp. 1-3.

[17] J. Demkowicz, "MEMS Gyro in the Context of Inertial Positionning," 2017 Baltic Geodetic Congress (BGC Geomatics), 2017, pp. 247-251, doi: 10.1109/BGC.Geomatics.2017.40.

[18] Tereshkov, Vasiliy. (2012). An Intuitive Approach to Inertial Sensor Bias Estimation. International Journal of Navigation and Observation. 2013. 10.1155/2013/762758.

[19] N. Kuxdorf-Alkirata, F. Kasolis, D. Brückmann and O. Koch, "Linear error modeling and noise smoothing for improved low-cost IMU-based indoor positioning," 2019 IEEE 62nd International Midwest Symposium on Circuits and Systems (MWSCAS), 2019, pp. 1069-1072.

[20] R. I. Sokolov, "Theoretical investigation of Gaussian and non-Gaussian noise masking properties," 2016 2nd International Conference on Industrial Engineering, Applications and Manufacturing (ICIEAM), 2016, pp. 1-4.

[21] D. T. Thinh, N. B. H. Quan and N. Maneetien, "Implementation of Moving Average Filter on STM32F4 for Vibration Sensor Application," 2018 4th International Conference on Green Technology and Sustainable Development (GTSD), 2018, pp. 627-631.

[22] R. Gonzalez and C. A. Catania, "A statistical approach for optimal order adjustment of a moving average filter," 2018 IEEE/ION Position, Location and Navigation Symposium (PLANS), 2018, pp. 1542-1546.

[23] H. Fourati, N. Manamanni, L. Afilal and Y. Handrich, "Position estimation approach by Complementary Filter-aided IMU for indoor environment," 2013 European Control Conference (ECC), 2013, pp. 4208-4213.

[24] A. R. P. Andriën, D. Antunes, M. J. G. v. d. Molengraft and W. P. M. H. Heemels, "Similarity-Based Adaptive Complementary Filter for IMU Fusion," 2018 European Control Conference (ECC), 2018, pp. 3044-3049.

[25] D. Feng, C. Wang, C. He, Y. Zhuang and X. Xia, "Kalman-Filter-Based Integration of IMU and UWB for High-Accuracy Indoor Positioning and

Navigation," in IEEE Internet of Things Journal, vol. 7, no. 4, pp. 3133-3146, April 2020.

[26] S. Habbachi, M. Sayadi, F. Fnaiech, N. Rezzoug, P. Gorce and M. Benbouzid, "Estimation of IMU orientation using linear Kalman Filter based on correntropy criterion," 2018 IEEE International Conference on Industrial Technology (ICIT), 2018, pp. 1340-1344.

[27] J. L. Coyte, D. Stirling, M. Ros, H. Du and A. Gray, "Displacement profile estimation using low cost inertial motion sensors with applications to sporting and rehabilitation exercises," 2013 IEEE/ASME International Conference on Advanced Intelligent Mechatronics, 2013, pp. 1290-1295.

[28] J. L. Coyte, D. Stirling, M. Ros, H. Du and A. Gray, "Displacement profile estimation using low cost inertial motion sensors with applications to sporting and rehabilitation exercises," 2013 IEEE/ASME International Conference on Advanced Intelligent Mechatronics, 2013, pp. 1260-1265.

[29] T. Gujarathi and K. Bhole, "GAIT ANALYSIS USING IMU SENSOR," 2019 10th International Conference on Computing, Communication and Networking Technologies (ICCCNT), 2019, pp. 1-5.

[30] InvenSense Inc, MPU-6050-Datasheet1 Document Number: PS-MPU6050A-00 Revision: 3.4 Release Date: 08/19/2013

[31] S. Barai, D. Biswas and B. Sau, "Estimate distance measurement using NodeMCU ESP8266 based on RSSI technique," 2017 IEEE Conference on Antenna Measurements & Applications (CAMA), 2017, pp. 170-173.

[32] Espressif Systems, ESP8266EX Datasheet.

[33] D. K. Halim, T. C. Ming, N. M. Song and D. Hartono, "Arduino-based IDE for Embedded Multi-processor System-on-Chip," 2019 5th International Conference on New Media Studies (CONMEDIA), 2019, pp. 135-138.

[34] Jayanth S, Poorvi MB and Sunil MP, "Raspberry Pi based energy management system," 2016 Online International Conference on Green Engineering and Technologies (IC-GET), 2016, pp. 1-5.

[35] A. Karnik, D. Adke and P. Sathe, "Low-Cost Compact Theft-Detection System using MPU-6050 and Blynk IoT Platform," 2020 IEEE Bombay Section Signature Conference (IBSSC), 2020, pp. 113-118.

[36] A. Mikov, S. Reginya and A. Moschevikin, "In-situ Gyroscope Calibration Based on Accelerometer Data," 2020 27th Saint Petersburg International Conference on Integrated Navigation Systems (ICINS), 2020, pp. 1-5.

[37] P. Gui, L. Tang and S. Mukhopadhyay, "MEMS based IMU for tilting measurement: Comparison of complementary and kalman filter based data fusion," 2015 IEEE 10th Conference on Industrial Electronics and Applications (ICIEA), 2015, pp. 2004-2009.

[38] Y. Pititeeraphab, T. Jusing, P. Chotikunnan, N. Thongpance, W. Lekdee and A. Teerasoradech, "The effect of average filter for complementary filter and Kalman filter based on measurement angle," 2016 9th Biomedical Engineering International Conference (BMEiCON), 2016, pp. 1-4.